SiFi: Self-updating of Indoor Semantic Floorplans for Annotated Objects

DEKE GUO, College of System Engineering, National University of Defense Technology, P. R. China and College of Intelligence and Computing, Tianjin University, P. R. China
XIAOQIANG TENG, College of System Engineering, National University of Defense Technology, P. R. China
YULAN GUO, College of Electronic Science, National University of Defense Technology, P. R. China and School of Electronics and Communication Engineering, Sun Yat-sen University, P. R. China
XIAOLEI ZHOU, 63rd Research Institute, National University of Defense Technology, P. R. China
ZHONG LIU, College of System Engineering, National University of Defense Technology, P. R. China

Due to the rapid development of indoor location based services, automatically deriving an indoor semantic floorplan becomes a highly promising technique for ubiquitous applications. To make indoor semantic floorplan fully practical, it is essential to handle the dynamics of semantic information. Despite several methods proposed for automatic construction and semantic labeling of indoor floorplans, this problem has not been well studied and remains open. In this paper, we present a system called SiFi to provide accurate and automatic self-updating service. It updates semantics with instant videos acquired by mobile devices in indoor scenes. First, a crowdsourced-based task models are designed to attract users to contribute semantic-rich videos. Second, we use the maximum likelihood estimation method to solve the text inferring problem as the sequential relationship of texts provides additional geometrical constraints. Finally, we formulate the semantic update as an inference problem to accurately label semantics at correct locations on the indoor floorplans. Extensive experiments have been conducted across nine weeks in a shopping mall with more than 250 stores. Experimental results show that SiFi achieves 84.5% accuracy of semantic update.

CCS Concepts: • **Networks** → **Network services**; *Location based services*; • **Computer systems organization** → *Real-time systems*;

Additional Key Words and Phrases: Indoor semantic floorplan, crowdsourcing, self-updating system

ACM Reference Format:
Deke Guo, Xiaoqiang Teng, Yulan Guo, Xiaolei Zhou, and Zhong Liu. 2020. SiFi: Self-updating of Indoor Semantic Floorplans for Annotated Objects. *ACM Trans. Internet Things* 1, 1 (October 2020), 20 pages. https://doi.org/0000001.0000001

Authors' addresses: Deke Guo, College of System Engineering, National University of Defense Technology, Changsha, Hunan, 410073, P. R. China, College of Intelligence and Computing, Tianjin University, Tianjin, 300350, P. R. China, guodeke@gmail.com; Xiaoqiang Teng, College of System Engineering, National University of Defense Technology, Changsha, Hunan, 410073, P. R. China, tengxiaoqiang13@nudt.edu.cn; Yulan Guo, College of Electronic Science, National University of Defense Technology, Changsha, Hunan, 410073, P. R. China, School of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou, 510275, P. R. China, yulan.guo@nudt.edu.cn; Xiaolei Zhou, 63rd Research Institute, National University of Defense Technology, Nanjing, Jiangsu, 210089, P. R. China, xl.zhou@nudt.edu.cn; Zhong Liu, College of System Engineering, National University of Defense Technology, Changsha, Hunan, 410073, P. R. China, liuzhong@nudt.edu.cn.

2577-6207/2020/10-ART $15.00
https://doi.org/0000001.0000001

1 INTRODUCTION

Location based services (LBS) have been developed for various applications, including navigation [30], geo-social networks [14], and location-based advertising [9]. The aim of LBS is to provide a service using the geographic location of a person [24]. Outdoor LBS using GPS and Google maps are well studied and has been adopted over the world. However, similar LBS are unavailable in indoor environments due to the lack of GPS signals. Therefore, it is essential to provide indoor LBS applications.

The availability of indoor floorplan is a major factor for practical applications of indoor LBS. Consequently, the automatic construction of detailed indoor floorplan becomes an important technique for ubiquitous indoor LBS. Extensive investigation has been conducted to construct indoor floorplans using crowdsourced data (e.g., motion trajectories, images, and WiFi signals) [2, 8, 15, 23, 31, 37]. The constructed indoor floorplan, however, is unaware of various semantics in indoor scenes. On the contrary, the appearance of semantic-rich indoor floorplan can be used to improve existing indoor LBS methods and to design new indoor LBS applications.

Recently, several techniques have been proposed to manually label or learn semantics for objects in an indoor floorplan [10, 11, 17, 22, 27]. For example, SemSense [10] requires each user to actively assign a semantic name to a physical location during check-in operations. Mobile sensing data (e.g., images and WiFi signals) are required by ShopProfiler [17] and AutoLabel [27] to label store names in a shopping mall with learning methods. TransitLabel [11] is developed to recognize user activities in transit stations and to infer the functionalities around the physical areas of users. Despite the progress made, existing works do not account for dynamically changing semantics in indoor environments. For example, when a store closes and a new one opens in the exact same location of a shopping mall, the performance of LBS applications may deteriorate or even break down, due to the out-of-date semantics (e.g. name label). This problem is known as semantic updating of indoor floorplan.

A straightforward approach is to ask crowd to manually label changed semantics. Unfortunately, this method is high-cost and cannot guarantee high accuracy due to the uncertainty of crowd behavior. To make things worse, some trick users are deliberate to label incorrect semantics in the indoor semantic floorplans. An alternative solution is to use existing semantic labeling methods [10, 11, 17, 22, 27] during each particular period. This updating strategy, however, is very labor-intensive and time-consuming, and may introduce unnecessary updates for unchanged environment. Additionally, such labeling methods are designed for automatic construction of the complete floorplan. They are unsuitable for continuous semantics updating with high accuracy in complex indoor spaces.

This paper presents SiFi as a mobile crowdsourcing system to update indoor semantic floorplans automatically, continuously, and accurately. An indoor semantic floorplan has semantic-rich labels for those indoor objects in the floorplan. Thus, it represents both the spatial structures and their semantics (e.g., categories, functionalities, and other non-spatial attributes) of an indoor space. An indoor objects can refer to any location, area, or general entities. Such indoor objects can be divided into two types: annotated objects and non-annotated objects. An annotated object implies that its semantic information has been manually labeled with texts. For example, the name and functionality of an indoor object are usually labeled with texts in complex commercial places. A non-annotated object (e.g., a fine-grained general entity) still lacks accurate semantic labels. This paper focuses on automatic and continuous updating of annotated objects in indoor space, i.e., texts.

Instant videos are the latest and short (e.g., 30 seconds) videos captured in indoor scenes. They provide sufficient semantics for indoor objects and are easy for users to record and share. Users are motivated by dedicated task models to capture instant videos in a large open indoor environment, which incurs low mobile device power consumptions (Section 4.1). The recorded instant videos are loaded to SiFi server for further processing in an online or offline manner, which requires only a small amount of network resources. Texts are extracted from videos and sequential relationships are established among texts to generate *text sequences*. Compared to uncorrelated texts, it is demonstrated by SiFi that the sequential relationships among texts provide more valuable information in the indoor crowdsourced settings. The accurate text inferring task is formulated as a Maximum Likelihood Estimation (MLE) problem since the sequential relationships provide additional geometrical constraints (Section 4.2). These text sequences can be matched with the indoor floorplans and then the changed semantics can be updated. Consequently, starting from a pre-established indoor semantics floorplan of a general indoor environment, SiFi detects and removes the out-of-date semantics and localizes the new semantics in the indoor floorplan timely (Section 4.3). Therefore, the quality of indoor LBS can be persistently maintained after long-term deployment. Moreover, SiFi does not rely on any indoor localization system or extra dedicated hardware. Our method is orthogonal to existing semantic labeling methods for indoor floorplans [10, 11, 17, 22, 27]. They can be combined to achieve indoor LBS even for long-term deployment.

The major contributions of this paper can be summarized as follows.

- We propose an automatic and continuous method (namely SiFi) to update indoor semantic floorplan using instant videos. Our method does not require any dedicated device or additional indoor localization system.
- We investigate the sequential relationships among texts and use the MLE method to improve the update performance of texts. We further propose localization algorithm of text sequence to update out-of-date texts on indoor floorplans.
- A SiFi prototype system is implemented and extensive experiments have been conducted in a shopping mall. Experimental results demonstrate that SiFi achieves promising performance for indoor semantic floorplan updating.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 gives an overview of SiFi. Section 4 describes the design details of SiFi. Section 5 presents the implementation and evaluation of SiFi, followed by technical discussions and limitations in Section 6. We conclude this paper in Section 7.

