IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 January 2024, accepted 22 January 2024, date of publication 19 February 2024, date of current version 23 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3359908

RESEARCH ARTICLE

Tightly-Coupled SLAM Integrating LiDAR and INS for Unmanned Vehicle Navigation in Campus Environments

LINSHUAI ZHANG[1,2,3], QIAN WANG[2], SHUOXIN GU[2], TAO JIANG[1,2], SHIQI JIANG[2,3], JIAJIA LIU[2,4], SHUANG LUO[2], AND GONGJUN YAN[5]

[1]School of Intelligent Medicine, Chengdu University of Traditional Chinese Medicine, Chengdu, Sichuan 611137, China
[2]School of Automation, Chengdu University of Information Technology, Chengdu, Sichuan 610225, China
[3]International Joint Research Center of Robotics and Intelligence System of Sichuan Province, Chengdu University of Information Technology, Chengdu, Sichuan 610225, China
[4]Unmanned System Intelligent Perception Control Technology Engineering Laboratory of Sichuan Province, Chengdu University of Information Technology, Chengdu, Sichuan 610225, China
[5]Department of Computer Science, University of Southern Indiana, Evansville, IN 47712, USA

Corresponding authors: Shuoxin Gu (gsx@cuit.edu.cn) and Tao Jiang (jiangtop@cdutcm.edu.cn)

This work was supported in part by the Key Research and Development Support Program of Chengdu Science and Technology Bureau of Sichuan Province under Grant 2022-YF05-01128-SN; in part by the Research Fund of the Chengdu University of Information Technology under Grant KYTZ202108; in part by the Opening Project of Unmanned System Intelligent Perception Control Technology Engineering Laboratory of Sichuan Province under Grant WRXT2021-002; in part by the Opening Project of International Joint Research Center for Robotics and Intelligence System of Sichuan Province under Grant JQZN2021-001; in part by the Sichuan Science and Technology Planning Project under Grant 2021YFH0069, under Grant 2023YFG0178, and under Grant 2023YFG0045; and in part by the Science and Technology Innovation Ability Improvement Plan of Chengdu University of Information Technology under Grant KYTD202228.

ABSTRACT Simultaneous Localization and Mapping (SLAM) is one of the key issues for mobile robots to achieve true autonomy. The implementations of SLAM could rely on a variety of sensors. Among many types of them, the laser-based SLAM approach is widely used owing to its high accuracy, even in poor lighting conditions. However, when in structure-less environments, laser modules will fail due to a lack of sufficient geometric features. Besides, motion estimation by moving lidar has the problem of distortion since range measurements are received continuously. To solve these problems, we propose a tightly-coupled SLAM integrating LiDAR and an integrated navigation system (INS) for unmanned vehicle navigation in campus environments. On the basis of feature extraction, a constraint equation for inter-frame point cloud features is constructed, and the pose solution results of the INS are added as a priori data for inter-frame point cloud registration. The Levenberg-Marquardt nonlinear least square method is used to solve the constraint equation to obtain inter-frame pose relationships. Map matching and loop closure detection methods are used to optimize the odometer, and the optimal pose information is obtained. The proposed SLAM algorithm is evaluated by comparing with the classic open-source laser SLAM algorithms on the campus dataset. Experimental results demonstrate that our proposed algorithm has certain advantages in estimating the trajectory error of the unmanned vehicle and has higher mapping performance.

INDEX TERMS SLAM, mobile robot, localization and navigation, multi-sensor data fusion, liDAR and INS, high-precision point cloud map.

I. INTRODUCTION

In recent years, the artificial intelligence (AI) industry has obtained unprecedented development opportunities, and various intelligent devices have begun to enter people's daily lives, such as smart homes, cleaning robots, cars, and so on. However, with the continuous improvement of people's quality of life, people hope that robots can complete more human repetitive work, or replace people to complete some difficult and dangerous work. At present, robot technology

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

has received extensive attention, especially the research on mobile robot technology has become a research hotspot. Intelligent mobile robots such as explosive handling robots, substation detection robots and pipe gallery detection robots have been produced and successfully applied in military, civil, industrial and scientific research fields.

Mobile robot technology is a multifunctional integrated system integrating environment perception, dynamic decision and planning, and low-level control. To realize an intelligent mobile robot, it is necessary to solve several problems such as decision processing, autonomous control, environment perception, positioning and navigation. Decision processing includes receiving and analyzing various types of data and making corresponding decisions; Autonomous control includes motion control and control algorithm analysis; Environmental perception, including vision, millimeter-wave radar and liDAR technology, is used to identify the surrounding environment; Positioning and navigation Obtain real-time vehicle position information, including Global Positioning System (GPS), Inertial Navigation System (INS) and Simultaneous Localization and Mapping (SLAM) technology. SLAM technology has always been the core issue of positioning and navigation of mobile robots. As a technology of great practical significance, the core process can be summarized into three steps, including pre-processing, matching and map fusion [1]. The mobile robot carrying the sensor can explore the unknown environment, process the sensor data to estimate the robot's state, and then complete the construction of the geometric information of the environment, that is, build the map according to the measurement of environmental information. The robot's state is described by its attitude, and the state can also contain other quantities, such as robot speed, sensor deviation, and calibration parameters. The built map is usually a representation of the area of interest. The constructed environment map can be used for robot motion planning and auxiliary positioning.

The concept of SLAM began in the 1980s and was first proposed by Cheeseman [2] at the Internet Content Rating Association (ICRA) in 1986. In 1995, Durrant-Whyte [3] first proposed the algorithm framework for SLAM problems, including data processing, odometer estimation, back-end optimization and map construction, and gave the convergence test results for SLAM problems. Probability estimation theory is also gradually applied to SLAM, including the famous Extended Kalman Filters (EKF) [4] and Rao-Blackwellized Particle Filters (RBPF) [5] and Maximum Likelihood Estimation (MLE), the difficulty in classical times is efficiency and robustness of data association. Gmapping [6], a mapping algorithm widely used at present, uses particle filtering to locate and construct raster maps. Yang et al. [7] from Didi proposed a robust mapping method suitable for city-scale liDAR, which introduced a graph optimization structure and used a multi-hypothesis extended Kalman filter to remove dynamic objects. SLAM++ proposed by Salas-Moren et al. [8] in 2013 can be regarded as the earliest SLAM algorithm using semantic information. In recent years, a series of LLOAM [9], SegMap [10], CubeSLAM [11], [12] based on point cloud segmentation, SuMa++ [13] based on semantic information, ApriISAM [14] and deep learn-based GEN-SLAM [15] have appeared in various peaks [16]. Gmapping is based on 2D liDAR and LOAM is based on 3D liDAR [17], [18]. Among them, as 3D liDAR technology is paid more and more attention and applied more and more widely, 3D liDAR SLAM technology is also more and more. Shan et al. [19] proposed a SLAM system capable of real-time pose estimation and ground optimization in ground vehicles with low-power embedded systems. Lin et al. [20] proposed A robust and real-time liDAR SLAM algorithm for small scenes and irregular sampling, which improved the accuracy and efficiency compared with the existing A-LOAM [21] by improving the front-end and back-end. Reference [22] open source a correction framework for 3D laser SLAM, Interactive SLAM. Compared with the current framework of pure laser SLAM, Interactive SLAM can manually add or subtract some constraint information, to make the point cloud map better.

Since a single sensor often has its observation error which is inevitable and will gradually increase with time, the equipotential pose measurement sensor of an inertial navigation system is usually introduced to estimate the current pose as prior information of the perception sensor, which can eliminate certain accumulated errors. Zuo et al. [23] proposed a tightly coupled mileage calculation method based on the Fusion of multiple sensors including liDAR, inertial navigation and a camera, LIC Fusion, which can calibrate the spatial and temporal relationship parameters of the three sensors online. The robustness is superior to the existing visual inertial odometer and liDAR odometer methods in severe motion. Koide et al. [24] open-source a 6-DOF 3D liDAR SLAM algorithm, HDL_graph_SLAM, on GitHub. 3D liDAR is mainly used to construct laser odometer, loop-back detection and back-end image optimization, and IMU, GPS and other information are also taken as additional constraints. Shao et al. [25] proposed a SLAM framework based on binocular vision inertial radar in IROS2019, VIL-SLAM. By combining a tightly coupled stereovision inertial odometer with liDAR construction drawing and liDAR enhanced visual closed-loop, the closed-loop corrected 6-DOF 3D liDAR attitude can be obtained in real time. Mc2-SLAM [26] is a real-time laser odometer system based on non-rigid matching, which performs point cloud distortion compensation and point cloud matching in one optimization task and also uses IMU pre-integration to improve accuracy, which is second only to LOAM in KITTI data performance. LIO jointly minimizes the cost of LiDAR and IMU measurement values, and the drift of the odometer in long-term operation is also within an acceptable range, which makes up for the deficiency of independent sensors and proposes a tightly coupled method [27], [28]. A computationally efficient and robust LiDAR-inertial odomy framework was proposed that fuses

LiDAR feature points with IMU data to enable robust navigation in fast-moving, noisy or cluttered environments [29], [30]. Geneva et al. [31] proposed a planar representation method based on the nearest point, constituting a three-dimensional planar SLAM system of liDAR and inertial elements. Qin et al. [32], Hong Kong University of Science and Technology, proposed a robust and universal method for inertial state estimation of deep coupled monocular vision.

This paper focuses on the current research work on the 3D laser SLAM algorithm and high-precision point cloud map construction. Firstly, the system scheme is designed. The initialization process of the multi-sensor fusion laser system designed in this paper is introduced, including the time synchronization of the multi-sensor. Then the basic processing method of 3D laser point cloud and feature extraction method based on roughness is introduced. Finally introduced pose estimation based on point cloud registration, using the integrated navigation system for calculating the position of the results as a condition of registration at the beginning, reusing the LM algorithm point cloud registration, and using map matching and loopback detection position optimization, finally get a picture can be used for driverless cars road campus auxiliary positioning test point cloud maps.

