

Mobile Augmented Reality Framework with Fusional Localization and Pose Estimation

Songlin HOU^{1,*†}, Fangzhou LIN^{2,3,†}, Yunmei HUANG^{4,†}, Zhe PENG⁵ and Bin XIAO⁵

¹*Department of Computer Science, Worcester Polytechnic Institute, Worcester 01609, U.S.A.*

²*Department of Robotics Engineering, Worcester Polytechnic Institute, Worcester 01609, U.S.A.*

³*Unprecedented-scale Data Analytics Center, Tohoku University, Sendai 980-8577, Japan*

⁴*Department of Forestry and Natural Resources, Purdue University, West Lafayette 47907, U.S.A.*

⁵*Department of Computing, The Hong Kong Polytechnic University, Hong Kong 999-077, China*

As a novel way of presenting information, augmented reality (AR) enables people to interact with the physical world in a direct and intuitive way. While there are some mobile AR products implemented with specific hardware at a high cost, the software approaches of AR implementation on mobile platforms (such as smartphones, tablet PC, etc.) are still far from practical use. GPS-based mobile AR systems usually perform poorly due to the inaccurate positioning in the indoor environment. Previous vision-based pose estimation methods need to continuously track predefined markers within a short distance, which greatly degrade user experience. This paper first conducts a comprehensive study of the state-of-the-art AR and localization systems on mobile platforms. Then, we propose an effective indoor mobile AR framework. In the framework, a fusional localization method and a new pose estimation implementation are developed to increase the overall matching rate and thus improving AR display accuracy. Experiments show that our framework has higher performance than approaches purely based on images or Wi-Fi signals. We achieve low average error distances (0.61–0.81m) and accurate matching rates (77–82%) when the average sampling grid length is set to 0.5m.

KEYWORDS: augmented reality, indoor localization, fusional localization, location based service

1. Introduction

Wireless techniques are extensively utilized in various mobile applications, significantly driving the rapid expansion of personal use services such as location-based services (LBS). As a subset of promising LBS applications, indoor mobile augmented reality (AR) has been attracting considerable attention in personal use, facilitated by advancements in the computational capabilities of mobile devices. While most existing AR applications on the market rely on vision tracking, inertial tracking, or GPS tracking [1], implementing indoor mobile AR is often considered challenging due to 1) the difficulty of achieving accurate user localization in indoor environments with minimal or no GPS signals [2] and 2) the limitations of sensor types available in standard mobile devices or specialized settings which are not common in most mobile phones, such as point cloud technologies [3]. Furthermore, while coarse-grained localization may suffice for outdoor environments, it is often inadequate for indoor scenarios, where the intricate structures and confined spaces of indoor environments pose additional challenges.

In indoor environments, a mobile device needs to offer the most probable position before providing augmented reality to users by predicting or matching based on sensor inputs. Thus, for an indoor mobile AR application, indoor localization and pose estimation are two key components in implementation.

1.1 Indoor localization techniques

Many indoor localization approaches [4–9] based on different mobile phone sensors have been proposed to facilitate indoor AR, such as wireless local area network (WLAN), Bluetooth, inertial sensors, geomagnetism, vision analysis or audible sound based systems. With most methods using vision analysis adopting different fundamental techniques from the others, we categorize indoor localization methods as non-vision-based methods and vision-based methods. As for the vision-based methods, we use image matching and image retrieval interchangeably with no discrimination. Several sensors used in indoor localization such as radio-frequency identification (RFID), infrared are not discussed in this article since the extra customized hardware is often needed other than mobile phones.

For indoor localization, non-vision-based approaches with inertia sensors and WLAN are most widely studied for its low cost both on users and infrastructure deployment. For inertia sensors [9, 10] like gyroscope and accelerator, relative

Received July 31, 2024; Accepted January 3, 2025

*Corresponding author. E-mail: shou@wpi.edu

†equally contribution as co-first authors



CC BY 4.0 (Attribution 4.0 International) License. This license permits others to distribute, remix, adapt, and build upon the Article, including for commercial purposes, provided that the original source and Authors are properly credited (<https://creativecommons.org/licenses/by/4.0/>).

movements are measured and accumulated in order to calculate user's moving trajectory and estimate the position. Localization methods [5, 8, 11] using WLAN are usually based on trilateration or fingerprinting. Trilateration-based methods [12] largely depend on the signal-strength-to-distance relationship with generally lower precision than methods using fingerprinting, while a large amount of sampling work is usually required by fingerprinting. The combination of WLAN and inertia sensors to improve precision can also be found in many studies [9]. With the booming of machine learning and deep learning, many learning-based methods are also proposed for this task. Sebetic *et al.* [13] provided a wireless channel-aware data augmentation method that can significantly improve localization accuracy with reduced measurement data. Some researchers [14, 15] also applied Long Short-Term Memory on wireless signals to enhance generalization and reliability.

On the other hand, vision-based indoor localization uses image pattern recognition and matching (or retrieval) to deal with AR-based indoors localization problem. The methods can be categorized into marker-based localization, semantics-based localization, and photo matching. Marker-based methods use pattern recognition to detect predefined markers for tracking and are now mostly used on robots [16]. Semantics-based localization [17] uses semantic descriptors as features to locate users. Applications in [18] demonstrate how to use Optical Character Recognition (OCR) as semantic features for localization and AR. A similar pedestrian localization system called OCRAPOSE II [18] also locate users using OCR in detecting numbers. Feature-based localization uses image features taken from the camera to retrieve possible images of indoor environments on servers. Works from [19] use features extracted from images to identify locations. Apart from that, learning-based methods become popular due to their high performance exhibited in image recognition. Works such as [20, 21] formalized indoor localization as an image classification task. Moreover, some works use techniques usually found in vision-related tasks to process signal data. For example, Saideep *et al.* [22] and Ayush *et al.* [23] use convolutional neural network (CNN) to classify images generated from WLAN signals. Huiqing *et al.* [24] adopt both CNN and wavelet transform to process WLAN data directly.

