Resource-efficient and Automated Image-based Indoor Localization

QUN NIU and MINGKUAN LI, Sun Yat-sen University, Guangzhou, China
SUINING HE, The Hong Kong University of Science and Technology, Hong Kong, China
CHENGYING GAO, Sun Yat-sen University, Guangzhou, China
S.-H. GARY CHAN, The Hong Kong University of Science and Technology, Hong Kong, China
XIAONAN LUO, Guilin University of Electrical Technology, Guilin, China

Image-based indoor localization has aroused much interest recently because it requires no infrastructure support. Previous approaches on image-based localization, due to their computation and storage requirements, often process queries at servers. This does not scale well, incurs round-trip delay, and requires constant network connectivity. Many also require users to manually confirm the shortlisted matched landmarks, which is inconvenient, slow, and prone to selection error.

To overcome these limitations, we propose a **h**ighly **a**utomated (in terms of image confirmation after taking images) **i**mage-based **l**ocalization algorithm (HAIL), distributed in mobile devices. HAIL achieves resource efficiency (in terms of storage and processing) by keeping only distinguishing visual features for each landmark, and employing the efficient k-d tree to search for features. It further utilizes motion sensors and map constraints to enhance the localization accuracy without user operation. We have implemented HAIL on Android platforms and conducted extensive experiments in a food plaza and a premium shopping mall. Experimental results show that it achieves much higher localization accuracy (reducing the localization error by more than 20%) and computation efficiency (by more than 40% in time) as compared with the state-of-the-art approaches.

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

Additional Key Words and Phrases: Image-based localization, feature selection, automated image selection, joint constraints, smartphone-based localization

ACM Reference format:
Qun Niu, Mingkuan Li, Suining He, Chengying Gao, S.-H. Gary Chan, and Xiaonan Luo. 2019. Resource-efficient and Automated Image-based Indoor Localization. *ACM Trans. Sen. Netw.* 15, 2, Article 19 (February 2019), 31 pages.
https://doi.org/10.1145/3284555

This work is supported, in part, by National Natural Science Foundation of China under Grant 61472455, Guangzhou Science Technology and Innovation Commission GZSTI16EG14/201704030079 and Guangdong Provincial Department of Science and Technology GDST16EG04/2016A050503024.

Authors' addresses: Q. Niu, M. Li, and C. Gao (corresponding author), School of Data and Computer Science, Sun Yat-sen University, Guangzhou, 510006, China; emails: {niuqun, limkuan}@mail2.sysu.edu.cn, mcsgcy@mail.sysu.edu.cn; S. He, S.-H. Gary Chan, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China; emails: {sheaa, gchan}@cse.ust.hk; X. Luo, School of Computer Science and Information Technology, Guilin University of Electronic Technology, Guilin, 541004, China, email: luoxn@guet.edu.cn.

1550-4859/2019/02-ART19 $15.00
https://doi.org/10.1145/3284555

1 INTRODUCTION

Image-based indoor localization has aroused much interest in commercial sites, mainly because it does not require infrastructure support and has large commercial and social values. Image-based localization techniques are mainly based on *landmarks*, i.e., physical prominent objects (such as store logos or wall paintings) of known locations. These landmarks have a relatively long lifetime in the system (in weeks or months), and they provide more robust and stable location clues as compared with fluctuating Radio Frequency (RF) signals. In this article, though, for concreteness, we consider a "user" or "pedestrian" carrying a "smartphone" for image-based localization; our technique can be broadly applied to devices equipped with multiple cameras, such as robots, drones, head-mounted gears, smartglasses, and the like [23, 38, 41, 47].

Image-based indoor localization is often conducted in two phases: *offline phase* (site survey) and *online phase* (location query). In the offline phase, surveyors collect images of landmarks, the so-called "training images" (Figure 1). The localization systems then extract visual features (e.g., Speeded-Up Robust Features [2], or SURF for short) of these images to build the feature database. (Note that besides collecting images by professional surveyors, crowdsourcing [14, 55] or web crawling from popular guide websites, such as guides on restaurants, are also alternative solutions to the construction of the image database.)

In the online phase, a user takes images of the nearby environment (e.g., the logo of a store or a restaurant). The system then compares features of images taken by the user with those in the feature database and estimates the location based on top matches. To further enhance localization accuracy, some works [48, 52] employ distances or relative angles between the user and matched images, as measured by device sensors.

While many works have appeared on image-based localization, most of them are based on client-server architecture where the servers are responsible for processing location queries [14]. Such approaches are not scalable to a large number of users, incur round-trip delay, and raise privacy concerns. They also require network connectivity and extra infrastructure, which may not be (always) available in practice.

To overcome these limitations, we study an image-based localization system at mobile devices. Our problem can be formally defined as follows. Given a floor plan and landmarks indoors, find the location of the client with several input images and sensors automatically without network access. The problem is challenging, due to the following:

- *Limited storage on devices*: The storage on mobile devices is often limited. Existing image-based approaches, due to their steep requirement on storage, can be hardly deployed in mobile devices. For example, the size of the feature database in GoGo Plaza (a food plaza covering around 6,000m^2 in Guangzhou, China) is around 275 MB.[1] For a larger mall, the feature size can be orders of magnitude larger, which is too costly to be stored locally at a smartphone.
- *Limited processing on devices*: Traditional server-based approaches often suffer from high-processing requirements, which lacks distributability on resource-constrained smartphones. Using our example of the food plaza above, many features need to be processed (in this case 470,400 features). Processing and comparison of them at smartphones create computation and power concerns.

[1] There are 84 landmarks in the food plaza. We take seven training images for each of the landmarks. Then we extract 800 SURF features for each training image and store them in a feature file, whose size is 480KB on average. The total size of the feature database is, hence, $84 \times 7 \times 480 = 275$ MB.

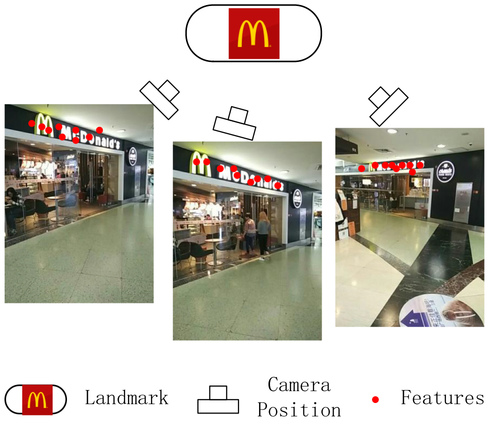


Fig. 1. Illustration of a landmark (McDonald's) for image-based localization. The features in the images form the feature database or are used for localization.

- *The need for manual image confirmation*: To achieve acceptable accuracy, many previous approaches require users to manually select and confirm the shortlisted matched images. This is inconvenient (cumbersome), unreliable (susceptible to selection error), and slow (leading to non-realtime operation). Furthermore, such a selection mechanism is not applicable to robotic devices (such as autonomous robots and drones) where a higher intelligence is not always available.

To address the above, we propose HAIL, a **h**ighly **a**utomated **i**mage-based **l**ocalization system deployable on resource-limited devices. Different from the state-of-the-arts [11, 52], HAIL is automated in that it determines the target landmark without manual confirmation of shortlisted images. More specifically, HAIL makes the following contributions:

- *Offline feature selection:* In contrast to the traditional work that stores all image features, HAIL employs a novel strategy to select only relevant and *distinguishing* visual features in the training images. This greatly reduces features and, hence, storage requirement (by 99% compared with the comprehensive image database). Furthermore, the time consumption for landmark comparison and localization is significantly lower (by 40%) without sacrificing the matching accuracy.
- *Automated image selection:* HAIL uses an efficient kurtosis method to select the target landmarks automatically and efficiently. Kurtosis measures the "peakedness" of a distribution. Given a query image, if one landmark has more matched features with it than others, the "peakedness" of matched feature numbers is high. To increase the applicability, HAIL employs the efficient logistic regression to select the target landmark with the kurtosis. Consequently, HAIL can be easily extended to different scenarios.
- *Highly accurate localization:* As HAIL only stores distinguishing features for image matching, it greatly reduces comparison redundancy, resulting in lower matching error. To further enhance localization accuracy, it uses floor plans as constraints, and locates the target by *jointly* considering the compass, gyroscope, and mutual distances between landmarks.

We show the system framework of HAIL in Figure 2. The workflow consists of two phases: an offline phase and an online phase.

In the offline phase, surveyors collect images of the site and store them in an image database. Then HAIL extracts features from these images and selects distinctive ones for each landmark in

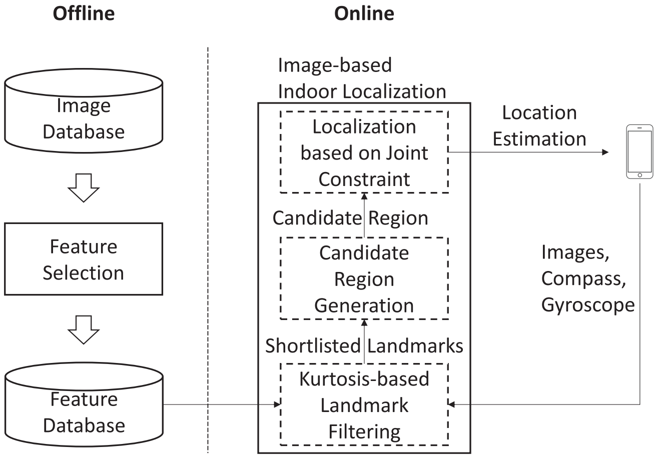

Offline
Online
Image Database
Feature Selection
Feature Database
Image-based Indoor Localization
Localization based on Joint Constraint
Candidate Region
Candidate Region Generation
Shortlisted Landmarks
Kurtosis-based Landmark Filtering
Location Estimation
Images, Compass, Gyroscope

Fig. 2. The system workflow in HAIL.

order to differentiate it from all others (discussed in Section 3). Afterward, HAIL employs a k-d tree structure to store the feature database for efficient landmark searching. Before localization, a user can download the database to the mobile device for efficient query. In the online phase, a user first collects several images of the environment. HAIL then automatically finds a landmark for each image via the efficient kurtosis method. Each identified landmark indicates a candidate region the user may be in. With the compass readings of the smartphone, HAIL finally finds the constrained region for the user and estimates the location (discussed in Section 4).

Due to its great values in commercial sites, we conduct extensive experiments in the GoGo Plaza (a food plaza) and the Grandview Mall (a premium shopping mall) in Guangzhou, China. We validate that HAIL can be easily deployed on resource-constrained mobile devices (smartphones). Experimental results show that HAIL significantly reduces the localization error (by more than 20%) as compared with the state-of-the-art schemes. A user study also proves that many users think that HAIL is automated, easy to use, and accurate. We present the full questionnaire in the appendix.

HAIL is most effective to be applied to sites with rich features, such as shopping malls, food plaza, galleries, museums, exhibitions, and the like. For sites with less features, posters and banners may be used as landmarks. HAIL can also be integrated with Optical Character Recognition (OCR) to further enhance its accuracy. The edges of tiles on the ground, walls, and ceilings may also be used to localize a target [3] or calibrate the compass [39]. We have conducted another experiment in the office section of the Hong Kong University of Science and Technology where visual landmarks are sparsely distributed. Experimental results demonstrate that the introduction of posters, banners, and door plates can improve the localization accuracy.

The feature selection strategy in HAIL is different from Bag-of-Words (BoW) models in that we do not conduct any quantization or clustering, which reduces the chance of omitting the best correspondence between features. This is because our objective is to achieve sufficient accuracy with limited storage capacity. As in previous fingerprint-based localization, they consider updating the database to achieve higher accuracy. Since the landmarks are stable over a long time (often in weeks or months), HAIL does not require frequent updates. The feature database can be retrained with crowdsourced data as presented in, for example, Refs. [13] and [40]. Recent research [9, 12] considers constructing the floor plan effectively. These are orthogonal to ours in that we consider efficient localization on mobile devices. However, we can integrate them into the current work to make it more deployable.

The remainder of this article is organized as follows. In Section 2, we review the related work. We present our feature selection strategy in Section 3 and propose the localization algorithm and the formulation in Section 4. We discuss experimental results in Section 5 and conclude in Section 6.

