Robotica (2025), **43**, pp. 3200–3240
doi:10.1017/S0263574725102178

CAMBRIDGE UNIVERSITY PRESS

REVIEW ARTICLE

A review of LiDAR simultaneous localization and mapping techniques for multi-robot

Rui Yuan, Bo Ji, Yuheng Gao and Honghui Tao

School of Computer Science and Artificial Intelligence, Zhengzhou University, Zhengzhou, China
Corresponding author: Rui Yuan; Email: rryuan@outlook.com

Received: 13 July 2024; **Revised:** 10 May 2025; **Accepted:** 18 July 2025; **First published online:** 1 September 2025

Keywords: multi-robot systems; simultaneous localization and mapping; LiDAR; collaborative mapping; map fusion; frontier detection

Abstract

Simultaneous localization and mapping technology is the basis for multi-robot systems to complete navigation, path planning, and autonomous exploration in complex, dynamic, and Global Positioning System (GPS)-denied environments. This paper reviews the current status and progress of multi-robot simultaneous localization and mapping (SLAM) technology based on LiDAR. First, this paper studies the basic principles of LiDAR SLAM. It analyzes the system model construction of LiDAR SLAM, including the mobile robot coordinate system model, kinematic model, sensor model, map presentation, LiDAR SLAM framework, and classic algorithms. Then, this paper discusses the basic framework of collaborative SLAM, analyzes the key issues such as data association, loop closure detection, and global graph optimization in collaborative SLAM, and conducts a detailed literature review on the solutions to key problems in sub-fields of multi-robot SLAM such as frontier detection, task allocation, map fusion, and compares the advantages and disadvantages of various algorithms. Finally, this paper outlines the challenges and future research directions of multi-robot LiDAR SLAM.

1. Introduction

Simultaneous localization and mapping (SLAM) is a technique used by mobile robots equipped with sensors to estimate their motion state and incrementally build a map of the surrounding environment without prior knowledge [1]. SLAM was initially proposed by Smith and Cheeseman [2] in 1988 and has been further improved by researchers such as Leonard and Durrant-Whyte [3]. It has become a cornerstone of mobile robot technology and finds widespread applications in autonomous intelligent driving [4, 5], unmanned aerial vehicles (UAVs) [6], autonomous underwater vehicles (AUVs) [7, 8], three-dimensional reconstruction [9], etc. SLAM is one of the most extensively studied topics in mobile robotics.

The early SLAM algorithms relied mainly on sonar sensors, but their performance was unsatisfactory. Currently, the main sensors used in SLAM for mobile robots include odometry, LiDAR, and depth cameras. Based on the primary sensors used in modern SLAM algorithms, SLAM can be divided into two categories: visual SLAM and LiDAR SLAM. Visual SLAM usually utilizes cameras, which have lower costs and provide rich information. However, it is prone to errors due to lighting variations and fast motion, leading to blurred images [10], making it challenging to directly construct two-dimensional grid maps suitable for localization and navigation [11]. LiDAR SLAM relies predominantly on LiDAR sensors due to their exceptional detection accuracy and high-resolution data. One key advantage of LiDAR is its immunity to lighting conditions, allowing it to capture reliable scan data consistently.

Currently, research on single-robot SLAM has reached a mature stage [12]. However, when it comes to complex tasks and large-scale environments, using a single robot alone poses challenges such as

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

low work efficiency, poor stability, and low map accuracy. Compared with single-robot SLAM, multi-robot SLAM considers the process of multiple robots simultaneously positioning and mapping in the same environment. Collaborative SLAM (CSLAM) is a specific implementation of multi-robot SLAM, which emphasizes information sharing and fusion between robots to achieve collaborative mapping, especially when global optimization is required. It offers advantages such as high efficiency, robustness, and improved map accuracy, making it more suitable for large-scale or sparse environments [13]. However, extending single-robot SLAM to multi-robot SLAM introduces new challenges. These include communication issues between multiple robots [14], efficient frontier detection, collaborative planning [15] and task allocation for robot groups, and concerns regarding the accuracy and efficiency of global fusion when integrating locally constructed maps from individual robots [16].

So far, there have been many surveys and reviews on SLAM and CSLAM. Many of the existing surveys focus on visual SLAM methods [17–20]. Huang et al. [21] and Xu et al. [22] summarized the common LiDAR SLAM frameworks and core module functions and studied multi-sensor fusion technology, but provided inefficient literature reviews and were limited to single-robot LiDAR SLAM. Yue et al. [23] summarized three different types and configurations of LiDAR, but discussed the issues of multi-robot LiDAR systems less. Rizk et al. [24] focus on robot agents and emphasize the challenges in the MRS sub-fields, including task decomposition, coalition formation, task allocation, perception, and multi-agent planning and control. Their work was an excellent and worthwhile review but lacked an in-depth analysis of LiDAR sensors. Refs. [25, 26] reviewed and analyzed the key issues and current challenges in CSLAM, but provided insufficient literature reviews and limited reference for LiDAR SLAM.

Lajoie et al. [27] provided a complete overview of the CSLAM research landscape, detailing the overall framework, popular algorithms, opening issues, and ongoing development trends of CSLAM. However, they haven't mentioned the analysis of LiDAR SLAM exploration, planning, and mapping issues. Zhang et al. [28] conducted a comprehensive review of the scientific connotation, key difficulties, research status, and future development trends of 3D LiDAR but did not consider the task challenges encountered in practical SLAM applications, such as map fusion. Zhang et al. [29] conducted a comprehensive review of the location recognition SLAM methods and related research using LiDAR sensors but only focused on one module of LiDAR scanning. Chen [30] reviewed multi-robot collaborative SLAM from the perspective of multimodal data fusion and discussed existing algorithms from the perspective of machine learning classification.

Different from prior research, our review focuses on the literature on multi-robot cooperative SLAM framework based on LiDAR. Figure 1 shows the overall framework of this study. The main contributions are as follows:

- **Key SLAM Techniques:** It discusses the theoretical basis of LiDAR SLAM technology, SLAM front-end and back-end, and popular 2D/3D LiDAR SLAM algorithms.
- **Multi-robot LiDAR CSLAM Systems:** This paper reviews CSLAM in detail, including the most advanced CSLAM systems and the framework of CSLAM systems, and analyzes the core issues of CSLAM such as data association, loop closure detection, and global optimization.
- **Challenges of MRS Sub-fields:** The study decomposes the multi-robot LiDAR SLAM system into three core sub-fields: frontier detection, task allocation, and map fusion. This paper provides a comparative analysis of the algorithms addressing these issues.
- **Open Challenges and Trends:** Finally, the paper discusses the development challenges and emerging trends in LiDAR SLAM, offering insights into future research directions.

This study provides a valuable reference for researchers aiming to understand various LiDAR SLAM applications. The paper is organized as follows. The theoretical foundations of LiDAR SLAM are briefly reviewed in Section 2. Section 3 focuses on analyzing the system framework of CSLAM, including the system form and key challenges of CSLAM. Then, Section 4 reviews three subfields of

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

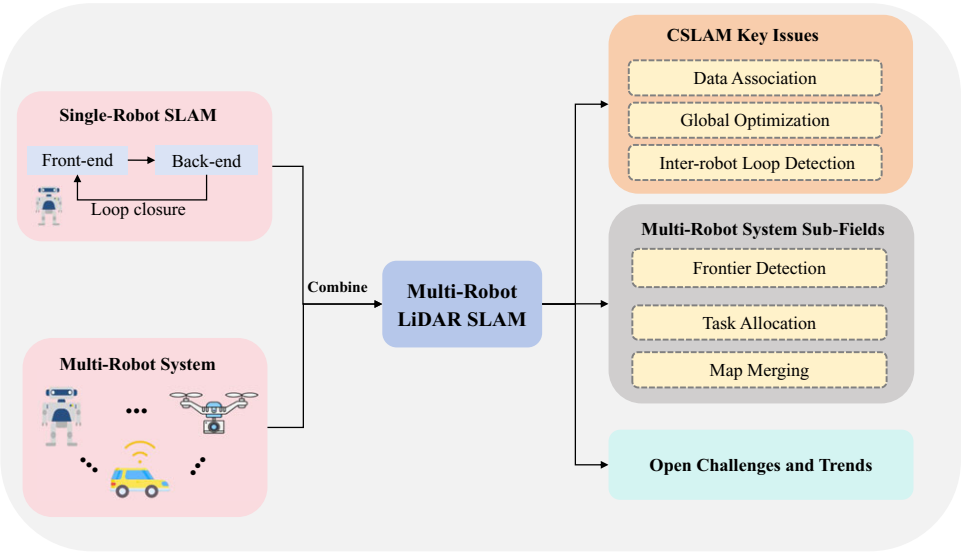


Figure 1. *Multi-robot LiDAR SLAM structure diagram.*

multi-robot LiDAR systems: frontier detection, task allocation, and map fusion, as well as existing methods to address these challenges. Therefore, in Section 5, several open challenges and future development trends of multi-robot LiDAR SLAM are pointed out. Finally, this paper summarizes the conclusions in Section 6.

2. Key SLAM techniques

Before formally introducing the LiDAR SLAM system, we first review the relevant knowledge of building the LiDAR SLAM system model to provide some reference for beginners.

2.1. Kinematic model of mobile robot

The kinematic model of a mobile robot is the basis for estimating the robot's posture. Mobile robot systems commonly employ Cartesian coordinate systems and polar coordinate systems to represent the relative positions of robots and obstacles [31]. When calculating the robot's position and orientation, the Cartesian coordinate system is simpler and more intuitive. Therefore, the model construction in this chapter adopts the Cartesian coordinate system.

Figure 2 depicts the coordinate system utilized by the mobile robot. The world coordinate system, denoted as *XwOwYw*, acts as a reference frame for measuring the displacement of all objects within the system. In the world coordinate system, the robot's pose at time k is $X_k = (x_r, y_r, \theta_r)$. Let the robot coordinate system be *XrOrYr*. This coordinate system moves with the movement of the robot. The coordinate of its origin in the world coordinate system represents the robot's pose, and the angle between the horizontal axis and the horizontal axis of the world coordinate system represents the posture of the robot [32].

If a point $X_0 = [x_0, y_0]^T$ in the robot coordinate system, then the coordinates of this point $X_w^0 = [x_w^0, y_w^0]^T$ in the world coordinate system can be obtained by (1):

$$\begin{cases} x_W^0 = x_r + x_0 \cos\theta_r - y_0 \sin\theta_r \\ y_W^0 = y_r + x_0 \sin\theta_r + y_0 \cos\theta_r \end{cases} \tag{1}$$

The motion of a mobile robot occurs in a three-dimensional environment, with the coordinate system centered at the robot's chassis center point. The X-axis represents the forward direction, the Y-axis is

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

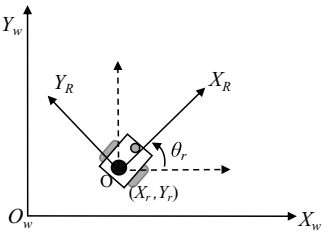


Figure 2. *Mobile robot coordinate system.*

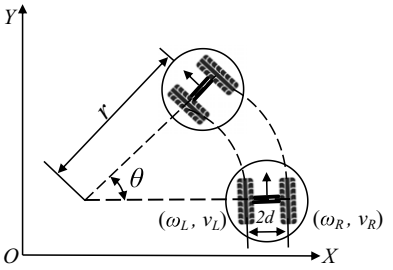


Figure 3. *Mobile robot kinematic model.*

perpendicular to the forward direction, and the Z-axis passes through the center of the chassis and is perpendicular to the robot's base. Simultaneously, it is necessary to simplify the robot model appropriately by considering the robot's motion in a two-dimensional plane. Therefore, the three-dimensional coordinate system can be simplified to a two-dimensional XOY coordinate system, and the robot's pose representation consists of the two-dimensional plane coordinates and the steering angle.

Due to its simple structure and easy control, the two-wheel differential motion structure is widely used in wheeled mobile robots. The robot moves in a circular arc around a certain center, as shown in Figure 3.

The calculation formulas for angular velocity and linear velocity are as follows:

$$\omega = \frac{v_R - v_L}{2d} \tag{2}$$

$$v = \frac{v_R + v_L}{2} \tag{3}$$

Where v and ω represent the linear velocity and angular velocity of the robot during movement, v_L and v_R are the linear velocities of the left and right wheels, and ω_L and ω_R are the angular velocities of the left and right wheels, respectively. r represents the radius of rotation around the center of the circle, and the distance between the two wheels is $2d$.

From the above formula (3), we can get that in the kinematic model of the mobile robot, the linear velocity and angular velocity of the robot's motion are only related to the linear velocity of the robot's left and right wheels, and the wheel spacing. And half of the wheel spacing d is a constant, which has been set when the robot leaves the factory. Therefore, the robot's motion can be calculated by measuring v_L and v_R through the encoders on the wheels.

2.2. Sensor model

2.2.1. Odometer model

The odometer calculates the distance traveled by the mobile robot per unit of time by using the pulse data output by the photoelectric encoder within a certain period to estimate the robot's movement over time. The odometer can mitigate many errors arising from uncertainties in the robot's navigation process and enhance the accuracy of map construction. The two-wheel differential chassis has a simple structure,

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

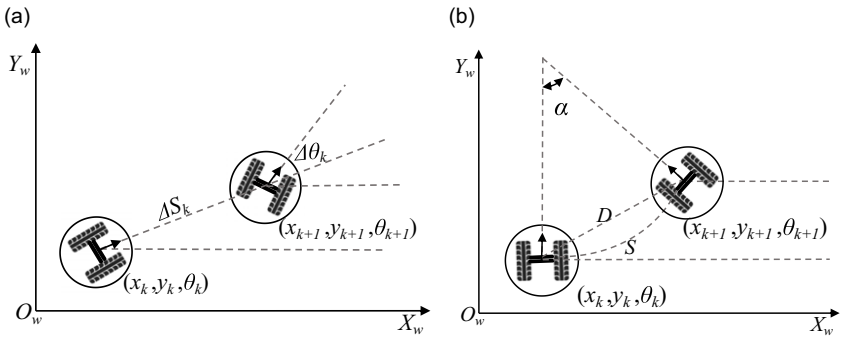


Figure 4. *Mobile robot odometer linear model and odometer arc model.*

low cost, and a relatively simple model to establish, so it is used in various types of robot production [33]. Figure 4 shows two models of odometer: the straight line model and the arc model.

- **Straight Line Model**. The motion of the robot is continuous, but it is usually considered to be discrete in time to simplify the calculation process [34]. The straight line model is represented in Figure 4(a). We can know that the dead reckoning Eq. (4) for the robot's linear motion is:

$$\begin{cases} x_{k+1} = x_k + \Delta S_k \cos\theta_k \\ y_{k+1} = y_k + \Delta S_k \sin\theta_k \\ \theta_{k+1} = \theta_k + \Delta\theta_k \end{cases} \tag{4}$$

  Where(x_k, y_k, θ_k) represents the robot coordinates, ΔS_k, $\Delta\theta_k$ represents the robot's linear motion distance and rotation angle. Substituting the angular velocity and linear velocity ω, v of the robot's motion, the robot's posture at the next moment is:

$$\begin{cases} x_{k+1} = x_k + v\Delta t \cos\theta_k \\ y_{k+1} = y_k + v\Delta t \sin\theta_k \\ \theta_{k+1} = \theta_k + \omega\Delta t \end{cases} \tag{5}$$

- **Arc model**. The robot rotates around a point in an arc and moves to the next position. This model considers both the movement of the robot and the change of the direction angle [35], which is shown in Figure 4(b). The dead reckoning equation for the robot's circular motion is (6):

$$\begin{cases} x_{k+1} = x_k + D\cos\left(\theta_k + \frac{\alpha}{2}\right) \\ y_{k+1} = y_k + D\sin\left(\theta_k + \frac{\alpha}{2}\right) \\ \theta_{k+1} = \theta_k + \alpha \end{cases} \tag{6}$$

  Where α represents the angle that the robot rotates in time Δt and S represents the arc of rotation, which is approximately the displacement D between two points. Substituting the angular velocity ω and linear velocity v into the robot's motion equation, the robot's posture changes to:

$$\begin{cases} x_{k+1} = x_k + v\Delta t\cos\left(\theta_k + \frac{\omega\Delta t}{2}\right) \\ y_{k+1} = y_k + v\Delta t\sin\left(\theta_k + \frac{\omega\Delta t}{2}\right) \\ \theta_{k+1} = \theta_k + \omega\Delta t \end{cases} \tag{7}$$

  Based on the formula (7) provided above, the future posture of the robot can be calculated by considering its current posture, velocity, and elapsed time.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

2.2.2. IMU model

An inertial measurement unit (IMU) is a precision sensor module consisting of an accelerometer and a gyroscope that is used to capture inertial data, including the linear acceleration and angular velocity of an object. This technology is widely used in various fields such as aerospace engineering, robotic systems, and automotive navigation. In contemporary SLAM systems, the integration of LiDAR and IMU technology is becoming more and more common [36–39].

Using the measurement data and kinematic equations of the IMU, the position and attitude of the robot can be determined. However, there are noise and drift errors when the robot moves, and the raw data often leads to inaccurate pose estimates. The measurement formulas of the angular velocity Wt and linear velocity at of the IMU are as follows:

$$\hat{a}_t = R_t^{BW}(a_t + g) + b_a + n_t^a \tag{8}$$

$$\hat{w}_t = w_t + b_g + n_t^g \tag{9}$$

Where n_t^a and n_t^g represent the measurement noise of the accelerometer and gyroscope at time t, respectively, b_a and b_g represent the deviations of the accelerometer and gyroscope, respectively, which can be considered as constant deviations in a short period of time. g represents the gravitational acceleration of the current coordinate system, R_t^{BW} indicates the attitude of system B relative to system W at the t moment.

IMU error models can be categorized into two primary types: deterministic errors and stochastic errors during operation. Deterministic errors, including bias and scale factor, can be effectively compensated through pre-calibration procedures. In contrast, stochastic errors are typically modeled under the assumption of Gaussian distribution characteristics, encompassing phenomena such as Gaussian white noise and biased random walk [40]. From a performance perspective, IMUs can be classified into three distinct tiers: (1) high-precision marine and aerospace-grade inertial navigation systems, which are characterized by their large size, exceptional accuracy, and substantial cost; (2) tactical and industrial-grade inertial sensors, which offer significant application value and are widely employed in scenarios requiring high-performance measurements at relatively moderate costs, such as in delivery robots and UAVs; (3) consumer-grade sensors designed for mass-market applications, prioritizing cost-effectiveness over precision.

2.2.3. LiDAR model

In mobile robot SLAM, the availability of accurate and reliable sensor data is crucial for constructing precise maps. Various sensors are commonly employed, including IMU, wheel encoders, sonar, LiDAR, and depth cameras [41]. Distance sensors like sonar and LiDAR actively detect objects by transmitting and receiving signals, allowing for effective measurement of the distance between the robot and obstacles. However, sonar sensors are susceptible to environmental noise, which can introduce disturbances into the data readings. On the other hand, visual sensors such as depth cameras are significantly influenced by ambient light conditions, posing challenges for their operation in both dimly and brightly lit environments.