Apart from the two categories mainly discussed above, there are techniques that have been widely used to facilitate the task of indoor localization. Structure from Motion (SfM) reconstructs 3D environments from 2D images, which can be used as a navigable maps, setting the stage for localization. Apart from it, point cloud, which is usually generated with depth camera, is often used for map generation. Siddharth *et al.* [25] focus on the image generation with point cloud registration. Some deep learning-based methods [26, 27] can also be used to help recover defeats commonly found in point cloud during map construction. Xinying He *et al.* [28] proposed a framework without keypoint detection and achieved enhanced performance in texture-poor scenes. Besides, Simultaneous Localization and Mapping (SLAM) enables a system to determine its position within an unknown environment while simultaneously building a map of that environment. The SLAM system is usually implemented in a way in which features collected from multiple sensors (including vision and signal data) are fused, for example, Niraj Pudasaini *et al.* [29] designed a SLAM system (SPAQ-DL-SLAM) using neural networks to enhance efficiency and accuracy in robots in resource-constrained embedded platforms. Muhammad *et al.* [30] utilize a graph optimization algorithm combining YOLOv5 and Wi-Fi fingerprint sequence matching, achieving satisfactory accuracy in complex environments.

The performance of non-vision-based indoor localization methods varies. Methods based on inertia sensors are usually hampered by the inferior hardware quality. Trilateration with WLAN [12] can grow unsteady even in general occasions due to signal fluctuation. Fingerprinting is comparatively usable since it's less susceptible to signal changes. Vision-based indoor localization methods can provide good performance most of the time, but they share some restriction by some inevitable limitations. A proper illumination environment with minimal obstruction is needed, and the details of varieties of patterns and colors can strongly affect the precision. When these methods are used in motion, motion blur and shake of camera can degrade the precision too. With the boom of machine learning and deep learning, some models are also used for localization, like applying K nearest neighbors (KNN) [11, 31], support vector machine (SVM) [32] on RSSI in non-vision-based methods. Or using CNN [30, 33] to retrieve images to help locate users. These methods turn out to be more accurate in localization; however, a drastically larger amount of data is usually required to ensure the performance, thus making it limited in some scenarios. For image matching methods using CNN, a large computational cost should be considered in real-time localization since high delay and power consumption can easily occur. Besides, we noticed while the matching methods based on CNN work generally well in outdoor environments (where landmarks are visible), they tend to generalize poorly on environment images taken indoors, especially when they are taken in motion. Moreover, pure learning-based image matching does not account for the relative distance between sampling positions and localized positions, which can introduce additional localization errors. These disadvantages suggest that learning-based image matching alone might not be the best choice for indoor localization when high precision is required.

1.2 Pose estimation techniques

Pose estimation [34] is the key technique to determine the locations and orientations of virtual objects. Its implementation can also be either vision-based or non-vision-based. For most AR applications, pose estimations are vision-based. These approaches seek matches for the point correlations between the current views and the predefined patterns or 3D digital models and determine the ways of rendering in perspective modes. The non-vision-based

Table 1. Comparasions of several indoor mobile AR or localization systems.

Methods/Authors	Data Sources	Approaches
OCRAPOSE II [18]	Images	OCR
Yong Deng [14]	iBeacon (FP)	Deep Learning
Taira [37]	Images	Dense Matching
Abbas [38]	Wi-Fi (FP)	Deep Learning
Xie [31]	Wi-Fi (FP)	KNN
Chen [39]	Wi-Fi, PDR, Images	Kalman Filter
Chriki [32]	Wi-Fi	SVM
Yang [40]	Wi-Fi, Images	CNN, Dense Matching

implementations use some data sources other than visual information, such as users' localization, orientations detected using mobile phones' sensors, etc.

Vision-based implementations of pose estimation can be divided into Pose from Orthography and Scaling & Iteration (POSIT) [35], fiducial marker detection, and contour tracking. As a standard way to solve the PnP problem in pose estimation, POSIT estimates the position and the rotation of the camera relative to the target through 3D coordinates of feature points from the real model, and the corresponding 2D coordinates of image points from the perspective view. Planar fiducial markers use image patterns [9, 16] in the real world as a marker for detection and tracking, and usually realize pattern recognition and camera pose estimation simultaneously. Contour tracking uses the contours of a 3D digital model to match the models.

Vision-based pose estimation implementations share some common limitations: they can be strongly affected by the input image data. An image without proper focusing, proper illumination and occlusion can result in bad performance. Vision-based AR implementations are sensitive to distance since necessary information for recognition and tracking can only be captured when targets close to the camera. While in indoor LBS, objects to be augmented can be far from users. Traditional vision-based methods will be unsuitable for indoor uses. Non-vision-based pose estimation implementations [6, 9] have less disturbance than vision-based ones since they do not need any optical marker. However, they can also be vibration-prone and less visually appealing since the alignments and poses of visual objects are entirely dependent on the hardware qualities which are usually poor. Besides, GPS-based AR systems will fail to provide precise localization information in indoor environments.

To illustrate some classic cases, we compare the implementations of several indoor AR systems. The results are shown in Table 1.