2 RELATED WORK

Our work constructs a model to update indoor semantic floorplans with crowdsourced data. It is closely related to works that address the following three problems.

Digital Indoor Floorplan Construction. Several systems have been proposed to construct digital indoor floorplans using the general layout of a building. FootSLAM [3] was proposed using a Simultaneous Localization and Mapping (SLAM) based approach for pedestrians to generate a probabilistic map. Puyol et al. [32] proposed an autoregressive integrated moving average model for the odometry error along the vertical component to extend FootSLAM to multistory buildings. CrowdInside [2] constructed the floorplan of the building using a mass of walking traces of humans in an indoor space. MapGENIE [31] was proposed to perform indoor mapping using the exterior information and the grammar generator [5] to encode structural information about the building. Jiang et al. [23] used WiFi fingerprints and user motion information to propose an automatic indoor map construction system. Shen et al. [37] proposed an indoor pathway mapping system, called Walkie-Markie, to automatically reconstruct internal pathway maps of buildings. They used novel landmarks, i.e., WiFi-defined landmarks, to calibrate the partial walking traces collected by different users. Jigsaw [15] used a computer vision approach to extract the location and orientation of landmarks from images taken by users. It then used the walking traces of users and the location of cameras to reconstruct the indoor floorplans. However, we cannot assume that all edges and corners of the room could be covered with user traces [8]. CrowdMap [8] reconstructed indoor floorplans using videos and inertial sensor data from users. It employed computer vision algorithms to exploit the sequential relationship between each consecutive frame abstracted from the video to generate accurate spatial information of the indoor environment. Nevertheless, these systems do not provide rich semantic information to the indoor floorplans [10, 11].

Indoor Semantic Floorplan Construction. Recently, several techniques have been proposed to manually label or learn semantics for objects in an indoor floorplan [10, 11, 17, 22, 27]. ShopProfiler [17] was proposed to refine floorplans and characterize shops in terms of location, category, and name using crowdsourced sensor readings from mobile phones of users. Semsense [10] used sensor data collected by mobile devices from users during their normal check-in operations to associate a place name with its location on an unlabeled floorplan. AutoLabel [27] was proposed to automatically label clusters of pictures and the corresponding WiFi Access Points (APs) with store names in a shopping mall using images and WiFi signals collected by mobile devices. It uses a mobile device to scan the WiFi APs and recognizes the store the user is in. Overlay [22] was proposed to register objects or places of interest into an augmented reality system by collecting a set of images. It has to manually label all the images. TransitLabel [11] was developed to recognize user activities in transit stations and to infer the functionalities around the physical areas of users. However, those methods are designed for automatic construction of complete floorplan rather than continuous updating of semantics in complex indoor space.

Vision-based Text Recognition. Text recognition has become a very active research topic in the computer vision community. Many methods have been proposed to recognize texts in images. Almazán et al. [1] addressed the spotting and recognition tasks by learning a common representation for word images and text strings. Rodríguez-Serrano et al. [33] embedded word attributes/labels and word images into a common subspace for word spotting and text recognition. Then, the text recognition problem is transformed to a retrieval problem. Yao et al. [44] used a representation of strokes to produce more semantic description of characters, that are then classified using random forests. Recently, deep neural network models, and specifically Deep Convolutional Neural Networks (DCNN), have been rapidly developed in text recognition. Wang et al. [41] firstly detected individual characters and then recognized these characters with DCNN models. Bissacco et al. [6] used binarization and a sliding window classifier to generate candidate character regions and then used classifier scores into recognizing words. Jaderberg et al. [21] used the convolutional nature of CNNs to generate response maps for characters, which are then integrated to score lexicon words. Wang et al. [40] used the Vanishing Points method [19] to extract text boxes and used the N-gram method to recognize texts. However, due to the complexity of indoor environments and the presence of large perspective distortions, it is difficult for those methods to achieve the desired text recognition accuracy in practical applications. In contrast, SiFi exploits sequential relationships among texts and uses additional geometrical constraints for text inference.

3 OVERVIEW

In this section, we first give the problem description and then describe the architecture of the proposed system.

3.1 Problem Description

An indoor semantic floorplan contains semantics (e.g., texts) extracted from objects (e.g., store logos in a shopping mall) in an indoor space. These semantics attached in the indoor floorplan can enhance the performance of the LBS applications. However, the performance of LBS applications

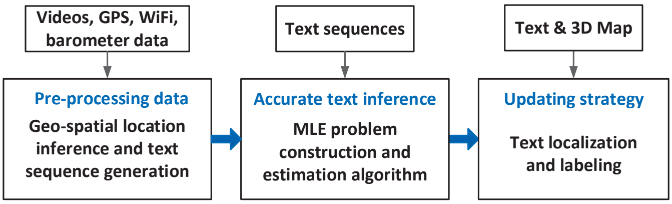

Videos, GPS, WiFi, barometer data
Text sequences
Text & 3D Map
Pre-processing data
Geo-spatial location inference and text sequence generation
Accurate text inference
MLE problem construction and estimation algorithm
Updating strategy
Text localization and labeling

Fig. 1. System architecture.

can deteriorate and even break down due to the out-of-date semantics in an indoor space. For example, a closing store is closed and a new restaurant is open in the same location. Thus, the semantics of the closing store should be removed and the semantics of the new restaurant should be labeled in the indoor floorplan. This general problem is formulated as indoor semantic floorplan updating.

3.2 Challenges

This paper presents a novel method to make an indoor semantic floorplan last for a long time by adapting to environmental dynamics. Translating such an intuitive idea to a practical system, several challenges have to be addressed by SiFi.

First, it is difficult to collect semantic-rich instant videos by mobile crowdsourcing. Usually, most crowdsourced data not only exhibit low quality but also lack required semantics, because users may be unprofessional for crowdsourcing tasks. Such data would cause unnecessary consumptions of energy and bandwidth at mobile devices. To address this problem, several dedicated task models are designed to guide unprofessional users to collect semantic-rich instant videos.

Second, it is challenging to accurately detect and recognize texts in instant videos. The texts extracted from each individual instant video suffer considerable false positive and false negative errors, even with state-of-the-art techniques. These errors will be significantly accumulated. Furthermore, these errors would lead to incorrect updating for the indoor semantic floorplan. To address this problem, the text inferring is formulated as a Maximum Likelihood Estimation (MLE) problem since sequential relationships among texts provide additional geometrical constraints.

Third, it is unknown how to update changed texts at correct positions in indoor semantic floorplans without any indoor localization system. To solve this issue, the text sequence matching and localization methods are proposed to remove the out-of-date texts and label new texts in an indoor floorplan incorporating with the Longest Common Subsequence (LCS) [20] and the Structure from Motion (SfM) [36] algorithms.

3.3 System Architecture

Figure 1 illustrates the architecture of the proposed SiFi system. SiFi utilizes crowdsourced semantic-rich instant videos. It consists of two components: mobile application software and a server. To collect semantic-rich instant videos, two crowdsourcing tasks are performed by users in indoor spaces using mobile devices deployed by the application software (Section 4.1). The recorded semantic-rich instant videos are then automatically compressed and uploaded to the server for further processing. Most of the computational burden is enforced in the server where the uploaded instant videos are processed. Since instant videos can be recorded by different users, there are significant diversities in mobile devices, usage poses, camera positions, and view directions. Besides, the camera motion traces are unknown in advance. These factors make the use of videos highly challenging. Therefore, semantic-rich instant videos received at the server are then fed into the text extraction module (Section 4.2). Specifically, the text sequences are generated by a text recognition algorithm, grouped using the Jaccard similarity metric and extracted using the MLE approach. Once text sequences are obtained, texts are updated by localizing text sequences in the indoor floorplan using both room facades and unchanged texts cues (Section 4.3).

4 SYSTEM DESIGN

In this section, we first describe the crowdsourced task modeling for indoor semantic floorplan updating. We further formulate the text extraction task as a MLE problem and design an efficient estimation algorithm to solve this problem. Finally, we present a method to update texts at correct locations of indoor floorplans.

4.1 Crowdsourced Task Modeling

Crowdsourcing provides access to a large number of mobile devices and people in a cost-effective manner. Usually, most crowdsourced data are of low quality and lack the required semantics, as users may be unqualified for crowdsourcing tasks. The low-quality data may cause unnecessary energy and bandwidth consumptions at mobile devices.

