Autonomous Robots (2022) 46:535–551
https://doi.org/10.1007/s10514-022-10037-w

Semantic scan context: a novel semantic-based loop-closure method for LiDAR SLAM

Lin Li[1] · Xin Kong[1] · Xiangrui Zhao[1] · Tianxin Huang[1] · Yong Liu[1]

Received: 10 August 2021 / Accepted: 25 February 2022 / Published online: 4 April 2022

Abstract
As one of the key technologies of SLAM, loop-closure detection can help eliminate the cumulative errors of the odometry. Many of the current LiDAR-based SLAM systems do not integrate a loop-closure detection module, so they will inevitably suffer from cumulative errors. This paper proposes a semantic-based place recognition method called Semantic Scan Context (SSC), which consists of the two-step global ICP and the semantic-based descriptor. Thanks to the use of high-level semantic features, our descriptor can effectively encode scene information. The proposed two-step global ICP can help eliminate the influence of rotation and translation on descriptor matching and provide a good initial value for geometric verification. Further, we built a complete loop-closure detection module based on SSC and combined it with the famous LOAM to form a full LiDAR SLAM system. Exhaustive experiments on the KITTI and KITTI-360 datasets show that our approach is competitive to the state-of-the-art methods, robust to the environment, and has good generalization ability. Our code is available at:https://github.com/lilin-hitcrt/SSC.

Keywords Loop-closure · 3D scene understanding · Semantic feature · SLAM

1 Introduction

Simultaneous Localization and Mapping (SLAM) has rapidly developed in recent decades as critical technologies for autonomous vehicles and robots. Loop-closure detection represents the ability of robots to recognize previously visited places, which can build global constraints for the SLAM system to eliminate the odometry's cumulative errors and establish a globally consistent map (Angeli et al. 2008). Loop-closure detection is usually conducted by using images or point clouds. After a long period of research on image-based loop-closure detection, many successful methods (Negre Carrasco et al. 2016; Muhammad et al. 2019; Han et al. 2018) have been proposed. Since point cloud data is rarely affected by environmental factors such as illumination and seasonal changes, LiDAR-based methods have received widespread attention in recent years.

✉ Yong Liu
yongliu@iipc.zju.edu.cn

Lin Li
22032043@zju.edu.cn

[1] Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, People's Republic of China

Most existing works on LiDAR-based loop-closure detection are achieved by encoding the point cloud into global or local descriptors and then matching the descriptors. They usually use low-level features such as coordinates (Johnson and Hebert 1999; Kim and Kim 2018; Kim et al. 2019; He et al. 2016; Yin et al. 2018), normal (Chen et al. 2020), reflection intensity (Cop et al. 2018; Guo et al. 2019; Wang et al. 2020a; Chen et al. 2020, 2021), etc. In recent years, with the development of the deep learning, many LiDAR-based object detection (Shi et al. 2020) and semantic segmentation (Zhu et al. 2021; Tang et al. 2020) methods have been proposed, making it possible to obtain semantic information from point clouds. However, there are still only a few LiDAR-based works trying to use semantic information (Kong et al. 2020; Zhu et al. 2020; Chen et al. 2020, 2021).

In the field of visual SLAM, the loop-closure detection based on local features (Bay et al. 2006; Rublee et al. 2011) and Bag-of-Words (BoW) (Galvez-López and Tardos 2012)have been very mature and have been widely used (Mur-Artal and Tardós 2017; Qin et al. 2018). Unlike images containing rich texture features, point clouds are almost pure geometric information, making loop-closure detection based on point clouds challenging. As a result, there are few effective methods integrated into LiDAR SLAM sys-

tems yet. The odometry and mapping algorithms represented by LOAM (Zhang and Singh 2017) have achieved very high accuracy on the KITTI dataset (Geiger et al. 2013). However, due to the lack of a back-end optimization module, these methods cannot establish global pose constraints, so they will inevitably suffer from cumulative errors.

In this paper, we propose a semantic-based place recognition method called Semantic Scan Context (SSC). Unlike most existing works that encode low-level features such as coordinates, normals, and reflection intensity as local or global descriptors, we explore the use of high-level semantic information to represent scenes more effectively. Our method mainly consists of two parts, which are the two-step global ICP and the semantic aided descriptor. The two-step global ICP can estimate the 3-DOF pose (x, y, yaw) between the point cloud pairs without initial prior. This pose is used to eliminate the effects of rotation and translation in the descriptor matching stage. To verify the performance of our algorithm, we integrate the proposed approach into LOAM to build a full LiDAR SLAM system. We combine the odometry information and the similarity between the descriptors to detect loop-closure candidates. Then we perform geometric verification to reduce mismatches. The 3-DOF pose obtained by the global ICP provides a good initial value for the geometric verification. After detecting the loop-closures, we add constraints to the pose graph and optimize it. We conduct extensive experiments on the KITTI and KITTI-360 (Liao et al. 2021) datasets to verify the effectiveness of the proposed method. Figure 1 is a demonstration of our results. The main contribution is summarized as follows:

- We propose a semantic-based place recognition method called Semantic Scan Context, which explores high-level semantic information to represent scenes more effectively.
- We propose a global ICP to estimate the 3-DOF pose (x, y, yaw) between the point cloud pairs. The pose is used to eliminate the influence of rotation and translation during descriptor matching and provide initial values during geometric verification.
- We combine the odometry information and the similarity between descriptors to detect loop-closure candidates.
- Our method can help the state-of-the-art LiDAR SLAM system eliminate cumulative errors and build a globally consistent map.
- Exhaustive experiments on the KITTI and KITTI-360 datasets show that our approach is competitive to the state-of-the-art methods, robust to the environment, and has good generalization ability.

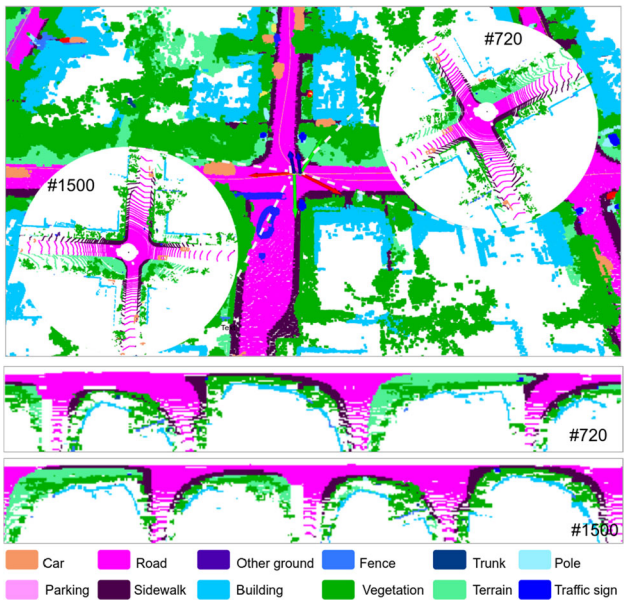


Fig. 1 An example of loop-closure detection using semantic scan context. It is a partial map of KITTI sequence 08, where frames 720 and 1500 form a reverse loop. The lower part of the figure is the semantic scan context corresponding to the two frames. Since the directions of them are opposite, the descriptors are quite different, while the aligned one shown in Fig. 2 is easy to distinguish

2 Related works

According to the features used, we can divide the loop-closure detection methods into three categories: geometry-based, semi-semantic-based, semantic-based.

Geometry-based methods: Spin image (Johnson and Hebert 1999) establishes a local coordinate system for each point, then projects the point into the 2D space and counts the number of points in different areas in the 2D space to form a spin image. ESF (Wohlkinger and Vincze 2011) proposes a shape descriptor that combines angle, point-distance, and area to boost the recognition rate. Röhling et al. (Röhling et al. (2015)) encode the distance between the point and the robot as a histogram. M2DP (He et al. 2016) projects the point cloud into multiple 2D planes and generates a density signature for each plane's points. The left and right singular vectors of those signatures are used as the global descriptors. Scan context (Kim and Kim 2018; Kim et al. 2019) converts the point cloud to polar coordinates and then divides it into blocks along the azimuth and radial directions. Lastly, it encodes the z coordinate of the highest point in each block as a 2D global descriptor. LocNet (Yin et al. (2018)) divides a point cloud into rings, generates a distance histogram for each ring, and stitches all histograms to form a global descriptor. Then a siamese network is used to score the similarity between the descriptors. LiDAR Iris (Wang et al. (2020b)) extracts a binary signature image for each point cloud then uses the

Hamming distance of two corresponding binary signature images as the similarity. Seed (Fan et al. (2020)) segments the point cloud into different objects and encodes the topological information of the segmented objects into the global descriptor. The above methods have achieved good results by encoding low-level geometric structures into descriptors. Integrating more advanced features can further enhance the discriminative power of descriptors.

