Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 3279059, 9 pages
https://doi.org/10.1155/2021/3279059

*Research Article*

# Image-Based Indoor Localization Using Smartphone Camera

**Shuang Li,[1,2] Baoguo Yu,[1] Yi Jin,[3] Lu Huang,[1,2] Heng Zhang,[1,2] and Xiaohu Liang[1,2]**

[1]*State Key Laboratory of Satellite Navigation System and Equipment Technology, China*
[2]*Southeast University, China*
[3]*Beijing Jiaotong University, China*

Correspondence should be addressed to Yi Jin; yjin@bjtu.edu.cn

Received 17 April 2021; Revised 30 May 2021; Accepted 20 June 2021; Published 5 July 2021

Academic Editor: Mohammad R. Khosravi



With the increasing demand for location-based services such as railway stations, airports, and shopping malls, indoor positioning technology has become one of the most attractive research areas. Due to the effects of multipath propagation, wireless-based indoor localization methods such as WiFi, bluetooth, and pseudolite have difficulty achieving high precision position. In this work, we present an image-based localization approach which can get the position just by taking a picture of the surrounding environment. This paper proposes a novel approach which classifies different scenes based on deep belief networks and solves the camera position with several spatial reference points extracted from depth images by the perspective-*n*-point algorithm. To evaluate the performance, experiments are conducted on public data and real scenes; the result demonstrates that our approach can achieve submeter positioning accuracy. Compared with other methods, image-based indoor localization methods do not require infrastructure and have a wide range of applications that include self-driving, robot navigation, and augmented reality.

## 1. Introduction

According to statistics, more than 80 percent of people's living time is in an indoor environment such as shopping malls, airports, libraries, campuses, and hospitals. The purpose of the indoor localization system is to provide accurate positions in large buildings. It is vital to applications such as evacuation of trapped people at fire scenes, tracking of valuable assets, and indoor service robot. For these applications to be widely accepted, indoor localization requires an accurate and reliable position estimation scheme [1].

In order to provide a stable indoor location service, a large number of technologies are researched including pseudolite, bluetooth, ultrasonic, WiFi, ultra wideband, and LED [2, 3]. It is almost impossible to obtain very accurate results for a radio-based approach in view of the multipath interference through arrival time and arrival angle methods. The time-varying indoor environment and the movement of pedestrians also have adverse effects on the stability of fingerprint information [4–6]. In addition, the high cost of hardware equipment, construction, and installation as well as maintenance and update is also an important factor limiting the development of indoor positioning technology. Besides, these kinds of methods can only output the position ($X$, $Y$, and $Z$ coordinates) but not the view angle (pitch, yaw, and roll angles).

The vision-based positioning method is a kind of passive positioning technology which can achieve high positioning accuracy and does not need extra infrastructure. Moreover, it can not only output the position but also the view angle at the same time. Therefore, it has gradually become a hotspot of indoor positioning technology [7, 8]. Such methods typically involve four steps: first, establishing an indoor image dataset collected by depth cameras with exact positional information; second, comparing the images collected by a camera to the images in the database which established the last step; third, retrieving some of the most similar pictures, then extracting the feature and matching the points; at last, solving the perspective-*n*-point problem [9–12]. However, the application of scene recognition to mobile location implies several challenges [13–15]. The complex three-dimensional shape of the environment results in occlusions, overlaps, shadows, and reflections which require a robust description of the scene [16]. To address these issues,

we propose a particularly efficient approach based on a deep belief network with local binary pattern feature descriptors. It enables us to find out the most similar pictures quickly. In addition, we restrict the search space according to adaptive visibility constraints which allows us to cope with extensive maps.

## 2. Related Work

Before presenting the proposed approach, we review previous work on image-based localization methods and divide these methods into three categories roughly.

Manual mark-based localization methods completely rely on the natural features of the image which lacks robustness, especially under conditions of varying illumination. In order to improve the robustness and accuracy of the reference point, special coding marks are used to meet the higher positioning requirements of the system. There are three benefits: simplify the automatic detection of corresponding points, introduce system dimensions, and distinguish and identify targets by using a unique code for each mark. Common types of marks include concentric rings, QR codes, or patterns composed of colored dots. The advantage is raising the recognition rate and effectively reducing the complexity of positioning methods. The disadvantage is that the installation and maintenance costs are high, some targets are easily obstructed, and the scope of application is limited [17, 18].

