An Automatic Mapping Method of Navigation Map for Outdoor Road Scenes

Xiaoyu Guo [123], Zhizhong Kang [123] , Chenming Ye [123], Xiaoran Wang [123]

[1] School of Land Science and Technology, China University of Geosciences, Xueyuan Road, Beijing, 100083 CN
[2] Research Center of Lunar and Planetary Remote Sensing Exploration, China University of Geosciences (Beijing) , No. 29 Xueyuan Road, Haidian District, Beijing
[3] Subcenter of International Cooperation and Research on Lunar and Planetary Exploration, Center of Space Exploration, Ministry of Education of The People’s Republic of China, No. 29 Xueyuan Road, Haidian District, Beijing 100083 China - guoxiaoyu@email.cugb.edu.cn, zzkang@cugb.edu.cn, yechenming@email.cugb.edu.cn, wangxiaoran@email.cugb.edu.cn.

Keywords: Automatic Mapping, Semantic Segmentation, SLAM, Vector Navigation Map, Lightweight Map, Lanelet2.

Abstract

In the process of autonomous navigation of outdoor mobile robots, four modules are involved: perception, localization, planning, and controlling. The perception module utilizes sensors such as cameras and radars based on various principles to analyze the robot's surroundings in real-time. The localization module uses GPS (Global Positioning System), IMU (Inertial Measurement Unit), and prior maps for real-time positioning analysis. The planning module plans optimal path based on the outputs of the first two modules, and the controlling module directs the robot's chassis to move along the planned optimal path. For the localization module, the accuracy of GPS positioning results heavily depends on weather conditions and GPS signal receptions. Even if the positioning results of imu are integrated, the positioning accuracy still cannot meet the needs of robot navigation. Therefore, using prior maps for repositioning can compensate for this accuracy deficiency. The planning module also requires path planning based on prior maps. If the a priori map storage is large, it will lead to difficulties in usage, maintenance, and updates. Therefore, it is crucial to research lightweight navigation map mapping methods. In this paper, an automatically mapping method of lightweight navigation maps is proposed, combining cameras and LiDAR (Light Detection and Ranging), including semantic informations necessary for outdoor navigation positioning, such as pole-like objects and traffic signs for robot longitudinal positioning, and lane line elements for robot lateral positioning. This method automatic generates robot navigation maps in Lanelet2 format, providing support for subsequent positioning and path planning modules.

1. Introduction

With the development of three-dimensional scene reconstruction and deep learning technologies, research combining map construction and semantic segmentation is becoming more prevalent. Various devices are primarily used for three-dimensional reconstruction, including monocular cameras, binocular cameras, RGB-D depth cameras, and LiDAR. Semantic segmentation tasks have made significant progress through convolutional neural network-based methods, with various structures continuously enhancing the segmentation accuracy.

Li et al. (Li et al., 2017) integrated LSD-SLAM framework with convolutional neural networks, selected keyframes for deep learning, and achieved semantic segmentation. Keisuke Tateno et al. (Tateno et al., 2017) utilized monocular vision to estimate absolute scale and used convolutional neural networks to generate dense semantic labels. Yi Yang et al. (Yang et al., 2018) used stereo vision to generate depth information, refining the results of semantic segmentation. Additionally, they removed moving objects by combining semantic labels, resulting in improved accuracy. Unlike binocular cameras, RGB-D cameras can directly generate depth information, avoiding stereo matching computations in SLAM solutions, presenting an advantage. Zhe Zhao et al. (Zhao & Chen, 2016) fused depth images obtained from RGB-D cameras with semantic labels to construct three-dimensional semantic maps of indoor scenes. While RGB-D cameras can directly receive depth information, they are sensitive to ambient light and are only suitable for small-scale environments such as indoors, unable to handle large-scale outdoor environments. These SLAM (Simultaneous Localization and Mapping) methods are vision-based, reconstructing three-dimensional models of the environment through stereo image pairs. In these methods, errors arise during three-dimensional reconstruction for creating the initial map of the environment, impacting the accuracy of the final semantic map and increasing computational resource consumption. LiDAR SLAM, by directly utilizing LiDAR point clouds for environmental mapping, can avoid these shortcomings.

