Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Visual-feature-assisted mobile robot localization in a long corridor environment*

Gengyu GE[†1,3], Yi ZHANG[†‡2], Wei WANG[1], Lihe HU[1], Yang WANG[1], Qin JIANG[1]

[1]*School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2]*School of Advanced Manufacturing Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[3]*School of Information Engineering, Zunyi Normal University, Zunyi 563006, China*
[†]E-mail: gegengyu_2021@163.com; zhangyi@cqupt.edu.cn

Received May 14, 2022; Revision accepted Nov. 15, 2022; Crosschecked Mar. 31, 2023

Abstract: Localization plays a vital role in the mobile robot navigation system and is a fundamental capability for autonomous movement. In an indoor environment, the current mainstream localization scheme uses two-dimensional (2D) laser light detection and ranging (LiDAR) to build an occupancy grid map with simultaneous localization and mapping (SLAM) technology; it then locates the robot based on the known grid map. However, such solutions work effectively only in those areas with salient geometrical features. For areas with repeated, symmetrical, or similar structures, such as a long corridor, the conventional particle filtering method will fail. To solve this crucial problem, this paper presents a novel coarse-to-fine paradigm that uses visual features to assist mobile robot localization in a long corridor. First, the mobile robot is remote-controlled to move from the starting position to the end along a middle line. In the moving process, a grid map is built using the laser-based SLAM method. At the same time, a visual map consisting of special images which are keyframes is created according to a keyframe selection strategy. The keyframes are associated with the robot's poses through timestamps. Second, a moving strategy is proposed, based on the extracted range features of the laser scans, to decide on an initial rough position. This is vital for the mobile robot because it gives instructions on where the robot needs to move to adjust its pose. Third, the mobile robot captures images in a proper perspective according to the moving strategy and matches them with the image map to achieve a coarse localization. Finally, an improved particle filtering method is presented to achieve fine localization. Experimental results show that our method is effective and robust for global localization. The localization success rate reaches 98.8% while the average moving distance is only 0.31 m. In addition, the method works well when the mobile robot is kidnapped to another position in the corridor.

Key words: Mobile robot; Localization; Simultaneous localization and mapping (SLAM); Corridor environment; Particle filter; Visual features
https://doi.org/10.1631/FITEE.2200208 **CLC number:** TP242.6

1 Introduction

Mobile robots are becoming more and more common as intelligent devices which can serve human beings. For instance, there are many commercial service robots used for transporting goods in factory workshops, industrial parks, hotels, restaurants, hospitals, and so on. They were especially important during the COVID-19 epidemic (Wang XV and Wang, 2021). With people needing to avoid close contact, a mobile robot becomes very valuable. Intelligence and autonomy are first manifested in a mobile robot's ability to move and navigate autonomously. In the navigation process, localization is the fundamental capability because it is the basis of path planning. Generally,

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (Nos. 61703067, 61803058, 51604056, and 51775076), the Science and Technology Research Project of Chongqing Education Commission, China (No. KJ1704072), and the Doctoral Talent Train Project of Chongqing University of Posts and Telecommunications, China (No. BYJS202006)
ORCID: Gengyu GE, https://orcid.org/0000-0001-9913-0785; Yi ZHANG, https://orcid.org/0000-0001-6935-5721

localization refers to the process of finding the pose of a mobile robot given a map. It is divided into three categories: global localization, local localization, and the kidnapped robot problem (Meng et al., 2021). Global localization means that the mobile robot is powered on and started at an unknown initial position. Many commercial robots require humans to give them an initial pose and the target location manually. The mobile robot then moves to its goal using a navigation system. However, for an autonomous robot facing the future in an unmanned application scenario, autonomous localization is a necessary skill. Therefore, finding feature information that can be located in the environment becomes very important. Local localization or pose tracking means that the mobile robot has a known initial pose and tracks the pose according to a given map and motion prediction. The kidnapped robot problem refers to a moving mobile robot being kidnapped to another place, and then needing to recover its correct pose (Chen RJ et al., 2021). The kidnapped robot problem can be thought of as another global localization problem and the key is how to find itself when kidnapped.

Global positioning system (GPS) is an ideal positioning method in an outdoor environment and has high accuracy and robustness (Yousuf and Kadri, 2021). However, in an indoor environment, the GPS signal may be denied or lose its effectiveness. To solve this crucial problem, researchers from different disciplines have proposed many different positioning methods. Some of them are beacon-based solutions, such as radio frequency identification (RFID) (Motroni et al., 2021), WiFi (Zhang et al., 2020), light emitting diode (LED) (Wu et al., 2017), ultra-wideband (UWB) technology (Djosic et al., 2021), and quick response (QR) codes (Katsikis et al., 2022), which can be used as references for mobile robot localization. However, these methods require artificial beacons or landmarks to be placed in fixed locations in advance, which is not flexible. Besides, if the mobile robot's task and moving route change, the beacon arrangement also needs to be changed. In contrast, other methods using onboard sensors can actively perceive environmental information. The commonly used sensor is a laser light detection and ranging (LiDAR) or laser rangefinder which can measure distances from the robot to the obstacles (Kim and Chung, 2016). Three-dimensional (3D) LiDAR can achieve abundant point clouds of the 3D space environment, and is usually used in outdoor driverless application scenes, but the cost is very high (Chen XYL et al., 2020). Two-dimensional (2D) LiDAR scans the horizontal plane at a certain height in the vertical direction, obtains a set of distance values, and builds an occupancy grid map using the laser-based simultaneous localization and mapping (SLAM) method (Hess et al., 2016).

Given a previously built map, a mobile robot can obtain a pose estimation by combining the laser scan observation and motion model prediction. Among all the localization algorithms, the particle filter is the most widely used, and can solve all the three localization issues. Inspired by Monte Carlo thinking, a particle filter is a non-parametric and arbitrary distribution implementation scheme based on a Bayesian filter (Wang FS et al., 2018). If the environment has enough geometrical features, the particle filter can easily deal with the localization task. However, in real-world scenes, especially in man-made structures, there are many areas with similar, repeated, or symmetrical geometrical features. The mobile robot will fail to estimate its pose in these areas without a known initial pose or the assistance of third-party tools.

By contrast with the 2D laser rangefinder sensor, a cheap and lightweight camera provides more dense information, such as point features, colors, textures, and lines. In addition, there is semantic information such as objects and texts that can be extracted from an image (Zhao ZQ et al., 2019; Long et al., 2021). It is one of the most valuable sensors that can be used to perceive the environment and localize the pose for mobile robots. However, due to the lack of metric information, pure visual features are not suitable for flexible path planning and navigation. To solve the above problems, we propose an alternative scheme that uses a monocular camera to extract visual features of the indoor environment to assist mobile robot localization. In addition, if necessary, it can be extended to object detection and pedestrian tracking tasks using semantic visual information. A coarse-to-fine paradigm is used in our method. It uses image retrieval to obtain a coarse position candidate and then an improved Monte Carlo localization (MCL) algorithm (Thrun et al., 2001) to achieve a fine pose.

