RESEARCH ON LIDAR SLAM METHOD WITH FUSED POINT CLOUD INTENSITY INFORMATION

He Huang [1], Renzhong Wang [1], Junxing Yang [1 *], Chaowei Ma [1], Tianjiao Wang [1]

[1] School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing, China - huanghe@.bucea.edu.cn,2108570022064@stu.bucea.edu.cn,yangjunxing@bucea.edu.cn,machaowei@stu.bucea.edu.cn,2108160122006@stu.bucea.edu.cn

KEY WORDS: SLAM, point cloud intensity information, scene geometry information, point cloud alignment, loopback detection.

ABSTRACT:

Under the current trend of intelligence and automation, simultaneous positioning and mapping technology has become one of the research hotspots. The main problems of SLAM technology research are to improve the robustness of mapping and positioning, establish an efficient back-end optimization system, and improve the generalization of SLAM technology. This paper proposes to fuse the intensity information of the point cloud and the geometric information of the environment scene to construct a globally consistent environment feature descriptor and use the non-iterative two-step method to perform the nearest neighbour search on the point cloud in the point cloud registration stage. Then use the globally consistent descriptor that has been constructed to extract the laser point cloud descriptor by using the ring partition method, combine the method based on domain search to search for the closest point cloud frame, and finally use Intensity-ICP to complete the loopback frame. Fine registration, outputs the optimal pose transformation, to complete the loop detection. We use our self-built mobile platform to verify the robustness and generalization of the improved laser SLAM algorithm in public datasets and campus datasets. Experimental results show that the improved algorithm reduces the trajectory drift of the mobile platform and improves the efficiency of point cloud registration and loop closure detection.

1. INTRODUCTION

In today's rapidly advancing era of technology, the increasing demands of social production and everyday life have placed higher requirements on positioning and environmental perception. These requirements primarily revolve around achieving high precision, robustness, real-time performance, and applicability. Positioning information, being time-sensitive data, plays a crucial role in various practical applications. Delays or inaccuracies in positioning can have a significant impact on tasks such as planning and emergency response. Therefore, to leverage the full potential of spatiotemporal big data, it is vital to unify space and time. This entails minimizing positioning delays and improving the real-time performance and accuracy of data to support the smooth operation of diverse application scenarios. In outdoor environments, the Global Navigation Satellite System (GNSS) remains the most widely adopted positioning system. Several countries and regions are actively developing global positioning and navigation systems, including the United States GPS, Russia's GLONASS, China's Beidou, and the European Galileo satellite system. Current research in the field of GNSS primarily focuses on improving positioning accuracy and robustness in challenging scenarios and environments, such as urban canyons and indoor areas where GNSS signals may be weak. However, these single-sensor positioning systems have inherent limitations. GNSS, for instance, is susceptible to signal weakness and rapid attenuation in urban operations, leading to poor robustness, low accuracy, and the "last mile" problem. Inertial sensors, on the other hand, suffer from accumulating errors over time, resulting in rapid divergence of positioning accuracy. Additionally, current outdoor positioning systems do not fully exploit the available sensor data, leading to suboptimal positioning performance.

Simultaneous Localization and Mapping (SLAM) offers a solution to the "last mile" problem in outdoor positioning. The concept of SLAM was initially proposed by Cheeseman et al. at the IEEE Platform and Automation Conference in 1986. Its objective is to introduce estimation-based theoretical methods to address the challenges of platform mapping and positioning. Over the course of more than three decades, SLAM technology has evolved from using Kalman filtering to estimate the attitude of mobile systems (Smith et al., 1988) to the emergence of SLAM based on graph optimization theory (A.-Latif et al., 2013). Modern SLAM technology emphasizes robustness, efficiency, and multi-threaded processing (Huang et al., 2020). Depending on the sensor category, SLAM systems can be categorized into laser SLAM, visual SLAM, and multi-sensor fusion SLAM. Among various sensors, lidar stands out due to its all-weather usability, high precision, and strong reliability. In recent years, the development trend of laser SLAM has focused on achieving high robustness, precision, and generalization capabilities. Compared to other laser SLAM algorithms, LOAM introduced a pioneering dual-thread simultaneous positioning and mapping system based on environmental geometric features (Ji and Singh, 2017). This approach significantly improved the usability and real-time performance of laser SLAM. Since then, many laser SLAM algorithms and multi-sensor fusion SLAM algorithms have been developed based on LOAM. However, there are still some challenges in the current mainstream laser SLAM methods. These include the high computing power required by point cloud distortion methods, resulting in poor real-time performance; insufficient utilization of lidar sensor information, leading to limited positioning accuracy; and existing loop detection algorithms being prone to missing loopback frame detection when trajectory drift errors are significant.

To address these issues, this paper builds upon the original LOAM architecture and proposes improvements. Specifically, a simultaneous positioning and mapping method is presented that

incorporates both point cloud geometric feature information and environmental point cloud intensity information. The enhanced laser SLAM algorithm's robustness and generalization are validated using a self-built mobile platform on campus and public datasets. The rest of this paper discusses the details of the proposed method and is structured as follows. Section II reviews the proposed methods, including point cloud registration by fusing point cloud intensity information and loop closure detection based on intensity information optimization. Section III presents the experiments, including tests and performance comparisons of the algorithms on two datasets. Finally, Section IV summarizes the conclusions.

