

ASL-SLAM: A LiDAR SLAM With Activity Semantics-Based Loop Closure

Baoding Zhou^{ID}, Member, IEEE, Chunyu Li, Shoubin Chen^{ID}, Doudou Xie, Min Yu, and Qingquan Li^{ID}

Abstract—A practical back-end module with loop closure detection is very useful and important for a LiDAR simultaneous localization and mapping (SLAM) system to perform high-precision positioning and mapping tasks. However, most existing loop closure detection methods are based on images or point clouds, and these methods may produce errors when the structure or texture is similar. To overcome this problem, we propose a complete LiDAR SLAM system, including a front-end odometry module based on normal distribution transform (NDT)-LOAM and a back-end optimization module with loop closure based on activity semantics.

Through the analysis and calculation of inertial measurement unit (IMU) data from SLAM platforms such as unmanned ground vehicles (UGVs), the activity semantics of turning and passing over a speed bump are detected based on the peak z-axis angular velocity and z-axis acceleration, respectively. Then, according to this activity semantics information and its unique and definite attributes, we establish correct loop closure detection using rough geometric detection, activity semantics matching, and point cloud rematching for validation. Finally, graph optimization theory is utilized to reduce the global cumulative error, improve the global trajectory accuracy and map consistency, and obtain the final global motion trajectory and point cloud map. We collected a dataset for evaluation, which contains indoor data, outdoor data, and indoor–outdoor integration data, and we also evaluated our method on the KITTI dataset. The experimental results for different scenes show that the addition of activity semantics can effectively help loop closure detection and improve LiDAR SLAM system performance.

Index Terms—Activity semantics, graph optimization, inertial measurement unit (IMU), LiDAR, loop closure detection, simultaneous localization and mapping (SLAM).

Manuscript received 15 March 2023; revised 23 April 2023; accepted 23 April 2023. Date of publication 1 May 2023; date of current version 14 June 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 42171427 and Grant 42101445, in part by the Shenzhen Scientific Research and Development Funding Program under Grant KQTD20180412181337494, in part by the Central Guided Local Science and Technology Development Special Project under Grant 20222ZDH04090, and in part by the Shenzhen Key Laboratory of Structure Safety and Health Monitoring of Marine Infrastructures under Grant ZDSYS20201020162400001. The associate editor coordinating the review of this article and approving it for publication was Prof. Huang Chen Lee. (*Corresponding author: Shoubin Chen*.)

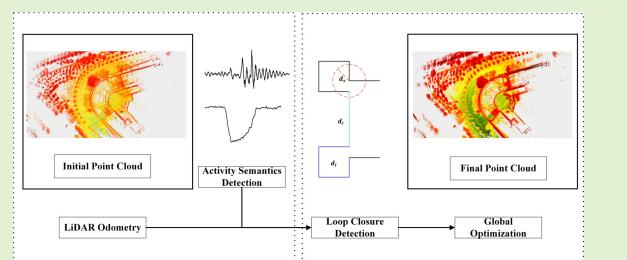
Baoding Zhou, Chunyu Li, and Doudou Xie are with the Institute of Urban Smart Transportation and Safety Maintenance, the Key Laboratory for Resilient Infrastructures of Coastal Cities, Ministry of Education, and the College of Civil and Transportation Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: bdzhou@szu.edu.cn; 2070474152@email.szu.edu.cn; 2070474167@email.szu.edu.cn).

Shoubin Chen is with the Guangdong Provincial Laboratory of Artificial Intelligence and Digital Economy (Shenzhen), Shenzhen University, Shenzhen 518060, China, also with the School of Architecture and Urban Planning, Shenzhen University, Shenzhen 518060, China, and also with Orbbee Research, Shenzhen 518052, China (e-mail: shoubin.chen@whu.edu.cn).

Min Yu is with the College of Communication, Electronic Engineering, and Computer Science, Jiangxi Normal University, Nanchang 330022, China (e-mail: myu@jxnu.edu.cn).

Qingquan Li is with the Guangdong Key Laboratory of Urban Informatics and the Guangdong Laboratory of Artificial Intelligence and Digital Economy (Shenzhen), Shenzhen University, Shenzhen 518060, China (e-mail: liqq@szu.edu.cn).

Digital Object Identifier 10.1109/JSEN.2023.3270871



I. INTRODUCTION

LiDAR simultaneous localization and mapping (SLAM) has been widely studied over the past 30 years and is now commonly used in robotics, unmanned ground vehicles (UGVs) [1], [2], environmental 3-D modeling [3], [4], complex structure scene reconstruction [5], and other fields. At the same time, LiDAR SLAM has gradually attracted the attention of researchers in the field of photogrammetry due to its deep relation to point cloud processing, location-based services (LBSs), and mobile mapping system (MMSs) [6].

The core of LiDAR SLAM utilizes LiDAR sensors to perceive an environment and estimate states, i.e., position and attitude. Generally, a typical LiDAR SLAM system can be divided into a front-end odometry module and a back-end optimization module. The front-end module collects a 3-D point cloud of the environment, calculates the position and posture transformation of two adjacent frames using either a iterative closest point (ICP) [7] or NDT [8] method, and finally outputs the MMS odometry and trajectory. The back-end optimization corrects the cumulative errors of the front-end odometer and improves the state estimation accuracy. However, some SLAM algorithms contain only a front end, such as classical LOAM [9], [10] and its many variants, including LOAM-Livox [11], LeGO-LOAM [12], and NDT-LOAM [13].

In the back-end module of an LiDAR SLAM system, loop closure detection is of great significance. Loop closure detection methods include scan context [14], scan context++ [15], ISC [16], M2DP [17], and PointNetVLAD [18]. Since loop closure detection associates recent data and all historical data, it helps improve the accuracy, robustness, and consistency of the entire SLAM system. Some methods [19], [20] identify the environment semantics and use the semantic information to match them. However, conventional SLAM mainly relies on visual image or LiDAR point cloud environmental perception information and calculates the environmental similarity discrimination to detect loop closure. Image-based methods may find incorrect loops in places where the environmental texture is similar, causing mapping to fail. For point cloud-based methods, repeated and similar scenarios in an environment will make the method ineffective [21].

Rather than environmental perception information, MMSs, such as wheeled robots and UGVs, incorporate mobile activity semantic information during data collection, such as when passing over a speed bump or turning around. Activity semantic detection has become a mature method used in pedestrian navigation and positioning algorithms [22], [23]. This technology uses an inertial measurement unit (IMU) to detect pedestrian activity semantic information, such as turning, and uses this information to retrieve pedestrian positions in a global activity semantic map.

In this article, we propose ASL-SLAM, which is an LiDAR SLAM with activity semantics-based loop closure. UGVs will generate activity semantics when passing over speed bumps and turning. This method performs loop closure detection by detecting the UGV activity semantics, and then performs global optimization and builds a global point cloud map. The main contributions of this work are summarized as follows.

- 1) We build a complete LiDAR SLAM system, including a front-end odometry module based on NDT-LOAM and a back-end optimization module with loop closure based on activity semantics.
- 2) We propose a loop closure detection method based on activity semantics detection, which can be applied to scenes where traditional point cloud and image-based methods are not applicable, such as underground parking lots and corridors where structures and textures are similar. To the best of our knowledge, this article is the first to use activity semantics for loop closure.
- 3) Experiments using an SZU-Litchi UGV and the KITTI dataset show that the accuracy of our method is better than that of other advanced methods in complex scenes such as underground parking lots and parks.

The rest of this article is organized as follows. Section II reviews related work. Section III introduces the system and methods. Section IV provides the experimental results, and Section V is the conclusion.

II. RELATED WORK

A. LiDAR SLAM

To calculate the pose transformation between two adjacent frames, LOAM [9], [10] selects the edge and planar

points as feature points by sorting the curvature on each scan. Then, the corresponding feature points are found in the data of two adjacent frames, and these points are used to calculate the pose transformation. As an improved LOAM algorithm, LeGO-LOAM [12] divides a point cloud into ground points and nonground points, extracts feature points from these points, and calculates the transformation. IMLS-SLAM [24] uses a scan-to-model framework and a special sampling strategy, matches the points to the surface reconstructed using the IMLS method, and finally minimizes the distance between the points and the surface. Unlike LOAM, SuMa [25] expresses the map using surface elements and minimizes the distance between points and faces using the ICP method. SuMa++ [26] further improves SuMa using semantic information to remove dynamic objects and using semantic information for ICP. MULLS [27] extracts five feature points, the ground, facade, pillar, beam, and roof, to match and calculate the transformations.

