Semantic SLAM for Mobile Robot with Human-in-the-Loop

Zhenchao Ouyang[1,2(✉)], Changjie Zhang[2], and Jiahe Cui[1,2]

[1] Beihang Hangzhou Innovation Institute Yuhang, Beihang University, Yuhang, Hangzhou 311100, Zhejiang, China
ouyangkid@buaa.edu.cn

[2] School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Abstract. Mobile robots are an important participant in today's modern life, and have huge commercial application prospects in the fields of unmanned security inspection, logistics, express delivery, cleaning and medical disinfection. Since LiDAR is not affected by ambient light and can operate in a dark environment, localization and navigation based on LiDAR point clouds have become one of the basic modules of mobile robots. However, compared with traditional binocular vision images, the sparse, disordered and noisy point cloud poses a challenge to efficient and stable feature extraction. This makes the LiDAR-based SLAM have more significant cumulative errors, and poor consistency of the final map, which affects tasks such as positioning based on the prior point cloud map. In order to alleviate the above problems and improve the positioning accuracy, a semantic SLAM with human-in-the-loop is proposed. First, the interactive SLAM is introduced to optimize the point cloud pose to obtain a highly consistent point cloud map; then the point cloud segmentation model is trained by artificial semantic annotation to obtain the semantic information of a single frame of point cloud; finally, the positioning accuracy is optimized based on the point cloud semantics. The proposed system is validated on the local platform in an underground garage, without involving GPS or expensive measuring equipment.

Keywords: Semantic SLAM · Robot · Point cloud segmentation · Human-in-the-loop · Interactive SLAM

1 Introduction

With the rapid development of technology, the related applications of mobile robots are gradually entering daily life from the research stage. Especially during the fight against the COVID-19, mobile robots have undertaken a series of tasks such as material distribution, disinfection and sterilization, service, and cleaning. At the same time, as China enters an aging society, the birth rate continues to decline, the demographic dividend disappears, and the labor shortage will

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 461, pp. 289–305, 2022.
https://doi.org/10.1007/978-3-031-24386-8_16

become increasingly significant [1]. The demand for unmanned operations in society and the market continues to rise, and mobile robots [2] will become one of the important guarantees to fill the workforce gap.

Autonomous environment perception and self-positioning are the basic functions of mobile robots. Robots need to complete the acquisition of their own poses before they can carry out more complex mobile tasks. The current mainstream autonomous positioning of robots includes Simultaneous Localization and Mapping (SLAM) [3], odometer of wheel speed encoder [4], Ultra-Wide-Band (UWB) [5], GPS [6], and multi-sensor fusion solutions [7]. Among them, GPS cannot locate targets in occluded environments or indoors; UWB requires construction on the environment; wheel speedometers are easily affected by slippage and wear, and dynamic modeling of the robot chassis is required, and the estimation accuracy of the rotational pose is also poor. The SLAM algorithm based on open-loop control is less dependent on equipment, and has a wider range of stability and applicable scenarios.

Based on environmental perception sensors and data structures, existing SLAM algorithms can be roughly divided into two categories: vision [8] and LiDAR-based SLAM. LiDAR based on active perception is not affected by ambient light and can perceive in a dark environment, which can provide reliable guarantees for mobile robots. LiDAR-based SLAM technology [9] can provide mobile robots with localization and stable environment-dense mapping information, which is the key module of robot mobility. However, the sparseness and randomness of LiDAR point clouds make the registration features much lower than stereo-based images [10], thus introducing more significant cumulative errors, resulting in the degradation of the point cloud map and poor consistency of the final map. This in turn affects prior map-based localization and a range of downstream tasks.

Although a series of optimization methods have been carried out for the front-end and back-end of the SLAM system, the existing algorithms are still not comparable to the human perception ability. One main reason is that the current SLAM algorithm lacks the ability to understand semantic information of the environment–human cognition. Existing systems [11] try to optimize SLAM systems by introducing environmental semantic information to simulate primary human-like cognition, which is the recently developed semantic SLAM. However, due to the scarcity of point cloud-based semantic segmentation datasets for large-scale scenes, the development of deep learning models has just begun. Annotation of sparse point cloud data, especially for low-cost LiDAR is also extremely challenging.

To solve the above problems, a semantic SLAM with human-in-the-loop is proposed in this paper, we focus on improving mobile robot localization with only low cost LiDAR for the indoor environment. The system takes human collaboration into consideration from the following two aspects: 1) we first introduce the interactive SLAM [12] to refine normal SLAM results, and generate corrected pose based on global point cloud map. 2) Then the point clouds with high global consistency are manually labeled with target-level semantics. By labeling the overlapped map [13] instead of a single point cloud frame, not only the labeling efficiency is greatly

improved, but fewer errors are introduced. 3) The point cloud segmentation model [14] is trained for the extraction of semantic labels, and 4) a novel semantic prior map-based localization method is proposed. The algorithm utilizes point cloud semantic label information to optimize the global map search and local pose estimation of the localization process, and is validated in a large local underground parking garage.

2 Related Works

Considering we only focus on SLAM with semantic information, this section briefly summarized the recent development of semantic visual and LiDAR SLAM.

2.1 Visual Semantic SLAM

Zhang et al. [15] presented a semantic SLAM system for RGB-D cameras under the ORB-SLAM2 [16] framework. The YOLO [17] is introduced as an obstacle detector to extract object-level features in the scene. With this operation, unstable features belonging to moving objects are removed, and the localization accuracy is improved. They also use the fast line rasterization algorithm to speed up the construction of Octomap. However, Yolo can only provide bounding box (BBox)-level detection accuracy, especially when the irregular target is close to the lens, the detection frame will contain a lot of background information. This means that the features in the background will also be eliminated, resulting in the failure of inter-frame registration. Wang et al. [18] use depth map-based flood filling to extract the contour of objects, and acquire highly precise semantic segmentation results.