Natural mark-based localization methods usually detect objects on the image and match them with an existing building database. The database contains the location information of the natural marks in the building. The advantage of this method is that it does not require additional local infrastructure. In other words, the reference object is actually a series of digital reference points (control points in photogrammetry) in the database. Therefore, this type of system is suitable for large-scale coverage without increasing too much cost. The disadvantage is that the recognition algorithm is complex and easy to be affected by the environment, the characteristics are easy to change, and the dataset needs to be updated [19–22].

Learning-based localization methods have emerged in the past few years. It is an end-to-end method that directly obtains 6dof pose, which has been proposed to solve loop-closure detection and pose estimation [23]. This method does not require feature extraction, feature matching, and complex geometric calculations and is intuitive and concise. It is robust in weak textures, repeated textures, motion blur, and lighting changes. In the training phase, the calculative scale is very large, and GPU servers are usually required, which cannot run smoothly on mobile platforms [20]. In many scenarios, learning-based features are not as effective as traditional features such as SIFT, and the interpretability is poor [24–27].

## 3. Framework and Method

In this section, first, we introduce the overview of the framework. Then, the key modules are explained in more detail in the subsequent sections.

*3.1. Framework Overview.* The whole pipeline of the visual localization system is shown in Figure 1. In the following, we briefly provide an overview of our system.

In the offline stage, the RGB-D cameras are held to collect enough RGB images and depth images around the indoor environment. At the same time, the pose of the camera and the 3D point cloud are constructed. The RGB image is used as a learning dataset to train the network model, and then, the network model parameters are saved until the loss function value does not decrease. In the online stage, after the previous step is completed, anyone enters the room, downloads the trained network model parameters to the mobile phone, and takes a picture with the mobile phone, and the most similar image is identified according to the deep learning network. The unmatched points are eliminated, and the pixel coordinates of the matched points and the depth of the corresponding points are extracted. According to the pinhole imaging model, the $n$-point perspective projection problem-solving method can be used to calculate the pose of the mobile phone in the world coordinate system. Finally, the posture is converted into a real position and displayed on the map.

*3.2. Camera Calibration and Image Correction.* Due to the processing error and installation error of camera lens, the image has radial distortion and tangential distortion. Therefore, we must calibrate the camera and correct the images in the preprocessing stage. The checkerboard contains some calibration reference points, and the coordinates of each point are disturbed by the same noise. Establishing the function $\gamma$:

$$\gamma = \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| p_{ij} - p\wedge(A, R_i, t_i, P_i) \right\|^2, \tag{1}$$

where $p_{ij}$ is the coordinate of the projection points on image $i$ for reference point $j$ in the three-dimensional space. $R_i$ and $t_i$ are the rotation and translation vectors of image $i$. $P_i$ is the three-dimensional coordinate of reference point $i$ in the world coordinate system. $\widehat{p}(A, R_i, t_i, P_i)$ is the two-dimensional coordinate in the image coordinate system.

*3.3. Scene Recognition.* In this section, we use the deep belief network (DBN) to categorize the different indoor scenes. The framework includes image preprocessing, LBP feature extracting, DBN training, and scene classification.

*3.3.1. Local Binary Pattern.* The improved LBP feature is insensitive to rotation and illumination changes. The LBP operator can be specifically described as the following: the gray values in the window center pixel are defined as the threshold, and the gray values of the surrounding 8 pixels are, respectively, compared with the threshold in a clockwise direction, and if the gray value is bigger than the threshold, then mark the pixel as 1; otherwise, mark 0, and then get an 8-bit binary number through the comparison. After the decimal conversion, get the LBP value of the center pixel in this window. The value reflects the texture information of the point at this position. The calculation process is shown in Figure 2.

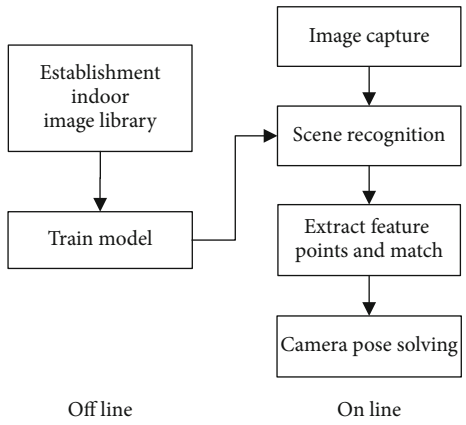



Figure 1: The framework of the visual localization system.

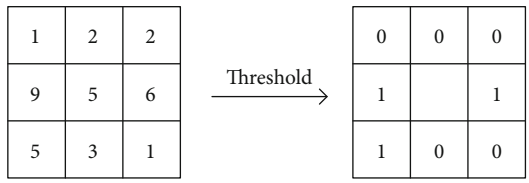



Figure 2: Local binary pattern calculation process.

