




## Article

# Multi-Floor Indoor Localization Based on Multi-Modal Sensors

Guangbing Zhou <sup>1,2,3</sup> , Shugong Xu <sup>1,\*</sup>, Shunqing Zhang <sup>1</sup> , Yu Wang <sup>1</sup> and Chenlu Xiang <sup>1</sup> 

<sup>1</sup> School of Information and Communication Engineering, Shanghai University, Shanghai 200444, China; zhou020@shu.edu.cn (G.Z.); shunqing@shu.edu.cn (S.Z.); wangyu042@shu.edu.cn (Y.W.); xcl@shu.edu.cn (C.X.)

<sup>2</sup> Institute of Intelligent Manufacturing, Guangdong Academy of Sciences, Guangzhou 510070, China

<sup>3</sup> South China Robotics Innovation Research Institute, Foshan 528300, China

\* Correspondence: shugong@shu.edu.cn

**Abstract:** High-precision indoor localization is growing extremely quickly, especially for multi-floor scenarios. The data on existing indoor positioning schemes, mainly, come from wireless, visual, or lidar means, which are limited to a single sensor. With the massive deployment of WiFi access points and low-cost cameras, it is possible to combine the above three methods to achieve more accurate, complete, and reliable location results. However, the existing SLAM rapidly advances, so hybrid visual and wireless approaches take advantage of this, in a straightforward manner, without exploring their interactions. In this paper, a high-precision multi-floor indoor positioning method, based on vision, wireless signal characteristics, and lidar is proposed. In the joint scheme, we, first, use the positioning data output in lidar SLAM as the theoretical reference position for visual images; then, use a WiFi signal to estimate the rough area, with likelihood probability; and, finally, use the visual image to fine-tune the floor-estimation and location results. Based on the numerical results, we show that the proposed joint localization scheme can achieve 0.62 m of 3D localization accuracy, on average, and a 1.24-m MSE for two-dimensional tracking trajectories, with an estimation accuracy for the floor equal to 89.22%. Meanwhile, the localization process takes less than 0.25 s, which is of great importance for practical implementation.

**Keywords:** indoor localization; visual-based localization; WiFi signal; multi-floor; lidar SLAM



**Citation:** Zhou, G.; Xu, S.; Zhang, S.; Wang, Y.; Xiang, C.

Multi-Floor Indoor Localization Based on Multi-Modal Sensors.

*Sensors* **2022**, *22*, 4162. <https://doi.org/10.3390/s22114162>

Academic Editors: Adrian Kliks, Pawel Sroka, Cynthia Hood, Nikos Dimitriou and Marcin Dryjanski

Received: 22 April 2022

Accepted: 27 May 2022

Published: 30 May 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Location-based services (LBS) [1] have been identified as a promising technology, with which to merge social daily lives with geographic information, which triggers a paradigm shift in shopping, entertainment, and other business activities. Typical localization applications, including express delivery, turn-by-turn navigation, and travel arrangement, have, dramatically, changed people's lives in outdoor environments, in recent years. In terms of contributing to social civilization, the improved Tiny-Yolov3 model [2] can help visually impaired people to navigate indoors and outdoors. Visually impaired people wearing Google Glass can be guided to target locations inside the building, using its floor plan [3]. A more promising area for the future lies in the indoor scenario, where indoor localization and navigation, using smartphones or Internet of things devices, have been considered important applications for this rapidly growing market. As reported in [4], the global LBS market is expected to reach a total of USD 226 billion by 2022, where 18% of the market share belongs to indoor LBS applications, with a more than 20% compound annual growth rate (CAGR).

The Global Navigation Satellite System (GNSS) can provide precise positioning services, for moving objects on the ground. Xu L. et al. [5] used GNSS to locate vehicles that were moving fast on expressways, achieving a positioning accuracy of 4–6 m. With the aid of Inertial Navigation Systems (INS), GNSS localization accuracy can be provided [6]. In the indoor scenario, Bluetooth low-energy (BLE) [7], Ultra-Wide Band (UWB) [8], 3rd-Generation

Partnership Project (3GPP) long-term evolution (LTE)/5G, and WiFi signals [9,10] are adopted, instead, since the GNSS signals suffer from building shielding. Indoor-localization technology, based on static cameras, mobile cameras, wireless, IMU, and other sensor components has received increasing attention from scholars. As cameras become more affordable, as well as their integration with smart devices, visual indoor positioning is becoming popular. It is, increasingly, widely applicable in the fields of auxiliary equipment, autonomous robots, monitoring, and positioning [11]. However, reaching the same level of localization accuracy is quite challenging, due to the complexity and variability of the indoor environment, so the large-scale application of indoor-location solutions has, yet, to be realized.

With the massive deployment of low-cost digital cameras in smart entities, a potential approach to further improve the localization accuracy is to incorporate visual information, which is often referred to as visual-based localization (VBL) [12]. Simultaneous Localization and Mapping (SLAM) achieves simultaneous positioning and map construction, based on self-perception [13].

SLAM is a self-localization technology in intelligent mobile devices. It realizes its own localization and environment mapping, through a lidar-sensing environment. The lidar SLAM localization scheme is a mature theory and technology, and [14] has carried out a complete mathematical deduction and verification of the SLAM localization scheme, from the perspective of probability theory. The commonly used 2D lidar SLAM methods are GMapping and Cartographer. Gmapping adds lidar data to the proposed distribution on the basis of particle filtering positioning and puts forward the effective particle number, as the resampling constraint [15]. Cartographer is Google's open-source indoor location technology for mobile devices, with a front end for scanning and matching as well as a back end for closed-loop detection and optimization [16]. These SLAM localization schemes can be used as a theoretical reference for wireless and visual localization schemes because they can achieve single-floor millimeter-level indoor localization.

Since the signal feature-based localization (SFBL) and the VBL schemes share similar design philosophy, e.g., to establish databases offline and perform pattern matching online, a natural extension is to, jointly, minimize the localization errors, via combining databases and matching algorithms in a brute-force manner, and the localization performance can be improved to the meter level [17] or sub-meter level, with the help of high-cost lidar. A smarter approach is to decouple SFBL and VBL processing, in a hierarchical way [18], e.g., to perform coarse-grained localization using SFBL and fine-tune the intermediate positions using VBL, and the resultant errors can be reduced to less than 2 m. This approach requires significant manpower for image-database generation and online-processing complexity. To make this more practical, ref. [19] proposes projecting the query images into a two-dimensional floor plan for pattern-matching, and [20] utilizes the special 'EXIT' signs, to reduce the processing complexities.

The above hybrid SFBL and VBL approaches provide a promising direction for high-precision indoor localization, by utilizing the corresponding advantages, in a separate manner. However, it fails to explore the interactions between the two schemes, especially when the localization task is mixed with multi-floor scenarios. In this paper, we propose a joint visual and wireless signal feature-based solution for high-precision multi-floor indoor localization. As shown later, our proposed scheme can utilize the coarse localization results from SFBL, to assist the later VBL procedures, which, eventually, simultaneously reduces the localization errors as well as the processing complexities. The main contributions are listed below.

- **Signal-Feature-Assisted VBL.** The conventional hybrid SFBL and VBL scheme, simply, selects some candidate regions using SFBL, to restrict the processing complexities in the VBL stage. If we regard the region index identification as a 'hard decision', a more reasonable scheme is to consider a 'soft decision', instead. Hence, in this paper, we propose a joint visual and wireless signal feature-based localization (JVWL),

by considering the likelihood distribution of potential positions, which eventually helps to improve the localization accuracy.

- **Single-Floor to Multi-Floor Extension.** Different from the single-floor cases, where the horizontal dimension is neglected, multi-floor localization raises many challenging issues, regarding the existing localization mechanisms. For example, the floor structure is more or less the same for different floors, which is generally difficult for VBL. Therefore, we utilize a multi-domain translation architecture, on top of signal-feature-based coarse localization, to learn the minor changes in different floor environments.
- **Low-Complexity Few-Shot Learning.** In addition, we study a low-complexity dataset construction mechanism and, numerically, analyze the relationship between the localization accuracy and the number of sampling images. Through some numerical results, we show that high-accuracy localization results can be achieved with low-complexity few-shot learning methods.

The rest of the paper is organized, as follows. Section 2 analyzes the related work in learning-based visual localization, few-shot learning, and fusion-based localization. The entire localization system model is described in Section 3, and the proposed joint localization scheme for the multi-floor scene is discussed in Section 4. In Section 5, we present our experimental results, and the concluding remarks are provided in Section 6.

## 2. Related Works

### 2.1. Learning-Based Visual Localization

Learning-based visual localization systems perform as image regression, with a large number of datasets. For example, PoseNet has been proposed in [21], which was recognized as the first successful end-to-end pre-trained deep CNNs approach, for 6-DoF pose regression. Long-Short-Term Memory (LSTM) units have been discussed in [22], to avoid overfitting issues in the traditional CNNs approaches. Moreover, the feature-fusion mechanism was later incorporated in NetVLAD [23], with multiple-CNN-based feature extraction.