II. MATERIALS AND METHODS

In this paper, multi-sensor fusion is adopted to construct the SLAM system, and the pose calculation results of the integrated navigation system are introduced into the interframe registration of a three-dimensional laser point cloud. To a certain extent, the cumulative errors caused by using lidar alone can be reduced, and the accuracy of the final output pose information and the constructed point cloud map can be ensured. Figure 1 is the system block diagram designed in this paper. With 3D liDAR and an integrated navigation system as input, it is divided into four parts: SLAM system initialization, point cloud processing and feature extraction, pose estimation and map construction and pose optimization. Finally, the point cloud map is output. The components of each part are as follows:

System initialization: Because this article uses a combination of 3D liDAR and navigation systems belonging to two different data types of sensors, and data update frequency difference is bigger, with time by using the method of data synchronization software implementation for the two kinds of data, in addition to facilitate the subsequent data processing, will transition to the two sensors under the same coordinate system, is needed between the sensor and calibration; The integrated navigation system calculates the pose of the current carrier and takes the position of the first frame as the world coordinate system.

Point cloud processing and characteristics: 3D laser point cloud is corrected based on the external parameter calibration and pose solution results of system initialization; Due to the large amount of 3D laser point cloud data and many invalid points affected by noise, a voxel raster filtering method is used to filter the point cloud. Finally, the point cloud is segmented and the plane and edge features are extracted by the roughness of the point cloud.

Pose estimation: After extracting the characteristics of the point cloud, constructing the frame, point cloud feature constraint relations between constraint equation is obtained, and on this basis, the use of position information of integrated navigation system as the relations between the two at the beginning of the frame of point cloud registration, using LM nonlinear least squares method to solve the point cloud transformation matrix between frames, realize the point cloud registration between frames, and estimate the current position, And unify to the world coordinate system to get odometer information.

Map and Pose optimization: Sitting posture estimation, on the basis of using the map matching correction posture information in time, however, map matching optimization is based on the current point cloud and the relationship between the point cloud maps, unavoidably will still be affected by the accumulated error, so you also to join, a new method for the optimization of the closed-loop detection using ICP registration method to pose global optimization, and on this basis, the construction of road environment map of the point cloud.

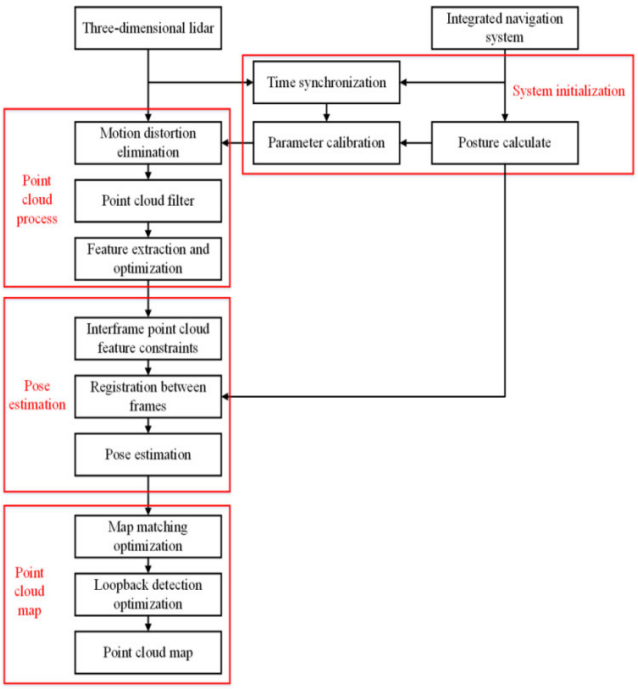


FIGURE 1. System chart.

A. SYSTEM INITIALIZATION

1) SENSOR TIME SYNCHRONIZATION

In SLAM, when faced with some extreme environments or too fast motion and too much jitter, the system using only liDAR may lose frames, leading to the wrong estimation of the pose at the next moment. Generally speaking, all pose estimations of laser SLAM are based on the world coordinate

system established for the first frame data, which may cause a large cumulative error over time. In this paper, an integrated navigation system is introduced for data fusion of multiple sensors to solve this problem. However, before the fusion of 3D liDAR and integrated navigation systems, the relationship between sensors and relevant data acquisition needs to be initialized.

The 3D liDAR [33], [34] sensor and the integrated navigation system used in this paper are two completely different sensors with different properties and independent data, and the scanning frequency of 3D liDAR is set at 10Hz, while the data update frequency of the integrated navigation system is set at 100Hz. There is no strict hardware time synchronization between them. Due to the influence of trigger delay, transmission delay and asynchronous clock sampling, time offset often exists at different measurement moments, as shown in Figure 2. Therefore, soft time synchronization processing is carried out for the two sensors.

According to ROS robot operating system subscription and distributed process mechanism, as a result of the integrated navigation system data update frequency is higher than 3D liDAR, the first to subscribe to news topic of 3D liDAR and get the timestamp, then subscribe to the news of the integrated navigation system is the subject to get away from the 3D liDAR data frame with the shortest time stamp and get the time difference value Δt_k, Then the linear acceleration and angular acceleration obtained by the integrated navigation system can be integrated to obtain the displacement and Angle offset of the carrier. Finally, the displacement and Angle offset are reflected on the point cloud data of the 3D liDAR to complete the soft time synchronization processing.

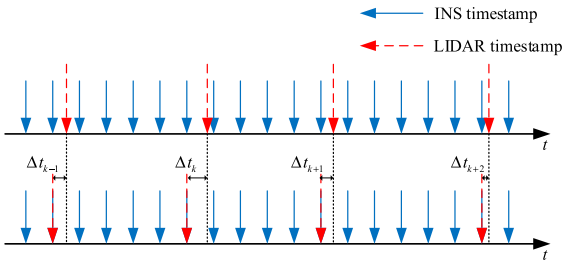


FIGURE 2. Schematic diagram of data frame offset of the sensor at different time.

2) MULTI-SENSOR EXTERNAL PARAMETER CALIBRATION

For multiple sensor fusion, the sensor coordinate system to the outside reference parameters [35] relationship between the data level fusion, unity, and the general for like liDAR and integrated navigation system of different data types of sensors, using the hand-eye calibration [36], [37] method can solve the two sensors' coordinate system between the parameters, The integrated navigation system and 3D liDAR used in this paper are fixed to the same mounting frame and can be seen as a rigid body.

External rotation matrix of calibration [38], [39], the first step to liDAR carrier to coordinate rotation matrix, calculate the rotation matrix can be used to minimize differences in function. However, because of the existence of the constraint in the three-dimensional space of a Jacobian matrix, the rotation matrix computation is more complex, so this article uses iterative ideas to solve this problem. In each iteration, the calibration results of the rotation matrix at the last time are used to estimate the zero deviation of the gyroscope, and then the rotation matrix is corrected by the zero deviation. Finally, a linear overdetermined equation is constructed to calculate the new rotation matrix.

3) IMU AND GPS FUSION LOCALIZATION

Navigation parameters of inertial navigation system (INS) in integrated navigation system mainly come from the solution of IMU, which can also be simply understood as INS inertial navigation system composed of IMU and related algorithms. IMU is a MEMS sensor commonly used for carrier pose estimation, consisting of an accelerometer, gyroscope and magnetometer. The current acceleration, angular velocity and magnetic field intensity of the carrier are measured.

IMU measurements contain two systematic errors and random errors. The systematic error includes scale factor deviation, zero deviation and axis deviation of accelerometer and gyroscope, while random error includes quantization noise, zero deviation instability and random walk noise. It is usually necessary to calibrate the systematic error and random error of IMU to obtain the true value of IMU. Systematic errors can be calibrated using the six-sided method and random errors can be analyzed using the Allan variance method.

GPS positioning inevitably has some positioning errors, the error sources are generally GPS own error, GPS signal transmission error and client receiver error. Differential GPS is a positioning method that eliminates positioning error sources of GPS reference receiver (reference station) and client (mobile station) by using real-time or post-processing technology. Differential GPS can be divided into position differential, pseudo-range differential and carrier phase differential according to different signal modes. Their working principle is the same, the difference lies in the content of the correction signal is not the same, and its positioning accuracy is also different. Carrier phase differential technology, also known as Real Time Kinematic technology (RTK), is a positioning technology that processes the carrier phase between the reference station and mobile station in Time. The mobile station receives carrier phase from the GPS satellite and reference station respectively, and the phase differential observation value is formed for timely processing, with accuracy up to centimeter level. In this paper, RTK location information is used as a real reference to compare the location results.

This paper mainly uses the Kalman filtering algorithm to fuse the IMU and GPS data of integrated navigation through four steps: data preprocessing, attitude calculation, position calculation, and data fusion, providing prior data for subsequent laser odometry, and using RTK as the standard value.

B. POINT CLOUD PROCESS

1) POINT CLOUD FILTER

When collecting 3D laser point cloud data, due to the influence of electromagnetic wave diffraction characteristics, surface properties of the measured object, human factors, environmental factors and other factors, it is inevitable to cause some outliers in point cloud data. To avoid the influence of these outliers on feature extraction, point cloud registration, visualization and other processes in subsequent point cloud processing, it is necessary to filter the point cloud data to eliminate the influence of these outliers.

Because of Velodyne HDL-32E [40] amount of a single frame of point cloud data collected is larger, the point cloud was ray traits, dense areas near the liDAR point cloud, far area is thin, it is not using the algorithm of a point cloud of follow-up to deal with, so this article use of voxel filtering for 3D laser point cloud filter, not only can remove outliers, also can undertake under the point cloud sampling at the same time keep the shape feature point cloud.

To better retain the characteristic shape of the point cloud, the voxel size used in this paper is 0.2.