The formula of local binary pattern:

$$\mathrm{LBP}(x_c, y_c) = \sum_{n=0}^{N-1} 2^n s(i_n - i_c),$$
$$s(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{else}, \end{cases} \tag{2}$$

where $(x_c, y_c)$ is the horizontal and vertical coordinate of the center pixel; $N$ is number 8; $i_c$, $i_n$ are the gray values of the center pixel and the neighborhood pixel, respectively; and $s(\cdot)$ is the two-valued symbol function.

The earliest proposed LBP operator can only cover a small range of images, so the optimization and improvement methods for the LBP operator are constantly proposed by researchers. We adopt the method which improves the insufficiency of the window size of the original LBP operator by replacing the traditional square neighborhood with a circular neighborhood and expanding the window size as shown in Figure 3.

In order to make the LBP operator have rotation invariance, the circular neighborhood is rotated clockwise to obtain a series of binary strings, and the minimum binary value is obtained, and then, the value is converted into decimal, which is the LBP value of the point. The process of obtaining the rotation-invariant LBP operator is shown in Figure 4.

*3.3.2. Deep Belief Network.* The deep belief network consists of a multirestricted Boltzmann machine (RBM) and a backpropagation (BP) neural network. The Boltzmann machine is a neural network based on learning rules. It consists of a visible layer and a hidden layer. The neurons in the same layer and the neurons in different layers are connected to each other. There are two types of neuron output states: active and inactive, represented by numbers 1 and 0. The advantage of the Boltzmann machine is its powerful unsupervised learning ability, which can learn complex rules from a large amount of data; the disadvantages are the huge amount of calculation and the long training time. The restricted Boltzmann machine canceled the connection between neurons in the same layer; each hidden unit and visible layer unit are independent of each other. Roux and Bengio theoretically prove that as long as the number of neurons in the hidden layer and the training samples are sufficient, the arbitrary discrete distribution can be fitted. The structure of BM and RBM is shown in Figure 5.

The joint configuration energy of its visible and hidden layers is defined as

$$E(v, h|\theta) = -\sum_{i=1}^{m} b_i v_i - \sum_{j=1}^{n} c_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i w_{ij} h_j, \tag{3}$$

where $\theta = \{W_{ij}, b_i, c_j\}$ are parameters in RBM, $b_i$ is bias of visible layer $i$, $c_j$ is bias of visible layer $j$, and $w_{ij}$ is the weight.

The output of the hidden layer unit is

$$h_j = \sum_{j=1}^{n} v_i w_{ij} + b_j. \tag{4}$$

When the parameters are known, based on the above energy function, the joint probability distribution of $(v, h)$

$$P(v, h|\theta) = \frac{e^{-E(v,h|\theta)}}{Z(\theta)},$$
$$Z(\theta) = \sum_{v,h} e^{-E(v,h|\theta)}, \tag{5}$$

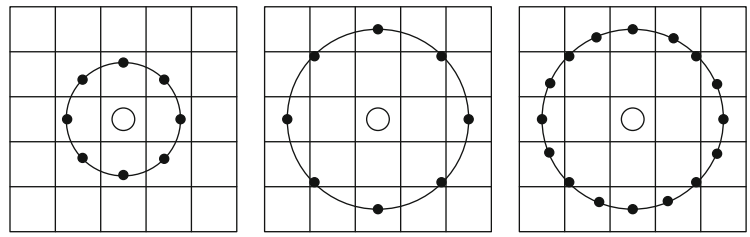
where $Z(\theta)$ is the normalization factor. Distribution of $v$ is $P(v|\theta)$, joint probability distribution $P(v, h|\theta)$:

$$P(v|\theta) = \sum_{h} p(v, h|\theta) = \frac{1}{Z(\theta)} \sum_{h} e^{-E(v,h|\theta)}. \tag{6}$$

Since the activation state of each hidden unit and visible unit is conditionally independent, therefore, when the state of the visible and hidden units is given, the activation probability of the first implicit unit and visible elements is

$$P(h_j = 1|v, \theta) = \sigma\left(b_j + \sum_{i=1}^{m} v_i w_{ij}\right),$$
$$P(v_i = 1|h, \theta) = \sigma\left(c_i + \sum_{j=1}^{n} h_j w_{ij}\right), \tag{7}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid activation function.



(a) $LBP_5^1$ (b) $LBP_5^2$ (c) $LBP_{16}^2$

Figure 3: Three types of LBP.

