Hindawi
Advances in Multimedia
Volume 2022, Article ID 3941995, 10 pages
https://doi.org/10.1155/2022/3941995

Research Article

An Improved VSLAM for Mobile Robot Localization in Corridor Environment

Gengyu Ge,[1,2] **Zhong Qin**,[1] **and Lilve Fan**[1]

[1]*School of Information Engineering, Zunyi Normal University, Zunyi 563006, China*
[2]*School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

Correspondence should be addressed to Gengyu Ge; gegengyu_2021@163.com

Received 25 February 2022; Revised 12 April 2022; Accepted 6 May 2022; Published 23 May 2022

Academic Editor: Qiangyi Li

Localization is a fundamental capability for an autonomous mobile robot, especially in the navigation process. The commonly used laser-based simultaneous localization and mapping (SLAM) method can build a grid map of the indoor environment and realize localization task. However, when a robot comes to a long corridor where there exists many geometrically symmetrical and similar structures, it often fails to position itself. Besides, the environment is not represented to a semantic level that the robot cannot interact well with users. To solve these crucial issues, in this paper, we propose an improved visual SLAM approach to realize a robust and precise global localization. The system is divided into two main steps. The first step is to construct a topological semantic map using visual SLAM, text detection and recognition, and laser sensor data. The second step is the localization which repeats part work of the first step but makes the best use of the prebuilt semantic map. Experiments show that our approach and solutions perform well and localize successfully almost everywhere in the corridor environment while traditional methods fail.

1. Introduction

Nowadays, many commercial service robots are used for transporting goods in restaurants, hotels, and hospitals, especially during the COVID-19 epidemic time. Among them, the localization research for autonomous mobile robot navigation in a man-made structural environment is an ongoing challenge. In indoor scenes, the most used method is using a 2D laser rangefinder and laser-based SLAM to construct a 2D occupancy grid map [1, 2]. Then the mobile robot performs localization task by AMCL algorithm, which is a particle filter solution [3]. However, when the robot is in a long corridor, the mapping result is always shorter than the real scene and the localization is inaccurate. More seriously, it is easy for the localization process to fall into a symmetrical or similarly wrong position [4]. The reason is that, for the laser sensor used in the symmetrical and similar long corridor environment, the data collected at different times are similar. Therefore, the mobile robot can not get accurate pose information when it performs global localization tasks; besides, it is easy for the robot to converge to the wrong unimodal distribution. More than that, the number of particles increases with the size of the map; then the computing and memory costs also increase. Of all the shortcomings, the main one is the limited amount of data information collected by a 2D laser sensor.

In comparison to a 2D laser sensor, cameras provide more dense information such as point features, textures, lines, objects, and texts. It is one of the most potential sensors that can be used to perceive the environment and localize the pose for mobile robotics.

To address these problems, we propose a novel visual SLAM-based method for mobile robot localization, which is assisted by text information and laser data features extracted from the indoor scene, especially the long, symmetrical, and similar corridor environment. Firstly, the mobile robot initializes the system at the starting position. Secondly, it moves along the middle line of the corridor and constructs a features-based visual map using the visual SLAM method. Thirdly, the robot stops when passing through door areas, rotates the camera toward the doorplate, and records the text content together with the current keyframe node. Lastly, the mobile robot navigates and localizes itself according to the previously built map. The whole framework of the mapping and localization system is depicted in Figure 1.

This paper is organized as follows. The related work of visual SLAM, text detection, and recognition are discussed in Section 2. A proposed method using visual SLAM for localization is presented in Section 3. The experiments and discussion are described in Section 4 and Section 5 concludes the paper.

2. Related Work

2.1. Visual SLAM. In the field of visual SLAM, feature-based indirect approaches and photometric-based direct methods are currently two mainstream techniques. The former one extracts salient image features, like points, lines, and planar features, to realize the target. By minimizing the reprojection errors of the matched feature pairs, the camera motion and the depth of map points can be computed. PTAM [5] firstly used the parallel threads to solve the visual SLAM problem; it estimated pose in the tracking thread and refined camera motion in the local mapping thread. ORB-SLAM series [6–8] adopted the idea of parallel threads, added a loop closure thread, and used ORB features in the overall process. It is a state-of-the-art solution in the research field and can be used directly in applications.

The direct methods solve pose estimation by minimizing the pix-level intensity errors from two adjacent images. LSD-SLAM [9] built a semidense map compared to the sparse points map by feature-based SLAM. However, it still needed features extraction for loop closure purposes. SVO [10] used a depth filter model to estimate the depth and filtered outliers. It tracked sparse pixels using the FAST corners and modeled the triangulated depth observations with a Gaussian Uniform distribution. DSO was direct sparse odometry and a probabilistic model without computing keypoints or descriptors [11].

2.2. Text Detection and Recognition. Text signs in the indoor environment have semantic content, which makes easy to realize human-computer interaction. To achieve and understand the scene text information, the OCR techniques usually have two phases, text detection and text recognition, respectively [12].