Kang et al. [19] tried to reduce the error introduced by the BBox while providing the 3D space information of the targets with a robust edge detector. The edge detector divides indoor objects into two wrappers-cuboid or cylinder, and uses 2D-3D transformation to generate the object into 3D landmarks for later usage. But their work only considered a simple indoor environment with limited targets.

PSPNet-SLAM [20] is another improved version of ORB-SLAM2 framework. In this system, the image segmentation-based pyramid-structured PSPNet [21] is used to get a segmentation mask instead of a bounding box for each object. The masks of moving objects can effectively reduce the background introduced by the bounding boxes, thereby increasing the registration features to improve the overall accuracy and system robustness. Zhao et al. [22] follow the same workflow of PSPNet-SLAM while adding the GPS and landmarks from google map to enable the system to be used in outdoor scenarios for self-driving vehicles.

Kimera [23] built a local mesh of the scene based on multi-frame stereo data to guarantee globally-consistent trajectory estimation, but is also not suitable for open areas.

Other visual semantic SLAM [24–27] follow the same trends of using different deep learning based object detection [28,29] or segmentation models [30,31] as a filter to remove the unstable object and get a refined local image/depth map.

Some researchers try to use other sensor information to help visual SLAM adapt to outdoor environments. However, the unstable light and limited FOV make visual SLAM unable to meet safety requirements for self-driving vehicles.

2.2 LiDAR Semantic SLAM

Although RGB-D camera and stereo parallax estimation can offer point clouds, they are either within a short distance or inaccurate under bad light conditions. Only the LiDAR (multi-beam) based point cloud is considered.

Since there are almost no point cloud-based segmentation dataset, early algorithms, such as LIMO [32] and [33], fused the camera and LiDAR to get the semantic information of point clouds. The moving targets are first detected or segmented from the image view, and their BBoxes or masks are projected into the 3D point cloud for later filtering based on the external parameter matrix [10] of camera-LiDAR calibration system. On the one hand, the inherent difference between the FOV of the camera and the LiDAR will introduce projection errors. On the other hand, the camera FOV is relatively limited and can only filter part of dynamic targets, especially for 360-degree ring-like LiDAR.

To achieve fast segmentation for 3D sparse point clouds in large distances without involving camera image, LiSeg [34] follows the RangeNet++ [35] which directly deployed the segmentation on the 2D spherical mapping of raw point cloud. And then, the point cloud after removing the dynamic target is projected back to the 3D space for subsequent SLAM. SUMA++ [11] combines the multi-class flood fill with RangeNet++ to refine the 2D segmentation result of the spherical projection map. With semantic constraints from above operations, the projected scans matching through ICP are improved, and SUMA++ is able to work with very few static structures on the highway. OverlapNet [36] also benefits from the segmentation results of RangeNet++, and combines the semantic class probability with other point cloud cues for prediction of overlap of the current map and heading yaw of the agent.

Recurrent-OctoMap [37] uses a Nap-LSTM model to learn the semantic state transition between different time-scales of observation for the long term SLAM requirement. Different processing strategies will be used to construct different maps based on dynamic objects, such as moving and potential moving vehicles. The test shows that the OctoMap built from 7-day-long mapping data can maintain semantic memory using long-term experience. But long-term SLAM mapping ignores fine-grained spatial features, which makes real-time positioning accuracy poor.

With the release of the SemanticPOSS [38] and SemanticKITTI [13], it is possible to directly perform 3D point cloud semantic segmentation [14,39], and related semantic SLAM. The model directly performs semantic segmentation in three-dimensional space, which can effectively use the complete spatial information without additional projection and back-projection operations. The artifacts introduced back-projection can also be avoided.

3 System Workflow

Our workflow consists of the following four main steps (as shown in Fig. 1). 1) A normal SLAM-based data collection and refinement based on offline interactive SLAM is first adopted, the later operation involved human collaboration in closed-loop optimization. 2) The point clouds are then overlapped based on the refine posed from the last step, and the human is involved again in labeling the point cloud with cognitive prior knowledge. This operation generates both a point cloud map (through voxelization) and a per-frame labeled point cloud dataset with predefined semantic labels. 3) A point cloud segmentation model is then trained based on the collected and labeled dataset. 4) Finally, each scan is first fed into the segmentation model to get per-point labels, and matching to the semantic map in a 'global-local' paradigm.

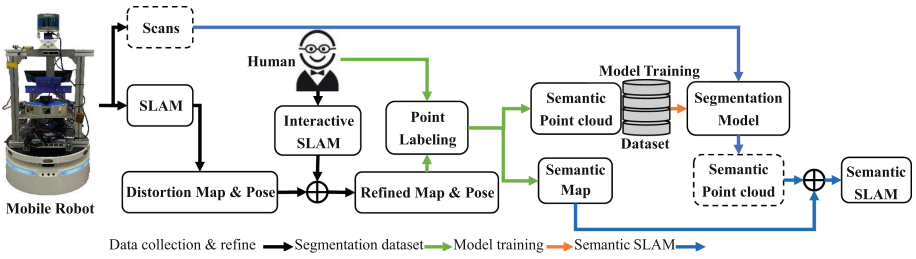
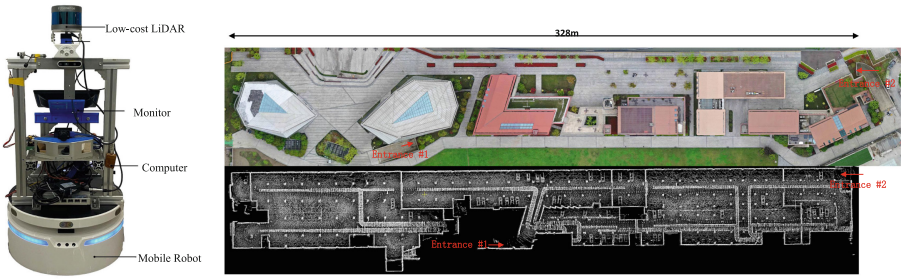


