This document is downloaded from DR-NTU (https://dr.ntu.edu.sg)
Nanyang Technological University, Singapore.

VisioMap : lightweight 3-D scene reconstruction toward natural indoor localization

Li, Feng; Hao,Jie; Wang, Jin; Luo, Jun; He, Ying; Yu, Dongxiao; Cheng, Xiuzhen

2019

Li, F., Hao, J., Wang, J., Luo, J., He, Y., Yu, D. & Cheng, X. (2019). VisioMap : lightweight 3-D scene reconstruction toward natural indoor localization. IEEE Internet of Things Journal, 6(5), 8870-8882. https://dx.doi.org/10.1109/JIOT.2019.2924244

https://hdl.handle.net/10356/148581

https://doi.org/10.1109/JIOT.2019.2924244

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: https://doi.org/10.1109/JIOT.2019.2924244.

Downloaded on 28 Aug 2022 07:57:10 SGT

VisioMap: Lightweight 3D Scene Reconstruction towards Natural Indoor Localization

Feng Li, *Member, IEEE,* Jie Hao, *Member, IEEE,* Jin Wang, *Member, IEEE,* Jun Luo, *Member, IEEE,* Ying He, *Member, IEEE,* Dongxiao Yu, *Member, IEEE,* Xiuzhen Cheng, *Fellow, IEEE*

***Abstract*—Most existing proposals for indoor localization are "unnatural", as they rely on sensing abilities not available to human beings. While such a mismatch causes complications in human-computer interactions and thus potentially reduces the usability and friendliness of a localization service, it is partially entailed by the need for low-cost/effort sensing with resource-limited mobile devices. Fortunately, recent developments in smart glasses (e.g., Google Glasses) signal a trend towards realistic visual sensing and hence make the sensing ability of mobile devices more compatible to that of human users. Leveraging such front-end developments, we propose VisioMap as a *natural* indoor localization system that intentionally mimics the human skills in visual localization. VisioMap uses very sparse photo samples to reconstruct 3D indoor scenes; this is facilitated by the facts that photos are taken at the eye-level with high stability and regularity, and that the reconstruction is lightweight as it exploits geometric features rather than image pixels. Localization is in turn performed by matching the geometric features extracted online to the reconstructed 3D scene, making VisioMap i) natural to users as they can see the matched 3D scene, and ii) dispensed with the need for dense fingerprints/POIs towards accurate localization.**

***Index Terms*—Indoor localization, 3D scene reconstruction, floor plan generation, smart glasses.**

I. Introduction

Stimulated by the high demands (from, e.g., *mCommerce* [31]) for human indoor positions, indoor localization has been attracting extensive interests for more than a decade and has led to a huge amount of developments [14], [30]. Unfortunately, no full-fledged deployment has been derived from these proposals by far. The main reason, beside unsatisfactory localization accuracy and lack of floor plans, could be the incompatible sensing/presentation ability between mobile devices used for localization and their human users. In particular, meter-level errors are normal for fingerprint-based and AoA-based schemes (e.g., Horus [40] and CUPID [28]) while ranging-trilateration methods (e.g., EZ [7]) can be worse. Besides, although the floor plan can generated by, e.g., the labor-consuming crowdsensing method [10], [43], [5], [4], [38], [23], the noise introduced by the inherent errors in individual crowd-sensed components and in assembling them together may result in ambiguity and further confuse users.

Furthermore, the information delivered by existing localization systems (i.e., pin-pointing a user location on a given floor plan) is almost identical to the widely used plain *you-are-here* map. Since people often find it is difficult to make the connection between a 3D structure and its 2D projection, such a map fails to really guide a user under many circumstances due to the mismatch between its 2D representation and human users' 3D visual ability. Therefore, the location indicator suggested by the localization system may be frustrating to users. Hence, some proposals (e.g., [32], [38]) suggest using photos of close-by *point-of-interests* (POIs) for location estimation. This approach provides friendly localization service, but significantly increases the complexity in system operation during both initialization and run-time phases, due to its dependence on intensive survey and image processing.

In reality, human beings are used to applying their 3D vision to figure out locations in a natural manner, but 3D visual sensing (i.e., taking, processing and assembling photos) is rarely adopted in existing proposals due to its high computational cost, as well as its high demand on the user effort such as shooting at a right angle and avoiding hand tremors. Fortunately, recent developments on smart glasses [1] have paved the way towards more convenient visual sensing in mobile devices, since the effort users have to make in shooting photos can be reduced to the largest extent thanks to the upright and stably positioned camera (as will be shown by our extensive experiments in Sec. III-A). Also, the camera is almost aligned with the user's sight line, which considerably improves the naturality in the procedures of map generation and localization. Nevertheless, smart glasses are usually equipped with very limited resources (e.g., Google Glass carries a 570 mAh lithium-polymer battery, while a normal smart phone, such as Samsung S4, has a battery capacity of 2600 mAh), it is highly *non-trivial* to design sufficiently light-weight algorithms for adapting 3D vision sensing to the smart devices.

In this paper, we promote what we term *natural localiza-*

This work is partially supported by NSFC (Grant No. 61702304, 61832012, 61602195 and 61771289), Shandong Provincial Natural Science Foundation, China (Grant No. ZR2017QF005), and AcRF Tier 2 Grant MOE2016-T2-2-022.

F. Li and D. Yu are with School of Computer Science and Technology, Shandong University, Qingdao, China. Email: {fli, dxyu}@sdu.edu.cn.

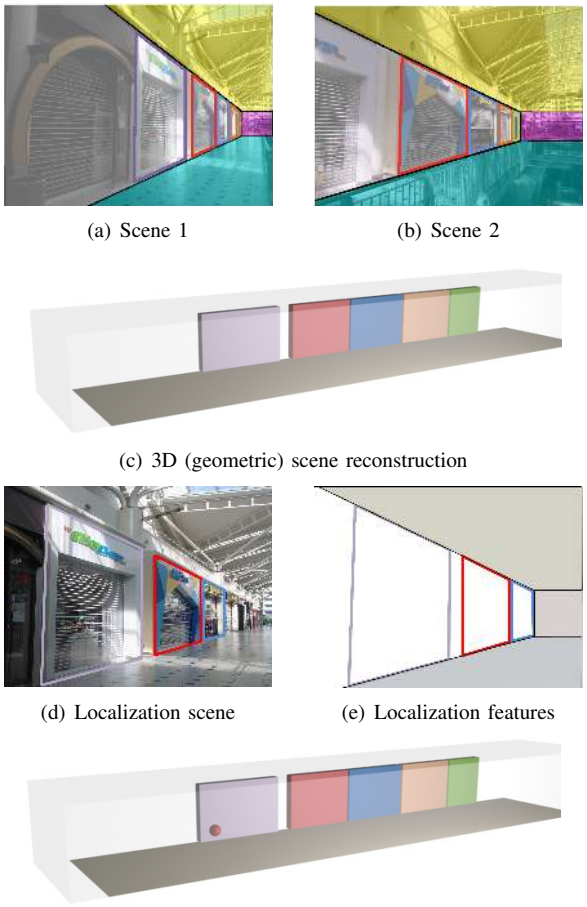
Jie Hao (corresponding author) is with College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. E-mail: haojie@nuaa.edu.cn.

J. Wang, J. Luo and Y. He are with School of Computer Science and Engineering, Nanyang Technological University, Singapore. E-mail: {jwang033, junluo, yhe}@ntu.edu.sg

X. Cheng are with School of Computer Science and Technology, Shandong University, Qingdao, China, and Department of Computer Science, The George Washington University, Washington DC, USA. Email: xzcheng@sdu.edu.cn.

[1]For example, as the most remarkable smart wear device in recent years, smart glasses (e.g., Google Glass and Vuzix M100 Smart Glass) have received increasing attentions in many emerging areas and various applications [26], [35], [33], [42], [39].

tion,[2] in the sense that we try to unify the sensing (hence presentation) ability of a localization system with that of human users. Specifically, we intend to build a system in which both floor plan generation and location estimation are done mainly through visual sensing. As such a system closely mimics the location identification ability of human beings, it provides a location indicator in the form of *scene* rather than *point*, delivering an immediate sense of location to users. Also, such an indicator is insensitive to localization errors: as it is adapted to human vision, meter-level errors can be easily corrected by users. Moreover, we reconstruct indoor 3D scenes using geometric features extracted from images; hence, the resulting floor plan contains far more information for localization than a commonly used 2D plain map, and it avoids errors mostly caused by applying inertial sensing in the existing proposals [10], [43]. Also, the smart glass provides a friendly visual interface, through which we can provide feedbacks to user to guide scene survey. We illustrate the basic idea of our natural localization in Fig. 1.



