

Review

Not peer-reviewed version

A Review of Simultaneous Localization and Mapping Algorithms Based on Lidar

[Yong Li](#)*, [Jiexin An](#), Na He, Yanbo Li, Zhenyu Han, Zishan Chen, Yaping Qu

Posted Date: 28 November 2024

doi: 10.20944/preprints202411.2260.v1

Keywords: simultaneous localization and mapping; lidar; multi-sensor fusion; deep learning; development tendency



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Review

A Review of Simultaneous Localization and Mapping Algorithms Based on Lidar

Yong Li ^{1,*}, Jiexin An ¹, Na He ², Yanbo Li ^{3,4}, Zhenyu Han ³, Zishan Chen ³ and Yaping Qu ⁵

¹ Automotive Engineering Research Institute, Jiangsu University, 301 Xuefu Road, Zhenjiang 212013, China

² Beijing Institute of Space Launch Technology, Beijing 100076, China

³ Lingyun Industrial Co. Ltd, Shanghai, 201708, China

⁴ Hebei Province Automotive Safety Parts Technology Innovation Center, Zhuozhou 072750, China

⁵ School of Management, Jiangsu University, 301 Xuefu Road, Zhenjiang 212013, China

* Correspondence: liyongthinkpad@outlook.com

Abstract: Simultaneous localization and mapping (SLAM) is one of the key technologies for mobile robots to achieve autonomous driving, and the lidar SLAM algorithm is the mainstream research scheme. Firstly, this paper introduces the overall framework of lidar SLAM, elaborates on the functions of front-end scanning matching, closed-loop detection, back-end optimization, and map building module, and summarizes the algorithms used. Then, the classical representative SLAM algorithms are described and compared from three aspects: pure lidar SLAM algorithm, multi-sensor fusion SLAM algorithm, and deep learning lidar SLAM algorithm. Finally, the challenges faced by the lidar SLAM algorithm in practical use are discussed. The development trend of the lidar SLAM algorithm is prospected from five dimensions: lightweight, multi-sensor fusion, combination of new sensors, multi-robot collaboration, and deep learning. This paper can provide a brief guide for novices entering the field of SLAM, and provide a comprehensive reference for experienced researchers and engineers to explore new research directions.

Keywords: simultaneous localization and mapping; lidar; multi-sensor fusion; deep learning; development tendency

1. Introduction

With the release of China's 2035 vision goal outline and the proposal of new quality productivity, mobile robot technology has also received extensive attention. Simultaneous localization and mapping technology is a mobile robot in an unfamiliar environment, using its various sensors to carry out environmental data acquisition, attitude estimation, and self-positioning, to build the surrounding environment map [1]. The term "SLAM" was initially proposed by Smith and Cheeseman at the IEEE Robotics and Automation Conference. The technology has been in development for over three decades. The current mainstream SLAM algorithms are mainly divided into three categories, depending on the type of sensors used: lidar SLAM, visual SLAM, and multi-sensor fusion SLAM.

Visual SLAM mainly uses the camera as the core sensor. The camera is favored by many researchers because of its rich texture information, simple operation, and low cost [2]. The camera is mainly divided into monocular camera, binocular camera, and depth camera. The monocular camera uses a single camera to obtain environmental information. Its advantages are simple structure and low cost. However, due to structural limitations, the depth information of the object cannot be obtained. The binocular camera can estimate the depth information of the object by calibration and matching, but the calibration process is complex and the calculation load is large [3]. The depth camera can obtain the depth information and color of the object, and the information collection speed is fast, but its cost is high [4]. Due to the limitation of field of view and resolution, the depth camera is mainly used in indoor scenes.

Compared with visual sensors, lidar has the advantages of high-ranging accuracy, is not susceptible to external interference such as illumination and perspective changes, and stable and accurate map construction [5]. It is widely used in the construction of large-scale complex indoor and

outdoor scene maps. Lidar SLAM mainly uses 2D lidar or 3D lidar to collect information. 2D lidar can identify obstacles in the lidar scanning plane, with fast scanning speed and low cost. However, 2D lidar can not obtain the height information of the object, so it is often used in indoor service robots [6]. The 3D lidar uses multiple laser beams to scan the environment to obtain three-dimensional structural information such as the distance and shape of the object, but the measurement distance is greatly affected by the weather and is expensive.

Inertial measurement unit (IMU), millimeter wave radar, ultrasonic radar, and GPS are also used in multi-sensor fusion SLAM. IMU can measure the acceleration and attitude angle of the carrier and can provide the initial value for the prior pose estimation. However, as the running time increases, the cumulative error is large. Millimeter wave radar has a strong penetration ability to haze, rain, and other weather, but its accuracy is low [7]. Ultrasonic radar is easy to install and suitable for close-range accurate detection, but the ultrasonic transmission speed is easily affected by the environment, and it is difficult to detect objects in non-linear directions. GPS can provide accurate positioning information for mobile robots, with fast update speed and high precision, but trees and buildings may block GPS signals [8]. The advantages and disadvantages of various sensors in the lidar SLAM algorithm are shown in Table 1.

Table 1. Analysis table of advantages and disadvantages of each sensor.

Sensor type		Advantages	Disadvantages
Camera	Monocular camera	Simple structure and low cost	Depth information cannot be obtained
	Binocular camera	Depth information can be measured	The calibration process is complex and the calculation load is large
	Depth camera	Depth information and the color of the object can be obtained	High cost, limited by field angle and resolution
Lidar	2D lidar	Fast scanning speed and low cost	The height information of an object cannot be measured
	3D lidar	Three-dimensional structure information such as the distance and shape of an object can be obtained	Affected by the weather, the price is high
Inertial measurement unit		The acceleration and attitude angle of the carrier can be measured	Easy to produce cumulative errors
Millimeter wave radar		Strong penetration for rain, snow, and haze	Low data accuracy
Ultrasonic radar		Easy to install, suitable for close-range accurate detection	Easy to be affected by the environment, it is difficult to detect objects in a non-linear direction
GPS		Can provide accurate positioning information, update speed, high precision	GPS signals may be blocked by trees and buildings

SLAM technology has been continuously developed and improved in recent years. Many scholars have summarized the related research of SLAM, described the basic problems, models, frameworks, and technical difficulties of SLAM, reviewed the progress of related improved algorithms, and discussed the development trend of the SLAM algorithm. Among them, most of the literature reviews mainly introduce the application of visual SLAM algorithms or SLAM algorithms in specific scenarios, and there are few reviews on lidar SLAM algorithms. Kazerouni et al [9] provide an overview of recent research advances in visual SLAM for mobile robots and introduce machine vision algorithms for feature extraction and matching. Zhu et al [10] summarize the formation of state estimators in SLAM algorithms and give the current state of research on multi-sensor fusion

algorithms, but do not expand on this in the context of specific SLAM algorithms. Yarovoi et al [11] discuss the background information of SLAM algorithms, and the challenges faced, and summarize the recent SLAM directions. However, they only discuss the application of SLAM algorithms in the construction industry. Saleem et al [12] analyze recent advances in neural network techniques for SLAM-based ground vehicle localization, but only summarize the localization module of the SLAM algorithmic framework.

This paper systematically introduces the system architecture of the lidar SLAM algorithm and the current mainstream lidar SLAM algorithm scheme and summarizes the challenges and future development trends of the lidar SLAM algorithm. The main contributions of this paper are as follows.

- (1) In this paper, the system architecture of the lidar SLAM algorithm is analyzed comprehensively. The article introduces in detail the four key modules of scanning matching, closed-loop detection, back-end optimization, and map construction in the SLAM algorithm framework.
- (2) The current mainstream lidar SLAM algorithm is summarized and described. The principles, advantages, and disadvantages of various SLAM algorithms are compared and analyzed from three aspects: pure lidar SLAM algorithm, multi-sensor fusion SLAM algorithm, and deep learning SLAM algorithm.
- (3) The challenges and future development trends of the lidar SLAM algorithm are discussed and summarized. In practical applications, the lidar SLAM algorithm faces the problems of difficult effective fusion of multiple sensor data, inherent measurement problems of lidar, and poor robustness. Five major trends in the future development of the lidar SLAM algorithm are summarized.

The rest of this review is organized as follows. In Section 2, the overall framework of the lidar SLAM algorithm is described in detail. Section 3 describes and summarizes the latest representative lidar SLAM algorithms. Section 4 discusses the challenges faced by the lidar SLAM algorithm. Section 5 summarizes the future development trend of lidar SLAM. Section 6 is the conclusion of this paper.

2. Lidar SLAM System Architecture

The SLAM system architecture can be roughly divided into four key parts: sensor data scanning and matching, closed-loop detection, back-end optimization, and map construction. The data collected by lidar and other types of sensors are processed and analyzed by the SLAM front-end to estimate the pose transformation between adjacent lidar data frames. At this time, the calculated pose contains cumulative errors. The back-end optimization is responsible for global trajectory optimization, obtaining accurate pose, and constructing a global consistency map. Throughout the process, the closed-loop detection has been performed. It is used to identify historically identical scenes, achieve closed loops, and eliminate cumulative errors [13]. The system architecture of SLAM is shown in Figure 1.

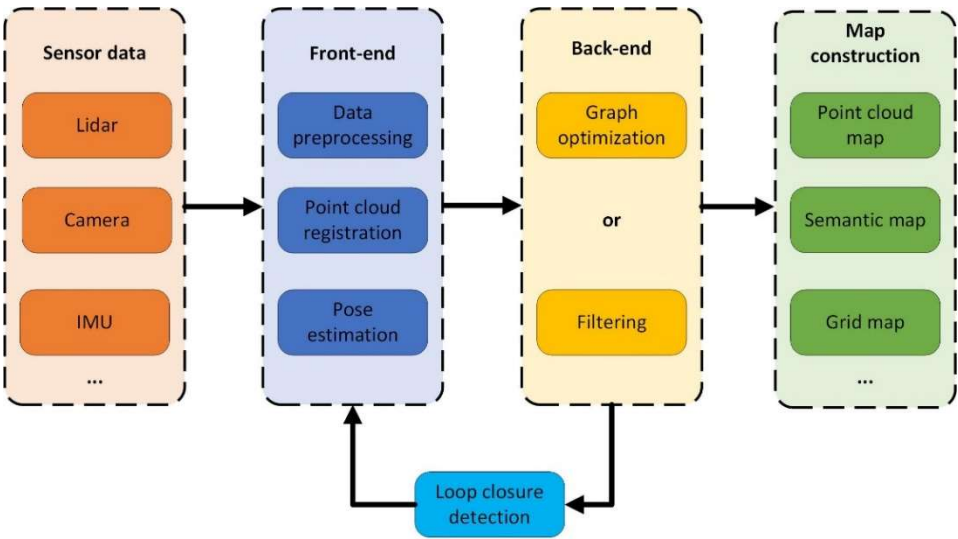


Figure 1. System framework of SLAM.