2 RELATED WORK

As a Global Positioning System (GPS) signal cannot penetrate indoors, researchers have explored many other RF signals for indoor localization, including Wi-Fi [16, 28, 42, 50], Bluetooth [51], Frequency Modulation (FM) [5], and Radio Frequency IDentification (RFID) [46]. Despite promising results, many RF-based indoor localization systems still suffer from the following limitations. First, they require extra infrastructures (e.g., beacons, RFID readers, synchronized transmitters, and receivers). Second, the measured RF signal often suffers from a multipath effect [15, 20, 31, 49]. Third, the alteration of Access Points (APs) incurs constant radio map updates. These limitations increase the maintenance cost and degrade the localization accuracy [17]. To address these, we study image-based indoor localization, which leverages environmental landmarks for localization. Compared with RF signals, these landmarks are more stable and, hence, provide more informative location clues. Furthermore, HAIL can work with the above RF-based localization systems and achieve higher positioning accuracy.

However, images usually require more storage than numerical Wi-Fi fingerprints without compression. To reduce the storage burden on smartphones, Sextant [11] selects only one representative image out of many for each landmark, which may lose useful information from other training images. Different from Sextant, HAIL selects distinguishing features from *all* training images rather than one. Therefore, it is able to reduce storage without sacrificing much accuracy in landmark matching.

Instead of storing all images in the database, researchers propose feature-based localization methods to reduce storage. These methods can be broadly divided into two categories. The first category of methods reduce storage by space projection and dimension reduction. For example, Lu et al. [24] propose to project high-dimensional SIFT descriptors (128 real-valued vectors) to low-dimensional Hamming space. Jegou et al. [18, 19] further refine visual words by adding binary signatures to descriptors. Although efficient, they could result in loss of feature distinctiveness and, consequently, degraded matching accuracy.

The second category of methods reduce storage by selecting a *subset* of original features [1, 44, 54]. Turcot et al. [44] select features by adapting the term-frequency inverse-document-frequency (tf-idf) weighting with only geometrically verified images. NaviGlass [54] reduces the storage by eliminating features if they are physically correlated, i.e., those of two nearby landmarks. This can remove useful features from adjacent landmarks because they are observable in different positions, leading to degraded matching accuracy. Torri et al. [43] propose a scalable representation of repetitive features of outdoor buildings to facilitate place recognition. Li et al. [22] find repetitive features and prioritize the matching with them. Sattler et al. [35, 36], on the other hand, determine the probability of a feature to be matched and the *cost* of matching. By sorting these features according to the cost, they find top ones that yield a sufficient number of 2D-to-3D matches. Different from above, we select features based on *landmarks* and measure their uniqueness with both *in-class* similarity and *out-class* differences. Therefore, we are able to retain distinguishing features and remove frequent ones detectable in different landmarks. In addition, we select SURF from 2D images directly without constructing 3D models, which is more computationally efficient. Knopp et al. [21] enhance landmark matching by identifying and removing repetitive objects and features using geo-tags. Our feature selection differs from this in two aspects. First, we do not employ bag-of-word clustering to retain the distinctiveness of visual features. Second, we simultaneously consider the in-class and out-class landmarks to select both distinctive and robust features. Bae et al. [1] select features based on the historical queries, where a feature is more likely to be selected if it is related to many queries. Although effective, it does not work well when the system first deploys. HAIL advances in that it does not require historical data and provides efficient localization once it is deployed.

Recent image-based localization techniques have employed 3D models [7, 9, 34, 37, 53]. Cavallari et al. [4] propose to learn the 3D point registration with Regression Forest. Although effective when executed in servers, it requires many (1,024) initial pose guesses and RANdom SAmple Consensus (RANSAC) optimization, which is not applicable for resource-constrained mobile devices. Lu et al. [25] use the optical flow method to build an accurate 3D model using input videos. ClickLoc [53] fuses Wi-Fi with a 3D model to achieve accurate localization. Unlike ClickLoc, HAIL does not need a Wi-Fi fingerprint to obtain coarse location, thus reducing the survey cost. Apart from that, it constrains the *user location*, rather than candidate landmarks with the smartphone orientation to reduce the computational cost. iMoon [8] builds the Structure from Motion (SfM) model in the offline stage and locates the client accurately by image registration. In case the input image is distant from 3D models, Deng et al. [7] form a local 3D model with multiple input images so that they can enhance localization with more information. However, 3D-based techniques are not suitable for smartphones due to the steep requirement of battery. In contrast, HAIL does not need to build complicated 3D models. Thus, it is more suitable for mobile platforms with limited computational power.

Many 3D model-based localization systems discuss generating scene candidates before view registration to speed up the localization. In Ref. [27], researchers employ multi-task learning to retrieve the candidate location corresponding to the query image. Lu et al. [26] infer the scene candidate with the detected object in the constructed 3D model. Our HAIL is different from the above in that we do not need to construct complicated 3D models. Apart from that, HAIL reduces the computation by leveraging off-the-shelf sensors, e.g., GPS, compass, to infer the candidate scene, thus achieving higher efficiency.

To reduce computational complexity, recent works locate users using 2D images [10, 11, 32]. Piciarelli et al. [33] propose to locate a user by comparing the input image with the localized features of the environment. MoVIPS [48] further considers the distance from the input image to the most matched image and its location. However, it requires many training images to infer the fine-grained location of the user and is prone to the estimation error of mutual distance. HAIL advances in that it requires far fewer training images and does not rely on any distance estimations. With an efficient image structure and a novel fusion scheme, it is more lightweight and robust to measurement noise. Sextant [11] also locates users with three matched 2D images. However, it asks the user to select shortlisted landmarks before localization. If some of the matching results are not correct, Sextant estimates the mismatched landmarks and then locates the user by triangulation. HAIL achieves automation using the kurtosis method, which eliminates the need for inconvenient and error-prone manual confirmation. Furthermore, it utilizes a unified localization strategy to estimate user locations with joint constraints, which make use of the angular similarity between the user and landmarks and, hence, improve the localization accuracy. NaviGlass [54] locates the client *primarily* using the motion sensors and recovered trajectory. To reduce the drift, it *opportunistically* calibrates the current location based on the image matching. Since it calibrates the current location based on the matching results, it needs dense sampled training images to achieve sufficient accuracy. Different from it, HAIL does not rely on noisy motion sensors and needs a small number of training images to locate clients.

We summarize the strengths and weaknesses of typical state-of-the-arts in Table 1, where we divide related work into two main categories: mobile platform-based methods and pure vision-based ones. Those in the first category take the advantage of abundant sensors on mobile

Table 1. Qualitative Comparisons of the State-of-the-Arts

Category	Scheme	Input	Memory	Automation	Localization Error	Network Dependency
Mobile platform-based	iMoon [8]	Image, Wi-Fi, motion sensors	High	High	<2m (90th percentile)	Always
	Sextant [11]	Image, motion sensors	Low	Low	<8m (80th percentile)	No
	MoVIPS [48]	Image	High	High	<1m	Always
	ClickLoc [53]	Image, Wi-Fi, motion sensors	High	Low	<1m (80th percentile)	Always
	NaviGlass [54]	Image, motion sensors	Low	High	<6m (80th percentile)	No
Pure vision-based	Lu et al. [26]	Video	High	High	83.5% retrieval accuracy	Always
	Lu et al. [27]	Image	High	High	97.8% retrieval accuracy	Always
	Opdenbosch et al. [32]	Image	Low	High	<10m (80th percentile)	Always

platforms or consider the applicability in a mobile. In contrast, those in the second category achieve sufficient accuracy with only visual information. They usually require fast and reliable network access and a powerful backend server to realize localization. In summary, HAIL takes the advantage of multiple sensors on mobile devices and achieves accurate localization on resource-constrained smartphones. Thus, it is significantly different from pure vision-based methods.

3 OFFLINE FEATURE SELECTION

Traditional 2D image-based localization systems [11, 48] store training images of landmarks. This could incur frequent feature detection and extraction during localization, which increases computational overhead on servers. Instead of storing training images, we first extract *SURF* of landmarks and store them in the smartphones. This is because SURF is robust with different view points. Every landmark is, hence, a combination of features rather than a collection of training images. The process of finding similar images becomes finding similar landmarks, which is termed *landmark searching*.

As mentioned, keeping all the SURF of landmarks in smartphones leads to the high cost of storage and landmark searching. To achieve storage and computational efficiencies, we, hence, propose a novel feature selection strategy to select distinguishing features for each landmark automatically. To put it another way, we filter frequent features, which appear in many landmarks because they do not help differentiating the landmarks. In this way, we significantly reduce the database size without sacrificing matching accuracy (or even improving it).

We elaborate the objective of feature selections as follows. Suppose there are M landmarks and the number of training images for each landmark is N. Let i_{mn} be the n-th $(1 \leq n \leq N)$ training image of landmark m $(1 \leq m \leq M)$. In the experiment, HAIL extracts H features from each training image. Finally, it selects Q features for each landmark. We summarize the major symbols used in the article in Table 2. Then, we present the three stages of feature selection, image comparison (Section 3.1), feature voting (Section 3.2), and distinguishing feature selection (Section 3.3). Finally, we summarize the overall workflow of the feature selection and evaluate the computational complexity (Section 3.4).

Table 2. Major Symbols in HAIL

Notation	Definition
M	Number of landmarks.
N	Number of training images for each landmark.
$\boldsymbol{\gamma}_m$	2D location of landmark m.
$\boldsymbol{\Gamma}$	$M \times 2$ matrix of landmark locations.
i_{mn}	n-th training image of landmark m.
H	Number of extracted features for each training image.
Q	Number of features selected for each landmark.
S	Number of landmarks chosen by a user.
μ_s	Kurtosis value corresponding to queries s.
α_s	Rotation angle between s-th and $(s+1)$-th image.
θ_s	Compass reading when taking the s-th image.
$\Phi(\mathbf{x})$	Sum of Euclidean distances between location estimation $\mathbf{x}$ and estimated landmarks.
Ψ	Sum of mutual Euclidean distances between estimated landmarks.

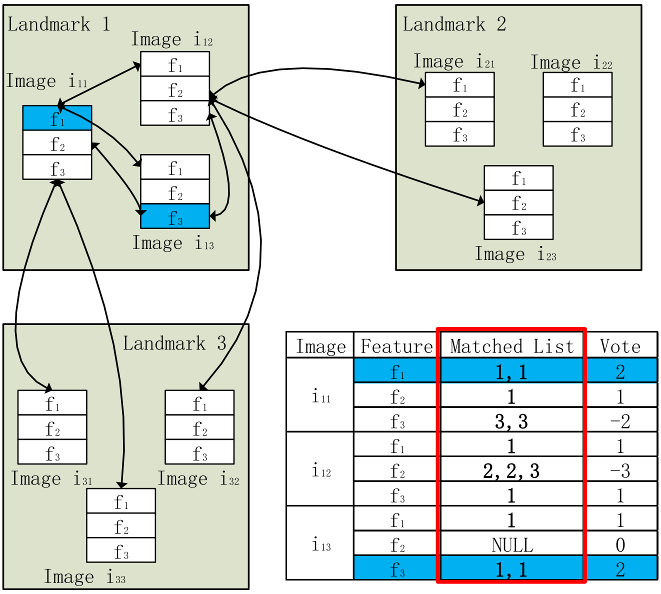


Image	Feature	Matched List	Vote
i_{11}	f_1	1, 1	2
	f_2	1	1
	f_3	3, 3	−2
i_{12}	f_1	1	1
	f_2	2, 2, 3	−3
	f_3	1	1
i_{13}	f_1	1	1
	f_2	NULL	0
	f_3	1, 1	2

Fig. 3. An illustration of the proposed feature selection. For concreteness, we only present the matched lists of features and their votes in Landmark 1 in the table.

3.1 Image Comparison with Local Features

In the first image comparison stage, HAIL compares each training image with all others in the image database using local features, such as SURF. To illustrate, we present the process of image comparison among images from three landmarks (Landmarks 1, 2, and 3, respectively) in Figure 3. Each of these landmarks has three training images (e.g., Landmark 1 has three images, Image i_{11}, i_{12}, and i_{13}) in the image database. We extract a given number of most stable features from each image, e.g., 3 (numbered f_1, f_2, and f_3). By doing this, we are able to reduce the impact of noisy

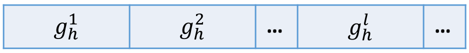


Fig. 4. The matched list of the h-th feature in image i_{mn}.

features. Then, HAIL first compares each image, e.g., Image i_{11} of Landmark 1, to other images of this landmark and all images from other landmarks.

Due to the long distance, cluttered environment, and noise of images, the algorithm can detect unstable SURF. To reduce the impact of these inaccurate features, we employ several techniques to detect strong and stable features from input images. First, we set the threshold of SURF to 1,000. With higher thresholds, we are able to eliminate weak SURF. In addition, we also adopt the ratio test in Ref. [30] to reduce false matches. To determine if two features are matched, we compare them by calculating their Euclidean distance. If their distance is smaller than a given threshold, they are matched in this case. With the above strategies, we can detect more strong features. We use double-headed arrows to indicate two matched features. In the meantime, HAIL records the identities of the matched features, i.e., the landmarks to which these features belong, in a list of the source feature (the feature being matched).