Fig. 1. The workflow of the semantic SLAM for mobile robots with human-in-the-loop.

With the involvement of human cooperation, we can refine the robot pose and final map without GPS positioning information or high-end laser trackers (such as Leica and Focus). At the same time, the point cloud is batch-labeled, based on the cumulative point cloud map instead of a single frame, with the help of human cognition. This greatly improves the labeling efficiency. Through the collaboration between human and the mobile robot, the subsequent semantic SLAM can be carried out. Next, the specific workflow will be introduced based on the local mobile robot and experimental environment.

3.1 Local Robot Platform

Our mobile robot is a differential two-wheel chassis equipped with a low-cost LiDAR with 16 beams@10 Hz and 360° (Robosense, RS-16), a monitor and embedded computer unit (Nvidia Jetson AGX), as shown in Fig. 2(a). The experimental environment is a large underground parking garage (about 328 m long), GPS signals cannot be received in this environment. A comparison between the bird's-eye view above ground building map based on UAV image stitching and the underground point cloud map optimized by this algorithm is shown in Fig. 2(b), it can be found that the final point cloud map has a high match with the surface scene.



(a) Mobile Robot (b) Underground garage

Fig. 2. Local experiment platform and environment.

3.2 Interactive SLAM-Based Data Collection and Refinement

Firstly, the data collection is performed by manually controlling the robot to traverse the scene. We use the lightweight Lego-LOAM as the initial mapping algorithm. On the basis of LOAM, the Lego-LOAM adds ground segmentation and clustering-based segmentation for front-end optimization, and optimizes the back-end of SLAM through a graph. The lightweight algorithm can be deployed in robotic embedded computing units.

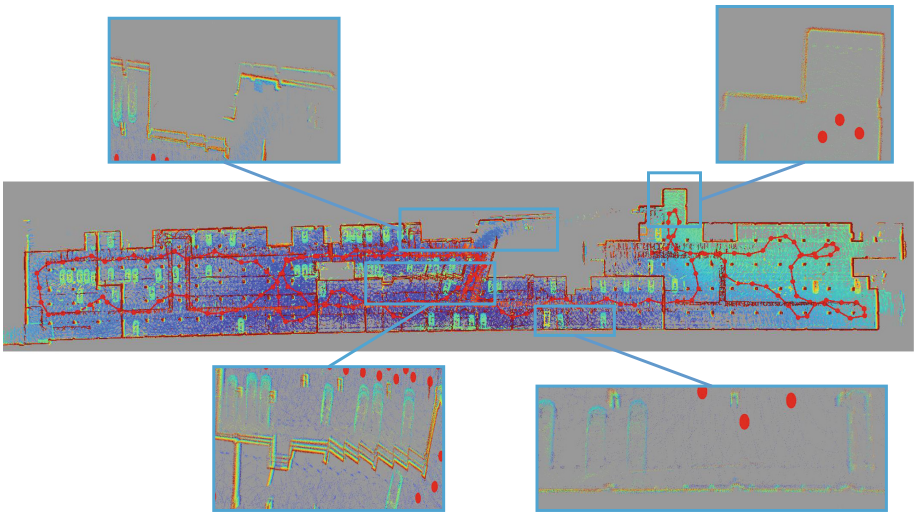


Fig. 3. The point cloud map output by the algorithm contains some significant mapping errors.

Due to the large size of the scene, the pose estimation bias will be introduced during robot motion (especially rotation), resulting in the shift or degradation

of the final point cloud map. Some large pose offsets or long-term accumulated errors can lead to closed-loop detection errors and failures. For man-made buildings, mapping errors are easily observed (as shown in Fig. 3), such as excessively thick walls, wall ghosting, ground noise, etc. At the same time, for the lower wall of the entire map, it can also be seen that there are obvious arcs instead of straight lines.

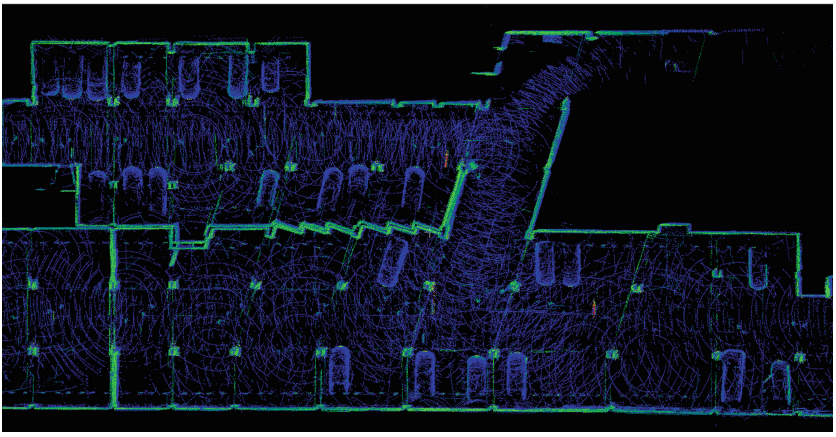


Fig. 4. The point cloud map output by the algorithm contains some significant mapping errors.