2. METHOD

2.1 SLAM Algorithm Fused With Point Cloud Intensity Information

This paper proposes a lidar SLAM algorithm framework (Figure 1), including front-end laser odometer and back-end loopback detection. Front-end tasks include preprocessing and motion distortion compensation for point clouds, extracting local features, such as line and surface features, and adding them to the local feature map to estimate the pose of the platform. Meanwhile, the intensity information of the point cloud is also calibrated and encoded into the ISC feature map. Back-end tasks include: loop detection, calculating the feature similarity score to determine whether a loop occurs and publish loop information, global consistency optimization, by constructing a factor map and receiving the current line and surface features, pose estimation and loop constraints. An optimized pose is generated, and finally, a final trajectory and a global consistency map are generated (Debeunne and Vivet, 2020).

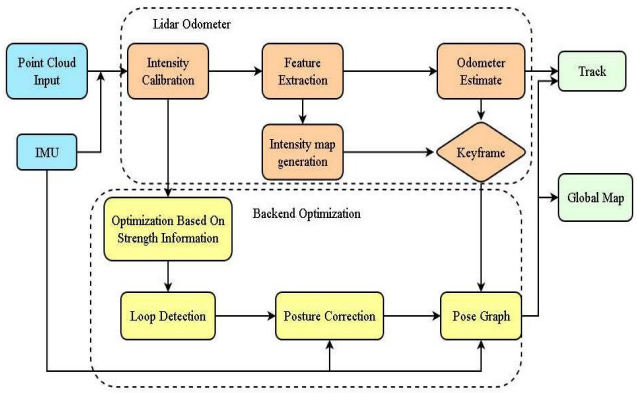

Figure 1. Algorithm framework

2.2 Point Cloud Registration by Fusion of Point Cloud Intensity Information

Point cloud registration is the process of unifying point clouds in different coordinate systems into the same coordinate system. It is realized by finding an optimal rigid body transformation matrix from the current coordinate system to the reference coordinate system. This transformation matrix can be in different coordinates. The point clouds in the same coordinate system are registered to point clouds in the same coordinate system so that they have similar spatial positions in the same coordinate system (Bing and Vemuri, 2011). In this paper, feature point selection and feature matching use geometric information features and intensity information features at the same time. Finally, in the rigid body transformation matrix estimation, the best pose estimation is calculated by minimizing the geometric error and intensity error.

2.2.1 Feature Point Selection

This paper utilizes features based on environment geometry information and intensity information instead of using only geometry features. The calibrated intensity information contains the reflectivity profile of objects in the environment, reflecting the distribution of different objects in the environment. Therefore intensity information is helpful for point cloud registration. For each point $p_i \in P$ and its intensity value $\tilde{\eta}_i \in \tilde{\tau}$, by searching for the closest point $N_i \in P$, the local distance distribution σ_i^{ς} and its intensity distribution value are computed using σ_i^{τ}:

$$\sigma_i^{\varsigma} = \frac{1}{|N_i|} \cdot \sum_{p_j \in N_i} (|P_i - P_j|) \tag{1}$$

$$\sigma_i^{\tau} = \frac{1}{|N_i|} \cdot \sum_{p \in N_i} (|\eta_i - \eta_j|) \tag{2}$$

where $\sigma_i^{\tau}, \sigma_i^{\varsigma}$ = distribution values for intensity and distance

p_i, p_j = a point cloud

η_i, η_j = a point cloud intensity value

$|N|$ = point cloud collection

By weighting and summing these features, the salient features are selected:

$$\sigma_i = w^{\varsigma} \cdot \sigma_i^{\varsigma} + w^{\tau} \cdot \sigma_i^{\tau} \tag{3}$$

where $I = (a_{ij}) \in \mathbb{R}^{N_r \times N_s}, a_{ij} = \phi(P_{ij})$ = final distribution value

w^{ς}, w^{τ} = weighted values for environment geometry and intensity values

2.2.2 Build an Intensity Map

The intensity distribution map contains the reflectance distribution of the environment around the mobile device. For most SLAM systems that only use the geometric features of the environment for point cloud registration, the map is maintained and updated by the occupancy grid method or the octree method (Arnesen et al., 2018). This paper adopts the method of dividing the three-dimensional space into grid cells, expressing each cell with a probability function, and using this method to construct and update an intensity map. The probability function used in the past is replaced by using intensity values $I(\eta_i \mid z_{1:t})$ for each grid cell. For the observation of each grid cell at time t, the surface reflectance can be updated by the following method:

$$M(m_i \mid z_{1:t}) = M(m_i \mid z_{1:t-1}) + \frac{\eta_{m_i} - M(m_i \mid z_{1:t-1})}{n_{m_i}} \tag{4}$$

Where $M(m_i \mid z_{1:t})$ = current observed intensity

n_{m_i} = The total number of observations for the grid cell

And when there is no environment object in the grid cell, the intensity value will be marked as 0.

