Robust and Effort-Efficient Image-Based Indoor Localization With Generative Features

Zhenhan Zhu, Yanchao Zhao, *Member, IEEE*, Maoxing Tang, Yanling Bu, *Member, IEEE*, and Hao Han, *Member, IEEE*

***Abstract*—Image-based indoor localization using smartphones has become popular, leveraging visual landmarks and fingerprint extraction for localization. Fingerprint density significantly affects accuracy, but collecting dense, high-resolution fingerprints during on-site surveys is labor-intensive and incurs high computation/storage costs during matching. Additionally, efficient fingerprint extraction often constrains users to specific shooting poses, with deviations markedly reducing localization accuracy. To address these challenges, we introduce ARGILS, an Automated Real-time Generative Image Localization System. The key idea is to use cross sparse sampling instead of dense sampling, generate fingerprint features for missing locations, and quickly match locations through feature orthogonal decomposition. Cross sparse sampling ensures full coverage of scene features and helps to generate missing fingerprints. To maintain high localization resolution with sparse sampling, we designed a distance-constrained generative adversarial network to generate fingerprints for unsampled locations. Additionally, we developed an orthogonal fingerprint extraction method to decompose image features into horizontal and vertical directions in 2D space. To improve robustness against obstacles, we implemented a scanning localization scheme using key frame filtering and clustering. We have implemented ARGILS and performed extensive real-world evaluations. Experiment results show that when reducing 75% site survey effort, the average location error of ARGILS is around 2.5m in a shopping mall, 48% higher than state-of-the-art methods. ARGILS can also efficiently speed up localization process, with the time consumption ranging from 0.1 to 0.3 seconds on smartphones of various configurations.**

***Index Terms*—Image-based localization, generative adversarial networks, mobile device.**

I. Introduction

A. Motivation

INDOOR localization is a critical component of location-based services, which can greatly facilitate people's daily indoor activities [1], [2], such as smart home [3], indoor navigation [4], [5] and so on. With the advent of 5G technology, localization capabilities are expected to advance even further [6], [7]. In large shopping malls, for example, visitors often have difficulty finding their destination, leading to frustration and disorientation. Indoor localization provides a solution to this problem. The advent of smartphones has fueled the demand for accurate indoor localization in large open spaces like shopping malls, libraries, and theaters. Various wireless signal-based technologies, including WiFi [8], Bluetooth [9], [10], RFID [11] and etc., have been used to explore effective indoor localization. However, these methods usually rely on pre-installed infrastructure and suffer from multi-path propagation, device orientation, and signal reflection. Therefore, the localization accuracy is vulnerable to complex indoor environments when the single signal feature is used as fingerprint [12]. On the contrary, vision-based indoor localization has emerged as a promising technology that eliminates the need for complex infrastructure pre-installation, which makes it more practical and scalable. It solely relies on unique, feature-rich images captured by smart devices to deliver robust, powerful, and cost-effective indoor localization solutions. Unlike other image-based tasks, the lighting conditions of indoor scenes tend to be relatively stable so they do not have much impact. Based on these advantages, we hope to use a simple-to-deploy localization system to quickly locate photos taken by users while ensuring a user-friendly experience.

B. Limitation of Prior Art

Vision-based indoor localization methods can generally be divided into two types: model-based and retrieval-based. Model-based methods usually use structure from motion to reconstruct the 3D scene to locate the user. However, a point cloud model of a scene may contain millions of 3D points [13], [14], [15], which will generate high storage consumption and greatly increase the time complexity of model inference.

Retrieval-based approaches have been widely investigated to achieve accurate and lightweight localization [16] for mobile terminals with limited resources. It typically involves three steps: site survey, image retrieval, and location estimation. The site

Received 15 May 2024; revised 4 February 2025; accepted 7 February 2025. Date of publication 11 February 2025; date of current version 5 June 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62172215 and Grant 62402217, in part by the A3 Foresight Program of NSFC under Grant 62061146002, in part by the Jiangsu Natural Science Foundation under Grant BK20241377, and in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization. Recommended for acceptance by E. Ngai. *(Corresponding authors: Yanchao Zhao; Yanling Bu.)*

Zhenhan Zhu, Yanchao Zhao, Maoxing Tang, and Hao Han are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China (e-mail: zhuzhenhan@nuaa.edu.cn; yczhao@nuaa.edu.cn; tangmaoxing@nuaa.edu.cn; hhan@nuaa.edu.cn).

Yanling Bu is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China, and also with the State KeyLab. for Novel Software Technology, Nanjing University, Nanjing 210093, China (e-mail: byling@nuaa.edu.cn).

Digital Object Identifier 10.1109/TMC.2025.3541045

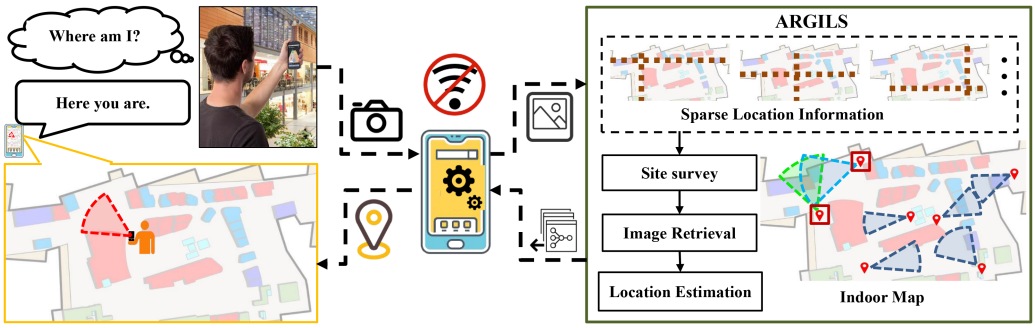


Fig. 1. Background of image-based indoor localization. How to use sparse database location to estimate user location?.

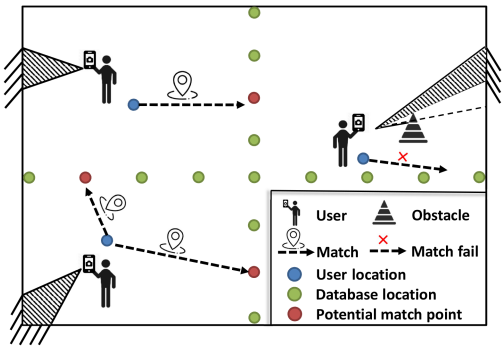


Fig. 2. Challenges of image-based localization using sparsely collected databases.

survey is an offline operation process, which requires environmental image collection at labeled locations. This process is inherently labor-intensive and tedious as the density of sampling in space directly affects the final localization accuracy of the system. This greatly affects the deployment efficiency and cost of the localization system. Some studies [17], [18] use landmarks with unique characteristics, such as trademarks, posters, etc., for localization, but this also requires extremely complex detailed investigation of the scene. Furthermore, this requires users to choose a specific angle to ensure that specific landmarks are photographed, which greatly affects the user experience. To address these limitations, as illustrated in Fig. 1, we explore the question of *whether robust and accurate indoor image-based localization can be achieved with sparsely collected image databases without relying on specific landmarks*.

C. Proposed Approach

To solve the problem mentioned above, we present ARGILS, an **a**utomated **r**eal-time **g**enerative **i**mage **l**ocalization **s**ystem for robust, low-overhead, and user-friendly indoor localization solution. ARGILS greatly reduces the workload of site surveys by cross-sparse image sampling. The feature vectors extracted from images using the EfficientNet(b_0) [19] neural network without the classification layer serve as fingerprint features. It then uses a distance-constrained generative adversarial network(DistanceGAN) to generate features at missing locations, which can improve the spatial resolution of the fingerprint feature database in an effort-efficient manner. ARGILS exploits the similarity between feature segments to decompose features both horizontally and vertically, thereby facilitating the evaluation of feature offsets to complete the position estimate. Notably, the terms "vertical" and "horizontal" are defined within the context of the two-dimensional plane of the scene. To ensure the robustness of localization in real scenarios, we also propose a scanning localization algorithm to minimize the impact of local obstacles and errors caused by single-frame localization. By obviating the necessity for specific landmarks with strict constraints, such as logos, our system showcases practical utility across diverse indoor environments.

D. Technical Challenges and Solutions

Basically, we are facing three major challenges as shown in Fig. 2 when using sparsely collected databases for localization:

- *Missing Location Features:* Using sparse acquisition methods can reduce the workload but may result in missing features at various positions, decreasing localization accuracy. To address this, we propose generating missing features to compensate for the limitations of sparse acquisition. However, constraining the positional relationship between generated and sampled features is challenging, making it difficult to directly generate corresponding features based on distance changes.
- *Missing Viewing Angle Features:* Different viewing angles at the same location result in various viewing angle features. Locating under a missing viewing angle may lead to multiple matching results, making it difficult to determine the relationship between existing angles and synthesize them to generate view features.
- *Complex and Variable Obstacles:* Dynamic obstacles often appear in indoor scenes. When the user's shooting angle is blocked by an obstacle, incorrect matching can occur, leading to unacceptable errors.

To address these challenges, the main solutions and contributions of ARGILS are summarized as follows:

- *Distance-Constraint Generative Model for Feature Generation to Achieve Effort-Efficient Site Survey:* We develop a distance-constraint generative adversarial network to generate features of missing locations. This model uses distance constraints to generate features at varying distances from the same viewing angle, enabling the creation of

features for front or rear locations. This can effectively improve the spatial resolution of the fingerprint feature database to improve localization accuracy.

- *An Orthogonal Feature Decomposition and Localization Framework for Addressing Missing Views to Achieve Robust Localization:* This framework includes a feature projection algorithm based on feature similarity, effectively projecting features to the target perspective. By decomposing features vertically and horizontally, we optimize the two dimensions of coordinates simultaneously. This method effectively addresses the issue of missing viewing angle features, as it can accommodate any shooting angle of the user.
- *Scanning Localization Algorithm for Addressing Variable Obstacles to Achieve Robust and Accurate Localization:* We implemented a scanning localization algorithm that allows users to scan the scene. It automatically selects the best frame from the video for localization and uses clustering to refine the user's position. This approach reduces errors from mismatches and dynamic obstacles, improving system robustness.

E. Summary of Experimental Results

We have implemented and tested the performance of ARGILS using mobile devices in shopping malls, office buildings, and laboratory buildings. ARGILS can achieve an average localization accuracy of 1.6 m in scenes with clear and simple features such as a lobby. Even in a shopping mall environment with complex lighting conditions and complex random obstacles such as pedestrians, the average error is 2.5 m, which is 48% lower than the current state-of-the-art method. More importantly, ARGILS only utilizes 25% of the original database collection density, significantly reducing the workload of site survey collection. And it can ensure a stable localization effect under different lighting environments. At the same time, the single-frame localization time of our method does not exceed 400 ms on mobile devices with different configurations. On an ordinary OnePlus8 phone, the single localization time is 105 ms, while on the poorly configured 360N6 Pro, a single localization can also be completed within 332 ms.

II. Related Works

A. Wireless Indoor Localization

With the emergence of the Internet of Things(IoT), localization within indoor environments like supermarkets, airports, train stations, and hospitals becomes inevitable. Indoor localization using wireless technologies such as WiFi, 5G, Bluetooth low energy(BLE), Zigbee, and radio-frequency identification(RFID) has seen significant advancement, offering good accuracy in various indoor environments. WiFi-based [20], [21] localization technology stands out due to its high popularity and has become the most widely used solution. [22] proves that KNN is the most accurate localization technique as well as the most precise. [6], [7], [23], [24] use 5G new radio (NR) signals to achieve indoor localization. Zigbee [25] is used for long-distance transmission which has low cost, low data transfer rate, and short latency time, compared to WiFi standards. At the same time, many studies [26], [27] use BLE technology to solve indoor localization problems.

