World Electric Vehicle Journal

MDPI

Article

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: liyong@ujs.edu.cn

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 scan matching, loop closure 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

check for updates

Academic Editor: Grzegorz Sierpiński

Received: 27 November 2024
Revised: 14 January 2025
Accepted: 15 January 2025
Published: 21 January 2025

Citation: Li, Y.; An, J.; He, N.; Li, Y.; Han, Z.; Chen, Z.; Qu, Y. A Review of Simultaneous Localization and Mapping Algorithms Based on Lidar. *World Electr. Veh. J.* **2025**, *16*, 56. https://doi.org/10.3390/wevj16020056

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. Lidar SLAM technology has been widely used in many fields in recent years, especially in autonomous driving and exploration robots [2]. In the autonomous driving system, lidar SLAM uses lidar to scan the surrounding environment, collect three-dimensional point cloud data, and build a high-precision environmental map to help autonomous vehicles perceive surrounding obstacles, road signs, road markings, traffic signals, and other information in a complex environment. In mines and tunnels, lidar SLAM can help explore robots to avoid obstacles, plan safe paths, and ensure that robots can work smoothly in small or irregular environments. In space exploration, SLAM technology helps the exploration robot to perform self-positioning and map construction on the surface of the unknown planet to ensure the smooth progress of the exploration mission.

Visual SLAM mainly uses the camera as the core sensor [3]. The camera is favored by many researchers because of its rich texture information, simple operation, and low cost [4]. The camera is mainly divided into a monocular camera, a binocular camera, and a depth camera. The monocular camera uses a single camera to obtain environmental information. Its advantages are a 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 [5]. 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 [6]. Due to the limitation of field of view and resolution, the depth camera is mainly used in indoor scenes.

Visual SLAM relies on the camera to obtain environmental information. In the dark and strong light environment, the image quality captured by the camera is poor, which leads to inaccurate feature extraction and affects the accuracy of positioning and mapping. Visual SLAM makes it difficult to extract effective feature points in low-texture regions, failing feature matching. Compared with visual SLAM, lidar SLAM has the advantages of high-ranging accuracy, insusceptibility to external interference such as illumination and perspective changes, and stable and accurate map construction. It is widely used in the construction of large-scale complex indoor and outdoor scene maps [7]. 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. Two-dimensional lidar can identify obstacles in the lidar scanning plane with fast scanning speed and low cost. However, 2D lidar cannot obtain the height information of the object, so it is often used in indoor service robots [8]. 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 [9]. 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 a fast update speed and high precision, but trees and buildings may block GPS signals [10]. The advantages and disadvantages of various sensors in the lidar SLAM algorithm are shown in Table 1.

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. [11] 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. [12] 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. [13] 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. [14] analyze recent advances in neural network techniques for SLAM-based ground vehicle localization but only summarize the localization module of the SLAM algorithmic framework.

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

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 scan matching, loop closure 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

Lidar measures the location and shape of objects in the surrounding environment by transmitting a laser beam and receiving the signal that the laser beam reflects from the object, calculating the distance of the beam to the target. In 2D lidar SLAM, lidar measures in a single horizontal plane to obtain the plane structure of the environment and generate a plane map. In 3D lidar SLAM, the lidar measures in multiple horizontal planes to obtain the three-dimensional structural information of the object and generate a three-dimensional map. Lidar measurements can provide accurate range information and improve the 3D localization capability of lidar SLAM. Lidar measurements do not depend on the lighting conditions in the environment and can ensure that lidar SLAM can work stably at night, in haze, strong light, or poor lighting environments.

The SLAM system architecture can be roughly divided into four key parts: sensor data scanning and matching, loop closure detection, back-end optimization, and map construction. The SLAM system needs to obtain environmental data through various sensors. The type and number of sensors directly affect the performance and accuracy of the SLAM system. Common sensors include lidar, camera, and IMU. The sensor module sends the collected environmental data to the SLAM front-end for processing and analysis. In the front end, noise outlier removal and environmental feature extraction are performed through data preprocessing. According to these environmental characteristics, point cloud registration is performed between continuous data frames to estimate the pose transformation between adjacent lidar data frames, but 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. Common back-end optimization methods are graph optimization and filtering-based methods. Throughout the SLAM process, loopback detection has been performed, which is used to identify historically identical scenes, achieve closed loops, and eliminate cumulative errors [15]. The system architecture of SLAM is shown in Figure 1.

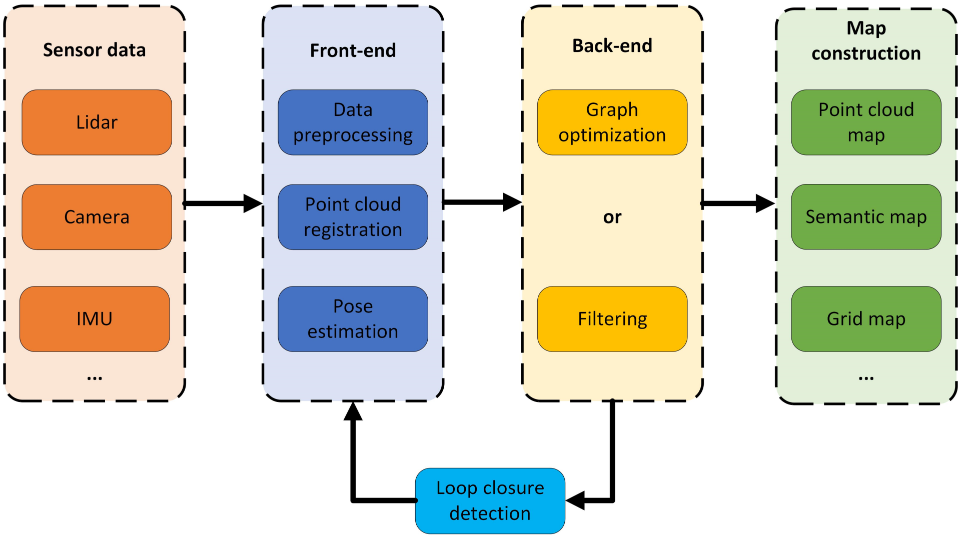

Figure 1. System framework of SLAM.

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. Scan 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 [16] and Chen et al. [17]. 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.

The ICP algorithm finds the optimal rotation and translation transformation between the source point cloud and the target point cloud in an iterative manner to minimize the matching error between the two. The specific steps of ICP registration are as follows. Input two point cloud sets, the source point cloud set $P = \{p_1, p_2, p_3, \ldots, p_\mathrm{n}\}$ and the target point cloud set $Q = \{q_1, q_2, q_3, \ldots, q_\mathrm{n}\}$. The KD-Tree algorithm is used to find the corresponding point of p_i in the source point cloud in the target point cloud q_i so that the Euclidean distance between the two is minimized. According to the corresponding points, the rotation matrix $\boldsymbol{R}$ and the translation matrix $\boldsymbol{T}$ are calculated by the least square method to minimize the error function.

$$E(\boldsymbol{R},\boldsymbol{T}) = \frac{1}{n}\sum_{i=1}^{n} \|p_i' - q_i\|^2 \tag{1}$$

The source point cloud is updated to p' according to the rotation matrix and the translation matrix.

$$P' = \boldsymbol{R}p_i + \boldsymbol{T} \tag{2}$$

Calculate the square distance between the corresponding points. When the distance error d_k is less than the threshold or reaches the number of iterations, the iterative calculation is stopped, otherwise, it returns to Formula 1. According to the final $\boldsymbol{R}$ and $\boldsymbol{T}$, the point cloud P is converted to the coordinate system of point cloud Q to complete the final registration.

$$d_k = \frac{1}{n}\sum_{i=1}^{n} \|p_i' - q_i\|^2 \tag{3}$$

To solve the problem that the traditional ICP algorithm makes it easy to fall into local optimization and to 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 which are suitable for lidar SLAM include PL-ICP (point-to-line ICP) [18], PP-ICP (point-to-plane ICP) [19], GICP (generalized ICP) [20], NICP (normal ICP) [21], 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 ICP algorithm needs to calculate the nearest points between all points in the point cloud and the target point cloud at each iteration, which will increase the amount of calculation, especially for large point cloud data. When processing a point cloud of thousands to tens of thousands of points, the registration time of ICP is from about tens of milliseconds to seconds. For larger point clouds, if the number of iterations is more, the time may increase to a few seconds to tens of seconds. The ICP algorithm is very sensitive to noise. If there are a large number of outliers or wrong matching points in the point cloud, the algorithm may converge to the wrong result.

