Rise of the Indoor Crowd: Reconstruction of Building Interior View via Mobile Crowdsourcing

Si Chen†, Muyuan Li†, Kui Ren†, Xinwen Fu‡, Chunming Qiao†
†Department of Computer Science and Engineering, SUNY at Buffalo
‡Department of Computer Science, University of Massachusetts Lowell
{schen23, muyuanli, kuiren, qiao}@buffalo.edu, xinwenfu@cs.uml.edu

ABSTRACT

Crowdsourcing is a technology with the potential to revolutionize large-scale data gathering in an extremely cost-effective manner. It provides an unprecedented means of collecting data from the physical world, particularly through the use of modern smartphones, which are equipped with high-resolution cameras and various micro-electrical sensors. In this paper, we address the critical task of reconstructing the indoor interior view of a building from crowdsourced data. We propose, design, and prototype IndoorCrowd2D, a smartphone-empowered crowdsourcing system for indoor scene reconstruction. We first formulate the problem via trackable models and then employ a divide and conquer approach to address the inherently incomplete, opportunistic, and noisy crowdsourced data. By utilizing the image information and sensory data in a coordinated way, our system demonstrates high result-accuracy, as well as allows a gradual build-up procedure of the hallway skeleton. Our evaluation result shows that IndoorCrowd2D achieves a precision around 85%, a 100% recall and a F-score around 95% for reconstructing college buildings from 1,151 datasets uploaded by 25 users. This reveals that our image and sensor hybrid method is more robust to overcome errors and outliers as compared to image-only method.

Categories and Subject Descriptors

H.3.4 [[**Information Storage and Retrieval**]: Systems and Software

General Terms

Algorithms; Design; Experimentation; Performance

Keywords

Crowdsourcing; indoor scene; panorama; multi-dimensional sensing

SenSys'15, November 1–4, 2015, Seoul, South Korea..

DOI: http://dx.doi.org/10.1145/2809695.2809702.

1. INTRODUCTION

The emerging technology has witnessed the birth of virtual tour applications that connect the physical and cyber world by enabling an immersive visual experience. The industrial state-of-the-art Google Street View [20] project provides 360 degree panoramic views along many public streets in the world, covering about 5 million miles of roads and more than 3,000 cities. However, unlike outdoor environment, only 91 indoor street view transit locations around the world are currently available from Google Map. The major obstacle to ubiquitous coverage is the complexity of the indoor environment [24]. As a result, existing outdoor street-view reconstruction techniques either cannot be directly applied to an indoor environment or become very costly.

Currently, indoor visualization has been studied by robotic and computer vision communities. One of the state-of-the-art solution is named simultaneous localization and mapping (SLAM) [13, 24, 44]. In [24], the authors present a SLAM-based human-operated backpack system which equipped with 2D laser scanners and inertial measurement units (IMU) to automatically reconstruct building interior view. Xiao and Furukawa [44] reconstruct the structure and interior view of world museums by jointly using several advanced computer vision algorithms and large-scale laser scanning 3D points. However, most of current SLAM-based indoor scene reconstruction techniques require specialize equipment to capture indoor scene and cannot be scaled well. Alternatively, some researches focus on utilizing image-based pose estimation techniques such as Structure from Motion (SfM) [1,36,37,40] and Multiview-stereo [17, 18] to process image collections and perceive spatial relationships of these images. However, indoor scenes are usually full of structurally similar and textureless objects, which violates assumptions of vision-based algorithms. Hence, these vision based solutions are usually unable to provide accurate spatial structure results at a large scale [14].

In this paper, we propose IndoorCrowd2D, a smartphone empowered system utilizing the power of the crowd for helping us reconstruct the building interior views at large scale and with low cost. Our system is able to provide immersive panoramic image viewing experience for building interior views with an effective scene navigation mechanism. It breaks away from established approaches to reconstruct indoor scenes, and explores an alternative architecture based on crowdsourcing and mobile-sensing. IndoorCrowd2D is expected to extend existing online map service to indoor environment at a large scale. Moreover, IndoorCrowd2D can

enable a variety of applications including indoor navigation, localization and emergency management.

The power of crowdsourcing has been confirmed by the success of such projects as OpenStreetMap. It attracts more than one million contributors, and has achieved impressive accuracy in labeling approximately 21,107,200 miles of road across the world. Meanwhile, the ubiquity of smartphones and wearable devices facilitates extending the scale of such systems, liberating users from the need for a computer. Recent work has used crowdsourced smartphone inertial data to infer user trajectories and use them to reconstruct indoor digital map [2, 9, 21, 27, 34, 49]. [19] takes one step further, both image and inertial data are used to generate an indoor floor plan.

To the best of our knowledge, we are the first to propose and implement a smartphone-based crowdsourcing system that explores the power of untrained users to generate building interactive panoramic maps at large scale following an open-source approach. Our system design focuses on indoor panoramic map generation. The final output of our system contains multiple panoramic images for visualizing the building interior view. We also provide a navigable hallway skeleton for each floor that enable users to easily navigate through a large scale indoor environment. Users can take an immersive virtual tour of a building by using an Internet browser.

We solve two major challenges within the IndoorCrowd2D design. Unlike the traditional indoor scene reconstruction technique, which utilizes professional equipment [13, 20, 24] to capture images in a pre-defined path, the image upload from the crowd are determined by many individual users. Therefore, the camera positions, view directions and camera moving trajectories are unknown in advance. This in turn creates a challenge when crowdsourcing and positioning indoor interior image at a large scale. To overcome the first challenge, we draw on multi-dimensional sensing. First, the inertial sensor inside smartphones combine with image data is utilized to track user movements. Then, we leverage multiple users' tracking information to re-establish the building hallway skeleton. Moreover, the output hallway skeleton offers an auxiliary information to help generating indoor scene panorama with the correct position. Regarding to the second challenge in our design, the crowdsourced data is inherently incomplete, opportunistic, and noisy. For instance, the image data is captured and uploaded by different users with a wide variety of mobile devices under various lighting conditions. Hence, our system should be robust and be able to handle heterogeneous, redundant, inaccurate and low-quality data. To solve this issue, IndoorCrowd2D employs a divide and conquer approach: i) the mobile application with a real-time data quality feedback mechanism is designed to guide users to provide good quality data. ii) a hierarchical processing pipeline at the cloud computing backend is designed to gradually filter out redundant and low-quality data. In addition, the building interior view reconstruction algorithm is engineered to guarantee a high accuracy and noise-tolerant performance. We summarize our contributions as follows:

- We formulate the problem via designing two trackable models, and also employ the divide and conquer approach to address the inherently incomplete and noisy crowdsourced data. Image information and sensory data are both unitized in a synergetic manner for the purpose of improving the results accuracy and allowing gradual building-up of the hallway skeleton. In particular, we design an aggregation algorithm, which utilize the visual information as "anchor point" to aggregate crowdsourced sensory data.

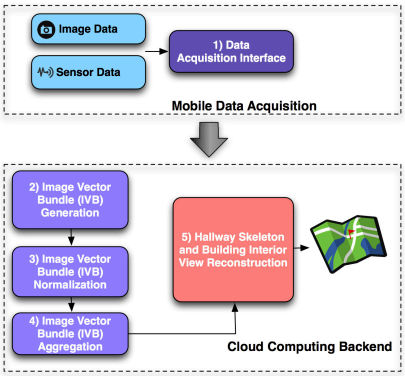


Figure 1: The architecture of IndoorCrowd2D.

- We prototype the IndoorCrowd2D system by implementing a mobile application and a cloud computing backend. The mobile application offers the users a convenient method to capture the indoor interior view. Furthermore, the cloud computing backend is able to automatically generate the interactive panoramic maps.