(a) Scene 1 (b) Scene 2

(c) 3D (geometric) scene reconstruction

(d) Localization scene (e) Localization features

(f) Location indicator in 3D

Fig. 1. A 3D scene (c) is reconstructed based on two photos (a) and (b) (as well as their rectilinear features). During the localization phase, a new photo (d) is taken and its features are extracted (e) and matched to the 3D scene, resulting in an accurate location indicator (f).

To realize the aforementioned ideas, we hereby present VisioMap as an infrastructure-free localization system that mainly applies visual sensing for both floor plan generation and localization. Leveraging the high quality visual sensing of smart glasses and efficient computer vision algorithms, VisioMap infers geometric information (including dimensions and patterns) of an indoor scene from very sparse image samples. Consequently, the computational cost is reduced to the largest extent, so that one person wearing a smart glass and equipped with a smart phone can accomplish 3D scene reconstruction for a large indoor space within an hour. Our system totally frees us from the burden of shooting thousands photos and hence eliminates the need for the noise-prone crowdsensing. The extracted geometric information, after being used for scene reconstruction, is also stored in a database as fingerprints for later localization. These fingerprints are more natural than, say, WiFi fingerprints to human visions and are far more available than dense POIs. In summary, our contributions in VisioMap are as follows:

- We design an indoor localization system that reconstructs 3D scenes as its floor plans and adopts the geometric information contained in these scenes as localization fingerprints.
- We engineer VisioMap to fully exploit the power of smart glasses in assisting both lightweight floor plan generation and natural localization.
- We extend the existing computer vision algorithms for VisioMap to infer various geometric information from sparsely sampled images.
- We implement VisioMap with both Google Glass and smart phone, and perform extensive experiments on it in various indoor spaces. The results strongly confirm the usability of VisioMap as the first prototype of natural indoor localization.

For the purpose of indoor localization, VisioMap is, in several aspects, superior to both visual SLAM [8] and SfM [29] (the latter is adopted by Jigsaw [10] and SnapTask [23] for floor plan generation). First of all, VisioMap requires very sparse image samples for map generation and localization, whereas both SLAM and SfM demand a high image sampling rate to acquire a huge amount of photos. Secondly, VisioMap is computationally lightweight in scene reconstruction as it focuses on geometric features rather than point clouds. Thirdly, VisioMap adopts highly available geometric information as fingerprints for localization, as opposed to SLAM's reliance on point features inapplicable to indoor localization using resource-constrained mobile devices.

The rest of our paper is organized as follows. We first briefly describe the VisioMap architecture in Sec. II. Then we present the design rationales and technical details of VisioMap in Sec. III and IV. The extensive evaluations on VisioMap are reported in Sec. V. We survey related literature in Sec. VI, and finally conclude our paper in Sec. VII.

II. System Overview

As demonstrated in Fig. 2, VisioMap mainly consists of five hardware/software components: smart glasses, smart phones, *Feature Extraction Module* (FEM), *Scene Assembly Module*

[2]In coining this term, we draw inspiration from the well-known term *natural language* where "natural" distinguishes the communication ability of human from that of machines.

(SAM), and *Scene Matching Module* (SMM). The first three serve the goals of both floor plan generation and localization, and SAM and SMM are dedicated to these two functionalities respectively. Readings from both inertial sensors and WiFi radio are collected at the background; they assist scene reconstruction and also act as supplementary location fingerprints.

A. Sensing and Presentation Interfaces

VisioMap combines the sensing and presentation capabilities of both smart glasses and smart phones to produce information in a straightforward and natural manner. Leveraging the upright and stable positions of smart glasses' cameras, VisioMap is able to obtain high quality visual and inertial sensing data (see Sec. III-A for details). Moreover, the matched scenes can be displayed on the smart glasses for a user to visually judge the correctness of each location estimation. To alleviate the overhead of the smart glasses, we employ smart phones serving as computing and data relaying platforms. Finally, the reconstructed 3D (geometric) scenes can be displayed on the phone screen, giving users a natural sense of the environment and location.

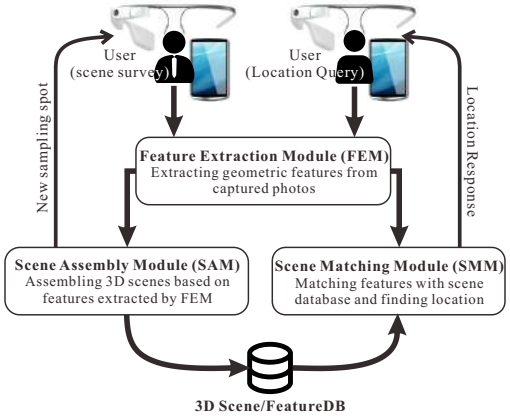


Fig. 2. VisioMap architecture

B. FEM: Extracting Typical Indoor Features

It is well known that rectilinear geometric structures most commonly appear in indoor environments; such structures may include doors, windows, display/sign boards, and even the whole side of a wall. FEM aims to extract such structures as the features of scenes. In fact, not only the individual rectangular shapes can act as features, but the combination of a set of rectangles, along with their relative positions in the scene, can also serve as a rather unique feature. FEM further processes the pixels within each rectangle to enrich the feature space [24], better guaranteeing the discriminability of applying such feature vectors as fingerprints for scenes. In fact, these scene fingerprints are universally available for any locations indoors, so using them for discriminating locations frees us from the reliance on particular POIs. In addition, FEM measures the dimensions of the indoor structures, e.g., the length and width of a hallway, by combining computer vision techniques [16] with the relatively accurate inertial sensing performed by smart glasses. The outcome serves as an input to SMM so that the 3D scene reconstruction can be conducted in a geometric manner.

C. SAM: Assembling Scenes Geometrically

As the output of FEM for a given scene includes a set of line segments (the basis of recognizing rectilinear structures) and the corresponding dimensions, a 3D geometric model can be constructed out of it readily. Moreover, assembling two scenes is made easy by simply concatenating the line segments belonging to the same lines. As shown in Fig. 2, VisioMap invokes FEM and SAM in an iterative manner: while making progress in scene reconstruction by taking input from FEM, SAM also feedbacks to the smart glass so that the human user is guided to proceed to the next image sampling spot, allowing FEM to acquire a new scene. The iteration between FEM and SAM ends with the user exploring the whole indoor space and results in a set of features extracted by FEM along with their relative positions in 3D, i.e., a 3D geometric scene.

SAM refrains from using optimization approaches for assembling a floor plan [10], [26] since high computational cost would be induced. Instead, it applies a light-weight geometric relaxation method to handle the mismatches: it preserves the intrinsic linearity in indoor scenes and avoids producing floor plans with artificially twisted hallways as often resulted from the existing approaches [1], [43], [26], [10].

D. SMM: Scene-Base Localization

During the localization phase, a user again uses smart glass to take a photo and has it processed by FEM. The output is posted as a location query to SMM that in turn attempts to match the incoming feature vector with the reconstructed scene. To facilitate the matching procedure, VisioMap organizes features into FeatureDB and applies both a B-tree and an R-tree [19] to index them. The matching algorithm recursively goes through these indices in order to narrow down the region potentially containing the user's location. Though VisioMap can also use other fingerprints (e.g., WiFi fingerprints collected by mobile phones) to help confining the search region for accelerating the matching, we refrain from discussing this supplement as it may not be always feasible. Different from the POI-based visual localization schemes, SMM exploits scene features that are more general than POIs and a location query can be processed wherever it is issued.

III. Reconstructing 3D Scenes Geometrically

In this section, we explain how VisioMap performs scene reconstruction while effectively controlling its computational complexity. We first explain in Sec. III-A why we choose smart glasses from a technical perspective, and then discuss how we extract features from images and construct 3D scenes accordingly in Sec. III-B and III-C, respectively.

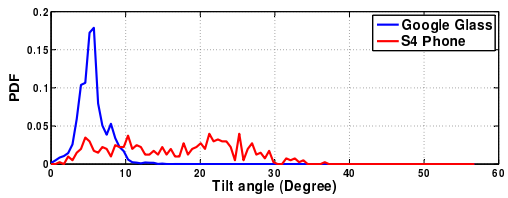
A. Why Smart Glasses?

