Multiscale Space Adaptation-based Dynamic Object Removal for LiDAR Mapping

Haitao Xu
Institutes of Physical Science and Information Technology, Anhui University
Hefei 230031, China
q21301384@stu.ahu.edu.cn

Hanqi Wang
Hefei Institutes of Physical Science, Chinese Academy of Sciences
Hefei 230031, China
whq97@mail.ustc.edu.cn

Xueshi Zhang
Hefei Institutes of Physical Science, Chinese Academy of Sciences
Hefei 230031, China
zhangxs@mail.ustc.edu.cn

Pengfei Zhou
Hefei Institutes of Physical Science, Chinese Academy of Sciences
Hefei 230031, China
Anhui Engineering Laboratory for Intelligent Driving Technology and Application
Hefei 230031, China
pfzhou@hfcas.ac.cn

Huawei Liang*
Hefei Institutes of Physical Science, Chinese Academy of Sciences
Hefei 230031, China
Innovation Research Institute of Robotics and Intelligent Manufacturing, Chinese Academy of Sciences
Hefei 230031, China
hwliang@iim.ac.cn

2023 IEEE International Conference on Unmanned Systems (ICUS) | 979-8-3503-1630-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICUS58632.2023.10318308

***Abstract*—Simultaneous Localization And Mapping (SLAM) is a crucial technology for mobile robots to locate themselves when working in a location environment. However, when a large number of dynamic objects appear in the environment, the positioning and mapping results of the robot will be affected. At present, online real-time removal algorithms can effectively solve such problems. Therefore, this paper proposes a dynamic object removal algorithm based on multi-scale spatial information, which identifies and eliminates moving objects in the environment during self-localization and environmental modeling. In this paper, the Inertial Measurement Unit (IMU) data is used to preliminarily judge the robot's position and attitude transformation, and the dynamic target distance is combined to adjust the target feature extraction scale, and modify the dynamic object detection conditions to screen objects with high confidence. To remove misjudgment information caused by irregular objects in the environment, road information is used to remove misjudgment information and obtain dynamic object coordinate information in the environment. Through quantitative and qualitative experiments, it has been proven that this algorithm can improve the positioning accuracy of robots in urban road scenes and optimize environmental models. This algorithm reduces Absolute Pose Error by 0.11032m.**

***Index Terms*—SLAM, ground segmentation, dynamic object removal**

I. Introduction

With the development of mobile robots, the requirements for robot positioning accuracy are constantly improving. Simultaneous Localization And Mapping(SLAM) [1] technology is widely used in unmanned delivery vehicles, unmanned buses, and the construction of high-precision maps [2]. SLAM refers to the process and analysis of sensor data by a robot in an unknown environment to determine its current pose and model the environment. SLAM can be divided into LiDAR-based SLAM and camera-based SLAM based on sensor types. Because LiDAR-based SLAM technology is not constrained by natural light and can adapt to various complex environments while still maintaining high accuracy, a large number of excellent algorithms for LiDAR-based SLAM have been published in recent years, including LOAM [3], LeGO-LOAM [4], SC-LeGO-LOAM, LIO-SAM [5], FAST-LIO [6].

This paper improves on the SC-LeGO-LOAM algorithm. The original algorithm combines the ScanContext loop detection algorithm with the LeGO-LOAM algorithm, removing ground points at first, and then using the segmentation module to limit feature points to extract the target, extracting edge point classes and plane point classes in the segmented target. The IMU pre-integration module calculates the transformation amount between two feature points to obtain the robot's transformation matrix. Finally, This paper uses the ScanContext loop detection algorithm to optimize the backend mapping process. However, when the environment is complex and a large number of dynamic objects appear, the dynamic environment will affect the robot's judgment of its own posture. In addition, dynamic objects can also affect the mapping process [7]. To eliminate the impact of dynamic objects in the environment on SLAM, there are two mainstream methods: online real-time removal and post-processing removal. However, the post-processing removal method only removes dynamic objects

*Corresponding Author: Huawei Liang.

This work was supported by the National Key Research and Development Program of China (Grant No. 2020AAA0108103).

979-8-3503-1630-8/23/$31.00 ©2023 IEEE

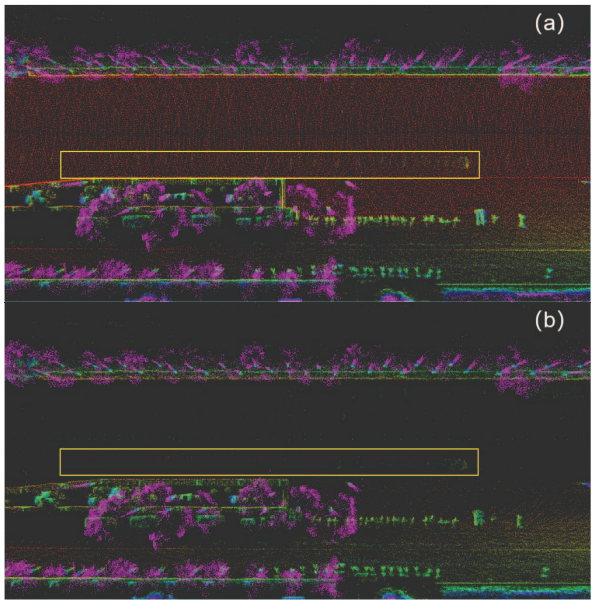

Fig. 1. Ghosts left by dynamic objects in the environment can be clearly seen in the yellow box in Figure (a). After removing out the dynamic objects in Figure (b), the ghost phenomenon in the yellow box disappears.

from the map and cannot eliminate the impact of dynamic objects on robot positioning.

