Proceedings of 2022 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Guilin China

Semantic Lidar Odometry and Mapping for Mobile Robots Using RangeNet++

Xiangyu Dong[1], Guo He[2], Peipei Fan[1], Fei Zhang[2], Teng Li[1]

[1]State Grid Anhui
Ultra High Voltage Company
Limited Hefei, Anhui Province, China

Junyi Zhou[2], Jia Xie[1], Junjie Zhang[1], Jie Huang[1],
Weiwei Shang[2]

[2]Department of Automation
University of Science and Technology of China
Hefei, Anhui Province, China

2022 IEEE International Conference on Mechatronics and Automation (ICMA) | 978-1-6654-0853-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMA54519.2022.9856091

***Abstract* - The point cloud registration methods in the existing Simultaneous Localization and Mapping (SLAM) systems are all based on static assumptions, and dynamic points will affect the registration accuracy. Therefore, it is difficult for SLAM systems to operate in a highly dynamic environment. If only the geometric information of the point cloud data is used, further processing is required to avoid the influence of dynamic objects on the point cloud registration algorithm. To this end, we utilize the RangeNet++ method to identify the semantic information of 3D laser point clouds. By combining with semantic information, outliers caused by highly dynamic objects can be directly filtered out, and the objective function of point cloud registration in laser odometry is optimized. We can build high-quality semantic maps that are semantically and geometrically consistent. The experimental results prove that more accurate lidar odometry results can be obtained. The high-quality full-static semantic maps will be established to ensure the completeness of the traversable area of the mobile robot in subsequent navigation planning.**

Index Terms - SLAM; dynamic points; RangeNet++; semantic information; semantic maps.

I. Introduction

Among the various functions of robots, odometer estimation and mapping are the basic conditions for mobile robots to achieve intelligence. High-resolution 3D lidar has become one of the main sensors in SLAM research because of its wide horizontal field of view (FOV), unaffected by illumination changes, and ability to perceive environmental details from a distance. After more than 30 years of research since SLAM was proposed, the technology of using lidar sensors for odometer estimation and mapping has become more and more mature. Accurate positioning and environmental maps can be obtained [1-3].

However, the point cloud registration methods in SLAM systems are all based on static assumptions. When operating in highly dynamic environments, SLAM systems that only use point cloud geometry information for point cloud registration struggle to generate consistent maps due to the presence of moving objects. At this time, dynamic points need to be detected and filtered in real time to ensure the reliability of the entire SLAM system. Yoon *et al*. [4] used two frames of laser scans before and after the current frame as reference frames, and compared the point-to-point distances to confirm the dynamic points of the current frame. Schauer *et al*. [5] determined that the grid is dynamic based on whether the grid is hit or not. Kim *et al*. [6] projected the current scan and nearby local maps as depth images, respectively. The dynamic points in the local map can be judged by comparing the depth of the same position on the two depth maps. Similarly, Lim *et al*. [7] compared the difference in point cloud distribution between the current scan and the nearby local map in the divided fan-shaped small area, and eliminated the dynamic points in the local map.

With the continuous development of deep learning and the popularization of 3D lidar, algorithms applied to semantic segmentation of laser point cloud data have also received extensive attention and research. Combined with the semantic information of point cloud segmentation, the influence of highly dynamic objects can be directly avoided [8], and the semantic information of the mapping area can also be obtained to realize the intelligent navigation behavior of the robot. Semantic segmentation is also an important task in semantic scene understanding. In recent years, thanks to some large datasets for point cloud training and learning [9-11], 3D semantic segmentation methods based on deep learning have been developed. The RangeNet++ method proposed by Milioto *et al*. [12] is also inspired by projection-based methods. Using the range 2D image as input and reprojecting the segmented point clouds into 3D point clouds, the entire lidar point clouds can be accurately segmented at a higher frequency. This method of identifying the semantic information of each point in the point clouds also lays an important foundation for the lidar semantic SLAM algorithm [13].

