SweepLoc: Automatic Video-based Indoor Localization by Camera Sweeping

MINGKUAN LI, Sun Yat-sen University, China
NING LIU*, Sun Yat-sen University, China
QUN NIU, Sun Yat-sen University, China
CHANG LIU, Sun Yat-sen University, China
S.-H. GARY CHAN, The Hong Kong University of Science and Technology, China
CHENGYING GAO, Sun Yat-sen University, China

Indoor localization based on visual landmarks has received much attention in commercial sites with rich features (e.g., shopping malls, museums) recently because landmarks are relatively stable over a long time. Prior arts often require a user to take multiple independent images around his/her location, and manually confirm shortlisted landmarks. The process is sophisticated, inconvenient, slow, unnatural and error-prone. To overcome these limitations, we propose *SweepLoc*, a novel, efficient and automatic video-based indoor localization system. SweepLoc mimics our natural scanning around to identify nearby landmarks in an unfamiliar site to localize.

In SweepLoc, a user simply takes a short video clip (about 6 to 8 seconds) of his/her surroundings by sweeping the camera. Using correlation and scene continuity between successive video frames, it automatically and efficiently selects key frames (where potential landmarks are centered) and subsequently reduces the decision error on landmarks. With identified landmarks, SweepLoc formulates an optimization problem to locate the user, taking compass noise and floor map constraint into account. We have implemented SweepLoc in Android platform. Our extensive experimental results in a food plaza and a premium mall demonstrate that SweepLoc is fast (less than 1 second to localize), and achieves substantially better accuracy as compared with the state-of-the-art approaches (reducing the localization error by 30%).

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**; • **Networks** → *Location based services*;

Additional Key Words and Phrases: Video-based indoor localization, key frame selection, scene-continuity constraint

ACM Reference Format:
Mingkuan Li, Ning Liu, Qun Niu, Chang Liu, S.-H. Gary Chan, and Chengying Gao. 2018. SweepLoc: Automatic Video-based Indoor Localization by Camera Sweeping. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 120 (September 2018), 25 pages. https://doi.org/10.1145/3264930

*This is the corresponding author

Authors' addresses: Mingkuan Li, Sun Yat-sen University, Guangzhou, Guangdong, 510006, China, limkuan@mail2.sysu.edu.cn; Ning Liu, Sun Yat-sen University, Guangzhou, Guangdong, 510006, China, liuning2@mail.sysu.edu.cn; Qun Niu, Sun Yat-sen University, Guangzhou, Guangdong, 510006, China, niuqun@mail2.sysu.edu.cn; Chang Liu, Sun Yat-sen University, Guangzhou, Guangdong, 510006, China, liuchng2@mail2.sysu.edu.cn; S.-H. Gary Chan, The Hong Kong University of Science and Technology, Hong Kong, Hong Kong, 510006, China, gchan@cse.ust.hk; Chengying Gao, Sun Yat-sen University, Guangzhou, Guangdong, 510006, China, mcsgcy@mail.sysu.edu.cn.

2474-9567/2018/9-ART120 $15.00
https://doi.org/10.1145/3264930

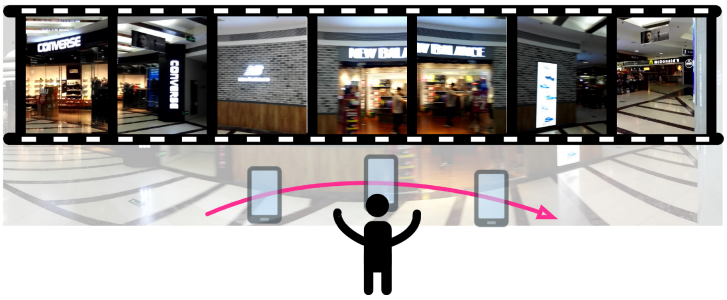


Fig. 1. A user sweeps the camera to take a video of the environment and records the compass bearings.

1 INTRODUCTION

Indoor localization using visual landmarks, such as store logos, signs and wall paintings have attracted much attention recently [6, 33, 34, 38]. This is because indoor landmarks are visually conspicuous (easy to notice and distinguishable from the background and nearby environment), pervasive and stable. Thus, they are easily distinguishable and provide long-term stable location clues (in months or years) as compared with other signals. For example, radio frequency signals, such as Wi-Fi, Bluetooth Low Energy (BLE), are easily affected by multi-path propagation, signal reflection and device orientation, thus they may not provide stable localization in the complicated indoor environment. In addition, they have to deploy and maintain a large set of devices such as Wi-Fi access points or Bluetooth beacons, which significantly increases the deployment and maintenance cost of such localization systems. In a visual localization system, surveyors first collect training images of landmarks and construct a database (the offline stage). In the online stage, a user queries his/her location using the phone camera. A localization algorithm returns the estimated location after comparing the images stored in the database.

Although traditional vision-based localization approaches have achieved sufficient accuracy, they have several limitations for practical applications. First, many of them require intrusive user operations, such as distinguishing landmarks, pointing to them and taking multiple independent *images* with their cameras [34, 39]. Even more, some require users to manually confirm shortlisted landmarks [6, 38], which is error-prone for novice users.

To address above limitations, we propose a novel video-based indoor localization system termed SweepLoc. It mimics the natural visual behavior of human to localize oneself in an environment: we scan around and determine our locations based on nearby visual landmarks. We illustrate SweepLoc in Figure 1, where a user first takes a short video clip by sweeping the camera (about 6-8 seconds). Based on the hundreds of image frames collected, the localization system selects the best frames with the landmarks in the center (i.e., *key frames*) and then triangulates the user given the noisy compass reading. Note that in this paper we use "user" and "client" interchangeably to refer to a human, a smartphone, a robot, a drone, or a wearable device (e.g. Google Glass [47]) with onboard camera.

SweepLoc is natural to use, intuitive to unskilled users, automatic in operation and highly accurate in localization (due to many image frames to select from). It is general enough to integrate with other landmark-based localization system based on key frames selection. It is particularly applicable to indoor environment with abundant visual clues, textures and signs, such as food plazas, shopping malls, museums, etc. Even in a site with fewer visual landmarks (such as hospitals or offices), it can be integrated with other features such as line segments, optical characters (such as doorplates), RF signals (such as Wi-Fi) or geomagnetism to enhance its accuracy.

The key contributions of SweepLoc are as follows:

- *Efficient key frame selection based on ROI tracking:* Previous works [34, 39] require users to choose landmarks and point to them, which is tedious and inefficient. Different from them, we propose a key frame selection scheme with *Faster Region-based Convolutional Neural Network* (Faster R-CNN) [24]. Based on Faster

R-CNN, our system is able to detect Regions of Interest (ROIs) with potential landmarks and selects key frames with ROI in the center automatically, which is efficiency. Nevertheless, it is time-consuming to select key frames by frame-wise detection in a video. To reduce the time consumption, we employ a novel *ROI* tracking strategy based on temporal correlation between frames. Moreover, we utilize landmark region for preliminary identification or ROIs, which reduces the impact of crowded pedestrians in indoor videos.

- *Automatic landmark identification based on scene continuity:* Due to view differences, distortion and motion blur, it is difficult to identify each individual landmark accurately in an image. We compute the transitional probability between landmarks and formulate a Hidden Markov Model (HMM) to jointly identify landmark sequence in the video. By considering the scene continuity, we are able to significantly reduce decision errors and find the landmarks accurately.
- *Accurate localization given noisy compass reading:* After selecting landmarks in the video, we formulate a localization problem which simultaneously considers compass noise and refines the bias estimation through float-encoded genetic algorithm. To further improve the localization accuracy, we constrain the estimated location to accessible regions in the floor plan.

We prototype SweepLoc and conduct extensive experiments in a food plaza and a premium mall. Experimental results demonstrate that SweepLoc is fast (localize within one second), and achieves much higher localization accuracy (improve by more than 30%) compared with state-of-the-art Sextant [6] and MoVIPS [34].

Realizing accurate and automatic visual indoor localization plays a fundamental role in a wide range of compelling applications, one of which is location sharing between friends. When two close friends decide to meet in an unfamiliar shopping mall, they may find it difficult to find each other. In this case, one may open the camera, sweep around. Then our system is able to infer the current location, share with friends and help them find each other quickly. Our system can also be deployed on mobile and wearable platforms, such as Google Glass, to help eye-impaired users due to its high automation and low localization error. According to recent researches, these users can benefit from the localization system to explore indoor sites by themselves, thus help to improve the quality of their lives [8, 13]. In addition, automatic video-based localization can benefit the indoor robotic navigation as well. For example, a robot (e.g., a wheeled robot or a drone) can initialize its starting location by sweeping the camera. Then it can use the inertial navigation system (INS) to infer consecutive locations accurately and efficiently without building a complicated indoor 3D model as Simultaneously Localization And Mapping (SLAM) or Structure from Motion (SfM) does. Furthermore, these robots can opportunistically calibrate the noisy INS by sweeping for another time. Since the video taking and localization is fully automatic, it is able to find the current location easily without human intervention. Furthermore, video localization can enhance the user experience of augmented reality (AR) based gaming (e.g., Pokémon Go!) as well. As these AR games usually require location priors, our system can provide localization services transparently, which can improve the user experience significantly.

The rest of this paper is organized as follows. We review the related work in Section 2, followed by system overview in Section 3. We elaborate our automatic ROI detection algorithm in Section 4, and landmark sequence identification in Section 5. In Section 6, we formulate the localization problem to estimate the user position. We present illustrative experimental results in Section 7, followed by our discussion and future work in Section 8. Finally, we conclude in Section 9.

2 RELATED WORK

Since the GPS signal is usually not available indoors, researchers begin to study indoor localization with various other signals, such as Wi-Fi [19, 28, 42], Bluetooth [16, 50] magnetic field [9, 22], visible light [32, 49] and acoustic signals [37]. Of all above signals, Wi-Fi based localization has emerged as a promising one due to the pervasiveness

Table 1. Qualitative comparisons of SweepLoc and the RF-based localization.

Category	Scheme	Device Dependency	Calibration Cost	Localization Accuracy	Maintenance Expectation
Vision-based Localization	SweepLoc	No	Low	2~4m	Monthly
RF-based Localization	Wu et al. [36]	Yes	Medium	<2m	Daily
	Ye et al. [42]		High	~3m	Weekly
	Liu et al. [17]		High	~1m	
	Luo et al. [19]		Low	1~3m	
	Wu et al. [35]		Low	~2m	
	Zuo et al. [50]		High	<3m	

of Wi-Fi devices, such as Wi-Fi interface cards. Other signals, such as Bluetooth, visible light and sound, require deploying specific devices, which significantly increases the deployment and maintenance overhead.