As the application scope of mobile robots continues to expand, the range of movement for outdoor mobile robots is also increasing. Therefore, the issues of the storage volume of prior maps and the computational load during localization cannot be ignored. Additionally, the maintenance of maps is also crucial. In practical applications, it is not feasible to frequently reconstruct maps as changes in roadside building appearances or seasonal variations in tree states can affect map usage. Overly dense and detailed maps can actually increase the difficulty of localization. Hence, simplifying map elements and storage formats to construct lightweight semantic maps, and then utilizing lightweight semantic map relocation methods, is a preferred solution for many road environment SLAM modules (Zhao et al., 2019).

Schreiber et al. (Schreiber et al., 2013) proposed using lane markings as localization clues. To achieve this, they manually labeled lane markings on LiDAR intensity maps, then used stereo cameras to detect lane markings online and match them with lane markings in the map. Welzel et al. (Welzel et al., 2015) and Qu et al. (Qu et al., 2015) used traffic signs to assist image-based vehicle positioning. Inspired by this, Ma et al. (Ma et al., 2019) proposed a lightweight localization method that does not require detailed knowledge about the world's appearance (e.g., dense geometric structures or textures). Instead, it utilizes vehicle

dynamics and a semantic map containing lane markings and traffic sign. Traffic signs provide longitudinal information, while lanes help avoid lateral drift. Experiments conducted over a test drive of over 300 kilometres showed that this method achieved a lateral accuracy of 0.05 meters and longitudinal accuracy of 1.12 meters.

Jie Jin et al. (Jin et al., 2018) conducted similar research, utilizing LiDAR data and its semantic extraction results, along with GPS poses, to construct semantic maps. They used GPS as an initial value to determine the search radius and directly searched for the positioning result within the search range based on the 3D landmark's reprojection result on images. Due to the potential mismatch of lane markings and traffic lights, and relatively sparse features, the method primarily used 2D-3D matching of traffic signs during initialization, which could be achieved after multiple frames due to the sparse distribution of traffic signs.

Relocation algorithms based on prior semantic maps have been well-validated in underground garages, industrial parks, and small-scale road scenes, making them an important module in mobile robot research. Prior maps provide bounded error assurance for the localization system, serving as a reliable way to improve localization accuracy. Lightweight semantic maps address the storage and maintenance challenges of large-scale high-precision maps. The lightweight characteristics of navigation prior maps can be reflected in two aspects: 1. Lightweight elements, selecting a small number of elements with obvious geometric or texture feature. For this study, they are poles and traffic signs, and lane line elements. In this way, when using a prior map for relocation, only real-time extraction of these distinctive features is needed, and then matching them with the prior map can achieve relocation, avoiding positioning errors caused by unclear geographical features leading to real-time extraction errors. At the same time, due to the reduced number of map elements, the computational complexity of feature matching during positioning is reduced. 2. Lightweight storage format, using the simplest vector format to store geographic features, simplifying the complex world into points, line strings, and polygons, reducing the storage capacity of prior maps.

2. Method

In this paper, a road scene automatic mapping method is proposed, combining cameras and LiDAR. The image is fed into a lightweight semantic segmentation network BiseNETv2 (Bilateral Segmentation Network) and SCNN (Spatial CNN) to provide 2D semantic information. Through the calibration of cameras and LiDAR, the 2D semantic information is mapped onto a 3D point cloud. The point cloud is used to construct a 3D point cloud of the scene through LiDAR SLAM, resulting in a 3D semantic point cloud map of the scene. In the process of SLAM, considering the temporal and spatial character of point cloud semantic information, the optimization of the current frame point cloud semantic labels is achieved by extracting point clouds from previous and subsequent frames and calculating the normals of points. After obtaining the 3D semantic point cloud map of the scene, interested elements are finely extracted through denoising, clustering, fitting, and other methods. Finally, the 3D semantic point cloud of the scene is stored as a vector navigation feature map using the Lanelet2 map format. The framework of the entire system is depicted in Figure 1.