However, these methods often require additional infrastructure setup, which can be costly and complex. Moreover, moving objects, human interference, or architectural modifications can affect signal propagation, leading to fluctuations in localization accuracy over time. This instability is particularly problematic in scenarios requiring real-time tracking or precise location-based services.

B. Image-Based Indoor Localization

Image-based indoor localization has two primary implementation types: model-based methods and image fingerprint retrieval-based methods. Model-based methods use structure from motion(SFM) or simultaneous localization and mapping(SLAM) [28] technology to construct indoor 3D models and perform 2D-3D mapping for localization [15], [29], [30], [31]. The characteristics of high storage and high computing performance make these methods unsuitable for deployment on mobile platforms. The image fingerprint retrieval-based method only requires storing 2D image features and performing simple vector operations for localization [32], [33]. Retrieval-based method [34], [35] offers better efficiency and is more feasible for real-time applications on mobile devices. Gao et al. [17] proposed a lightweight site survey method that can automatically mark image coordinates.

The image-based indoor localization method can also be categorized into two approaches: landmark location labeling [18], [36] and camera location labeling [37], [38]. The landmark location labeling approach is obviously unstable and not robust in scenes with sparse landmarks. In contrast, the camera location labeling method uses the camera's location as image coordinates. This approach is primarily influenced by the number of pictures taken, which increases the workload of site surveys to achieve higher localization accuracy. However, these two methods are naturally difficult to cope with missing angle labels and dynamic obstacles such as pedestrians. Our proposed method ARGILS is based on the camera location labeling approach.

C. Feature Generation

Generative algorithms, like GAN proposed by Goodfellow et al. [39], have found applications in various domains, such as image inpainting [40], super-resolution [41], and image synthesis [42]. Conditional GANs, introduced by Mirza et al. [43], enhance interpretability by allowing conditional value input to produce more accurate results. Stable Diffusion [44] has made important contributions to the development of image generation.

Different from image generate tasks, the information of feature generation is relatively limited, but the generated result is more abstract, and the whole process is similar to the middle part of image generation. Feature generation tasks combine aspects of image inpainting and super-resolution. Generating foreground features resembles super-resolution while generating background features resembles image inpainting and extends

to edge features. Pathak et al. [45] proposed context-encoder can train the model by using L_2loss and $adversarialloss$ to form a joint loss, so that the training process can be carried out under supervision, and has become the backbone of many subsequent image inpainting work. Isola et al. [46] proposed pix2pix, the basis of the super-resolution algorithm pix2pixHD [47]. It trains the model through L_1loss and adversarial loss and is the benchmark for many image conversion methods.

For our DistanceGAN, ARGILS, we employ an encoder-decoder network structure. The triplet loss by Hermans et al. [48] ensures that the distance between negative samples is greater than that of positive samples while considering their correlation. ARGILS employs a modified version of the triplet loss, augmented by an L_2 loss, to generate analogous features that facilitate precise retrieval.

III. Preliminary Analysis

In this section, we first introduce the basic image localization method, which is used as a baseline for comparison with our method. Then we use preliminary analysis to verify the impact of the location density and angle number on the localization accuracy and explore the distance information hidden in the image features to provide theoretical support for the generated features. We explored the regularity of features under different view angles to inspire us to decompose the features for better utilization. Finally, we compare the impact of obstacles on localization results to verify that random obstacles in a single frame image will affect the localization effect.

A. Basic Image-Based Localization Method

The basic method relies on camera location labeling and involves two main steps. First, during the site survey, workers capture images of the scene and record the camera's position as the image's location. After extracting features, these images and location tags form a feature database. Second, in online localization, the user captures an image, and the best-matching image is retrieved from the database, using its location tag as the baseline localization result.

In the process of image retrieval, feature extraction and feature matching are required. To meet mobile deployment needs, we use the lightweight EfficientNet(b_0) model [19], pre-trained on the ImageNet dataset [49]. This model delivers strong feature extraction performance suitable for most indoor scenes, with the highest similarity result providing the final localization output.

The basic method has minimal application requirements, relying only on image retrieval without additional position calculations. When the image database is dense, it achieves high localization accuracy. We aim to optimize this method for reliable localization even with sparse image databases.

B. Impacts of Image Database Density

The localization accuracy of the basic localization method is determined by the density of the database, which in turn is influenced by two factors: the number of collection locations l and the number of viewing angles v at each position. To investigate the impact of image database density on localization accuracy, we conducted tests in a 200 m^2 test environment using the basic method proposed in Section III-A. We varied the number of collection locations $l \in \{40, 80, 160\}$ and the number of viewing angles $v \in \{3, 4, 5, 6\}$. The viewing angle represents any angle within the 360-degree horizontal circle around the user. In each localization test, we randomly selected l localization points from the scene and gathered v pictures of incompletely overlapping shooting perspectives to form an image database.

Observation: Fig. 5 illustrates the comparison of localization accuracy under different numbers of positions and viewing angles. The results show that the higher the number of positions and viewpoints, the higher the localization accuracy, but the improvement of the accuracy slows down. This indicates that the denser the coverage of the scene is, the better it is for localization. However, too much redundant coverage is not worthwhile.

C. Distance and View Angle Related Feature

Intuitively, the disparity between image features grows when the view angle is fixed but the distance increases. Similarly, when the distance is fixed but the view angle changes, the feature differences also increase. To explore the relationship between image features, distance, and view angle, we conducted two experiments. Fig. 3(a) shows images of the same landmark taken from different positions below a bird's-eye view, with consistent distances between adjacent positions. Using EfficientNet-b0 as the feature extractor (classification layer removed), we compared the Euclidean distances between features of images captured from these positions.

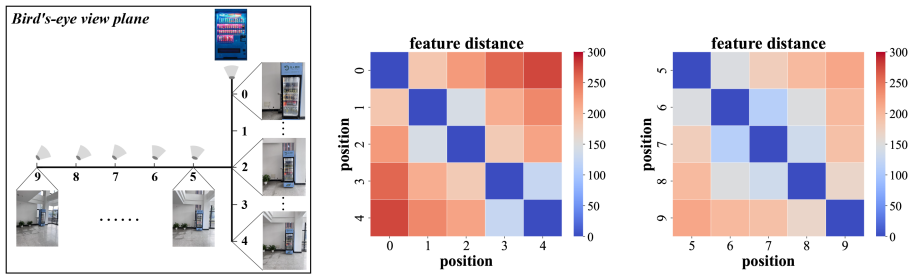
Distance and Feature Distance: Fig. 3(b) shows feature distances between images taken from positions 0 to 4, where the view angle remains constant, but the distances vary. Feature distances are small when images are captured from nearby positions but increase with greater capture distances. This relationship can be leveraged to generate distance-dependent features that are sensitive to distance changes under the same view angle, reducing the need for labor-intensive data collection.

View Angle and Feature Distance: Fig. 3(c) shows the feature distances between images from positions 5 to 9, which have the same vertical distance but varying view angles. It is evident that as the view angle shifts, the feature distance gradually increases, indicating that changes in view angle impact feature matching accuracy.

Furthermore, the overall feature distances in Fig. 3(b) are higher than those in Fig. 3(c). This suggests that when moving the same distance, the feature changes induced by view angle shifts are smaller than those induced by distance changes. And the larger the distance between features, the more accurate it is for retrieval. This observation guides us to decompose the features along the direction with the largest feature changes, which improves the accuracy of retrieval.

D. Impacts of Random Obstacles

In practical indoor localization scenarios, random obstacles (e.g., pedestrians) often appear in the user's field of view, and



(a) Image positions under the bird's eye view plane (b) Position 0-4 (c) Position 5-9

Fig. 3. Effect of different distances(b) and view angles(c) on feature distance. A distance of 0 indicates complete similarity and larger values correspond to higher feature distances.

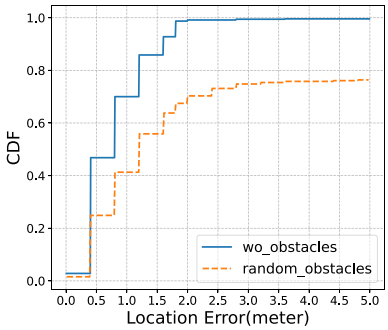
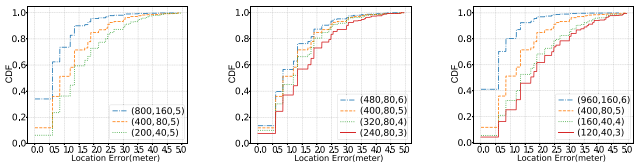


Fig. 4. Comparison of localization results with and without obstacles.



(a) Number of locations (b) Number of views (c) Locations and views

Fig. 5. Notation: (image number, location number, view number). The impact of different acquisition parameters on localization accuracy. (a)change only the number of locations (b)change only the number of views (c)change the number of locations and angles at the same time.

the extent of occlusion caused by these obstacles is variable. While users may try to avoid such obstacles when capturing images, eliminating them entirely is nearly impossible. To assess the impact of random obstacles, we conducted comparative experiments using a basic image-based localization method. One group used test images without significant obstacles, while the other simulated random obstacles by adding random white blocks to the images. The localization error results, shown in Fig. 4, reveal significantly worse performance in the group with random obstacles. This indicates that image-based localization systems are influenced by feature variations in the images, and random obstacles inevitably introduce matching challenges. Therefore, addressing the issue of randomly appearing obstacles during localization is one of the key challenges tackled in this study.

IV. System Framework

Our primary goal in designing the ARGILS is to achieve higher localization accuracy with minimal site survey. The system, shown in Fig. 6, consists of two main phases: offline and online.

In the offline stage(Section V), the localization service provider defines appropriate orthogonal x-y axis directions indoors. Then, several suitable locations are selected as collection points for the site survey, and images are captured in one or more suitable directions from the positive x-axis direction, negative x-axis direction, positive y-axis direction, or negative y-axis direction. Features are then extracted from these images to establish the original fingerprint feature database. Utilizing the trained DistanceGAN model, the original fingerprint feature database is expanded by generating features at the same viewing angle but varying displacements, thereby constructing a generative feature database for online localization.

In the online phase(Section VI), ARGILS initially employs a single-frame orthogonal localization method. After the user captures a scene image with their mobile devices at their location, ARGILS performs orthogonal decomposition of the image features along the x-y axis to extract vertical and horizontal features. The features in these two directions are subjected to two-stage retrieval to quickly obtain the localization result. To better handle random obstacles in the scene, ARGILS also offers a scanning localization option that leverages multiple frames. If the user is convenient to provide a panoramic scan video of the indoor environment from a fixed position, ARGILS will offer a more robust localization service through keyframe filtering and continuous aggregation methods.

V. Offline Fine-Grained Feature Database Generation

In this section, we propose a method to expand the original site survey database using a generative adversarial network. This can build a high-density feature database with as little manual collection work as possible. During the site survey, we set the vertical and horizontal feature perspectives in the scene by selecting the x-y orthogonal direction. Then, sufficient but non-redundant collection positions are selected in the scene to comprehensively collect indoor scenes. This allows us to generate features for the missing locations through DistanceGAN to expand the database.