LiDAR uses the time difference between the laser beam emitted and the laser reflected by the obstacle to calculate the distance between the obstacle and itself. The collected object information presents a series of scattered points with accurate angle and distance information, which is called a point cloud. Compared with other sensors, LiDAR has the characteristics of high accuracy, fast solution speed, and no restrictions on light environment. It finds extensive applications in autonomous driving, mapping, positioning, and navigation domains [42]. Figure 5 is divided into the device appearance of the laser (a) and the scanning model (b).

There are two main types of LiDAR ranging models: triangulation and ToF (Time of Flight). Triangulation obtains distance information by solving triangles based on the camera's spot imaging

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

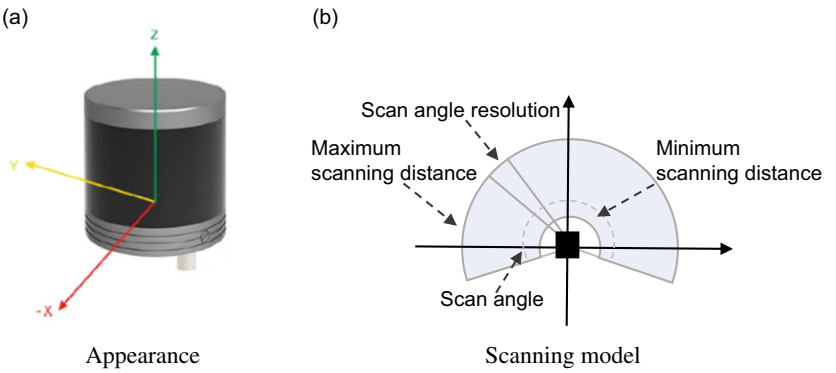


Figure 5. *Appearance and scanning model of LiDAR.*

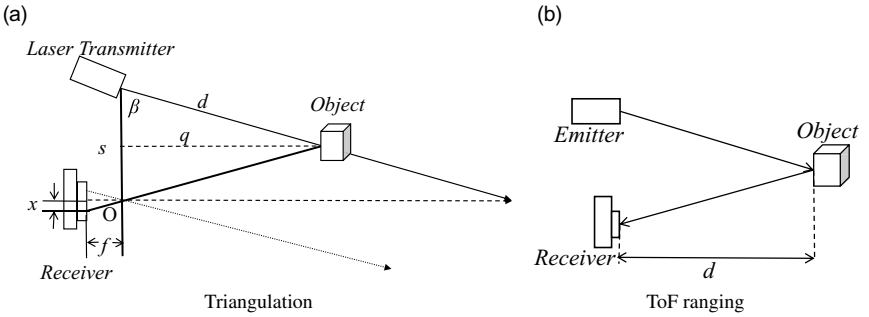


Figure 6. *Two model of LiDAR: triangulation and ToF.*

position, while ToF calculates distance based on the measured light's flight time, which has a higher accuracy [43].

In the triangulation distance measurement of Figure 6(a), the distance to the object is proportional to the angle of the reflected light. The concept of the triangle equation can be used to estimate the actual distance [44]. There is a distance s between the transmitter and the center o of the receiver of the LiDAR using the triangulation distance measurement method, which is called the baseline. The distance from the object to the baseline is q. The transmitter emits a laser beam at a certain angle β, which hits an object at a distance d and is received by the receiver after reflection. The focal length of the receiver is f, and the distance from the receiving point to the center o of the receiver is x. The LiDAR measures the distance d of an object using the triangulation formula, as depicted in calculation formula (10).

$$d = \frac{fs}{2x \sin \beta} \tag{10}$$

Figure 6 (b) shows the working principle of ToF ranging. A high-frequency laser beam of a specific power is emitted to the target object, and the laser signal reflected to the receiver is processed with the emission signal. The target position information is obtained by calculating the product of the time difference between the emitted laser and the reflected laser and the speed of light [45]. Compared with triangulation-ranging LiDAR, ToF LiDAR has higher accuracy and stability. ToF calculates the distance d to the object based on the light flight time. The calculation formula is shown in (11). Where d is the distance to the object, c is the speed of light, Δt is the time difference between the laser emission and reception, and n is the average reflectivity of the transmission medium.

$$d = \frac{c\Delta t}{2n} \tag{11}$$

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

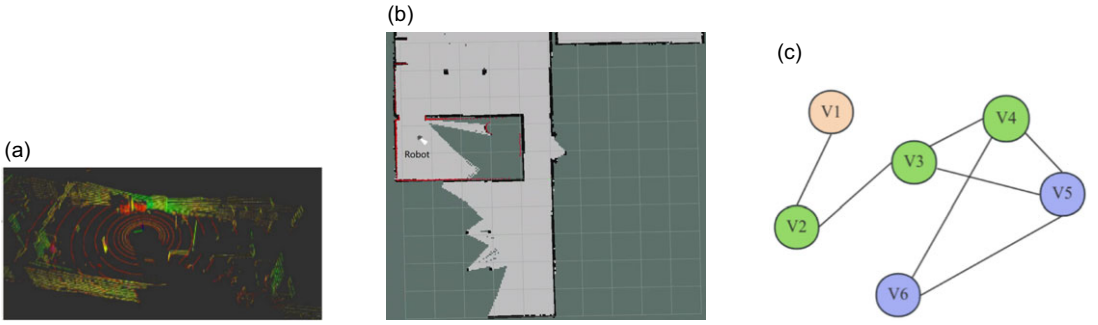


Figure 7. *Representation of 3D point cloud map (a), occupancy grid map (b), and topological map (c).*

2.3. Map representation

After obtaining data through sensors such as LiDAR, the raw data needs to be converted into a visual data representation, namely a map representation, which is a structure used to store and express environmental information. Commonly used map representation methods can be roughly divided into three types: metric map, topological map, and metric semantic map. Each map representation method has its advantages and disadvantages. The following is a brief introduction to these different map types.

2.3.1. Metric map

Metric maps are the most commonly used map representations in SLAM, with real physical dimensions. Typical representatives include the Occupancy grid Map, Point cloud Map, and Feature Map [46]. The point cloud map [47] is shown in Figure 7(a). The map created using the LiDAR sensor is represented as a grid map, as shown in Figure 7(b). The occupation grid map was first proposed by Elfs [48] and Morave [46] in the late 1980s, and its basic idea is to use different pixel values to represent the occupation state of the corresponding area, that is, the existence of obstacles.

In general, a two-dimensional raster map represents a specific height slice of information within a three-dimensional space in a particular state. The grid cells in a raster map can be categorized into three states: occupied, vacant, and unknown. Raster maps provide comprehensive spatial distribution and scale information, serving as the foundation for multi-robot exploration map fusion and finding applications in various SLAM frameworks [49]. A feature map is an environmental map constructed by a robot to extract distinctive features from the surrounding environment. It is primarily employed in indoor scenarios, utilizing geometrically prominent elements such as tables, chairs, doors, and windows to represent the environment. Scale maps find primary applications in map construction, localization, and small-scale path planning tasks.

2.3.2. Topological map

Topological maps are widely used in SLAM, consisting of nodes and edges that represent the connected relationships between different locations. Unlike physical maps, topological maps lack real-world dimensions and focus solely on spatial connections. The structure of a topological map, as depicted in Figure 7(c) [50], simplifies the requirement for precise details, resulting in a more compact representation. The benefits of topological maps include high efficiency in autonomous routing, fast data processing, and reduced memory consumption for mobile robots. Consequently, they are frequently employed in large-scale robot path planning.

2.3.3. Metric-semantic map

Metric-semantic map is a map representation method combined with SLAM and deep learning that correlates semantic information with traditional metric maps [51]. It adds labels based on scale maps and

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

is often used for human-computer interaction [52] to capture the state and shape of objects in the environment for description and to build environmental maps. Qin [53] proposed a lightweight positioning scheme that relied on low-cost cameras and compact visual semantic maps, and its semantic elements included lane lines and walkways. The metric semantic map has become one of the development prospects of SLAM research in the future.

Different maps have advantages and disadvantages, so using a combination of maps is an optional solution. Zhang et al. [54] constructed a metric semantic map, which removes unrelated redundant points compared with traditional maps and is convenient for robots to locate targets, which is suitable for object search. Kuang et al. [55] improved the initial particle swarm distribution in the global positioning of the adaptive Monte Carlo localization algorithm based on the semantic grid map. The semantic grid map solved the problem of environmental information degradation caused by 2D LiDAR and improved the robot's perception of the environment.

2.4. LiDAR SLAM framework

In an unknown environment, the problem of SLAM can be defined as follows: the mobile robot relies exclusively on its onboard sensors to achieve self-localization while concurrently constructing a map of the environment. The self-localization of the robot (pose estimation) and the construction of the unknown environment map are mutually dependent. Assuming that the mobile robot is equipped with sensors and is navigating within the unknown environment, and taking into consideration that the sensor measurements are discrete due to sampling at discrete time intervals, the robot's motion can be mathematically described using discrete time steps, denoted as $t = 1, \ldots, T$. The sensors integrated into the robot's system include LiDAR, odometry, and IMU. The relationship between the robot's sensor measurements and its motion can be expressed using motion (12):

$$x_T = f\left(x_{T-1}, u_T, w_T\right) \tag{12}$$

Among them, x_T represents the position of the mobile robot at a certain moment, the random variable sequence $x_{1:T} = \{x_1, \ldots, x_T\}$ is the motion trajectory of the mobile robot in an unknown environment; u_T represents the reading of the motion sensor it carries, such as acceleration, angular velocity, and other parameters; w_T represents the motion noise in the process, that is, the deviation between the actual motion process and the issued motion command.

The map constructed by the mobile robot is composed of different landmarks. $z_{T,j}$ is the observation data generated by the mobile robot observing the landmark y_j through the sensors it carries at the location x_T. The process is described by an abstract function h. The noise in the observation process is $v_{T,j}$, then the observation (13) can be obtained:

$$z_{T,j} = h\left(y_j, x_T, v_{T,j}\right) \tag{13}$$

Single-robot SLAM is the basis of the multi-robot SLAM. Figure 8 provides a common framework of single robot LiDAR SLAM framework, which includes two components: front-end and back-end [56]. Sensor data mainly comes from LiDAR and wheel odometers. The LiDAR obtains environmental information through laser scanning. The wheel odometer continuously obtains the robot's mileage information through the encoder of the mobile chassis [57]. The SLAM front-end collects data through scanning measurement, and the back-end performs state estimation and map optimization.

2.4.1. SLAM front-end

Front-end matching is a common data association method in LiDAR SLAM, where a mobile device reads a previous position and uses the current position to measure the relationships between neighboring devices.

The accuracy and computational efficiency of the scan matching algorithm directly affect the accuracy of SLAM algorithm position estimation and map construction, as well as providing back-end optimization capabilities. Theoretically, it can be divided into iterative closest point (ICP)-based

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

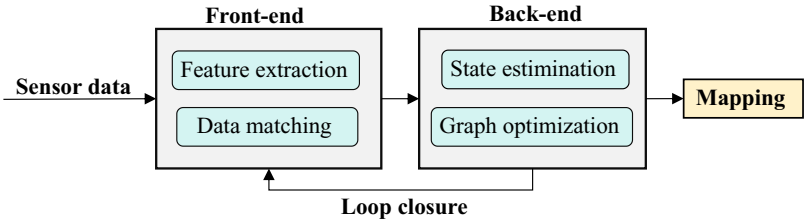


Figure 8. *Single-robot LiDAR SLAM framework.*

scanning, feature-based scanning, mathematical property-based scanning, correlation-based scanning, optimization-based scanning [58], etc.

The iterative closest point algorithm was proposed by Besl [59] and Chen [60] in 1992. The essence of scanning matching is to calculate the translation matrix and rotation matrix with the smallest Euclidean distance between the corresponding point pairs between two point cloud frames and then use this translation matrix and rotation matrix as the pose change relationship between the two point cloud frames to be matched through iteration [61]. However, this algorithm is prone to fall into local optimality. In the improved algorithm for solving the optimal pose, Low [62] and others proposed an improved solution to expand the distance from point to point to point to line or even point to surface [63]. In addition, Nicolai [64] and others combined the three-dimensional point cloud with the CNN network for training, which improved the real-time performance.

The feature-based scan matching algorithm removes key elements from the laser scan data through matching, such as point, line, surface, and other set primitives or combinations, features such as curvature and normal vector, and some custom feature descriptors to achieve fast correspondence matching. It can solve the pose change without providing initial values. Zhang [65] proposed feature points for scan matching using point-to-boundary and point-to-plane methods, and the effect of the LiDAR odometer is relatively accurate.

The mathematical property-based scan matching algorithm converts the point cloud registration problem into a mathematical formulation. A notable example of such an algorithm is the distribution-based Normal Distribution Transform (NDT) [66]. NDT is a mathematical point cloud registration algorithm that utilizes Gaussian distribution to model the captured point cloud data [67]. Compared with ICP and feature-based algorithms, NDT exhibits fast speed, short calculation time, and the ability to compensate for a wider range of initial poses [68].

The correlation scan matching method (CSM) conducts point cloud matching utilizing a probabilistic grid map. Each grid retains a logarithmic probability of occupancy. In the case of a new laser scan frame, all point estimations are projected onto the grid map. The current pose score, which represents the credibility of the pose, is determined by summing the logarithmic probabilities of the grid occupied by the laser. Nevertheless, the brute force matching approach is slow, leading to the proposal of a branch and bound acceleration method that subdivides the environmental resolution to compute the optimal pose. Du et al. [69] proposed a localization method for indoor mobile robots based on the CSM technique, which uses 2D LiDAR to provide accurate and fast pose estimation based on a public space map.

The optimization-based scan matching method is essentially a gradient-based method and solution. Hector-SLAM and Cartographer are two representative optimization-based algorithms, among which Hector-SLAM is a pure map optimization method, while Cartographer is an optimization method that combines CSM and gradient optimization.

2.4.2. SLAM back-end

Back-end optimization is an effective method to reduce the accumulated local errors in front-end matching. In principle, it can be mainly divided into filtering-based optimization methods and graph optimization-based methods [70]. For filtering-based LiDAR SLAM, the front-end estimates the pose

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Table I. *Filter forms.*

Filter form	Description	Characteristic
KF	Optimal estimation for recursive linear Gaussian systems	Only applicable to linear system
EKF	Linearization of nonlinear and non-Gaussian systems	High computational complexity, poor robustness to model uncertainty
UKF	Nonlinear transformation of sigma points followed by Gaussian distribution approximation	Higher estimation accuracy compared to EKF
IF	Representation of Gaussian function using canonical parameters (information matrix and information vector)	Dual to Kalman filter
PF	Approximation of state distribution using particles as sampling points based on Monte Carlo methods	Particle depletion issue
EM	A probabilistic model between the state to be estimated and the observable variables is established	When data is missing or the noise is large, state estimation is achieved through probabilistic reasoning

of the mobile robot through point cloud data registration, and the back-end optimizes the pose calculated by the front-end using filtering theory [71]. Representative filtering methods include the Kalman filter (KF), extended KF (EKF), unscented KF (UKF), information filter (IF), Particle Filter (PF), and expectation maximization. The filter forms are shown in Table I.

The more classic one is the EKF method, which solves the problem of nonlinear non-Gaussian application scenarios and has been applied to LiDAR CLSAM. The EKF-based SLAM algorithm will initialize the map, update the current robot posture and state according to the odometer data and kinematic model, initialize the road sign, correct the current state through the road sign, and update the new roadside information. The disadvantage of the EKF method is that it needs to store the mean and variance of all states, and the computational complexity is high. The UKF method is a combination of the unscented transform (UT) and the standard KF system, using UT to handle the nonlinear transfer of the mean and covariance. The PF method uses the recursive filter of the Monte Carlo method to generate particles, uses the system model to predict the state, updates the particle weights, resamples and discards particles with smaller weights, and finally calculates the estimated value of the state variable. The IF method uses the information matrix to represent and update the state estimate and its uncertainty.

Compared to filtering-based SLAM solutions, graph optimization-based approaches can effectively adapt to larger and more complex scenarios. Graph optimization involves expressing a conventional optimization problem in the form of a graph theory, where the robot's posture is represented as a node and the relationships between postures form the edges. Once these nodes and edges are established, the robot's posture is adjusted to satisfy the imposed constraints [72]. Factor graph optimization is a specific method within graph optimization that partitions the graph into variable nodes and factor nodes. In the case of a long path, check whether it has returned to the previous position. If a closed loop is detected, add corresponding constraints to improve the global consistency of the map. After the optimization is completed, update the robot's posture and the position of the map features.

In an unknown environment, for a highly uncertain robot, it is worth considering whether to continue exploring the unknown area or revisit the already explored region after completing exploration in a partial area. This situation poses the "exploration-exploitation dilemma" in frontier detection. Grabowski [73] observed that regions of interest where sensor readings overlap may provide more informative data than unexplored areas. In the SLAM mapping process, the front-end odometry only considers keyframes

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Table II. *Comparison of LiDAR SLAM methods.*

Methods	Year	Sensors	Algorithms	Applicable scenarios	Innovation
EKF-SLAM	1990	2D LiDAR	Filtering	Low-dimensional SLAM, 2D mapping	The EKF was applied to 2D LiDAR SLAM for the first time
Gmapping [75]	2007	2D LiDAR	Filtering	2D Indoor SLAM	Separate robot path and map estimation, and use Grid Mapping for probabilistic grid mapping
Karto [77]	2010	2D LiDAR	Graph Optimization	2D Indoor SLAM	Back-end optimization with loop-back detection was introduced for the first time in laser SLAM
Hector [76]	2011	2D LiDAR	Filtering	UAV, Indoor SLAM	The 2D SLAM system was combined with 3D scan-matching technology and an inertial sensing system
LOAM [224]	2014	2D LiDAR	Graph Optimization	3D Mapping	Good real-time performance, constant velocity motion assumption, no closed-loop detection
Cartographer [78]	2016	2D/3D LiDAR	Filtering, Graph Optimization	2D & 3D SLAM	Using a submap and closed-loop, suitable for large indoor and outdoor scenes
LeGO-LOAM [80]	2018	3D LiDAR, IMU	Graph Optimization	Low-cost 3D SLAM	Compared to LOAM, back-end optimization is added to make the diagram more complete
LIO-SAM [82]	2020	3D LiDAR, IMU, GPS	Graph Optimization	Autonomous driving, UAV	Multiple sensors are tightly coupled to generate precise environmental maps
FAST-LIO2 [84]	2022	3D LiDAR	Filtering	High-speed SLAM	Based on ESKF, good real-time performance
LOCUS 2.0 [85]	2022	3D LiDAR	Filtering	Extreme Environment SLAM	NASA mission, Factor Graph Optimization

from adjacent periods, leading to the accumulation of errors over time and resulting in unreliable long-term estimation results. Loop closure is employed to detect potential loops based on the constraint relationship between point cloud data frames. By correcting drift errors, globally consistent trajectories, and maps are constructed, enabling global optimization. From an algorithmic perspective, loop closure can be broadly categorized as scan-to-scan, scan-to-map, or map-to-map loop closure.