Visual sensing is not new to mobile computing, but applying it to indoor localization is very recent. We attribute this to the inconvenience and intrusiveness of visual sensing by phone cameras, and we explain how the introduction of camera-equipped smart glasses eliminates these weaknesses while bringing further benefits. As mentioned in Sec. I, our prototype system is built with Google Glass equipped with a 5-megapixel camera. The resolution of each photo is 528×1856.

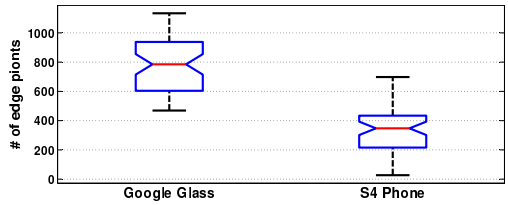
1) Stronger Guarantees on Photo Quality: Conventional computer vision based scene reconstruction often requires either a large number of photos to be taken for a single POI [29], [10] or a video with high frame rate to be recorded [9]. Such a requirement more or less reduces the dependence on the quality of individual images, as the information loss caused by one low-quality image can be compensated by others. However, to make VisioMap a lightweight mobile system, we have more stringent demands on individual photos that may not be satisfied by smart phones. Fortunately, smart glasses do offer a higher quality photo shooting than smart phones, mainly in the following aspects:

- Better Guarantee on Upright Perspective: Arbitrary shooting gestures and/or shaky hands can cause the photo perspective be tilted or skewed, resulting in troubles for feature extraction. Mounting cameras on users' heads allows smart glasses to largely avoid these problems.
- Higher Stability during Shooting: Shooting photos indoors with dim light results in a relatively long exposure time and thus a higher chance to blur photos. Braced by human heads (rather than shaky hands), smart glasses produce photos with far less blurring, which facilitates feature detection.

We ask a few users to arbitrarily choose 100 sampling spots and shoot two photos with Google Glass and a Samsung S4 phone, respectively. We compute the tilt angle of each photo and plot the distribution of these angles in Fig. 3 (a). As blurring photos tend to have less detectable edges, we compare the two set of photos in terms of their detectable edge points in Fig. 3 (b). It is shown that Google Glass performs much better on both aspects than the mobile phone.



(a)



(b)

Fig. 3. Comparing smart glasses and smart phones in shooting performance.

2) Higher Accuracy in Inertial Sensing: Although inertial sensing has been widely used in indoor localization, it is well known to be error-prone. In particular, as a smart phone can be in any position on a human body with various attitudes and its position may keep changing even if the body remains static, it causes unpredictable errors in measuring, e.g., turning angles [36], which has forced several localization systems to require unrealistically that users have to hold smart phones in fixed positions (e.g., [43]). Nevertheless, this is not a problem for smart glasses, as human heads has very restricted movements mostly within the horizontal planes even when the bodies are moving, resulting in high quality inertial sensing. In Fig. 4, we compare them in term of accuracy in angle detection, and the results clearly show that the smart glasses induce very small errors.

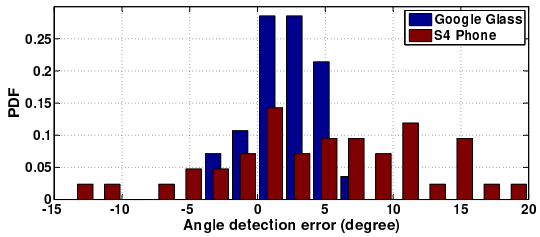


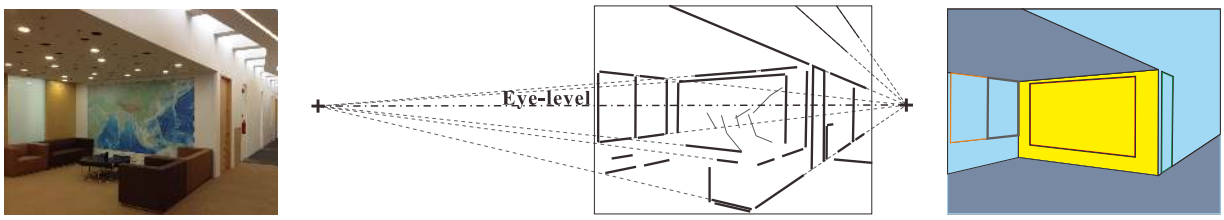
Fig. 4. Comparisons in terms of angle detection.

3) Reference for Scene Dimensioning: Dimensioning is an important aspect in generating floor plans, as otherwise individual sensing patches could not be easily assembled. Some of existing approaches rely on point clouds sampled along overlapped motion traces for a rough dimensioning (with the stride length as the default unit of length) [1], [43]. The above "footprints" can be further complemented by computer vision techniques (e.g., SfM [29]) that require plenty of photos and heavy computations [10]. In contrast, the camera of a smart glass sits at a known height: the eye-level of a certain user. This reference height gives us an extraordinary leverage in dimensioning a 3D scene with even one photo, as we will demonstrate in Sec. III-B3.

B. Extracting Features from Images

Feature Extraction Module (FEM) of VisioMap takes images captured at individual sampling spots as input and recovers (local) scenes by extracting their geometric features.

1) Detecting Line Segments and Vanishing Points: Given a sampled image shown in Fig. 5(a), we first detect the line segments using the state-of-the-art LSD algorithm [34]. We filter out the very short ones in order to i) avoid unnecessary computations, and ii) alleviate interference incurred by obstacles (as such objects normally produce trivial short line segments). We adopt a "single-floor-single-ceiling" assumption, which is commonly used in indoor scene modeling [16] and is also applied in existing state-of-the-art indoor localization systems, e.g., [10]. In this model, a 3D indoor scene is composed of only one floor and ceiling as well as multiple walls. These planes are represented by their respective boundary lines that are in turn classified according to three *vanishing points* (VPs). We follow [16] to detect the VPs in RANSAC manner and identify the ceiling-wall and floor-wall boundary lines. Thanks to the upright shooting angle of smart glasses, we have the advantage of detecting only two horizontal VPs, while the vertical one can be set to infinity. Using these VPs, each line segment is classified into one of the three types according to the VP it vanishes at, or is eliminated if it does not vanish at any VP. Then it is labeled by a triple $L = (A, B, K)$, where



(a) Photo by smart glasses (b) Detected line segments and VPs (c) Recognized features

Fig. 5. The workflow of VisioMap's Feature Extraction Module (FEM)

A and B are the two end points and $K = 1, 2, 3$ indicates the type. We illustrate these results in Fig. 5(b).

2) Recognizing Rectilinear Structures: With all line segments detected and classified, there exist generic algorithms for recognizing rectilinear structures (e.g., [22]). However, the algorithms are not adapted to indoor scenes and hence incur rather high computational costs. We hereby propose a light-weight heuristic algorithm to identify the rectilinear structures for indoor scenes. Our heuristic again relies on the VPs. We build a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of all detected line segments $\{L_i = (A_i, B_i, K_i)\}$ and $\mathcal{E}$ is a set of edges representing the neighboring relationship between the line segments. Two line segments are said to be neighbors if they satisfy one of the following conditions: 1) Crossing Lines: vanishing at two different VPs, and the sum of the shortest distances between the ends of each line and the line intersection is within a pre-defined threshold τ_c, and 2) Collinear Lines: vanishing at the same VP, difference in slopes is within threshold τ_s, and the shortest distance between their ends is within distance threshold τ_d. If L_i and L_j are neighbors, we add an edge $e = (i, j, P)$ to $\mathcal{E}$ with P being the intersection of L_i and L_j. We empirically set $\tau_c = \tau_d = 4$ in pixels, and $\tau_s = 0.03$.

In a given $\mathcal{G}$, we heuristically recognize rectilinear structures through an adapted depth-first-search. In each iteration, the algorithm uses a state machine with 4 states corresponding to the 4 edges of a rectangle. Note that, each rectilinear structure in our model is associated with two of the VPs (one is horizontal and the other one is vertical), and adjacent edges are associated with different ones. The initial state is chosen as a line segment vanishing at the vertical (virtual) VP and we initialize the search by going upwards. Once we search a line segment whose VP differs from the current one, the state machine proceeds to the next state. Each iteration succeeds once it reaches the fourth state and finds the starting line segment as a neighbor. To prevent never-end loops in the search, a threshold for the number of line segments that form a rectangle is set in advance (we empirically set it to 8). To make the algorithm robust to the broken wall-floor intersection lines (e.g., due to doors or obstacles), a search failing to proceed to the fourth state is allowed to be completed with an artificial wall-floor line (i.e., a line segment which vanishes at the associated horizontal VP and connects the starting point in the first state and the end point in the third state), if the Crossing Lines condition is not violated. The successful search returns the set of line segments $\mathcal{R}$ that form a rectangle. We iteratively conduct this for all the line segments vanishing at the vertical VPs in attempt to recognize all the rectangles on the wall planes, as shown in Fig. 5(c). Although our heuristic entails only marginal complexity, we can sketch the whole geometric (i.e., rectilinear) structures in the indoor scene.