The whole framework of the mapping and localization system is depicted in Fig. 1. In the mapping

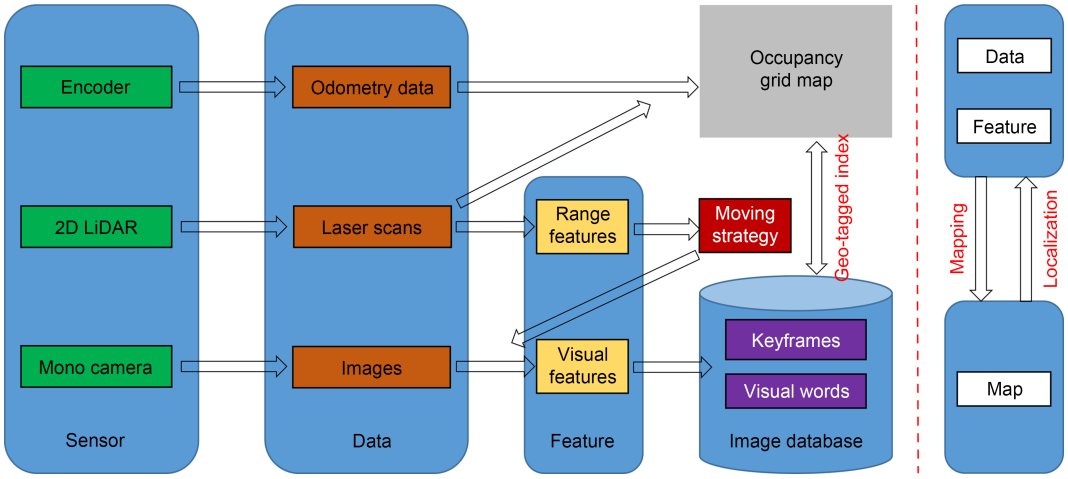

Fig. 1 Framework of the mapping and localization system

phase, an image database consisting of keyframes and visual words is created along with the building of an occupancy grid map. In the localization phase, according to our proposed moving strategy, a coarse localization is obtained using an image retrieval method, and then a fine localization is performed by employing an improved MCL approach. The main contributions of this work are as follows:

1. A hybrid map consisting of screened images and an occupancy grid map is built when the mobile robot moves from one end of the corridor to the other. The screened images are keyframes that are selected based on a selection strategy. Compared with conventional visual SLAM methods, our strategy can achieve dense and stable keyframes.

2. Each keyframe is associated with a mobile robot's pose relative to the global metric map. The associations or geo-tagged indexes are associated with each other by timestamps. A compound two-tuple data structure is used to store the associated information and is convenient for later image-pose searching.

3. A coarse-to-fine paradigm is used to achieve two-stage localization. According to the moving strategy, the mobile robot can find the best perspective to capture images and match them with the image database, to obtain a coarse localization. Based on the approximate location, an improved particle filtering approach is used to scatter the particles in a small range, instead of spreading them over the whole map. Experimental results show that our proposed approach is efficient and achieves a 98.8% successful localization rate where traditional MCL methods fail.

2 Related works

Localization usually needs to solve the question "Where am I?" and find the pose of the mobile robot according to a given map. Specifically, localization refers to a mobile robot achieving its position and orientation relative to the world coordinate system, and usually has three modes which are global localization, local localization, and robot kidnapping (Fox et al., 1999b). With a constructed map, a mobile robot can accomplish navigation tasks through real-time localization and path planning (Muhammad et al., 2022). To facilitate path planning and obstacle avoidance, the 2D probability occupancy grid map is the usual mode in an indoor environment (Qian et al., 2019; Xu et al., 2019). The occupancy grid map is a metric map that is constructed using a 2D laser rangefinder and laser-based SLAM solutions (Valente et al., 2019; Zhao JH et al., 2020, 2021).

The Markov localization method is one of the earliest approaches proposed for solving the robot positioning problem (Fox et al., 1999a). The grid localization approach employs histogram filters to estimate the mobile robot's pose according to the grid decomposition of a state space (Thrun et al., 2005). However, the computational load of the specific algorithm increases when the grid resolution increases. Consequently, this method is not suitable for real-time localization applications. The Kalman filter (KF) approach (Liu et al., 2021), especially its extended version, the extended Kalman filter (EKF) (Ullah et al., 2021), is usually employed to solve part of the localization problems

based on a map with landmarks or features. However, the KF series solutions need the initial pose information to process the pose tracking issue. In addition, this kind of method requires that the position and orientation obey a Gaussian distribution, which obviously does not conform to the actual situation in grid map localization. In contrast, the Monte Carlo approach uses a particle filter to realize a non-parametric implementation of the recursive Bayesian filter and can deal with these three localization modes. Thrun et al. (2005) used the MCL method to realize mobile robot localization even when the sensor data are noisy. The MCL series methods can realize multi-modal beliefs and have many advantages over KF series methods. However, they still fail when encountering similar environments or repeated geometric structures. Therefore, it is necessary to introduce other sensor data or features to assist mobile robot localization. Ge et al. (2022) and Zimmerman et al. (2022) used text information to achieve initialization and a coarse global localization, and then used the improved MCL methods to achieve other localization tasks. Ito et al. (2014) proposed a method that integrates WiFi and an RGB-D camera to solve the global localization problem. All those approaches have been compared with traditional MCL methods, and their localization results have better robustness and efficiency. As an alternative solution, we exploit visual features (Naseer et al., 2018) to assist the mobile robot localization in a long corridor. The system uses a visual camera which can be easily extended to semantic information acquisition applications.

3 Proposed methodology

At the mapping stage, a hybrid map consisting of an occupancy grid map and an image map is built. The grid map is built by performing the laser SLAM in an online mode, while the image map is completed by extracting keyframes and computing visual words. The coordinate of each keyframe is associated with the robot's pose where it performs laser SLAM and captures the images. At the navigation stage, the localization module based on sensor data and the previously built map works in real time. To ensure that the currently captured image can always search for a matched one in the database, the mobile robot needs to follow a relatively fixed moving strategy, especially in a corridor environment.

3.1 Moving strategy in a long corridor

First, we explain why the mobile robot needs a moving strategy rather than random moving. The camera used in visual SLAM or visual navigation applications is handheld or mounted on a driverless car, and consequently the moving routes are relatively fixed because of human subconscious behaviors. However, when it comes to the applications of autonomous mobile robots, for instance, the delivery robot moves in a corridor from one room to another and cannot ensure the same route every time. The result of changing the moving route is that the mobile robot fails to track the visual features built previously. A vivid description is shown in Fig. 2.

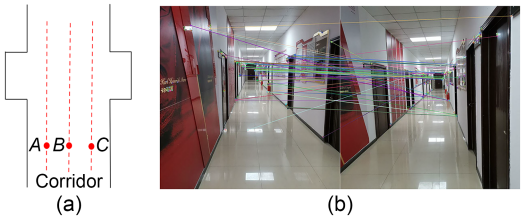

Fig. 2 Perspective of different positions: (a) different routes; (b) few and wrong matches using ORB features (References to color refer to the online version of this figure)

Three routes indicated by red dashed lines are shown in Fig. 2a, and three positions are indicated by red points. The images captured from different positions in a lateral line have significant changes in perspective because the corridor is long and narrow. Although there are several robust image feature descriptors like SIFT, SURF, and ORB (Rublee et al., 2011), which can be used for place recognition, these features are invariant in a limited variation range. There are only a few matched pairs between the visual features extracted from the image captured in position *A* and those extracted from the image captured in position *C*. In Fig. 2b, the left image is captured from position *A* and the right from position *C*.