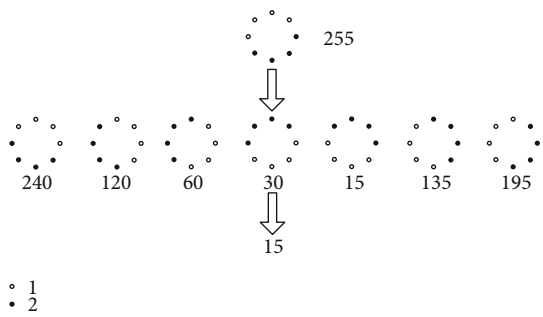



Figure 4: Rotation-invariant LBP schematic.

*3.4. Feature Point Detection and Matching.* In this paper, we propose a multifeature point fusion algorithm. The combination of the edge detection algorithm and the ORB detection algorithm enables the detection algorithm to extract the edge information, thereby increasing the number of matching points with fewer textures. The feature points of the edge are obtained by the Canny algorithm to ensure that the object with less texture has feature points. ORB have scale and rotation invariance, and the speed is faster than SIFT. The BRIEF description algorithm is used to construct the feature point descriptor [28–31].

The Brute force algorithm is adopted as the feature matching strategy. It calculates the Hamming distance between each point of the template image and each feature point of the sample image. Then compare the minimum Hamming distance value with the threshold value; if the distance is less than the threshold value, regard these two points as the matching points; otherwise, they are not matching points. The framework of feature extraction and matching is shown in Figure 6.

*3.5. Pose Estimation.* The core idea is to select four noncoplanar virtual control points; then, all the spatial reference points are represented by the four virtual control points, and then, the coordinates of the virtual control points are solved by the correspondence between the spatial reference points and the projection points, thereby obtaining the coordinates of all the spatial reference points. Finally, the rotation matrix and the translation vector are solved. The specific algorithm is described as follows.

Given $n$ reference points, the world coordinate is $\tilde{P}_i^w = (x_i, y_i, z_i)^T$, $i = 1, 2, \cdots, n$. The coordinates of the corresponding projection point in the image coordinate system are $\tilde{u}_i = (u_i, v_i)^T$, and the corresponding homogeneous coordinates are $P_i^w = (x_i, y_i, z_i, 1)^T$ and $u_i = (u_i, v_i, 1)^T$. The correspondence between the reference point $P_i^w$ and the projection point $u_i$:

$$\lambda_i u_i = K[R\,t]P_i^w, \tag{8}$$

where $\lambda_i$ is the depth of the reference point and $K$ is the internal parameter matrix of the camera:

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{9}$$

where $f = f_u = f_v$ is the focal length of the camera and $(u_0, v_0) = (0, 0)$ is the optical center coordinate.

First, select four noncoplanar virtual control points in the world coordinate system. The relationship between the virtual control points and their projection points is shown in Figure 7.

In Figure 7, $\mathbf{C}_1^w = [0, 0, 0, 1]^T$, $\mathbf{C}_2^w = [1, 0, 0, 1]^T$, $\mathbf{C}_3^w = [0, 1, 0, 1]^T$, and $\mathbf{C}_4^w = [0, 0, 1, 1]^T$. $\{\mathbf{C}_j^c, j = 1, 2, 3, 4\}$ are homogeneous coordinates of the virtual control point in the camera coordinate system, $\{\tilde{\mathbf{C}}_j^c, j = 1, 2, 3, 4\}$ is the corresponding nonhomogeneous coordinate, $\{\mathbf{C}_j, j = 1, 2, 3, 4\}$ is the homogeneous coordinate of the projection point corresponding in the image coordinate system, and $\{\tilde{\mathbf{C}}_j, j = 1, 2, 3, 4\}$ is the corresponding nonhomogeneous coordinate. $\{\mathbf{P}_i^c, i = 1, 2, \cdots, n\}$ is the homogeneous coordinate of the reference point in the camera coordinate system; $\{\tilde{\mathbf{P}}_i^c, i = 1, 2, \cdots, n\}$ is the corresponding nonhomogeneous coordinate. The relationship between the spatial reference points and the control points in the world coordinate is as follows:

$$\mathbf{P}_i^w = \sum_{j=1}^{4} \alpha_{ij} \mathbf{C}_j^w,\ i = 1, 2, \cdots, n, \tag{10}$$