Semi-semantic-based methods: Some methods use non-geometric information to construct descriptors, such as reflection intensity or learning-based features. These features are related to the object type but do not clearly indicate the semantic category, so we classify these methods as semi-semantic based. ISHOT (Guo et al. 2019) and ISC (Wang et al. 2020a) exploit the intensity information of the point cloud for loop-closure detection. SegMatch (Dubé et al. 2017) and SegMap (Dubé et al. 2019) cluster a point cloud into segments. Then they extract features for each segment and use the kNN algorithm to identify correspondences. Based on SegMatch, LOL (Rozenberszki and Majdik 2020) proposes a method to correct the accumulated drift of the LiDAR-only odometry. LLOAM (Ji et al. 2019) propose a complete 3D LiDAR-based SLAM system by combining LOAM with a SegMatch-based loop-closure detection module. PointNetVLAD (Uy and Lee 2018) combines PointNet (Qi et al. 2017) and NetVLAD (Arandjelovic et al. 2016) to extract global descriptors from the 3D point clouds. L^3-Net (Lu et al. 2019) selects key points from the given point cloud then uses a PointNet to learn local descriptors for each key point. OREOS (Schaupp et al. 2019) projects the 3D point cloud into a 2D range image and proposes a convolutional neural network to extract the global descriptor. DH3D (Du et al. 2020) designs a siamese network to learn 3D local features from the raw 3D point clouds, then use an attention mechanism to aggregate these local features as the global descriptor. LPD-Net (Liu et al. 2019) proposes the adaptive local feature extraction module and the graph-based neighborhood aggregation module to extract local features of the point cloud; then, as the PointNetVLAD, they use the NetVLAD to generate the global descriptor. MinkLoc3D (Komorowski 2021) uses a sparse voxelized point cloud representation and sparse 3D convolutions to compute a discriminative 3D point cloud descriptor. SeqSphereVLAD (Yin et al. (2020)) projects the point cloud onto a spherical view, extracts features on it and sequences those features to form a descriptor. SpoxelNet (Chang et al. (2020)) voxelized the point cloud in spherical coordinates and defines the occupancy of each voxel in ternary values. Then they use a neural network to extract the global descriptor. The above methods combine more advanced features with geometric features. However, most of them use neural networks to extract abstract features, which are more complicated and not well interpretable.

Semantic-based methods: SGPR (Kong et al. 2020) represents the scene as a semantic graph then score their similarity through a graph similarity network. GOSMatch (Zhu et al. 2020) proposes a new global descriptor that is generated from the spatial relationship between semantics. It also proposes a coarse-to-fine strategy to efficiently search loop-closures and gives an accurate 6-DOF initial pose. The two methods represent the scene as a graph and abstract the object as a node in the graph, which will cause the loss of features such as the size of each object. OverlapNet (Chen et al. 2020, 2021) designed a deep neural network that uses different types of information, such as intensity, normal, and semantics generated from LiDAR scans, to provide overlap and relative yaw angle estimates between paired 3D scans. However, it is too slow in preprocessing due to the need to calculating the normal and inferring the complex network backbone. To use the semantic information more effectively, we propose our Semantic Scan Context approach.

3 Methodology

In this section, we present our semantic scan context approach. Different from other scan context-based methods that use incomplete semantic information and ignore small translations between point clouds, we explore to exploit full semantic information and emphasize that the small translation between point cloud pairs has a significant influence on the accuracy of recognition.

As shown in Fig. 2, our method consists of two main parts: two-step global semantic ICP and Semantic Scan Context. The two-step global semantic ICP is divided into Fast Yaw Angle Calculate and Fast Semantic ICP. First, we define a point cloud frame as $P = \{p_1, p_2, \cdots, p_n\}$, with each point $p_i = [x_i, y_i, z_i, \eta_i]$, η_i represent the semantic label of p_i. Given a pair of point clouds (P_1, P_2), we first use our Fast Yaw Angle Calculate method to get the relative yaw angle θ between them. Then we use the Fast Semantic ICP to calculate their relative translation $(\Delta x, \Delta y)$ in the x-y plane. Through the above two steps, we get the relative poses $(\Delta x, \Delta y, \theta)$ of the two frames of point clouds in the 2D subspace pose. In order to eliminate the influence of rotation (e.g., reverse loop-closures) and small translation on recognition, we use the obtained relative pose to align point cloud P_2. We mark the aligned point cloud as P_a. Finally, we use our global descriptor – the Semantic Scan Context to describe (P_1, P_a) as (S_1, S_2). The similarity score is obtained by comparing S_1 and S_2.

3.1 Global semantic ICP

It is known that the general ICP algorithm based on local iterative optimization is susceptible to local minimums (Yang

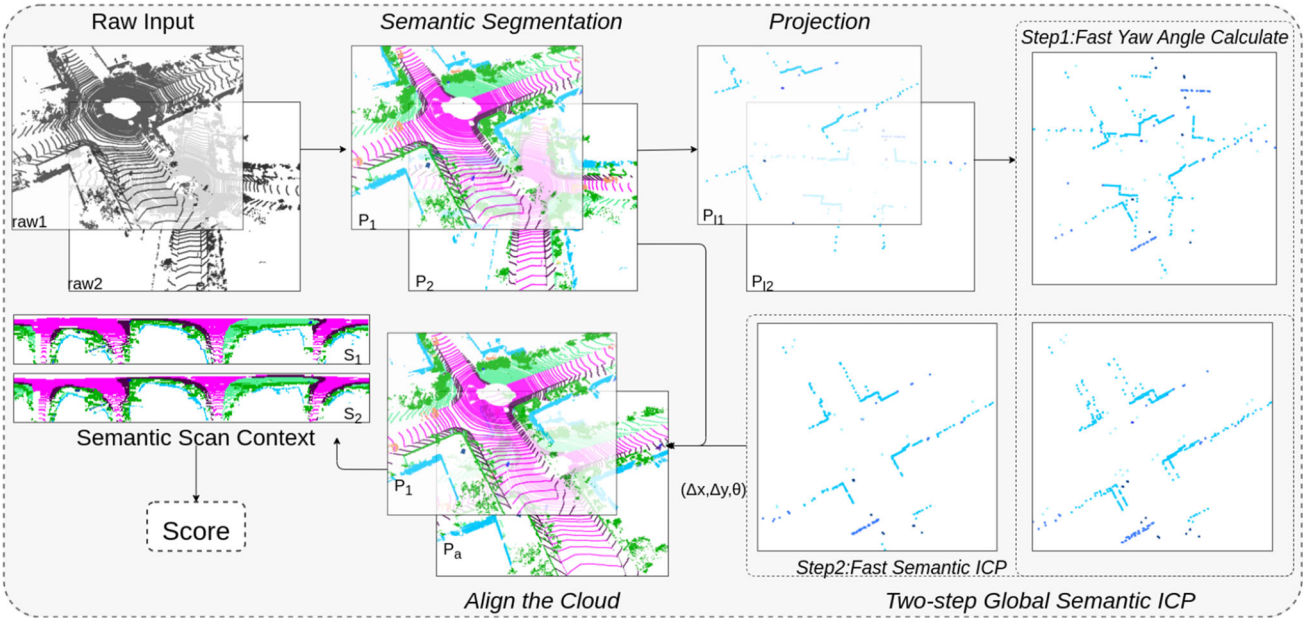


Fig. 2 The pipeline of our approach. It mainly consists of two parts: two-step global semantic ICP and Semantic Scan Context. First, we conduct semantic segmentation on the raw point cloud. Then we use semantic information to retain representative objects and project them onto the x-y plane. The two-step global semantic ICP is performed on the projected cloud to get the 3-DOF pose $(\Delta x, \Delta y, \theta)$. Finally, we use the 3-DOF pose to align the original clouds and generate global descriptors (Semantic Scan Context). The similarity score is obtained by matching SSC

et al. 2016). For loop-closure detection, we usually cannot get a valid initial value, which leads to the failure of the general ICP algorithm. To solve this, we propose the two-step global semantic ICP algorithm consisting of Fast Yaw Angle Calculate and Fast Semantic ICP. Benefited from the use of semantic information, our algorithm does not require any initial values to get satisfactory results.