2.5. LiDAR SLAM algorithms

The LiDAR-centric solution is one of the mainstream solutions for SLAM [74]. According to the number of laser beams in a LiDAR sensor, it can be divided into single-line LiDAR (2D LiDAR) and multi-line LiDAR (3D LiDAR). In this section, we will conduct a comprehensive literature review on two LiDAR SLAM algorithms: (1) 2D LiDAR SLAM, and (2) 3D LiDAR SLAM. Table II summarized classical LiDAR SLAM algorithms.

2.5.1. 2D LiDAR algorithms

The SLAM algorithm that utilizes a single-line radar to create a two-dimensional map is known as 2D LiDAR SLAM. 2D LiDAR can be divided into two types: filter-based and scan-matching-based methods. Common algorithms for 2D LiDAR SLAM include Gmapping [75], Hector [76], Karto [77], and Cartographer [78]. Typically, both the data and movement are confined to a 2D plane, which is parallel to the laser scanning plane.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Gmapping. Gmapping is one of the most representative 2D LiDAR SLAM systems based on particle filters and is widely used for constructing two-dimensional grid maps. It employs a particle filter framework that integrates the robot's motion model with sensor observation data to create a globally consistent map.

Since the Gmapping method only supports simple loop detection, cumulative errors may occur in large and complex environments, resulting in map-building failure. In addition, in the Gmapping algorithm, each particle will build a map, which requires a lot of memory and is therefore not suitable for large scenes.

Hector. Hector is a 2D LiDAR SLAM algorithm that combines a robust scan-matching method with navigation technology using an inertial sensor system. The algorithm framework is based on Gauss-Newton technology, which has high requirements for sensors, requires the use of laser scanners with high update frequency and low measurement noise, and does not require an odometer. It is suitable for aerial drones and ground cars driving in uneven areas.

Karto. Karto is a 2D SLAM algorithm based on graph optimization, which uses highly optimized and non-iterative Cholesky matrices for sparse system decoupling as a solution. The graph optimization method uses the mean of the graph to represent the map. Each node represents a position point of the robot trajectory and a sensor measurement data set. The connection pointed by the arrow represents the movement of continuous robot position points. When each new node is added, the map will be calculated and updated according to the constraints of the node arrows in space. The more landmarks there are, the greater the memory requirements. However, the graph optimization method has greater advantages in mapping in a large environment than other methods.

Cartographer. Cartographer is an open-source mapping algorithm that combines 2D and 3D LiDAR, landmarks, odometry, and IMU measurements [79]. It adopts a laser slam framework based on graph optimization, proposes the definition of submap, and uses the branch and bound method to solve the problem of submap construction and global map matching, achieving loop closure detection and better global optimization.

The core of the entire algorithm is divided into two parts: the first part is the introduction of submap, also known as local map, which builds a large-scale submap in a short time through laser scan, and inserts the new scan-to-scan frame into the best position of the submap, and convert the scan-to-scan matching into scan-to-map matching. The second part introduces the concept of a global map. The error in the submap created over a short period is minimal, but over a longer duration, cumulative errors can arise. At this point, loop closure detection is necessary to eliminate these cumulative errors. When a submap is built, loop closure detection is performed through loop closure, and no new laser scan is inserted into the current submap. The current Cartographer has been improved to support 3D SLAM mapping.

2.5.2. 3D LiDAR algorithms

2D SLAM cannot be applied to indoor scenes with large dynamic scene changes, let alone complex outdoor environments. Visual SLAM is highly dependent on light and cannot be used in dark places. 3D LiDAR SLAM is applicable both indoors and outdoors and is not affected by the light environment.

3D LiDAR SLAM mainly targets multi-line laser radars, such as Laser Odometry and Mapping (LOAM), Lightweight and Ground-Optimized LiDAR Odometry and Mapping (LeGO-LOAM) [80], LiDAR Inertial Odometry via Smoothing and Mapping (LIO-SAM), Fast Direct LiDAR-Inertial Odometry (FAST-LIO2), LUCUS 2.0 etc. It performs positioning through feature point matching of 3D data and has a high degree of restoration of the 3D mapping environment [81]. The following will review several state-of-the-art 3D LiDAR algorithms.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

LOAM. LOAM is a real-time method that can simultaneously estimate odometry and create maps using 3D LiDAR. Solving SLAM tasks based on the LOAM algorithm can be decomposed into three steps: (1) Feature detection: LOAM extracts edge and plane features of the current frame and the matching frame based on local smoothness, and matches the points in the current point cloud with the edges and planes in the map. (2) Odometer measurement: Calculate odometer information. 3)Map construction: LOAM does not include a loop detection module, which means it cannot revisit previously accessed locations. However, accurate map construction can still be obtained through feature detection and feature point matching, in conjunction with the calculated odometer information.

LeGO-LOAM. LeGO-LOAM is lightweight as real-time pose estimation and mapping can be implemented on embedded systems [80]. Point cloud segmentation is applied to filter out noise, and feature extraction is performed to obtain unique plane and edge features. The two-step Levenberg-Marquardt optimization method uses plane and edge features to solve different components of the six-degree-of-freedom transformation across continuous scans. The algorithm pays special attention to the extraction and utilization of ground points, can effectively handle uneven ground and complex environments, and has strong adaptability.

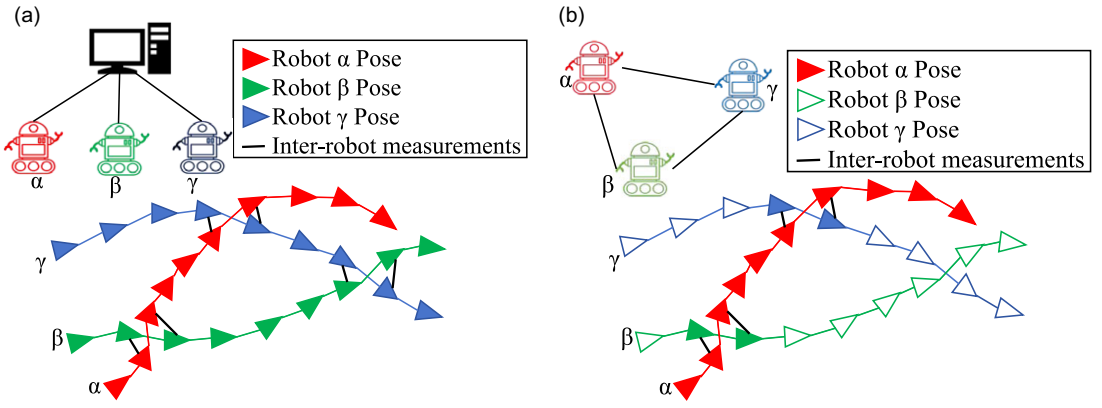
LIO-SAM. Based on LeGO-LOAM, Shan proposed LIO-SAM, which can generate high-precision three-dimensional maps and provide rich environmental information by tightly coupling LiDAR and IMU [82]. LIO-SAM fuses LiDAR and IMU data, using the high-frequency dynamic information provided by the IMU to improve the low-frequency updates of the LiDAR. LIO-SAM formulates LiDAR-Inertial Odometry on a factor graph, allowing a large number of relative and absolute measurements (including loop closures) from different sources to be factored into the system. The motion estimated by the IMU pre-integration removes the tilt of the point cloud and provides an initial guess for LiDAR odometry optimization.

FAST-LIO2. FAST-LIO2 is based on the FAST-LIO algorithm and is a LiDAR-inertial odometry framework [83] that uses a tightly coupled iterative KF to fuse LiDAR feature points with IMU measurements. In this approach, planar and edge features are extracted from the LiDAR point cloud, and in the next step, these features are used together with IMU measurements for state estimation. The feature points are then matched to the global frame using the estimated pose and the global map is updated. Xu et al. [84] developed and validated an incremental k-d tree *(ikd-Tree)* data structure that supports dynamic insertion, deletion, and parallel reconstruction of dynamic points. At the same time, direct matching of point clouds with maps is achieved based on this data structure. The scan-to-map matching method improves the accuracy of pose estimation while reducing the amount of computation. It supports incremental map updates and efficient KNN queries at each step.

Compared to tightly coupled LiDAR-Inertial methods using feature points, the FAST-LIO2 is lighter and achieves higher mapping rates and odometry accuracy. Another advantage of FAST-LIO2 is that it can naturally adapt to different LiDARs since feature extraction is removed. This is because feature extraction needs to be carefully designed for different LiDARs according to their respective scanning patterns and densities.

LOCUS 2.0. LOCUS 2.0 [85]: A robust and computationally efficient lidar odometry system for real-time subsurface 3D mapping. It includes: a new normal-based generalized iterative closest point (GICP) formulation that reduces the computation time of point cloud alignment; an adaptive voxel grid filter that maintains the required computational load regardless of the environment geometry; and a sliding window mapping method that limits memory consumption. The system is able to achieve accurate and real-time ego-motion estimation under challenging perception conditions through large-scale exploration under severe computational and memory constraints.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press



Centralized multi-robot system. Decentralized multi-robot system.

Figure 9. *Centralized and decentralized CSLAM approach. The centralized CSLAM estimates a global map and trajectories, while the decentralized CSLAM estimates a globally consistent local map (from the perspective of robot α).*

3. Multi-robot LiDAR SLAM system

Multi-robot SLAM, as known as CSLAM, involves multiple, collaborating homogenous or heterogenous robots operating simultaneously in the same unknown environment but at different locations. Each robot uses its sensor perception to analyze, integrate, and fuse data from diverse information sources through robot-to-robot interactions. This process aims to construct a global map of the environment [16].

When the initial states of all robots are known or can be estimated, the CSLAM problem is a simple extension of the single-robot SLAM problem, which includes the states of all robots, measurements, and additional inter-robot measurements that connect the maps of different robots. Assume a robot group $(X_1 \cdots X_n)$, where X_i is the position of the i-th mobile robot at a certain moment, and Z_i is the observation data collected by the i-th robot, which can be the measurement result from the sensor. $Z_{1\ldots n}$ represents the relative measurement between a group of robots, including the relative position estimation of the robot's position in each trajectory. At this time, the definition formula of CSLAM can be expressed as:

$$\begin{aligned}\left(X_1^*, \ldots, X_n^*\right) &= \underset{X_1,\ldots,X_n}{\operatorname{argmax}} p\left(X_1, \ldots, X_n \mid Z_1, \ldots, Z_n, Z_{1\ldots n}\right) \\ &= \underset{X_1,\ldots,X_n}{\operatorname{argmax}} p\left(Z_1, \ldots, Z_n, Z_{1\ldots n} \mid X_1, \ldots, X_n\right) p\left(X_1, \ldots, X_n\right)\end{aligned} \tag{14}$$

However, if the initial relative poses of the robots are unknown and cannot be estimated, CSLAM is reduced to the following Maximum Likelihood Estimation (MLE):

$$\left(X_1^*, \ldots, X_n^*\right) = \underset{X_1,\ldots,X_n}{\operatorname{argmax}} p\left(Z_1, \ldots, Z_n, Z_{X1\ldots n} \mid X_1, \ldots, X_n\right) \tag{15}$$

As a result, the pose and detected landmarks of each robot need to be expressed in a global reference frame. The following will introduce the system framework of CSLAM and the three core issues of multi-robot systems.

3.1. CSLAM frameworks

In a multi-robot system, according to the different data communication methods, the multi-robot SLAM can be divided into a centralized CSLAM and a decentralized CSLAM. Figure 9 provides two different forms of CSLAM systems.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

3.1.1. Centralized CSLAM

In the centralized CSLAM approach shown in Figure 9(a), data is processed on a central or external agent of a multi-robot system, and the central processor is responsible for complex optimization tasks, while individual robots do not need to have high configurations and can be as simple as possible. The central unit fuses the communication information of all robots and provides feedback to individual robots or shares the fused information with them to better enhance their self-localization. However, as the number of robots increases, communication and bandwidth limitations become a thorny issue [86].

Most CSLAM systems are vision-based, such as CCM-SALM [87], C2VIR-SLAM [88], and CORB2I-SLAM [89], but research on LiDAR CSLAM is relatively limited. LAMP 2.0 [90] is a typical LiDAR-centric multi-robot system that has been experimented with in unknown large-scale underground environments. It combines a single-robot front-end interface that can adapt to different range sources and LiDAR configurations and a scalable multi-robot front-end that can support large-scale environments and multi-robot inter- and intra-loop closure detection. Liu et al. [91] proposed a centralized multi-robot collaborative LiDAR SLAM framework, in which each robot is equipped with only one 2D LiDAR and a loop closure selection method is used to ensure the correctness and consistency of global optimization. Jalil [92] proposed an enhanced multi-robot fast localization, ranging, and mapping algorithm (EMR-FLOAM), which uses a two-stage non-iterative distortion compensation technique to address computational complexity and resource requirements. Zhong et al. [93] proposed CoLRIO, an online, centralized, and robust collaborative SLAM framework, which enables multiple robots to generate global SLAM estimates in real-time from simultaneously collected data and achieve co-localization without deploying anchors.

3.1.2. Decentralized CSLAM

In the decentralized CSLAM approach shown in Figure 9(b), the decentralized system typically consists of several interconnected robots that are managed by a coordinating architecture and can communicate with other robots to share environmental information, poses, or maps. Distributed systems can cope with partial failures, and even if some distributed robots fail or lose contact, other robots can still complete the task or at least part of the task.

A notable advantage of distributed systems is the capability to alleviate the bandwidth burden. Xie [94] developed a real-time distributed collaborative SLAM system based on 3D LiDAR: RDC-SLAM, which uses the DELIGHT global descriptor instead of traditional point cloud matching, significantly reducing the communication bandwidth and time consumption pressure between multiple robots. Huang et al. [95] proposed a distributed multi-robot SLAM, DiSCo-SLAM, which is the first to apply scan context descriptor to multi-robot SLAM and 3D LiDAR observation, including a global-local optimization framework. The system is suitable for low communication bandwidth scenarios. Xie et al. [96] proposed an online multi-robot LiDAR system that hierarchically divides the path planning problem into global planning and local navigation. However, it requires LiDAR and Global Positioning System (GPS) data for data association and is only applicable to partially unknown environments.

DOOR-SLAM [97] is a fully distributed SLAM system consisting of two parts: position graph optimization and distributed SLAM front-end. It uses point-to-point communication, performs distributed outlier rejection to remove false loop closures, and performs distributed pose graph optimization to retrieve the robot's trajectory estimate. Also based on point-to-point communication, DCL-SLAM [98] focused on data-efficient and resilient communication between robots, enabling consistent trajectory estimation through partial attitude measurement and indirect data association. Swarm-SLAM [99] is an open-source CSLAM system specifically designed for swarm robots that emphasizes scalability, flexibility, decentralization, and sparsity, with a new budget approach to select inter-robot loop closure to reduce data exchange.

Table III summarizes several classic CSLAM frameworks. However, decentralized or distributed systems are also limited by the on-board computing and communication of robots, which brings additional challenges such as complex data management capabilities and bookkeeping strategies, information double counting, and high latency [100].

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Table III. *Summary of the most representative multi-robot SLAM frameworks.*

Multi-robot SLAM frameworks	Sensors	Data interaction	Data association	Agents types	Mapping	Open source
DOOR-SLAM [97]	LiDAR	Decentralized	NetVLAD descriptor	UGVs	3D	✓
DisCo-SLAM [95]	LiDAR, IMU	Decentralized	Scan Context descriptor	UGVs	3D	✓
LAMP2.0 [90]	LiDAR	Centralized	Flexible selection	UGVs	3D	✓
DCL-SLAM [98]	LiDAR, IMU	Decentralized	LiDAR-Iris descriptor	UGVs	3D	✓
RDC-SLAM [94]	LiDAR	Decentralized	DELIGHT global descriptor	UGVs	3D	–
MR-SLAM [91]	LiDAR	Centralized	Loop closure selection	Robots	2D	–
Swarm-SLAM [99]	Stereo, RGB-D, LiDAR	Decentralized	Scan Context descriptor, Cosplace	UGVs	3D	✓
Online MR-SLAM [96]	LiDAR, IMU, GPS	Decentralized	Scan Context descriptor, GPS data	UGVs	3D	–
EMR-FLOAM [92]	LiDAR	Centralized	Scan Context descriptor	Robots	3D	–
CoLRIO [93]	LiDAR, IMU, UWB	Centralized	Scan-Context++	UAVs	3D	✓

3.2. Key issues of CSLAM

3.2.1. Robust data association

Robust data association is the key technology to promote multi-autonomous robot collaboration. Data association refers to matching current sensor data with stored map data (or map data shared by other robots) to find the correct corresponding points or features [101], which is usually done at the front end of collaborative SLAM. Robust data association can effectively handle factors such as sensor noise, occlusion, and dynamic scene changes, avoid mismatching, and ensure the accuracy and stability of the multi-robot SLAM system.

In multi-robot LiDAR SLAM, LiDAR sensors are used to map the environment, which usually generates a point cloud dataset. Data association involves the following aspects: (1) point cloud registration: avoiding local mismatches and large-scale drift [102]; (2) sparsity of point clouds: sparse point cloud data has a low density and it is difficult to provide sufficient feature information for effective matching. The ICP algorithm and its various variants are known point cloud registration application methods but are only suitable for single-robot SLAM to calculate relative poses and are not well suited for multi-robot operations because they rely on a good initial guess, which is usually not available between the local maps of the robots.

The data association of the LiDAR CSLAM system is divided into feature matching and direct matching methods. Direct matching methods include: probabilistic model NDT with poor predictability [103], correlation matching method ICP with high computational complexity, etc. Feature matching methods perform data association and pose estimation based on the features with high significance in the

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

LiDAR scan, and align the relative pose by aligning two LiDAR scans to improve computational efficiency. 3D feature description is similar to image matching, but due to problems such as disordered organization, uneven density, lack of texture, and structural occlusion, dense point cloud information often contains more redundant data, and the outliers of 3D feature matching are very high. In addition, the descriptor-based method has high computational complexity and is difficult to process in real-time.

Shin et al. [104] argued that relatively sparse point clouds have no meaning for the depth associated with visual features. They apply the direct method to the combination of low-line LiDAR and camera to realize a loosely coupled SLAM system and solve the sparsity problem in data association. Xie et al. [94] proposed a descriptor-based approach to multi-robot SLAM and designed a communication protocol to facilitate data association between robots. However, with the increase in running time, descriptor matching time increases, making it difficult to guarantee the real-time performance of SLAM.

Perceptual aliasing can lead to incorrect inter-robot data association. Tian et al. [105] introduced a multi-robot system called Kimera-Multi that can identify and reject incorrect inter-robot and intra-robot loops caused by perceptual aliasing, improving global mapping consistency. Liu et al. [106] introduced a semantically assisted LiDAR tightly coupled SLAM method that incorporates constraints based on semantic information in the point cloud feature extraction and matching process to improve the accuracy and pose estimation precision and promote globally consistent map construction.