2.1. Scan Matching

The accuracy and computational efficiency of the scan matching algorithm directly affect the accuracy of trajectory estimation and map construction of the SLAM algorithm, which provides the initial value of the robot state and constraint relationship for the back-end optimization module. Scanning matching algorithms can be broadly categorized into iterative closest point (ICP) based algorithms, geometric feature-based algorithms, and mathematical feature-based algorithms.

2.1.1. ICP-Based Matching Algorithm

The ICP matching algorithm was originally proposed by Best [14] and Chen et al [15]. The algorithm's principle is to find the best pose transformation relationship between two scanning points of adjacent frames using continuous iteration with the minimum distance error as the objective function. The algorithm is simple and easy to implement, but it is easy to fall into local optimization, and the computational cost is too high when the number of scanning points increases.

To solve the problem that the traditional ICP algorithm is easy to fall into local optimization and improve the accuracy and efficiency of registration, many researchers have proposed an improved ICP algorithm. The improved algorithm optimizes the ICP algorithm steps utilizing point selection, matching, weighting, eliminating false associations, and minimizing cost functions. The ICP matching algorithms suitable for lidar SLAM include PL-ICP (point-to-line ICP) [16], PP-ICP (point-to-plane ICP) [17], GICP (generalized ICP) [18], NICP (normal ICP) [19], etc. The PL-ICP algorithm takes the distance from the point to the line as the error value, while the PP-ICP algorithm takes the distance from the point to the surface as the error cost. The GICP algorithm combines the ICP algorithm and the PL-ICP algorithm to perform point cloud registration on the probabilistic framework model, which improves the accuracy of matching. The NICP algorithm applies the normal vector and curvature information of the point cloud surface to the ICP algorithm, which improves the robustness of the registration.

The Go-ICP algorithm [20] introduces global 3D Euclidean point cloud matching based on the branch and bound algorithm, which solves the problem that the ICP algorithm is sensitive to initial values. The GP-ICP algorithm [21] introduces ground plane constraints for ICP scan matching of ground vehicle point clouds. The algorithm is based on the boundary of height information to find the corresponding relationship between two frames of point clouds, which overcomes the problem of ground slope change. The Faster-GICP algorithm [22] filters the lidar points through a two-step point filter. First, the lidar points are filtered, only the points with higher flatness are retained, and then further filtered according to the contribution to the optimized objective function. With the help of a two-step point filter, the Faster-GICP algorithm improves the efficiency and accuracy of point cloud matching. A large number of iterations, requirements for initial values, and sensitivity to noise limit the ICP and its variant algorithms, and the computational cost of directly using the original point cloud matching is usually high.

2.1.2. Geometric Feature-Based Matching Algorithm

The geometric feature-based scan matching algorithm speeds up the efficiency of point cloud matching by extracting object edge features and planar features from the scanned point cloud and matching the corresponding features. Scanning matching based on geometric features is shown in Figure 2.



Figure 2. (a) Point cloud data obtained by lidar; (b) The extracted edge and plane features.

LOAM [23] extracts the corner points and plane points of the current frame and the matching frame based on the local curvature of the point cloud constructs the constraints through point-line and point-surface, takes the point-line distance and point-surface distance as the error function, and performs point cloud matching and pose optimization based on the nonlinear least squares method. R-LOAM [24] combines the point features in the LOAM algorithm and the grid features in the reference object to achieve fast matching of the corresponding point cloud features. However, the algorithm has higher requirements for the positioning accuracy of the reference object and the three-dimensional grid. E-LOAM [25] extends the pre-extracted geometric feature information to the local point cloud information around the geometric feature points for matching, which improves the registration accuracy of the algorithm. To solve the problem of sparse geometric features in unstructured environments, the E-LOAM algorithm uses the intensity information of the point cloud to extract points with large intensity changes as additional feature points for matching. Surfel is a point cloud-based representation method that uses point clouds rather than polygonal meshes to represent three-dimensional shapes but requires GPU processing. SuIn-LIO [26] combines the invariant extended Kalman filter state estimator with the map based on the facet feature, which can not only achieve the accuracy of state estimation and map construction but also realize the real-time registration of lidar scanning frames.

The matching algorithm based on geometric features does not need the prior information on point cloud transformation relationship and only uses the geometric features of the point cloud for iterative solution. It has good real-time performance and is widely used. In structured scenes, the feature-based scan matching method can achieve better scan matching, but there are also some problems. For sparse scanning point clouds or unstructured environments, it is difficult to extract reliable features, which can easily lead to mismatches [27]. Compared with the ICP algorithm, the registration accuracy of the matching algorithm based on geometric features is poor, and the extraction of robust features requires a higher computational cost [28]. Given the above problems, adopting a coarse-to-fine hybrid scanning matching method has gradually become a trend. Firstly, the initial pose estimation is obtained by using the feature-based method, and then the ICP algorithm is run to further correct the pose, to obtain an accurate pose finally. He et al [29] propose an ICP registration algorithm based on point cloud local features. The algorithm combines the density, curvature, and normal information of the point cloud to design an efficient and robust three-dimensional local feature descriptor. Then, according to the feature descriptor, the initial registration result of the point cloud is found. Finally, the result is used as the initial value of the ICP algorithm to achieve accurate point cloud registration.

2.1.3. Mathematical Feature-Based Matching Algorithm

Mathematical feature-based matching algorithms are scanning matching methods that use mathematical properties to describe the point cloud data and the pose changes between frames, and the most famous one is the matching algorithm based on the normal distribution transform (NDT). The NDT algorithm [30] divides the point cloud into grids calculates the normal distribution of each grid to represent point cloud features, and then calculates the relative positional relationship between point clouds, which has the advantages of fast matching speed and good stability.

SRG-NDT [31] is a segmented region-growing NDT algorithm. The algorithm first removes the ground points in the scanned point cloud to speed up the point cloud registration and then performs Gaussian clustering on the remaining point clouds to ensure the matching accuracy of the algorithm. In MI-NDT [32], the reference scan voxel is transformed into a multi-scale voxel based on the optimal plane ratio criterion. The pose obtained from the coarse voxel is used as the input of the fine voxel, and it is refined to the finest voxel level. The final pose parameter is the best transformation. The MI-NDT algorithm has greatly improved the registration accuracy and robustness. NDT-6D [33] is a color-based normal distribution transformation point cloud registration method. The algorithm uses geometric and color information to calculate the correspondence between point clouds and uses three-dimensional geometric dimensions to minimize the distance of these correspondences, which improves the robustness of the algorithm to noise. NDT-LOAM [34] proposes a method combining weighted NDT and local feature (LFA) to match point clouds and estimate the motion of the lidar. The algorithm has the characteristics of low drift, high precision, and strong real-time performance.

The ICP algorithm has higher accuracy when registering between adjacent frame point clouds, but the initial pose has higher requirements. The matching algorithm based on geometric features can balance computational efficiency and matching accuracy, but it is not suitable for sparse point clouds and unstructured scenes. The calculation efficiency of the NDT algorithm is relatively high, but the matching accuracy is not as good as the ICP algorithm. The characteristics of the typical lidar scanning matching algorithm are summarized as shown in Table 2.

Table 2. Features of typical scan matching methods.

Features	ICP-type methods	Feature-based methods	NDT-type methods
Iteration	Need	Optional	Need
Initial value	Need	No need	Need
Domain of convergence	Small	/	Large
Operating speed	Slow	Fast	Medium
Robustness	Poor	Medium	Good
Precision	High, Affected by outliers and noise	Low, Related to feature extraction accuracy	Medium, Related to voxel size
Scope of application	Wide	Structured scene	Wide

Based on the characteristics of different matching algorithms, some scholars have proposed a combined scan-matching algorithm. Wang et al [35] use the NDT point cloud registration algorithm to roughly estimate the pose of the unmanned vehicle and then use the ICP point cloud registration algorithm to correct the registered point cloud to achieve an accurate estimation of the pose of the unmanned vehicle. Yang et al [36] propose a point cloud registration algorithm based on 3D NDT-ICP, which uses a 3D NDT algorithm to solve the coordinate transformation of the point cloud in the roadway, and optimizes the unit resolution of the algorithm according to the environmental characteristics of the roadway, introduces kd-tree in the ICP algorithm for the search of point pairs, and optimizes the nonlinear objective function of the algorithm by using the Gaussian-Newton method to complete the point cloud of the roadway with Accurate alignment.

2.2. Closed Loop Detection

Both scan matching at the front end and map building at the back end of the SLAM algorithm generate computational errors due to deviations in sensor information, which accumulate over runtime. Closed-loop detection is a key module that can avoid error accumulation, which is very important for SLAM systems running in large environments, many loops, and a long time. The main goal of closed-loop detection is to detect whether the robot has visited the same position or not, thus correcting localization errors due to drift or environmental changes [37]. If the closed loop is correctly identified, the global pose and map information errors can be corrected. If the two frames of point clouds that are far apart are mistakenly identified as closed loops, it may cause a large deviation in pose estimation and even lead to map construction failure. The process of closed-loop detection is shown in Figure 3.

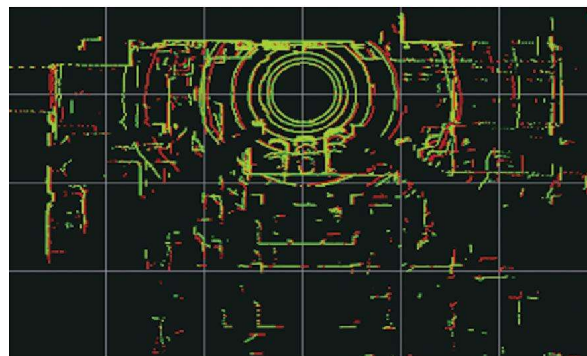


Figure 3. Closed loop detection (The green is the current frame point cloud, and the red is the historical loop-closure frame point cloud).

Tomono [38] uses coarse-to-fine point cloud registration to detect the closed-loop constraints of the robot. The coarse estimation used geometric constraints such as planes and lines to shorten the time of closed-loop detection and then used the iterative closest point algorithm for fine estimation. Meng et al [39] developed a closed-loop detection algorithm based on a point cloud histogram in multi-resolution mode. This algorithm can accurately detect the closed-loop in the process of vehicle driving, and obtain accurate attitude information and map through global optimization. However, the recognition accuracy is poor in scenes with large vertical changes.