Fast Yaw Angle Calculate. For scan context based methods, columns of their descriptor represent the yaw angle. The pure rotation of the LiDAR in the horizontal plane will cause the column shift of their descriptor. Scan context and Intensity Scan Context get the similarity score and the yaw angle at the same time. Specifically, they calculate similarity (or distance) with all possible column-shifted descriptors and find the maximum similarity (or minimum distance). However, there are two main disadvantages. Firstly, it's inefficient to compare the whole 2D descriptors by shifting. Secondly, they still try to get the maximum score for point clouds from different places (not loop-closure). This obviously makes it more prone to false positives. To draw the above issues, we propose the semantic-based fast yaw angle calculate method.

Given a point cloud pair (P_1, P_2), we select some representative objects (building, fence, trunk, pole, traffic-sign) based on semantic information. Then we convert the filtered clouds to polar coordinate in the x-y plane:

$$p_i = [r_i, \varphi_i, x_i, y_i, \eta_i]$$

$$\begin{aligned} r_i &= \sqrt{x_i^2 + y_i^2} \\ \varphi_i &= \arctan\left(\frac{y_i}{x_i}\right) \end{aligned} \tag{1}$$

where p_i is the i-th point in each converted cloud, r_i and φ_i represent polar diameter and polar angle, respectively. Each converted cloud is then segmented to N_a sectors by yaw angle. We only keep the point with the smallest polar diameter in each sector. Finally, we get two clouds P_{I1} and P_{I2}, with N_a elements. We sort the points in P_{I1} and P_{I2} according to the azimuth angle and save their corresponding polar diameters as vectors R_1 and R_2. Similar to the scan context, the shift of the column vector is related to the yaw angle:

$$\begin{aligned} \text{shift} &= \underset{i, i\in[0,N_a]}{\text{argmin}}\ \Psi\left(R_1, R_2^i\right) \\ \theta &= 360\left(1 - \frac{\text{shift}}{N_a}\right) \end{aligned} \tag{2}$$

where R_2^i is R_2 shifted by i-th element and Ψ is defined as:

$$\Psi(R_1, R_2^i) = \left\| R_1 - R_2^i \right\|_1 \tag{3}$$

Compared with Scan Context and Intensity Scan Context, our method only needs to compare one-dimensional vectors; therefore, it is more efficient. Moreover, our method does not

obtain the angle via maximizing the score, which is helpful to identify non-loop-closure point-cloud pairs.

Fast Semantic ICP. Though most works ignore translation between point clouds, ignoring the translation causes considerable declines in our experiments. In fact, for methods based on scan context, translation will affect both the row and column of the descriptor. We cannot get the best result just by the column-shifted descriptor. Therefore, we propose a fast semantic ICP algorithm to correct the translation between point clouds.

To find the relative translation, we firstly rotate P_{I2} to the same direction as P_{I1}, and the rotated point cloud is P_{Ia}, which is defined as:

$$\begin{aligned} x_{ai} &= x_i \cos(\theta) - y_i \sin(\theta) \\ y_{ai} &= x_i \sin(\theta) + y_i \cos(\theta) \end{aligned} \tag{4}$$

where (x_i, y_i) and (x_{ai}, y_{ai}) represent the i-th point in P_{I2} and P_{Ia} respectively. Our ICP problem can be defined as:

$$\begin{aligned} (\Delta x, \Delta y) = \underset{\Delta x, \Delta y}{\operatorname{argmin}} L = \underset{\Delta x, \Delta y}{\operatorname{argmin}} \sum_{i=1}^{N_a} \Gamma(\eta_{ai}, \eta_{ri}) \\ \cdot \frac{(x_{ai} + \Delta x - x_{ri})^2 + (y_{ai} + \Delta y - y_{ri})^2}{2} \end{aligned} \tag{5}$$

where (x_{ri}, y_{ri}) represents the corresponding point of (x_{ai}, y_{ai}), which is the point closest to (x_{ai}, y_{ai}) in P_{I1}, η_{ai} and η_{ri} are semantic labels of the points. If η_{ai} is equal to η_{ri}, then the output of $\Gamma(\eta_{ai}, \eta_{ri})$ is 1; otherwise, 0. As our point clouds are ordered, we can search for the corresponding points near the position where the yaw angle is consistent with the target point. Specifically, our search interval for the i-th target point is:

$$\left[i + \text{shift} - \frac{N_l}{2}, i + \text{shift} + \frac{N_l}{2} \right] \tag{6}$$

where N_l is the length of search interval and shift is defined in Eq. (2). After a certain number of iterations, we can get the relative translation between the input point clouds.

3.2 Semantic scan context

Scan Context and Intensity Scan Context uses the points' height and reflection intensity as features, respectively. Their methods essentially take advantage of the different characteristics of different objects in the scene. However, height and reflection intensity is only low-level features of the object which are not representative enough. We explore to use the high-level semantic features to represent scenes and thus propose the Semantic Scan Context descriptor.

Descriptor definition. Given a point cloud P, we first convert it to the polar coordinate system as we did in Sect. 3.1. Then, like scan context, we divide the point cloud into $N_s \times N_r$ blocks along the azimuthal and radial directions. Each block is represented by:

$$\begin{aligned} B_{ij} = \Bigg\{ \eta_k \Bigg| \frac{(i-1) R_{\max}}{N_r} \le r_k < \frac{i R_{\max}}{N_r}, \\ \frac{2\pi(j-1)}{N_s} - \pi \le \varphi_k < \frac{2\pi j}{N_s} - \pi \Bigg\} \end{aligned} \tag{7}$$

where R_{max} is the the maximum effective measurement distance of LiDAR, $i \in [1, N_r]$ and $j \in [1, N_s]$. Our descriptor can be defined by:

$$S(i, j) = f\left(B_{ij}\right) = \underset{\eta \in B_{ij}}{\operatorname{argmax}} E(\eta) \tag{8}$$

f is an encoding function to encode features of B_{ij}. Note that if $B_{ij} = \varnothing$, $f\left(B_{ij}\right) = 0$. Function E is used to evaluate the representative ability of each semantic object. Unlike the global ICP in Sect. 3.1, we use eleven types of semantic objects (road, parking, sidewalk, other-ground, building, fence, vegetation, trunk, terrain, pole, traffic-sign) to construct the descriptor. We manually set the priority of different semantics in function E to show their representativeness. We believe objects that appear less frequently in the scene are more representative (e.g., traffic signs are more representative than roads).

Similarity Scoring. Given aligned clouds P_1 and P_a, we can get their descriptors S_1 and S_2 by Eq. (8). Then the similarity score between them can be calculated by:

$$\text{score}_1 = \frac{\sum_{1 \le i \le N_r} \sum_{1 \le j \le N_s} I(S_1(i, j) = S_2(i, j))}{\sum_{1 \le i \le N_r} \sum_{1 \le j \le N_s} I(S_1(i, j) \ne 0 \text{ or } S_2(i, j) \ne 0)} \tag{9}$$

where I is the indicator function, defined by:

$$I(x) = \begin{cases} 1 & \text{x is true} \\ 0 & \text{x is false} \end{cases} \tag{10}$$

Figure 3 shows Semantic Scan Context creation.