One of the most critical issues for CNN-based approaches is the limited generalization capability, e.g., the related algorithms can, hardly, adapt to the changing environment [24]. To address this, generative adversarial networks (GANs) [25] have been proposed in the conventional computer-vision tasks, including CycleGAN [26] and ComboGAN [27], which are shown to achieve superior feature-extraction abilities, compared to conventional CNNs. Motivated by this fact, ToDayGAN was proposed in [28], to use GAN-based architecture for localization, which is shown to achieve a five-meter accuracy, with a 52.9% probability, for both daytime and nighttime. Based on the ComboGAN architecture, a novel domain-invariant-feature-learning approach has been proposed in [29], and the resultant probability can be improved to 87.2%. This is, partially, because ComboGAN's flexible combination of encoder-decoder pairs can, effectively, learn and extract domain-invariant features across multiple image domains.

Although the learning-based visual localization can outperform the conventional feature-extraction-based scheme, such as speeded-up robust features (SURF), it usually requires a huge amount of high-quality training data, e.g., hundreds of thousands of images taken from different positions, under diverse conditions, to guarantee robustness, as mentioned in [28]. To reduce the complexities in collecting high-quality images, an unpaired image-to-image translation scheme was, also, proposed in [27].

### 2.2. Few-Shot Learning

Few-shot learning is another promising approach, to reduce the requirement for high-quality images. Depending on the number of sampling images, few-shot learning [30] can, roughly, be categorized into two types, including meta-learning-based methods and transfer-learning-based methods.

Meta-learning-based methods, also known as learning to learn, aim to learn a generalized model that can be adapted to infer new classes, using only few-shot training samples.

For instance, an optimization framework, to update hyper-parameters in neural networks, between the meta-training and meta-testing stages, has been proposed in [31], and an abstracted learning metric, to measure the similarities between training images and test images, has been exploited in [32]. All the above meta-learning-based methods require only a limited number of sampling data, with the episodic training strategy.

Transfer-learning-based methods [33], however, apply conventional approaches to pre-train a generalized model from the basic dataset, and are adapted to some specific tasks, with few-shot training samples. Since the classifier weights in the neural networks are critical to the adaptation, the existing transfer-learning-based mechanisms focus on analyzing the feature embeddings of few-shot samples. As an example, the mean vectors of feature embeddings have been utilized in [34], and a generalized mapping function from the feature embeddings to the weights of classifiers, has been studied in [35]. Meanwhile, an attention module, to dynamically predict the weights of classifiers, has been proposed in [36], which outperforms meta-learning-based methods, as illustrated in [37].

### 2.3. Fusion-Based Localization

As mentioned before, both the SFBL methods and the VBL schemes have pros and cons, which have not been widely implemented to date [38]. In order to be more cost-efficient, a hybrid SFBL and VBL scheme has been proposed in [39], where context information from visual data and signal features from WiFi data are fused together, to provide highly accurate localization results. A deep-fusion mechanism for wireless signals and visual images is proposed in [40], which incorporates the wavelet-transformed-signal features and a scale-invariant feature, from sampled images. The famous LASSO algorithm is adopted, to achieve 0.83-m localization accuracy. In [41], the extended naive Bayes and SURF algorithms are utilized, to extract the features of WiFi signals and visual images, respectively. A particle-filter-based fusion scheme is, then, proposed for localization estimation, which achieves 1.9-m accuracy. The above particle-filter-based fusion framework has been extended to incorporate INS signals in [17], where improved two-dimensional convolutions are applied, to generate RGB-WM image features. According to the experimental results, this fusion-based localization scheme can achieve less than 1.23-m accuracy.

The aforementioned fusion-based localization schemes can achieve superior localization accuracy, in general. However, the computational and storage costs of image processing are much more significant than the SFBL. In addition, if we consider the feature extraction and fusion complexities, the real-time fusion-based localization scheme cannot be installed on mobile devices.

### 2.4. Lidar-Slam Localization

As shown in Figure 1, Cartographer is a scan-matching algorithm based on graph optimization. Cartographer incorporates laser, odometer, IMU and other multi-sensor data. Local SLAM (Frontend): laser data through voxel filtering, odometer and IMU data through track calculation. The Scan data are matched with the latest Submap, so the Scan data of this frame are inserted into the optimal position on the Submap. The Submap is updated, as new data frames are inserted. A certain amount of Scan data forms a Submap. If no new Scan is inserted into the Submap, the Submap is considered to have been created, and the next Submap is created, according to the step size. Global SLAM (Backend): each Submap has cumulative errors. The cumulative error is optimized by loop detection. If the current Scan and each created Submap is close enough, loopback detection is performed. To reduce the computation, the branch and bound method is used to search [42].

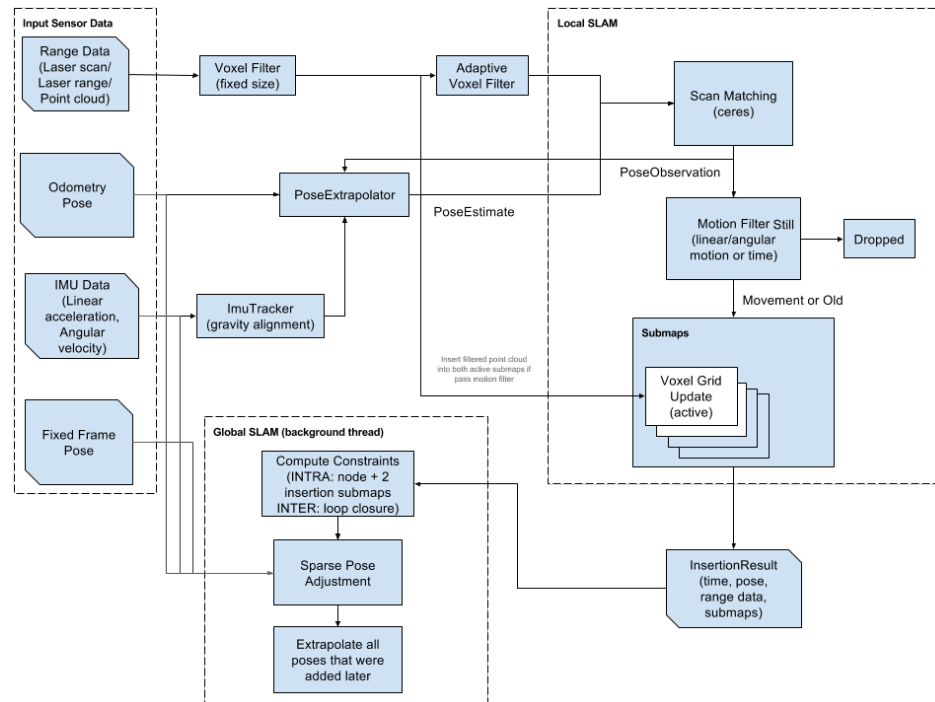


Figure 1. Block diagram of Cartographer scheme.

### 3. System Model

In this section, we introduce the overall procedures of the proposed JVWL scheme and discuss the construction of databases in what follows.

#### 3.1. Overall Description

As shown in Figure 2, the proposed JVWL scheme first collects  $N_s$  sample received signal strength indications (RSSIs) from  $N_{AP}$  WiFi access points, and  $N_p$  query images with  $N_w \times N_l$  pixels and  $N_{RGB}$  color channels from on-device cameras, where the corresponding observations are denoted as  $\mathbf{R}(\mathcal{L}) \in \mathbb{R}^{N_s \times N_{AP}}$  and  $\mathbf{I}(\mathcal{L}) \in \mathbb{Z}^{N_p \times N_w \times N_l \times N_{RGB}}$  for a given location  $\mathcal{L}$ , respectively. RSSIs of  $N_{RP}$  reference points (RPs) are collected offline to construct WiFi fingerprint database,  $\mathcal{DB}_W$ , which consists of  $N_{RP}$  RP locations,  $\{\mathcal{L}_{RP}^i\}$ , and the measured RSSIs,  $\{\mathbf{R}(\mathcal{L}_{RP}^i)\}$ . The image database,  $\mathcal{DB}_I$ , is constructed in a similar manner, which contains the location  $\mathcal{L}_{\mathcal{I}}$ , and the associated images,  $\{\mathbf{I}(\mathcal{L}_{\mathcal{I}})\}$ .

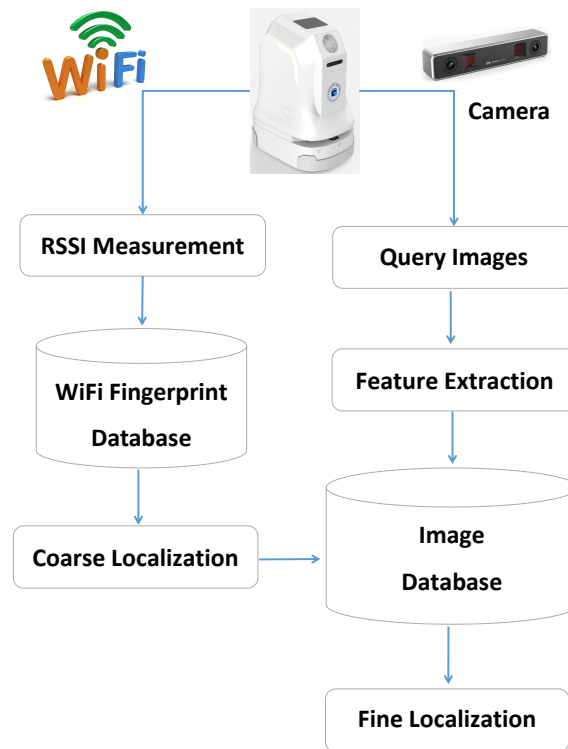
The entire localization procedures consist of a WiFi RSSI-based coarse localization and a visual-image-based fine localization (kindly note that the above RSSI and visual information can easily be obtained by the sensors on the mobile robots, such as WiFi receivers and cameras.), as explained below.