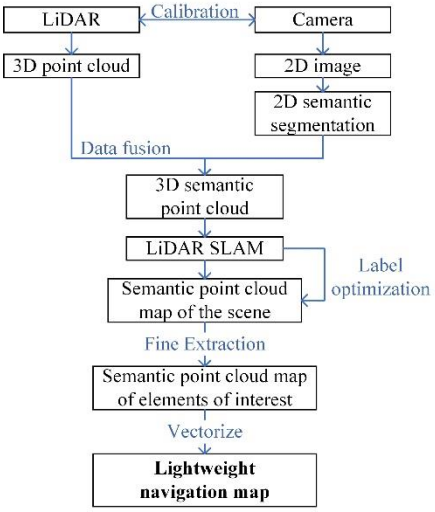


Figure 1. Framework of the automatic mapping method.

2.1 The extraction of semantic information in outdoor scenes

2.1.1 2D image semantic segmentation: For real-time mapping, real-time semantic segmentation of scenes is required. Thus demanding an efficient semantic segmentation network. To balance accuracy and efficiency, this paper uses the BiseNETv2 (Yu et al., 2021) with a dual-channel network structure to perform semantic segmentation on scene images. The network consists of three main components: Detail branch, Semantic branch, and Feature Fusion module. The Detail branch has fewer network layers but larger feature map sizes, suitable for extracting spatial detail information; the Semantic branch is relatively deeper with more down sampling operations, resulting in smaller feature map sizes and larger receptive fields, suitable for extracting high-level semantic information; the Feature Fusion module is used to merge the features of the Detail and Semantic branches to fully utilize the information from both branches. The network is trained using the cityscapes (Cordts et al., 2016), a publicly available road scene semantic segmentation dataset, which includes fine labels for 19 classes of road scene basic elements.

To provide lateral constraints for navigation positioning, this paper also uses the SCNN (Pan et al., 2018) semantic segmentation network to extract lane line elements. In SCNN, the feature map obtained from the backbone network is sliced and convoluted in four directions (up, down, left, and right) to enhance spatial information fusion, and facilitate the segmentation tasks of elongated objects like lane line. Compared to the redundant data issue caused by the transmission of feature pixel information from various directions in traditional networks, the SCNN model transmits information in a sequential manner, where each pixel is transmitted to the next layer pixel either by row or by column, thereby saving significant computational time. The SCNN network is trained using CUlane dataset (Pan et al., 2018), a widely used dataset for lane line detection.

2.1.2 3D semantic point cloud generation： The mapping of 2D images and 3D point clouds in the KITTI dataset involves each data sample containing an RGB image, a binary file of laser point clouds, and a calibration file. The calibration file is presented in floating-point form with seven rows of floating-point sequences, each representing a calibration matrix: $P_{\text{rect}}^{(0)}$, $P_{\text{rect}}^{(1)}$, $P_{\text{rect}}^{(2)}$, $P_{\text{rect}}^{(3)}$, $R_{\text{rect}}^{(0)}$, $T_{\text{velo}}^{\text{cam}}$ 和 $T_{\text{imu}}^{\text{velo}}$ .Since four cameras

were used during data collection in the KITTI dataset, 0 for the left grayscale camera, 1 for the right grayscale camera, 2 for the left color camera, and 3 for the right color camera. The $P_{\text{rect}}^{(1)}$ matrix represents the intrinsic matrix of camera i multiplied by the extrinsic matrix from camera 0 to camera i, $R_{\text{rect}}^{(0)}$ is the rectifying rotation matrix of camera 0, and $T_{\text{velo}}^{\text{cam}}$ is the extrinsic matrix from LiDAR to camera 0. This study utilizes the color monocular camera 2 and, following the KITTI dataset paper (Geiger et al., 2013) the 2D-3D mapping relationship is given by the equation:

$$y_{3\times1} = P_{\text{rect}}^{(2)} \cdot R_{\text{rect}}^{(0)} \cdot T_{\text{velo}}^{\text{cam}} \cdot x_{4\times1} \tag{1}$$

where $y_{3\times1} = \{u, v, 1\}^T$ represents the transformed RGB image pixel coordinates and $x_{4\times1} = \{x, y, z, 1\}^T$ represents the coordinates of the 3D point cloud.