3.2.2. Inter-robot loop closures

The CSLAM front-end is responsible for generating landmark estimates, odometry measurements, and loop closures within and between robots. The computation of odometry and intra-robot loop closure measurements can be done completely locally on each robot [107]. Loop closure detection, also known as place recognition, refers to the robot group identifying the places they have visited before and performing data correction to eliminate drift errors. Incorrect loop closures can cause robots to connect their current location to a different location they visited previously, resulting in incorrect data association and global inconsistency issues. Correct loop closures can significantly reduce cumulative errors, helping robots avoid obstacles more accurately and quickly, especially in large-scale environments. Establishing closed-loop detection between robots requires a lot of key human resources to exchange sensor data.

The generation of inter-robot loops is crucial to ensure estimation consistency. Inter-robot closure detection can be divided into two forms: direct and indirect. Direct inter-robot closure detection is to calculate the relative pose based on the position coordinates when the robots meet, but this process is called "rendezvous." After the robots meet, they exchange observation data to identify potential loop closures. Finding an optimal lossless swap strategy (i.e., a minimum weighted vertex cover) is NP-hard. Tian et al. [108] proposed a resource-adaptive framework for distributed loop closure detection to address the resource consumption problem in inter-robot loop closure detection, enabling rendezvousing robots to collaboratively search for loop closures between robots by exchanging a near-optimal subset of their observations without exceeding the allocated data transmission budget.

Indirect inter-robot closure detection estimates the corresponding transformation by reviewing the overlapping regions in the maps. After the location recognition matching, geometric estimation is performed to calculate the relative pose between the two locations. A common method for robots equipped with 3d LiDAR to perform location recognition is to use compact and robust global point cloud descriptor matching, such as Scan Context descriptor, global descriptor LiDAR-Iris [98], etc. The similarity score between the global descriptors from the two robots is calculated to identify the locations or overlapping parts between their respective maps. Wang et al. [109] proposed a robust loop closure detection method and a global three-dimensional descriptor to improve robot mapping accuracy, but it was only applied to a single robot. Feature-based visual CSLAM [110] uses the bag-of-words (BoWs) approach to find multi-robot loop closures. Shan et al. [111] proposed a method for robust real-time place recognition using imaging LiDAR, extracting ORB feature descriptors from images and encoding them into BoWs vectors, which can quickly and robustly search for loop closures and produce 3D point clouds with image quality and high resolution.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Chen [112] proposed an inter and intra-robot loop closure selection algorithm based on robust consistency for multi-robot pose graph map fusion, which provides initial relative pose estimation between robots by introducing a new consistency score and based on a consistent loop closure selection strategy. LAMP2.0 proposed a modular loop closure method that can adapt to various environmental representations (2D grid maps and 3D maps). DCL-SLAM used the Iris descriptor derived from LiDAR data and uses the Hamming distance and KD tree method for loop closure detection. Swarm-SLAM proposed a new inter-robot loop closure priority technology that can reduce communication and accelerate convergence.

3.2.3. Global optimization

CSLAM backend optimization is an effective way to reduce the accumulated local errors in front-end matching. The backend solves nonlinear pose graph optimization, and global graph optimization generates a global factor graph to consistently and effectively represent the information of multiple robots [113]. The pose graph uses the robot pose (position and orientation) as the nodes in the graph, and the robot motion constraints (odometer data) and loop detection constraints as the edges in the graph.

Pose graph optimization (PGO) is an effective method for global optimization. PGO methods [114, 115] are also divided into centralized and distributed types. Centralized PGO collects all measurements at a central station, which is responsible for calculating the trajectory estimates of all robots. However, when the number of robots increases, problems such as excessive computational burden on the central station, reduced optimization speed, and high communication overhead will be encountered. Distributed PGO is more suitable for multi-robot systems. Each robot only maintains the pose and edges related to itself and performs local optimization. Finally, a distributed optimization algorithm is used for collaborative calculation to converge to a consistent global optimal solution.

Choudhary et al. [116, 117] proposed a distributed two-stage Gauss-Seidel method, which iteratively recovers the posture by decoupling the rotation and translation components. However, this method has high initialization requirements and cannot guarantee convergence to the first-order critical point. Fan and Murphey [118] proposed an optimization minimization method for distributed PGO that is guaranteed to converge to the first-order critical point under mild conditions. Li et al. [102] proposed a distributed pose graph optimization algorithm that combines multi-level partitioning and accelerated optimization methods. The communication overhead of multi-robot parallel computing is reduced by balancing the posture graph. Based on the non-convexity of the PGO problem, the IRBCD method is used to quickly converge to the optimal solution. Optimizing the global coordinate transformation graph can ensure high-quality initial guesses, thereby stably and accurately optimizing the local pose graph of the robot. Disco-SLAM [95] contains a two-stage global-local optimizer. In the global optimization, the factor graph containing the transformation of the robot's local frame to the global frame is optimized. In the local optimization, the local pose graph of the robot's local odometry, the robot's internal constraints, and the constraints between adjacent robots are optimized.

4. Challenges of MRS sub-fields

At present, CSLAM is based on multi-robot systems execution of mapping tasks. For multiple robots that make up a robot cluster, the pose and position of each robot are first estimated and calculated, that is, localization. The frontier exploration strategy is used for frontier detection, and tasks are assigned to each robot through the task allocation algorithm. The robots go to the target points for exploration according to the assigned task and use the environmental data information sensed by their sensors to build their local map, which is then fused to obtain a global map.

The framework diagram of multi-robot SLAM is depicted in Figure 10. After each function, the content and details of the global map will increase until the exploration ends. Compared with single-robot SLAM, multi-robot SLAM expands three key sub-fields: frontier detection, task allocation, and map fusion.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

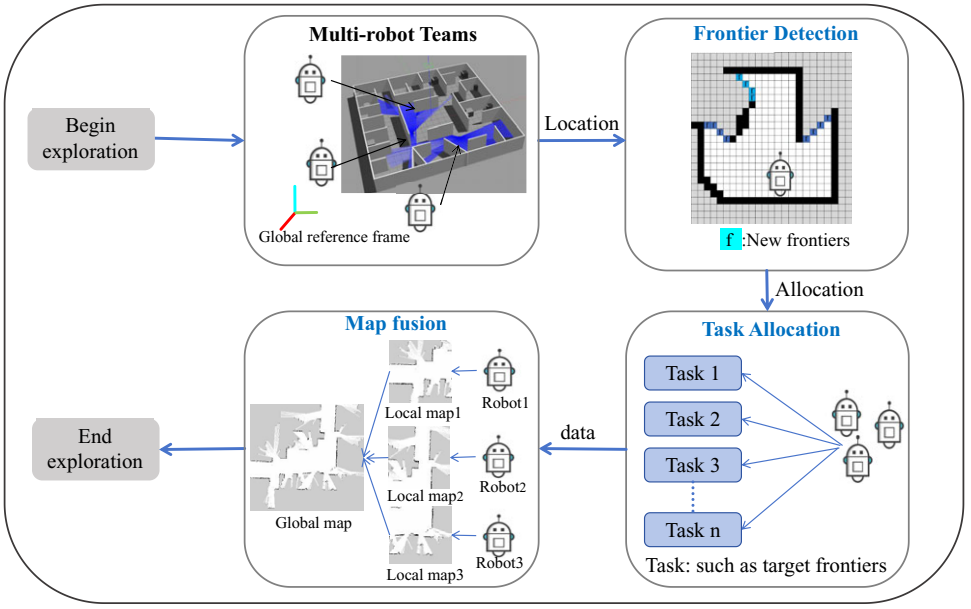


Figure 10. *Multi-robot SLAM workflow framework. The framework includes three core issues: frontier detection, task allocation, and map fusion. The robot teams perform exploration tasks in a global coordinate system.*

4.1. Frontier detection

Frontier detection is a widely used exploration method that plays a crucial role in identifying frontiers, which act as boundaries between explored and unexplored regions on the global map. As illustrated in Figure 11, the frontier refers to the boundary between the unknown area and the known area in the current map, which consists of a set of unknown points. In frontier-based exploration, each point is connected to at least one open space. The robot engages in exploration by continuously detecting and navigating toward these boundaries, which separate familiar regions from unexplored areas. Frontier detection involves the identification of boundaries between unknown areas and known areas without obstacles on the map. The significance of frontiers lies in determining optimal locations for the robot's future exploration within the environment [119].

Yamauchi [120] introduced the concept of boundaries and pioneered active exploration based on boundary detection and tracking. Yamauchi's work involved each robot moving towards the centroid of the nearest boundary. The robots independently navigate toward the target while sharing a common map, and boundary detection is performed only upon reaching the target. This work laid the foundation for future mobile robots to explore unknown maps. Subsequently, detected boundaries have become commonly used as navigation targets for mobile robots in exploration mission planning. Building on this foundation, more sophisticated exploration strategies have emerged. Table IV provides a summary comparing different boundary detection algorithms.

4.1.1. WFD, FFD

Early scholars used image processing technology, such as Canny edge detection, to extract boundary points from map images. Once the boundary is detected, the frontier on the boundary will be assigned as targets for the robot to explore. However, this method is only suitable for small-scale scenes. When the scene gets large, the efficiency of the image processing process for the entire scene will be significantly reduced.

Most early frontier detection algorithms dealt with the entire map data. This process was time-consuming and could impact exploration decisions. Keidar and Kaminka proposed the Wavefront

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

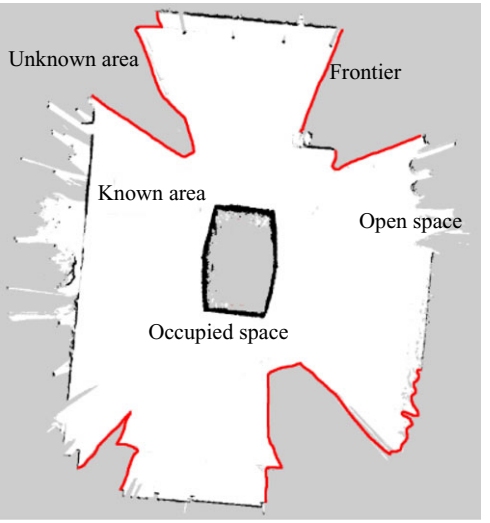


Figure 11. *Frontier-based exploration process. The unknown area refers to the gray area around the map, and the known area is the obstacle (such as a wall) detected by the black line on the left. The frontier is the boundary between the known and unknown areas marked in red, the open space is the area to be explored on the right, and the occupied space is the gray portion marked by the black border in the middle of the map (such as obstacles).*

Frontier Detector (WFD) and the Fast Frontier Detector (FFD) [121, 122] to address this issue. The WFD algorithm is a graph-search-based approach that focuses on examining known areas of the map. Limiting the analysis to known areas reduces the computational burden compared to processing the entire map. The WFD algorithm efficiently detects frontiers by examining the boundaries between known and unknown regions. However, as exploration progresses, the known area that needs to be processed becomes larger. The FFD algorithm only extracts the exploration frontier each time it acquires sensor data. However, it can only extract new frontiers from sensor data when the robot obtains unknown environmental information.

Quin [123] extended the WFD algorithm by utilizing the frontier cells from the previous time step as the starting point for detecting the frontier in the current time step. This extension combines the ideas of WFD and FFD. Instead of exploring all unoccupied areas like FFD, it searches for new unoccupied regions. In addition, the Frontier-Tracing Frontier Detection (FTFD) algorithm proposed by Quin et al. [124] uses the perimeter observed by the current sensor combined with the laser scanning endpoint and the previously known boundary to perform boundary detection. FTFD is suitable for applications that require frontier detection after each observation and can be extended from 2D to 3D, but the effect is not obvious. Sun et al. [125] extended the DFD method [119] by first expanding the submap and using WFD to detect the local reachable frontier, and adaptively selecting the span according to the posture deviation of the submap, which is more effective and flexible than the DFD method.

4.1.2. RRT

The Rapid Random Tree (RRT) algorithm is currently one of the most mainstream boundary-based exploration methods for robot autonomous exploration. Umari [126] proposed using RRT for boundary detection. RRT involves random sampling within a specified region and creating tree edges and nodes based on the sampled points and their nearest valid tree nodes. Through iterative iterations, a comprehensive tree-like structure is formed. This approach integrates the generation of local and global trees to effectively and efficiently select boundary points. In this method, the RRT algorithm is utilized for boundary point selection, performing path planning on a global scale and refining it locally.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Table IV. *Algorithms of frontier detection.*

Method	Reference	Expand	Advantages	Limitations
WFD	[121]	WFD	It focuses on checking the known areas on the map and has strong stability	The amount of calculation increases significantly over time and the real-time performance is poor
	[123]	E-WFD	It has high real-time performance and is suitable for large-scale environment exploration	Dependent on the initial conditions
FFD	[122] [121]	FFD	High real-time performance	The accuracy is relatively low and it is sensitive to environmental changes
	[124]	FTFD	The algorithm can be extended from 2D to 3D, but the effect is not obvious	When dealing with complex environments, the computational load of tracking the previous frontier and scanning endpoint information will increase significantly
RRT	[126]	M-RRTs	The simultaneous exploration of multi-RRT has increased the exploration speed	Work is limited to a single robot case
	[127]	ID-RRT	It can effectively explore unknown environments while obtaining maximum information gain	The computational complexity of complex environments is relatively high, and dynamic environmental factors have not been considered
	[129]	SMF	Delete redundant tree nodes to reduce memory consumption	The complexity of parameter tuning for multi-tree fusion needs to be considered
	[128]	E-RRT	Evaluation is conducted based on the market-based task allocation strategy and multiple evaluation rules	The parameter dependence is strong and the communication dependence is relatively high
	[133]	HBS-RRT	The semantic information of the environment is used as a heuristic to guide the robot exploration	Rely on the accuracy of target recognition
	[132]	A-RRT	Real-time incremental frontier detection, adaptive sampling space improves RRT sampling rate	It has a strong reliance on map updates
	[131]	HFD	Effectively handle large-scale environments with multiple obstacles	It integrates three different cutting-edge detection algorithms, and the overall algorithm complexity has increased
	[134]	HMS-RRT	Consider the effectiveness of boundary points to improve detection efficiency	Rely on effective communication between robots
DRL	[136]	DRL-based	Optimize exploration paths, and have strong learning ability and adaptability	Environmental adaptability is limited and is assumed to be based on a static environment
	[137]	DRL-based	Efficient path planning and strong generalization ability of training network	Large computing resource requirements

Pimentel [127] enhanced the traditional RRT algorithm by incorporating Shannon entropy to estimate the expected information gain (ID-RRT) for each candidate boundary. Their approach generates paths for navigation to selected destinations. Zhang [128] extended the RRT algorithm by introducing a cost function that considers passing through unknown and unexplored regions. The trajectory length from the initial position to the target frontier point is defined as an unconstrained optimization problem.

However, the randomness of the RRT algorithm may cause some boundary points to fail to be detected in time, resulting in backtracking. Therefore, Qiao [129, 130] introduced a multi-tree fusion algorithm

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

called Sample-based Multi-tree Fusion (SMF), which combines sampling with RRT. SMF retains tree nodes while discarding tree edges after boundary detection, allowing the tree to grow across obstacles. The proposed block structure significantly reduces the number of tree nodes stored in the search process, thus reducing the memory consumption. Xu et al. [131] proposed an autonomous exploration strategy based on hybrid frontier detection to deal with large-scale environment problems with multiple obstacles. It consists of a variable step-size random tree global frontier detector, a multi-rooted random tree boundary detector, and a grid-based boundary detector. With the increase of expansion, the number of nodes is increasing, which affects the exploration efficiency. Sun et al. [132] proposed an efficient frontier detector method based on adaptive RRT to achieve real-time incremental frontier detection. Its adaptive sampling space improves the successful sampling rate of RRT. The detector tends to sample in the newly explored area, which improves the efficiency of boundary detection.

The discrete detection strategy based on the RRT algorithm ignores the geometric continuity of the environment, resulting in disordered robot motion, and the efficiency of the traditional RRT algorithm decreases as the exploration area increases. To address this problem, Liu [133] proposed a heuristic-based biased sampling robot exploration strategy, which uses the connection between environmental semantic information and geometric heuristics as heuristic information to guide robot exploration. After obtaining the estimated area of the heuristic object, a biased RRT algorithm is proposed to accelerate the frontier detection of the estimated area.

Furthermore, Ning [134] proposed a frontier detection algorithm based on a hybrid multi-strategy RRT. This method combines an adaptive distance increment strategy, subregion sampling strategy, and greedy boundary exploration strategy. The effectiveness of the identified boundary points is carefully considered, resulting in enhanced detection efficiency.

4.1.3. DRL

Currently, deep reinforcement learning (DRL) is in a boom state. The traditional heuristic method is efficient and easy to use, but it is limited to specific cases. Because they lack the ability to integrate the intentions of teammates, they perform poorly in decentralized execution. Due to the use of manually designed utility functions and the lack of structural information in the field, these methods are difficult to be extended to different chaotic fields. Therefore, the advantages of deep reinforcement learning are embodied [135]. Leong [136]'s method optimizes the robot's movement through reinforcement learning and uses a reward mechanism to guide the robot to select the optimal frontier point. Compared with traditional frontier exploration strategies, it can effectively improve the problem of non-optimal paths in scenarios with multiple frontiers and similar distances, enabling the robot to choose a more reasonable path during the exploration process and improve exploration efficiency. Wang [137]'s enhanced frontier exploration method, combined with DRL, uses global and local networks to extract map features, allowing the robot to better understand the environment than traditional boundary-based methods and complete exploration at a lower distance cost.

4.2. Task allocation

Through frontier detection, several target points need to be explored by robots, and these target points need to be reasonably allocated to multiple robots to complete the SLAM task under the condition of optimal target. How to assign appropriate tasks to different robots to optimize the objective function is a process called Multi-Robot Task Allocation (MRTA) [138]. MRTA is characterized as a combinatorial optimization problem that aims to identify the optimal solution from a finite set of feasible solutions. The objective function in MRTA operates on a discrete domain with a vast search space.

Each robot explores based on simple criteria for the nearest boundary, which can lead to problems with repeated exploration and even collisions. Therefore, we need to allocate the boundary points reasonably through evaluation. A single factor is not sufficient to effectively disperse multiple robots between

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Table V. *Multi-robot task allocation categorization.*

Classification	Type 1	Type 2
Robot	ST, robots complete one task at a time	MT, robots complete multiple tasks at the same time
Task	SR, each task is completed by one robot	MR, a task requires multiple robots to complete
Allocation	IA, robots are assigned tasks to be completed immediately	TA, robots are given a task schedule to complete

tasks, and the expected costs and benefits of executing each candidate boundary point must be estimated. Evaluating all boundary points to estimate the effect of performing a set of actions to reach each boundary point is utility computing [139]. The commonly used methods of utility evaluation include information theory (IT), optimal experimental design theory (TOED), etc. Gao [140] designed a new utility function that considers the optimal frontier allocation of the robot's moving distance, boundary size, and turning cost. By constructing a topological graph and recording the key nodes visited and unknown states, this method can better balance map coverage and time consumption. Sun et al. [141] proposed a detection and maintenance method for incremental addition of frontiers and a multi-path generation method, using the proposed multi-objective utility function for frontier evaluation to obtain the optimal path and improve exploration efficiency.