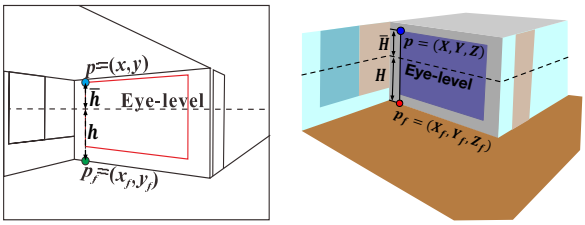
In order to increase the number of detectable rectilinear structures (especially in face of obstacles), we require a user to shoot up to three photos at the same spot by panning the camera. Leveraging the accurate gyroscope sensing of pan angles explained in Sec. III-A, FEM directly performs perspective transformation to merge the detected features; this significantly reduces the computational complexity compared with computer vision approaches that perform pixel-level reasoning [29], [9].

3) Dimensioning and "Rendering" a 3D Scene: Given the extracted features and recognized rectangles, a 3D geometric scene is logically rebuilt. However, in order to facilitate the overall scene reconstruction, each scene has to be properly dimensioned. To this end, we need to "translate" the dimension (in pixels) of a 2D photo to their corresponding 3D coordinates. As mentioned in Sec. III-A3, we reconstruct the 3D scene according to only one photo, by taking the height of Google Glass (with respect to ground) as reference.

Let us take a particular point p with 2D coordinates (x, y) in Fig. 6 as an example. Since the rectilinear features we concern in VisioMap are on the walls, we take into account only the vertices of the detected rectilinear structures. According to the perspective effect, its 3D coordinates (X, Y, Z) can be determined by

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi & 0 & 0 \\ 0 & \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}. \quad (1)$$

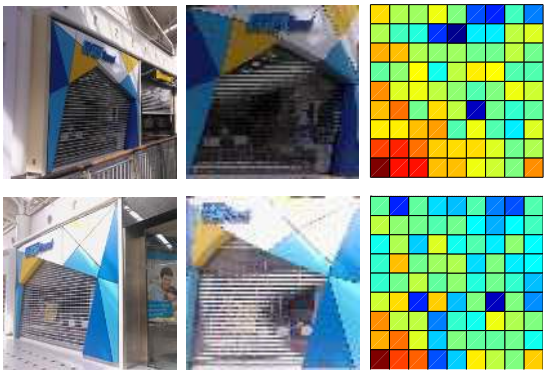
where ϕ is a camera parameter that needs to be calibrated in an off-line fashion only once. Since the 2D coordinates x and y can be obtained directly from the photo, we now have two equations but three unknowns (i.e., the 3D coordinates X, Y and Z); that is an under-determined system. We denote by p_f the projection of p on the floor. In the 2D photo, p_f is the intersection point of the vertical line going through p and the wall-floor boundary line of the plane containing p. Since p and p_f are on the same vertical line, we have $X = X_f$ and $Z = Z_f$. Recalling that the camera of the smart glass is usually fixed beside the user's eyes, the height of the user's eye-level H is known (as explained in Sec. III-A3). Hence, we have $Y_f = -H$, if we deem the eye-level as Y-coordinate in the 3D scene (or y-coordinate in the 2D photo) 0 . Solving the equation system gives us X_f and Z_f (and thus X and Z). Finally, since $\bar{H}/H = \bar{h}/h$, we can calculate $Y = \bar{H}$, according to $\bar{h}$ and h known in the photo.



(a) 2D (b) 3D

Fig. 6. Dimensioning a 3D scene. The eye-level is determined in the photo by the two horizontal VPs.

To improve the discriminability for localization and to avoid scene rendering that would heavily load a mobile device, we use Discrete Cosine Transformation (DCT) to produce a *region descriptor* [22], which can be used to describe the detected rectilinear image patch. In fact, DCT works well in resource limited mobile devices, since i) computing DCT is quite efficient and fast with a low time complexity of $O(n \log n)$, and ii) robustness to frame misalignment can be achieved by storing only low-frequency coefficients. In particular, the recognized rectilinear image patch is first normalized to have the same size (80×80), zero mean, and unit variance. This normalization procedure helps alleviating the impact of illumination variety, out-of-focus blur, etc. Then DCT is applied on the normalized image patch to obtain the descriptor as the 10×10 low frequency coefficients (see Fig. 7).



(a) Original (b) 80×80 (c) DCT

Fig. 7. The same doorway viewed from different angles (a) are normalized to 80×80 squares (b), and their DCT coefficients (c) are similar.

C. Assembling 3D Scenes

With features extracted from individual photos and the corresponding reconstructed 3D sub-region models, Scene Assembly Module (SAM) attempts to build a 3D model of the entire indoor space by interacting with users on-line, thanks to the friendly vision-based interface between the users and the smart glasses. Since an indoor space is usually composed by hallways, we focus on reconstructing the hallway system, especially considering it is the most crucial part for both localization and navigation.

1) Iterative Photographing and Progressing: VisioMap mimics human's natural behavior in exploring a certain space. Specifically, each time SAM processes a new input scene from FEM and extends the scene database, it returns a feedback to the user in the form of a picture displaced on smart glass, and this picture is one of the photos (or a transformed version if necessary) overlaid with the detected rectilinear structures, as shown in Fig. 8. Based on this feedback, VisioMap directs the user to proceed towards the next sampling spot: right before the furthest detected feature (the green frame shown in Fig. 8). Upon reaching that spot, the same feature extracting and scene extending procedure get repeated, and another feedback/guidance will be issued, driving the scene reconstruction to make progress till the end. Determining the termination of the reconstruction process does require a bit of user's subjective intervention, but this is needed for any indoor floor plan generation method.

Fig. 8. A feedback on Google Glass display. We mark the rectilinear structures detected in the current scene with red and green colors, while the green frame can be employed to guide the users to proceed.

2) Extending Scenes by Geometric Concatenation: Since users take photos according to the feedbacks mentioned above, the photos taken at a sampling spot has at least one overlapped feature with those taken in the previous spot. In other words, the 3D scenes reconstructed from consecutive photos share at least one common rectilinear feature, although they may be under different coordinate systems. According to [11], SAM calculates a transformation, by applying which, the two consecutive 3D scenes can be assembled such that the overlapped rectilinear features are matched with each other. The transformation is rigid and thus is easy to calculate, since the scenes have been properly dimensioned as shown in Sec. III-B. Furthermore, as one rectangle correspondence between two photos is sufficient to calculate the relative pose and merge the two scenes [11], such an assembling method does not compromise the sparsity of the photo samples. Also, we extend the wall-floor and wall-ceiling intersection lines as follows to make the 3D model more completed. If the two consecutive scenes are captured within a hallway, they should share the same wall-floor and/or wall-ceiling intersection lines, which VisioMap connects to extend the scene. If the hallway is turning, VisioMap by default sets the turning angle to be 90° unless the gyroscope reading shows a significant deviance, in which case the turning angle can be accurately measured using the features detected earlier (e.g., the eye-level). We illustrate an 3D scene reconstruction example relying on a few regularly spaced photos in Fig. 9.

3) Global Dimension Unification: One major issue with VisioMap's simple geometric scene concatenation is potential mismatches in case of route loop. As errors exist in dimensioning individual scenes, the error accumulation may cause the starting and ending point of a loop missing each other. Fortunately, such errors are systematic as they are caused by the errors in detecting VPs, as opposed to those

Fig. 9. Assembling a 3D scene from photos. As VisioMap only reconstructs the scenes of the hallway system where the detected features are actually facing inside, while the outside spaces are rooms whose scenes are not reconstructed for brevity.

random errors caused by using inertial sensing for measuring length [17]. As we assume that the vertical VP is at infinity given smart glasses' upright shooting angle, the estimated horizontal VPs can be either above or below the actual eye-level if there exist slight tilting, which in turn causes either overestimation or underestimation in dimensioning. Instead of correcting such errors for individual scenes, VisioMap records the tilt angle for each scene. Upon a mismatch, scenes have their length adjusted according to their respective tilt angels. This computationally efficient method effectively preserves the default linearity of hallway systems.