Therefore, to improve the positioning accuracy of SLAM in complex environments, this paper will start from the front end and calculate the pose transformation matrix of the robot using Inertial Measurement Unit (IMU) data [8]. Due to the working principle of LiDAR, the data distribution features are presented as near dense and far sparse. To improve the ability of object detection, this paper will adjust the grid scale and object detection threshold according to the distance of the target, and modify the conditions of dynamic object detection. Using the pose transformation matrix, project multiple frames of spatial information data onto the current radar coordinate system. By overlapping multiple frames of data, objects that meet dynamic conditions are removed. This article combines road boundary information to remove erroneous judgment data caused by irregular objects and obtain real dynamic object spatial position information. The final effect is shown in Fig. 1. In Fig. 1 (a), due to the lack of removal of dynamic objects, dynamic object data was retained during the construction phase, forming a ghost image. The result of using the method proposed in this paper is shown in Fig. 1 (b), where the dynamic object data in the map is deleted and the ghost phenomenon disappears.

II. RELATED WORKS

In recent years, people have been continuously optimizing the SLAM algorithm from different directions. This section will introduce the relevant work in the ground segmentation module and dynamic object removal module.

A. Ground Segmentation

As one of the most important parts of SLAM preprocessing, ground segmentation can not only improve the accuracy of pose estimation in the feature matching stage but also reduce the amount of data that needs to be calculated in the future and reduce computational time [9] [10]. At present, the mainstream ground segmentation methods mainly include model fitting and geometric feature-based ground segmentation algorithms.

1) Model fitting method: The model fitting method relies on multiple models to fit ground points, commonly including line fitting [11], Gaussian fitting, and plane fitting [12]. The point where the parameters are controlled within the threshold range through multiple iterations is considered a ground point. Hyungtae Lim [13] used the method of local ground plane fitting to divide the space into circular grid areas; Seungjae Lee [14] adaptively calculated appropriate parameters based on adaptive ground likelihood estimation. However, when the robot encounters a curved road surface or a ground with a certain slope, it will determine it as a ground point and the computational complexity is high.

2) Based on geometric feature method: Based on geometric feature methods, classification is usually carried out based on geometric features, such as normal vectors, grid height differences, and so on. The normal vector method calculates the normal vector between adjacent points and judgment points. By setting an angle threshold, the angle between the normal vector of each point and the ground normal vector is calculated, and compared with the threshold for classification. This method can achieve high scores in accuracy, but there may be misjudgments. LeGO-LOAM [4] calculates the horizontal angle between two points to determine whether the point cloud meets the ground point threshold condition and obtains the segmented ground point cloud. The grid height difference divides the environment into grids of specific shapes and then projects the original point cloud onto the grid where it is located. The maximum height difference is calculated for each point in the grid. If the difference meets the ground point condition, the grid is the ground. However, this algorithm is limited by distance and equipment. When the distance is too far and the point hitting the ground cannot be determined whether it is a ground point, it becomes invalid.

B. Dynamic Object Removal

1) Online real-time removal: Online real-time removal, including pre and post-frame comparison, frame-to-local map comparison, clustering, and other methods. David Yoon [15] used the front and back frames as reference frames, and the front reference frame is used to compare the distance between points. Points with excessive distance are identified as potential dynamic points and then placed in the back reference frame for verification. The confirmed dynamic points are clustered and grown as seed points in the point cloud to obtain dynamic objects. The idea of RF-LIO [16] is to construct a local map in real-time near the robot and extract dynamic points from the local map by comparing real-time environmental data with the point cloud distribution of the local map through depth images.