Recent Wi-Fi based indoor localization can be broadly classified into two categories: geometry-based approaches and fingerprint-based approaches. Previous geometry-based Wi-Fi localization employs triangulation of arrival angles or time of flight [36] to determine the current location. Although efficient, they usually require line-of-sight with Access Points (APs), which is not always available due to indoor obstructions. With the multipath, the localization error increases. Wi-Fi fingerprint-based indoor localization does not require line-of-sight with APs, thus it is more robust indoors. RADAR [1] and Horus [43] pioneer the Wi-Fi fingerprint-based indoor localization. However, Wi-Fi signal is not stable indoors due to environmental changes, such as moving pedestrians, opening and closing of doors. This renders collected fingerprint inaccurate and degrades the localization accuracy. To address this, recent researchers study using spatial or temporal signal patterns to reduce the signal fluctuation. For example, Ye et al. [42] exploit the spatial dependency among Wi-Fi fingerprints with sub-sequence dynamic time warping algorithm. Tian et al. [29] improve the indoor localization accuracy by leveraging the temporal correlation of the fingerprints. Liu et al. [17] propose a localization framework to learn the distance metrics between fingerprints based on transfer learning. Furthermore, more recent studies update signals with crowdsourcing. Luo et al. [19] propose self-calibrating radio map by opportunistically data collection of smartphone users. Wu et al. [35] propose a survey-free localization system, which automatically associated unlabeled radio map generated by crowdsourcing to indoor floor plan.

Despite promising results [13, 41], current RF-based indoor localization, such as Wi-Fi, BLE, are device-dependent. This could incur deployment or maintenance overhead and reduces the deploy-ability of such systems. We summarize the strengths and weaknesses of RF-based state-of-the-arts in Table 1. In contrast to RF-based localization systems, SweepLoc requires less calibration cost due to its stability and achieves sufficient accuracy with distinguishing visual features. Thus, the proposed system is more deployable.

Vision-based indoor localization approaches can be divided into two categories: model-based and retrieval-based approaches. Model-based approaches locate users based on 3D model of the test site constructed by SfM. Argus [38] builds a coarse model for each landmark and reduces Wi-Fi localization error through the relative distances derived though SfM. Different from Argus, iMoon [3] builds a comprehensive point cloud with SfM. To reduce the time consumption of direct image registration, it partitions the 3D model and uses the Wi-Fi to infer the candidate partition of current user location. Dong et al. [4] derive a floor plan from 3D point cloud and track pedestrians using particle filters to track the client continuously.

Instead of constructing the 3D model for the whole test site, recent works build models for each single landmark. ClickLoc [39] infers the possible region based on the camera focus and size of the landmark in the image. Then, it locates the client with Wi-Fi fingerprint. Knitter [7] estimates the spatial and geometrical relationship, and

calibrate the current user location using ceilings and ground in the view. Since these require selecting landmarks and taking specific images, they are tedious for novice users. Our proposed SweepLoc is significantly different from the above in several aspects. First, it does not require tedious landmark selection and image taking. Instead, it just asks a user to open the camera and sweep. Then SweepLoc can find landmarks in videos automatically and efficiently with high accuracy, thus increasing the usability of visual localization systems. Second, it is computationally expensive to construct a dense point cloud for a large indoor site. Even more, a large test site may have millions of 3D points in the point cloud, which incurs severe memory consumption [26]. Third, SweepLoc does not introduce additional signals into the system, thus it is able to reduce the survey cost and more deployable.

Another category of model-based localization is SLAM [20], which has many advantages such as no need for prior site configuration. However, it needs to store densely sampled key frames, and to establish the frame-to-frame correspondence, which could incur heavy computational load [14]. Apart from that, SLAM cannot locate the client upon launching the system. SweepLoc, however, does not require initial location and can be integrated into other localization systems to provide initial location or sensor calibration.

Compared with model-based indoor localization approaches, retrieval-based methods do not need to construct a complicated 3D point cloud or align models with the floor plan, thus they are more deployable. Piciarelli et al. [23] locate the user by comparing the query image to geo-tagged visual features. Travi-navi [48] navigates users with images. These are different from SweepLoc in that they determine the current location based on the geo-location of matched images, which requires dense samples to achieve high accuracy. Different from these, SweepLoc achieves accuracy with sparse images by data augmentation and employs rich sensibilities of smartphone to increase its accuracy. Werner et al. [34] further consider the distance between the input image and the most matched one in the database. Although efficient, it is not sufficiently accurate with coarsely sampled images. Apart from that, the employed local point descriptors (e.g. SURF [2], ORB [25]) are prone to blur and repetitive textures. SweepLoc however, estimates all landmarks in the video simultaneously with the indoor floor plan. Thus, it is able to reduce random noise and achieve higher accuracy. Sextant [6] locates users with three independent images. It requires user confirmation, which is cumbersome and difficult for ordinary users. Different from Sextant, our proposed SweepLoc does not involve user confirmation and thus is more automated. Wang et al. [31] employ optical character recognition to facilitate landmark recognition and localization. This may not always work well in practice as texts are difficult to recognize due to long distance, varied fonts, colors and sizes compared with printed texts.

Due to the abundant information derived from video, researchers also study video-based localization. Papaioannou et al. [21] fuse surveillance cameras and Wi-Fi received by workers in the construction site for tracking. Their method is significantly different from ours because they use videos from surveillance cameras rather than devices carried by users to localize. VMag [18] proposes a network to fuse single frame with corresponding magnetic readings for localization. Our system advances it in several aspects. First, SweepLoc is more deployable as it does not require additional survey of magnetism to localize. Second, SweepLoc further exploits the correlation between consecutive frames to identify landmarks in videos accurately, which reduces the impact of noise from sensors, such as motion blur and strong light. Third, SweepLoc employs the scene-continuity constraints to jointly identify landmark sequences. By doing this, SweepLoc is able to further reduce the identification error and improve the localization accuracy.

3 SYSTEM OVERVIEW

In this section, we overview the work flow of SweepLoc in Figure 2, which consists of two main modules: the client and the server. We first overview the localization process from the viewpoint of a user. To localize, a user starts the localization application and sweeps the camera, then the application initiates video streaming and collects compass readings continuously and automatically. In the meantime, it sends collected video and compass

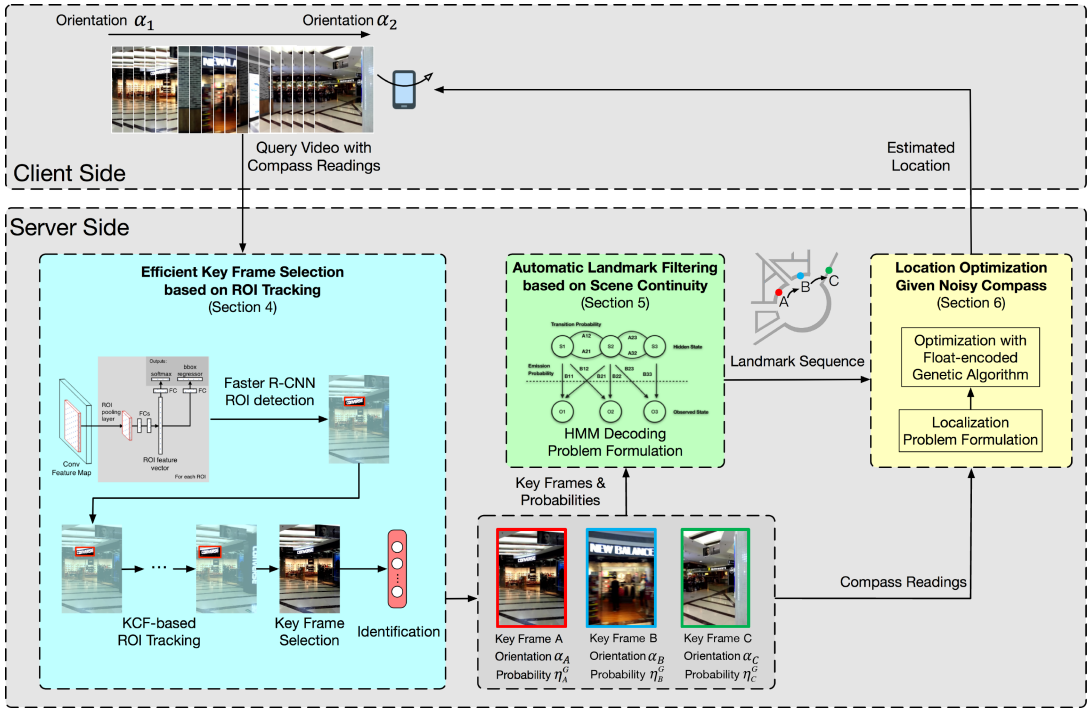


Fig. 2. System framework of SweepLoc with two main modules: client and server module. The client collects video and compass bearings while the server processes them and locates the client.

readings to a remote server. After video taking, the user obtains current location within a short time (usually 1 second).

Upon receiving the data, the server processes them in three main steps: potential landmark bounding and key frame selection, landmark identification and triangulation. These correspond to the following three submodules, respectively: *Efficient Key Frame Selection based on ROI Tracking*, *Automatic Landmark Filtering based on Scene Continuity*, and *Localization Optimization Given Noisy Compass*. Afterwards, server returns an estimated location to the user. We discuss these submodules as follows:

1) *Efficient Key Frame Selection based on ROI Tracking* (Section 4): To increase the applicability of SweepLoc, we propose to automatically select key frames with landmarks in the center with Faster R-CNN. In the offline phase, surveyors collect images for each landmark in the site and construct a landmark database. Then we fine-tune the Faster R-CNN to detect landmarks in frames. However, it is computationally expensive to conduct frame-wise ROI detection in videos, due to a large number of frames. To address this, SweepLoc initializes Kernelized Correlation Filters (KCF) tracker [11] with ROIs detected by Faster R-CNN and conducts bi-directional tracking of them in adjacent frames. Then SweepLoc automatically selects the key frame with an ROI in the center and preliminarily identifies the probability of each landmark. After an ROI moves out of the view, we initialize Faster R-CNN again to detect in the new frame and track ROIs accordingly. By tracking across the whole video, SweepLoc extracts a list of key frames with ROI confidence and corresponding compass readings.
2) *Automatic Landmark Filtering based on Scene Continuity* (Section 5): Due to the environmental noise, such as motion blur, strong light, Faster R-CNN may mis-identify landmarks in ROIs. To address this, we propose to filter mis-identified landmarks in the sequence jointly with Hidden Markov Model (HMM). SweepLoc first

Table 2. Major symbols in SweepLoc.