The entire projection process can be summarized as follows: projecting the points from the LiDAR coordinate system into the camera 0 coordinate system, performing projection correction for camera 0, projecting the points from the camera 0 coordinate system into the camera 2 coordinate system, and finally, based on the intrinsic parameters of camera 2, projecting the points into the pixel coordinate system of camera 2, thereby completing the mapping of 2D images and 3D point clouds, resulting in a semantic 3D point cloud.

2.2 The construction of scene semantic point cloud map and vector navigation map

2.2.1 LiDAR SLAM: This paper uses LeGO-LOAM (Shan & Englot, 2018) and the each frame of semantic point cloud obtained in Section 1 to construct the scene and obtain the semantic point cloud of the scene. Initially, the semantic point cloud is divided into ground points and non-ground points. Then, calculating the curvature of points using formula 2 and extracting feature plane points and feature corner points by sorting curvature.

$$c = \frac{1}{|s| \cdot \| X_{(k,i)}^L \|} \left\| \sum_{j \in S, j \neq i} \left(X_{(k,i)}^L - X_{(k,j)}^L \right) \right\| \tag{2}$$

where $X_{(k,i)}^L$ is the coordinate of the point p_i , $X_{(k,j)}^L$ is the coordinate of p_i's adjacent point and s is the number of point p_i's adjacent point points excluding the point p_i.

Feature matching is then carried out to find the corresponding feature plane and feature line in previous frame for each feature plane and feature corner point in the current frame. Minimizing the distance between feature points and feature lines and planes. This process yields the coordinate changes between the two frames, providing an estimate of the current frame's pose. Finally, the fused odometry matches the keyframe point cloud to generate a global point cloud, conducts scan-map loop closure detection, optimizes the generated map, and outputs the final pose estimation of each frame and the semantic point cloud of the scene.

2.2.2 Label optimization based on semantic information: At this moment, every frame of semantic point cloud in the semantic point cloud map is obtained by 2D semantic segmentation and 3D point cloud mapping, where errors may exist leading to incorrect semantic labels of point cloud. Considering that the semantic information of points is temporal, it is possible to optimize the semantic labels of the current frame by using the semantic labels of the point clouds from the previous and subsequent frames. The semantic information of point cloud also has spatial characteristics, where points that are close in distance and have similar normal vectors tend to have the same semantic label. Therefore, in this study, the semantic labels of each key frame point cloud were optimized by combining the temporal and spatial characteristics of semantic labels in the thread of publishing maps in LeGO-LOAM. The original thread of publishing maps publishes local maps consisting of key frames within a certain range relative to the current frame based on the pose information of key frame point cloud. This thread runs at a specific frequency, and ultimately, several local maps constitute the global map of the scene.

In this map publishing thread, this study optimized the labels for each current key frame point cloud $F^p = \{F_1^P, F_2^P \dots F_i^P\}$ that constitute the local map. Firstly, based on the key frame estimate pose $F^e = \{F_1^e, F_2^e \dots F_i^e\}$, the point clouds of the current key frame F_i^P and the 3 frames before and after it were obtained to form neighboring point clouds F_i^n. Then, the voxel grid of the neighboring point cloud F_i^n was partitioned with a voxel size of 0.2, the label in each voxel was counted, and the label that appeared most frequently in a voxel represented that grid. Finally, the label of each point in the current key frame was determined based on the voxel it belongs to. Using this method, temporal optimization was achieved for the current key frame F_i^P , followed by spatial optimization of the frame's point cloud. Specifically, the method involved computing the 10 nearest neighbors of each point, comparing the normal vectors of each neighboring point with the point in question, counting the labels of neighboring point with similar normal vectors, and assigning the most frequent label to the point. This method achieved spatial optimization of the labels.