Afterward, we show the list used to store the landmark IDs of the matched features in Figure 4. In this figure, we present the matched list corresponding to the h-th feature in Image i_{mn}. Let L be the length of the matched list. The l-th element in this list is denoted by g_h^l, which is the landmark ID of the l-th matched feature, and is defined as follows:

$$g_h^l = z, \tag{1}$$

where z is a scalar indicating the matched landmark of feature h. Please note that z is not a constant value. Instead, it is based on the comparison results of features.

Take the first element ($l = 1$) of the matched list corresponding to the first feature f_1 of the Image i_{11}; for example, $g_1^1 = 1$ because this feature matches with the feature f_1 of the Image i_{12}, which is a training image of Landmark 1.

We present lists of matched features in the red bounding box in the table in Figure 3.

3.2 Feature Voting

In the second feature voting stage, we calculate the number of votes for each feature. For instance, the number of votes for feature f_h of landmark m is:

$$v_h = \sum_{l=1}^{L} u\left(m, g_h^l\right), \tag{2}$$

where $u(c, d)$ is a function defined as follows:

$$u(c, d) = \begin{cases} 1 & \text{if } c = d \\ -1 & \text{otherwise} \end{cases}. \tag{3}$$

We determine the votes of each feature based on the votes from matched features. If a matched feature belongs to the same landmark with the source feature, it casts a positive vote. Otherwise, it casts a negative vote (Equation (3)). By evaluating the number of votes, we are able to reduce the impact of false matches because we jointly consider the matches between in-class images (images belonging to the same landmark) and out-class images (images belonging to different landmarks).

3.3 Distinguishing Feature Selection

In this last stage, we select distinguishing features (marked blue) in Figure 3 based on the number of votes for each feature. For example, the number of votes for the feature f_1 in Image i_{11} of

Landmark 1 is two because it has two matched features in Landmark 1 (two positive votes) and no matched features from other landmarks (zero negative vote). Similarly, the number of votes for the feature f_3 in Image i_{13} of Landmark 1 is two because it also has two matched features in Landmark 1. However, for the feature f_2 in Image i_{12} of Landmark 1, its number of votes is −3 because it does not match with any features in Landmark 1 but three different features in Landmark 2 (three negative votes). Taking the example from Section 3.2, the feature f_1 in Image i_{11} of Landmark 1 and the feature f_3 in Image i_{13} of Landmark 1 have the most votes among features of Landmark 1. Consequently, HAIL selects them as the distinguishing features for Landmark 1. Combined with the threshold and ratio test (Section 3.1), we are able to reduce the false matches of SURF.

3.4 Overall Workflow of Feature Selection

We summarize the process of feature selection in this section. Algorithm 1 details the process of feature selection. First, HAIL compares every image in the database with all other images and adds the landmark IDs of matched features to the corresponding matched list (Lines 4–12). Function FindMatchedPairs($i_{mn}, i_{m'n'}$) compares the image i_{mn} with the image $i_{m'n'}$ and returns a t-by-2 index array of matched features, where t is the number of matched features between these images. Afterward, HAIL calculates the number of votes for each feature: the feature that belongs to the same landmark as the source feature casts a positive vote while that which belongs to different landmarks casts a negative vote (Lines 15–19). Finally, HAIL selects the top Q features for each landmark as the distinguishing features (Lines 21–22). After selecting distinguishing features, we further reduce storage of feature files by adopting the feature compression method in Ref. [29].

We elaborate the process of feature selection by giving an example. Suppose we have three landmarks (Landmark 1, Landmark 2, and Landmark 3) in a site (Figure 3), each of which has three training images. The objective is to select two features for each landmark. In the first stage, we extract a fixed number of features (for example, three) from each training image, numbered as f_1, f_2, and f_3. Afterward, we compare these images with all other training images. For example, we compare Image i_{11} with the rest of the training images.

After image comparison, we show matched features connected by double-headed arrows in Figure 3. We can see from this figure that for Landmark 1, the feature f_1 in Image i_{11} matches with the features f_1 in Image i_{12} and Image i_{13} of the same landmark. Afterward, we record the matched list corresponding to each feature. Take feature f_1 in Image i_{11} of Landmark 1 for example. Its matched list has two elements: the landmark IDs of the matched feature in Image i_{12} and Image i_{13} both belong to Landmark 1. In this example, g_1^1 and g_1^2 are both equal to one. Since feature f_2 in Image i_{11} of Landmark 1 matches with one feature in Image i_{13} of Landmark 1, it has only one element in its matched list, which means $g_2^1 = 1$.

Finally, we select distinguishing features based on feature voting for each landmark. These features should have the most number of votes for the corresponding landmark. In this example, the feature f_1 of Image i_{11} and the feature f_3 of Image i_{13} are distinguishing features of Landmark 1 because they have the highest number of votes among features of Landmark 1 (two votes).

In indoor scenarios, the number of landmarks usually ranges from 50 to 100, which is different from large-scale image retrieval. For that reason, we utilize a scalable k-d tree structure [30] to efficiently search the matching landmarks.

We end this section by analyzing the computational complexity of feature selection. Given M landmarks, each with N training images, the complexity of comparing each image to other images is $O(M^2N^2)$. The complexity of selecting one feature for each landmark is bound by $O(HMN^2)$, where H is the number of features extracted for one training image. As the offline training stage is conducted once before online query, matching accuracy, rather than efficiency, is often the main focus (the selected features must be helpful for the online process). As this is conducted offline, it

ALGORITHM 1: Feature selection

```
input: Number of landmarks $M$; Number of training images for each landmark $N$; Number of features
       to extract from each image $H$; Number of features to select for each landmark $Q$.
output: Indices of selected features.
RefList ← Zeros($M$,$N$,$H$);
Score ← List($M$,$N$,$H$);
SelectedFeatures← List($M$);
for $m \leftarrow 1$ to $M$, $n \leftarrow 1$ to $N$, $m' \leftarrow 1$ to $M$, $n' \leftarrow 1$ to $N$ do
    if $m == m'$ && $n == n'$ then
        Continue;
    end
    indp ← FindMatchedPairs($i_{mn}, i_{m'n'}$);
    for $k \leftarrow 1$ to $t$ do
        RefList($m$,$n$,indp($k$,1)).Add(indp($k$,2));
    end
end
for $m \leftarrow 1$ to $M$, $n \leftarrow 1$ to $N$, $h \leftarrow 1$ to $H$ do
    for $l \leftarrow 1$ to RefList($m$,$n$,$h$).Length() do
        if $m ==$ RefList($m$,$n$,$h$,$l$) then
            Score($m$,$n$,$h$)++;
        else
            Score($m$,$n$,$h$)- -;
        end
    end
    indices ← GetTopFeatureIndices(Score($m$,:,:));
    SelectedFeatures($m$).Add(indices);
end
```

does not impact the online user experience of clients. For simplicity, we therefore conduct brute-force feature selection to select distinguishing features, though we have considered some efficient methods; for example, we can transfer to more efficient programming languages, such as Python or C++ rather than MATLAB. Apart from that, we can utilize multi-process programming to further speed up the feature selection. Based on these optimization strategies, we can reduce the time consumption by around 75% with four concurrent processes selecting distinguishing features for 21 landmarks each. Finally, we compare the complexity of landmark searching between sequential search and k-d tree. Given a feature, the complexity of searching a matching feature sequentially in the database with N features is $\mathcal{O}(N)$. With the help of k-d tree, the complexity is reduced to $\mathcal{O}(\log_k N)$, where k denotes the number of children in a node in k-d tree. Consequently, HAIL significantly reduces the time consumption of landmark searching.

4 AUTOMATED IMAGE-BASED LOCALIZATION

In this section, we present the strategy to locate users. First, we give a preliminary explanation of the operations needed by users to locate themselves in Section 4.1. Second, we elaborate the kurtosis-based landmark filtering in Section 4.2. Third, we generate the candidate region of the user based on the remaining landmarks in Section 4.3. Fourth, we elaborate the joint constraints-based localization in Section 4.4, followed by complexity analysis in Section 4.5.

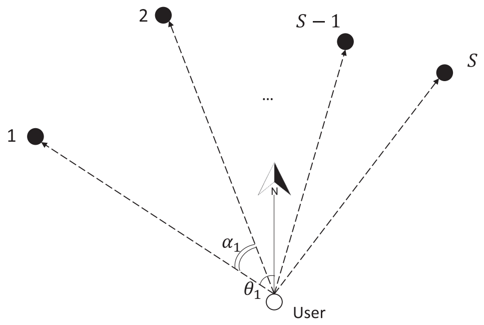


Fig. 5. The illustration of image and sensor fusion based localization.

4.1 Overview of Location Estimation

We first overview the entire localization process of HAIL. To achieve fast and accurate localization, we propose automated localization with joint constraints. The motivations are as follows:

- Reduce erroneous user operation. Selecting the target landmark manually is slow and error-prone. In our experiment, we observe that kurtosis is an effective indicator of whether HAIL successfully identifies the target landmark. Consequently, we select or filter the top matching landmark based on kurtosis, which reduces user operation and time consumption.
- Improve the localization accuracy. Due to the measurement noise and landmark mismatch, locating users based on either distances or angles is not always satisfactory. Hence, we jointly consider distances and angles in our problem. Furthermore, we employ a unified formulation to locate users to consider the error mitigation.

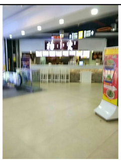
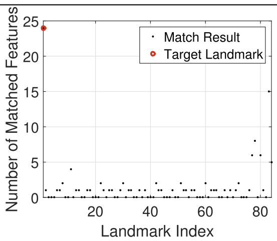
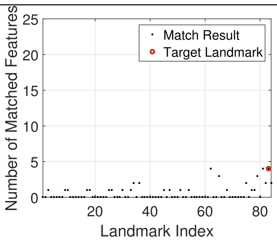
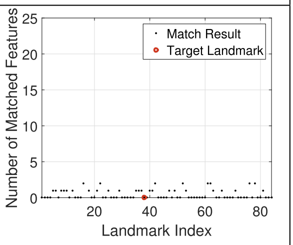
HAIL locates users based on several images of landmarks and corresponding gyroscope and compass readings. More specifically, the process of location estimation can be detailed as follows. Given M landmarks in the environment, the locations of these landmarks are denoted by $\boldsymbol{\Gamma} = \{\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_M\}$. Before localization, a user arbitrarily selects S landmarks in the environment. During the localization, that user stands on the current location, and spins arms and body to take one image for each of these S selected landmarks (Figure 5). Then the user queries the location based on motions sensors on the smartphone and these taken images, the so-called *input images*. To achieve sufficient localization accuracy, we recommend users to take three images ($S = 3$) to locate themselves. However, the localization error can decrease as the number of input images increases (refer to Figure 19). For the input image of landmark $s(1 \leq s \leq S)$, the angle relative to the true north determined based on the compass reading is θ_s. The rotation angle from $s-1$ to s relative to the user is α_{s-1}, which can be calculated based on gyroscope readings. HAIL jointly locates the user with input images, along with compass and gyroscope readings. Finally, HAIL presents the location of the user on the screen of mobile devices.

4.2 Kurtosis-based Landmark Filtering

In this section, we discuss how to identify the target landmark based on the kurtosis measure on the number of matched features.

After landmark matching, we obtain the number of matched features between the input image and each landmark. Take the input image of landmark s, for example. The numbers of matched features to M landmarks are denoted by $t_{s1}, t_{s2}, \ldots, t_{sM}$, respectively. Normally, the landmark with the largest number of matched features (the top candidate) is the target landmark (marked by the red circle) in the second example. However, this is not always the case. When comparing another

Table 3. Kurtosis Values in Different Test Scenarios

Scenario	Normal	Long Distance	Motion Blur
Query Image			
Matched Features			

input image to the database, the target landmark does not have the largest number of matched features due to practical challenges. For example, images of a far away landmark (with fewer detected features) and motion blur. Table 3 exemplifies these special scenarios. In the first case, a user takes a clear image of a nearby landmark. The kurtosis value is large and the target landmark has a significantly larger number of matched features. However, in some scenarios where a user may take an image of a far-away landmark (around 15m), the landmark is smaller and has fewer detected features. In this case, the kurtosis value is small. In addition to that, a user sometimes may take images with a shaking hand. In such case, the motion blur may be obvious, thus resulting in low-quality images. The number of strong features in the blurred images is also small, resulting in small kurtosis value and degraded matching accuracy. Based on above observations, we conclude that the landmark with a significantly larger number of matched features than others should be the target landmark.

