Journal of Imaging Science and Technology® 66(4): 040412-1–040412-14, 2022.

Medical Mobile Robot Localization in Hospital Corridor Environment Using Laser SLAM and Text Features

Gengyu Ge
School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
School of Information Engineering, Zunyi Normal University, Zunyi 563006, China

Yi Zhang
School of Advanced Manufacturing Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
E-mail: zhangyi@cqupt.edu.cn

Wei Wang
School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Qian Wang
School of Information Engineering, Zunyi Normal University, Zunyi 563006, China

Abstract. *Localization is an important capability for a medical service robot navigating in the hospital environment. Recently, by using laser-based simultaneous localization and mapping (SLAM) and Monte Carlo localization (MCL) techniques, the mobile robot can localize itself in those places with obvious geometric features, like corners, lines, breakpoints, and curves. However, when a robot moves in a long corridor environment inside a building, there are only a few geometric features for reference, especially in symmetrical or similar areas. Consequently, the mobile robot can neither localize itself correctly in the global localization phase without a known initial pose nor recover the correct pose from the kidnapped robot problem. To solve this problem, we propose an approach that uses text features on the doorplate to assist localization. In the mapping stage, we control the mobile robot moves along a fixed path using both text features and laser scans data. In the localization stage, the mobile robot moving according to a strategy to find text features and initializes the particles, then navigates and localizes using an improved MCL method. The experimental results show that our approach achieves over 90% mapping accuracy and 93.3% successful localization rate, which outperforms traditional methods.*

[DOI: 10.2352/J.ImagingSci.Technol.2022.66.4.040412]

1. INTRODUCTION

Nowadays, many commercial service robots are used for transporting goods in hospitals, especially during the Coronavirus disease 2019 (COVID-19) epidemic time. Localization is one of the most fundamental capabilities for an autonomous mobile service robot that navigates in the man-made environment. When a mobile robot moves from one place to the target position, it needs to know its position and orientation at every moment because of the need for path planning and obstacle avoidance. In most outdoor environments, the problem can be easily solved by using the global positioning system (GPS) method [1], which has been used in many commercial electronic products, like mobile phones and car navigators. In an indoor environment, a variety of techniques are used to solve mobile robot localization problems. Radio frequency identification (RFID) [2], QR code [3], WiFi [4], and ultra-wide band (UWB) [5] can be effectively used in the robot localization field, but these methods need external artificial beacons and should be placed in a fixed position in advance. In addition, the mobile robot needs to carry communication devices or readers with those external tags. When a mobile robot moves to an unknown environment without artificial beacons, the mainstream approach is using SLAM techniques to build a map for later positioning and navigation tasks [6, 7]. The commonly used SLAM methods are laser-based SLAM [8] and visual SLAM [9], which are very effective in their respective application scenarios.

Like other areas in the indoor environment, long corridors are very common in the man-made world, where room layouts are highly structured and repetitive. There are usually symmetrical rooms on both sides of the corridor in the hospital inpatient department. Unfortunately, indoor corridors usually have few geometry or visual features to extract and track. Laser-based SLAM often has accumulated errors, which will lead to inaccurate mapping and localization. As shown in Figure 1(a), when a mobile robot moves from positions A and B in the corridor without a known pose, the data results of laser scanning are the same. Therefore, the mobile robot cannot get a correct global localization in the initialization phase, even after moving in a small range. Similarly, when a mobile robot is kidnapped from position A to position B, it still cannot find the scanning data difference. If the robot moves to a unique room and

Received Jan. 15, 2022; accepted for publication May 5, 2022; published online June 13, 2022. Associate Editor: Bo Wang.
1062-3701/2022/66(4)/040412/14/$25.00

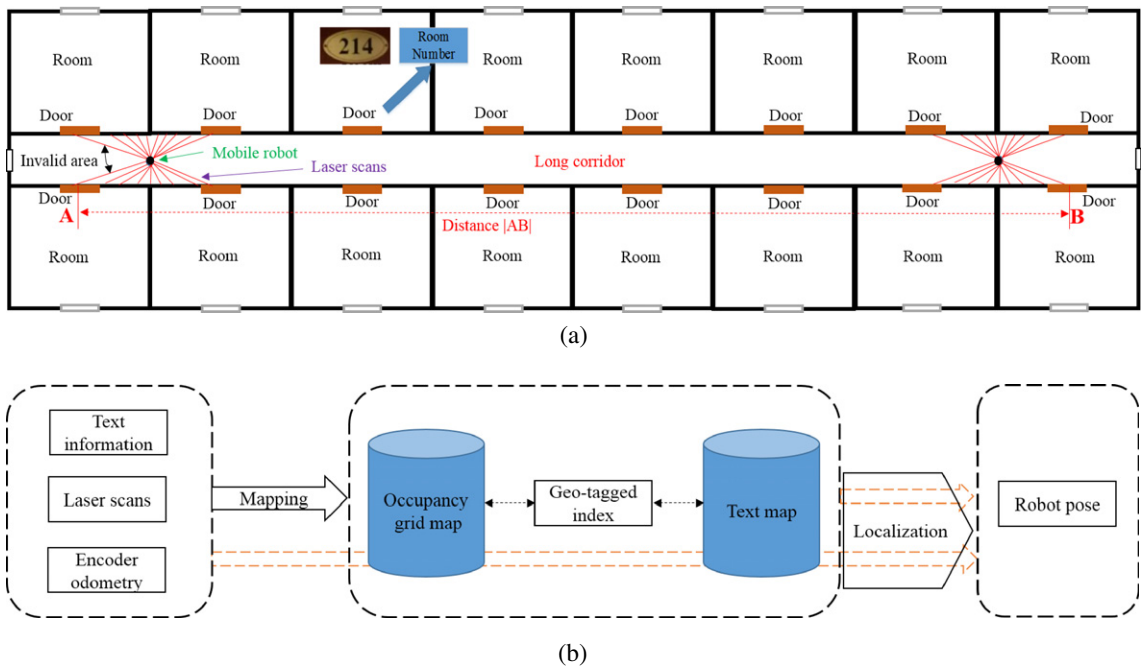


Figure 1. (a) Text information on the doorplate in an indoor long corridor scene; (b) Framework of the mapping and localization process.

comes out again, then it will locate successfully. Because of light changes, texture sparsity, and other factors, visual SLAM will lose track of the last frame or previous frame captured from the same place. Bayramoglu et al. used line segmentation to implement visual odometry, which was used to help robot localization [10]. However, lines are relatively stable, but can not provide more information for positioning, for instance, without text-level semantic information which can be used for a further intelligent purpose. The most important issue of pure visual localization is that the sparse feature points are not suitable for flexible path planning.

Fortunately, indoor corridors rarely exist independently, but with rooms and doors on one side or both sides. Typically, rooms have doorplates or nameplates marked with room numbers and label annotations, as Fig. 1(a) shows. Consequently, there are many text features available for indoor corridor environments which can assist localization. Radwan et al. [11] used the textual information from the shops on the street for global localization in public maps based on a compass and RGB-D camera, which got higher positioning accuracy relative to GPS in urban environments. Li et al. [12] integrated planner text features into the visual SLAM pipeline and got a more robust and accurate map. A work similar to ours is [13], which used object and character recognition features to help the mobile robot navigate in an indoor environment, especially in a corridor. The robot needs to find a door first, then find the room number plate, and recognized the text content on the plate at last. However, their work needs to build the models of door and text-plate as prior knowledge which is complex and unnecessary. Besides, they used four ultrasonic sensors that are not suitable to construct an occupancy grid map and cannot plan paths or navigate inside a room.

In this paper, we propose a novel approach that combines laser-based SLAM with text features to assist mobile robot localization in a long corridor environment. The main framework of the mapping and localization process is shown in Fig. 1(b). The main contributions of this article are:

(1) This paper proposed to use text features to assist robot mapping and localization. The text features were mainly extracted from the doorplates and the experimental results demonstrated that our method was effective.
(2) A monocular camera used to capture images was mounted on the mobile robot and oriented sideways. We proposed a mobile robot moving strategy that keeps a fixed distance from the wall. This was beneficial for capturing high-quality images which were used to detect text information. As a result, text detection and recognition accuracy improved.
(3) The paper proposed extracting fixed areas of the image to reduce computing costs and get as little but accurate text information as possible.
(4) Compared with conventional laser SLAM and MCL localization methods, text features used for mapping with laser-based SLAM could improve the accuracy of the map and the success rate of localization. Mapping accuracy can achieve over 90% accuracy while traditional methods have lower accuracy, and the success rate can reach 93.3% accuracy while traditional methods always fail.