In addition, after the task assignment, the robot also needs to perform corresponding path planning. RRT algorithm and A* algorithm are both commonly used path planning algorithms. Fusic [142] introduced an improved fast random tree (IRRT*) algorithm, which uses triangle inequality rewiring technology to find collision-free paths for drones in three-dimensional environments. After that, they [143] proposed a mobile robot path planning method based on adaptive learning particle swarm optimization, aiming to find the most effective path in an obstacle environment mapped by LiDAR. For a more detailed review of path planning, please refer to [144, 145]. This section focuses on the relevant algorithms for task allocation.

Multi-robot task allocation can be classified into single-task (ST) and multi-task (MT) based on the number of tasks a robot can handle concurrently. Additionally, they can be categorized as single-robot task or multi-robot task (MR), depending on the number of robots required for each task. In terms of the time plan, assignments can be further divided into instantaneous assignments and time-extended assignments (TA) [146, 147]. The classification criteria are summarized in Table V.

Scholars at home and abroad have conducted in-depth research on the multi-robot task allocation problem and proposed various algorithms and variants. The currently popular task allocation algorithms can be roughly divided into five categories: market-based allocation strategies [148], optimization-based [149], bio-inspired, behavior-based [150] allocation strategies, and reinforcement learning-based allocation strategies. We review some classic task allocation algorithms and compare them in Table VI.

4.2.1. Market-based

Market-based strategies, inspired by economic theory, provide an effective approach for coordinating robot activities and are commonly used to solve MRTA problems. Auctions, which consider robot bids and auction criteria (objective functions), are employed to allocate tasks to robots. Market-based methods utilize robust communication mechanisms for coordination, with a central unit collecting auction bids and performing task assignments [148]. In these strategies, robots are treated as bidders and tasks as commodities. Robots detect boundaries and bid on them, adding the won boundaries to the task list of the winners. Tasks for each robot are sorted based on their new auction prices at each iteration [151]. Please refer to the works of De Ryck [152], Yu [151], Ning [134], and others for more details on market-based strategies for multi-robot task allocation.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Table VI. *Algorithms of task allocation.*

Classification	Reference	Algorithm	System type	Question type
Based on optimization	[155]	Based on improved ant colony algorithm	Centralized	MT-MR-TA
	[153]	PSO + Greedy	Centralized	ST-SR-TA
	[154]	expand PSO, based on Pareto frontier	Centralized	MT-MR-TA
	[156]	STADAA	Distributed	ST-SR-IA
	[157]	IMMODE-SA	Distributed	ST-SR-IA
	[147]	Clustering and game theory	Centralized	ST-SR-TA
	[158]	Multi-Objective Teaching-Learning-Based Optimization Algorithm	Centralized	MT-SR-TA
Based on market	[151]	Based on market	Centralized	ST-MR-TA
	[152]	Sequential single-item auction	Distributed	ST-MR-TA
	[134]	Based on improved market mechanisms	Distributed	ST-MR-TA
Based on behaviers	[161]	Alliance-based MRTA architecture	Distributed	ST-SR-TA
Based on reinforcement learning	[163]	Based on reinforcement learning	Centralized	ST-MR-TA
Bio-inspired	[159]	Locust-based heuristic algorithm	Distributed	MT-MR-TA
	[160]	Animal foraging behavior subject to area-restricted search	Distributed	ST-MR-IA
Based on random conflicts	[225]	Based on random conflicts	Distributed	ST-SR-TA

4.2.2. Optimization-based

The optimization-based task allocation strategy generalizes the dynamic MRTA problems into multiple traveling salesman problems and classifies them as combinatorial optimization problems. Multi-objective optimization tasks improve the task completion rate in less time but have poor adaptability to objective factors and low robustness. Classic optimization-based strategies include the ant colony algorithm, particle swarm algorithm [153, 154], and their variants.

The basic ant colony algorithm suffers from slow convergence and a tendency to get stuck in local optima. Qin [155] addressed these issues by introducing a local optimization mutation operator and an improved simulated annealing algorithm. The mutation operator was used for local optimization, and the algorithm employed the Metropolis criterion from the simulated annealing mechanism to probabilistically accept worse solutions. This allowed the algorithm to escape local extreme values and achieve better optimization results, while also exhibiting faster convergence speed.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Lindsay [156] proposed a Sequential Task Addition Distributed Allocation Algorithm that enables fast and scalable allocation in a specific class of MRTA problems. However, its applicability is limited to this particular class of problems. Miao [157] proposed an improved multimodal multi-objective differential evolution algorithm hybrid with a simulated annealing algorithm to solve MRTA problems. Martin [147] introduced a cooperative game framework to address MRTA problems. They established a cooperative game to evaluate the performance of robots and task sets. In their experiments, they compared the performance and computation time of their game theory-based approach against a benchmark genetic algorithm. The results showed that the game theory-based method outperformed the genetic algorithm in terms of performance and computation time. Kang [158] proposed a teaching-based multi-objective optimization algorithm and established a mathematical model to study the task allocation problem of multiple weeding robots.

4.2.3. Bio-inspired

Bio-inspired algorithm is a method to solve the task assignment problem by drawing on the self-organization, adaptability, and optimization ability of biological system. Bionic algorithms are typically rooted in the concept of swarm intelligence, which involves abstracting a problem into the interactions and cooperation among individuals. These algorithms aim to find optimal solutions by simulating the behaviors exhibited by biological groups. Kurdi [159] proposed a novel autonomous bionic method for allocating tasks among multiple drones during a mission. Aljalaud [160] proposed a multi-UAV task allocation system inspired by area-restricted search, which is inspired by animal foraging behavior, that is, after finding prey, animals usually spend more time searching in the nearby area to maximize food intake. The algorithm shows better net throughput than the baseline performance, especially in the case of small fleet size, without affecting runtime performance.

4.2.4. Behavior-based

Behavior-based dynamic task allocation assumes that the frontier is attractive to robots, while obstacles and other robots are repellent. The multi-robot exploration problem involves a two-layer behavior-based control structure, where task identification and inter-robot communication are referred to as high-level behaviors, and obstacle avoidance, navigation, and task switching are referred to as low-level behaviors. When necessary, high-level behaviors can inhibit or prevent low-level behaviors. This strategy is particularly well-suited for MRTA in scenarios with weak communication capabilities and uncertain, dynamic real-world environments. Dos [161] proposed an axiomatic framework based on Social Choice Theory to analyze the task allocation problem in intentional cooperation multi-robot systems.

4.2.5. Reinforcement learning-based

The reinforcement learning-based approach [162] makes decisions by learning from the behavior of other robots, but it may suffer from overfitting to specific environments. Commonly used reinforcement learning techniques include Q-learning. Task allocation strategies based on reinforcement learning present a promising direction for further research. By employing a reward strategy, rewards or penalties are assigned based on the chosen actions. Park [163] proposed a Markov decision process model for multi-robot task allocation using reinforcement learning to address the problem that the performance of existing metaheuristic methods deteriorates as the number of robots or tasks increases. The proposed formula follows a sequential assignment approach, aiming to minimize the overall completion time of the tasks. It incorporates a cross-attention mechanism to determine the preference of robots for each task. This enables the algorithm to identify the optimal allocation solution for each problem and improve the solution quality, especially when dealing with large-scale allocation problems.

DRL combines the powerful nonlinear function-fitting capabilities of deep neural networks with the decision-making and control abilities of reinforcement learning, offering a novel approach to solving the multi-agent task planning problem. Xie et al. [164] used DRL to solve the three-dimensional path

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

planning problem of UAVs. Zhang et al. [165] proposed a multi-unmanned surface vessel (USV) mission planning method based on improved DRL to solve the problem of how to design a suitable reward function in mission planning. Jin et al. [166] proposed a multi-robot collaborative path planning and task allocation method based on a dual deep Q-learning network and ant colony optimization algorithm in a 3D environment.

4.3. Map fusion

Map fusion can be viewed as a specialized instance of image registration in computer vision, wherein the image corresponds to a map with a distinct geometric structure. Occupancy grid maps, which consist of matrices storing the probability of each grid cell being occupied, are commonly used in this context [167]. By treating each value in the occupancy grid map matrix as a pixel in an image, the representation of grid map fusion becomes analogous to image fusion. Grid map fusion is a crucial challenge in both multi-robot SLAM and single-robot SLAM. In computer vision, the process of merging occupancy grid maps is transformed into an image registration problem by establishing an objective function to calculate the transformation matrix.

Occupancy grid maps have been the subject of extensive research, and various merging methods have been proposed. These methods include probabilistic methods, optimization-based techniques, and feature-based methods. In addition, map merging algorithms have gradually evolved from 2D to 3D point clouds. This paper focuses on feature-based map fusion methods.

The feature-based method consists of three stages: feature detection, feature description, and feature matching. Common feature detection methods include SIFT detector [168], SURF detector [169], ORB detector [170], Accelerated-KAZE (AKAZE) detector [171], etc. Common feature description methods include SIFT, SURF, Brisk [172], Freak [173], etc. In the feature matching stage, the Random Sample Consensus [174] method is often used for fine matching, and the transformation matrix T is calculated based on the matched feature pairs to perform image transformation and fusion.

Early methods for multi-robot map fusion represented advancements over existing single-robot map fusion techniques adapted for multi-robot systems. Thrun [175] used a combination of maximum likelihood estimation and particle filters to create a consistent global grid map on multiple robots, but this required the relative initial positions of multiple robots to be known and had high requirements for data communication. Subsequently, Howard [176] used Rao-Blackwellized particle filters for grid map fusion and did not require the initial positions of the robots to be known. Table VII presents a comprehensive summary and comparison of various map fusion algorithms.

4.3.1. Optimization-based methods

Optimization-based methods aim to find the best map configuration through optimization algorithms (such as particle swarm optimization, genetic algorithms, etc.) or objective functions. Lee [177] proposed a sinogram-based particle swarm optimization multi-hypothesis map merging algorithm to address the challenge of merging maps without features, local maxima, and initial relative pose information. Wang [51] proposed a heuristic search method to guide the fusion of repeated areas of local maps. Each robot sends local map information to the same workstation, and uses the similarity of local maps as the judgment indicator for fusion to obtain the optimal global map.

4.3.2. Feature-based methods

Feature-based methods are one of the most commonly used map fusion methods. The feature-based map fusion method is as follows: After obtaining the local maps of the two robots, the images can be preprocessed, such as binarization, to make the detected feature points more accurate. Next, the feature points of the two maps are detected and described, and then the feature points are matched. According to the matched feature point pairs, the transformation matrix T is calculated. After obtaining T, the image

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Table VII. *Algorithms of map fusion.*

Category	Algorithm	Reference	Characteristic
Based on optimization	Based on multi-hypothesis analysis with sinograms	[177]	Improved flexibility in multi-robot systems without requiring initial relative poses
	Overlap-based heuristic objective function	[51]	Easily scalable to more robots, maximizing MR-SLAM efficiency
	MCS	[167]	No assumptions are needed about the robot pose
Feature based	Based on Maximum Empty Rectangle (MER)	[178]	Map merging using MER as local map features, independent of relative robot positions but increases complexity
	A own corner detector and a cylindrical descriptor	[226]	Fast generation of consistent collaborative maps
	Based on Hypothesis Boxes	[179]	Significantly improves map merging efficiency in low feature matching scenarios
	Extract 2D edge point set registration	[227]	Supports merging grid maps with different resolutions
	A speed-up version of KAZE	[228]	It reduces the computational complexity and improves the robustness of the search process
3D point cloud fusion	Deep learned descriptors	[180]	Fast and Robust Autonomous 3D point cloud map merging for Egocentric multi-robot exploration
	A temporal bidirectional filtering module based on M-Detector	[182]	An outlier removal mechanism is designed for loop closure to reject false positives, significantly reduce outliers
	A graph-based robust 3D point cloud map merging approach for large-scale applications	[181]	Improving the robustness of point cloud map fusion

is translated or rotated according to T, and finally, the map is fused to obtain the global map. Figure 12 shows a process flow chart of feature-based map fusion.

Common features include points, lines, surfaces, and some geometric features such as triangles and rectangles. Sun [167] proposed a map fusion algorithm based on the maximum common subgraph (MCS), which uses Harris corner points to extract corner points from the grid map, constructs a triangular isomorphic graph, calculates the initial optimal transformation matrix, and finally iteratively finds the MCS in the generated polygonal isomorphic graph and optimizes the transformation matrix. Jazi [178] proposed a map merging algorithm based on the maximum empty rectangle (MER). Map merging is performed by constructing the MER as a local map feature. Experimental results show that the algorithm can merge local maps with a small overlapping area relative to the entire map. However, the sensitivity of the MER to rotated maps increases the complexity of the fusion process. Chen [179] proposed a hypothesis-based map fusion method. First, they extracted rectangular corners

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

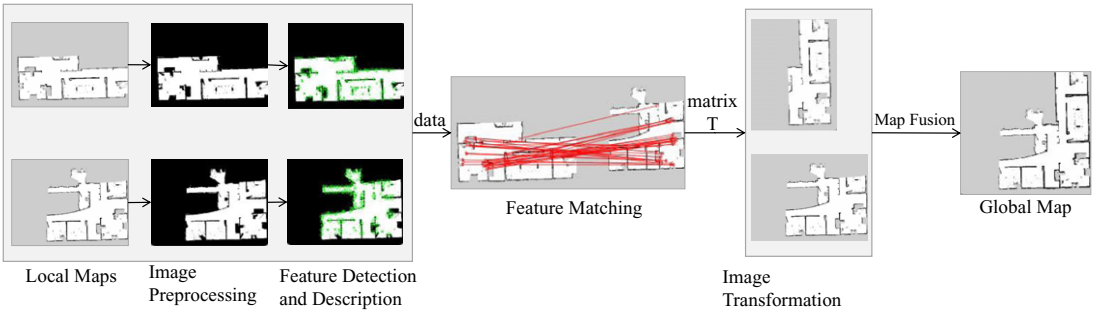


Figure 12. *Flow chart of feature-based map fusion.*

based on their shared edge features and constructed virtual bounding boxes in the map. Then, they obtained a transformation matrix based on a matched pair of virtual bounding boxes and optimized the transformation matrix using a KF. Wang [44] applied AKAZE to map fusion, using the AKAZE features of overlapping areas to estimate the relative pose of the robot, described by the modified local difference binary descriptor. The algorithm reduces the computational complexity and improves the robustness of the search process.

4.3.3. 3D point cloud fusion methods

3D maps are now crucial in various applications of mobile robots, and SLAM can generate dense 3D point cloud maps (PCM). Stathoulopoulos [180] proposed a 3D point cloud map merging framework based on overlap detection and alignment for egocentric heterogeneous multi-robot exploration that does not rely on manual initial guesses or prior knowledge of the robot poses. Their solution leverages unknown recognition learning descriptors to provide fast and robust region overlap estimation through the framework's main pipeline, eliminating the need for time-consuming global feature extraction and feature matching typically used in 3D map integration. Gui [181] introduces a graph-based robust 3D PCM merging method for large-scale applications. Wei et al. [182] propose an emerging 3D point cloud map merging framework, design a temporal bidirectional filtering method to remove dynamic objects from 3D LiDAR point cloud data, and use a binary triangle combination descriptor to provide accurate location description, thereby improving the accuracy of the merged result.

5. Open questions of LiDAR SLAM

With the further development of multi-robot LiDAR SLAM technology, based on the existing technology, its future development trends include the following aspects.

5.1. Challenges and solutions in unstructured scenarios

Robots have achieved relatively high levels of autonomy in controlled and static environments. However, in unstructured environments, complex environmental perception makes it challenging to develop fully autonomous mobile robots. Liyana et al. focused on the problems of robot perception and scene understanding in unstructured environments and their applications in related positioning, mapping, and path planning, and conducted an extensive literature review [183]. Min et al. [184] discussed autonomous driving technology in unstructured environments, including challenges such as environment perception, pose estimation, and mapping, and pointed out the future research direction of autonomous driving technology in unstructured scenarios.

Robot perception, scene analysis, and deeper scene understanding are key issues for autonomous robot navigation in unstructured outdoor environments. In some unstructured environments and dynamic

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

scenes with uneven terrain and cluttered objects, there are many limitations on robot perception capabilities. Vision-based SLAM is very unstable, and feature-based SLAM will also cause feature loss due to some dynamically moving scenes [185]. Under changing lighting conditions and shadows, LiDAR SLAM using point-cloud registration methods is more stable than visual SLAM and feature SLAM, but at the cost of losing some environmental semantic details. It is difficult to achieve global perception using only low-dimensional LiDAR feedback, especially in environments with highly repetitive features or rich semantic information. Therefore, heterogeneous multi-robots equipped with multiple sensors can build maps more effectively in complex environments [186].

The combination of LiDAR SLAM and visual SLAM has been studied and applied, using visual sensors to introduce rich information and the robustness of LiDAR, and is widely used in 3D maps. Park et al. [187] introduced a two-stage cascade deep neural network that integrates 3D LiDAR and stereo disparity maps to yield high-precision disparities. Chen et al. [188] introduced a location system based on sensor fusion, which integrated LiDAR and visual sensors, constructed a hybrid grid map for global localization, used a visual global descriptor to accelerate positioning convergence, and combined with a posture refinement pipeline to improve positioning accuracy.

RGB cameras capture high-resolution images, but lack the precision needed for accurate 3D positioning and camera motion tracking. High-resolution LiDAR systems offer precise depth estimation; however, their high cost restricts their use in various applications. Although 2D single-beam radars are more affordable, their depth estimation is insufficient for effective environmental perception. Therefore, the combination of 2D LiDAR and RGB sensors is a good feasible solution. Lu et al. [189] used a new self-supervised network and an economical single-beam LiDAR and camera sensor fusion to achieve performance comparable to expensive LiDAR systems. Best et al. [190] proposed a collaborative autonomy pipeline for multi-sensor detection in constrained environments that effectively exploits range and visual sensing modalities and is resilient to adverse events.

In addition, it is crucial to build robust datasets that encompass the diversity and complexity of unstructured environments. Future work should focus on creating synthetic and real-world datasets, including dynamic and richly varied terrains, for training machine learning models. We summarize some datasets [191–195] for multi-robot autonomous navigation in unstructured scenarios to support interested researchers.

5.2. Semantic maps and deep learning in dynamic environments

Most SLAM methods assume successful execution in a static environment. Once in a highly dynamic environment, this assumption exposes the weakness of the SLAM algorithm, which lacks a module that pays special attention to dynamic objects [196]. In highly dynamic environments, where large areas may be significantly occluded by objects [197, 198], the entanglement of dynamic object features can introduce errors that can be interpolated over and over, ultimately causing the algorithm to fail. Furthermore, without loop closure, the inclusion of dynamic object features can lead to errors.