- *Coarse Localization*  $f(\cdot)$  In the coarse localization, the proposed JVWL scheme first computes the likelihood probability with respect to (w.r.t.)  $N_{RP}$  RPs, i.e.,  $\hat{\mathbf{p}}_{RP}(\mathcal{L}) = [\hat{p}_{RP}^1(\mathcal{L}), \dots, \hat{p}_{RP}^i(\mathcal{L}), \dots, \hat{p}_{RP}^{N_{RP}}(\mathcal{L})]$ , by inquiring the WiFi fingerprint database,  $\mathcal{DB}_W$ . By comparison with the observed WiFi RSSIs, the likelihood probability w.r.t. the  $i^{th}$  RP can be obtained via a standard support vector machine (SVM) scheme, which has proven to be effective in fingerprint classification tasks [43], e.g.,

$$\hat{p}_{RP}^i(\mathcal{L}) = f_1(\mathbf{R}(\mathcal{L}), \mathcal{DB}_W). \quad (1)$$

To reduce the searching complexity of the latter fine localization, we partition the target areas into  $N_A$  consecutive areas based on  $N_{RP}$  RPs,  $\{\mathcal{A}^j(\{\mathcal{L}_{RP}^i\})\}$ , as shown in Figure 3. The likelihood probabilities of  $N_A$  areas are, as follows:

$$\hat{\mathbf{p}}_{\mathcal{A}}(\mathcal{L}) = [\hat{p}_{\mathcal{A}}^1(\mathcal{L}), \dots, \hat{p}_{\mathcal{A}}^j(\mathcal{L}), \dots, \hat{p}_{\mathcal{A}}^{N_A}(\mathcal{L})] \quad (2)$$



**Figure 2.** The overall procedures of the proposed JVWL scheme. It contains two stages, including WiFi RSSI-based coarse localization and visual-image-based fine localization.

This can be calculated by summing over the likelihood probabilities of RPs, where each element  $\hat{p}_{\mathcal{A}}^j(\mathcal{L})$  is given by,

$$\hat{p}_{\mathcal{A}}^j(\mathcal{L}) = f_2(\hat{\mathbf{p}}_{RP}(\mathcal{L})) = \sum_{i \in \mathcal{A}^i(\{\mathcal{L}_{RP}^i\})} \hat{p}_{RP}^i(\mathcal{L}). \quad (3)$$

The coarse localization results are, thus, given by selecting  $J^*$  most possible areas according to the likelihood probabilities,  $\hat{\mathbf{p}}_{\mathcal{A}}(\mathcal{L})$ . Mathematically, if we denote  $\Omega_j^*(\mathcal{L})$  and  $\overline{\Omega_j^*(\mathcal{L})}$  to be the index set of selected areas and its complementary set, the candidate localization area  $\mathcal{A}^*(\mathcal{L})$  and the corresponding likelihood probability  $\hat{\mathbf{p}}_{\mathcal{A}^*}(\mathcal{L})$  can be expressed as

$$\mathcal{A}^*(\mathcal{L}) = \bigcup_{j \in \Omega_j^*(\mathcal{L})} \mathcal{A}^j(\{\mathcal{L}_{RP}^i\}), \quad (4)$$

$$\hat{\mathbf{p}}_{\mathcal{A}^*}(\mathcal{L}) = \{\hat{p}_{\mathcal{A}}^j(\mathcal{L})\}, \forall j \in \Omega_j^*, \quad (5)$$

where  $\hat{p}_{\mathcal{A}}^j(\mathcal{L}) \geq \hat{p}_{\mathcal{A}}^{j'}(\mathcal{L})$  for any  $j \in \Omega_j^*(\mathcal{L})$  and  $j' \in \overline{\Omega_j^*(\mathcal{L})}$ , and the cardinality of  $\Omega_j^*(\mathcal{L})$  is  $J^*$ . By cascading (1)–(5), we denote the entire coarse localization process as

$$(\mathcal{A}^*(\mathcal{L}), \hat{\mathbf{p}}_{\mathcal{A}^*}(\mathcal{L})) = f(\mathbf{R}(\mathcal{L}), \mathcal{DB}_W, \{\mathcal{A}^i(\{\mathcal{L}_{RP}^i\})\}).$$

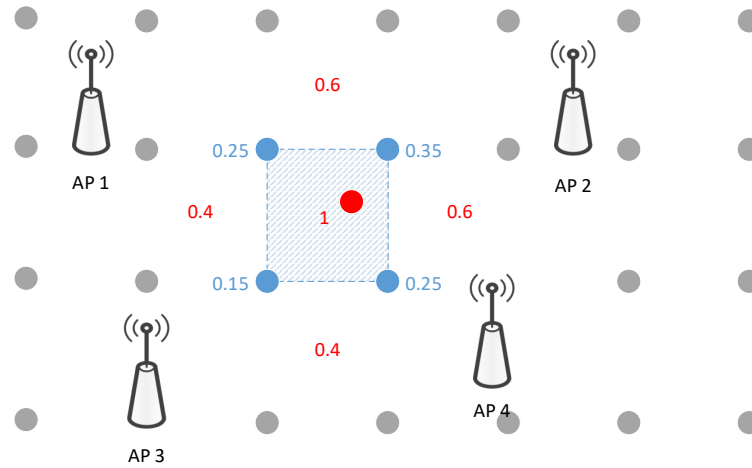
- *Fine Localization*  $g(\cdot)$  In the fine localization, the proposed JVWL scheme maps the  $N_p$  query images,  $\mathbf{I}(\mathcal{L})$ , to the estimated location,  $\hat{\mathcal{L}} \in \mathcal{A}^*(\mathcal{L})$ , according to the image database  $\mathcal{DB}_I$ . To control the searching complexity, we only use a subset of the entire image database in the practical deployment, e.g.,



$$\mathcal{DB}_I(\mathcal{A}^*(\mathcal{L})) \triangleq \{\mathbf{I}(\mathcal{L}_I), \forall \mathcal{L}_I \in \mathcal{A}^*(\mathcal{L})\} \subset \mathcal{DB}_I, \quad (6)$$

and the mathematical expression of the fine localization process is given by

$$\hat{\mathcal{L}} = g(\mathbf{I}(\mathcal{L}), \mathcal{DB}_I(\mathcal{A}^*(\mathcal{L}))). \quad (7)$$



**Figure 3.** The illustration of coarse localization. The nearest  $N_p$  RPs are selected by the KNN algorithm and form a circle area, which is the coarse localization area. The gray dots and red dots are RP positions and test positions, respectively.

### 3.2. Database Construction

In order to construct the databases  $\mathcal{DB}_W$  and  $\mathcal{DB}_I$ , a site survey of WiFi RSSI fingerprints and camera images is conducted with  $N_{RP}$  and  $|\mathcal{L}_I|$  positions, respectively. To make a more reliable database,  $N_W$  rounds of RSSIs collections and  $N_{IR}$  rounds of images are performed to construct  $\mathcal{DB}_W$  and  $\mathcal{DB}_I$ , e.g.,  $\mathcal{DB}_W = \{(\mathbf{R}^{N_W}(\mathcal{L}_{RP}^i), \mathcal{L}_{RP}^i)\}$  and  $\mathcal{DB}_I = \{(\mathbf{I}^{N_I}(\mathcal{L}_I), \mathcal{L}_I)\}$ , respectively. Since  $\mathcal{DB}_I(\mathcal{A}^*(\mathcal{L}))$  is equal to

$$\cup_{j \in \Omega_j^*(\mathcal{L})} \mathcal{DB}_I(\mathcal{A}^j(\{\mathcal{L}_{RP}^i\})) \quad (8)$$

we can partition the image database  $\mathcal{DB}_I$  into  $N_A$  parts in the offline stage, e.g.,

$$\{\mathcal{DB}_I(\mathcal{A}^j(\{\mathcal{L}_{RP}^i\}))\} \quad (9)$$

and efficiently construct  $\mathcal{DB}_I(\mathcal{A}^*(\mathcal{L}))$  in the online stage. For the convenience of data collection and future update, a mobile robot equipped with WiFi, camera, lidar and IMU sensors are used to construct  $\mathcal{DB}_W$  and  $\mathcal{DB}_I$ , with corresponding ground-truth positions. More implementation details are presented in Section 5.