2.2.3 The construction of vector navigation map: After obtaining the semantic point cloud map of the scene through SLAM, it is necessary to extract the elements of interest, such as lane lines, pole-like objects, and traffic signs required for creating a vector navigation location map, based on the semantic labels. Subsequently, through denoising, clustering, and line fitting operations, the mapping elements mentioned above were finely extracted to eliminate noise points and erroneous points resulting from inaccuracies in 2D image semantic segmentation and 3D point cloud mapping. Specifically, the entire point cloud representing the elements of interest was denoised, pole-like objects and traffic signs were subjected to Euclidean clustering, and lane lines were fitted using RANSAC for line fitting.

Upon completion of the fine extraction of the elements of interest, the Lanelet2 map format (Poggenhans et al., 2018) was used to construct the vector navigation location map. The Lanelet2 map is a 2D vector map, consisting of basic units, including points, line strings, and polygons. Line strings and areas are composed of points, while polygons are formed by a line string. In this study, lane lines, traffic signs, and pole-like objects were added as elements in the navigation feature map. Lane lines and traffic signs were stored using line strings, while pole-like objects were stored as points.

3. Experimental Result

The KITTI dataset is the largest computer vision evaluation dataset in the field of autonomous driving, collected from urban, rural, and highway scenes. The data collection platform of the KITTI dataset consists of 2 grayscale cameras, 2 color cameras,

a 3D LiDAR-64, and a GPS navigation system. This experiment uses the data package from the 2011_09_26_drive_0059 sequence in the KITTI dataset, which includes color images, grayscale images, 64-line LiDAR point cloud, sensor calibration files, and more. The duration of this sequence is 37 seconds, containing 379 frames of data, providing extensive information about cars, people, buildings, and more. In this experiment, the segmentation model is trained using the Cityscapes dataset. The Cityscapes dataset includes 2975 street scene images with dense labels for 19 object classes, such as roads, sidewalks, pedestrians, vehicles, buildings, and traffic lights.

Table 1 shows the IOU and MIOU of 2D image semantic segmentation of 19 classes of road scene. Figure 2 display the results of 2D semantic segmentation and 3D point cloud mapping. Figure a represents the original image, while figure b shows the semantic segmentation of 19 classes of road scene elements. Figure c displays the lane line semantic segmentation, and figure d illustrates the 3D semantic point cloud obtained by mapping the 2D semantic segmentation results onto the 3D point cloud.

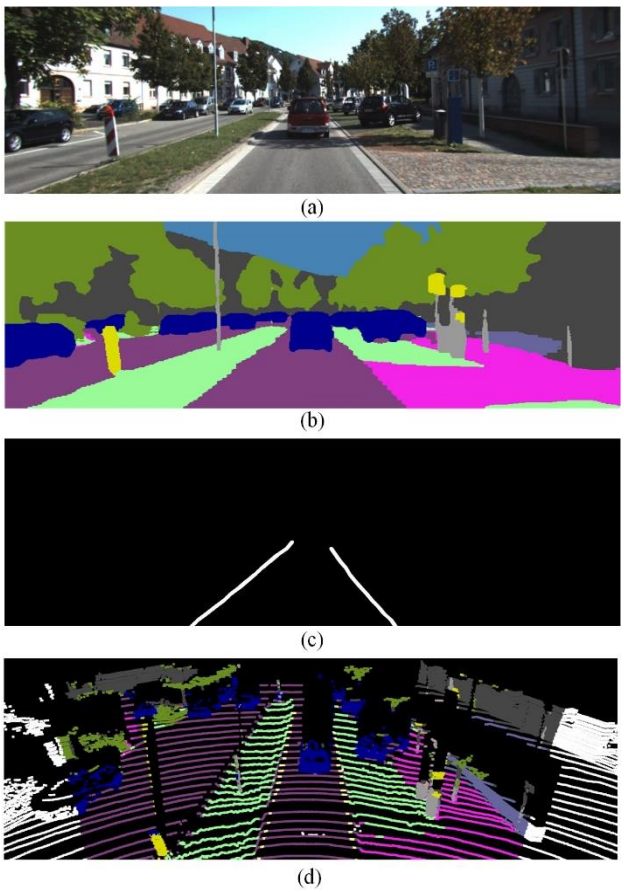


Figure 2. Result of 2D and 3D semantic segmentation.