In this paper, we utilize the semantic information provided by the fully convolutional neural network RangeNet++ [12] to implement semantic lidar odometry and build high-quality semantic maps. The spherical projection of 3D point cloud data is performed and different point cloud semantic datasets are constructed for the network. By combining the acquired semantic information, we add a semantic weight factor to the objective function of point cloud registration and filter out the corresponding dynamic objects in the semantic maps. Finally, the experimental results show that the proposed system improves the accuracy of lidar odometry and the quality of semantic maps, laying an important foundation for subsequent intelligent navigation planning.

The remainder of this paper is organized as follows: Section II presents various aspects of the proposed system in detail. In Section III, we analyze the experimental results and the conclusion is presented in Section IV.

978-1-6654-0853-0/22/$31.00 ©2022 IEEE

II. Proposed System

To enhance the semantic understanding of the scene, we use Convolutional Neural Network (CNN) to perform full semantic segmentation on lidar point cloud data. In addition, we construct a semantic SLAM system based on the semantic information of point cloud, which improves the robustness of the system and the applicability of the maps. The overall system framework is shown in Fig. 1.

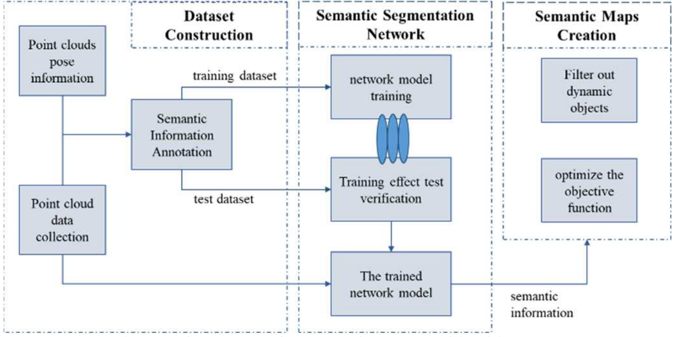

Dataset Construction
Semantic Segmentation Network
Semantic Maps Creation
Point clouds pose information
Semantic Information Annotation
Point cloud data collection
training dataset
test dataset
network model training
Training effect test verification
The trained network model
Filter out dynamic objects
optimize the objective function
semantic information

Fig. 1. The system framework proposed in this paper

We provide semantic information of point clouds by using the fully convolutional network RangeNet++ for semantic segmentation of point clouds. The network model input parameters are modified according to the applied lidar sensor model. In order to achieve good training results, we use the large-scale KITTI semantic dataset [11] for training and testing, which is a large number of dense semantic annotations for the original KITTI dataset. At the same time, a semantic dataset of the corresponding actual environment is also constructed to ensure the semantic segmentation effect of the trained network on the actual test environment. Finally, based on the obtained semantic information, a robust semantic SLAM algorithm is constructed to build an accurate and informative semantic map. The specific design analysis of each module in the entire algorithm is given below.

A. Semantic Segmentation Network for Point Clouds

Mechanical LiDARs scan on a vertical plane by mechanical rotation to image. This method similar to the camera's line-by-line exposure will bring about the jelly effect, that is, when the carrier robot or the detected object is moving, different parts of the moving object will not be exposed at the same time due to the line-by-line exposure, which will lead to shaking and slope images. Therefore, to obtain a geometrically consistent representation of the environment, we convert each frame's 3D point cloud data into a 2D image data representation, which is called a spherical projection image of the point cloud. As shown in the schematic diagram of projected coordinate transformation in Fig. 2, the projection is not a simple expansion of the lidar scanning surface, but a process of transforming the base coordinate system. Each pixel in the projected image can contain multiple measurements in descending order of point cloud distance, which ensures that all points presented in the image are within the current field of view of the sensor.

As shown in Fig. 2, for each point $\mathbf{p}_i\left(x,y,z\right)$ in the point cloud, the origin of lidar coordinates is placed in the center of the image. The yaw angle and pitch angle can correspond to the position of each pixel in the image, and the final coordinate transformation is:

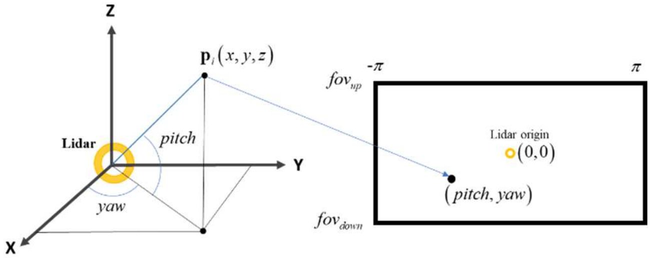


Fig. 2. Schematic diagram of spherical projection coordinate transformation

$$\begin{cases} u=\left(yaw+\pi\right)\cdot\left(col/2\pi\right) \\ v=\left(fov_{up}\text{-}pitch\right)\cdot\left(row/fov_{up}+fov_{down}\right) \end{cases}. \tag{1}$$

for most 3D lidars, $yaw=\arcsin\left(y/\sqrt{\left(x^2+y^2\right)}\right)\in\left[-\pi,\pi\right]$, $pitch=\arcsin\left(z/\sqrt{\left(x^2+y^2+z^2\right)}\right)\in\left[fov_{down},fov_{up}\right]$, where fov_{down} is the vertical downward field of view angle of the lidar sensor, and fov_{up} is the vertical upward field of view angle of the lidar sensor, *col* is image width and *row* is image height.

To obtain the semantic segmentation of the current projected image, we use the RangeNet++ method with real-time segmentation efficiency. The network is first a down-sampled encoder that can encode contextual information and run faster. Because the information in the vertical direction of the image projected from the point cloud data is determined by the lidar beams, there is very little information in the vertical direction. Therefore, the information in the vertical direction is preserved during downsampling, and downsampling is only performed in the horizontal direction. The RangeNet++ downsampling is followed by a decoder module. Inspired by the network backbone architecture of DarkNet, the extracted features are up-sampled to the original image resolution. The final layer of the network inference process is to use the softmax function to classify point cloud semantics:

$$\hat{y}_c=e^{res_c}\Big/\sum_{c=1}^{C}e^{res_c}, \tag{2}$$

where res_c is the unbounded output corresponding to category c, and C is the number of semantic categories. During the training process, the network uses the stochastic gradient descent algorithm and the weighted cross-entropy loss function L for end-to-end iterative optimization:

$$L=-\sum_{c=1}^{C}w_c y_c\log\left(\hat{y}_c\right), \tag{3}$$

among them, $w_c=1/\log\left(f_c+\varepsilon\right)$, f_c represents the frequency of occurrence of category c.

B. Semantic Dataset Construction

In order to fully evaluate and apply the semantic segmentation network, we train and evaluate the network on the SemanticKITTI dataset and a self-built dataset. The SemanticKITTI dataset is a lidar-based dataset for semantic scene understanding, covering the full 360° field of view of the

Velodyne 64-line lidar. The dataset consists of 22 sequences in total, including 28 semantic classes of different ground, vehicles, vegetation, etc. Different environmental scenes such as urban traffic, residential areas, motorways and country roads around Karlsruhe, Germany are shown. 23201 complete 3D scans of sequence 00-10 were segmented as training set, and 20351 complete 3D scans of sequence 11-21 were used as test set, which is by far the largest publicly available laser semantic dataset.

For the practical application environment of robots, we established and marked the corresponding datasets according to the SemanticKITTI dataset format for training and verification. Using this dataset format, we convert each frame scan of Velodyne lidar data to ******.bin format. At the same time, the traditional SLAM method is used to estimate the pose of each frame scan, which is stored in the poses.txt file. Each frame pose includes a 3 × 3 rotation matrix **R** and a 1 × 3 translation matrix **T**, and the expansion is stored in one line in one-dimensional form, which can be written as:

$$pose = [\mathbf{R} \quad \mathbf{T}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad . \quad (4)$$

$$= [r_{11} \quad r_{12} \quad r_{13} \quad t_1 \quad r_{21} \quad r_{22} \quad r_{23} \quad t_2 \quad r_{31} \quad r_{32} \quad r_{33} \quad t_3]$$

As shown in Fig. 3, our work can not only perform detailed semantic annotation for each frame of point cloud data, but also integrate multiple frames of lidar data into a 3D point cloud map by combining the scanning poses in the poses.txt file. The overall semantic annotation of the data improves the efficiency of annotation and saves time.