B. Loop Closure Detection

Loop closure detection methods can be divided into geometry-based methods and appearance-based methods [28]. Geometry-based methods search the vicinity of a certain point according to the spatial geometric relationship of the front-end odometry track to determine loop closure. The advantage of this method is that it is intuitive, but the disadvantage is that loop closure cannot be performed when the cumulative error is large.

Appearance-based methods can be divided into three types: image-based methods, point cloud-based methods, and fusion-based methods [21]. Among image-based methods, bag-of-words (BOW) [29], [30] is the most famous. In this method, an image is represented as a histogram of words existing in a dictionary, which is used to compare the similarity between an image and the previous image. Another method uses deep learning to identify images. Hou et al. [31] used a convolution neural network (CNN) to identify images and then looked for two similar images as loops.

Point cloud-based methods can usually be categorized into local, global, and segment-based methods. Local descriptor methods [32], [33] usually describe the local information around key points and use this information for similarity matching. Global descriptor methods such as M2DP [17], Scan Context [14], Scan Context++ [15], LocNet [34], OverlapNet [35], and ISC [16] find loops by comparing 2-D images from 3-D point cloud transformation. PointNetVLAD [18], SeqLPD [36], and LPD-Net [37] extract features using PointNet [38] and look for similar features via NetVLAD [39]. Charroud et al. [40] proposed a localization method based on an improved clustering particle filter. And Charroud et al. [41] proposed a method using clustering to identify features in a point cloud, and used these features for positioning. Segmentation-based approaches such as SegMatch [42] divide the point cloud into different fragments for feature extraction, and then use a random forest to match the features. SegMap [43] uses 3-D CNNs to extract point cloud features.

Fusion-based methods usually combine point clouds and images. Such methods include LV-SLAM [28], which uses

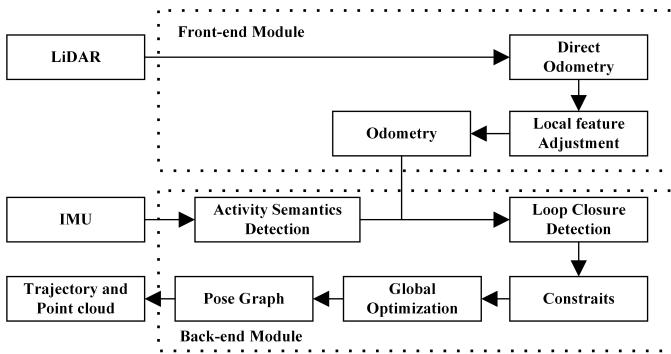


Fig. 1. ASL-SLAM system structure.

the BOW method to find loop closures in an image and uses the LiDAR point cloud to check whether they are real loop closures.

C. Activity Semantics

Activity semantics methods are usually used for pedestrian navigation and positioning. Zhou et al. [22] detected pedestrian activity semantics and used the pedestrian dead reckoning (PDR) [44] method and the activity-based map matching (AMM) [45] method to locate pedestrians.

Zhou et al. [23] used crowdsourcing data from mobile phone sensors to detect pedestrian activity semantics and then aligned multiple tracks according to the same activity semantics. Finally, a method based on graph optimization was used to construct indoor maps.

Liu et al. [46] used turning motion to assist positioning. They used WiFi for trajectory estimation and then looked for loop closure by identifying the same turning activity.

III. SYSTEM AND METHODS

A. System Overview

As shown in Fig. 1, the system consists of two parts: a front-end odometry module and a back-end optimization module.

We use the improved NDT-LOAM algorithm [13], which replaces the ICP module of LOAM with the NDT module to complete the front end. The ICP algorithm is used by LOAM to match frames. First, the matching feature points in the point clouds of two adjacent frames are calculated, and the Euclidean distance of these matching points is taken as the error objective function. The error of the method will increase if the feature point matching is wrong. The NDT algorithm first divides the space into several grids; second, it projects the point cloud to each grid; third, it calculates the normal distribution function of the grid; and fourth, it projects the next frame point cloud into the grid according to a transformation matrix, calculates the probability distribution function of the response and constructs the error function. According to the characteristics of our method, we optimized the NDT-LOAM parameters so that it can achieve higher accuracy. The front end of our algorithm can run at a high frequency of 10 Hz and can accurately register a point cloud.

The back-end global map optimization module is used to create global pose map optimization and point cloud maps

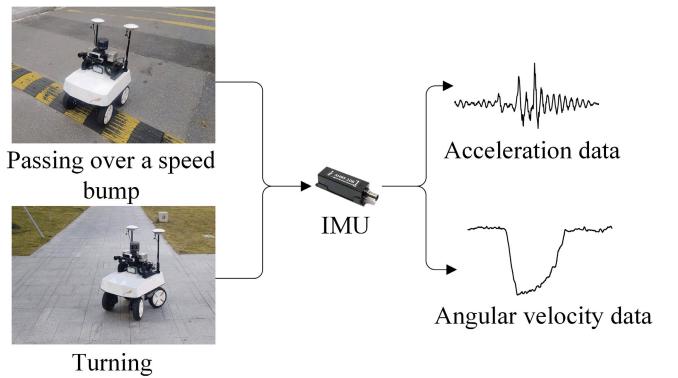


Fig. 2. Diagram of activity semantics detection.

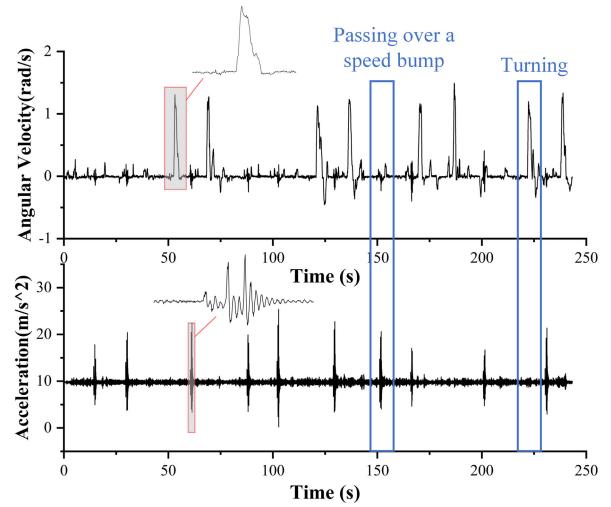


Fig. 3. Diagram of the IMU data.

after loop detection. First, this module detects the activity semantics when passing over speed bumps and turning by extracting IMU data. Second, the activity semantics are associated with the corresponding key frame of the point cloud according to the nearest neighbor timestamp. Third, within a certain threshold space range of key frames with active semantic labels, the module determines whether there are other key frames with the same label. If one exists, it will be considered a loop closure. Finally, graph optimization theory corrects the error of the global graph, which improves the accuracy of the global graph trajectories, and the results are output.

B. Activity Semantics Detection

UGVs have many activity semantics, such as turning, going uphill, going downhill, passing over a speed bump, and turning around. In this study, we selected the activity semantics of turning and passing over a speed bump. We assume that the IMU z-axis direction is upward. A diagram of the activity semantics detection method is shown in Fig. 2.