In addition to the direct matching of two sets of point cloud frames, another approach is to construct descriptors with geometric information, intensity information, or others, and then compare the descriptors to measure the similarity of the closed-loop detection, which is based on both artificial design and deep learning. The former can directly process the 3D point cloud, including local-based descriptors, global-based descriptors, and hybrid-based descriptors, while the latter requires representing the 3D point cloud into a format that can be processed by deep learning, such as preprocessing it into a structured, fixed-size representation. Effectively extracting feature descriptors with discriminative ability and robustness is also one of the issues that scholars focus on and study [40].

The local descriptor is extracted from the position of each feature point and matched based on the historical scene, which improves the efficiency of closed-loop detection. Some key point detection methods and local descriptors are proposed, such as 3DSift, 3DHarris, 3D-SURF, ISHOT, and FPFH. MF-LIO [41] uses the fast point feature histogram feature descriptor to perform closed-loop detection, which improves the positioning and mapping accuracy of the SLAM algorithm. FELC-SLAM [42] introduces a variety of feature descriptors at the back end of the SLAM algorithm for closed-loop detection, which effectively solves the problem of sparse point clouds at the remote end.

Local descriptors ignore the relationship between local features and lack robustness when facing scenes with sparse lidar point clouds and insufficient features. The global descriptor can make full use of point cloud data and improve the accuracy of closed-loop detection. Inspired by Mercator projection and depth map, Wang et al [43] propose a new global Mercator descriptor to improve the accuracy and recall rate of closed-loop detection. Kim et al [44] propose a method based on scanning context, Scan Context, which is a non-histogram global descriptor. It divides each frame of the point cloud into several regions, selects representative points, and saves them to a two-dimensional array. The kd-tree is used to find the historical Scan Context descriptor for similarity calculation and detect whether to loop back. The Scan Context algorithm is not affected by the change of lidar viewpoint, and can still achieve efficient loop closure detection in real environments such as large scenes, noise, and partial occlusion.

In recent years, many scholars have integrated the intensity features, semantic information, and deep learning of lidar into the closed-loop detection module to achieve robust and efficient results. Zhang et al [45] propose a new Intensity Cylindrical Projection Shape Context (ICPSC) descriptor and ICPSC-based row-column similarity estimation and use a double-valued ring candidate verification strategy to achieve accurate closed-loop detection in sparse scenes. Li et al [46] propose a semantic scanning context location recognition method, which uses semantic descriptors to encode scene information and uses two-step global ICP to eliminate the influence of rotation and translation on descriptor matching. Cattaneo et al [47] introduce a closed-loop detection network (LCDNet). The deep learning network consists of a shared encoder, a position recognition head that extracts the global descriptor, and a relative pose head that estimates the conversion between two point clouds. It can identify the closed-loop and estimate the 6-DOF pose transformation.

2.3. Back-End Optimization

In the lidar SLAM algorithm, the back-end optimization is a process of integrating the lidar pose and inter-frame motion constraints of each frame to achieve overall optimization and generate an environment map based on the scanning data provided by the front end. According to different key algorithm theories, lidar SLAM can be divided into filter-based lidar SLAM and graph-based lidar SLAM. The filter method mainly uses the Kalman filter or extended Kalman filter to obtain the optimal state estimation. There are problems such as slow update efficiency, poor environmental adaptability, and lack of closed-loop detection modules. It is difficult to apply in large-scale long-distance and multi-loop scenarios. The method based on graph optimization will consider all the

sensing data for global optimization. Although the calculation amount is large, it can make full use of the data collected by the sensor, and the fusion accuracy is high.

2.3.1. Algorithm Based on Filtering

The filter-based back-end optimization algorithm is mainly based on Bayesian estimation, and the posterior probability is calculated by prior probability, sensor measurement, and state transition model. In the state estimation, the commonly used filtering algorithms are the Kalman filter (KF) [48], extended Kalman filter (EKF), unscented Kalman filter (UKF), error state Kalman filter (ESKF), and particle filter (PF) [49]. Table 3 summarizes several filtering algorithms.

Table 3. Filtering algorithm in back-end optimization.

Filtering algorithm	Description	Limitation
Kalman filter (KF)	Recursive optimal estimation of linear Gaussian system	Only applicable to linear systems
Extended Kalman filter (EKF)	The first-order Taylor is used to linearize the nonlinear system to solve the nonlinear problem to a certain extent.	When it is far away from the working point, the degree of linearization is not enough, which will have a greater impact.
Unscented Kalman filter (UKF)	The Sigma sampling points are used to approximate the nonlinear Gaussian transform.	Depending on the number of sampling points, it can deal with nonlinear problems but has high computational complexity.
Error state Kalman filter (ESKF)	The error state variable is used instead of the original variable for iterative update, which has small dimension and high linearization degree.	It is sensitive to initial estimates and has accumulated errors.
Particle filter (PF)	Based on the Monte Carlo method, a set of particles is constructed to approximate the target state distribution and update the particles.	There is a particle dissipation problem, and the computational complexity is proportional to the number of particles.

To balance efficiency and accuracy, Gao et al [50] propose a GNSS-LiDAR-inertial state estimator based on the iterative error state Kalman filter algorithm, which fuses GNSS, LiDAR, and IMU data to achieve coarse-to-fine state estimation and improves the accuracy and stability of the SLAM system. EKF-LOAM [51] uses the extended Kalman filter algorithm to fuse the wheel odometer and IMU data into the LeGO-LOAM algorithm, which improves the positioning ability of the mobile robot in the absence of GPS data. FAST-LIO [52] uses a tightly coupled iterative extended Kalman filter to fuse lidar feature points with IMU data to achieve navigation in fast motion or noisy environments. This algorithm proposes a new Kalman gain calculation formula, which makes the calculation amount depend on the state dimension rather than the measurement dimension, which greatly improves the operation efficiency of the algorithm. Compared with the traditional Kalman filter algorithm, researchers are more inclined to use the variants of the EKF algorithm to improve efficiency and state estimation accuracy.

2.3.2. Algorithm Based on Graph Optimization

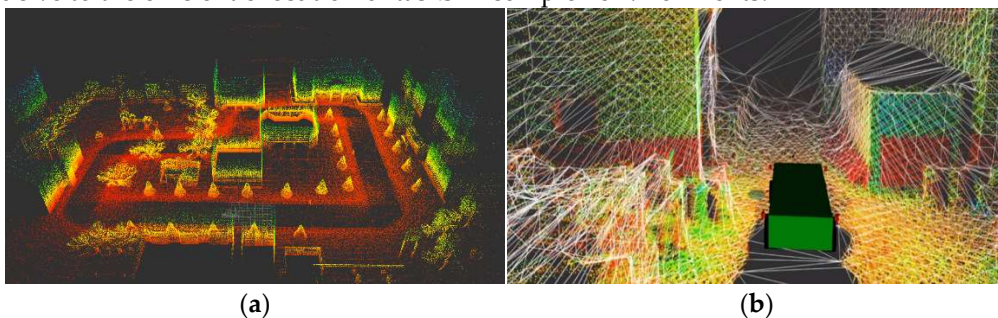
The disadvantage of the filtering algorithm is that it cannot correct the cumulative error, and it will produce serious error drift when running for a long time in the scene with large noise and many loops. Graph optimization uses the method of graph theory to model and solve the system. It is an optimization scheme adopted by most SLAM algorithms at present and is deeply loved by the majority of researchers. Factor graph optimization is a special graph optimization method. It represents the pose state of the robot with nodes and represents the constraint relationship between nodes with edges. The nonlinear least squares method is used to iteratively optimize the cumulative error in the mapping process.

The core idea of the graph optimization algorithm is to use the graph optimization framework to estimate the pose and construct the map in real time. The SLAM algorithm is divided into front-end and back-end. The front end is responsible for scanning matching, feature extraction, and data association. The back end is responsible for pose graph optimization and closed-loop detection is added to correct the cumulative error. It can be applied to outdoor and indoor large-scale environments with good robustness, but the amount of calculation is too large. Lang et al [53] design a dynamic weight estimation algorithm based on sensor and closed-loop detection information. The algorithm uses the different accuracy of LiDAR, IMU, and closed-loop detection in different scenarios to calculate the dynamic weight of the graph optimization edge, which improves the accuracy of the back-end state optimization. Du et al [54] use the nonlinear least squares method to optimize the factor graph composed of the INS factor, camera factor, lidar odometer factor, and motion model factor to improve the attitude estimation accuracy of the algorithm in the sparse feature outdoor environment.

At present, the open-source optimization algorithm libraries based on graph optimization include iSAM (incremental smoothing and mapping), GTSAM (georgia tech smoothing and mapping), G2O (general graph optimization), Ceres, BA (bundle adjustment), etc. With the help of these optimization libraries, the time of back-end iteration to solve the optimization value can be saved. Using these open-source algorithm libraries, developers can focus on the construction of system graph models rather than specific mathematical calculations to achieve the purpose of simple and efficient development.

2.4. Map Construction

The map constructed by the lidar SLAM algorithm is a model of the surrounding environment of the mobile robot and a prerequisite for positioning and autonomous navigation [55]. In the field of SLAM, the constructed map is mainly divided into point cloud map, octree map, semantic map, and so on. The 2D lidar SLAM generally uses a grid map to represent the surrounding obstacle information. The algorithm divides the real environment into grids and uses the probability 0~1 to represent the probability that the grid is occupied by obstacles. The 3D lidar SLAM algorithm intuitively describes the surrounding environment information through the point cloud map, as shown in Figure 4a. The point cloud map has a large amount of data, which generally needs to be downsampled by voxel filtering to display normally. A simple point cloud map cannot distinguish obstacle information and cannot be used for path planning and obstacle avoidance, but it can be used for point cloud registration research. Mesh maps represent a 3D model of the environment by capturing key points and features in the environment and generating a mesh consisting of triangular facets, as shown in Figure 4b. The Mesh map can be used for path planning, but it needs to filter out sensor noise. OctoMap is a three-dimensional spatial map representation method based on an octree data structure, which efficiently processes and stores large-scale environmental information by recursively decomposing three-dimensional space into eight equal sub-units, as shown in Figure 4c. The octree map is a special occupied grid map. In this structure, the grids with the same occupancy probability can be merged to reduce the storage space of the map, which is suitable for obstacle avoidance and path planning. The semantic map uses semantic segmentation to classify objects in the environment. It not only contains spatial information but also integrates a semantic understanding of various elements in the environment, as shown in Figure 4d. Semantic maps combine geometric and semantic information to improve the ability of mobile robots to perceive the environment, which is conducive to the efficient execution of tasks in complex environments.