Num.	Class	IOU	Num.	Class	IOU
0	Road	0.979	10	Sky	0.945
1	Sidewalk	0.83	11	Person	0.808
2	Building	0.918	12	Rider	0.579
3	Wall	0.465	13	Car	0.945
4	Fence	0.56	14	Truck	0.689
5	Pole	0.617	15	Bus	0.787
6	Traffic light	0.687	16	Train	0.75
7	Traffic sign	0.773	17	Motorcycle	0.588
8	Vegetation	0.92	18	Bicycle	0.768
9	Terrain	0.63			
MIOU: 0.75					

Tabel 1. Results of 2D semantic segmentation.

Figure 3 demonstrate the scene semantic point cloud map obtained by inputting a single-frame 3D semantic point cloud into the LeGO-LOAM laser SLAM system, presenting two different perspectives of the scene semantic point cloud map. The "2011_09_26_drive_0059" sequence in the KITTI dataset depicts a straight road segment with a lane, featuring grassland, trees, stationary vehicles, pole-like objects, traffic signs, buildings, and other prominent elements along the road. Furthermore, the processing speed of semantic segmentation is around 200ms per frame, and the frequency of semantic point cloud input into the SLAM system is 5Hz, meeting the real-time requirements of the SLAM system for point cloud input.

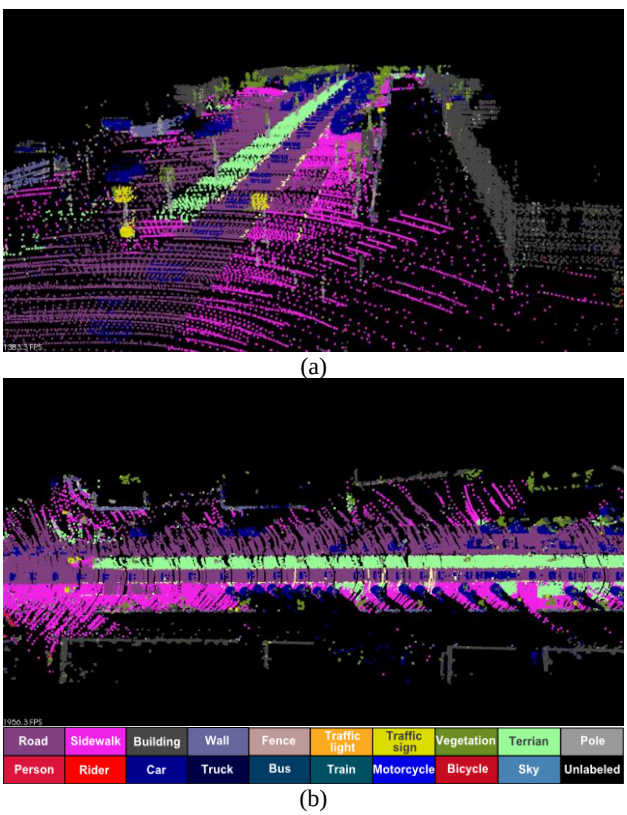
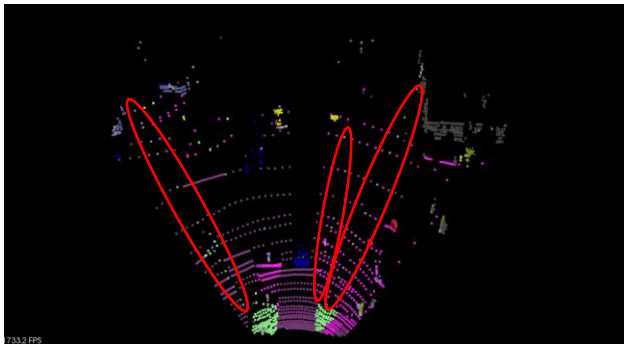
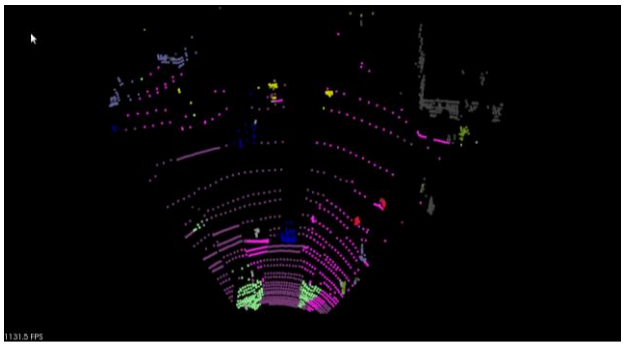