The Go-ICP algorithm [22] 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 [23] 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 [24] filters the lidar points through a two-step point filter. First, the lidar points are filtered, only the points in the plane 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. Scan matching based on geometric features is shown in Figure 2.

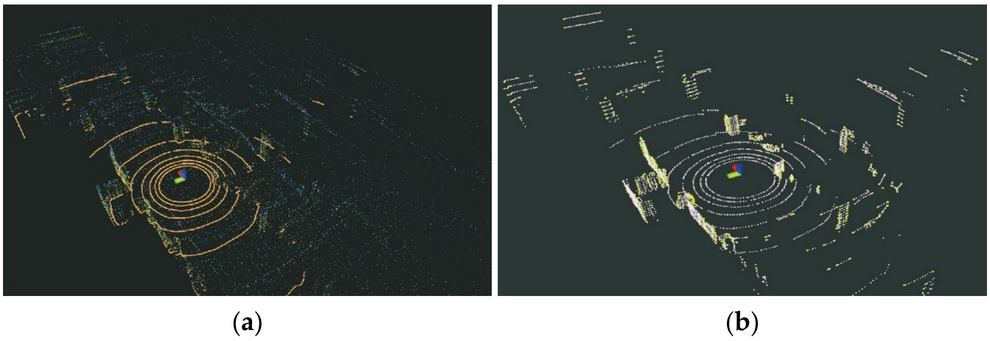

Figure 2. Point cloud matching based on geometric features. (**a**) Point cloud data obtained by lidar (in orange). (**b**) The extracted edge and plane features (in green).

Lidar odometry and mapping (LOAM) [25] 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.

The LOAM algorithm sorts the lidar point cloud according to the curvature size to extract the edge feature points and the plane feature points. The curvature calculation formula of point cloud features is as follows:

$$c = \frac{1}{|S| \cdot \left\| P^L_{(k,i)} \right\|} \left\| \sum_{j \in S, j \neq i} \left(P^L_{(k,i)} - P^L_{(k,j)} \right) \right\| \tag{4}$$

In the formula, S is the neighborhood of the k-frame point cloud, $p^L_{(k,i)}$ and $p^L_{(k,j)}$ are two lidar points in the neighborhood S, and L is the lidar coordinate system.

R-LOAM [26] 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 [27] 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 [28]. SuIn-LIO [29] 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 the point cloud transformation relationship and only uses the geometric features of the point cloud for the 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 [30]. 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 [31]. Given the above problems, adopting a coarse-to-fine hybrid scan 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 and to obtain an accurate pose finally. He et al. [32] 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 scan 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 [33] 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. The NDT algorithm divides the point cloud into small voxels and performs registration by optimizing these voxel models. Since NDT usually only needs to consider the distribution of voxel models rather than each point in the optimization process, the computational efficiency is relatively higher, especially when dealing with large-scale point cloud data. For hundreds of thousands of point clouds, the NDT registration time is usually only a few seconds. Since NDT focuses on the distribution of points within each voxel, rather than the exact pairing between each point, outliers and noise have little effect on the registration results.

The NDT algorithm is calculated by the gridding and normal distribution of the point cloud, and the transformation matrix is solved according to the probability distribution function of the two frame point clouds. The specific process of NDT matching is as follows. The point cloud space is divided into 3D cube cells with uniform size, and the Gaussian probability distribution function is used to represent the probability distribution of lidar points in the cube cells. The expression is as follows:

$$p(x) = \frac{1}{a}\exp\left[-\frac{(x_i - \boldsymbol{q})^{\mathrm{T}}\boldsymbol{C}^{-1}(x_i - \boldsymbol{q})}{2}\right] \tag{5}$$

In the formula, x_i is the point in each cell, $\boldsymbol{C}$ is the covariance matrix of each cell, $\boldsymbol{q}$ is the mean vector of each cell, and a is a constant. $\boldsymbol{C}$ and $\boldsymbol{q}$ in each cell data can be expressed as follows:

$$\boldsymbol{q} = \frac{1}{n}\sum_{i-1}^{n} x_i \tag{6}$$

$$\boldsymbol{C} = \frac{1}{n-1}\sum_{i-1}^{n}(x_i - \boldsymbol{q})(x_i - \boldsymbol{q})^{\mathrm{T}} \tag{7}$$

where x_i (i = 1, 2, 3, . . ., n) is all the points in the cube cell.

The points in the point cloud to be registered are converted to the reference point cloud coordinate system, and then the probability distribution of each projection point is solved and summed.

$$s(p) = \sum_{i=1}^{n}\exp\left[-\frac{(x_i' - \boldsymbol{q})^{\mathrm{T}}\boldsymbol{C}^{-1}(x_i' - \boldsymbol{q})}{2}\right] \tag{8}$$

In the formula, x_i' represents the projection point of each point in the point cloud to be registered in the reference point cloud coordinate system.

The Newton iterative optimization method is used to optimize the objective function $s(p)$, and the pose rotation matrix and translation vector are obtained. Let $f = s(p)$. To minimize the objective function f, the following processing should be carried out in each iterative optimization process.

$$\boldsymbol{H}\Delta p = -\boldsymbol{g} \tag{9}$$

where $\boldsymbol{g}$ is the transpose gradient of function f, and $\boldsymbol{H}$ is the Hessian matrix of function f.

SRG-NDT [34] 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 [35], 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 [36] 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 [37] proposes a method combining weighted NDT and local features 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 scan 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	Medium
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

The matching effect of different lidar scan matching algorithms is shown in Figure 3. Figure 3a is a two-frame point cloud with a large difference in initial position. It can be seen from Figure 3b that the ICP algorithm is sensitive to the initial position of the point cloud, which leads to the failure of point cloud matching. Figure 3c is the matching result based on feature matching. Under the condition of extracting enough feature points, the method based on feature matching has better accuracy. Figure 3d is the matching result of the NDT algorithm. The NDT algorithm has strong robustness and realizes the point cloud registration effect, but the accuracy is not as good as the method based on feature matching.

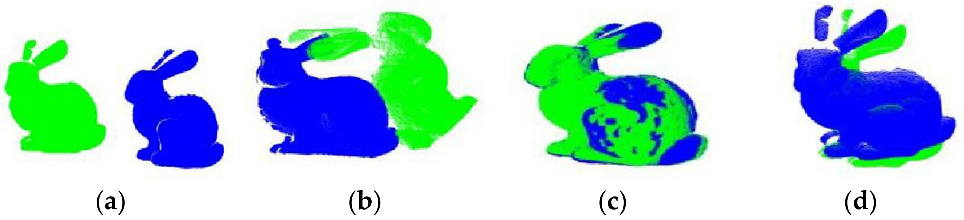

Figure 3. Comparison of the effect of different point cloud matching methods (Green is the source point cloud, and blue is the point cloud to be matched). (**a**) Initial position of point cloud; (**b**) ICP algorithm; (**c**) feature-based algorithm; (**d**) NDT algorithm.