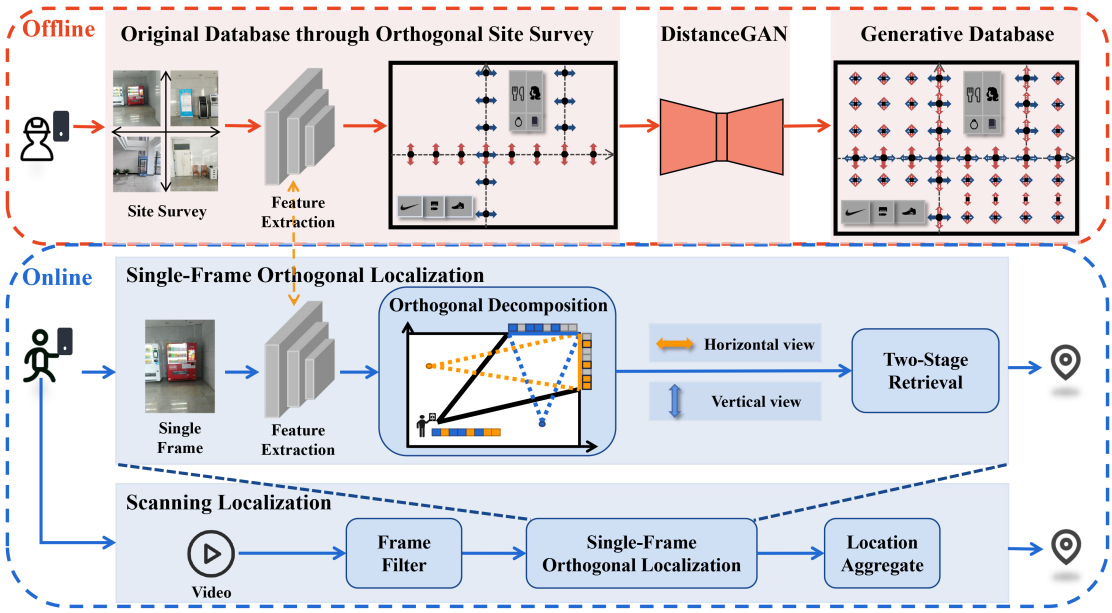


Fig. 6. System overview.

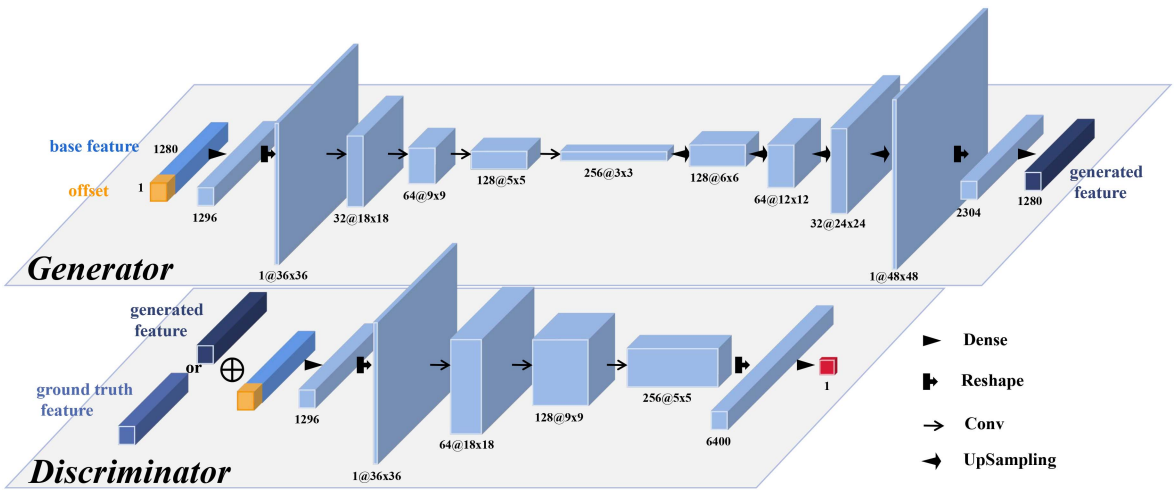


Fig. 7. Structure of DistanceGAN.

Next, we will sequentially introduce the site survey method, the design and training of DistanceGAN, and the construction of the generative feature database.

A. Original Database Through Orthogonal Site Survey

Experimental results in Section III-B show that four viewing angles typically ensure high localization accuracy, as they cover most scene features. Based on this, we conducted the site survey along orthogonal directions, selecting collection points to achieve comprehensive scene coverage with minimal effort.

Specifically, the orthogonal measurement method is shown in Fig. 8. First, two orthogonal directions within the scene must be defined: horizontal and vertical angles. Next, for common scenes, a crosshair can be used to select enough collection points to ensure the complete coverage of the environment. Then, images are captured at the collection points along the horizontal or vertical directions. The occlusion of random obstacles(pedestrians) should be avoided as much as possible during the collection process. It is important to note that although the collection points in Fig. 8 are evenly distributed along a straight line, the actual positions are flexible. Since real-world scenes can be complex, it is only necessary to ensure that the collection viewpoints are either horizontal or vertical. Features extracted from images using EfficientNet-b0 are used as fingerprints. By combining location labels with features and their corresponding directions, we were able to construct the original site survey database with very minimal effort. We use $\mathbb{L}, \mathbb{F}, \mathbb{D}$ and n to represent coordinates, features, view directions and the number of features respectively. Then the original site survey database

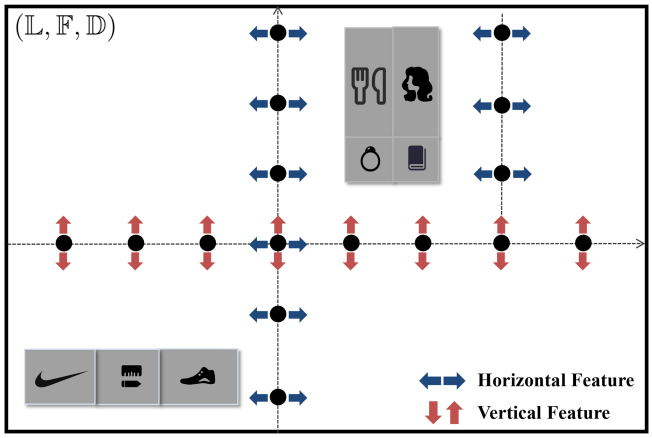


Fig. 8. Original feature database established through orthogonal site survey.

is defined as:

$$(\mathbb{L},\mathbb{F},\mathbb{D}) = \{\langle l_i, f_i, d_i\rangle \,|i = 1,2,\ldots,n\} = \{\langle (x_1,y_1), f_1, d_1\rangle, \ldots, \langle (x_n,y_n), f_n, d_n\rangle\}, \quad (1)$$

where $d_i \in \mathbb{D} = \{1,2,3,4\}$. d equals 1, 2, 3, or 4 indicates that the shooting angle is positive x-axis, negative x-axis, positive y-axis, or negative y-axis, respectively.

B. Distancegan

1) Structure of DistanceGAN: We constructed the DistanceGAN based on CGAN [43]. The network structure is depicted in Fig. 7 and consists of two main parts: the generator and discriminator. The generator takes the base feature and displacement splice as input and aims to generate the feature vector after displacing the base feature by the corresponding distance. Our network design draws inspiration from the context-encoder [45], which exhibits strong feature learning capabilities and serves as the foundation for various image inpainting tasks. In order to align the distribution of image features extracted by EfficientNet, we substitute the original RELU [45] with the ELU [50], which is capable of generating negative values. This modification also serves to reduce the computational burden on the model.

2) Training of DistanceGAN: The purpose of feature generation is to enhance localization performance. Conventional generative tasks using GANs often employ content-based loss, like Mean Squared Error (MSE), for supervised training. However, as validated by [51] and [52], MSE is unable to guide the model in learning the structured information within the image. At the same time, due to the consistent view angles and close proximity between the features in this study, the differences between the features are minimal, requiring contrastive methods to enhance the model's learning capability. To address this, we propose a joint loss, combining adversarial loss and content loss with distance loss during training.

To prepare the training data, our dataset contains numerous single-view image sets. For each perspective image set, we extract three different feature vectors from pictures with various displacements Fig. 9(a): positive feature, base feature, and negative feature. The base feature serves as a reference to generate features corresponding to the position of the positive feature, which also serves as the ground truth for the generated feature. To better learn the position distinction between features, we also include any non-ground truth position feature as a negative feature.

Adversarial Loss: Our adversarial loss aligns with that of GAN, aiming to generate features that closely resemble the original feature distribution. During the training process (Fig. 9(b)), we use the base feature f_b, offset value o, and ground truth feature f_p at the offset for adversarial training. During discriminator(D) training, the concatenation of basic features f_b, ground truth feature f_p, and corresponding offset value o will be marked as true, otherwise it will be marked as false if it contains the generated offset features. Correspondingly, in the generator(G) training stage, the ground truth feature f_p is used for supervised training. The adversarial loss is expressed as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{f_p\sim F}\left[\log D\left(f_p \mid f_b, o\right)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G\left(f_b, o\right)\right)\right)\right] \quad (2)$$

Distance Loss: As shown in Fig. 9(c), distance loss aims to learn the distance discrimination between features. We design the distance loss as a combination of MSE loss and Triplet loss [48], defined as follows:

$$\begin{aligned} L_{dis} &= L_{MSE} + L_{Triplet} \\ &= \frac{1}{n}\sum_{i=1}^{n} \|G(f_b,o) - f_p\|_2^2 \\ &\quad + \sum_{i=1}^{n} \max\left(\|G(f_b,o) - f_p\|_2 \right. \\ &\quad \left. - \|G(f_b,o) - f_n\|_2 + \alpha, 0\right) \end{aligned} \quad (3)$$

where f_n is a negative sample indicating any other feature except f_p under the same perspective. α is a margin that controls the minimum difference between the positive and negative pairs in the feature space. Specifically, the margin α ensures that the model is penalized if the distance between the anchor and the negative sample becomes smaller than the distance between the anchor and the positive sample, thus forcing the model to learn a clear distinction between positive and negative pairs. The Euclidean distance $\|a-b\|_2$ measures the difference between two feature vectors.

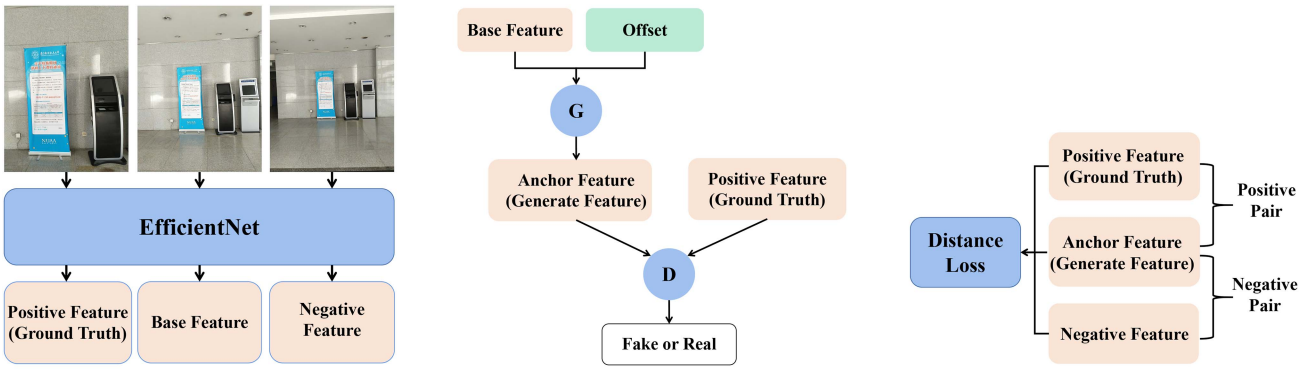
In this way, the MSE loss focuses on minimizing the overall error between generated and ground truth features, while the Triplet loss ensures that the model learns to distinguish between similar and dissimilar features effectively.

Joint Loss: We define joint loss as follows:

$$L = \lambda_{adv} L_{adv} + \lambda_{dis} L_{dis}, \quad (4)$$

where λ_{adv} and λ_{dis} are the coefficients of loss respectively.

Displacement Retrieval Error: In order to evaluate the performance of the generative model, we propose the displacement



(a) Training data prepare. (b) Adversarial training. (c) Distance loss training.