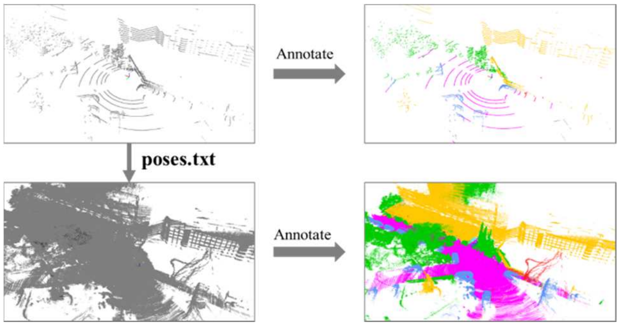


Fig. 3. Sketch map of point cloud semantic annotation

C. *Semantic SLAM*

The main contribution of the semantic SLAM method used in our work is to combine point cloud semantic information and geometric information. The advantages of the two different modal information are complemented to improve the accuracy of pose estimation in robot SLAM system and the practical applicability of 3D point cloud maps. As shown in Fig. 1, we combine semantic information to optimize the SLAM method mainly in two aspects: on the one hand, we combine semantic information to correlate the corresponding features of the current observation scan and the local registration map. The weight of semantic connection is added to each optimized error term, which optimizes the registration objective function and improves the accuracy of point cloud registration; On the other hand, the semantic map is optimized by combining semantic information. Positive moving objects are processed to filter out the influence of highly dynamic objects on traditional SLAM feature matching. The static semantic map established at the end can be more accurately applied to the subsequent intelligent navigation planning. All in all, this paper pursues the accuracy of robot pose estimation and the applicability of semantic maps. The following paper will describe these two aspects in detail.

1. Lidar Odometer

The quality of the point cloud registration objective function is the key to the feature matching of the SLAM system. It determines the accuracy of the final laser odometry estimation, so we combine the point cloud semantic information to optimize the registration objective function in the feature matching process. According to the local characteristics of the point cloud proposed in LOAM [1], we choose to extract the corresponding edge feature points F_e and plane feature points F_p. Before establishing the objective function, we need to find the edge line composed of the corresponding edge feature points and the plane composed of the corresponding plane feature points in the established local map. The current robot pose is estimated by using the distance errors between edge feature points to edge lines and plane feature points to planes as the scan matching objective function. That is, we need to find the laser measurement corresponding to the current feature point set $\{F_e, F_p\}$ in the feature set $\{M_e, M_p\}$ of the local map. The distance error term between the corresponding measurements is taken as the objective function. In this process, we use the semantic feature labels of each point to improve the accuracy and efficiency of feature association correspondence.

使用每个点的语义特征标签来提高特征关联对应的准确性和效率

We use the transformation result of the previous moment or the current pose transformation result assisted by other sensors as the initial transformation to convert the current feature point to the map coordinate system. According to the Euclidean distance, the nearest points are found in the corresponding feature set of the local map as the corresponding measurement point set S, and Principal Component Analysis (PCA) is performed on the point set S.

For the current edge feature point $\mathbf{p}_i = (x, y, z) \in F_e$, the adjacent point set S_e is selected in the local map edge feature point set M_e. We calculate the center coordinates of each point in the point set:

$$[cx \quad cy \quad cz] = \left[\frac{1}{N}\sum_{k=1}^{N} x_k \quad \frac{1}{N}\sum_{k=1}^{N} y_k \quad \frac{1}{N}\sum_{k=1}^{N} z_k \right], \quad (5)$$

where N represents the number of points in the point set S_e. The average error in each direction can be: 去中心化

$$[ax \ ay \ az] = \left[\frac{1}{N}\sum_{k=1}^{N}(x_k - cx) \quad \frac{1}{N}\sum_{k=1}^{N}(y_k - cy) \quad \frac{1}{N}\sum_{k=1}^{N}(z_k - cz) \right], (6)$$

then the covariance matrix of S_e can be expressed as:

$$\mathbf{A}_e = [ax \ ay \ az]^{\mathrm{T}} \times [ax \ ay \ az] = \begin{bmatrix} ax \cdot ax & ax \cdot ay & ax \cdot az \\ ax \cdot ay & ay \cdot ay & ay \cdot az \\ ax \cdot az & ay \cdot az & az \cdot az \end{bmatrix}. (7)$$