1.3 Our contributions

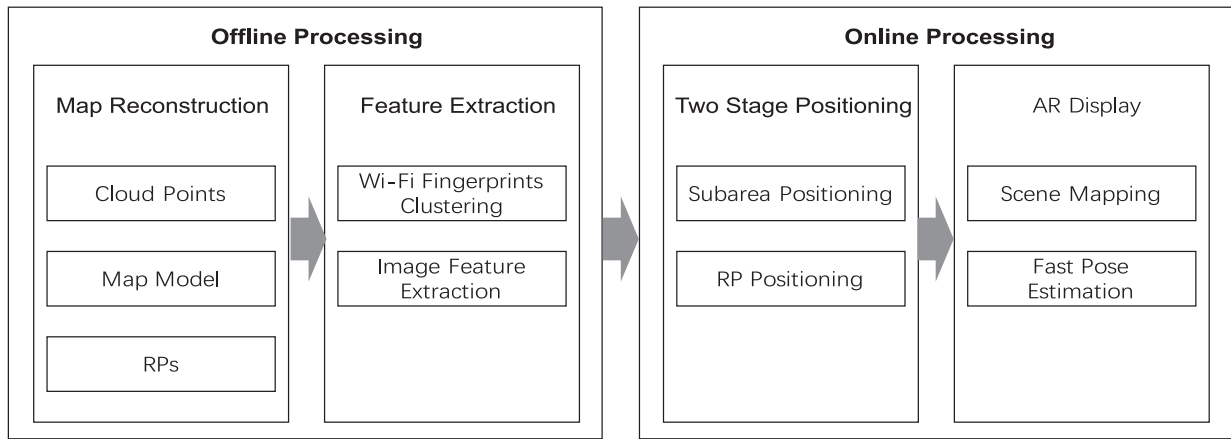
In order to mitigate the precision loss in current localization and pose estimation approaches and provide a more robust indoor AR implementation, we propose an indoor mobile AR framework which is deployed in android smartphones by 1) fusional localization with Wi-Fi and images as well as 2) a location-based pose estimation. In the fusional localization, distance ratio (DR) is introduced to compensate for the distance loss in image matching. We also give the implementation of location-based pose estimation based on relative Cartesian coordinate system for AR display. An android application using the framework is developed and tested in the experiments.

We conclude our contributions as follows:

- We present a comprehensive review of the two key techniques in indoor mobile AR implementation, indoor localization, and AR display.
- We explore and test the current vision-based indoor localization options and identify the common generalization issue widely present in deep learning methods (CNN) for indoor LBS applications.
- We propose a complete framework which fully supports indoor mobile AR with optimizing methods to improve image matching accuracy and pose estimation implementation for AR display.
- We evaluate and compare the performance of the proposed system with and without the methods proposed in the framework. The proposed method gives lower average error distance (0.61–0.81 meters) than methods only based on Wi-Fi (3.81–4.18 meters), and on SIFT [36] (1.29–1.44 meters). The reference point matching rate of the proposed method (77–82%) is much higher than methods only based on Wi-Fi (27–31%), and on SIFT (55–59%). Our method also shows a more stable performance and a higher generalization ability for CNN-based methods.

2. A New Indoor AR Framework

We propose a new indoor AR framework based on smartphone localization and pose estimation. The proposed framework provides real-time point-level AR display with accurate environment information. Users can get information about the surroundings as well as their current positions through AR information. Since the heavy



computing operations (Wi-Fi and image matching) are done on the server, and smartphones simply act as the role to collect and send data, so the computation cost on the smartphones can be treated acceptable for most scenarios.

2.1 Map reconstruction

To reconstruct a 3D indoor map, we capture the 3D structure data using a depth camera built on Google’s Tango device. Since the 3D reconstruction of a large environment can be time-consuming, we divide the whole environment into smaller areas and point clouds of these areas are collected in parallel. The point cloud of the whole environment can be assembled easily based on the overlaps of these sub-regional point clouds. The point cloud can be served as the 3D map directly, but in our case, we handcraft a 3D model using simple geometric meshes based on the point cloud to make it more user friendly. Annotated Reference Points (RPs) are marked evenly and of grid distribution with 0.5 average meters’ distance interval and confined in the walkable area in this map. The reconstruction process can be illustrated in Fig. 2.

The grid distance interval is a trade-off factor in our implementation. A smaller grid length can improve localization precision in most cases, however more labor-cost is required especially when the sampling environment is large. If the AR display precision is not critical, higher sampling distances (2.0–10.0m) can be used for distant AR display in spacious areas and lower sampling distances (0.5–2.0m) are only reserved for delicate near field AR display in smaller areas for labor saving.

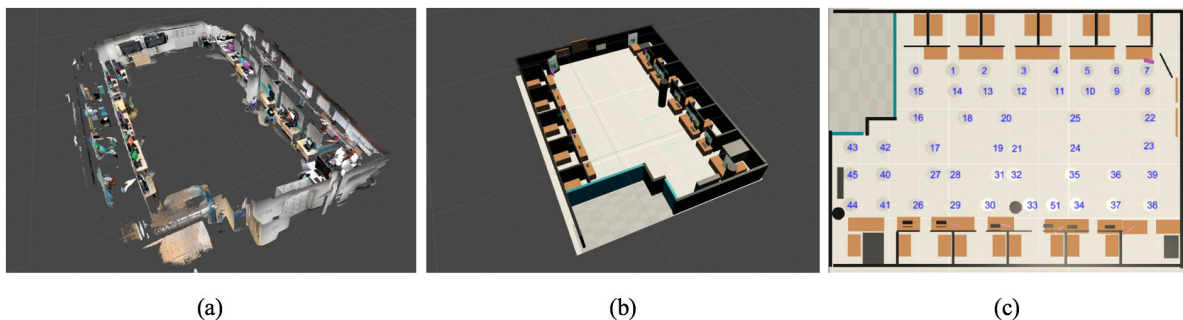


Fig. 2. (a) The original cloud point captured using Google’s Tango device. (b) The 3D model handcrafted based on the cloud point. (c) The top-view of the 3D map with RPs annotated.