ORB features are extracted from the two images, and then matched with each other. Only a few, and wrong, matching lines are achieved. Besides, the camera oriented in different directions has different matching results, even if it is in the same position. Therefore,

if a mobile robot moves along the left red dashed line as the route, and maps the corridor using visual SLAM, it will fail to localize itself when moves along the right red dashed line in the subsequent process. Consequently, the mobile robot needs to move within a fixed route range, preferably the middle line of the corridor, like the middle red dashed line, shown in Fig. 2a.

To make the mobile robot move along a relatively fixed route at different time points, range information relative to the walls on both sides of the corridor needs to be accurately known. The 2D LiDAR is used to measure the ranges from the center of the sensor to the obstacle. As shown in Fig. 3a, the left detected distance plus the right one equals the width of the corridor. However, due to sensor noise and measurement error, the equation is not accurate and can be expressed by

$$\left|d_{\mathrm{L}}+d_{\mathrm{R}}-w_{\mathrm{corridor}}\right|\leq e_1, \tag{1}$$

where d_{L} means the shortest distance between the left wall and the center of the laser sensor, d_{R} means the shortest one on the other side, w_{corridor} is the width of the corridor, and e_1 is the tiny error. When the robot passes through the doorway area, the following inequalities hold:

$$\left|d_{\mathrm{L}}+d_{\mathrm{R}}-(w_{\mathrm{corridor}}+2d_{\mathrm{da}})\right|\leq e_2, \tag{2}$$

$$\left|d_{\mathrm{L}}+d_{\mathrm{R}}-(w_{\mathrm{corridor}}+d_{\mathrm{da}})\right|\leq e_3, \tag{3}$$

where d_{da} means the depth of the door frame recessed into the wall, and e_2 and e_3 are tiny errors. Inequality (2) represents the rooms and doors in a face-to-face layout, as in Fig. 3a. Inequality (3) means that the doors on both sides of the corridor are staggered.

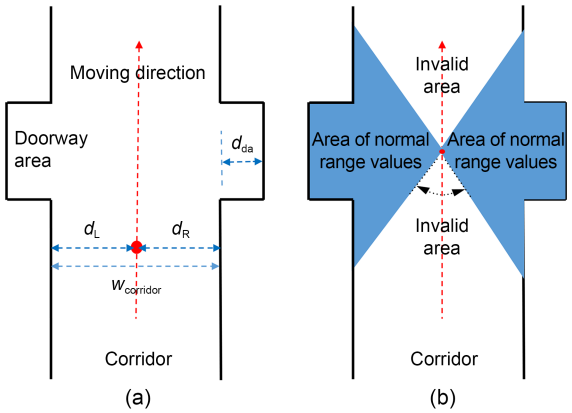

Fig. 3 Basic information and laser scans in a corridor environment: (a) annotation data; (b) valid and invalid laser scanning areas

When a mobile robot knows itself to be in a corridor area, it is not difficult to move along the middle line of the corridor. The robot can slightly adjust its position and orientation, so that the data measured by the laser sensor satisfy the following inequality:

$$\left|d_{\mathrm{L}}-d_{\mathrm{R}}\right|\leq e_4, \tag{4}$$

where e_4 is a tiny error. Combined with inequalities (1)–(3), the mobile robot can know whether it is near a doorway area and then decides whether d_{L} or d_{R} equals $0.5w_{\mathrm{corridor}}$ or $0.5w_{\mathrm{corridor}}+d_{\mathrm{da}}$.

Another important problem that needs to be considered is how a robot can know that it is in a corridor area. In most indoor scenes, especially inside a room, the laser sensor can obtain the whole valid distance data around the robot. If it cannot, the invalid areas most likely exceed the limit of the LiDAR scanning radius, and there is a high probability that the robot is in a corridor, as shown in Fig. 3b. To describe these problems objectively, we give the following definitions:

Definition 1 Given a 2D laser rangefinder that has a measurement angle range of 360°, the maximum measuring distance is defined as r_{max}, the minimum measuring distance as r_{min}, and the angular resolution as ϕ_{min}. Then the raw data can be described by the following formula (Fig. 4a):

$$R=\{(r_i,\phi_i)|i=1,2,...,N\}, \tag{5}$$

where N equals $360/\phi_{\mathrm{min}}$ and represents the total number of scanning points $P=\{p_1,p_2,...,p_N\}$.

Definition 2 An invalid scan area angle θ_{null} has a dynamically variable range. The maximum value θ_{max} is obtained when the laser sensor mounted on the robot is close to the wall, while the minimum value θ_{min} is obtained when the robot is located on the middle line along the corridor. The two extreme values are defined as follows:

$$\theta_{\mathrm{max}}=\arcsin\frac{w_{\mathrm{corridor}}}{r_{\mathrm{max}}}, \tag{6}$$

$$\theta_{\mathrm{min}}=\arcsin\frac{w_{\mathrm{corridor}}}{2r_{\mathrm{max}}}. \tag{7}$$

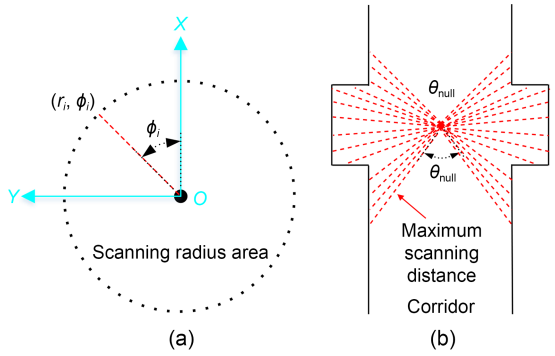

Fig. 4 Laser scans description (a) and the angle of invalid laser scanning area (b)

If the mobile robot is moving in a corridor, then inequality (8) holds, which is vividly shown in Fig. 4b.

$$\theta_{\min} < \theta_{\text{null}} < \theta_{\max}. \tag{8}$$

We record the first and last angles of the invalid area and compute the invalid area angle θ_{null} according to Eq. (9):

$$\theta_{\text{null}} = \left| \phi_i - \phi_j \right|, \tag{9}$$

where ϕ_i means the i^{th} angle whose return distance value from the detected point p_i is invalid, and ϕ_j represents the j^{th} one. The constant angle range between these 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 inequality (8). Similarly, if two invalid areas satisfy inequality (8) and are distributed like that shown in Fig. 3b, then the robot is most likely to be located away from the end of the corridor. According to the information extracted from the laser scanning data and the moving strategy described above, the mobile robot can autonomously move to the free area along a preset fixed route.