The detection method for passing over a speed bump is based on the peak of the z-axis acceleration (peak_ACC). As shown in Fig. 3, when UGV passes over the speed bump, the z-axis acceleration of the UGV generates a specific

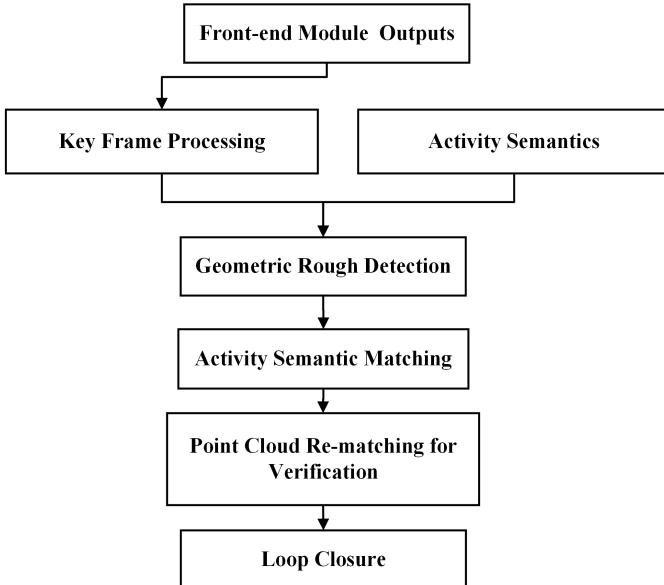


Fig. 4. Flowchart of the loop closure detection method.

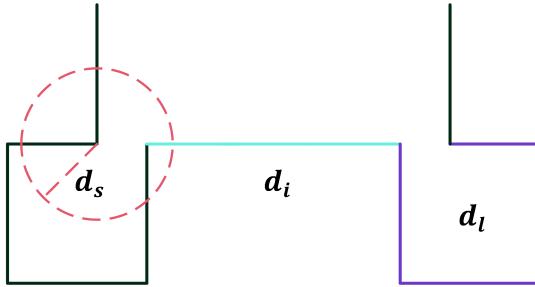


Fig. 5. Thresholds of geometric rough detection.

waveform. In Fig. 3, we can see that most of the data are approximately 9.8 m/s^2 , which means that the UGV is driving on a flat road. In each fluctuation, there are two peaks, representing the front and rear wheels of the UGV passing over the speed bump. ACC_TH represents the threshold of detection classification. If $\text{peak_ACC} > \text{ACC_TH}$, the UGV passed over a speed bump. According to the results of many experiments, ACC_TH is set to 16 m/s^2 in this study.

The detection method for turning is based on the peak z-axis angular velocity (peak_ANG). As shown in Fig. 3, the z-axis angular velocity of the UGV will generate a specific waveform. When the UGV moves straight, the angular velocity is close to 0 rad/s. When the UGV turns, the angular velocity changes dramatically, to approximately $\pm 1 \text{ rad/s}$. After turning, it recovers to straight movement, and the angular velocity gradually recovers to approximately 0 rad/s. ANG_TH represents the threshold of detection classification. If $|\text{peak_ANG}| > |\text{ANG_TH}|$, then the UGV is turning. According to the results of many experiments, ANG_TH is set to $\pm 0.6 \text{ rad/s}$ in this article.

C. Loop Closure Detection

Combined with the geometry-based method, we designed a new loop closure detection method, as shown in Fig. 4. The principle of this method is that if a certain activity semantics

is detected, and within a certain spatial threshold range, there is another detected activity semantics of the same type in the previous trajectory, then it can be considered that the UGV has passed the previous trajectory again. The trajectory points corresponding to the semantics of these two activities form a closed loop.

Key frame processing must be performed first. The output frequency of the front-end module is 10 Hz, the same as that of the data collected by LiDAR. However, such a high-processing frequency is not needed for loop closure detection. Therefore, whenever the motion state exceeds a distance threshold δ_d or angle threshold δ_a , a frame of data is extracted as a key frame.

After filtering out the key frames, the appropriate candidate loop closure frames are selected for each key frame through geometric rough detections. When processing the current key frame, the candidate loop closure frame is selected from the historical key frames according to three threshold judgment conditions.

- 1) *Search Area Threshold d_s* : The key frame can be selected as a candidate loop closure frame only when the space position is smaller than this threshold.
- 2) *Loop Interval Threshold d_i* : The interval between two loops should be greater than this threshold.
- 3) *Loop Length Threshold d_l* : The loop length must be greater than this threshold.

Only key frames that meet the above conditions can be selected as preliminary candidate loop closure frames. A diagram of the geometric coarse detection threshold is shown in Fig. 5.

Then we can use activity semantics for accurate loop closure detection. After the key frame selection and candidate loop closure frame selection are processed, the results of activity semantics detection as described in Section III-B are matched to the key frame nearest to the timestamp. After this step, each key frame has its own activity semantic label, where the label 0 indicates there are no activity semantics, the label 1 indicates passing over a speed bump, and the label 2 means turning. The key frames with activity semantic attributes are compared with the activity semantic attributes of all candidate loop closure frames. If the same activity semantics exist, then the frame is the loop frame of the current frame. This means that if the UGV turns at the time and place of this key frame and there is also a turn in the historical data within the space threshold range, then it can be judged that the two activities pass through the same turn. That is, the UGV returns to the loop closure point. After detecting a loop closure, we also determine whether this is a reverse loop according to the odometry direction. If it is a reverse loop, we give the point cloud an initial coarse yaw rotation according to the angle of the odometer and finally perform ICP on the point cloud to compute the exact transformation.

Finally, for matching accuracy, we need to perform point cloud rematching verification. We calculate the Euclidean consistency score, which is the mean value of the square of the distances from the source point clouds to the target point clouds. If this value is greater than the threshold, the

loop closure will be discarded. In our experiments, we set the threshold to 3.0.

D. Global Graph Optimization

The front-end module uses point clouds to match the current frame with one or more neighboring frames to estimate the current poses. Over time, the track becomes longer, and the map becomes larger. At the same time, both the direct matching method and the feature point method have errors. Therefore, the front-end odometry trajectory error will become increasingly large, and the inconsistency of the global graph will become increasingly obvious.

To improve the key frame pose precision and the quality of the global map, we can keep the track results of the front-end odometry and build another back-end global map optimization module to reduce the cumulative error. In the global pose map, the key frame pose is taken as the node, and the relative motion estimator between the two pose nodes obtained using the point cloud matching method is taken as the constraint edge. Finally, the nonlinear least-squares method is used to remove the adjustment to obtain more accurate and more consistent results.

After finding the loop closure, we use G²o [47] to construct a pose graph and optimize it. According to graph optimization theory, in this optimization problem, a common edge comes from the pose estimation between adjacent key frames of the point cloud, and a loop edge comes from the loop closure based on activity semantics detection. Suppose that the poses corresponding to each key frame of the point clouds are represented by $T_1, T_2, T_3, \dots, T_n$, which is a Lie group and Lie algebra. The relative pose transformation of frame i and frame j can be expressed as

$$\Delta T_{ij} = T_i^{-1} T_j. \quad (1)$$

The error function between frame i and frame j is

$$e_{ij} = \ln \left(\Delta T_{ij}^{-1} T_i^{-1} T_j \right)^{\vee}. \quad (2)$$

The formula can be written as follows, where ξ_i and ξ_j are the variables to be estimated and ξ_{ij} represents the transformation between frames

$$e_{ij} = \ln \left(\exp \left((-\xi_{ij})^{\wedge} \right) \exp \left((-\xi_i)^{\wedge} \right) \exp \left(\xi_j^{\wedge} \right) \right)^{\vee}. \quad (3)$$

Our goal is to make e_{ij} as small as possible. Open-source libraries such as Ceres and g2o also provide some solution methods for graph optimization.

After optimization, the cumulative errors can be eliminated, resulting in a more accurate trajectory and a more consistent global map.

IV. EXPERIMENTS AND RESULTS

A. Dataset and Implementation Details

As shown in Fig. 6, the platform of our experiment is an SZU-Litchi UGV equipped with multiple sensors. The UGV is equipped with a Robosense RS-LiDAR-16 LiDAR, an InertialLabs AHRS-10 IMU, a BDStar Navigation NC502-D RTK, two Livox Mid-40 LiDARs and several cameras with different



Fig. 6. SZU-Litchi UGV.

functions. The RS-LiDAR-16 LiDAR measurement range is 150 m with an accuracy of ± 2 cm. Its vertical field of view (FOV) is 30° , and its horizontal FOV is 360° . It is a 16-line LiDAR that can provide a vertical resolution of 2° and a horizontal resolution ranging from 0.1° to 0.4° depending on the rotation rate. In this article, we set the scanning frequency to 10 Hz, providing a horizontal angular resolution of 0.2° . According to the needs of our method, after a large number of experimental tests, we set the key frame distance parameter in the front-end module to 0.1 m. The x -axis of the IMU is to the right, the y -axis is forward, and the z -axis is upward. For the RTK, we purchased the QianXun FindCM service, which can provide a horizontal positioning accuracy of 2 cm and an elevation positioning accuracy of 5 cm. In this article, we set the frequency of the IMU to 200 Hz and that of the RTK to 10 Hz.