2.2.3 Build Strength Residual

Intensity residuals are computed by matching features to the intensity map. It can be achieved by minimizing the intensity residual between the current point p_i (including edge features and planar features) and the transformed point $\hat{p}_i$ in the intensity map:

$$f_I(\hat{p}_i) = \eta_i - M(\hat{p}_i) \tag{5}$$

Where $M(\hat{p}_i)$ = Intensity values in the intensity map M

To search for intensity information in the intensity map, we introduce a trilinear interpolation method. Although it is more straightforward to represent the intensity using the nearest grid cell, the intensity information is less accurate, especially for large-scale maps with lower grid resolution.

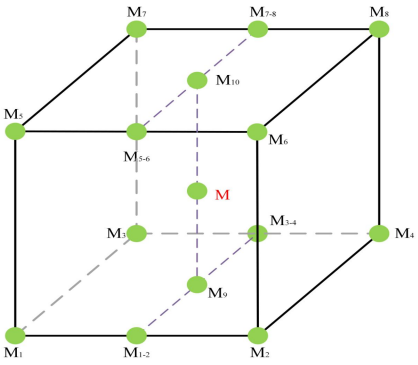

Figure 2. Linear interpolation

For each transformation point $\hat{p}_i = [x_i, y_i, z_i]^T$, we can search the eight surrounding grid cells, as illustrated in Figure 2. The intensity values of these grid cells are denoted as $M_1(\hat{p}_i)$,⋯, $M_8(\hat{p}_i)$ respectively. We define the point closest to the centre of the grid cell as $p_1 = [x_1, y_1, z_1]^T$ and the point farthest from the centre of the grid cell as $p_2 = [x_2, y_2, z_2]^T$. Based on the width, height, and depth of the grid cell, the calculation method for determining the intensity value of the target point is as follows:

$$M_{1-2}(\hat{p}_i) = \frac{x_i - x_1}{x_2 - x_1} \cdot M_1(\hat{p}_i) + \frac{x_i - x_1}{x_2 - x_1} \cdot M_2(\hat{p}_i) \tag{6}$$

Where $x_2 - x_1$ 、 $y_2 - y_1$ 、 $z_2 - z_1$ = The width, height and depth of each grid cell

Likewise, the values of M_{3-4}, M_{5-6} and M_{7-8} can be computed. Subsequently, the intensity value is estimated as follows:

$$M_9(\hat{p}_i) = \frac{y_i - y_1}{y_2 - y_1} \cdot M_{1-2}(\hat{p}_i) + \frac{y_i - y_1}{y_2 - y_1} \cdot M_{3-4}(\hat{p}_i) \tag{7}$$

$$M_{10}(\hat{p}_i) = \frac{y_i - y_1}{y_2 - y_1} \cdot M_{5-6}(\hat{p}_i) + \frac{y_i - y_1}{y_2 - y_1} \cdot M_{7-8}(\hat{p}_i) \tag{8}$$

$$M_{10}(\hat{p}_i) = \frac{y_i - y_1}{y_2 - y_1} \cdot M_{5-6}(\hat{p}_i) + \frac{y_i - y_1}{y_2 - y_1} \cdot M_{7-8}(\hat{p}_i) \tag{9}$$

2.3 Loop Closure Detection Based on Intensity Information Optimization

The loop closure detection algorithm in this paper uses the environmental geometric information and point cloud intensity information to construct a scene descriptor to describe the scene. In the part of environment geometric information, a local key point will be selected, and the geometric shape of the point cloud around the local key point will be encoded once. In this process, the three-dimensional information of the point cloud is converted into two-dimensional encoding, and the height information of the local point cloud is saved, which helps to preserve the three-dimensional information of the scene during the information conversion process

2.3.1 Scene Descriptor Construction

The 3D point cloud scanned by the lidar is divided vertically and horizontally based on the sensor coordinates. N_s and N_r are the number of sectors and rings, respectively. Assuming that the maximum sensing range of the lidar sensor is $L_{\max}$, the central angle of the sector is equal to $2\pi N_s$. In the experiments in this paper, use $N_s = 60$ and $N_r = 20$. Through vertical and horizontal division, this frame of the point cloud is divided into several areas, as illustrated in Figure 3, which is a collection of points belonging to the overlapping area of the ith ring and the jth sector.

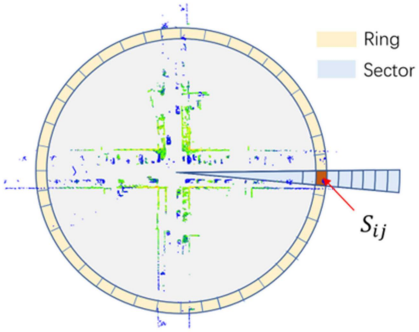

Figure 3. Point cloud frame space division

The division of the point cloud area is based on the geometric centre of the scanned point cloud of one frame as the centre of the divided area so that the physical area of the area away from the centre is wider, but in the process of encoding, the point cloud areas of different sizes are equally encoded into the same location descriptor. Therefore, dynamic objects in the environment are considered noise.