- We evaluate the IndoorCrowd2D prototype in real-world scenarios. The performance of IndoorCrowd2D exhibits a significantly high recall, good precision and high F-measure. Based on our evaluation result, with approximately 425 crowdsourced datasets from 25 users, we can integrate the hallway skeleton and also generate the panoramas within it.

The rest of this paper expands on each of these contributions, beginning with the design overview and system architecture (Section 2), followed by the system modeling (Section 3), design details (Section 4) and the implementation of a prototyped IndoorCrowd2D (Section 5). Experiments are conducted to assess this system (Section 6). After discussing the limitations of our system (Section 7), our work is compared with related works (Section 8). We summarize this paper in Section 9.

2. DESIGN OVERVIEW

IndoorCrowd2D is a smartphone-empowered crowdsourcing system for building interior view reconstruction. It leverages crowdsourced image and sensory data and does not rely on the priori that knowing any building interior information. The result of IndoorCrowd2D is an interactive panoramic map, which can be divided into two parts: i) indoor panoramic images and ii) building hallway skeleton. The first part aims at creating appealing panoramic images for visualizing the indoor interior view. The second part is built for providing an interactive navigation mechanism through these panoramas.

Compare with conventional approach, which hires volunteers to take pictures and logs the location and direction of

each picture on the digital floor plan manually, it is very challenging to build up an accurate and complete interactive panoramic map by leveraging crowdsourced data only. This is because the crowdsourced individual user data are inherently opportunistic, incomplete, and noisy. Hence, we employ the divide and conquer approach, which is built upon two models: *indoor spatial model* (Section 3.1) and *indoor user trajectory model* (Section 3.2). The first model discretizes the continuous indoor space into grid matrix and represents it as unit cells in metric terms. The second model defines a special data structure named *image vector bundle*, which aims at providing accurate user movement information by combining both the sensory and the image data. Moreover, we apply a reduced Manhattan World assumption (RMW) [12] to simplify the structure of the hallway skeleton. In addition, this hallway skeleton structure is tailored for panoramic image navigation. Based on these two models, a complete building interior map is gradually discovered as long as sufficient crowdsourced data are collected.

As shown in Fig. 1, the architecture of the IndoorCrowd2D system consists of two components: an Android application runs on a mobile platform and a cloud computing backend. The mobile application is responsible for crowdsourced data acquisition. It allows users to shoot and upload building interior scenes annotated with synchronized sensory data including compass, gyroscope and accelerometer to the cloud. Then, the cloud processes multiple datasets simultaneously. For each dataset, the cloud first convert the sensory data and image data to an image vector bundle to represent the geospatial relationship among each consecutive image. Moreover, the cloud normalizes each image vector bundle based on the RMW assumption. For multiple user datasets, the cloud aggregates geospatially-similar image vector bundles together. Finally, our system completely reconstruct the indoor panoramic images, and also the hallway skeleton of the building.

Based on the flow of operations, we further divide the whole system into five modules, namely: 1) data acquisition interface (Section 4.1); 2) image vector bundle generation (Section 4.2); 3) image vector bundle normalization (Section 4.3); 4) image vector bundle aggregation (Section 4.4); 5) hallway skeleton and building interior view reconstruction (Section 4.5).

Since our approach does not require any professional equipments or special trainings, it alleviates the overhead of building interior view reconstruction compared to the conventional approach. Particularly, users are only involved in the data acquisition module. A typical use case could be: a user holds his/her smartphone, starts our mobile application, enters a room (our mobile application takes pictures automatically), walks around inside the room, then moves towards another room through a hallway. To save energy, our mobile application only captures the sensory data and the image data when user starts shooting. Also, it logs out the sensory data when user stops capturing the scene.

3. SYSTEM MODELING

3.1 Indoor Spatial Model

In IndoorCrowd2D, we design our own indoor spatial model to build the coordinates of the 2D indoor space. It represents the indoor environment by a homogeneous grid matrix, which reflects the accessibility of the indoor environment

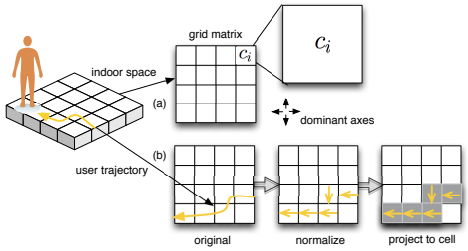


Figure 2: Indoor spatial model for IndoorCrowd2D

(shown in Fig. 2). Specifically, each cell contains a numerical value and is initialized as 0. It increased by 1 every time if a user trajectory is successfully projected to the cell grids. In addition, we store the user trajectory information to the cell for further processing. This model is tailored to IndoorCrowd2D because it provides trackable results and is applicable for our divide and conquer approach.

In order to initialize the grid matrix for each floor as well as preserve the structural and geometrical properties of the space at the same time, we introduce the reduced Manhattan World (RMW) assumption. According to the RMW assumption, we assume two perpendicular dominant axes are existed for each building floor. Also, each line segment (most corridors) inside the building floor is supposed to align with one of the two axes. Compare with predominant path-based space assumption that applied in [2, 19], the RMW assumption offers the following advantages in our building interior view reconstruction system: i) it produces clean, simple hallway skeleton as outputs, which is good enough for indoor panoramic map navigation. ii) it normalizes the orientation of each image, which benefits the reconstruction of the indoor interior views. Note that the RMW assumption may not suitable for some particular buildings, which are not predominantly rectangular (e.g. the Pentagon). However, according to [38], more than 90% of modern buildings are rectangular actually. We moreover discuss how to reconstruct non-predominantly rectangular building and the possible impact of accuracy in Section 7.

Based on the RMW assumption, the grid matrix initialization process can be divided into two steps: First, we detect the dominant axes of the building (See Section 4.3). Second, we divide the indoor building space into a uniform square unit cells $C = \{c_{0,0}, c_{0,1}, .., c_{n,n}\}$ upon the detected dominant axes of the building. Then, we initialize each cell with value 0.

3.2 Indoor User Trajectory Model

We further put forward an indoor user trajectory model to provide accurate matches among the inherently incomplete, uncoordinated, and noisy user data. In this model, we propose to use a tailored image vector data structure to not only represent user movement inferred from the sensory data, but also include the corresponding image information. This proposed model takes an individual user trajectory, i.e., a sequence of captured images annotated with sensor data as the input. Next, it outputs a discretized image vector bundle, which is comprised of many image vectors and the geospatial information of these image vectors (shown in Fig. 3). Then, each image vector is mapped to a unit cell in the space model. The spatial relationships of the unit cells

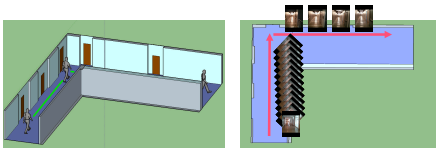


Figure 3: Indoor user trajectory model for IndoorCrowd2D

are further deduced from the image vectors. In general, our proposed models spontaneously allow the hallway skeleton to be built-up gradually.

We define the user trajectory by a sequence of image vectors $\boldsymbol{Z} = \{\vec{z}_1, \vec{z}_2, ..., \vec{z}_t, ..., \vec{z}_T\}$ represents the geospatial characteristics of user movement at each time t during the total track life T. We also refer $\boldsymbol{Z}^{\kappa}$ as an image vector bundle (IVB) generated by user with a reference number κ.