Notations	Definitions
K	Number of landmarks in a site
N	Number of ROI sequences in query video
τ_s	Sharpness index of ROI s
$\boldsymbol{p}_s$	$1 \times K$ probabilities of ROI s being each landmark
η_j^G	The probability of ROI sequence G corresponding to landmark j
λ	The Hidden Markov Model
A	$K \times K$ transition matrix of λ
B	$N \times K$ observation matrix of λ
$\boldsymbol{\pi}$	$1 \times K$ initial probability distribution of λ
$\mathbf{z}_k$	2-D coordinate of landmark k
$\mathbf{R}_s$	Rotation matrix corresponding to ROI s in an ROI sequence
$\mathbf{x}_{ij}$	Estimated 2-D location with respect to landmark i and j
$\hat{\mathbf{x}}$	Estimated 2-D location of the client
ω_{ij}	Weight of estimated 2-D location $\boldsymbol{x}_{ij}$

generates landmark transition matrix in the offline stage, which represents the probability of moving from the one landmark to the next one. To obtain the probability, we generate many test locations and simulate the video taking at these locations. Based on the simulation, we are able to estimate the transitional matrix. After key frame selection and identification in the previous phase, SweepLoc estimates the emission matrix and initial distribution vector then solve the HMM encoding problem by Viterbi Algorithm [5], which finds the landmark sequence with maximum probability and filters mis-identified landmarks.

3) *Localization Optimization Given Noisy Compass* (Section 6): Smartphone compass is highly noisy indoors due to nearby ferromagnetic objects, such as iron doors, lifts and escalators. The noise can incur distant location estimations without calibration. According to our previous experiments, we observe that location estimations based on two close landmarks usually give accurate localization results. Consequently, we develop a weighted triangulation algorithm with identified landmarks, distances between them and corresponding compass bearings. In the localization stage, the final location should be close to the center of location estimations with two landmarks. Based on the above intuitions, we formulate an optimization function with compass noise as bias to achieve higher accuracy. To obtain the global minimum of the optimization function, we solve this problem with float-encoded genetic algorithm [45].

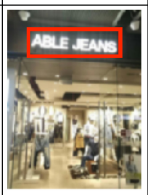
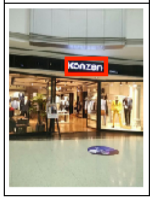
Table 2 exhibits the major symbols used in this paper.

4 EFFICIENT KEY FRAME SELECTION BASED ON ROI TRACKING

Previous image-based localization systems ask users to point to landmarks and take images. This is tedious for ordinary users as they may not be clear what is a landmark, and not applicable for robots or drones. To increase the applicability of SweepLoc, we propose to automatically select key frames with landmark in the center. Concretely, we employ the Faster R-CNN to detect ROIs with potential landmarks. However, it is computationally expensive to conduct frame-wise ROI detection with Faster R-CNN.

To address this, we propose a bi-directional ROI tracking paradigm to avoid the frame-wise detection. First, we utilize the fine-tuned Faster R-CNN to initialize the ROI (Section 4.1). Then we employ an efficient ROI selection algorithm to select an ROI sequence (Section 4.2). Finally, we select the key frames with ROI in the center and present the strategy to identify the landmark in each key frame using consecutive visual clues in Section 4.3.

Table 3. An example of data augmentation. The red rectangles indicate ROIs with landmarks.

Original Image	Image Blurring	Projective Transformation	Brightness Adjustment
			
			

4.1 Fine-tuned Faster R-CNN for Efficient ROI Detection

For automatic key frame detection, we fine-tune Faster R-CNN to select ROIs with potential landmarks in video frames. To this end, we first collect images for each landmark in the test site. Then we manually annotate a bounding box for each landmark in the image with the landmark ID. Next we construct a landmark database with these labeled boxes. We augment the database using various image transformations, e.g., resizing, projective transformation, brightness adjustment and image blurring, as shown in Table 3. The bounding boxes are transformed accordingly. By database augmentation, more training images are generated, which prevents over-fitting and facilitates accurate ROI detection.

Faster R-CNN consists of two modules: Region Proposal Networks (RPN) for computing candidate regions of an input image and Fast R-CNN for generating a tight bounding box around target object and classifying the object. In order to achieve higher mean Average Precision (mAP, which is the actual metric for object detection), we select the deeper VGG-16 model [27] which has 16 convolution layers as the convolutional layers of Faster R-CNN, instead of the 5-convolutional-layers ZF model [44] based on the experimental results in [24].

In the training stage, we modify the number of nodes in classification output layer and bounding box regression layer of Faster R-CNN and set them to $K + 1$ and $4(K + 1)$, respectively. One produces the probabilities of K landmarks in the indoor site and a catch-all background class, and the other encodes the bounding box regression offsets.

After specifying network model, we utilize the 4-step training algorithm proposed in [24] and train the Faster R-CNN with our augmented landmark database. First, we fine-tune RPN using Stochastic Gradient Descent (SGD). Second, we train the Fast R-CNN detection network using the proposals generated by RPN in the first step. Then we initialize the RPN with Fast R-CNN generated in the second step and train the RPN again with the weights of convolutional layers fixed. Finally we keep the convolution layers fixed and fine-tune the output layers of Fast R-CNN. Note that the convolutional layers are shared between RPN and Fast R-CNN, which reduces duplicate feature extraction and accelerates the training.

With the trained model, SweepLoc is able to segment ROIs in input frames and output the possibility of being a landmark. If the possibility is large, it indicates that the landmark in the ROI is very likely to be a landmark. To reduce the noise, we filter the regions with probability lower than a threshold δ. We discuss how to set δ in Section 7.

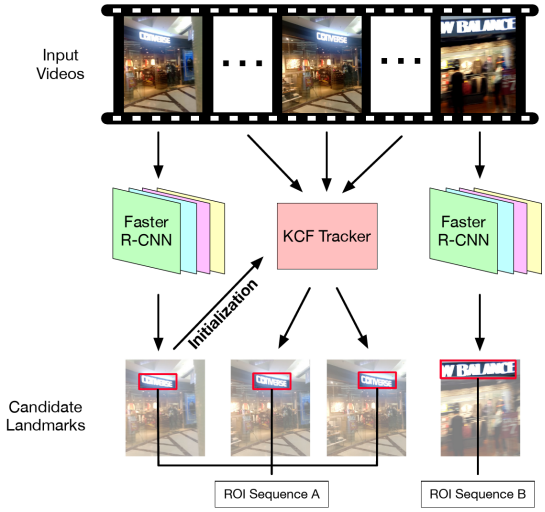


Fig. 3. ROI selection using Faster R-CNN and KCF tracker.

4.2 Efficient Key Frame Selection in Video Streams

In our experiment, it takes around 0.2s to process one query image with the trained ROI detection network. Thus it takes around 30 seconds to process a typical 5-second video with 150 frames. To facilitate efficient localization, we propose a *detect-and-track* strategy to monitor ROIs in videos, instead of detecting them frame-by-frame.

We first select ROIs in the starting frame (not necessarily the first frame). Then we employ Kernelized Correlation Filters (KCF) to track each detected ROI in the following frames efficiently (around 172 frames per second). Moreover, to reduce the possibility that ROI detection network misses an ROI in previous frames, the KCF tracker also moves in the reverse direction to track the ROI. When the tracked ROI moves out of the view, the tracking stops for each specific ROI. We then group these tracked ROIs (both directions) in each trace together. These tracked ROIs are termed ROI sequence. After generating an ROI sequence, SweepLoc initializes ROI in the next frame with Faster R-CNN as another starting frame and conducts tracking afterwards as shown in Figure 3. By doing so, we are able to extract the ROIs and select those frames with ROI in the center.

4.3 Landmark Identification in Key Frame

As Faster R-CNN detects ROIs with potential landmarks, it also outputs the possibility of each ROI belonging to a specific landmark. Since the parameters are shared between the RPN and Fast R-CNN, we can determine the probability of tracked ROIs without running Faster R-CNN. Intuitively, the landmark with the largest probability should be the landmark. However, the motion blur, which is the main source of noise of indoor videos, reduces the video quality and decreases the identification accuracy [15]. To reduce the impact of motion blur, we take all ROIs in the same ROI sequence into consideration as this retains all the available visual clues in videos instead of barely identifying landmark with the ROI inside the key frame.

To evaluate the impact of motion blur on the identification accuracy of Faster R-CNN, we simulate different levels of blur and evaluate the accuracy. To this end, we adjust the length of an image filter, denoted by $\mathcal{F}^l$,

$$\mathcal{F}^l = [1/l \quad 1/l \quad \cdots \quad 1/l], \tag{1}$$

where l is the length of the image filter. We present the output of original images with different blur filters in Figure 4(a)-Figure 4(f). Figure 4(g) shows that the probability corresponding to the correct landmark decreases

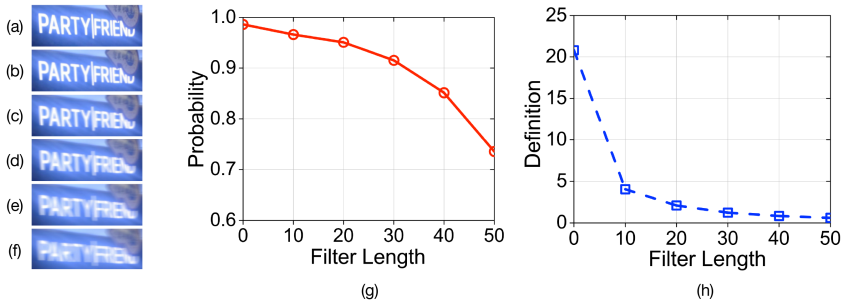


Fig. 4. (a) The original image. (b)-(f) The original image with filter length 10px, 20px, 30px, 40px, 50px, respectively. (g) The relationship of the probability corresponding to the correct landmark with motion blur. (h) The relationship of the definition measured by *Tenengrad* with motion blur.

significantly as the length of motion blur filter increases in the ROI. This is because larger filters is able to smooth out more detailed textures in images, leading to erroneous landmark estimations.