The encoding value of each point cloud area is $\phi : P_{ij} \to \mathbb{R}$, which contains the encoding of the maximum height of the point cloud area $\phi(P_{ij}) = \max_{p \in P_{ij}} \cdot z(p) p_k = [\rho_k, \theta_k, z_k, \eta_k]$. where $z(\cdot)$ is a function of the Z coordinate value of point p . Finally, the point cloud position descriptor is represented as a matrix of $N_r \times N_s$.

$$I = (a_{ij}) \in \mathbb{R}^{N_r \times N_s}, a_{ij} = \phi(P_{ij}) \tag{10}$$

Where I = location descriptor
P_{ij} = point cloud area

2.3.2 Intensity Descriptor Construction

The construction of local descriptors will consume more computing power in the actual processing process. In this process, local geometric features, such as the local norm of each key point, need to be identified. Therefore, to improve computational efficiency, this paper constructs the intensity information as a global descriptor, which can effectively integrate geometric information and intensity features into a

global descriptor. The global feature descriptor, η denotes the intensity reading, the three-dimensional space position of the point is denoted by $[x, y, z]$, n denotes the number of points in the point cloud, a lidar scan is denoted by $P = [p_1, p_2, p_3, \cdots, p_n]$, and each point is denoted by $p_k = [x_k, y_k, z_k, \eta_k]$ in a frame of lidar point cloud coordinate system. Carry out coordinate system conversion, and express the spatial position of the point cloud in polar coordinate system, so:

$$p_k = [\rho_k, \theta_k, z_k, \eta_k] \tag{11}$$

$$\rho_k = \sqrt{x_k^2 + y_k^2} \tag{12}$$

$$\theta_k = \arctan\frac{y_k}{x_k} \tag{13}$$

Where x_k, y_k, z_k = 3D coordinates

ρ_k, θ_k = polar angle and radius

η_k = intensity value

Next, the divided point cloud regions are represented as follows:

$$S_{ij} = \{p_k \in P \mid \frac{i \cdot L_{\max}}{N_r} \le \rho_k < \frac{(i+1) \cdot L_{\max}}{N_r}, \frac{j \cdot 2\pi}{N_s} - \pi \le \theta_k < \frac{(j+1) \cdot 2\pi}{N_s} - \pi\} \tag{14}$$

Where $i \in [|1, N_s|], j \in [|1, N_r|]$

N_s, N_r = number of sectors and rings

$L_{\max}$ = radar sensor maximum range

S_{ij} = point cloud area

Since each subspace is much smaller than a point cloud frame, the intensity readings are assumed to be constant. Therefore, for each subspace, an encoding function can be applied to reduce the intensity dimensionality. The definition of this function is:

$$\eta_{ij} = k(S_{ij}) = \max_{p_k \in S_{ij}} \eta_k \tag{15}$$

If there is no available data in the S_{ij} and $\eta_{ij} = 0$, the intensity geometry descriptor can be expressed as follows:

$$\Omega(i, j) = \eta_{ij} \tag{16}$$

Where Ω = global feature descriptor

η_{ij} = a point cloud intensity reading

3. EXPERIMENT

The hardware platform used in this article is a self-built mobile platform equipped with a chassis and sensors including lidar and IMU. Install the sensor on the mobile platform bracket, and use a Lenovo notebook as the algorithm layout platform, and deploy the algorithm on it. The mobile platform is shown in Figure 4, where the laser radar is Pandar40p, a product of HESAI, with 40 laser lines, a maximum detection distance of 200 meters, a horizontal market angle of 360 degrees, a horizontal angle resolution of 0.2 degrees, and a vertical angle of view The field is 40 degrees, the vertical angle range is -25° ~ +15°and the data acquisition frequency is 10HZ and 20HZ. The chassis of the autonomous mobile platform used is the HUNTER chassis of the Songling platform, which reserves a variety of interfaces and supports the access of a variety of sensors, which facilitates the secondary development of researchers. The full name of the industrial computer is industrial control computer. The autonomous mobile platform uses an industrial computer as the computing centre and embeds the Ubuntu system as the operating system to provide the hardware foundation for data transmission and sensor control.

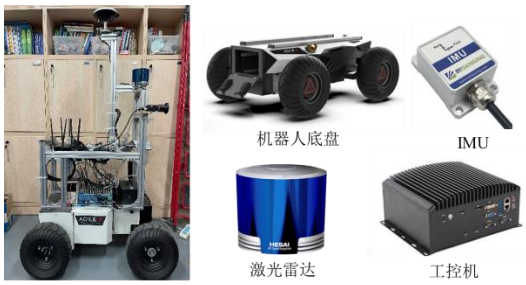

Figure 4. Hardware Platform and Composition

In the SLAM algorithm, the current standard accuracy evaluation metric involves utilizing the evaluation of the odometry and SLAM (evo) algorithm to assess the accuracy of the trajectory. The primary evaluation metrics encompass the Absolute Pose Error (APE) and Relative Pose Error (RPE) (Sturm et al., 2012). The APE calculation involves comparing the pose information calculated by the system with the assumed true trajectory information based on continuous time data. By measuring the discrepancies between the poses at corresponding time points, the trajectory error at each time point is obtained. RPE, on the other hand, calculates the root mean square error of the Lie algebra of each pose within the trajectory over continuous time. This error estimation incorporates both rotational and translational errors of the motion pose. The error values of APE and RPE encompass various dimensions, including the maximum error value, minimum error value, average error value, median error value, standard deviation, root mean square error, and other indices.