For each time step, we generate an image vector $\vec{z}$ by converting the raw sensor and the image data through image vector bundle generation algorithm (Section 4.2). Each of the image vectors $\vec{z}_t = [x^t, y^t, \vec{\psi}^t, \boldsymbol{I}^t, t]$ represents one user's movement, which include relative spatial location $[x, y]$, heading direction $\vec{\psi}$, image $\boldsymbol{I}$ at time t and timestamp. All the information is all extracted through the sensor data and camera visual tracking. Since a discretized indoor spatial model is applied, the normalization of each image vector $\vec{z}$ and the projection of the $\vec{z}$ to a cell $c_{i,j}$ are necessary.

3.3 Basic Elements of Building Interior View

In IndoorCrowd2D, we consider five basic elements of every building interior view, each of which defines a particular indoor scenario: corridor, room, wing, intersection, and open area (shown in Fig. 4). All of these elements have its own special visual or sensory features. We now briefly explain these basic elements: 1) Corridor, it is a narrow tract and usually have a long passageway. It is full of textureless walls and have various lightning conditions. The users' moving trajectories inside corridor are expected as a bidirectional flow; 2) Room, it is a portion of space separated by walls. An occupied room usually contains a few texture-rich man-made objects. The moving patterns of the user inside a room may differ a lot and may cover all locations where are reachable to the user. 3) Wing, it is a special corridor which one side is wall and the other side is transparent windows or open space. It usually contains more visual textures compare to corridor. However, the lightning conditions may various a lot due to the affect of sun-light. 4) Intersection, a place at which two or more corridors (or wings) cross. Users take turns at the intersection point and therefore create a unique moving pattern. 5) Open Area, an open space where it is like a room without walls (e.g. a lobby). Users could reach this place through nearly all possible directions. Therefore, the aggregated user trajectories could be complex at this place.

4. DESIGN DETAILS

4.1 Data Acquisition Interface Design

Data acquisition interface is designed for collecting images and sensory data from each user. In order to enhance the overall crowdsourced data quality from users, we design a

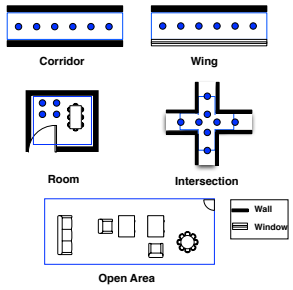


Figure 4: Basic elements of building interior view

mobile application that is deployed on user's smartphone. Compare to conventional approach that user requires to use specialize equipment, locate and logs the direction of each picture and walk through a pre-define path to capture the indoor scene, our application provides a much more handy data capture process. Indeed, we simplify the whole data capturing process and only require users to put minimal effort in capturing the indoor scene by using our mobile application. We further conduct a mobile application usability study in Section 6.

Proactive Data Collection. IndoorCrowd2D follows a proactive data collecting mechanism. Our application uses the smartphone's back camera to capture the indoor scene and does not require any other specialized equipment. Once start capturing, users can move freely and steadily holding the smartphone vertically or horizontally (landscape mode). Our mobile application automatically takes pictures for every t seconds and records the sensory data (accelerometer, gyroscope and compass) with the timestamp simultaneously.

Real-time Data Quality Feedback. In our smartphone application, we implement a real-time data quality feedback mechanism similar to [49] to guide the users to provide high quality data. While the user is capturing the scene, our application continues estimating the data quality metrics by processing the sensor data and the image data in real-time. The metrics are measured by the smartphone application include: i) linear acceleration, ii) angular acceleration and iii) the number of SURF [4] features in each picture. Moreover, the value of linear acceleration and angular acceleration can be directly read from mobile platform API, whereas, the number of SURF features should be obtained from the SURF feature detection algorithm by leveraging the OpenCV [7] library.

If the measured values of i) and ii) are beyond a certain threshold $(\vec{v}_h, \vec{\omega}_h)$, it indicates that the user either moves or turns too fast. In this case, a hint appears on the screen to remind the user to slow down. If our application detects the number of features in iii) falls below a predefined threshold (τ_h), this exhibits that the user shoots feature-less objects, such as walls, and thereby, our application displays a hint that guides the user to record other places and avoids some feature-less objects.

Fig.5 shows the screen displays the SURF features of a image on the mobile platform in real-time.

4.2 Image Vector Bundle Generation

IndoorCrowd2D uses both the image and the sensor data to detect user motions and fuses these two orthogonal domains of data together to generate the image vector bun-

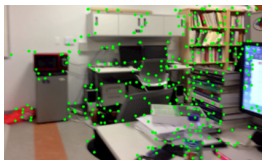


Figure 5: Detect SURF feature points of a image in real-time

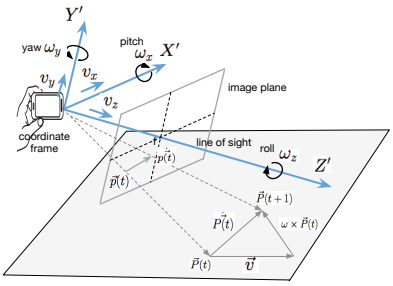


Figure 6: Optical flow model

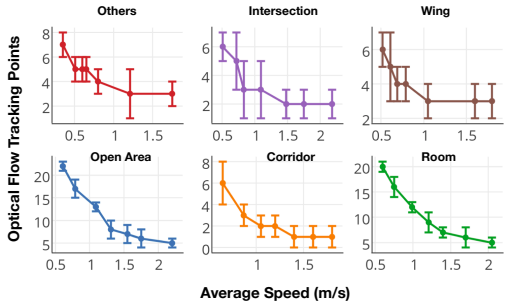


Figure 7: The relationship between user average speed and the number of optical flow tracking points

dle $\boldsymbol{Z}$. For each image vector $\vec{z}_t = [x^t, y^t, \vec{\psi}^t, \boldsymbol{I}^t, t] \in \boldsymbol{Z}$, the image data $\boldsymbol{I}^t$ and timestamp t are obtained directly from smartphone. The relative spatial location $[x^t, y^t]$ is initialized as 0 and calculated based on the difference of smartphone spatial location between the consecutive image vectors. The initial heading direction $\vec{\psi}^0 \in \vec{z}_0$, is acquired by the compass as long as the user starts capturing. As it is pointed out by [31], the compass is erroneous in indoor environment due to the higher magnetic inference in indoors than outdoors. Therefore, a compass calibration is necessary before starting the scene capturing. Note that the calibrated compass value is only applied to set the initial heading direction $\vec{\psi}^0$ of the user.

Relative Spatial Location Calculation. For each two consecutive image vector $\vec{z}_{t-1}$ and $\vec{z}_t$, the angle difference of heading direction between $\vec{\psi^{t-1}}$ and $\vec{\psi^t}$ is derived by the heading offset estimation method [31] and the reading of gyroscope [29, 42]. Furthermore, the distance $|z_{t-1}, z_t|$ is calculated by fusing the sensory data and the image data.

For the sensory data (accelerometer, gyroscope and compass), step counting method [29,48] is applied to measure the distance. For the image data, the relative speed of motion of

Figure 8: (a). Contour of a building (b). Two dominant axes in red and blue picked up by the contour sample in cyan

the user between $\boldsymbol{I}^{t-1}$ and $\boldsymbol{I}^t$ is measured by using an optical flow model [25]. For a given pair of consecutive images $\boldsymbol{I}^{t-1}$ and $\boldsymbol{I}^t$ with the same or extremely similar indoor scenes, we can use the corresponding image feature movements (feature vectors) to estimate user velocity. The image feature is defined as some specific structures in a image. Fig. 6 illustrates the optical flow model used in the IndoorCrowd2D, which the smartphone held by a user moves in the ground-plane with the velocity $\vec{v} = (v_x, v_y, v_z)^t$ and the rotational velocity $\vec{\omega} = (\omega_x, \omega_y, \omega_z)^t$. Also, through using a perspective projection, the user movement trajectory $\dot{\vec{P}}(t)$ in the ground plane is expressed by the dynamics of the feature vector $\dot{\vec{p}}(t)$ in the image plane. Importantly, we require users holding their camera at the same height when applying the optical flow model. This criteria can be ensured by checking the relative angle change of the smartphone through gyroscope. Based on the optical flow model, the feature vector $\dot{\vec{p}}(t)$ is calculated using the iterative pyramidal Lucas-Kanade [6] method. The relationship between user average speed and the number of optical flow tracking points is shown in Fig. 7. According to the result, the optical flow based method provides auxiliary result when the user average speed is less than 1m/s.