The rest of the paper is organized as follows. Section 2 gives an overview of related work. Section 3 presents the main methods which are the mathematical model of mapping and

localization, text detection and recognition, map-building with text information, and localization with text information. In Section 4, we provide experiments and results in both simulated and real-world environments. Section 5 is the discussion and conclusion of this paper.

2. RELATED WORK

2.1 *Laser-based Mapping and Localization*

When a mobile robot needs to perform tasks in a new environment, for instance, delivering drugs and goods to a patient room in the hospital during the COVID-19, the first task is to build a map of the new environment. For autonomous mobile robots, the occupancy grid map which is a metric representation of the environment, is suitable for path planning and accurate moving. To this end, the laser-based SLAM is used to accomplish the mapping task. A laser-based SLAM usually uses a laser range finder or laser LIDAR as the measurement sensor. The sensor collects the distance of each angle, from the sensor transmitting end to the obstacle end. At present, there are two mainstream laser SLAM methods used for mapping an unknown environment. The first one is a two-step iterative process filter-based method which is derived from Bayesian filtering. One of the most popular filter SLAM approaches is the Extended Kalman Filter (EKF) SLAM and is firstly used in sonar sensors [14], then extended to laser and vision sensors. Another major branch of filter methods is particle filter, which can accommodate any form of probability distribution and does not need a Gaussian noise assumption. A representative work is gmapping [15], which has been used in the third-party software packages of the ROS (a robotics community) and many commercial service mobile robots. Contrary to filter-based SLAM approaches, graph-based SLAM approaches process a collection of nodes representing landmarks and poses. Edges of a graph are those constraints generated from sensor observations and optimization methods are used to minimize the error of the constraints. One of the most popular projects is Cartographer [16], which is implemented by Google's engineers. In an indoor environment where features are salient, both the two different SLAM methods can get high precision grip map, which is suitable for navigation and path planning. However, when the environment is a long corridor, the mapping result has poor quality. The reason is due to the accumulated error of odometer and the limited measurement range of laser sensor. Usually, the transverse error is almost negligible, however, the longitudinal error is large.

Localization means the mobile robot gets the position and orientation in a known environment with a previously built map. It refers to three modes which are global localization, local localization and robot kidnapped problem. Global localization means the robot achieves its pose without a known setup. Local localization usually assumes that the initial pose of the robot is known, and then the local localization problem becomes a pose tracking issue. A kidnapped robot problem means a moving mobile robot is kidnapped to another location in a very short time, and it belongs to an upgraded version of global localization. The most widely used localization method is particle filter-based Monte-Carlo localization (MCL), which is proposed by Thrun et al. [17]. Even though the sensor data and control commands (odometry data) are noisy, MCL can estimate the position and orientation of a mobile robot using particle filters, and realize a non-parametric implementation of the recursive Bayes filter. Because the particle filter-based localization method can handle multi-model beliefs, the MCL has been used with many types of sensors, like laser range finders, sonars, and many cameras [18, 19]. Unfortunately, for a long corridor environment, if the initial pose is not given manually, it is difficult for the robot to achieve global localization. Moreover, after the robot kidnapping problem, it faces difficulties to recover the robot pose.

2.2 *Text Detection and Recognition*

Man-made world has rich text information such as billboards, the building or room number, and labels on objects. Due to various fonts, multi-oriented directions, multilingual, uneven illumination, and perspective transformation, natural scene text detection and recognition is more difficult than traditional optical character recognition (OCR) techniques which focus on printing and scanning document level recognition task. To extract and understand the scene text information, especially the semantic level content, the OCR usually has two stages which are text detection and text recognition [20], and its essence is feature detection and recognition [21].

As a prerequisite of the subsequent textual information processing, text detection mainly distinguishes text regions from the backgrounds of an image according to the features designing. In the case of obvious contrast between text and the background, traditional approaches depending on manually designed features can achieve good results. Neumann et al. [22] proposed the maximally stable extremal region (MSER) method, which has a real-time detection result and is robust to blur, illumination, and color variation. Epshtein et al. [23] put forward the stroke width transform (SWT) method, which used a local image operator to compute the width of approximately textual pixels, then found letter candidates, and lastly grouped letters into text lines. Busta et al. [24] presented the FASText approach, which used FAST keypoint detector to extract the strokes of texts. For those multi-directional and multi-font text scene images, the detection tasks have difficulties. Then, text detection methods based on deep learning neural networks become popular and effective. Tian et al. proposed a connectionist text proposal network called CTPN [25], which localized text lines in scene text images accurately. The CTPN used CNN to detect a text line in a sequence of fine-scale text proposals, while the sequential proposals were connected naturally using RNN. EAST was a powerful pipeline which predicted words or text lines of arbitrary orientation in full images with a single neural network [26]. IncepText was an end-to-end natural scene detector based on instance

segmentation [27]. The authors combined PSROI pooling with the inception-text module, which was effective for multi-oriented scene text detection.

Before the popularization of deep learning, traditional text recognition usually uses manually designed features to abstract the image at a semantic level. There are many open-sourced OCR engines and software that can be used, like Google Docs OCR, Tesseract, ABBYY FineReader, and Transym [28, 29]. As a traditional text recognition method, Tesseract has many steps to solve the problem. For instance, at the line and word-finding stage, there are steps such as line finding, baseline fitting, fixed-pitch detection and chopping, and proportional word finding. While at the word recognition stage, there are steps such as chopping joined characters, associating broken characters, training data, and adaptive classification. Nowadays, because of the powerful computing capability of graphics processing unit (GPU), research works are focused on deep learning approaches. Two main-stream methods are all based on CNN and RNN frameworks. One of them uses CNN to extract features, then combines RNN with connectionist temporal classification (CTC) to predict the sequence. The representative work is CRNN, which is proposed by Shi et al. [30]. The network of CRNN has three components, that are convolution layers, recurrent layers, and followed by a transcript layer. The other one uses CNN, the Seq2Seq model, and the Attention framework [31], which gets inspiration from machine translation and includes an encoder and decoder. There are many important text datasets used for text detection and recognition, such as ICDAR, COCO-Text, IIIT5K-Word, SVT, Synth800K, and MSRA-TD500 [32, 33].

3. METHODS

3.1 *Mathematical Model of Laser-based Mapping and Localization*

When a mobile robot maps an environment with known poses, the occupancy grid mapping algorithm is used to compute the posterior probability of the whole map [34]. The formula is defined as in Eqs. (1) and (2):

$$p\left(\boldsymbol{m}|z_{1:t}, x_{1:t}\right) = \prod_i p\left(m_i|x_{1:t}, z_{1:t}\right) \tag{1}$$

$$\boldsymbol{m} = \{m_i | i = 1, 2, 3, \ldots, n\}, \tag{2}$$

where m is the whole map while m_i is the ith grid cell. $x_{1:t} = x_1, x_2 \ldots, x_t$ represents the robot trajectory from time 1 to time t. $z_{1:t} = z_1, z_2, \ldots, z_t$ is the external laser sensor observations. The problem can be easily solved by using the binary Bayes filter method. The occupancy grid map construction algorithm uses the logarithmic occupancy probability expression as shown in Eq. (3):

$$l_{t,i} = \log \frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})}. \tag{3}$$

The advantage of logarithmic probability expression is that it can avoid numerical instability near 0 and 1. The posterior probability is calculated as Eq. (4):

$$p\left(m_i|z_{1:t}, x_{1:t}\right) = 1 - \frac{1}{1 + \exp\{l_{t,i}\}}. \tag{4}$$