3.3 SLAM system

Although the loop-closures can help eliminate the cumulative error of the SLAM system, the wrong loop-closures will cause serious damage to the reliability of the SLAM system. Therefore, it is often necessary to add additional steps in the actual SLAM system to avoid false loop-closures. We will use the actual system as an example to describe how to apply our method.

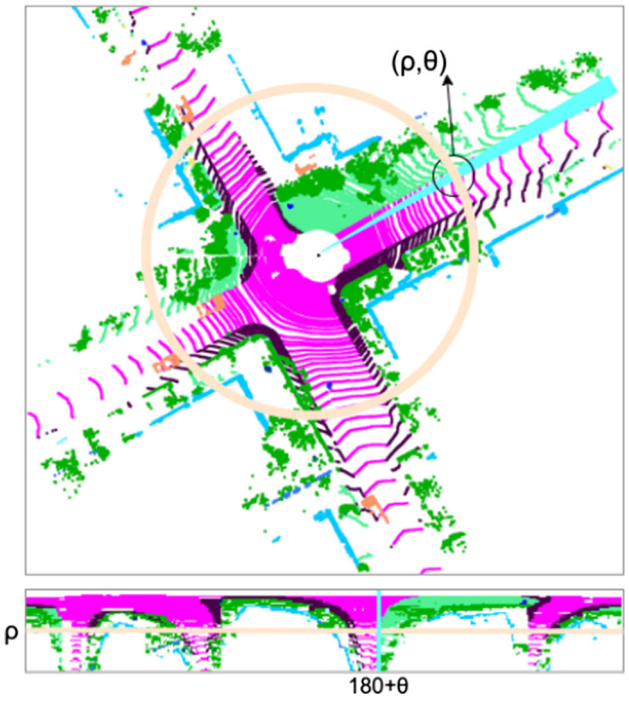


Fig. 3 An example of generating SSC. ρ and θ represent the polar diameter and polar angle, respectively. A sector corresponds to a descriptor column, while a ring corresponds to a row of the descriptor

We integrate our proposed SSC into LOAM, a famous LiDAR-based SLAM system, to test its performance. Figure 4 illustrates how our method works with LOAM.

Candidate selection based on odometry. When the new scan comes, we first use LOAM to calculate the odometry. Then we calculate the global descriptor S_c and range vector R_c from the current point cloud according to the process described in the previous two sections. To avoid redundant calculations, we store S_c and R_c in the database for subsequent use. Like most other methods, we first use the odometry to roughly screen candidates to speed up processing. We set a circular search area with the current position as the center. The radius of the area is defined as follows:

$$R = \zeta D \tag{11}$$

where D is the cumulative distance between the current frame and the target frame, ζ is a manually set constant-coefficient representing the drift rate of the odometry. We take the point cloud scans located in this area as candidates and exclude recent ones.

Similarity verification and coarse pose estimation. We load its corresponding global descriptor S_t and range vector R_t from the database for each possible candidate point cloud. Then we use the global semantic ICP method described in Sect. 3.1 to calculate the 3-DOF pose $(\Delta x, \Delta y, \text{yaw})$ between R_c and R_t. The obtained 3-DOF pose is used to align the descriptors. Finally, we get the similarity score score_1 between the current and target point clouds by comparing the descriptors. Only when score_1 is greater than the threshold α_1 will the subsequent steps be performed.

Geometric verification and pose refinement. We use geometric verification to reduce the probability of false loop-closures further. Specifically, we use LOAM's point cloud registration method to register the current point cloud and the candidate point cloud. Due to the odometry drift, we use the coarse pose obtained in the similarity verification step as the initial pose of the registration. The initial pose is defined as follows:

$$T_0 = \begin{pmatrix} \cos(\text{yaw}) & -\sin(\text{yaw}) & 0 & \Delta x \\ \sin(\text{yaw}) & \cos(\text{yaw}) & 0 & \Delta y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

Finally, we can get the registration score score_2 and refined pose. If score_2 is greater than the threshold α_2, we consider that the current point cloud and the candidate point cloud form a loop-closure. Then we use the refined pose to construct a loop-closure constraint in the pose graph and optimize it to eliminate the cumulative error.

4 Experiments

4.1 Experiment setup

We conduct experiments on the KITTI and KITTI-360 datasets. The point clouds in both datasets are collected by a 64-beam LiDAR (Velodyne HDL-64E). Table 1 shows the details of the two datasets.

KITTI. The KITTI dataset contains 11 training sequences (00-10) with ground truth poses. We choose sequences with loop-closure (00,02,05,06,07,08) for evaluation and note that sequence 08 has reverse loops while others are in the same direction. The ground-truth semantic labels are from the SemanticKITTI dataset (Behley et al. 2019). We also test our method with the semantic segmentation algorithm (RangeNet++ (Milioto et al. 2019)) to prove that our method can be applied to noisy predictions in real situations.

KITTI-360. The KITTI-360 dataset contains 9 training sequences. Similar to the KITTI dataset, we select sequences (0000, 0002, 0004, 0005, 0006, 0009) that contain loop-closures for testing. Unlike the KITTI dataset, each sequence of the KITTI-360 dataset contains a large number of reverse loops, which brings great challenges to the loop-closure detection algorithms. Another difference from the KITTI dataset is that the KITTI-360 dataset does not label single-frame point clouds with semantic information. Instead, it stitches all the point clouds into a map and then annotates

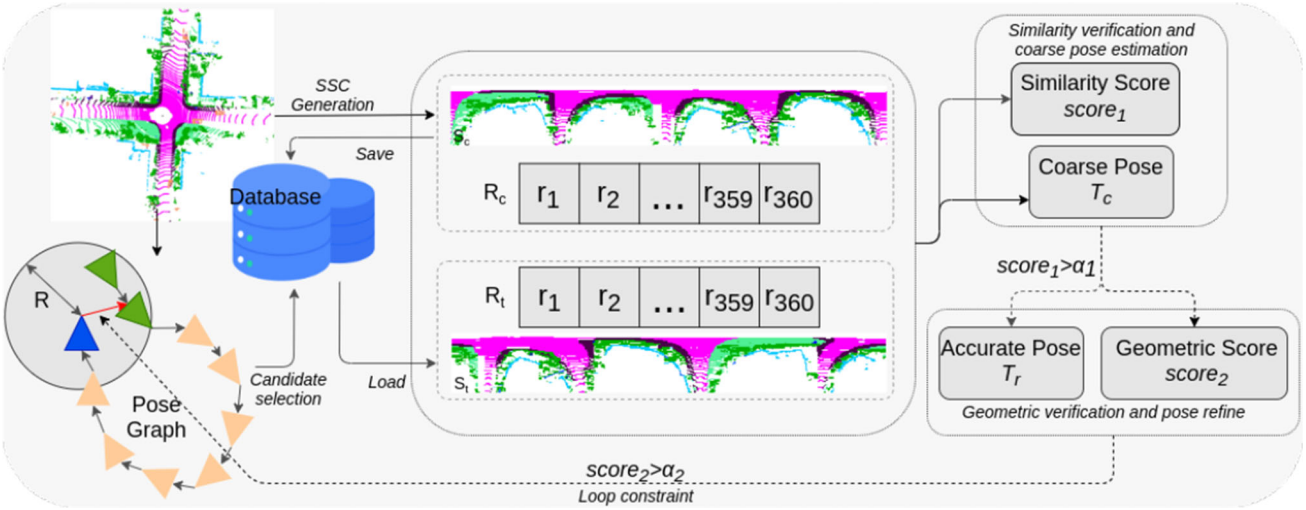


Fig. 4 This figure illustrates how our method works in an actual SLAM system. First, we generate the global descriptor S_c and range vector R_c for the input point cloud and save them in the database for reuse. Then we roughly screen loop candidates according to the odometry. For each candidate, we load its corresponding descriptor S_t and range vector R_t from the database. We use the range vectors and the descriptors to calculate the coarse pose (T_c) and similarity score ($score_1$). If the similarity verification is successful ($score_1 > \alpha_1$), we will further perform geometric verification. Specifically, we use point cloud registration to refine T_c while obtaining geometric similarity $score_2$. If the geometry verification is successful ($score_2 > \alpha_2$), we will establish a loop constraint in the pose graph. Finally, we optimize the pose graph to eliminate the cumulative error of the odometry

Table 1 Statistics of evaluation dataset

	KITTI						KITTI-360					
	00	02	05	06	07	08	00	02	04	05	06	09
Num of scans	4541	4661	2761	1101	1101	4071	10518	19240	11587	11587	9186	13247
Num of loops	7555	1684	4785	1578	1833	1994	19211	19514	11569	15150	19173	39137
Direction	Same	Same	Same	Same	Same	Reverse	Both	Both	Both	Both	Both	Both

The KITTI-360 dataset is more complex than the KITTI dataset. Compared with the KITTI dataset, each sequence of the KITTI-360 dataset contains more loop-closures. What's more, the KITTI-360 dataset includes lots of reverse loop-closures, which brings significant challenges to the loop-closure detection algorithms

the map. To obtain the semantic information of a point in the original point cloud, we use a KD-tree to find the closest point in the map and classify the original point and the found point into the same category.