Since the rotational velocity $\vec{\omega}$ can be directly acquired from the gyroscope sensor, the velocity of the feature vector $\vec{v}$ in the image plan can be simplified as:

$$\frac{\partial \dot{\vec{p}}(t)}{\partial t} = \vec{v} = \tilde{v}_x * \boldsymbol{i} + \tilde{v}_y * \boldsymbol{j} \tag{1}$$

Based on the equation (1), if the focal length of the smartphone camera f and the current distance H to the scene are given, we are able to calculate user velocity as:

$$\vec{v} = \vec{\tilde{v}} \frac{H}{f} \tag{2}$$

Next, the timestamp t is used to measure the time duration between every two consecutive images, and thereby, calculate the distance $|z_{t-1}, z_t|$. To merge the distance inferred from both sensory and image data, IndoorCrowd2D assigns a weight for each result and use the weighted linear combination method to achieve the merging purpose. Thus, the relative spatial location $[x^t, y^t]$ for $\vec{z}_t$ is easily acquired based on the distance $|z_{t-1}, z_t|$ and the heading direction $\vec{\psi}^t$.

4.3 Image Vector Bundle Normalization

After the image vector generation process, we further normalize the heading direction $\vec{\psi}^t$ for each image vector $\vec{z}_t \in \boldsymbol{Z}$ using the reduced Manhattan World (RMW) assumption. Under the RMW assumption, we assume that each building has two perpendicular dominant axes in 2D and each line segment (mostly corridors) inside the building is aligned to

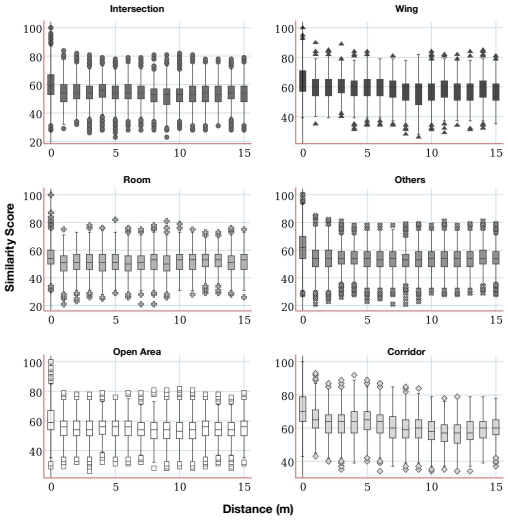


Figure 9: The relationship between image location and pHash similarity score (fixed walking direction).

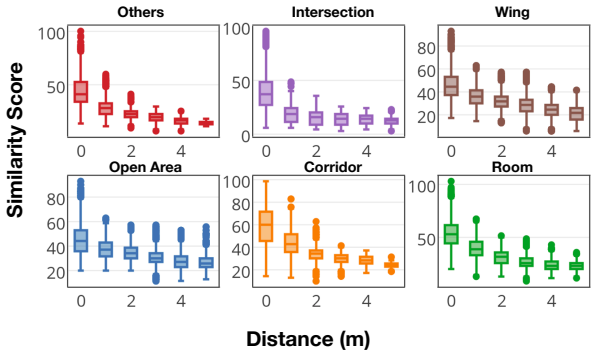


Figure 10: The relationship between image location and the similarity score of image feature matching (fixed walking direction).

one of the two axes. IndoorCrowd2D applies the following steps to detect the dominant axes of the building and align the user heading direction to one of the two dominant axes:

Dominant Axes Detection. We determine the two dominant axes by analyzing the building outlines that obtains directly from the Google Maps. This concept is identical to that in [39]. However [39] does not present an implementation and we hence conduct our own. First, we apply both the thresholding and the edge detection to the target image. Then, through utilizing the Canny Detector from the OpenCV, it extracts the contour of the building, which in the form of a series of clockwise vectors as illustrated in Fig. 8 (a). The next step is to normalize the vector directions to $0-90$ degrees, and also place those angles into a histogram. Through sampling the contour points every few pixels, the impact of short contours (contours only one pixel in length) is avoided. The average angle from the most frequent bins is taken as the direction of the first dominant axis, whereas the second dominant axis is perpendicular to the first dominant axis, as shown in Fig. 8 (b).

Normalize Heading Direction. we check the heading direction $\vec{\psi^t}$ for each image vector $\vec{z}_t$. As the gyroscope is accurate enough for detecting the change of the heading direction [29,42], it is sufficient to normalize the heading direction into four possible movement direction $(\uparrow, \rightarrow, \downarrow, \leftarrow)$ in a two-dimensional space. For each image vector bundle $\boldsymbol{Z}$, the heading direction $\vec{\psi^t}$ is normalized by checking whether the value difference for each consecutive image vector is larger than 45 degrees.

4.4 Image Vector Bundle Aggregation

The image vector bundle aggregation module is designed to merge multiple image vector bundles. The main objective of the image vector bundle aggregation module is to achieve robust performance across a large variety of image vector bundles generated by different users, different smartphone models or various indoor environments. In order to decrease the complexity of its input, IndoorCrowd2D adapts a hierarchical approach wherein image vector bundle matching and merging take place in the following three steps.

pHash Filtering. The primary focus of the pHash algorithm [47] is calculating a fingerprint or hash of a particular image based upon the various visual features of its content, rather than processing the raw bits in the image. Unlike cryptographic hash functions such as SHA-1, in which small changes in input lead to drastically different hash values, pHash produces similar hash values if the visual features of the two images are similar. Thus, the similarity between two images is computed by calculating the Hamming distance between the two output fingerprints produced by pHash. pHash is very fast compared to other image similarity comparison methods, it only takes 65 seconds to hash 100 images. Fig. 9 shows the relationship between image location and pHash similarity score under different building interior scenes. The x-axis in this figure represents the linear distance between the shooting locations of the two images. According to the result, if we set the similarity score threshold properly, the pHash algorithm is able to reject most of the incorrect match but only preserve structurally identical indoor images (images shoot at the same indoor location).

Image Feature Extraction and Matching. To precisely match two image vector bundles, the SURF descriptors is selected for representing the points of interest on each image. We choose the SURF algorithm instead of SIFT because it is more robust, and its speed is sufficient for real-time processing, which is critical when dealing with crowdsourced data. In our model, given a query image $\boldsymbol{I}^a \in \boldsymbol{Z}^A$ and a set of candidate images in $\boldsymbol{Z}^B$, we perform a match in the following manner: i) Construct a codebook of "visual features" through using the SURF algorithm; ii) Quantize these visual feature descriptors by the k-nearest neighbor (kNN) algorithm where $k = 2$; iii) Use Euclidean distance as a similarity metric for computing the number of good matches. The total process take around 1.5 seconds to match two images. Fig. 10 shows the correlation between distance and the similarity score of image feature matching. Similar to Fig. 9, the x-axis in this figure represents the linear distance between the shooting locations of the two images.The result exhibits that by using our kNN-based image feature matching algorithm, we are able to set a similarity threshold that only preserve the images, which belongs to the same location (distance less than 1m).