We filter incorrect landmarks based on kurtosis, which measures the distribution of the matched feature numbers. Given M landmarks, the corresponding kurtosis value for the s-th input image is

$$\mu_s = \frac{E[(t_{sm} - \bar{t}_s)^4]}{(E[(t_{sm} - \bar{t}_s)^2])^2}, \tag{4}$$

where $\bar{t}_s$ denotes the average value of the matched feature numbers corresponding to the s-th input image. The number of matched features between this input image and landmark m is denoted as t_{sm}, where $1 \leq m \leq M$. The corresponding kurtosis value is denoted as μ_s.

To accept or filter landmarks automatically, we utilize binomial logistic regression [6] and the kurtosis value corresponding to each input image. The probability of accepting the top candidate corresponding to the s-th input image is:

$$P(Y_s = 1|\mu_s) = \frac{\exp(w\mu_s + b)}{1 + \exp(w\mu_s + b)}, \tag{5}$$

while the probability of filtering the top candidate of s is:

$$P(Y_s = 0|\mu_s) = \frac{1}{1 + \exp(w\mu_s + b)}, \tag{6}$$

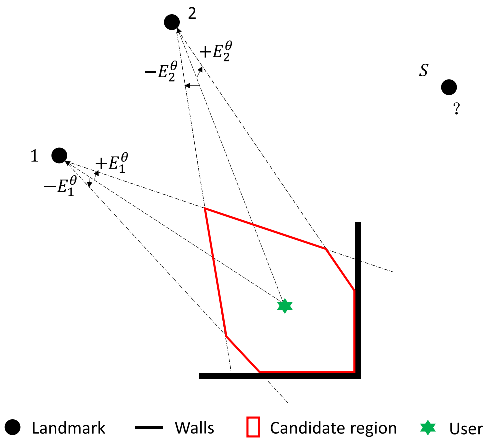


Fig. 6. The candidate region of the user is the intersection of possible regions. The measurement error of the compass is denoted as E_s^θ.

where $Y_s \in \{0, 1\}$ is the acceptance indicator of the top candidate, w is the weight, while b is the offset scalar. Finally, we can estimate the parameters by maximizing the likelihood estimation function with pairs of kurtosis and ground truth indicator values.

If $P(Y_s = 1|\mu_s)$ is larger than $P(Y_s = 0|\mu_s)$, we decide the top candidate is probably the target landmark (positive) and discard other candidate landmarks. Thus, the corresponding candidate set $\mathbf{U}_s$ has only one element, the top candidate. Otherwise, it is unlikely that the top candidate is the target landmark (negative). In this case, the candidate set $\mathbf{U}_s$ consists of the landmarks visible to the user. Moreover, we measure the performance of the landmark matching for the positive class by *F-score*, i.e.,

$$F = \frac{2 \times p \times r}{p + r}, \tag{7}$$

where the precision and recall of landmark matching are denoted by p and r, respectively. The F-score is 92.3% in our experiment.

4.3 Candidate Region Generation

After identifying the target landmark in the s-th input image, we can narrow down the location of the user to a corresponding constrained region, termed the *possible region*. This is a sector centered at $\boldsymbol{\gamma}_s$ and its central angle is two times the error of the compass, E_s^θ (Figure 6). With multiple input images, the location of the user can be narrowed down to the intersection of these possible regions. Due to the influence of magnetic disturbances from steel indoors, the compass readings are error-prone. We evaluate the performance of the system with random compass noise in the Section 5.

Apart from region constraint, the location of the user should follow the map constraint, i.e., the user should be in public areas. We incorporate the constraints of landmarks and maps and generate the candidate for each user (region with red border lines in Figure 6).

4.4 Optimization Under Joint Constraints of Motion Sensors and Floor Plan

In this section, we elaborate the localization strategy based on joint constraints.

The objective in our localization problem is to find a set of target landmarks, $1, \ldots, S$ and the location of user, $\mathbf{x}$, such that the angular differences

$$\mathrm{cost}(\mathbf{x}) = (1-\lambda)\sqrt{\sum_{s=1}^{S} (\theta_s^{\mathbf{x}} - \theta_s)^2} \; + \; \lambda\sqrt{\sum_{s=1}^{S-1} (\alpha_s^{\mathbf{x}} - \alpha_s)^2}, \tag{8}$$

are minimal. In Equation (8), λ is a weight constant. θ_s is the angle relative to true north measured by a compass, while α_s is the measured rotation angle from s to $s-1$. The corresponding calculated angles given estimated landmarks and $\mathbf{x}$ are $\theta_s^{\mathbf{x}}$ and $\alpha_s^{\mathbf{x}}$, respectively.

Via the experimental studies, we have observed that using angular difference alone to constrain localization may suffer from noisy angle measurement. Due to the complexity of a crowded indoor environment, different combinations of the landmarks may have similar angle measurements with respect to the target location. In order to mitigate the influence of noisy measurements, we fuse different sensing techniques to improve the localization accuracy. Next, we discuss our techniques as follows.

First, the location of the user should be in the region generated by landmarks and maps. In implementation, we use $\mathbf{A}$ to denote a set of lines (or rays that start at locations of landmarks). To sum up, the region constraints are denoted by

$$\mathbf{A}\mathbf{x} \geq \mathbf{b}, \tag{9}$$

where $\mathbf{b}$ is a column vector.

In addition to the region constraint, we use the mutual distances of landmarks and the distances between the user and landmarks to constrain the location estimation. Specifically, the sum of mutual distances between the estimated landmarks is:

$$\Psi = \sum_{\forall i, j \in [1,S], i \neq j} \frac{K(\boldsymbol{\gamma}_i, \boldsymbol{\gamma}_j)}{2}, \tag{10}$$

where $K(\boldsymbol{\gamma}_i, \boldsymbol{\gamma}_j)$ denotes the Euclidean distance between the estimated landmark i and j.

Moreover, we calculate the sum of distances between estimated location $\mathbf{x}$ and each of the estimated landmarks by the following equation:

$$\Phi(\mathbf{x}) = \sum_{s=1}^{S} K(\mathbf{x}, \boldsymbol{\gamma}_s), \tag{11}$$

where $K(\mathbf{x}, \boldsymbol{\gamma}_s)$ denotes the Euclidean distance between estimated location $\mathbf{x}$ and the landmark s.

In an indoor environment, users are unlikely to take images of distant landmarks. Therefore, the estimated location and the combination of landmarks should also comply with the distance constraint. As a result, we set an upper bound for the Ψ and $\Phi(\mathbf{x})$, respectively, according to the configurations of the specific site, i.e.,

$$\Psi \leq \tau_{\psi}, \quad \Phi(\mathbf{x}) \leq \tau_{\phi}, \tag{12}$$

where τ_{ψ} denotes the threshold for the sum of mutual distances between landmarks, and τ_{ϕ} denotes the threshold for the sum of distances between the user and landmarks.

To sum up, given S input images, corresponding readings from motion sensors and the floor plan, we want to find a combination of landmarks $[1, .., S]$ and location estimation $\mathbf{x}$ that satisfy

$$\begin{aligned} \underset{\mathbf{x}}{\arg\min} \quad & \text{cost}(\mathbf{x}), \\ \text{subject to} \quad & \mathbf{a_i} \in \mathbf{A}, \\ & \Psi \leq \tau_{\psi}, \\ & \Phi(\mathbf{x}) \leq \tau_{\phi}. \end{aligned} \tag{13}$$

We solve the above localization problem by a two-stage localization algorithm. The first stage of the algorithm is estimating the candidate region for the user. The second stage is locating the user by finding the estimated landmarks and location estimation with minimal angular differences.

In the first stage, we find the possible region relative to each target landmark based on the compass reading and sensor error. Afterward, HAIL generates several possible regions for all the estimated landmarks and constrains the user location estimation to the largest intersection of all the possible regions, the candidate region.

In solving the above problem, after finding the candidate region for the user, HAIL enumerates the combinations of landmarks and locates the user at the position with minimal angular differences.

4.5 Complexity Analysis

Finally, we analyze the computational complexity of our localization algorithm in HAIL. Given P possible locations of a user and M landmarks, the complexity of determining the user's possible regions of each estimated landmark is $O(PM)$. Let P' be the number of possible locations subject to region constraints and M' be the subset of landmarks that are visible from the possible locations. Suppose HAIL filters s queries with low kurtosis values, and then the complexity of finding the location and landmarks combination with minimal angular difference is $O(P'M'^s)$. To summarize, the total complexity of localization is $O(PM + P'M'^s)$.

5 EXPERIMENTAL EVALUATION

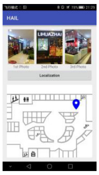
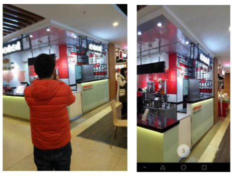
To evaluate its performance, we have developed a mobile application of HAIL so that users can obtain their locations by taking photos (this can be easily extended to video stream by extracting key frames). We show in Table 4 the user interface and an illustration of localization procedures. Users need to select at least three landmarks and take images of them to locate themselves. Then this application displays images taken by this user and indicates the estimated user location as a blue marker. We have conducted extensive experiments in GoGo Plaza (a food plaza in Figure 7(a)) and Grandview Mall (a premium shopping mall in Figure 7(b)) in Guangzhou. To demonstrate the generality of HAIL in a featureless environment, we conduct another experiment in the academic building of the Hong Kong University of Science and Technology (HKUST) (Figure 7(c)), which consists of long corridors with many offices (termed *office area*). This area has sparse landmarks and repetitive structures. We evaluate in this site how the introduction of door plates and banners helps by improving the localization accuracy.

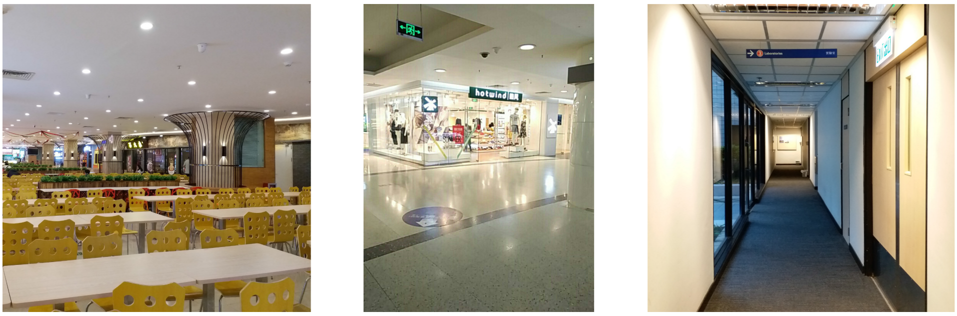
In this section, we first discuss our experimental settings (Section 5.1). Afterward, we evaluate the accuracy of landmark matching (Section 5.2), localization error (Section 5.3), the system overhead (Section 5.4), and the user study (Section 5.5).

5.1 Experimental Settings and Comparison Schemes

Figure 8(a)–(c) exhibits floor plans of three experimental sites. The food plaza is compact, covering around 6,000m^2, while the shopping mall is large, covering around 16,000m^2. We regard logos

Table 4. The User Interface of HAIL and the Procedures to Localization

			
User interface.	Step 1. Take the first input image.	Step 2. Take the second input image.	Step 3. Take the third input image.



(a) Food plaza. (b) Shopping mall. (c) Office area.