Rapid advances in deep learning in recent years have introduced novel perspectives for semantic maps-based SLAM. Integrating deep learning algorithms with visual semantic SLAM empowers robots to conduct more precise semantic segmentation, autonomously comprehend image features, identify targets, and simultaneously assess image similarity to accomplish closed-loop detection. This capability enables efficient completion of SLAM tasks in dynamic environments [199]. Wu et al. [200] proposed a dynamic environment robust visual SLAM system YOLO-SLAM, which closely combines the target detection method and the geometric constraint method to effectively reduce the impact of dynamic targets. Li et al. [201] integrated YOLOv8's real-time object detection and image segmentation technology into the ORB-SLAM3 framework, achieving high-precision and robust visual SLAM in dynamic environments. To address problems such as semantic inconsistency in highly dynamic scenes, Kong et al. [202] proposed a place recognition method based on LiDAR semantic consistency, combining a LiDAR semantic image restoration network and a PR network based on a semantic pyramid transformer to perform LiDAR semantic restoration.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Currently, the method used to solve the SLAM problem in dynamic environments is mainly visual SLAM, such as RGB-SLAM. RGB-D SLAM is a more popular type of visual SLAM, especially in indoor environments [203]. In addition, combined with deep learning-based semantic information, dynamic object class semantics are classified, invalid points are eliminated based on multiple loop detections, features are tracked, and local mappings are continuously corrected. Semantic segmentation is used to extract semantic information, predict labels, and generate masks for detected objects. The performance of visual SLAM in dynamic scenes can be greatly improved by identifying and removing potential dynamic objects. Blitz-SLAM [204], a semantic SLAM system in dynamic indoor environments, combines the advantages of semantic and geometric information of mask, RGB, and depth images to remove noise blocks formed by moving objects in local point clouds.

Deep learning-based semantic information can solve this problem, but this method has a high computational cost and cannot handle unknown objects. Ji et al. [205] proposed a real-time semantic RGB-D SLAM system for dynamic environments that can detect known and unknown moving objects. The system only performs semantic segmentation on keyframes, reduces computational costs, removes known dynamic objects, and maintains stable static mapping. The learning-based semantic information method is used to handle potential dynamic objects, and an effective geometric module is proposed to handle unknown moving objects. Kenye et al. [196] proposed a semantic module that can be incorporated into SLAM to identify dynamic objects, classifying semantics according to the probability that they belong to the dynamic object class. They also proposed a depth masking method based on depth-first search, which uses the depth image and the detected bounding box to generate a lower-resolution binary mask image. However, the current research in this direction is still insufficient, and there are still many points worthy of attention and exploration, such as the practical application in multi-robot systems.

5.3. Active exploration and active SLAM

SLAM research has been ongoing for decades, resulting in a relatively mature technology. However, the field of active SLAM is still in the developmental stage. Unlike traditional SLAM, active SLAM involves a decision-making process where the robot autonomously selects its future path to control its actions [206]. Zhang et al. [207] applied the active SLAM system to improve exploration efficiency, using LiDAR SLAM and Cartographer as benchmarks for environment modeling to provide theoretical support for active exploration. Alcalde et al. [208] used deep reinforcement learning to implement active SLAM in complex environments. Taking into account map information and robot posture uncertainty, they designed a reward function based on map completeness, uncertainty, and closed loop, which improved the robot's path planning and generalization capabilities, and provided new ideas and methods for the development of active SLAM technology.

Promising research directions include leveraging advancements in neural networks to extend the scope of SLAM beyond line-of-sight and exploring new environmental representations, as well as environmental perception and exploration-utilization balance during active exploration [209]. Furthermore, achieving repeatability in active SLAM poses a fascinating challenge. Currently, practical applications of active SLAM are limited, and the field is in its nascent phase. However, these circumstances offer ample opportunities for future trends and advances.

5.4. Limitations of LiDAR SLAM in practical applications

While SLAM has improved its ability to adapt to the surrounding environment, traditional 2D exploration strategies remain a significant limitation, mainly for ground robots. This also limits the ability of SLAM systems to reconstruct 3D environments in the real world autonomously. In addition, the high cost of advanced SLAM systems poses a significant barrier to research and development in this field, while many low-cost SLAM systems operating in resource-constrained situations cannot achieve high-precision real-time mapping and localization, making them unsuitable for practical applications. Pang et al. [210] introduce a cost-effective SLAM system design that significantly reduces costs while maintaining high performance. First, a LiDAR-inertial odometry framework is used with a fast ICP algorithm

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

to adjust the SLAM algorithm to balance accuracy and real-time performance. The 3D multi-target RRT algorithm is combined with a nonlinear model predictive control for autonomous exploration in complex environments. Its accuracy and efficiency are comparable to those of more expensive SLAM systems, and the research results are open.

The high cost and fragility of multi-line mechanical LiDARs limit their application in real-world mapping environments. To reduce costs, multiple LiDAR odometries can be used to solve a 360-degree field of view, thereby building dense 3D maps. Chen et al. [211] proposed a real-time, computationally efficient multi-lidar-inertial odometry system that supports multiple LiDAR settings and adopts a novel parallel state update strategy and voxelized map format to optimize computational efficiency and achieve efficient mapping.

Swarm navigation tasks [212] have become one of the hot research topics in the field of MRS and have been widely studied in practical applications of multi-robots. EGO-Swarm [213] is a decentralized quadrotor swarm system that proposes a robust topology planning method to utilize onboard resources for swarm navigation. The framework is insensitive to unreliable communications and positioning drift. Park [214] proposed an online distributed trajectory planning algorithm for a swarm of quadrotors to ensure the return of collision-free, dynamically feasible trajectories in complex environments. He conducted comparative experiments with Ego-Swarm in two obstacle environments, and the success rate was higher than that of Ego-Swarm. Swarm-LIO [215] is a fully decentralized state estimation method for aerial swarm systems. Each robot exchanges self-state estimates and mutual observation information, using 3D LiDAR for drone detection, identification, and tracking. It is applicable to various extreme environments, for example GPS-denied scenes, degenerate scenes for camera or LiDAR. The method combines LiDAR, IMUs, and mutual observation measurements to provide accurate ego state and relative state estimation. Swarm-LIO2 [216] is an extension of this work, further improving the scalability of the group and showing excellent robustness in LiDAR degradation scenarios.

The SubT Challenge seeks to develop novel integrated solutions capable of mapping and navigating complex and dynamic terrain, sensors, and computation for perception in adverse conditions, distributed information sharing in degraded communications environments, and collaborative autonomy for extended operations with minimal human intervention [217]. In the DARPA SubT Challenge, mobile robot teams must detect, locate, and report the location of specific objects in underground environments. There are many excellent works in the challenge [218–223], which have promoted the development of artificial intelligence robots.

6. Conclusion

This paper outlines the theory and framework of LiDAR SLAM technology, discusses several state-of-the-art LiDAR algorithms, and reviews the framework of CSLAM. Then, it focuses on a comprehensive review of the three core sub-fields of multi-robot system boundary detection, task allocation, and map fusion, and summarizes the existing solutions, their characteristics, and applicable scenarios. It also looks forward to the future development of LiDAR SLAM and lists some problems that need to be solved. In general, multi-robot LiDAR SLAM is still an important topic worthy of in-depth research and discussion. The fusion of LiDAR and camera sensors provides strong support for environmental perception, improves the performance of robots and autonomous driving systems in complex dynamic environments, and is also one of the hottest research hotspots in the future.

Author contributions. All authors contributed to the study conception and design. Rui Yuan, Yuheng Gao, and Honghui Tao conducted data gathering. The first draft of the manuscript was written by Rui Yuan.

Financial support. This work was supported by the National Natural Science Youth Foundation Project of China under Grant No.62206254.

Competing interest. The authors declare that no conflicts of interest exist.

Ethical approval. Not applicable.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Rob. Autom. Mag.* **13**(2), 99–110 (2006).

[2] R. Smith, M. Self and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Mach. Intell. Pattern Recogn.* **5**(5), 435–461 (1988). Elsevier.

[3] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Rob. Autom.* **7**(3), 376–382 (1991).

[4] Y. Zhang, S. Zhao and D. An, "An efficient LiDAR-based localization method for self-driving cars in dynamic environments," *Robotica* **40**(1), 38–55 (2022).

[5] Q. Liu, X. Di and B. Xu, "Autonomous vehicle self-localization in urban environments based on 3D curvature feature points–Monte carlo localization," *Robotica* **40**(3), 817–833 (2022).

[6] T. Elmokadem and A. V. Savkin, "A method for autonomous collision-free navigation of a quadrotor UAV in unknown tunnel-like environments," *Robotica* **40**(4), 835–861 (2022).

[7] N. Palomeras, M. Carreras and J. Andrade-Cetto, "Active SLAM for autonomous underwater exploration," *Remote Sens-BASEL* **11**(23), 2827 (2019).

[8] S. Zhang, S. Zhao and D. An, "Visual SLAM for underwater vehicles: A survey," *Comput. Sci. Rev.* **46**, 100510 (2022).

[9] H. Wang, C. Zhang, Y. Song, "Three-dimensional reconstruction based on visual SLAM of mobile robot in search and rescue disaster scenarios," *Robotica* **38**(2), 350–373 (2020).

[10] Y. Wang and W. Zhang, "Research progress on SLAM technology in unstructured lunar complex environments," *J. Zhengzhou Univ. (Eng. Sci.).* **39**(003), 45–50 (2018).

[11] K. Yi and Y. Shen, "A survey on laser SLAM navigation and mobile robot localization algorithms," *Rob. Technol. Appl.* **5**, 25–28 (2019).

[12] Y. Hu, Q. Zhou, Z. Miao, "Outdoor LiDAR-inertial SLAM using ground constraints," *Robotica* **42**(4), 1246–1261 (2024).

[13] K. S. Choi and S. G. Lee, "An enhanced CSLAM for multi-robot based on unscented Kalman filter," *Int. J. Control Autom. Syst.* **10**(1), 102–108 (2012).

[14] W. Liu, "SLAM algorithm for multi-robot communication in unknown environment based on particle filter," *J. Amb. Intell. Hum. Comput.*, 1–9 (2021).

[15] R. Dubois, A. Eudes and V. Frémont, "Sharing visual-inertial data for collaborative decentralized simultaneous localization and mapping," *Rob. Auton. Syst.* **148**, 103933 (2022). Elsevier.

[16] Z. Li, B. He, L. Zhang and J. Zhang, "Multi-robot CSLAM in the Unknown Environment," *IEEE International Conference on Information and Automation (ICIA)* (IEEE, 2018) pp. 45–50.

[17] D. Zou, P. Tan and W. Yu, "Collaborative visual SLAM for multiple agents: A brief survey,"Virtual," *Reality Intell. Hardware* **1**(5), 461–482 (2019).

[18] J. Cheng, L. Zhang, Q. Chen, "A review of visual SLAM methods for autonomous driving vehicles," *Eng. Appl. Artif. Intell.* **114**, 104992 (2022).

[19] K. Wang, G. Zhao and J. Lu, "A deep analysis of visual SLAM methods for highly automated and autonomous vehicles in complex urban environment," *IEEE Trans. Intell. Transp. Syst.* **25**(9), 10524–10541 (2024).

[20] Z. Zheng, K. Su, S. Lin, "Development of vision–based SLAM: From traditional methods to multimodal fusion," *Rob. Intell. Autom.* **44**(4), 529–548 (2024).

[21] L. Huang, "Review on LiDAR-based SLAM Techniques," *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)* (IEEE, 2021) pp. 163–168.

[22] X. Xu, L. Zhang, J. Yang, "A review of multi-sensor fusion SLAM systems based on 3D LIDAR," *Remote Sens-BASEL* **14**(12), 2835 (2022).

[23] X. Yue, Y. Zhang, J. Chen, J. Chen, X. Zhou and M. He, "LiDAR-based SLAM for robotic mapping: State of the art and new frontiers," *Ind. Robot: Int. J. Rob. Res. Appl.* **51**(2), 196–205 (2024).

[24] Y. Rizk, M. Awad and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey," *ACM Comput. Surv. (CSUR)* **52**(2), 1–31 (2019).

[25] X. Zhao, S. Shao, T. Wang, C. Fang, J. Zhang and H. Zhao, "A Review of Multi-robot Collaborative Simultaneous Localization and Mapping," *2023 IEEE International Conference on Unmanned Systems (ICUS)* (2023) pp. 900–905.

[26] K. Aloui, A. Guizani, M. Hammadi, M. Haddar and T. Soriano, "Systematic Literature Review of Collaborative SLAM Applied to Autonomous Mobile Robots," *2022 IEEE Information Technologies & Smart Industrial Systems (ITSIS)* (2022) pp. 1–5.

[27] P.-Y. Lajoie, B. Ramtoula, F. Wu and G. Beltrame, "Towards Collaborative Simultaneous Localization and Mapping: A Survey of the Current Research Landscape," arXiv preprint arXiv:2108.08325 (2021).

[28] Y. Zhang, P. Shi and J. Li, "3D LiDAR SLAM: A survey," *Photogramm. Rec.* **39**(186), 457–517 (2024).

[29] Y. Zhang, P. Shi and J. Li, "LiDAR-based place recognition for autonomous driving: A survey," *ACM Comput. Surv.* **57**(4), 1–36 (2024).

[30] W. Chen, X. Wang, S. Gao, G. Shang, C. Zhou, Z. Li, C. Xu and K. Hu, "Overview of multi-robot collaborative SLAM from the perspective of data fusion," *Machines* **11**(6), 653 (2023).

[31] F. Zhu, Research on Indoor Environment Perception and Map Building for Mobile Robots Based on SLAM *Master's Thesis* (Guangdong University of Technology, 2016).

[32] X. Zhao, Research on Simultaneous Localization and Indoor Map Building Algorithms Based on LIDAR *Master's Thesis* (Harbin Institute of Technology, 2017).

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

[33] T. Hang, Research on Indoor Mobile Robots Based on Ground Feature Navigation *Master's thesis* (Soochow University, 2020).
[34] G. Zhou, Research on SLAM Algorithms for Mobile Robots Based on LIDAR Navigation *Master's thesis* (Zhejiang University, 2021).
[35] Z. Qin, Research on Map Building Method for Mobile Robots by Fusing WiFi and LIDAR *Master's thesis* (Southwest University of Science and Technology, 2022).
[36] S. Li, L. Wang, J. Li, "3D LiDAR/IMU calibration based on continuous-time trajectory estimation in structured environments," *IEEE Access* **9**, 138803–138816 (2021). doi: 10.1109/ACCESS.2021.3117966.
[37] H. Huang, X. Zhang, J. Yang, "Research on sensor fusion-based calibration and real-time point cloud mapping methods for laser liDAR and IMU," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **48**, 1199–1206 (2023).
[38] L. Zhang, L. Zhang, R. Gao, "Indoor mobile robot localization applying IMU/stereo camera/LiDAR and graph-based optimization," *IEEE Sens. J.* **24**(13), 21466–21478 (2024).
[39] Y. Zhang, "LILO: A novel lidar–IMU SLAM system with loop optimization," *IEEE Trans. Aerospace Electron. Syst.* **58**(4), 2649–2659 (2022). doi: 10.1109/TAES.2021.3135234.
[40] M. Brossard and S. Bonnabel, "Learning Wheel Odometry and IMU Errors for Localization," *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019) pp. 291–297. doi: 10.1109/ICRA.2019.8794237.
[41] Y. Wei, Real-Time Localization and Mapping Techniques Based on LIDAR *Master's Thesis* (Harbin Institute of Technology, 2017).
[42] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama and S. Mita, "Lidar Scan Feature for Localization with Highly Precise 3-D Map," *IEEE Intelligent Vehicles Symposium Proceedings* (IEEE, 2014) pp. 1345–1350.
[43] J. Lu, Research on Engineering Applications of SLAM for Mobile Robots Based on LIDAR *Master's thesis* (Shanghai Jiao Tong University, 2020).
[44] W. Wang, Research on Autonomous Navigation Path Planning Algorithm Based on Multi-Sensor Information Fusion *Master's thesis* (Hebei University of Architecture and Engineering, 2022).
[45] Y. Zheng, Simulation Study on Autonomous Navigation Algorithm for Mobile Robots Based on LIDAR SLAM *Master's thesis* (Harbin University of Commerce, 2022).
[46] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.* **9**(2), 61–61 (1988).
[47] P. Kim, J. Chen and Y. K. Cho, "SLAM-driven robotic mapping and registration of 3D point clouds," *Autom. Constr.* **89**, 38–48 (2018). Elsevier.
[48] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer* **22**(6), 46–57 (1989). IEEE.
[49] Y. Cai, F. Kong, Y. Ren, F. Zhu, J. Lin and F. Zhang, "Occupancy grid mapping without ray-casting for high-resolution LiDAR sensors," *IEEE Trans. Rob.* **40**, 172–192 (2023).
[50] M. Shienman, A. Kitanov and V. Indelman, "Ft-bsp: Focused topological belief space planning," *IEEE Rob. Autom. Lett.* **6**(3), 4744–4751 (2021). IEEE.
[51] T. Wang, G. Shang and S. Gao, "Research on multi-robot SLAM map fusion method based on heuristics," *J. Syst. Simul.* **34**(7), 1449–1458 (2022).
[52] Y. Bao, Y. Pan, Z. Yang and R. Huan, "Utilization of semantic planes: Improved localization and dense semantic map for monocular SLAM in urban environment," *IEEE Rob. Autom. Lett.* **6**(3), 6108–6115 (2021). IEEE.
[53] T. Qin, Y. Zheng, T. Chen, Y. Chen and Q. Su, "A Light-Weight Semantic Map for Visual Localization Towards Autonomous Driving," *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021) pp. 11248–11254.
[54] Y. Zhang, G. Tian, X. Shao, S. Liu, M. Zhang and P. Duan, "Building metric-topological map to efficient object search for mobile robot," *IEEE Trans. Ind. Electron.* **69**(7), 7076–7087 (2021). IEEE.
[55] H. Kuang, Y. Li, Y. Zhang, "Research on rapid location method of mobile robot based on semantic grid map in large scene similar environment," *Robotica* **40**(11), 4011–4030 (2022).
[56] S. Wu, Simultaneous Localization and Mapping in Unknown Indoor Environments Using Laser Radar *Master's thesis* (Guangxi University, 2020).
[57] S. Song, SLAM Algorithm Design for Indoor Mobile Robots *Master's thesis* (Harbin Institute of Technology, 2019).
[58] S. Liang, Z. Cao, C. Wang and J. Yu, "A novel 3D LiDAR SLAM based on directed geometry point and sparse frame," *IEEE Rob. Autom. Lett.* **6**(2), 374–381 (2020). IEEE.
[59] P. J. Besl and N. D. McKay, "Method for Registration of 3-D Shapes," **In:** *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. **1611** (SPIE, 1992) pp. 586–606.
[60] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.* **10**(3), 145–155 (1992). Elsevier.
[61] J. Qi, SLAM Algorithm Design and Implementation for Unmanned Vehicle Based on LIDAR *Master's thesis* (Harbin Institute of Technology, 2021).
[62] K. L. Low, "Linear least-squares optimization for point-to-plane ICP surface registration," *Chapel Hill Univ. N. C.* **4**(10), 1–3 (2004).
[63] J. M. Phillips, R. Liu and C. Tomasi, "Outlier Robust ICP for Minimizing Fractional RMSD," *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)* (IEEE, 2007) pp. 427–434.
[64] A. Nicolai, R. Skeele, C. Eriksen and G. A. Hollinger, "Deep Learning for Laser Based Odometry Estimation," *RSS Workshop Limits and Potentials of Deep Learning in Robotics*, vol. **184** (2016) p. 1.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

[65] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," *Rob.: Sci. Syst.* **2**(9), 1–9 (2014). Berkeley, CA.