Text detection is to distinguish text regions from the background of a captured image. Traditional methods based on manually designed features perform well when there exists an obvious construct between the text region and the background part. The stroke width transform (SWT) employed a local image operator to compute the width of approximately textual pixels, searched letter candidates, and grouped letters into text lines [13]. The maximally stable extremal region (MSER) had a real-time detection performance [14]. In addition, it was robust to blur, illumination, and color variation. The morphology-based method extracted high contrast areas as text line candidates [15] and was also invariant to different image changes like lighting, translation, rotation, and complicated backgrounds. CTPN [16] was a connectionist text proposal network and used CNN to detect a text line in a sequence of fine-scale text proposals which were then connected naturally with RNN. The other famous works based on deep learning networks were EAST [17] and IncepText [18].

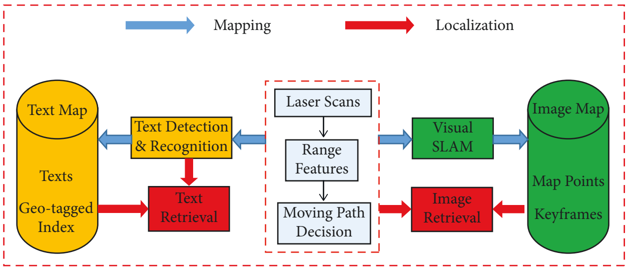


Figure 1: Framework of the mapping and localization system.

There are many open-source OCR engines and software using traditional text recognition algorithms can be used, such as Tesseract, Google Docs OCR, and Transym [19]. These methods have relatively high accuracy when the text region has large contrast with the backgrounds and simple text lines. In addition, it does not need a GPU configuration. If the scene texts are multiple fonts, colorful, and complicated backgrounds, then the deep learning approaches based on GPU will be essential. There are two mainstream solutions based on CNN and RNN. The first one uses CNN to extract image features and combines RNN with connectionist temporal classification (CTC) to predict sequence, like CRNN [20]. Another one employs CNN, the Seq2Seq model, and Attention framework [21], which includes encoder and decoder, which adopts ideas from machine translation techniques.

3. Method

When the mobile robot comes into a new environment for the first time, it needs to construct a map and then navigate in the environment according to the built map. We use a monocular camera to perform visual SLAM for both mapping and localization purposes. Different from the traditional laser-based SLAM method, our method uses laser sensor data only for basic geometry features extraction, such as door area, middle of the corridor, or end of the corridor. In addition, text information extracted from the doorplates region is used for semantic localization. The detailed descriptions are as follows.

3.1. Moving Strategy. In the map building phase, the mobile robot mainly runs in the SLAM mode utilizing the ORB-SLAM framework. It will create a sparse feature map of the environment from scratch or incrementally update an existing map. However, most of the visual SLAM solutions are used for handheld mode or driverless scenes. In these cases, the moving trajectory of the camera will always be artificially moved on a fixed route and will certainly not lose the tracking of the route according to a prebuilt map. When it comes to the applications of autonomous mobile robots, for instance, the delivery robot moves in a corridor and needs to avoid obstacles when encountering pedestrians; changing the moving route will lose visual feature tracking. Consequently, the mobile robot needs to move within a fixed route range, preferably the middle line of the corridor, as shown in Figure 2, where a robot passes through a doorway area. Three

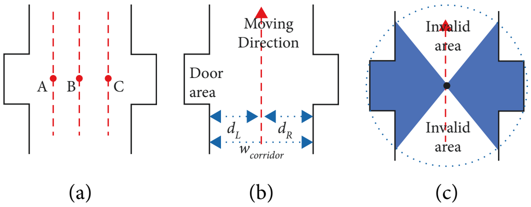


Figure 2: Mobile robot passes through the doorway area. (a) Different routes; (b) annotation data; (c) valid and invalid laser scanning areas.

routes indicated by red dotted lines are shown in Figure 2(a), and three positions are indicated by red points. There are only a few matched pairs between the visual features extracted from the image captured in position A and those in position C. Besides, the cameras oriented to different directions also have different matching results, even if in the same position. Therefore, if the mobile robot moves along the left red dotted line as the route and maps the corridor using visual SLAM, it will fail to localize itself when moving along the right red dotted line in the subsequent process. Ideally, the mobile robot moves along the middle red dotted line in Figure 2(a).

To ensure the mobile robot moves along a relatively fixed route, a laser range finder is used to measure the ranges from the center of the sensor to the obstacle, thus analyzing a high-precision position relative to the walls on both sides. As shown in Figure 2(b), the left detected distance plus the right one equals the width of the corridor.

$$d_L + d_R = w_{\text{corridor}}, \tag{1}$$

where d_L means the shortest distance between the left wall and the center of the laser sensor, and d_R means the one of another side. When the robot passes through the doorway area, the following equation holds:

$$d_L + d_R = w_{\text{corridor}} + 2^{*} d_{\text{door}}. \tag{2}$$

Consequently, when the mobile robot determines that it is in the corridor area, it is easy for the robot to move along the middle line of the corridor. The robot can slightly adjust its position so that the data measured by the laser sensor satisfy the following equation:

$$d_L = d_R. \tag{3}$$

The next problem that needs to be solved is to determine whether the robot is in the corridor area. In most indoor scenes, the laser sensor can get whole valid distance data around the robot; if not, the robot is most likely in the corridor area, as shown in Figure 2(c) where the laser sensor gets two invalid range data regions. The blue areas belong to the range that can be covered by the scanning radius of the laser sensor.