To address this problem, we first leverage GPS and barometer data to infer the geo-spatial information of collected videos and then propose dedicated task models to guide unprofessional users to collect videos. In this task, videos are recorded to cover as many texts as possible using the rear cameras embedded in mobile devices. In addition, users are required to hold mobile devices in front of their free hands and to keep mobile devices stable during recording. Moreover, users are motivated to record instant indoor videos when they are in a large indoor space.

4.1.1 Geo-spatial Information Inferring. Geospatial information is used to accelerate the text updating process on indoor floorplans by providing the coarse locations of the users. To obtain a building location and locations of a user in the building, the localization software development kit (SDK) of existing indoor methods is used. To obtain the current floor where the user is on, the barometer is used because of its low power consumption and high accuracy [28].

Given an initial floor number, we infer the users' floor number using altitude data. It is hard to determine the initial floor number since the floor number where the user starts to use SiFi is usually unknown. Therefore, when SiFi is started, a new dialog box is promoted to ask a user to give its current floor number F_0. If the floor height h_i is known, where i represents i-th floor, the floor number F_c can be inferred. As shown in Fig. 2, the altitude data were collected when a user climbed from floor 1 to floor 4 and then walked down to floor 1. The dotted line represents the smoothed altitude data generated by averaging all samples within every 6 seconds. We also recorded the timestamps when the user started walking on the stairs and arrived at a new floor as the ground truth floor number (red circles in Fig. 2). Let H_p be the initial altitude data from the floor given by a user, H_c be the current altitude data, the floor difference j is calculated by

$$\sum_{i=F_0+s\times 1}^{F_0+s\times j} h_i = |H_c \text{-} H_p|, \tag{1}$$

where $s=\frac{H_c\text{-}H_p}{|H_c\text{-}H_p|}$. Thus, we have $F_c=F_0+s\times j$.

Note that, the floor inferring method is an online process. It only requires a one-time interaction of the user, that is a user only once gives his floor number. Afterwards, the method can be automatically and continuously executed to infer the floor.

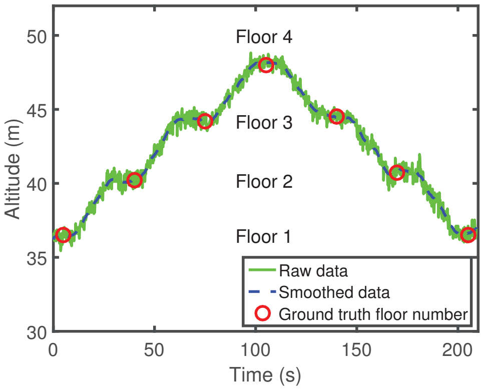


Fig. 2. Data processing for floor number inferring.

4.1.2 Task modeling. The task model is used to guide a user to collect videos based on motion states of the user. The states of a user are classified into two categories: static state and walking state.

Task 1**: Static State.** The quality of videos recorded by a static mobile device is higher than that recorded by a moving mobile device. Therefore, users are suggested to record semantic-rich instant videos statically. That is, the camera can be rotated by the user to record a semantic-rich instant video.

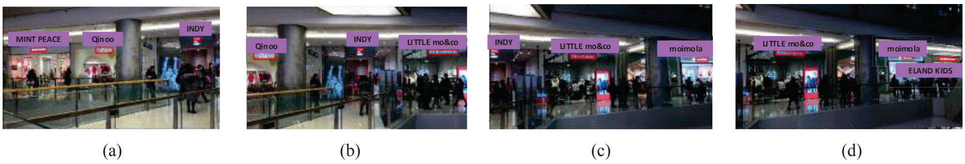
Task 2**: Walking State.** The user records a semantic-rich instant video at a location *A*, and then holds the mobile device and walks to another location *B*.

Note that, SiFi does not allocate collection tasks for users. Indeed, task allocation algorithms in crowdsourcing settings can be used to improve the data quality collected by users [25, 42]. However, that is out of the scope of this paper and can be employed to improve data quality in the future. These two crowdsourcing tasks are released by the server. When participating in these crowdsourcing tasks, users consume their own resources such as battery and computing power and expose themselves to potential privacy risks by sharing videos. Therefore, an incentive mechanism [43, 45] has to be designed to encourage users to perform these crowdsourcing tasks.

4.2 Text Inference

Since SiFi aims to update texts in indoor semantic floorplans, text recognition from an image is an important stage. Recently, Wang et al. [40] used the Vanishing Points (VP) method [19] to extract text boxes and the N-gram and the sequential character classification methods to recognize texts. The accuracy of the text recognition is about 71% when tested in a shopping mall and it produces many false texts. These errors can be accumulated in the indoor semantic floorplans and degrade the performance of the LBS applications. In this paper, we remove these recognition errors of texts using the sequence relationship among texts rather than developing a more advanced text recognition algorithm. Our method is orthogonal to existing text recognition algorithms and they can be combined to improve the accuracy of text recognition.

4.2.1 Key-frame Selection. In SiFi, the videos are recorded by a camera at 30Hz and many frames can be extracted from an instant video. Considering the motion of a camera in a mobile device usually is slow and texts are sparse, neighboring frames in a video are highly similar. Therefore, several representative frames (namely key-frame) can be selected and processed instead of processing all of the frames. Specifically, we use ORB [34] features to select key-frames. If the number of ORB



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

Fig. 3. An illustration of a text sequence generated from four key-frames. Texts recognized from images are (a) MINTPEACE, Qinoo, INDY, (b) Qinoo, INDY, LITTLEmo&co, (c) INDY, LITTLEmo&co, moimola, and (d) LITTLEmo&co, moimola, ELANDKIDS. The duplicated texts are Qinoo, INDY, LITTLEmo&co, and moimola. Finally, the text sequence is generated as MINTPEACE, Qinoo, INDY, LITTLEmo&co, moimola, ELANDKIDS.

features in a frame is larger than threshold TH_K, the frame is selected as a candidate key-frame. Each candidate key-frame is matched against its previous key-frame using ORB features. If the percentage of matched features is less than a threshold R_K, the candidate key-frame is selected as a new key-frame. This process is repeated for remaining candidate key-frames. The selection of parameters TH_K and R_K is further discussed in Section 5.3.3. Furthermore, blurred key-frames are removed. The blurred images can decrease the accuracy of text recognition [40]. In this paper, the blurriness values of a key-frame and its four adjust frames are calculated using the method proposed in [12]. The frame with the smallest blurriness value is selected as the key-frame. Finally, a key-frame sequence I=$\{I_1, I_2, \ldots, I_m\}$ can be obtained from a video, where m is the number of key-frames.

4.2.2 Text Sequence Generation. A text sequence denotes the sequence relationship among texts. Texts are first recognized from a key-frame [40], resulting in both texts and their coordinates. The spatial correlation between two texts can be obtained using their coordinates and the temporal correlation of texts can also be extracted from with timestamps of key-frames. The spatial-temporal correlation provides us an important clue to generate a text sequence.

As illustrated in Fig. 3, the text sequences are obtained, i.e., (a) MINTPEACE, Qinoo, INDY, (b) Qinoo, INDY, LITTLEmo&co, (c) INDY, LITTLEmo&co, moimola, and (d) LITTLEmo&co, moimola, ELANDKIDS. We propose two rules to identify duplicate texts: (1) If a text has existed in the sequence (namely the second text), it is labeled as candidate. (2) If the neighboring texts of the second text and the first text (the same text as the second text in the sequence) are also the same, the second text is the duplicated text. As illustrated in Fig. 3, the duplicated texts are Qinoo, INDY, LITTLEmo&co, and moimola. The duplicated texts can be a landmark to merge these text sequences. Finally, the text sequence is generated as {MINTPEACE, Qinoo, INDY, LITTLEmo&co, moimola, ELANDKIDS}.

4.2.3 Text Sequence Grouping. A list of shop names in a mall is first generated manually. Second, the Simhash algorithm [35] and Hamming distance [29] are used to refine the text sequence. The Simhash algorithm is a locality sensitive hashing based fingerprinting technique. It uses random projections to generate a compact representation of a high dimension vector [35]. The Hamming distance $d(\boldsymbol{x}, \boldsymbol{y})$ between two vectors $\boldsymbol{x}, \boldsymbol{y} \in F^{(n)}$ is the number of coefficients in which they differ, such as, $d(10111, 11001)$=3. In SiFi, the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are obtained by the Simhash algorithm. The text with the highest score is selected to be added to the text sequence. As illustrated in Table 1, the false texts $MINTPAACE$, $Qimoo$, and $moonola$ are amended to $MINTPEACE$, $Qinoo$, and $moimola$.