Longest Common Subsequence Image Vector Aggregation. Once a match between two images is detected, IndoorCrowd2D calculates the similarity scores of the two corresponding image vector bundles (IVBs) based on the longest common subsequence (LCS) metric. Let $\boldsymbol{Z}^A, \boldsymbol{Z}^B$ be the two IVBs with the length of i and j, respectively. Also, given that system metric is δ and matching threshold is ϵ, the LCS metric for the two IVBs is defined as follows:

$$L\left(\boldsymbol{Z}_i^A, \boldsymbol{Z}_j^B\right) = \begin{cases} 0, if\ i = 0\ or\ j = 0; \\ 1 + L(\boldsymbol{Z}_{i-1}^A, \boldsymbol{Z}_{j-1}^B), \\ \quad if\ d(\vec{z}_i^A, \vec{z}_j^B) \leq \epsilon\ and\ |i-j| < \delta; \\ max(L(\boldsymbol{Z}_i^A, \boldsymbol{Z}_{j-1}^B), L(\boldsymbol{Z}_{i-1}^A, \boldsymbol{Z}_j^B)), \\ \quad otherwise; \end{cases}$$

where parameter δ is used to control the maximum length difference between the two IVBs and ϵ represents the distance threshold. The similarity score SS for the two IVBs based on [50] is expressed as follows:

$$SS(\boldsymbol{Z}^A, \boldsymbol{Z}^B) = max_{f \in \mathscr{F}} \frac{L(\boldsymbol{Z}^A, f(\boldsymbol{Z}^B))}{min(i,j)} \tag{3}$$

where $\mathscr{F}$ stands for a set of all possible translations. In our prototyped IndoorCrowd2D, if SS is larger than threshold ss_h, two IVBs can be merged into one larger IVB.

4.5 Hallway Skeleton and Building Interior View Reconstruction

In the last module, we leverage aggregated image vector bundles to generate the hallway skeleton and the indoor interior view. The hallway skeleton represents the topology of the hallway for navigation purpose. As it is mentioned before, IndoorCrowd2D selects an indoor spatial model to represent the environment. Hence, we need to further project our aggregated image vector bundles to the indoor space cell to reconstruct the hallway skeleton.The indoor interior view is generated separately by leveraging a δ-building interior view generation algorithm. This algorithm is a combination of several state-of-the-art panorama reconstruction algorithms. The end result of IndoorCrowd2D is an interactive building interior view reconstructed from the output results.

Reconstruct Hallway Skeleton. For each floor, we select the aggregated image vector bundles $\boldsymbol{Z}^{aggr}$ and project it to the indoor space matrix based on our indoor spatial model. The whole procedure is as the following: First of all, there is a homogeneous grid cell matrix to represent the indoor space, and also, each cell inside the matrix is initialize as number 0. Then, we map $\boldsymbol{Z}^{aggr}$ to the cell matrix by checking the relative spatial location $[x, y]$ of each image vector $\vec{z}_i \in \boldsymbol{Z}^{aggr}$. If there exists an image vector $\vec{z}_i$, which corresponds to a particular cell $c_{i,j}$, we therefore increase the value of $c_{i,j}$ by 1. We then store the reference number $[i, aggr]$ of image vector to $c_{i,j}$ The process is continuously repeated until all image vectors in the aggregated image vector bundle $\boldsymbol{Z}^{aggr}$ are projected to the indoor space matrix. Finally, the cells with value higher than a threshold h_{cell} (means this position is accessible) inside the indoor space matrix is chosen to represent the hallway skeleton.

δ-Building Interior View Generation. Up to this step, the hallway skeleton is completely generated. However, we also interested in the generation of a visually appealing building interior view. Therefore, we choose to use a cylindrical panorama, as such panorama present users with a more realistic representation of the indoor environment.

To generate a cylindrical panorama, the qualified images inside the aggregated image vector bundle Z^{aggr} have to satisfy with: i). the user's locations of these images should be as close as possible. ii). the viewing direction of these images should cover the scene in a wide angle (e.g. 360 degree for a 360-panorama). iii). every two images should have appropriate overlap part. For criteria i and ii, we use the reconstructed hallway skeleton for accurately selecting the candidate images. First, for a particular cell c_i inside the indoor space matrix, we check if it is accessible (the value is larger than h_{cell} if it's accessible). Second, if their exists more than one image vector $\vec{z}$ corresponding to c_i due to image vector aggregation, we then select images inside these image vectors as candidate images because they belongs to the same location, and therefore they should satisfy criteria i. Third, we check the normalized heading direction of these candidate images to make sure these images share the same or consecutive orientation values, which satisfies criteria ii. Finally, we send the images passed the check to a panorama reconstruct pipeline.

The panorama reconstruct pipeline performs incremental image stitching through a feature detector module, a feature matching module, and an inlier pair match estimation module to select the best matches (shown in Fig. 11 (a)). Although the shortcomings of our approach for image stitching include a slightly slower speed and a less accurate process comparing to state-of-the-art softwares (e.g., Autostich [8]), the results are more accessible dealing with non-overlap and textureless objects (criteria iii), which makes it particularly suitable for our crowdsourced image data.

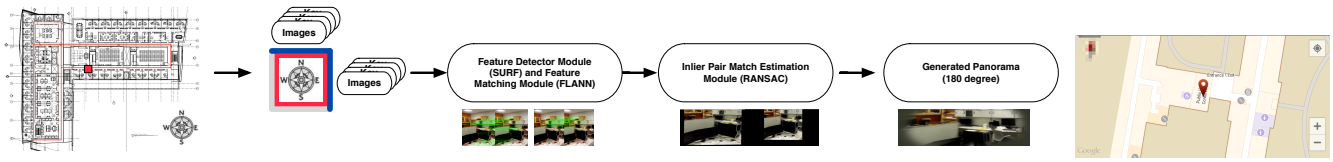
Fig. 11 (a) exhibits the δ-building interior view generation process in a typical scenario. The output of RANSAC algorithm is a homography matrix, which establishes the number of features pairs that match closer than a predefined threshold. Once the homography matrix is computed, it is utilized to overlap the source image with the target image. In our scenario, however, the crowdsourced images may exhibit strong differences in light and white balance due to the same scene being captured from different perspectives and by different people. In order to obtain the good panorama images, it is necessary to perform images blending[8] first to smooth out the discontinuities between the two overlapping pictures before applying the homography matrix.

5. IMPLEMENTATION

The prototype of the IndoorCrowd2D system is composed of two parts: a data acquisition interface runs on Android mobile device and an indoor interior view pipeline deploys on a cloud.

For the data acquisition interface, we implement and test it on Android 4.0 KitKat. Three different smartphone models are chosen to test the compatibility, includes Google (LG) Nexus 5, Google (LG) Nexus 4, and SAMSUNG Galaxy Nexus. The reconstruction pipeline is implemented on Microsoft Azure platform using a network optimized A9 virtual machine (16 cores, 112 GB memory and Ubuntu Linux).

1)Mobile Application. Our mobile application enables the crowd to capture the building interior view. These user-generated-images can be further uploaded to cloud from user smartphones. For saving bandwidth, we pre-process the



(a) (b)

Figure 11: (a) The δ-building interior view generation process in a room scenario, (b) Building location detection by dropping pinpoint

sensing data and filter out the redundant and low-quality part. The qualified datasets are zipped and separated into 5MB chunks for transmitting. The transmission only starts when Wi-Fi connections are available.

Real-time Data Quality Detection. To support on-board real-time image processing, we leverage an open-source OpenCV library to perform SURF feature extraction, and to process the extracted image and detect its quality further. To reduce the running time of the SURF feature detection, we tune the parameters to produce fewer SURF features per image. Note that this modification can reduce the feature detection accuracy. However, since we only need to judge if the numbers of features are fall below a threshold τ_{fea} (prevent user shooting feature-less object), the loss of accuracy does not affect our system.