In this experiment, we used only the mechanical RS-LiDAR-16 LiDAR at the top, the RTK at the back, and the IMU in the middle. The average moving speed of the UGV is 1 m/s, and the operating speed of our algorithm is 10 Hz.

In outdoor scenes, RTK signals are available, so our ground-truth data come from the RTK. However, there is no RTK signal in indoor scenes. Therefore, we control the start and end points of the UGV to be the same point for accuracy evaluation. Some important threshold parameters used in the algorithm are set as follows. The translation and rotation thresholds for key frame selection are 2.0 m and 0.17° , respectively. The thresholds d_s , d_i , and d_l for rough geometric detection are 5.0, 7.0, and 10.0 m, respectively.

Using the UGVs, we set up four representative scenes, sequences #01, #02, #03, and #04, for the experiments, as shown in Fig. 7. Scene #01 is an underground parking lot at the Zhi Li Building at Shenzhen University, with a 50×15 m floor plan, as shown in Fig. 7(a). The scene has many similar structures, as well as many turns and speed bumps. There are six speed bumps along the rectangular road network. The data collection path is planned as two rectangles, one large and one small. On this path, the UGV passed over ten speed bumps and made eight turns. Since there is no RTK

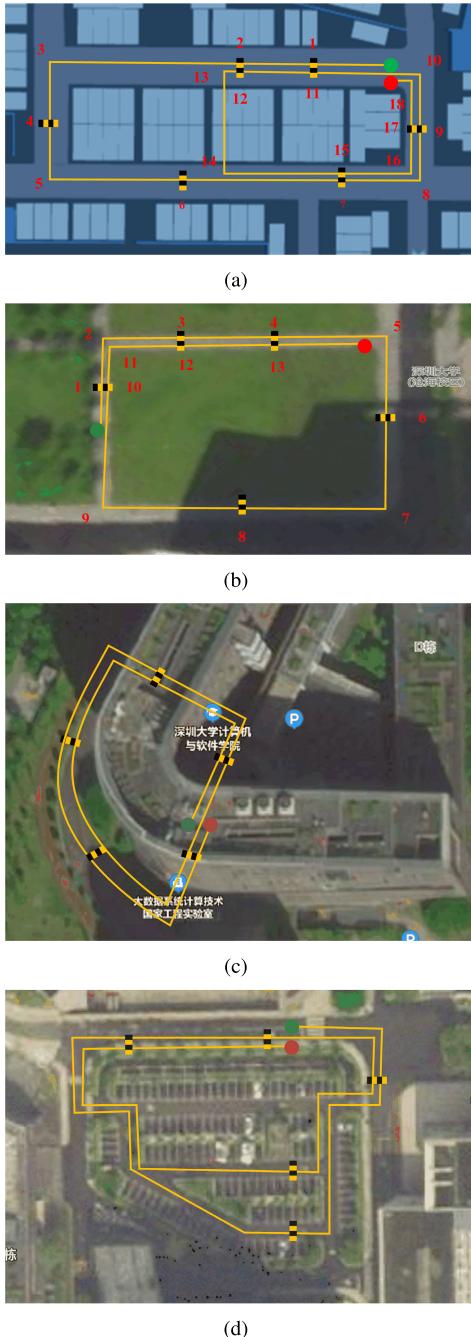


Fig. 7. Experimental scene and trajectory diagram. (a) Scene #01, (b) scene #02, (c) scene #03, and (d) scene #04. The yellow line represents the track and direction of the UGV, the green point represents the starting point, the red point represents the endpoint, and a short line segment with yellow and black represents a speed bump. The red numbers represent the sequence number of the activity semantics. The starting and ending points in scene #01 and scene #03 overlap, but we did not draw them together in the diagram for visualization.

signal underground, there is no effective method to collect the ground-truth value, so we purposely overlapped the starting and ending points. Then we used the absolute trajectory error (ATE) for evaluation.

Scene #02 is a rectangular road network around lawn in front of the Zhi Li Building at Shenzhen University, with an area of 40×60 m, as shown in Fig. 7(b). It is an open outdoor

TABLE I
RESULTS OF ACTIVITY SEMANTICS DETECTION

Sequence	Activity Semantic	Actual Quantity	Detected Quantity	Accuracy	Recall
#01	Speed Bump	10	10	100%	100%
	Turn	8	8	100%	100%
#02	Speed Bump	8	8	100%	100%
	Turn	5	5	100%	100%
#03	Speed Bump	10	14	71.43%	100%
	Turn	6	12	50%	100%
#04	Speed Bump	8	8	100%	100%
	Turn	17	17	100%	100%
KITTI 00	Speed Bump	0	-	-	-
	Turn	26	28	92.86%	100%

space with an available RTK signal and many moving objects. There are five speed bumps along the road network. On this path, the UGV passed over eight speed bumps and made five turns.

Scene #03 is a fan-shaped road with a perimeter of 200 m, which is located at the School of Computer Science of Shenzhen University, as shown in Fig. 7(c). It is an indoor-outdoor fusion scene, and its curved road can simulate the effect of continuous turns on activity semantics. This is a complex scene. There are many potholes and small stones on the road in this scene, and there are many moving cars and pedestrians on the road. Because the RTK signal in this environment is very poor, we used the ATE for evaluation. This scene has five speed bumps and three turns. A path of two consecutive laps with the same start and end points was used to collect data. On this path, the UGV passed through ten speed bumps and six turns.

Scene #04 is a complex outdoor parking lot at Shenzhen University, with a 90×80 m area, as shown in Fig. 7(d). There are many green belts in this scene, which will block the view, so the traditional method will be limited. There are also a large number of similar turns in this parking lot, and the distance between turns is very close. These turns interfere with the detection of activity semantics and loop closures, which can be used to evaluate the robustness of our method. Because there are many buildings near the scene, the RTK signal is very poor. Therefore, we use the ATE for evaluation. This scene has five speed bumps and ten turns. The UGV passed through eight speed bumps and 17 turns.

To evaluate whether our method works in complex urban environments, we used sequence 00 in the KITTI dataset, which are urban environment data, for evaluation.

B. Activity Semantics Detection

We used the threshold method to detect acceleration and angular velocity to identify activity semantics. The complete raw data from scene #01, scene #02, and scene #03 detected using the IMU are shown in Fig. 8. The detection results are shown in Table I.

As shown in Fig. 8, the activity semantics detected by our method correspond exactly to the actual activity semantics. In experiments #01, #02, and #04, our paths featured a total of 26 speed bumps and 30 turns. The peak detection algorithm successfully detected all the activity semantics, and

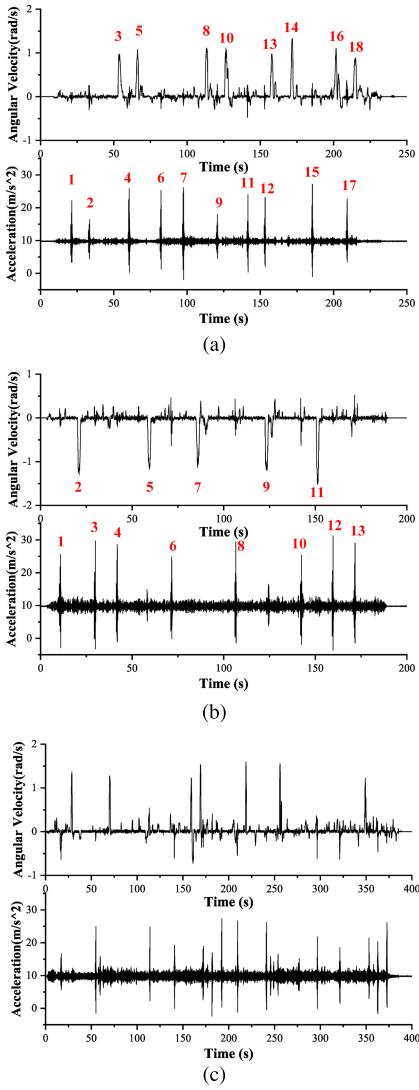


Fig. 8. Activity semantics detection result. (a)–(c) Results of scene #01, scene #02, and scene #03.

the detection accuracy of the two activity semantics reached 100%.