However, in most cases, mobile robots don't know their position and orientation in a strange environment. In those unknown environments, the traditional mapping approach based on a particle filter and a laser range finder is called laser SLAM, which is to estimate the joint posterior $p(x_{1:t}, \boldsymbol{m}|z_{1:t}, u_{1:t-1})$ about the probability grid map $\boldsymbol{m}$ and the robot trajectory $x_{1:t} = x_1, x_2 \ldots, x_t$. This estimation needs the external laser sensor observations $z_{1:t} = z_1, z_2, \ldots, z_t$ and the internal odometry measurements $u_{1:t} = u_1, u_2, \ldots, u_{1:t}$. Fortunately, the estimation is equivalent to the following factorization as in Eq. (5):

$$\begin{aligned} &p(x_{1:t}, \boldsymbol{m}|z_{1:t}, u_{1:t}) \\ &\quad = p(x_{1:t}|z_{1:t}, u_{1:t}) \cdot p(\boldsymbol{m}|x_{1:t}, z_{1:t}, u_{1:t}) \qquad (5) \\ &\quad = p(\boldsymbol{m}|x_{1:t}, z_{1:t}) \cdot p(x_{1:t}|z_{1:t}, u_{1:t}), \qquad (6) \end{aligned}$$

where the map can be computed when given the trajectory, which should be estimated firstly before further operation. When given the poses and the corresponding observations of the robot, the map is assumed independent of the odometry data. This is why Eq. (6) can be obtained from Eq. (5). With an estimated trajectory of the mobile robot, the problem becomes a map construction with a known pose. Then, the posterior map can be computed by using Eqs. (1) and (2). A graphical model of the SLAM problem is shown in Figure 2. The data of the blue nodes are known, while the values in the red dotted box are to be determined.

Here, the global localization problem is mainly considered when given a known environment map and observations at every moment t. The robot pose $x_t = (xy\theta)^T$ can be computed using the formula $p(x_{1:t}|z_{1:t}, m)$. A graphical model of localization is shown in Figure 3. The poses in the red dotted box are unknown and need to be determined.

$$\begin{aligned} p(x_t|\boldsymbol{m}, z_{1:t}, u_{1:t}) \propto p(z_t|m, x_t) \cdot \int p(x_t|x_{t-1}, u_t) \\ \cdot p(x_{t-1}|m_{1:t-1}, z, u_{1:t-1}) dx_{t-1}. \end{aligned} \tag{7}$$

Given a built grid map, odometry data, and laser scans, the basic Monte Carlo localization algorithm is shown in Algorithm 1.

In Algorithm 1, the particles set $\chi_t = \left\{x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[N]}\right\}$ represent the belief confidence $bel(x_t)$ at current time t. x_t^j can be estimated from the last moment pose x_{t-1}^j and the current moment sample odometry motion model data u_t. The weight w_t^j of each particle represents the probability of each estimated robot pose. Most localization problems can be solved by using the MCL algorithm and its extended versions, for instance, Augmented MCL and KLD sampling MCL. The technology of these combinations is called AMCL, which has been integrated into ROS software packages [35].

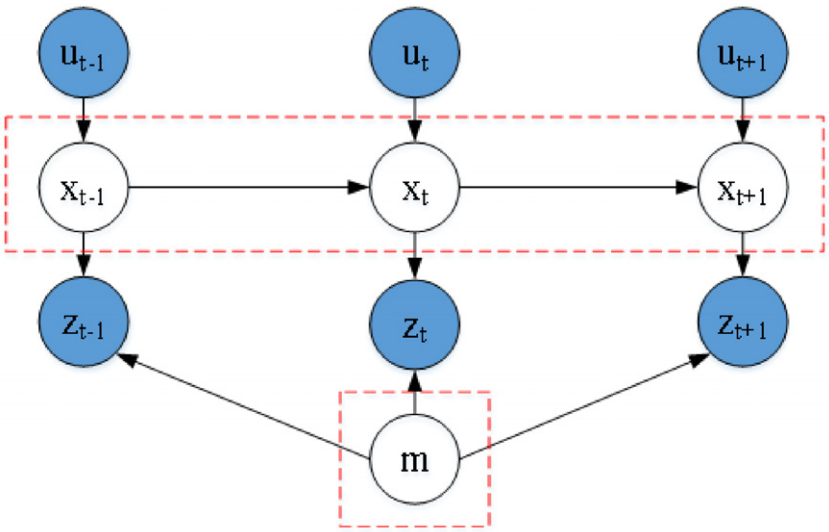


Figure 2. Graphical model of SLAM.

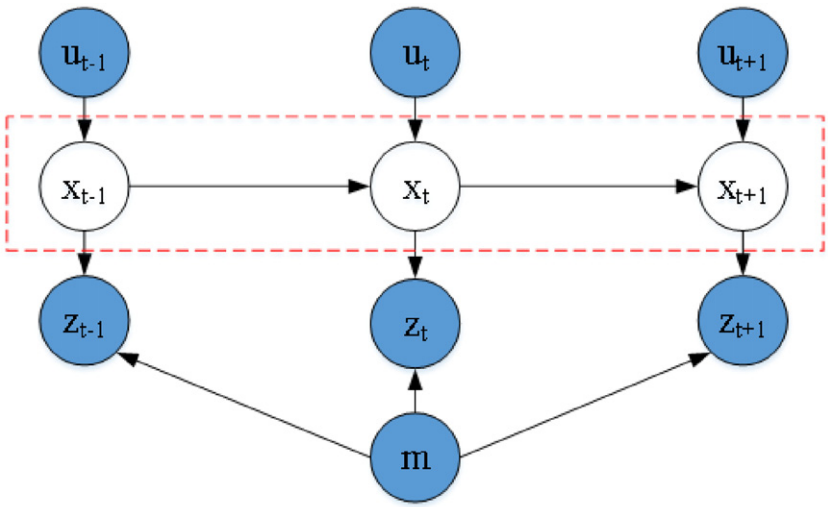


Figure 3. Graphical model of localization.

3.2 *Text Detection and Recognition Strategy*

Different from research that only focuses on scene text detection from images captured from natural scenes without purposive selection. The mobile robot in an indoor environment needs not to consider many complex cases. In a long corridor environment inside a building, text features are used for localization mainly on the doorplates or next to the door frames. In addition, most texts are room numbers and have a simple or single background color. An interesting phenomenon is that text arrangements used for the function of signs usually are horizontal, though some are vertical.

Due to the limitation of the computing device mounted on the mobile robot and a low power consumption need, we use a traditional text detection and recognition method. MSER is a text detection method which groups pixels into regions using connected component analysis [22]. The method assumes that pixels from the same character have similar properties, then finds individual characters. The improved version combines intensity, hue, saturation channels, and intensity gradient, can detect 94.8% of characters by using ER detector. In addition, MSER has been adopted by OpenCV (an open-source software library for image processing). As a text recognition solution, Tesseract is an OCR engine, which is released for open source and is now available at https://digi.bib.uni-mannheim.de/tesseract/. Figure 4 shows the text detection and recognition results of the methods used in this paper. Fig. 4(a) is the source image captured from a student dormitory building, which is a door with a doorplate and room number on it. The background color is dark red, which is relatively simple. As shown in the figure, the doorplate is a yellow metal plate with black numbers printed on it. Bounding boxes in Fig. 4(b) show that there are two error detection regions, even if the true text region is successfully detected. Fig. 4(c) shows detection results using MSER method after image binarization, which has more error detection regions.