Based on the characteristics of different matching algorithms, some scholars have proposed a combined scan-matching algorithm. Wang et al. [38] 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. [39] 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 Gauss–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. Loop closure detection is a key module that can avoid error accumulation, which is very important for SLAM systems running in large environments with many loops over a long period of time. The main goal of loop closure detection is to detect whether the robot has visited the same position or not, thus correcting localization errors due to drift or environmental changes [40]. 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 loop closure detection is shown in Figure 4.

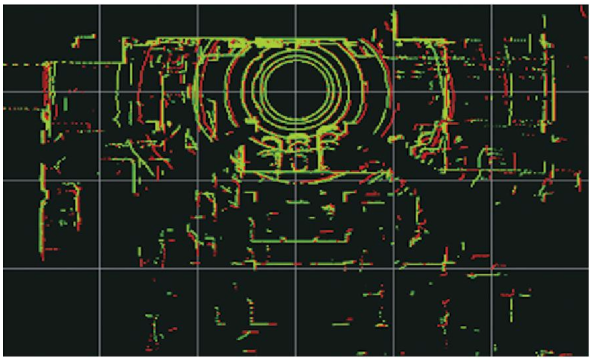

Figure 4. Closed loop detection (the green is the current frame point cloud, and the red is the historical loop-closure frame point cloud).

Tomono [41] 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 loop closure detection and then used the iterative closest point algorithm for fine estimation. Meng et al. [42] developed a loop closure 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, 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 loop closure 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 [43].

The local descriptor is extracted from the position of each feature point and matched based on the historical scene, which improves the efficiency of loop closure detection. Some key point detection methods and local descriptors are proposed, such as 3DSift, 3DHarris, 3D-SURF, ISHOT, and FPFH [44]. MF-LIO [45] uses the fast point feature histogram feature descriptor to perform loop closure detection, which improves the positioning and mapping accuracy of the SLAM algorithm. FELC-SLAM [46] introduces a variety of feature descriptors at the back end of the SLAM algorithm for loop closure 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 loop closure detection. Inspired by the Mercator projection and depth map, Wang et al. [47] propose a new global Mercator descriptor to improve the accuracy and recall rate of loop closure detection. Kim et al. [48] 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 loop closure detection module to achieve robust and efficient results. Zhang et al. [49] 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 loop closure detection in sparse scenes. Li et al. [50] propose a semantic scanning context location recognition method that uses semantic descriptors to encode scene information and uses two-step global ICP to eliminate the influence of rotation and translation on descriptor matching.

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 a lack of loop closure 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) [51], extended Kalman filter (EKF) [52], unscented Kalman filter (UKF) [53], error state Kalman filter (ESKF) [54], and particle filter (PF) [55]. 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. [56] 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 [57] 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 [58] 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 extended 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 main disadvantage of the filtering algorithm is that the computational complexity is significantly increased due to the growth of the state vector. Another disadvantage is that the cumulative error cannot be corrected, and long-term operation in the scene with large noise and multiple loops will cause serious error drift. 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 it 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 factor graph is composed of the state node $\boldsymbol{x}_k$ representing the optimization variable and the factor node f_k representing the observation constraint. When the state change of the mobile robot exceeds a certain threshold, new state nodes and corresponding constraint factor nodes will be added to the factor graph to form the objective function as shown below.

$$\begin{aligned} F(\boldsymbol{x}_k) = &\| f_k^I(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k) \|^2 \boldsymbol{C}_k^I + \| f_k^L(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k) \|^2 \boldsymbol{C}_k^L \\ &+ \| f_k^G(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k) \|^2 \boldsymbol{C}_k^G + \| f_k^B(\boldsymbol{x}_{k-1}, \boldsymbol{x}_k) \|^2 \boldsymbol{C}_k^B \end{aligned} \tag{10}$$

In the formula, $f_k^I(\cdot)$ is the IMU pre-integration factor, $f_k^L(\cdot)$ is the lidar odometer factor, $f_k^G(\cdot)$ is the ground constraint factor, $f_k^B(\cdot)$ is the loop closure detection factor, and $C_k^{\cdot}$ is the covariance matrix of different factors.

The Baidu Apollo automatic driving platform performs automatic driving in a complex urban environment, involving multi-lane, long-distance, and high-density road conditions, requiring high-precision real-time positioning and map construction. The Apollo platform first uses the EKF method to optimize the back end of SLAM. However, in the case of many obstacles and large road fluctuations, the EKF is prone to filter error accumulation, resulting in a decrease in positioning accuracy. To overcome the error accumulation problem of the filtering method, the Apollo platform introduces a back-end optimization method based on graph optimization. Graph optimization globally optimizes vehicle positioning and map construction by minimizing the error of all constraints, which improves positioning accuracy and reduces map drift.

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 graph optimization algorithm transforms the SLAM problem into an optimization problem of the graph structure, where the nodes of the graph represent the robot pose, and the edges represent the constraint relationship between different nodes. When the closed-loop is detected, the graph optimization introduces new global constraints, performs global optimization, eliminates the previous cumulative error, and obtains more accurate maps and trajectories. 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. [59] designed a dynamic weight estimation algorithm based on sensor and loop closure detection information. The algorithm uses the different accuracies of LiDAR, IMU, and loop closure 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. [60] 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. [61]. 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 [62]. In the field of SLAM, the constructed map is mainly divided into point cloud map, octree map, semantic map, and so on [63].

The point cloud map is a collection of three-dimensional space points obtained by sensors such as lidar. Each point represents an actual position in the space, and each point usually contains position coordinates (x, y, z), and sometimes includes additional information such as color, reflection intensity, etc. The point cloud map has high precision and can reconstruct the three-dimensional structure of the real environment, but there are problems with large amounts of data and a lack of semantic information. The octree map is a spatial data structure. Based on the principle of octree, the three-dimensional space is recursively divided into small cube regions. Each node corresponds to a space area, and the state information of the area, such as occupancy or not, is usually stored in the node. The octree map can effectively save storage space, but the construction process of the octree is more complicated. The semantic map contains not only the structural information of the environment but also the semantic information of the objects in the scene (such as object category, function, state, etc.). Semantic maps can provide rich environmental information, but data acquisition is complex and computing resources are consumed.

The 2D lidar SLAM generally uses a grid map to represent the surrounding obstacle information. The 3D lidar SLAM algorithm intuitively describes the surrounding environment information through the point cloud map, as shown in Figure 5a. 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 5b. 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 5c. The octree map is a special occupancy 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 5d. 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.

Tsai et al. [64] 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. [65] propose a local three-dimensional environment representation method for unmanned ground vehicle navigation, the 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. [66] 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. [67] 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.

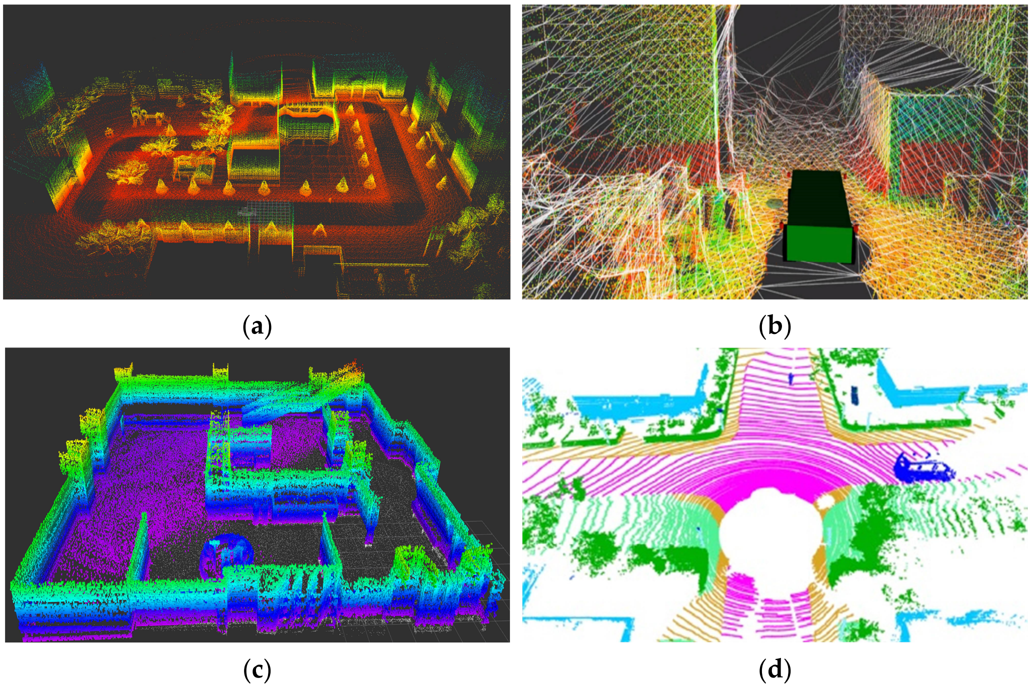

Figure 5. The representation of a map. (**a**) Point cloud map; (**b**) mesh map; (**c**) octree map; (**d**) semantic map.