3.1 Public Dataset Experiment

The data set used in the experiment is KITTI's Odometry collection, which was co-founded by the Karlsruhe Institute of Technology in Germany and the Toyota American Institute of Technology. It is one of the largest open-source data sets currently used in autonomous driving (Geiger et al. ., 2012).

Regarding the robustness of the laser SLAM system, this paper conducts five experiments on the four parts of the Odometry part 00-03 of the Kitti dataset to test the robustness and robustness of the algorithm in different scenarios. Compared with the Lego-Loam algorithm (Shan and Englot, 2018) with better performance, and analyzed various error indicators, the results are as follows:

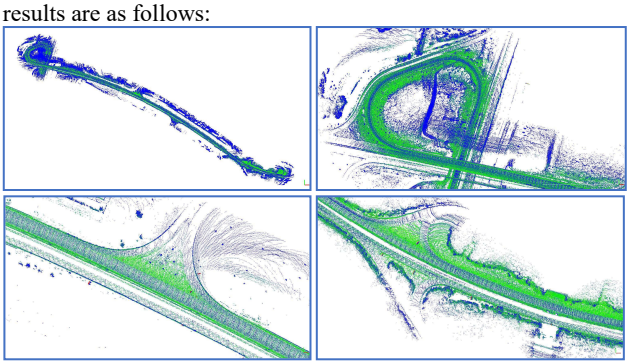

Figure 5. Kitti odometry dataset 00 run experiment

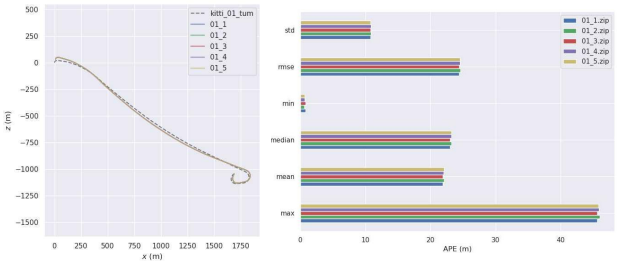

Figure 6. Kitti odometry dataset 00 runs five times trajectory overlap with real trajectory comparison and each error value

Through experiments on the Kitti Odometry dataset 00 (Figure 5, Figure 6), it is found that the improved algorithm maintains stable mapping and positioning in non-closed-loop and structured scenes, compared with the given true value The trajectory deviation is small, and the error dispersion keeps fluctuating within a certain range, about 2%. Compared with the algorithm before improvement, it is found that the trajectory deviation is small, and the differences in standard deviation, medium error, and covariance remain in a stable range. The algorithm as a whole is more stable than before the improvement.

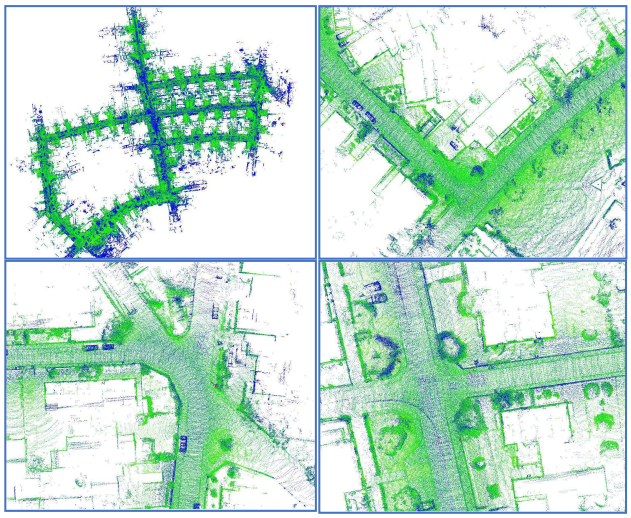

Figure 7. Kitti odometry dataset 01 algorithm run results

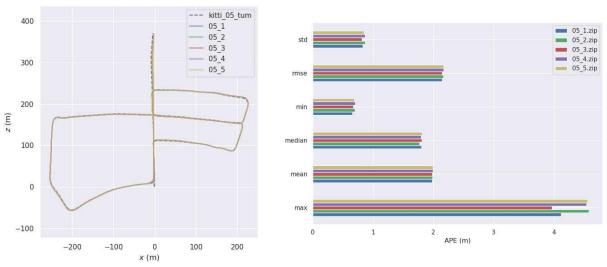

Figure 8. Kitti odometry dataset 01 runs five times trajectory overlap with real trajectory comparison and each error value

Combined with the analysis of the experimental results in Figures 7 and 8, when the improved algorithm runs in a closed-loop environment, the trajectory and mapping accuracy obtained from multiple experiments and the error of the true trajectory are relatively small. The error values remain stable, the standard deviation and the mean value of the error fluctuate at 1% for multiple experiments, the root mean square error, the minimum value of the error and the median of the error fluctuate at 3%, the maximum error fluctuates at 13%, and the maximum error shows unstable. Overall, the trajectory error remains stable, and the algorithm is robust.