Fig. 9. Overview of DistanceGAN training.

retrieval error. We use $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$ to represent the set of features obtained at the same angle but different distances, and calculate the corresponding distance set $\mathcal{S} = \{s_1, s_2, \ldots, s_k\}$ from the coordinates corresponding to these features. Then we can get:

$$(\mathcal{S}, \mathcal{F}) = \{\langle s_i, f_i \rangle | i = 1, 2, \ldots, k\}, \tag{5}$$

from which we can get the offset set for the generator based on base feature $\langle s_b, f_b \rangle \in (\mathcal{S}, \mathcal{F})$:

$$\begin{aligned} O &= \{s_b - s_i | s_i \in \mathcal{S}\} \\ &= \{o_1, o_2, \ldots, o_k\}. \end{aligned}$$

Then we can use the generator(G) to get the generative set

$$\begin{aligned} (\mathcal{S}, \mathcal{F}_b^*) &= \{\langle s_i, G(f_b, o_i) \rangle | i = 1, 2, \ldots, k\} \\ &= \{\langle s_1, f_1^* \rangle, \ldots, \langle s_b, f_b \rangle, \ldots, \langle s_k, f_k^* \rangle\}. \end{aligned} \tag{6}$$

Suppose $\langle s_t, f_t \rangle \in (\mathcal{S}, \mathcal{F})$ is chosen as the test case. The best matching result retrieved in the generative set is $\langle s_j, f_j^* \rangle \in (\mathcal{S}, \mathcal{F}_b^*)$, thus the displacement retrieval error can be obtained:

$$DRE = |s_t - s_j|, \tag{7}$$

C. Generative Feature Database Using DistanceGAN

After training the DistanceGAN, we can use it to expand the original feature database. For any feature in the original database, we can select a set of appropriate offsets O according to its viewing direction. When the viewing direction is horizontal, the offset corresponds to the change of the horizontal coordinate x, and vice versa for the change of the vertical coordinate y. For any original feature $\langle (x_i, y_i), f_i, d_i \rangle \in (\mathbb{L}, \mathbb{F}, \mathbb{D})$, suppose the offsets set is:

$$O_i = \begin{cases} \Delta X_i = \{\Delta x_{i1}, \ldots, \Delta x_{im}\}, d_i = 1, 2 \\ \Delta Y_i = \{\Delta y_{i1}, \ldots, \Delta y_{im}\}, d_i = 3, 4 \end{cases} = \{o_{i1}, \ldots, o_{im}\}, \tag{8}$$

where m represents the number of features to be generated. Then we use the generator(G) in DistanceGAN to obtain a set of feature groups $\mathcal{F}_i^*$ based on f_i as:

$$\mathcal{F}_i^* = \{G(f_i, o_k) | o_k \in O_i\} \tag{9}$$

A positive value of O_i corresponds to moving forward from the current perspective, whereas a negative value corresponds to moving backward from the current perspective. Consequently, the coordinates are updated differently under different shooting angles d_i. The coordinates $\mathcal{L}_i^*$ of the generated features can be obtained from the offset and the viewing direction:

$$\mathcal{L}_i^* = \begin{cases} \langle x_i, y_i \rangle + \Delta X_i, d_i = 1 \\ \langle x_i, y_i \rangle - \Delta X_i, d_i = 2 \\ \langle x_i, y_i \rangle + \Delta Y_i, d_i = 3 \\ \langle x_i, y_i \rangle - \Delta Y_i, d_i = 4 \end{cases} = \{l_{i1}^*, \ldots, l_{im}^*\}. \tag{10}$$

Consequently, for each feature $\langle (x_i, y_i), f_i, d_i \rangle$ in the original database, a set of generated features with the same view direction but different positions can be obtained as:

$$(\mathcal{L}_i^*, \mathcal{F}_i^*, \mathcal{D}_i) = \{\langle l_{i1}^*, f_{i1}^*, d_i \rangle, \ldots, \langle l_{im}^*, f_{im}^*, d_i \rangle\}.$$

We use $\mathbb{L}^*, \mathbb{F}^*, \mathbb{D}$ and n to represent generative coordinates, generative features, view directions and the number of features respectively. Finally, all the generated features together form a generative feature database as follows:

$$(\mathbb{L}^*, \mathbb{F}^*, \mathbb{D}) = \{(\mathcal{L}_1^*, \mathcal{F}_1^*, \mathcal{D}_1), \ldots, (\mathcal{L}_n^*, \mathcal{F}_n^*, \mathcal{D}_n)\}. \tag{11}$$

The generated database is shown in Fig. 10. Note that the set of generated offsets in the horizontal and vertical directions here is flexible.

VI. Online Real-Time Localization With Orthogonal Feature Decomposition

In this section, we propose a single-frame orthogonal localization algorithm based on orthogonal feature decomposition, designed to address the issue of missing viewing angles in the generative database. Orthogonal localization first extracts features from the user-provided image using the same method as the offline process, and then decomposes the features along the horizontal and vertical directions. By conducting a two-stage retrieval for the horizontal and vertical features in the feature database, the user's positional information in the two orthogonal directions can be obtained separately. This orthogonal feature decomposition approach aligns with the site survey method,

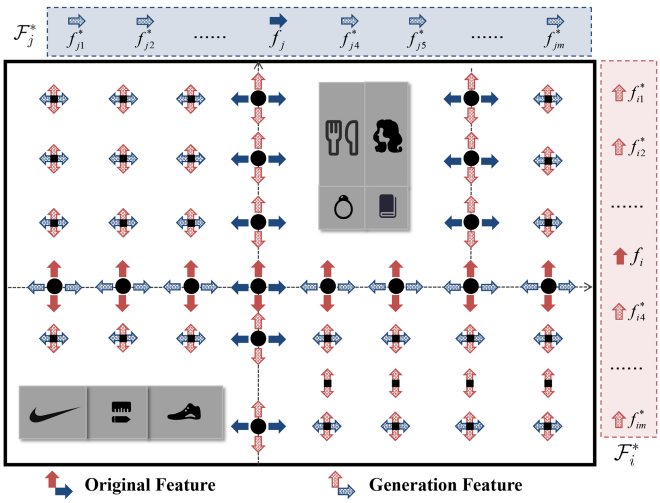


Fig. 10. Generative features database construction.

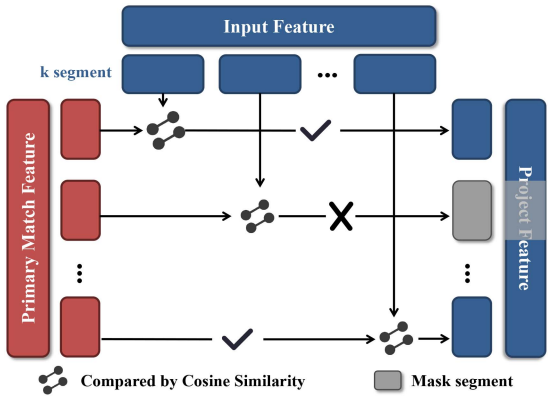


Fig. 11. Feature projection.

allowing ARGILS to effectively utilize the generative database to enhance localization accuracy, even when the user's captured angle differs from the feature library's angle. However, single-frame images are susceptible to random obstacles, which may interfere with localization. To mitigate this, we offer a scanning localization option. Scanning localization filters keyframes from the multi-frame video provided by the user and continuously aggregates the results from single-frame orthogonal localization, thereby reducing the impact of random obstacles and improving localization accuracy.

A. *Single-Frame Orthogonal Localization*

As shown in Section V-C, the generative feature database contains only features from four fixed view directions. However, the user's shooting angles are random, and there is no guarantee that they will be consistent with those in the generated database.

To address the issue of limited perspectives in the generative feature database, We propose a single-frame orthogonal localization method for fast and efficient real-time localization. After obtaining the features through feature extraction, the single-frame orthogonal localization method employs a feature projection method to decompose single image features along the horizontal and vertical directions. Subsequently, a two-stage retrieval process is conducted for the horizontal and orthogonal features, respectively. Ultimately, the positional information in the horizontal and vertical directions is utilized to obtain accurate localization results. Next, we will introduce the idea of orthogonal feature decomposition and two-stage retrieval in detail. Finally, the complete single-frame orthogonal localization algorithm is given.

1) Orthogonal Feature Decomposition: When the user's shooting angle lies between the horizontal and vertical axes, the extracted features from the image will contain elements from both directions. Therefore, we propose a feature projection strategy. Feature projection begins by selecting a reference feature (Primary Match Feature). If the reference feature and the input feature share overlapping perspectives, the corresponding overlapping features will exhibit high similarity. Based on this idea, we segment both the input feature vector and the reference feature and then calculate the similarity between each segment sequentially. When segment features are similar, that segment is retained as the projected feature. Conversely, masked segments will replace the original segments and will not be used in subsequent retrieval and matching processes. The feature projection process is shown in Fig. 11. Through feature projection, we can retain the features of the input features that match the target viewing direction, thereby reducing the interference caused by irrelevant features when the input features are generated and positioned in this direction.

To enable feature decomposition along the horizontal and vertical directions, we first classify the original feature database according to direction as:

$$(\mathbb{L}_h, \mathbb{F}_h, \mathbb{D}_h)_{horizontal} = \{\langle l_i, f_i, d_i\rangle | d_i \in \{1, 2\}, f_i \in \mathbb{F}\}$$
$$(\mathbb{L}_v, \mathbb{F}_v, \mathbb{D}_v)_{vertical} = \{\langle l_i, f_i, d_i\rangle | d_i \in \{3, 4\}, f_i \in \mathbb{F}\}.$$

Then, the user's input feature I is retrieved in both $\mathbb{F}_v$ and $\mathbb{F}_h$ to obtain the optimal results through

$$\begin{cases} f_v = \max_{\mathbf{f}\in\mathbb{F}_v} \mathrm{Sim}(\mathbf{f}, I) \\ f_h = \max_{\mathbf{f}\in\mathbb{F}_h} \mathrm{Sim}(\mathbf{f}, I) \end{cases} \tag{12}$$

where Sim represents the operation of calculating the cosine similarity of two feature vectors.

Then the reference features in the vertical and horizontal directions are f_v and f_h respectively. Through feature projection, we can separately obtain the vertical projection feature *(vpf)* and the horizontal projection feature *(hpf)*. At this point, the orthogonal feature decomposition of the single-frame feature has been completed.

2) Two-Stage Retrieval: As the generated features in the generative database are expanded separately according to each single original feature, this sequential relationship can be utilized in the retrieval process. At the same time, the original features can quickly and accurately help the system to narrow down the retrieval scope. In light of these considerations, we propose a two-stage retrieval method.

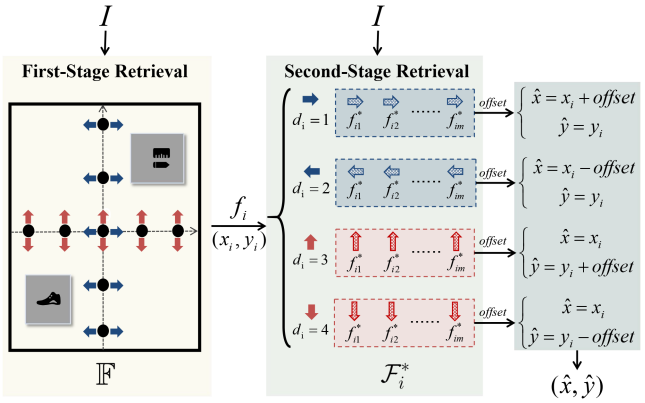


Fig. 12. Two-Stage retrieval.

We will input *vpf* and *hpf* obtained through orthogonal feature decomposition into the two-stage retrieval process separately. Therefore, the two-stage retrieval process will use I to generally refer to the input feature. The process of two-stage retrieval is shown in Fig. 12.