IV. Localization by Matching Geometric Features

VisioMap performs localization by again calling FEM to extract features from a photo sampled right at the to-be-located spot, then it searches the scene database to identify the best match. The focus of this section is to explain how VisioMap's Scene Matching Module (SMM) organizes the scene database so as to efficiently answer location queries. In this section, we present specifically designed indexing approach and query processing method.

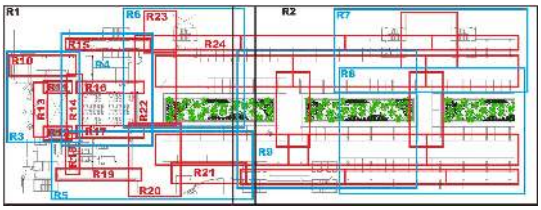
A. Indexing the Spatial Database

SMM stores all rectilinear features as records in FeatureDB. As shown in Fig. I, each item (i.e., feature) consists of three fields: aspect ratio, geographic location and region descriptor. The aspect ratios and geographic locations can be directly obtained from the reconstructed 3D scene, and the region descriptors are calculated based on DCT, as shown in Sec. III. For each rectilinear feature, its aspect ratio is unique under certain perspective effect; therefore, we organize all the recognized rectilinear features in the FeatureDB as a B-tree with respect to the aspect ratios, in order to gain efficiency in searching, especially in the face of large scenes. In particular, given a feature input, we search the B-tree firstly according to its aspect ratio, and then further compare it with the selected items in the FeatureDB based on region descriptors, and finally get the corresponding locations.

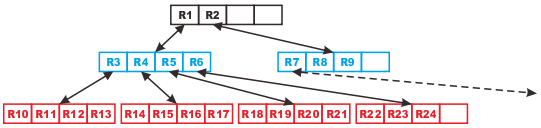
Nevertheless, the decoupled features do not have sufficient discernibility to locate users. We have to make better use of the spatial proximity relation among these features; the features captured in a scene are geographically close to each other. Hence, SMM adopts R-tree [19] for indexing all features in terms of their locations, especially considering most of the indoor plans can be well partitioned into rectangles. We choose to have a degree 4 for each R-tree nodes as each scene captured at a certain spot normal contains 1 to 4 features. Consequently, each leaf node contains up to 4 pointers to the close-by features in FeatureDB. Fig. 10 illustrates an R-tree for a floor plan. It is a partial construction, as we do not have space to show all nodes. Whereas R-tree is usually used for handling location-based queries with high efficiency (e.g., which features are close to a certain geographic location?), SMM uses it to retrieve nearby features. Therefore, our R-tree differs from a normal one by having bi-directional pointers, such that we can retrieve its parent node from a child node.

TABLE I
Records in FeatureDB.

Aspect ratio	Region descriptor	Location
0.32	D_1^r	$(10.5, 22.3)$
0.56	D_2^r	$(35.6, 18.0)$
2.85	D_3^r	$(38.1, 55.8)$
...	...	...



(a) Partitioning a floor plan into rectangular sub-regions.

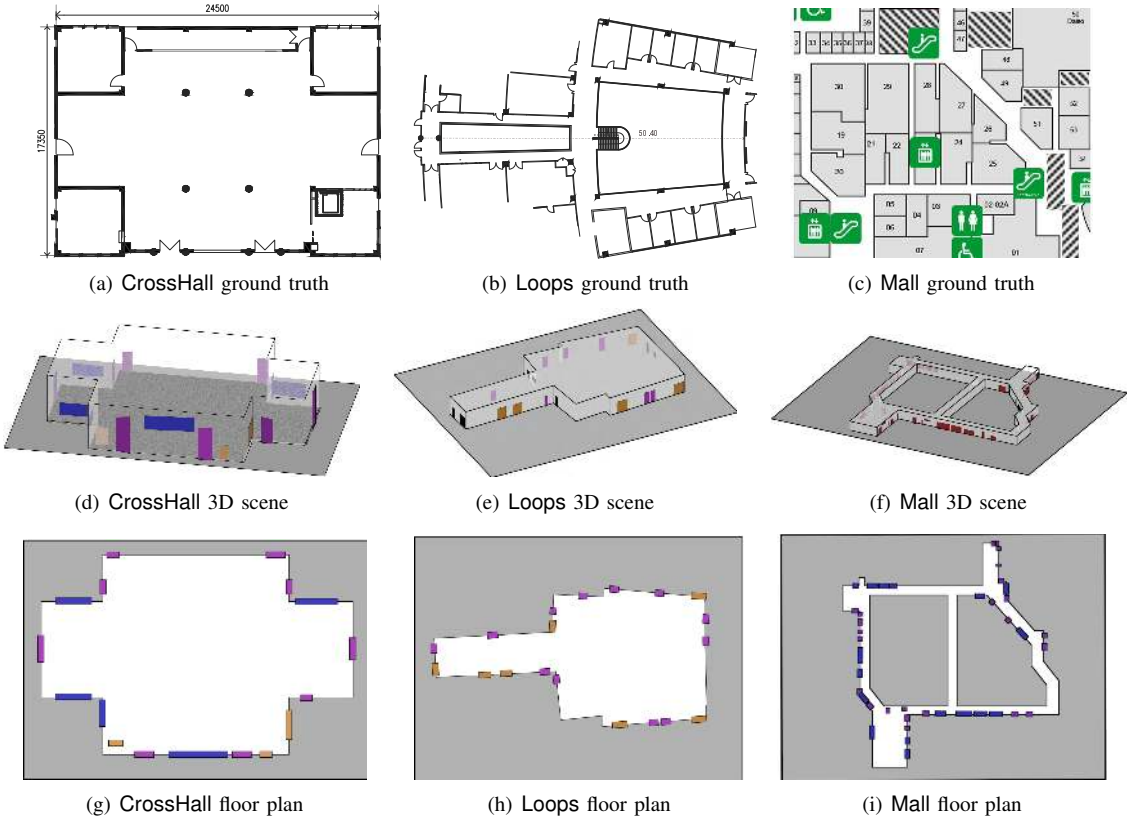


(b) R-tree representation of the partitions.

Fig. 10. An example of using R-tree for indexing a partitioned floor plan. Note that, these are rectangular areas on the 2D floor plan, having nothing to do with the rectilinear features detected earlier.

B. Location Query Processing

When a location query arrives with a set of newly extracted features, SMM conducts an ϵ-approximation nearest neighbor search in FeatureDB with respect to the first feature. Thanks to the B-tree indexing of the aspect ratio, the search procedure is rather efficient. The results of this step are a set of records enabling SMM to trace back to the leaf nodes of the R-tree, which in turn leads to a new table containing features that are geographically close to the first one. In fact, the table indicates a sub-region containing the user, and thus can be considered as a coarse estimation of the user's location. Our next step is to refine the sub-region by exploiting the spatial relations among the features. In particular, we recursively apply the above search procedure to the features one by one, and each recursion results in a smaller table containing the unmatched features geographically close to the input features that already have



(a) CrossHall ground truth (b) Loops ground truth (c) Mall ground truth

(d) CrossHall 3D scene (e) Loops 3D scene (f) Mall 3D scene

(g) CrossHall floor plan (h) Loops floor plan (i) Mall floor plan

Fig. 11. Visual demonstration of VisioMap's scene reconstruction.

been visited and matched. The recursion is terminated until either the newly derived table contains only one feature or the features carried by the query run out. The former case yields a sub-region containing exactly the features carried by the query, based on which SMM further exploits the known locations of the matched features to calculate the user's location. The latter case implies an ambiguity among multiple sub-regions, and SMM further uses the relative positions among the features to filter out non-matching cases.

In fact, the user's location can estimated by referring to only one rectilinear structure, as the aspect ratio is unique for each rectilinear structure under the perspective effect. To improve the robustness of the localization procedure, our method is to average multiple location estimations, each of which is based on one rectilinear structure. Since our localization approach can work with a handful of rectilinear features, the presence of obstacles does not impair the localization performance much, especially considering a scene captured in one shot usually involves multiple rectilinear structures.