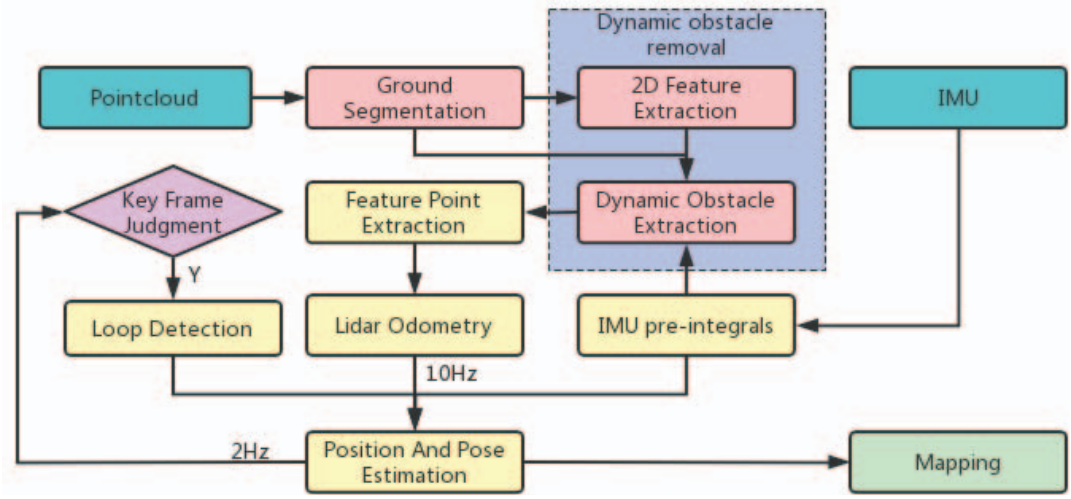

Fig. 2. Flow chart of Multiscale Space Adaptation-based Dynamic Object removal for LiDAR Mapping.

LIO-Livox used the Euclidean clustering method to cluster the original point cloud into ground points, background points, and front scenic spots. The front scenic spots are considered dynamic objects and are removed during the feature extraction process. Due to real-time limitations, online real-time removal cannot perform a large amount of recognition and judgment. And there may be misjudgments. Moreover, the algorithm cannot meet the robustness requirements and cannot adapt to various environments and devices.

2) Post processing removal: The post-processing removal method does not require high real-time performance, so all frames within the entire SLAM cycle can be used as reference information to identify dynamic points. Compared to real-time methods, the post-processing method has higher accuracy and sufficiency in removing dynamic point clouds. J Schauer and A Nuechter [17] proposed a grid-based method for dynamic removal, which determines whether a grid is dynamic based on its hit and pass through. Giseop Kim [18] adopted the method of viewpoint visibility. The basic idea is to project the original data into a depth map, and then project the local map near the original data into a depth map from the same viewpoint. Finally, they compared the pixel depths at the same location on two depth maps. If the latter has a shallower depth, the corresponding point on the local map for that pixel position is a dynamic point. The basic idea of ERASOR [19] is to divide the original data and local maps in the original data into sector-like grids according to the same location, and then compare the grid of the original data with the grid of the local map, according to the difference in the distribution of point clouds in the grid, remove the dynamic points in the local map grid. Post-processing removal usually involves identifying dynamic targets in the environment after the robot has judged the current pose. By comparing the built models, dynamic targets that may exist in the environment are removed, which does not improve the accuracy of the front-end robot pose and only optimizes the map accuracy in the mapping process.

III. METHODS

A. System Overview

The algorithm flowchart of this paper is shown in Fig. 2. Based on the SC-LeGO-LOAM algorithm, the proposed method optimized the ground segmentation algorithm in the front-end and added a dynamic obstacle removal algorithm. Firstly, the ground segmentation module removes ground points from the LiDAR data and records the information on the ground points in the two-dimensional plane. Next, the dynamic object removal algorithm projects the segmented point cloud onto the grid based on multi-scale spatial information. Calculating the transformation matrix between the current robot pose and the historical pose using IMU data and using the transformation matrix to project multiple frames of information onto the current radar coordinate system. Comparing and processing the multi-scale spatial information of multiple frames to obtain the spatial coordinate information of dynamic obstacles in the environment. Due to the complex urban road environment, irregular objects on both sides of the road can affect the accuracy of the dynamic obstacle recognition process. Therefore, this paper will use road information to constrain dynamic object detection and remove the information of wrong judgment, and transmit the processed point cloud data to the feature extraction module. Then, the feature point matching module is responsible for extracting line surface features from the processed points and performing scan-to-scan feature matching, using two nonlinear optimizations to estimate the relative motion transformation matrix between two radar frames. The loop detection module saves the calculated global coordinates of the radar and determines whether it is a keyframe. If the current frame is a keyframe, loop detection will be performed to optimize the global path. Finally, the algorithm saves the historical point cloud according to the radar trajectory.

B. Ground Segmentation Algorithm based on Multi-scale Ray Slope Threshold

This paper adopts a ground segmentation algorithm based on a multi-scale ray slope threshold, which is suitable for urban roads with small slopes or multiple obstacles. The principle of the ray slope threshold method is that there is a difference in the angle between the front and rear points of the same ray when scanning the road surface and obstacles. The judgment is based on the angle difference between the point to be judged and the front and rear points on the ray, as well as the height value of the point. At the same time, the distance difference between the front and rear points and the angle change rate is used to adaptively adjust the global height threshold to adapt to different urban road surfaces, and removing the ground point cloud can ensure accurate removal of slope points, as shown in Fig. 3.