Given a 2D laser range finder that has a measurement angle range of 360 degrees, the maximum measuring distance is defined as $\mathrm{D}_{\max}$, the minimum measuring distance as $D_{\min}$, and the angular resolution as $\phi_{\min}$. Then the raw data can be described by the following formula:

$$R = \{(r_i, \phi_i) \mid i = 1, 2, \ldots, N\}, \tag{4}$$

where N equals the value $360/\phi_{\min}$ and represents the total number of scanning points $P = \{p_1, p_2, \ldots, p_N\}$. An invalid scan area angle θ_{null} has a dynamically variable range. The maximum value $\theta_{\max}$ is got when the laser sensor mounted on the robot is close to the wall, while the minimum value $\theta_{\min}$ is got when the robot is located on the middle line along the corridor. The two values are defined as

$$\begin{cases} \theta_{\max} = \arcsin \dfrac{w_{\text{corridor}}}{D_{\max}}, \\ \\ \theta_{\min} = \arcsin \dfrac{w_{\text{corridor}}}{2D_{\max}}. \end{cases} \tag{5}$$

If the mobile robot is moving in a corridor, then (6) holds.

$$\theta_{\min} \leq \theta_{\text{null}} \leq \theta_{\max}. \tag{6}$$

We count the constant number of invalid return distances and compute the invalid area angle θ_{null} according to

$$\theta_{\text{null}} = \left| \phi_i - \phi_j \right|. \tag{7}$$

where ϕ_{i} means that the *i-th* angle whose return distance value from the detected point $\mathrm{p_i}$ is invalid and ϕ_{j} represents the *j-th*. The constant angle range between the two also has invalid return distance values.

It is easy to find that the mobile robot is at the end of the corridor if only one invalid area satisfies (6). Similarly, if two invalid areas satisfy (6) and are distributed like Figure 2(c), then the robot is most likely located in the middle of the corridor.

According to the above information extracted from the laser scanning data, the mobile robot can autonomously move to the free area along a preset fixed route. In addition, the robot can get a coarse position estimation relative to the corridor.

3.2. Build an Image Map

3.2. Build an Image Map. The next work is using visual SLAM to construct a visual features map along the fixed route which is got based on the previous laser data.

The motion of a single camera should be solved by the principle of epipolar geometry, as shown in Figure 3. Define x_2 as the coordinate on the normalized plane of the pixel point p_2 in the current image I_2, and x_1 is the pixel p_1 in the previous image I_1. The two points are matched by visual feature; then, the following holds:

$$\begin{cases} x_1 = [u_1, v_1, 1]^T, \\ x_2 = [u_2, v_2, 1]^T, \\ s_1 x_1 = s_2 R x_2 + t, \\ x_2^T t^{\wedge} R x_1 = 0, \\ E = t^{\wedge} R, \\ F = K^{-T} \mathrm{E} K^{-1}, \end{cases} \tag{8}$$

where R means the camera rotation motion, t is the translation, E represents the essential matrix, and F is the

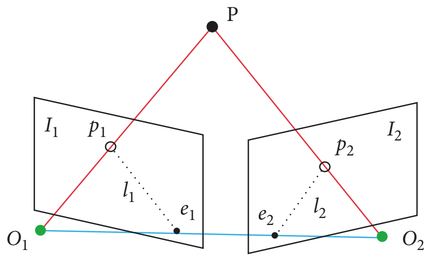


Figure 3: One point in two adjacent image frames (epipolar geometry).

fundamental matrix. If the parameters of the essential matrix are calculated, then the basic matrix is easy to solve. Similarly, the rotation and translation matrices can be obtained by decomposing the essential matrix.

The essential matrix E is a 3×3 matrix. Consider a pair of matching points whose normalized coordinates are $x_1 = [u_1, v_1, 1]^T$ and $x_2 = [u_2, v_2, 1]^T$. According to the polar constraint, the following equation holds:

$$(u_1, v_1, 1) \times \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} \times \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = 0. \tag{9}$$

Expand matrix E and write it in the form of vector; then put all the points into one equation to become a system of linear equations as (10) shows. u_i and v_i represent the i-th feature point. The coefficient matrix of linear equations consists of the position of characteristic points, with a size of 8×9. If the matrix composed of eight pairs of matching points satisfies the condition of rank 8, then the elements of E can be solved by this equation.

$$\begin{pmatrix} u_1^1u_2^1 & u_1^1v_2^1 & u_1^1 & v_1^1u_2^1 & v_1^1v_2^1 & v_1^1 & u_2^1 & v_2^1 & 1 \\ u_1^2u_2^2 & u_1^2v_2^2 & u_1^2 & v_1^2u_2^2 & v_1^2v_2^2 & v_1^2 & u_2^2 & v_2^2 & 1 \\ u_1^3u_2^3 & u_1^3v_2^3 & u_1^3 & v_1^3u_2^3 & v_1^3v_2^3 & v_1^3 & u_2^3 & v_2^3 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_1^8u_2^8 & u_1^8v_2^8 & u_1^8 & v_1^8u_2^8 & v_1^8v_2^8 & v_1^8 & u_2^8 & v_2^8 & 1 \end{pmatrix} \times \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \end{pmatrix} = 0. \tag{10}$$