- First-Stage Retrieval: The query image feature I provided by the user is utilized to retrieve the optimal match $\langle (x_i, y_i), f_i, d_i \rangle$ in the original feature library $(\mathbb{L}, \mathbb{F}, \mathbb{D})$ through (12).
- Second-Stage Retrieval: As illustrated in Section V-C, the generated features with different distances for the same viewpoint are generated through f_i as follows:

$$\langle l_i, f_i, d_i \rangle \xrightarrow{\text{Extension}} (\mathcal{L}_i^*, \mathcal{F}_i^*, \mathcal{D}_i).$$

Then I is further retrieved from this set of generated features to obtain the optimal match as well as the corresponding offset. The optimization method of offset-based coordinates varies depending on the d_i of the feature. At this point, the user's position information in direction d_i has been confirmed.

For feature I without decomposition, two-stage can only optimize position information in one direction at a time. But when we implement two-stage retrieval on *vpf* and *hpf* respectively, the user's position information can be confirmed in the vertical and horizontal directions simultaneously. Specifically, $vpoint = \langle (x_v, y_v), f_v, d_v \rangle$ and $hpoint = \langle (x_h, y_h), f_h, d_h \rangle$ represent the matching points of *vpf* and *hpf* in the first stage respectively. $offset_v$ and $offset_h$ represent the offsets obtained in the second stage. Take $d_v = 3$ and $d_h = 1$ as an example, as shown in Fig. 13. Then the final localization result $(\hat{x}, \hat{y})$ of the user is:

$$\begin{cases} \hat{x} = x_v + offset_v \\ \hat{y} = y_h + offset_h \end{cases} \tag{13}$$

Compared with traversing and searching directly in the generative database, two-stage retrieval has the advantage of reduced time complexity. According to (1) and (8), the number of features in the original database is n, and each original feature corresponds to the generation of m generated features. Therefore, the time complexity of traversing retrieval is $O(m \times n)$, while the two-stage method is $O(m + n)$. This reduction in time complexity greatly reduces the retrieval time, thereby improving the localization speed.

Algorithm 1: Single-Frame Orthogonal Localization Algorithm.

Require: input feature I; original feature database ofd; expand feature database efd; threshold of check one view or two views cth; segment length sl; threshold of cosine similarity cos;
Ensure: Location

function ProjectFeature($fsrc, fdest$):
  $fproj = fsrc.copy()$
  $k = fsrc.length \div sl$
  for $i = 0; i < k; i++$ **do**
    $sim =$ CosineSimilarity($fsrc[i \times sl, (i+1) \times sl], fdst[i \times sl, (i+1) \times sl]$)
    if $sim < cos$ **then**
      $fproj[i \times sl, (i+1) \times sl] = Mask()$
    end if
  end for
  return $fproj$
$vpoint =$ Retrieval($I, ofd.verticalView$)
$hpoint =$ Retrieval($I, ofd.horizontalView$)
$vpf =$ ProjectFeature($I, vpoint.feature$)
$hpf =$ ProjectFeature($I, hpoint.feature$)
if $|vpoint.confidence - hpoint.confidence| > cth$ **then**
  if $vpoint.confidence > hpoint.confidence$ **then**
    $b_v =$ GenerativeSetRetrieval($vpf, vpoint, efd$)
    $b_h = 0$
  else
    $b_h =$ GenerativeSetRetrieval($hpf, hpoint, efd$)
    $b_v = 0$
  end if
else
  $offset_v =$ GenerativeSetRetrieval($vpf, vpoint, efd$)
  $offset_h =$ GenerativeSetRetrieval($hpf, hpoint, efd$)
end if
if $hpoint.d == 1$ **then**
  $location.x = hpoint.x + offset_h$
else if $hpoint.d == 2$ **then**
  $location.x = hpoint.x - offset_h$
end if
if $vpoint.d == 3$ **then**
  $location.y = vpoint.y + offset_v$
else if $vpoint.d == 4$ **then**
  $location.y = vpoint.y - offset_v$
end if
return location

3) Single-Frame Orthogonal Localization Algorithm: The single-frame orthogonal localization algorithm process is shown as Algorithm 1. Both the orthogonal decomposition and the first stage of the two-stage retrieval require searching within the original database, so we can merge this step. It is worth noting that when there is a significant confidence gap between *vpoint*

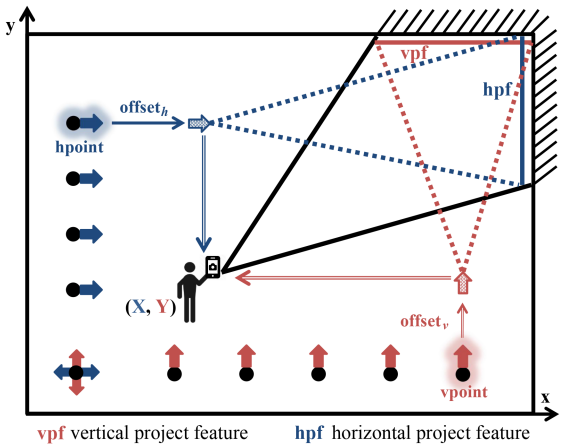


Fig. 13. Scene representation of single-frame orthogonal localization.

and *hpoint* (line 15 in Algorithm.1), we assume that the shooting angle is either horizontal or vertical. In this case, we only use the feature with a higher confidence level for two-stage retrieval to optimize the localization along a single direction.

In summary, the single-frame orthogonal localization algorithm resolves missing perspectives in the generative database and optimizes localization accuracy in both directions. The two-stage retrieval improves real-time performance by reducing localization time.

B. Scanning Localization

While orthogonal localization enables effective localization from a single image across various scenarios, challenges remain in certain conditions. Specifically, occlusions or interfering objects within the scene can adversely affect localization results. Additionally, images that lack distinctive features, such as large, textureless white walls, can lead to localization errors. These issues often arise when users select inappropriate shooting angles. When only a single image is captured for localization, there may be significant discrepancies between the localization outcome and the actual position, resulting in a suboptimal user experience, such as repeated location requests.

To address these challenges, we have developed an optional localization mode called scanning localization. In this mode, users are required to rotate their phones to capture a video of the scene while staying at a fixed point. Suitable frames are then selected from the video stream for single-frame orthogonal localization, allowing us to continuously update and optimize the user's location in real-time through clustering techniques.

The key issue of scanning localization is how to automatically select appropriate keyframes in the video, and how to fuse the localization results of these keyframes to obtain the final precise localization. The flow of the scanning localization algorithm we designed is shown in Algorithm 2.

1) Key Frames Filtering: A single video contains a large number of frames with little variability between adjacent frames, making frame-by-frame localization increase the system overhead and latency without the corresponding improvement in localization accuracy. Therefore, we need to perform efficient and reasonable keyframe selection to select the images that can be efficiently localized and filter out the redundant and repetitive images. Therefore, we use three dimensions: image sharpness, number of features and keyframe variability to evaluate the criticality of frames.

- *Image sharpness:* A sharp image means that the difference between light and dark of its texture boundaries is obvious, which is more conducive to feature extraction and matching. We use the Laplace operator f_1 to describe the clarity of the image.
- *Number of features in the image:* We want the image to have more features so that its differentiation will be more obvious and the accuracy in image feature retrieval will be higher. We choose to use ORB [53] to extract the local features of the image, which can achieve a feature extraction speed of more than 30fps on the mobile side. By setting the intensity threshold of the features, we consider the local features that meet the threshold as valid local features and the number of valid features as f_2.
- *Keyframe discrepancy:* When the variability of adjacent keyframes is high, different keyframes can cover more scene features. Considering that each frame image has the same size as each other and we want different keyframes to contain different structural information, this paper uses the Hamming distance in the perceptual hashing algorithm [54] to represent the difference between two images. For the previous keyframe Pre and the current frame Cur, the keyframe discrepancy f_3 is

$$\begin{aligned} f_3 &= HammingDistance(hash_{Pre}, hash_{Cur}) \\ &= \sum_{i=0}^{n} hash_{Pre}[i] \oplus hash_{Cur}[i] \end{aligned} \tag{14}$$

where $\oplus$ represents XOR operation.

After the above calculation, we can get the frame features $x_i = \{f_1, f_2, f_3\}$ of the i-th candidate image. We use a binary classification model to classify the keyframes, and the classification result y_{pre} is 1 when the candidate image is a keyframe, and 0 when it is a non-key frame.

$$y_{pre} = W * x + b \tag{15}$$

where W, b are the model parameters. To train the model, we construct the training data using the collected data in the offline phase and optimally train these parameters using the Binary Cross-Entropy function:

$$\text{Loss} = -(y \cdot \log(y_{pre}) + (1 - y) \cdot \log(1 - y_{pre})) \tag{16}$$

We collected landmark images and surround video data of the environment respectively in the offline stage, where the image data are used as landmark data and the video sayings are used as training data. For the i-th frame of the j-th video sample, we calculate the localization error e_i^j of the frame feature x_i^j. we identify frame i with e_i^j less than 15% as an alternative keyframe. We establish a filtering criterion for alternative frames with intervals not less than 10, selecting some as key frames

Algorithm 2: Scanning Localization Algorithm.

Require: input video frame stream fs; original feature database ofd; expand feature database efd
Ensure: Location

```
location = NULL
frame, status = ReadFrameFromStream(fs)
while status == true do
    fil = FilterFrame(frame)
    if fil == false then
        frame = Preprocess(frame)
        feature = FeatureExtract(frame)
        points = TopMatchRetrivial(feature, ofd)
        clustersNum = Cluster(points) // KMeans
        if clustersNum == 1 then
            currentLocation = GenerativeLocalization(feature, point, efd)
        else if clustersNum > 1 then
            verticalPoint, horizontalPoint = GetOrthogonalPoints(points)
            currentLocation = OrthogonalLocalization(verticalPoint, horizontalPoint, efd)
        end if
        location = Aggregate(currentLocation, location)
    end if
    frame, status = ReadFrameFromStream(fs)
end while
return location
```

labeled as 1, while the remaining frames are labeled as 0. This process forms the training dataset for the model.

At this point, we get a binary classification model that can perform keyframe detection. In the online localization process, we use a sliding window of length 10 to limit the selection range of keyframes, so that at most one keyframe is included in the same window.

2) Aggregate Update: After keyframe filtering for the query video submitted by the user, we perform single-frame orthogonal localization on the obtained keyframes respectively. Ideally, the localization results of these images are the same, but the localization results of a single image will be biased due to the existence of viewing angle and obstacle defects, so we need to aggregate these positions to obtain a better aggregation center as the localization result.

During location aggregation, we define an image sequence as I, where I_i represents the i-th view image. We can obtain a single localization result $L_i : (x_i, y_i)$ through position estimation during sequence processing. The confidence C_i of L_i is represented by the matching error. Aggregating the position in the sequence is an iterative process during which the user's current position $L^* = (x^*, y^*)$ is continuously calculated. Therefore, we can treat location aggregation as using the location L_i in the image sequence to continuously update L^*.

$$x^* = x* + (x_i - x^*) \times \frac{C_i}{C_i + C*} \tag{17}$$

(a) Shopping mall

(b) Office building

(c) Laboratory lobby

Fig. 14. Experimental areas.

TABLE I
DATA ACQUISITION IN DIFFERENT SCENARIOS

#	Scenarios	Size(m^2)	Map locations	Map images	Test locations / videos
1	Shopping mall(one floor)	5120	210	430	160
2	Office Building	1080	80	172	60
3	Laboratory Building Lobby	231	20	42	20

$$y^* = y* + (y_i - y^*) \times \frac{C_i}{C_i + C*} \tag{18}$$

$$C^* = \max(C^*, C_i) \tag{19}$$

As a result, we achieve position aggregation in the iterations of the image sequence, and the position confidence increases with the number of images.