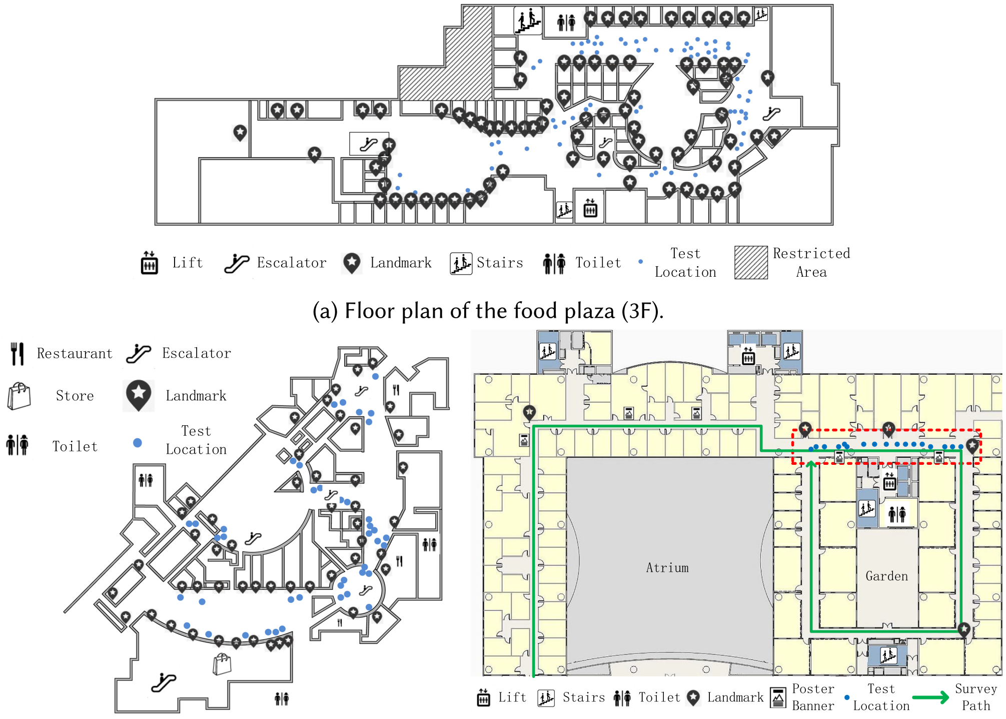
Fig. 7. Three test sites in our experiment. There are abundant visual features in the food plaza and shopping mall. While in the office area of our university, visual features are sparse.

of stores as landmarks in these experiments. Figure 8(c) shows the test site (around 5,400m^2) on the third floor of the HKUST academic building. It is different from the shopping mall and food plaza in that it has repetitive structures and sparse visual features. We demonstrate that other complementary visual clues, such as banners and door plates, help by improving the localization accuracy in this site.

There are 84 landmarks in the food plaza and 51 in the shopping mall. We acquire an accurate floor plan for each of the testing sites and carefully annotate the locations of all landmarks on the map. In the experiment, we implement an Android application to take training images with a fixed resolution (800 × 600). The quality of the image is set to the highest to ensure high matching accuracy (the size of an image is around 120KB). The total number of training images in the food plaza is 588 and the number of training images in the shopping mall is 357. To train the logistic regression, we take another 71 images in the food plaza and 42 images in the shopping mall.

We conduct experiments at 73 test locations in the food plaza and 41 test locations in the shopping mall (denoted by the blue points in the floor plans). In addition, we select 45 test points in the food plaza and take seven images at each location. The test locations are also manually tagged based on nearby landmarks of known locations. Figure 9 shows the distribution of distances from test locations to corresponding landmarks in these sites. It depicts that distances between stores in this shopping mall are longer than those in the food plaza. This is because stores in this shopping mall are generally larger and are more sparsely distributed.

Unless otherwise stated, we use parameters in Table 5 as the baseline. We implement HAIL on the Android platform and conduct experiments with different devices. To extract SURF from images, our system uses VLFeat library [45]. We compare the performance of HAIL with the following state-of-the-art image-based localization schemes on smartphones:



(a) Floor plan of the food plaza (3F).

(b) Floor plan of the shopping mall (3F).

(c) Office area in our university.

Fig. 8. Floor plans of three test sites, where blue points denote test locations.

- Sextant [11]: Sextant employs the benchmark selection algorithm [11] to select images and construct the image database. In the localization phase, Sextant asks each user to take three images of the environment and confirm the candidate images. After user confirmation, it uses triangulation to locate the user. In case image matching fails, Sextant estimates the location of the user based on geographical constraint.
- MoVIPS [48]: MoVIPS locates each user with the location of the most matched image in the database. It first finds the most similar training image to the input image. Afterward, it estimates the distance between the user and the location of the matched image based on the camera field-of-view and the average ratio of distances between each pair of matched local features. Finally, MoVIPS locates the user based on location of the matched image and the estimated distance.

HAIL constructs a feature database with distinguishing features. For Sextant, we implement a benchmark selection algorithm to select one image for each landmark. The database of MoVIPS consists of all training images. In the localization stage, the number of features extracted for comparison is equal among training images. For instance, if the total number of features for each landmark is 700 and the number of training images is seven, then the number of features extracted in each training image is 100 (=700÷7). In addition to the above localization algorithms, we compare our feature selection with that proposed in Ref. [36]. Specifically, we build one 3D model for each landmark with our training images and select 3D points as well as corresponding 2D features to build the feature database. Finally, we select different numbers of features using HAIL and Ref. [36], and compare their matching accuracy in our test site.

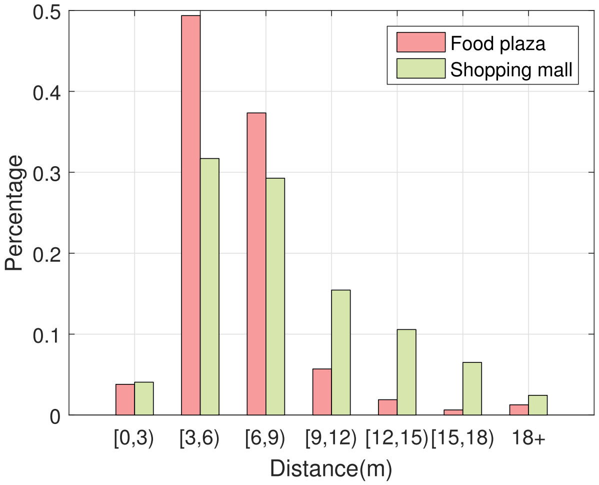


Fig. 9. The distribution of distances between stores and corresponding test locations.

Table 5. Baseline Parameters Used in HAIL

Name	Value	
	Food plaza	Shopping Mall
# of training images	6	
# of features per landmark	500	
Image resolution	800×600	
τ_ψ	30m	
τ_ϕ	30m	

We evaluate the performance of HAIL using the following metrics:

(1) *Matching accuracy*: Let w_i denote the ground truth landmark for the input image i, and let w_i' be the estimated landmark for the same input image. We use q_i to indicate whether the estimated landmark $w_i' = w_i$. If the estimated landmark is the ground truth ($w_i' = w_i$), the corresponding q_i is set to one. Otherwise, its value is zero. Given a set of images W, the mean matching accuracy is

$$\text{acc} = \frac{1}{|W|}\sum_{i=1}^{|W|} q_i. \tag{14}$$

(2) *Localization error*: Denote the ground truth location of the user by $\mathbf{x}_i$ and the estimated location of the user by $\mathbf{x}_i'$. Suppose there are Z test locations; the mean localization error is given by

$$e = \frac{1}{|Z|}\sum_{i=1}^{|Z|} ||\mathbf{x}_i - \mathbf{x}_i'||_2, \tag{15}$$

where $||\cdot||_2$ is an L^2 norm operator.

(3) *Computation time*: In order to evaluate the time consumption, we measure the mean time consumption of landmark matching and localization by summing up the time consumption of the corresponding process of all the test cases and dividing them by the number of test cases.

Table 6. Main Technical Specifications of Test Smartphones

Specifications	Mate 7	MI 4	C5 Pro	Mate 9
CPU	HiSilicon Kirin 925	Qualcomm Snapdragon 801	Qualcomm Snapdragon 626	Hisilicon Kirin 960
Battery Capacity	4100mAh	3080mAh	2600mAh	4100mAh

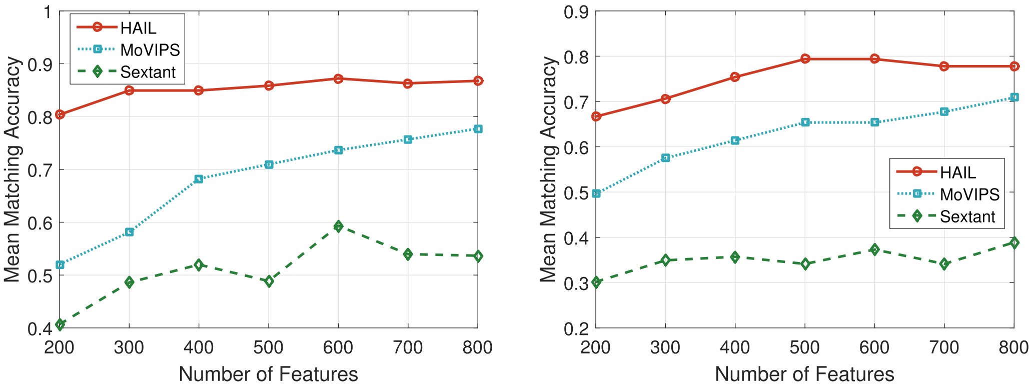
As discussed in Section 3, the feature selection algorithm selects a few distinguishing features for each landmark (say, 1,000). These features are in the descending order of the distinctiveness. To evaluate the matching accuracy with different numbers of features, we select the top 200, 300, 400, 500, 600, 700, and 800 features for each landmark and evaluate the matching accuracy. To achieve a fair comparison, we consider the matching accuracy of different strategies based on the top candidate (the landmark with the largest number of matched features) without landmark filtering. Apart from the illustrative results of landmark matching, we use the kurtosis method to filter landmarks and present the localization error. Moreover, we compare the performance of HAIL and Sextant when not all of the landmarks are correctly matched. That is, when these localization systems fail to identify target landmark(s) from one or two input images correctly. To this end, we select the test cases where not all the landmarks are correctly matched and compare the localization error between HAIL and Sextant.

In order to evaluate the impact of resolution on localization error, we upsample and downsample the images and repeat the experiments for each scale. We also evaluate the performance of our algorithm when the user fails to point at the center of the landmark and the impact of nearby ferromagnetic objects. Although different, they lead to similar numerical errors on the location estimation. Without loss of generality, we conduct a simulation to evaluate the performance of the proposed method with compass error. To this end, we add Gaussian noise with zero mean and different variances to the measured compass readings and simulate the localization results.

In the office area, we select logos of departments as landmarks (five in the test site) and augment the visual database with posters on the wall (four in this site). Similar to the above, we take seven images for each landmark and poster. We also conduct 17 tests in the dashed red bounded area in Figure 8(c) where we are able to see at least three landmarks or posters. To evaluate the localization accuracy with door plates, we walk along the corridor (path denoted by the green solid lines) and take one image every second. The number of frames along the path is 393, where 96 of which have clear views of door plates. We use the Tencent Youtu OCR[2] to recognize room numbers in these 96 images. In this corridor, if OCR recognizes the door number in the image successfully, the localization error is zero. This is because the corridor is narrow (1.5m to 3m in width), and we have to be in close range to take clear images with recognizable plate numbers. Otherwise, we localize this user to the last known position. Please note that many other companies, such as Baidu, Google, and Microsoft, provide free OCR APIs or toolkits, which can also be easily integrated into our system.

We evaluate the power consumption of these localization systems with four different Android devices: Huawei Mate 7, Xiaomi MI 4, Samsung Galaxy C5 Pro, and Huawei Mate 9 (with dual cameras). Table 6 presents the detailed evaluations of these devices. In this experiment, we randomly select 200 localization test cases (with repetition) and conduct localization consecutively

[2] https://ai.qq.com/hr/youtu.shtml.



(a) Food plaza.

(b) Shopping mall.

Fig. 10. Accuracy of landmark matching vs. number of features per landmark.

with Espresso.[3] Afterward, we calculate the overall power drop after the test. Then we determine the average power consumption of each test by dividing the number of test cases.

In order to evaluate the response from users to the system, we conducted user studies. Fifty-seven volunteers from different laboratories in our school took part in this study including 21 female and 36 male volunteers. In this study, we asked each volunteer to take three images and locate the tester using Sextant and our HAIL, respectively. Afterward, we asked the tester to give scores on the localization accuracy, automaticity, time consumption, and the like.

5.2 Accuracy of Landmark Matching

First, we evaluated the accuracy of landmark matching in different test sites. Then we presented the matching accuracy in the food plaza with different parameter settings.

Figure 10(a) shows that the mean accuracy of landmark matching increases with more features. This is because more features can provide more comprehensive information of the landmark. However, the accuracy becomes stable with more than around 600 features. This is because we have sufficient information. However, the time consumption of landmark matching increases with more features. To achieve tradeoff between the accuracy and time consumption, we select the top 500 for each landmark in our experiment. With feature selection, HAIL is able to achieve higher landmark matching accuracy than MoVIPS and Sextant because it discards noisy and frequent features. MoVIPS achieves sufficient matching accuracy at the cost of higher storage consumption (seven times larger than Sextant). Different from MoVIPS, Sextant uses the feature information from one benchmark image. In addition, it does not discriminate robust features from noisy ones. As a result, the matching accuracy of Sextant is lower than HAIL and MoVIPS in our experiment.