3.2 Keyframe selection

Keyframes are selected when the mobile robot moves along a middle line in a corridor. While the laser SLAM builds the grid map, continuous image frames are recorded by the camera mounted on the robot platform. However, not all the images are used for building an image map or database because a large number of adjacent images have almost the same content. To cull the redundant frames, we need to decide which of them is not necessary. A keyframe selection strategy is designed based on the ORB features (Rublee et al., 2011) that are computed in every image. Then, the similarity between two adjacent images is computed for the decision. The detailed description is as follows:

ORB features combine the orientated 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

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y), \tag{10}$$

where the values of p and q are limited to 0 or 1, and $I(x, y)$ is the pixel value of the pixel position (x, y). Then the intensity centroid is computed from those moments:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right). \tag{11}$$

The orientation vector can be constructed by connecting two points. One of them is the center of the keypoint corner, and the other is the centroid of the patch. The orientation is then computed as

$$\theta = \arctan\left(m_{01}, m_{10}\right). \tag{12}$$

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

$$\tau(p;x,y) := \begin{cases} 1, p(x) < p(y), \\ 0, p(x) \geq p(y), \end{cases} \tag{13}$$

where $p(x)$ and $p(y)$ are the intensities at points x and y, respectively. There are usually 256 pairs of points selected to express a keypoint. The descriptor is defined as a vector of 256 binary tests:

$$f_n(p) := \sum_{i=1}^{n} 2^{i-1} \tau(p;x_i,y_i), \tag{14}$$

where n equals 256. Then a learning method that uses principal component analysis (PCA) or other dimensionality reduction strategy is used to assist in realizing

a rotation-invariant BRIEF descriptor. The combination of oFAST and rBRIEF is called an ORB feature, and an example is shown in Fig. 5. The center of a red cycle is the keypoint and the radius means the orientation. In addition, the features are extracted mainly from the texture information such as bulletin boards, stickers, or room numbers on the wall.

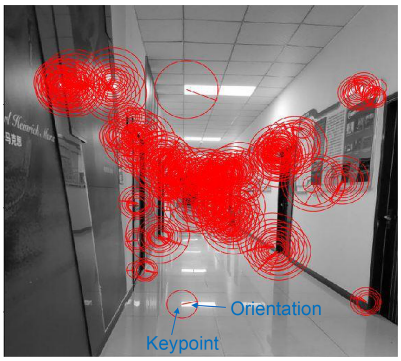

Fig. 5 ORB features extracted from an image (References to color refer to the online version of this figure)

The subsequent work is to decide which image can be thought of as a keyframe. Different from the monocular camera visual SLAM system which needs map initialization, the first image captured by the robot in our proposed method is treated as a keyframe by default. After that, a new image that is considered a keyframe should meet the following conditions:

(1) The total number of ORB features extracted from the image needs to exceed the minimum threshold. To give an objective numeral rather than a random value, some preparation needs to be completed in advance. First, we record a video at a distance of 10 m in the corridor. Then, we calculate the number of features of each frame offline and take a parameter $F_{\min}$ as the reference threshold which is expressed as Eq. (15), where N is the total number of images, I_i is the number of features of the i^{th} image, and $I_{\min}$ is the image which has the minimum number of features.

$$F_{\min} = \max\left\{0.5 \times \frac{1}{N}\sum_{i=1}^{N} I_i, I_{\min}\right\}. \tag{15}$$

(2) More than 10 frames must have passed from the last keyframe decision, due to the high frame rate of the camera and the low moving speed of the mobile robot.

(3) The current frame shares fewer than 90% but more than 70% matched points with the last keyframe.

(4) The mobile robot has moved about 0.02 m from the position of the last keyframe.

Fig. 6 shows the results of different keyframe selection strategies in a corridor environment. The blue trapezoidal blocks are camera's poses that represent the keyframes of those captured images and will be saved as one part of the image map. The green one indicates the current frame. Map points are indicated by the black and red points which are not saved in the image map in our proposed system.

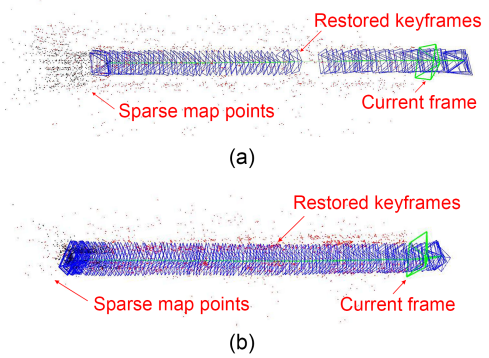

Fig. 6 Keyframes and map points: (a) ORB-SLAM; (b) improved strategy of keyframe selection (References to color refer to the online version of this figure)

Compared with the visual SLAM approach which refers to epipolar geometry and removes more redundant images, the strategy of retaining only keyframe images reduces the computational workload of geometric transformation between two adjacent image frames. Moreover, the map points are not needed to be stored, which also reduces computing sources involving triangulation. Table 1 is an extraction result compared with the ORB-SLAM method (Mur-Artal et al., 2015) by intercepting a section of 10 m from a corridor inside an office building in our university. Our strategy can

Table 1 Keyframes using different approaches

Method	Total number of images	Initial number of keyframes	Total number of keyframes	Extraction rate (%)
ORB-SLAM	1502	9	42	2.8
Ours	1502	1	103	6.9

obtain dense and uniform keyframes which are conducive to later place recognition.

3.3 Visual word generation

The achieved keyframes can be stored directly in an image database. However, when a newly captured image needs to be retrieved from the database, the workload of feature point matching between images is particularly large. A popular method is to use the bag-of-words technique to train a large number of images into a visual vocabulary at the offline stage and then convert a new image into a numerical vector at the online stage. The advantage of using the bag-of-words model is in reducing the time needed for image matching.

As shown in Fig. 7, the right part is a visual vocabulary k-dimensional tree (k-d tree) which is created by discretizing the ORB descriptor space into m visual words. Using the k-means approach, the descriptors are clustered into k branches in each iteration of the d times. The k^d leaf nodes are the visual words with weighting values according to the inverse document frequency (IDF).

After creating the bag-of-words vocabulary, the mobile robot moves in the environment and acquires new images continuously. Every newly selected keyframe will be converted to a visual vector $v_t=\{(w_i, \eta_i)|i=1, 2, ..., m\}$, where m is the number of visual words in the vocabulary database. The weight η_i is computed from the product of term frequency TF_i and the inverse document frequency IDF_i:

$$\begin{cases} \mathrm{TF}_i = \dfrac{n_{ik}}{n_k}, \\ \mathrm{IDF}_i = \log_2 \dfrac{N}{n_i}, \\ \eta_i = \mathrm{TF}_i \times \mathrm{IDF}_i, \\ v_t = \{(w_1, \eta_1), (w_2, \eta_2), ..., (w_m, \eta_m)\}, \end{cases} \tag{16}$$

where n_k is the number of all feature points in the k^{th} image, n_{ik} is the number of occurrences of the i^{th} word in the k^{th} image, TF_i represents the weight of the i^{th} word in the k^{th} image, N is the total number of images of the training database used to build the vocabulary tree, and n_i is the number of occurrences of the i^{th} word in the database. Actually, n_i/N means the weight of this word in the database; the larger the value, the lower the recognition rate. Consequently, the suitable weight is replaced by IDF_i.