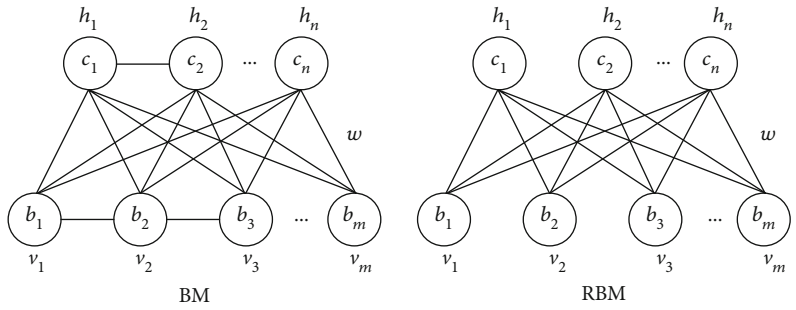



Figure 5: Boltzmann machine and restricted Boltzmann machine. $v$ is the visible layer, $m$ indicates the number of input data, $h$ is the hidden layer, and $w$ is the connection weight between two layers,$\forall i, j, v_i \in \{0, 1\}, h_j \in \{0, 1\}$.

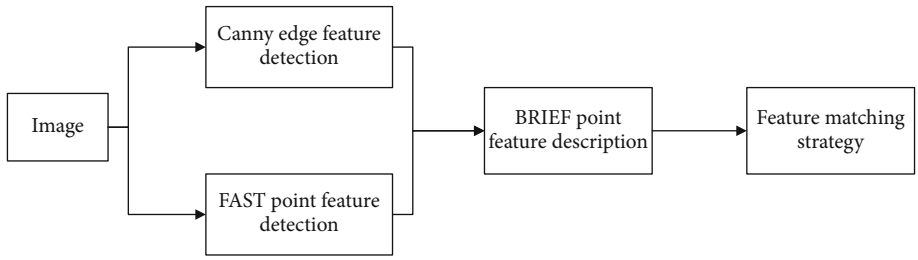



Figure 6: The process of multifeature fusion extraction and matching.

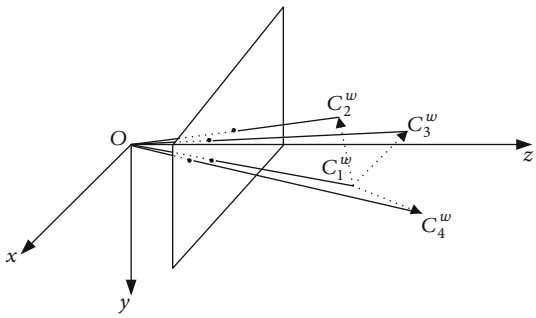



Figure 7: Virtual control point and its projection point correspondence.

where vector $[\alpha_{i1}, \alpha_{i2}, \alpha_{i3}, \alpha_{i4}]^T$ is the coordinate of the Euclidean space based on the control point $\mathbf{C}_i^c$. From the invariance of the linear relationship under the Euclidean transformation,

$$\begin{aligned} \mathbf{P}_i^c &= \sum_{j=1}^{4} \alpha_{ij} C_j^c, \quad i = 1, 2, \cdots, n, \\ \lambda_i u_i &= \mathbf{K}\tilde{\mathbf{P}}_i^c = \mathbf{K} \sum_{j=1}^{4} \alpha_{ij} \tilde{\mathbf{C}}_j^c, \quad i = 1, 2, \cdots, n. \end{aligned} \tag{11}$$

Assume $\tilde{\mathbf{C}}_j^c = [x_j^c, y_j^c, z_j^c]^T$, then

$$\lambda_i = \sum_{j=1}^{4} \alpha_{ij} z_j^c. \tag{12}$$

Then, obtain the equation:

$$\begin{aligned} \sum_{j=1}^{4} \alpha_{ij} f x_j^c - \alpha_{ij} u_i z_j^c = 0, \\ \sum_{j=1}^{4} \alpha_{ij} f y_j^c - \alpha_{ij} v_i z_j^c = 0. \end{aligned} \tag{13}$$

Assume $Z = [Z_1^{cT}, Z_2^{cT}, Z_3^{cT}, Z_4^{cT}]^T$, $Z_j^c = [f x_j^c, f y_j^c, z_j^c]^T$, $j = 1, 2, 3, 4$, then the equations are obtained from the correspondence between spatial points and image points as follows:

$$MZ = 0. \tag{14}$$

The solution $Z$ is the kernel space of the matrix $M$:

$$Z = \sum_{i=1}^{N} \beta_i W_i, \tag{15}$$

where $W_i$ is the eigenvector of $M^T M$, $N$ is the dimension of the kernel, and $\beta_i$ is the undetermined coefficient. For a perspective projection model, the value of $N$ is 1, resulting in

$$Z = \beta W, \tag{16}$$

where $W = [w_1^T, w_2^T, w_3^T, w_4^T]^T$, $w_j = [w_{j1}, w_{j2}, w_{j3}]^T$; then, the image coordinates of the four virtual control points are

$$\mathbf{c}_j = \left\{ \frac{w_{j1}}{w_{j3}}, \frac{w_{j2}}{w_{j3}}, 1 \right\}, \quad j = 1, 2, 3, 4. \tag{17}$$