2.2 Feature extraction

Based on the reconstructed 3D indoor map, we extract features from the data collected at every reference point (RP) in two steps: Wi-Fi fingerprints clustering and image feature extraction. At every RP, we collect many Wi-Fi fingerprints and images of surrounding with fixed focal length. The fingerprints and visual features derived from images are regarded as features used in this framework. A typical fingerprint of a RP takes the form of a set which contains signal strengths from different Access Points (APs). The fingerprint of RP b is defined as Eq. (2.1) ($FP(\cdot)$ is used to denote the fingerprint of a given RP). And AP_n^b indicates the signal strength of the n th AP collected at RP b .

$$FP(b) := \{AP_1^b, AP_2^b, AP_3^b, \dots, AP_n^b\} \quad (2.1)$$

Similarly, the visual features of RP b can be expressed as a set containing multiple key point collections of images taken at RP b . It takes the form of Eq. (2.2).

$$VF(b) := \{KP_1, KP_2, KP_3, \dots, KP_n\} \quad (2.2)$$

KP_i denotes the key point collection of the i th image taken at RP b , which is defined as Eq. (2.3). The j th element of the collection p_{tj} is a vector contains the pixel characteristics of the j th key point.

$$KP_i := \{p_{t1}, p_{t2}, p_{t3}, \dots, p_{tn}\} \quad (2.3)$$

In order to facilitate this process, we designed an Android application, which was installed on Samsung Galaxy S6, to collect fingerprints and images at the same time. At each RP, Received Signal Strength (RSS) of different APs and photos at different viewpoints are recorded. The phone camera used for collection has 16 MP resolution (f/1.9 aperture) and 1/2.6" sensor size. The original image file collected each has 2656×1494 pixels (16 : 9 aspect ratio), and was taken without Auto HDR mode and LED flash. At each RP, we collected 50 fingerprints and 50 images; although many of the photos lack salient visual features, they are captured to truthfully and clearly depict the environment.

Wi-Fi fingerprints clustering. When size of Wi-Fi fingerprints grows, matching process can soon become time-consuming. Clustering is generally used to reduce the number of fingerprint matchings and improve robustness against outliers. K-Medoids and DBSCAN are usually used to cluster fingerprints. We propose a two-step approach to cluster Wi-Fi fingerprints efficiently.

In the first step, for all the fingerprints collected at one RP, we need to find the centroid of these fingerprints. RL-clustering (combination of the authors names) [43] can be used to identify the central point. In our experiment, 45 fingerprints are selected to represent 45 RPs accordingly. Then, we apply RL-clustering again on these selected central fingerprints to find K subarea clusters. These K subarea clusters defines K subareas and we store these K central points on server. We also store the list of RPs which are related to each subarea cluster. In our experiment, K is set to 3.

Image feature extraction. Image features extraction is the process of getting numeric representation of prepared images. An image can contain a lot of features which can describe the content. Although human are good at differentiate various types of images, it becomes difficult for computers. Image features are commonly used for recognition. Since similar images tend to be close in features, we can retrieve similar images by comparing image features. Scale-Invariant Feature Transform (SIFT) is usually used in image retrieval for its generally higher accuracy. In SIFT, the operations on Difference of Gaussian (DoG) images can be approximated as the first order derivation in the scale space [44]. To generate more key points, we can calculate the n th order derivations, which takes the form of Eq. (2.4).

$$\frac{\partial^n L(x, y, \sigma)}{\partial \sigma^n} = 0 \quad (2.4)$$

In our experiment, key points of up to 4th order derivatives are calculated. We extract the key points from all the images from every RP and store them in the database. For the sake of brevity, we denote the SIFT key points with higher order derivatives as SIFT since they are equivalent in nature.

Challenges with CNN-based matching. CNN is widely used for image classification. The images features generated by CNN can also be used for image matching. However, there are several challenges in terms of accurate indoor positioning.

First, it is generally impractical to train a CNN for image classification (with ID of each RP as labels) because of extensive data required for training and the self-similarity nature of indoor images. Extant vision-based approaches using CNN generally requires extensive labor for data collection and labeling, which is not applicable when images taken at different positions are too similar or when no salient features are available.

Second, extracting features with CNN for matching requires lots of additional work. Using the machine-generated features can help find the image closest to the current captured photo, but the distance difference cannot be directly measured since these features are not related to pixel locations. Taira *et al.* [37] proposed a method to do pose estimation after matching images with CNN-based features, but it requires a lot more computation such as working with both low and high resolution of the query image during matching and requires multiple verification steps for pose

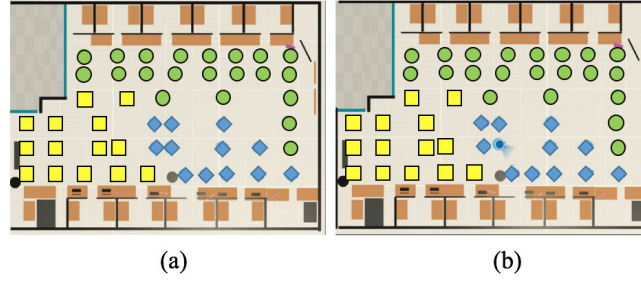


Fig. 3. Two steps in determining the user position. (a) subareas on maps. (b) user position.

estimation. The precision of map construction can also greatly influence the performance.

2.3 Two stage positioning

During the online processing, we first propose a new method to locate users from Wi-Fi fingerprints and images. This method includes two steps: 1) subarea positioning using Wi-Fi fingerprints matching, and 2) RP positioning using image matching. The combination of Wi-Fi fingerprints and image matching is able to determine the user's current position and orientation. Compared with methods solely depending on vision, our approaches can be more time-efficient since only a small number of images are to be compared after Wi-Fi fingerprints eliminating most impossible images before matching. It is also more robust since methods depending on solely vision analysis may fail when many photos taken at different places look similar. This process is shown in Fig. 3. In image (a), three subareas are created, each consisting of several RPs, which are distinguished by different colors and shapes. In image (b), the user is localized to be on one of the RP.

Subarea positioning. To determine which subarea the user is currently in, we compare the observed fingerprints captured by user's device with the premade K central points of subarea clusters. Various metrics can be chosen, like Euclidean distance, cosine similarity and Manhattan distance. In our experiment, we choose cosine similarity since it gives us least error rate.

RP positioning. Accurate positions can be further obtained through image matching (or retrieval) with photos taken at the RPs in this subarea when subarea positioning is done. In order to find the image most similar to the user's current scene photo, image feature extracted from the photo captured by the user's camera is compared with the image features stored in database. To calculate the similarity, the key points generated from 1–4th order derivatives are matched.

There have been a lot of key point based retrieval methods [44, 45], however, most are not directly applicable to the images with different shooting distances because they do not consider the distance diversity (which can be reflected in differences in scales and orientations, etc.) between matched images. Two images of the same environment taken at different distances can be of high similarity in these methods, but a huge loss in will occur when it is used in localization. Distance compensation is needed in this scenario as the offset for key point similarity and can leverage matching accuracy.