2) FEATURE EXTRACTION AND OPTIMIZATION

To efficiently represent environmental information, a small amount of data is often needed to describe the environment, which requires that these data can be repeatedly observed from multiple angles and can clearly distinguish the surrounding points, namely feature points. In the 3D laser SLAM algorithm, the feature points need to meet the conditions of a sufficient number, static feature points and a simple and efficient feature extraction method. Compared with the image, the 3D laser point cloud can directly obtain the depth information of the surrounding environment, but due to the lack of texture information, the characteristics of the point cloud are monotonous. Although point cloud features can describe some shapes, such as points, lines and planes, it is very challenging to realize repeatable observation of extracted point cloud features due to the influence of point dispersion and data loss.

To make point cloud feature extraction efficient, accurate and reliable, this paper introduces a feature extraction method by calculating the barycenter distance between the point to be measured and the neighborhood point set, which can also be seen as calculating the roughness of the point to be measured and the neighborhood point set, similar to curvature. The points with large or small fluctuations in the point cloud are found as feature points according to the roughness. Due to a large amount of point cloud data, the horizontally adjacent point sets are from the same laser channel relative to the vertical point sets, so it is of greater reference significance to search for their features. Therefore, point clouds in the same channel need to be classified, and then point cloud features are extracted.

Figure 3 shows the feature extraction diagram between the points to be measured and the neighborhood points. Red points are the points to be measured, and green points are

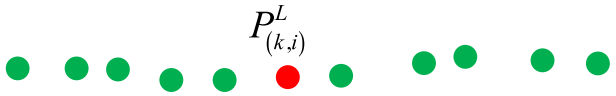


FIGURE 3. Schematic diagram of feature extraction.

the neighborhood points with the same ID of the points to be measured. Point cloud roughness is calculated according to equation (1).

$$c = \frac{1}{|S| \left\| P_{(k,i)} \right\|} \left\| \sum_{j \in S, j \neq i} \left(P_{(k,i)} - P_{(k,j)} \right) \right\| \tag{1}$$

where, $P_{(k,i)}$ and $P_{(k,j)}$ respectively represent the coordinate value of the point i and point j of the unified ID in the point cloud P_k. The point i is the point to be measured, S is the number of the point's neighborhood point set, and c is the roughness of the point and neighborhood point set. The larger c, the larger the roughness of the point, and the farther the distance from the neighboring point set, the point may be an edge point. On the contrary, the smaller c, the smaller the roughness of this point, and the closer the distance to the neighborhood point set, so this point may be a plane point. The edge feature point set extracted from P_k the point cloud is represented by ε_k, and the plane feature point set is represented by ξ_k.

3) POINT CLOUD SEGMENTATION

To improve the efficiency of plane feature extraction, a fast point cloud plane acquisition method ground estimation is used to improve the efficiency of plane point feature extraction. The point cloud of 3D liDAR is distributed in ray form, and the adjacent relation between channels is calculated to determine whether it is ground or not.

$$\theta = \arctan \frac{\sqrt{\left(x_{(k,i)} - x_{(k-1,i)}\right)^2 + \left(y_{(k,i)} - y_{(k-1,i)}\right)^2}}{\left|\left(z_{(k,i)} - z_{(k-1,i)}\right)\right|} \tag{2}$$

where, k and $k-1$ are the channel ID of the liDAR, $\left(x_{(k,i)}, y_{(k,i)}, z_{(k,i)}\right)$ and $\left(x_{(k-1,i)}, y_{(k-1,i)}, z_{(k-1,i)}\right)$ are the coordinate values of the two points of the adjacent channel, θ is the included angle between the two points and the liDAR coordinate system, if less than the threshold, the point is the ground.

In addition, there will be a large number of trees and vegetation on both sides of the campus road, so it is difficult to achieve repeatable observation of features extracted from leaves and grass leaves. To improve the efficiency of the algorithm and filter out invalid features, the point cloud after filtering ground was classified based on the fast image segmentation algorithm, and the categories less than 30 were ignored.

As shown in Figure 4, O is the origin of the liDAR coordinate system, A and B are two points in the point cloud respectively, OA and OB represent the relative distance between the two points and the origin of coordinates. Whether

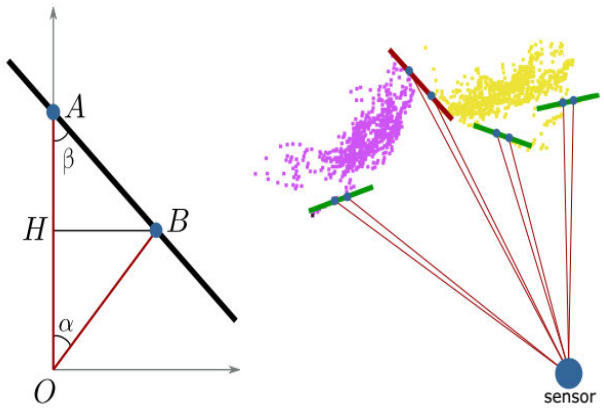


FIGURE 4. Point cloud feature object classification.

points A and B belong to the same object can be estimated by the angles of the triangle formed by points A, B and O.

$$\beta = arctan\frac{\|BH\|}{\|HA\|} = arctan(\frac{d_{OB}sin\alpha}{d_{OA} - d_{OB}cos\alpha}) \quad (3)$$

d_{OA} and d_{OB} are the distances of points A and B to the origin of the liDAR coordinate system respectively, and α represents the horizontal angular resolution of the liDAR. Combined with Figure 4, it can be seen that when β is greater than a certain threshold, two points are regarded as the same object, as shown in The green line in Figure 4 When β is less than a certain threshold, it indicates that the depth of two points varies greatly and they do not belong to the same object, as shown in The red line in Figure 4.

C. POSE ESTIMATION AND OPTIMIZATION

1) INTERFRAME POINT CLOUD FEATURE CONSTRAINTS

Due to the huge point cloud data of each frame, each point in the point cloud occupies different time stamps with a distant distribution. Before inter-frame feature constraint, the point cloud data of the two adjacent frames need to be unified to the same moment.

As shown in Figure 5, the blue line represents the point cloud P_k in the frame k of point cloud data obtained by liDAR, and t_k is the time stamp of the point cloud P_k. Similarly, the red line represents the point cloud P_{k+1} in the frame $k+1$ of point cloud data obtained by liDAR, and t_{k+1} is the time stamp of the point cloud P_{k+1}. The point cloud P_k is represented by $\bar{P}_k$ when it is re-projected to t_{k+1}, and the green line in Figure 5. In this way, the point cloud P_k of the frame k and the point cloud P_{k+1} of the frame $k+1$ can be unified at the same time for inter-frame feature constraints. In this paper, the constraints of point-to-line and point-to-plane distance between the features of point clouds in adjacent frames are used to form nonlinear optimization problems, and the pose relationship is solved.

Through the feature extraction method, the edge point set ε_{k+1} and flat point set ξ_{k+1} can be obtained in the point cloud

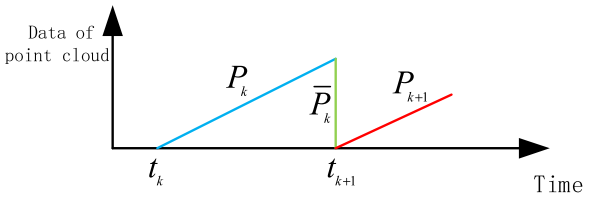


FIGURE 5. Point cloud scanning update process.

P_{k+1}, and feature lines and feature planes corresponding to ε_{k+1} and ξ_{k+1} can be found in the point cloud $\bar{P}_k$. As time goes on, more point cloud data frames P_{k+1} will be acquired gradually. In each iteration, ε_{k+1} and ξ_{k+1} are projected to the timestamp of the current data frame according to the currently estimated conversion relationship, represented by $\tilde{\varepsilon}_{k+1}$ and $\tilde{\xi}_{k+1}$ respectively. For each point at $\tilde{\varepsilon}_{k+1}$ and $\tilde{\xi}_{k+1}$, the corresponding nearest point can be found in the point cloud $\bar{P}_k$. Different inter-frame feature constraint relations can be constructed by the feature edge point set and feature plane point set of each frame data.

a: CONSTRAINTS OF POINT AND LINE

As shown in Figure 6 (a), let i be a point in the set of feature edge points $\bar{\varepsilon}_{k+1}$, $i \in \tilde{\varepsilon}_{k+1}$. The edge line $j \in \bar{P}_k$, m is the sub-adjacent point from the point i on the front and back channels of the channel where point j, $m \in \bar{P}_k$. Points j and m form an edge line in $\bar{P}_k$ and are located in different channels. Then the distance between the edge point i and the edge line constitutes the point line constraint relation.

$$d_\varepsilon = \frac{\left|(P_{(k+1,i)} - P_{(k,j)}) \times (P_{(k+1,i)} - P_{(k,m)})\right|}{\left|P_{(k,j)} - P_{(k,m)}\right|} \quad (4)$$

where, $P_{(k,j)}$ and $P_{(k,m)}$ are the coordinate values of point j and point m in point cloud $\bar{P}_k$ respectively, and $P_{(k+1,i)}$ is the coordinate values of point i in the set $\bar{\varepsilon}_{k+1}$ after the projection of feature edge points in the point cloud P_{k+1}.

b: CONSTRAINTS OF POINT AND SURFACE

In Figure 6 (b), i is set as a point in the set of feature plane points $\tilde{\xi}_{k+1}$, $i \in \tilde{\xi}_{k+1}$.Similar to the point-line constraint, the nearest proximity of the KD tree is used to search for the nearest point j of the point cloud $\bar{P}_k$ midpoint i, $j \in \bar{P}_k$. Then find point m and point n as the next adjacent points of point i, $m \in \bar{P}_k$, $n \in \bar{P}_k$, point m and point j are on the same channel, and the point n is on the adjacent channel of point j.

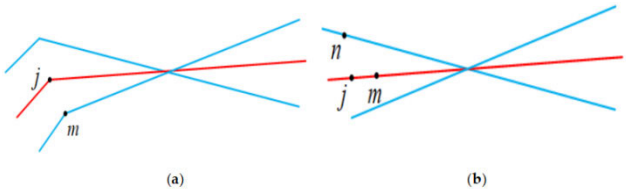


FIGURE 6. Schematic diagram of feature constraint construction.