Once text sequences are obtained, multiple text sequences are divided into different groups based on their similarity values. In this paper, the widely used *text Jaccard similarity* is adopted. The

Jaccard text similarity of T_1 and T_2 is:

$$sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}, \tag{2}$$

where, T_1 and T_2 are the text sequences. Note that, other popular similarity measures can be converted to Jaccard similarity, such as Hamming distance, cosine similarity, and overlap similarity [4]. If $sim(T_1, T_2)$ is larger than a threshold τ, the two text sequences are considered as similar and classified into a group, where the threshold is empirically set based on experiments (Set to be 0.8 in our experiments).

4.2.4 Accurate Text Inference. Text recognition results usually contain false positives and false negatives. These errors introduce challenges for texts updating. To reduce these errors, we infer accurate texts using the *configuration* of text sequences. The configuration is defined as the true value of texts and the sequence relationship among texts in the text sequence, where the sequence relationship denotes the relative location relationship among the text.

Notations. Suppose there are N local coordinate systems corresponding to N groups of text sequences, a group is indexed by k as a superscript. $\boldsymbol{t_e}$ is the true value of text $\boldsymbol{t}$ and $\boldsymbol{S}$ is the sequence relationship in a text sequence. $\boldsymbol{Z}$ gives the coordinates of texts in the local coordinate system of a group, that is, the observation of $\boldsymbol{S}$. We formulate the *accurate text inferring (ATI)* task as a Maximum Likelihood Estimation (MLE) problem. Its goal is to infer the best configuration of $\boldsymbol{t_e}$ and S with the measurements $\boldsymbol{t}$ and $\boldsymbol{Z}$ provided by a group of text sequences. We consider this estimation problem as finding $\boldsymbol{t_e}$ and $\boldsymbol{S}$ to maximize the joint likelihood based on conditional dependence:

$$\begin{aligned} \{\boldsymbol{t_e}, S\} &= \underset{\boldsymbol{t_e}, S}{\arg\max}\ Pr(\boldsymbol{t}, \boldsymbol{Z} | \boldsymbol{t_e}, S) \\ &= \underset{\boldsymbol{t_e}, S}{\arg\max}\ Pr(\boldsymbol{t} | \boldsymbol{t_e}, S) Pr(Z | \boldsymbol{t_e}, S). \end{aligned} \tag{3}$$

Likelihood $Pr(\boldsymbol{t}|\boldsymbol{t_e}, S)$. $Pr(\boldsymbol{t}|\boldsymbol{t_e}, S)$ measures the likelihood of measurement $\boldsymbol{t}$ given the true value of texts $\boldsymbol{t_e}$ and sequence relationship configurations $\boldsymbol{S}$. This term can be estimated by computing the agreement between observations and estimate values of the true values. That is to say that the estimated values of the true values what we want to obtain can be calculated by observations of text sequences via maximizing this probability function. Observations include two probability expressions. The first one is $P_q = \frac{N(t_i^q)}{N(t_i)}$, which is computed by the candidate t_i^q of the i-th ground-truth, where $N(t_i)$ is the number of candidate of the i-th ground-truth and $N(t_i^q)$ is the number of the q-th candidate of the i-th ground-truth. The second one is $P_q^j = \frac{N(S^j_{t_i^q})}{N(S^j_{t_i})}$, which is computed by the number of text sequences, where $N(S^j_{t_i})$ is the number of text sequences containing the ith text, $N(S^j_{t_i^q})$ is the number of text sequences containing the q-th candidate and is indexed by j as a

Table 1. Text corrections using the Simhash algorithm.

Text	*MINTPEACE*	*Qinoo*	*INDY*	*LITTLEmo&co*	*moimola*	*ELANDKIDS*
MINTPAACE	100%	0.05%	0.00%	40.03%	19.05%	17.75%
Qimoo	0.05%	100.00%	0.00%	0.02%	0.01%	0.01%
moonola	19.05%	0.01%	8.83%	47.58%	99.99%	93.16%

Algorithm 1: Estimation Algorithm for Accurate Text Inferring

```
Input: Text sequences and a uniform random variable $q \sim U(0,1)$.
Output: The final estimation of $\boldsymbol{t_e}$ and $S$.
1  /*Likelihood Maximization*/
2  for each text do
3      Propose $\boldsymbol{t'_e}$ and $S'$ with Gaussian probability;
4      Compute the acceptance ratio $\alpha = \frac{Pr(t,Z|t'_e,S')}{Pr(t,Z|t_e,S)}$;
5      if $\alpha \geq q$ then
6          Accept $\boldsymbol{t'_e}$ and $S'$;
7      end
8  end
9  Compute the maximum of $Pr(\boldsymbol{t},Z|\boldsymbol{t_e},S)$ for accepted $\boldsymbol{t'_e}$ and $S'$;
10 return $\boldsymbol{t_e}$ and $S$;
```

superscript. Overall, we obtain the following likelihood:

$$
\begin{aligned}
Pr(\boldsymbol{t}|\boldsymbol{t_e},S) &= \prod_{i}^{N_i} \prod_{q}^{N_q} \prod_{j}^{N_j} P_q P_q^j \\
&= \prod_{i}^{N_i} \prod_{q}^{N_q} \prod_{j}^{N_j} \frac{N(t_i^q) N(S^j_{t_i^q})}{N(t_i) N(S^j_{t_i})}.
\end{aligned} \tag{4}
$$

where N_i is the number of texts in a text sequence, N_q is the number of candidates of a text, N_j is the number of text sequences in a group.

Likelihood $Pr(Z|\boldsymbol{t_e},S)$. $Pr(Z|\boldsymbol{t_e},S)$ measures the likelihood of the sequence relationship among texts in the local coordinate system. The intuition is that the best sequence relationship among texts in a group maximizes the *coincidence degree* of text sequences. In this paper, the coincidence degree of text sequences is measured by the Jaccard similarity. Therefore, the likelihood term can be estimated by computing the Jaccard similarity among texts and among text sequences. $sim(t_i^j, t_{i'}^{j'})$ is the Jaccard similarity between the i-th text in the j-th text sequence and the i'-th text in the j'-th text sequence. $sim(T_j, T_{j'})$ is the text Jaccard similarity between the j-th text sequence and the j'-th text sequence, where $i \neq i'$ and $j \neq j'$. Those Jaccard similarities can be calculated using Eq. 2. Thus, we obtain the following likelihood:

$$
Pr(Z|\boldsymbol{t_e},S) = \prod_{j}^{N_j} \prod_{i}^{N_i} sim(t_i^j, t_{i'}^{j'}) sim(T_j, T_{j'}). \tag{5}
$$

where N_i is the number of texts in a text sequence, N_j is the number of text sequences in a group.

Estimation Algorithm. Our goal is to estimate the true value of texts and the sequence relationship among texts to maximize Eq. 3. Therefore, an estimation algorithm is proposed to solve the MLE problem. Algorithm 1 shows the details of our estimation algorithm.

To initialize the true value of texts t_e and the sequence relationship of texts S, the largest number of texts in the text sequence are selected from $\boldsymbol{t}$. To maximize $Pr(\boldsymbol{t},Z|\boldsymbol{t_e},S)$, we sample $\boldsymbol{t_e}$ and S in each position in the text sequence using the Metropolis sampling method [18]. For each text, we first propose new $\boldsymbol{t_e}$ and S (denoted as $\boldsymbol{t'_e}$ and S') with Gaussian probability and then compute the acceptance ratio (Lines 3-6 in Algorithm 1). The process continues until all $\boldsymbol{t'_e}$ and S' are checked.

The final parameters are obtained by maximizing $Pr(\boldsymbol{t}, \boldsymbol{Z}|\boldsymbol{t_e}, \boldsymbol{S})$ for accepted $\boldsymbol{t'_e}$ and $\boldsymbol{S'}$ (Line 9 in Algorithm 1).