Figure 10(b) shows the accuracy of landmark matching in the shopping mall, which first increases with more features. This is because we have more information for each landmark. However, as the feature number is larger than 500, the matching accuracy becomes stable and even begins to decrease. This is because we have redundant or noisy features in the database. Experimental results show that the matching accuracy in the shopping mall is lower than that in the food plaza. The reasons are as follows. The number of detectable features of the distant landmark is fewer, therefore, less information is provided to search the target landmark. Apart from that, due

[3] https://developer.android.com/training/testing/espresso/index.html.

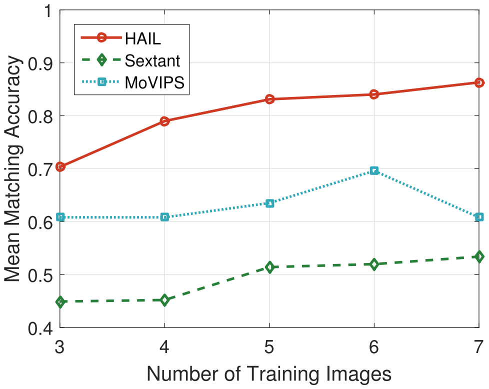


Fig. 11. Accuracy of landmark matching vs. number of training images (Food plaza).

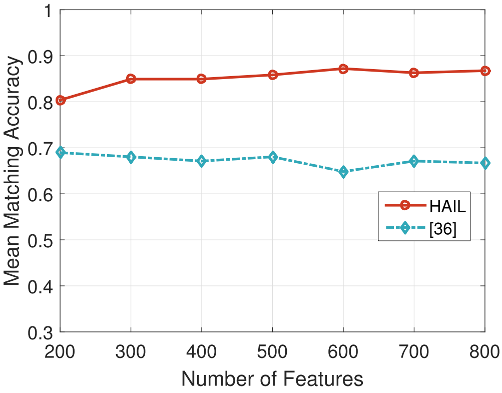


Fig. 12. Comparison of feature selection (Food plaza).

to longer distances, it is more likely to extract noisy features from the input image. Consequently, the overall matching accuracy in the shopping mall is lower.

Figure 11 presents the accuracy of landmark matching with different numbers of training images. The accuracy of landmark matching increases with the number of training images. This is because distinguishing features can get more votes with more training images. With more votes, these distinguishing features can be selected. Based on the experimental results, we suggest taking six training images for each landmark to achieve sufficient accuracy without much survey cost.

Figure 12 compares the matching accuracy of different feature selection schemes. It demonstrates that our algorithm is able to achieve higher accuracy than that proposed in Ref. [36]. This is because our scheme finds distinguishing features based on both in-class similarity and out-class distinctiveness. Thus, we are able to retain distinguishing features and remove noisy and duplicate ones in different landmarks. In contrast to ours, Sattler et al. [36] select features for each landmark without considering other landmarks. Furthermore, they may select multiple features corresponding to one 3D point, resulting in redundancy and a lack of diversity. The matching accuracy of Ref. [36] remains stable with more features. This is because SIFT is of a high dimension and contains sufficient information with a small number of features. Consequently, the accuracy does not change much with more features. However, it is possible to add noise with more features, thus degrading the matching accuracy slightly.

5.3 Localization Error

In this part, we evaluate the localization error in different test sites.

Figures 13 and 14 present the localization error of HAIL at two experimental sites. Frequent features indoors can lead to a low matching accuracy because they are more likely to match with other features. Since the method estimates the location of the user based on locations of matched images, the incorrect match result can increase the localization error. HAIL is able to select distinguishing features and discard the noisy and frequent ones with feature selection. As a result, the accuracy of landmark matching is higher. Combined with joint constraints, the localization error is smaller. The localization error in the shopping mall is larger. The reason is that in the shopping mall, the distances between the user and landmarks are longer. As a result, the number of detectable features from distant landmarks is fewer, which can decrease the matching accuracy. In addition, the mentioned methods infer the location of the user based on positions of matched landmarks. Therefore, the localization error increases as the matching accuracy decreases.

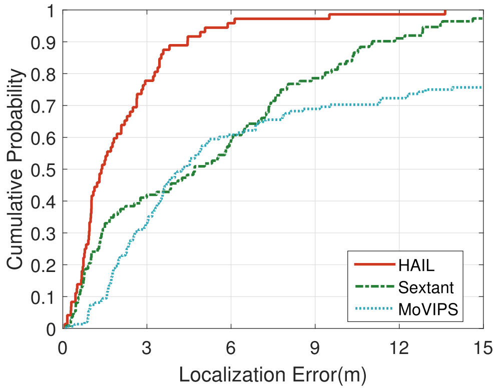


Fig. 13. CDF of localization error in the food plaza.

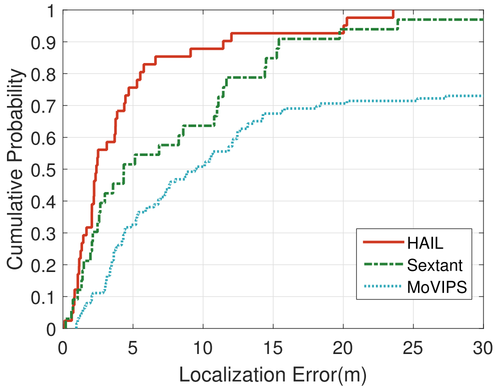


Fig. 14. CDF of localization error in the shopping mall.

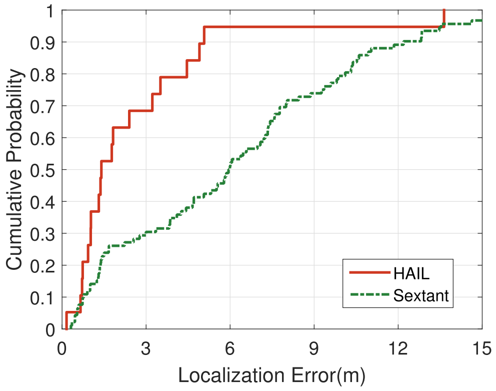


Fig. 15. CDF of the localization error with matching failure (Food plaza).

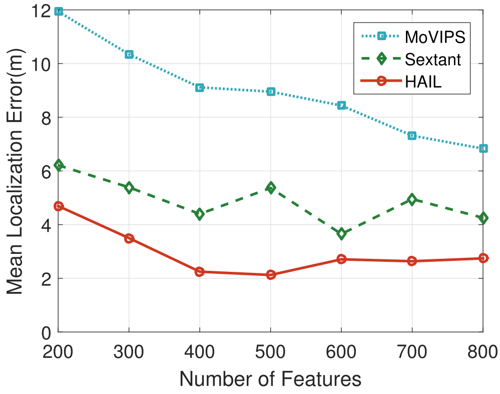


Fig. 16. Localization error vs. number of features per landmark (Food plaza).

Figure 15 presents the localization error when localization systems fail to identify one or two of the landmarks correctly. It shows that HAIL outperforms Sextant in our experiment. The reason is that HAIL jointly utilizes motion sensors, distance, and region constraints to restrain the location estimation. Therefore, HAIL is less sensible to sensor error and can achieve higher accuracy compared with Sextant.

Figure 16 depicts the mean localization error with different numbers of features per landmark. We can infer from this figure that more features can lead to a lower localization error. It is because more features per landmark can provide more comprehensive descriptions of the corresponding landmark and, hence, the matching accuracy is improved. If the number of features is around 500, the localization error remains stable because we have a sufficient number of features to represent the landmarks. As more features are used, noise can be added to the feature database. Therefore, the localization error of HAIL gradually increases. To achieve tradeoff between the localization accuracy and efficient storage, we select 500 features in our experiment.

Figure 17 shows the impact of image resolution on the localization error. It shows that the localization error decreases with the increase of image resolution. This is because high-resolution images usually contain more information. As a result, more features can be utilized to represent each landmark, which leads to higher accuracy of landmark matching. Combined with region and

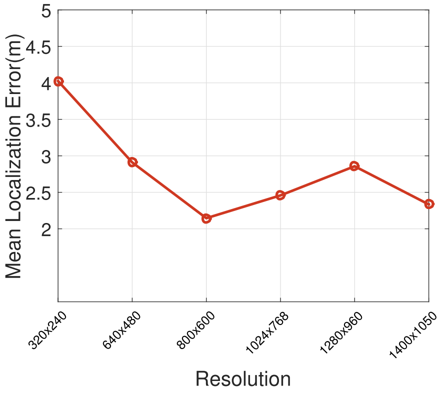


Fig. 17. Mean localization error vs. image resolution (Food plaza).

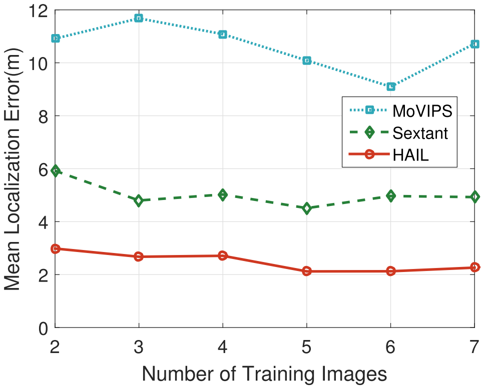


Fig. 18. Mean localization error vs. number of training images (Food plaza).

distance constraints, the overall localization error decreases significantly (from 4.0m at 320 × 240 to 2.3m at 1400 × 1050). However, we can see that the localization error increases from 800 × 600 to 1280 × 960. This is because the noise in original images can be amplified in the upsampling process, which degrades the accuracy of landmark matching. If the resolution is larger than 1280 × 960, the single noisy pixel in original images can become several more pixels, even *blobs*. In this case, these blobs are no longer considered noise. Instead, they become *objects*. With the feature selection, these blobs can be filtered due to their repetitive patterns and, thus, the matching accuracy increases. In this case, the localization error starts to decrease again. However, it takes more time to extract features from images with higher resolution. To achieve tradeoff between localization accuracy and efficiency, the resolution of images is 800 × 600 in our experiment.

Figure 18 presents the localization error with respect to the number of training images. It shows that the localization error decreases with the number of training images. The reason is that more training images can help by selecting distinguishing features and discarding frequent ones, thus, it leads to higher accuracy of landmark matching and lower localization error. However, the improvement of localization accuracy decreases with more training images. It is because after giving six training images, the information is sufficient for localization. Newly added images may add noise to the location estimations. As a result, taking around six training images for each landmark can achieve sufficient localization accuracy.

Figure 19 presents the localization error with different numbers of input images. It shows that the mean localization error decreases with more input images. This is because with more input images and sensor readings, HAIL is able to reduce the impact of wrong matching results. After a certain number of input images (three in our experiment), the improvement on localization accuracy is small. To reduce the user operation, we think it enough to take three input images to achieve sufficient localization accuracy.

Figure 20(a) illustrates the localization with different compass error ranges. This value corresponds to the E_i^θ in Figure 6. It shows that the localization error decreases with a larger error range. This is because a larger estimation of compass error allows the algorithm to generate a larger possible region based on each landmark, which reduces the chance to incorrectly filter the ground truth location of the user. Consequently, our algorithm can estimate a more accurate location of the user. However, the search space of the user location and the localization time can increase with the compass error range. In our experiment, setting the compass error to 20 can achieve sufficient localization accuracy without much time overhead.

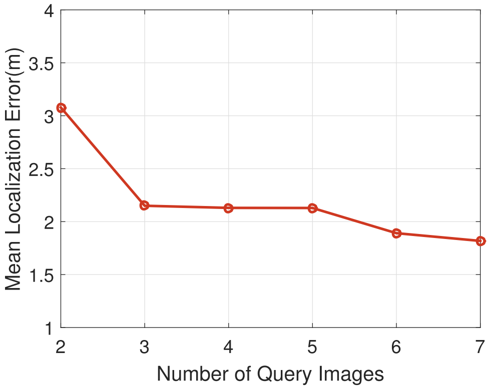
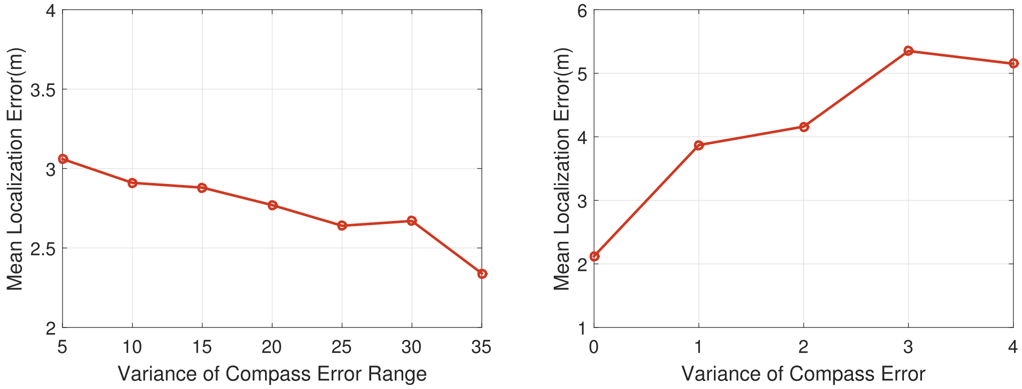


Fig. 19. Mean localization error vs. number of input images (Food plaza).