Application Settings. We create a interactive setting interface to allow users input the building floor information. As shown in Fig.11 (b), once a user clicks the setting button, a pinpoint on the map appears to indicate their last known GPS location. These results may be inaccurate, which we address by allowing the user to drag and drop the pinpoint onto the correct map position. Note that we do not need users to provide very accurate location information, as this location information is only used for building detection. Our system works well as long as the pinpoint falls inside the outline of the target building. Once the user complete the previous step, a new screen will appear prompting the user to input their current floor number. Their input is then sent to the cloud along with building location data. Also, the input data is stored in our application. User does not need to re-input it unless he/she starts reconstruct a new building floor.

2)Cloud Computing Backend. The IndoorCrowd2D cloud computing backend is built upon Windows Azure virtual machine instances with Ubuntu Linux. Based on function, we can divide cloud computing backend into two layers: i) a communication layer which receives and stores the incoming crowdsourced image and sensory data. ii) a data processing layer to process received data and generate hallway skeleton and the interior views of the building.

Communication Layer. In this layer, we select an Tornado web server to receive crowdsourced data. The Tornado http server is a powerful, flexible web server. The mobile application send zipped data to Tornado web server in real-time through HTTP requests.

Data Processing Layer. Data processing layer process the crowdsourced data and output the final result. In the first step, our program first stores the received raw data into MongoDB. Then, our task scheduler (APScheduler) loads the data and send it to the hallway skeleton and the interior view reconstruct pipeline. Several function modules compose the pipeline and are all controlled by a task coordinator. Once the reconstruction is finished, the task scheduler stores the result back to the database and wait for new tasks. For visualizing the result, the WebGL technique is applied to create an interactive online GUI. It in charges of the user input and the result display (shown in Fig. 12). However, users can also ported our data to other online map services by using its API.

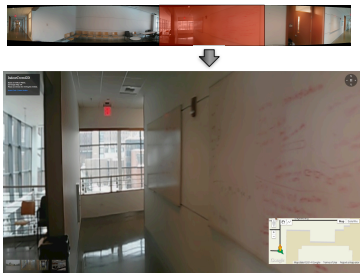


Figure 12: Screenshot for visualizing building interior view and hallway skeleton

6. PERFORMANCE EVALUATION

We evaluate our IndoorCrowd2D prototype in the following scenario: untrained and uncorrelated volunteers use our mobile application capturing indoor scenes in a typical college building to reconstruct the building interior and skeleton. We collect data on the college buildings at different times of day, and over a period of five months. Before conducting the experiment, 55,453 images of two different buildings (Teaching Building (TB) dataset and GYM dataset) from 1,151 datasets are successfully uploaded by 25 users. Some places were captured multiple times. Fig. 13 (a) shows the distribution of the basic elements inside the data.

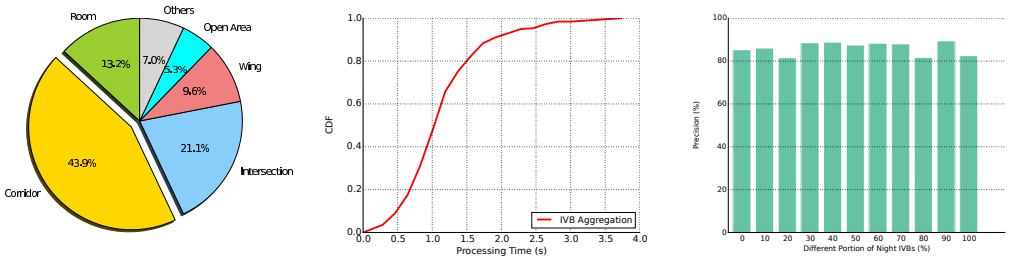
6.1 Evaluation Methodologies

The ground truth hallway skeleton $\mathcal{S}_{true} = \{c_1, c_2, .., c_n\}$ is learned from the building floor map. Therefore, once we generate a building hallway skeleton $\mathcal{S}_{gen}$, the precision, recall, and F-measure of IndoorCrowd2D can be defined as follows:

$$precision = \frac{|\mathcal{S}_{gen} \bigcap \mathcal{S}_{true}|}{|\mathcal{S}_{gen}|} \quad (4)$$

$$recall = \frac{|\mathcal{S}_{gen} \bigcap \mathcal{S}_{true}|}{|\mathcal{S}_{true}|} \quad (5)$$

$$F - measure = 2 * \frac{precision * recall}{precision + recall} \quad (6)$$



(a) The distribution of the crowdsourced building interior data (b) Building hallway skeleton reconstruct computational latency (c) Tolerance of various lightning conditions

Figure 13: IndoorCrowd2D performance evaluation for: (a) The distribution of the crowdsourced building interior data (b) Building hallway skeleton reconstruct computational latency (c) Tolerance of various lightning conditions

Note that, these performance metrics not only reveals the accuracy of the generated building hallway skeleton, but also affects the quality of panorama generation (because we use the hallway skeleton as the auxiliary information to help generate indoor panorama with the correct position).

For comparison purpose, we compute the precision, recall, and F-measure for both our method (with sensory and image data) and image only method (without sensory data). The image only approach is widely used in computer vision community. They assume the images are crowdsourced from the Internet. In this approach, the spatial relationship of each image is deduced by processing the visual information (e.g. through feature matching).

6.2 Evaluation Results

Hallway Skeleton Reconstruction Performance. As shown in Fig. 16, we compares the performance of IndoorCrowd2D with the image only approach. The precision, recall, and fall-out are computed over the entire set of pictures. The results exhibit that IndoorCrowd2D achieves a precision around 85%, a 100% recall and a F-score around 95% for the two datasets. We find that existing state-of-the-art vision algorithms still exist some drawbacks by further checking the results. In our crowdsourced test datasets, these computer vision algorithms tends to give incorrect results when encounter images of extreme distance/angle, feature-less objects (e.g. walls). According to the results, our image and sensor hybrid method is more robust to errors and outliers compare to the image only method.

Computational Latency. IndoorCrowds2D computational latency is highly dependent upon the time needed for the computer vision based image processing, especially the dominating subtask for aggregating two image vector bundles. Fig. 13 (b) plots the CDF graph of the single-threaded performance for matching two image vectors inside two different vector bundles $\boldsymbol{Z}$. Given that each image vector bundle contains around 20 image vectors in general, the performance of our system is comparable to the state-of-the-art.

Tolerance of Various Lightning Conditions. To test IndoorCrowd2D under varying lighting conditions, we manually select datasets that are captured from different periods of time in a day and categorize them into two groups: daylight datasets and night datasets by judging its images and timestamp. We keep the size of daylight dataset and night datasets equal.