The confidence of identification in sharp images is usually higher than in blurred images. We select *Tenengrad* [40] as the metric of gradient magnitude and use it to measure the sharpness of images. Tenengrad uses the Sobel operator to compute the gradient, and decreases with higher level of motion blur, shown in Figure 4(h). The Sobel operator calculates the gradient both in the vertical direction (s_x) and horizontal direction (s_y). Formally, it is defined as follows:

$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad s_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \tag{2}$$

Given an ROI sequence, let R be the landmark in it. We obtain $\nabla S_x(i,j)$, the value of pixel (i,j) after applying convolutional operation to s_x and R. Similarly, $\nabla S_y(i,j)$ is the convolutional result of s_y. Thus, the definition of R measured by *Tenengrad* is defined as:

$$\tau = \frac{1}{XY} \sum_{i=1}^{X} \sum_{j=1}^{Y} [\nabla S(i,j)]^2, \tag{3}$$

where X and Y are the height and width of R respectively and $\nabla S(i,j)$ is the Sobel gradient magnitude value expressed by:

$$\nabla S(i,j) = \sqrt{\nabla S_x(i,j)^2 + \nabla S_y(i,j)^2}. \tag{4}$$

Combined with the sharpness of each ROI, we illustrate the process of landmark identification in an ROI sequence in Figure 5. Let M be the number of ROI in the given ROI sequence G. Note that $M = 3$ in Figure 5. We denote the probabilities after Fast R-CNN module and the definitions measured by *Tenengrad* of M ROIs as $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M\}$ and $\{\tau_1, \tau_2, \ldots, \tau_M\}$, respectively. Thus, after a Softmax classifier, the probability of G corresponds to landmark j can be expressed as

$$\eta_j^G = \frac{e^{\hat{\eta}_j}}{\sum_{k=1}^{K} e^{\hat{\eta}_k}} \quad \text{for} \quad j = 1, 2, \ldots, K, \tag{5}$$

where $\hat{\eta}_j$ is the j-th element of $\hat{\boldsymbol{\eta}}$ and $\hat{\boldsymbol{\eta}}$ is evaluated by assigning τ_i to $\boldsymbol{p}_i$ as weight, i.e.,

$$\hat{\boldsymbol{\eta}} = \sum_{i=1}^{M} \tau_i \boldsymbol{p}_i. \tag{6}$$

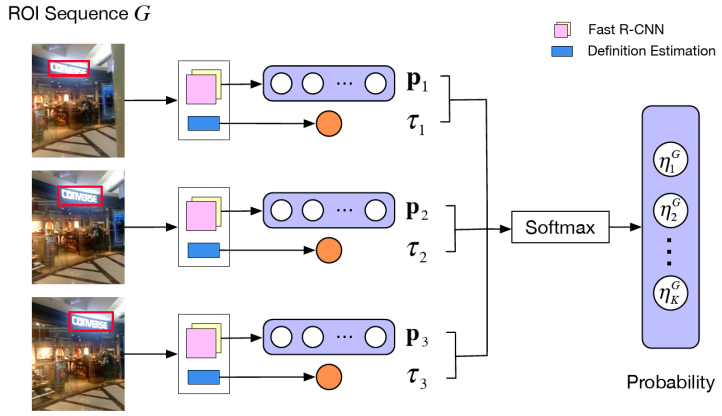


Fig. 5. Landmark identification by jointly considering the probabilities and the definitions of each ROI.

5 AUTOMATIC LANDMARK FILTERING BASED ON SCENE CONTINUITY

Although SweepLoc identifies the landmark in each key frame with consecutive visual clues, the landmark with the largest probability may still not be the target one due to the long distance, extreme viewing angle and the impact of light or motion blur. If we identify landmarks based on the landmarks with maximal probability, the localization error can be large. To address this, we propose to identify the *whole* landmark sequence (consisted of all detected candidate landmarks) detected in the video and filter those erroneous ones based on the scene continuity. Based on the above, we first present the problem in Section 5.1. Then we formulate this to the Hidden Markov Model (HMM) in Section 5.2 and find the landmark sequence with maximal probability. We present the complexity analysis of the algorithm in Section 5.3.

5.1 Problem Formulation

Given K landmarks and N key frames in a video, let $\mathbf{Q}$ be the collection of landmark and $\mathbf{V}$ be the key frames, i.e.,

$$\begin{aligned} \mathbf{Q} &= \{q_k | k \in \{1, \ldots, K\}\} \\ \mathbf{V} &= \{v_n | n \in \{1, \ldots, N\}\}. \end{aligned} \tag{7}$$

Given a key frame v_n, let x_n be the corresponding candidate landmark. The landmark sequence identification problem is to find the most likely sequence $\{x_n\}$ that complies with the indoor floor map:

$$\arg\max_{\{x_n\} \in \mathbf{Q}} P(x_1) \prod_{i=2}^{N} P(x_i | x_{i-1}) \prod_{j=1}^{N} P(v_j | x_j), \tag{8}$$

where $P(x_1)$ is the probability of the landmark in the first key frame being x_1, $P(x_i | x_{i-1})$ is the probability of transition from landmark x_{i-1} at time $i-1$ to landmark x_i at time i, and $P(v_j | x_j)$ gives the likelihood of the key frame v_j corresponding to landmark x_j.

An HMM is a stochastic finite state machine in which the internal states are hidden and only the outputs of the states are observable. In HMM modeling for landmark identification, $\mathbf{Q}$ and $\mathbf{V}$ are treated as the hidden states and observable states, respectively. Let $\pi = \{\pi_1, \ldots, \pi_K\}$ be the initial probability distribution. Let $A = \left[a_{ij}\right]_{i,j \in \mathbf{Q}}$ be the transition probability from hidden state q_i to state q_j. By definition, a_{ij} is given by

$$a_{ij} = P(q_i | q_j) \quad i, j \in \mathbf{Q}. \tag{9}$$

Similarly, let $B = [b_i(j)]_{i \in \mathbf{Q}, j \in \mathbf{V}}$ be the probability of observing v_j from state q_i and $b_i(j)$ is given by

$$b_i(j) = P\left(v_j | q_i\right) \quad i \in \mathbf{Q}, j \in \mathbf{V}. \tag{10}$$

We represent the HMM λ by a quintuple, i.e.,

$$\lambda = < \mathbf{Q}, \mathbf{V}, \pi, A, B > . \tag{11}$$

Thus, the landmark sequence identification problem can be reduced to the HMM decoding problem, a problem to find the most likely state sequence in the model that produced the observations.

5.2 Hidden Markov Model Parameter Initialization

In this section, we discuss how to estimate the preliminary parameters of HMM described in Equation (11) by jointly considering indoor configuration and match probabilities.

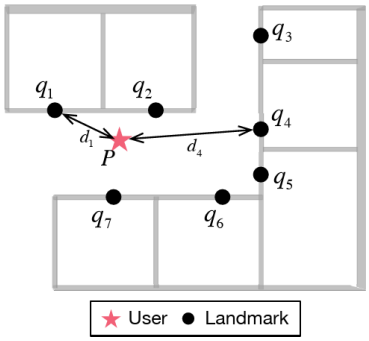


Fig. 6. HMM-based landmark sequence identification.

The transition matrix A specifies the distribution of landmarks in indoor scenes. To estimate A, we *simulate* the process of an user taking videos for localization and generate a large number of possible landmark sequences. Please note that our aim here is to simulate the transitional probability between landmarks. As a result, we accomplish this with the floor plan and does not consider the non-line-of-sight of landmarks due to temporary camera occlusions. Assume that the probability ξ_i of the landmark q_i appearing in the query video reduces as the distance between them increases by intuition. For example, Let P be the user's location in Figure 6, we have $\xi_1 > \xi_4$ since $d_1 < d_4$, where d_i is the distance between P and landmark q_i. Thus, we formulate ξ_i as

$$\xi_i = \begin{cases} \min\left\{1, \hat{d} \Big/ d_i\right\} & q_i \text{ in line-of-sight} \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

where $\hat{d}$ is the distance between user's position and a reference landmark, which is the first landmark select by user in our case. We also define a landmark is in *line-of-sight* as there are no opaque walls between user and the landmark, in other words, user can observe the landmark directly. As the floor plan shown in Figure 6, the landmark q_3 is not in line-of-sight for the user at P.

After generating sufficient landmark sequences, we estimate a_{ij} by the empirical probability

$$a_{ij} = \frac{\phi_{ij}}{\sum_{i=1}^{K} \phi_{ij}}, \tag{13}$$

where ϕ_{ij} is the frequency of situation that landmark q_i appears at time t and landmark q_j appears at time $t+1$ in the generated sequence.

We details how to generate the transition matrix A according to statistics of N_s landmark sequences below. First of all, We select N_s locations, which cover all public area in the indoor site. For each location P, we initialize the landmarks set $\mathbf{Q}^*$ which are in line-of-sight and select a subset $\mathbf{S}$ of $\mathbf{Q}^*$ with probability ξ_i. After that we sort S to generate a landmark sequence according to the relative rotation angle between q_{t_1} and q_i, where q_{t_1} is the first landmark in the generated sequence and $q_i \in \mathbf{S}$. As the situation shown in Figure 6, we can obtain the possible landmark sequence $\mathbf{S} = \{q_2, q_4, q_6, q_7, q_1\}$ while user selects landmark q_2 in the beginning. Note that the transition matrix A_{CW} while user rotating in clockwise direction is different from the transition matrix A_{CCW} in counter-clockwise direction, but A_{CCW} is the transpose of A_{CW}, intuitively, e.g.,

$$A_{CCW} = A_{CW}^T. \tag{14}$$

Hence, we sort $\mathbf{S}$ in ascending order to generate A_{CW} then obtain A_{CCW} by transpose operator. During online phase, we can determine the user's direction of rotation by the overall trend of the compass reading and choose the corresponding transition matrix.

To estimation emission matrix B, we consider η_i^j, which is the probability of key frame v_j corresponds to landmark q_i and calculated by Equation (5) in Section 4.3. Assume that every landmark and key frame is equally likely to be appeared, e.g.,

$$\begin{aligned} P(q_1) = P(q_2) = \cdots = P(q_K) = 1/K \\ P(v_1) = P(v_2) = \cdots = P(v_N) = 1/N, \end{aligned} \tag{15}$$

the probability $P\left(v_j|q_i\right)$ that observing v_j while observing landmark q_i is then calculated using Bayes rules, as follows:

$$P\left(v_j|q_i\right) = \frac{P(v_j)P(q_i|v_j)}{P(q_i)} = \frac{K \cdot \eta_i^j}{N}. \tag{16}$$

As a result, the emission matrix B is denoted by

$$B = \frac{K}{N}\left[\eta_i^j\right]_{N\times K} \quad i = 1, \ldots, K,\ j = 1, \ldots, N. \tag{17}$$

The initial probability distribution π provides information on the starting probability of each landmark. Let $\mathbf{Q}^*$ be the set of nearby landmarks which are observable for user. We assume that every landmark in $\mathbf{Q}^*$ is equally likely to be a starting landmark, and the landmarks that not in $\mathbf{Q}^*$ cannot be the beginning since they are unobservable for user. Thus, the initial probability distribution $\pi = [\pi_i]_{1\times K}$ is expressed as:

$$\pi_i = \begin{cases} \frac{1}{|\mathbf{Q}^*|} & , q_i \in \mathbf{Q}^* \\ 0 & , \text{otherwise} \end{cases}, \tag{18}$$

where $|\mathbf{Q}^*|$ is the number of landmark of $\mathbf{Q}^*$. Note that $\mathbf{Q}^*$ is generally determined by a rough location estimation of user (such as Wi-Fi, INS or the recent location). If the location is inexistent, we define that $\mathbf{Q}^* = \mathbf{Q}$.

When the parameter $\lambda =< \mathbf{Q}, \mathbf{V}, \pi, A, B >$ is given, the most likely landmark sequence $\left\{x_1^*, x_2^*, \ldots, x_N^*\right\}$ can be identified by the Viterbi Algorithm.

We illustrate the landmark sequence identification problem by giving a toy example at the test place illustrated in Figure 6. Suppose the client sweeps the camera in clockwise direction at point P and starts taking video from q_2. Thus the corresponding true landmark sequence is $\{q_2, q_4, q_5, q_6, q_7, q_1\}$, while the estimated one is $\{q_2, q_4, q_6, q_3, q_1\}$ as Faster R-CNN fails to detect the ROI of landmark q_5 and misidentifies landmark q_7 to landmark q_3. After identifying the most likely landmark sequence by HMM, SweepLoc correct the erroneous estimation q_3 to q_7 with indoor configuration and locates the client with adjusted landmark sequence $\{q_2, q_4, q_6, q_7, q_1\}$.