Scene #03 is a complex scenario we set up to evaluate the robustness of our method. There are many potholes and small stones on the road in this scene, which will affect the detection effect of the speed bump. Moreover, there are many cars and moving pedestrians in this scene, so the UGV must perform operations such as braking, avoiding, and changing lanes during operation, which will affect the turning detection effect. The UGV actually goes through ten speed bumps and six turns, and our method detects 14 speed bumps and 12 turns. In this scenario, the accuracy of our method is 71.43% and 50.00%, respectively, and the recall both reach 100%.

There are 26 turns and zero speed bumps in the KITTI 00 dataset. The accuracy of our method is 92.86%, and the recall is 100%.

C. Loop Closure Detection

The detection results are shown in Table II. The actual loops in Table II refers to the loops formed by the activity

TABLE II
RESULTS OF LOOP CLOSURE DETECTION

Sequence	Actual Loop	Detected Loop	Accuracy	Recall
#01	6	5	100%	83.33%
#02	4	4	100%	100%
#03	8	7	100%	87.50%
#04	10	10	100%	100%
KITTI 00	7	7	100%	100%

semantics. The UGV passed through six real loops, of which four were formed by speed bumps and two by turns in scene #01. Our loop closure detection algorithm detects five loops. At the last turn, there was actually a loop closure, but it was not detected because its distance from the previous loop closure was less than the threshold d_i . This does not affect the final optimization because the nearby loop has been detected, so this segment was already optimized. In this experiment, the accuracy reached 100%, and the recall reached 83.33%.

In scene #02, The UGV passed through four real loops, of which three were formed by speed bumps and one by turns. Our method detected all four loops. The accuracy and recall both reached 100%.

The UGV passed through eight real loops in scene #03, of which five were formed by speed bumps and three by turns. Although there are many disturbances in activity semantic detection, our method excluded those disturbances and obtained good results, detecting seven loops. The last loop was not detected because it was closer to the end and there were no keyframes there. In scene #03, the accuracy of our method was 100%, and the recall was 87.5%.

In scene #04, The UGV passed through ten real loops, of which three were formed by speed bumps and seven by turns. Our method detected all ten loops. The accuracy and recall both reached 100%.

There are actually seven loop closures in KITTI 00, two of which are reverse loops. Our detection accuracy and recall were both 100%.

D. Graph Optimization

We choose three classical algorithms, A-LOAM, LeGO-LOAM, and SC-LeGO-LOAM, to compare with our method. Among these methods, the A-LOAM and LeGO-LOAM algorithms have no loop optimization module, while SC-LeGO-LOAM has a loop optimization module named the scan context [14]. LeGO-LOAM uses IMU data to improve accuracy. In scene #01, because of the indoor environment, we cannot obtain RTK signals, so we cannot obtain ground truth. Therefore, the evaluation index we choose is the ATE. Similarly, scene #03 also includes some indoor environments, so we also used ATE for evaluation. In scene #02 and the KITTI 00 dataset, we use the RTK trajectory as the ground truth, so the absolute pose error (APE) is used as the evaluation index. The APE includes maximum error, minimum error, and so on.

To further evaluate the effect of active semantics, we use the ICP method for comparison, and combined a new method. The front-end module of this method uses the

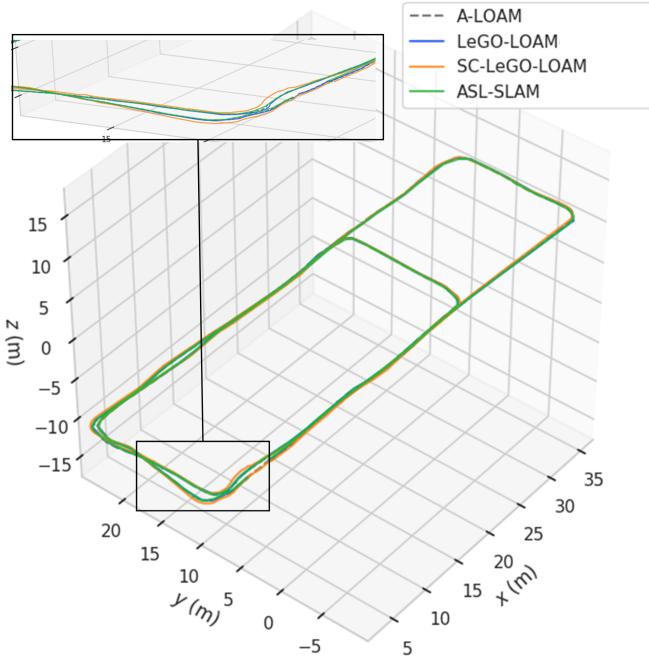


Fig. 9. Comparison results of scene #01.

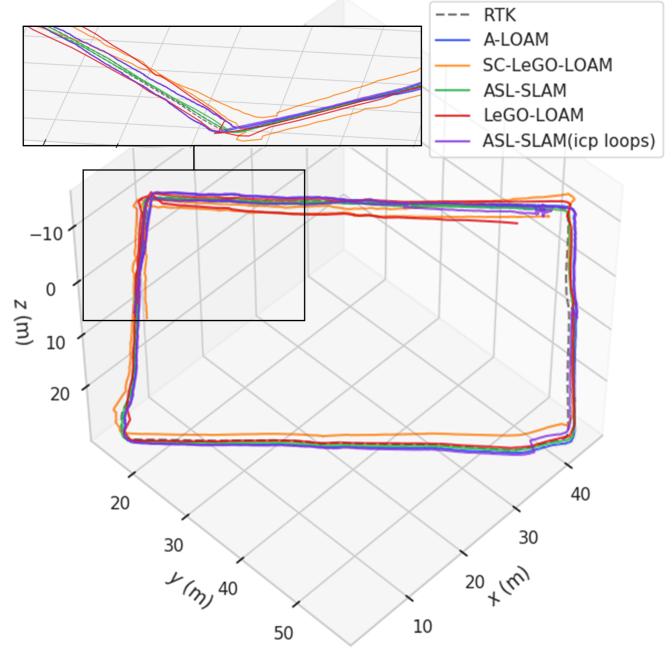


Fig. 10. Comparison results of scene #02.

ASL-SLAM front-end, but the back-end module is replaced by an ICP-based loop closure detection method. This method fails in scene #01. Because there are many similar structures in this scene, some false loop closures occurred when using the point cloud for loop closure detection. These false loop closures lead to global optimization failures. In scene #02, the results obtained by this method is worse than when the loop closure module is not added. In the loop obtained by the ICP-based method, sometimes the actual positions of the two frames of point clouds are too far apart, which will cause the error of the matching calculation results to increase, and finally lead to an increase in the global error. In scene #03, this method has a small optimization effect compared to the ASL-SLAM method without a backend module. On the KITTI 00 dataset, this method has no optimization effect.

Fig. 9 shows the trajectory comparison of scene #01. The results are shown in Table III. In Table III, the last column in Table III represents the optimization degree of ASL-SLAM with loops compared to ASL-SLAM without loops. The results show that our back-end optimization method can better correct the trajectory and reduce the error. Even if the original front-end trajectory accuracy is very high, the accuracy will be improved after our optimization method is applied. Compared with the other three methods, our method has better accuracy before adding the loop closure optimization module. After adding the module, the ATE is further reduced by 78.71%. At the same time, Fig. 9 shows that the other methods have large errors in the z-axis, but our method corrects this error. Because its accuracy was already high before optimization, we do not show the comparison results of the point cloud.