[66] K. Chiang, Y. Chiu, S. Srinara and M. Tsai, "Performance of LiDAR-SLAM-based PNT with initial poses based on NDT scan matching algorithm," *Satell. Navig.* **4**(1), 3 (2023). Springer.

[67] M. Liu, G. Xu, T. Tang, X. Qian and M. Geng, "A survey of laser SLAM algorithms," *J. Comput. Eng. Appl.* **60**(1) (2024).

[68] M. Magnusson, The Three-Dimensional Normal-Distributions Transform: An Efficient Representation for Registration, Surface Analysis, and Loop Detection *PhD thesis* (Örebro Universitet, 2009).

[69] S. Du, T. Chen, Z. Lou, "A 2D-LiDAR-based localization method for indoor mobile robots using correlative scan matching," *Robotica*. **43**(2), 1–28 (2024).

[70] Y. K. Tee and Y. C. Han, "Lidar-based 2D SLAM for Mobile Robot in an Indoor Environment: A Review," *International Conference on Green Energy, Computing and Sustainable Technology (GECOST)* (IEEE, 2021) pp. 1–7.

[71] I. Grooms, "A comparison of nonlinear extensions to the ensemble Kalman filter: Gaussian anamorphosis and two-step ensemble filters," *Comput. Geosci.* **26**(3), 633–650 (2022). Springer.

[72] Y. Fang, A. Panah, J. Masoudi, B. Barzegar and S. Fatehi, "Adaptive unscented Kalman filter for robot navigation problem (Adaptive unscented Kalman filter using incorporating intuitionistic fuzzy logic for concurrent localization and mapping)," *IEEE Access* **10**, 101869–101879 (2022). IEEE.

[73] R. Grabowski, P. Khosla and H. Choset, "Autonomous Exploration via Regions of Interest," *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 2 (IEEE, 2003) pp. 1691–1696.

[74] K. Ebadi, L. Bernreiter, H. Biggie, "Present and Future of SLAM in Extreme Underground Environments," arXiv preprint arXiv:2208.01787 (2022).

[75] G. Grisetti, C. Stachniss and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Trans. Rob.* **23**(1), 34–46 (2007).

[76] S. Kohlbrecher, O. Von Stryk, J. Meyer and U. Klingauf, "A Flexible and Scalable SLAM System with full 3D Motion Estimation," *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (IEEE, Kyoto, Japan, 2011) pp. 155–160.

[77] K. Konolige, "Efficient Sparse Pose Adjustment for 2D Mapping," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2010).

[78] W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-Time Loop Closure in 2D LIDAR SLAM," *2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEEStockholm, Sweden, 2016) pp. 1271–1278.

[79] J. G. Rogers, J. M. Gregory, J. Fink and E. Stump, "Test Your SLAM! The subT-Tunnel Dataset and Metric for Mapping," *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020) pp. 955–961. doi: 10.1109/ICRA40945.2020.9197156.

[80] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized LiDAR Odometry and Mapping on Variable Terrain," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, Madrid, 2018) pp. 4758–4765.

[81] D. Fasiolo, L. Tiozzo, L. Scalera and E. Maset, "Comparing LiDAR and IMU-based SLAM approaches for 3D robotic mapping," *Robotica* **41**(9), 2588–2604 (2023).

[82] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti and D. Rus, "LIO-SAM: Tightly-Coupled Lidar Inertial Odometry via Smoothing and Mapping," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, Las Vegas, NV, USA, 2020) pp. 5135–5142.

[83] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Rob. Autom. Lett.* **6**(2), 3317–3324 (2021).

[84] W. Xu, Y. Cai, D. He, "Fast-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Rob.* **38**(4), 2053–2073 (2022).

[85] A. Reinke, M. Palieri, B. Morrell, "Locus 2.0: Robust and computationally efficient LiDAR odometry for real-time 3D mapping," *IEEE Rob. Autom. Lett.* **7**(4), 9043–9050 (2022).

[86] S. Saeedi, M. Trentini, M. Seto, "Multiple-robot simultaneous localization and mapping: A review," *J. Field Rob.* **33**(1), 3–46 (2016).

[87] P. Schmuck and M. Chli, "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *J. Field Rob.* **36**(4), 763–781 (2019).

[88] J. Xie, X. He, J. Mao, "C2VIR-SLAM: Centralized collaborative visual-inertial-range simultaneous localization and mapping," *Drones* **6**(11), 312 (2022).

[89] A. Saha, B. C. Dhara, S. Umer, "Corb2i-SLAM: An adaptive collaborative visual-inertial SLAM for multiple robots," *Electronics* **11**(18), 2814 (2022).

[90] Y. Chang, K. Ebadi, C. E. Denniston, "LAMP 2.0: A robust multi-robot SLAM system for operation in challenging large-scale underground environments," *IEEE Rob. Autom. Lett.* **7**(4), 9175–9182 (2022).

[91] E. Liu, H. Li, S. Li, "Centralized Multi-Robot Collaborative LiDAR SLAM Utilizing Loop Closure Selection," *2023 China Automation Congress (CAC)* (IEEE, 2023) pp. 463–468.

[92] B. A. Jalil and I. K. Ibraheem, "Enhancing multi-robot SLAM: Centralized liDAR-based loop closure detection approach," *Iraqi J. Comput. Sci. Math.* **5**(4), 24 (2024).

[93] S. Zhong, H. Chen, Y. Qi, "Colrio: Lidar-Ranging-Inertial Centralized State Estimation for Robotic Swarms," *2024 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2024) pp. 3920–3926.

[94] Y. Xie, Y. Zhang, L. Chen, "RDC-SLAM: A real-time distributed cooperative SLAM system based on 3D liDAR," *IEEE Trans. Intell. Transp.* **23**(9), 14721–14730 (2021).

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

[95] Y. Huang, T. Shan, F. Chen, "Disco-slam: Distributed scan context-enabled multi-robot lidar slam with two-stage global-local graph optimization," *IEEE Rob. Autom. Lett.* **7**(2), 1150–1157 (2021).

[96] H. Xie, D. Zhang, X. Hu, "Autonomous multirobot navigation and cooperative mapping in partially unknown environments," *IEEE Trans. Instrum. Meas.* **72**, 1–12 (2023).

[97] P. Y. Lajoie, B. Ramtoula, Y. Chang, "DOOR-SLAM: Distributed, online, and outlier resilient SLAM for robotic teams," *IEEE Rob. Autom. Lett.* **5**(2), 1656–1663 (2020).

[98] S. Zhong, Y. Qi, Z. Chen, "Dcl-slam: A distributed collaborative lidar slam framework for a robotic swarm," *IEEE Sens. J.* **24**(4), 4786–4797 (2023).

[99] P. Y. Lajoie and G. Beltrame, "Swarm-SLAM: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems," *IEEE Rob. Autom. Lett.* **9**(1), 475–482 (2023).

[100] S. Ahlström and M. Schmidt, "Collaborative SLAM for ground robotics in search and rescue missions (2024).

[101] D. Liu, "A data association algorithm for SLAM based on central difference joint compatibility criterion and clustering," *Robotica* **39**(9), 1674–1691 (2021).

[102] C. Li, G. Guo, P. Yi and Y. Hong, "Distributed pose-graph optimization with multi-level partitioning for multi-robot SLAM," *IEEE Rob. Autom. Lett.* **9**(6), 4926–4933 (2024). doi: 10.1109/LRA.2024.3382531.

[103] K. He, R. Jia, H. Hong, "LDG-CSLAM: Multi-robot collaborative SLAM based on curve analysis, normal distribution, and factor graph optimization," *J. Field Rob.* **42**(5), 2173–2191 (2025).

[104] Y. Shin, Y. Park and A. Kim, "Direct Visual SLAM Using Sparse Depth for Camera-LiDAR System," *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21-25 May 2018*, Brisbane, QLD, Australia (2018) pp. 5144–5151.

[105] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic SLAM for multi-robot systems," *IEEE Trans. Rob.* **38**(4), 2022–2038 (2022). doi: 10.1109/TRO.2021.3137751.

[106] P. Liu, Y. Bi, J. Shi, T. Zhang and C. Wang, "Semantic-assisted LiDAR tightly coupled SLAM for dynamic environments," *IEEE Access* **12**, 34042–34053 (2024). doi: 10.1109/ACCESS.2024.3369183.

[107] J. Yan, X. Lin, Z. Ren, "Mui-tare: Cooperative multi-agent exploration with unknown initial position," *IEEE Rob. Autom. Lett.* **8**(7), 4299–4306 (2023).

[108] Y. Tian, K. Khosoussi, M. Giamou, "Near-optimal budgeted data exchange for distributed loop closure detection. arXiv preprint, arXiv:1806.00188 (2018).

[109] H. Wang, C. Wang and L. Xie, "Intensity Scan Context: Coding Intensity and Geometry Relations for Loop Closure Detection," *2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France* (2020). doi: 10.1109/ICRA40945.2020.9196764.

[110] D. Zhu, G. Xu, X. Wang, X. Liu and D. Tian, "PairCon-SLAM: Distributed, online, and real-time RGBD-SLAM in large scenarios," *IEEE Trans. Instrum. Meas.* **70**, 1–14 (2021).

[111] T. Shan, B. Englot, F. Duarte, C. Ratti and D. Rus, "Robust Place Recognition Using an Imaging LiDAR," *2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China* (2021) pp. 5469–5475. doi: 10.1109/ICRA48506.2021.9562105.

[112] Z. Chen, J. Zhao, T. Feng, "Robust loop closure selection based on inter-robot and intra-robot consistency for multi-robot map fusion," *Remote Sens-BASEL* **15**(11), 19 (2023). doi: 10.3390/rs15112796.

[113] K. Ebadi, L. Bernreiter, H. BiggiePresent and future of SLAM in extreme environments: The DARPA SubT challenge," *IEEE Trans. Rob.* **40**, 936–959 (2023).

[114] G. Shin, H. Sim, S. Nam, Y. Kim, J. Heo and K.-K. K. Kim, "Multi-robot relative pose estimation in SE(2) with observability analysis: A comparison of extended Kalman filtering and robust pose graph optimization," *IEEE Trans. Intell. Veh.*, 1–23 (2024). doi: 10.1109/TIV.2024.3478290.

[115] R. Bai, S. Yuan, H. Guo, P. Yin, W.-Y. Yau and L. Xie, "Multi-robot Active Graph Exploration with Reduced Pose-SLAM Uncertainty via Submodular Optimization," *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2024) pp. 10229–10236. doi: 10.1109/IROS58592.2024.10802691.

[116] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen and F. Dellaert, "Distributed Trajectory Estimation with Privacy and Communication Constraints: A Two-Stage Distributed Gauss-Seidel Approach," *2016 IEEE International Conference on Robotics and Automation (ICRA)* (2016) pp. 5261–5268. doi: 10.1109/ICRA.2016.7487736.

[117] S. Choudhary, L. Carlone, C. Nieto, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *Int. J. Rob. Res.* **36**(12), 1286–1311 (2017).

[118] T. Fan and T. D. Murphey, "Majorization minimization methods for distributed pose graph optimization," *IEEE Trans. Rob.* **40**, 22–42 (2024). doi: 10.1109/TRO.2023.3324818.

[119] J. Oršulić, D. Miklić and Z. Kovačić, "Efficient dense frontier detection for 2-D graph slam based on occupancy grid submaps," *IEEE Rob. Autom. Lett.* **4**(4), 3569–3576 (2019). IEEE.

[120] B. Yamauchi, "A Frontier-based Approach for Autonomous Exploration," *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97: Towards New Computational Principles for Robotics and Automation* (1997) pp. 146–151.

[121] M. Keidar and G. A. Kaminka, "Efficient frontier detection for robot exploration," *Int. J. Rob. Res.* **33** (2), 215–236 (2014). SAGE Publications.

[122] M. Keidar and G. A. Kaminka, "Robot Exploration with Fast Frontier Detection: Theory and Experiments," *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1* (2012) pp. 113–120.

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

[123] P. Quin, A. Alempijevic, G. Paul and D. Liu, "Expanding Wavefront Frontier Detection: An Approach for Efficiently Detecting Frontier Cells," *Australasian Conference on Robotics and Automation, ACRA* (2014).
[124] P. Quin, D. D. K. Nguyen, T. L. Vu, "Approaches for efficiently detecting frontier cells in robotics exploration," *Front. Rob. AI* **8**, article616470 (2021).
[125] Z. Sun, B. Wu, C. Xu, S. E. Sarma, J. Yang and H. Kong, "Frontier Detection and Reachability Analysis for Efficient 2D Graph-SLAM Based Active Exploration," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2020) pp. 2051–2058.
[126] H. Umari and S. Mukhopadhyay, "Autonomous Robotic Exploration Based on Multiple Rapidly-Exploring Randomized Trees," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017) pp. 1396–1402.
[127] J. M. Pimentel, M. S. Alvim, M. F. Campos and D. G. Macharet, "Information-driven rapidly-exploring random tree for efficient environment exploration," *J. Intell. Rob. Syst.* **91**(2), 313–331 (2018). Springer.
[128] L. Zhang, Z. Lin, J. Wang and B. He, "Rapidly-exploring random trees multi-robot map exploration under optimization framework," *Rob. Auton. Syst.* **131**, 103565 (2020). Elsevier.
[129] W. Qiao, Z. Fang and B. Si, "A sampling-based multi-tree fusion algorithm for frontier detection," *Int. J. Adv. Rob. Syst.* **16**(4) (2019). doi: 10.1177/1729881419865427. SAGE Publications.
[130] W. Qiao, Z. Fang and B. Si, "Sample-based Frontier Detection for Autonomous Robot Exploration," *IEEE International Conference on Robotics and Biomimetics (ROBIO)* (IEEE, 2018) pp. 1165–1170.
[131] G. Xu, L. Zhang, M. Chen, "Hybrid Frontier Detection Strategy for Autonomous Exploration in Multi-obstacles Environment," *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (IEEE, 2021) pp. 1909–1915.
[132] Z. Sun, B. Wu, C. Xu, "Ada-Detector: Adaptive Frontier Detector for Rapid Exploration," *2022 International Conference on Robotics and Automation (ICRA)* (IEEE, 2022) pp. 3706–3712.
[133] J. Liu, Y. Lv, Y. Yuan, W. Chi, G. Chen and L. Sun, "An efficient robot exploration method based on heuristics biased sampling," *IEEE Trans. Ind. Electron.* **70**(7), 7102–7112 (2022). IEEE.
[134] Y. Ning, T. Li, C. Yao, W. Du and Y. Zhang, "HMS-RRT: A novel hybrid multi-strategy rapidly-exploring random tree algorithm for multi-robot collaborative exploration in unknown environments," *Expert Syst. Appl.* **247**, 123238 (2024). Elsevier.
[135] A. Mete, M. Mouhoub and A. Farid, "Coordinated Multi-Robot Exploration using Reinforcement Learning," *2023 International Conference on Unmanned Aircraft Systems (ICUAS)* (2023) pp. 265–272.
[136] K. Leong, "Reinforcement Learning with Frontier - Based Exploration via Autonomous Environment," arXiv preprint arXiv:2307.07296 (2023).
[137] R. Wang, J. Zhang, M. Lyu, "An improved frontier - based robot exploration strategy combined with deep reinforcement learning," *Rob. Auton. Syst.* **181**, 104783 (2024).
[138] A. Khamis, A. Hussein and A. Elmogy, "Multi-Robot Task Allocation: A Review of the State-of-the-Art," **In:** *Cooperative Robots and Sensor Networks 2015* (2015) pp. 31–51. Springer.
[139] I. Scheler and R. Dietrich, "coExplore: Combining Multiple Rankings for Multi-robot Exploration," arXiv preprint arXiv:2303.17459 (2023).
[140] W. Gao, M. Booker, A. Adiwahono, M. Yuan, J. Wang and Y. Yun, "An Improved Frontier-based Approach for Autonomous Exploration," *15th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (IEEE, 2018) pp. 292–297.
[141] Y. Sun and C. Zhang, "Efficient and safe robotic autonomous environment exploration using integrated frontier detection and multiple path evaluation," *Remote Sens-BASEL* **13**(23), article 4881 (2021).
[142] S. J. Fusic and R. Sitharthan, "Improved RRT* algorithm-based path planning for unmanned aerial vehicle in a 3D metropolitan environment," *Unmanned Syst.* **12**(05), 859–875 (2024).
[143] J. Fusic and R. Sitharthan, "Self-adaptive learning particle swarm optimization-based path planning of mobile robot using 2D Lidar environment," *Robotica* **42**(4), 977–1000 (2024).
[144] B. Hichri, A. Gallala, F. Giovannini, "Mobile robots path planning and mobile multirobots control: A review," *Robotica* **40**(12), 4257–4270 (2022).
[145] M. Reda, A. Onsy, A. Y. Haikal, "Path planning algorithms in the autonomous driving system: A comprehensive review," *Rob. Auton. Syst.* **174**, 104630 (2024).
[146] C. Sarkar, H. S. Paul and A. Pal, "A Scalable Multi-robot Task Allocation Algorithm," *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018) pp. 5022–5027.
[147] J. G. Martin, F. J. Muros, J. M. Maestre and E. F. Camacho, "Multi-robot task allocation clustering based on game theory," *Rob. Auton. Syst.* **161**, 104314 (2023). Elsevier.
[148] F. Quinton, C. Grand and C. Lesire, "Market approaches to the multi-robot task allocation problem: A survey," *J. Intell. Rob. Syst.* **107**(2), 29 (2023).
[149] H. Chakraa, F. Guérin, E. Leclercq and D. Lefebvre, "Optimization techniques for multi-robot task allocation problems: Review on the state-of-the-art," *Rob. Auton. Syst.*, **168**,104492 (2023).
[150] N. Seenu, R. M. K.C., M. Ramya and M. N. Janardhanan, "Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems," *Ind. Robot: Int. J. Rob. Res. Appl.* **47**(6), 929–942 (2020).
[151] T. Yu, B. Deng, W. Yao, X. Zhu and J. Gui, "Market-based Robots Cooperative Exploration in Unknown Indoor Environment," *2021 IEEE International Conference on Unmanned Systems (ICUS)* (IEEE, 2021) pp. 414–419.
[152] M. De Ryck, D. Pissoort, T. Holvoet and E. Demeester, "Decentral task allocation for industrial agv-systems with routing constraints," *J. Manuf. Syst.* **62**, 135–144 (2022).

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

[153] X. Kong, Y. Gao, T. Wang, J. Liu and W. Xu, "Multi-robot Task Allocation Strategy Based on Particle Swarm Optimization and Greedy Algorithm," *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* (IEEE, 2019) pp. 1643–1646.