In particular, points j, m, and n are not collinear. In this way, these three points can form a plane $\bar{P}_k$, and the distance between the plane point i and the plane constitutes the point-plane relation.

$$d_\xi = \frac{\left|(P_{(k+1,i)} - P_{(k,j)})((P_{(k,j)} - P_{(k,m)}) \times (P_{(k,j)} - P_{(k,n)}))\right|}{\left|(P_{(k,j)} - P_{(k,m)}) \times (P_{(k,j)} - P_{(k,n)})\right|} \tag{5}$$

where, $P_{(k,j)}$, $P_{(k,m)}$ and $P_{(k,n)}$ are the coordinate values of point j, point m and point n in the point cloud $\bar{P}_k$ respectively, and $P_{(k+1,i)}$ is the coordinate values of point i in the set $\tilde{\xi}_{k+1}$ after the projection of feature plane points in the point cloud P_{k+1}.

2) POSE ESTIMATION

In this section, we need to use the nonlinear least square LM method to solve the inter-frame pose conversion, carry out the point cloud inter-frame accurate registration, and build the pose odometer. The Levenberg-Marquardt (LM) algorithm [41], [42] is an optimization method commonly used in nonlinear least squares problems. It has the characteristics of a gradient descent algorithm and a Gauss-Newton algorithm. It can solve the case of the Hessian Matrix with non-rank and non-positive definite.

The 3D liDAR and integrated navigation system used in this paper can be regarded as the same rigid body as the carrier, and the movement state of the carrier can be estimated by a 3D laser point cloud and inertial navigation system. In each point cloud data acquisition cycle, the movement of the carrier can be regarded as being carried out at constant angular velocity and linear velocity. Therefore, linear insertion pose transformation can be used for points in the same period obtained at different times. Suppose that the current timestamp of the point cloud of the frame $k+1$ is t, t_{k+1} is the starting timestamp of the point cloud of the frame $k+1$, and T_{k+1} is the 6-DoF pose transformation of 3D liDAR in $[t_{k+1}, t]$, $T_{k+1} = \left[t_x, t_y, t_z, \theta_x, \theta_y, \theta_z\right]$. Where, $\left[t_x, t_y, t_z\right]$ and $\left[\theta_x, \theta_y, \theta_z\right]$ represent the translation and rotation Angle corresponding to the X, Y and Z axes of the 3D liDAR coordinate system respectively. If t_i is set as the timestamp of a point in the point cloud P_{k+1}, the corresponding pose transformation matrix $T_{(k+1,i)}$ at $[t_{k+1}, t_i]$ can be obtained by linear interpolation of T_{k+1}.

$$T_{(k+1,i)} = \frac{t_i - t_{k+1}}{t - t_{k+1}} T_{k+1} \tag{6}$$

Therefore, according to the feature point-line constraint and feature point-plane constraint constructed in section C(1), the constraint relation between point cloud coordinates and pose transformation can be obtained.

$$f_\varepsilon(P_{(k+1,i)}, T_{k+1}) = d_\varepsilon, \quad i \in \varepsilon_{k+1} \tag{7}$$

$$f_\xi(P_{(k+1,i)}, T_{k+1}) = d_\xi, \quad i \in \xi_{k+1} \tag{8}$$

Because different feature points have different advantages, plane feature points are extracted based on ground segmentation in the early stage, which can intuitively reflect the height change of the Z axis, roll Angle change around the X axis and pitch Angle change around the Y axis. Therefore, the whole pose solution is divided into two steps. First, $\left[t_z, \theta_x, \theta_y\right]$ in the pose is calculated by using plane features, which are taken as the initial iterative value of edge feature points. Finally, $\left[t_x, t_y, \theta_z\right]$ is calculated as the unknown quantity of the rest of the pose.

Therefore, the LM algorithm can obtain the solution formula of pose transformation.

$$T_{k+1} \leftarrow T_{k+1} - (J^T J + \lambda I)^{-1} J^T f \tag{9}$$

where, f is the constraint equation, J represents f for the Jacobian matrix of the variable T_{k+1} to be solved. According to the chain method.

$$J_{k+1} = \frac{\partial f}{\partial T_{k+1}} = \frac{\partial f}{\partial P_{(k+1,i)}} \frac{\partial P_{(k+1,i)}}{\partial T_{k+1}} \tag{10}$$

D. POINT CLOUD MAP CONSTRUCTION

The purpose of this paper is to build a point cloud map for driverless vehicles to assist in locating on campus roads. Based on the optimal pose, the point cloud of the corresponding pose is converted to the world coordinate system (map coordinate system), that is, the point cloud map can be obtained. On the basis of point cloud registration, multi-sensor fusion laser odometer, map matching and loopback detection are used to optimize the pose.

1) MAP MATCHING OPTIMIZATION

Based on the least square method introduced in Section C(2), the rotation and displacement pose transformation relations of the point cloud of adjacent frames are solved. With the continuous increase of point cloud data frames, the position and pose transformation relations of adjacent points are continuously superimposed to obtain the liDAR moving track, namely the laser odometer. Since a single sensor cannot effectively eliminate the accumulated errors caused by point registration of each adjacent frame, the data information of the integrated navigation system should be introduced as the prior information of point cloud registration to reduce the impact of the accumulated errors of a single sensor on pose estimation. After the time synchronization processing of 3D liDAR and integrated navigation system, the pose solution method can be used to obtain the initial conversion relation of point clouds of adjacent frames for the initial registration of point clouds. Then, on the basis of initial registration, the point cloud registration method in Section C(2) can be used for precise registration. Finally, the laser odometer after data fusion is obtained after two-point cloud registration.

To make pose estimation more accurate, map matching optimization is carried out based on the laser odometer with multi-sensor fusion. Map matching optimization is to optimize the part of the constructed map and the currently estimated pose. According to Section C(1), the adjacent frame point clouds within time $[t_k, t_{k+1}]$ are first re-projected, and the pose transformation relation of the adjacent frame

point clouds within time intervals $[t_k, t_{k+1}]$ is estimated as T_{k+1}^k through the above method. Suppose M_k is the map at t_k moment, and the pose in the world coordinate system is T_k^W, then T_{k+1}^k and T_k^W can get the pose T_{k+1}^W of P_{k+1} in the world coordinate system, and then transform $\bar{P}_{k+1}$ into the world coordinate system, to get the map $\tilde{M}_{k+1}$ at t_{k+1}. At this point, according to the feature constraint thought in Section C(1), M_k and $\tilde{M}_{k+1}$ constraint relations were constructed by using the feature points of the current point cloud PK $+1\sim$. The point cloud registration method in Section C(2) was used to complete the registration of the point cloud $\bar{P}_{k+1}$ and map M_k, and the optimized pose T_{k+1}^W and map M_{k+1} were obtained.

2) LOOPBACK DETECTION OPTIMIZATION

The position that the robot passed at a certain moment before it passed (called revisiting), and the effective recognition of the revisiting position (namely loopback detection) are related to posing optimization. Although the first two sections have used multi-sensor fusion and map matching to optimize the pose, both of them are based on the relationship between adjacent data frames to optimize the pose, which still has a certain cumulative error, so global pose optimization is needed, namely loopback detection and poses optimization.

As time goes by, the number of point clouds in the map will increase more and more, and the global optimization using the whole established map will greatly increase the computational burden of the algorithm. In this paper, with 0.5m as a unit, the calculated pose is saved as a key data frame, and the features and poses of the key data frame are saved in the KD tree. Then, the nearest neighbor search method of the KD tree is used to calculate the pose of the key data frame within the radius of 20m of the current pose and obtain its features. Finally, the ICP point cloud matching method is used to match the point cloud of the current point cloud and the keyframe, and the relative pose is calculated. Gtsam [42] is used to optimize the pose of the stored keyframe to eliminate accumulated errors and complete the global pose optimization.

In general, it is considered that the odometer pose data output at a high frequency in a short time is relatively accurate, and the global pose will not be affected by reducing the processing frequency of map matching and loop optimization.

III. RESULTS

A. POINT CLOUD SEGMENTATION AND FEATURE EXTRACTION EXPERIMENTS

In this section, based on the feature extraction method introduced above, a feature extraction experiment is designed based on point cloud segmentation using the characteristics of point cloud roughness, and the feature extraction effect is compared with whether point cloud segmentation is carried out. Figure 7 shows the point cloud after point cloud segmentation, where the yellow point cloud refers to ground point cloud segmentation and the black point cloud refers to point cloud segmentation.

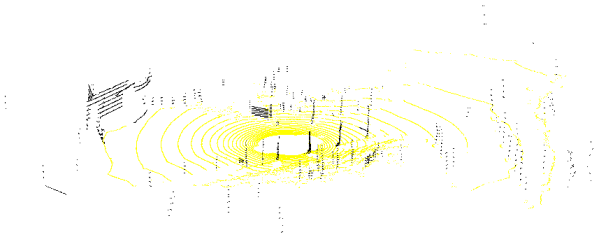


FIGURE 7. Point cloud segmentation results.