2.5. Dynamic Point Cloud Recognition

There will inevitably be dynamic objects in the actual working scene of lidar SLAM, which will affect the positioning and mapping of the algorithm. Therefore, identifying and removing dynamic point clouds is also one of the key technologies of lidar SLAM. The conventional lidar SLAM assumes that the environment is static, and the scanned point cloud is directly spliced into a map. The trajectory of the dynamic object motion will be retained in the point cloud map to form a residual shadow that does not exist in the real environment. Figure 6 is a schematic diagram of the point cloud of the dynamic object. The red point cloud in the figure is the residual shadow left by the dynamic object. This will affect the subsequent navigation and path planning of autonomous vehicles. It is necessary to identify and remove the dynamic point cloud in the point cloud map so that only the static point cloud is retained in the map.

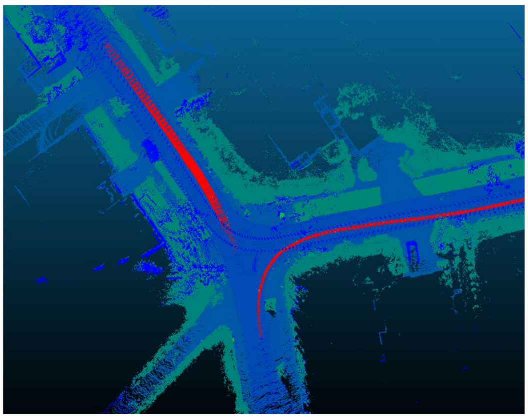

Figure 6. Dynamic object point cloud.

According to the principle, the dynamic point cloud recognition method is mainly divided into grid-based, visibility-based, and deep learning-based methods. The principle of the grid-based method is to divide the three-dimensional space into a spatial grid; the lidar point cloud falls into different grids according to the coordinates and the dynamic object is identified by comparing the situation of the grid being hit and penetrated. The principle of the visibility-based method is to compare the point clouds of the current frame and the historical frame. If the point cloud of the historical frame passes through the point cloud of the current frame, it can be considered that the point cloud of the historical frame is dynamic. This kind of method does not need to establish a grid map and has low requirements for computing resources. The principle of the deep learning method is to semantically segment the lidar point cloud through a neural network, classify each point in the point cloud, and identify and remove dynamic points according to the category of points. Identifying and removing dynamic point clouds can not only improve the accuracy of lidar SLAM but also provide a static environment map with practical application value for subsequent path planning and control.

3. SLAM Algorithm Scheme Based on Lidar

3.1. SLAM Algorithm Based on Pure Lidar

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

In the 2D lidar SLAM algorithm, Lee et al. [68] 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, more particles are needed to obtain better estimation, and the resampling of particles will also lead to particle dissipation. GMapping [69] 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 Rao–Blackwellized Particle Filter 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 loop closure detection, and there is a large cumulative error in long-term operation in large outdoor scenes.

Karto [70] 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 loop closure 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 [71] uses the Gauss–Newton method to directly perform scan-to-map matching to construct a 2D grid map. Due to the lack of loop closure detection in Hector, once the map is wrong, it will affect the subsequent mapping effect, and, since 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 [72] algorithm, adding sensor time synchronization and laser data preprocessing, introducing the concept of submap, using a combination of correlative scan matching 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. Table 4 is a summary of the 2D lidar SLAM algorithm.

Table 4. Two-dimensional 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 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	Without a loop closure detection function, it is easy to drift when rotating too fast.
Cartographer	Graph optimization	Yes	High precision and strong real-time, accelerate loop closure 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 7. LOAM divides the SLAM problem into high-frequency, low-precision motion estimation and low-frequency, high-precision map construction. Then, it fuses the two pieces of data to obtain an accurate point cloud map. The shortcoming of the LOAM algorithm is the lack of back-end optimization and loop closure detection, which will cause drift in large-scale and multi-loop scenarios. LeGO-LOAM [73] 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.

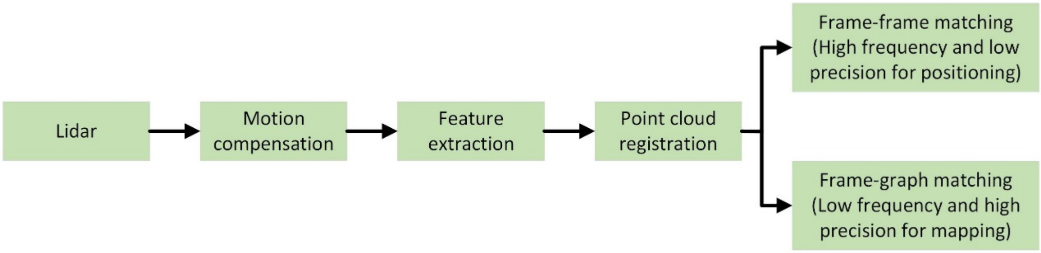
Lidar
Motion compensation
Feature extraction
Point cloud registration
Frame-frame matching (High frequency and low precision for positioning)
Frame-graph matching (Low frequency and high precision for mapping)

Figure 7. 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 [74]. 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 [75]. 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 [76].

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 used 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 [77] 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. [78] use the lidar inertial odometry of the iterative extended Kalman filter to fuse lidar keyframes and IMU information. In loop closure 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 loop closure 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 [79] 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 8. The algorithm uses the factor graph framework to fuse the IMU pre-integration factor, the lidar odometer factor, the GPS factor, and the loop closure 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 [80] 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.

Lidar cannot work normally in degraded scenes and has a poor global positioning ability, while the camera has a 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 [81]. V-LOAM [82] is a general architecture combining a lidar odometer and a visual odometer. The block diagram of the algorithm system is shown in Figure 9. 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 loop closure detection module. LVI-SAM [83] 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 10. 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 scan matching. The advantage of the algorithm is that, if one of the VIS or LIS fails, the LVI-SAM can also work normally, which makes the algorithm have better robustness in an environment with less texture and a lack of features. R^2LIVE [84] 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 easily fails in an unstructured environment. To improve the accuracy and robustness of the algorithm, R^3LIVE is proposed. R^3LIVE [85] 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 R^3LIVE has stronger robustness and accuracy in state estimation than existing systems.