Similar to SGPR (Kong et al. 2020), we regard the point cloud pair with a relative distance less (greater) than 3m (20m) as a positive (negative) sample. In addition, we excluded positive samples whose collection interval is too short because they cannot characterize the ability of loop-closures detection. Since there are too many negative samples, we only select a part of the negative samples for evaluation. Specifically, if N_p positive samples are in a sequence, we will randomly select $\alpha \cdot N_p$ negative samples. We can adjust the proportion of negative samples by changing the coefficient α. In our experiments, we set $N_a = 360$, $N_l = 20$, $N_s = 360$, $N_r = 50$. All experiments are done on the same system with an Intel i7-9750H @3.00GHz CPU with 16 GB RAM.

4.2 Loop-closure detection performance

As mentioned in Sect. 4.1, we use both ground-truth semantic labels (Ours-SK) and predicted semantic labels (Ours-RN) for testing. We compare our approach with the state-of-the-art methods, including Scan Context (Kim and Kim 2018) (SC), Intensity Scan Context (Wang et al. 2020a) (ISC), M2DP (He et al. 2016), LiDAR Iris (Wang et al. 2020b) (LI), PointNetVLAD (Uy and Lee 2018) (PV), OverlapNet (Chen et al. 2020, 2021) (ON), and SGPR (Kong et al. 2020). For SGPR, we use their pre-trained models trained with the 1-fold strategy. As we cannot reproduce the results of OverlapNet, we use the pre-trained model provided by the author. The model is trained on sequences 03-10, so sequences 05, 06, 07, 08 are included in the training set.

Fixed α. In this experiment, we set α to 100, which means the number of negative samples is $100N_p$. Figure 5 shows the precision-recall curve of each method. Additionally, we also

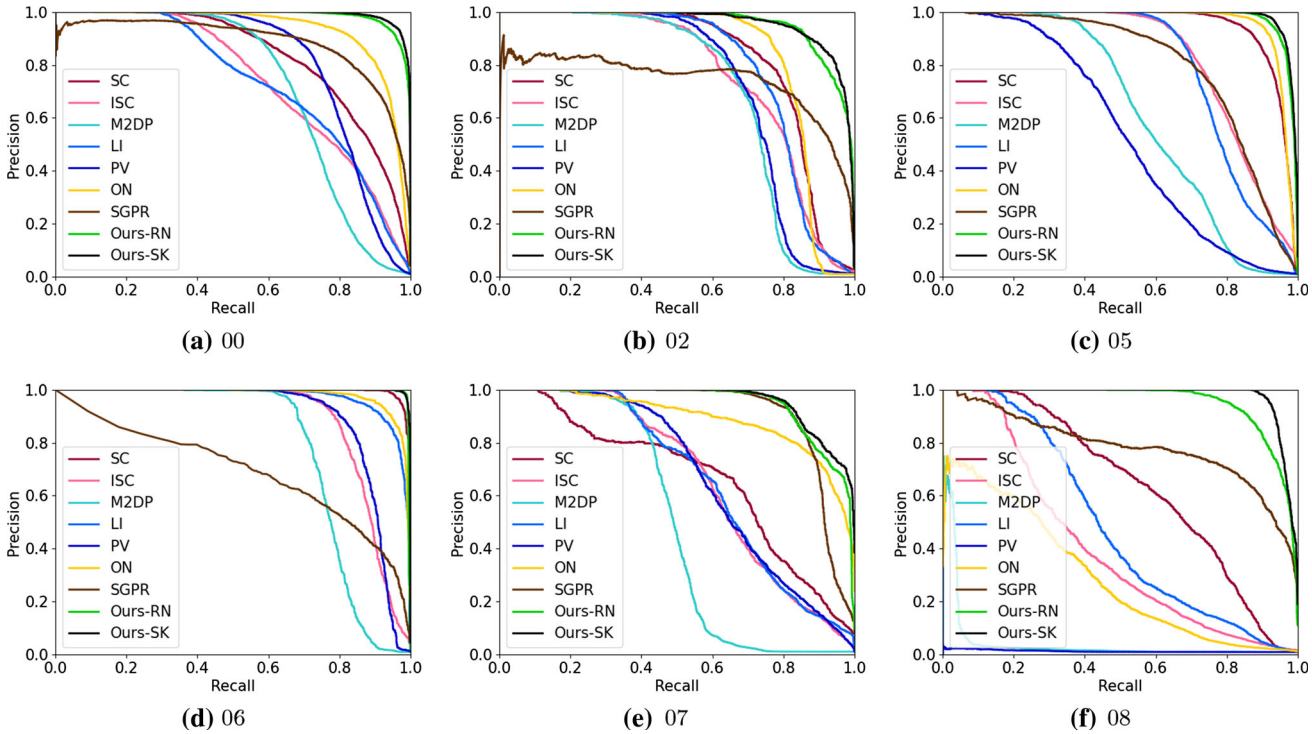


Fig. 5 Precision-Recall curves on KITTI dataset

Table 2 F_1 max scores and Extended Precision on KITTI dataset

Methods	00	02	05	06	07	08	Mean
SC	0.750/0.609	0.782/0.632	0.895/0.797	0.968/0.924	0.662/0.554	0.607/0.569	0.777/0.681
ISC	0.657/0.627	0.705/0.613	0.771/0.727	0.842/0.816	0.636/0.638	0.408/0.543	0.670/0.661
M2DP	0.708/0.616	0.717/0.603	0.602/0.611	0.787/0.681	0.560/0.586	0.073/0.500	0.575/0.600
LI	0.668/0.626	0.762/0.666	0.768/0.747	0.913/0.791	0.629/0.651	0.478/0.562	0.703/0.674
PV	0.779/0.641	0.727/0.691	0.541/0.536	0.852/0.767	0.631/0.591	0.037/0.500	0.595/0.621
ON	0.869/0.555	0.827/0.639	0.924/0.796	0.930/0.744	0.818/0.586	0.374/0.500	0.790/0.637
SGPR	0.820/0.500	0.751/0.500	0.751/0.531	0.655/0.500	0.868/0.721	0.750/0.520	0.766/0.545
Ours-RN	0.939/0.826	0.890/0.745	0.941/0.900	**0.986/0.973**	0.870/0.773	0.881/0.732	0.918/0.825
Ours-SK	**0.951/0.849**	**0.891/0.748**	**0.951/0.903**	0.985/0.969	**0.875/0.805**	**0.940/0.932**	**0.932/0.868**

F_1 max scores and Extended Precision: F_1 max scores / Extended Precision.
The best scores are marked in bold, and the second-best scores are underlined

use the maximum F_1 score and Extended Precision (Ferrarini et al. 2020) (EP) shown in Table 2 to analyze the performance. The F_1 score is defined as:

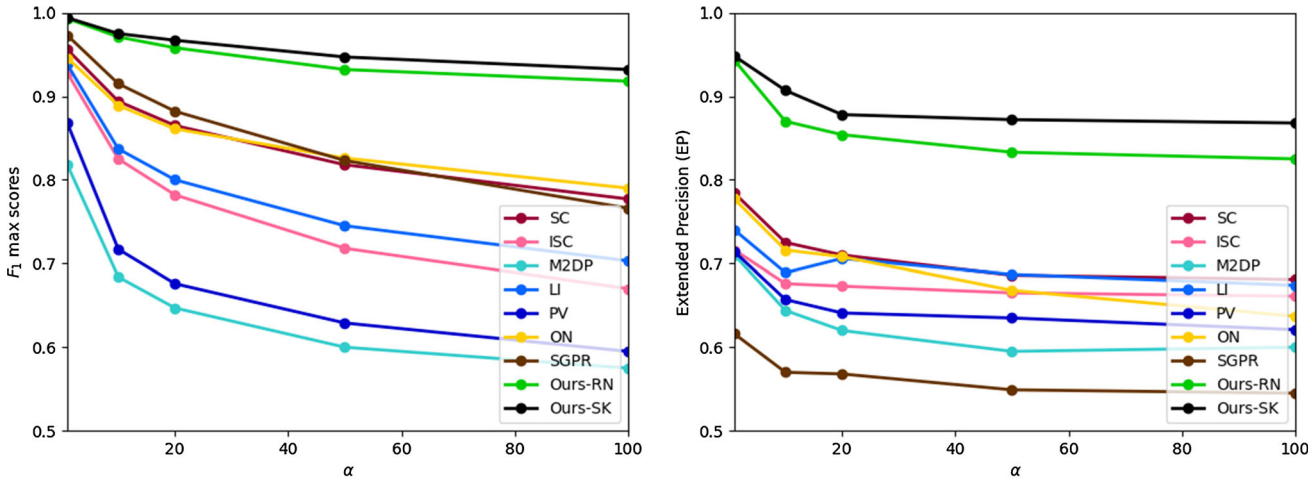
$$F_1 = \frac{2PR}{P+R} \tag{13}$$

where P and R represent the Precision and Recall, respectively; F_1 is the harmonic mean of P and R. It treats P and R as equally important and measures the overall performance of classification. The Extended Precision is defined as:

$$\text{EP} = \frac{1}{2}\left(P_{R0} + R_{P100}\right) \tag{14}$$

where P_{R0} is the precision at minimum recall, and R_{P100} is the max recall at 100% precision. EP is specifically designed metrics for loop-closure detection algorithms.

As shown in Fig. 5 and Table 2, Ours-SK surpasses other methods in all indicators of all sequences with a large margin. Especially in sequence 08, which has only reverse loops, the performance of other methods drops significantly while our method still performs well. This indicates that our method is robust to view angle changes. OverlapNet performs well on most sequences except 08. We guess this is because it uses the normal of the point cloud, which will change as the point cloud rotates. Therefore, this method cannot robustly handle



(a) Average F_1 max scores **(b)** Average EP

Fig. 6 Average F_1 max score and Average Extended Precision corresponding to different α

reverse loops. SGPR works well on indicator the F_1 max score but poorly on the Extended Precision. We find that it gives some negative samples a huge score, which causes the recall to be almost zero when the accuracy reaches 100%. The result of Ours-RN is slightly worse than Ours-SK as expected. As the difference is not obvious, it means that our approach can adapt to semantic segmentation algorithms for actual systems.

Change α. In this experiment, we change the value of α to analyze the influence of the number of negative samples on those algorithms. Figure 6 shows the Average F_1 max score and Average Extended Precision corresponding to different α. It clearly shows that our method performs better than others no matter how much α is taken. As α increases, the performance of all methods gradually decreases, but our method is less affected, showing that our method can effectively identify negative samples. For loop-closure detection, negative samples are generally far more than positive samples, which is one key reason why our method leads in metrics far ahead. Moreover, identifying negative samples is significant as false positives will bring fatal crashes to the SLAM system.

4.3 Pose accuracy

As described in Sect. 3.1, our approach can estimate the 3D relative pose $(\Delta x, \Delta y, \theta)$, while most other methods cannot estimate pose or can only estimate 1D pose (yaw). We compare our method with Scan Context, Intensity Scan Context, and Overlap. The ground-truth pose is calculated by:

$$T = T_1^{-1} T_2$$
$$(\Delta x, \Delta y, \theta) = \left(T\,(1,3)\,, T\,(2,3)\,, \arctan\left(\frac{T(2,1)}{T(1,1)}\right)\right) \quad (15)$$

Table 3 Yaw error on KITTI dataset

sequences	SC (deg)	ISC (deg)	ON (deg)	Ours-SK (deg)
00	11.526	**0.829**	2.595	0.891
02	11.301	1.343	4.911	**1.142**
05	18.394	0.904	3.329	**0.653**
06	4.074	**0.534**	1.124	0.759
07	21.862	0.684	2.233	**0.512**
08	49.170	3.856	68.622	**1.878**
Average	19.388	1.358	13.802	**0.973**

where $T_1 \in \mathrm{SE}(3)$ and $T_2 \in \mathrm{SE}(3)$ represent the pose of P^1 and P^2, respectively. Since the pitch and roll angles are hardly changed in autonomous vehicles, we ignore them.

Table 3 shows the relative yaw error on the KITTI dataset. We can see that our method outperforms other methods in terms of the average relative yaw error. Especially in the challenging sequence 08, affected by the reverse loop, most methods perform poorly, while our method can still accurately estimate the yaw angle. This again shows that our method can handle the reverse loop well. As mentioned in Sect. 4.2, OverlapNet performs poorly due to its inability to handle reverse loops.

Figure 7 shows the relative translation error of our approach on the KITTI dataset. As shown, our method can estimate accurate relative translation, which is currently not possible with other methods to our knowledge. Thus, our Fast Yaw Angle Calculate and Fast Semantic ICP approaches can give accurate 3-DOF pose estimation. This can provide a good initial value for the ICP algorithm to obtain a 6D pose or directly serve as a global constraint in the SLAM system.

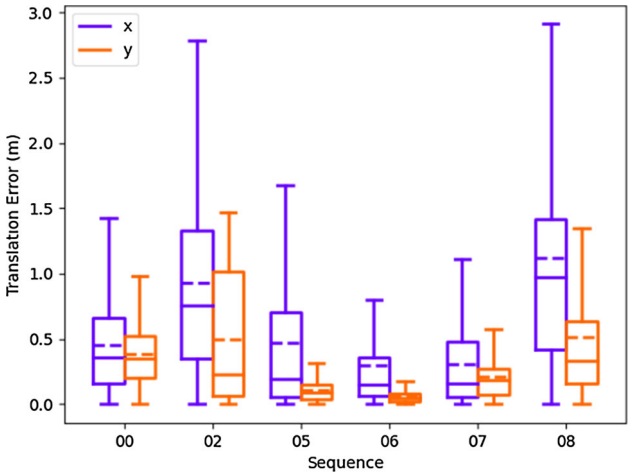


Fig. 7 Translation error

4.4 Robustness test

Occlusion Test: In the actual scene, part of the point cloud may be occluded due to the change of perspective or the influence of dynamic objects. To test the robustness of our approach to this situation, we simulated the occluded point cloud on the KITTI dataset. We randomly remove points within 30 degrees along the azimuth direction for each point cloud frame to simulate occlusion. We also design experiments to test whether occlusion will deteriorate semantic segmentation results and indirectly affect our algorithm. Specifically, we conduct experiments to simulate occlusion before semantic segmentation (Ours-RN*) and after semantic segmentation (Ours-RN). As shown in Table 4, the performance of all methods has declined as expected. However, compared to other methods, our method is less affected, and the performance of our method is far superior to other methods in most sequences. Almost all learning-based methods (PV, ON, SGPR) have been greatly affected, which shows that the generalization ability of these methods is limited. M2DP is not a learning-based method, but it suffers the most from all methods. This is because it uses PCA to establish a coordinate system for the point cloud, and occlusion will greatly affect the results of PCA. Comparing the results of Ours-RN and Ours-RN*, we can see that occlusion does indirectly affects our algorithm. However, the result of Ours-RN* is only slightly worse than that of Ours-RN. We guess that the semantic segmentation algorithm mainly relies on local features, so occlusion will not have a huge impact.