[154] C. Wei, Z. Ji and B. Cai, "Particle swarm optimization for cooperative multi-robot task allocation: A multi-objective approach," *IEEE Rob. Autom. Lett.* **5**(2), 2530–2537 (2020).

[155] X. Qin, Q. Zong, X. Li, B. Zhang and X. Zhang, "Multi-robot task allocation based on improved ant colony algorithm," *Space Control Technol. Appl.* **44**(5), 55–59 (2018).

[156] N. Lindsay, R. K. Buehling and L. Sun, "A sequential task addition distributed assignment algorithm for multi-robot systems," *J. Intell. Rob. Syst.* **102**(2), 51 (2021).

[157] Z. Miao, W. Huang, Q. Jiang and Q. Fan, "A novel multimodal multi-objective optimization algorithm for multi-robot task allocation," *Trans. Inst. Meas. Control* (2023). doi: 10.1177/01423312231183588. SAGE Publications.

[158] N. Kang, Z. Miao, Q. K. Pan, "Multi-objective teaching-learning-based optimizer for a multi-weeding robot task assignment problem," *Tsinghua Sci. Technol.* **29**(5), 1249–1265 (2024).

[159] H. A. Kurdi, E. Aloboud, M. Alalwan, S. Alhassan, E. Alotaibi, G. Bautista and J. P. How, "Autonomous task allocation for multi-uav systems based on the locust elastic behavior," *Appl. Soft. Comput.* **71**, 110–126 (2018).

[160] F. Aljalaud and H. A. Kurdi, "Autonomous task allocation for multi-uav systems based on area-restricted search behavior in animals," *Procedia Comput. Sci.* **191**, 246–253 (2021).

[161] W. P. N. dos Reis, G. L. Lopes and G. S. Bastos, "An arrovian analysis on the multi-robot task allocation problem: Analyzing a behavior-based architecture," *Rob. Auton. Syst.* **144**(9), 103839 (2021).

[162] R. J. Alitappeh and K. Jeddisaravi, "Multi-robot exploration in task allocation problem," *Appl. Intell.* **52**(2), 2189–2211 (2022).

[163] B. Park, C. Kang and J. Choi, "Cooperative multi-robot task allocation with reinforcement learning," *Appl. Sci.* **12**(1), 272 (2021).

[164] H. Xie, D. Yang, L. Xiao, "Connectivity-aware 3D UAV path design with deep reinforcement learning," *IEEE Trans. Veh. Technol.* **70**(12), 13022–13034 (2021).

[165] J. Zhang, J. Ren, Y. Cui, "Multi-USV task planning method based on improved deep reinforcement learning," *IEEE Internet Things J.* **11**(10), 18549–18567 (2024).

[166] B. Jin, Y. Sun, W. Wu, "Deep reinforcement learning and ant colony optimization supporting multi-UGV path planning and task assignment in 3D environments," *IET Intell. Transp. Syst.* **18**(9), 1652–1664 (2024).

[167] Y. Sun, R. Sun, S. Yu and Y. Peng, "A Grid Map Fusion Algorithm Based On Maximum Common Subgraph," *2018 13th World Congress On Intelligent Control and Automation (WCICA)* (IEEE, 2018) pp. 58–63.

[168] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. **2** (IEEE, 1999).

[169] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understanding* **110**(3), 346–359 (2008).

[170] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," *2011 International Conference on Computer Vision* (IEEE, 2011) pp. 2564–2571.

[171] Z. Qu, W. Bu and L. Liu, "The algorithm of seamless image mosaic based on A-KAZE features extraction and reducing the inclination of image," *IEEJ Trans. Electr. Electron. Eng.* **13**(1), 134–146 (2018).

[172] S. Leutenegger, M. Chli and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," *2011 International Conference on Computer Vision* (IEEE, 2011).

[173] A. Alahi, R. Ortiz and P. Vandergheynst, "Freak: Fast Retina Keypoint," *2012 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2012) pp. 510–517.

[174] O. Chum and J. Matas, "Optimal randomized RANSAC," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(8), 1472–1482 (2008).

[175] S. Thrun, "A probabilistic on-line mapping algorithm for teams of mobile robots," *Int. J. Rob. Res.* **20**(5), 335–363 (2001).

[176] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *Int. J. Rob. Res.* **25**(12), 1243–1256 (2006).

[177] H. Lee, B. Roh and B. Lee, "Multi-hypothesis map merging with sinogram-based PSO for multi-robot systems," *Electron. Lett.* **52**(14), 1213–1214 (2016).

[178] S. H. Jazi, "Map-merging using maximal empty rectangles in a multi-robot slam process," *J. Mech. Sci. Technol.* **34**(6), 2573–2583 (2020).

[179] B. Chen, S. Li, H. Zhao and L. Liu, "Map merging with suppositional box for multi-robot indoor mapping," *Electronics* **10a**(7), 815 (2021a).

[180] N. Stathoulopoulos, A. Koval, A.-a. Agha-mohammadi and G. Nikolakopoulos, "Frame: Fast and Robust Autonomous 3D Point Cloud Map-Merging for Egocentric Multi-robot Exploration," *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023) pp. 3483–3489.

[181] L. Gui, C. Zeng, J. Luo, "Graph-based robust 3D point cloud map merging approach for large scale," *Cluster Comput.* **28**(1), 1–15 (2025).

[182] H. Wei, R. Li, Y. Cai, C. Yuan, Y. Ren, Z. Zou, H. Wu, C. Zheng, S. Zhou, K. Xue and F. Zhang, "Large-scale multi-session point-cloud map merging," *IEEE Rob. Autom. Lett.* **10**(1), 88–95 (2025). doi: 10.1109/LRA.2024.3504317.

[183] L. Wijayathunga, A. Rassau and D. Chai, "Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: A review," *Appl. Sci.* **13**(17), 9877 (2023).

[184] C. Min, S. Si, X. Wang, "Autonomous Driving in Unstructured Environments: How Far Have We Come?," arXiv preprint arXiv:2410.07701 (2024).

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

[185] X. Liu, S. Wen, H. Liu and F. R. Yu, "CPL-SLAM: Centralized collaborative multi-robot visual-inertial SLAM using point-and-line features," *IEEE Internet Things J.*, 1–1 (2025). doi: 10.1109/JIOT.2025.3548958.
[186] J. Bayer, P. Cížek and J. Faigl, "Autonomous multi-robot exploration with ground vehicles in DARPA subterranean challenge finals," *Field Rob.* **3**, 266–300 (2023). doi: 10.55417/fr.2023008.
[187] K. Park, S. Kim and K. Sohn, "High-Precision Depth Estimation with the 3D Lidar and Stereo Fusion," *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018).
[188] Y. Chen, W. Chen, L. Zhu, "A study of sensor-fusion mechanism for mobile robot global localization," *Robotica* **37**(11), 1835–1849 (2019).
[189] G. Lu, "SLAM Based on Camera-2D LiDAR Fusion," *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2024) pp. 16818–16825.
[190] G. Best, R. Garg, J. Keller, "Multi-robot, multi-sensor exploration of multifarious environments with full mission aerial autonomy," *Int. J. Rob. Res.* **43**(4), 485–512 (2024).
[191] J. Yin, "M2DGR: A multi-sensor and multi-scenario slam dataset for ground robots," *IEEE Rob. Autom. Lett.* **7**(2), 2266–2273 (2021).
[192] H. Wei, "Fusionportablev2: A unified multi-sensor dataset for generalized slam across diverse platforms and scalable environments," *Int. J. Rob. Res.* (2024). doi: 10.1177/02783649241303525,
[193] D. Feng, "S3E: A multi-robot multimodal dataset for collaborative SLAM," *IEEE Rob. Autom. Lett.* **9**(12), 11401–11408 (2024).
[194] J. Kim, "DiTer^{++}: Diverse Terrain and Multi-modal Dataset for Multi-Robot SLAM in Multi-session Environments," arXiv preprint arXiv:2412.05839 (2024).
[195] C. Yao, Y. Ge, G. Shi, "Tail: A terrain-aware multi-modal slam dataset for robot locomotion in deformable granular environments," *IEEE Rob. Autom. Lett.* **9**(7), 6696–6703 (2024).
[196] L. Kenye and R. Kala, "Improving RGB-D SLAM in dynamic environments using semantic aided segmentation," *Robotica* **40**(6), 2065–2090 (2022).
[197] R. Long, C. Rauch, T. Zhang, "Rigidfusion: Robot localisation and mapping in environments with large dynamic rigid objects," *IEEE Rob. Autom. Lett.* **6**(2), 3703–3710 (2021).
[198] R. Long, C. Rauch, V. Ivan, "RGB-D-inertial SLAM in Indoor Dynamic Environments with Long-Term Large Occlusion," arXiv preprint arXiv:2303.13316 (2023).
[199] Y. Wang, Y. Tian, J. Chen, "A survey of visual SLAM in dynamic environment: The evolution from geometric to semantic approaches," *IEEE Trans. Instrum. Meas.* **73**, 1–21 (2024).
[200] W. Wu, L. Guo, H. Gao, "YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint," *Neural Comput. Appl.*, **34**(8), 6011–6026 (2022).
[201] Y. Li, H. Shen, Y. Fu, "A method of dense point cloud SLAM based on improved YOLOV8 and fused with ORB-SLAM3 to cope with dynamic environments," *Expert Syst. Appl.* **255**, 124918 (2024).
[202] D. Kong, X. Li, Q. Xu, "SC_LPR: Semantically consistent LiDAR place recognition based on chained cascade network in long-term dynamic environments," *IEEE Trans. Image Process.* **33**, 2145–2157 (2024).
[203] X. Wang, S. Zheng, X. Lin, "Improving RGB-D SLAM accuracy in dynamic environments based on semantic and geometric constraints," *Measurement* **217**, 113084 (2023).
[204] Y. Fan, Q. Zhang, Y. Tang, "Blitz-SLAM: A semantic SLAM in dynamic environments," *Pattern Recogn.* **121**, 108225 (2022).
[205] T. Ji, C. Wang and L. Xie, "Towards Real-Time Semantic RGB-D SLAM in Dynamic Environments," *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021) pp. 11175–11181.
[206] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Trans. Rob.* **39**(3), 1686–1705 (2023).
[207] Y. Zhang, L. Chen and Q. Xu, "An Improved Frontier Point Exploration System for Mobile Robots Based on LiDAR SLAM," *IEEE International Conference on Development and Learning (ICDL)* (IEEE, 2023) pp. 282–287.
[208] M. Alcalde, M. Ferreira, P. González, "DA-SLAM: Deep Active SLAM Based on Deep Reinforcement Learning," *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)* (IEEE, 2022) pp. 282–287.
[209] K. J. Singh, D. S. Kapoor, K. Thakur, "Map making in social indoor environment through robot navigation using active SLAM," *IEEE Access* **10**, 134455–134465 (2022).
[210] C. Pang, L. Zhou and X. Huang, "A low-cost 3D SLAM system integration of autonomous exploration based on fast-ICP enhanced LiDAR-inertial odometry," *Remote Sens-BASEL* **16**(11), 1979 (2024).
[211] Q. Chen, G. Li, X. Xue and J. Pu, "Multi-LIO: A Lightweight Multiple LiDAR-Inertial Odometry System," *2024 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2024).
[212] L. Yin, "Decentralized Swarm Trajectory Generation for Lidar-based Aerial Tracking in Cluttered Environments," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2023).
[213] X. Zhou, J. Zhu, H. Zhou, C. Xu and F. Gao, "Ego-Swarm: A Fully Autonomous and Decentralized Quadrotor Swarm System in Cluttered Environments," *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2021) pp. 4101–4107. doi: 10.1109/ICRA48506.2021.9561902.
[214] J. Park, D. Kim, G. C. Kim, D. Oh and H. J. Kim, "Online distributed trajectory planning for quadrotor swarm with feasibility guarantee using linear safe corridor," *IEEE Rob. Autom. Lett.* **7**, 4869–4876 (2022).

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

[215] F. Zhu, Y. Ren, F. Kong, H. Wu, S. Liang, N. Chen, W. Xu and F. Zhang, "Swarm-LIO: Decentralized Swarm LiDAR-Inertial Odometry," *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2023) pp. 3254–3260. doi: 10.1109/ICRA48891.2023.10161355.
[216] F. Zhu, Y. Ren, L. Yin, F. Kong, Q. Liu, R. Xue, W. Liu, Y. Cai, G. Lu, H. Li and F. Zhang, "Swarm-LIO2: Decentralized efficient liDAR-inertial odometry for aerial swarm systems," *IEEE Trans. Rob.* **41**, 960–981 (2025). doi: 10.1109/TRO.2024.3522155.
[217] T. H. Chung, V. Orekhov and A. Maio, "Into the robotic depths: Analysis and insights from the darpa subterranean challenge," *Ann. Rev. Control Rob. Auton. Syst.* **6**(1), 477–502 (2023).
[218] T. Rouček, M. Pecka, P. ČíŽek, "Darpa Subterranean Challenge: Multi-Robotic Exploration of Underground Environments," *Modelling and Simulation for Autonomous Systems: 6th International Conference, MESAS 2019, Palermo, Italy, October 29-31, 2019, Revised Selected Papers 6* (Springer International Publishing, 2019) pp. 274–290.
[219] M. Petrlik, P. Petracek, V. Kratky, "UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT," arXiv preprint, arXiv:2206.08185 (2022).
[220] A. Koval, C. Kanellakis and G. Nikolakopoulos, "Evaluation of lidar-based 3D SLAM algorithms in subT environment," *IFAC-PapersOnLine* **55**(38), 126–131 (2022).
[221] M. Tranzatto, T. Miki, M. Dharmadhikari, "CERBERUS in the DARPA subterranean challenge," *Sci. Rob.* **7**(66), article eabp9742 (2022).
[222] A. Agha, K. Otsu, B. Morrell, "NeBula: TEAM coSTAR's robotic autonomy solution that won phase II of DARPA subterranean challenge," *Field Rob.* **2**, 1432–1506 (2022).
[223] K. Ebadi, L. Bernreiter, H. Biggie, "Present and Future of SLAM in Extreme Underground Environments," arXiv preprint, arXiv:2208.01787 (2022).
[224] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-Time," *Rob.: Sci. Syst.* **2**(9), 1–9 (2014).
[225] S. Choudhury, J. K. Gupta, M. J. Kochenderfer, D. Sadigh and J. Bohg, "Dynamic multi-robot task allocation under uncertainty and temporal constraints," *Auton. Rob.* **46**(1), 231–247 (2022).
[226] C. A. V. Hernández and F. A. P. Ortiz, "A real-time map merging strategy for robust collaborative reconstruction of unknown environments," *Expert Syst. Appl.* **145**, 113109 (2020).
[227] Z. Lin, J. Zhu, Z. Jiang, Y. Li, Y. Li and Z. Li, "Merging grid maps in diverse resolutions by the context-based descriptor," *ACM Trans. Internet Technol. (TOIT)* **21**(4), 1–21 (2021).
[228] L. Zhang, C. Jiao, J. Huang and X. Su, "AKAZE Feature-Based Map Merging for Multi-Robot SLAM with Unknown Initial Pose," *2022 34th Chinese Control and Decision Conference (CCDC)* (IEEE, 2022) pp. 5637–5642.
[229] J. McConnell, Y. Huang, P. Szenher, "DRACo-SLAM: Distributed Robust Acoustic Communication-Efficient SLAM for Imaging Sonar Equipped Underwater Robot Teams," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2022) pp. 8457–8464.
[230] Y. Rizk, M. Awad and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey," *ACM Comput. Surv. (CSUR)* **52**(2), 1–31 (2019). ACM New York, NY, USA.
[231] H. Lee, "Tomographic feature-based map merging for multi-robot systems," *Electronics* **9**(1), 107 (2020).
[232] W. Chen, J. Sun and Q. Zheng, "Fast loop closure selection method with spatiotemporal consistency for multi-robot map fusion," *Appl. Sci.* **12**(11), 5291 (2022).
[233] H. Dong, J. Yu, Y. Xu, Z. Xu, Z. Shen, J. Tang, Y. Shen and Y. Wang, "Mr-gmmapping: Communication efficient multi-robot mapping system via gaussian mixture model," *IEEE Rob. Autom. Lett.* **7**(2), 3294–3301 (2022).
[234] T. Ma, G. Jiang, Y. Ou, "Semantic geometric fusion multi-object tracking and lidar odometry in dynamic environment," *Robotica* **42**(3), 891–910 (2024).

Appendix

This article mentions a large number of abbreviations. For ease of reference and consistency, we summarize the term abbreviations in Table VIII.

Table VIII. *Glossary of terms cited.*

Full terms	**Abbreviation**
Simultaneous Localization and Mapping	SLAM
Unmanned Aerial Vehicles	UAVs
Autonomous Underwater Vehicle	AUVs
Unmanned Surface Vessel	USV
Light Detection and Ranging	LiDAR

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press

Table VIII. *Continued.*

Full terms	Abbreviation
Multi Robot System	MRS
Inertial Measurement Unit	IMU
Time of Flight	ToF
Iterative closest point	ICP
Normal Distribution Transform	NDT
Correlation Scan Matching	CSM
Kalman Filter	KF
Extended Kalman Filter	EKF
Unscented Kalman Filter	UKF
Particle Filter	PF
Expectation Maximization	EM
Unscented Transform	UT
Laser Odometry and Mapping	LOAM
Lightweight and Ground-Optimized LiDAR Odometry and Mapping	LeGO-LOAM
LiDAR Inertial Odometry via Smoothing and Mapping	LIO-SAM
Fast Direct LiDAR-Inertial Odometry	FAST-LIO
Generalized Iterative Closest Point	GICP
Maximum Likelihood Estimation	MLE
Pose Graph Optimization	PGO
Enhanced Multi-Robot Fast Localization, Ranging, and Mapping	EMR-FLOAM
Wavefront Frontier Detector	WFD
Fast Frontier Detector	FFD
Frontier-Tracing Frontier Detection	FTFD
Rapid Random Tree	RRT
Information Gain RRT	ID-RRT
Sample-based Multi-tree Fusion	SMF
Multi-Robot Task Allocation	MRTA
Information Theory	IT
Deep Reinforcement Learning	DRL
Scale-Invariant Feature Transform	SIFT
Speeded Up Robust Features	SURF
Oriented FAST and Rotated BRIEF	ORB
Accelerated-KAZE	AKAZE
Random Sample Consensus	RANSAC
Maximum Empty Rectangle	MER
Maximum Common Subgraph	MCS
Modified Local Difference Binary	MLDB
Point Cloud Maps	PCM
Binary Triangle Combination	BTC
Depth-First Search	DFS
Advanced Research Projects Agency	DARPA

Cite this article: R. Yuan, B. Ji, Y. Gao and H. Tao (2025). "A review of LiDAR simultaneous localization and mapping techniques for multi-robot", Robotica **43**, 3200–3240. https://doi.org/10.1017/S0263574725102178

https://doi.org/10.1017/S0263574725102178 Published online by Cambridge University Press