VII. Implementation and Evaluation

A. Experimental Settings

1) Experimental Scenarios: We conducted experiments in three scenarios: a shopping mall's one floor, an office building's first floor, and a laboratory building's lobby. As shown in Fig. 14, these scenes have distinct layout distributions, with the mall having a large area with more iconic elements, the office building being relatively empty with sparse features, and the laboratory space being compact with repeated elements and low scene discrimination. The data collection situation is summarized in Table I.

2) Experimental Environment: The feature generative model was trained using Keras on an Nvidia GeForce RTX 2080Ti

and deployed on an Ubuntu 20.04 machine for expanding the feature database. Our localization system was primarily tested on three mobile phone models: OnePlus 8 T, HUAWEI Mate 20, and 360 N6 Pro, to evaluate different models' performance. Please note that different mobile devices have different sizes of photosensitive components and zoom ranges, so the images taken at the same location with different focal lengths by users may vary significantly. When the equivalent focal length used by the user for shooting differs significantly from the equivalent focal length at the time of data collection, differences in localization retrieval may occur. To avoid the inconvenience caused by the heterogeneity of device focal lengths, this paper uniformly uses the most common equivalent focal length size at present, which is 26 mm, for both data collection and retrieval, thus ensuring consistency in the perspective of map collection and user retrieval. In the application of smartphones, the API provided by the zoom lens can be utilized to set the equivalent focal length.

3) Data Acquisition: Data collection involved obtaining a generative model dataset and a localization dataset. We developed a mobile application to capture images quickly with manual marking for ease of use.

The generative model requires a single-view dataset. For each view, we will take multiple displacement pictures, and use the relative displacement value and the view id as their labels. In the process of generating data collection, we use multiple different types of mobile phones to capture images and keep the distance between any two adjacent images within the range of 0.4 m-1 m. Finally, we take 4544 pictures of 227 different views for GAN model training and testing.

The localization dataset is divided into map data and test data. Initially, we partition each experimental scene into several approximate rectangular areas. We then capture vertical and horizontal perspective images along the centerline of each rectangle at a distance of 0.8 m to 1 m. These images are annotated with camera coordinates. Finally, we randomly select several locations within each scene to record a 360-degree video, marking the corresponding coordinates. All test images are sourced from these randomly selected locations, simulating real-world usage scenarios.

4) Comparative Methods: In order to verify the performance of our system, we compare with a number of different state-of-the-art approaches: HAIL [18], Sextant [17], GLN [55], PixLoc [56] that uses DISK [57] for feature extraction and LightGlue [58] for feature matching. Our ARGILS aims to achieve lightweight real-time and reliable localization services based on a sparse image database. We will compare ARGILS and state-of-the-art methods from multiple aspects: localization accuracy, multi-scene verification, mobile terminal localization performance, and storage overhead. To verify the effects of orthogonal localization and test the feasibility of mobile platform deployment.

B. Performance Evaluation

1) Real Localization Case: We illustrate the localization process and real effects using a real case in an office building scene, as depicted in Fig. 14(b). In Fig. 15, random perspective pictures taken by the user at their location serve as retrieval inputs. Initially, in step 1, the input image features undergo orthogonal decomposition in both vertical and horizontal directions. Subsequently, matches are retrieved separately in the original database. If the confidence of retrieval in both directions is similar, the offset position is determined within the respective feature databases in step 2. Finally, the final localization result is derived in step 3 by utilizing the matching positions in both vertical and horizontal directions. However, in the scenario depicted in the upper left corner, due to significant disparity in confidence between the features obtained in the vertical and horizontal directions, only the generated feature database in the horizontal direction is utilized to obtain the localization result.

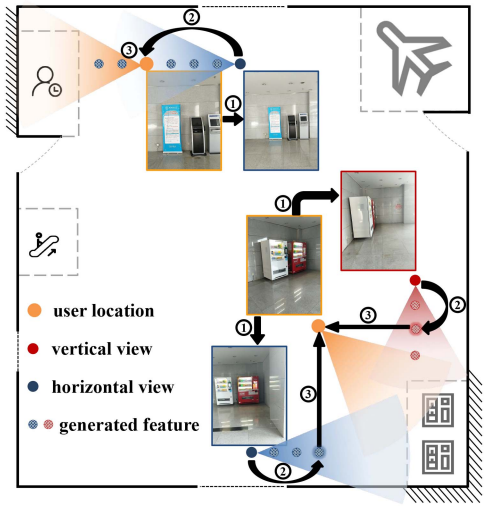
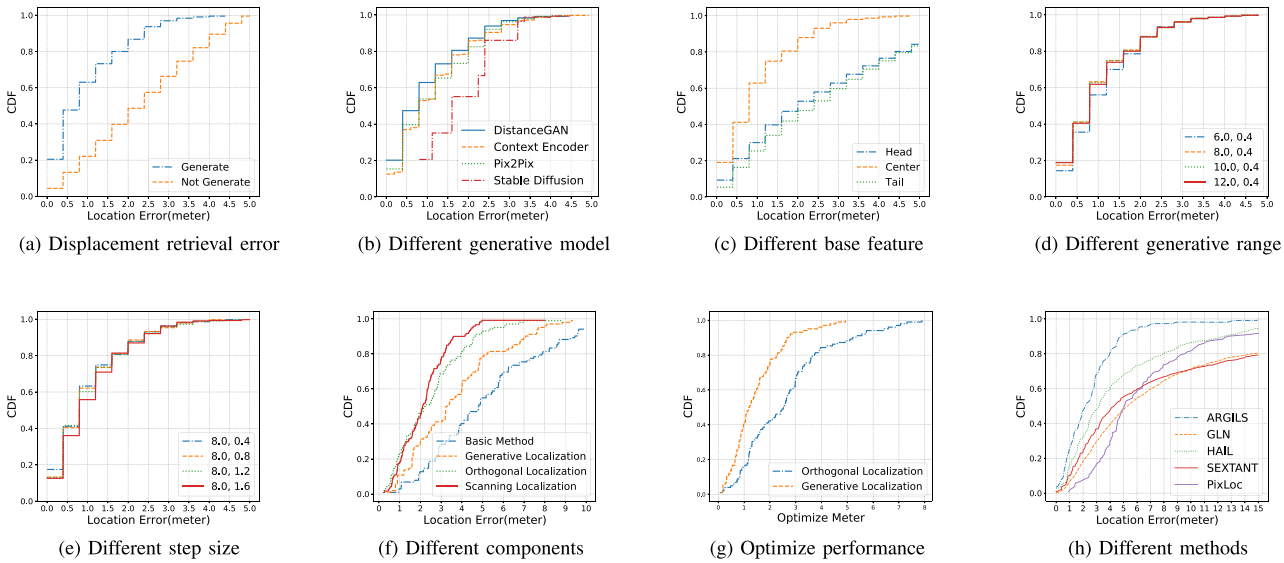


Fig. 15. Real case.

2) Performance of DistanceGAN:

- Reduction of Location Error: DistanceGAN generates missing location features in the map. We evaluate the generative model's effectiveness using displacement retrieval error in Section V-B-2. For comparison, we select a base feature from each viewing angle feature set and generate features based on displacement. Without generation, the base feature is used to construct the map, and the retrieval error for any feature in the set is measured as its relative displacement to the base feature. With generation, the error is measured as the displacement retrieval error. Fig. 16(a) illustrates the generative model's performance compared to the non-generated retrieval error. The average error after generation is 0.99 m, while without generation, it is 2.3 m.
- Different Generative Model: Fig. 16(b) further compares the generative effects of different GAN models: Stable Diffsuion [44], Context-Encoder [45] and Pix2Pix [46] that use U-Net [59] structure for image translation. We adjusted the feature and displacement input for Pix2Pix and Context-Encoder during training and evaluated their generation effect using displacement retrieval error. For Stable Diffusion, prompts with distance constraints are used directly for generating. DistanceGAN achieved the



(a) Displacement retrieval error (b) Different generative model (c) Different base feature (d) Different generative range

(e) Different step size (f) Different components (g) Optimize performance (h) Different methods

Fig. 16. Performance evaluation.(d)(e)The parameters represent range size and step size respectively.

best generation effect, with mean error decreased by 15.0% and 15.7% respectively compared to context-encoder and pix2pix. In comparison to Stable Diffusion, which is incapable of understanding actual physical distances, DistanceGAN has undergone significant advancements.

- Different Base Feature: Choosing the right base feature is crucial for better generation. Under the same viewing angle, the front relative position has a higher definition but fewer perceptible features, while the back position's image area is relatively blurred, but more features can be perceived. Therefore, we experimented with features from the front, center, and tail ends as the base feature, setting the generating range to 8 m and the step size to 0.8 m. As shown in Fig. 16(c), selecting the middle position as the base feature yields an average error of only 1.02 m, significantly lower than the errors caused by selecting the front or back ends. The middle position maintains a relatively close distance to both ends simultaneously, resulting in lower generation difficulty.
- Different Generative Range: The generation range will not only affect the performance of the generation but also the site survey step. The larger the generation range, the smaller the sampling density. We choose the feature at the center position as the base feature and test the generation range of 6 m, 8 m, 10 m, and 12 m with a generation step length of 0.4 m. As shown in Fig. 16(d), when the generation range exceeds 8.0 m, the localization accuracy is relatively close. Although the increase in the generation range can reduce the sampling density, the indoor scene is not open. The generation under the same viewing angle usually does not exceed 10 m. Its localization effect is not much different from the 8 m generation range, but it will increase the size of the feature library, so choose 8 m. The generation range is more suitable for mobile applications.
- Different Step Size: The main factor affecting the generation step size is the size of the generated database. We set the generation range to 8.0 m, and select the center location feature as the base feature for experimentation. As shown in Fig. 16(e), the average localization accuracy of 0.8 m step length is 1.02 m, which is slightly lower than 0.99 m of 0.4 m step length, but the size of the feature library generated by 0.4 m step length is twice the 0.8 m step length, so choose 0.8 m step length is more conducive to mobile terminal deployment.

The above experimental results demonstrate that our generative model optimizes localization results under the same perspective and significantly improves localization accuracy. Unlike the direct feature generation of the confrontation network, DistanceGAN enhances feature discrimination under different displacements and improves generation effectiveness using distance loss. The experimental results indicate that selecting the center location feature for generation results in higher localization accuracy. Since the intermediate position can be generated in both forward and backward directions simultaneously, in some narrow spaces, it is possible to choose only one position for data collection when facing different directions on the same straight line.

3) Performance of Different Components: We compare different modules of ARGILS in shopping malls, including orthogonal localization, generative localization, and scanning localization, using a basic method as a benchmark. Generative localization employs the two-stage retrieval method from Section VI-A-2 in the generative database. As shown in Fig. 16(f), their mean errors are 5.14 m, 3.67 m, 2.50 m, and 2.20 m, respectively, with orthogonal and scanning localization showing higher accuracy. Scanning localization can achieve better localization accuracy than orthogonal localization based on a single frame, because single frame localization is often more easily affected by random obstacles(pedestrians). This

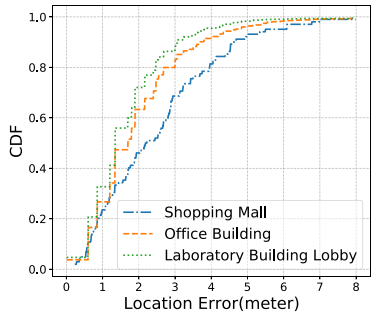


Fig. 17. Different areas.

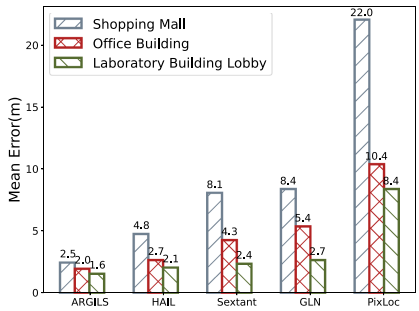


Fig. 18. Mean error in different scenes.