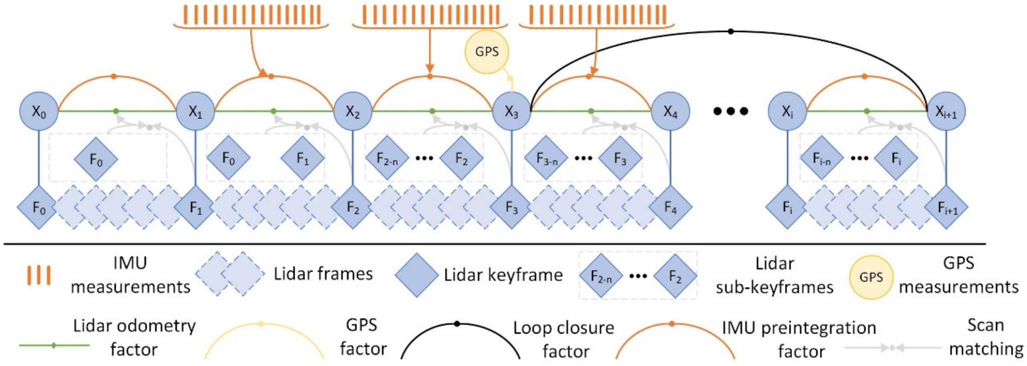

Figure 8. The system architecture of factor graph optimization.

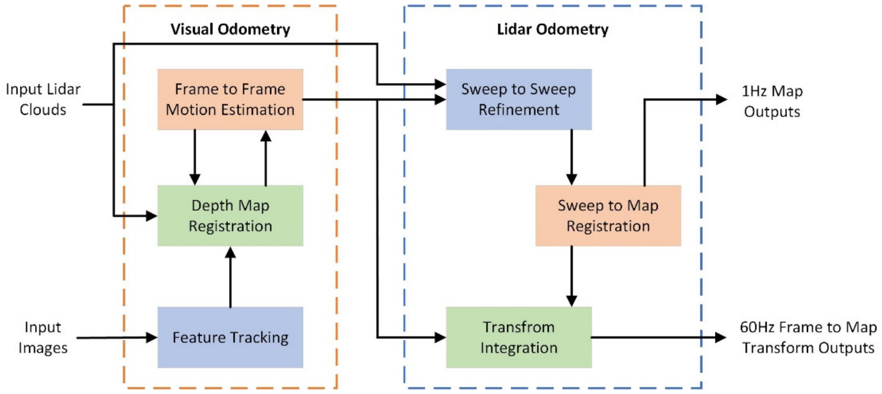

Figure 9. Block diagram of V-LOAM algorithm system.

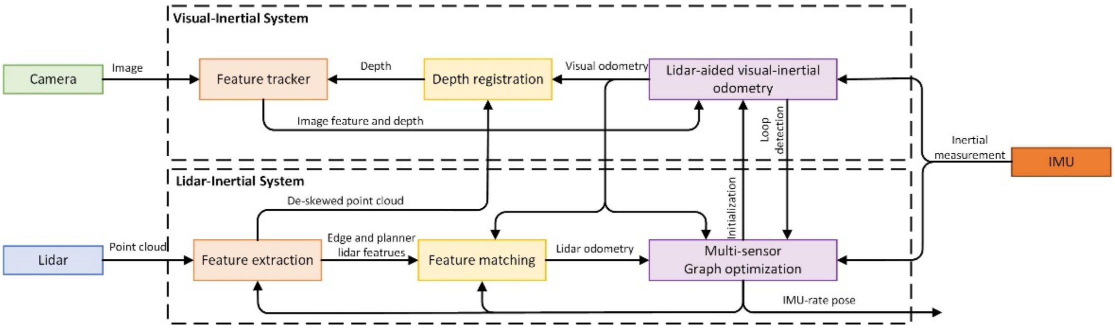

Figure 10. Lidar–visual–inertial tightly coupled system architecture.

To cope with fast motion, noise, and complex application scenarios, many researchers have begun to integrate different types of sensors into the SLAM system [86]. RadarSLAM [87] 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 [88] 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. [89] 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 the lidar SLAM algorithm is mainly applied to several modules in SLAM architecture, such as feature extraction and registration of point clouds, loop closure detection, and the construction of semantic maps. The method based on deep learning uses data-driven learning to obtain more accurate results than manual design [90]. The application of deep learning in feature extraction and loop closure 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 an 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 [91]. Sun et al. [92] innovatively propose an MLP_GCN module that combines a multi-layer perceptron and a 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. [93] 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. [94] 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 the 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 [95].

Accurate loop closure detection can eliminate the cumulative error of the SLAM algorithm and improve the accuracy of the algorithm. The traditional loop closure 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 loop closure detection features, deep learning can extract features with high discrimination and improve the accuracy of the closed loop through a large amount of data training and a reasonable network model [96].

In recent years, loop closure detection algorithms based on deep learning have been continuously developed. DeLightLCD [97] is a deep and lightweight loop closure 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 loop closure detection without pre-pose, but the input of the network must be a two-dimensional depth image after point cloud conversion. Wang et al. [98] propose a loop closure 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. The Overlap-Transformer [99] is a lightweight transformer network, which uses the distance image of a 3D point cloud projection and transformer attention mechanism to generate yaw angle invariant descriptors for loop closure 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 [100]. SuMa++ [101] 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. [102] 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. [103] 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. [104] 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 that cannot 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 to problem arises that the data sampling time is not synchronized [105]. 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. [106] 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. [107] designed 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 scan 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 the lidar SLAM algorithm [108]. 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 [109]. 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 [110]. DL-SLOT [111] 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 the multi-beam lidar, high-precision IMU, and differential GPS positioning modules used in the 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 or on small equipment and improve the popularization of its applicable objects [112]. 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 [113]. The pursuit of lightweight products and the miniaturization of the 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 in the lidar SLAM algorithm. Single-sensor information has certain limitations and cannot meet the mapping needs of complex environments [114]. 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 highly robust maps. MSF-SLAM [115] 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 occupy a small amount of memory, are of high quality, and do 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 a high frame rate [116]. The use of new sensors provides more possibilities for the development of lidar SLAM algorithms.

(4) Multi-robot collaborative SLAM algorithm.

The multi-robot collaborative SLAM algorithm is the current research hotspot and development direction. The 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 nature and accuracy of collaborative mapping. The distributed multi-robot cooperative SLAM system should have strong robustness and fault tolerance and be able to deal with the problems 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 [117], Swarm-SLAM [118], and LAMP 2.0 [119].

(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 loop closure 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 [120]. The addition of semantic information realizes the classification and labeling of point cloud objects and enriches map information [121]. The loop closure 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, loop closure 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 the 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 its practical use, such as the difficulty of the 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.

Author Contributions: Conceptualization, Y.L. (Yong Li) and J.A.; methodology, J.A.; software, J.A. and Y.L. (Yong Li); validation, J.A.; formal analysis, N.H. and Y.L. (Yong Li); investigation, J.A. and Y.L. (Yong Li); resources, Z.H. and Z.C.; data curation, J.A. and Y.Q.; writing—original draft preparation, J.A.; writing—review and editing, J.A. and Y.L. (Yong Li); visualization, Y.Q. and Z.C; supervision, Z.H. and Y.L. (Yanbo Li); project administration, N.H. and Y.L. (Yong Li); funding acquisition, Y.L. (Yong Li). All authors have read and agreed to the published version of the manuscript.

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 Mountainious 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: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Yanbo Li, Zhenyu Han, and Zishan Chen are employees of Lingyun Industrial Co., Ltd. The paper reflects the views of the scientists, and not the company.