Figure 3. Semantic point cloud map of the scene from two perspectives.

Figure 4 reveal the optimized labeling results of a single frame point cloud, indicating misclassified points from the 2D image semantic segmentation and 3D point cloud mapping in gray within the red markers. By considering the temporal and spatial characteristics of point cloud semantic labels, misclassified points are corrected through semantic label optimization.



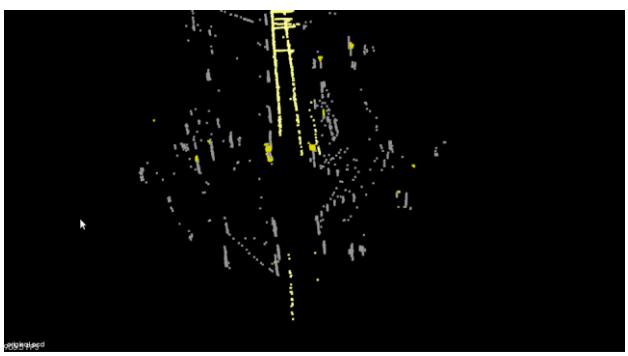
(a)



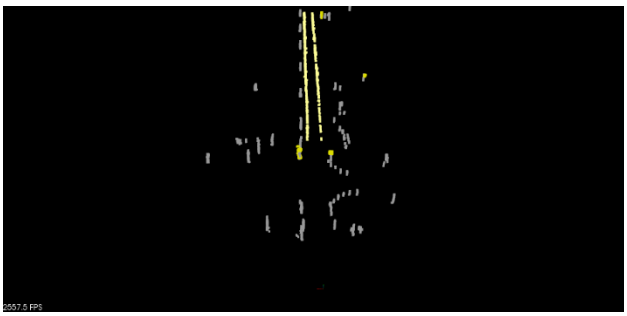
(b)

Figure 4. Single frame semantic point cloud before and after label optimization

The results displayed in Figure 5 show the refined extraction of navigation elements, the gray points represent pole-like objects, the yellow points represent traffic signs, and the light yellow points represent lane lines. Parameters set as follows: the number of neighboring points searched during denoising is 30, the outlier threshold is 5; the Euclidean clustering tolerance is 0.5, the minimum cluster point number is 3, and the maximum cluster point number is 200; the distance threshold for RANSAC line fitting is 0.3. The result image demonstrates that through refined extraction operations, most of the noise points have been removed, making the pole-like objects next to the lane lines clearly visible and guiding poles for right turns on the road also clearly visible. Additionally, the misclassified lane lines point cloud have been filtered out through line fitting.



(a)



(b)

Figure 5. semantic point cloud map of elements of interest before and after fine extraction.

Figure 6 depicts a vector map of the road drawn using the online mapping tool (Autoware Vector Map Builder), generating a vector navigation element map containing lane lines, pole-like objects, and traffic signs. This map is a 2D map where lane lines and traffic signs are stored as line strings, shown in blue in the image, while pole-like objects are stored as points, shown in yellow in the image.