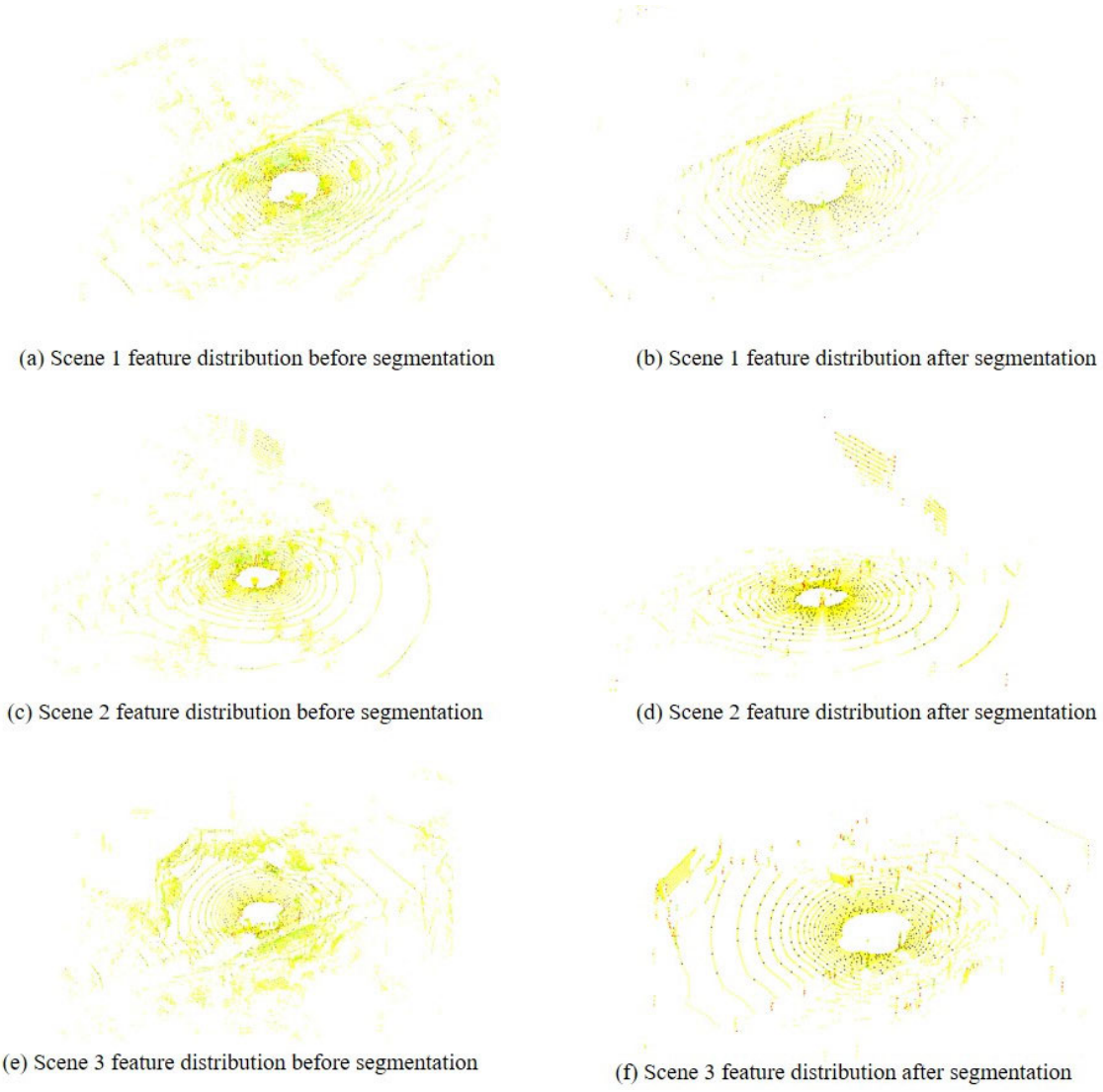
A comparative experiment of feature distribution extracted before and after point cloud segmentation was designed for three different scenes. Experimental results as shown in Figure 8, for feature extraction of ordinary road scene 1 distribution Figure 8 (a) and (b), from close to the road building scene 2 distribution of feature extraction for Figure 8 (c) and (d), a wide range of scenario3Distribution of feature extraction for Figure 8 (e) and (f), Figure 8 (a), (c) and (e) not on point cloud segmentation to extract the characteristics of distribution, Figure 8 (b), (d) and (f) is the feature distribution extracted after point cloud segmentation, wherein yellow point cloud is the planar feature, the black point cloud is the largest planar feature, the red point cloud is the edge feature, and the green point cloud is the smallest edge feature. As can be seen from Figure 8, red and green edge features are mostly distributed on tree trunks, grass and the edge of buildings, while yellow and black plane features are distributed on the ground and surface of buildings. It shows that feature extraction based on roughness is effective and can be used as an effective feature point of the scene. Among them, edge features and plane features extracted before point cloud segmentation are easily affected by small objects, grassland and tree canopies, and their distribution is chaotic without obvious separation. The features extracted based on point cloud segmentation are distributed evenly and rationally, and the number of features is greatly reduced, excluding the influence of grass and tree canopy.

The plane features come from the point cloud after ground segmentation, and the edge features come from the point cloud after object segmentation.

Table 1 shows the comparison between the number of features and the corresponding computing time of the above three scenarios. It can be seen from the table that the number of features extracted based on point cloud segmentation is significantly reduced and the computing time is significantly reduced, which is more conducive to the subsequent point cloud feature registration.

B. POINT CLOUD REGISTRATION EXPERIMENTS

As shown in Figure 9, point cloud P_k at t_k moment (yellow) and point cloud P_{k+1} at t_{k+1} moment (red) respectively. By using the point cloud registration method, the transformation matrix T_{k+1}^k between the point cloud P_{k+1} and point cloud P_k can be obtained, and the value of T_{k+1}^k is shown in Formula 11, as shown at the bottom of the next page.



(a) Scene 1 feature distribution before segmentation

(b) Scene 1 feature distribution after segmentation

(c) Scene 2 feature distribution before segmentation

(d) Scene 2 feature distribution after segmentation

(e) Scene 3 feature distribution before segmentation

(f) Scene 3 feature distribution after segmentation

FIGURE 8. Comparison of feature extraction distribution before and after point cloud segmentation.

P_{k+1} is unified into the P_k coordinate system through the transformation matrix T_{k+1}^k, and the point cloud registration result as shown in Figure 10 is obtained.

From Figure 10, point cloud registration results and transformation between the frame matrix T_{k+1}^k can be seen, due to the time interval of the two point clouds being small, the rotation matrix transformation matrix is close to the unit matrix, the indication of the relative momentum transfer of two point cloud is very small, and translation vector responses in three directions of translational value, close to the vehicle moving displacement.

C. MAP CONSTRUCTION AND POSE OPTIMIZATION

The point cloud map constructed in this paper represents the height distribution of the point cloud map based on the color distribution. The cool color indicates the lower the height, and the warm color indicates the higher the height. Figure 11 is the point cloud map directly constructed by using the position and pose solved by the integrated navigation system, that is, the initial registration of the point cloud. Each frame of the point cloud is converted to the map coordinate system according to the position and pose solved by the corresponding integrated navigation system, and the map coordinate system

$$T_{k+1}^k = \begin{bmatrix} 0.999996 & 0.0013327 & -0.0023995 & 0.15093 \\ -0.0013264 & 0.999996 & 0.0026444 & -0.002399 \\ 0.0024030 & -0.0026412 & 0.999994 & 0.064744 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

TABLE 1. The comparison results of feature extraction.

Scene	Maximum edge features/piece		Ordinary edge features/piece		Minimum plane features/piece		Ordinary plane features/piece		Calculation time/ms	
	Original point cloud	Point cloud segmentation	Original point cloud	Point cloud segmentation	Original point cloud	Point cloud segmentation	Original point cloud	Point cloud segmentation	Original point cloud	Point cloud segmentation
1	351	218	2130	301	545	393	18756	5175	44.67	20.85
2	349	186	1697	259	564	423	17781	5424	45.10	19.78
3	284	197	1830	399	630	375	19992	5455	47.06	21.48

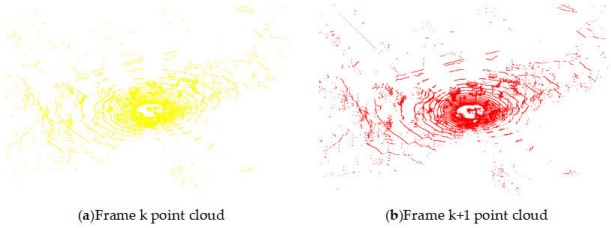


FIGURE 9. Point clouds of two adjacent frames before registration.

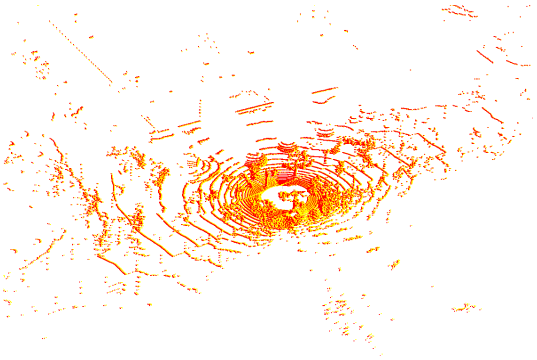


FIGURE 10. Point cloud after registration.

and the world coordinate system are defined as the same coordinate system. The constructed point cloud map does not carry out inter-frame registration of the point cloud. Although it roughly conforms to the actual scene around campus roads, the point cloud distribution is chaotic and cannot be used directly.

Figure 12 is the point cloud map constructed by using only a 3D laser point cloud for interframe registration of the point cloud. Compared with Figure 11, the point cloud is much cleaner. However, due to accumulated error, according to the height color information of the point cloud map, the

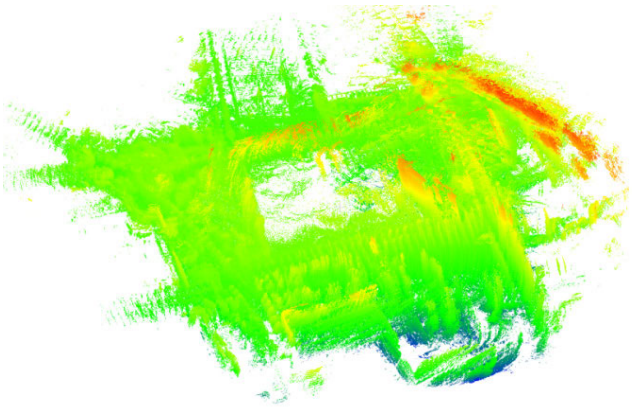


FIGURE 11. Point cloud map based on the pose of the integrated navigation system.