However, these errors can be easily detected by a human. Therefore, the offline interactive SLAM is introduced by adding closure and other pose constraints through a human operator during the back-end optimization. The interactive SLAM [12] mainly realizes map correction by introducing manual closed-loop detection in back-end optimization. Figure 4 shows the corrected results for the most heavily drifted region in Fig. 3. The whole process requires constant human-in-the-loop iterative optimization, and the specific optimization time depends on factors such as the scene size and the scale of draft areas. Finally, we get a refined pose for each frame, and a map from overlapped and voxelized point clouds.

3.3 Semantic Point Cloud Labeling and Segmentation Model Training

We then separately label the overlapped point cloud and the voxelized map according to the predefined classes with **Point Labeler**[1]. Table 1 illustrates the eight common targets appearing in the local underground garage, they can be roughly divided into stable senses (road, column, wall and ceiling), moving subjects (vehicle, motorcycle, pedestrian), and unrecognized noise. In the current

[1] https://github.com/jbehley/point_labeler.

stage, manual collaboration is introduced again, and data annotation is performed with the help of human cognition, which is used to build a semantic point cloud dataset and a global map with semantic labels.

Table 1. The predefined semantic labels for underground garage.

Label	Color	Example	Explanation
0	[0,0,0]	null	unlabeled noise or outliers
1	[0,0,255]		Vehicle
2	[245,150,100]		Motorcycle
3	[245,230,100]		Pedestrian
4	[250,80,100]		Road
5	[150,60,30]		Column
6	[255,0,0]		Wall
7	[180,30,80]		Ceiling

The **Point Labeler** organized the point cloud and corresponding relative pose ($[x, y, z, roll, pitch, yaw]$) as input, and overlapped the point cloud into a scene, and divided the whole scene into smaller square cells. Therefore, during the labeling process, the complete structure based on the nearest neighbor point cloud can give a more complete target space shape. And at the same time, with the help of the pose information optimized by interactive SLAM, batch annotation (relative to a single frame) can be easily and quickly performed. Figure 5 illustrates an example of the labeling tool UI with corresponding sense from the image, the annotator can easily identify the class of the object through the map-level point cloud (first annotate and filter the ground and ceiling).

We divide the labeled data into two disjoint subsets, one is used for model training, and the other is used for model evaluation. Comprehensively considering both computational efficiency and segmentation accuracy, the following three models are considered in the current study, i.e., RandlaNet [39], PolarNet [14], and Cyclinder3D [40].

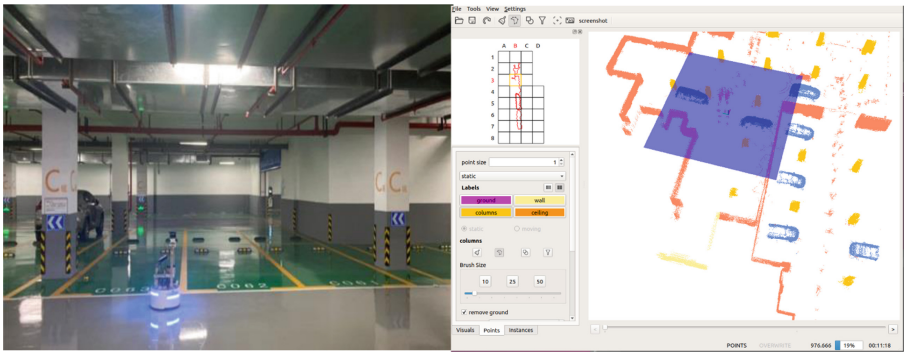


Fig. 5. The underground sense (left) and related point cloud map by removing ground and ceiling in **Point Labeler** (right).

3.4 Location Based on Semantic Point Cloud

Before starting work, the robot may start from different locations (for example, get items from different sources for delivery, or wake up from any map location for clearing), and it is critical to obtain an accurate current location through the prior semantic map. To further optimize the storage and subsequent searching of the point cloud map, the point cloud normal are calculated, and the point cloud is rasterized to obtain the triangular mesh-grid representation. This operation is done offline before integration into the robot localization system (Fig. 6).

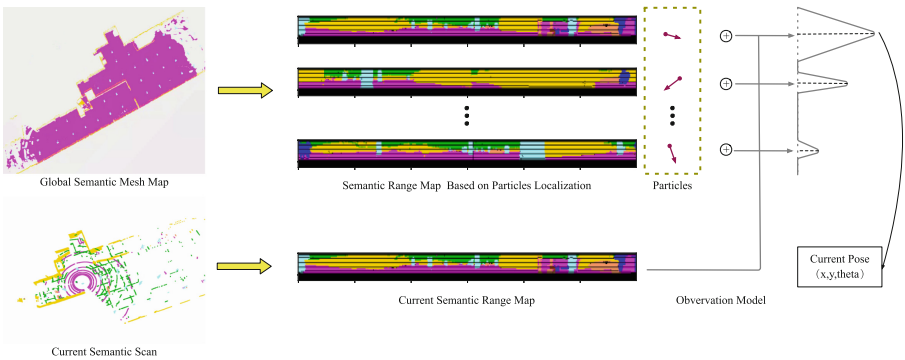


Fig. 6. A mesh grid based global semantic map is first generated from the labeled point cloud map. And the local semantic range-map is used for coarse localization with MCL.