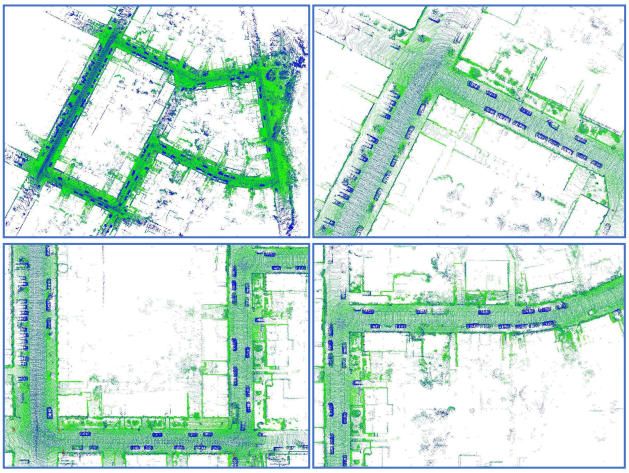

Figure 9. Kitti odometry dataset 02 algorithm run results

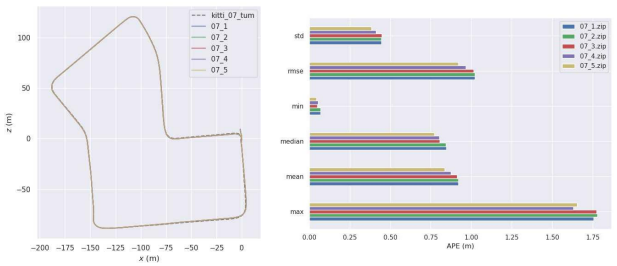

Figure 10. Kitti odometry dataset 02 runs five times trajectory overlap with real trajectory comparison and each error value

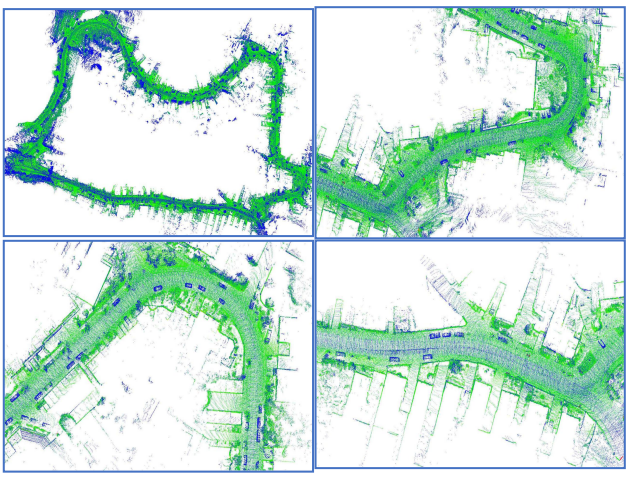

Figure 11. Kitti odometry dataset 03 algorithm run results

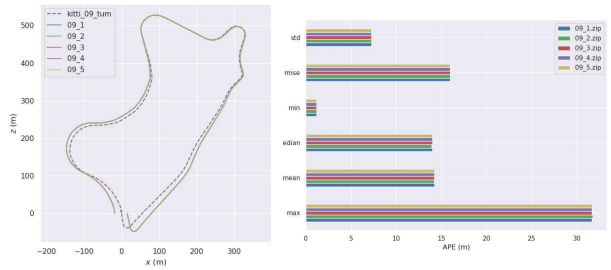

Figure 12. Kitti odometry dataset 03 runs five times trajectory overlap with real trajectory comparison and each error value

Figures 5 to 12 show the effect of the improved algorithm tested on the Kitti dataset Odometry set. Combining the above figure, it can be seen that the point cloud image built by the improved algorithm has no obvious trajectory offset, the structured scene map is clear, and the point cloud registration effect has no obvious overlapping shadows. By comparing the improved algorithm with five experiments on the data set, it is found that the algorithm shows good robustness and robustness, and the algorithm performance is stable under the data sets of different scenarios. By observing Figures 5 and 11, it is found that when simultaneous positioning and mapping are performed on straight and non-loop roads, as the running time increases, the degree of deviation of the mapping trajectory from the given true value also increases. The loopback detection module has a poor backend optimization effect. Comparing Figure 7, Figure 9, and Figure 11, it is found that when the algorithm is running on a straight road, the trajectory error gradually increases. When the trajectory forms a closed loop, when loop closure detection and back-end optimization are performed, the error quickly converges, forming a better map. Effect. Next, through quantitative error analysis, compare the errors between the improved algorithm and previous algorithms, and judge the mapping effect of the improved algorithm.

KITTI	00					01				
Algorithms	Mean	Std	Rmse	Maxe	Min	Mean	Std	Rmse	Max	Min
A-LOAM	19.34	8.35	17.46	47.55	0.95	133.64	102.98	168.71	188.68	31.79
L C-Fusion	19.21	12.42	22.88	46.01	0.68	19.12	12.39	22.78	57.11	19.44
Lego-Loam	19.57	11.15	21.92	64.58	0.61	3.68	4.65	5.93	27.75	0.97
ISC-LIO	20.01	10.79	24.51	45.79	0.70	1.99	0.85	2.16	4.35	0.68
ORB-SLAM2	22.01	10.29	24.31	46.88	0.66	22.18	10.95	24.95	64.46	7.49

Table 1. The improved algorithm is compared with the mainstream algorithm, and ISC-LIO is the improved algorithm