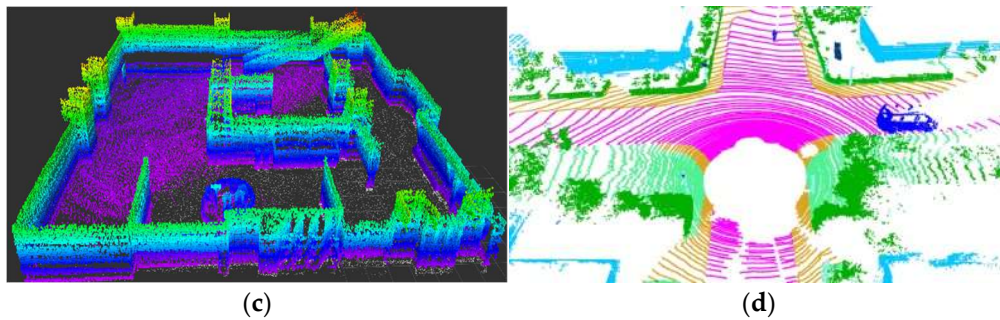


Figure 4. The representation of a map.

Tsai et al [56] use point cloud data preprocessing and feature extraction methods such as ground segmentation to improve the recognition accuracy of vertical objects, reduce rotation and translation errors, and construct an accurate global point cloud map. Ruetz et al [57] propose a local three-dimensional environment representation method for unmanned ground vehicle navigation, visible point cloud mesh (OVPC Mesh). This method uses local point cloud data to generate 3D meshes to represent the free space around the robot, which balances the representation accuracy and computational efficiency. Cai et al [58] propose an efficient occupancy grid map D-Map based on an octree map. The algorithm uses a depth image to determine the area occupancy state, and deletes the known state of the grid when updating the state, ensuring low memory consumption and high operating efficiency. Hu et al [59] designed a neural network architecture RandLA-Net, using random sampling and local feature clustering methods to achieve semantic segmentation of large-scale 3D point clouds.

3. SLAM Algorithm Scheme Based on Lidar

3.1. SLAM Algorithm Based on Pure Lidar

The pure lidar SLAM algorithm can be divided into 2D lidar SLAM algorithm and 3D lidar SLAM algorithm according to the type of lidar used.

In the 2D lidar SLAM algorithm, Montemerlo et al [60] propose the Fast SLAM algorithm, which is a SLAM algorithm based on a particle filter. The algorithm decomposes the SLAM problem into two independent problems: robot pose estimation and map coordinate point posterior probability distribution estimation. The robot pose estimation is realized by particle filter, and the map coordinates posterior probability distribution is estimated by an extended Kalman filter. Fast-SLAM can deal with nonlinear systems due to the use of particle filtering technology. However, in a large environment or when the odometer error is large, more particles are needed to obtain better estimation, and the resampling of particles will also lead to particle dissipation. GMapping [61] improves based on particle filter and alleviates the risk of memory explosion by reducing the number of particles. The selective resampling method is used to rank the importance weights of particles, and the particles with low weights are resampled to solve the problem of particle dissipation in the RBPF algorithm. The Gmapping algorithm has good real-time performance and high mapping accuracy in the indoor environment. It is one of the most commonly used SLAM algorithms, but it lacks closed-loop detection, and there is a large cumulative error in long-term operation in outdoor large scenes.

Karto [62] is the first open-source algorithm based on graph optimization, which makes full use of the sparsity of the system. In the front end, correlation scan matching is used to match the data frame with the local map to calculate the pose. In the back end, the graph-based optimization method is used for global optimization. The closed-loop detection uses the candidate data frame to construct the local map and uses the scan-to-map method to match and detect the closed-loop. Hector [63] does not require odometer data and uses the Gauss-Newton method to directly perform scan-to-map matching to construct a 2D grid map. Due to the lack of closed-loop detection in Hector, once the map is wrong, it will affect the subsequent mapping effect, and the scan-to-map method has a large amount of calculation, the robot must move at a lower speed. In 2016, the Google team open-sourced the Cartographer [64] algorithm, adding sensor time synchronization and laser data preprocessing, introducing the concept of submap, using a combination of CSM and gradient optimization to match frames with submaps, using branch-and-bound to accelerate loopback detection, and combining with

lazy decision making to reduce the likelihood of loopback errors. Cartographer is one of the open-source algorithms with the highest accuracy, the best robustness, and the most convenient secondary development. Table 4 is a summary of the 2D lidar SLAM algorithm.

Table 4. 2D lidar SLAM algorithm.

Algorithm name	Algorithm framework	Loop closure detection	Merits and demerits
Fast-SLAM	Particle filter	No	Pose estimation and mapping are optimized separately; it can deal with nonlinear problems; but there is a problem of particle degradation.
Gmapping	Particle filter	No	Improve particle degradation; it relies heavily on odometers and cannot build large-scale maps.
Karto	Graph optimization	Yes	The first open-source SLAM based on graph optimization, with loopback detection function; poor real-time performance
Hector	Gauss-Newton	No	No odometer information is required; without a closed-loop detection function, it is easy to drift when rotating too fast.
Cartographer	Graph optimization	Yes	High precision and strong real-time, accelerate closed-loop detection; high requirements for computing resources

LOAM is a classic and representative 3D lidar SLAM algorithm, and many other algorithms are improved based on it. The framework of the LOAM algorithm is shown in Figure 5. The algorithm divides the feature points into edge feature points and plane feature points according to the curvature of the lidar points in the local range. LOAM divides the SLAM problem into high-frequency, low-precision motion estimation and low-frequency, high-precision map construction. Then it fuses the two data to obtain an accurate point cloud map. The shortcoming of the LOAM algorithm is the lack of back-end optimization and closed-loop detection, which will cause drift in large-scale and multi-loop scenarios. LeGO-LOAM [65] is a lightweight SLAM algorithm that projects three-dimensional point clouds onto two-dimensional images to separate ground points and non-ground points to remove noise. The algorithm divides the ground point cloud before feature extraction, which speeds up the operation efficiency and enables it to work on low-power embedded systems.

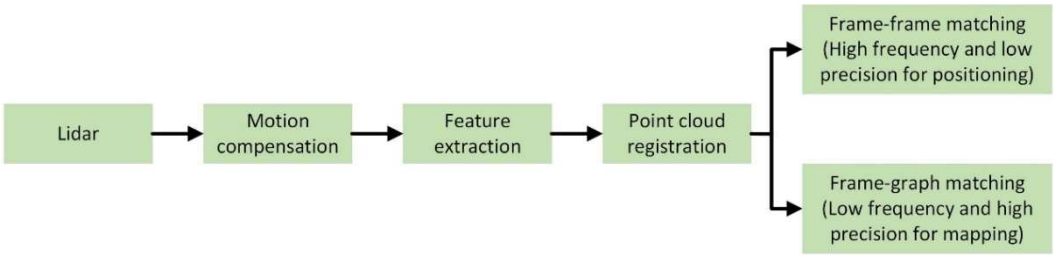


Figure 5. Algorithm framework for LOAM.

With the emergence of new sensors, Lin of the University of Hong Kong proposes the Loam-livox algorithm for the new solid-state lidar scanning characteristics [66]. The algorithm is optimized in terms of effective point screening and iterative pose optimization, which is of great significance for the subsequent research of other solid-state lidar SLAM algorithms. The traditional SLAM algorithm based on lidar mainly focuses on geometric features and time information, while ignoring the intensity information of lidar reflection. Wang et al. propose the Intensity-SLAM algorithm using geometric and intensity features [67]. The algorithm uses intensity information to construct an intensity map to assist feature point extraction and pose estimation. In the closed-loop module, a Scan Context descriptor based on intensity information is used to be robust to rotation.

3.2. Lidar SLAM Algorithm Based on Multi-Sensor Fusion

A single sensor cannot independently complete the global map construction in all scenarios, and the fusion of multiple sensors can solve the limitations of a single sensor and obtain a more accurate, efficient, and adaptable SLAM effect [68].

The fusion of lidar and IMU can overcome the problems of low update frequency and motion distortion of lidar in the lidar SLAM algorithm. According to the data fusion method, the fusion of lidar and IMU can be divided into loose coupling and tight coupling. Loose coupling is to process the lidar and IMU data separately, and then combine the parameters estimated by both for fusion. Compared with loose coupling, tight coupling uses lidar data and IMU data to jointly construct parameter vectors in the data processing process, and then estimates and optimizes them, showing better robustness and accuracy. Tight coupling is also one of the research hotspots of the multi-sensor fusion SLAM algorithm.

LIO-mapping [69] is a tightly coupled fusion algorithm of lidar and IMU proposed for the first time. By jointly optimizing the measurement data of IMU and lidar, there is no obvious drift even in the case of Lidar degradation. The mapping accuracy of the algorithm is improved, but the complexity of the tightly coupled algorithm is high. Pan et al [70] use the lidar inertial odometry of the iterative extended Kalman filter to fuse lidar keyframes and IMU information. In closed-loop detection, the global descriptor of the scanning context is constructed by using the de-distortion feature points of the front-end IMU to improve the accuracy of the closed-loop detection. In the back end, GTSAM is used for global optimization and GPS constraints are introduced to obtain globally consistent trajectories and maps. LIO-SAM [71] follows the idea of the LOAM algorithm to extract feature points, using IMU data to correct point cloud distortion and provide the initial value of pose transformation. The back end uses a factor graph-optimized architecture, as shown in Figure 6. The algorithm uses the factor graph framework to fuse the IMU pre-integration factor, the lidar odometer factor, the GPS factor, and the closed-loop detection factor, which can obtain better accuracy and real-time performance. However, in unstructured scenes, there will be a lack of effective feature points and failure. FAST-LIO2 [72] inherits the tightly coupled architecture of FAST-LIO. It makes full use of the subtle features in the environment by updating the map directly using the original lidar points and reduces the computational complexity significantly by using the ikd-tree data structure. In summary, the study of lidar inertial tight coupling mainly improves the accuracy, efficiency, and robustness of mapping through the tight coupling of lidar and IMU. However, the joint calibration method for lidar and IMU is not yet mature.

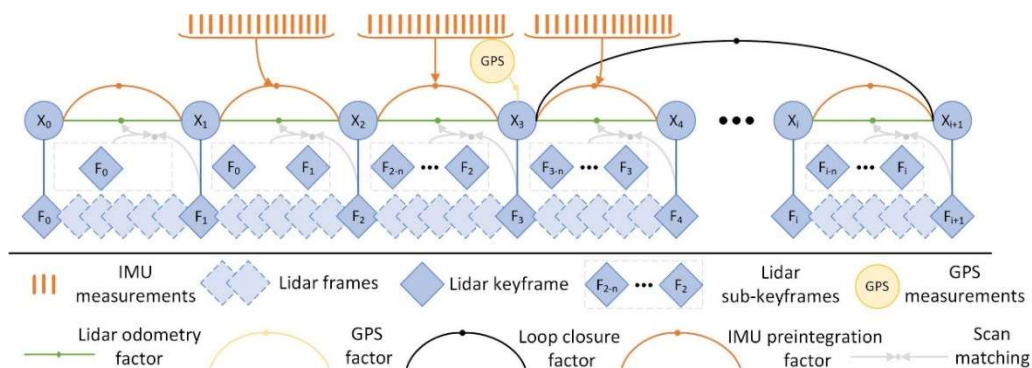


Figure 6. The system architecture of factor graph optimization.