Algorithm 1. Basic MCL Algorithm

Input: χ_{t-1}, u_t, z_t, m

Output: χ_t

1: Initialization, $\overline{\chi}_t = \chi_t = \varnothing$

2: For $j = 1$ to N do

3: $x_t^{[j]} = \text{sample\_motion\_model}\ (u_t, x_{t-1}^{[j]})$

4: $w_t^{[j]} = \text{measurement\_model}\ (z_t, x_t^{[j]}, m)$

5: $\overline{\chi}_t = \overline{\chi}_t + \left\langle x_t^{[j]}, w_t^{[j]} \right\rangle$

6: End for

7: For $i = 1$ to N do

8: Draw i with probability $\propto w_t^{[i]}$

9: Add $x_t^{[i]}$ to χ_t

10: End for

11: Return χ_t

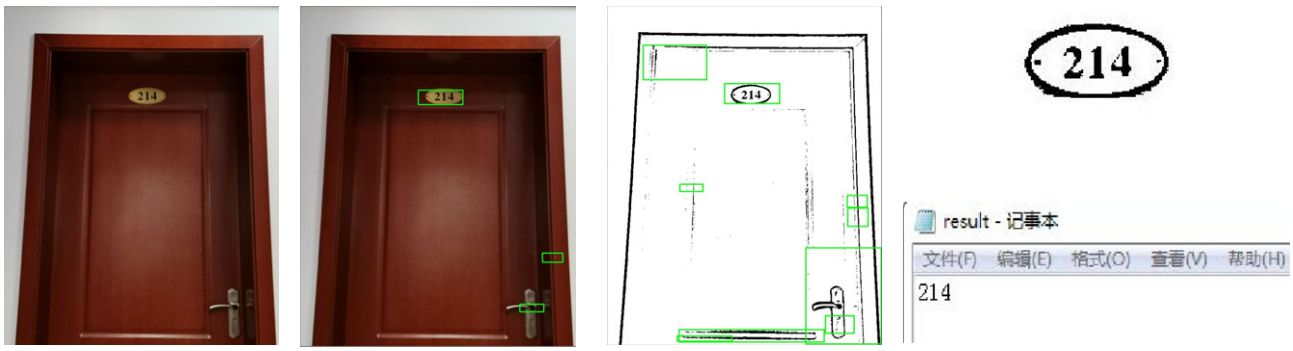


Figure 4. Results of text detection and recognition. (a) Source image, a door with a room number on the top half; (b) Text detection using MSER method; (c) Text detection using MSER method after image binarization; (d) Text recognition result after detection and binarization.

However, the recognition result has low accuracy if uses source image to recognize directly. Therefore, we detect the text regions first, then convert the image to a binary one, and the corresponding region of binarization is used for text recognition. The recognition result of Fig. 4(c) is shown in Fig. 4(d), which has the correct text content.

However, the above tests are implemented in static images, which are so simple and not the same as the images captured by a mobile robot in the real environment. Actually, most areas in each image do not contain text information like the room number and are not useful for robot localization. For instance, the doorplate with text content is usually on the upper part of the door. If the mobile robot moves along a fixed route with a camera faced to one side, theoretically, the height of the doorplate on the image of each different door is constant. Consequently, there is no need to process these areas which consumes a lot of computing resources, memory space, and time.

To improve the efficiency of text detection and recognition for the mobile robot, following are our steps. A new image is defined as I_i, where $i = \{1, 2, \ldots, n_I\}$, and n_I represents the number of all images that the camera captured. The number of doors on both sides of the corridor is defined as N_D, correspondingly, the number of doorplates and text contents is N_T. Theoretically, N_T is equal to N_D. B_{ij} means the jth rectangular text box in ith image, where $j = \{1, 2, \ldots, n_T\}$, and n_T represents the number of all text boxes in image I_i. For every rectangular text box, the coordinates of 4 vertices are defined as, (x_{ij0}, y_{ij0}) represents the upper left vertex, (x_{ij1}, y_{ij1}) is the upper right vertex, (x_{ij2}, y_{ij2}) is the lower left vertex, and (x_{ij3}, y_{ij3}) represents the lower right vertex. The center point coordinate of the rectangle is defined as B_{ijc}, which is as following equation.

$$B_{ijc} = (x_{ijc}, y_{ijc}). \tag{8}$$

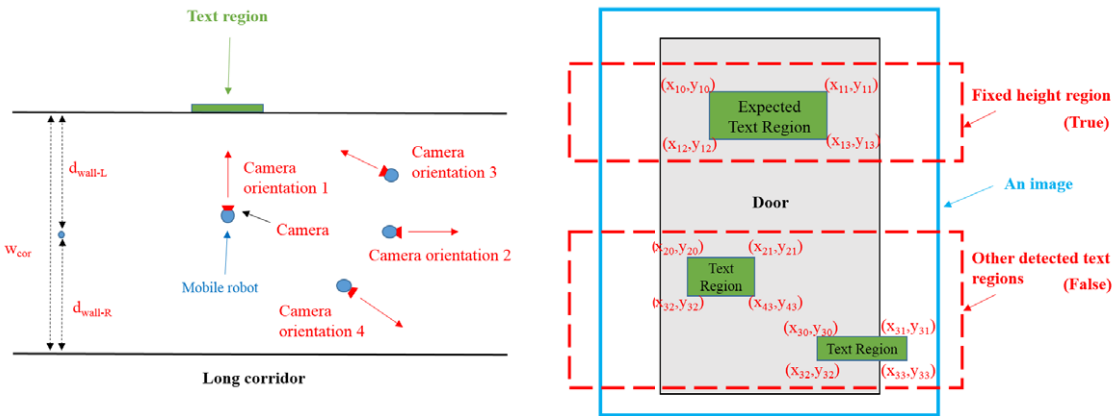


Figure 5. Text detection and recognition. (left) Several camera perspectives; (right) Expected text region and other detected text regions.

Furthermore, Eq. (8) can be represented as Eq. (9):

$$(x_{ijc}, y_{ijc}) = \left(\frac{1}{2}(x_{ij0} + x_{ij1}), \frac{1}{2}(y_{ij0}, y_{ij2})\right). \quad (9)$$

Besides, the width and height of the jth rectangle text box can be represented as Eq. (10) and Eq. (11):

$$w_{ij} = x_{ij1} - x_{ij0} \quad (10)$$

$$h_{ij} = y_{ij2} - y_{ij0}. \quad (11)$$

The corresponding coordinate symbols are shown in the right part of Figure 5.

The width of the corridor is usually a constant value and is defined as w_{cor}, shown in the left part of Fig. 5. Two walls on both sides of the corridor are defined as wall-L and wall-R, where L means left and R means right. In addition, the size of the image collected every time by the mobile robot is also constant. We control the mobile robot to move along one side of the corridor, keeping a fixed distance $d_{\text{wall}-R}$ from the wall. Then the distance between the mobile robot and the other side of wall is $d_{\text{wall}-L}$. It is easy to know that the width of the corridor is the sum of these two distances:

$$w_{\text{cor}} = d_{\text{wall}-L} + d_{\text{wall}-R}. \quad (12)$$

When the mobile robot needs to search text features on the left side of the corridor, the ideal case is $d_{\text{wall}-L} > d_{\text{wall}-R}$ and keeping the camera oriented to the left side. The reason is to collect high-quality images as much as possible and vice versa on the right side. In the left part of Fig. 5, there are four camera orientations from different perspectives. Camera orientation 1 is the most suitable case and can get a high-quality image that includes the expected text region. Other camera orientations are difficult to achieve the same effect. Therefore, we should make the strategy that the mobile robot moves along the corridor and keeps a fixed distance from one sidewall.

As the doorplates are installed deliberately, the heights of all plates are the same. As a result, the center point of the doorplate in each image has the same vertical coordinate value. In the right part of Fig. 5, we will get only one expected and true text region from the image captured by the mobile robot. However, there are usually other extracted text regions in the image, these text areas may be originally existed or were incorrectly detected. If we keep the mobile robot moving in a fixed line or path in the corridor, theoretically, the desired text region should be detected at the same height and region in each image. With this prior knowledge, we do not need to deal with the information of other regions in the images. Consequently, we get an improving processing speed and ensure the real-time performance of computing.

3.3 *Map-building with Text Information*

In the mapping stage, the mobile robot moves forward along the corridor, using laser-based SLAM method and text features assisted information. The odometry records the moving distance of the mobile robot and the distance between adjacent room number text regions. We define the distance between adjacent text regions as $Dsit_{(i,i+1)}$, which can be thought as the distance between two center points of the adjacent doorplates. The distance can also be considered as the difference between the two positions of the odometry readings, which can be expressed in the following equation:

$$Dist_{(i,i+1)} = u_{i+1} - u_i. \quad (13)$$

The straight line distance from the first door to the last one on the left side of a corridor is the sum of the distances between all the adjacent room number text regions. Similarly, the right side is the same. The accumulated odometer data has such a relationship:

$$\sum_{i=1}^{n-1} Dist_{(i,i+1)} \geq u_n - u_1. \quad (14)$$

The room number texts distribution is shown in Figure 6.