Viewpoint Changes: Usually, the robot can return to the original position in different directions. Therefore, being able to adapt to changes of viewpoint is very important for loop-closure detection algorithms. We randomly rotate the point cloud on the KITTI dataset to simulate the change of viewpoint. As shown in Table 5, our method is almost unaffected thanks to the global ICP, which shows that our method is rotation invariant. LiDAR Iris uses Fourier transform to achieve rotation invariance, so this method is not affected by viewpoint changes. To reduce the influence of rotation, SGPR performs data enhancement in the network training stage, SC and ISC perform column shifts on the descriptors. However, their results were still slightly affected. The performance of the several remaining methods is greatly reduced due to their sensitivity to changes of viewpoint. The results of Ours-RN and Ours-RN* are almost identical, which shows that the rotation has nearly no effect on the semantic segmentation algorithm we use.

4.5 Generalization ability

To further explore the generalization ability of our method, we conducted experiments on the KITTI-360 dataset. As described in Sect. 4.1, the KITTI-360 dataset is more complex than the KITTI dataset. Compared with the KITTI dataset, the KITTI-360 dataset contains both forward and reverse loop-closures in each sequence. This requires the algorithm to be able to handle rotation. As shown in Fig. 8, because the semantic information of the single-frame point cloud on KITTI-360 is acquired manually, there are errors in the annotation. Obviously, this will have an impact on semantic-based methods.

Since the number of loop-closures on the KITTI-360 dataset is much more than that on the KITTI dataset, we set α to 10 instead of 100 to reduce the experiment time. As shown in Fig. 9 and Table 6, our method can also achieve good results on the KITTI-360 dataset. This proves that our method has good generalization ability. As mentioned in Sect. 4.4, because M2DP, OverlapNet, and PointNetVLAD are not rotation-invariant, they perform poorly on the KITTI-360 dataset.

4.6 Performance in the SLAM system

This experiment is designed to show how our method can benefit the SLAM system. We integrate the proposed method into the existing SLAM system LOAM and evaluate it on the KITTI dataset.

Quantitative Results. Table 7 shows the relative pose error (RPE) and absolute translation error (ATE) on the KITTI dataset. It can be seen from the table that our method can significantly reduce the relative pose error and absolute trajectory error of the odometry. Figure 10 shows the trajectory on the 00 sequences of the KITTI dataset. From this figure, we can intuitively see that the consistency of the trajectory has been greatly improved after applying our method.

Qualitative Results. Visualizations of the similarity are shown in Fig. 11. The first row (a-c) and the second row (d-f) are the visualization results of sequence 00 and sequence

Table 4 Table caption

Methods	00	02	05	06	07	08	Mean	Cmp
SC	0.724/0.619	0.751/0.591	0.845/0.692	0.904/0.836	0.616/0.554	0.552/0.561	0.732/0.642	−0.045/−0.039
ISC	0.620/0.579	0.686/0.590	0.711/0.633	0.812/0.725	0.589/0.606	0.387/0.538	0.634/0.612	−0.036/−0.049
M2DP	0.199/0.510	0.138/0.510	0.283/0.512	0.140/0.518	0.113/0.506	0.046/0.500	0.153/0.509	−0.422/−0.091
LI	0.627/0.585	0.710/**0.628**	0.679/0.688	0.859/0.759	0.585/0.590	0.383/0.532	0.641/0.630	−0.062/−0.044
PV	0.547/0.534	0.570/0.529	0.295/0.502	0.589/0.581	0.444/0.513	0.031/0.500	0.413/0.527	−0.182/−0.094
ON	0.756/0.549	0.706/0.593	0.791/0.584	0.781/0.538	0.712/0.502	0.246/0.500	0.665/0.544	−0.125/−0.093
SGPR	0.649/0.500	0.604/0.500	0.619/0.557	0.542/0.501	0.625/0.571	0.531/0.500	0.595/0.522	−0.171/−**0.023**
Ours-RN	0.900/0.684	0.870/0.596	0.918/0.846	**0.950/0.874**	0.838/0.711	0.863/0.772	0.890/0.747	−0.028/−0.078
Ours-RN*	0.900/0.693	0.863/0.595	0.920/0.856	0.944/0.853	0.834/0.711	0.852/0.726	0.886/0.739	−0.032/−0.086
Ours-SK	**0.919/0.766**	**0.881**/0.609	**0.929/0.862**	0.948/0.853	**0.847/0.765**	**0.911/0.853**	**0.906/0.785**	−**0.026**/−0.083

F_1 max scores and Extended Precision: F_1 max scores / Extended Precision. We randomly remove points within 30 degrees along the azimuth direction for each point cloud frame to simulate occlusion. The best scores are marked in bold.
Ours-RN: remove points after semantic segmentation; Ours-RN*: remove points before semantic segmentation

Table 5 Viewpoint Changes

Methods	00	02	05	06	07	08	Mean	Cmp
SC	0.719/0.599	0.734/0.627	0.844/0.754	0.898/0.864	0.606/0.542	0.546/0.572	0.725/0.660	−0.052/−0.021
ISC	0.659/0.627	0.701/0.582	0.769/0.722	0.840/0.770	0.629/0.651	0.403/0.540	0.667/0.649	−0.003/−0.012
M2DP	0.276/0.543	0.282/0.545	0.341/0.533	0.316/0.549	0.204/0.534	0.201/0.502	0.270/0.534	−0.305/−0.066
LI	0.667/0.624	0.764/0.661	0.772/0.749	0.912/0.792	0.633/0.663	0.470/0.567	0.703/0.676	0.000/+0.002
PV	0.083/0.504	0.090/0.506	0.490/0.518	0.094/0.506	0.064/0.503	0.086/0.504	0.151/0.507	−0.444/−0.114
ON	0.130/0.501	0.092/0.505	0.113/0.501	0.114/0.502	0.173/0.507	0.117/0.501	0.123/0.503	−0.667/−0.134
SGPR	0.772/0.501	0.716/0.501	0.723/0.534	0.640/0.502	0.748/0.624	0.678/0.506	0.713/0.528	−0.053/−0.017
Ours-RN	0.939/0813	0.888/**0.756**	0.939/0.878	**0.989/0.980**	0.868/0.757	0.905/0.814	0.921/0.833	+0.003/+0.008
Ours-RN*	0.937/0.782	0.880/0.725	0.936/0.886	0.979/0.956	0.874/0.782	0.938/0.871	0.924/0.834	**+0.006/+0.009**
Ours-SK	**0.955/0.850**	**0.889**/0.730	**0.952/0.899**	0.986/0.969	**0.876/0.795**	**0.943/0.933**	**0.934/0.863**	+0.002/−0.005

F_1 max scores and Extended Precision: F_1 max scores / Extended Precision. We randomly rotate the point cloud on the KITTI dataset to simulate the change of viewpoint. The best scores are marked in bold.
Ours-RN: rotate the point cloud after semantic segmentation; Ours-RN*: rotate the point cloud before semantic segmentation

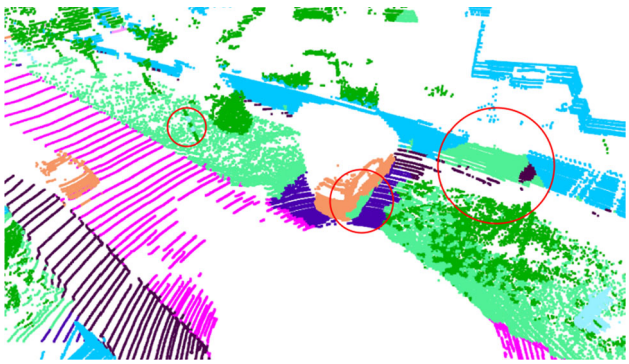


Fig. 8 The semantic information on the KITTI-360 dataset is worse

08. The first column (a and d) shows the similarity between a single frame point cloud and all other point clouds in the sequence. We randomly select a single frame point cloud to compare it with all the remaining point clouds and use the color shade to indicate the similarity. It can be seen from the figure that only scenes near the selected scene have high similarity, which shows that our method rarely has false detections. The second column (b and e) is the similarity matrix, and both rows and columns represent

nodes in the sequence. Specifically, the value in the i-th row and j-th column of the similarity matrix is the similarity between the point cloud of the i-th frame and the j-th frame obtained by our method. Similar to the second column, the third column (c and f) is the distance matrix. The value of the i-th row and j-th column of the distance matrix is calculated by the following formula:

$$v_{ij} = \begin{cases} (20 - d)/20 & d < 20 \text{ and } |i - j| > 100 \\ 0 & d \geq 20 \text{ or } |i - j| \leq 100 \end{cases} \tag{16}$$