Lidar cannot work normally in degraded scenes and has poor global positioning ability, while the camera has good global positioning ability and can obtain rich image texture information. The fusion of the two can improve the accuracy and application scenarios of the SLAM system [73]. V-LOAM [74] is a general architecture combining a lidar odometer and a visual odometer. The block diagram of the algorithm system is shown in Figure 7. The algorithm uses a visual odometer to process fast motion, while the lidar odometer guarantees low drift and robustness under poor lighting conditions. V-LOAM ranks first in the average translation and rotation error benchmark test of the KITTI dataset but lacks a closed-loop detection module. LVI-SAM [75] is a SLAM framework for tightly coupled lidar-vision-inertial odometry, which consists of a lidar-inertial system (LIS) and

a visual-inertial system (VIS). The architecture of the tightly coupled lidar-visual-inertial system is shown in Figure 8. LIS provides accurate depth information for VIS and improves the accuracy of VIS. In turn, LIS uses the initial pose estimation of VIS to perform point cloud scanning matching. The advantage of the algorithm is that one of the VIS or LIS fails, and the LVI-SAM can also work normally, which makes the algorithm have better robustness in the environment with less texture and lack of features. R²LIVE [76] uses the error state iterative Kalman filter to fuse the measurement data of lidar, vision, and inertial sensors for state estimation and further improves the overall accuracy through factor graph optimization. However, the visual system in the algorithm adopts the feature point method, which is easy to fail in an unstructured environment. To improve the accuracy and robustness of the algorithm, R³LIVE is proposed. R³LIVE [77] improves the visual system by using the optical flow method instead of the original feature point method. It effectively fuses the visual data by minimizing the photometric error from the frame to the map and renders the RGB color for the map. Experiments show that R³LIVE has stronger robustness and accuracy in state estimation than existing systems.

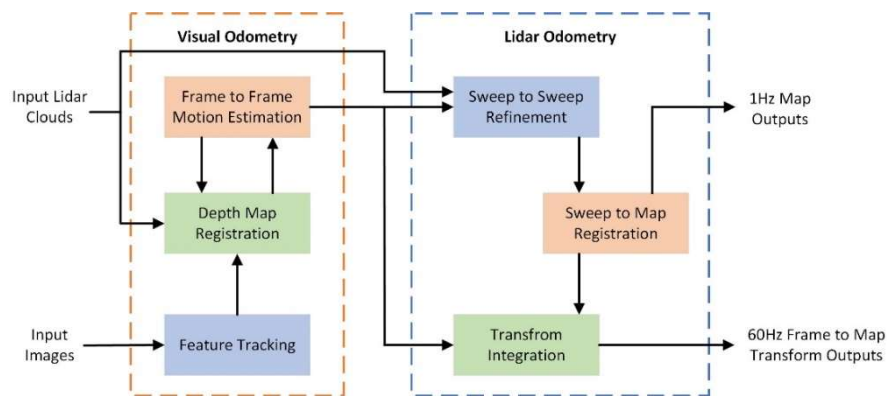


Figure 7. Block diagram of V-LOAM algorithm system.

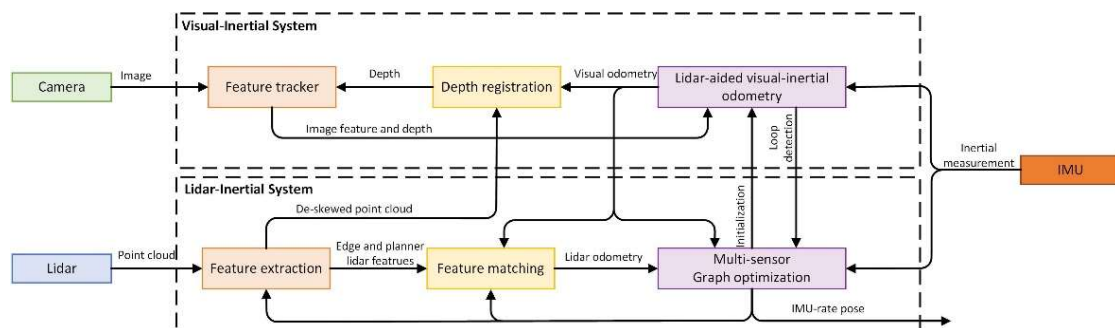


Figure 8. Lidar-visual-inertial tightly coupled system architecture.

To cope with fast motion, noise, and complex application scenarios, many researchers have begun to integrate different types of sensors into the SLAM system [78]. RadarSLAM [79] is a SLAM system suitable for frequency-modulated continuous wave radar. The algorithm uses radar to track the geometric structure reliably and compensates for the motion distortion through joint optimization. It can also work stably under extreme weather conditions such as heavy snow and fog. GR-LOAM [80] fuses the measurement data of wheel encoder, lidar, and IMU in a tightly coupled way, and estimates the pose change of the robot through IMU and wheel encoder. The algorithm detects abnormal data by adjusting the optimal weight of each sensor, and eliminates the moving point cloud of dynamic objects, but does not integrate camera data. Liu et al [81] propose a multi-sensor SLAM fusion framework integrating GPS, IMU, lidar, and camera. Different sensor data are fused in a tightly coupled manner and then optimized by a factor graph. The algorithm reduces the error of the sensor due to accidental changes through the visual-lidar automatic calibration method and uses the target detection module to establish a semantic map, which provides rich map information for navigation and obstacle avoidance.

3.3. Lidar SLAM Algorithm Based on Deep Learning

The combination of deep learning and lidar SLAM algorithm is mainly applied to several modules in SLAM architecture, such as feature extraction and registration of point clouds, closed-loop detection, and construction of semantic maps. The method based on deep learning uses data-driven learning to obtain more accurate results than manual design [82]. The application of deep learning in feature extraction and closed-loop detection improves the efficiency, accuracy, and robustness of the SLAM algorithm. Constructing semantic maps through deep learning methods can improve the ability of mobile robots to perceive the surrounding environment and adapt to different environments.

Traditional point cloud matching methods have achieved good matching results in some scenes, but they have the problems of high requirements on initial values, poor matching accuracy, and inability to adapt to the whole scene. With the rapid development of deep learning technology, researchers apply deep learning to point cloud feature extraction and point cloud matching to solve the drawbacks of traditional point cloud matching [83]. Sun et al [84] innovatively propose an MLP_GCN module that combines multi-layer perceptron and graph convolutional network. The MLP_GCN module is integrated into the point cloud registration detector and descriptor. The detector is used to extract key points, and the descriptor is used to describe the depth features of each key point, which improves the accuracy of point cloud registration. Based on deep learning, Poiesi et al [85] propose a GeDi descriptor for point cloud registration. The deep learning network uses a quaternion conversion module to ensure that the 3D descriptor can cope with different scaling and rotation and has strong robustness. Zheng et al [86] designed a PBNet that combines a point cloud network with global optimization. The network first uses the feature information of the target for coarse registration, then searches the entire three-dimensional motion space, and performs branch and bound method and iterative closest point method. PBNet reduces the influence of initial values on point cloud registration, but there is a local optimal problem. Deep learning brings certain convenience to SLAM and improves the accuracy and robustness of point cloud matching and feature extraction. However, deep learning models require data for training and also consume GPU resources.

Accurate closed-loop detection can eliminate the cumulative error of the SLAM algorithm and improve the accuracy of the algorithm. The traditional closed-loop detection is mostly detected by manually setting closed-loop features, which has the problems of low recognition accuracy and large amounts of calculation. Compared with the method of manually designing closed-loop detection features, deep learning can extract features with high discrimination and improve the accuracy of closed-loop through a large amount of data training and a reasonable network model [87].

In recent years, closed-loop detection algorithms based on deep learning have been continuously developed. DeLightLCD [88] is a deep and lightweight closed-loop detection neural network. The network includes a Siamese feature extraction module and a dual attention difference module. Based on the dual attention feature difference network, a feature difference map is generated to identify the difference between the two frame point clouds. A fast selection method of closed-loop candidate points is used to extract closed-loop candidate points from a large number of lidar scanning sequences. The DeLightLCD can perform efficient closed-loop detection without pre-pose, but the input of the network must be a two-dimensional depth image after point cloud conversion. Wang et al [89] propose a closed-loop detection method based on a multi-scale point cloud feature transformer. This method uses voxel sparse convolution to obtain the features of original point clouds with different resolutions and uses a transformer network to establish the relationship between different resolution features, to realize multi-scale feature fusion, and to obtain global descriptors. This method uses the Euclidean distance between point cloud descriptors to measure the similarity between point clouds and does not need to manually create descriptors, which improves the detection efficiency. Overlap-Transformer [90] is a lightweight Transformer network, which uses the distance image of 3D point cloud projection and Transformer attention mechanism to generate yaw angle invariant descriptors for closed-loop detection.

Semantic SLAM can accurately identify and classify target objects, and deep learning is precisely the most advantageous object recognition method at present. Therefore, the combination of deep learning and semantic SLAM has received extensive attention [91]. SuMa++ [92] uses the fully convolutional neural network to extract the semantic information in the point cloud and combines

the geometric information of the lidar point cloud to improve the accuracy of pose estimation. The algorithm filters the point cloud of moving objects through semantic segmentation and improves the accuracy of point cloud matching. Wen et al [93] study a semantic segmentation neural network architecture Hybrid CNN-LSTM, which combines the advantages of the CNN network and LSTM network. The LSTM network is used to segment the point cloud image obtained by PointNet, and the image block is input into the LSTM network according to the Hilbert curve sequence, which improves the segmentation accuracy of the neural network for smaller targets, but the real-time performance of the algorithm is poor. Li et al [94] propose the neural radiation field convolution (NeRFConv) method for large-scale point cloud semantic segmentation. The algorithm clusters the features by learning the density distribution and adaptive weights of the point cloud and preserves the spatial information and semantic features as much as possible. Luo et al [95] propose a multi-scale region relation feature segmentation network MS-RRFSegNet for urban scenes. The original point cloud is reorganized by uniform super voxels, which effectively reduces the computational complexity. The region relation feature inference module based on super voxels is used to extract the local region context information, and then the semantic segmentation point cloud map of the scene is obtained.