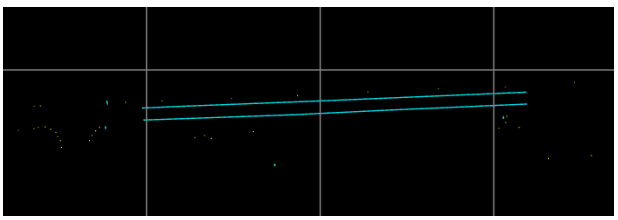


Figure 6. lightweight navigation map of the scene.

4. Conclusion

For the purpose of facilitating the storage, maintenance, and updating of robot prior navigation maps, this study proposes the concept of lightweight maps. "Lightweight maps" are mainly reflected in two aspects. The first aspect is to select a small number of map elements, which are elements with clear geometric or texture features, which also need to serve positioning and navigation. In this study, lane lines, pole-like objects, and traffic signs are selected as map elements. The second aspect is to use the most simplified vector format to store the prior map, thereby reducing the storage of maps. Therefore, this study proposes an automatically mapping method of lightweight navigation maps, combining data from cameras and LiDAR. Through experiments, this method successfully obtain a vector navigation map of outdoor scenes, including the three elements required for the subsequent navigation and positioning. Traffic signs and pole-like objects can provide longitudinal information, while line lanes help avoid lateral drift.

References

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. *Proceedings of the IEEE conference on computer vision and pattern recognition.*

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, *32*(11), 1231-1237.

Jin, J., Zhu, X., Jiang, Y., & Du, Z. (2018). Localization based on semantic map and visual inertial odometry. *2018 24th International Conference on Pattern Recognition (ICPR).*

Li, X., Ao, H., Belaroussi, R., & Gruyer, D. (2017). Fast semi-dense 3D semantic mapping with monocular visual SLAM. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC).*

Ma, W.-C., Tartavull, I., Bârsan, I. A., Wang, S., Bai, M., Mattyus, G., Homayounfar, N., Lakshmikanth, S. K., Pokrovsky, A., & Urtasun, R. (2019). Exploiting sparse semantic HD maps for self-driving vehicle localization. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

Pan, X., Shi, J., Luo, P., Wang, X., & Tang, X. (2018). Spatial as deep: Spatial cnn for traffic scene understanding. *Proceedings of the AAAI conference on artificial intelligence*.

Poggenhans, F., Pauls, J.-H., Janosovits, J., Orf, S., Naumann, M., Kuhnt, F., & Mayr, M. (2018). Lanelet2: A high-definition map framework for the future of automated driving. *2018 21st international conference on intelligent transportation systems (ITSC).*

Qu, X., Soheilian, B., & Paparoditis, N. (2015). Vehicle localization using mono-camera and geo-referenced traffic signs. *2015 IEEE Intelligent Vehicles Symposium (IV).*

Schreiber, M., Knöppel, C., & Franke, U. (2013). Laneloc: Lane marking based localization using highly accurate maps. *2013 IEEE Intelligent Vehicles Symposium (IV).*

Shan, T., & Englot, B. (2018). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

Tateno, K., Tombari, F., Laina, I., & Navab, N. (2017). Cnn-slam: Real-time dense monocular slam with learned depth prediction. *Proceedings of the IEEE conference on computer vision and pattern recognition.*

Welzel, A., Reisdorf, P., & Wanielik, G. (2015). Improving urban vehicle localization with traffic sign recognition. *2015 IEEE 18th International Conference on Intelligent Transportation Systems.*

Yang, Y., Qiu, F., Li, H., Zhang, L., Wang, M.-L., & Fu, M.-Y. (2018). Large-scale 3D semantic mapping using stereo vision. *International Journal of Automation and Computing, 15*(2), 194-206.

Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., & Sang, N. (2021). Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision, 129*, 3051-3068.

Zhao, Z., & Chen, X. (2016). Building 3D semantic maps for mobile robots using RGB-D camera. *Intelligent Service Robotics, 9*, 297-309.

Zhao, Z., Mao, Y., Ding, Y., Ren, P., & Zheng, N. (2019). Visual-based semantic SLAM with landmarks for large-scale outdoor environment. *2019 2nd China symposium on cognitive computing and hybrid intelligence (CCHI).*