The traditional laser-based SLAM framework uses laser and odometry data to map a occupancy grid map. The raw two-dimension laser scanning distances around the mobile robot are thought of as observations, while the odometry data from the encoder sensor are thought of as the control quantity. Our method in this paper adds the text features to the mapping system as shown in Figure 7. We record the

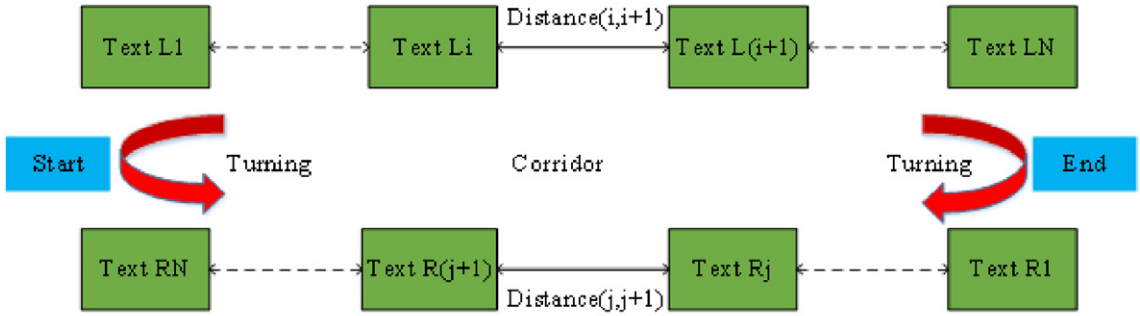


Figure 6. Texts distribution on both sides of a long corridor.

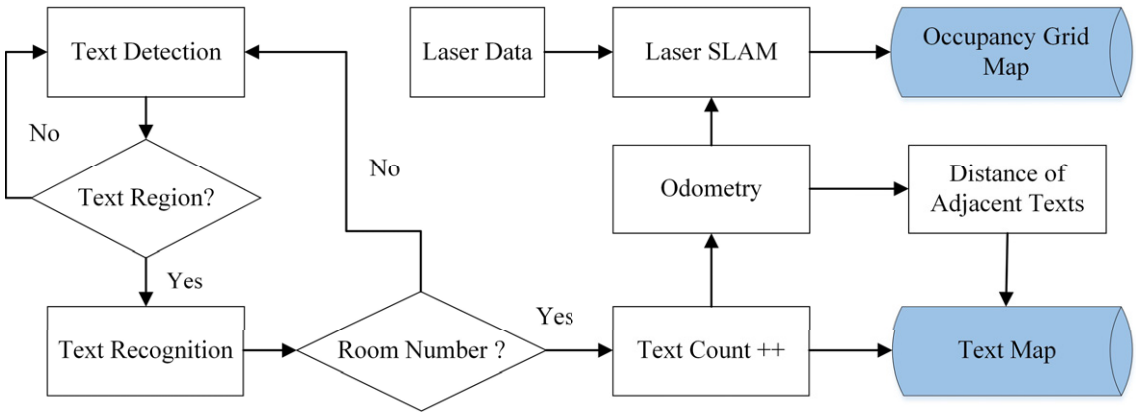


Figure 7. Map-building progress.

adjacent distance between two adjacent room numbers or texts.

Two processes run in the controller system simultaneously. One of the them performs the traditional SLAM algorithm to build a map in the process of the mobile robot moving forward. The mobile robot is controlled to keep $d_{\text{wall}-R}$ distance from the wall on the right side and capture images from the left side. The other process mainly detects the room number text region from the image area where the doorplate is mostly to appear, then recognizes the text region and gets the text content. If the room number is identified, waiting for the bounding box appears in the middle line of the image, then we record it, and save the text content and distance between the adjacent room numbers to the text map. The adjacent distance between the two adjacent doorplates is used as a reference in the localization phase.

The mobile robot needs to turn around and return near the left side of the wall, then repeat the previous working procedure to process the text regions on the right side. The finally constructed maps are an occupancy grid map based on 2D laser SLAM and a text map with the text content of the room numbers. In addition, the text map also includes the adjacent distances data between each pair of adjacent room numbers. The adjacent distance refers to the odometer data.

3.4 *Localization with Text Information*

When the mobile robot needs to perform tasks and navigate in the long corridor environment to find a target room, it should locate itself first, especially the global localization in the initialization phase. Traditional occupancy grid maps cannot provide semantic information, only laser scanning data can be used. In case of a kidnapped robot problem or restart from a computer system crash, the mobile robot can not localize itself quickly because of the similar surroundings. Besides, current service mobile robots are all artificially set target coordinate position, it is easy to locate incorrectly if the odometry has errors. Based on traditional MCL serious algorithms, our method adds text features as semantic landmarks to assist localization and navigation. Given a target room number, the localization progress is shown in Figure 8.

If the mobile robot has a known initial pose, the localization is a pose tracking problem, and then the AMCL localization method can determine the later poses. In Fig. 8, when a user inputs a room number into the robot system, the robot computes the target room position in the grid map and moves forward to the place. To reduce the complexity of navigation and localization, the robot moves along one side of the corridor and keeps a constant distance from the wall according to the strategy described in Section 3.2, accompanied by a text extraction strategy. The camera captures images from the other side to detect doorplate region and recognize the text information. According to the recognized room number information and the distance of the two adjacent doorplates, the mobile robot will correct its pose.

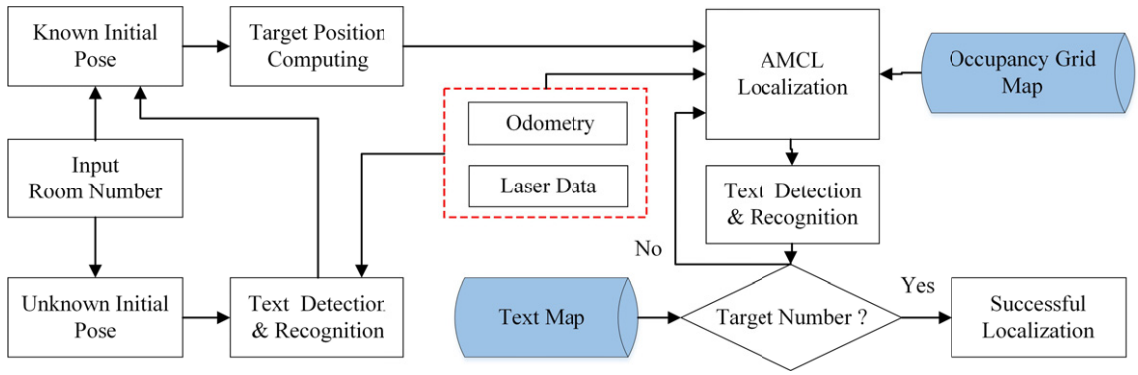


Figure 8. Localization progress.

When the robot does not have a known initial pose, or there has a robot kidnapped case, the localization program will fail. For this instance, the execution strategy we set up for the mobile robot has three steps. First, the mobile robot adjusts its moving direction and moves forward along the corridor according to the laser scanning data shown in Fig. 1. Then, the robot detects doorplates in the images captured from the other side of the wall and recognizes the text content from the text region. At last, the robot matches the recognized room number with a prebuilt text map and computes the most likely pose. As described in Ref. [13], the mobile robot recognizes a room number, and the position and pose can be computed based on the coordinate of the door and the doorplate. Given a known and accurate initial pose, the localization problem becomes much easier. If the current position of the mobile robot is near the ith door and the target room place is the jth, then the robot needs to move a distance like equation 15 shows.

$$\begin{cases} \sum_{i}^{j-1} \mathrm{Dist}_{(i,i+1)} & (i < j) \\ \sum_{j}^{i} \mathrm{Dist}_{(i,i+1)} & (j < i) \end{cases} \tag{15}$$

4. EXPERIMENTS AND RESULTS

Due to COVID-19 pandemic, we could not do experiments in real hospitals or hotels outside the university. Instead, we did experiments in the simulation environment and the real environment on campus, respectively. In the simulation environment, we mainly presented the mapping results using several popular SLAM solutions. A corridor without doorways which was thought as the environment without geometrical structure features, and a corridor with doorways as that with geometrical features. The experiments showed that traditional methods were not suitable for mapping in the long corridor environment and a necessary improvement method needed to be adopted. In the real-world environment, we tested our proposed methods in the text detection and recognition, mapping, and localization phases, respectively. It should be noted that the final localization tests were based on the scheme of manual mapping to ensure the accuracy of the metric map.