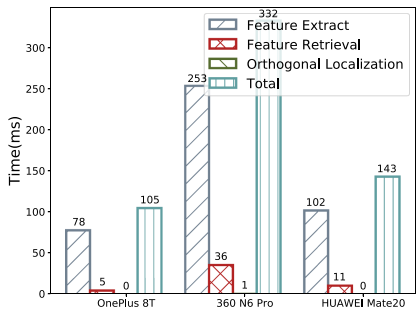


Fig. 19. Time consumption.

suggests scanning localization enhances fault tolerance and addresses partial orthogonal localization failures. Fig. 16(g) further compares orthogonal and generative localization's optimization relative to the basic method, showing that orthogonal localization offers more optimization potential and better overall performance. Both methods improve basic method errors due to sparse database construction, demonstrating the feasibility of optimizing localization from both vertical and horizontal perspectives.

4) Performance Comparison: The results of comparison with other state-of-the-art methods are shown in Fig. 16(h). We compared Pixloc(DISK+LightGlue), HAIL, Sextant, and GLN using the orthogonal localization method and achieved the best performance. The mean localization error of our method is 2.50 m, which is 48.2% lower than the best-performing HAIL among other methods. At the same time, the localization error of our method reaches 90.7% within 5 m, which is 34.1% higher than that of HAIL. PixLoc builds 3D cloud maps using the same original site survey database as in ARGILS. The results show that PixLoc has poor localization results due to the sparsity of the raw acquisition, but ARGILS is able to compensate for this by generating features. Experiments show that, on the one hand, ARGILS can effectively reduce matching errors through deep feature retrieval, thereby reducing large localization errors, and keeping localization accuracy within a relatively controllable range. On the other hand, by generating a combination of localization and orthogonal localization, our method is also superior to other methods in low-error accuracy.

5) Performance in Different Areas: We test ARGILS in three different scenarios. In the test, using orthogonal localization to calculate the position, the results are shown in Fig. 17. The localization errors of ARGILS in a shopping mall, an office building, and a lobby are 2.50 m, 2.0 m, and 1.6 m respectively. The accuracy of localization error within 2 m is 46.0%, 63.3%, and 72.0%. The maximum error is basically controlled within 5 m. Localization in shopping malls is suboptimal due to the complexity and variability of the lighting environments. Variations in lighting equipment, floor and glass reflections can compromise the extraction of key features from images, resulting in Localization errors that frequently exceed 4 m. This in turn negatively affects the overall accuracy of localization results.

Furthermore, we also test the localization accuracy of other methods in various scenarios, and the average error of each method is shown in Fig. 18. In all scenarios, ARGILS achieves the best performance, especially in the most complex shopping mall scenarios. The localization accuracy of PixLoc is susceptible to significant inaccuracies due to the limited number of captured images utilized to construct the 3D point cloud. Conversely, when a sufficient number of matching features are present in the user-supplied images, the localization accuracy is highly reliable. Therefore, the workload of the field survey plays a pivotal role in determining the effectiveness of PixLoc.

6) Time Consumption on Different Devices: We hope that ARGILS can be fully deployed on the mobile terminal to provide location services. To this end, we analyze the running time of ARGILS on OnePlus8, HUAWEI Mate20, and 360 N6 Pro. As shown in Fig. 19, we test the running time of different steps of ARGILS on different models, including feature extraction, feature retrieval, orthogonal localization, and complete operation of full localization. In the experiment, we set the feature database size to 1000 and used 1000 images for testing. On the well-configured OnePlus8, the single localization time is 105 ms, and in the poorly-configured 360N6 Pro equipped with Snapdragon 660 CPU (8 cores, 2.2 GHz) and 6 GB RAM, a single localization can also be completed within 332 ms, which proves that our system can run in real-time on most current mobile phones. In contrast, HAIL and Sextant extract SURF features to work, and the time cost is relatively large. SURF also has a large overhead in the matching stage, which limits their real-time application on the mobile phone. In addition, the orthogonal localization proposed by us does not significantly increase the time overhead of the system and has good practicality.

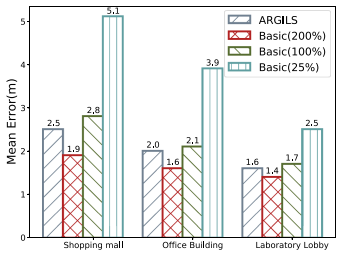


Fig. 20. Different site survey workload.

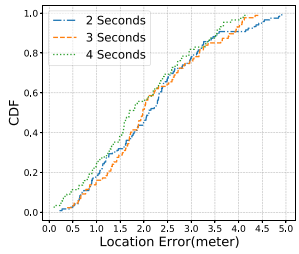


Fig. 21. Different scanning time.

7) Performance of the Site Survey Workload: We conducted a comparison of the site survey workload and localization accuracy between the orthogonal localization method and the basic localization method proposed in Section III-A across different scenarios. Basic(25%), Basic(100%), and Basic(200%) represent the performance of the basic localization method under different site survey densities. The percentage represents the proportional relationship of the number of features used in the feature database for localization. Basic(100%) and Basic(200%) are databases with the sampling positions and angles doubled on the basis of Basic(25%). Meanwhile, ARGILS represents the performance of orthogonal localization under the same survey density as Basic(25%), but ARGILS generates fingerprints for other locations to extend the database density. As shown in Fig. 20, the average error of orthogonal localization in each scene is slightly higher than Basic(200%), and the error is much lower than other schemes. This means that ARGILS is able to achieve the same accuracy as the basic localization method with 200% of the site survey workload while paying only 25% of the site survey workload.

8) Performance of Different Scanning Time: As the scan time increases, the more input features the system can acquire. In order to verify the effect of scan time on the system, we chose segments with different time lengths for testing. As shown in Fig. 21, the average error of the 4 s scan segment is 1.91 m, which is 11.1% higher than the 2 s duration, and as the duration increases, the maximum localization error of the system gradually decreases, which proves that our scanning method can reduce a lot of the error and provides a more reliable localization result.

9) Storage Consumption of Different Methods: Another factor that restricts the application of mobile application platforms is storage. We set a space of 10×10 meters to analyze the storage overhead of each method. Both HAIL and Sextant rely on landmarks for their work. The greater the number of landmarks, the greater the workload of collecting images and the higher the localization accuracy. We assume that the number of landmarks in this scenario is 5, and the results are shown in Table II. The data in the table are averaged from the data in the corresponding papers. In terms of storage consumption, ARGILS exhibits a lower rate than HAIL, Sextant, and PixLoc. Among these, PixLoc necessitates the storage of the matching relationship between features, resulting in a particularly high memory consumption. In contrast, GLN employs a classification-based localization network, obviating the necessity for a feature database for matching. However, an additional map structure and coordinates must be stored. In the area of $100m^2$, our method requires only 800 KB of storage, which is lower than other methods and can also meet the needs of mobile terminals in terms of storage.

TABLE II
STORAGE CONSUMPTION OF DIFFERENT METHODS

Methods	Label	Map Locations or Landmarks	Map images	Num of Features	Size
ARGILS	Camera Location	9	20	160	800KB
HAIL	Landmark Location	5	30	7840	4693KB
Sextant	Landmark Location	5	30	None	900KB
GLN	Camera Location	25	100	None	None
PixLoc	None	9	20	9088	264MB

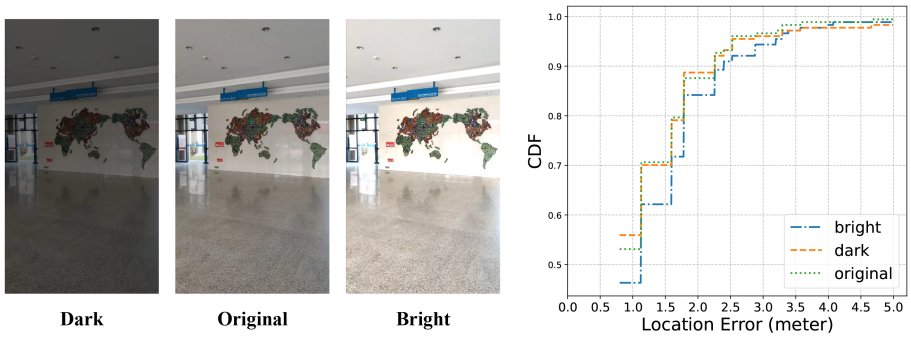
10) Performance Under Varying Light Conditions: We simulate changes in indoor lighting conditions by adjusting the image's contrast coefficient to represent both light and dark scenarios. In Fig. 22(a), we establish two control groups, one with strong lighting and the other with dim lighting, within the setting depicted in Fig. 14(c). The localization accuracy results are illustrated in Fig. 22(b). ARGILS is able to maintain better localization ability when facing the overall change of ambient light brightness. This resilience is attributed to the strong generalization capabilities of EfficientNet and DistanceGAN.

11) Ablation Study on Loss Functions: In this experiment, we evaluate the impact of different loss function combinations on model performance using displacement retrieval error as described in Section V-B-2. The following configurations were tested: 1) Adv Loss, 2) Adv + MSE Loss, and 3) Adv + MSE + Triplet Loss. The corresponding mean errors for these configurations were 0.9476 m, 2.7805 m, and 4.0220 m, respectively.

As shown in Fig. 23, the configuration with "Adv + MSE + Triplet Loss" achieved the lowest mean error. This combination enables the model to better learn feature representations by minimizing content discrepancies (via MSE) and enforcing stronger feature distance discrimination (via Triplet Loss).

The Adv Loss performs the worst, as adversarial loss only brings the generated feature distribution closer to the real features, without refining the feature distinctions across different distances. When MSE Loss is added, the generated features are forced to be closer to the ground-truth features, resulting in improved localization performance. Finally, adding Triplet Loss further enhances feature discrimination, making the differences between features at varying distances more pronounced, which in turn leads to better retrieval and localization accuracy.

The performance degradation observed in configurations without "Triplet Loss" (i.e., "Adv + MSE" and "Adv" alone) underscores the critical importance of learning distance-based



(a) Examples of different light intensities (b) Varying light conditions

Fig. 22. Performance under varying light conditions.

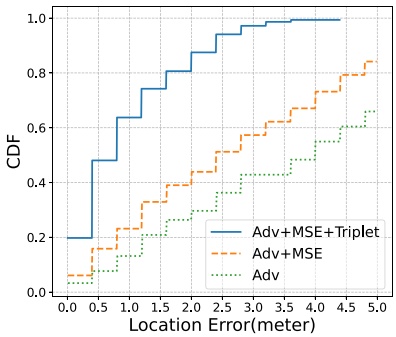


Fig. 23. Performance under different loss function combinations.

relationships. The experimental results confirm the effectiveness of the proposed loss function combination.

VIII. Conclusion

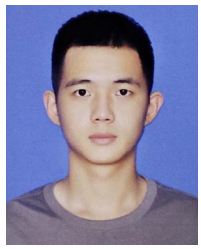
In this paper, we proposed a robust and effort-efficient indoor localization system that featured low computation and storage costs for clients, and low site survey costs on the server side, thus it could be deployed on a large scale for online map providers, e.g. Google map, Baidu Map and etc. Basically, our system explores the possibility of efficiently generating the localization features from a sparse collected feature database and proposes a distance-constraint generative neural network to do so. Then, we propose an orthogonal feature decomposition and localization framework, which not only greatly reduces the costs of feature generation in arbitrary angles, but also reduces the localization computation and storage cost in the clients. In addition, a scanning mechanism is proposed as an extension to achieve on-device video localization, making our method more practical, eliminating the impact of various obstacles on localization, and improving the robustness of the system. Real implementation experiments in three different scenarios have been conducted. Compared with the state-of-the-art methods, our system can obtain an average localization accuracy of 2.5 m, which is 48% better than the baselines, and only requires 25% of the site survey density. In addition, all the operations could be finished in less than 400 ms on mobile devices.