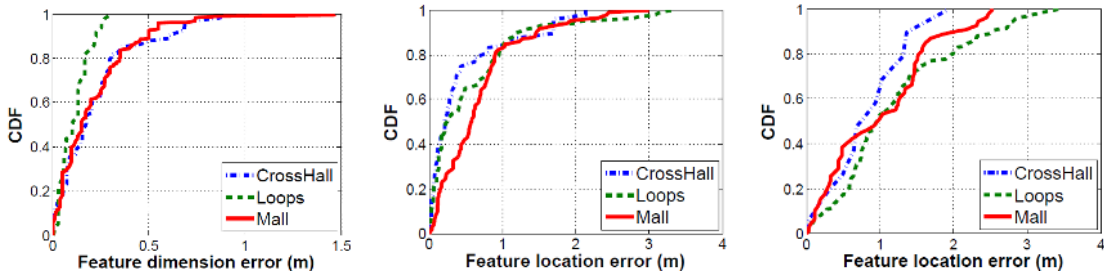
V. Performance Evaluation

A. Experimental Methodology

The basic equipments used in our experiments are two: one Google Glass and one Samsung S4 smart phone. As explained in Sec. II-A, they are mainly used as the sensing front-ends and user interfaces, respectively. Since the line segment detection is rather light in computation (see Sec. III-B), we let the phone to perform it. Furthermore, we offload the scene reconstruction to a back-end server, considering the required transmission between the phone and the server is not high given that only geometric features are to be conveyed. We also use a Bosch laser rangefinder DLE40 to obtain the ground truth distances and locations for all test sites. It is worth mentioning that our system is not platform specific and is compatible with other smart glasses (e.g., Vuzix M100 smart glass which is also equipped with a 5-megapixel camera and has similar computation and communication modules) and smart phones.

Experiments are performed for both 3D scene reconstruction and localization. The former is conducted by two of the volunteers, with one wearing the Google Glass and another holding the S4 phone for shooting photos at the same spots. While the photos shot by the Google Glass are used for scene reconstruction and localization, those shot by the phone are used for comparison purpose, already presented in Sec. III-A. The localization are tested by 12 users wandering around each site and shooting photos freely.

We have worked on several test sites including office buildings, libraries, and shopping malls during the past six months. In each site we repeat experiments every two weeks but at different time slots. This allows us to evaluate VisioMap under varying illumination, crowdedness, furniture positions, etc. Among all these sites, we choose to present three representative ones: CrossHall is a 24.50m$\times$17.35m hall of cross shape, Loops is a 50.40m$\times$40m storey of a university department building with two loops, and Mall is a 160m$\times$100m storey of the shopping mall. The number of sample spots for each site will be reported later: they are determined in an online manner as explained in Sec. III-C1.



(a) Feature dimension errors. (b) Feature location errors. (c) Hallway length errors.

Fig. 12. Numerically evaluating the performance of VisioMap's scene reconstruction.

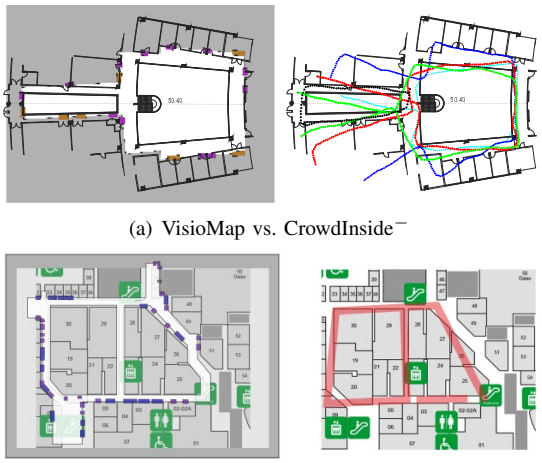
B. Performance of 3D Scene Reconstruction

We first graphically illustrate the scene reconstruction results for the three test sites in Fig. 11. These test sites are chosen such that, on one hand, we may have rather accurate ground truth data, and on the other hand, they represent diversified scenarios. CrossHall is selected to demonstrate that VisioMap works for room-like units even of a large scale, and hence we can focus only on the hallway system in other venues (where a room-level survey may not be possible). Then Loops is chosen to represent a hallway system with features available only on one side, as the loops are circulating atriums. Finally, the plan of Mall is very typical for relative large indoor spaces. We have 4, 14, 36 sample spots for these three venues, respectively, and majority of the spots require only one photo.

Although VisioMap does not illustrate the very detailed features for Loops and Mall at this zoom level, it is already clear that VisioMap faithfully reproduces the 3D scene structures and thus offers more features for location indications than standalone POIs (e.g., doorways or elevators). Also, the 3D presentations are far more natural to human eyes, making the reconstructed scene even more useful than the accurate ground truth floor plans. Finally, our reconstructions clearly maintain the linearity of the hallway systems. Therefore, there virtually is no orientation errors, and the (inevitable but tolerable) floor plan error does not affect users' perceptions to their locations.

We then numerically evaluate the accuracy of 3D scene reconstruction on three aspects: i) dimensioning individual features, ii) positioning these features, and iii) dimensioning the overall scene. Since the first two measurements are vectors, we evaluate them by their respective *root mean square error* (RMSE). We measure the absolute errors of the lengths of hallways and their arbitrarily chosen segments to evaluate the last aspect. The results are presented in Fig. 12 as *cumulative distribution function* (CDF) of errors. All these errors are within a reasonable range, given that VisioMap demands very sparse image samples and only applies lightweight processing techniques. Loops has relatively low dimensioning errors as its features are very regular (e.g., academic posters), whereas it has high hallway length errors due to a slightly curved hallway on the right that has been linearly approximated.

We also briefly compare the floor plans generated by VisioMap with those produced by CrowdInside [1] and Travi-Navi [43] in Fig. 13. As CrowdInside requires a large amount of inertia sensing data whereas what we gathered are rather sparse (as the tests with VisioMap only require sparse sample spots), we term the method we compare with CrowdInside$^-$. It is demonstrated that inertia sensing creates very irregular floor plans in both cases as we would expect. In fact, we cannot even properly align the inertia sensing traces using the methods suggested by CrowdInside$^-$ due to the lack of detectable POIs. Travi-Navi aligns various traces better as it makes use of shared segments, but the plan can be deformed when we force individual hallways to be linear. We do not make an unfair comparison with Jigsaw [10] and SnapTask [23] due to the huge difference in the demand on photos.



(a) VisioMap vs. CrowdInside$^-$

(b) VisioMap vs. Travi-Navi

Fig. 13. Performance comparisons with CrowdInside$-$ and Travi-Navi.

C. Performance of Localization

We first demonstrate the effectiveness of using aspect ratio to index the FeatureDB. Our point is that rectilinear features are naturally classified so that the query processing can be made efficient by searching in B-tree rather than matching with every region descriptor in FeatureDB. In Fig. 14, the aspect ratios sampled from Mall are well clustered, which almost could endow the features with semantics: e.g., the cluster with low aspect ratios represents posters or sign boards and that with high ratios includes doorways.

Recall that, upon an unambiguous bounding rectangle is identified in the R-tree, VisioMap's SMM uses the geometric relations to refine the user location within this sub-region (see Sec. IV-B). In fact, a simplified alternative is to simply report the centroid of this rectangular region as the estimated location, which we term "localization without refining". Also, the localization accuracy can be evaluated against either the ground truth floor (absolute) or the VisioMap generated floor

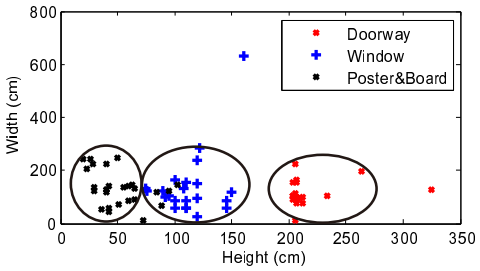


Fig. 14. The distribution of aspect ratios of the rectilinear features in Mall.

plan (relative). Therefore, we further evaluate the performance of scene-based localization in terms of three metrics: i) absolute accuracy without refining, ii) absolute accuracy with refining, and iii) relative accuracy with refining; the results are shown in Fig. 15. As expected, refining using geometric relations leads to a substantial reduction in localization errors, but the mean error of less than 7m for unrefined results is not bad either. Because we rely only on simple geometric reasoning rather than pixel-level processing (e.g. [29]) to estimate camera/user locations, the refined results may not appear to be accurate down to the sub-meter level. We should interpret these errors differently from what we normally do for "unnatural" localization schemes: as VisioMap already puts a user close to the scene that is supposedly seen by the user, meter-level errors can be easily corrected by human vision.

D. Energy Consumption

We hereby briefly evaluate the energy consumption of VisioMap in smart phones. Given a 350KB image captured by each Google Glass shot, default settings for LSD, and the 2600mAh phone battery fully charged before each experiment, our VisioMap system costs about 2%, 4%, and 10% of the phone battery in surveying the three sites, as shown in Fig. 16. Also, VisioMap incurs much lower energy consumption for smart phones than CrowdInside$^-$ and Travi-Navi. In fact, since both CrowdInside$^-$ and Travi-Navi utilize crowdsourcing to reconstruct floor plans, their total energy consumptions are much higher than the ones shown in Fig. 16.