## 4. Proposed Multi-Floor Scheme

In this section, we describe the problem formulation and the proposed multi-floor-localization scheme. To be more specific, we propose a joint-optimization framework for visual and wireless localization, based on which a novel neural network structure and loss function are, then, presented.

### 4.1. Problem Formulation

To obtain a reliable localization error performance, we introduce the subscript  $k$  to the ground-truth and estimated locations and formulate the multi-floor localization problem, as follows.

**Problem 1** (Multi-Floor Localization).

$$\underset{\hat{\mathcal{L}}_k}{\text{minimize}} \quad \frac{1}{K} \sum_{k=1}^K \min_{j \in \Omega_j^*(\hat{\mathcal{L}}_k)} \frac{\|\hat{\mathcal{L}}_k - \mathcal{L}_k\|_2^2}{\hat{p}_{\mathcal{A}}^j(\hat{\mathcal{L}}_k)} \quad (10)$$

$$\text{subject to} \quad (\mathcal{A}^*(\hat{\mathcal{L}}_k), \hat{\mathbf{p}}_{\mathcal{A}^*}(\hat{\mathcal{L}}_k)) = f(\mathbf{R}(\hat{\mathcal{L}}_k), \mathcal{DB}_W, \{\mathcal{A}^i(\{\mathcal{L}_{RP}^i\})\}), \quad (11)$$

$$\hat{\mathcal{L}}_k, \hat{\mathcal{C}}_k = g(\mathbf{I}(\mathcal{L}_k), \mathcal{DB}_I(\mathcal{A}^*(\hat{\mathcal{L}}_k))), \quad (12)$$

$$\hat{\mathcal{L}}_k \in \mathcal{A}^*(\hat{\mathcal{L}}_k), \forall k, \quad (13)$$

where  $K$  is the total number of localization tasks,  $\hat{\mathcal{C}}_k$  represents the category of the current user's floor.

We have a large-scale dataset  $\mathcal{D}_{base}$  from  $\mathcal{DB}_I$ , containing many-shot-labeled examples from each base class (domain)  $\mathcal{C}_{base}$  and a small-scale dataset  $\mathcal{D}_{novel}$  of only few-shot-labeled examples. The task of few-shot learning in our scheme is to learn a robust classifier using the few-shot-labeled examples in  $\mathcal{D}_{novel}$ , with the examples in  $\mathcal{D}_{base}$  as auxiliary data.

The main component of all few-shot algorithms is a feature extractor  $g_\theta(\cdot)$ , which is a convolutional neural network with parameters  $\theta$ . Given an image  $\mathbf{x}$ , the feature extractor will output a  $d$ -dimensional feature  $g_\theta(\mathbf{x})$ . How can the feature extractor  $g_\theta(\cdot)$  learn image features that can be readily exploited for novel classes with few training data during the second stage? With this goal in mind, we propose leveraging the recent progress in GAN feature learning, to further improve the current few-shot-learning approaches.

Based on the above multi-floor-localization problem, we decompose the original problem into two stages. It computes the feature vector, with respect to database images in the first stage, and compares the feature with every feature vector in the database, to obtain the final results in the second stage. The corresponding mathematical formulation is given below.

**Problem 2** (Regression-Based Localization).

$$\underset{g_\theta(\cdot), \hat{\mathcal{L}}_k}{\text{minimize}} \quad \frac{1}{K} \sum_{k=1}^K \min_{j \in \Omega_j^*(\hat{\mathcal{L}}_k)} \frac{\|\hat{\mathcal{L}}_k - \mathcal{L}_k\|_2^2}{\hat{p}_{\mathcal{A}}^j(\hat{\mathcal{L}}_k)} \quad (14)$$

$$\text{subject to} \quad (\mathcal{A}^*(\hat{\mathcal{L}}_k), \hat{\mathbf{p}}_{\mathcal{A}^*}(\hat{\mathcal{L}}_k)) = f(\mathbf{R}(\hat{\mathcal{L}}_k), \mathcal{DB}_W, \{\mathcal{A}^i(\{\mathcal{L}_{RP}^i\})\}), \quad (15)$$

$$v_k = g_\theta(\mathbf{I}(\mathcal{L}_k), \mathcal{DB}_I(\mathcal{A}^*(\hat{\mathcal{L}}_k))), \quad (16)$$

$$\hat{\mathcal{L}}_k, \hat{\mathcal{C}}_k = \tilde{g}(v_k), \quad (17)$$

$$\hat{\mathcal{L}}_k \in \mathcal{A}^*(\hat{\mathcal{L}}_k), \forall k, \quad (18)$$

where  $v_k$  denotes the extracted feature vector of all the images at the current floor.

In the formulation of Problem 2,  $g(\cdot)$  has been decomposed into two simplified functions,  $g_\theta(\cdot)$  and  $\tilde{g}(\cdot)$ . The function  $g_\theta(\cdot)$  and  $\tilde{g}(\cdot)$  are modeled as a typical feature-extract and feature-retrieve problem, to predict the precise location and floor index. The function  $g_\theta(\cdot)$  consists of a generator  $G$  and discriminator  $D$ , for each domain.

In the feature-extraction model  $g_\theta(\cdot)$ , assuming that a generator function  $G_{AB}$  that can transform domain  $A$  into domain  $B$  exists, such that  $b = G_{AB}(a)$ ,  $a \in A$ ,  $b \in B$ . Additionally, another generator  $G_{BA}$  transforms in the reverse direction, that is,  $\tilde{a} = G_{BA}(\tilde{b})$ . Similarly,  $G_{BA}$  should, also, transform domain  $B$  to domain  $A$ .

In the function  $\tilde{g}(\cdot)$ , we first use the corresponding feature extractor to extract the feature vector for the query image, then compare the feature with every feature vector in the database, using a cosine distance metric, choosing the most similar feature as the



retrieval result. Compared with the conventional visual-based localization technologies, the proposed multi-floor-localization scheme only needs to evaluate over the potential area  $A^*(\hat{\mathcal{L}}_k)$ , which undoubtedly reduces the computational complexity.

**Problem 3** (GAN-Based Localization).

$$g_{\theta}(\cdot)^* = \arg \min_{g_{\theta}(\cdot)} \{L_{Gan}(G_{AB}, D_B) + L_{Gan}(G_{BA}, D_A) + \lambda_1 L_{cyc}(G_{AB}, G_{BA}) + \lambda_2 L_{fea}(E_A, E_B, G_{AB}, G_{BA})\} \quad (19)$$

where  $\lambda_1$  and  $\lambda_2$  are hyperparameters used to balance the training direction.

We apply a cycle consistency  $L_1$  loss between domain  $A$  and  $B$ , as follows:

$$L_{cyc}(G_{AB}, G_{BA}) = E_{a-p(a)} [\|G_{BA}(G_{AB}(a)) - a\|_1] + E_{b-p(b)} [\|G_{AB}(G_{BA}(b)) - b\|_1], \quad (20)$$

where  $a - p(a)$  and  $b - p(b)$  are the image collections at the correspond domain, and  $E$  is the expectation function. Discriminator  $D_A$  and  $D_B$  work in each domain and try to discriminate between  $A, \tilde{A}$  and  $B, \tilde{B}$ , respectively. We apply adversarial losses, such as:

$$L_{Gan}(G_{AB}, D_B) = E_{b-p(b)} [\log D_B(b)] + E_{a-p(a)} [1 - \log D_B(G_{AB}(a))], \quad (21)$$

$$L_{Gan}(G_{BA}, D_A) = E_{a-p(a)} [\log D_A(a)] + E_{b-p(b)} [1 - \log D_A(G_{BA}(b))]. \quad (22)$$

To improve the training efficiency and make the model more practical for the localization task, we adopt a feature consistency loss of [29], built on the encoded features of different domains.

$$L_{fea}(E_A, E_B, G_{AB}, G_{BA}) = E_{a-p(a)} [\|E_B(G_{AB}(a)) - E_A(a)\|_2] + E_{b-p(b)} [\|E_A(G_{BA}(b)) - E_B(b)\|_2], \quad (23)$$