4. Challenges of the Lidar SLAM Algorithm

Through the combing and analysis of different lidar SLAM algorithms, it can be seen that the system framework and related theories of lidar SLAM have developed more maturely, but in practical applications, the process of realizing lidar SLAM technology from theory to practice still faces many challenges.

(1) Difficulty in fusing data from different sensors

Relying on a single sensor to perceive the surrounding environment has certain limitations, which can not meet the SLAM mapping algorithm in different scenarios. Reasonable fusion of multi-sensor measurement data can improve the performance of the mapping algorithm, but it is necessary to calibrate the relative pose transformation and timestamp between different sensors before fusion. Due to the different measurement frequencies of different sensors, there is a problem that the data sampling time is not synchronized [96]. In the actual use process, the external parameters of different sensors will affect the relative pose matrix due to mechanical deformation and changes in the external environment, which makes it difficult to effectively synchronize the data of different sensors. Li et al [97] propose a method to calibrate the parameters between lidar and IMU in time and space. By querying the accurate transformation on the continuous time trajectory, the deformation point and the time displacement point are modeled to realize the effective fusion of sensor data.

(2) Inherent problems of lidar measurement

When lidar is used to scan the surrounding environment and output a 3D point cloud, a huge amount of data is often generated. Although a large amount of point cloud data provides rich environmental information, it seriously affects the real-time performance of the algorithm and brings challenges to the lidar SLAM algorithm. Lidar will distort the scanning point cloud due to the rapid movement of the carrier, which will affect the matching effect of the point cloud. Yang et al [98] design a point cloud distortion correction method that combines lidar and a camera. The camera is used to provide dense angular resolution for the lidar, and the probabilistic Kalman filter method is used to match the real-time speed of the moving object with the correct point cloud to complete the point cloud distortion correction. In an environment with sparse or repetitive features, due to the limitation of the detection distance of the lidar, the point cloud scanning matching is mismatched, which in turn affects the accuracy of the algorithm pose estimation.

(3) Robustness in complex scenes

Real-time localization is a very important problem in lidar SLAM algorithm. In the process of switching between indoor scenes and outdoor scenes, how to ensure accurate positioning is a huge challenge for the algorithm. There are a large number of dynamic objects in some scenes, such as vehicles, electric bicycles, pedestrians, and other traffic participants [99]. The lidar SLAM algorithm needs to realize real-time detection and tracking of these dynamic objects and separate them from static obstacles during the construction of the map to ensure the accuracy and stability of the map [100]. DL-SLOT [101] is a tightly coupled dynamic lidar SLAM method. The algorithm uses object detection to identify point clouds of all potential dynamic objects and uses point clouds that filter out

dynamic objects for map construction. The state of the vehicle, stationary object, and dynamic object is jointly estimated by the sliding window graph optimization method.

5. The Development Trend of Lidar SLAM Algorithm

In recent years, with the development of robot technology and the progress of sensor technology, robot technology has been integrated into people's daily lives. As a technical support in the field of robotics, the development of SLAM technology is crucial to many fields. This section summarizes the future development of lidar SLAM technology.

(1) Lightweight and miniaturization

At present, sensors such as multi-beam lidar, high-precision IMU and differential GPS positioning module used in lidar SLAM algorithm are expensive, which limits the popular application of SLAM technology. Therefore, it is a development trend for the lidar SLAM algorithm in the future to realize the stable operation of the lidar SLAM algorithm embedded on small equipment and improve the popularization of its applicable objects. The purpose of the lidar SLAM algorithm is to realize the application of the underlying motion module, such as navigation, teaching, and service. At the same time, the SLAM algorithm also provides positioning and mapping functions for the upper module, which does not consume a lot of computing resources [102]. The pursuit of lightweight and miniaturization of lidar SLAM algorithm application is a direction in its future development process.

(2) Multi-source sensor information fusion

In the face of a complex surrounding environment, multi-source information fusion of different sensors is an inevitable trend of the lidar SLAM algorithm. Single-sensor information has certain limitations and cannot meet the mapping needs of complex environments [103]. To make up for the shortcomings of a single sensor, domestic and foreign scholars have designed a variety of multi-sensor fusion algorithms to build high-precision and high-robust maps. MSF-SLAM [104] is a multi-sensor fusion SLAM algorithm for complex dynamic scenes. At the front end of the system, the lidar point cloud provides three-dimensional information for the camera's image feature points. Image-based moving object detection can remove the moving point cloud in the lidar. At the system's back end, a multi-constraint factor graph of different sensors is constructed, and nonlinear optimization is performed to obtain the system's optimal state estimation. Many SLAM algorithms based on multi-sensor fusion have been proposed, but related research is still one of the research hotspots and development trends of the lidar SLAM algorithm.

(3) SLAM algorithm combined with the new sensor

With the continuous development of sensor technology, more functional and efficient sensors continue to appear. The appearance of new sensors can inject new vitality into the further development of SLAM technology. The data acquired by new sensors occupies little memory, is of high quality, and does not require much processing, which can improve the operational efficiency of SLAM algorithms. For example, the Event Camera is beginning to be used in the SLAM field with the advantages of low power consumption and high frame rate [105]. The use of new sensors provides more possibilities for the development of lidar SLAM algorithms.

(4) Multi-robot collaborative SLAM algorithm

Multi-robot collaborative SLAM algorithm is the current research hotspot and development direction. Multi-robot collaborative SLAM is a collaborative operation of multiple agents in an unknown environment. It exchanges information such as maps and self-positioning obtained by each other through communication protocols to achieve rapid perception of the environment, precise positioning, and map construction. However, multi-robot collaborative SLAM technology has the problems of communication bandwidth limitation and data sharing delay, which affects the real-time and accuracy of collaborative mapping. The distributed multi-robot cooperative SLAM system should have strong robustness and fault tolerance and can deal with the problem of single-robot failure and network communication interruption. Therefore, designing a SLAM algorithm with multi-sensor data fusion, distributed function, and coordinated allocation of robot resources is one of the future development directions. At present, the multi-robot cooperative SLAM algorithms with better performance include DiSCo-SLAM [106], Swarm-SLAM [107] and LAMP 2.0 [108].

(5) SLAM algorithm combined with deep learning

Combining deep learning with the lidar SLAM algorithm can improve the efficiency and robustness of the system. Deep learning can be applied to feature extraction and matching, semantic SLAM, and closed-loop detection modules in lidar SLAM algorithms. Deep learning improves the efficiency and matching accuracy of feature point recognition in the front-end matching of lidar point clouds [109]. The addition of semantic information realizes the classification and labeling of point cloud objects and enriches map information [110]. The closed-loop detection combined with deep learning has significantly improved accuracy, robustness, and intelligence. At present, there are still some shortcomings in the SLAM algorithm based on deep learning. Deep learning requires a lot of training data and GPU resources, and it is difficult to run in real-time on embedded platforms. In the future, the lidar SLAM algorithm based on deep learning is still a research hotspot for many scholars.

6. Conclusions

The SLAM algorithm is an effective means for mobile robots to construct an unknown environment map and obtain its pose information to achieve autonomous driving. It can assist robots in performing tasks such as path planning, autonomous exploration, and navigation. Starting from the system framework of lidar SLAM, this paper systematically introduces each module of the lidar SLAM system, then expounds and summarizes the mainstream lidar SLAM algorithm scheme, and finally discusses and summarizes the challenges and future development directions of lidar SLAM. The conclusions of this paper are as follows.

(1) Based on the lidar SLAM algorithm, the overall system architecture of SLAM is introduced. The functions of front-end matching, closed-loop detection, back-end optimization, and map construction modules are described in detail, and the algorithms used are summarized.

(2) With the development of lidar technology, lidar SLAM algorithms have also achieved rich research results. The classical SLAM algorithm schemes are sorted out from pure lidar SLAM algorithm, multi-sensor fusion SLAM, and deep learning SLAM.

(3) The algorithm scheme of lidar SLAM has been developed maturely, but there are still some defects in practical use, such as the difficulty of fusion of different sensor data, the inherent defects of lidar measurement, and the lack of robustness in complex scenes. Lightweight, multi-sensor data fusion, in combination with new sensors, multi-robot collaboration, and deep learning will be the main directions for the future development of lidar SLAM algorithms.

Funding: This work is supported by National Natural Science Foundation of China (Grant No. 51705213), State Key Laboratory of Intelligent Green Vehicle and Mobility (Grant No. KFY2409), Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province (Grant No. 2023ZHNY02), Key Laboratory of Agricultural Equipment Technology for Hilly and Mountainous Areas, Ministry of Agriculture and Rural Affairs (Grant No. 2023KLOP05), Opening Project of Automotive New Technique of Anhui Province Engineering Technology Research Center (Grant No. QCKJ202201A).

Data Availability Statement: Data will be made available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cai, Y., Lu, Z., Wang, H., Chen, L., & Li, Y. 2022. A lightweight feature map creation method for intelligent vehicle localization in urban road environments. *IEEE Transactions on Instrumentation and Measurement*, 71, 1-15.
2. Jiang, Z., Gao, X., Li, Y., Wang, H., Zhang, Y., & Fang, X. 2022. Multilayer map construction and vision-only multi-scale localization for intelligent vehicles in underground parking. *Measurement Science and Technology*, 33(11), 115021.
3. Zhang, X., Yan, Y., Liu, Y., & Yuan, C. 2023. Vision detection method of car body relative parking position based on close-range images of binocular lens. *Journal of Jiangsu University (Natural Science Edition)*, 44(3), 262-269.
4. Rao, Z., Cai, Y., Wang, H., Chen, L., & Li, Y. 2023. A multi-stage model for bird's eye view prediction based on stereo-matching model and RGB-D semantic segmentation. *IET Intelligent Transport Systems*.
5. Li, S., Sun, X., & Zhang, M. 2023. Vehicle recognition technology at urban intersection based on fusion of LiDAR and camera. *Journal of Jiangsu University (Natural Science Edition)*, 45(6), 621-628.
6. Wu, C., & Li, X. 2023. Autonomous navigation method of orchard robot based on single laser. *Journal of Jiangsu University (Natural Science Edition)*, 44(5), 530-539.