The comparison trajectories between our algorithm and the other algorithms for scene #02 are shown in Fig. 10. The errors of the other algorithms mainly appear on the z-axis and are

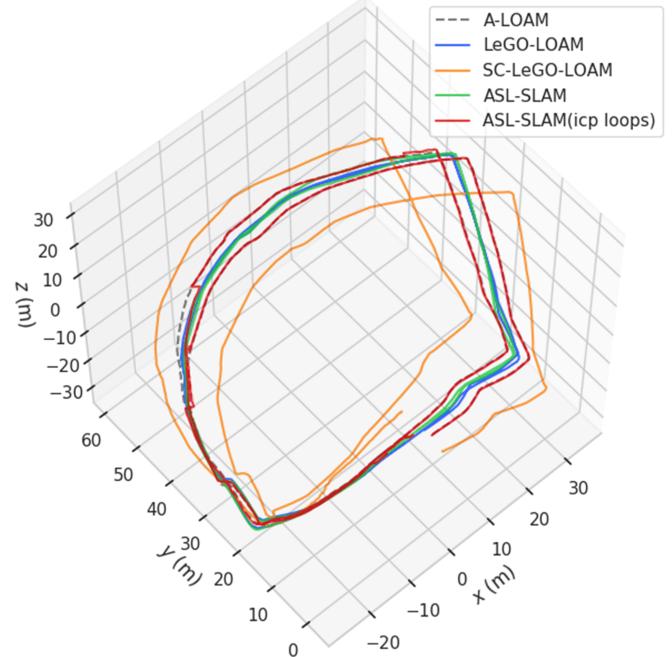


Fig. 11. Comparison results of scene #03.

most obvious at turns. The first cycle trajectory and the second cycle trajectory of the other algorithms are not on the same plane, and their coincidence with the RTK trajectory is poor. The trajectory of our method has the highest coincidence with the RTK trajectory.

The comparison results for scene #03 are shown in Fig. 11. As seen in Fig. 11, the gap between the results is quite striking. The trajectory results of the other algorithms are not closed, but the trajectory of our method is well closed, and the highest accuracy is achieved. This demonstrates that our approach

TABLE III
COMPARISON OF ERROR RESULTS

Sequence	Distance (m)	Error Type	A- LOAM	LeGO- LOAM	SC-LeGO- LOAM	ASL-SLAM (icp loops)	ASL-SLAM (without loops)	ASL-SLAM (semantic loops)	Improvement		
#01	237.065	ATE(m)	0.213	0.088	0.301	Failed	0.155	0.033	78.71%		
		Max(m)	1.772	2.570	3.284	2.281	1.785	0.775	56.58%		
		Mean(m)	0.924	1.440	1.921	1.033	0.732	0.362	50.55%		
		Median(m)	0.927	1.444	1.815	0.982	0.614	0.316	48.53%		
		Min(m)	0.136	0.128	0.582	0.073	0.078	0.041	47.44%		
		RMSE	0.990	1.609	1.989	1.114	0.830	0.398	52.05%		
#02	261.285	Std	0.356	0.717	0.514	0.415	0.392	0.166	57.65%		
		#03	438.325	ATE(m)	6.816	1.283	11.420	6.017	7.308	0.371	94.92%
		#04	537.478	ATE(m)	13.211	20.090	16.841	6.536	7.727	1.797	76.74%
		Max(m)	8.110	10.135	12.246	6.750	6.755	6.632	1.82%		
		Mean(m)	2.694	3.536	3.686	1.967	1.976	1.953	1.16%		
KITTI 00	3724.187	Median(m)	2.368	2.762	3.603	1.826	1.717	1.621	5.59%		
		Min(m)	0.212	0.246	0.369	0.356	0.354	0.336	5.08%		
		RMSE	3.083	4.135	4.230	2.341	2.360	2.340	0.85%		
		Std	1.498	2.311	2.074	1.269	1.290	1.255	2.71%		

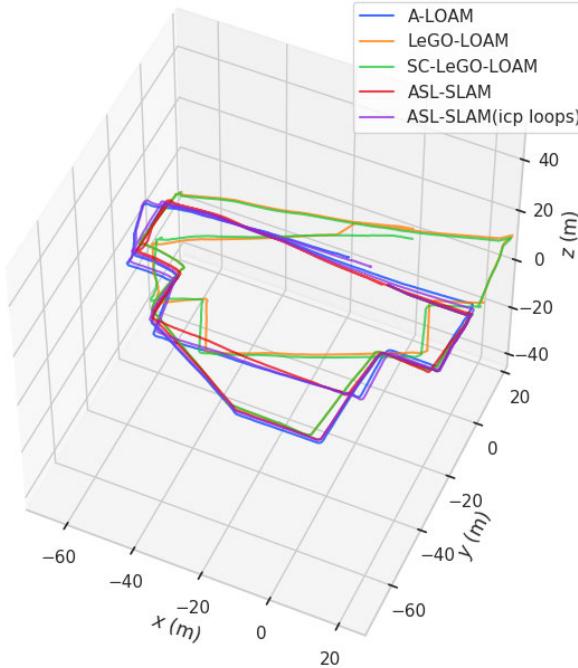


Fig. 12. Comparison results of scene #04.

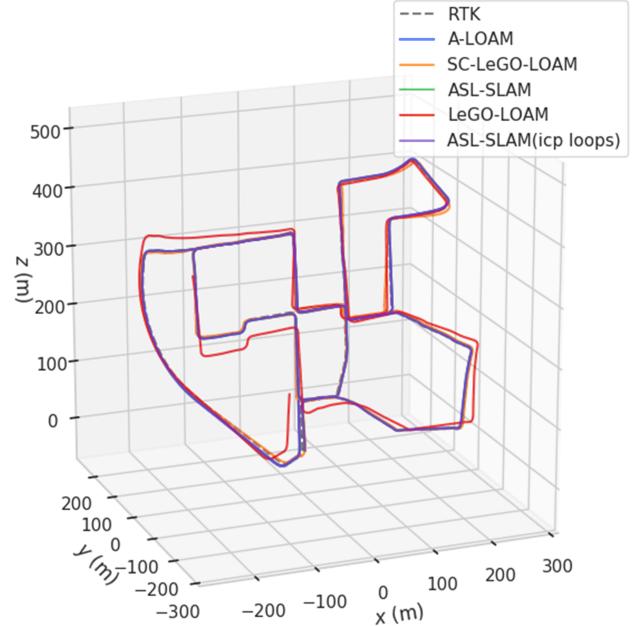


Fig. 13. Comparison results of KITTI 00.

can achieve high-precision results in complex indoor-outdoor scenarios and overcome complex road environments such as continuous curved turns, multiple bumps, and frequent obstacle avoidance.

The comparison results for scene #04 are shown in Fig. 12. Other algorithms have large errors, especially LEGOLoAM and A. Their trajectories are shifted a lot in the middle of the scene because of the large number of turns in the middle of the scene. Frequent turns will lead to larger errors. Moreover, the structures of these turns are similar, so the scan context loop closure detection method does not work. The loop closure detection method based on ICP does not have a good effect either. The best effect is the ASL-SLAM method.

We also selected sequence 00 in the KITTI dataset for evaluations. The results are shown in Fig. 13. In sequence 00, the largest errors in the results of the front-end odometry

are mainly generated in the last segment, but the car does not generate activity semantics on this section of the road, so loop closure based on activity semantics cannot be detected at the end of the odometry. Our loop closure detection method only detects loops in the middle of the odometry, but the accuracy of odometry in this area is already very high. Therefore, the effect of global optimization is not obvious compared to the odometry. Nevertheless, our results are still much higher than those of the SC-LeGO-LOAM method based on loop closure detection with point clouds. The scan context method successfully fixes the last segment of the map where there are dense turns, resulting in a final global result that is worse than ours. It is important to note that the area where the scan context method has a large error is exactly where our method is effective.