In addition, an inverse index list is created to store a list of the most similar images for each word w_i, as the left part of Fig. 7 shows. The scoring value $s(v_A, v_B)$ measures the similarity of two images A and B using their vocabulary vector pattern, which is calculated based on an L1-score as Eq. (17). Once a new image is captured to match with the visual word database, the inverse index list improves the matching speed.

$$s(v_A, v_B) = 1 - \frac{1}{2}\left|\frac{v_A}{|v_A|} - \frac{v_B}{|v_B|}\right|. \tag{17}$$

3.4 Building a hybrid map

For many application scenarios, the mobile robot does not need to enter a specific room; for example, during the COVID-19 epidemic, the quarantined persons in the hotel could open the room door after receiving the message, and then take away the items. Therefore, it is not necessary to acquire images inside the rooms. We remotely control the mobile robot to move from one end of the corridor along the middle line to the other end. A loosely coupled hybrid map is constructed in the moving process.

The hybrid map consists of an image map and an occupancy grid map. The former consists of keyframes and visual words. The latter is the probability-occupied

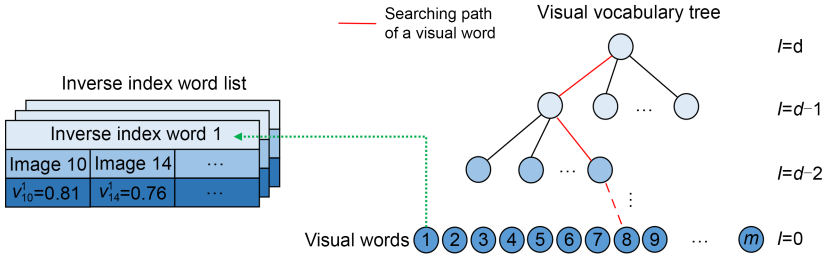

Fig. 7 Visual vocabulary tree and inverse index

grid map constructed by the laser SLAM method. The link between these two is the timestamp which can be achieved by the robot operating system (ROS) software. Through time alignment, a keyframe can be associated with the pose coordinates of the robot. The geo-tagged index associates a keyframe with a mobile robot's pose, and is defined as follows:

$$\text{GeoIndex}_{\text{id}}=\{\text{KF}_{\text{id}}, \text{Pose}_{\text{id}}=(x_k, y_k, \theta_k)\}. \quad (18)$$

The schematic diagram is shown in Fig. 8.

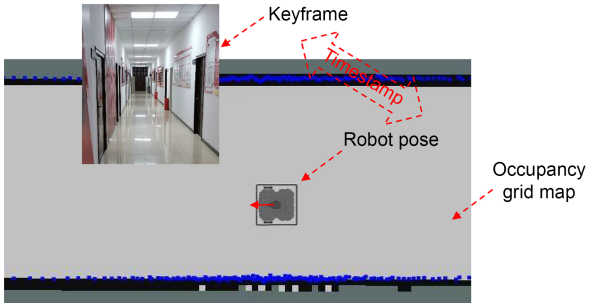

Fig. 8 Hybrid map based on the timestamp

3.5 Localization mode

Given a known map, the localization mode uses a coarse-to-fine paradigm to achieve the mobile robot's pose. Fig. 9 gives a framework of the paradigm: the left part is a visual image retrieval process that can acquire a coarse candidate location, and the right part is a fine pose estimation process that combines laser data, odometry data, and the occupancy grid map.

In the left part of Fig. 9, after the mobile robot is powered on, initialized, or recovered from a fault, it will adjust to a suitable position according to the moving strategy described in Section 3.1, and then start to capture images and handle them. A newly captured image will be transformed into a visual bag-of-words vector. Then a similarity value is computed by matching the visual vector with those in the image map. The vector and keyframe with the highest score are what we are looking for. According to Eq. (18), the pose of the mobile robot is obtained by looking up the geo-tagged index. However, the actual situation is not perfect, and most of the time, multiple high-score keyframes are retrieved which will lead to longitudinal errors. Due to the observation errors of the sensors, lateral errors also occur. Consequently, the image retrieval part obtains only a candidate location with a small range of uncertainty.

In the right part of Fig. 9, an improved MCL method is used to obtain a fine pose estimation. In the particle initialization phase, different from a traditional MCL method which generates particles uniformly throughout the whole grid map, we initialize the particles in a small range as Fig. 10 shows. The particles are shown as the red dots and spread evenly inside an ellipse, where the mobile robot is most likely located according to the image retrieval result. Then, the robot executes the classical particle filter localization mode. To detect the kidnapped robot problem and successfully recover the correct pose from the abduction, a fixed time interval of image retrieval strategy is used.

The blue star in Fig. 10 means the mobile robot's pose with the maximum probability, and the probabilities of other particles which are represented by red circles decrease towards the edge of the ellipse. Due to the moving strategy in Section 3.1, the directions of particles are consistent and have less uncertainty. Based on prior knowledge, the initial weights are set based on the Gaussian density function:

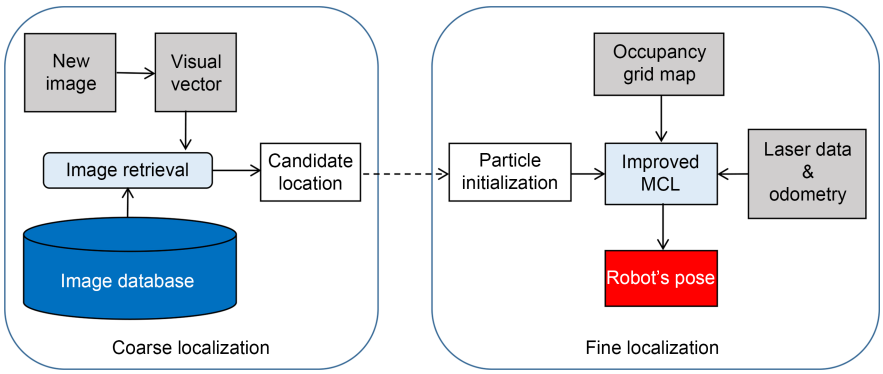

Fig. 9 Localization mode (MCL: Monte Carlo localization)

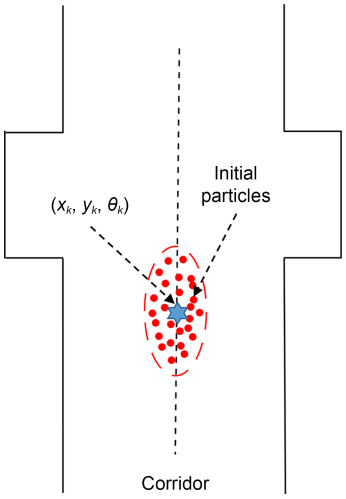

Fig. 10 Potential area of particle initialization (References to color refer to the online version of this figure)

$$w_{\text{initial}} = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\left(\frac{(x-x_k)^2}{2\sigma_x^2}+\frac{(y-y_k)^2}{2\sigma_y^2}\right)\right), \quad (19)$$

where (x_k, y_k, θ_k) is the most likely pose of the mobile robot, and the weight of each particle is set depending on its distance from (x_k, y_k). σ_x and σ_y are the variances in the x and y directions, respectively. After the initialization, the iterative filtering scheme is used to complete the fine positioning work. A detailed description is shown in Algorithm 1.