The image coordinates of the four virtual control points obtained by the solution and the camera focal length obtained during the calibration process are taken into the absolute positioning algorithm to obtain the rotation matrix and the translation vector.

## 4. Experiments

We conducted two experiments to evaluate the proposed system. In the first experiment, we compare the proposed algorithm with other state-of-the-art algorithms on public datasets and then perform numerical analysis to show the accuracy of our system. The second experiment evaluated the performance of accuracy in the real world.

*4.1. Experiment Setup.* The experimental devices include an Android mobile phone (Lenovo Phab 2 Pro) and a depth camera (Intel RealSense D435) as shown in Figure 8. The user interface of the proposed visual positioning system on a smart mobile phone running in an indoor environment is shown in Figure 9.

*4.2. Experiment on Public Dataset.* In this experiment, we adopted the ICL-NUIM dataset which consists of RGB-D images from camera trajectories from two indoor scenes. The ICL-NUIM dataset is aimed at benchmarking RGB-D, Visual Odometry, and SLAM algorithms [32–34]. Two different scenes (the living room and the office room scene) are provided with ground truth. The living room has 3D surface ground truth together with the depth maps as well as camera poses and as a result perfectly suits not only for benchmarking camera trajectory but also for reconstruction. The office room scene comes with only trajectory data and does not have any explicit 3D model with it. The images were captured at 640∗480 resolutions.

Table 1 shows localization results for our approach compared with state-of-the-art methods. The proposed localization method is implemented on Intel Core i5-4460 CPU@3.20 GHz. The total procedure from scene recognition to pose estimation takes about 0.17 s to output a location for a single image.

*4.3. Experiment on Real Scenes.* The images are acquired by a handheld depth camera at a series of locations. The image size is 640 × 480 pixels, and the focal length of the camera is known. Several images of the laboratory are shown in Figure 10.

Using the RTAB-Map algorithm, we get the 3D point cloud of the laboratory. It is shown in Figure 11. The blue points are the position of the camera, and the blue line is the trajectory.

The 2D map of our laboratory is shown in Figure 12. The length and width of the laboratory are 9.7 m and 7.8 m, respectively. First, select a point in the laboratory as the origin of the coordinate system and establish a world coordinate system. Then, hold the mobile phone, walk along different routes, and take photos, respectively, as indicated by the arrows.

In the offline stage, we get a total of 144 images. Due to some images captured at different scenes being similar, we divide them into 18 categories. In the online stage, we captured 45 images at different locations on route 1 and 27 images on route 2. The classification accuracy formula is

$$P = \frac{N_i}{N}, \tag{18}$$

where $N_i$ is the correct classified number of scene images and $N$ is the total number of scene images. The classification accuracy of our method is 0.925.

Most mismatched scenes concentrate in the corner, mainly due to the lack of significant features or mismatches. Several mismatched scenes are shown in Figure 13.

After removing the wrong matched results, the error cumulative distribution function graph is shown in Figure 14.

The trajectory of the camera is compared with the predefined route. After calculating the Euclidean distance between the results through our method and the true position, we get the error cumulative distribution function


Figure 8: Intel RealSense D435 and Lenovo mobile phone.


Figure 9: The user interface of the proposed visual positioning system on a smart mobile phone running in an indoor environment.

Table 1: Comparison of mean error in ICL-NUIM dataset.

| Method | Living room | Office room |
|---|---|---|
| PoseNet | 0.60 m, 3.64° | 0.46 m, 2.97° |
| 4D PoseNet | 0.58 m, 3.40° | 0.44 m, 2.81° |
| CNN+LSTM | 0.54 m, 3.21° | 0.41 m, 2.66° |
| Ours | 0.48 m, 3.07° | 0.33 m, 2.40° |

Figure 10: Images captured from different scenes.

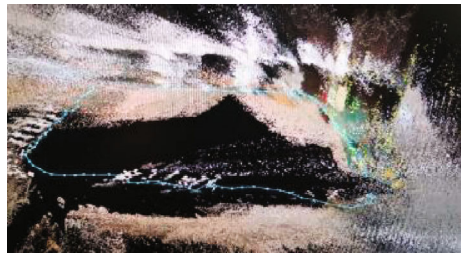


Figure 11: 3D point cloud of laboratory.