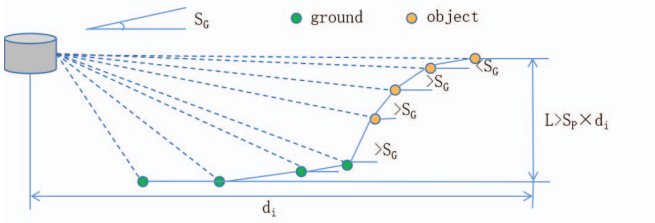

Fig. 3. Schematic diagram of ground segmentation based on multi-scale slope threshold.In the figure, the green points are ground points, and the yellow points are obstacle points.

Firstly, the pre-processed point cloud is organized in an orderly manner in the form of rays, and the entire scene is evenly divided into $360°/\theta$ fan-shaped regions, where θ represents the horizontal resolution of the LiDAR. Then it calculates the angle between each point and the x-positive direction in the robot coordinate system, divides by the horizontal resolution, and rounds down to assign each point to its subregion A_m:

$$m = \lfloor \frac{arctan(y_i/x_i)}{\theta} \times \frac{180°}{\pi} \rfloor \tag{1}$$

where m is the index of the sector subregion. x_i and y_i are the horizontal and vertical coordinates of the i-th point. Therefore, the point cloud in each region is equivalent to the formation of the same vertical ray cluster, so the position information of points in the same region can be transformed from three-dimensional (x_i, y_i, z_i) to the corresponding horizontal distance d_i and vertical distance z_i of the distance radar. d_i is defined as:

$$d_i = \sqrt{x_i^2 + y_i^2} \tag{2}$$

The use of local height threshold S_P and global angle threshold S_G can improve recognition accuracy. The angle α_i is calculated between adjacent two points in the same area and the xoy plane:

$$\alpha_i = \arctan(\frac{z_i - z_{i+1}}{d_i - d_{i+1}}) \tag{3}$$

After obtaining α_i, it is compared with the global angle threshold S_G. If the angle between adjacent two points is less than S_G, it is considered that the range between these two points is the ground area. However, it may encounter some special situations, such as when judging the roof area, the angle between adjacent two points is also less than the threshold. If the judgment still follows this standard, there will be a misjudgment. This paper introduces a height threshold H to address this issue. If an obstacle point with a slow angle change is encountered and its height is greater than H, it is considered an obstacle point.

However, when encountering sloping ground after scanning for obstacles, it is also possible to determine them as obstacle points due to the impact of the slope. At this point, this paper considers using a multi-scale local height threshold to fuse the height threshold with the object distance d_i and defines the local height threshold H_i as:

$$H_i = S_P \times d_i \tag{4}$$

where S_P represents the local height threshold coefficient of the i-th point. d_i represents the horizontal distance from the i-th point to the radar.

At this point, this paper sequentially traverses the point cloud in sequence to determine whether it meets the global angle threshold. If the calculated angle is less than the S_G of the point, then calculate the local height threshold H_i of the point. If it meets the threshold range, it is determined as a ground point. If it is greater than the local height threshold, it is determined as an obstacle point. While traversing the point cloud, the proposed method constructs a two-dimensional grid plane to record the distribution information of ground points within a range of 100 meters in front of the robot, as shown in Fig. 5 (c). The method adjusts the grid size to one grid every 0.5 meters, and records the number of ground points in the vertical space:

$$Ground(i,j)_t = num(points[a], \cdots, points[b]) \tag{5}$$

where $num()$ calculates the number of points that fall in the grid.

C. Dynamic Object Removal

The dynamic obstacle detection module receives the ground-segmented point cloud, projects it onto a two-dimensional plane, records the corresponding spatial information, and performs residuals on the two-dimensional point cloud information of the first two frames. The remaining part is the dynamic objects in the environment. However, the working environment is not only dynamic, but the robot itself is also dynamic. If the robot is in a stationary state, it is very easy to obtain dynamic objects in the environment. However, due to the robot moving, it needs to obtain the pose transformation matrix between the current state and the previous time. At this time, this paper needs to pre-integrate the IMU data.

1) Posture estimation based on IMU pre-integration: In this paper, the method will represent L as the radar coordinate system, I as the IMU coordinate system, W as the world coordinate system, and P as the radar point cloud. Firstly, it considers the robot itself as a rigid body, and the motion state