5.3 Complexity Analysis

The landmark sequence identification consists of two stages: offline stage and online stage. In the offline stage, we estimate the transition matrix A of HMM. Generating N_s landmark sequences takes $O(N_s \cdot K)$ time. Note that the parameter N_s is constant. The complexity of evaluating transition matrix by Equation (13) is $O(K^2)$. Therefore, the overall offline complexity of landmark sequence identification is then

$$O\left(N_s \cdot K + K^2\right). \tag{19}$$

In the online stage, after selecting key frames with framework described in Section 4, we first estimate the emission matrix B and the initial state distribution π. After that, we solve the landmark identification problem with Viterbi Algorithm. As there are K landmarks in database and N key frames in query video, the complexity of estimate B is $O(N \cdot K)$ while the complexity of π is $O(K)$ and the complexity of dynamic-programming-based Viterbi Algorithm is $O\left(N \cdot K^2\right)$. Thus, the overall complexity of landmark sequence identification problem in online stage is:

$$O\left(N \cdot K^2\right), \tag{20}$$

where K and $N \cdot K$ are asymptotically smaller than $N \cdot K^2$ and hence can be ignored.

6 LOCATION OPTIMIZATION GIVEN NOISY COMPASS

As ferromagnetic materials are pervasive indoors (e.g., doors, elevators and escalators), the compass bearings are usually noisy, leading to erroneous location estimations. Thus, we propose a novel algorithm to estimate the current location. Section 6.1 gives a preliminary example with two landmarks, followed by the overall optimization problem in Section 6.2. We present the complexity analysis of SweepLoc in Section 6.3.

6.1 Angle-based Localization with Two Landmarks

Given two key frames detected in video, we can obtain the corresponding landmarks by the algorithm described in Section 5. Note that the coordinates of two landmarks ($\mathbf{z}_1$, $\mathbf{z}_2$) and the compass readings while the user facing them (θ_1 and θ_2) (Figure 7(a)), the straight line that connects landmark q_s and the user location is denoted as $l_s : A_s x + B_s y + C_s = 0$. Note that the compass reading is the relative angle between user's orientation and the truth north ($\vec{\mathbf{n}}$). We calculate A_s, B_s and C_s as follows:

$$\begin{cases} [-B_s, A_s]^T = R_s \cdot \vec{\mathbf{n}} \\ C_s = -[A_s, B_s]\mathbf{z}_s \end{cases}, \tag{21}$$

where R_s is the rotation matrix which transfers $\vec{\mathbf{n}}$ to direction vector of l_s:

$$R_s = \begin{bmatrix} \cos\theta_s & \sin\theta_s \\ -\sin\theta_s & \cos\theta_s \end{bmatrix}. \tag{22}$$

The client location determined by Landmark 1 and 2, $\mathbf{x}_{12}$, is defined as follows:

$$\mathbf{x}_{12} = \left[\frac{B_1C_2 - C_1B_2}{A_1B_2 - B_1A_2}, \frac{C_1A_2 - A_1C_2}{A_1B_2 - B_1A_2}\right]^T. \tag{23}$$

6.2 Localization with Multiple Landmarks

In our experiments, we can identify several landmarks from the video, which determine a few intersections of line segments defined by identified landmarks and the camera. If the compass is free of error, these intersections should converge to a point. However, due to the compass noise, the estimated locations (circles in Figure 7(b)) are usually dispersed.

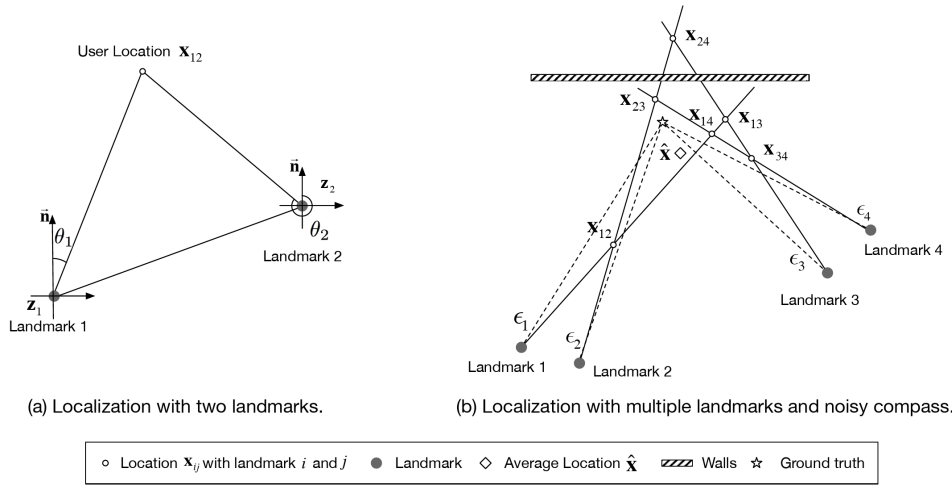


Fig. 7. Illustration of absolute angle based localization.

Suppose we are able to identify N landmarks in a query video, then we need to determine the true location based on $N(N-1)/2$ estimated locations (intersections). We model the compass noise as follows:

$$\theta_n = \hat{\theta}_n + \epsilon_n, \tag{24}$$

where θ_n denotes the compass bearing corresponding the n-th ($1 \le n \le N$) landmark, $\hat{\theta}_n$ denotes the ground truth orientation of user and ϵ_n denotes the compass error. Thus, the localization estimation based on landmark i and j ($0 < i, j < N$) is defined as:

$$\mathbf{x}_{ij} = \mathbf{Loc}\left(\mathbf{z}_i, \theta_i, \mathbf{z}_j, \theta_j\right), \tag{25}$$

where $\mathbf{Loc}(\cdot,\cdot,\cdot,\cdot)$ is the algorithm defined in Equation (23) and $\mathbf{z}_i, \mathbf{z}_j$ are coordinates of the landmark i and j, respectively.

In our experiment, if the distance between landmark i and j is shorter, the confidence of their location estimation $\mathbf{x}_{ij}$ is higher. However, if landmark i and j are not in the line-of-sight of each other (a client cannot see landmark i and j simultaneously), the identification results can be wrong. Based on our observations, we employ a weighted localization strategy to calculates the location $\hat{\mathbf{x}}$ of the client:

$$\hat{\mathbf{x}} = \frac{1}{Y}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\omega_{i,j}\mathbf{x}_{ij}, \tag{26}$$

where $Y = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\omega_{ij}$ is the normalization factor and the weight corresponding to the intersection $\mathbf{x}_{ij}$ is:

$$\omega_{ij} = \begin{cases} 1 \Big/ \left|\mathbf{z}_i - \mathbf{z}_j\right|^2 & \text{in the line-of-sight} \\ 0 & \text{otherwise} \end{cases}. \tag{27}$$

Based on the above, we formulate our localization problem as follows. The objective is to minimize the distance between the weighted location $\hat{\mathbf{x}}$ and $\mathbf{x}_{ij}$ by estimating a potential compass drift $\mathbf{E} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_N]^T$.

$$J(\mathbf{E}) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left|\mathbf{x}_{ij} - \hat{\mathbf{x}}\right|^2 + \alpha\sum_{i=1}^{N}\epsilon_i^2, \tag{28}$$

where $\sum_{i=1}^{N}\epsilon_i^2$ is the regularization term that keeps ϵ_i small to avoid overfitting, and α is regularization parameter used to adjust the weight of regularization term and localization error. We minimize $J(\mathbf{E})$ with float-encoded

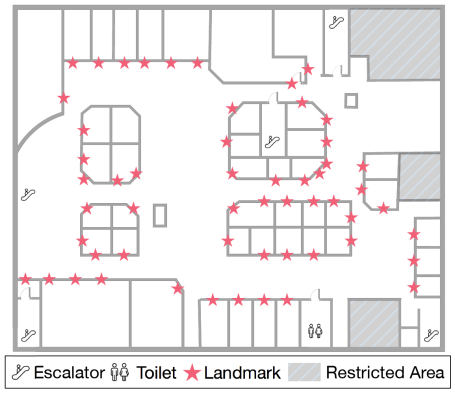


Fig. 8. The floor plan of the food plaza.

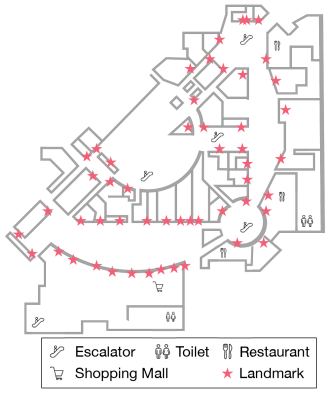


Fig. 9. The floor plan of the shopping mall.

genetic algorithm. With the consideration of compass error $\mathbf{E}$, we are able to reduce the impact of compass error, thus increasing the localization accuracy.

6.3 Complexity Analysis

Given two landmarks and the compass reading while the user is facing them, we can estimate the user location by Equation (23). As there are N landmarks detected in the query video, the number of intersections of each two landmarks is $N(N-1)/2$. Therefore, we can evaluate the objective function $J(\mathbf{E})$ in Equation (28) within $O(N^2)$.

Assume that the population size and the number of generations of Genetic Algorithm are G_p and G_e, respectively, the complexity of minimizing the object function and solving the compass-robust user location is given by

$$O\left(N^2 G_p G_e\right). \tag{29}$$

Note that the parameter G_p and G_e are constants in the localization algorithm, thus the overall computational complexity is quadratic of N. In our experiment, the number of N ranges from 3 to 7. As a result, the computational cost in the localization is low.

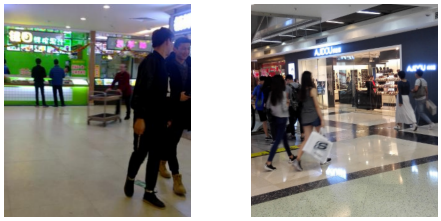
7 ILLUSTRATIVE EXPERIMENTAL RESULTS

To evaluate the performance of SweepLoc, we implement a client-server-based localization system to conduct experiments in two test sites. The client side of SweepLoc is an Android-based application responsible for capturing videos and recording compass bearings. While taking the video, the client sends the recorded video and compass bearings to the server by streaming. The server simultaneously receives the data and processes them. We present our experimental settings and comparison schemes in Section 7.1. Then we evaluate the landmark identification in Section 7.2 and the localization error with different parameter settings in Section 7.3, followed by discussing the system overhead in Section 7.4.