As the point cloud segmentation model is trained, we encapsulate the semantic cognitive ability of humans to point clouds into the map and per-frame point cloud. Each time the robot gets a new scan, the point clouds are first

sent to the segmentation model, and the point-wise labels can be obtained. And we project a single frame point cloud $[x, y, z]$ onto a 2D range-map $[W, H]$ with the label index based on the LiDAR Cartesian to Polar projection formula (as shown in Eq. 1). Where $R = \sqrt{x^2 + y^2 + z^2}$ is the distance of points, $F_v = Fov_{up} + Fov_{down} = 30°$ is the vertical angle, $H_{scale} = 2°$ is the vertical resolution $H_r = Fov_v / H_{scale} + 1 = 16$, $W_r = 1800$ is the Horizontal angular resolution of the LiDAR.

$$\begin{pmatrix} W \\ H \end{pmatrix} = \begin{pmatrix} \frac{1}{2}[1 - \frac{arctan(y,x)}{\pi}]W_r \\ [1 - \frac{arcsin(z/R) + F_{up}}{F_v}]H_r \end{pmatrix} \tag{1}$$

Every time the robot is initially started, or needs to be relocated at intervals, the particles are evenly distributed on the semantic map through the Monte Carlo localization (MCL) based on particle filters. Once particles are scattered anywhere on the map (limit the z-axis height to the sensor installation height), we can generate a semantic range-map from the corresponding particle. Different from the previous work [41] that uses z-axis value as pixel or point distance R, we replaced it with the semantic label index, this helps reduce the variability of range-map distributions. To reduce the complexity, we constrain the robot motion from 6°C of freedom (DoF) $[X, Y, Z, Roll, Pitch, Yaw]$ to three DoF $[X, Y, Yaw]$, considering only motion in the 2D plane of bird's eye view (BEV). Therefore, the robot localization is $L_r = (x, y, yaw)$, and the corresponding observation at L_r is a semantic range-map $SRM_r = (W, H)$.

$$d = \frac{\sum |SRM_r - SRM_p|}{W * H} \tag{2}$$

We compared the current range-map generated from LiDAR scan $SRM_r = (W, H)$ with all the particles' range-map $SRM_{p(i=0,\ldots,n)} = (W, H)$, and calculated their similarities. The robot's current location is inferred from the semantic map with the highest similarity. The observation model can be defined as Eq. 2, the mean of the absolute pixel-wise difference of two images with the same scale $(W * H)$.

4 Experimental Study

4.1 Data Collection and Refinement

Through multiple (different dates) manual control of the robot in the underground parking lot, the random walk method is used to collect data to ensure the diversity of data. Figure 7 compares the different trajectories generated based on Lego-LOAM, the poses may contain drift, and will be corrected in the next operation. Table 2 illustrates more detailed statistical information of the point cloud sequences, and during the collection *seq1* and *seq2* use a limited maximum speed of 1.5 m/s.

On the one hand, the slow speed of the mobile robot leads to the high similarity of continuing frames; on the other hand, considering the computational

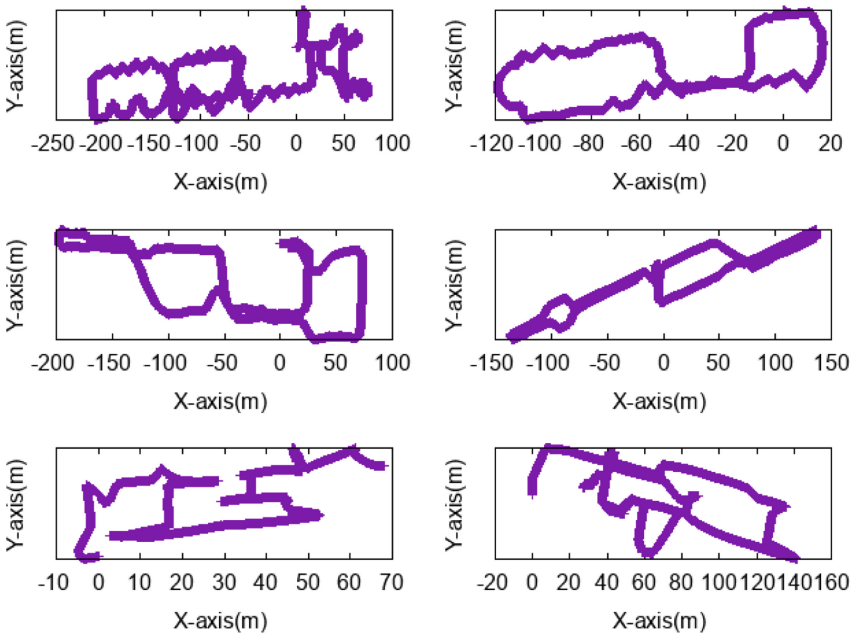


Fig. 7. Six random walks in the underground parking garage during data collection.

Table 2. Statistics of the six sequence trajectories.