4.3 Indoor Floorplan Updating

Once text sequences are obtained, the next step is to update texts at correct locations of the indoor semantic floorplans using instant videos. Given a collected text sequence T=$\{t_j|0 \leq j \leq n\}$, where n is the number of the texts in the text sequence, a text set M=$\{t_i|0 \leq i \leq m\}$ of an indoor floorplan, where m is the number of texts in the indoor floorplan. We need to locate the text sequence, detect and remove the out-of-date texts, and then label the new text in the indoor floorplan.

4.3.1 Location Inference of Text Sequence. Location inference of a text sequence in the text set of an indoor floorplan is not trivial. To formally define the problem, let query text sequence be denoted as the text sequence from new collected data and data text sequence be denoted the text sequence in the set of an indoor floorplan. Each data text sequence is partitioned by the text set of the indoor floorplan and its length is the same as that of the query text sequence. Naive methods such as using the number of same texts between a query text sequence and the data text sequences could be used, but may produce a false match. That is because there are several shops with the same name in different areas of a mall.

To address this issue, we propose a location inference algorithm of text sequence based on the similarity between the query text sequence and the data text sequences, as calculated using Eq. 2. The data text sequences with top-5 scores are selected as candidate text sequences.

Second, given a vector $\boldsymbol{T_a}$ of a query text sequence with length n and a vector $\boldsymbol{T_b}$ of a candidate text sequence with length m. The similarity score of the two vectors is calculated by the Longest Common Subsequence (LCS) metric [20]. Given the system metric δ and matching threshold ϵ, the LCS metric for the two vectors is defined as follows:

$$L(\boldsymbol{T_{a,n}}, \boldsymbol{T_{b,m}}) = \begin{cases} 0, \text{if } n{=}0 \text{ or } m{=}0; \\ 1{+}L(\boldsymbol{T_{a,n\text{-}1}}, \boldsymbol{T_{b,m\text{-}1}}), \text{if } d(\boldsymbol{t_a}, \boldsymbol{t_b}) {\leq} \epsilon \text{ and } |n\text{-}m| {<} \delta; \\ \max(L(\boldsymbol{T_{a,n}}, \boldsymbol{T_{b,m\text{-}1}}), L(\boldsymbol{T_{a,n\text{-}1}}, \boldsymbol{T_{b,m}})), \\ \quad \text{otherwise.} \end{cases} \tag{6}$$

where δ is the maximum length difference between two text sequences and ϵ is the distance threshold.

The similarity score S_T is defined as:

$$S_T = \max_{f \in F} \frac{L(\boldsymbol{T_a}, f(\boldsymbol{T_b}))}{\min(n, m)}, \tag{7}$$

where F represents a set of sliding windows. The candidate text sequence with the largest S_T is selected.

4.3.2 Text Localization. Once those text sequences are localized, the next step is to detect the out-of-date, new, and unchanged texts, and then to remove those out-of-date texts and label those new texts in the floorplan.

Structure from Motion (SFM) technique is used to reconstruct the 3D model using a set of images captured in the scene from different viewpoints [36]. Specifically, the colmap [36] is used in SiFi. It recovers a representation of a scene using the SFM technique, resulting in point clouds in the reference coordinate system, locations of 2D feature points in images, correspondences between 3D point clouds and 2D feature points, and pose of the cameras for images. Each point in point clouds represents a physical point in the scene and each 2D feature point detected by Scale Invariant Feature Transform (SIFT) algorithm [26]. The location of a text in a query text sequence can be

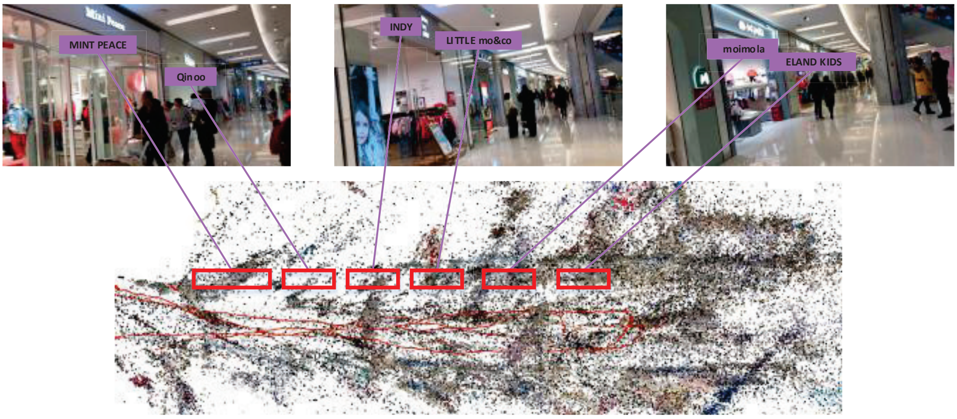


Fig. 4. An illustration of the text localization method. The bottom figure represents the sparse point clouds of the part of the indoor scene generated by the colmap algorithm. The red rectangles in the point clouds represent the text locations extracted from the top three images in the indoor scene.

obtained using the point clouds and 2D feature points. As illustrated in Fig. 4, the sparse point clouds are generated by the colmap algorithm. The red rectangles in the point clouds represent the text locations extract from the three images.

Note that, the text location is expressed in terms of the reference coordinate system obtained by the colmap algorithm. Therefore, we need to align the reference coordinate system with the floorplan coordinate system, described as follows. Assuming that most people were holding the phone in portrait orientation during recording videos. The pose of a camera for each image provides a cue to align the reference coordinate system and the floorplan coordinate system. First, a plane is fitted using the camera poses resulting from colmap algorithm and its normal vector ($\boldsymbol{v}$) is estimated using the Singular Value Decomposition (SVD) solutions [7]. Let vector $\boldsymbol{y}$ be the axis y direction in the reference coordinate system. Thus, given a vector $\boldsymbol{x}$ in the reference coordinate system, a vector $\boldsymbol{x'}$ in the floorplan coordinate system can be calculated by Rodrigues' formula[1] as:

$$\boldsymbol{x'}{=}cos(\theta)\boldsymbol{x}{+}(1-cos(\theta))(\boldsymbol{u}\cdot\boldsymbol{x})\boldsymbol{u}{+}sin(\theta)(\boldsymbol{u}\times\boldsymbol{x}), \tag{8}$$

where, $\boldsymbol{u}{=}\boldsymbol{v}\times\boldsymbol{y}$ and θ is the included angle between vectors $\boldsymbol{v}$ and $\boldsymbol{y}$. Therefore, the location of a text in the reference coordinate system can be transformed into the floorplan coordinate system using the Eq. 8.

4.3.3 Updating Indoor Floorplan. Given two text sequences T^a and T^b, n and m are the number of texts in T^a and in T^b, respectively. t_i^a is the i-th text in T^a and t_j^b is the j-th text in T^b, $t{\in}\{t_i^a{=}t_j^b, i{=}1,\ldots,n, j{=}1,\ldots,m\}$. Thus, our objective is to maximize N_t, where N_t is the number of t. To solve this problem, we conduct exhaustive search in T^a and T^b. Accordingly, these aligned texts of T^a and T^b are unchanged texts (called landmarks). The texts $t{\in}\{t_k|t_k{\in}T^a, t_k{\notin}T^b, k{=}1,\ldots,N_t\}$ are the out-of-date texts. The texts $t{\in}\{t_k|t_k{\notin}T^a, t_k{\in}T^b, k{=}1,\ldots,N_t\}$ are the new texts. Therefore, the out-of-date texts are removed and the new texts are directly labeled in the indoor semantic floorplan.

Once a set of sufficient number of query text sequences are available, the update procedure is executed once to adapt the current indoor semantic floorplan to the newer measurements. The

[1] http://mathworld.wolfram.com/RodriguesRotationFormula.html

Algorithm 2: Updating Algorithm for indoor floorplans

```
Input: The initial indoor floorplan $M_0$ and its text sequence $TM$, the newest set of text sequences $TS$.
Output: The complete updated indoor floorplan $M_t$.
for each text sequence $T \in TS$ do
    Localize $T$ in $TM$; (Section 4.3.1)
    3D modeling using the colmap algorithm and align the coordinate systems; (Section 4.3.2)
    for each text $t \in T$ do
        Get the position of each text using the 3D model and the 2D features of the text in a key-frame;
          (Section 4.3.2)
    end
    Find the out-of-date and the new texts in $T$, remove the out-of-date texts, and label the new texts in $M_0$;
      (Section 4.3.3)
end
```

updating algorithm is shown in Algorithm 2. The indoor floorplans can be always up-to-date thanks to a timely adaptation to indoor environmental changes.