E. Lessons from Experiments

During our extensive experiments with VisioMap, we have realized that it can be extended in a few ways. Although VisioMap requires only sparsely sampled images and hence crowdsensing is not necessary, performing the survey task with more than one user may still be preferred. As individual scenes are accurately dimensioned based on the eye-level of corresponding samplers, seamlessly welding them should be feasible. Currently, no semantics are given to individual features, but the results in Fig. 14 show a potential to infer semantics through simple data analytics. It would indeed be better if we could label the features with proper semantics, so that the 3D scene reconstructed by VisioMap can become even more understandable for the users.

VI. Related Work

Given the huge body of literature on indoor localization [7], [18], [15], [28], [40], [25], [36], [3], [37], [12], [27], [6], [41], a comprehensive survey on the recent progress has bee given in [14], [30]. Due to the limit on space, we hereby give a discussion only on optical-enabled methods as well as floor plan generation schemes, both of which are closely related to our VisioMap.

A. Optical-Enabled Indoor Localization Systems

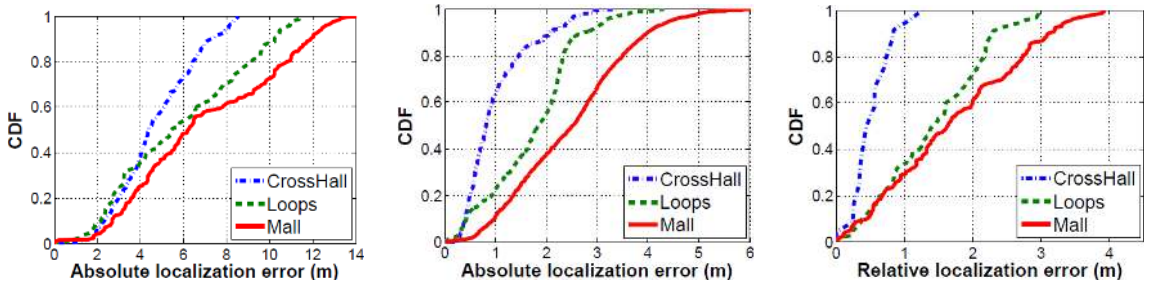
Though optical-enabled systems have been developed for decades in robot navigation [8], [21], [9], those systems demand an image sampling rate (normally beyond 10 Hz) much higher than a human user (with a mobile device) can handle. Within smart phone based optical localization systems, OPS [20] pioneers in applying computer vision techniques for recognizing and locating outdoor POIs with photos taken at multiple viewpoints. As OPS is not meant for indoor localization, Sextant [32] takes a reverse thinking and relies on three photos of nearby POIs to locate a user indoors. While POIs can be too sparse to have three nearby, it is not clear whether such cumbersome user-machine interactions can be widely acceptable. Not relying on vision-based mechanisms, Luxapose [13] innovates in using programmable LEDs to transit location identifiers and smart phones to capture the information. VisioMap again turns to vision-based techniques for natural localization, but it is lightweight by matching only geometric features rather than pixel-level features [2], [29].

B. Floor Plan Generation

Most of the aforementioned systems rely on the existence of a digitized floor plan, but a few recent proposals aim to automatically generate floor plans for scalable indoor localization. While UnLoc [36] is among the first to hint the possibility of generation floor plan through crowdsensing, CrowdInside [1] actually realizes the plan: it relies on POIs to align users' motion traces so that overlapped traces yield a point cloud roughly characterizing the floor plan geometry. To overcome the sparsity of POIs, Travi-Navi [43] aligns traces using the overlapped segments through fingerprint similarity. Jigsaw [10] extends CrowdInside by considering all doorways as POIs. As a result, the point cloud is further supplemented with walls constructed by connecting and expanding doorways through computer vision techniques. ThirdEye [26] combines CrowdInside with Google Glass to reconstruct only the layout of a store. All these proposals produce only 2D floor plans, hence inferior to VisioMap as explained in Sec. I. Although [23] can construct 3D indoor models, it employs SfM algorithm and thus has to collect a huge amount of photos by a crowdsourcing approach . Also, we have to pinpoint quite a few images in the map as features for localization purpose.

VII. Conclusions

VisioMap is an indoor localization system fully based on the visual sensing capability of smart glasses. We consider this as the first attempt to make localization really natural to human users who are used to explore visual sensing for self-localization purpose. VisioMap requires only a single user to perform a sparse photo survey for geometrically reconstructing



(a) Absolute errors without refining. (b) Absolute errors with refining. (c) Relative errors with refining.

Fig. 15. Numerical evaluations of VisioMap's localization accuracy.

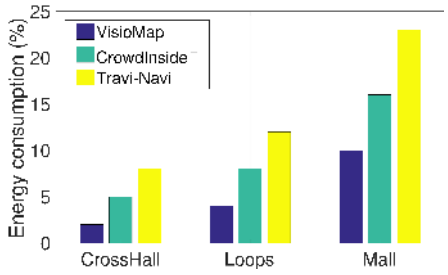


Fig. 16. Energy consumptions in different venues.

the 3D indoor scene, then it allows other users to issue location queries carrying newly taken photos and handles such a query through scene matching against the reconstructed scene. VisioMaps' success is attributed to our three main contributions: i) systematic exploration of the strength of smart glasses, ii) intensive exploitation of the geometric features in photos, and iii) novel idea of scene as location fingerprint. All these have made VisioMap a realistic system for natural indoor localization, as firmly demonstrated by our extensive experiments in various indoor venues.

Although we pioneer in reconstructing 3D scenes for natural indoor localization through light-weight visual sampling, there are plenty of rooms for us to further improve our system. Based on what we have learned from our experiments (see Sec. V-E), we are on the way of integrating crowdsensing and semantic understanding with our system. We are also in the attempt of taking ubiquitous fingerprints in indoor environments (e.g., WiFi fingerprints) as supplements.

REFERENCES

[1] M. Alzantot and M. Youssef. CrowdInside: Automatic Construction of Indoor Floorplans. In *Proc. of ACM SIGSPATIAL*, pages 99–108, 2012.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. *Compt. Vis. Image. Und.*, 110(3):346–359, 2008.

[3] C. Chen, Y. Chen, Y. Han, H. Lai, and K.J. Ray Liu. Achieving Centimeter-Accuracy Indoor Localization on WiFi Platforms: A Frequency Hopping Approach. *IEEE Internet of Things Journal*, 4(1):111–121, 2017.

[4] S. Chen, M. Li, K. Ren, X. Fu, and C. Qiao. Rise of The Indoor Crowd: Reconstruction of Building Interior View via Mobile Crowdsourcing. In *Proc. of the 13th ACM SenSys*, pages 59–71, 2015.

[5] S. Chen, M. Li, K. Ren, and C. Qiao. Crowd map: Accurate Reconstruction of Indoor Floor Plans from Crowdsourced Sensor-Rich Video. In *Proc. of the 35th IEEE ICDCS*, pages 1–10, 2015.

[6] Z. Chen, Z. Li, X. Zhang, G. Zhu, Y. Xu, J. Xiong, and X. Wang. Awl: Turning Spatial Aliasing from Foe to Friend for Accurate WiFi Localization. In *Proc. of the 13th ACM CoNEXT*, pages 238–250, 2018.

[7] K. Chintalapudi, A. Padmanabha Iyer, and V. Padmanabhan. Indoor Localization Without the Pain. In *Proc. of the 16th ACM MobiCom*, pages 173–184, 2010.

[8] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[9] J. Engel, J. Sturm, and D. Cremers. Semi-Dense Visual Odometry for a Monocular Camera. In *Proc. of IEEE ICCV*, pages 1449–1456, 2013.

[10] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li. Jigsaw: Indoor Floor Plan Reconstruction via Mobile Crowdsensing. In *Proc. of the 20th ACM MobiCom*, pages 249–260, 2014.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[12] H.Gan, M. Khir, G. Djaswadi, and N. Ramli. A Hybrid Model Based on Constraint OSELM, Adaptive Weighted SRC and KNN for Large-Scale Indoor Localization. *IEEE Access*, 7:6971–6989, 2019.

[13] Y. Kuo, P. Pannuto, K. Hsiao, and P. Dutta. Luxapose: Indoor Positioning with Mobile Phones and Visible Light. In *Proc. of the 20th ACM MobiCom*, pages 447–458, 2014.