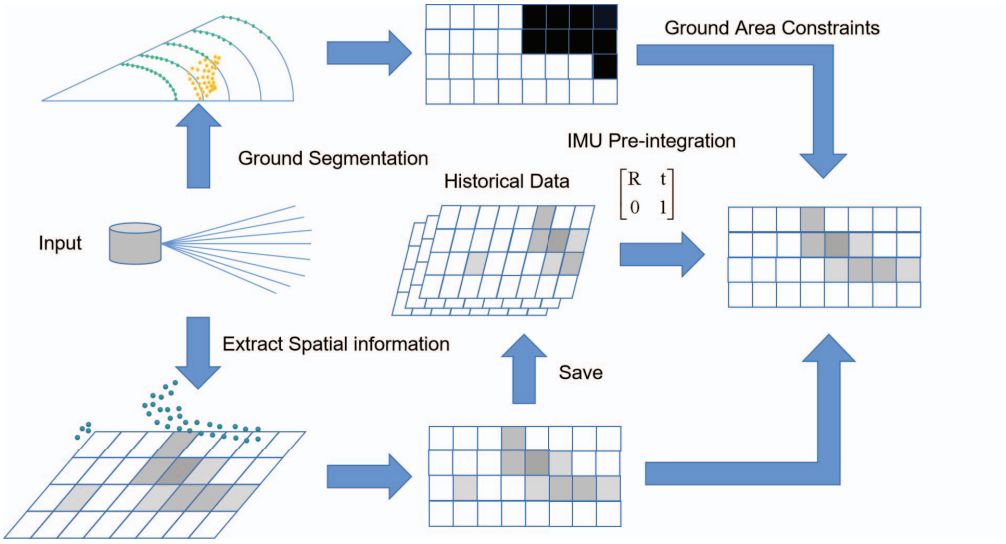

Fig. 4. Schematic diagram of dynamic Object removal.As shown in the figure, after the radar data is transferred in, the final processing result is obtained through the surface point extraction and dynamic object detection module and the combination of the two processing results.

of the IMU device is consistent with the radar state. Therefore, the state of the robot itself can be written as:

$$S = [V^{\tilde{T}}, P^{\tilde{T}}, R^{\tilde{T}}, b^{\tilde{T}}] \tag{6}$$

where V is the speed of IMU relative to the world coordinate system, b is the acceleration and gyroscope offset in IMU, T is the conversion from radar to the world coordinate system, and R is the Rotation matrix of T.

The robot state at time t can be calculated as:

$$\hat{\omega_t} = \omega_t + b_{\omega t} + n_\omega \tag{7}$$

$$\hat{a_t} = a_t + b_{at} + n_a \tag{8}$$

where n_ω and n_a represent the white noise generated by the IMU itself, $b_{\omega t}$ represents the bias of the gyroscope, and b_{at} represents the bias of the accelerator.

Next, we can use the IMU pre-integration method to calculate the current state at the time:

$$p^{\omega}_{b_{k+1}} = p^{\omega}_{b_k} + v^{\omega}_{b_k}\Delta t_k + \iint_{t\in[k,k+1]} [R^{\omega}_t(\hat{a_t} - b_{a_t}) - g^{\omega}]dt^2 \tag{9}$$

$$v^{\omega}_{b_{k+1}} = \int_{t\in[k,k+1]} [R^N_t(\hat{a_t} - b_{a_t}) - g^{\omega}]dt \tag{10}$$

$$q^{\omega}_{b_{k+1}} = q^{\omega}_{b_k} \otimes \int_{t\in[k,k+1]} \frac{1}{2}\Omega(\hat{\omega_t} - b_{\omega t}) q^{b_k}_t dt \tag{11}$$

where $q^{\omega}_{b_{k+1}}$ represents the transformation matrix of the pose from time k to time $k+1$ in the robot's own coordinate system.

The pose transformation matrix between two frames of IMU data is obtained through IMU pre-integration. However, the input frequency of radar data is slower than the IMU frequency. At this point, this paper needs to calculate the transformation matrix of the robot between the two radar frames:

$$q^{\omega}_{b_t} = q^{\omega}_{b_k} \otimes q^{\omega}_{b_{k+1}} \otimes \cdots \otimes q^{\omega}_{b_{k+t}} \tag{12}$$

where $q^{\omega}_{b_t}$ represents the transformation matrix of the robot at time t.

2) Dynamic object removal: Next, the method will remove the point cloud data after ground segmentation based on distance. Due to the working principle of mechanical LiDAR, the vertical distribution density is lower than the horizontal distribution density, resulting in a low probability of long-distance moving objects being scanned and low point density, which affects the subsequent judgment of dynamic objects. Therefore, this paper removes areas with long distances and low density based on point cloud distribution characteristics.

The proposed method projects point cloud data onto a two-dimensional grid plane and record the distribution information of obstacles within a range of 100 meters in front and back, and 50 meters in the left and right of the robot, with a grid size of 0.2 meters. Then it records the height of the highest point of the grid and the number of points it falls on, and save:

$$Maxh(i,j)_t = max(points[a].z, \cdots, points[b].z) \tag{13}$$

$$Density(i,j)_t = num(points[a], \cdots, points[b]) \tag{14}$$

where $max()$ calculates the height value of the highest point that falls on the grid. As shown in Fig. 5 (a), the maximum height information is saved and it can be seen that there are two vehicles driving on the road.