4.1 *Mapping in a Simulated Environment*

In this experiment, we modeled two distinct long corridors in the Gazebo simulation software using a desktop computer that had installed Ubuntu16.04 distribution operating system of Linux and ROS Kinetic version software. Both the two corridors were 15.5 meters long and 2.0 meters wide and had ten rooms of the same size, five on each side. One of them had closed doors, which were in the same plane with the wall, as shown in Figure 9(a). The other one had opened doors, which was shown in Figure 10(a). A simulated mobile robot named turtlebot3-waffle was placed in the corridor environment. The robot was configured with a laser range finder sensor, an RGB-D camera, and an odometer. We used three different SLAM methods to construct grid maps, and then analyzed the results.

In Fig. 9, the turtlebot3 was placed in the corridor environment with closed doors, which could be seen as a rectangle. Fig. 9(b) was a grid map constructed by using the gmapping SLAM method, Fig. 9(c) was a map got from hector SLAM method [36], and Fig. 9(d) was from the cartographer. These methods were integrated into the ROS system and had been used for many commercial service robots. From the figures above, we found that the lengths of the map differed from each other. The lengths of the map from gmapping and hector methods were shorter than that from the cartographer, while the widths of the three were equal and had high accuracy. There were two reasons for this result. One of them was that the scanning radius of the laser range finder sensor mounted on turtlebot3 was far shorter than the long corridor length. The other was the defects of the two methods. Gmapping depended on the odometry data and the number of particles. Because the symmetrical and featureless corridor had a simple geometry structure, particle weights could not be updated correctly, and the particles could not concentrate. Similarly, the hector SLAM method only depended on laser scans as the observations. In a long corridor environment like a rectangle, the observations along the corridor were always the same. On the contrary, the result got from cartographer was almost perfect.

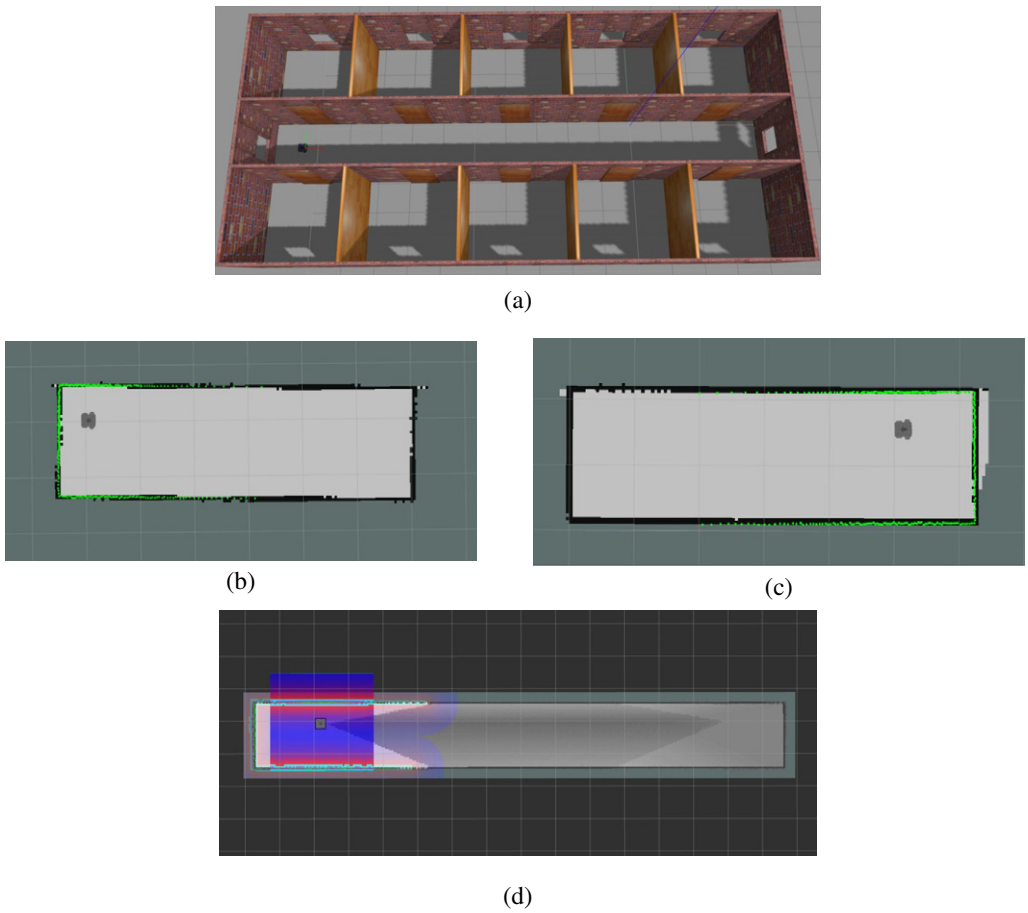


Figure 9. Mapping results from several SLAM methods with the closed doors. (a) Simulated long corridor environment; (b) Map using gmapping SLAM; (c) Map using hector SLAM; (d) Map using cartographer SLAM.

The branch-and-bound approach was used in the SLAM to compute scan-to-submap matches as constraints.

In Figure 10, the turtlebot3 was placed in a corridor with opened doors which could be determined by the features. we used the same three mapping methods to construct grid maps. The turtlebot3 moved along the corridor without entering the rooms. We found that three grid maps looked infinitely close.

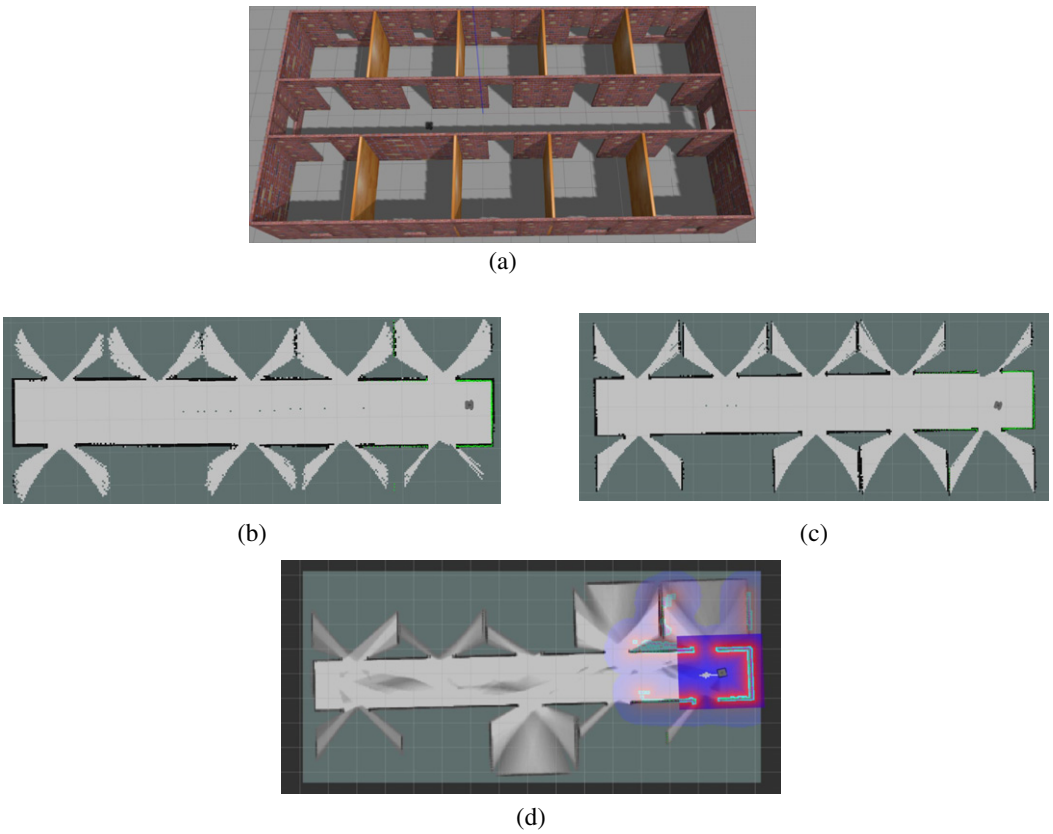
From the above results, we can get a basic fact. When a long corridor environment has enough geometry features, the grip map constructed by using laser-based SLAM, is usually accurate. However, the mobile robot still cannot localize correctly even if there had a highly accurate map. This situation always appears in symmetrical or similar environments [37], because there are many possible positions where the mobile robot may locate. Without the help of other sensor information, the robot cannot distinguish itself at the door of room 214 or room 215 using data from the laser sensor.