Distance compensation. In our system, we propose a new metric, i.e., distance ratio (DR), to implement distance compensation and assist matching similar images. The DR is used to measure the proximity of a photo relative to another of the same scene. Relative to photo p , $DR(q|p) > 1$ when objects in photo q are closer in physical space than these objects in p , while $DR(q|p) < 1$ means more distant. For a photo A captured from the user's camera, the distance ratio DR of the image B_i from m images ($B_1, B_2, \dots, B_m$) with top keypoint similarity relative to A is defined as follows:

$$DR(B_i|A) = \frac{1}{(N-1)N} \sum_{n=1}^N \sum_{j=1}^N \frac{|B_i^{(n)}, B_i^{(j)}|}{|A^{(n)}, A^{(j)}|}, \quad 1 \leq i \leq m \quad (2.5)$$

where

$$|P, Q| = \sqrt{(P_x - Q_x)^2 + (P_y - Q_y)^2}. \quad (2.6)$$

In (2.5), $B_i^{(n)}$ and $A^{(n)}$ are two key points in image B_i and A respectively belonging to the n th match between the two images, N is set to 10 in our experiment ($N > 1$), and the 10 matches with the shortest distance between key points are chosen. In (2.6), $|P, Q|$ function stands for the Euclidean distance between P and Q .

Then, we utilize the DR to compensate for the distance loss after finding the top m premade images with highest key point similarity. By measuring with DR , the image with DR closest to 1 chosen from the m images with highest similarity is selected as the most similar one, and the RP tagged with this image is returned as the current user's position.

2.4 AR display

After locating the users, we conduct the AR display of various indoor environment information in two steps: scene mapping and pose estimation. Compared with most vision-based methods, our method does not require users to put their phone camera within a near distance (0.5–2 meters on average). And it does not need to extract the patterns of observed objects (like 3D model or image key points) beforehand to enable AR tracking.

Scene mapping. Scene mapping is a process which maps the positions of the physical space to virtual space to enable location-based AR. The RP which can be regarded as the discretized position of the current user can be determined through two-step positioning.

The virtual space is presented by a 3D spatial coordinate system constructed from the 3D indoor map. By setting the center of the 3D indoor map as the origin, the RPs corresponding with the physical positions can be represented as coordinates in the virtual space (the directions of x and z axes representing right and forward while y axis representing vertical). There is a size ratio between the virtual space and physical space. For simplicity, we adjust the scale of coordinates to make the size ratio equal to one. Through two-stage positioning, the RP closest to users can be chosen, and the coordinates in virtual space can be determined.

After mapping the physical position to a position in virtual space, we insert virtual objects into the virtual space. The coordinates of the virtual objects can be easily computed within virtual space. In our experiment, we add 3D texts near the tables where different projects are placed. The contents of 3D texts are the names of the projects. Note that while only 3D texts are shown in the virtual space, any 3D objects can be easily integrated similarly like the 3D texts.

Pose estimation. In virtual space, the coordinate of the user (which is the coordinate of the closest RP) as well as the coordinates of all the virtual objects placed in can be determined, so the relative orientation and distance can also be computed. In order to show AR information based on the user's location, a normalized vector f representing the face direction is calculated through the built-in compass and accelerator. By adding a constant h representing the body height to the y coordinate of the RP, the coordinate of the eyes e of the current user is derived. Then coordinates of e and direction of vector f are used to describe the pose of the user's eyes.

When position and orientation of the user's eyes are determined, a pyramid representing the scope of view is calculated based on the pixel aspect ratio of the camera lens, vector f and the max observe distance beyond which virtual objects will not be displayed.

The scope of view is used in the next step to choose which virtual objects will be displayed, and the vector f is used to estimate the relative angle between the users. By setting up a local Cartesian coordinate system L with e being the origin and f being the axis with zero degree, the poses of virtual objects can be expressed in the Cartesian coordinate form. The pose information of the virtual objects in physical space can be obtained by mapping Cartesian coordinates relative to L back to physical space. This process can be illustrated in Fig. 4. Sub-figure (b) shows the virtual space. In (b), vector f representing the user's orientation is denoted as the blue arrow, and the scope of view is shown as the pyramid in white lines. The blue texts are virtual objects put in the virtual space, while are also rendered in the physical space with relative orientations and distances in (a).

3. Experiment Results

The experiment site is an exhibition hall. We used three types of smartphones in our experiments—Qihoo N4S, Huawei Mate 9 and Samsung Galaxy S6. Wi-Fi fingerprints are collected with these three phones randomly on all the

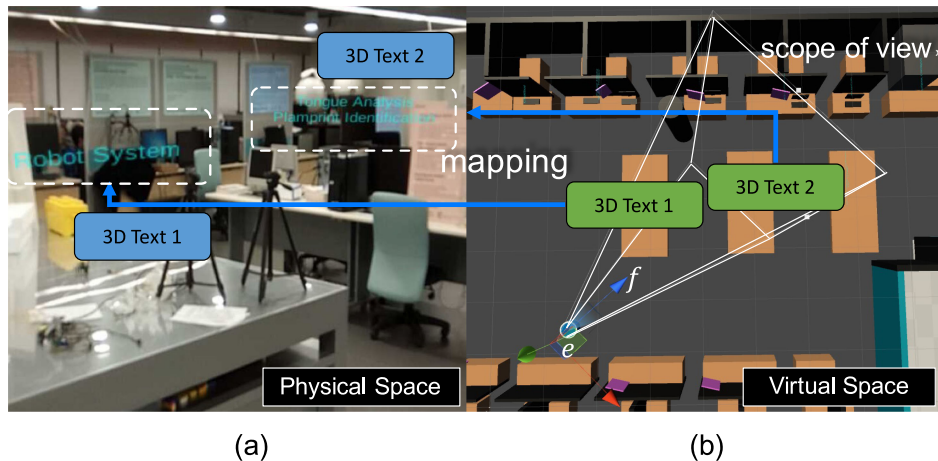


Fig. 4. The augmented view which implemented using the proposed method. (a) The view users see. (b) The corresponding pose of users in 3D virtual space.