Algorithm 1 Improved adaptive Monte Carlo localization (AMCL) algorithm

Input: Sample set $\bar{\chi}_{t-1}$, control u_t, observation z_t (laser data, keyframe), and hybrid map M

Output: Particle set with the optimal estimated pose χ_t

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

2 Uniformly sample poses according to a keyframe retrieval

3 $w_{\text{initial}} = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\left(\frac{(x-x_k)^2}{2\sigma_x^2}+\frac{(y-y_k)^2}{2\sigma_y^2}\right)\right)$

4 Robot moves forward

5 **for** j=1 to N **do**

6 $\hat{x}_t^{[j]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[j]})$

7 $\hat{w}_t^{[j]} = \text{measurement\_model}(z_t, \hat{x}_t^{[j]}, M)$

8 $\bar{\chi}_t = \bar{\chi}_t \cup \left\{< \hat{x}_t, \hat{w}_t >\right\}$

9 **end for**

10 **for** k=1 to N **do**

11 Draw sample $\hat{x}_t^{[k]}$ from $\bar{\chi}_t$ with probability $\propto \hat{w}_t^{[k]}$

12 Add $\hat{x}_t^{[k]}$ to χ_t

13 **end for**

14 **if** time interval equals Δt **then**

15 go to step 2

16 **end if**

17 **return** χ_t

4 Experiments and discussion

Our experimental mobile robot is a modified two-wheel differential driving chassis based on a Kobuki robot (Yujin Robot Company, Korea), which is also introduced in the ROS community. The mobile robot shown in Fig. 11a is equipped with an RPLIDAR A2 laser LiDAR (SLAMTEC Company, China), a monocular camera with 1280×960 resolution and 30 frames per second, and a microcomputer with 1.5 GHz ARM Cotex-A72 CPU and 8 GB RAM. The laser LiDAR has a 360° rotation range and an effective measurement radius of 8–10 m. It is noteworthy that the camera is mounted lower than the LiDAR sensor to avoid blocking the LiDAR beam. The experimental environment is a long corridor with many symmetrical and similar areas inside an office building next to our laboratory (Fig. 11b). The length of the corridor is 36.10 m, the width is 2.17 m, and the depth of the door frames is 0.07 m. There are two types of door-frame widths, 1.00 and 2.25 m. The ground truths of the signed positions and the distances are measured manually using a tape ruler.

We removed items such as fire extinguishers and garbage cans in the corridor to reduce the impact on the experiments. The mobile robot was remote-controlled to move from the building entrance to the other end of the corridor, and a probabilistic occupancy grid map was built using the Gmapping SLAM solution (Grisetti et al., 2007). It should be noted that the mobile robot did not reach the wall at the endpoint, but stopped at a position about 0.8 m away from it. Fig. 12 shows the mapping result, and the red solid rectangles represent the door-frame areas. Inside the dashed parallelograms are several geometrically similar areas. In addition, 357 images were extracted and determined as the keyframes using our keyframe selection strategy.

4.1 Global localization

In the global localization experiment, we chose 10 different initial locations (marked with red dots in

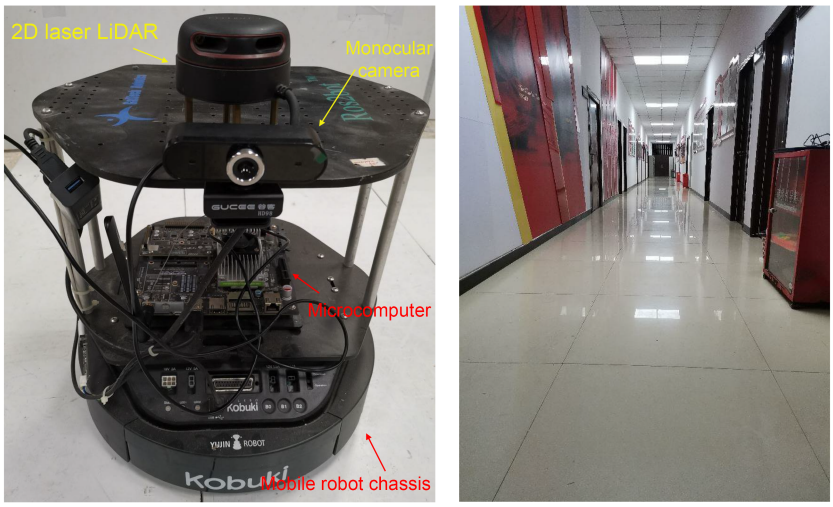

Fig. 11 Experimental setup and environment: (a) the mobile robot platform used to verify the proposed method; (b) a long corridor inside an office building

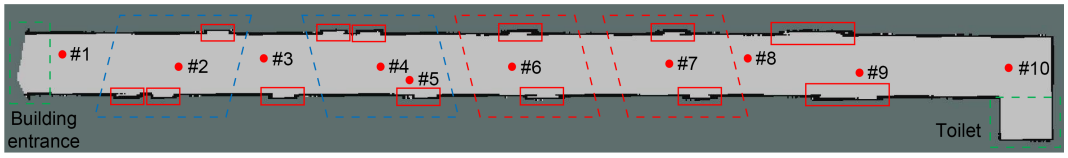

Fig. 12 Occupancy grid map the environment

Fig. 12) to test our proposed localization method and the traditional MCL method. To obtain the average and generality of the experimental results, each of the methods in each specific location was tested 10 times. The mobile robot was placed in the fixed positions and restarted from there. Fig. 13 shows an experiment of the global localization process using the traditional adaptive Monte Carlo localization (AMCL) method. The mobile robot restarted from position #8 and scattered particles evenly on the whole map to realize initialization. Due to the randomness, the pose of the robot that shown in Fig. 13a is not the real one. After the mobile robot moved a short distance as shown in Fig. 13b, the particles gradually converged to many clusters which were the potential poses. In Fig. 13c, the mobile robot moved to an area near position #10, and the particles had converged into a single cluster. Unfortunately, the final localization result was wrong.

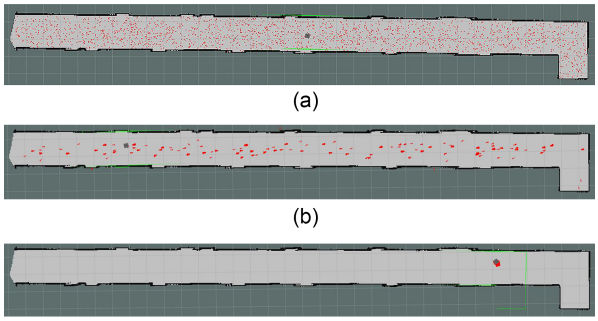

Fig. 13 Global localization results of the adaptive Monte Carlo localization (AMCL) method: (a) initialization of the particles; (b) particle distribution after moving a short distance; (c) wrong localization when particles converged

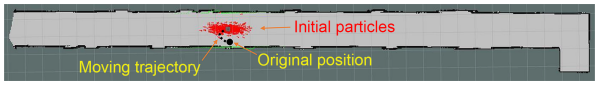

Fig. 14 Initialization of the proposed method

Fig. 14 is the initialization of our proposed method. The mobile robot restarted from position #5 and adjusted its pose to a suitable position. According to our proposed moving strategy and the laser scan data, the mobile robot moved through a small range to the middle line of the corridor. Then the robot captured images and matched them with the image database. The most similar ones were the candidate areas where the mobile robot was most likely to be. Finally, the

particle initialization was performed based on Fig. 10 and Eq. (19). After several iterations, the particles were more concentrated and the position with the maximum weight was the estimated pose of the mobile robot.