By comparing the improved algorithm with the current mainstream SLAM algorithm, it is found that on the Kitti dataset 00, the road is straight, the scene is single and structured, and the current SLAM algorithm performs similarly. On the Kitti dataset 01, there are closed-loop scenes on the road with high scene richness, and there are unstructured scenes. The improved algorithm shows excellent performance. The error of the current mainstream algorithm gradually increases with time. In unstructured scenes, the point cloud registration effect is poor, and the overall trajectory error is highly discrete. Compared with the closest LEGO-LOAM, the improved algorithm improves the Mean value by 45%, the error range is narrowed, and the standard deviation and root mean square error are the best. In structural scenes and scenes with closed loops, the mapping error is reduced and the positioning accuracy is improved.

3.2 Campus Dataset Experiment

The improved laser SLAM algorithm was tested using the campus dataset, with the primary focus on Building A of the Foundation Building and Building D of the College Building. These locations are marked in Figure 13. Building A of the Foundation Building and Building D of the College Building represent structured and closed-loop scenes within the dataset.

Figure 13. Campus experiment scene

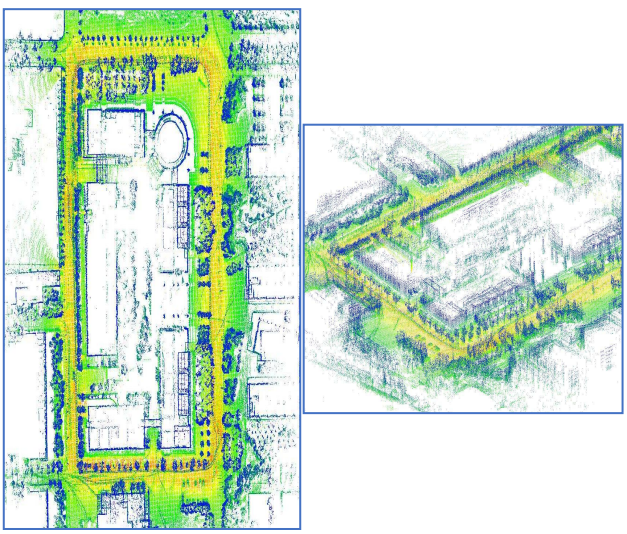

Figure 14. Experimental Effects of the Foundation Building Building A

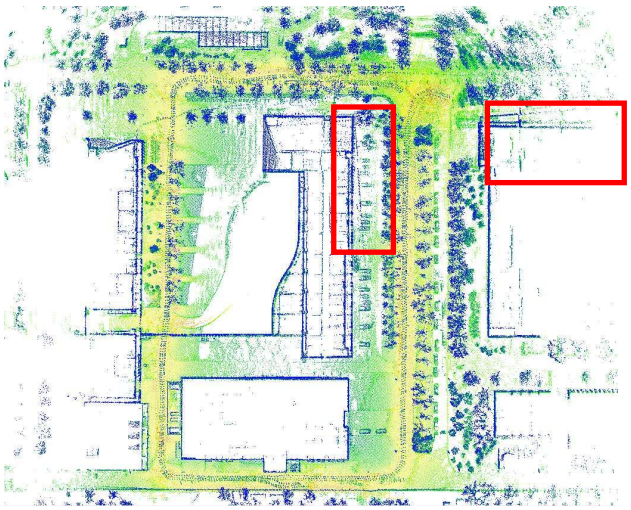

Figure 15. The effect of running the algorithm in College Building D before the improvement

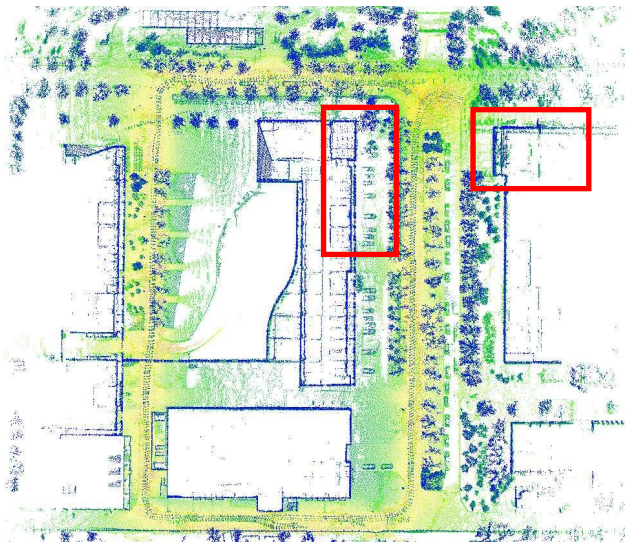

Figure 16. The effect of the improved algorithm running in College Building D

The collected campus environment data was integrated into the algorithm, and the point cloud was played in chronological order to generate a real-time point cloud map of the campus environment and the mobile platform's movement trajectory. Figure 14 displays the experimental results for Block A of the Foundation Building. It can be observed that the improved algorithm demonstrates favourable performance in closed-loop environments, particularly during loop closure. Comparing the experimental outcomes depicted in Figure 15 and Figure 16 for Block D of the College Building, the results obtained using the improved algorithm (highlighted within the red box) exhibit several notable improvements. The outline of the surrounding objects in the point cloud map is more distinct, and the road point cloud map shows no significant shadows or overlapping issues. Overall, the mapping and positioning effects are enhanced.