[14] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione. A Survey of Enabling Technologies for Network Localization, Tracking, and Navigation. *IEEE Communications Surveys Tutorials*, 20(4):3607–3644, 2018.

[15] P. Lazik and A. Rowe. Indoor Pseudo-Ranging of Mobile Devices uisng Ultrasonic Chirps. In *Proc. of ACM SenSys*, pages 99–112, 2012.

[16] D. Lee, M. Hebert, and T. Kanade. Geometric Reasoning for Single Image Structure Recovery. In *Proc. of IEEE CVPR*, 2009.

[17] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A Reliable and Accurate Indoor Localization Method Using Phone inertial Sensors. In *Proc. of the 14th ACM UbiComp*, pages 421–430, 2012.

[18] 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 *Prof. of the 18th ACM MobiCom*, pages 305–316, 2012.

[19] Y. Manolopoulos, A. Nanopoulos, A. Papadopoulos, and Y. Theodoridis. *R-Trees: Theory and Applications*. Springer-Verlag, 2006.

[20] J. Manweiler, P. Jain, and R. Choudhury. Satellites in Our Pockets: An Object Positioning System Using Smartphones. In *Proc. of the 10th ACM MobiSys*, pages 211–224, 2012.

[21] R. Mautz and S. Tilch. Survey of Optical Indoor Positioning Systems. In *Proc. of the 2nd IEEE IPIN*, pages 1–7, 2011.

[22] B.Branislav Mičušík, Horst Wildenauer, and Jana Košecká. Detection and Matching of Rectilinear Structures. In *Prof. of IEEE CVPR*, 2008.

[23] M. Noreikis, Y. Xiao, J. Hu, and Y. Chen. SnapTask: Towards Efficient Visual Crowdsourcing for Indoor Mapping. In *Proc. of the 38th IEEE ICDCS*, pages 578–588, 2018.

[24] Štěpán Obdržálek and Jiří Matas. Object Recognition Using Local Affine Frames on Maximally Stable Extremal Regions. In *Toward Category-Level Object Recognition*, LNCS 4170, pages 83–104. 2006.

[25] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an Organic Indoor Location System. In *Proc. of the 8th ACM MobiSys*, pages 271–284, 2010.

[26] S. Rallapalli, A. Ganesan, K. Chintalapudi, V. Padmanabhan, and L. Qiu. Enabling Physical Analytics in Retail Stores Using Smart Glasses. In *Proc. the 20th ACM MobiCom*, pages 115–126, 2014.

[27] S. Sadowski and P. Spachos. RSSI-Based Indoor Localization With the Internet of Things. *IEEE Access*, 6:30149–30161, 2018.

[28] S. Sen, J. Lee, K. Kim, and P. Congdon. Avoiding Multipath to Revive Inbuilding WiFi Localization. In *Proc. of ACM MobiSys*, 2013.

[29] N. Snavely, I. Simon, M. Goesele, R. Szeliski, and S. Seitz. Scene Reconstruction and Visualization From Community Photo Collections. *Proceedings of the IEEE*, 98(8):1370–1390, 2010.

[30] P. Spachos, I. Papapanagiotou, and K. Plataniotis. Microlocation for Smart Buildings in the Era of the Internet of Things: A Survey of Technologies, Techniques, and Approaches. *IEEE Signal Processing Magazine*, 35(5):140–152, 2018.

[31] T. Stafford and M. Gillenson. Mobile Commerce: What It is and What It Could Be. *Commun. ACM*, 46(12):33–34, 2003.

[32] Y. Tian, R. Gao, K. Bian, F. Ye, T. Wang, Y. Wang, and X. Li. Towards Ubiquitous Indoor Localization Service Leveraging Environmental Physical Features. In *Proc. of the 33rd IEEE INFOCOM*, pages 55–63, 2014.

[33] Y. Tung, C. Hsu, H. Wang, S. Chyou, J. Lin, P. Wu, A. Valstar, and M. Chen. User-Defined Game Input for Smart Glasses in Public Space. In *Proc. of the 33rd ACM CHI*, pages 3327–3336, 2015.

[34] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.

[35] C. Wang, W. Chu, P. Chiu, M. Hsiu, Y. Chiang, and M. Chen. PalmType: Using Palms As Keyboards for Smart Glasses. In *Proc. of the 17th ACM MobileHCI*, pages 153–160, 2015.

[36] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. Choudhury. No Need to War-drive: Unsupervised Indoor Localization. In *Proc. of the 10th ACM MobiSys*, pages 197–210, 2012.

[37] X. Wang, L. Gao, and S. Mao. CSI Phase Fingerprinting for Indoor Localization With a Deep Learning Approach. *IEEE Internet of Things Journal*, 3(6):1113–1123, 2016.

[38] H. Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, and Y. Liu. Indoor Localization via Multi-Modal Sensing on Smartphones. In *Proc. of ACM UbiComp*, pages 208–219, 2016.

[39] W. Xu, Y. Shen, N. Bergmann, and W. Hu. Sensor-assisted Face Recognition System on Smart Glass via Multi-view Sparse Representation Classification. In *Proc. of the 15th ACM/IEEE IPSN*, 2016.

[40] M. Youssef and A. Agrawala. The Horus WLAN Location Determination System. In *Proc. of the 3rd ACM MobiSys*, pages 205–218, 2005.

[41] C. Zhang, F. Li, J. Luo, and Y. He. iLocScan: Harnessing Multipath for Simultaneous Indoor Source Localization and Space Scanning. In *Proc. of the 12th ACM SenSys*, pages 91–104, 2014.

[42] Y. Zhang, W. He, W. Xu, H. Wen, and C. Chou. NaviGlass: Indoor localisation Using Smart Glasses. In *Proc. of EWSN*, 2016.

[43] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao. Travi-Navi: Self-deployable Indoor Navigation System. In *Proc. of the 20th ACM MobiCom*, pages 471–482, 2014.

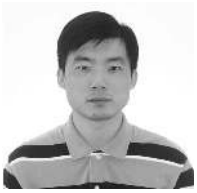
Feng Li received his BS and MS degrees in Computer Science from Shandong Normal University, China, in 2007, and Shandong University, China, in 2010, respectively. He got his PhD degree (also in Computer Science) from Nanyang Technological University, Singapore, in 2015. From 2014 to 2015, he worked as a research fellow in National University of Singapore, Singapore. After that, he joined the School of Computer Science and Technology, Shandong University, China, where he is currently an associate professor. His research interests include distributed algorithms and systems, wireless networking, mobile computing, and Internet of Things.

Jie Hao received the B.S. degree from the Beijing University of Posts and Telecommunications, China, in 2007, and the Ph.D. degree from the University of Chinese Academy of Sciences, China, in 2014. From 2014 to 2015, she was a Post-Doctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. She is currently a Lecture with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. Her research interests are visible light sensing and IoT.

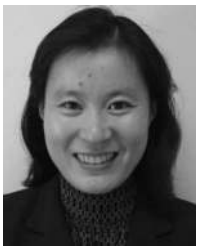
Jin Wang received her BS degrees in Computer Science from Nanyang Technological University, Singapore, in 2010. She is currently a PhD student in Nanyang Technological University, Singapore, and meanwhile working in SAP Innovation center as a research associate. Her research interests include mobile sensing, physical analytics and human-computer interaction.

Jun Luo got his PhD degree in computer science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland, in 2006. From2006 to 2008, he has worked as a post-doctoral research fellow in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of the School of Computer Science and Engineering, Nanyang Technological University in Singapore, where he is currently an Associate Professor. His research interests include wireless networking, distributed systems, multimedia protocols, network modeling and performance analysis, applied operations research, as well as network security. He is a Member of both IEEE and ACM.

Ying He received the BS and MS degrees in Electrical Engineering from Tsinghua University, China, and the PhD degree in Computer Science from the State University of New York (SUNY), Stony Brook, USA. He is currently an associate professor at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the broad areas of visual computing, with a focus on the problems that require geometric computation and analysis. More information can be found at http://www.ntu.edu.sg/home/yhe.



Dongxiao Yu received the BSc degree in 2006 from the School of Mathematics, Shandong University and the PhD degree in 2014 from the Department of Computer Science, The University of Hong Kong. He became an associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2016. He is currently a professor in the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing and graph algorithms.



Xiuzhen Cheng received her M.S. and Ph.D. degrees in computer science from the University of Minnesota – Twin Cities in 2000 and 2002, respectively. She is a professor in the School of Computer Science and Technology, Shandong University. Her current research interests include cyber physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time), and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is Fellow of IEEE and a member of ACM.

View publication stats