We recorded the number of iterations when the particles converged to a single cluster in each of the 10 experiments and computed the average value. If the mobile robot did not move, but stayed where it was, the particles could not be converged. Therefore, the mobile robot had to move around to find the salient geometric or visual features for localization. Similarly, the average moving distance was recorded and computed when the particles had converged. When the particles had converged to a cluster, we considered that a global localization task was completed. A successful global localization criterion is that the pose difference of the mobile robot between the map and that of the real-world environment is within a small range. The distance threshold is 0.05 m and the angle threshold is 0.5°.

Table 2 shows the statistical results of our proposed method and the AMCL approach. The results show that the conventional AMCL method needs the mobile robot to move a relatively long distance to obtain particle convergence and has a low successful localization rate. The situation did not improve significantly, even though more particles were added. By contrast, our proposed method achieved a high, 98.8%, success rate while the average moving distance was 0.31 m.

4.2 Recovery from robot kidnapping

As shown in Fig. 12, there are several areas that are geometrically similar, for instance, the area near position #6 and the area near position #7, which are inside the red parallelogram dashed line boxes. There are another two areas near position #2 and position #4 which are similar in the opposite direction. These two areas are inside the blue parallelogram dashed line boxes. To perform the kidnapped robot experiment, we placed the mobile robot in the corridor and controlled the robot's moving from the entrance to the toilet position. When the mobile robot moved to position #6 where we placed a marker in advance, we lifted it up and placed it at position #7. We used both our proposed method and the traditional AMCL method to test the pose recovery experiment.

Fig. 15 shows the recovery results of these two different methods. The X-axis is the time interval

Table 2 Statistical results of our proposed method and the AMCL approach

Method	Number of particles	Average number of iterations	Moving distance (m)	Success rate (%)
AMCL	500	69	9.43	47.3
AMCL	500	85	12.27	52.9
Ours	50	6	0.31	98.8

AMCL: adaptive Monte Carlo localization

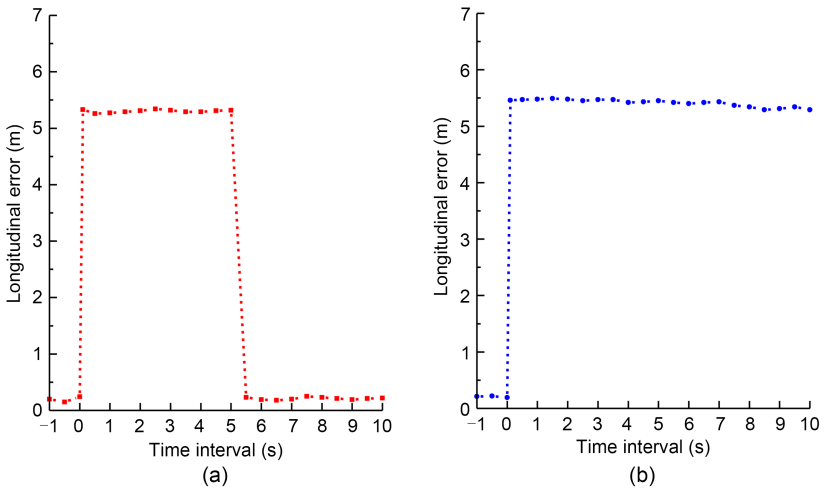

Fig. 15 Recovery from the kidnapped robot problem: (a) result of the proposed method; (b) result of the traditional adaptive Monte Carlo localization (AMCL) method

after the kidnapping event and the *Y*-axis is the longitudinal error. A lateral error was not considered in this experiment because the width of the corridor was limited and the error came mainly from the laser accuracy, not the algorithm. Commonly, the longitudinal error was about 0.19 m. We defined the moment when the robot kidnapping occurred as time 0. When the robot was abducted, the actual longitudinal error came to an average of 5.34 m. The traditional AMCL method could not find the difference, and the error would continue as shown in Fig. 15b. By contrast, our proposed method found the difference after 5 s which was set as a trigger mechanism. Then, the newly captured image was matched with the image database to trigger a new particle initialization process, as described in Algorithm 1. We found that the longitudinal error of the mobile robot returned to about 0.19 m after 5 s. The mobile robot successfully recovered its pose from the kidnapping problem.

5 Conclusions

In this work, we propose a novel approach that uses visual features to assist mobile robot localization in a long corridor. A coarse-to-fine paradigm is used to realize the localization task, and the system is divided into two phases. In the mapping phase, we control the mobile robot moving along the middle line from the entry position to the end of the corridor. A hybrid map consisting of an occupancy grid map and an image database is built in the moving process. The image database includes keyframes which are screened according to an image selection strategy and visual words that are converted from the features of keyframes using the bag-of-words approach. The geo-tagged index between a keyframe and the associated robot's pose is decided by the timestamp. In the localization phase, the mobile robot adjusts its pose to find the best perspective to capture images and match them with the image database. The matching result provides a rough initial position candidate which is a coarse localization. Then, an improved AMCL method is used to realize a fine localization by scattering particles according to the reference position. The results indicate that our proposed method performs well in a corridor environment and can solve both global localization and kidnapped robot problems.

In the future, we will integrate more location areas into our research work, for example, the multiple similar bookshelf areas in the library or the different carriages of a train, and further expand to environments inside the 2D geometrically similar rooms. In addition, more visual features can be extracted from the image to assist robot localization or perception tasks, such as objects and texts.

Contributors

Gengyu GE designed the research. Gengyu GE, Yi ZHANG, and Wei WANG proposed the methods. Gengyu GE, Lihe HU, and Yang WANG conducted the experiments. Gengyu GE processed the data. Wei WANG participated in the visualization. Gengyu GE drafted the paper. Yi ZHANG and Qin JIANG revised and finalized the paper.

Compliance with ethics guidelines

Gengyu GE, Yi ZHANG, Wei WANG, Lihe HU, Yang WANG, and Qin JIANG declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Chen RJ, Yin H, Jiao YM, et al., 2021. Deep samplable observation model for global localization and kidnapping. *IEEE Robot Autom Lett*, 6(2):2296-2303. https://doi.org/10.1109/LRA.2021.3061339

Chen XYL, Läbe T, Nardi L, et al., 2020. Learning an overlap-based observation model for 3D LiDAR localization. Proc IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.4602-4608. https://doi.org/10.1109/IROS45743.2020.9340769

Djosic S, Stojanovic I, Jovanovic M, et al., 2021. Fingerprinting-assisted UWB-based localization technique for complex indoor environments. *Exp Syst Appl*, 167:114188. https://doi.org/10.1016/j.eswa.2020.114188

Fox D, Burgard D, Thrun S, 1999a. Markov localization for mobile robots in dynamic environments. *J Artif Intell Res*, 11:391-427. https://doi.org/10.1613/jair.616

Fox D, Burgard W, Dellaert F, et al., 1999b. Monte Carlo localization: efficient position estimation for mobile robots. Proc 16th National Conf on Artificial Intelligence and 11th Conf on Innovative Applications of Artificial Intelligence, p.343-349.