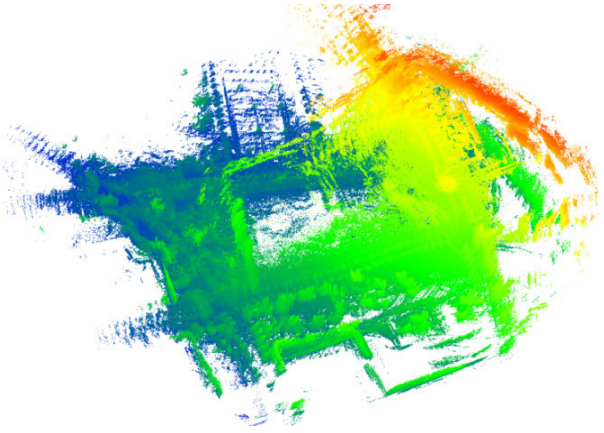


FIGURE 12. Point cloud map is constructed using only 3D laser point cloud interframe registration.

whole point cloud map is not on the same plane, so there is no closure and obvious faults. Therefore, before inter-frame

registration, the position and pose calculated by the integrated navigation system can be used as the initial registration information to reduce a certain cumulative error.

Figure 13 is a point cloud map constructed by map matching optimization based on the prior information of accurate interframe registration of the point cloud solved by the integrated navigation system. Compared with the point cloud map constructed by using only a 3D laser point cloud, the distribution of the point cloud is cleaner and the surrounding environment of campus road is more specific. However, due to the influence of cumulative error, trees on both sides of the road appear obvious dislocation at the loopback position marked in the red box in Figure 13, as shown in Figure 14 (a).

After loopback detection optimization, point clouds at loopback locations are optimized, as shown in Figure 14 (b). Finally, a globally consistent point cloud map is constructed, as shown in Figure 15.

The pose estimation of pure 3D laser point cloud was compared with the optimized pose, and the RTK positioning result was taken as the standard quantity. The experimental results are shown in Figure 16 and Figure 17.

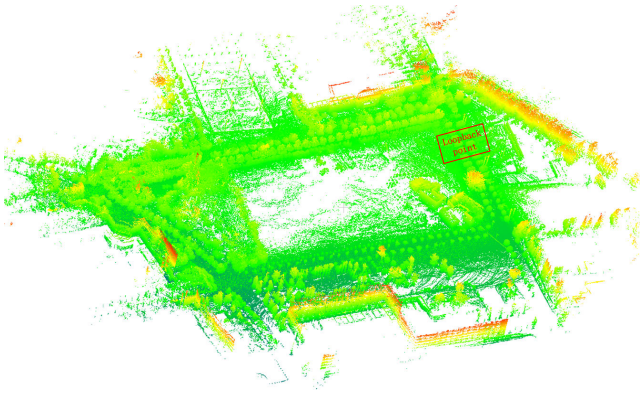


FIGURE 13. Point cloud map constructed by data fusion of 3D liDAR and integrated navigation system.

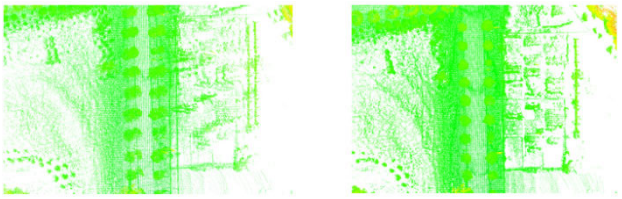


FIGURE 14. Optimization comparison before and after loopback points in the point cloud map.

Figure 16 and Figure 17 respectively show the position comparison of different pose estimation results in the X-Y plane and Z axis. The green curve is the pose trajectory estimated using only a 3D laser point cloud; the red curve is the optimized pose trajectory after inter-frame registration based on the initial pose registration of the integrated navigation system; and the black curve is the RTK positioning result. It can be seen that the pose estimation based only

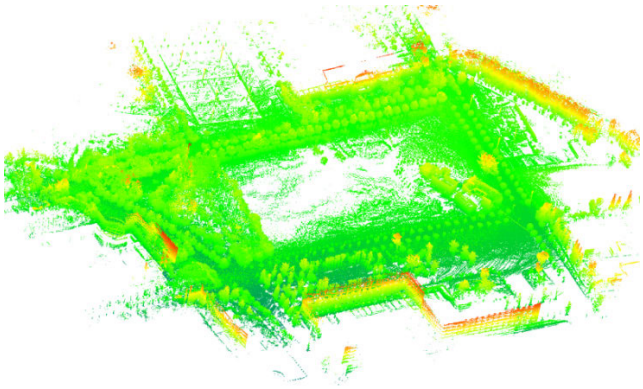


FIGURE 15. Optimized global consistency point cloud map.

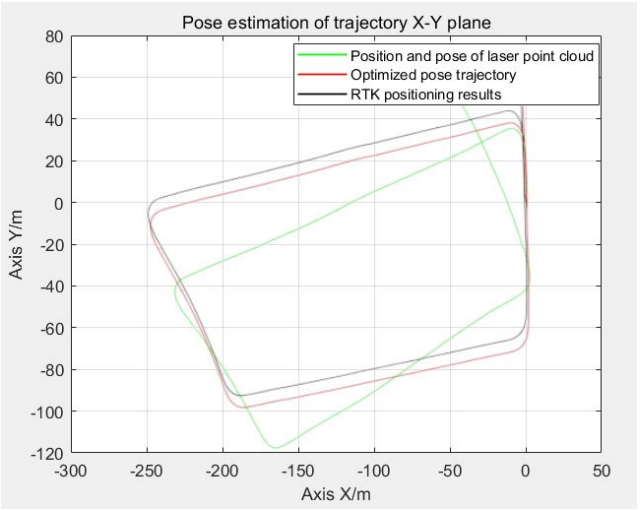


FIGURE 16. Pose estimation of the trajectory X-Y plane.

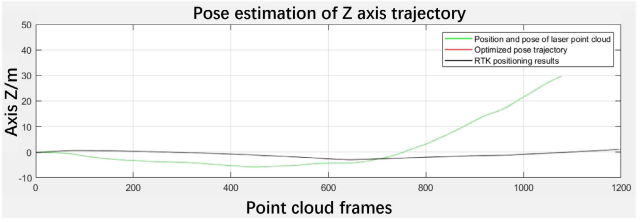


FIGURE 17. Pose estimation of Z-axis trajectory.

on a 3D laser point cloud has high accuracy in the initial period, but with the continuous increase of data frames, the cumulative error increases, leading to the divergence of the pose. After introducing the pose information of the integrated navigation system for initial registration and optimization, the pose estimation result is significantly improved and roughly consistent with the RTK result.

Due to the flat terrain and small difference in roll Angle and pitch Angle during the collection of a campus environment data, a course angle comparison of pose estimation results was conducted, as shown in Figure 18. It can be seen from the figure that the heading Angle estimated only by the 3D laser

TABLE 2. Comparison of measurement values of road width of sections.

Numbers	Data source	Number of measurement					Mean value	Error
		1	2	3	4	5		
Section 1	Point cloud map	7.104	7.117	7.064	7.067	7.045	7.079	0.045
	Actual measurement	7.058	7.015	7.036	7.028	7.034	7.034	
Section 2	Point cloud map	7.015	7.066	6.988	7.045	7.012	7.025	0.021
	Actual measurement	7.019	7.011	7.005	6.979	7.008	7.004	
Section 3	Point cloud map	6.157	6.146	6.079	6.106	6.141	6.128	0.061
	Actual measurement	6.065	6.071	6.066	6.059	6.075	6.067	

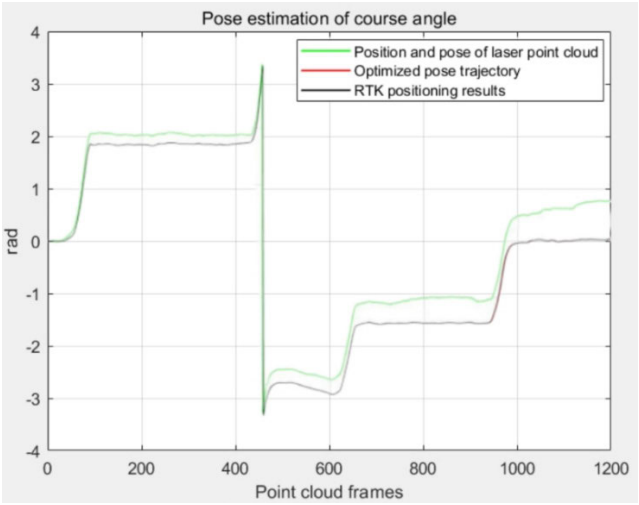


FIGURE 18. Pose estimation of course angle.

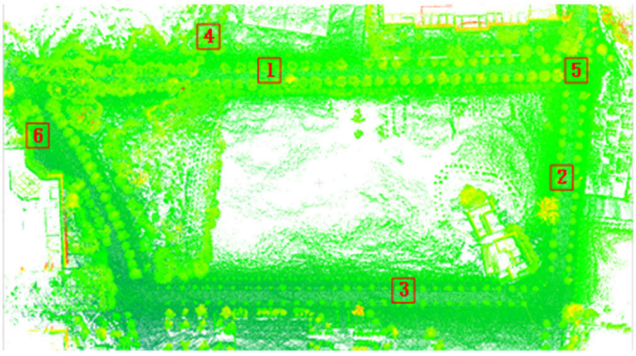


FIGURE 19. The position mark of a measurement position in a point cloud map.

point cloud gradually diverges over time. After optimization, the accuracy of the heading angle is improved and can be roughly consistent with the heading result measured by RTK.