45 RPs on different days in a week. Photos captured at each RP are taken using these three smartphones in the offline process, and the original photos are all compressed with resolution 800×600 before feature extraction. In the online process, the photos are automatically taken and uploaded to the server by an android app we developed for every 2 s, and different methods are used on the photos to evaluate. We compared our method with other three different methods, including only utilizing Wi-Fi fingerprints, only utilizing key point matching, and combining Wi-Fi fingerprints with key point matching. In order to avoid noise, the results are averaged on 300 samples collected in corresponding smartphones.

Since the alignment and trigger of virtual objects are both decided by the relative poses of these objects, so to these objects the pose of eyes relative to these virtual objects, which is also relative to the virtual space, is the only factor to affect the performance of the indoor AR in the proposed system. While the directional vector f is dependent on the built-in compass and accelerator, we use the localization precision to evaluate the indoor AR performance. In this experiment, RP matching rate and average distance error are used as metrics to evaluate and compare.

The result of RP matching rate is shown in the top diagram of Fig. 5. By evaluating the matching rate, we find indoor localization purely depending on Wi-Fi fingerprints is not accurate since Wi-Fi signals can be affected by a large amount of factors and RSS can be distinct for different kinds of smartphones. Image matching using SIFT is much better, and these three phones have similar matching rates. Combining these two methods, the matching rates increase for all three smartphones. It is a good evidence that through clustering, some incorrect images are filtered out before matching. The integration of distance compensation largely improves the matching rates, which shows the distance compensation makes closer images in a group of similar images more identifiable.

The result of average distance error is shown in bottom diagram of Fig. 5. For a mismatch, the predicted RP is relatively better when it is closer to the correct RP. So we also evaluate the precision of the proposed algorithm by calculating the average distance between the predicted RPs and the correct RPs, since matching rate alone cannot reveal the distance precision. The average error in distance is also measured under different conditions. The average error in distance of each method is shown on the top of each corresponding bar. By analyzing the distance error, it can be shown that the precision is within 1 meter, and most of the time (over 80%) the correct RP can be chosen. It is feasible to be used to integrate with the location based system proposed in this article.

We utilized CNN as a feature extractor to replace SIFT during the RP positioning step. Features derived from ResNet-152 [46], DenseNet-121 [47], and ConvNeXt-V2 [48] were compared with those in the database. Due to the nature of the generated features, distance compensation was omitted for these methods. DenseNet-121 achieved the best result with an average error distance of approximately 1.3, while the other two methods exhibited average error distances exceeding 1.6, performing less effectively than our proposed approach. This experiment demonstrates that while the proposed method, based on SIFT and distance compensation, is simpler in nature, it achieves superior precision.

To test the actual performance of the indoor AR system, the tester using the android app implemented with the proposed method in the exhibition hall. The screenshots of the AR application on Qihoo N4S are shown on the first row of Fig. 6. The blue augmented 3D texts are overlaid before the exhibition booths on the real scenes, which indicate

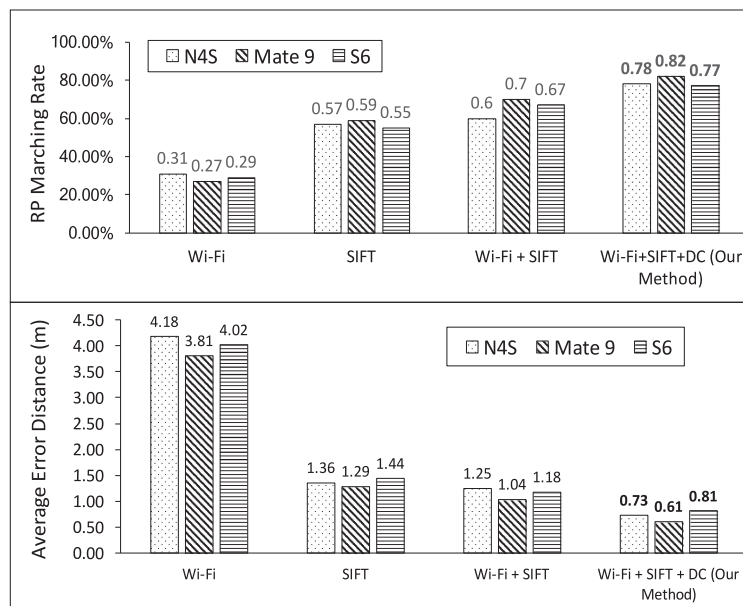


Fig. 5. Performance of localization accuracy (RP matching rates and average error distances of 3 phones using different methods).

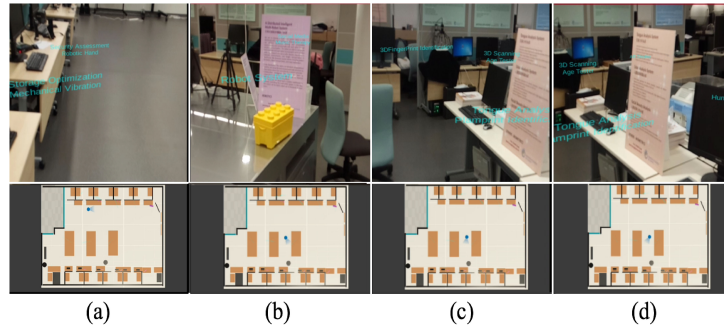


Fig. 6. The screenshots of the android app with proposed indoor AR implementation. **Row 1:** screenshots of the augmented scenes with virtual texts. **Row 2:** the positions and orientations of the users corresponding to the Row 1 view.

the project names of booth in front of the tester. The second row of Fig. 6 displays the tester's current position and orientation in the hall. The 2D projection of the 3D map now serves as the indoor map.