We find the eigenvalues and eigenvectors of matrix $\mathbf{A}_e$. Because the point set S_e are all edge feature points corresponding to the current point, its eigenvalues contain one element that is significantly larger than the other two, and the eigenvector corresponding to this eigenvalue represents the direction of the corresponding edge line. To obtain the distance residual from point $\mathbf{p}_i(x,y,z)$ to the corresponding edge line, two nearby points $A=(x_1,y_1,z_1)$ and $B=(x_2,y_2,z_2)$ are arbitrarily selected on the line passing through the center mean point in this direction. The corresponding distance residual is:

$$d_i^e = \frac{\left|\overrightarrow{PA}\times\overrightarrow{PB}\right|}{\left|AB\right|}. \qquad (8)$$

For this residual term, we combine the distance geometric weight and the point cloud semantic weight to optimize its proportion in the residual of the entire system to ensure the accuracy of the optimized registration. In the continuous scanning of lidar, the distance between edge feature points and corresponding edge lines is zero under ideal conditions. Therefore, the distance geometric weight is based on the distance residual as a reference, that is, the smaller the distance, the greater the weight of the current item, which has stronger geometric consistency. Similarly, in successive scans, the corresponding measurements are most likely of the same object. We judge the semantic information identity of the corresponding feature point set S_e and the current feature point $\mathbf{p}_i$ as a reference for the semantic weight of the point cloud. To be consistent with distance geometric weights, we define semantic inconsistency parameters: 在进行线特征匹配时，匹配成功的点和线应该是同一物体

定义了一个语义不一致参数：pi与Se中每个点进行匹配，语义标签一致则为1，否则为0

$$c_i^e = 1-\frac{1}{N}\sum\nolimits_{k=1}^{N}\left(p_k \oplus p_i\right), \qquad (9)$$

where the function $(p_k \oplus p_i)$ is defined to indicate whether the semantics of the kth point in S_e and the current point $\mathbf{p}_i$ are consistent, that is, the semantic information labels of the two points are the same, then the function value is 1. Otherwise it is 0. Like the distance geometry weight, when the semantic inconsistency is smaller, the weight of the current item is larger and has stronger semantic consistency. Then the total weight of the current item can be defined by the following formula:

$$w_i^e = 1-\alpha * d_i^e - \beta * c_i^e, \qquad (10)$$

where α and β represent the geometric weight factor and the semantic weight factor respectively, and $\alpha+\beta=1$. When the distance residual is within the range of $[0,\ 1]$, we consider it as a valid measurement correlation, otherwise it is discarded. Therefore, the total weight $w_i^e \in [0,\ 1]$.

For the current plane feature point $\mathbf{p}_i = (x,y,z) \in F_p$, the adjacent point set S_p is selected in the local map plane feature point set M_p. Because S_p is generally distributed on a plane, its covariance matrix has one eigenvalue that is significantly smaller than the other two. The eigenvector corresponding to the smallest eigenvalue represents the direction of the plane. Then by combining the geometric center points of the point set S_p, a plane equation can be determined:

$$AX + BY + CZ + D = 0. \qquad (11)$$

Calculate the distance from the current plane feature point to this plane as the residual:

$$d_i^p = \frac{\left|Ax+By+Cz+D\right|}{\sqrt{A^2+B^2+C^2}}. \qquad (12)$$

Similar to the residual term of edge feature points, we also set the corresponding weight $w_i^p \in [0,1]$ for the residual term corresponding to the plane feature points.

Finally, the point cloud registration objective function we established is as follows: 目标函数的权重里包含了d，相当于使用了两遍d，感觉有点冗余，没解耦

$$e_{(\mathbf{T})} = \sum_{p_i \in F_e} w_i^e \cdot d_i^e + \sum_{p_i \in F_p} w_i^p \cdot d_i^p, \qquad (13)$$

we then use the Gauss-Newton iteration method to minimize this objective function to find the optimal transformation. The increment of each iteration is $\Delta\mathbf{T}=\left(J^T J\right)^{-1} J^T e$, where J is the Jacobian matrix of the objective function with respect to the pose transformation $\mathbf{T}$, that is, $J=\partial e/\partial\mathbf{T}$. When $\Delta\mathbf{T}$ is small enough or meets the maximum number of iterations, the iteration is stopped to obtain the optimal pose transformation of the current lidar.