References

1. Cai, Y.; Lu, Z.; Wang, H.; Chen, L.; Li, Y. A lightweight feature map creation method for intelligent vehicle localization in urban road environments. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 8503115. [CrossRef]
2. Jin, Y.; Liu, J.; Xu, Z.; Yuan, S.; Li, P.; Wang, J. Development status and trend of agricultural robot technology. *Int. J. Agric. Biol. Eng.* **2021**, *14*, 1–19. [CrossRef]
3. Chen, J.; Song, J.; Guan, Z.; Lian, Y. Measurement of the distance from grain divider to harvesting boundary based on dynamic regions of interest. *Int. J. Agric. Biol. Eng.* **2021**, *14*, 226–232. [CrossRef]
4. Jiang, Z.; Gao, X.; Li, Y.; Wang, H.; Zhang, Y.; Fang, X. Multilayer map construction and vision-only multi-scale localization for intelligent vehicles in underground parking. *Meas. Sci. Technol.* **2022**, *33*, 115021. [CrossRef]
5. Zhang, X.; Yan, Y.; Liu, Y.; Yuan, C. Vision detection method of car body relative parking position based on close-range images of binocular lens. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2023**, *44*, 262–269.
6. Rao, Z.; Cai, Y.; Wang, H.; Chen, L.; Li, Y. A multi-stage model for bird's eye view prediction based on stereo-matching model and RGB-D semantic segmentation. *IET Intell. Transp. Syst.* **2023**, *18*, 2552–2564. [CrossRef]
7. Li, S.; Sun, X.; Zhang, M. Vehicle recognition technology at urban intersection based on fusion of LiDAR and camera. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2023**, *45*, 621–628.
8. Wu, C.; Li, X. Autonomous navigation method of orchard robot based on single laser. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2023**, *44*, 530–539.
9. Yang, S.; Dai, Z.; Yang, X.; Tang, L. Bridge state evaluation and numerical simulation analysis based on millimeter-wave radar measurement and control. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2023**, *44*, 606–613.
10. Yuan, C.; Zhu, H.; He, Y.; Jie, S.; Chen, L. Autonomous navigation algorithm for intelligent vehicle in weak GPS environment. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2023**, *44*, 22–28.
11. Kazerouni, I.A.; Fitzgerald, L.; Dooly, G.; Toal, D. A survey of state-of-the-art on visual SLAM. *Expert Syst. Appl.* **2022**, *205*, 117734. [CrossRef]
12. Zhu, J.; Li, H.; Zhang, T. Camera, LiDAR, and IMU based multi-sensor fusion SLAM: A survey. *Tsinghua Sci. Technol.* **2023**, *29*, 415–429. [CrossRef]
13. Yarovoi, A.; Cho, Y.K. Review of simultaneous localization and mapping (SLAM) for construction robotics applications. *Autom. Constr.* **2024**, *162*, 105344. [CrossRef]
14. Saleem, H.; Malekian, R.; Munir, H. Neural network-based recent research developments in SLAM for autonomous ground vehicles: A review. *IEEE Sens. J.* **2023**, *23*, 13829–13858. [CrossRef]
15. Chen, S.; Zhou, B.; Jiang, C.; Xue, W.; Li, Q. A lidar/visual slam backend with loop closure detection and graph optimization. *Remote Sens.* **2021**, *13*, 2720. [CrossRef]
16. Best, P.J. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Vis.* **1992**, *14*, 239–256.
17. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155. [CrossRef]
18. Censi, A. An ICP variant using a point-to-line metric. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 19–25.
19. Low, K.L. *Linear Least-Squares Optimization for Point-to-Plane icp Surface Registration*; University of North Carolina: Chapel Hill, NC, USA, 2004; Volume 4, pp. 1–3.
20. Chen, Z.; Xu, Y.; Yuan, S.; Xie, L. iG-LIO: An incremental Gicp-based tightly-coupled LiDAR-inertial odometry. *IEEE Robot. Autom. Lett.* **2024**, *9*, 1883–1890. [CrossRef]
21. Serafin, J.; Grisetti, G. NICP: Dense normal based point cloud registration. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 742–749.
22. Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2241–2254. [CrossRef]
23. Kim, H.; Song, S.; Myung, H. GP-ICP: Ground plane ICP for mobile robots. *IEEE Access* **2019**, *7*, 76599–76610. [CrossRef]
24. Wang, J.; Xu, M.; Foroughi, F.; Dai, D.; Chen, Z. Fastergicp: Acceptance-rejection sampling based 3d lidar odometry. *IEEE Robot. Autom. Lett.* **2021**, *7*, 255–262. [CrossRef]
25. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*; MIT Press: Cambridge, MA, USA, 2014; Volume 2, pp. 1–9.
26. Oelsch, M.; Karimi, M.; Steinbach, E. R-LOAM: Improving LiDAR odometry and mapping with point-to-mesh features of a known 3D reference object. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2068–2075. [CrossRef]
27. Guo, H.; Zhu, J.; Chen, Y. E-LOAM: LiDAR odometry and mapping with expanded local structural information. *IEEE Trans. Intell. Veh.* **2022**, *8*, 1911–1921. [CrossRef]
28. Zuo, C.; Feng, Z.; Xiao, X. CMDS-SLAM: Real-time efficient centralized multi-robot dense surfel SLAM. *Meas. Sci. Technol.* **2024**, *35*, 116303. [CrossRef]

29. Zhang, H.; Xiao, R.; Li, J.; Yan, C.; Tang, H. A High-Precision LiDAR-Inertial Odometry via Invariant Extended Kalman Filtering and Efficient Surfel Mapping. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 8502911. [CrossRef]
30. Liu, H.; Gong, J.; Zhang, Y. Extraction method of small agricultural AGV navigation baseline based on crop row characteristics. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2022**, *43*, 296–301.
31. Fan, Y.; Shi, L.; Su, W.; Yan, H. Lane detection algorithm based on PINet + RESA network. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2023**, *44*, 373–378.
32. He, Y.; Yang, J.; Hou, X.; Pang, S.; Chen, J. ICP registration with DCA descriptor for 3D point clouds. *Opt. Express* **2021**, *29*, 20423–20439. [CrossRef] [PubMed]
33. Biber, P.; Straßer, W. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, NV, USA, 27–31 October 2003; Volume 3, pp. 2743–2748.
34. Das, A.; Waslander, S.L. Scan registration using segmented region growing NDT. *Int. J. Robot. Res.* **2014**, *33*, 1645–1663. [CrossRef]
35. Shen, Y.; Zhang, B.; Wang, J.; Zhang, Y.; Wu, Y.; Chen, Y.; Chen, D. MI-NDT: Multi-scale Iterative Normal Distribution Transform for Registering Large-scale Outdoor Scans. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5705513. [CrossRef]
36. Gupta, H.; Lilienthal, A.J.; Andreasson, H.; Kurtser, P. NDT-6D for color registration in agri-robotic applications. *J. Field Robot.* **2023**, *40*, 1603–1619. [CrossRef]
37. Chen, S.; Ma, H.; Jiang, C.; Zhou, B.; Xue, W.; Xiao, Z.; Li, Q. NDT-LOAM: A real-time LiDAR odometry and mapping with weighted NDT and LFA. *IEEE Sens. J.* **2021**, *22*, 3660–3671. [CrossRef]
38. Wang, Q.; Zhang, J.; Liu, Y.; Zhang, X. Point Cloud Registration Algorithm Based on Combination of NDT and ICP. *Comput. Eng. Appl.* **2020**, *56*, 88–95.
39. Yang, J.; Wang, C.; Luo, W.; Zhang, Y.; Chang, B.; Wu, M. Research on point cloud registering method of tunneling roadway based on 3D NDT-ICP algorithm. *Sensors* **2021**, *21*, 4448. [CrossRef] [PubMed]
40. Jiang, H.; Chen, Y.; Shen, Q.; Yin, C.; Cai, J. Semantic closed-loop based visual mapping algorithm for automated valet parking. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2024**, *238*, 2091–2104. [CrossRef]
41. Tomono, M. Loop detection for 3D LiDAR SLAM using segment-group matching. *Adv. Robot.* **2020**, *34*, 1530–1544. [CrossRef]
42. Meng, Q.; Guo, H.; Zhao, X.; Cao, D.; Chen, H. Loop-closure detection with a multiresolution point cloud histogram mode in lidar odometry and mapping for intelligent vehicles. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 1307–1317. [CrossRef]
43. Shang, G.; Jiang, K.; Han, J.; Ni, W. An environmental fruit detection method for light weight orchard. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2024**, *45*, 46–52.
44. Liu, L.; Bai, Y.; Wang, Y.; Sun, Z. Point cloud registration method based on 3DSIFT and BSHOT features. *Laser Infrared* **2021**, *51*, 848–852.
45. Song, S.; Shi, X.; Ma, C.; Mei, X. MF-LIO: Integrating multi-feature LiDAR inertial odometry with FPFH loop closure in SLAM. *Meas. Sci. Technol.* **2024**, *35*, 086308. [CrossRef]
46. Gao, R.; Li, Y.; Li, B.; Li, G. FELC-SLAM: Feature extraction and loop closure optimized lidar SLAM system. *Meas. Sci. Technol.* **2024**, *35*, 115112. [CrossRef]
47. Wang, Z.; Xie, D.; Wu, Y.; Wu, H.; Qi, X.; Huang, D.; Zhong, R. Mercator Descriptor: A Novel Global Descriptor for Loop Closure Detection in LiDAR SLAM. *IEEE Sens. J.* **2024**, *24*, 35730–35742. [CrossRef]
48. Kim, G.; Kim, A. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4802–4809.
49. Zhang, X.; Zhang, H.; Qian, C.; Li, B.; Cao, Y. A LiDAR-intensity SLAM and loop closure detection method using an intensity cylindrical-projection shape context descriptor. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *122*, 103419. [CrossRef]
50. Li, L.; Kong, X.; Zhao, X.; Huang, T.; Liu, Y. Semantic scan context: A novel semantic-based loop-closure method for LiDAR SLAM. *Auton. Robot.* **2022**, *46*, 535–551. [CrossRef]
51. Qin, Y.; Zhang, C. Optimization of automatic navigation control system of unmanned working ship. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2024**, *45*, 417–425.
52. Zhou, Z.; Wang, D.; Xu, B. A multi-innovation with forgetting factor based EKF-SLAM method for mobile robots. *Assem. Autom.* **2021**, *41*, 71–78. [CrossRef]
53. Bahraini, M. On the efficiency of SLAM using adaptive unscented Kalman filter. *Iran. J. Sci. Technol. Trans. Mech. Eng.* **2020**, *44*, 727–735. [CrossRef]
54. Xu, Z.; Li, H.; Zi, Y.; Guo, S.; Cui, G. An ESKF Based SLAM Approach with Millimeter Wave Radar and IMU. In Proceedings of the International Conference on Autonomous Unmanned Systems, Xi'an, China, 23–25 September 2022; pp. 2301–2310.
55. Zhang, Z.; Liang, J.; Zhang, Z.; Chen, X.; Chen, L.; Wei, W.; Li, H. Cooperative tracking of multiple maneuvering targets based on AIMM-PF. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2024**, *45*, 434–440.