According to the estimated essential matrix E and the camera motion R, t can be recovered. This process is obtained by singular value decomposition (SVD), as follows:

$$\begin{cases} E = U\Sigma V^T, \\ t_1^\wedge = \mathrm{UR}_Z\left(\dfrac{\pi}{2}\right)\Sigma U^T, \\ t_2^\wedge = \mathrm{UR}_Z\left(-\dfrac{\pi}{2}\right)\Sigma U^T, \\ R_1 = \mathrm{UR}_Z^T\left(\dfrac{\pi}{2}\right)V^T, \\ R_2 = \mathrm{UR}_Z^T\left(-\dfrac{\pi}{2}\right)V^T, \end{cases} \tag{11}$$

where U and V are orthogonal matrices, Σ is a singular value matrix, and there are four possible solutions in the SVD decomposition. Point P in world coordinate system has a positive depth in both cameras, so the depth of the point under the two cameras can be used as the basis for judging the positive solution. The final decomposition result is shown in

$$\begin{cases} \Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3), \ \ \sigma_1 \geq \sigma_2 \geq \sigma_3, \\ E = \mathrm{Udiag}\left(\dfrac{\sigma_1 + \sigma_1}{2}, \dfrac{\sigma_1 + \sigma_1}{2}, 0\right)V^T. \end{cases} \tag{12}$$

If all feature points in the scene fall on the same plane, then motion estimation can be carried out through homography and the following equation holds:

$$\begin{cases} n^T P + d = 0, \\ p \cong 2K\left(R - \dfrac{\mathrm{tn}^T}{d}\right)K^{-1}p_1. \end{cases} \tag{13}$$

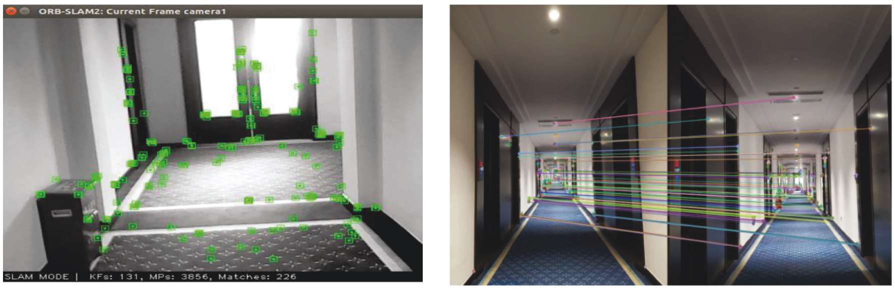
Homography matrix is related to rotation, translation, and plane parameters. Solving motion is similar to essence matrix E. According to matching point pairs and (14) and (15), the H is decomposed to calculate rotation and translation.

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \times \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}. \tag{14}$$

A set of matching point pairs can construct three constraints (only two are linearly independent), so the homography matrix with a degree of freedom of 8 can be calculated through four pairs of matching feature points (these feature points cannot have three collinear points), that is, solve the following linear equations (when $h\ 9 = 0$, the right side is zero).

$$\begin{pmatrix} u_1^1 & v_1^1 & 1 & 0 & 0 & 0 & -u_1^1u_2^1 & -v_1^1u_2^1 \\ 0 & 0 & 0 & u_1^1 & v_1^1 & 1 & -u_1^1v_2^1 & -v_1^1v_2^1 \\ u_1^2 & v_1^2 & 1 & 0 & 0 & 0 & -u_1^2u_2^2 & -v_1^2u_2^2 \\ 0 & 0 & 0 & u_1^2 & v_1^2 & 1 & -u_1^2v_2^2 & -v_1^2v_2^2 \\ u_1^3 & v_1^3 & 1 & 0 & 0 & 0 & -u_1^3u_2^3 & -v_1^3u_2^3 \\ 0 & 0 & 0 & u_1^3 & v_1^3 & 1 & -u_1^3v_2^3 & -v_1^3v_2^3 \\ u_1^4 & v_1^4 & 1 & 0 & 0 & 0 & -u_1^4u_2^4 & -v_1^4u_2^4 \\ 0 & 0 & 0 & u_1^4 & v_1^4 & 1 & -u_1^4v_2^4 & -v_1^4v_2^4 \end{pmatrix} \times \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{pmatrix} = \begin{pmatrix} u_2^1 \\ v_2^1 \\ u_2^2 \\ v_2^2 \\ u_2^3 \\ v_2^3 \\ u_2^4 \\ v_2^4 \end{pmatrix}. \tag{15}$$

In monocular SLAM, the depth information of pixels cannot be obtained only through a single image, and the depth of map points needs to be estimated by triangulation. Triangulation refers to determining the distance of the same point by observing the included angle of the same point at two places. According to (8) and (15), we can calculate the world coordinate value of map points. However, due to the influence of noise, the two lines often cannot intersect. Therefore, it can be solved by the least square method.