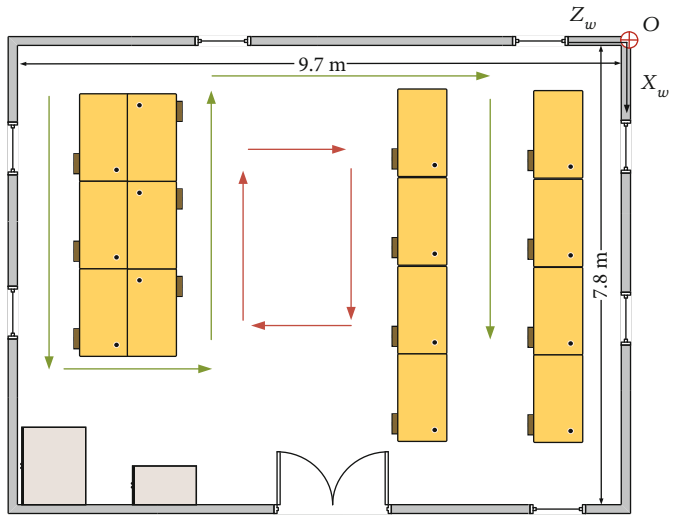



Figure 12: Environmental map and walking route.

Figure 13: Mismatched scene.

graph (Figure 14). It can be seen that the average positioning error is 0.61 m. Approximately 58% point positioning error is less than 0.5 m, about 77% point error is less than 1 m, about 95% point error is less than 2 m, and the maximum error is 2.55 m.

Since the original depth images in our experiment are based on RTAB-Map, its accuracy is not accurate. For example, in an indoor environment, intense illumination and strong shadows may lead to inconspicuous local features. It is also difficult to construct a good point cloud model. In

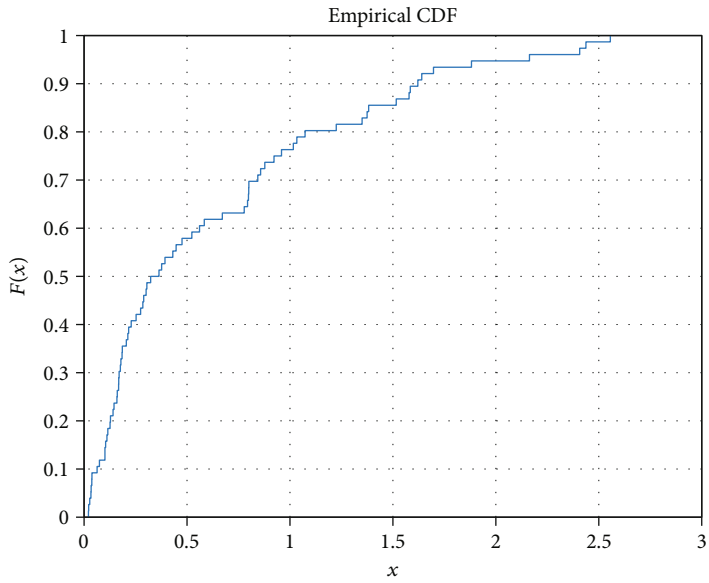



Figure 14: Error cumulative distribution function graph.

the future, we plan to use laser equipment to construct a point cloud.

## 5. Conclusions and Future Work

In this article, we have presented an indoor positioning system based only on cameras. The main work is to use deep learning to identify the category of the scene and use 2D-3D matching feature points to calculate the location. We implemented the proposed approach on a mobile phone and achieved a positioning accuracy of decimeter level. The preliminary indoor positioning experiment result is given in this paper. But the experimental site is a small-scale place. The following work needs to be done in the future: with the rapid development of deep learning, it can generate high-level semantics and effectively solve the limitations caused by artificial design features, use a more robust lightweight image retrieval algorithm, and carry out tests under different lighting and dynamic environments, system tests under large-scale scenarios, and long-term performance tests.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This study was partially supported by the Key Research Development Program of Hebei (Project No. 19210906D).

## References

[1] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, "Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2389–2406, 2018.

[2] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe, "Alps: s bluetooth and ultrasound platform for mapping and localization," in *Proceedings of the 13th ACM Conferenceon Embedded Networked Sensor Systems, ACM*, pp. 73–84, New York, NY, USA, 2015.

[3] S. He and S. Chan, "Wi-Fi fingerprint-based indoor positioning: recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2017.