56. Gao, J.; Sha, J.; Wang, Y.; Wang, X.; Tan, C. A fast and stable GNSS-LiDAR-inertial state estimator from coarse to fine by iterated error-state Kalman filter. *Robot. Auton. Syst.* **2024**, *175*, 104675. [CrossRef]
57. Júnior, G.P.C.; Rezende, A.M.; Miranda, V.R.; Fernandes, R.; Azpúrua, H.; Neto, A.A.; Freitas, G.M. EKF-LOAM: An adaptive fusion of LiDAR SLAM with wheel odometry and inertial data for confined spaces with few geometric features. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 1458–1471. [CrossRef]
58. Xu, W.; Zhang, F. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3317–3324. [CrossRef]
59. Lang, F.; Ming, R.; Yuan, Z.; Xu, X.; Wu, K.; Yang, X. Adaptive Global Graph Optimization for LiDAR-Inertial SLAM. *IEEE Robot. Autom. Lett.* **2024**, *9*, 9868–9875. [CrossRef]
60. Du, T.; Shi, S.; Zeng, Y.; Yang, J.; Guo, L. An integrated INS/LiDAR odometry/polarized camera pose estimation via factor graph optimization for sparse environment. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–11. [CrossRef]
61. Guo, J.; He, B. Improved iSAM based on flexible re-linearization threshold and error learning model for AUV in large scale areas. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7678–7687. [CrossRef]
62. Zhao, C.; Ma, W.; Su, D.; Xu, H.; Tan, Y. Development of harvesting and handling robot system for hilly citrus orchard. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2024**, *45*, 167–172.
63. Sousa, R.; Sobreira, H.; Moreira, A. A systematic literature review on long-term localization and mapping for mobile robots. *J. Field Robot.* **2023**, *40*, 1245–1322. [CrossRef]
64. Tsai, T.C.; Peng, C.C. Ground segmentation based point cloud feature extraction for 3D LiDAR SLAM enhancement. *Measurement* **2024**, *236*, 114890. [CrossRef]
65. Ruetz, F.; Hernández, E.; Pfeiffer, M.; Oleynikova, H.; Cox, M.; Lowe, T.; Borges, P. Ovpc mesh: 3d free-space representation for local ground vehicle navigation. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8648–8654.
66. Cai, Y.; Kong, F.; Ren, Y.; Zhu, F.; Lin, J.; Zhang, F. Occupancy grid mapping without ray-casting for high-resolution LiDAR sensors. *IEEE Trans. Robot.* **2024**, *40*, 172–192. [CrossRef]
67. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Markham, A. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 8338–8354. [CrossRef] [PubMed]
68. Lee, S.; Eoh, G.; Lee, B. Relational FastSLAM: An improved Rao-Blackwellized particle filtering framework using particle swarm characteristics. *Robotica* **2016**, *34*, 1282–1296. [CrossRef]
69. Grisetti, G.; Stachniss, C.; Burgard, W. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans. Robot.* **2007**, *23*, 34–46. [CrossRef]
70. Konolige, K.; Grisetti, G.; Kümmerle, R.; Burgard, W.; Limketkai, B.; Vincent, R. Efficient sparse pose adjustment for 2D mapping. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 22–29.
71. Kohlbrecher, S.; Von Stryk, O.; Meyer, J.; Klingauf, U. A flexible and scalable SLAM system with full 3D motion estimation. In Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 1–5 November 2011; pp. 155–160.
72. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
73. Shan, T.; Englot, B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
74. Lin, J.; Zhang, F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 3126–3131.
75. Wang, H.; Wang, C.; Xie, L. Intensity-slam: Intensity assisted localization and mapping for large scale environment. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1715–1721. [CrossRef]
76. Li, Z.; Ma, L.; Han, C.; Wang, S. Obstacle detection method of lawn mowing robot based on multi-sensor fusion. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2024**, *45*, 160–166.
77. Ye, H.; Chen, Y.; Liu, M. Tightly coupled 3d lidar inertial odometry and mapping. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3144–3150.
78. Pan, H.; Liu, D.; Ren, J.; Huang, T.; Yang, H. LiDAR-IMU Tightly-Coupled SLAM Method Based on IEKF and Loop Closure Detection. *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.* **2024**, *17*, 6986–7001. [CrossRef]
79. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 5135–5142.

80. Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Trans. Robot.* **2022**, *38*, 2053–2073. [CrossRef]
81. Li, C.; Jiang, H.; Wang, C.; Ma, S. Estimation method of vehicle position and attitude based on sensor information fusion. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2022**, *43*, 636–644.
82. Zhang, J.; Singh, S. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2174–2181.
83. Shan, T.; Englot, B.; Ratti, C.; Rus, D. Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 5692–5698.
84. Lin, J.; Zheng, C.; Xu, W.; Zhang, F. R2LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7469–7476. [CrossRef]
85. Lin, J.; Zhang, F. R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 10672–10678.
86. Liu, J.; Hou, G. Application prospect of voice technology in agricultural intelligence. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2021**, *42*, 540–545.
87. Hong, Z.; Petillot, Y.; Wallace, A.; Wang, S. RadarSLAM: A robust simultaneous localization and mapping system for all weather conditions. *Int. J. Robot. Res.* **2022**, *41*, 519–542. [CrossRef]
88. Su, Y.; Wang, T.; Shao, S.; Yao, C.; Wang, Z. GR-LOAM: LiDAR-based sensor fusion SLAM for ground robots on complex terrain. *Robot. Auton. Syst.* **2021**, *140*, 103759. [CrossRef]
89. Liu, X.; Wen, S.; Jiang, Z.; Tian, W.; Qiu, T.Z.; Othman, K. A multi-sensor fusion with automatic vision-LiDAR calibration based on Factor graph joint optimization for SLAM. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 9513809. [CrossRef]
90. Ma, Z.; Jiang, H.; Ma, S.; Li, Y. Intelligent Parking Control Method Based on Multi-Source Sensory Information Fusion and End-to-End Deep Learning. *Appl. Sci.* **2023**, *13*, 5003. [CrossRef]
91. Niu, J.; Qian, K.; Bu, X. Place recognition method based on YOLOv3 and deep features. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2021**, *42*, 733–737.
92. Sun, L.; Zhang, Z.; Zhong, R.; Chen, D.; Zhang, L.; Zhu, L.; Wang, Y. A weakly supervised graph deep learning framework for point cloud registration. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–12. [CrossRef]
93. Poiesi, F.; Boscaini, D. Learning general and distinctive 3D local deep descriptors for point cloud registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3979–3985. [CrossRef] [PubMed]
94. Zheng, Y.; Li, Y.; Yang, S.; Lu, H. Global-PBNet: A novel point cloud registration for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 22312–22319. [CrossRef]
95. Liu, W.; Hu, J.; Liu, J.; Yue, R.; Zhang, T.; Yao, M.; Li, J. Method for the navigation line recognition of the ridge without crops via machine vision. *Int. J. Agric. Biol. Eng.* **2024**, *17*, 230–239.
96. Wang, W.; Luo, S.; Geng, G.; Liu, J. Pedestrian trajectory and intention estimation under low brightness based on human key points. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2022**, *43*, 400–406.
97. Xiang, H.; Zhu, X.; Shi, W.; Fan, W.; Chen, P.; Bao, S. Delightlcd: A deep and lightweight network for loop closure detection in lidar slam. *IEEE Sens. J.* **2022**, *22*, 20761–20772. [CrossRef]
98. Wang, S.; Zheng, D.; Li, Y. LiDAR-SLAM loop closure detection based on multi-scale point cloud feature transformer. *Meas. Sci. Technol.* **2023**, *35*, 036305. [CrossRef]
99. Ma, J.; Zhang, J.; Xu, J.; Ai, R.; Gu, W.; Chen, X. Overlaptransformer: An efficient and yaw-angle-invariant transformer network for lidar-based place recognition. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6958–6965. [CrossRef]
100. Li, Y.; Feng, F.; Cai, Y.; Li, Z.; Sotelo, M.A. Localization for intelligent vehicles in underground car parks based on semantic information. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 1317–1332. [CrossRef]
101. Chen, X.; Milioto, A.; Palazzolo, E.; Giguere, P.; Behley, J.; Stachniss, C. Suma++: Efficient lidar-based semantic slam. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4530–4537.
102. Wen, S.; Wang, T.; Tao, S. Hybrid CNN-LSTM architecture for LiDAR point clouds semantic segmentation. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5811–5818. [CrossRef]
103. Li, W.; Zhan, L.; Min, W.; Zou, Y.; Huang, Z.; Wen, C. Semantic Segmentation of Point Cloud With Novel Neural Radiation Field Convolution. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 6501705. [CrossRef]
104. Luo, H.; Chen, C.; Fang, L.; Khoshelham, K.; Shen, G. MS-RRFSegNet: Multiscale regional relation feature segmentation network for semantic segmentation of urban scene point clouds. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8301–8315. [CrossRef]
105. Fu, X.; He, Z.; Huang, B.; Pei, B.; Yang, F. Slope identification algorithm based on multi-information data fusion filtering. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2021**, *42*, 173–179.

106. Li, S.; Wang, L.; Li, J.; Tian, B.; Chen, L.; Li, G. 3D LiDAR/IMU calibration based on continuous-time trajectory estimation in structured environments. *IEEE Access* **2021**, *9*, 138803–138816. [CrossRef]
107. Yang, W.; Gong, Z.; Huang, B.; Hong, X. Lidar with velocity: Correcting moving objects point cloud distortion from oscillating scanning lidars by fusion with camera. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8241–8248. [CrossRef]
108. Ma, Z.; Yang, S.; Li, J.; Qi, J. Research on slam localization algorithm for orchard dynamic vision based on YOLOD-SLAM2. *Agriculture* **2024**, *14*, 1622. [CrossRef]
109. Tian, L.; Jin, J.; Zheng, Q. Road scene pedestrian detection based on detection-enhanced YOLOv3-tiny. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2024**, *45*, 441–448.
110. Gong, J.; Chen, T.; Zhang, Y.; Lan, Y. An improved interframe differential target detection and tracking algorithm based on multi-region information fusion constraints. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2022**, *43*, 302–309.
111. Tian, X.; Zhu, Z.; Zhao, J.; Tian, G.; Ye, C. Dl-slot: Tightly-coupled dynamic lidar slam and 3d object tracking based on collaborative graph optimization. *IEEE Trans. Intell. Veh.* **2023**, *9*, 1017–1027. [CrossRef]
112. Zhang, T.; Zhou, J.; Liu, W.; Yue, R.; Shi, J.; Zhou, C.; Hu, J. SN-CNN: A Lightweight and Accurate Line Extraction Algorithm for Seedling Navigation in Ridge-Planted Vegetables. *Agriculture* **2024**, *14*, 1446. [CrossRef]
113. Yuan, C.; Song, J.; He, Y.; Jie, S.; Chen, L.; Weng, S. Active collision avoidance algorithm of autonomous vehicle based on pedestrian trajectory prediction. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2021**, *42*, 1–8.
114. Ren, X.; Duan, N.; Cheng, Y.; Miao, Z. Statistics method of basic seedling based on machine vision. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2022**, *43*, 191–194.
115. Lv, X.; He, Z.; Yang, Y.; Nie, J.; Dong, Z.; Wang, S.; Gao, M. MSF-SLAM: Multi-Sensor-Fusion-Based Simultaneous Localization and Mapping for Complex Dynamic Environments. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 19699–19713. [CrossRef]
116. Cui, M.; Zhu, Y.; Liu, Y.; Liu, Y.; Chen, G.; Huang, K. Dense depth-map estimation based on fusion of event camera and sparse LiDAR. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–11. [CrossRef]
117. Huang, Y.; Shan, T.; Chen, F.; Englot, B. DiSCo-SLAM: Distributed scan context-enabled multi-robot lidar slam with two-stage global-local graph optimization. *IEEE Robot. Autom. Lett.* **2021**, *7*, 1150–1157. [CrossRef]
118. Lajoie, P.Y.; Beltrame, G. Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. *IEEE Robot. Autom. Lett.* **2023**, *9*, 475–482. [CrossRef]
119. Chang, Y.; Ebadi, K.; Denniston, C.E.; Ginting, M.F.; Rosinol, A.; Reinke, A.; Carlone, L. LAMP 2.0: A robust multi-robot SLAM system for operation in challenging large-scale underground environments. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9175–9182. [CrossRef]
120. Yu, Q.; Wang, K.; Wang, H. A multi-scale YOLOv3 detection algorithm of road scene object. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2021**, *42*, 628–633.
121. Wang, L.; Mou, C. Few shot semantic segmentation algorithm based on meta-learning. *J. Jiangsu Univ. (Nat. Sci. Ed.)* **2024**, *45*, 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.