Seq #	Frame	Point # (M)	Trajectory (m)	Duration (s)	Avg-speed (m/s)	Max-speed (m/s)
0	7814	318.37	944.31	793.50	1.19	2.37
1	4458	189.48	375.57	726.27	0.34	1.5
2	5471	242.08	652.10	854.24	0.76	1.5
3	5183	238.87	587.29	773.04	0.76	2.11
4	5725	270.13	294.01	863.12	0.57	2.27
5	7571	353.08	585.97	1219.34	1.48	2.27

cost of closed-loop optimization of the interactive SLAM process, the poses with a certain distance ($> 5\,m$) are extracted as candidate key frames by downsampling. Considering human-in-the-loop-based interactive SLAM is mainly optimized based on human subjective observation. Therefore, after optimizing six sets of collected data, an evaluation based on map topology entropy is introduced. In highly structured indoor environments, both Mean Map Entropy (MME) and Mean Plane Variance (MPV) are shown to be highly correlated with the trajectory error of SLAM [42]. Moreover, both the two metrics depend on the total number of points for the final map, the voxelization is first adopted to each map to normalize the map. Table 3 compares the average MME and MPV of the final maps with three scales of voxelization (0.2, 0.4 and 0.6). It can be seen that, when generating a point cloud map with degenerated raw poses,

those maps will contain drift frames and lead to higher MME and MPV. When those degenerated poses are corrected, the map consistency is improved, both of the two topological entropy drops.

Table 3. Evaluation based on point cloud topological entropy.

Voxelization method	0.2		0.4		0.6	
	MME	MPV	MME	MPV	MME	MPV
Raw pose	1.08	0.06	1.07	0.06	0.86	0.05
Interactive SLAM	0.81	0.05	0.77	0.04	0.58	0.04
LOAM with semantic	1.04	0.05	1.02	0.05	0.81	0.04

4.2 Semantic Dataset and Model Selection

Table 4 listed the information about the local labeled dataset based on the six collections and interactive SLAM refinement, we also compare the local dataset with previously published datasets (contains some tasks for architectural semantic segmentation). However, our local dataset is the only one that is collected based on low-cost LiDAR with a mobile robot in an underground situation. The biggest challenge of point cloud map construction and data annotation for this kind of scene is the lack of accurate positioning benchmarks, such as GPS or high-end laser trackers that can cooperate. 6261 and 1565 labeled frames are used for model training and testing, respectively, the two subsets are disjoint. Semantic labeling only relies on the downsampled poses ($> 5\,m$), this operation can reduce the similarity of the overall data and improve the learning efficiency. And that is why the final labeled points are less than the total frames of the six sequences in Table 2.

Table 4. Comparison between local and published semantic datasets.

Dataset	Frame	Point Scale (M)	Class #	Sensor	Scenes
HYY(ours)	6261+1565	325.9	8	RS-16	Underground
SemanticKITTI	23201+20351	4549	25	VLP-64E	Street
OakLand3d	17	1.6	5	SICK LMS	Street
Freiburg	77	1.1	4	SICK LMS	Street
Wachtberg	5	0.4	5	VLP-64E	Street
Semantic3D	15/15	4009	8	Terrestrial Laser Scanner	Street
Paris-Lille-3D	3	143	9	VLP-32E	Street

All the models are trained on a GPU desktop with Intel i7-9700 CPU, 16G memory, HDD disk and a single NVIDIA 2080ti GPU. We set the batch = 4, use Adam optimizer with learning rate = 0.0001 in the beginning, and train all

models for 80 epochs. Table 5 illustrated the comparison results on the testing set of our local data. The point-wise mean-Intersection over Union (mIoU) is selected to evaluate the model performance. The mIoU is defined according to Eq. 3, where TP_c, FP_c and FN_c correspond to the number of true positive, false positive, and false negative predictions labels for class c compare to the ground truth (GT) of testing set, and $C = 8$ is the total number of classes. However, we ignored the noise or outlier points in Table 5.

$$mIoU = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FP_c + FN_c} \tag{3}$$

Table 5. The semantic segmentation results on test set.

Model	mIoU	Per-Class mIoU							FPS
		Vehicle	Motorcycle	Pedestrian	Road	Column	Wall	Ceiling	
RandlaNet	89.93	96.50	92.67	67.52	93.19	94.42	97.22	87.99	62.50
PolarSeg	94.43	97.78	98.48	81.64	94.36	97.54	98.74	92.45	50.11
Cyclinder3D	95.57	98.50	98.64	82.70	95.86	98.20	99.22	95.87	12.29

It can be seen from the table that the overall mIoU of PolarSeg is slightly lower than that of the Cyclinder3D, but its speed is 50.11 frames per second (FPS). Although RandlaNet achieves the highest FPS at 62.5, however, its mIoU is much lower than the other two models. And Cyclinder3D is too slow which cannot meet the needs of in-vehicle computing units. Taking the mIoU and FPS into consideration, we prefer to choose PolarSeg. Its mIoU is only 1.14 lower than Cyclinder3D, but about 4 times faster. Subsequent semantic localization optimization is based on the point cloud semantic information of vehicles, motorcycles, pedestrians, ground bearing columns, walls and roofs obtained in this step, the category of noise points will be directly eliminated.

4.3 Localization Performance

We first compare our semantic range-map based location (refer as semantic-based) with the original range-map (refer as range-based). Each time we randomly selected one point from the collected sequence (with interactive SLAM refinement) as GT, and adopt the two different localization methods, 10000 particles are used for searching. We repeat this operation 30 times (i.e., 30 poses are used) and calculate the average time and total time as shown in Table 6.

It can be seen that, except when dealing with the seq1 (the trajectory of seq1 is very simple as shown in Fig. 7 column 1-right), our semantic-based method can achieve faster convergence time with 10000 particles MCL, the average time for the six sequences is 10.02 s, involving the semantic guidance reduces 13% calculating time. However, using this algorithm only when the robot is initialized for localization, the improvement is not significant.

Table 6. Comparison of the calculation time.