[4] C. L. Wu, L. C. Fu, and F. L. Lian, "WLAN location determination in e-home via support vector classification," in *Networking Sensing and Control, IEEE International Conference, 2004*, pp. 1026–1031, Taipei, Taiwan, 2004.

[5] G. Ding, Z. Tan, J. Wu, and J. Zhang, "Efficient indoor fingerprinting localization technique using regional propagation model," *IEICE Transactions on Communications*, vol. 8, pp. 1728–1741, 2014.

[6] G. Ding, Z. Tan, J. Wu, J. Zeng, and L. Zhang, "Indoor fingerprinting localization and tracking system using particle swarm optimization and Kalman filter," *IEICE Transactions on Communications*, vol. 3, pp. 502–514, 2015.

[7] C. Toft, W. Maddern, A. Torii et al., "Long-term visual localization revisited," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2020.

[8] A. Xiao, R. Chen, D. Li, Y. Chen, and D. Wu, "An indoor positioning system based on static objects in large indoor scenes by using smartphone cameras," *Sensors*, vol. 18, no. 7, pp. 2229–2246, 2018.

[9] E. Deretey, M. T. Ahmed, J. A. Marshall, and M. Greenspan, "Visual indoor positioning with a single camerausing PnP," in *In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–9, Banff, AB, Canada, October 2015.

[10] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2969–2976, Colorado Springs, CO, USA, 2011.

[11] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2d-to-3dmatching," in *2011 IEEE International Conference on Computer Vision, IEEE*, pp. 667–674, Barcelona, Spain, 2011.

[12] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3d point clouds," in *European Conference on Computer Vision (ECCV)*, Berlin, Heidelberg, 2012.

[13] M. Larsson, E. Stenborg, C. Toft, L. Hammarstrand, T. Sattler, and F. Kahl, "Fine-grained segmentation networks: self-supervised segmentation for improved long-term visual localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 31–41, Seoul, Korea, 2019.

[14] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool, "Night-to-day image translation for retrieval-based localization," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5958–5964, Montreal, QC, Canada, 2019.

[15] J. X. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: large-scale scene recognition from abbey to zoo," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, San Francisco, CA, USA, 2010.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[17] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarseto fine: robust hierarchical localization at large scale," in *Proceedingsof the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, California, 2019.

[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 Transactions on Sensor Networks*, vol. 15, no. 2, pp. 1–31, 2019.

[19] Y. Chen, R. Chen, M. Liu, A. Xiao, D. Wu, and S. Zhao, "Indoor visual positioning aided by CNN-based image retrieval: training-free, 3D modeling-free," *Sensors*, vol. 18, no. 8, pp. 2692–2698, 2018.

[20] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *IEEE International Conference on Robotics & Automation*, pp. 4762–4769, Stockholm, Sweden, 2016.

[21] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 9, pp. 1744–1756, 2016.

[22] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-scale localization for cameras with known vertical direction," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 7, pp. 1455–1461, 2016.

[23] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera pose voting for large-scale image-based localization," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2704–2712, Santiago, Chile, 2015.

[24] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: a convolutional network for real-time 6-dof camera relocalization," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2938–2946, Santiago, Chile, 2015.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, Salt Lake City, Utah, 2018.

[26] Z. Chen, A. Jacobson, N. Sunderhauf et al., "Deep learning features at scale for visual place recognition," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017.

[27] S. Lynen, B. Zeisl, D. Aiger et al., "Large-scale, real-time visual–inertial localization revisited," *The International Journal of Robotics Research*, vol. 39, no. 9, pp. 1–24, 2020.

[28] M. Dusmanu, I. Rocco, T. Pajdla et al., "D2-net: a trainable cnn for joint description and detection oflocal features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8092–8101, California, 2019.

[29] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *IEEE International Conference on Robotics and Automation*, pp. 1848–1853, Kobe, Japan, 2009.

[30] A. Xu and G. Namit, "SURF: speeded-up robust features," *Computer Vision & Image Understanding*, vol. 110, no. 3, pp. 404–417, 2008.

[31] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision*, pp. 2564–2571, Barcelona, Spain, 2012.

[32] A. Handa, T. Whelan, J. Mcdonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *2014 IEEE international conference on Robotics and automation (ICRA)*, pp. 1524–1531, Hong Kong, China, 2014.

[33] M. Labbe and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[34] Z. Gao, Y. Li, and S. Wan, "Exploring deep learning for view-based 3D model retrieval," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 1, pp. 1–21, 2020.