7.1 Experimental Setting and Comparison Schemes

We collect test videos with Xiaomi 4 and Huawei Mate 7 and deploy our server on a PowerMax Workstation running on Ubuntu 14.04 with Intel Xeon E5 Processor, 128GB RAM and NVIDIA Tesla K40c GPU. The system trains the ROI selection and landmark identification network with Caffe [12]. We use the genetic algorithm library, GALib [30], in our system.

Figure 8 and Figure 9 exhibit the floor plans of the food plaza and the shopping mall, respectively. The testing area of food plaza is around 3,500 m^2 while the shopping mall is 16,000 m^2. Figure 10 presents the corresponding



(a) The food plaza. (b) The shopping mall.

Fig. 10. Two frames from the test videos in our experiment. We do not avoid humans during the test.

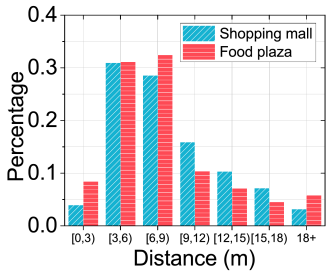


Fig. 11. The distribution of distances between landmarks and test locations.

test sites in these test sites. In our experiment, we regard store logos as landmarks and construct the landmark database by a comprehensive site survey. In addition to that, we can also construct the database by crawling images from guide websites. There are 55 landmarks in food plaza and 51 landmarks in shopping mall, marked by red stars. The locations of landmarks are manually tagged on the floor plan. Since the average distance between one landmark and the nearest neighbor is around $3.7m$ in the food plaza and the distance in the shopping mall is $8.4m$, the distribution of landmarks in the shopping mall is more dispersed than the food plaza. It takes us 2-3 hours to collect images for all landmarks in each test site (each with around 15 images). To sum up, we have 825(=55×15) images in the food plaza and 765(=51×15) images in the shopping mall. Then we manually label the landmark regions in these training images by rectangles. After data augmentation, we fine-tune the Faster R-CNN for each site with these manually labeled regions on a GPU server for 8-10 hours. Note that the fine-tuning is conducted offline. Thus it does not incur time overhead in the localization stage. The number of label regions is around 19,656 and 18,034 training images in the food plaza and the shopping mall, while numbers of manually taken images are 825 and 765 respectively. Therefore, data augmentation significantly reduces the survey cost. In addition, we simulate the video taking at different places and generate 16,000 possible landmark sequences to estimate the transition matrix of HMM.

We conduct experiments at 113 test locations in the food plaza and 41 test locations in the shopping mall. These test locations are also carefully tagged. We do not specifically avoid pedestrians during the test. To evaluate the localization error, we take a short video at each test location. Meanwhile, the corresponding compass readings on client during the video are recorded. Figure 11 illustrates the distribution of distances between test location and the corresponding landmarks in different sites.

Given |C| location queries (each with a query video) and the i-th query video c_i. We evaluate localization performance using the following metrics.

- *Localization error*: Let the ground truth location of the user be $\mathbf{x}_i$ in query video c_i and the estimated location of the user be $\hat{\mathbf{x}}_i$. The mean localization error is given by $e = \sum_{i=1}^{|\mathbf{C}|} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\| \Big/ |\mathbf{C}|$ where $\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|$ is the Euclidean distance (in meters) between $\hat{\mathbf{x}}_i$ and $\mathbf{x}_i$.

Table 4. Baseline parameters used in SweepLoc.

Name	Value	
	Food plaza	Shopping Mall
# of landmarks	55	51
# of sequence N_s in HMM training	16000	
Faster R-CNN threshold δ	0.5	
Video resolution	800 x 600	

- *Success rate*: The unsuccessful located queries are usually caused by insufficient number of landmark detected in the query videos. Let $\mathbf{C}^*$ be the collection of successful located queries. The success rate defines the percentage of the successful located queries among all the queries, e.g., $r = |\mathbf{C}^*| \big/ |\mathbf{C}|$.
- *Average identification accuracy*: Let s_i denote the true landmark sequence of query c_i and the detected sequence be $\hat{s}_i$. Let $LCS(s_i, \hat{s}_i)$ be the Longest Common Subsequence between s_i and $\hat{s}_i$. The length of a sequence s is $|s|$. Thus, we evaluate the average landmark sequence identification accuracy by $\sigma = \sum_{i=1}^{|\mathbf{C}|} \frac{|LCS(s_i, \hat{s}_i)|}{|s_i|} \Big/ |\mathbf{C}|$.
- *Precision and Recall of key frame selection*: Suppose that tp_i is the number of key frame selected by SweepLoc in the landmark database. As mentioned above, there are N_i landmarks detected in video c_i, the key frame selection precision is $\sum_{i=1}^{|\mathbf{C}|} \frac{tp_i}{N_i} \Big/ |\mathbf{C}|$. On the other hand, the recall of key frame selection (given T_i landmarks in the video c_i) is defined as: $\sum_{i=1}^{|\mathbf{C}|} \frac{tp_i}{T_i} \Big/ |\mathbf{C}|$.

We compare SweepLoc with the state-of-the-art Sextant[6] and MoVIPS[34]. Apart from those, we also evaluate the performance of SweepLoc without landmark sequence identification (denoted by SweepLoc w/o sequence identification) and SweepLoc without reducing the impact of noisy compass (denoted by SweepLoc w/o compass optimization). We briefly discuss these schemes as follows:

- *Sextant*[6]: Sextant employs the benchmark selection algorithm to select benchmark images for each landmark and construct the image database. In the localization phase, Sextant compares photos taken by user from three nearby landmarks with image database. Then it asks user to confirm the correct landmarks in the top 3 matching results. After user confirmation, Sextant utilizes relative angle based triangulation to locate the user.
- *MoVIPS*[34]: MoVIPS constructs the database with many geo-referenced images of the environment in training phase. In the testing phase, it finds the most similar training image to the input image and estimates the distance between the user and the location of matched training image according to the ratio of distance between corresponding points in the image. Finally, MoVIPS estimates the location of the user based on the location of the matched image and the distance.
- *SweepLoc w/o sequence identification* (SweepLoc w/o Seq-Id): This comparison is same to SweepLoc except it obtains the landmark sequence with the maximal probability of each key frame, which the landmark sequence $\{x_1, x_2, \ldots, x_N\}$ is expressed as $x_i = \arg\max_j \eta_j^i, i = 1, 2, \ldots, N$.
- *SweepLoc w/o compass optimization* (SweepLoc w/o Opt): This comparison is same to SweepLoc except it locates the user without considering the noisy compass. The user's location is the mean of each intersection, which can be estimated as $L = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathbf{x}_{i,j} \Big/ C_n$, where N is the number of landmark in the query video and $C_n = N(N-1)/2$ is the number of landmark combinations.

Unless otherwise stated, we show the baseline parameters of SweepLoc in Table 4. For Sextant and MoVIPS, we randomly select 9 training images for each landmark before data augmentation as suggested in [6]. We implement the benchmark selection algorithm to select three benchmark images for each landmark and construct the image

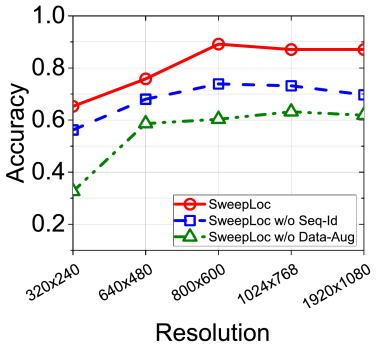


Fig. 12. Average identification accuracy vs. video resolution (food plaza).

database in the training phase of Sextant while the image database of MoVIPS consists of all training images. As Sextant and MoVIPS accept images rather than video clips, we randomly select three (or one) pivot frames where ROI is in the center of the video frame, respectively.

In order to evaluate the impact of video length on performance, we obtain several fixed-length queries by randomly selecting continuous part of the videos and the corresponding compass readings. We repeat the experiments for different length of query videos. To study the impact of video resolution, we upsample and downsample each frame in query video. To evaluate the impact of data augmentation, we retrain the detection network with the training database without augmentation (SweepLoc w/o Data-Aug).

We also conduct a simulation that evaluates the localization accuracy of SweepLoc with different numbers of landmarks in the database. To this end, we randomly select 20%, 40%, 60%, 80% and 100% of landmarks and construct the landmark database with them. Then we repeat the experiments with these databases. To evaluate the robustness of angle based localization schemes in SweepLoc with noisy compass sensor, we assume that the compass error with Gaussian noise with zero mean and conduct another simulation by adding noise with different standard deviation values (in degree) to the measured values.

Furthermore, we run the localization client in two different modes: no other application running and conducting localization queries. Then we consecutively record battery level to evaluate the power consumption of client.

7.2 Landmark Identification

Figure 12 shows the landmark sequence identification accuracy with different video resolution. It shows that the average identification accuracy increases with video resolution. The reason is that high resolution images contain more information. As a result, more features can be learned by deep neural network, which leads to higher accuracy. However, high resolution takes more network bandwidth consumption. To achieve trade-off between identification accuracy and bandwidth, we choose 800×600 as our baseline parameter in our experiments to achieve sufficient identification accuracy (around 89.1%). It also shows that data augmentation increases the accuracy with more diverse training images. It is because the augmentation prevents overfitting during training process and allows neural network become more robust to various resolution and view points. The identification accuracy of SweepLoc is higher than that without HMM-based landmark sequence identification. The reason is that it utilizes indoor configuration and takes the whole landmark sequence into consideration, which reduces noise from long distance, extreme angle or motion blur.

7.3 Localization Accuracy

Figure 13 and Figure 14 illustrate the overall localization performance in the food plaza and the shopping mall, respectively. In our experiments, the localization error of SweepLoc is smaller than other schemes in both

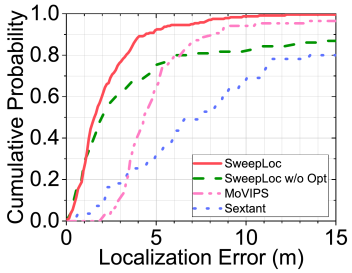


Fig. 13. CDF of localization error in the food plaza.

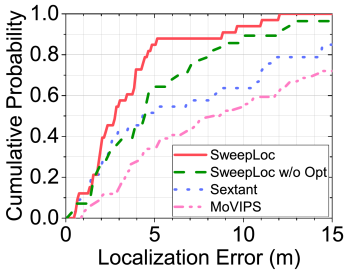


Fig. 14. CDF of localization error in the shopping mall.

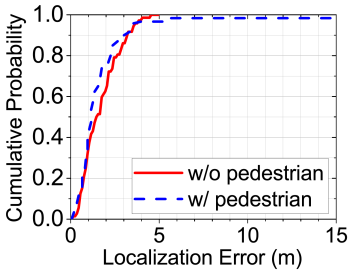


Fig. 15. Localization error v.s. pedestrians (food plaza).