Next it read the coordinate transformation $q^{\omega}_{b_{t-1}}$ and $q^{\omega}_{b_{t-2}}$ between the current frame pose and the previous two frames pose, and project the historical frames $Maxh(i,j)_{t-1}$, $Maxh(i,j)_{t-2}$, $Density(i,j)_{t-1}$, and $Density(i,j)_{t-2}$ into the radar coordinate system at the current time t through the transformation matrix:

$$Maxh(i,j)^t_{t-1} = q^{\omega}_{b_{t-1}} \otimes Maxh(i,j)_{t-1} \tag{15}$$

$$Maxh(i,j)^t_{t-2} = q^{\omega}_{b_{t-2}} \otimes q^{\omega}_{b_{t-1}} \otimes Maxh(i,j)_{t-2} \tag{16}$$

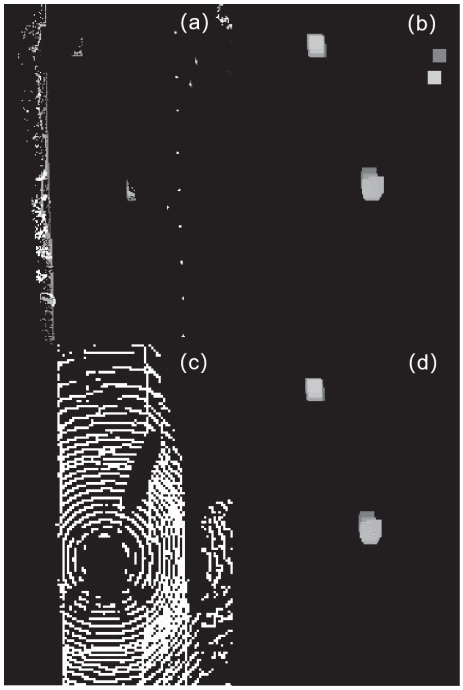

Fig. 5. Figure (a) is the spatial grid information extracted from the environment, Figure (b) is the dynamic Object detection grid map without adding ground constraints, Figure (c) is the ground point grid information after ground segmentation, and Figure (d) is the dynamic Object detection grid map after adding ground constraints.

$$Density(i,j)^{t}_{t-1} = q^{\omega}_{b_{t-1}} \otimes Density(i,j)_{t-1} \quad (17)$$

$$Density(i,j)^{t}_{t-2} = q^{\omega}_{b_{t-2}} \otimes q^{\omega}_{b_{t-1}} \otimes Density(i,j)_{t-2} \quad (18)$$

The method traverses the spatial and historical information in the grid in sequence and determines whether there are dynamic objects in the grid by setting a height threshold D_{ij} based on the distance between objects. If there is object information in $Maxh(i,j)_t$ and both $Maxh(i,j)_t - Maxh(i,j)^t_{t-1}$ and $Maxh(i,j)_t - Maxh(i,j)^t_{t-2}$ are less than the height threshold D_{ij}, it can be determined that there are dynamic objects in the grid.

However, due to the irregular environment and the presence of positional changes in the robot itself, the confidence level of dynamic obstacle recognition cannot be guaranteed. In this case, density information can be used to assist in judgment. Set c as the density change coefficient. If it is determined that there are obstacles in the grid (i,j) at this time, then compare the grid with historical spatial information. When $Density(i,j)_t - Density(i,j)^t_{t-1}$ and $Density(i,j)_t - Density(i,j)^t_{t-2}$ are smaller than $c \times Density(i,j)_t$, it is considered that there are dynamic objects in the grid.

If all the above conditions are met, the probability of dynamic objects in the grid is high, and the height of the highest point in the grid is saved:

$$Dynamic(i,j) = Maxh(i,j)_t \quad (19)$$

Due to the fact that the information currently calculated is only a partial point on a dynamic object, it has now grown from known dynamic points to the entire dynamic class based on spatial information. When it is determined that there is a dynamic object in grid $Dynamic(i,j)$, as shown in Fig. 5 (b), based on $Maxh$ information, determine whether the height difference between adjacent grids is within the set height difference threshold. Assuming λ is the height difference coefficient, calculate the value of $|Maxh(i,j)_t - Maxh(i+a, j+b)_t|(a,b \in [-1,0,1])$ to determine whether it falls within the range of $[0, \lambda \times Maxh(i,j)]$. If the condition is met, it is considered that the adjacent two grids are the same object.

By utilizing expansion and corrosion operations, accurate spatial information on dynamic objects can be obtained. As shown in the figure, the information of two dynamic vehicles in the environment in the grid is obtained through recognition.

However, due to the presence of irregular objects in the environment, they lead to misjudgment. To remove misjudgment information, the method uses information $Gound(i,j)_t$, as shown in Fig. 5 (d), to determine whether the dynamic obstacle is on the road. If the object exists on the road, the dynamic information of the grid is considered true. If it does not appear on the road, it is judged as misjudgmental information and deleted.

IV. EXPERIMENTS

A. Experimental Platform

Fig. 6. Data Collection PlatForm.The data collection vehicle is equipped with 128-line LiDAR, XW-GI7660 inertial navigation equipment, GPS Antenna satellite positioning equipment, and ARK-3500 industrial control computer.