5 IMPLEMENTATION AND EVALUATION

In this section, we describe the implementation details of SiFi, and the evaluation setups. We then test the performance of each component of SiFi.

5.1 Implementation

The SiFi prototype consists of two parts: a mobile application and an updating pipeline working on a server.

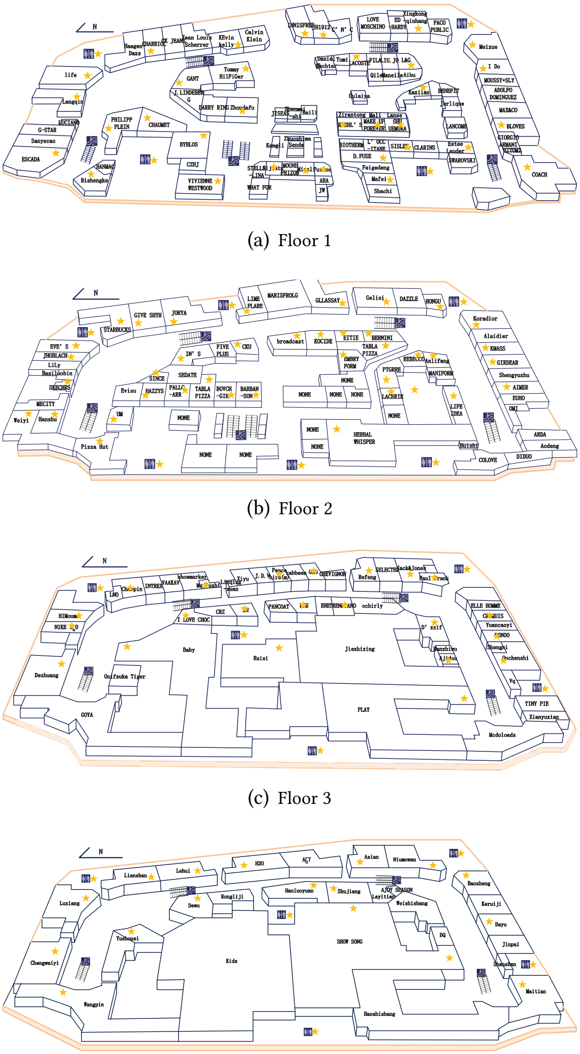
Mobile Application. The mobile application software was used by the crowd to record semantic-rich instant videos with timestamps. It was installed in different Android mobile devices with WiFi and cameras. Videos were automatically compressed and divided into 6MB chunks data for transmission through WiFi network.

Server Configuration. The indoor semantic floorplan updating pipeline was implemented on a Lenovo computer with 32GB RAM, an i7 CPU processor, a 12GB Titan GPU, and WiFi device. The pipeline was implemented in Ubuntu Linux using two threads, one is used to receive and store incoming videos, the other is used to process videos and produce the newest indoor semantic floorplan.

5.2 Evaluation Methodology and Setups

Extensive experiments were conducted on data collected in Wanda mall, Changsha, China. It has four floors with over 250 stores. The floorplan of Wanda mall contains rich information including store names, store location, promotion information, and widths of the corridors. Figure 5 shows the semantic floorplan of four floors in Wanda mall. During the initialization of our experiment, all texts (e.g., store names, promotion information, and restrooms) were labeled manually, since the available floorplans of the shopping mall were out of date.

Five volunteers were invited to participate in the data collection procedure. Each volunteer carried a mobile device to obtain semantic-rich instant videos at different times of a day. Application software was installed in the mobile devices for automatic video collection. The walking path of each volunteer for video acquisition was determined by the volunteer. Each volunteer was asked to cover the entire experimental area as much as possible. Eventually, we found that the walking paths of volunteers covered 80% of all the routes in the shopping mall. The videos were collected



(a) Floor 1

(b) Floor 2

(c) Floor 3

(d) Floor 4

Fig. 5. Updating results achieved on four floors of the Wanda shopping mall. Store names are directly marked in indoor semantic floorplans, other texts are marked in indoor semantic floorplans using golden stars.

in most areas of the shopping mall and to cover all available texts. In addition, these videos are different in many aspects, e.g., the area and size they covered. The dataset was used to test the performance of each component of SiFi.

5.3 Performance Evaluation

5.3.1 Performance of Indoor Semantic Floorplan Updating. Precision is the most important performance metric for indoor semantic floorplan updating. Five volunteers were invited to participate in this experiment. The Wanda mall has more than 250 store names and 100 other texts (e.g., promotion posters). Store names are directly marked in the indoor semantic floorplans, while other texts are marked in the indoor semantic floorplans using golden stars. The updated indoor semantic floorplan achieved by SiFi on four floors of Wanda mall.

To measure the performance of SiFi, the S_{acc} metric is used:

$$S_{acc}=\frac{N_{corr}}{N_t}\times 100, \tag{9}$$

where N_t is the number of the manually marked ground-truth changed texts on the whole floorplan. N_{corr} is the number of correct texts updated by SiFi. We tested the text updating performance achieved by SiFi and the Pure Optical Character Recognition (POCR) based method [40]. Experiments were conducted on four floors of Wanda mall. SiFi achieves an average accuracy of the text updating about 84.5%, while the POCR-based method achieves about 78.4%. That is because SiFi uses sequential relationship among texts while the POCR-based method uses individual text only. The error sources of our method are from the text recognition errors and localization errors of texts (See Section 5.3.4).

5.3.2 Performance of Crowdsourcing Task Model. To test the performance of crowdsourcing task model, extraction ratio of the texts (r_s) is calculated by dividing the number of extracted texts with the number of texts in an indoor space. Videos were recorded under two cases: with or without our crowdsourcing task model. Each video is about 200 seconds long. It can be seen from Table 2 that the extraction ratio of texts is higher with our crowdsourcing task model as compared to the results achieved without our crowdsourcing task model. That is because videos recorded with our crowdsourcing task model contains clear images and richer texts. Therefore, these individual videos recorded with our crowdsourcing task model provide more valuable information about an indoor scene as compared to those recorded without our crowdsourcing task model.

Accuracy of Floor Number Inferring. We further evaluated the floor number inferring performance based on two walking states, including walking up the stairs and walking down the stairs. The input floor numbers are floor 1, floor 2, floor 3, and floor 4. It can be seen from Table 3 that SiFi achieves a floor number inferring accuracy of 100% for any initial floor number. s

5.3.3 Performance of Text Extraction. This section gives the details of the performance of key-frame selection, the performance of text sequence merging, and the accuracy of text extraction.

Performance of Key-frame Selection. The key-frame selection performance depends on the threshold for ORB feature number TH_K and the match percentage R_K. TH_K and R_K values can be determined using the number of key-frames in a specific indoor scene, and a tradeoff lies between

Table 2. Extraction ratio of texts (r_s)

Floor	1	2	3	4
With our task model	0.89	0.91	0.91	0.90
Without our task model	0.71	0.75	0.73	0.75

Table 3. The floor number inferring accuracy produced by SiFi

		Input floor number			
		1	2	3	4
Inferred floor number	1	-	100%	100%	100%
	2	100%	-	100%	100%
	3	100%	100%	-	100%
	4	100%	100%	100%	-

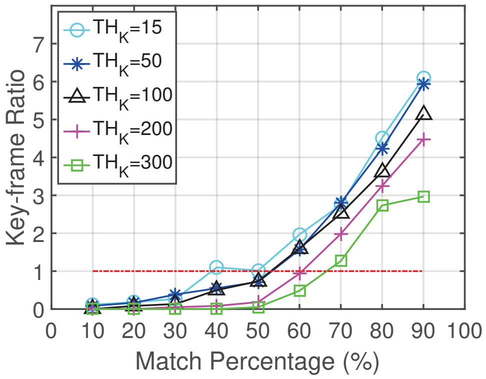


Fig. 6. Effect of the key-frame selection parameters.

the number of extracted texts from a video and the computational cost. We tested the key-frame selection performance (denoted as key-frame ratio) with respect to different TH_K and R_K on 20 videos. The key-frame ratio is calculated by dividing the number of key-frames selected by SiFi with the baseline number of key-frames in a video. The baseline number of key-frames is defined as the smallest number of frames to cover all texts in a video. The average results are given in Fig. 6.