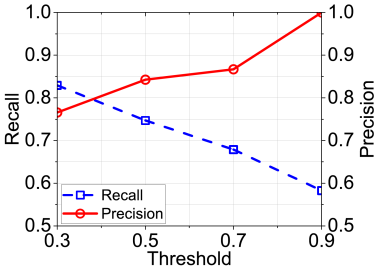


Fig. 16. Recall and precision vs. Faster R-CNN threshold δ (food plaza).

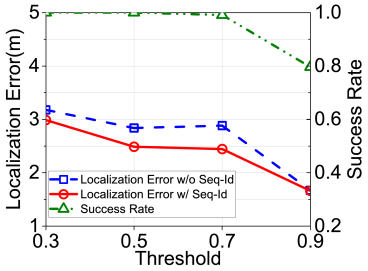


Fig. 17. Localization error and success rate vs. Faster R-CNN threshold δ (food plaza).

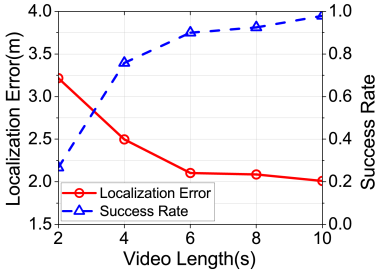


Fig. 18. Localization error and time consumption vs. video length (food plaza).

experimental sites. The reasons are two-fold. First, the video sequence-based landmark identification strategy takes consecutive visual clues into consideration, thus is more robust in complex indoor environment and achieves higher accuracy. Second, the proposed weighted localization algorithm locates users with multiple landmarks, which provides more robust location clues than image-based methods.

However, the localization error in the shopping mall is larger than the food plaza because the shopping mall is larger and the distribution of stores is dispersed. These result in less information of visual details and thus lower matching accuracy and larger localization error. In our test, Sextant does not achieve satisfactory localization accuracy, which is mainly due to the low matching accuracy. This is because our test environment is highly dynamic (with walking pedestrians) and contains much noise, such as drastic illumination changes at different test locations, extreme viewing angle, which can result in degraded matching accuracy with traditional SURF. In addition, Sextant does not consider reducing drift of noisy motion sensor, which can also result in degraded localization accuracy.

Figure 15 compares the cumulative error of SweepLoc with and without pedestrians in view. We can infer that the localization error remains stable even if pedestrians are present. This is mainly due to following reasons. First, we choose store logos as landmarks, which is high above the ground and not easily obstructed by walking pedestrians. Second, we recognize landmarks with landmark regions instead of whole image. Therefore SweepLoc is robust to humans in sight. Third, we are usually able to detect multi landmarks in a single video (refer to the discussion of Figure 18). With one of landmarks obstructed, our algorithm is able to estimate the location with remaining visible landmarks. Consequently, SweepLoc is able to achieve stable accuracy.

Figure 16 depicts the recall and precision of ROI selection module with different thresholds in the food plaza. We can infer from this figure that precision keeps rising while recall decreasing. The reason is that false alarms are filtered with large threshold. However, if the threshold is too large, we may mistakenly reject genuine

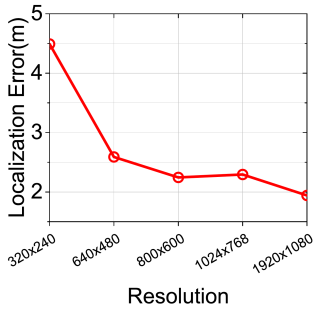


Fig. 19. Localization error vs. video resolution (food plaza).

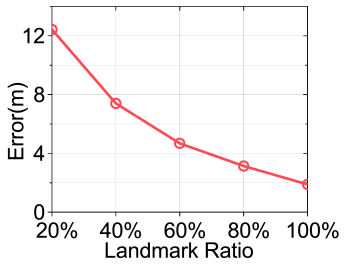


Fig. 20. Localization error vs. landmark ratio (food plaza).

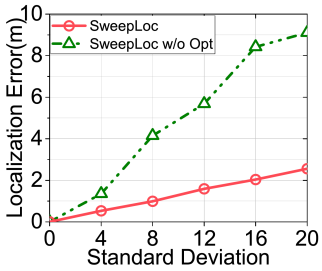


Fig. 21. Localization error vs. compass drift (food plaza).

landmarks, resulting in degraded localization due to the lack of landmarks. In contrast to large values, we may extract obsessive number of landmarks with small threshold. In this case, many of potential ROIs may be false alarms, which incurs high cost of tracking and potentially erroneous landmark identification. To achieve trade-off between recall and precision, we set the threshold to 0.5 in our experiment.

Figure 17 illustrates the mean localization error with the threshold in ROI selection module. It shows that SweepLoc with landmark sequence identification based on indoor configuration outperforms that with the maximal probability only. The reason is that HMM-based landmark identification increases accuracy and reduce the impact of erroneous localization results. We can also infer from this figure that the localization error decreases with the threshold. This is because the precision increases and the false positive ROIs without real landmark are filtered with the increase of the threshold. However, if the threshold increase, the recall of ROI identification module decrease, which causes the number of detected ROI with potential landmark is insufficient for localization and the success rate decrease in the meantime. As a result, we choose $\delta = 0.5$ as the trade-off threshold in the ROI identification module.

Figure 18 depicts the localization error and success rate with the length of query videos. It shows that the longer videos lead to lower localization error and higher success rate. This is because a longer video usually covers larger field of view with more landmarks, which provides more constraints of optimization in joint localization algorithm. This figure shows that the decrease of localization error slows down with videos longer than 6 seconds. Thus, we conclude that the number of landmarks detected in a 6-second video is able to provide strong location clue with high success rate. According to our experiment, the average number of landmarks in a 6-second video usually ranges from 4 to 6. Therefore, we come to the conclusion that SweepLoc is able to achieve sufficient accuracy with around 5 landmarks.

Figure 19 presents the localization error with different video resolution. It shows that the localization accuracy decreases with higher video resolution. This is because frames with higher resolution usually contain more detailed information about the landmark, thus Faster R-CNN is able to recognize landmark regions and identify them with higher accuracy. This further leads to lower localization error. However, the decrease of localization error slows down with resolution larger than 800x600. This is because we have enough detailed information of landmarks. With larger resolution, the time consumption of video uploading and Faster R-CNN detection can increase. Therefore, we set the resolution to 800x600 in our experiment to achieve trade-off between transmission overhead, Faster R-CNN detection and localization accuracy.

Figure 20 depicts the localization error with different number of landmarks. It shows that the localization error decreases with more landmarks in our test site. This is because more landmarks can provide more location clues. Based on these landmarks, we are able to estimate the landmark sequence and filter erroneous ones with

higher confidence. Furthermore, more landmarks also reduce the compass noise and thus increase the overall localization accuracy.

Figure 21 presents the simulation results with random compass noise. We can see that the localization error of SweepLoc w/ optimization is smaller and increases more slowly compared with that without optimization, which demonstrates the robustness of SweepLoc. The reason is that we specifically design the cost function to select a point with minimal sum of distance to all other estimated locations with two landmarks. In addition, we add the compass noise as bias to the formulation so that we are able to estimate the compass error in the meantime. Finally we use the genetic algorithm progressively to estimate the compass error and the client position. Thus SweepLoc is more robust to compass noise and achieve higher accuracy.

7.4 System Overhead

It takes around 265ms to select the potential landmark corresponding to a key frame with the Faster R-CNN on the NVIDIA Tesla K40c GPU. Apart from that, it takes around 12ms to identify the maximal probability landmark sequence with the indoor configurations and 118ms to locate the client with a genetic algorithm. As the localization server concurrently detects ROIs and tracks them in video frames, SweepLoc is able to locate the client efficiently with streaming. As a result, the total time consumption during a location query is less than $t + 1$ seconds, where t is the length of query video (usually 6-8s).

Then we present the energy consumption of SweepLoc. Experimental results show that the average current is 289.01mA (no foreground application running) and 737.109mA (taking and uploading videos through Wi-Fi). As one localization takes around 6-8 seconds, the total power consumption is around 21 Joules. Therefore, we conclude that SweepLoc does not consume much energy compared with the power capacity of current smartphones (around 20k Joules).

8 DISCUSSION AND FUTURE WORK

Indoor localization has large social and commercial values in public sites, such as shopping malls, museums, to name a few. These sites usually have rich visual features, such as store logos, wall paintings, banners and posters which serve as landmarks for localization. Therefore, our proposed SweepLoc is most effective and popular in these sites. Armed with automatic potential landmark detection, effective landmark filtering and robust localization, SweepLoc can facilitate a wide range of applications, e.g., indoor localization, augmented gaming. To achieve trade-off between localization accuracy, ease of use and processing overhead, system develop could use videos of medium resolution (800x600) and length (around 5s). As for landmark selection, developers should select landmarks high above the ground to achieve stable localization accuracy in crowded scenes as they are distinguishable and not easily occluded by pedestrians. To achieve sufficient accuracy, developer should select many landmarks as they are able to provide stronger location constraints and thus higher localization accuracy.

Our key frame selection strategy can be applied to a number of other research areas as well, such as video abstraction (a small part of frames from the whole video that presents the main content of the video). For example, many museums have taken videos of their collections to public visitors. Our key frame selection algorithm can be applied to extract key frames from videos, where cameras aim at objects. By doing so, content holders can easily tag interesting frames and viewers can grasp the big picture of the site.

In practical deployment, indoor landmarks can change from time to time due to renovation. In this case, we need to identify changed landmarks and collect new images for them manually, which is time-consuming and tedious. In the future, we plan to develop a scheme to identify changed landmarks automatically by implicit crowdsourcing [10, 46]. By doing this, we are able to identify changed landmarks and update the image database automatically and reduce the maintenance cost of the system.

9 CONCLUSION

Vision-based indoor localization has attracted much attention lately, mainly because it does not require the deployment of external devices. Current image-based indoor localization methods suffer from several limitations, such as sophisticated and slow user operations, error-prone landmark identification and noisy motion sensors. These limitations reduce the automation and degrade the accuracy of such systems. To address above, we propose SweepLoc, an automatic video-based indoor localization system. SweepLoc achieves automation by employing a customized ROI detection network to detect ROIs. It speeds up the key frame selection by temporal tracking in consecutive video frames without frame-wise expensive ROI detection. To reduce the impact of motion blur and camera noise, SweepLoc finds the target landmark by jointly considering an ROI sequence. It further filters the erroneous landmarks based on the scene continuity. Moreover, it employs a localization scheme that locates the client with noisy sensor readings. We conduct extensive experiments in two test sites, a shopping mall and a food plaza. Compared with competing schemes, SweepLoc is able to improve the localization by 30% with a little time overhead (less than 1s).

ACKNOWLEDGMENTS

This work was supported, in part, by National Natural Science Foundation of China under Grant 61472455, Guangzhou Science Technology and Innovation Commission (GZSTI16EG14/201704030079).

REFERENCES

[1] P. Bahl and V. N. Padmanabhan. 2000. RADAR: an in-building RF-based user location and tracking system. In *Proc. of IEEE INFOCOM*, Vol. 2. IEEE, 775–784 vol.2.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. SURF: speeded up robust features. In *Proc. of Springer ECCV*. Springer, Berlin, Heidelberg, 404–417.