Experiments on the KITTI dataset demonstrate that our method can detect correct loop closures and optimize

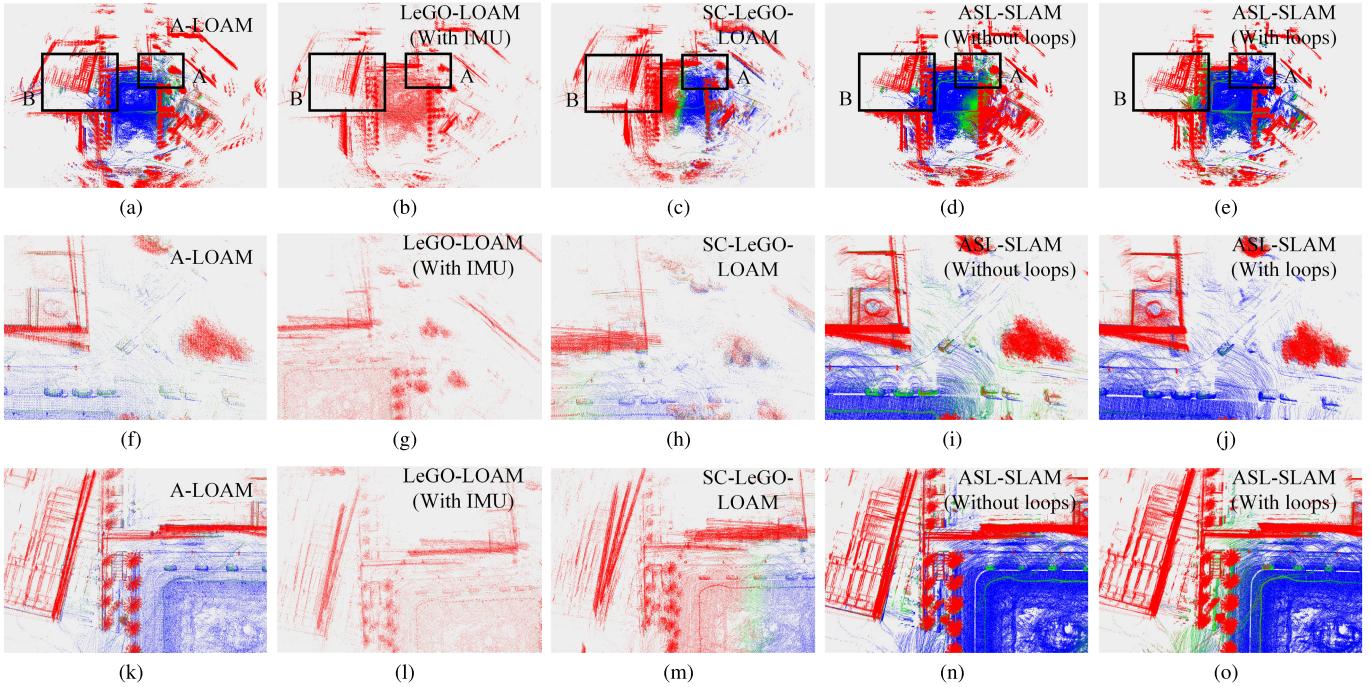


Fig. 14. Point cloud maps of scene #02, (a) is the point cloud map result of A-LOAM, (b) is LeGO-LOAM, (c) is SC-LeGO-LOAM, (d) is ASL-SLAM(Without loops), (e) is ASL-SLAM(With loops). (f) to (j) are the point cloud maps of area A, and (k) to (o) are the point cloud maps of area B.

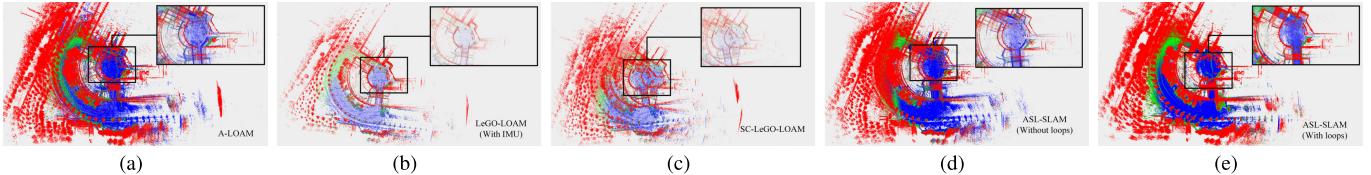


Fig. 15. Point cloud maps of scene #03, (a) is the point cloud map of A-LOAM, (b) is LeGO-LOAM, (c) is SC-LeGO-LOAM, (d) is ASL-SLAM(Without loops), and (e) is ASL-SLAM(With loops).

successfully in large and complex urban road environments. However, the number of loop closures detected in this scene is less compared to other scenes. If a loop closure is detected at a place where the accuracy of the front-end odometry is high, the effect after optimization will not be as obvious as in other scenarios. Even so, our method can achieve higher global accuracy than other methods.

The trajectories obtained using the other algorithms will produce larger errors, and the trajectories obtained using our method have the highest degree of coincidence.

After global optimization, the quality of the point cloud map is also greatly improved. The ASL-SLAM point cloud maps and other algorithms are shown in Figs. 14 and 15. Fig. 14 is the result of scene #02, and Fig. 15 is the result of scene #03. Because scene #01 is an indoor space, the visualization of its point cloud is not good, so it is not shown.

In Fig. 14, we can see that the point cloud map optimized using ASL-SLAM is the clearest. In Fig. 14(a)–(e), the outline of many objects show ghosting, and many objects are very blurred, which is caused by errors between multiple scans. There is obvious difference in area A and area B. In Fig. 14(f) and (i), we can see that there are two red boundaries in the buildings, and the cars parked on the roadside show ghosting, similar to two cars parked side by side. Because of the large error, there are multiple boundaries in the buildings,

and multiple ghosting events in the roadside cars are shown in Fig. 14(h). In contrast, we can see the clear building and car boundaries in Fig. 14(j). We can also see that the building has two boundaries, and the tree crowns are not clear enough in Fig. 14(k) and (n). In Fig. 14(m), there are three boundaries in the building and many ghosts in the tree crown. In contrast, Fig. 14(o) is clear in these places.

In Fig. 15, we can see a more pronounced difference in point cloud accuracy. Many border ghosts can be seen near the circular border in the middle of Fig. 15(a)–(d), but these borders are clear in Fig. 15(e).

In conclusion, the global consistency of the point cloud map obtained using ASL-SLAM is higher than that obtained using the other algorithms. This also proves that the proposed loop cloud detection and graph optimization methods can improve the accuracy of the point cloud map.

The above four representative scenarios and the KITTI dataset demonstrate that the ASL-SLAM proposed in this article can operate successfully and can significantly reduce the ATE and APE values in structured environments such as parking lots and parks. In these structured scenarios, the accuracy of global positioning and point cloud mapping can reach approximately 20 cm. It also proves that ASL-SLAM can not only be successfully optimized in complex large-scale urban road environments, but also achieve higher positioning

accuracy and global point cloud accuracy than other methods in these environments. In general, ASL-SLAM ensures the positioning accuracy and consistency of the point cloud map.

V. CONCLUSION

In this article, we propose an LiDAR SLAM with activity semantics-based loop closure, which first uses the NDT-LOAM algorithm as a front-end module, then uses active semantics to detect loop closures, and finally carries out graph optimization and global map construction. The main innovation of this article is the novel loop closure detection method using activity semantics. The detection of activity semantics uses the acceleration and angular velocity data from an IMU to detect the two activity semantics of turning and passing over a speed bump. After using our UGVs to conduct experiments, we obtain the following conclusions. Compared with A-LOAM, LeGO-LOAM, and SC-LeGO-LOAM, our method with loop closure detection and global graph optimization achieved much better performance, including better point cloud maps, APE, and ATE values.

Our method also has shortcomings. In complex large-scale urban road environments, our method detects fewer loop closures. Therefore, future work will extract more types of activity semantics and optimize loop closure detection methods.