(a) Mean localization error v.s. Compass error range (Food plaza).

(b) Mean localization error v.s. The variance of compass error (Food plaza).

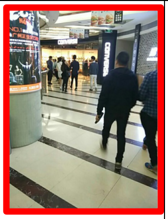
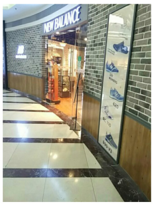
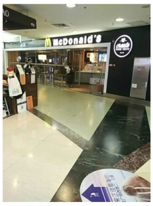
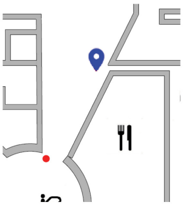
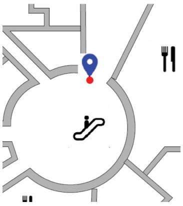
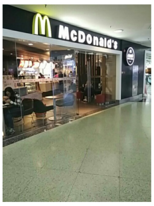
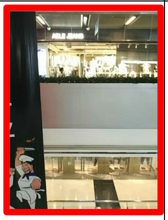
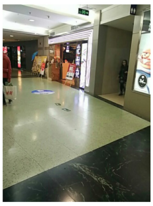
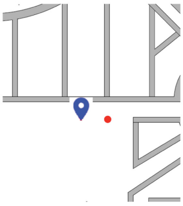
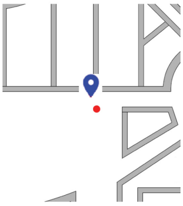
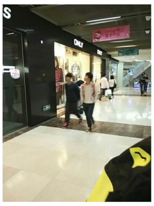
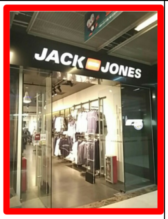
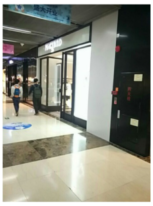
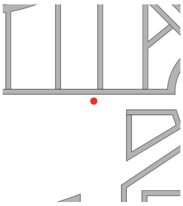
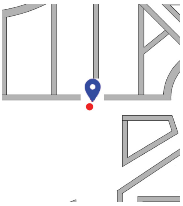
Fig. 20. Evaluation of compass noise on the localization error.

Figure 20(b) shows the localization error when the center of the camera does not align with that of a landmark. As in our formulation, we set λ to 0.74 according to our empirical deployment. We set a higher weight for the gyroscope difference because it is usually more accurate to reduce the impact of a noisy compass on the cost function to avoid large localization error. However, if the center of a camera is distant from that of a landmark, the localization error can increase.

We present three special test cases and the corresponding localization results in Table 7, where ground truth locations are denoted as red-filled circles and estimated locations are denoted as blue markers. In the first scenario, users take images of landmarks in a cluttered environment with many pedestrians (the first input image). In the second scenario, users take images from a long distance (the second input image is 15 meters away). In the third scenario, one tester accidentally confirms the wrong matching result for the second image. In this case, the triangulation fails since the confirmed landmarks do not match with the sensor readings. As HAIL is automatic, it does not require error-prone user operation, which makes it more robust and suitable for robotic applications.

We illustrate the mean localization error in the office area with only landmarks (denoted by LM) and with additional posters and door plates (denoted by LM+Poster+Door Plate) in Figure 21. It shows the introduction of posters and door plates improves the localization accuracy significantly

Table 7. Sample Localization Results in Challenging Scenarios

Scenarios	Input images			Localization results	
				Sextant	HAIL
Cluttered scenario with many pedestrians	 CONVERSE	 NEW BALANCE	 MacDonald's	 Distance: 17.2m	 Distance: 1.1m
Long distance	 MacDonald's	 ABLE JEANS	 Chinese cuisine	 Distance: 4.2m	 Distance: 2.5m
Careless user operation	 ONLY	 JACK & JONES	 PEACEBIRD	 Distance: Not available	 Distance: 0.625m

in our test. This is because OCR technique is able to recognize door plates in the constrained office area accurately. In addition, as door plates are more pervasive in the office environment, they can provide more general and accurate location clues even if no landmarks are in the line-of-sight. Based on our experimental results, we demonstrate that using posters and door plates can boost the localization accuracy and enhance the generality of visual indoor localization systems in these sites.

5.4 System Overhead

In this part, we evaluate the storage, time needed for feature selection and localization, as well as power consumption.

The feature database of 84 landmarks in our experiment requires 316MB of storage (described in Section 1). Through feature selection and compression, the size of our feature database in HAIL is 2MB. As a result, HAIL can be easily deployed on smartphones.

In the offline stage, it takes around an hour to select distinguishing features in the food plaza on a MacBook Pro with an Intel i7 CPU and 8GB RAM without performance optimization. We can further reduce the time consumption by multi-thread programming. Please note that service providers conduct feature selection in the offline stage, which does not affect localization experience. Moreover, feature selection reduces the storage significantly so that clients can download the specific database on demand and get the location estimation in a short time with sufficient accuracy.

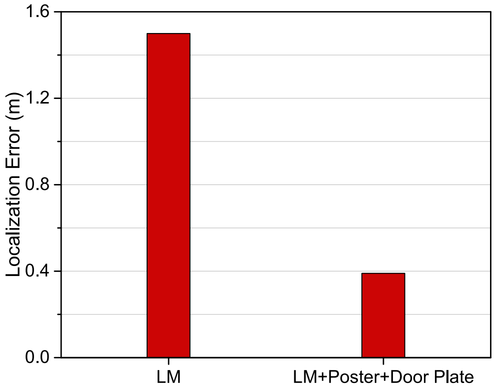


Fig. 21. Mean localization error without/with posters and door plates (Office area).

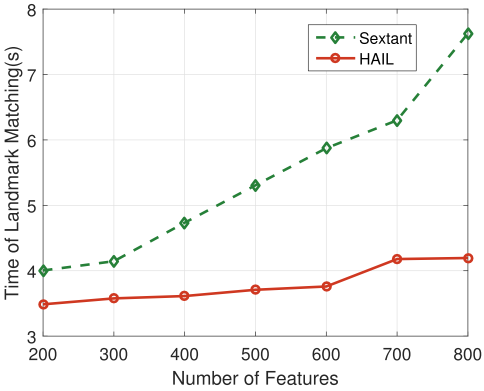


Fig. 22. Time of landmark matching vs. number of features per landmark (Food plaza).

Table 8. Detailed Evaluation of Power Consumption with Different Devices (HAIL)

Stages	Mate 7	MI 4	C5 Pro	Mate 9
Image Taking	0.819mAh	0.882mAh	0.456mAh	1.119mAh
Landmark Filtering	1.524mAh	1.188mAh	0.503mAh	0.668mAh
Localization	0.016mAh	0.015mAh	0.004mAh	0.007mAh
Total	2.359mAh (0.058%)	2.085mAh (0.067%)	0.963mAh (0.037%)	1.794mAh (0.043%)

Figure 22 shows the time of landmark matching with different numbers of features. It can be seen that the time consumed by Sextant increases faster with features per landmark. Due to the efficiency and scalability of k-d tree, the time consumption of HAIL is lower and increases more slowly with the number of features. In the food plaza, MoVIPS needs to make 6 × 84 image comparisons (there are 84 references in the food plaza; each of these references has six training images in the database), which is approximately two times more than that of Sextant (total number of comparisons, 3 × 84).

In our implementation, it takes around 0.8 second to extract features from one training image using OpenCV[4] SURF extractor. As a result, storing the features directly on a smartphone can help reduce the time consumption significantly (about 67 seconds in the food plaza). In the process of landmark matching, it takes around 3.7 seconds to retrieve landmarks for three input images. The localization takes around 12 milliseconds. As a result, the total time for localization takes less than 4 seconds. Compared with HAIL, Sextant needs more time to conduct image searching (around 5.2 seconds). Since it takes 1 second to confirm the matching result of one taken image, the user may need at least three more seconds before the location estimation is shown. Therefore, HAIL is able to reduce the time consumption of image matching and localization by around 40%.

Table 8 presents the average power consumption of HAIL in different stages. It shows that the power consumption is significantly lower than the capacity of batteries (usually less than 0.07%). The overall power consumption of Samsung C5 Pro and Huawei Mate 9 is lower than Huawei Mate 7 and Xiaomi MI 4. This is mainly because the latter models have more power-efficient processors.

Figure 23 illustrates the energy consumption of one localization query with different mobile phones. Since MoVIPS acquires only one image for input, the power consumption of MoVIPS is

[4] http://opencv.org/.

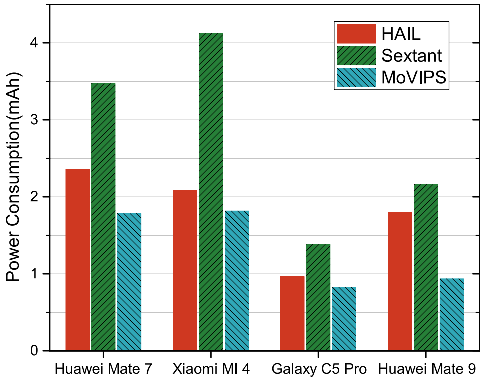


Fig. 23. Comparison of the power consumption of different devices.

Table 9. Statistical Analysis of HAIL

Metric	Negative	Neutral	Positive
Time	5%	25%	70%
Convenience	7%	25%	68%
More automated	7%	21%	72%
Localization accuracy	2%	18%	80%
Storage	35%	40%	25%
Preference	7%	19%	74%

less than HAIL and Sextant, with three input images. However, HAIL achieves a higher localization accuracy than MoVIPS with a similar power consumption. It is because we reduce the number of comparisons by feature selection and employ an efficient k-d tree searching algorithm. The overall power consumption with Samsung C5 Pro and Huawei Mate 9 is lower than Huawei Mate 7 and Xiaomi MI 4. The reasons are two-fold. First, the Samsung C5 Pro and Huawei Mate 9 employ more recent power-efficient CPUs. Second, the batteries of Huawei Mate 7 and Xiaomi MI 4 degrade after 3 years of usage; thus, the power levels drop at a faster rate.

5.5 User Study on the Usability of HAIL

To evaluate how users respond to our system, we conducted a user study and presented the evaluation results as follows.

We invited 57 volunteers to take part in this study. These volunteers are from three research laboratories in our school, including students and teachers. Each of these volunteers was asked to rate the above three systems on the time consumption, convenience, automation, localization accuracy, storage consumption, and preferences from 1 to 5. If the given score is 4 or 5, the volunteer feels positive (this indicates short time consumption, convenient to use, high localization accuracy, and low storage consumption). However, a user feels "negative" if the given score is lower than 2 (this indicates long localization time, inconvenient to localize, low localization accuracy, and high storage consumption). Otherwise, the volunteer feels neutral about the results.

We summarized the evaluation results in Table 9, which shows that many volunteers in our study feel that HAIL is convenient to use as it is more automated, takes shorter time to locate users, and consumes less storage on smartphones. Many of them are not reluctant to use HAIL in the real world. In this study, we find a few volunteers are not likely to use this application as they feel a bit embarrassed to take photos with smartphones indoors. Our HAIL is deployable on smart devices, such as smart glasses, drones, and robots with onboard cameras. This reduces tedious image-taking and, thus, is more user-friendly.

6 CONCLUSION

Image-based indoor localization has attracted much attention recently. Previous approaches in the area are often difficult to be deployed on resource-limited mobile devices due to their storage and processing requirements. Furthermore, some require manual user confirmation in order to achieve a satisfactory level of localization accuracy.

We have proposed HAIL, an automated image-based localization algorithm deployable and distributable on smartphones. To reduce memory/storage and processing requirements, we select and store only those distinguishing features of landmarks on smartphones. To achieve automation, HAIL employs an efficient kurtosis method to filter away the incorrect candidate images. It further improves the localization accuracy with joint constraints of motion sensors and floor plan. We have implemented HAIL in mobile phones, and conducted extensive experiments in a food plaza and a shopping mall in Guangzhou. Our results show that HAIL enhances image matching, and reduces the localization error significantly (by more than 20%) and is easily distributable in mobile platforms. An evaluation of 57 testers proves that HAIL is able to achieve higher automaticity than Sextant.