[3] Jiang Dong, Yu Xiao, Marius Noreikis, Zhonghong Ou, and Antti Ylä-Jääski. 2015. iMoon: using smartphones for image-based indoor navigation. In *Proc. of ACM SenSys*. New York, NY, USA, 85–97.

[4] Jiang Dong, Yu Xiao, Zhonghong Ou, Yong Cui, and Antti Ylä-Jääski. 2016. Indoor Tracking Using Crowdsourced Maps. In *Proc. of IEEE IPSN*. Article 5, 6 pages.

[5] G David Forney. 1973. The viterbi algorithm. *Proc. IEEE* 61, 3 (1973), 268–278.

[6] R. Gao, Y. Tian, F. Ye, G. Luo, K. Bian, Y. Wang, T. Wang, and X. Li. 2016. Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment. *IEEE Trans. Mob. Comput.* 15, 2 (Feb 2016), 460–474.

[7] R. Gao, B. Zhou, F. Ye, and Y. Wang. 2017. Knitter: Fast, resilient single-user indoor floor plan construction. In *Proc. IEEE INFOCOM*. IEEE, 1–9.

[8] Cole Gleason, Dragan Ahmetovic, Saiph Savage, Carlos Toxtli, Carl Posthuma, Chieko Asakawa, Kris M. Kitani, and Jeffrey P. Bigham. 2018. Crowdsourcing the Installation and Maintenance of Indoor Localization Infrastructure to Support Blind Navigation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1, Article 9 (March 2018), 25 pages.

[9] Fei Gu, Jianwei Niu, and Lingjie Duan. 2017. WAIPO: A Fusion-Based Collaborative Indoor Localization System on Smartphones. *IEEE/ACM Trans. Netw.* 25, 4 (Aug 2017), 2267 – 2280.

[10] B. Guo, Q. Han, H. Chen, L. Shangguan, Z. Zhou, and Z. Yu. 2017. The Emergence of Visual Crowdsensing: Challenges and Opportunities. *Commun. Surveys Tuts.* 19, 4 (Fourthquarter 2017), 2526–2543.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. 2015. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 3 (March 2015), 583–596.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: convolutional architecture for fast feature embedding. In *Proc. of ACM MM*. 675–678.

[13] Hernisa Kacorri, Eshed Ohn-Bar, Kris M. Kitani, and Chieko Asakawa. 2018. Environmental Factors in Indoor Navigation Based on Real-World Trajectories of Blind Users. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 56, 12 pages.

[14] Alex Kendall, Matthew Grimes, and Roberto Cipolla. 2015. PoseNet: a convolutional network for real-time 6-DOF camera relocalization. In *Proc. of IEEE ICCV*. 2938–2946.

[15] T. H. Kim, S. Nah, and K. M. Lee. 2017. Dynamic Video Deblurring using a Locally Adaptive Linear Blur Model. *IEEE Trans. Pattern Anal. Mach. Intell.* PP, 99 (Oct 2017), 1–1.

[16] Patrick Lazik, Niranjini Rajagopal, Oliver Shih, Bruno Sinopoli, and Anthony Rowe. 2015. ALPS: A Bluetooth and Ultrasound Platform for Mapping and Localization. In *Proc. ACM Sensys.* ACM, 73–84.
[17] Kai Liu, Hao Zhang, Joseph Kee-Yin Ng, Yusheng Xia, Liang Feng, Victor CS Lee, and Sang H Son. 2018. Toward Low-Overhead Fingerprint-Based Indoor Localization via Transfer Learning: Design, Implementation, and Evaluation. *IEEE Trans. Ind. Informat.* 14, 3 (2018), 898–908.
[18] Z. Liu, L. Zhang, Q. Liu, Y. Yin, L. Cheng, and R. Zimmermann. 2017. Fusion of magnetic and visual sensors for indoor localization: infrastructure-free and more effective. *IEEE Trans. Multimedia* 19, 4 (April 2017), 874–888.
[19] Chengwen Luo, Hande Hong, Mun Choon Chan, Jianqiang Li, Xinglin Zhang, and Zhong Ming. 2018. MPiLoc: Self-Calibrating Multi-Floor Indoor Localization Exploiting Participatory Sensing. *IEEE Trans. Mob. Comput.* 17, 1 (2018), 141–154.
[20] Raul Mur-Artal and Juan D Tardós. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* 33, 5 (2017), 1255–1262.
[21] S. Papaioannou, A. Markham, and N. Trigoni. 2017. Tracking People in Highly Dynamic Industrial Environments. *IEEE Trans. Mob. Comput.* 16, 8 (Aug 2017), 2351–2365.
[22] Valter Pasku, Alessio De Angelis, Darmindra D. Arumugam, Marco Dionigi, Paolo Carbone, Antonio Moschitta, and David S. Ricketts. 2017. Magnetic Field-Based Positioning Systems. *IEEE Commun. Surveys Tuts.* 19, 3 (Mar 2017), 2003 – 2017.
[23] Claudio Piciarelli. 2016. Visual indoor localization in known environments. *IEEE Signal Process. Lett.* 23, 10 (2016), 1330–1334.
[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In *Proc. of NIPS*. MIT Press, 91–99.
[25] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: an efficient alternative to SIFT or SURF. In *Proc. of IEEE ICCV*. 2564–2571.
[26] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. 2016. *Exploiting Spatial and Co-visibility Relations for Image-Based Localization.* Springer International Publishing, Cham, 165–187.
[27] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations* (2015).
[28] Masato Sugasaki and Masamichi Shimosaka. 2017. Robust Indoor Localization Across Smartphone Models with Ellipsoid Features from Multiple RSSIs. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 103 (Sept. 2017), 16 pages.
[29] Xiaohua Tian, Mei Wang, Wenxin Li, Binyao Jiang, Dong Xu, Xinbing Wang, and Jun Xu. 2018. Improve Accuracy of Fingerprinting Localization with Temporal Correlation of the RSS. *IEEE Trans. Mob. Comput.* 17, 1 (2018), 113–126.
[30] Matthew Wall. 1996. GAlib: A C++ library of genetic algorithm components. *Mechanical Engineering Department, Massachusetts Institute of Technology* 87 (1996), 54.
[31] S. Wang, S. Fidler, and R. Urtasun. 2015. Lost Shopping! Monocular Localization in Large Indoor Spaces. In *Proc. IEEE ICCV*. IEEE, 2695–2703.
[32] Yu-Lin Wei, Chang-Jung Huang, Hsin-Mu Tsai, and Kate Ching-Ju Lin. 2017. CELLI: Indoor Positioning Using Polarized Sweeping Light Beams. In *Proc. of ACM MobiSys.* ACM, 136–147.
[33] Hongkai Wen, Sen Wang, Ronnie Clark, Savvas Papaioannou, and Niki Trigoni. 2016. Poster: Efficient Visual Positioning with Adaptive Parameter Learning. In *Proc. of IEEE IPSN*. IEEE Press, Piscataway, NJ, USA, Article 34, 2 pages.
[34] M. Werner, M. Kessel, and C. Marouane. 2011. Indoor positioning using smartphone camera. In *Proc. of IEEE IPIN*. 1–6.
[35] Chenshu Wu, Zheng Yang, and Chaowei Xiao. 2018. Automatic Radio Map Adaptation for Indoor Localization Using Smartphones. *IEEE Trans. Mob. Comput.* 17, 3 (2018), 517–528.
[36] K. Wu, Jiang Xiao, Youwen Yi, Min Gao, and L. M. Ni. 2012. FILA: Fine-grained indoor localization. In *Proc. IEEE INFOCOM*. IEEE, 2210–2218.
[37] Han Xu, Zheng Yang, Zimu Zhou, and Chunyi Peng Ke Yi. 2017. TUM: Towards Ubiquitous Multi-Device Localization for Cross-Device Interaction. In *Proc. IEEE INFOCOM*.
[38] Han Xu, Zheng Yang, Zimu Zhou, Longfei Shangguan, Ke Yi, and Yunhao Liu. 2015. Enhancing wifi-based localization with visual clues. In *Proc. of ACM UbiComp*. ACM, New York, New York, USA, 963–974.
[39] Han Xu, Zheng Yang, Zimu Zhou, Longfei Shangguan, Ke Yi, and Yunhao Liu. 2016. Indoor Localization via Multi-modal Sensing on Smartphones. In *Proc. ACM UbiComp*. ACM, 208–219.
[40] Yi Yao, Besma Abidi, Narjes Doggaz, and Mongi Abidi. 2006. Evaluation of sharpness measures and search algorithms for the auto focusing of high-magnification images. In *Visual Information Processing XV*, Vol. 6246. International Society for Optics and Photonics, 62460G.
[41] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios. 2017. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. *Commun. Surveys Tuts.* 19, 2 (Secondquarter 2017), 1327–1346.
[42] Xuehan Ye, Yongcai Wang, Yuhe Guo, Wei Hu, and Deying Li. 2018. Accurate and Efficient Indoor Location by Dynamic Warping in Sequence-Type Radio-Map. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1 (2018), 50.
[43] Moustafa Youssef and Ashok Agrawala. 2005. The horus WLAN location determination system. In *Proc. of ACM MobiSys.* 205–218.

[44] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *Proc. of Springer ECCV*. Springer, Cham, 818–833.
[45] Hongwei Zhang, Barry Lennox, Peter R Goulding, and Andrew YT Leung. 2000. A float-encoded genetic algorithm technique for integrated optimization of piezoelectric actuator and sensor placement and feedback gains. *Smart Materials and Structures* 9, 4 (2000), 552.
[46] X. Zhang, A. K. S. Wong, C. T. Lea, and R. S. K. Cheng. 2018. Unambiguous Association of Crowd-Sourced Radio Maps to Floor Plans for Indoor Localization. *IEEE Trans. Mob. Comput.* 17, 2 (Feb 2018), 488–502.
[47] Yongtuo Zhang, Wen Hu, Weitao Xu, Hongkai Wen, and Chun Tung Chou. 2016. NaviGlass: indoor localisation using smart glasses. In *Proc. of ACM EWSN*. 205–216.
[48] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao. 2017. Travi-Navi: Self-Deployable Indoor Navigation System. *IEEE/ACM Trans. Netw.* 25, 5 (Oct 2017), 2655–2669.
[49] Shilin Zhu and Xinyu Zhang. 2017. Enabling High-Precision Visible Light Localization in Today's Buildings. In *Proc. of ACM MobiSys*. ACM, 96–108.
[50] Jinbo Zuo, Shuo Liu, Hao Xia, and Yanyou Qiao. 2018. Multi-Phase Fingerprint Map Based on Interpolation for Indoor Localization Using iBeacons. *IEEE Sensors J.* (2018), 3351 – 3359.

Received February 2018; revised May 2018; accepted September 2018