2. Filtering of dynamic object

In the process of building a semantic map, we distinguish dynamic objects into highly dynamic objects (such as walking people and moving vehicles) and semi-static objects (such as standing people and stopped vehicles). The roles and processing methods of the two in different function maps are different.

In scan-registered local maps, the method of correlating corresponding measurements with Euclidean distance as a reference generally works well in static environments. However, highly dynamic objects, i.e. moving objects, may lead to incorrect associations between the current scan observations and the local map. Therefore, when the local map is updated, the semantic consistency between the current laser scan and the local map is checked to filter out moving objects. Specifically, we utilize the semantic inconsistency parameter defined in Equation (9) to determine whether the current point belongs to the moving object. When the semantic inconsistency parameter c_i is greater than the set threshold, that is, $c_i > c_{th}$. The point is considered to belong to the moving object and is not updated in the local map. For semi-static objects, they are potentially moving objects. Because these objects are currently stationary and are also valuable features in the lidar scan

在更新局部地图时，检查当前激光扫描与局部地图之间的语义一致性，以过滤掉移动物体。

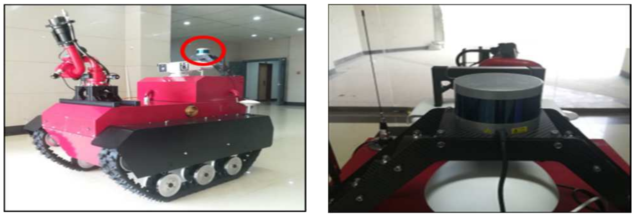
相当于在进行用于定位的局部地图时，保留了半静态物体；
而在最终的3D点云语义地图上，为了导航，将高动态物体和半静态物体全部剔除

matching process, simply removing them directly may result in a scarcity of valuable features. Conversely, keeping as many valuable features as possible can ensure the robustness and accuracy of scan registration.

For the final generated 3D point cloud semantic map, both highly dynamic objects and semi-static objects will be filtered out. Because a static map of the environment is always required in subsequent navigation planning, moving highly dynamic objects or potentially moving semi-static objects can be resolved by the autonomous obstacle avoidance module. Therefore, in this map, we only keep static objects to ensure a complete passable area in subsequent intelligent navigation planning.

III. Experiments

In this paper, the system algorithm is run in the Robot Operating System (ROS), and each module in the system is experimentally tested and verified. We use the SemanticKITTI dataset to conduct experiments to evaluate the performance of the system. At the same time, as shown in Fig. 4, a fire-fighting robot equipped with lidar is used as an experimental test platform to illustrate the applicability of the system proposed in the paper.



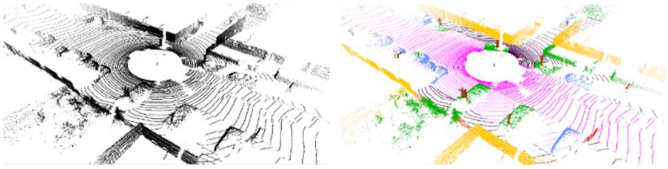
(a) Fire-fighting robot (b) Lidar sensor

Fig. 4. Test platform for practical environment experiments

A. Segmentation test for point cloud semantics

Before performing point cloud segmentation, we perform spherical projection of the original point cloud data, which converts the points in the lidar coordinate system to the image coordinate system. The KITTI dataset uses Velodyne HDL-64E lidar. Referring to its laser beams with 64 lines and the ease of subsequent semantic segmentation network training, We set the corresponding parameters (fov_{up}=2 , fov_{down}=24.8 , $row = 64$, col=1024).