4. Conclusion

In the increasing pursuit of indoor localization precision and public fever on AR, we believe indoor AR techniques will become even more widely demanded than ever. Although there are many types of implementations in both indoor localization and AR and some combination with these two, we assume the most promising implementations should have these features. First, they should be low in cost, both for deployment and maintenance. Second, they should be robust, resistant to ambient interference. And they can be easily fitted in various scenarios as a service. Unlike most implementations using extra hardware like iBeacons and RFID, the indoor AR framework proposed in this paper does not need any extra hardware other than users' ordinary smartphones. And compared with most systems using pure RSS, the camera as well as the built-in sensors like compass and accelerator is used to reduce occasional error in our framework. To further make the whole framework more accurate, image matching using SIFT is further optimized with distance compensation. To solve the limitations on images in vision-based AR, such as intrusive markers and limited observing distance, a new implementation based on relative Cartesian coordinates is also proposed.

Although the performance of this framework is strongly related to the value of sampling grid intervals, the selections of interval values can be flexible and customized in different small areas based on size of environments and requirements of AR objects. The major online operations, such as subarea positioning and RP positioning with distance compensation, are processed on servers without heavy computation cost (like CNN). And these methods all support parallel computation on several cores or slave servers (nodes) since the implementation only contains simple iterations without inter-dependencies. So, our framework can meet the requirements of most weak real-time scenarios while remaining scalable in large environments.

Acknowledgments

We would like to express our gratitude to WPI, Tohoku University and Purdue University, which provided the facilities and support necessary for the successful completion of this research. Special thanks to Haichong (Kai) Zhang, Ziming Zhang, Yamada K.D, and Emma Ge for their invaluable guidance, insightful comments, and constructive feedback throughout the development of this project.

We also acknowledge the contributions of our experiment participants, whose involvement was crucial in validating our method. Additionally, we appreciate the support of our families and friends for their encouragement and understanding during the research process.

Finally, we would like to extend our gratitude to the anonymous reviewers for their time and effort in reviewing this manuscript.

REFERENCES

- [1] Billinghurst, M., Clark, A., Lee, G., *et al.*, "A survey of augmented reality," *Foundations and Trends® in Human-Computer Interaction*, **8(2-3)**: 73-272 (2015).
- [2] Gladston, A., and Duraisamy, A., "Augmented reality indoor navigation using handheld devices," *International Journal of Virtual and Augmented Reality*, **3(1)**: 1-17 (2019).
- [3] Patel, V. N., and Grewal, D. R., "Augmented reality based indoor navigation using point cloud localization," *International Journal of Engineering Applied Sciences and Technology* (2021).
- [4] Liang, J. Z., Corso, N., Turner, E., and Zakhori, A., "Image based localization in indoor environments," *2013 IEEE Fourth International Conference on Computing for Geospatial Research and Application (COM. Geo)*, pp. 70-75 (2013).
- [5] Jiang, Y., Pan, X., Li, K., Lv, Q., Dick, R. P., Hannigan, M., and Shang, L., "Ariel: Automatic wi-fi based room fingerprinting