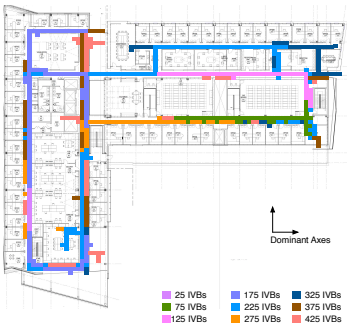


Figure 14: A typical gradually build-up scenario for IndoorCrowd2D (TB Dataset)

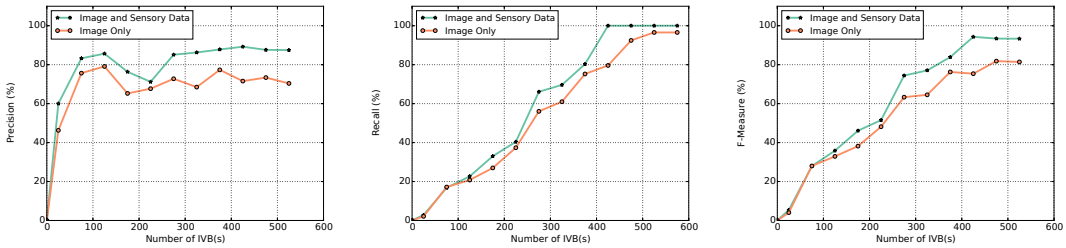
(a) Output Panorama

(b) Ground Truth Panorama

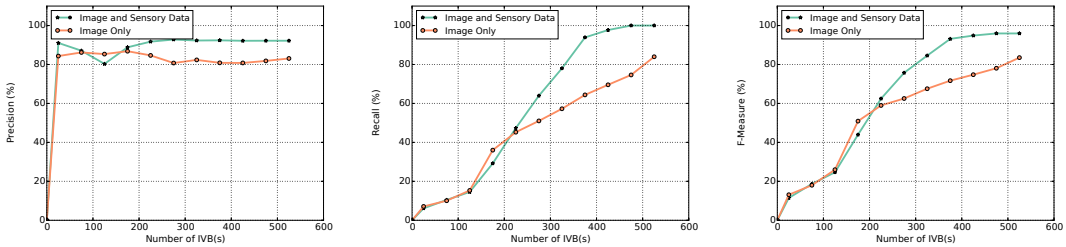
Figure 15: Quality comparison between the crowdsourced panorama (a) and the ground truth panorama (b). (TB Dataset)

We perform the following experiment: We randomly replace part of the daylight dataset, which is 10% of the dataset size in our experiment, with the same amount of night dataset. We keep doing this process and conducting the match on each newly generated dataset until the dataset becomes all-night. Fig. 13 (c) shows the skeleton precision with different portions of night IVB datasets. The result shows that the our building interior view reconstruction pipeline is robust to various lighting conditions.

Allowing Gradual Skeleton Build-up. Our building hallway skeleton reconstruction pipeline allows gradual



(a) Hallway skeleton precision for TB dataset (b) Hallway skeleton recall for TB dataset (c) Hallway skeleton F-measure for TB dataset



(d) Hallway skeleton precision for GYM dataset (e) Hallway skeleton recall for GYM dataset (f) Hallway skeleton F-measure for GYM dataset

Figure 16: IndoorCrowd2D performance evaluation (precision, recall, F-measure) for (a)-(c) TB dataset and (d)-(f) GYM dataset

build-up to save computational cost. IndoorCrowd2D automatically saves the previous reconstruction state when all the datasets are being processed. When new datasets arrive, it will restore the previous state and continue processing. Fig. 14 presents a typical gradually build-up process of the hallway skeleton construction, with new image vector bundles being added progressively. For the purpose of better visualization, we manually check each image inside the merged image vector bundles, and find its best possible position (by rotating it if possible) on the ground truth floor map. Note that, in this figure, we do not replace the previous longest skeleton with the newly generated one. Therefore, the final result shown in this figure is slightly different from the real data.

Quality Rank of the Resulting Panorama. Fig. 15 (a) presents a typical crowdsourced panorama output by the IndoorCrowd2D. Comparing with the ground truth, the output panorama preserve most of the details for an indoor scene. Thus, the overall quality of the crowdsourced panorama and the ground truth are evenly matched. However, since the crowdsourced image are taken by different users with different angles, and thereby, the resulting crowdsourced panorama inevitably containes some inconsistency and distortion parts.

Mobile Application Usability Study. 25 volunteers filled out a survey to evaluate the usability of our mobile application after they finished capturing the building interior. The survey contains two yes-no questions: a). "The usage instructions of the IndoorCrowd2D are easy to perform?" b). "Do you think the real-time feedback displays on the screen helps?" The results show that 80% of the volunteers agree on the first question and 88% of the volunteers agree on the second one. According to the results, the majority users believe that our mobile application alleviates the burden of building interior view reconstruction and encourages them to contribute high quality data.

7. DISCUSSIONS AND LIMITATIONS

Digital Floor Plan as A-priori Information. If the digital floor plan of a building is available as prior knowledge, then it can be utilized to detect the actual position of each room and corridor. In this case, we can adjust the image vector to the correct position by solving a non-linear least squares problem. We may for instance use the Ceres Solver to perform sensor data fusion, in order to estimate the image vectors' correct positions and orientations inside the building. Since we no longer rely on the Manhattan world assumption (we do not need to calculate the dominant axes), our system will thus be able to support non-predominantly rectangular buildings.

Reconstruct Building Skeleton for an Open-area. Unlike corridors and rooms, an open area such as a lobby may allow users to approach from different directions. Image vectors generated by different users will hence be aggregated in this area, due to our reliance on image features for matching pairs of image vectors. We state that IndoorCrowd2D is still able to provide reliable results in this special case due to the following reasons: As mentioned previously, for a particular location, if there exists more than one image, and these images share the same consecutive orientation values, a panorama will be generated by δ-building interior view generation algorithm. In an open-area, a 360° panorama would therefore be generated once sufficient key frames are gathered from all directions. The subsequent image vectors that captured the same open-area would then all be matched to the 360° panorama and converged into a point.

Reconstruct Multi-Floors in Single Round. IndoorCrowd2D requires users to input the number of floors they

are on before they start capturing a scene. Once they input the correct floor number, we ask the user to stay on the same floor while capturing data. Towards the goal of enhancing the usability of IndoorCrowd2D, however, we plan to support 2.5D building interior and skeleton reconstruction in the future. According to [42], stairways can be detected by comparing the accelerometer patterns when the user is walking on stairs.

Non-predominantly Rectangular Buildings. As IndoorCrowd2D assumes there is not any prior-knowledge available about the building skeleton, we utilize the well established reduced Manhattan world assumption to compute the dominant axes for each building based on its outline. For reconstructing non-predominantly rectangular buildings, the key challenge is the dominant axes may differ from our output result. Hence, we need to leverage a digital floor plan to detect the actual position of the main skeleton of the building. In our future work, we plan to solve this problem by creating a system letting user providing building skeleton information.

Energy Consumption. IndoorCrowd2D mobile application runs on a user's smartphone. We measure the energy consumption of our mobile application by using the drop rate and the residual capacity of the battery. The result shows that our mobile application takes 1800 mW on average for capturing the scene in four minutes. The inertial sensor consumes about 35 mW when user capturing the environment. Photo shooting takes an average of 300 mW for continues shooting with a resolution setting of 640x480 and a time interval of 3s. However, unlike other mobile crowdsourcing systems, our mobile application does not require users to continues running a daemon process in the background. Hence, capturing several indoor locations should not constitute significant power consumption.

8. RELATED WORK

Indoor Smartphone Sensing. Recently, many researches focus on utilizing smartphone sensing technique to determine the status of a pedestrian in an indoor environment [3,10,11,23,29,31,39,42,43,46,49], such as heading direction, location and walking trajectory. For example, [11] jointly utilizes accelerometer, gyroscope and compass to obtain the user heading direction information. This method is limited due to error accumulation of the inertial sensors. Furthermore, [31] develops a system to accurately detect user heading direction by analyzing user walking in depth. Additionally, it is able to predict and reduce the magnetic interference of compass data. [32] utilizes camera picture combined with computer vision technique to detect the changes of heading direction by calculating the vanishing points. [39], similar to [32], leverages front camera pictures to calculate the building internal line-shape objects as a hint to detect user walking direction. For determine user location, the traditional way is using Wi-Fi fingerprint to locate the user. [29] applies collected user trajectory information to constrains the possible user location by physical walls and use the crowdsourced Wi-Fi data to establish a fingerprint database.