ORB-SLAM2: Current Frame camera1
SLAM MODE | KFs: 131, MPs: 3856, Matches: 226

(a)

(b)

Figure 4: (a) ORB features and (b) matching pair.

$$\begin{cases} s_1 x_1 \wedge x_1 = 0, \\ s_2 x_1 \wedge R x_2 + x_1 \wedge t = 0. \end{cases} \tag{16}$$

We utilize the ORB-SLAM open-source library which extracts the ORB features to match the adjacent images. ORB features combine the oriented FAST detector with a rotated BRIEF descriptor and have low computational consumption, compared with SIFT and SURF features. To add an efficiently computed orientation to the FAST keypoint, an intensity centroid is designed for computing a vector. The moments of a patch around the FAST keypoint are defined as

$$m_{\mathrm{pq}} = \sum_{x,y} x^p y^q I(x, y), \tag{17}$$

where the value of p and q can only be limited to 0 or 1. Then, the intensity centroid is computed from those moments:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right). \tag{18}$$

The orientation vector can be constructed by connecting two points, one of them is the keypoint corner's center, and the other is the centroid of the patch. Then, the orientation is computed as shown in the following:

$$\theta = \arctan \left(m_{01}, m_{10} \right). \tag{19}$$

The work after keypoint detection is feature description which is convenient for feature matching. An original BRIEF descriptor is variant to in-plane rotation. It is a binary description of an image patch using a binary intensity test τ which is defined as follows:

$$\tau(p; x, y) := \begin{cases} 1, & p(x) < p(y), \\ 0, & p(x) \geq p(y), \end{cases} \tag{20}$$

where p(x) and p(y) are the intensity at point x and point y. There are usually 256 pairs of points selected to express a keypoint. Then, the descriptor is defined as a vector of 256 binary tests.

$$f_n(p) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i), \tag{21}$$

where n equals 256. Then, a learning method that uses PCA or other dimensionality reduction strategies is utilized to assist in realizing a rotation-invariant BRIEF descriptor. The combination of oFAST and rBRIEF is called ORB feature, and an example of the ORB features and matching pair is shown in Figure 4, which has eliminated the mismatched point pairs by using the RANSAC algorithm [22].

Then the depth of map points in the world coordinate can be computed by using the triangulation method. The constructed map of a corridor is shown in Figure 5, which has sparse points. The blue trapezoidal blocks are camera poses that represent the keyframes of those captured images and will be saved as one part of an image map. The green one means the current frame.

3.3. Extend to a Semantic Map. When the mobile robot performs a cargo transportation task and needs to interact with the user, it is more convenient to use the semantic map closed to human language expression and understanding. To realize this function, text detection and recognition techniques must be used to achieve the text-level information.

The text characters in the indoor environment are usually on the doorplate, room nameplate, signs, or billboard on the wall. The most common is the room number, which can uniquely determine the location of a room. Consequently, the mobile robot just needs to detect and recognize the text information in the door area and extract the room number; then the position of the robot can be determined.

However, the camera mounted on the robot platform captures images in real time and continuously. Most of the images do not have useful text information; processing these images will be a waste of time and computing power. Besides, the text information of a room number would probably only be partially captured in one image; this will lead to misjudgment of the whole semantic result. To solve this problem, we use the laser scanning data to get a preliminary judgment where the doorway area is in the corridor. Then the robot moves to a position facing the door and stops to capture an image with the best perspective. Lastly, detect the text region and recognize the correct room number.

We use two traditional approaches and one deep learning method to detect text regions in a corridor environment, respectively. Figure 6 shows the three detected results; it is obvious that the deep learning method has the best and most accurate detected box. Figure 6(a) is the detected result by using the morphological method which has wrong boxes including the door handle and other text-

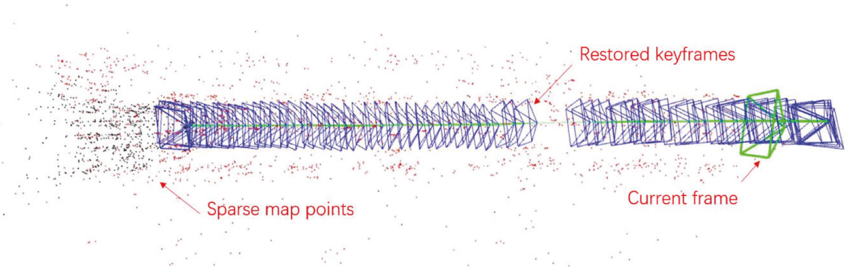
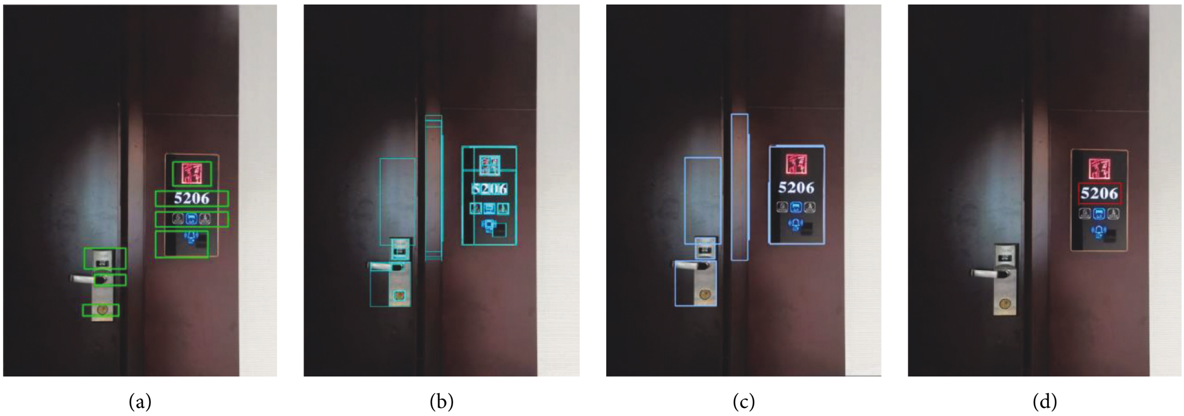


Figure 5: Image map with map points and keyframes.