REFERENCES

[1] Hyojoon Bae, Michael Walker, Jules White, Yao Pan, Yu Sun, and Mani Golparvar-Fard. 2016. Fast and scalable structure-from-motion based localization for high-precision mobile augmented reality systems. *mUX: The Journal of Mobile User Experience* 5, 1 (19 Jul 2016), 4.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-up robust features. *CVIU* 110, 3 (June 2008), 346–359.

[3] Raj Bista, Suman, Robuffo Giordano Paolo, and François Chaumette. 2016. Appearance-based indoor navigation by IBVS using line segments. *IEEE Rob. Autom. Lett.* 1, 1 (2016), 423–430.

[4] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Luigi Di Stefano, and Philip H. S. Torr. 2017. On-the-fly adaptation of regression forests for online camera relocalisation. In *Proc. IEEE CVPR.* 218–227.

[5] Yin Chen, Dimitrios Lymberopoulos, Jie Liu, and Bodhi Priyantha. 2013. Indoor localization using FM signals. *IEEE Trans. Mob. Comput.* 12, 8 (Aug 2013), 1502–1517.

[6] David W. Hosmer Jr., Stanley Lemeshow, and Rodney X. Sturdivant. 2013. *Applied Logistic Regression* (3 ed.). Wiley.

[7] Lei Deng, Zhixiang Chen, Baohua Chen, Yueqi Duan, and Jie Zhou. 2016. Incremental image set querying based localization. *Neurocomput.* 208, 5 (Oct 2016), 315–324.

[8] Jiang Dong, Yu Xiao, Marius Noreikis, Zhonghong Ou, and Antti Ylä-Jääski. 2015. iMoon: Using smartphones for image-based indoor navigation. In *Proc. ACM Sensys.* 85–97.

[9] Jiang Dong, Yu Xiao, Zhonghong Ou, Yong Cui, and Antti Ylä-Jääski. 2016. Indoor tracking using crowdsourced maps. In *Proc. IEEE IPSN.* 1–6.

[10] Moustafa Elhamshary, Anas Basalamah, and Moustafa Youssef. 2017. A fine-grained indoor location-based social network. *IEEE Trans. Mobi. Comput.* 16, 5 (May 2017), 1203–1217.

[11] Ruipeng Gao, Yang Tian, Fan Ye, Guojie Luo, Kaigui Bian, Yizhou Wang, Tao Wang, and Xiaoming Li. 2016. Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment. *IEEE Trans. Mob. Comput.* 15, 2 (Feb. 2016), 460–474.

[12] Ruipeng Gao, Bing Zhou, Fan Ye, and Yizhou Wang. 2017. Knitter: Fast, resilient single-user indoor floor plan construction. In *Proc. IEEE INFOCOM.* 1–9.

[13] Fei Gu, Jianwei Niu, and Lingjie Duan. 2017. WAIPO: A fusion-based collaborative indoor localization system on smartphones. *IEEE/ACM Trans. Networking* 25, 4 (Aug 2017), 2267–2280.

[14] Bin Guo, Qi Han, Huihui Chen, Longfei Shangguan, Zimu Zhou, and Zhiwen Yu. 2017. The emergence of visual crowdsensing: Challenges and opportunities. *IEEE Commun. Surv. Tutorials* 19, 4 (Aug 2017), 2526–2543.

[15] Suining He, S.-H. Gary Chan, Lei Yu, and Ning Liu. 2018. Maxlifd: Joint maximum likelihood localization fusing fingerprints and mutual distances. *IEEE Trans. Mob. Comput.* (2018), to appear.

[16] Suining He, S.-H. Gary Chan, Lei Yu, and Ning Liu. 2018. SLAC: Calibration-free pedometer-fingerprint fusion for indoor localization. *IEEE Trans. Mob. Comput.* 17, 5 (May 2018), 1176–1189.

[17] Suining He, Wenbing Lin, and S.-H. Gary Chan. 2017. Indoor localization and automatic fingerprint update with altered AP signals. *IEEE Trans. Mob. Comput.* 16, 7 (July 2017), 1897–1910.

[18] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2008. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. Springer ECCV.* 304–317.

[19] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2010. Improving bag-of-features for large scale image search. *Int. J. Comp. Vision* 87, 3 (01 May 2010), 316–336.

[20] Junghyun Jun, Liang He, Yu Gu, Wenchao Jiang, Gaurav Kushwaha, V. A, Long Cheng, Cong Liu, and Ting Zhu. 2018. Low-overhead WiFi fingerprinting. *IEEE Trans. Mob. Comput.* 17, 3 (Mar 2018), 590–603.

[21] Jan Knopp, Josef Sivic, and Tomas Pajdla. 2010. Avoiding confusing features in place recognition. In *Proc. Springer ECCV.* 748–761.

[22] Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher. 2010. Location recognition using prioritized feature matching. In *Proc. Spinger ECCV.* 791–804.

[23] Sikun Lin, Hao Fei Cheng, Weikai Li, Zhangpeng Huang, Pan Hui, and Christoph Peylo. 2017. Ubii: Physical world interation through augmented reality. *IEEE Trans. Mob. Comput.* 16, 3 (Mar 2017), 872–885.

[24] Guoyu Lu, Nicu Sebe, Congfu Xu, and Chandra Kambhamettu. 2015. Memory efficient large-scale image-based localization. *Multimedia Tools Appl.* 74, 2 (01 Jan 2015), 479–503.

[25] Guoyu Lu, Yan Yan, Abhishek Kolagunda, and Chandra Kambhamettu. 2016. *A Fast 3D Indoor-localization Approach Based on Video Queries.* Springer International Publishing, Cham, 218–230.

[26] Guoyu Lu, Yan Yan, Li Ren, Jingkuan Song, Nicu Sebe, and Chandra Kambhamettu. 2015. Localize me anywhere, anytime: A multi-task point-retrieval approach. In *Proc. IEEE ICCV.* 2434–2442.

[27] Guoyu Lu, Yan Yan, Nicu Sebe, and Chandra Kambhamettu. 2014. Knowing where I am: Exploiting multi-task learning for multi-view indoor image-based localization. In *Proc. BMVA BMVC.* 1–12.

[28] 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 (Jan 2018), 141–154.

[29] Kevin McGuinness, Kealan McCusker, Neil O'Hare, and Noel E. O'Connor. 2012. Efficient storage and decoding of SURF feature points. In *Proc. Springer-Verlag MMM.* 440–451.

[30] Marius Muja and David G. Lowe. 2014. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 11 (Nov 2014), 2227–2240.

[31] Hassan Naseri and Visa Koivunen. 2017. Cooperative simultaneous localization and mapping by exploiting multipath propagation. *IEEE Trans. Signal Process.* 65, 1 (Jan 2017), 200–211.

[32] Dominik Van Opdenbosch, Georg Schroth, Robert Huitl, Sebastian Hilsenbeck, Adrian Garcea, and Eckehard Steinbach. 2014. Camera-based indoor positioning using scalable streaming of compressed binary image signatures. In *Proc. IEEE ICIP.* 2804–2808.

[33] Claudio Piciarelli. 2016. Visual indoor localization in known environments. *IEEE Signal Processing Letters* 23, 10 (Oct 2016), 1330–1334.

[34] Muhamad Risqi U. Saputra, Andrew Markham, and Niki Trigoni. 2018. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Comput. Surv.* 51, 2, Article 37 (Feb 2018), 36 pages.

[35] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. 2012. Improving image-based localization by active correspondence search. In *Proc. Springer ECCV.* 752–765.

[36] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. 2017. Efficient effective prioritized matching for large-scale image-based localization. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 9 (Sept 2017), 1744–1756.

[37] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. 2017. Are large-scale 3D models really necessary for accurate visual localization? In *Proc. IEEE CVPR.* 6175–6184.

[38] Meina Song, Zhonghong Ou, Eduardo Castellanos, Tuomas Ylipiha, Teemu Kämäräinen, Matti Siekkinen, Antti Ylä-Jääski, and Pan Hui. 2017. Exploring vision-based techniques for outdoor positioning systems: A feasibility study. *IEEE Trans. Mob. Comput.* 16, 12 (Dec 2017), 3361–3375.

[39] Zheng Sun, Shijia Pan, Yu-Chi Su, and Pei Zhang. 2013. Headio: Zero-configured heading acquisition for indoor mobile devices through multimodal context sensing. In *Proc. ACM UbiComp.* 33–42.

[40] Xiaoqiang Teng, Deke Guo, Yulan Guo, Xiaolei Zhou, Zeliu Ding, and Zhong Liu. 2017. IONavi: An indoor-outdoor navigation service via mobile crowdsensing. *ACM Trans. Sens. Netw.* 13, 2, Article 12 (Apr 2017), 28 pages.

[41] Xiaohua Tian, Zhenyu Song, Binyao Jiang, Yang Zhang, Tuo Yu, and Xinbing Wang. 2017. HiQuadLoc: A RSS fingerprinting based indoor localization system for quadrotors. *IEEE Trans. Mob. Comput.* 16, 9 (Sept 2017), 2545–2559.

[42] X. Tian, M. Wang, W. Li, B. Jiang, D. Xu, X. Wang, and J. Xu. 2018. Improve accuracy of fingerprinting localization with temporal correlation of the RSS. *IEEE Trans. Mob. Comput.* 17, 1 (Jan 2018), 113–126.

[43] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. 2015. Visual place recognition with repetitive structures. *IEEE Trans. on Pattern Anal. Mach. Intell.* 37, 11 (Nov 2015), 2346–2359.

[44] Panu Turcot and David G. Lowe. 2009. Better matching with fewer features: The selection of useful features in large database recognition problems. In *Proc. IEEE ICCV Workshop*. 2109–2116.

[45] A. Vedaldi and B. Fulkerson. 2008. VLFeat: An open and portable library of computer vision algorithms. Retrieved from http://www.vlfeat.org/.

[46] Jue Wang and Dina Katabi. 2013. Dude, where's my card?: RFID positioning that works with multipath and non-line of sight. In *Proc. ACM SIGCOMM*. 51–62.

[47] Hongkai Wen, Sen Wang, Ronnie Clark, Savvas Papaioannou, and Niki Trigoni. 2016. Poster: Efficient visual positioning with adaptive parameter learning. In *Proc. ACM/IEEE IPSN*. 1–2.

[48] Martin Werner, Moritz Kessel, and Chadly Marouane. 2011. Indoor positioning using smartphone camera. In *Proc. IEEE IPIN*. 1–6.

[49] Chenshu Wu, Jingao Xu, Zheng Yang, Nicholas D. Lane, and Zuwei Yin. 2017. Gain without pain: Accurate WiFi-based localization using fingerprint spatial gradient. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 2, Article 29 (June 2017), 19 pages.

[50] Chenshu Wu, Zheng Yang, and Chaowei Xiao. 2018. Automatic radio map adaptation for indoor localization using smartphones. *IEEE Trans. Mob. Comput.* 17, 3 (Mar 2018), 517–528.

[51] Liyao Xiang, Tzu-Yin Tai, Baochun Li, and Bo Li. 2017. Tack: Learning towards contextual and ephemeral indoor localization with crowdsourcing. *IEEE J. Sel. Areas Commun.* 35, 4 (Apr 2017), 863–879.

[52] Han Xu, Zheng Yang, Zimu Zhou, Longfei Shangguan, Ke Yi, and Yunhao Liu. 2015. Enhancing WiFi-based localization with visual clues. In *Proc. ACM UbiComp*. 963–974.

[53] Han Xu, Zheng Yang, Zimu Zhou, Longfei Shangguan, Ke Yi, and Yunhao Liu. 2016. Indoor localization via multimodal sensing on smartphones. In *Proc. ACM UbiComp*. 208–219.

[54] Yongtuo Zhang, Wen Hu, Weitao Xu, Hongkai Wen, and Chun Tung Chou. 2016. NaviGlass: Indoor localisation using smart glasses. In *Proc. Junction Publishing EWSN*. 205–216.

[55] Yuanqing Zheng, Guobin Shen, Liqun Li, Chunshui Zhao, Mo Li, and Feng Zhao. 2017. Travi-Navi: Self-deployable indoor navigation system. *IEEE/ACM Trans. Networking* 25, 5 (Oct 2017), 2655–2669.

Received March 2017; revised July 2018; accepted October 2018