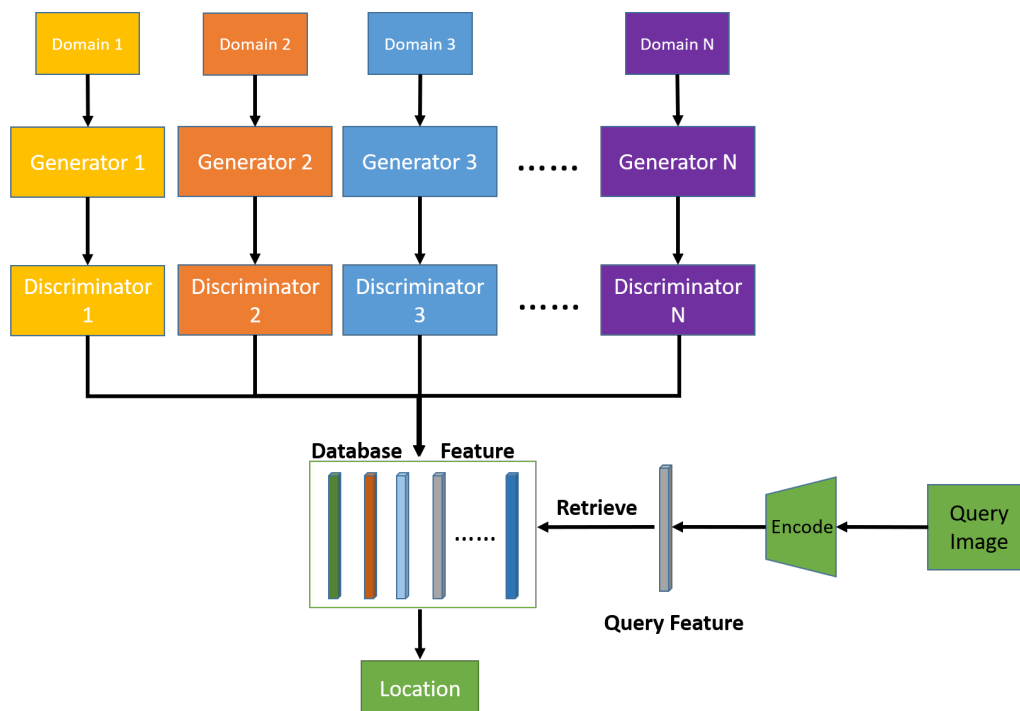
where  $E_A$  and  $E_B$  denote the encoder of the generator in domain  $A$  and domain  $B$ , respectively.

#### 4.2. Multi-Floor-Model Architecture

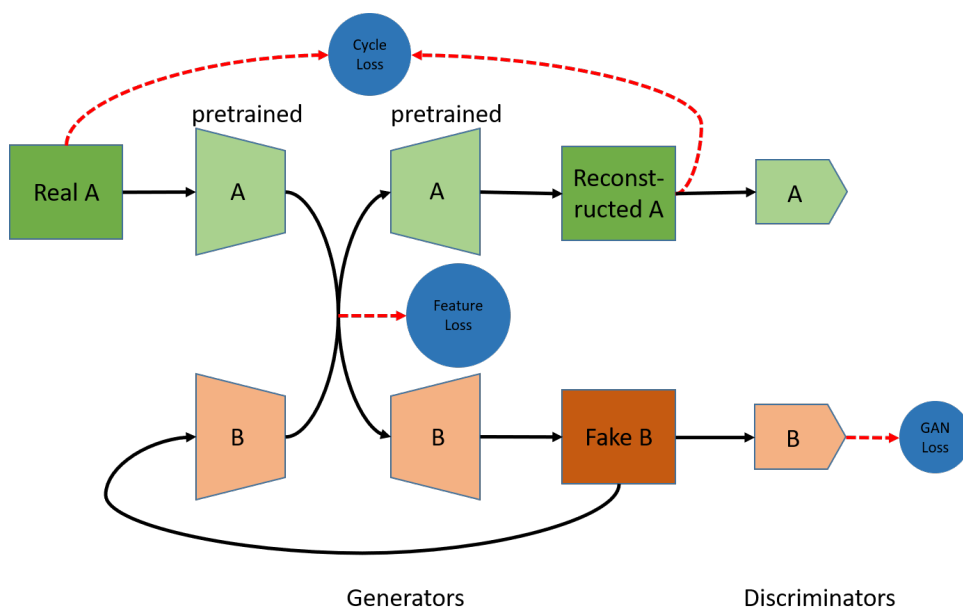
The overall procedures of the proposed multi-floor-localization scheme are shown in Figure 4, including the GAN-based feature-extraction model and the retrieval-based localization model.

In the GAN-based feature-extraction model  $g_{\theta}(\cdot)$ , we adopt the basic framework of multi-domain translation architecture ComboGAN [27]. Our proposed GAN-based localization architecture is shown as Figure 5, including GAN loss  $L_{Gan}$ , cycle consistency loss  $L_{cyc}$ , and feature-consistency loss  $L_{fea}$ . We design the loss function  $\mathbb{L}$  of the neural network, according to the previous problem formulation, which is given by

$$\mathbb{L} = \min \{L_{Gan} + \lambda_1 L_{cyc} + \lambda_2 L_{fea}\}. \quad (24)$$



**Figure 4.** The overall procedures of the proposed multi-floor-localization scheme. The query image is encoded as a feature vector and, then, used to retrieve the feature and image with the largest similarity in the database.



**Figure 5.** The proposed GAN-based localization architecture, which consists of generators and discriminators. The generators are divided into domain-specific pairs of encoders and decoders for each domain. The architecture of translation is  $A \rightarrow B$ , where  $A$  and  $B$  are randomly selected. The pass  $B \rightarrow A$  is performed in the same fashion.

The detailed configurations and parameters of our neural network are listed in Table 1 (Note that each convolutional layer in the Generators part corresponds to ‘Conv-InstanceNorm-ReLU’ )sequence. The transposed convolutional layer is denoted by Uconv1-3. The residual basic block is denoted as Res1-9.

The training procedure involves focusing on two of the total  $N$  domains at a time. At the beginning of each iteration, we selected two domains  $A, B \in \{1, 2, \dots, N\}, A \neq B$

from  $N$  domains, uniformly and at random. To make the GAN-based feature-extraction model more practical, for few-shot localization tasks, we proposed a training strategy based on a pre-trained model. We, first, trained a model using a large amount of labeled data from the base domains, encoding the knowledge from base domain data into the pre-trained model. Then, the pre-trained model was adopted as a feature extractor, to generate the feature embeddings of the labeled few-shot samples from the novel domain, which can be, directly, used as the initialization of the encoder for further fine-tuning.

**Table 1.** An Overview of Network Configuration and Parameters.

Module	Layers	Parameters
Generators	Conv1	$256 \times 256 \times 64$
	Conv2	$128 \times 128 \times 128$
	Conv3	$64 \times 64 \times 256$
	Res1-9	$64 \times 64 \times 256$
	Uconv1	$64 \times 64 \times 256$
	Uconv2	$128 \times 128 \times 128$
	Uconv3	$256 \times 256 \times 3$
Discriminator	Conv1	$128 \times 128 \times 64$
	Conv2	$64 \times 64 \times 128$
	Conv3	$32 \times 32 \times 256$
	Conv4	$16 \times 16 \times 512$
	Conv5	$16 \times 16 \times 1$

Our retrieval-based localization is based on GAN-based feature learning, as shown in Figure 4, which can better describe the function  $g(\cdot)$ . In the online stage, we trained the GAN-based feature-extraction model mentioned above, to pre-encode each database image into a one-dimensional vector, which can avoid redundant calculations. In the offline stage, we, first, used the corresponding trained encoder networks to extract features for the query image, then compared the features with every feature vector in the database, using a cosine distance metric. We chose the image with the most similar features, as the retrieval result.

## 5. Experiment Results

In this section, we conduct several numerical experiments to show the effectiveness of our proposed system.

### 5.1. Experimental Environment

The proposed localization scheme was verified in the corridor environment of a typical office building, as shown in Figure 6, which contains six floors, with 4000 square meters of area for each floor. The layout of each floor is more or less the same, with dramatically changing lighting conditions throughout the day. Meanwhile, the WiFi signals are, generally, unreliable, due to the regular daily activities of working staff. In the above settings, both the SFBL and the VBL schemes cannot achieve satisfying localization results, which is a great challenge for the fusion-based localization schemes.

To obtain the ground-truth positions and establish the databases, e.g.,  $DB_W$  and  $DB_I$ , we conducted the following implementation works. First, the Cartographer was used to build SLAM maps on each floor, as shown in Figure 7. The mobile robot could obtain its own positioning in real-time, and the reference coordinate origin was the starting point for mapping on each floor. Secondly, the coordinates of the camera and IMU module, relative to the robot center point, were calculated according to their installation positions. In SLAM, the transformation matrix from an image-coordinate system to a laser-coordinate system was calculated, and the image, IMU, wireless signal, and laser data are aligned. Third, several reference points were set on the 2D SLAM diagram to plan the s-shaped path of the mobile robot (the S-shaped path increases the z-axis Angle data in IMU), and the S-shaped trajectory passed through all the reference points. Finally, we set up several ramps, so that

the height varied and the positioning accuracy was measured, according to the 3D distance. To ensure the localization accuracy, the mobile robot periodically collected wireless signals and image data, to keep  $DB_W$  and  $DB_I$  updated.



**Figure 6.** The layout of the experimental corridor environment, the mobile robot constructing  $DB_W$  and  $DB_I$ , automatically, and several image samples from  $DB_I$ .



**Figure 7.** Partial SLAM map for floor 4.

All sampled images in our experiments were scaled to  $286 \times 286$  pixels, and randomly cropped to  $256 \times 256$  pixels, according to [28], for training and testing. In the training stage, learning rates were fixed at 0.0002 for generators and 0.0001 for discriminators, for the first half, and this, linearly, decreased to zero, during the second half. The batch size was fixed to 1 and the number of epochs was chosen as  $100 \times N$ . In our experiments,

the data-training processes were conducted on a localization server with NVIDIA Titan X GPU and the Pytorch platform. Other detailed parameter values are listed in Table 2. Kindly note that when the offline training processes were completed, we could provide the online localization service, immediately.