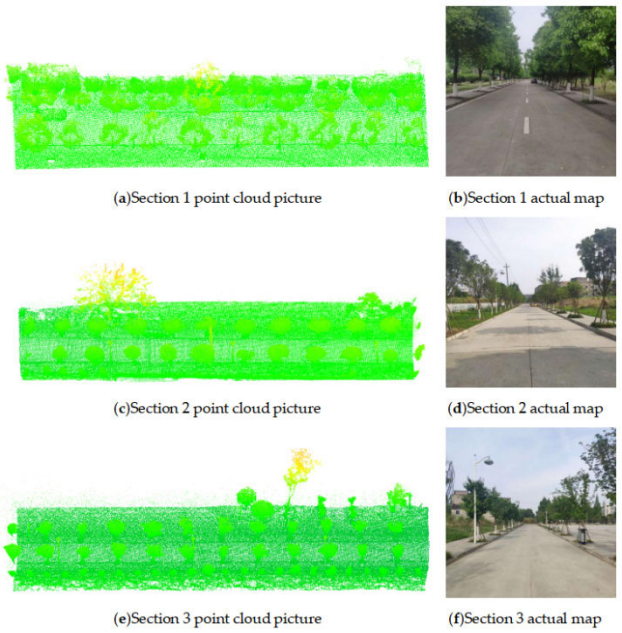


FIGURE 20. Point cloud images and actual images of three road sections were measured.

To verify the road accuracy of the point cloud map constructed in this paper, a relatively reliable accuracy value can be obtained by comparing the constructed point cloud map model with the real environment. Three sections of the road point cloud are taken from labels 1, 2 and 3 respectively, and their approximate positions are shown in Figure 19. The three sections of the road are shown in Figure 20 (a), (c) and (e), and the actual scene is shown in Figure 20 (b), (d) and (f). In the road point cloud model reconstructed from the point cloud map, the road edge and the road plane are divided. The measurement of road width is from the bottom of one side of the road to the bottom of the other side of the road, and the specific measurement data is shown in Table 2.

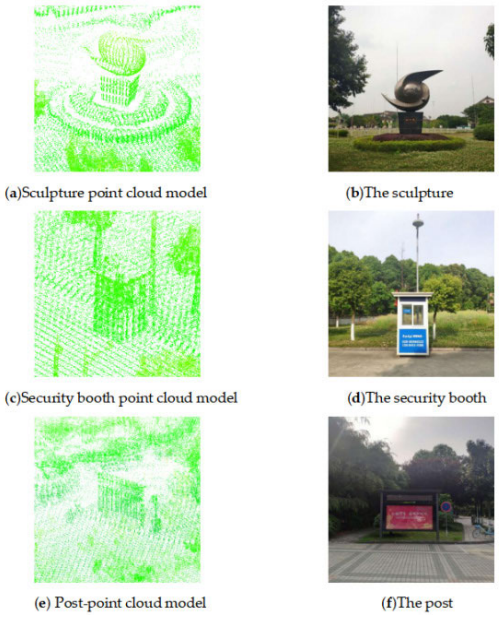


FIGURE 21. Point cloud model for measuring objects.

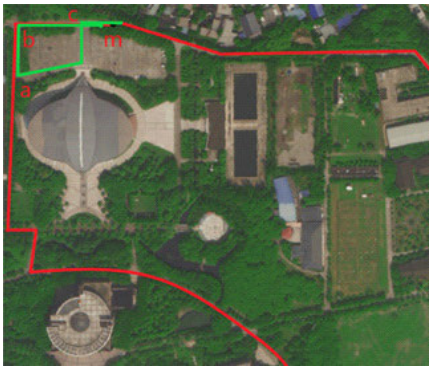


FIGURE 22. Trajectory map of the unmanned vehicle driving.

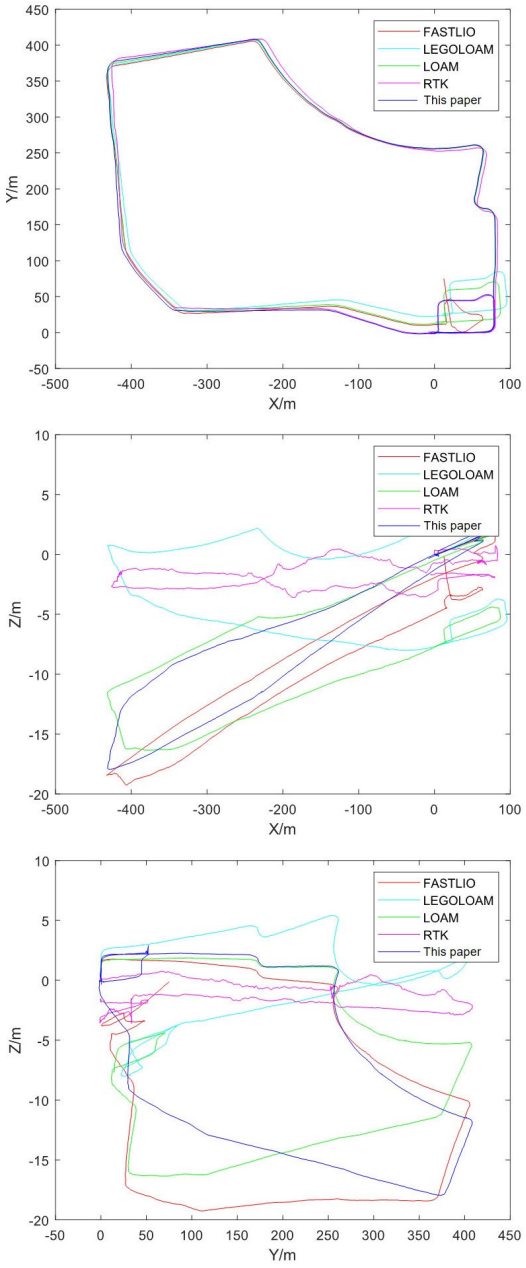


FIGURE 23. Trajectory comparison of multiple SLAM algorithms.

As can be seen from Table 2, the maximum error of the three sections is 0.061m, and the minimum error is 0.021m. In addition, the measurement data of the point cloud map is smooth and has no difference, which meets the accuracy requirements of the test of unmanned vehicles on campus roads.

To further verify the accuracy of the point cloud map, the accuracy measurement is carried out on three objects in the point cloud map environment, the specific locations are labeled 4, 5 and 6 in Figure 19. Label 4 is a sculpture model, as shown in Figure 21 (a). The width of the trapezoidal base of the sculpture is measured from bottom to top and the average value is taken. Label 5 is the security pavilion, as shown in Figure 21 (c), which mainly measures the width of the security pavilion. Label 6 is the newspaper column, as shown in Figure 21 (e), which mainly measures the width between two walls. (b), (d) and (f) in Figure 21 are the real images of corresponding objects respectively. The three-point cloud models were measured 5 times respectively, and the specific measured values are shown in Table 3.

It can be seen from the point cloud models of three different objects in Figure 21 that the point cloud map construction method in this paper can reconstruct the point cloud models of objects in the environment and intuitively distinguish the shapes and contours of objects.

According to the measurement data in Table 3, the measured objects are all standard sizes, and the error value is smaller than that of the section error. Because of the 3D laser point cloud radiated, with the increase of distance, measurement error will increase, so the distance of 8 m model 4 error relative to the distance of the putting error of 4 meters to a

TABLE 3. Comparison of measurement values of road width of sections.

Numbers	Data source	Number of measurement					Mean value	Error
		1	2	3	4	5		
Model 4	Point cloud map	1.756	1.806	1.851	1.967	2.034	1.882	0.035
	Actual measurement	1.568	1.702	1.846	1.998	2.122	1.847	
Model 5	Point cloud map	1.225	1.224	1.227	1.248	1.237	1.232	0.031
	Actual measurement	1.202	1.203	1.201	1.199	1.198	1.201	
Model 6	Point cloud map	3.155	3.169	3.162	3.141	3.164	3.158	0.014
	Actual measurement	3.139	3.148	3.150	3.138	3.146	3.144	

few bigger, but for driverless cars in the actual test, the error of 0.1 meters of the unmanned vehicle road test is within the acceptable range.

D. SLAM ALGORITHM COMPARISON EXPERIMENTS

In order to verify the performance of the overall mapping scheme in this paper, this section compares the effect of common SLAM algorithms in the campus dataset. The campus dataset is as follows: collect a trajectory map with a length of about 1940m as shown in Figure 22, where the black origin is the starting point, drive along the red route until passing the starting point for the second time, and the green circle represents the ending point. Assuming the starting point is m, and m is also the first loop closure point. This paper mainly compares LOAM, LEGO-LOAM and FAST-LIO2 algorithms. The running results of various algorithms are as follows:

According to Table 4, during the operation of the algorithm used in this article, the offset error is the average calculated error of the error in each axis direction of the trajectory point when the unmanned vehicle runs to 100m, 200m, 300m…1800m. Using the dataset collected from the campus to run all algorithms, the error of the proposed algorithm in the X-axis and Y-axis direction is much smaller than other algorithms. Compared with the elevation error, the proposed algorithm has a higher error than LEGO-LOAM, but the overall error is 0.453m, which is lower than the 8.54m error value of LEGO-LOAM. For the LOAM algorithm and FAST-LIO2 algorithm, due to the lack of a loop closure detection module, the accumulated errors estimated by inertial sensors and laser odometry can not be eliminated. This will result in the final estimated trajectory errors reaching 4.106m and 4.382m. For the LEGO-LOAM algorithm, due to the use of a loop closure detection module based on Euclidean distance, it is ultimately unable to loop back correctly, resulting in an estimated pose error of 8.54m.

TABLE 4. Comparison results of SLAM algorithm errors (m).