It can be found that as TH_K increases, the key-frame ratio decreases. As R_K increases, the key-frame ratio increases. When TH_K=200 and R_K=60% or TH_K=15 and R_K=50%, the key-frame ratio is close to 1. Therefore, we set TH_K=200 and R_K=60% in this paper as this setting makes the key-frame selection process more robust to camera motion.

Performance of Text Sequence Merging. The text sequence merging performance depends on the threshold for text Jaccard similarity τ. The text Jaccard similarity is used to divide text sequences into several groups for accurate text inferring. Two text sequences may describe the texts of the same scene due to text changes. Therefore, the text referring performance can be improved if a reasonable τ is set. We tested the false positive (FP_T) and false negative (FN_T) results of text grouping with respect to different τ values. FP_T is calculated by dividing the number of similar text sequences with the number of text sequences of a scene. FN_T is calculated by dividing the number of different text sequences which are considered to be similar with the number of text sequences of a scene. We randomly selected 20 text sequences of indoor scenes. The average results are given in Table 4. It is found that as τ is decreased, FP_T is increased and FN_T is decreased. Specifically, when τ is set to be 0.9, FP_T=0.05 and FN_T=0.05. Therefore, we set τ to 0.9 in SiFi.

Accuracy of Text Extraction. We then tested the text recognition accuracy achieved by our text extraction method and the method proposed in [40]. Volunteers were invited to record semantic-rich instant videos in different floors. The ground-truth of texts were labeled manually. The precision-recall metric is used to measure text recognition performance. Specifically, given the ground truth of text sequences TX_{true} and the result TX_{pro} produced by SiFi, precision P is calculated by dividing the number of correct texts with the number of texts in TX_{pro}. Recall R is calculated by dividing

Table 4. Effect of the text sequence grouping parameter

τ	1.0	0.9	0.8	0.7
FP_T	0.00	0.05	0.11	0.23
FN_T	0.09	0.05	0.02	0.01

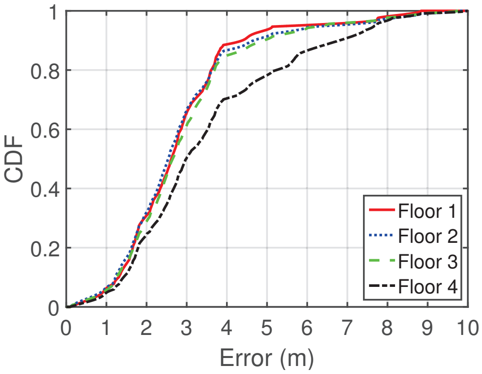


Fig. 7. The localization accuracy of new texts achieved by SiFi.

the number of correct texts with the number of texts in TX_{true}, that is:

$$P=\frac{|TX_{pro}\bigcap TX_{true}|}{|TX_{pro}|}, \tag{10}$$

$$R=\frac{|TX_{pro}\bigcap TX_{true}|}{|TX_{true}|}, \tag{11}$$

$$F=2\times\frac{P\times R}{P+R}, \tag{12}$$

where F represents the harmonic mean of precision and recall. Our method outperforms the method in [40]. The F value achieved by our method is 0.81, while the F value achieved by the method in [40] is less than 0.76 in all cases. The accuracy improvement in text recognition clearly demonstrates the effectiveness of our text extraction method.

5.3.4 Performance of Text Localization. This section gives the details of the performance of text sequence localization and the accuracy of text localization.

Performance of Text Sequence Localization. We tested the text sequence localization performance using localization rate, which is defined as the percentage of text sequences that can be correctly matched with the text sequence of the indoor semantic floorplan. If the text sequence estimated by our method matches the ground truth, we consider this localization as a correct localization. Otherwise, it is considered as an incorrect localization. The localization rate is calculated by dividing the number of correctly located text sequences with the total number of text sequences. As shown in Table 5, SiFi achieves an average localization rate of 95.7% for four floors in the mall.

Accuracy of Text Localization. We further tested the text localization performance. Specifically, the new text localization errors were calculated on four floors of the Wanda shopping mall. The results are given in Fig. 7. It is found that 50% of new texts have an error of less than 2.6m on floors 1 to 3. Besides, 50% of new texts have an error of less than 3.0m on the fourth floor. That is because the semantics on the fourth floor are relatively sparse than other floors, as shown in Fig. 5.

Table 5. Localization rate under different energy term configurations

Floor	Floor 1	Floor 2	Floor 3	Floor 4
Localization Rate	95%	93%	98%	97%

5.3.5 Other Factors. This section gives the details of the system performance to response delay and the energy consumption.

Response Delay. We deployed SiFi using a server equipped with a 32GB RAM, an i7 CPU processor, and a 12GB Titan GPU. The response delay was evaluated using 32 videos and each lasting about 30 seconds. 7880 key-frames were extracted. In our experiments, the colmap algorithm takes about 21 hours and other algorithms only take less than 1 hour totally. Therefore, SiFi achieves the day-level latency and the response delay is also determined by the number of videos for one update. Note that, the response delay can further be reduced using more powerful machines.

Energy Consumption. We tested the energy consumption of our SiFi mobile application software, including those consumed by cameras and WiFi network. The energy is calculated using the PowerTutor profiler [16] in a Google Nexus 7 tablet. During the experiment, we turned off all background applications and additional hardware components. The energy consumed by camera and WiFi network is 6.9 Joule and 1.6 Joule for a 6-second-long video, respectively. Compared to the battery capacity of 20k Joules, video capturing and uploading do not constitute any signification power consumption for a mobile device [15].

6 DISCUSSION AND LIMITATIONS

Although several promising results have been reported in our experiments, SiFi still has several limitations.

System Robustness and Scalability. SiFi was evaluated by different users using different mobile devices and images on various floors in a shopping mall. These extensive experiments have demonstrated the robustness of SiFi under a wide range of scenarios. SiFi can be extended to a worldwide scale using existing indoor semantic floorplans of indoor environments and the cloud. Particularly, semantics updating of each indoor space can be performed independently based on GPS and barometer data tagged on videos. It is true that the updating performance of our method will be decrease if most texts are not recorded on videos due to occlusion of crowds. In the future work, a pop will be added in our mobile applications of data collection to tips volunteers to avoid crowds.

Mixed Modality with More Techniques. In SiFi, we mainly use instant videos to extract semantic sequence and then update out-of-date semantics of annotated objects by locating these semantic sequences in indoor floorplans. Recent advances in indoor localization, especially those supported by mobile devices, have enabled meter or sub-meter level accuracy of localization [39, 40]. In the future, we would like to incorporate those techniques to build a mixed modality for semantic updating. For example, the localization accuracy of semantics can be improved by exploiting the walking traces and positions of users in indoor space.

Other Indoor Environments. Although SiFi is evaluated in a shopping mall, it can be customized to other indoor environments with a similar building structure, e.g., exhibition buildings. Moreover, SiFi can be extended to other types of buildings using appropriately adjusted parameters. Specifically, semi-unsupervised or unsupervised learning techniques [38] can be used to extend our method to new indoor environments. SiFi may have difficulties in indoor environments with a large open area and a small number of rooms, such as supermarkets. In the future, novel texts localization algorithms will be proposed to extend the application scenarios of our system.

Information Privacy. Since users can share videos, SiFi may pose a risk of privacy leakage. For example, building owners might not allow others to share some videos and audio recorded in their buildings. Therefore, an information privacy protection mechanism should be further designed, such as people face blurring [13].

7 CONCLUSION

In this paper, we have presented a method for automatic and continuous indoor semantic floorplan updating. A system called SiFi is proposed to perform floorplan updating using semantic-rich instant videos acquired by ordinary mobile devices. With appropriate computer vision techniques, images and text sequences are used in SiFi to achieve indoor semantic floorplan updating in an efficient, low-cost, and scalable manner. Extensive experiments have been conducted in a shopping mall with over 250 stores. Experimental results within nine weeks demonstrate that SiFi can effectively address semantic variations caused by environmental dynamics.