4.2 *Real-world Environments in Different Corridors*

In this section, we built a robot platform modified from a mobile chassis, which had two driving wheels and a passive wheel. The mobile robot was configured with a camera, a laser range finder sensor, and a wheel encoder as an odometer. The camera has a resolution of $480 * 640$ and 30 frames per second. The product model of the laser sensor was PRLIDAR A2(SLAMTEC, China), which had a 360-degree scanning range and a maximum 16 meters scanning radius. The hardware of the mobile robot was a Raspberry Pi, which had a quad-core 64-bit processor, 8GB of RAM, and other external devices. The software was Ubuntu16.04 Linux operating system with ROS Kinetic Kame installed. The robot platform was shown in Figure 11(a).

We did four different experiments, respectively. In the first experiment, we placed the mobile robot in a corridor to detect text regions from one side of the corridor. The corridor was in a student dormitory building (Chongqing university of posts and telecommunications, China) which was shown in Fig. 11(b). In the second experiment, we used the OCR engine to recognize the text contents of the text region boxes which were got from the previous step. The third experiment was mapping the corridor with both traditional methods and our proposed methods. In the last experiment, we did a localization experiment based on the third step.

In the text detection experiment, we controlled the mobile robot to move along one side of the wall throughout the whole corridor. Because the width of the corridor was



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

Figure 10. Mapping results from several methods with opened doors. (a) long corridor; (b) Gmapping; (c) Hector SLAM; (d) Cartographer.

(a) (b)

Figure 11. Robot platform and three corridor environments. (a) Robot platform; (b) A corridor in a dormitory building.

a constant value, theoretically, the sum of the minimum scanning value on the left side and the minimum scanning value on the right side of the lidar sensor was the width of the corridor. Consequently, we could easily control the robot to keep a fixed distance from the wall. In addition, we set the camera faced to the other side, perpendicular to the walking direction. We used three methods to detect the text regions from images captured by the mobile robot. Although the most valuable text information for the mobile robot was the room number on the doorplate, the text detection algorithm could not only detect the correct region. Error detection often occurred in most of the images, like Fig. 4(b), the door handle was mistaken as a text. MSER and MSER+NMS methods were used for detecting text regions in the whole area of every

Table I. Correct recognition rate of text content on doorplates.

Recognition area	Complexity of text content	Total doorplates	Test 1	Test 2	Test 3	Average accuracy
Fixed Areas	Numerals	16	15	15	16	95.8%

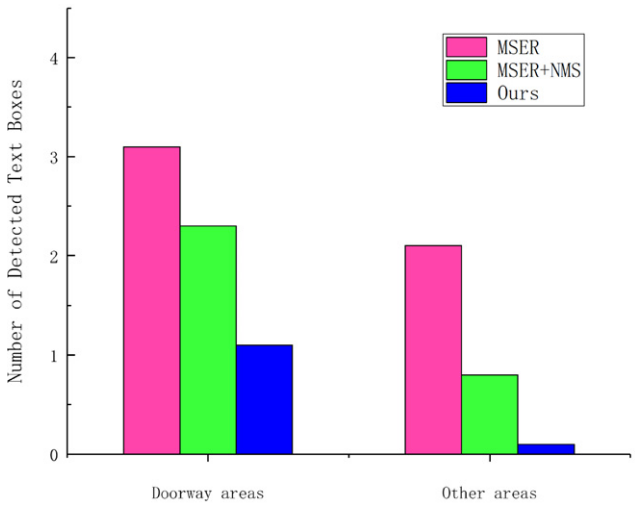


Figure 12. An average number of detected text boxes in different corridors.

image. Our method in this paper was to detect the fixed position area of the upper part of each picture, where the doorplates were most likely to appear. The detected results were shown in Figure 12.

In Fig. 12, the results were got from doorway areas when the mobile robot passed by and other areas on the wall. We found that the detected text boxes were over three when using the original MSER method. After using the non-maximum suppression (NMS) approach, the average number decreased. Fortunately, the result using our method was close to the real state, which had only one text box on the doorplate. Besides, most of the detected bounding boxes had the same height in the images. Text boxes of other areas in the images were filtered out. When the detected region was limited to being collected from the fixed height area of the images, the results changed greatly. There were negligible text regions detected using our method in other areas of the corridors.

In the second experiment, the text recognition was implemented based on the detected text boxes and the Tesseract OCR engine. We did three times tests in the corridor environment, mainly recognizing the text content of the doorplates. We counted the correctly recognized number of all doorplates in the corridor and computed the average correct rate, which was shown in Table I.

From Table I, we found that the text recognition system performs well in a simple text content environment like the dormitory building. The walls on both sides of the corridor in the dormitory building which was a place for students to rest at night had no more text information except for room numbers. What's more, the room numbers were usually on the doorplates posted on the top part of the doors. The recognized result of these text regions with numbers could get a 95.8% highly average correct rate. Actually, the result was worse if we detected and recognized the whole image area. Therefore, it was meaningful to focus on the information of fixed areas of each image.

Table II. Mapping results using different methods.

Method	Length (m)	Accuracy	Width (m)	Accuracy
	43.50		1.85	
Gmapping [16]	32.11	73.82%	1.85	100%
Hector [36]	31.85	70.31%	1.85	100%
Cartographer [17]	41.15	94.60%	1.85	100%
Gmapping+Text	41.75	95.58%	1.85	100%
Hector+Text	40.65	93.45%	1.85	100%
Cartographer+Text	42.30	97.24%	1.85	100%

In the third experiment, we chose a corridor environment from the second floor of dormitory building. The length and width of the corridor are 43.50 meters and 1.85 meters respectively. We used six SLAM methods to map the corridor and made statistics on the construction data. The mapping results was shown in Table II.

In Table II, the width was mapped with high accuracy because it was within the scanning radius of the laser sensor. We found that our method combined the SLAM methods with text features greatly improved the map accuracy, especially for the first two SLAM methods. The accuracy could not up to be 100% because of the laser sensor error and odometer error.

In the last experiment, we placed the mobile robot in 30 different positions in the second-floor corridor, then counted the success times of localization tests using AMCL and our improved MCL method respectively. The map was constructed manually to ensure the accuracy of the map, which was mainly used to test our localization algorithm and shown in Figure 13. In the initial state, the mobile robot was placed at each test position, and then three different methods were used to realize the global localization experiment. The mobile robot was allowed to move around. When the particles were converged to one cluster, the localization process ended. The result was shown in Table III and our method using both AMCL and text features had a higher localization success rate. It is evident that our result is better than the method in the paper [13].

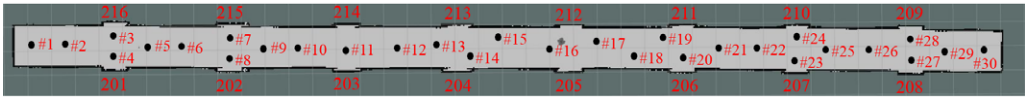


Figure 13. Constructed map and initial tested positions.