Seq #	Range-Based		Semantic-Based	
	Average Time (s)	Total Time (s)	Average Time (s)	Total Time (s)
0	11.87	341.29	**9.94**	**253.33**
1	**10.88**	**281.42**	11.39	306.88
2	11.45	304.78	**9.38**	**216.16**
3	11.53	308.82	**9.70**	**247.43**
4	11.36	325.79	**8.77**	**257.69**
5	11.97	274.02	**10.96**	**289.93**
Avg.	11.51	306.02	**10.02**	**261.90**

Considering that the relocation algorithm cannot meet the real-time requirements, we embed it as a module into the existing SLAM algorithm-LOAM. The algorithm will be used as a back-end optimization module to periodically start (that is, every one minute) to perform a global search and relocation to correct the drift of the robot. We refer to the corrected key frame poses with interactive SLAM from previous steps, which means only the poses with the shortest distance to the key frame poses are considered duration evaluation. It is easy to see from Table 3 that even involved with semantic information, the MME of the final point cloud map from LOAM is larger than Interactive SLAM. In the absence of GPS or other precise measurement equipment, it is feasible to use the results of interactive SLAM optimization as GT.

Table 7. Localization and rotation (Yaw) errors between different SLAM back-end optimization.

Seq #	A-LOAM		LOAM w Range-map		LOAM w Semantic Range-map	
	Location (m)	Yaw (deg)	Location (m)	Yaw (deg)	Location (m)	Yaw (deg)
0	1.01	0.16	0.96	0.11	0.84↓	0.07↓
2	0.23	0.32	0.20	0.21	0.16↓	0.08↓
4	1.58	1.18	1.52	0.81	1.19↓	0.50↓

Table 7 illustrates the evaluation results on the three challenge sequences of seq0, seq2 and seq4, their trajectories are more complex as shown in the Fig. 7 left column. The A-LOAM is the baseline algorithm without back-end optimization; LOAM w Range-map uses the range-based strategies and prior distance based point cloud map; LOAM with Semantic Range-map uses the semantic range-map and prior semantic point cloud map for periodic relocation optimization at every one minute period. It can be seen that, with a back-end optimization, both root mean square errors (RMSE) of the localization (x, y) and rotation (yaw) are reduced. In general, the search distance for closed-loop detection is short,

and there are few closed loops that can be formed in the trajectory of sequence 04. The estimated position and rotation RMSE of key frames are the largest, and the estimation error can also be significantly reduced by periodic relocation of the back-end. Moreover, with the semantic guidance, LOAM can achieve more remarkable results than the comparison methods.

5 Conclusion and Future Work

In order to improve the autonomous localization and mobility of mobile robots in unstable indoor illumination scenarios, this paper proposes a semantic optimization method based on low-cost LiDAR localization. In order to quickly and efficiently construct LiDAR point cloud data for semantic information acquisition without the aid of GPS or expensive measurement equipment, we introduced human collaboration twice in the entire workflow: 1) the interactive SLAM and 2) semantic data labeling. And the point cloud semantic segmentation model is used to simulate human cognitive ability for real-time point cloud semantic information acquisition. Based on the semantic information, we proposed a novel semantic range-map based MCL to improve the back-end of the A-LOAM. With multiple sequence data collected in a local underground garage, we perform extensive quantitative evaluations and comparative testing on the above workflow. The results show that periodic relocation optimization by introducing semantic information at the back-end can effectively reduce pose drift and overall map degradation. We also plan to introduce the valuable semantic information for the SLAM front-end in the future, improve the understanding of the dynamic environment, and optimize the overall workflow for different senses.

References

1. Luo, Y., Binbin, S., Zheng, X.: Trends and challenges for population and health during population aging-china, 2015–2050. China CDC Weekly **3**(28), 593 (2021)
2. Yang, G., et al.: Homecare robotic systems for healthcare 4.0: visions and enabling technologies. IEEE J. Biomed. Health Inform. **24**(9), 2535–2549 (2020)
3. Zhang, J., Singh, S.: Loam: lidar odometry and mapping in real-time. In: Robotics: Science and Systems, Berkeley, CA, vol. 2, no. 9, pp. 1–9 (2014)
4. Song, C.K., Uchanski, M., Karl Hedrick, J.: Vehicle speed estimation using accelerometer and wheel speed measurements. Soc. of Automotive Engineers (2002)
5. Barbieri, L., Brambilla, M., Trabattoni, A., Mervic, S., Nicoli, M.: UWB localization in a smart factory: augmentation methods and experimental assessment. IEEE Trans. Instrum. Meas. **70**, 1–18 (2021)
6. Wang, L., et al.: Initial assessment of the LEO based navigation signal augmentation system from Luojia-1a satellite. Sensors **18**(11), 3919 (2018)
7. Li, Y., He, L., Zhang, X., Zhu, L., Zhang, H., Guan, Y.: Multi-sensor fusion localization of indoor mobile robot. In: 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 481–486. IEEE (2019)
8. Campos, C., Elvira, R., Gómez Rodríguez, J.J., Montiel, J.M.M., Tardós, J.D.: Orb-slam3: an accurate open-source library for visual, visual-inertial, and multimap slam. IEEE Trans. Robot. **37**(6), 1874–1890 (2021)