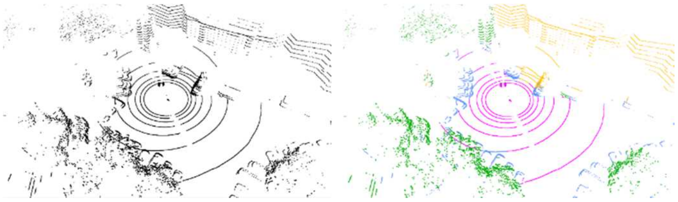
Because our main concern is to use point cloud semantic information to build high-quality semantic maps, laying a certain foundation for subsequent intelligent navigation planning. Therefore, the test result of point cloud semantic segmentation is directly and intuitively displayed in Fig. 5 here. Objects of different semantic categories are distinguished using different colored point clouds.



(a) Original point cloud data (b) Semantic segmentation result

Fig. 5. Semantic segmentation result of Velodyne HDL-64E lidar data

For the campus semantic dataset marked in this paper, we use Velodyne VLP-16 lidar and set the parameters (fov_{up}=2 , fov_{down}=24.8 , $row = 64$, col=1024). The different color point clouds in Fig. 6 show that it also has a good semantic segmentation effect in the campus data.



(a) Original point cloud data (b) Semantic segmentation result

Fig. 6. Semantic segmentation result of Velodyne VLP-16 lidar data

B. Accuracy test of lidar odometry

When the semantic information of the point cloud is obtained, we assign a weight factor to each error term in the point cloud registration objective function. This weighting factor comprehensively considers geometric information and semantic information. As shown in Fig. 7 and Fig. 8, we use the proposed system to perform semantic mapping tests on the KITTI dataset of sequence 10 and the campus data in the University of Science and Technology of China (USTC), respectively. We compared the positioning errors of the two methods that do not consider the weighting factor ($w_i = 1$) and the weighting factor that only considers geometric information ($w_i = 1 - d_i$), to illustrate that the algorithm in this chapter considers both geometry and semantics. As shown in Table I, the system proposed in this paper comprehensively considers the weight factors of geometric and semantic information, which improves the accuracy of lidar odometry.

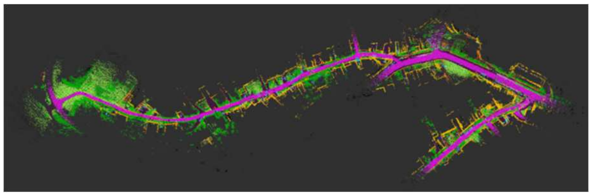


Fig. 7. Semantic map of sequence 10 in the KITTI dataset

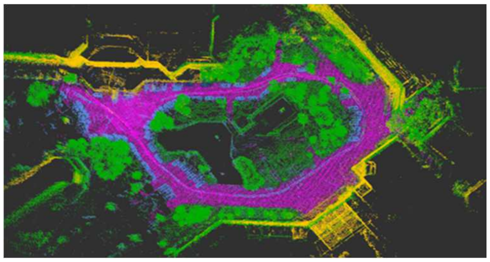


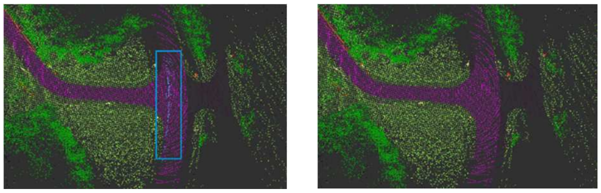
Fig. 8. Semantic map of the USTC campus dataset

TABLE I
Positioning Error (m) of LiDAR Odometry

Dataset	No Weighting Factor	Geometric Weighting Factor	The Proposed System
KITTI dataset	13.46	7.55	3.64
Campus dataset	1.72	1.69	1.46

C. Optimization test for different maps

In this paper, we consider the impact of highly dynamic objects on scan registration and mapping quality during the mapping process. As shown in Fig. 9(a), there are "ghosts" of highly dynamic objects (moving vehicle in blue boxes) in the generated map. The features of this object can cause false matching associations and adversely affect subsequent positioning and traversable area division based on the map. As shown in Fig. 9(b), we remove highly dynamic objects from the generated local map to avoid this adverse effect. Meanwhile, as shown in Fig. 10, we keep semi-static objects (vehicles parked on the side of the road) to preserve valuable features.