The dataset in this paper adopts urban environmental data collected on a laboratory unmanned vehicle platform. The data collection equipment is shown in Fig. 6. The unmanned vehicle platform in this paper is equipped with 128-line LiDAR, XW-GI7660 inertial navigation equipment, GPS Antenna satellite positioning equipment, and ARK-3500 industrial control computer. The data was collected on highly dynamic and complex urban roads, covering a range of 15 kilometers. It includes high-speed cars (60 km/h), medium-speed trucks (30 km/h), and low-speed pedestrians (10 km/h). The algorithm in this paper is deployed on an i7-8700 CPU desktop computer using the Ubuntu 16.04 Robot Operating System (ROS).

B. Qualitative Experiment

Fig. 7. Comparison between the effect of adding dynamic object removal module and real satellite maps.The red dots represent the map created by the algorithm in this paper for point cloud data, which is consistent with the actual satellite map.

This paper will qualitatively analyze the methods proposed in this paper and SC-LeGO-LOAM using real-life scenarios, as shown in Fig. 7. This paper simulates three scenarios for removing dynamic targets of different shapes: high-speed cars, low-speed bicycles, and medium-speed trucks. The experimental results are shown in Fig. 8. The method in this paper can accurately identify dynamic vehicles on the road, while the SC-LeGO-LOAM method cannot remove dynamic vehicles on the road, and ground points cannot be accurately removed. In Figure (a), there is a small car traveling at high speed on the road. The dynamic object removal method proposed in this paper can effectively remove point clouds on the car, and through the optimized algorithm in this paper, the ground points are removed more cleanly. From Figure (b), it can be seen that there are still a large number of feature points falling on the ground and on the dynamic car. In the scene shown in Figure (c), there is a bicycle. Due to its small size, there are fewer points that fall on the dynamic object itself. However, this paper accurately identifies moving objects of different sizes in the environment through changes in object spatial information. The original algorithm does not extract feature points due to its small size. In the scene shown in Figure (e), which includes a truck and a sedan, compared to Figure (f), our algorithm can still propose point clouds on dynamic objects. This proves that when multiple moving objects of different sizes appear in the environment, our algorithm can still achieve accurate recognition and removal of dynamic objects.

C. Quantitative Experiment

To visually see the impact of removing dynamic obstacles on the SLAM effect, this paper also designed quantitative

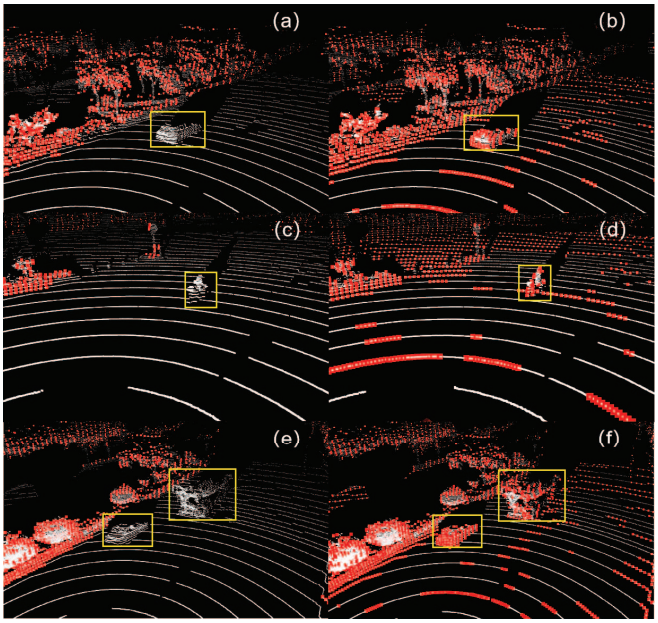

Fig. 8. Comparison of dynamic object removing effects in different scenes. The effectiveness of our algorithm is verified from three different scenarios in the figure. The yellow box represents dynamic objects in the environment. The white points are unprocessed original point clouds. The red dots are the processed point clouds passed to the odometer. The left side is the rendering of the algorithm processed in this paper. On the right is the rendering of the ground segmentation algorithm that has not been optimized and has not removed out dynamic objects in the environment. The yellow box retains a large number of dynamic object points and cannot accurately remove ground points.

experiments to intuitively demonstrate the algorithm's ability to improve the accuracy of the SLAM process, using absolute pose error(APE) as the evaluation criterion.

$$APE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} ||log(T_{gt,i}^{-1} T_{esti,i})||_2^2} \tag{20}$$

where $T_{esti,i}$ is the estimated trajectory and $T_{gt,i}$ is the actual trajectory.

By mapping the real environment, the algorithm proposed in this paper adds a dynamic object removal module to improve the accuracy of SLAM, eliminating the impact of multiple dynamic targets on positioning accuracy in urban scenes facing complex traffic environments. Under the same data and testing environment, the method proposed in this paper was compared with SC-LeGO-LOAM, as shown in Table I.