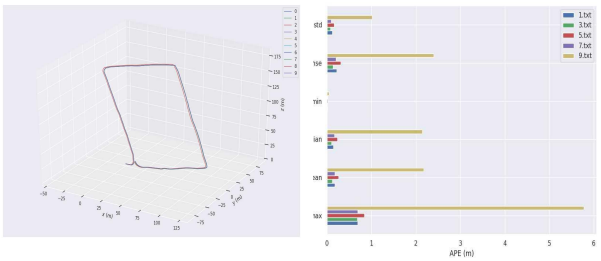

Figure 17. Comparison of the trajectory of ten experiments in Block D of the Foundation Building with the true value and the error value of each

The surrounding environment of the basic building D was selected to test the algorithm of the campus data set. After ten experiments (Figure 17), it can be found that the trajectory coincidence degree of the mobile platform is high, the algorithm performance is robust, and the algorithm robustness is good. Through the above experiments, it can be found that the improved algorithm performs better in various outdoor scenarios, the improved algorithm has good experimental results, and the campus test is relatively successful.

	Error	LEGO-LOAM	ISC-LIO
RPE	Max	1.750	0.726
	Min	0.017	0.009
	Mean	0.379	0.183
	Median	0.217	0.160
	Std	0.653	0.111
	RMSE	0.432	0.2l4
	SSE	308.480	222.565

Table 2. Comparison of Improved Algorithm and LEGO-LOAM Error Values

The improved algorithm and the LEGO-LOAM algorithm were tested using the campus data set. The improved algorithm showed a better effect in terms of relative error performance. Compared with the LEGO-LOAM algorithm, the improved maximum and minimum error ranges The LOAM is reduced by 30%, and the error value is small; the average error value is reduced by 20%, and the overall mapping and trajectory errors are reduced; the median error is 77% of the latter, and the overall error value is reduced; the root mean square error is reduced by 50%, and the overall The degree of dispersion of errors is better. It can be found that compared with LEGO-LOAM, the improved algorithm has greatly improved the construction of point cloud maps and the improvement of positioning accuracy on the campus dataset.

4. CONCLUSION

Based on the existing laser SLAM algorithm, this paper fuses point cloud intensity information and environmental geometric information to construct a globally consistent feature descriptor, improve the accuracy of point cloud registration, improve the loop detection algorithm, and ensure the effectiveness of loop detection. To improve the robustness, robustness and accuracy of laser SLAM algorithm mapping and positioning, and use network public datasets and campus datasets for experimental verification, analysis shows that mapping and positioning in environments with closed-loop trajectories and unstructured

scenes The accuracy of positioning has been improved, but there are still some shortcomings. For example, the improved algorithm requires a lot of computing power, and when the algorithm experiment is performed on a personal notebook, optimization failures and pose loss occur many times. And when the improved algorithm is used for mapping and positioning of large outdoor scenes, the robustness and robustness of the algorithm decrease, and when the data is run for more than 15 minutes, the trajectory drift will increase rapidly.

In the future development of this work, we plan to build a multi-sensor fusion SLAM system, using sensors such as lidar, binocular camera, GNSS, IMU, etc. to build a composite SLAM system that can support richer scenarios.

ACKNOWLEDGEMENTS

This research was funded by the National Natural Science Foundation of China (Grant Numbers 42201483) and the China Postdoctoral Science Foundation (Grant Numbers 2022M710332).

REFERENCES

Arnesen, B.O., Sandy, S.S., Schjlberg, I., et al., 2018. Probabilistic Localization and Mapping of Flexible Underwater Structures using Octomap[C]// European Control Conference 18.

A.-Latif, D.M., Salem, A.M., Ramadan, H., et al., 2013. Comparison of Optimization Techniques for 3D Graph-based SLAM[C]// Proceedings of the 4th European Conference of Computer Science (ECCS '13).

Bing, J., Vemuri, B.C., 2011. Robust Point Set Registration Using Gaussian Mixture Models[J]. IEEE Trans Pattern Anal Mach Intell, 33(8):1633-10645.

Debeunne, C., Vivet, D., 2020. A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping[J]. Sensors, 20(7):2068.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE.

Huang, H.B., Zhao, J., Liu, J. 2020. A Survey of Simultaneous Localization and Mapping with an Envision in 6G Wireless Networks.pdf.

Ji, Z., Singh, S., 2017. Low-drift and Real-time Lidar Odometry and Mapping[J]. Autonomous Robots, 41(2):401-416.

Smith, R., Self, M., Cheeseman, P., 1988. Estimating Uncertain Spatial Relationships in Robotics[J]. Machine Intelligence & Pattern Recognition, 5(5):435-461.

Sturm, J., Engelhard, N., 2012. Endres F., et al. A benchmark for the evaluation of RGB-D SLAM systems[J]. IEEE.

Shan, T., Englot, B., 2018. LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable ter-rain. 2018 IEEE/RSJ Int. Conf. on Int. Rob. and Syst., IEEE, 4758–4765