(a) (b) (c) (d)

Figure 6: Text detection results. (a) Morphological method; (b) MSER; (c) MSER and NMS; (d) deep learning.

like signs. Figure 6(b) uses the MSER approach and includes texture area affected by illumination. Figure 6(c) is a processed result from Figure 6(b) by using the nonmaximum suppression (NMS) algorithm [23]. The red rectangle in Figure 6(d) is the ideal result which is got by using the deep learning approach.

The subsequent recognizing phase is relatively easy if a correct text box is detected. In addition, the text information we need is only Arabic numerals. Many open-source OCR solutions can realize this function and have high accuracy [24]. However, to get a high overall text detection and recognition accuracy, we choose to utilize an online deep learning scheme which is called EasyDL from Baidu company [25], https://ai.baidu.com/tech/ocr/general. Thus, we do not need to use GPU as the deep learning processing module, which has high power consumption and cost. Instead, only a WIFI module is needed to access the Internet.

After achieving the text information, an extended semantic map can be accomplished and is shown in Figure 7. The node represents a geo-tagged place which has doors or an intersection, and the edge means a passable route. The images in the node are decided according to the keyframe selection strategy of ORB-SLAM. Three keyframes in each doorway or intersection area are chosen to generate a node. If a corridor is in the form of a straight line or circular, then the semantic map can be represented by the data structure of a two-way linked list. Other cases can be expressed as multilayer quadtree.

3.4. *Path Following.*

When a semantic map is constructed and the robot's initial pose is known, the navigation task is a path following problem. The path planning algorithm varies with the form of a map. The commonly used data structures are a two-way linked list, multifork tree, and graphs. Consequently, it is a look-up problem and finding the shortest path.

Compared with nodes in a social network or an electronic map of a city, the geo-tagged nodes in an indoor environment used by a personal robot are usually very few. Therefore, search algorithms commonly used in data structures are sufficient.

3.5. *Localization Mode.*

In the localization or navigation mode using the visual SLAM method, the system firstly loads the previously built sparse feature map, then extracts features of a newly captured image, and matches with those from the image map. The difficult thing is not the image matching, but how to capture images in a fixed route and direction that the robot ever traveled before. The method is using laser data to get a coarse place judgment according to (1)–(8).

Then the localization problem for an autonomous mobile robot becomes the problem of vision based global localization using a visual vocabulary [26]. For a geo-tagged place, the robot needs to perform a text retrieval task. Equation (22) shows a mathematical form to express an image. The image is expressed as a vector v_A which consisted

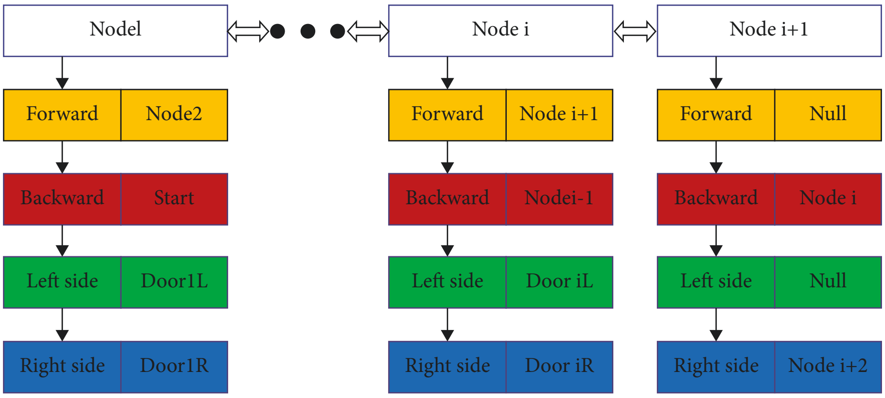


Figure 7: Topological semantic map diagram.

of N-words and the weight η_i is computed from the product of term frequency TF_i and the inverse document frequency IDF_i [27].

$$\begin{cases} \mathrm{TF}_i = \dfrac{n_i}{n}, \\ \mathrm{IDF}_i = \log\dfrac{n}{n_i}, \\ \eta_i = \mathrm{TF}_i \times \mathrm{IDF}_i, \\ v_A = \{(w_1,\eta_1), (w_2,\eta_2), \ldots, (w_N,\eta_N)\}. \end{cases} \tag{22}$$

The similarity of a new image translated to a form of bag-of-words with the image database can be computed using (23). When the robot moves to a doorway area, laser data feature extraction, visual image vocabulary matching, and text detection and recognition are combined to determine the localization result.

$$s(v_A - v_B) = 2\sum_{1}^{N} |v_{\mathrm{Ai}}| + |v_{\mathrm{Bi}}| - |v_{\mathrm{Ai}} - v_{\mathrm{Bi}}|. \tag{23}$$

4. Experiments and Discussion

Our experimental platform is a two-wheel differential driving mobile robot equipped with a RPLIDAR A2 and a single perspective camera. The laser sensor has a 360-degree scanning rotation range and 8–12 meters' distance. The CPU is ARM Cortex-A72 and the RAM memory size is 8 GB. The experimental environment is a long, symmetrical, and similar corridor inside a multiple-story hotel. The length of the corridor is 45 meters and the width is 1.85 meters. Figure 8 shows the corridor and platform. The laser sensor is used for collision detection and basic structural features extraction.