	X-axis error	Y-axis error	Z-axis error	RMSE
LOAM	7.62106	8.711132	7.68999	4.10558
LeGO-LOAM	9.65647	13.195	3.400000	8.54875
FAST-LIO2	11.8745	10.3897	9.64614	4.38188
Proposed algorithm	3.86218	2.6432	7.86165	0.45258

As shown in Figure 23, the trajectory diagrams of all comparison algorithms and the algorithm in this paper are presented. The RTK trajectory is used as the true reference value, and the trajectory estimated by the algorithm in this paper is roughly consistent with the RTK trajectory. The trajectory estimated by other SLAM algorithms ultimately failed to loop back successfully, and effective means were not used to eliminate cumulative errors, resulting in errors in the final estimated trajectory. Therefore, the overall comparison shows that the algorithm proposed in this paper has higher mapping performance compared to the classic open-source laser SLAM algorithms.

IV. CONCLUSION

In this paper, a tightly-coupled laser SLAM integrating LiDAR and INS is developed to solve the problems of low location and mapping accuracy in large-scale environments caused by the lack of information and poor robustness of a single sensor for autonomous unmanned vehicles. Firstly,the preprocessing and feature extraction methods of the 3D laser point cloud are studied. To keep the shape characteristics of the point cloud and reduce the noise of the point cloud, the voxel grid method is used to filter the point cloud. On the basis of point cloud segmentation, the roughness of the point cloud is used as the benchmark to extract point

cloud features, and plane features and edge features are extracted, and the features are optimized. Secondly, the laser SLAM algorithm based on point cloud registration is studied. On the basis of feature extraction, the characteristic constraint equation of the inter-frame point cloud was constructed, and the pose solution result of the INS was added as the prior data of inter-frame point cloud registration. The levenberg-Marquardt nonlinear least square method was used to solve the constraint equation to obtain the pose relationship between inter-frames. The odometry was optimized by map matching and loop closure detection. Finally, compared the proposed SLAM algorithm with the classic open-source SLAM algorithm by the campus dataset to verify the performance of the overall mapping scheme. The experimental results show that the trajectory error estimated by the proposed SLAM algorithm is smaller than that estimated by classic open-source SLAM algorithms. It has certain advantages in estimating the motion trajectory error of unmanned vehicles. The generated point cloud map can be further developed into a high-precision map for campus scene navigation of unmanned vehicles.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[2] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, Dec. 1986.

[3] H. Durrant-Whyte, "Where am I? A tutorial on mobile vehicle localization," *Ind. Robot, Int. J.*, vol. 21, no. 2, pp. 11–16, Apr. 1994.

[4] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Auton. Robot Vehicles*, pp. 167–193, Feb. 1990.

[5] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI/IAAI*, Jul. 2002, pp. 593–598.

[6] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[7] S. Yang, X. Zhu, X. Nian, L. Feng, X. Qu, and T. Ma "A robust pose graph approach for city scale LiDAR mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1175–1182.

[8] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1352–1359.

[9] X. Ji, L. Zuo, C. Zhang, and Y. Liu, "LLOAM: LiDAR odometry and mapping with loop-closure detection based correction," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2019, pp. 2475–2480.

[10] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: 3D segment mapping using data-driven descriptors," in *Proc. 14th Robot. Sci. Syst.*, Jun. 2018, pp. 1–5.

[11] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.

[12] X. Ji, Z. Gong, R. Miao, W. Xue, and R. Ying, "Monocular semantic mapping based on 3D cuboids tracking," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[13] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4530–4537.

[14] X. Wang, R. Marcotte, G. Ferrer, and E. Olson, "ApriISAM: Real-time smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2486–2493.

[15] P. Chakravarty, P. Narayanan, and T. Roussel, "GEN-SLAM: Generative modeling for monocular simultaneous localization and mapping," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 147–153.

[16] Z. Yan and H. Zha, "Flow-based SLAM: From geometry computation to learning," *Virtual Reality Intell. Hardw.*, vol. 1, no. 5, pp. 435–460, 2019.

[17] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot. Sci. Syst. Conf.*, vol. 2, no. 9, 2014, pp. 1–9.

[18] J. Zhang and S. Singh, "Low-drift and real-time LiDAR odometry and mapping," *Auton. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[19] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[20] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3126–3131.

[21] T. Qin and S. Cao. (2019). *A-LOAM [DB/OL]*. [Online]. Available: https://github.com/HKUST- Aerial-Robotics/A-LOAM

[22] K. Koide. (2019). *SMRT-AIST [DB/OL]*. [Online]. Available: https://github.com/SMRT-AIST /interactive_SLAM

[23] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-fusion: LiDAR-inertial-camera odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5848–5854.

[24] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, Mar. 2019, Art. no. 1729881419841532, doi: 10.1177/1729881419841532.

[25] W. Shao, S. Vijayarangan, C. Li, and G. Kantor, "Stereo visual inertial LiDAR simultaneous localization and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 370–377.

[26] F. Neuhaus, T. Koß, and R. Kohnen, "Mc2SLAM: Real-time inertial LiDAR odometry using two-scan motion compensation," in *Proc. German Conf. Pattern Recognit.*, Cham, Switzerland: Springer, 2018, pp. 60–72.

[27] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D LiDAR inertial odometry and mapping," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, 2019, pp. 3144–3150.

[28] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5135–5142.

[29] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.

[30] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.

[31] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "LIPS: LiDAR-inertial 3D plane SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 123–130.

[32] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[33] G. Pandey, J. Mcbride, S. Savarese, and R. Eustice, "Extrinsic calibration of a 3D laser scanner and an omnidirectional camera," in *Proc. IFAC*, vol. 43, no. 16, 2010, pp. 336–341.

[34] J. Jiang, P. Xue, S. Chen, Z. Liu, X. Zhang, and N. Zheng, "Line feature based extrinsic calibration of LiDAR and camera," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Sep. 2018, pp. 1–6.

[35] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, Jun. 1995.

[36] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Trans. Robot. Automat.*, vol. 14, no. 4, pp. 617–622, Aug. 1998.

[37] J. Jeong, Y. Cho, and A. Kim, "The road is enough! Extrinsic calibration of non-overlapping stereo camera and LiDAR using road information," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2831–2838, Jul. 2019.

[38] J.-K. Huang and J. W. Grizzle, "Improvements to target-based 3D LiDAR to camera calibration," *IEEE Access*, vol. 8, pp. 134101–134110, 2020.

[39] T. O. Chan, D. D. Lichti, and D. Belton, "Temporal analysis and automatic calibration of the velodyne HDL-32E LiDAR system," *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 2, pp. 61–66, Oct. 2013.

[40] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun. 1963.

[41] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, Feb. 2012.
[42] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," *Int. J. Robot. Res.*, vol. 31 no. 2, pp. 216–235, Feb. 2012.

LINSHUAI ZHANG received the Ph.D. degree in intelligent mechanical systems engineering from Kagawa University, Japan, and the Ph.D. degree in mechanical engineering from the Changchun University of Science and Technology, China. He is currently an Associate Professor with the School of Intelligent Medicine, Chengdu University of Traditional Chinese Medicine, Chengdu, China. He is also an Overseas High-Level Overseas Talent in Sichuan Province. His current research interests include medical robot systems for minimal invasive surgery, teleoperation systems, and intelligent mechanical system and sensing.

QIAN WANG received the B.S. degree from the Taiyuan University of Science and Technology, Taiyuan, China. He is currently pursuing the master's degree with the School of Automation, Chengdu University of Information Technology.

His research interests include laser SLAM, autonomous vehicle localization and mapping, indoor/outdoor seamless localization, and multi-sensor fusion localization technology.

SHUOXIN GU received the B.S. degree from the Changchun University of Science and Technology (CUST), China, in 2011, the M.S. degree from Jilin University (JLU), China, in 2015, and the Ph.D. degree in intelligent mechanical systems engineering from Kagawa University (KU), Japan, in 2019. After that, she came to the Chengdu University of Information Technology to work as a Lecturer and continued to engage in research on underwater robots. She is also an Overseas High-Level Overseas Talent in Sichuan Province. She has published more than ten papers in recent years, including two journal articles and three conference papers as the first author. Her current research interests include spherical underwater robotics and bio-inspired robotics.

TAO JIANG received the Ph.D. degree in pattern recognition and intelligent system from Universität Siegen, Siegen, German, in 2013. He has been a Professor with the School of Intelligent Medicine, Chengdu University of Traditional Chinese Medicine, since 2022. He is currently an Overseas High-Level Overseas Talent in Sichuan Province, an Outstanding Expert with Outstanding Contributions in Sichuan Province, and won the Second Prize of Sichuan Science and Technology Progress Award, in 2018. He has been published more than ten papers in high-level international journals and many inventions and new patents for use have been listed in China.

SHIQI JIANG received the master's degree in instrument engineering and the Ph.D. degree in testing and measuring technology and instruments from the University of Electronic Science and Technology of China, in 2003 and 2008, respectively. He is currently a Professor with the School of Automation, Chengdu University of Information Technology, Chengdu, China. His current research interests include robot systems for industrial applications, robot sensing, special test equipment and test technology, and aeronautical test technology. He is also a Teaching Steering Committee Member of Sichuan Ordinary Undergraduate Colleges and Universities.

JIAJIA LIU received the M.Sc. and Ph.D. degrees from Southwest Jiaotong University, Chengdu, China, in 2012 and 2016, respectively. He is currently an Independent Research Consultant with the Department of Control Engineering, Chengdu University of Information Technology. His research interests include machine learning, pattern recognition, computer vision, and industrial inspection based on computer vision.

SHUANG LUO received the B.S. and M.S. degrees from the Chengdu University of Information Technology, Chengdu, China, in 2017 and 2020, respectively.

His research interests include laser SLAM, autonomous vehicle localization and mapping, indoor/outdoor seamless localization, and multi-sensor fusion localization technology.

GONGJUN YAN received the Ph.D. degree in computer science from Old Dominion University, in 2010. He is currently an Associate Professor with the University of Southern Indiana and has been working on the issues about intelligent transportation, vehicular ad-hoc networks, sensor networks, wireless communication, and machine learning. His main research interests include intelligent vehicles, security, privacy, routing, and intelligent systems. In years, he applied mathematical analysis to model behavior of complex systems and integrated existing techniques to provide comprehensive solutions.