Table III. Localization results from different methods.

Method	Test times	Success times	Success rate
AMCL [18]	30	12	40.0%
Method [13]	–	–	80.0%
Text+AMCL	30	28	93.3%

5. CONCLUSIONS

In this paper, we presented an approach to assist the mobile robot localization in a long corridor environment by using laser-based SLAM and text features. In the mapping stage, we firstly control the mobile robot moving along the corridor and keeping a fixed distance from one side of the walls. Then, the camera oriented to the opposite wall, captures images to find text regions. Finally, the text features are used for building a grid map and saved into a text map. In the localization stage, the mobile robot needs a correctly initial pose, otherwise, the following positioning will be wrong. Before using the AMCL method, the robot moves according to a strategy along the corridor to find the nearest door. Then, by detecting and recognizing the text features, the mobile robot can get a precise pose. The above steps and methods are used to avoid the wrong positioning of similar doors or locations in the long corridor environment. In the experiments, we evaluate the performance of our proposed approach, and the results demonstrate a significant improvement in both the mapping and localization stages.

The future work we need to consider is to join the manipulator on the robot platform. When the door is closed or half-opened, the mobile robot can operate the door handle and open the door, then enter the room.

ACKNOWLEDGMENT

The authors would like to acknowledge all developers and maintainers of ROS and OpenCV, which are open-source software communities and provide convenient software packages. This work was supported by the National Natural Science Foundation of China (NSFC, grant number 61703067, 61803058, 51604056, 51775076), the Science and Technology Research Project of Chongqing Education Commission (grant number KJ1704072), the Science and Technology Project of Guizhou (grant number [2018]1180), and Doctoral Talent Train Project of Chongqing University of Posts and Telecommunications (grant number BYJS202006).

REFERENCES

[1] R. Ross and R. Hoque, "Augmenting GPS with geolocated fiducials to improve accuracy for mobile robot applications," Appl. Sci. **10**, 146 (2020).

[2] B. Tao, H. Wu, Z. Gong, Z. Yin, and H. Ding, "An RFID-based mobile robot localization method combining phase difference and readability," IEEE Trans. Autom. Sci. Eng. **18**, 1–11 (2020).

[3] X. Yu, Z. Fan, H. Wan, Y. He, J. Du, N. Li, Z. Yuan, and G Xiao, "Positioning, navigation, and book accessing/returning in an autonomous library robot using integrated binocular vision and QR code identification systems," Sensors **19**, 783 (2019).

[4] L. Zhang, Z. Chen, W. Cui, B. Li, C. Chen, Z. Cao, and K. Gao, "Wifi-based indoor robot positioning using deep fuzzy forests," IEEE Internet Things J. **7**, 10773–10781 (2020).

[5] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," Sensors **16**, 707 (2016).

[6] W. Deng, K. Huang, X. Chen, Z. Zhou, C. Shi, R. Guo, and R. Zhang, "Semantic RGB-D SLAM for rescue robot navigation," IEEE Access **8**, 221320–221329 (2020).

[7] Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li, "A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles," IEEE Trans. Intell. Transp. Syst. 1–15 (2021).

[8] H. Peel, S. Luo, A. G. Cohn, and R. Fuentes, "Localisation of a mobile robot for bridge bearing inspection," Autom. Constr. **94**, 244–256 (2018).

[9] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM.," IEEE Trans. Robot. **37**, 1874–1890 (2021).

[10] E. Bayramoglu, N. A. Andersen, N. K. Poulsen, J. S. Andersen, and O. Ravn, "Mobile robot navigation in a corridor using visual odometry," *Proc. IEEE Int'l. Conf. on Advanced Robotics* (IEEE, Piscataway, NJ, 2009), pp. 1–6.

[11] N. Radwan, G. D. Tipaldi, L. Spinello, and W. Burgard, "Do you see the bakery? leveraging geo-referenced texts for global localization in public maps," *Proc. IEEE Int'l. Conf. on Robotics and Automation* (IEEE, Piscataway, NJ, 2016), pp. 4837–4842.

[12] B. Li, D. Zou, D. Sartori, L. Pei, and W. Yu, "Textslam: Visual slam with planar text features," *Proc. IEEE Int'l. Conf. on Robotics and Automation* (IEEE, Piscataway, NJ, 2020), pp. 2102–2108.

[13] M. Tomono and S. Yuta, "Mobile robot navigation in indoor environments using object and character recognition," *Proc. IEEE Int'l. Conf. on Robotics and Automation* (IEEE, Piscataway, NJ, 2000), pp. 313–320.

[14] E. Hrustic and D. Vivet, "Using traffic signs as landmarks in object-oriented EKF-SLAM," *Int'l. Conf. on Control, Automation, Robotics and Vision (ICARCV)* (IEEE, Piscataway, NJ, 2020), pp. 273–280.

[15] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," IEEE Trans. Robot. **23**, 34–46 (2007).

[16] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D rangefinder SLAM," *Proc. IEEE Int'l. Conf. on Robotics and Automation* (IEEE, Piscataway, NJ, 2016), pp. 1271–1278.

[17] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," Artif. Intell. **128**, 99–141 (2001).

[18] S. Xu, W. Chou, and H. Dong, "A robust indoor localization system integrating visual localization Aided by CNN-based image retrieval with Monte Carlo localization," Sensors **19**, 249 (2019).

[19] S. H. Choi and C. G. Park, "Image-based Monte-Carlo localization with information allocation logic to mitigate shadow effect," IEEE Access **8**, 213447–213459 (2020).

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

[21] S. Ashraf, Z. Aslam, S. Saleem, S. Afnan, and M. Aamer, "Multibiometric sustainable approach for human appellative," CRPASE Trans. Electr. Electron. Comput. Eng. **6**, 146–152 (2020).

[22] L. Neumann and J. Matas, "Real-time scene text localization and recognition," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2012), pp. 3538–3545.
[23] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2010), pp. 2963–2970.
[24] M. Busta, L. Neumann, and J. Matas, "Fastext: Efficient unconstrained scene text detector," *Proc. IEEE Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2015), pp. 1206–1214.
[25] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," *Computer Vision – ECCV 2016. ECCV 2016*, Lecture Notes in Computer Science, Vol. 9912 (Springer, Cham, 2016).
[26] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2017), pp. 5551–5560.
[27] Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, W. Lin, and W. Chu, Inceptext: A new inception-text module with deformable psroi pooling for multi-oriented scene text detection. arXiv Preprint arXiv:1805.01167, (2018).
[28] A. P. Tafti, A. Baghaie, M. Assefi, R. H. Arabnia, Z. Yu, and P. Peissig, "OCR as a service: an experimental evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym," *Proc. Springer Int'l. Symposium on Visual Computing* (Springer, Cham, 2016), pp. 735–746.
[29] A. S. Agbemenu, J. Yankey, and E. O. Addo, "An automatic number plate recognition system using opencv and tesseract ocr engine," Int. J. Comput. Appl. **180**, 1–5 (2018).
[30] 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 Trans. Pattern Anal. Mach. Intell. **39**, 2298–2304 (2016).
[31] 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 Recognit. Lett. **136**, 134–141 (2020).
[32] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V.R. Chandrasekhar, S. Lu, and F. Shafait, "ICDAR 2015 competition on robust reading," *Proc. 13th Int'l. Conf. on Document Analysis and Recognition (ICDAR)* (IEEE, Piscataway, NJ, 2015), pp. 1156–1160.
[33] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv Preprint arXiv:1601.07140, (2016).
[34] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics* (MIT Press, Cambridge, MA, USA, 2005), pp. 279–484, Chapter 3.
[35] L. Xu, C. Feng, V. R. Kamat, and C. C. Menassa, "An occupancy grid mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments," Autom. Constr. **104**, 230–245 (2019).
[36] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," *2011 IEEE Int'l. Symposium on Safety, Security, and Rescue Robotics* (IEEE, Piscataway, NJ, 2011), pp. 155–160.
[37] G. Ge, Y. Zhang, Q. Jiang, and W. Wang, "Visual features assisted robot localization in symmetrical environment using laser SLAM," Sensors **21**, 1772 (2021).