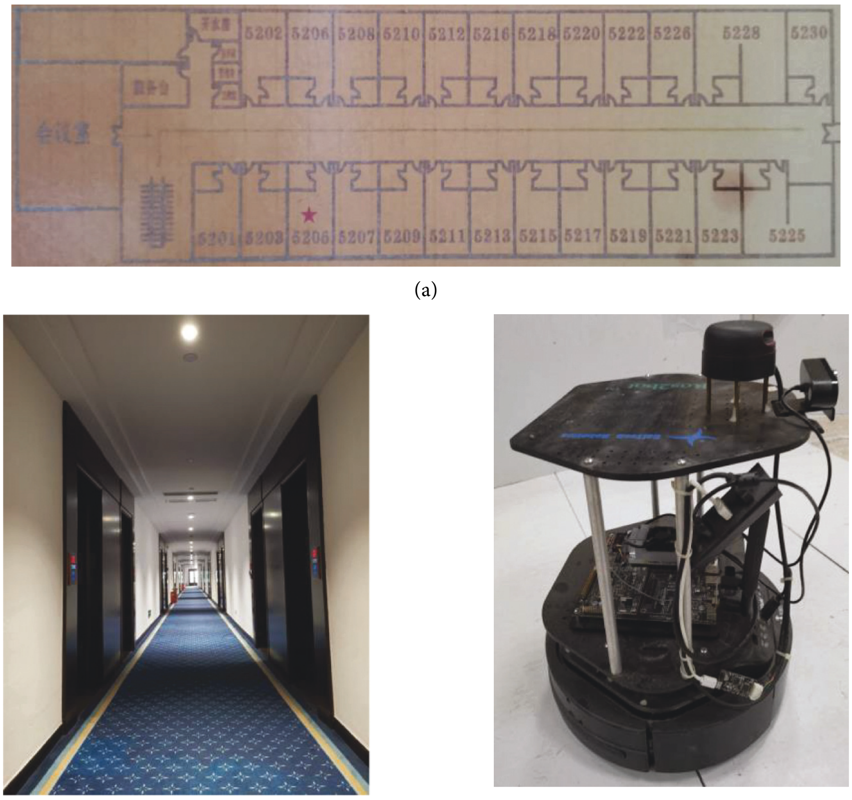
4.1. Lateral Deviation of Fixed Route. The mobile robot moves follow a fixed route which is the middle line along the corridor. According to the laser data computing and features extraction, the mobile robot adjusts its moving direction; a trajectory is shown in Figure 9.

From Figure 9, we find that the precision of laser data is very high, which is suitable for applications with accurate distance requirements.

4.2. Text Detection and Recognition. We collected 150 images from 30 doorway areas in a fifth-floor corridor inside the hotel. Five images in each area are collected from different perspectives and positions, but all of them are roughly facing the door. Because GPU device is not used in the microcomputer system, the offline deep learning model is not adopted for comparative experiments. Firstly, different text detection methods are tested and the average detected text boxes per image are counted. Table 1 shows the result which demonstrates that traditional methods detect many wrong areas as the text regions while online deep learning method performs better. Many lines or outlines in the image are misidentified as text information like numerals or characters. Consequently, WIFI based cloud platform solution is the best solution.

The second experiment is a comparison of the text recognition methods. The accuracy or successful recognition rate is calculated by counting the number of correct character recognitions out of the total tests (150 recognition experiments from 30 doorway areas). We used traditional method solution Tesseract and the online deep learning method to recognize the previously detected text boxes, respectively. Table 2 shows the recognition results. Compared with the traditional method, our online deep learning method achieved high accuracy.

4.3. Global Localization. In order to achieve a quantitative evaluation of the global localization performance, we placed the robot in 20 different positions in the corridor and used four different methods to perform global localization task, respectively. Due to the different moving strategies in different methods, we set the task ending condition for each positioning process. The AMCL method corresponded to condition when the particles were converged to a single cluster. The ORB-SLAM and text



(a)

(b)

(c)

Figure 8: Experimental environment and platform. (a) Floor plan of the second floor of a hotel; (b) corridor image of the second floor; (c) the robot platform.

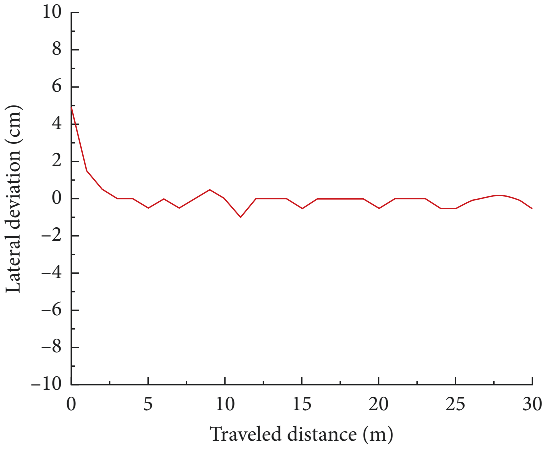


Figure 9: Lateral deviation.

identification methods allowed the robot to rotate one circle in place. In our proposed method, the robot was allowed to adjust its position near the middle line of the corridor, and it needed to move 0.24 meters and rotate 90 degrees on average which were the acceptable range and the requirement of our mobile strategy. The results are shown in Table 3. It is obvious that our approach achieves a higher localization success rate when compared with other methods.