- for indoor localization,” *ACM Conference on Ubiquitous Computing*, pp. 441–450 (2012).
- [6] Shi, D., Liu, F., Yutian, Q., and Ji, Y., “A wlan-based positioning system for indoor augmented reality services,” *International Conference on Information Science, Electronics and Electrical Engineering*, pp. 420–424 (2014).
 - [7] Zhang, C., Subbu, K. P., Luo, J., and Wu, J., “Groping: Geomagnetism and crowdsensing powered indoor navigation,” *Mobile Computing IEEE Transactions*, **14**(2): 387–400 (2015).
 - [8] Niu, J., Wang, B., Cheng, L., and Rodrigues, J. J., “Wicloc: An indoor localization system based on wifi fingerprints and crowdsourcing,” *2015 IEEE International Conference on Communications (ICC)*, pp. 3008–3013 (2015).
 - [9] Ahn, J., and Han, R., “An indoor augmented-reality evacuation system for the smartphone using personalized pedometry,” *Human-centric Computing and Information Sciences*, **2**(1): 1–23 (2012).
 - [10] Sun, J., Sun, W., Chen, Z.-H., Zhang, X., and Tang, C., “A large indoor localization system based on wifi and smartphone inertial sensors,” *42nd Chinese Control Conference*, pp. 3583–3588 (2023).
 - [11] Shi, D., Liu, F., Yutian, Q., and Ji, Y., “A wlan-based positioning system for indoor augmented reality services,” *International Conference on Information Science, Electronics and Electrical Engineering*, pp. 420–424 (2014).
 - [12] Cheng, V. C., Hao, L., Ng, J. K., and Cheung, W. K., “Fast setup and robust wifi localization for the exhibition industry,” *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pp. 472–479 (2018).
 - [13] Serbetci, O. G., Burghal, D., and Molisch, A. F., “Wireless channel aware data augmentation methods for deep learning-based indoor localization,” *IEEE Transactions on Wireless Communications* (2025).
 - [14] Deng, Y., and Li, C., “Research on indoor localization method based on deep learning and wireless sensing,” *Academic Journal of Science and Technology*, **11**(3): 97–101 (2024).
 - [15] de Aguiar, R. A., Paulino, N., and Pessoa, L. M., “A deep learning approach in ris-based indoor localization,” *Joint European Conference on Networks and Communications & 6G Summit*, pp. 523–528 (2024).
 - [16] López-Cerón, A., and Canas, J. M., “Accuracy analysis of marker-based 3d visual localization,” *XXXVII Jornadas de Automática, Madrid* (2016).
 - [17] Chen, Z., Chen, Y., Wang, S., and Zhao, Z., “A supervised learning based semantic location extraction method using mobile phone data,” *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, Vol. 3, pp. 548–551 (2012).
 - [18] Sadeghi, H., Valaee, S., and Shirani, S., “Ocrapose ii: An ocr-based indoor positioning system using mobile phone images,” *arXiv preprint arXiv:1704.05591* (2017).
 - [19] Lin, X., “Research on indoor target positioning system based on image feature extraction and recognition,” *2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, pp. 547–550 (2023).
 - [20] Higa, G. T. H., Monzani, R. S., da Silva Cecatto, J. F., de Souza, M. F. B. M., Weber, V., Pistori, H., and Matsubara, E. T., “Smartphone region-wise image indoor localization using deep learning for indoor tourist attraction,” *PLOS ONE*, **19**(9): e0307569–e0307569 (2024).
 - [21] Chen, D., Cui, L., Qin, L., and Feng-ying, W., “Gan and semi-supervised learning-based indoor visible light localization system for image sensors,” *International Conference on Remote Sensing, Mapping, and Image Processing (RSMIP 2024)*, p. 131670 (2024).
 - [22] Tiku, S., Mittal, A., and Pasricha, S., Convolutional neural networks for efficient indoor navigation with smartphones, *Embedded Machine Learning for Cyber-Physical, IoT, and Edge Computing*, pp. 3–20, Springer (2023).
 - [23] Mittal, A. K., Tiku, S., and Pasricha, S., “Adapting convolutional neural networks for indoor localization with smart mobile devices,” *GLSVLSI '18: Proceedings of the 2018 Great Lakes Symposium on VLSI*, pp. 117–122 (2018).
 - [24] Zhang, H., Shen, K., and Sun, H., “Indoor localization system based on wifi rssi using wavelet-cnn architecture,” *2023 China Automation Congress (CAC)*, pp. 2305–2309 (2023).
 - [25] Chauhan, S., Jayakumar, T., Mishra, D., and Ramiya, A. M., “1. Localization in mr-based indoor navigation system using point cloud registration,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (2024).
 - [26] Lin, F., Yue, Y., Hou, S., Yu, X., Xu, Y., Yamada, K. D., and Zhang, Z., “Hyperbolic chamfer distance for point cloud completion,” *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14595–14606 (2023).
 - [27] Lin, F., Yue, Y., Zhang, Z., Hou, S., Yamada, K., Kolachalama, V., and Saligrama, V., “Infocd: A contrastive chamfer distance loss for point cloud completion,” *Advances in Neural Information Processing Systems*, vol. 36 (2024).
 - [28] He, X., Sun, J., Wang, Y., Peng, S., Huang, Q., Bao, H., and Zhou, X., “Detector-free structure from motion,” *arXiv.org*, Vol. abs/2306.15669 (2023).
 - [29] Pudasaini, N., Hanif, M. A., and Shafique, M., “Spaql-dl-slam: Towards optimizing deep learning-based slam for resource-constrained embedded platforms,” *2024 18th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (2024).
 - [30] Shoukat, M. U., Yan, L., Deng, D., Imtiaz, M., Safdar, M., and Nawaz, S. A., “Cognitive robotics: Deep learning approaches for trajectory and motion control in complex environment,” *Advanced Engineering Informatics*, **60**: 102370 (2024).
 - [31] Xie, Y., Wang, Y., Nallanathan, A., and Wang, L., “An improved k-nearest-neighbor indoor localization method based on spearman distance,” *IEEE Signal Process. Lett.*, **23**(3): 351–355 (2016).
 - [32] Chriki, A., Touati, H., and Snoussi, H., “Svm-based indoor localization in wireless sensor networks,” *2017 IEEE 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1144–1149 (2017).
 - [33] Chen, Y., Chen, R., Liu, M., Xiao, A., Wu, D., and Zhao, S., “Indoor visual positioning aided by cnn-based image retrieval: Training-free, 3d modeling-free,” *Sensors*, **18**(8): 2692 (2018).
 - [34] Marchand, E., Uchiyama, H., and Spindler, F., “Pose estimation for augmented reality: A hands-on survey,” *IEEE Transactions on Visualization and Computer Graphics*, **22**(12): 2633–2651 (2016).
 - [35] DeMenthon, D. F., and Davis, L. S., “Model-based object pose in 25 lines of code,” *International Journal of Computer Vision*,

- 15:** 123–141 (1995).
- [36] Lowe, D. G., “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, **60**: 91–110 (2004).
 - [37] Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., and Torii, A., “Inloc: Indoor visual localization with dense matching and view synthesis,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7199–7209 (2018).
 - [38] Abbas, M., Elhamshary, M., Rizk, H., Torki, M., and Youssef, M., “Wideep: Wifi-based accurate and robust indoor localization system using deep learning,” *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10 (2019).
 - [39] Chen, Z., Zou, H., Jiang, H., Zhu, Q., Soh, Y. C., and Xie, L., “Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization,” *Sensors*, **15**(1): 715–732 (2015).
 - [40] Yang, Y., Toth, C. K., and Brzezinska, D., “A 3d map aided deep learning based indoor localization system for smart devices,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **XLIII-B4-2020**: 391–397 (2020).
 - [41] Peng, Z., Gao, S., Xiao, B., Wei, G., Guo, S., and Yang, Y., “Indoor floor plan construction through sensing data collected from smartphones,” *IEEE Internet of Things Journal* (2018).
 - [42] Peng, Z., Gao, S., Xiao, B., Guo, S., and Yang, Y., “Crowdgis: Updating digital maps via mobile crowdsensing,” *IEEE Transactions on Automation Science and Engineering*, **15**(1): 369–380 (2018).
 - [43] Rodriguez, A., and Laio, A., “Clustering by fast search and find of density peaks,” *Science*, **344**(6191): 1492 (2014).
 - [44] Park, U., Park, J., and Jain, A. K., “Robust keypoint detection using higher-order scale space derivatives: Application to image retrieval,” *IEEE Signal Processing Letters*, **21**(8): 962–965 (2014).
 - [45] Zheng, L., Yang, Y., and Tian, Q., “Sift meets cnn: A decade survey of instance retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**(5): 1224–1244 (2018).
 - [46] He, K., Zhang, X., Ren, S., and Sun, J., “Deep residual learning for image recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016).
 - [47] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q., “Densely connected convolutional networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708 (2017).
 - [48] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S., “A convnet for the 2020s,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986 (2022).