9. Shan, T., Englot, B.: 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), pp. 4758–4765. IEEE (2018)
10. Cui, J., Niu, J., Ouyang, Z., He, Y., Liu, D.: ACSC: automatic calibration for non-repetitive scanning solid-state lidar and camera systems. arXiv preprint arXiv:2011.08516 (2020)
11. Chen, X., Milioto, A., Palazzolo, E., Giguère, P., Behley, J., Stachniss, C.: Suma++: efficient lidar-based semantic slam. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4530–4537. IEEE (2019)
12. Koide, K., Miura, J., Yokozuka, M., Oishi, S., Banno, A.: Interactive 3d graph slam for map correction. IEEE Robot. Autom. Lett. **6**(1), 40–47 (2020)
13. Behley, J., et al.: Semantickitti: a dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 9297–9307 (2019)
14. Zhang, Y., et al.: PolarNet: an improved grid representation for online lidar point clouds semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9601–9610 (2020)
15. Zhang, L., Wei, L., Shen, P., Wei, W., Zhu, G., Song, J.: Semantic slam based on object detection and improved octomap. IEEE Access **6**, 75545–75559 (2018)
16. Mur-Artal, R., Tardós, J.D.: Orb-slam2: an open-source slam system for monocular, stereo, and RGB-D cameras. IEEE Trans. Robot. **33**(5), 1255–1262 (2017)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
18. Wang, Z., Zhang, Q., Li, J., Zhang, S., Liu, J.: A computationally efficient semantic slam solution for dynamic scenes. Remote Sen. **11**(11), 1363 (2019)
19. Kang, X., Yuan, S.: Robust data association for object-level semantic slam. arXiv preprint arXiv:1909.13493 (2019)
20. Long, X., Zhang, W., Zhao, B.: Pspnet-slam: a semantic slam detect dynamic object by pyramid scene parsing network. IEEE Access (2020)
21. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890 (2017)
22. Zhao, Z., Mao, Y., Ding, Y., Ren, P., Zheng, N.: Visual-based semantic slam with landmarks for large-scale outdoor environment. In: 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI), pp. 149–154. IEEE (2019)
23. Rosinol, A., Abate, M., Chang, Y., Carlone, L.: Kimera: an open-source library for real-time metric-semantic localization and mapping. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 1689–1696. IEEE (2020)
24. Fan, Y., et al.: Semantic slam with more accurate point cloud map in dynamic environments. IEEE Access **8**, 112237–112252 (2020)
25. Mahe, H., Marraud, D., Comport, A.I.: Real-time RGB-D semantic keyframe slam based on image segmentation learning from industrial cad models. In: 2019 19th International Conference on Advanced Robotics (ICAR), pp.s 147–154. IEEE (2019)
26. Nicholson, L., Milford, M., Sünderhauf, N.: Quadricslam: constrained dual quadrics from object detections as landmarks in semantic slam. IEEE Robot. Autom. Lett. (RA-L) (2018)
27. Li, R., Wang, S., Dongbing, G.: Ongoing evolution of visual slam from geometry to deep learning: challenges and opportunities. Cogn. Comput. **10**(6), 875–889 (2018)

28. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
29. Lehtonen, M., Ostojic, D., Ilic, A., Michahelles, F.: Securing RFID systems by detecting tag cloning. In: Tokuda, H., Beigl, M., Friday, A., Brush, A.J.B., Tobe, Y. (eds.) Pervasive 2009. LNCS, vol. 5538, pp. 291–308. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01516-8_20
30. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
31. Dong, X., Niu, J., Cui, J., Fu, Z., Ouyang, Z.: Fast segmentation-based object tracking model for autonomous vehicles. In: Qiu, M. (ed.) ICA3PP 2020. LNCS, vol. 12453, pp. 259–273. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60239-0_18
32. Graeter, J., Wilczynski, A., Lauer, M.: Limo: lidar-monocular visual odometry. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7872–7879. IEEE (2018)
33. Jian, R., et al.: A semantic segmentation based lidar SLAM system towards dynamic environments. In: Yu, H., Liu, J., Liu, L., Ju, Z., Liu, Y., Zhou, D. (eds.) ICIRA 2019. LNCS (LNAI), vol. 11742, pp. 582–590. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-27535-8_52
34. Zhao, Z., Zhang, W., Jianfeng, G., Yang, J., Huang, K.: Lidar mapping optimization based on lightweight semantic segmentation. IEEE Trans. Intell. Veh. **4**(3), 353–362 (2019)
35. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet++: fast and accurate lidar semantic segmentation. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4213–4220. IEEE (2019)
36. Chen, X.: et al.: Overlapnet: loop closing for lidar-based SLAM. In: Proceedings of the Robotics: Science and Systems (RSS), Freiburg, Germany, pp. 12–16 (2020)
37. Sun, L., Yan, Z., Zaganidis, A., Zhao, C., Duckett, T.: Recurrent-octomap: learning state-based map refinement for long-term semantic mapping with 3-d-lidar data. IEEE Robot. Autom. Lett. **3**(4), 3749–3756 (2018)
38. Pan, Y., Gao, B., Mei, J., Geng, S., Li, C., Zhao, H.: Semanticposs: a point cloud dataset with large quantity of dynamic instances. arXiv preprint arXiv:2002.09147 (2020)
39. Hu, Q., et al.: Randla-net: efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11108–11117 (2020)
40. Zhu, X., et al.: Cylindrical and asymmetrical 3D convolution networks for lidar segmentation. arXiv preprint arXiv:2011.10033 (2020)
41. Chen, X., Vizzo, I., Läbe, T., Behley, J., Stachniss, C.: Range image-based lidar localization for autonomous vehicles. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 5802–5808. IEEE (2021)
42. Razlaw, J., Droeschel, D., Holz, D., Behnke, S.: Evaluation of registration methods for sparse 3d laser scans. In: 2015 European Conference on Mobile Robots (ECMR), pp. 1–7. IEEE (2015)