REFERENCES

- [1] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): Part II,” *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.
- [2] C. Cadena et al., “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [3] J. Han, M. Rong, H. Jiang, H. Liu, and S. Shen, “Vectorized indoor surface reconstruction from 3D point cloud with multistep 2D optimization,” *ISPRS J. Photogramm. Remote Sens.*, vol. 177, pp. 57–74, Jul. 2021.
- [4] C. Wang et al., “Semantic line framework-based indoor building modeling using backpacked laser scanning point cloud,” *ISPRS J. Photogramm. Remote Sens.*, vol. 143, pp. 150–166, Sep. 2018.
- [5] S. Jia, C. Liu, H. Wu, D. Zeng, and M. Ai, “A cross-correction LiDAR SLAM method for high-accuracy 2D mapping of problematic scenario,” *ISPRS J. Photogramm. Remote Sens.*, vol. 171, pp. 367–384, Jan. 2021.
- [6] F. Jiang, J. Chen, and S. Ji, “Panoramic visual-inertial SLAM tightly coupled with a wheel encoder,” *ISPRS J. Photogramm. Remote Sens.*, vol. 182, pp. 96–111, Dec. 2021.
- [7] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” *Proc. SPIE*, vol. 1611, pp. 586–606, Apr. 1992.
- [8] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Oct. 2003, pp. 2743–2748.
- [9] J. Zhang and S. Singh, “LOAM: LiDAR odometry and mapping in real-time,” in *Proc. Robot. Sci. Syst.*, vol. 2, no. 9, Berkeley, CA, USA, 2014, pp. 1–9.
- [10] J. Zhang and S. Singh, “Low-drift and real-time LiDAR odometry and mapping,” *Auton. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.
- [11] J. Lin and F. Zhang, “Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3126–3131.
- [12] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.
- [13] S. Chen et al., “NDT-LOAM: A real-time LiDAR odometry and mapping with weighted NDT and LFA,” *IEEE Sensors J.*, vol. 22, no. 4, pp. 3660–3671, Feb. 2022.
- [14] G. Kim and A. Kim, “Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4802–4809.
- [15] G. Kim, S. Choi, and A. Kim, “Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments,” *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1856–1874, Jun. 2022.
- [16] H. Wang, C. Wang, and L. Xie, “Intensity scan context: Coding intensity and geometry relations for loop closure detection,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2095–2101.
- [17] L. He, X. Wang, and H. Zhang, “M2DP: A novel 3D point cloud descriptor and its application in loop closure detection,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 231–237.
- [18] M. A. Uy and G. H. Lee, “PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4470–4479.
- [19] T. Ran, L. Yuan, J. Zhang, L. He, R. Huang, and J. Mei, “Not only look but infer: Multiple hypothesis clustering of data association inference for semantic SLAM,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.
- [20] Q. Sun, J. Yuan, and X. Zhang, “IT-HYFAO-VO: Interpretation tree-based VO with hybrid feature association and optimization,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–18, 2021.
- [21] Z. Wang, Y. Shen, B. Cai, and M. T. Saleem, “A brief review on loop closure detection with 3D point cloud,” in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Aug. 2019, pp. 929–934.
- [22] B. Zhou, Q. Li, Q. Mao, W. Tu, and X. Zhang, “Activity sequence-based indoor pedestrian localization using smartphones,” *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 5, pp. 562–574, Oct. 2015.
- [23] B. Zhou et al., “A graph optimization-based indoor map construction method via crowdsourcing,” *IEEE Access*, vol. 6, pp. 33692–33701, 2018.
- [24] J.-E. Deschaud, “IMLS-SLAM: Scan-to-model matching based on 3D data,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2480–2485.
- [25] J. Behley and C. Stachniss, “Efficient surfel-based SLAM using 3D laser range data in urban environments,” in *Proc. Robot., Sci. Syst.*, 2018, p. 59.
- [26] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, “SuMa++: Efficient LiDAR-based semantic SLAM,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4530–4537.
- [27] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, “MULLS: Versatile LiDAR SLAM via multi-metric linear least square,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11633–11640.
- [28] S. Chen, B. Zhou, C. Jiang, W. Xue, and Q. Li, “A LiDAR/visual SLAM backend with loop closure detection and graph optimization,” *Remote Sens.*, vol. 13, no. 14, p. 2720, Jul. 2021.
- [29] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, “Fast and incremental method for loop-closure detection using bags of visual words,” *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1027–1037, Oct. 2008.
- [30] M. Cummins and P. Newman, “FAB-MAP: Probabilistic localization and mapping in the space of appearance,” *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 647–665, 2008.
- [31] Y. Hou, H. Zhang, and S. Zhou, “Convolutional neural network-based image representation for visual loop closure detection,” in *Proc. IEEE Int. Conf. Inf. Autom.*, Aug. 2015, pp. 2238–2245.
- [32] J. Guo, P. V. K. Borges, C. Park, and A. Gawel, “Local descriptor for robust place recognition using LiDAR intensity,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1470–1477, Apr. 2019.
- [33] S. Salti, F. Tombari, and L. Di Stefano, “SHOT: Unique signatures of histograms for surface and texture description,” *Comput. Vis. Image Understand.*, vol. 125, pp. 251–264, Aug. 2014.
- [34] H. Yin, L. Tang, X. Ding, Y. Wang, and R. Xiong, “LocNet: Global localization in 3D point clouds for mobile vehicles,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 728–733.
- [35] X. Chen et al., “OverlapNet: Loop closing for LiDAR-based SLAM,” 2021, *arXiv:2105.11344*.
- [36] Z. Liu et al., “SeqLPD: Sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1218–1223.
- [37] Z. Liu et al., “LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2831–2840.

- [38] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.
- [39] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5297–5307.
- [40] A. Charroud, K. E. Moutouakil, and A. Yahyaouy, "Fast and accurate localization and mapping method for self-driving vehicles based on a modified clustering particle filter," *Multimedia Tools Appl.*, vol. 82, pp. 18435–18457, Nov. 2022.
- [41] A. Charroud, A. Yahyaouy, K. E. Moutouakil, and U. Onyekpe, "Localisation and mapping of self-driving vehicles based on fuzzy K -means clustering: A non-semantic approach," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, May 2022, pp. 1–8.
- [42] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3D point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5266–5272.
- [43] R. Dubé, A. Craciuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: 3D segment mapping using data-driven descriptors," 2018, *arXiv:1804.09557*.
- [44] S. Beauregard and H. Haas, "Pedestrian dead reckoning: A basis for personal positioning," in *Proc. 3rd Workshop Positioning, Navigat. Commun.*, 2006, pp. 27–35.
- [45] D. Gusenbauer, C. Isert, and J. Krösche, "Self-contained indoor positioning on off-the-shelf mobile devices," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat.*, Sep. 2010, pp. 1–9.
- [46] R. Liu et al., "Collaborative SLAM based on WiFi fingerprint similarity and motion information," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1826–1840, Mar. 2020.
- [47] R. Kümmeler, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G²o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3607–3613.



Baoding Zhou (Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2015.

He is currently an Associate Professor with the College of Civil and Transportation Engineering, Shenzhen University, Shenzhen, China. His research interests include indoor localization and mapping, mobile computing, and intelligent transportation.



Chunyu Li received the B.S. degree in traffic engineering from Foshan University, Foshan, China, in 2019. He is currently pursuing the M.E. degree with Shenzhen University, Shenzhen, China.

His main research interests include SLAM and autonomous robots.



Shoubin Chen received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2020.

He currently holds a postdoctoral position with the School of Architecture and Urban Planning, Shenzhen University, Shenzhen, China. His research interests include robot navigation, LiDAR/Visual SLAM, and multisensor fusion.



Doudou Xie received the B.S. degree in traffic engineering from Shandong Jiaotong University, Jinan, China, in 2019. He is currently pursuing the M.E. degree with Shenzhen University, Shenzhen, China.

His main research interests include SLAM and autonomous robots.



Min Yu received the B.Eng. and master's degrees in communication and electronics engineering from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 1985 and 1988, respectively.

She was a Visiting Scholar with the University of California at Irvine, Irvine, CA, USA. She is currently a Professor in Communication, Electronic Engineering, and Computer Science with Jiangxi Normal University, Nanchang, China.

Her research interests include distributed computing, wireless sensor network, and indoor positioning.



Qingquan Li received the Ph.D. degree in photogrammetry and remote sensing from the Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 1998.

For over 20 years, Dr. Li has been engaged in the developments of surveying engineering and their innovation applications, contributing especially to the principles and practices of dynamic precision engineering survey. His research interests include the integration of GNSS, RS, and GIS; precision engineering survey; and spatiotemporal big data analytics.

Dr. Li is an Academician of IEAS, a member of the Committee at Science and Technology Commission of Ministry of Education, China, and the Chief-Editor of the *Journal of Geodesy and Geoinformation Science*. His work has been recognized with a variety of distinctions. His research was awarded the National Scientific and Technological Progress Second Prize (China) and the MoE Scientific and Technological Progress First Prize (China).