**Table 2.** The detailed parameter values of the experiments.

Parameter	Value	Parameter	Value
$N_s$	50	$N_{AP}$	5
$N_p$	492	$N_w$	752
$N_l$	780	$N_{RGB}$	3
$N_{RP}$	24	$N_A$	15
$N_W$	2	$N_I$	2
$J^*$	4	$N$	4
GPU	NVIDIA Titan X	Platform	Pytorch

### 5.2. Localization Accuracy and Computational Complexity

In the following experiments, we, first, investigated the localization accuracy, in terms of the cumulative distribution function (CDF) of 3D distance errors, to show the effectiveness of the proposed JVWL algorithm. To obtain an in-depth understanding, we plotted the estimation accuracy of floors and the corresponding mean squared errors (MSE) of estimated two-dimensional tracking trajectories (For simplicity, we only calculated the MSE of tracking trajectories, when the floor estimation was correct). Moreover, we, also, compared the associated computational complexities, in terms of the total computational times. All the above experiments were compared with the following two baseline schemes, e.g., *Baseline 1*, an SVM-based, WiFi-only localization scheme, and *Baseline 2*, a WiFi and vision-integrated scheme, as proposed in [20].

In Figure 8, we compare the localization accuracy in terms of the 3D distance errors of the proposed JVWL scheme with baselines. As shown in this figure, the proposed method (red) can achieve a 0.62-m localization accuracy with a CDF equal to 0.5, which achieves more accurate and reliable localization results than *Baseline 1* (green, 1.78 m) and *Baseline 2* (blue, 3.12 m).

In Figure 9, we plotted a snapshot view of tracking trajectories for different localization schemes, where our proposed JVWL scheme (red) is closer to the ground truth (yellow) than other schemes. The detailed estimation accuracy of floors and the corresponding MSE of the estimated two-dimensional tracking trajectories are summarized in Table 3. Numerically, the proposed JVWL scheme achieves an 89.22% estimation accuracy of floors, which is far superior to *Baseline 1* (57.93%) and *Baseline 2* (66.87%). In addition, we show that the proposed JVWL scheme can achieve an MSE of estimated tracking trajectory up to 1.24 m, which outperforms *Baseline 1* (3.43 m) and *Baseline 2* (2.64 m).

Although the proposed scheme provides a satisfactory localization performance in terms of 3D distance errors, the estimation accuracy of floors, and the corresponding MSE of two-dimensional tracking trajectory, the implementation complexity is still open. In this experiment, we compared the total computational time cost with two baseline schemes. As listed in Table 3 The floor estimation accuracy of *Baseline 1* is less than 60% and the MSE result is more than 3 m, which is not suitable for practical implementation., the average running time of the proposed JVWL scheme is around 0.25 s, which is five times less than *Baseline 2*. Meanwhile, the memory requirement for the proposed JVWL scheme is around 10 MB, which is much lower than *Baseline 2*. This is due to the fact that the matching algorithm adopted in *Baseline 2* requires significant storage and searching abilities.

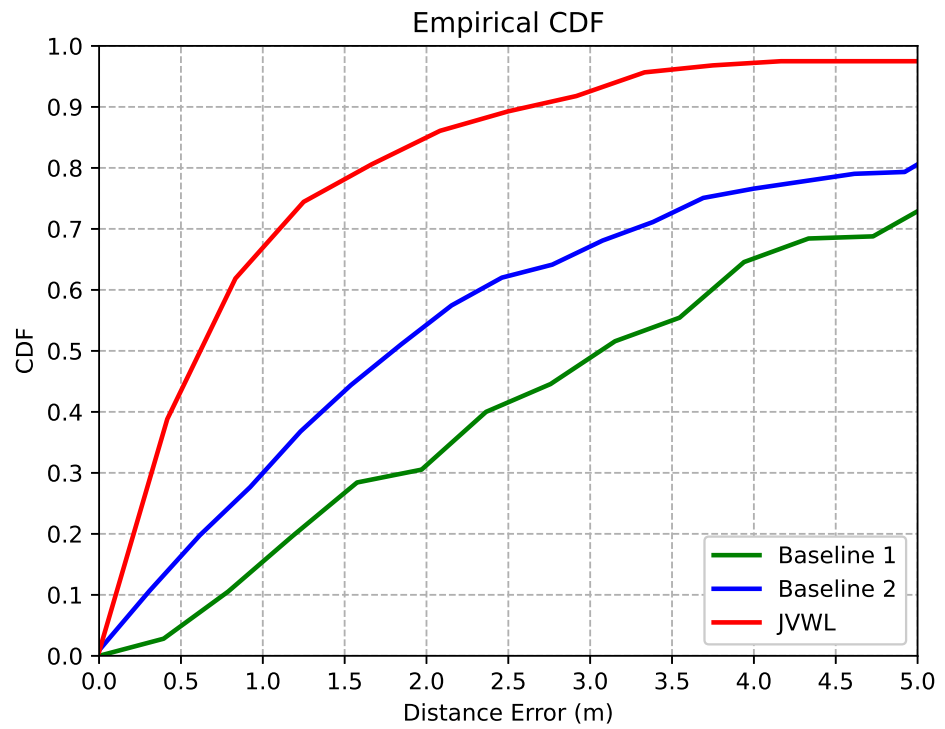


Figure 8. CDF of localization errors for different localization methods.

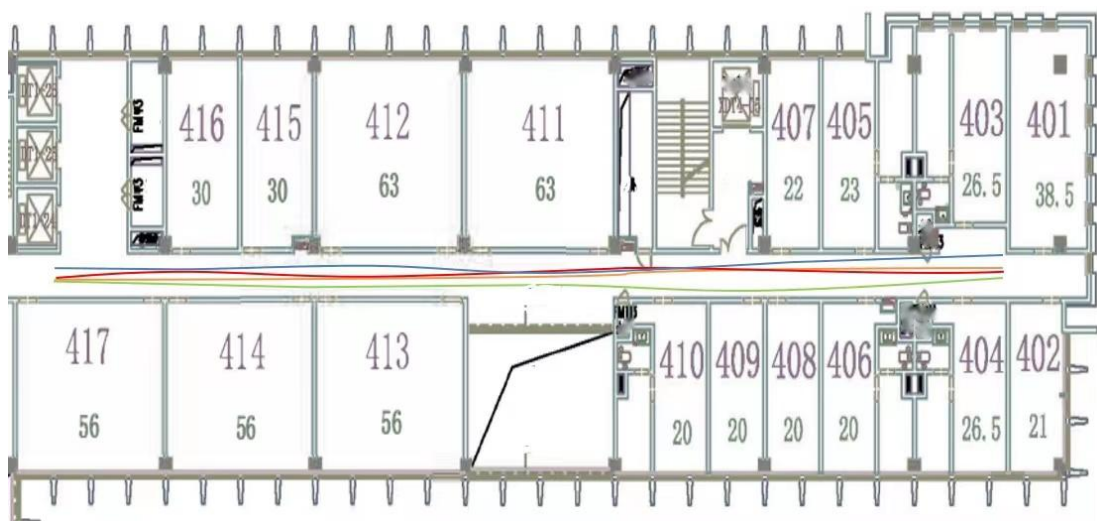


Figure 9. Localization trajectories for different schemes, including ground truth (yellow), JWVL (red), Baseline 1 (green), and Baseline 2 (blue).

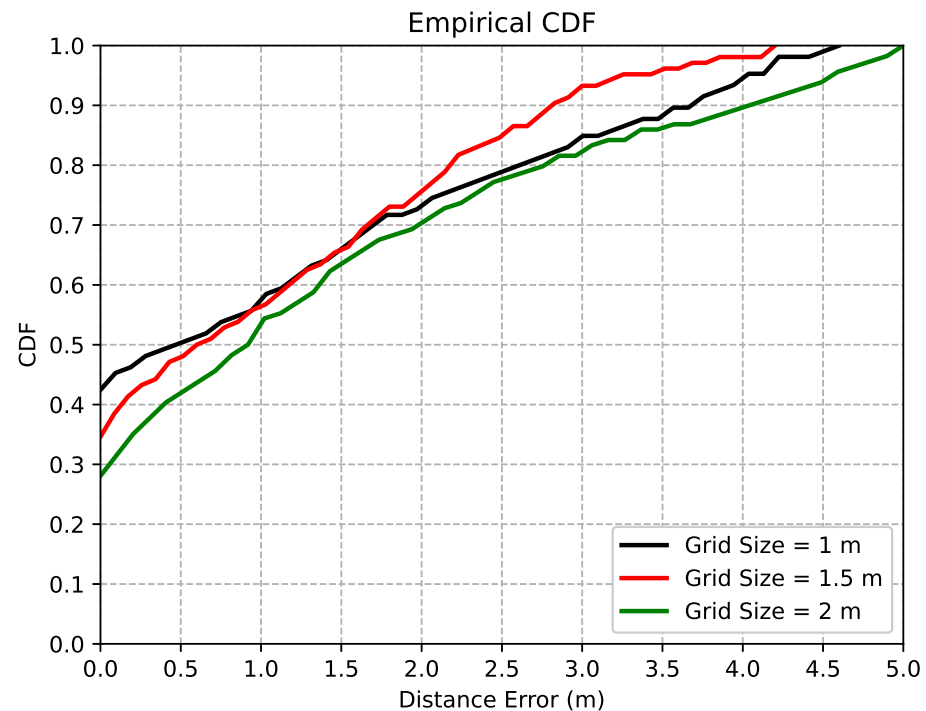
Table 3. Localization accuracy, total running time, and memory space comparison for different localization methods.