Table 1: Comparison of text detection results.

Method	Average detected boxes per image
Morphological method	7.4
MESR	9.2
MSER + NMS	6.9
Ours	1.1

Table 2: Recognition results.

Method	Total number of recognitions	Number of correct recognitions	Accuracy (%)
Tesseract	150	83	55.3
Ours	150	148	98.7

Table 3: Localization results from different methods.

Method	Test positions	Success times	Correct rate (%)
AMCL	20	9	45.0
ORB-SLAM	20	14	70.0
Text	20	4	20.0
Ours	20	19	95.0

5. Conclusions

This paper presented a novel mapping and localization approach for an autonomous mobile robot navigating in a long corridor environment. Laser data was used to keep a fixed moving route and extract features for coarse place judgment. Visual SLAM was used to get visual localization and text information for a semantic level purpose.

Although the proposed approach can solve most of the corridor environment localization and semantic interaction with users, the situation in which the mobile robot is inside a specific room was not considered. Therefore, in our future research work, we will pay attention to the topological metric map which can cover all indoor environments. In addition, the 5G communication and cloud computing technology can be used to achieve multiple semantic information in the field of robotics; thus, the mobile robot does not need to configure GPU and other large computing platforms.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Research Project of Guizhou Province Education Department (Grant no. KY[2017]023, Guizhou Mountain Intelligent Agricultural Engineering Research Center) and Doctoral Fund Research Project of Zunyi Normal University (Grant no. ZS BS[2016]01, Aerial Photography Test and Application of Karst Mountain Topography).

References

[1] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[2] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1271–1278, IEEE, Stockholm, Sweden, May 2016.

[3] R. P. Guan, B. Ristic, L. Wang, and J. L. Palmer, "KLD sampling with Gmapping proposal for Monte Carlo localization of mobile robots," *Information Fusion*, vol. 49, pp. 79–88, 2019.

[4] G. Ge, Y. Zhang, W. Wang, and Q. L. Y. Jiang, "Text-MCL: autonomous mobile robot localization in similar environment using text-level semantic information," *Machines*, vol. 10, no. 3, p. 169, 2022.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *proceedings of the IEEE and ACM international symposium on mixed and augmented reality*, pp. 225–234, IEEE, Nara, Japan, November 2007.

[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[7] R. Mur-Artal and J. D. Tardos, "Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[8] C. Campos, R. Elvira, J. J. G. Rodriguez, J M. Montiel, and J D. Tardos, "ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[9] J. Engel, T. Schöps, D. Cremers, and Lsd-Slam, "LSD-SLAM: large-scale direct monocular SLAM," *Computer Vision - ECCV 2014*, Springer, in *proceedings of European conference on computer vision*, pp. 834–849, September 2014.

[10] C. Forster, Z. Zhang, M. Gassner, and M. D. Werlberger, "SVO: semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[11] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[12] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020.

[13] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2963–2970, IEEE, San Francisco, USA, June 2010.

[14] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3538–3545, IEEE, Rhode Island, USA, June 2012.

[15] J.-C. Wu, J.-W. Hsieh, and Y.-S. Chen, "Morphology-based text line extraction," *Machine Vision and Applications*, vol. 19, no. 3, pp. 195–207, 2008.

[16] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proceedings of the European Conference on Computer Vision*, pp. 56–72, Computer Vision - ECCV 2016, Amsterdam, Netherlands, October 2016.

[17] X. Zhou, C. Yao, H. Wen et al., "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5551–5560, Honolulu, Hawaii, USA, July 2017.

[18] Q. Yang, M. Cheng, W. Zhou et al., "Inceptext: a new inception-text module with deformable psroi pooling for multi-oriented scene text detection," 2018.

[19] J. Yankey and O. Ernest, "An automatic number plate recognition system using opencv and tesseract ocr engine," *International Journal of Computer Application*, vol. 180, no. 43, pp. 1–5, 2018.

[20] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[21] N. T. Ly, C. T. Nguyen, and M. Nakagawa, "An attention-based row-column encoder-decoder model for text recognition in Japanese historical documents," *Pattern Recognition Letters*, vol. 136, pp. 134–141, 2020.

[22] S. Canaz Sevgen and F. Karsli, "An improved RANSAC algorithm for extracting roof planes from airborne lidar data," *Photogrammetric Record*, vol. 35, no. 169, pp. 40–57, 2020.

[23] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4507–4515, IEEE, Honolulu, Hawaii, July 2017.

[24] T. Hegghammer, "OCR with tesseract, amazon textract, and Google document AI: a benchmarking experiment," *Journal of Computational Social Science*, vol. 4, pp. 1–22, 2021.

[25] Y. Du, R. Yang, Z. Chen, and L. X. X. Wang, "A deep learning network-assisted bladder tumour recognition under cystoscopy based on Caffe deep learning framework and EasyDL platform," *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 17, no. 1, pp. 1–8, 2021.

[26] J. Niu and K. Qian, "Robust place recognition based on salient landmarks screening and convolutional neural network features," *International Journal of Advanced Robotic Systems*, vol. 17, no. 6, Article ID 172988142096696, 2020.

[27] L. Bampis and A. Gasteratos, "Sequence-based visual place recognition: a scale-space approach for boundary detection," *Autonomous Robots*, vol. 45, no. 5, pp. 1–14, 2021.