(a) Before filtering (b) After filtering

Fig. 9. Filtering of highly dynamic objects

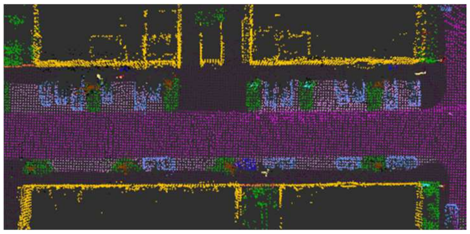


Fig. 10. Vehicles parked on the side of the road

Finally, we consider the need for intelligent autonomous navigation planning for robots and build a fully static environment semantic map. As shown in Fig. 11, compared to Fig. 8, the parked vehicles indicated in blue in the map are filtered out. We try to avoid the need for manual division of passable areas as much as possible. Moreover, the robot can distinguish different road conditions according to the current semantic information and adopt different navigation planning decisions.

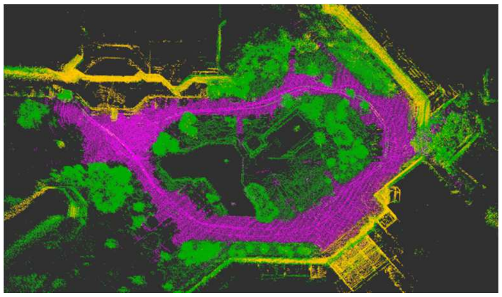


Fig. 11. Semantic map of fully static environment

IV. Conclusion

Conclusion In this paper, full semantic segmentation of 3D lidar point cloud data is performed using a fully convolutional neural network. By combining with the acquired semantic information, high-quality semantic maps are established. First, to ensure the efficiency of point cloud segmentation, we perform spherical projection to convert the disordered and complex point cloud into a two-dimensional image. The image is segmented using a semantic segmentation network to obtain 3D point cloud semantic information. Secondly, by combining with semantic information, we optimize the objective function of point cloud registration in laser odometry and filter out outliers caused by highly dynamic objects. Experimental results show that the proposed semantic SLAM system obtains high-precision laser odometry. The established fully static semantic map ensures a complete traversable area for the subsequent navigation the mobile robot.

Acknowledgment

Our research work was supported by the Project of State Grid Anhui Electric Power Company Limited with Grant No. 52120320006V.

References

[1] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," *Robotics: Science and Systems*, vol. 2, pp. 9, 2014.

[2] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-optimized Lidar Odometry and Mapping on Variable Terrain," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758-4765, 2018.

[3] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart and C. Cadena, "SegMap: 3d segment mapping using data-driven descriptors," *Robotics: Science and Systems (RSS)*, 2018.

[4] D. Yoon, T. Tang and T. Barfoot, "Mapless Online Detection of Dynamic Objects in 3D Lidar," *2019 16th Conference on Computer and Robot Vision (CRV)*, 2019, pp. 113-120.

[5] J. Schauer and A. Nüchter, "The Peopleremover—Removing Dynamic Objects From 3-D Point Cloud Data by Traversing a Voxel Occupancy Grid," in *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1679-1686, July 2018.

[6] G. Kim and A. Kim, "Remove, then Revert: Static Point cloud Map Construction using Multiresolution Range Images," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10758-10765.

[7] H. Lim, S. Hwang and H. Myung, "ERASOR: Egocentric Ratio of Pseudo Occupancy-Based Dynamic Object Removal for Static 3D Point Cloud Map Building," in *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2272-2279, April 2021.

[8] P. Pfreundschuh, H. F. C. Hendrikx, V. Reijgwart, R. Dubé, R. Siegwart and A. Cramariuc, "Dynamic Object Aware LiDAR SLAM based on Automatic Generation of Training Data," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11641-11647.

[9] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser and M. Nießner, "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2432-2443.

[10] T. Hackel, N. Savinov, L. Ladicky, J. Wegner, K. Schindler and M. Pollefeys, "Semantic3d. net: A new large-scale point cloud classification benchmark," *arxiv*, vol. abs/1704.03847, 2017.

[11] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9296-9306.

[12] A. Milioto, I. Vizzo, J. Behley and C. Stachniss, "RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4213-4220.

[13] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley and C. Stachniss, "SuMa++: Efficient LiDAR-based Semantic SLAM," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4530-4537.