7. Yang, S., Dai, Z., Yang, X., & Tang, L. 2023. Bridge state evaluation and numerical simulation analysis based on millimeter-wave radar measurement and control. *Journal of Jiangsu University (Natural Science Edition)*, 44(5), 606-613.
8. Yuan, C., Zhu, H., He, Y., Jie, S., & Chen, L. 2023. Autonomous navigation algorithm for intelligent vehicle in weak GPS environment. *Journal of Jiangsu University (Natural Science Edition)*, 44(1), 22-28.
9. Kazerouni, I. A., Fitzgerald, L., Dooly, G., & Toal, D. 2022. A survey of state-of-the-art on visual SLAM. *Expert Systems with Applications*, 205, 117734.
10. Zhu, J., Li, H., & Zhang, T. 2023. Camera, LiDAR, and IMU based multi-sensor fusion SLAM: A survey. *Tsinghua Science and Technology*, 29(2), 415-429.
11. Yarovoi, A., & Cho, Y. K. 2024. Review of simultaneous localization and mapping (SLAM) for construction robotics applications. *Automation in Construction*, 162, 105344.
12. Saleem, H., Malekian, R., & Munir, H. 2023. Neural network-based recent research developments in SLAM for autonomous ground vehicles: A review. *IEEE Sensors Journal*, 23(13), 13829-13858.
13. Wang, S., Zheng, D., & Li, Y. 2023. LiDAR-SLAM loop closure detection based on multi-scale point cloud feature transformer. *Measurement Science and Technology*, 35(3), 036305.
14. Best, P. J. 1992. A method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Vision*, 14, 239-256.
- Hu, C., Wang, R., Yan, F., and Chen, N. 2015. Output constraint control on path following of four-wheel independently actuated autonomous ground vehicles. *IEEE Transactions on Vehicular Technology*. 65(6): 4033-4043.
15. Chen, Y., & Medioni, G. 1992. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3), 145-155.
16. Censi, A. (2008, May). An ICP variant using a point-to-line metric. In 2008 IEEE international conference on robotics and automation (pp. 19-25). IEEE.
17. Low, K. L. 2004. Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina*, 4(10), 1-3.
18. Chen, Z., Xu, Y., Yuan, S., & Xie, L. 2024. iG-LIO: An incremental Gicp-based tightly-coupled LiDAR-inertial odometry. *IEEE Robotics and Automation Letters*.
19. Serafin, J., & Grisetti, G. (2015, September). NICP: Dense normal based point cloud registration. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 742-749). IEEE.
20. Yang, J., Li, H., Campbell, D., & Jia, Y. 2015. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11), 2241-2254.
21. Kim, H., Song, S., & Myung, H. 2019. GP-ICP: Ground plane ICP for mobile robots. *IEEE Access*, 7, 76599-76610.
22. Wang, J., Xu, M., Foroughi, F., Dai, D., & Chen, Z. 2021. Fastergicp: Acceptance-rejection sampling based 3d lidar odometry. *IEEE Robotics and Automation Letters*, 7(1), 255-262.
23. Zhang, J., & Singh, S. (2014, July). LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and systems* (Vol. 2, No. 9, pp. 1-9).
24. Oelsch, M., Karimi, M., & Steinbach, E. 2021. R-LOAM: Improving LiDAR odometry and mapping with point-to-mesh features of a known 3D reference object. *IEEE Robotics and Automation Letters*, 6(2), 2068-2075.
25. Guo, H., Zhu, J., & Chen, Y. 2022. E-LOAM: LiDAR odometry and mapping with expanded local structural information. *IEEE Transactions on Intelligent Vehicles*, 8(2), 1911-1921.
26. Zhang, H., Xiao, R., Li, J., Yan, C., & Tang, H. 2024. A High-Precision LiDAR-Inertial Odometry via Invariant Extended Kalman Filtering and Efficient Surfel Mapping. *IEEE Transactions on Instrumentation and Measurement*.
27. Liu, H., Gong, J., & Zhang, Y. 2022. Extraction method of small agricultural AGV navigation baseline based on crop row characteristics. *Journal of Jiangsu University (Natural Science Edition)*, 43(3), 296-301.
28. Fan, Y., Shi, L., Su, W., & Yan, H. 2023. Lane detection algorithm based on PINet + RESA network. *Journal of Jiangsu University (Natural Science Edition)*, 44(4), 373-378.
29. He, Y., Yang, J., Hou, X., Pang, S., & Chen, J. 2021. ICP registration with DCA descriptor for 3D point clouds. *Optics express*, 29(13), 20423-20439.
30. Biber, P., & Straßer, W. (2003, October). The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*(Cat. No. 03CH37453) (Vol. 3, pp. 2743-2748). IEEE.
31. Das, A., & Waslander, S. L. 2014. Scan registration using segmented region growing NDT. *The International Journal of Robotics Research*, 33(13), 1645-1663.
32. Shen, Y., Zhang, B., Wang, J., Zhang, Y., Wu, Y., Chen, Y., & Chen, D. 2024. MI-NDT: Multi-scale Iterative Normal Distribution Transform for Registering Large-scale Outdoor Scans. *IEEE Transactions on Geoscience and Remote Sensing*.
33. Gupta, H., Lilienthal, A. J., Andreasson, H., & Kurtser, P. 2023. NDT-6D for color registration in agri-robotic applications. *Journal of Field Robotics*, 40(6), 1603-1619.

34. Chen, S., Ma, H., Jiang, C., Zhou, B., Xue, W., Xiao, Z., & Li, Q. 2021. NDT-LOAM: A real-time LiDAR odometry and mapping with weighted NDT and LFA. *IEEE Sensors Journal*, 22(4), 3660-3671.
35. Wang, Q., Zhang, J., Liu, Y., & Zhang, X., 2020. Point Cloud Registration Algorithm Based on Combination of NDT and ICP. *Computer Engineering and Applications.*, 56(07), 88-95.
36. Yang, J., Wang, C., Luo, W., Zhang, Y., Chang, B., & Wu, M. 2021. Research on point cloud registering method of tunneling roadway based on 3D NDT-ICP algorithm. *Sensors*, 21(13), 4448.
37. Jiang, H., Chen, Y., Shen, Q., Yin, C., & Cai, J. 2024. Semantic closed-loop based visual mapping algorithm for automated valet parking. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 238(8), 2091-2104.
38. Tomono, M. 2020. Loop detection for 3D LiDAR SLAM using segment-group matching. *Advanced Robotics*, 34(23), 1530-1544.
39. Meng, Q., Guo, H., Zhao, X., Cao, D., & Chen, H. 2021. Loop-closure detection with a multiresolution point cloud histogram mode in lidar odometry and mapping for intelligent vehicles. *IEEE/ASME Transactions on Mechatronics*, 26(3), 1307-1317.
40. Shang, G., Jiang, K., Han, J., & Ni, W. 2024. An environmental fruit detection method for light weight orchard. *Journal of Jiangsu University (Natural Science Edition)*, 45(1), 46-52.
41. Song, S., Shi, X., Ma, C., & Mei, X. 2024. MF-LIO: integrating multi-feature LiDAR inertial odometry with FPFH loop closure in SLAM. *Measurement Science and Technology*, 35(8), 086308.
42. Gao, R., Li, Y., Li, B., & Li, G. 2024. FELC-SLAM: feature extraction and loop closure optimized lidar SLAM system. *Measurement Science and Technology*, 35(11), 115112.
43. Wang, Z., Xie, D., Wu, Y., Wu, H., Qi, X., Huang, D., ... & Zhong, R. 2024. Mercator Descriptor: A Novel Global Descriptor for Loop Closure Detection in LiDAR SLAM. *IEEE Sensors Journal*.
44. Kim, G., & Kim, A. (2018, October). Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4802-4809). IEEE.
45. Zhang, X., Zhang, H., Qian, C., Li, B., & Cao, Y. 2023. A LiDAR-intensity SLAM and loop closure detection method using an intensity cylindrical-projection shape context descriptor. *International Journal of Applied Earth Observation and Geoinformation*, 122, 103419.
46. Li, L., Kong, X., Zhao, X., Huang, T., & Liu, Y. 2022. Semantic scan context: a novel semantic-based loop-closure method for LiDAR SLAM. *Autonomous Robots*, 46(4), 535-551.
47. Cattaneo, D., Vaghi, M., & Valada, A. 2022. Lcdnet: Deep loop closure detection and point cloud registration for lidar slam. *IEEE Transactions on Robotics*, 38(4), 2074-2093.
48. Qin, Y., & Zhang, C. 2024. Optimization of automatic navigation control system of unmanned working ship. *Journal of Jiangsu University (Natural Science Edition)*, 45(4), 417-425.
49. Zhang, Z., Liang, J., Zhang, Z., Chen, X., Chen, L., Wei, W., & Li, H. 2024. Cooperative tracking of multiple maneuvering targets based on AIMM-PF. *Journal of Jiangsu University (Natural Science Edition)*, 45(4), 434-440.
50. Gao, J., Sha, J., Wang, Y., Wang, X., & Tan, C. 2024. A fast and stable GNSS-LiDAR-inertial state estimator from coarse to fine by iterated error-state Kalman filter. *Robotics and Autonomous Systems*, 175, 104675.
51. Júnior, G. P. C., Rezende, A. M., Miranda, V. R., Fernandes, R., Azpúrua, H., Neto, A. A., ... & Freitas, G. M. 2022. EKF-LOAM: An adaptive fusion of LiDAR SLAM with wheel odometry and inertial data for confined spaces with few geometric features. *IEEE Transactions on Automation Science and Engineering*, 19(3), 1458-1471.
52. Xu, W., & Zhang, F. 2021. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2), 3317-3324.
53. Lang, F., Ming, R., Yuan, Z., Xu, X., Wu, K., & Yang, X. 2024. Adaptive Global Graph Optimization for LiDAR-Inertial SLAM. *IEEE Robotics and Automation Letters*.
54. Du, T., Shi, S., Zeng, Y., Yang, J., & Guo, L. 2022. An integrated INS/LiDAR odometry/polarized camera pose estimation via factor graph optimization for sparse environment. *IEEE Transactions on Instrumentation and Measurement*, 71, 1-11.
55. Zhao, C., Ma, W., Su, D., Xu, H., & Tan, Y. 2024. Development of harvesting and handling robot system for hilly citrus orchard. *Journal of Jiangsu University (Natural Science Edition)*, 45(2), 167-172.
56. Tsai, T. C., & Peng, C. C. 2024. Ground segmentation based point cloud feature extraction for 3D LiDAR SLAM enhancement. *Measurement*, 236, 114890.
57. Ruetz, F., Hernández, E., Pfeiffer, M., Oleynikova, H., Cox, M., Lowe, T., & Borges, P. (2019, May). Ovp mesh: 3d free-space representation for local ground vehicle navigation. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 8648-8654). IEEE.
58. Cai, Y., Kong, F., Ren, Y., Zhu, F., Lin, J., & Zhang, F. 2023. Occupancy grid mapping without ray-casting for high-resolution LiDAR sensors. *IEEE Transactions on Robotics*.

59. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., ... & Markham, A. 2021. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11), 8338-8354.
60. Montemerlo, M. 2002. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *Proc. of AAAI02*.
61. Grisetti, G., Stachniss, C., & Burgard, W. 2007. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1), 34-46.
62. Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B., & Vincent, R. (2010, October). Efficient sparse pose adjustment for 2D mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 22-29). IEEE.
63. Kohlbrecher, S., Von Stryk, O., Meyer, J., & Klingauf, U. (2011, November). A flexible and scalable SLAM system with full 3D motion estimation. In *2011 IEEE international symposium on safety, security, and rescue robotics* (pp. 155-160). IEEE.
64. Hess, W., Kohler, D., Rapp, H., & Andor, D. (2016, May). Real-time loop closure in 2D LIDAR SLAM. In *2016 IEEE international conference on robotics and automation (ICRA)* (pp. 1271-1278). IEEE.
65. Shan, T., & Englot, B. (2018, October). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4758-4765). IEEE.
66. Lin, J., & Zhang, F. (2020, May). Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV. In *2020 IEEE international conference on robotics and automation (ICRA)* (pp. 3126-3131). IEEE.
67. Wang, H., Wang, C., & Xie, L. 2021. Intensity-slam: Intensity assisted localization and mapping for large scale environment. *IEEE Robotics and Automation Letters*, 6(2), 1715-1721.
68. Li, Z., Ma, L., Han, C., & Wang, S. (2024). Obstacle detection method of lawn mowing robot based on multi-sensor fusion. *Journal of Jiangsu University (Natural Science Edition)*, 45(2), 160-166.
69. Ye, H., Chen, Y., & Liu, M. (2019, May). Tightly coupled 3d lidar inertial odometry and mapping. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 3144-3150). IEEE.
70. Pan, H., Liu, D., Ren, J., Huang, T., & Yang, H. 2024. LiDAR-IMU Tightly-Coupled SLAM Method Based on IEKF and Loop Closure Detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.
71. Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., & Rus, D. (2020, October). Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5135-5142). IEEE.
72. Xu, W., Cai, Y., He, D., Lin, J., & Zhang, F. 2022. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4), 2053-2073.
73. Li, C., Jiang, H., Wang, C., & Ma, S. 2022. Estimation method of vehicle position and attitude based on sensor information fusion. *Journal of Jiangsu University (Natural Science Edition)*, 43(6), 636-644.
74. Zhang, J., & Singh, S. (2015, May). Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE international conference on robotics and automation (ICRA)* (pp. 2174-2181). IEEE.
75. Shan, T., Englot, B., Ratti, C., & Rus, D. (2021, May). Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In *2021 IEEE international conference on robotics and automation (ICRA)* (pp. 5692-5698). IEEE.
76. Lin, J., Zheng, C., Xu, W., & Zhang, F. 2021. R2LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping. *IEEE Robotics and Automation Letters*, 6(4), 7469-7476.
77. Lin, J., & Zhang, F. (2022, May). R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package. In *2022 International Conference on Robotics and Automation (ICRA)* (pp. 10672-10678). IEEE.
78. Liu, J., & Hou, G. 2021. Application prospect of voice technology in agricultural intelligence. *Journal of Jiangsu University (Natural Science Edition)*, 42(5), 540-545.
79. Hong, Z., Petillot, Y., Wallace, A., & Wang, S. 2022. RadarSLAM: A robust simultaneous localization and mapping system for all weather conditions. *The international journal of robotics research*, 41(5), 519-542.
80. Su, Y., Wang, T., Shao, S., Yao, C., & Wang, Z. 2021. GR-LOAM: LiDAR-based sensor fusion SLAM for ground robots on complex terrain. *Robotics and Autonomous Systems*, 140, 103759.
81. Liu, X., Wen, S., Jiang, Z., Tian, W., Qiu, T. Z., & Othman, K. M. 2023. A multi-sensor fusion with automatic vision-LiDAR calibration based on Factor graph joint optimization for SLAM. *IEEE Transactions on Instrumentation and Measurement*.
82. Ma, Z., Jiang, H., Ma, S., & Li, Y. 2023. Intelligent Parking Control Method Based on Multi-Source Sensory Information Fusion and End-to-End Deep Learning. *Applied Sciences*, 13(8), 5003.
83. Niu, J., Qian, K., & Bu, X. 2021. Place recognition method based on YOLOv3 and deep features. *Journal of Jiangsu University (Natural Science Edition)*, 42(6), 733-737.

84. Sun, L., Zhang, Z., Zhong, R., Chen, D., Zhang, L., Zhu, L., ... & Wang, Y. 2022. A weakly supervised graph deep learning framework for point cloud registration. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1-12.
85. Poiesi, F., & Boscaini, D. 2022. Learning general and distinctive 3D local deep descriptors for point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3), 3979-3985.
86. Zheng, Y., Li, Y., Yang, S., & Lu, H. 2022. Global-PBNet: A novel point cloud registration for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(11), 22312-22319.
87. Wang, W., Luo, S., Geng, G., & Liu, J. 2022. Pedestrian trajectory and intention estimation under low brightness based on human key points. *Journal of Jiangsu University (Natural Science Edition)*, 43(4), 400-406.
88. Xiang, H., Zhu, X., Shi, W., Fan, W., Chen, P., & Bao, S. 2022. Delightlcd: a deep and lightweight network for loop closure detection in lidar slam. *IEEE Sensors Journal*, 22(21), 20761-20772.
89. Wang, S., Zheng, D., & Li, Y. 2023. LiDAR-SLAM loop closure detection based on multi-scale point cloud feature transformer. *Measurement Science and Technology*, 35(3), 036305.
90. Ma, J., Zhang, J., Xu, J., Ai, R., Gu, W., & Chen, X. 2022. Overlaptransformer: An efficient and yaw-angle-invariant transformer network for lidar-based place recognition. *IEEE Robotics and Automation Letters*, 7(3), 6958-6965.
91. Li, Y., Feng, F., Cai, Y., Li, Z., & Sotelo, M. A. 2023. Localization for intelligent vehicles in underground car parks based on semantic information. *IEEE Transactions on Intelligent Transportation Systems*.
92. Chen, X., Milioto, A., Palazzolo, E., Giguere, P., Behley, J., & Stachniss, C. (2019, November). Suma++: Efficient lidar-based semantic slam. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4530-4537). IEEE.
93. Wen, S., Wang, T., & Tao, S. 2022. Hybrid CNN-LSTM architecture for LiDAR point clouds semantic segmentation. *IEEE Robotics and Automation Letters*, 7(3), 5811-5818.
94. Li, W., Zhan, L., Min, W., Zou, Y., Huang, Z., & Wen, C. 2023. Semantic Segmentation of Point Cloud With Novel Neural Radiation Field Convolution. *IEEE Geoscience and Remote Sensing Letters*.
95. Luo, H., Chen, C., Fang, L., Khoshelham, K., & Shen, G. 2020. MS-RRFSegNet: Multiscale regional relation feature segmentation network for semantic segmentation of urban scene point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12), 8301-8315.
96. Fu, X., He, Z., Huang, B., Pei, B., & Yang, F., 2021. Slope identification algorithm based on multi-information data fusion filtering. *Journal of Jiangsu University (Natural Science Edition)*, 42(2), 173-179.
97. Li, S., Wang, L., Li, J., Tian, B., Chen, L., & Li, G. 2021. 3D LiDAR/IMU calibration based on continuous-time trajectory estimation in structured environments. *IEEE Access*, 9, 138803-138816.
98. Yang, W., Gong, Z., Huang, B., & Hong, X. 2022. Lidar with velocity: Correcting moving objects point cloud distortion from oscillating scanning lidars by fusion with camera. *IEEE Robotics and Automation Letters*, 7(3), 8241-8248.
99. Tian, L., Jin, J., & Zheng, Q. 2024. Road scene pedestrian detection based on detection-enhanced YOLOv3-tiny. *Journal of Jiangsu University (Natural Science Edition)*, 45(4), 441-448.
100. Gong, J., Chen, T., Zhang, Y., & Lan, Y. 2022. An improved interframe differential target detection and tracking algorithm based on multi-region information fusion constraints. *Journal of Jiangsu University (Natural Science Edition)*, 43(3), 302-309.
101. Tian, X., Zhu, Z., Zhao, J., Tian, G., & Ye, C. 2023. DI-slot: Tightly-coupled dynamic lidar slam and 3d object tracking based on collaborative graph optimization. *IEEE Transactions on Intelligent Vehicles*.
102. Yuan, C., Song, J., He, Y., Jie, S., Chen, L., & Weng, S. 2021. Active collision avoidance algorithm of autonomous vehicle based on pedestrian trajectory prediction. *Journal of Jiangsu University (Natural Science Edition)*, 42(1), 1-8.
103. Ren, X., Duan, N., Cheng, Y., & Miao, Z. 2022. Statistics method of basic seedling based on machine vision. *Journal of Jiangsu University (Natural Science Edition)*, 43(2), 191-194.
104. Lv, X., He, Z., Yang, Y., Nie, J., Dong, Z., Wang, S., & Gao, M. 2024. MSF-SLAM: Multi-Sensor-Fusion-Based Simultaneous Localization and Mapping for Complex Dynamic Environments. *IEEE Transactions on Intelligent Transportation Systems*.
105. Cui, M., Zhu, Y., Liu, Y., Liu, Y., Chen, G., & Huang, K. 2022. Dense depth-map estimation based on fusion of event camera and sparse LiDAR. *IEEE Transactions on Instrumentation and Measurement*, 71, 1-11.
106. Huang, Y., Shan, T., Chen, F., & Englot, B. 2021. DiSCO-SLAM: Distributed scan context-enabled multi-robot lidar slam with two-stage global-local graph optimization. *IEEE Robotics and Automation Letters*, 7(2), 1150-1157.
107. Lajoie, P. Y., & Beltrame, G. 2023. Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. *IEEE Robotics and Automation Letters*, 9(1), 475-482.

108. Chang, Y., Ebadi, K., Denniston, C. E., Ginting, M. F., Rosinol, A., Reinke, A., ... & Carlone, L. 2022. LAMP 2.0: A robust multi-robot SLAM system for operation in challenging large-scale underground environments. *IEEE Robotics and Automation Letters*, 7(4), 9175-9182.
109. Yu, Q., Wang, K., & Wang, H. 2021. A multi-scale YOLOv3 detection algorithm of road scene object. *Journal of Jiangsu University (Natural Science Edition)*, 42(6), 628-633.
110. Wang, L., & Mou, C. 2024. Few shot semantic segmentation algorithm based on meta-learning. *Journal of Jiangsu University (Natural Science Edition)*, 45(5), 574-580.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.