Methods	Running Time	Memory Space	Accuracy	MSE
Baseline 1	0.01 s	4.8 MB	57.93%	3.43 m
Baseline 2	1.2 s	25 MB	66.87%	2.64 m
JWVL	0.25 s	10 MB	89.22%	1.24 m



### 5.3. Effects of Grid Sizes

In this experiment, we investigated the grid size effects to balance the localization accuracy and the associated deployment cost. Distances between adjacent RPs were selected as 1 m, 1.5 m, and 2 m, respectively. In Figure 10, we compared the 3D distance errors of the proposed JVWL scheme under different grid sizes. As shown in this figure, we were able to achieve 0.62 m, 0.83 m, and 1.19 m for grid sizes of 1 m, 1.5 m, and 2 m, respectively. Since the deployment cost of data collection and labeling for the grid size of 1.5 m was around 50% more than the grid size of 1 m and the localization accuracy improved by only 25.3%, we recommend choosing 1.5 m, as the grid size for the practical database construction.



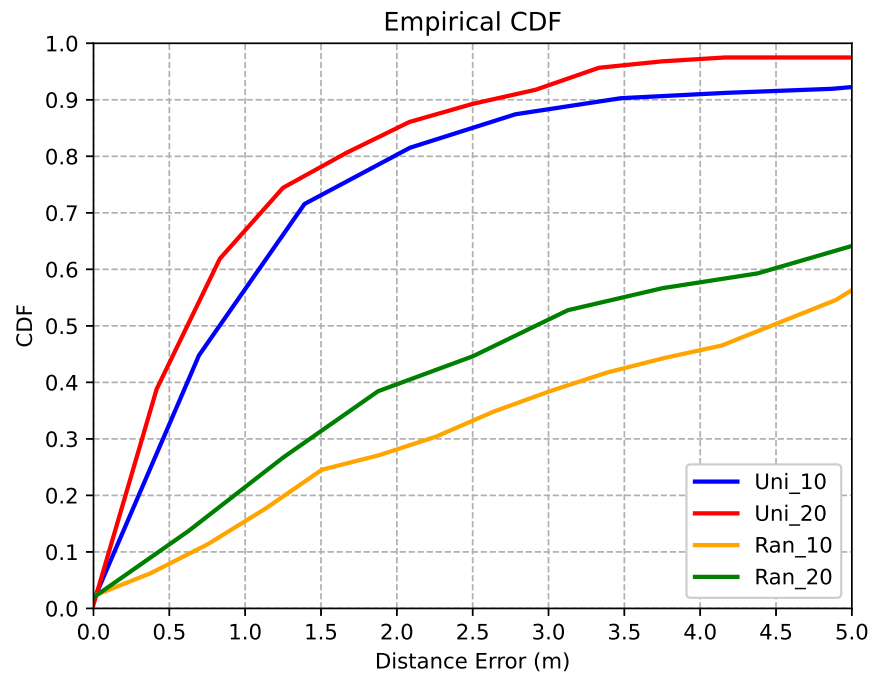
**Figure 10.** CDF of localization errors, for different training cell sizes. Three training datasets with different cell sizes were tested, to explore the most efficient deployment method.

### 5.4. Effects of Few-Shot

To demonstrate the effects of few-shot learning, we selected different “few-shot” approaches, e.g., to select 10%/20% of sampling images, uniformly (denoted as Uni\_10 and Uni\_20) and randomly (denoted as Ran\_10 and Ran\_20), to construct the training dataset.

In Figure 11 and Table 4, we compare the CDF of 3D distance errors, the estimation accuracy of floors, and the corresponding MSE of two-dimensional tracking trajectories of different “few-shot” approaches. As shown in Figure 11, selecting the sampling images uniformly is much better than selecting them randomly, and the averaged 3D distance errors can be improved, changing from more than 3 m to less than 1 m. A similar phenomenon occurs for the estimation accuracy of floors and the corresponding MSE of two-dimensional tracking trajectories, as listed in Table 4. For example, the estimation accuracy can be improved from less than 65% to more than 85%, and the corresponding MSE can be improved from more than 4.5 m to less than 2 m.

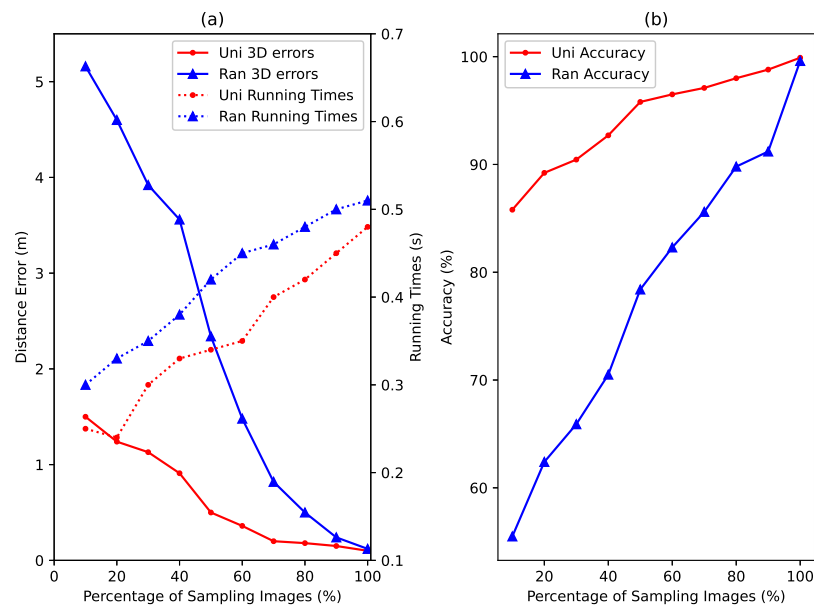
In Figure 12, we, linearly, increased the percentages of the sampling image selection, to demonstrate the corresponding “few-shot” effects. Intuitively, with more sampling images in the training dataset, the 3D distance errors and the estimation accuracy of floors can be improved. Moreover, as shown in Figure 12, the running times are controllable, when the percentages of sampling image selection are less than 20%. Based on the above results, we believe that our proposed JVWL scheme can, effectively, utilize few-shot samples, and 20% will be a reasonable number.



**Figure 11.** The CDF curves of 3D distance errors of different training sets for the proposed JWVL algorithm, in an experimental corridor environment.

**Table 4.** Localization accuracy with different numbers and distribution of few-shot samples.

Few-shot Samples	Accuracy	MSE
Uni_10	85.80%	1.50 m
Uni_20	89.22%	1.24 m
Ran_10	55.49%	5.16 m
Ran_20	62.40%	4.60 m



**Figure 12.** Influences of distribution of few-shot samples in our experiments. (a) represent results for localization distance error and average time consumption for different few-shot sample distributions. (b) shows results for the estimation accuracy of floors for different distributions of few-shot samples.

## 6. Conclusions and Discussion

In this paper, we propose a joint visual- and wireless-signal feature-based approach, for high-precision multi-floor indoor localization. By applying a hybrid coarse and fine localization framework, we could formulate the JVWL problem, accordingly. Through some theoretical analysis, a GAN-based-deep-learning scheme is proposed, for multi-floor localization architecture. Compared with the conventional SFBL and VBL schemes, our localization scheme could achieve 0.62 m 3D localization accuracy on average, and 1.24 m MSE of two-dimensional tracking trajectories, with a floor-estimation accuracy equal to 89.22%, which significantly improves the localization accuracy in the multi-floor scenarios. Meanwhile, the proposed JVWL scheme takes less than 0.25 s for the localization process, which is five times less than conventional WiFi and vision integrated schemes.

Compared with the industrial-laser SLAM solution, the JVWL solution has some advantages. First, the JVWL solution solves the multi-layer localization that laser SLAM cannot handle, especially the symmetrical environment of each floor. Secondly, JVWL scheme is suitable for people's livelihoods, such as in service positionings, which is a low-cost and efficient positioning scheme. For example, it can be used in shopping guidance in shopping malls, personnel positioning in COVID-19 makeshift hospitals, positioning in construction site clusters, etc. Finally, the JVWL scheme is convenient for mobile terminal applications.

Our proposed localization system, still, has limitations, such as a low positioning accuracy, the need to build fingerprint database, an offline map, etc. Although this accuracy is sufficient for ordinary scenes, it is, still, difficult for industrial and special scenes. Constructing a fingerprint database is complicated and heavy work, which is a challenge for field implementation. Lidar SLAM maps can be built by remote-controlled mobile robots, or by carrying lidar in a backpack. In the future, we plan to fuse visual, wireless, and lidar data through loose coupling, hoping to make improvements in terms of cost reductions, ease of use, accuracy, and other indicators.