ACKNOWLEDGMENTS

This work is partially supported by National Natural Science Foundation of China under Grant Nos. 61772544, 61602499, 61471371, and 71471175, National Basic Research Program (973 program) under Grant No.2014CB347800, the Hunan Provincial Natural Science Fund for Distinguished Young Scholars under Grant No.2016JJ1002, the Guangxi Cooperative Innovation Center of cloud computing and Big Data under Grant Nos. YD16507 and YD17X11, the Fundamental Research Funds for the Central Universities under Grant No. 18lgzd06, and the National Postdoctoral Program for Innovative Talents under grant No. BX201600172.

REFERENCES

[1] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. 2014. Word Spotting and Recognition with Embedded Attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 12 (2014), 2552–2566.

[2] M. Alzantot and M. Youssef. 2012. Crowdinside: Automatic construction of indoor floorplans. In *Proc. of SIGSPATIAL*. Redondo Beach, California.

[3] M. Angermann and P. Robertson. 2012. Footslam: Pedestrian simultaneous localization and mapping without exteroceptive sensorsąłhitchhiking on human perception and cognition. *Proc. of the IEEE* 100, Centennial-Issue (2012), 1840–1848.

[4] R. J. Bayardo, Y. Ma, and R. Srikant. 2007. Scaling up all pairs similarity search. In *Proc. of WWW*. Alberta, Canada.

[5] S. Becker, M. Peter, D. Fritsch, D. Philipp, P. Baier, and C. Dibak. 2013. Combined Grammar for the Modeling of Building Interiors. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2013), 1–6.

[6] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. 2013. PhotoOCR: Reading text in uncontrolled conditions. In *Proc. of IEEE ICCV*. Sydney, Australia.

[7] Tony F. Chan and Per Christian Hansen. 1990. Computing Truncated Singular Value Decomposition Least Squares Solutions by Rank Revealing QR-Factorizations. *SIAM Journal on Scientific Computing* 11, 3 (1990), 519–530.

[8] S. Chen, M. Li, K. Ren, and C. Qiao. 2015. CrowdMap: Accurate reconstruction of indoor floor plans from crowdscourced sensor-rich videos. In *Proc. of IEEE ICDCS*. Ohio, USA.

[9] S. Dhar and U. Varshney. 2011. Challenges and business models for mobile location-based services and advertising. *Commun. ACM* 54, 5 (2011), 121–128.

[10] M. Elhamshary and Y. Moustafa. 2015. SemSense: Automatic construction of semantic indoor floorplans. In *Proc. of IEEE IPIN*. Alberta, Canada.

[11] M. Elhamshary, M. Youssef, A. Uchiyama, H. Yamaguchi, and T. Higashino. 2016. TransitLabel: A crowd-sensing system for automatic labeling of transit stations semantics. In *Proc. of ACM MobiSys*. Florence, Italy.

[12] R. Ferzli and L. J. Karam. 2009. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB). *IEEE Transactions on Image Processing* 18, 4 (2009), 717–728.

[13] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent. 2009. Large-scale privacy protection in Google Street View. In *Proc. of IEEE ICCV*. Kyoto, Japan.

[14] H. Gao, J. Tang, X. Hu, and H. Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *Proc. of ACM RecSys*. Hong Kong, China.

[15] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li. 2014. Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In *Proc. of ACM MobiCom*. Maui, Hawaii.

[16] M. Gordon. 2013. PowerTutor: A power monitor for Android-based mobile platforms.

[17] X. Guo, E. C. Chan, C. Liu, K. Wu, S. Liu, and L. M. Ni. 2014. ShopProfiler: Profiling shops with crowdsourcing data. In *Proc. of IEEE INFOCOM*. Toronto, Canada.

[18] W. K. Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 1 (1970), 97–109.

[19] V. Hedau, H. Derek, and F. David. 2009. Recovering the spatial layout of cluttered rooms. In *Proc. of IEEE ICCV*. Kyoto, Japan.

[20] D. S. Hirschberg. 1977. Algorithms for the Longest Common Subsequence Problem. *J. ACM* 24, 4 (1977), 664–675.

[21] M. Jaderberg, A. Vedaldi, and A. Zisserman. 2014. Deep Features for Text Spotting. In *Proc. of ECCV*. Zurich, Switzerland.

[22] P. Jain, J. Manweiler, and R. R. Choudhury. 2015. OverLay: Practical mobile augmented reality. In *Proc. of ACM MobiSys*. Florence, Italy.

[23] Y. Jiang, X. Yun, X. Pan, K. Li, Q. Lv, R. P. Dick, L. Shang, and M. Hannigan. 2013. Hallway based automatic indoor floorplan construction using room fingerprints. In *Proc. of ACM UbiComp*. Zurich, Switzerland.

[24] I. A. Junglas and R. T. Watson. 2008. Location-based services. *Commun. ACM* 51, 3 (2008), 65–69.

[25] D. R. Karger, S. Oh, and D. Shah. 2014. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research* 62, 1 (2014), 1–24.

[26] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 2 (2004), 91–110.

[27] R. Meng, S. Shen, R. R. Choudhury, and S. Nelakuditi. 2016. AutoLabel: Labeling places from pictures and websites. In *Proc. of ACM UbiComp*. Heidelberg, Germany.

[28] K. Muralidharan, A. J. khan, A. Misra, R. K. Balan, and S. Agarwal. 2014. Barometric phone sensors: More hype than hope! (2014).

[29] M. Norouzi, D. J. Fleet, and R. Salakhutdinov. 2012. Hamming Distance Metric Learning. In *Proc. of Neural Information Processing Systems*. Lake Tahoe, Nevada.

[30] S. Panzieri, F. Pascucci, and G. Ulivi. 2002. An outdoor navigation system using GPS and inertial platform. *IEEE/ASME transactions on Mechatronics* 7, 2 (2002), 134–142.

[31] D. Philipp, P. Baier, C. Dibak, F. Dürr, K. Rothermel, S. Becker, M. Peter, and D Fritsch. 2014. Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data. In *Proc. of IEEE PerCom*. Budapest, Hungary.

[32] M. G. Puyol, D. Bobkov, P. Robertson, and T. Jost. 2014. Pedestrian simultaneous localization and mapping in multistory buildings using inertial sensors. *IEEE Transactions on Intelligent Transportation Systems* 15, 4 (2014), 1714–1727.

[33] J. A. Rodríguez-Serrano, A. Gordo, and F. Perronnin. 2015. Label embedding: A frugal baseline for text recognition. *International Journal of Computer Vision* 113, 3 (2015), 193–207.

[34] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *Proc. of IEEE ICCV*. Barcelona, Spain.

[35] C. Sadowski and G. Levin. 2007. Simhash: Hash-based similarity detection. In *http://simhash.googlecode.com/svn/trunk/paper/SimHashWithBib.pdf*.

[36] J. L. Schönberger and J. Frahm. 2016. Structure-from-Motion Revisited. In *Proc. of IEEE CVPR*. Las Vegas, NV.

[37] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. 2013. Walkie-Markie: Indoor pathway mapping made easy. In *Proc. of USENIX NSDI*. Lombard, IL.

[38] H. Shin, Y. Chon, and H. Cha. 2012. Unsupervised construction of an indoor floor plan using a smartphone. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 6 (2012), 889–898.

[39] X. Teng, D. Guo, Y. Guo, X. Zhou, Z. Ding, and Z. Liu. 2017. IONavi: An indoor-outdoor navigation service via mobile crowdsensing. *ACM Transactions on Sensor Networks* 13, 2 (2017), 12:1–12:28.

[40] S. Wang, F. Sanja, and U. Raquel. 2015. Lost shopping! Monocular localization in large indoor spaces. In *Proc. of IEEE ICCV*. Santiago, Chile.

[41] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. 2012. End-to-end text recognition with convolutional neural networks. In *Proc. of ICPR*. Tsukuba, Japan.

[42] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes. 2016. iCrowd: Near-optimal task allocation for piggyback crowdsensing. *IEEE Transactions on Mobile Computing* 15, 8 (2016), 2010–2022.

[43] D. Yang, G. Xue, X. Fang, and J. Tang. 2012. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proc. of ACM MobiCom*. Istanbul, Turkey.

[44] C. Yao, X. Bai, B. Shi, and W. Liu. 2014. Strokelets: A learned multi-scale representation for scene text recognition. In *Proc. of IEEE CVPR*. Columbus, OH.

[45] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang. 2015. Truthful incentive mechanisms for crowdsourcing. In *Proc. of IEEE INFOCOM*. Hong Kong.