Ge GY, Zhang Y, Wang W, et al., 2022. Text-MCL: autonomous mobile robot localization in similar environment using text-level semantic information. *Machines*, 10(3):169. https://doi.org/10.3390/machines10030169

Grisetti G, Stachniss C, Burgard W, 2007. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans Robot*, 23(1):34-46.

https://doi.org/10.1109/TRO.2006.889486

Hess W, Kohler D, Rapp H, et al., 2016. Real-time loop closure in 2D LIDAR SLAM. Proc IEEE Int Conf on Robotics and Automation, p.1271-1278. https://doi.org/10.1109/ICRA.2016.7487258

Ito S, Endres F, Kuderer M, et al., 2014. W-RGB-D: floor-plan-based indoor global localization using a depth camera and WiFi. Proc IEEE Int Conf on Robotics and Automation, p.417-422. https://doi.org/10.1109/ICRA.2014.6906890

Katsikis VN, Mourtas SD, Stanimirović PS, et al., 2022. Solving complex-valued time-varying linear matrix equations via QR decomposition with applications to robotic motion tracking and on angle-of-arrival localization. *IEEE Trans Neur Netw Learn Syst*, 33(8):3415-3424. https://doi.org/10.1109/TNNLS.2021.3052896

Kim J, Chung W, 2016. Localization of a mobile robot using a laser range finder in a glass-walled environment. *IEEE Trans Ind Electron*, 63(6):3616-3627. https://doi.org/10.1109/TIE.2016.2523460

Liu X, Zhou BD, Huang PP, et al., 2021. Kalman filter-based data fusion of Wi-Fi RTT and PDR for indoor localization. *IEEE Sens J*, 21(6):8479-8490. https://doi.org/10.1109/JSEN.2021.3050456

Long SB, He X, Yao C, 2021. Scene text detection and recognition: the deep learning era. *Int J Comput Vis*, 129(1):161-184. https://doi.org/10.1007/s11263-020-01369-0

Meng J, Wang ST, Xie YL, et al., 2021. Efficient re-localization of mobile robot using strategy of finding a missing person. *Measurement*, 176:109212. https://doi.org/10.1016/j.measurement.2021.109212

Motroni A, Buffi A, Nepa P, 2021. A survey on indoor vehicle localization through RFID technology. *IEEE Access*, 9: 17921-17942. https://doi.org/10.1109/ACCESS.2021.3052316

Muhammad A, Ali MAH, Turaev S, et al., 2022. Novel algorithm for mobile robot path planning in constrained environment. *Comput Mater Contin*, 71(2):2697-2719. https://doi.org/10.32604/cmc.2022.020873

Mur-Artal R, Montiel JMM, Tardós JD, 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans Robot*, 31(5):1147-1163. https://doi.org/10.1109/TRO.2015.2463671

Naseer T, Burgard W, Stachniss C, 2018. Robust visual localization across seasons. *IEEE Trans Robot*, 34(2):289-302. https://doi.org/10.1109/TRO.2017.2788045

Qian C, Zhang HJ, Tang J, et al., 2019. An orthogonal weighted occupancy likelihood map with IMU-aided laser scan matching for 2D indoor mapping. *Sensors*, 19(7):1742. https://doi.org/10.3390/s19071742

Rublee E, Rabaud V, Konolige K, et al., 2011. ORB: an efficient alternative to SIFT or SURF. Proc IEEE Conf on Computer Vision, p.2564-2571. https://doi.org/10.1109/ICCV.2011.6126544

Thrun S, Fox D, Burgard W, et al., 2001. Robust Monte Carlo localization for mobile robots. *Artif Intell*, 128(1-2):99-141. https://doi.org/10.1016/S0004-3702(01)00069-8

Thrun S, Burgard W, Fox D, 2005. Probabilistic Robotics. MIT Press, Cambridge, USA.

Ullah I, Qian SY, Deng ZX, et al., 2021. Extended Kalman filter-based localization algorithm by edge computing in wireless sensor networks. *Dig Commun Netw*, 7(2):187-195. https://doi.org/10.1016/j.dcan.2020.08.002

Valente M, Joly C, de La Fortelle A, 2019. Evidential SLAM fusing 2D laser scanner and stereo camera. *Unmanned Syst*, 7(3):149-159. https://doi.org/10.1142/S2301385019410012

Wang FS, Zhang JX, Lin BW, et al., 2018. Two stage particle filter for nonlinear Bayesian estimation. *IEEE Access*, 6: 13803-13809. https://doi.org/10.1109/ACCESS.2018.2808922

Wang XV, Wang LH, 2021. A literature survey of the robotic technologies during the COVID-19 pandemic. *J Manuf Syst*, 60:823-836. https://doi.org/10.1016/j.jmsy.2021.02.005

Wu N, Feng LH, Yang AY, 2017. Localization accuracy improvement of a visible light positioning system based on the linear illumination of LED sources. *IEEE Photon J*, 9(5): 7905611. https://doi.org/10.1109/JPHOT.2017.2727643

Xu LC, Feng C, Kamat VR, et al., 2019. An occupancy grid mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments. *Autom Constr*, 104:230-245. https://doi.org/10.1016/j.autcon.2019.04.011

Yousuf S, Kadri MB, 2021. Information fusion of GPS, INS and odometer sensors for improving localization accuracy of mobile robots in indoor and outdoor applications. *Robotica*, 39(2):250-276. https://doi.org/10.1017/S0263574720000351

Zhang L, Chen ZH, Cui W, et al., 2020. WiFi-based indoor robot positioning using deep fuzzy forests. *IEEE Int Things J*, 7(11):10773-10781. https://doi.org/10.1109/JIOT.2020.2986685

Zhao JH, Zhao L, Huang SD, et al., 2020. 2D laser SLAM with general features represented by implicit functions. *IEEE Robot Autom Lett*, 5(3):4329-4336. https://doi.org/10.1109/LRA.2020.2996795

Zhao JH, Li TC, Yang T, et al., 2021. 2D laser SLAM with closed shape features: Fourier series parameterization and submap joining. *IEEE Robot Autom Lett*, 6(2):1527-1534. https://doi.org/10.1109/LRA.2021.3058065

Zhao ZQ, Zheng P, Xu ST, et al., 2019. Object detection with deep learning: a review. *IEEE Trans Neur Netw Learn Syst*, 30(11):3212-3232. https://doi.org/10.1109/TNNLS.2018.2876865

Zimmerman N, Wiesmann L, Guadagnino T, et al., 2022. Robust onboard localization in changing environments exploiting text spotting. https://doi.org/10.48550/arXiv.2203.12647