Digital Floor Plan Generation. There are many studies about the building floor plan generation, most of which focus primarily on the sensor data aggregation [2,19,35,41]. [2] uses crowdsourced data from smart phone sensors to automatically and transparently construct accurate motion traces. [19] utilizes both image and inertial data to infer the spatial information for an indoor environment and eventually generate an indoor floor plan. Our work differs from theirs is that our system aims to reconstruct visually appealing indoor interior view, instead of a floor plan. Therefore, we only rebuild a simple hallway skeleton leads to accurately locate reconstructed interior views and helps user navigate through these views.

Indoor Scene Reconstruction. Lately, several researches are focused on indoor and outdoor scene reconstruction [30]. [33] develops a smartphone application to let user capture a panorama of indoor scenes. Basically, users are able to label the edge of the room directly at the panorama. According to the labels, the application is capable to reconstruct room shapes based on the Manhattan World assumption. [22] proposed a set of key-frame selection algorithms based on crowdsourced sensor-rich videos to automatic generate outdoor panorama. In addition to 2D scene reconstruction, indoor 3D modeling is also a hot research topic. Some studies leverage professional equipments, such as depth sensors [45] and laser scanners [5, 44] to reconstruct indoor scene. Other works like [13, 26] uses Kinect, a commercial depth sensor to reconstruct 3D model. Recently, Google lunches the Project Tango [28] that contributes to a tablet for mobile 3D sensing. Although the output 3D models of these approaches are impressive, specialized equipments are unadoptable in our crowdsourcing setting. Moreover, the computer vision community develops several computer vision techniques to reconstruct 3D models directly from images [15, 16]. They use some state-of-the-art 3D reconstruction techniques including Structure from Motion [1,36,37,40] and Multiview-stereo [17, 18]. To enhance the result quality, an user-labelling sub-system is further developed in a recent work [45]. However, because these computer vision based 3D reconstruction algorithm still exists some drawbacks, the images used for 3D reconstruction need to be carefully selected [14]. Thus, these techniques are not suitable for our crowdsourced scenario.

9. CONCLUSION AND FUTURE WORK

We have described the design, implementation, and evaluation of IndoorCrowd2D – a novel crowdsourcing system empowered by off-the-shelf smartphones for building interior reconstructions. The prototype is readily deployable in real-world scenarios. As our future work, specific issues related to a crowdsourcing deployment, such as user recruitment, incentive mechanism and privacy preservation, will be further focused on. Once fully hardened, IndoorCrowd2D is expected to provide indoor panorama and geo-data for each individual floor of any building around the world. IndoorCrowd2D is expected to extend existing online map services to the indoor environments at an unprecedented scale, which is currently cost prohibitive. IndoorCrowd2D can also serve an important stepping stone towards the ultimate goal of economically-viable massive indoor 3D model reconstruction.

10. ACKNOWLEDGEMENT

We thank the helpful comments from Fan Ye and the anonymous reviewers. We are grateful to David Chu for shepherding our paper. This work was supported in part by US National Science Foundation under grants CNS-1421903, CNS-1318948, and CNS-1262275.

11. REFERENCES

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] M. Alzantot and M. Youssef. Crowdinside: automatic construction of indoor floorplans. In *SIGSPATIAL GIS*, 2012.

[3] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide area localization on mobile phones. In *ISMAR*, 2009.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[5] Y. Bok, Y. Hwang, and I. S. Kweon. Accurate motion estimation and high-precision 3d reconstruction by sensor fusion. In *Robotics and Automation*, 2007.

[6] J.-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm.

[7] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[8] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[9] S. Chen, M. Li, K. Ren, and C. Qiao. Crowdmap: Accurate reconstruction of indoor floor plan from crowdsourced sensor-rich videos. In *ICDCS*, 2015.

[10] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Mobicom*, 2010.

[11] J. Chon and H. Cha. Lifemap: A smartphone-based context provider for location-based services. *IEEE Pervasive Computing*, (2):58–67, 2011.

[12] J. Coughlan and A. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *ICCV*, 1999.

[13] M. F. Fallon, H. Johannsson, J. Brookshire, S. Teller, and J. J. Leonard. Sensor fusion for flexible human-portable building-scale mapping. In *IROS*, 2012.

[14] A. Furlan, S. Miller, D. G. Sorrenti, L. Fei-Fei, and S. Savarese. Free your camera: 3d indoor scene understanding from arbitrary camera motion. In *BMVC*, 2013.

[15] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *CVPR*, 2009.

[16] Y. Furukawa, B. Curless, S. M. Seitz, and R. szeliski. Reconstructing building interiors from images. In *ICCV*, 2009.

[17] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010.

[18] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 32(8):1362–1376, 2010.

[19] 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.

[20] GoogleStreetView. https://www.google.com/maps/views/streetview.

[21] Y. Jiang et al. Hallway based automatic indoor floorplan construction using room fingerprints. In *Ubicomp*, 2013.

[22] S. H. Kim and others. Key frame selection algorithms for automatic generation of panoramic images from crowdsourced geo-tagged videos. In *WWGIS*. 2014.

[23] 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.

[24] T. Liu et al. Indoor localization and visualization using a human-operated backpack system. In *IPIN*, 2010.

[25] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Biological Sciences*, 208(1173):385–397, 1980.

[26] R. A. Newcombe et al. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011.

[27] D. Philipp et al. Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data. In *PerCom*, 2014.

[28] ProjectTango. https://www.google.com/atap/projecttango.

[29] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: zero-effort crowdsourcing for indoor localization. In *Mobicom*, 2012.

[30] V. Raychoudhury, S. Shrivastav, S. Sandha, and J. Cao. Crowd-pan-360: Crowdsourcing based context-aware panoramic map generation for smartphone users. *Parallel and Distributed Systems, IEEE Transactions on*, 26(8):2208–2219, Aug 2015.

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

[32] L. Ruotsalainen, H. Kuusniemi, and R. Chen. Heading change detection for indoor navigation with a smartphone camera. In *IPIN*, 2011.

[33] A. Sankar and S. Seitz. Capturing indoor scenes with smartphones. In *UIST*, 2012.

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

[35] H. Shin, Y. Chon, and H. Cha. Unsupervised construction of an indoor floor plan using a smartphone. *ITSMC*, 2012.

[36] N. Snavely, S. M. Seitz, and R. szeliski. Photo tourism: exploring photo collections in 3d. *TOG*, 2006.

[37] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2):189–210, 2008.

[38] P. Steadman. Why are most buildings rectangular? *Architectural Research Quarterly*, 10(02):119–130, 2006.

[39] 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.

[40] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3d reconstruction on mobile phones. In *ICCV*, 2013.

[41] TRXSystems. http://www.trxsystems.com/.

[42] 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.

[43] C. Wu, Z. Yang, Y. Liu, and W. Xi. Will: Wireless indoor localization without site survey. *TPDS*, 24(4):839–848, 2013.

[44] J. Xiao and Y. Furukawa. Reconstructing the world's museums. In *ECCV*. 2012.

[45] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013.

[46] J. Xiong and K. Jamieson. Arraytrack: a fine-grained indoor location system. In *Usenix NSDI*, 2013.

[47] C. Zauner. Implementation and benchmarking of perceptual image hash functions. 2010.

[48] X. Zhang, Z. Yang, C. Wu, W. Sun, and Y. Liu. Robust trajectory estimation for crowdsourcing-based mobile applications. 2013.

[49] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao. Travi-navi: Self-deployable indoor navigation system. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 471–482. ACM, 2014.

[50] Y. Zheng, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, 2009.