In future investigations, it is important to implement robust and efficient localization for heterogeneous mobile devices and varying focal lengths. For example, using monocular vision depth estimation to directly obtain device location information. Localization models applicable to zoom scenarios need to be investigated, enabling users to freely choose their focal lengths and improving user experience and system pervasiveness.

References

[1] S. Hayward, K. van Lopik, C. Hinde, and A. West, "A survey of indoor location technologies, techniques and applications in industry," *Internet Things*, vol. 20, 2022, Art. no. 100608.
[2] H. Obeidat, W. Shuaieb, O. Obeidat, and R. Abd-Alhameed, "A review of indoor localization techniques and wireless technologies," *Wireless Pers. Commun.*, vol. 119, pp. 289–327, 2021.
[3] S. Narayana et al., "LOCI: Privacy-aware, device-free, low-power localization of multiple persons using IR sensors," in *Proc. 19th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2020, pp. 121–132.
[4] P. Du and N. Bulusu, "An automated AR-based annotation tool for indoor navigation for visually impaired people," in *Proc. 23rd Int. ACM SIGACCESS Conf. Comput. Accessibility*, 2021, pp. 1–4.
[5] P. Du and N. Bulusu, "Indoor navigation for visually impaired people with vertex colored graphs," in *Proc. 20th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2022, pp. 587–588.
[6] A. Conti et al., "Location awareness in beyond 5G networks," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 22–27, Nov. 2021.
[7] S. Dwivedi et al., "Positioning in 5G networks," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 38–44, Nov. 2021.
[8] H. Rizk, H. Yamaguchi, M. Youssef, and T. Higashino, "Laser range scanners for enabling zero-overhead WiFi-based indoor localization system," *ACM Trans. Spatial Algorithms Syst.*, vol. 9, no. 1, Jan. 2023, Art. no. 4, doi: 10.1145/3539659.
[9] L. Bai, F. Ciravegna, R. Bond, and M. Mulvenna, "A low cost indoor positioning system using Bluetooth low energy," *IEEE Access*, vol. 8, pp. 136858–136871, 2020.
[10] G. Maus, H. Pörner, R. Ahrens, and D. Brückmann, "A phase normalization scheme for angle of arrival based Bluetooth indoor localization," in *Proc. IEEE 65th Int. Midwest Symp. Circuits Syst.*, 2022, pp. 1–5.
[11] A. Motroni, A. Buffi, and P. Nepa, "A survey on indoor vehicle localization through RFID technology," *IEEE Access*, vol. 9, pp. 17921–17942, 2021.
[12] Y. Zheng, J. Liu, M. Sheng, and C. Zhou, *Exploiting Fingerprint Correlation for Fingerprint-Based Indoor Localization: A Deep Learning-Based Approach*. Cham, Switzerland: Springer, 2023, pp. 201–237, doi: 10.1007/978-3-031-26712-3_9.
[13] V. Kachurka et al., "WeCo-SLAM: Wearable cooperative SLAM system for real-time indoor localization under challenging conditions," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5122–5132, Mar. 2022.
[14] T. Sattler, B. Leibe, and L. Kobbelt, *Exploiting Spatial and Co-Visibility Relations for Image-Based Localization*, Cham, Switzerland: Springer, 2016, pp. 165–187, doi: 10.1007/978-3-319-25781-5_9.

[15] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2019, pp. 12716–12725.
[16] M. Li, N. Liu, Q. Niu, C. Liu, S.-H. G. Chan, and C. Gao, "SweepLoc: Automatic video-based indoor localization by camera sweeping," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 3, pp. 1–25, 2018.
[17] R. Gao et al., "Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment," *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 460–474, Feb. 2016.
[18] Q. Niu, M. Li, S. He, C. Gao, S.-H. Gary Chan, and X. Luo, "Resource-efficient and automated image-based indoor localization," *ACM Trans. Sensor Netw.*, vol. 15, no. 2, pp. 1–31, 2019.
[19] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[20] X. Ge and Z. Qu, "Optimization WIFI indoor positioning KNN algorithm location-based fingerprint," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci.*, 2016, pp. 135–137, doi: 10.1109/icsess.2016.7883033.
[21] A. H. Salamah, M. Tamazin, M. A. Sharkas, and M. Khedr, "An enhanced WiFi indoor localization system based on machine learning," in *Proc. 2016 Int. Conf. Indoor Positioning Indoor Navigation*, 2016, pp. 1–8.
[22] S. Sadowski, P. Spachos, and K. N. Plataniotis, "Memoryless techniques and wireless technologies for indoor localization with the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10996–11005, Nov. 2020.
[23] A. A. Abdallah, K. Shamaei, and Z. M. Kassas, "Assessing real 5G signals for opportunistic navigation," in *Proc. 33rd Int. Tech. Meeting Satell. Division Inst. Navigation*, 2020, pp. 2548–2559.
[24] Y. Ruan, L. Chen, X. Zhou, G. Guo, and R. Chen, "Hi-loc: Hybrid indoor localization via enhanced 5G NR CSI," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5502415.
[25] V. Bianchi, P. Ciampolini, and I. De Munari, "RSSI-Based indoor localization and identification for ZigBee wireless sensor networks in smart homes," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 2, pp. 566–575, Feb. 2019, doi: 10.1109/tim.2018.2851675.
[26] P. Spachos and K. Plataniotis, "BLE beacons in the smart city: Applications, challenges, and research opportunities," *IEEE Internet Things Mag.*, vol. 3, no. 1, pp. 14–18, Mar. 2020, doi: 10.1109/iotm.0001.1900073.
[27] J. Rezazadeh, R. Subramanian, K. Sandrasegaran, X. Kong, M. Moradi, and F. Khodamoradi, "Novel iBeacon placement for indoor positioning in IoT," *IEEE Sensors J.*, vol. 18, no. 24, pp. 10240–10247, Dec. 2018.
[28] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Automat. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.
[29] J. Xia and J. Gong, "Precise indoor localization with 3D facility scan data," *Comput.-Aided Civil Infrastructure Eng.*, vol. 37, no. 10, pp. 1243–1259, 2022.
[30] M. Nimura, K. Kanai, and J. Katto, "Accuracy evaluations of real-time LiDAR-based indoor localization system," in *Proc. 2023 IEEE Int. Conf. Consum. Electron.*, 2023, pp. 1–5.
[31] A. Abozeid, A. I. Taloba, R. M. A. El-Aziz, A. F. Alwaghid, M. Salem, and A. Elhadad, "An efficient indoor localization based on deep attention learning model," *Comput. Syst. Sci. Eng.*, vol. 46, no. 2, pp. 2637–2650, 2023.
[32] I. Ha, H. Kim, S. Park, and H. Kim, "Image retrieval using BIM and features from pretrained VGG network for indoor localization," *Building Environ.*, vol. 140, pp. 23–31, 2018.
[33] E. Shahid and Q. A. Arain, "Indoor positioning: An image-based crowdsource machine learning approach," *Multimedia Tools Appl.*, vol. 80, no. 17, pp. 26213–26235, 2021.
[34] T. Kim, J. H. Lee, B. Shin, C. Yu, H. Kyung, and T. Lee, "Very fast fingerprinting DB construction for precise indoor localization," in *Proc. Int. Conf. Electron. Inf. Commun.*, 2022, pp. 1–4.
[35] Q. Niu, K. Zhu, S. He, S. Cen, S.-H. G. Chan, and N. Liu, "VILL: Toward efficient and automatic visual landmark labeling," *ACM Trans. Sensor Netw.*, vol. 19, no. 4, pp. 1–25, 2023.
[36] Y. Li, R. H. Kambhamettu, Y. Hu, and R. Zhang, "ImPos: An image-based indoor positioning system," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf.*, 2022, pp. 144–150.
[37] R. Gao and Y. Zhao, "A lightweight indoor location method based on image fingerprint," in *Proc. Int. Conf. Artif. Intell. Secur.*, 2020, pp. 763–774.
[38] M. Tang, Y. Zhao, Q. Ma, J. Hao, and B. Chen, "CrowdLoc: Robust image indoor localization with edge-assisted crowdsensing," *J. Syst. Archit.*, vol. 131, 2022, Art. no. 102732.
[39] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[40] W. Wang, L. Niu, J. Zhang, X. Yang, and L. Zhang, "Dual-path image inpainting with auxiliary GAN inversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11421–11430.
[41] Y. Shi, L. Han, L. Han, S. Chang, T. Hu, and D. Dancey, "A latent encoder coupled generative adversarial network (LE-GAN) for efficient hyperspectral image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5534819.
[42] J. Liang et al., "Sketch guided and progressive growing GAN for realistic and editable ultrasound image synthesis," *Med. Image Anal.*, vol. 79, 2022, Art. no. 102461.
[43] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[44] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10684–10695.
[45] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.
[46] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.
[47] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.
[48] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv: 1703.07737*.
[49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[50] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*.
[51] M. S. Rad, B. Bozorgtabar, U.-V. Marti, M. Basler, H. K. Ekenel, and J.-P. Thiran, "Srobb: Targeted perceptual loss for single image super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2710–2719.
[52] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 694–711.
[53] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
[54] N. Krawetz, "Perceptual hash algorithm," 2011. [Online]. Available: http://www.hackerfactor.com/blog/?/archives/432-Looks-Like-It.html
[55] M.-J. Chiou, Z. Liu, Y. Yin, A.-A. Liu, and R. Zimmermann, "Zero-shot multi-view indoor localization via graph location networks," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 3431–3440.
[56] P.-E. Sarlin et al., "Back to the feature: Learning robust camera localization from pixels to pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3247–3257.
[57] M. Tyszkiewicz, P. Fua, and E. Trulls, "Disk: Learning local features with policy gradient," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 14254–14265.
[58] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local feature matching at light speed," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 17627–17638.
[59] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. image Comput. Comput.- Assist. Intervention*, 2015, pp. 234–241.



Zhenhan Zhu received the BS degree in computer science from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China, in 2021. He is currently working toward the PhD degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include indoor localization, autonomous driving, cooperative perception, and sensor fusion.

Yanchao Zhao (Member, IEEE) received the BS and PhD degrees in computer science from Nanjing University, Nanjing, China, in 2007 and 2015, respectively. He is currently a professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing. In 2011, he was a visiting student with the Department of Computer and Information Sciences, Temple University, Philadelphia, Pennsylvania. His research interests include wireless network, mobile computing, edge computing, and device-free sensing. He has published more than 60 papers in prestigious conferences and journals such as *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Dependable and Secure Computing*, IEEE INFOCOM, (*International Journal of Human-Computer Studies*, *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Sensor Networks*, and AAAI. He is the recipient of the Jiangsu Provincial Excellent Youth Fund. Currently, he serves the organizing committees of several high-profile conferences including IEEE INFOCOM, ICCCN, ICPADS, and MASS. He is also a reviewer for top-tier international journals such as *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Network Magazine*, *IEEE Communication Magazine*, and *IEEE Internet of Things Journal*. He is honored with the Jiangsu Provincial Science and Technology Award for his contributions to the field.

Maoxing Tang received the ME degree from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2020. His research interests include edge computing, distributed systems, and computer vision. He has published in the Journal of Systems Architecture and has contributed to various research projects in his field.

Yanling Bu (Member, IEEE) received the PhD degree in computer science from the Nanjing University, China, in 2022. She is currently an associate researcher with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. Her research interests include the Internet of Things (IoT), mobile sensing, and RFID systems.

Hao Han (Member, IEEE) received the BS degree in computer science and technology from Nanjing University, Nanjing, China, in 2005, and the PhD degree in computer science from the College of William and Mary, Williamsburg, VA, USA, in 2014. He is currently a professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His research interests include system and software security against various types of threats.