**Author Contributions:** Conceptualization, S.Z. and S.X.; methodology, S.Z., S.X. and G.Z.; software, Y.W., C.X. and G.Z.; validation, Y.W., C.X. and G.Z.; formal analysis, S.Z. and S.X.; investigation, S.X. and G.Z.; data curation, G.Z. and Y.W.; writing—original draft preparation, Y.W. and G.Z.; writing—review and editing, G.Z.; visualization, Y.W. and G.Z.; supervision, S.Z. and S.X.; funding acquisition, G.Z. and S.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by The Key Area Research and Development Program of Guangdong Province(2020B0101130012) and The Foshan Science and Technology Innovation Team Project (FS0AA-KJ919-4402-0060).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Huang, H.; Gartner, G.; Krisp, J.M.; Raubal, M.; Van de Weghe, N. Location based services: Ongoing evolution and research agenda. *J. Locat. Based Serv.* **2018**, *12*, 63–93. [[CrossRef](#)]
2. Elgendy, M.; Sik-Lanyi, C.; Kelemen, A. A novel marker detection system for people with visual impairment using the improved tiny-yolov3 model. *Comput. Methods Programs Biomed.* **2021**, *205*, 106112. [[CrossRef](#)] [[PubMed](#)]
3. Al-Khalifa, S.; Al-Razgan, M. Ebsar: Indoor guidance for the visually impaired. *Comput. Electr. Eng.* **2016**, *54*, 26–39. [[CrossRef](#)]
4. Tomai, S.; Krjanc, I. An automated indoor localization system for online bluetooth signal strength modeling using visual-inertial slam. *Sensors* **2021**, *21*, 2857.
5. Xu, L.; Zhuang, W.; Yin, G.; Pi, D.; Liang, J.; Liu, Y.; Lu, Y. Geometry-based cooperative localization for connected vehicle subject to temporary loss of GNSS signals. *IEEE Sens. J.* **2021**, *21*, 23527–23536.
6. Onyekpe, U.; Palade, V.; Kanarachos, S. Learning to localise automated vehicles in challenging environments using Inertial Navigation Systems (INS). *Appl. Sci.* **2021**, *11*, 1270. [[CrossRef](#)]

7. Fischer, G.; Bordoy, J.; Schott, D.J.; Xiong, W.; Gabbrielli, A.; Hoflinger, F.; Fischer, K.; Schindelbauer, C.; Rupitsch, S.J. Multimodal Indoor Localization: Fusion Possibilities of Ultrasonic and Bluetooth Low-Energy Data. *IEEE Sens. J.* **2022**, *22*, 5857–5868. [[CrossRef](#)]
8. Mohanty, S.; Tripathy, A.; Das, B. An overview of a low energy UWB localization in IoT based system. In Proceedings of the 2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA), Goa, India, 20–22 September 2021; pp. 293–296.
9. Zhang, H.; Zhang, Z.; Zhang, S.; Xu, S.; Cao, S. Fingerprint-based localization using commercial lte signals: A field-trial study. In Proceedings of the IEEE Vehicular Technology Conference, Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
10. Xiang, C.; Zhang, S.; Xu, S.; Chen, X.; Cao, S.; Alexopoulos, G.C.; Lau, V.K. Robust sub-meter level indoor localization with a single WiFi access point—Regression versus classification. *IEEE Access* **2019**, *7*, 146309–146321. [[CrossRef](#)]
11. Morar, A.; Moldoveanu, A.; Mocanu, I.; Moldoveanu, F.; Radoi, I.E.; Asavei, V.; Gradinaru, A.; Butean, A. A comprehensive survey of indoor localization methods based on computer vision. *Sensors* **2020**, *20*, 2641. [[CrossRef](#)]
12. Piasco, N.; Sidibé, D.; Demonceaux, C.; Gouet-Brunet, V. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognit.* **2018**, *74*, 90–109. [[CrossRef](#)]
13. Huang, B.; Zhao, J.; Liu, J. A survey of simultaneous localization and mapping. *arXiv* **2019**, arXiv:1909.05214.
14. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005; pp. 309–384.
15. Grisetti, G.; Stachniss, C.; Burgard, W. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. Robot.* **2007**, *23*, 34. [[CrossRef](#)]
16. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real time loop closure in 2d lidar slam. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
17. Jiao, J.; Deng, Z.; Arain, Q.A.; Li, F. Smart fusion of multi-sensor ubiquitous signals of mobile device for localization in gnss-denied scenarios. *Wirel. Pers. Commun.* **2021**, *116*, 1507–1523. [[CrossRef](#)]
18. Dong, J.; Xiao, Y.; Noreikis, M.; Ou, Z.; Ylä-Jääski, A. iMoon: Using Smartphones for Image-based Indoor Navigation. *ACM Conf. Embed. Netw. Sens. Syst.* **2015**, *11*, 85–97.
19. Xu, H.; Yang, Z.; Zhou, Z.; Shangguan, L.; Yi, K.; Liu, Y. Indoor Localization via Multi-modal Sensing on Smartphones. In Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; pp. 208–219.
20. Hu, Z.; Huang, G.; Hu, Y.; Yang, Z. WI-VI Fingerprint: WiFi and Vision Integrated Fingerprint for Smartphone-based Indoor Self-localization. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 4402–4406.
21. Kendall, A.; Grimes, M.; Cipolla, R. Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2938–2946.
22. Walch, F.; Hazirbas, C.; Leal-Taixe, L.; Sattler, T.; Hilsenbeck, S.; Cremers, D. Image-based localization using lstms for structured feature correlation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 627–637.
23. Arjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. Netvlad: Cnn architecture for weakly supervised place recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *6*, 1437–1451.
24. Hu, H.; Qiao, C.; Cheng, M.; Liu, Z.; Wang, H. Dargil: Domain adaptation for semantic and geometric-aware image-based localization. *IEEE Trans. Image Process.* **2020**, *12*, 1342–1353. [[CrossRef](#)] [[PubMed](#)]
25. Liu, M.Y.; Breuel, T.; Kautz, J. Unsupervised image-to-image translation networks. *arXiv* **2017**, arXiv:1703.00848.
26. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 3, pp. 1857–1865.
27. Anoosheh, A.; Agustsson, E.; Timofte, R.; Van Gool, L. Combogan: Unrestrained scalability for image domain translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 896–8967.
28. Anoosheh, A.; Sattler, T.; Timofte, R.; Pollefeys, M.; Van Gool, L. Night-to-day image translation for retrieval-based localization. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5958–5964.
29. Hu, H.; Wang, H.; Liu, Z.; Yang, C.; Chen, W.; Xie, L. Retrieval-based localization based on domain-invariant feature learning under changing environments. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 3684–3689.
30. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–34. [[CrossRef](#)]
31. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
32. Li, W.; Wang, L.; Xu, J.; Huo, J.; Gao, Y.; Luo, J. Revisiting local descriptor based image-to-class measure for few-shot learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7253–7260.

33. Sun, Q.; Liu, Y.; Chua, T.S.; Schiele, B. Meta-transfer learning for few-shot learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 403–412.
34. Qi, H.; Brown, M.; Lowe, D.G. Low-shot learning with imprinted weights. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5822–5830.
35. Qiao, S.; Liu, C.; Shen, W.; Yuille, A.L. Few-shot image recognition by predicting parameters from activations. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7229–7238.
36. Gidaris, S.; Komodakis, N. Dynamic few-shot visual learning without forgetting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4367–4375.
37. Chen, W.Y.; Liu, Y.C.; Kira, Z.; Wang, Y.C.F.; Huang, J.B. A closer look at few-shot classification. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
38. Zhang, H.; Li, Y. Lightgbm indoor positioning method based on merged wi-fi and image fingerprints. *Sensors* **2021**, *21*, 3662.
39. Huang, G.; Hu, Z.; Wu, J.; Xiao, H.; Zhang, F. Wifi and vision-integrated fingerprint for smartphone-based self-localization in public indoor scenes. *IEEE Internet Things J.* **2020**, *7*, 6748–6761. [[CrossRef](#)]
40. Jiao, J.; Wang, X.; Deng, Z. Build a robust learning feature descriptor by using a new image visualization method for indoor scenario recognition. *Sensors* **2017**, *17*, 1569. [[CrossRef](#)] [[PubMed](#)]
41. Redžić, M.D.; Laoudias, C.; Kyriakides, I. Image and wlan bimodal integration for indoor user localization. *IEEE Trans. Mob. Comput.* **2020**, *19*, 1109–1122. [[CrossRef](#)]
42. Cartographer. Available online: <https://google-cartographer.readthedocs.io/en/latest/> (accessed on 12 March 2022).
43. Zhou, R.; Lu, X.; Zhao, P.; Chen, J. Device-free Presence Detection and Localization with SVM and CSI Fingerprinting. *IEEE Sens. J.* **2017**, *17*, 7990–7999. [[CrossRef](#)]