TABLE I
SCORE OF SC-LEGO-LOAM AND OUR ALGORITHM IN APE METRICS

	SC-LeGO-LOAM(m)	OURS(m)
APE_{max}	2.381198	2.341001
APE_{mean}	1.431347	1.316879
APE_{min}	0.758090	0.684640
APE_{median}	1.388065	1.305024
APE_{rmse}	1.479748	1.369428
APE_{sse}	4366.171344	3739.414962

Compared with the SC-LeGO-LOAM method, the method in this paper reduces 0.040197 meters at the maximum value, 0.114468 meters at the mean square deviation, 0.07345 meters at the minimum value, 0.083041 meters at the median value, 0.11032 meters at the root-mean-square deviation, and 626.75638 meters at the square sum of errors. And through calculation, the average time after using the method in this article is 0.9425ms. The algorithm ensures the real-time performance of mapping. Through quantitative experimental results, it is shown that the proposed method improves the performance and positioning accuracy of SLAM by removing dynamic obstacles. Compared with the original SC-LeGO-LOAM, the proposed method effectively reduces errors in high dynamic scenarios, and improves the robustness and positioning accuracy of SLAM.

V. CONCLUSION

This paper proposes a SLAM algorithm that can accurately identify and remove dynamic objects in the environment. This paper starts from the front-end and optimizes the ground segmentation algorithm. It adopts a ground segmentation algorithm based on a multi-scale ray slope threshold and saves the distribution information of ground points. Then, multi-scale spatial information is used to extract the spatial features of objects in the environment, and historical data is used to remove dynamic objects in the environment. Ground constraints are added to remove misjudgment information caused by irregular objects outside the road, and high-confidence dynamic object distribution information is obtained. Finally, the method uses distribution information to remove dynamic objects and input the remaining point cloud data into the SLAM system. The method proposed in this paper does not require training in advance or clustering of targets, which will improve the accuracy of dynamic Object detection, ensure that there is no impact of dynamic target point cloud on the back-end odometer in feature extraction, and improve the robustness and positioning accuracy of SLAM. In future work, we plan to study the removal of some ultra-low speed and temporarily stationary obstacles in special scenarios, in order to improve the effectiveness of later loop back, mapping, and positioning.

REFERENCES

[1] Jinseok Woo and Naoyuki Kubota. Simultaneous localization and mapping using a robot partner in dynamic environment. In *SICE Annual Conference 2011*, pages 524–529, 2011.

[2] Hanqi Wang, Zhiling Wang, Linglong Lin, Fengyu Xu, Jie Yu, and Huawei Liang. Optimal vehicle pose estimation network based on time series and spatial tightness with 3d lidars. *Remote Sensing*, 13(20), 2021.

[3] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems Conference*, 2014.

[4] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765, 2018.

[5] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142, 2020.

[6] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022.

[7] Jun Lee, Mikyoung Seo, JinHwan Kim, Soyoung Hwang, Taehoon Kim, and Kyoung-Sook Kim. Management of subdivided dynamic indoor environments by autonomous scanning system. In *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 224–227, 2018.

[8] Hanqi Wang, Zhiling Wang, Linglong Lin, Fengyu Xu, Jie Yu, and Huawei Liang. Optimal vehicle pose estimation network based on time series and spatial tightness with 3d lidars. *Remote Sensing*, 13(20), 2021.

[9] Zhihao Shen, Huawei Liang, Linglong Lin, Zhiling Wang, Weixin Huang, and Jie Yu. Fast ground segmentation for 3d lidar point cloud based on jump-convolution-process. *Remote Sensing*, 13(16), 2021.

[10] Weixin Huang, Huawei Liang, Linglong Lin, Zhiling Wang, Shaobo Wang, Biao Yu, and Runxin Niu. A fast point cloud ground segmentation approach based on coarse-to-fine markov random field. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7841–7854, 2022.

[11] M. Himmelsbach, Felix v. Hundelshausen, and H.-J. Wuensche. Fast segmentation of 3d point clouds for ground vehicles. In *2010 IEEE Intelligent Vehicles Symposium*, pages 560–565, 2010.

[12] Dimitris Zermas, Izzat Izzat, and Nikolaos Papanikolopoulos. Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5067–5073, 2017.

[13] Hyungtae Lim, Minho Oh, and Hyun Myung. Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3d lidar sensor. *IEEE Robotics and Automation Letters*, 6(4):6458–6465, 2021.

[14] Seungjae Lee, Hyungtae Lim, and Hyun Myung. Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13276–13283, 2022.

[15] David Yoon, Tim Tang, and Timothy Barfoot. Mapless online detection of dynamic objects in 3d lidar. In *2019 16th Conference on Computer and Robot Vision (CRV)*, pages 113–120, 2019.

[16] Rf-lio: Removal-first tightly-coupled lidar inertial odometry in high dynamic environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4421–4428, 2021.

[17] Johannes Schauer and Andreas Nüchter. The peopleremover—removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid. *IEEE Robotics and Automation Letters*, 3(3):1679–1686, 2018.

[18] Giseop Kim and Ayoung Kim. Remove, then revert: Static point cloud map construction using multiresolution range images. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10758–10765, 2020.

[19] Hyungtae Lim, Sungwon Hwang, and Hyun Myung. Erasor: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3d point cloud map building. *IEEE Robotics and Automation Letters*, 6(2):2272–2279, 2021.