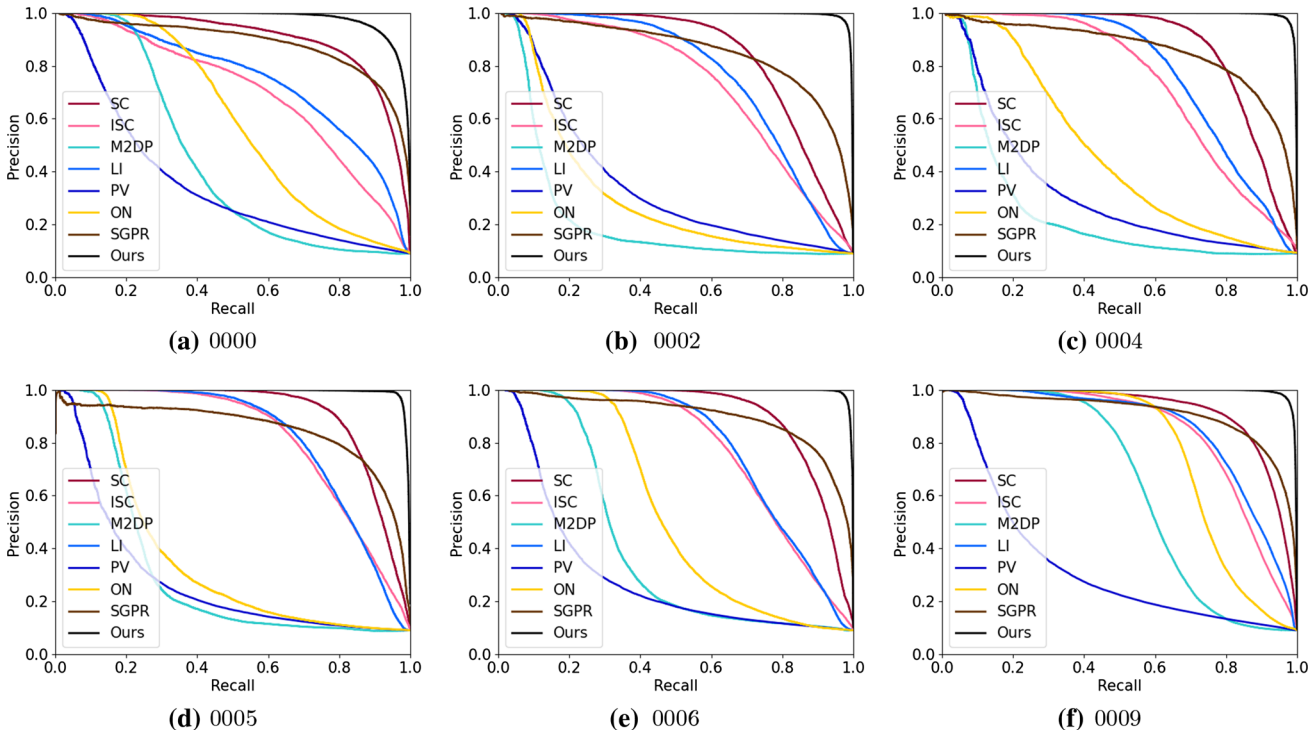


Fig. 9 Precision-Recall curves on KITTI-360 dataset

Table 6 F_1 max scores and Extended Precision on KITTI-360 dataset

Methods	0000	0002	0004	0005	0006	0009	Mean
SC	0.831/0.528	0.771/0.554	0.811/0.629	0.843/0.715	0.834/0.658	0.851/0.619	0.824/0.617
ISC	0.653/0.519	0.675/0.524	0.675/0.541	0.736/0.562	0.705/0.630	0.773/0.578	0.703/0.559
M2DP	0.423/0.557	0.209/0.509	0.246/0.512	0.311/0.534	0.397/0.544	0.620/0.570	0.368/0.538
LI	0.688/0.535	0.704/0.552	0.714/0.627	0.747/0.603	0.720/0.617	0.782/0.580	0.726/0.586
PV	0.352/0.511	0.349/0.515	0.325/0.515	0.285/0.508	0.295/0.510	0.330/0.510	0.323/0.512
ON	0.553/0.550	0.308/0.527	0.448/0.507	0.339/0.549	0.512/0.534	0.739/0.605	0.483/0.545
SGPR	0.818/0.505	0.788/0.505	0.795/0.504	0.798/0.500	0.833/0.514	0.843/0.501	0.813/0.505
Ours	**0.921/0.733**	**0.974/0.780**	**0.975/0.744**	**0.974/0.803**	**0.978/0.933**	**0.970/0.866**	**0.965/0.810**

F_1 max scores and Extended Precision: F_1 max scores / Extended Precision.
The best scores are marked in bold

Table 7 The relative pose error (RPE) and absolute translation error (ATE) on the KITTI dataset

	RPE						ATE					
	00	02	05	06	07	08	00	02	05	06	07	08
LOAM	0.657	0.832	0.419	0.368	0.411	0.909	4.029	10.257	3.412	0.671	0.596	3.117
LOAM+SSC	**0.634**	**0.778**	**0.279**	**0.359**	**0.316**	**0.876**	**0.873**	**5.256**	**0.344**	**0.470**	**0.302**	**1.814**

RPE: mean relative pose error over trajectories of 100 to 800 m. ATE: directly measures the difference between points of the true and the estimated trajectory
The best results are marked with bold

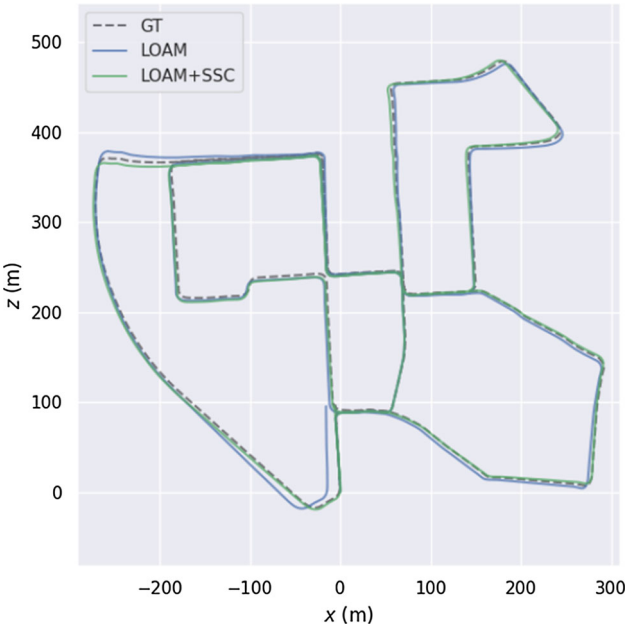


Fig. 10 Trajectory of sequence 00 on the KITTI dataset

Where d represents the distance between the point cloud of the i-th frame and the j-th frame. Comparing Figure b and Figure c (or Figure e and Figure f), we can find that the similarity distribution of the similarity matrix is very similar to the distance distribution of the distance matrix. This shows that the similarity distribution given by our method is ideal, which is an important reason why our method can achieve good results.

4.7 Ablation study

Contribution of individual components. We design an ablation study to investigate the contribution of each component. Specifically, we remove or replace a module at a time and then calculate the F_1 max scores and Extended Precision. To show the contribution of our Fast Yaw Angle Calculate method, we replace this module with the method used in scan context – shift the column of descriptors and calculate the maximum similarity score while obtaining the yaw angle. Similarly, we replace the semantic label in the descriptor by maximum z to see semantic contribution. To

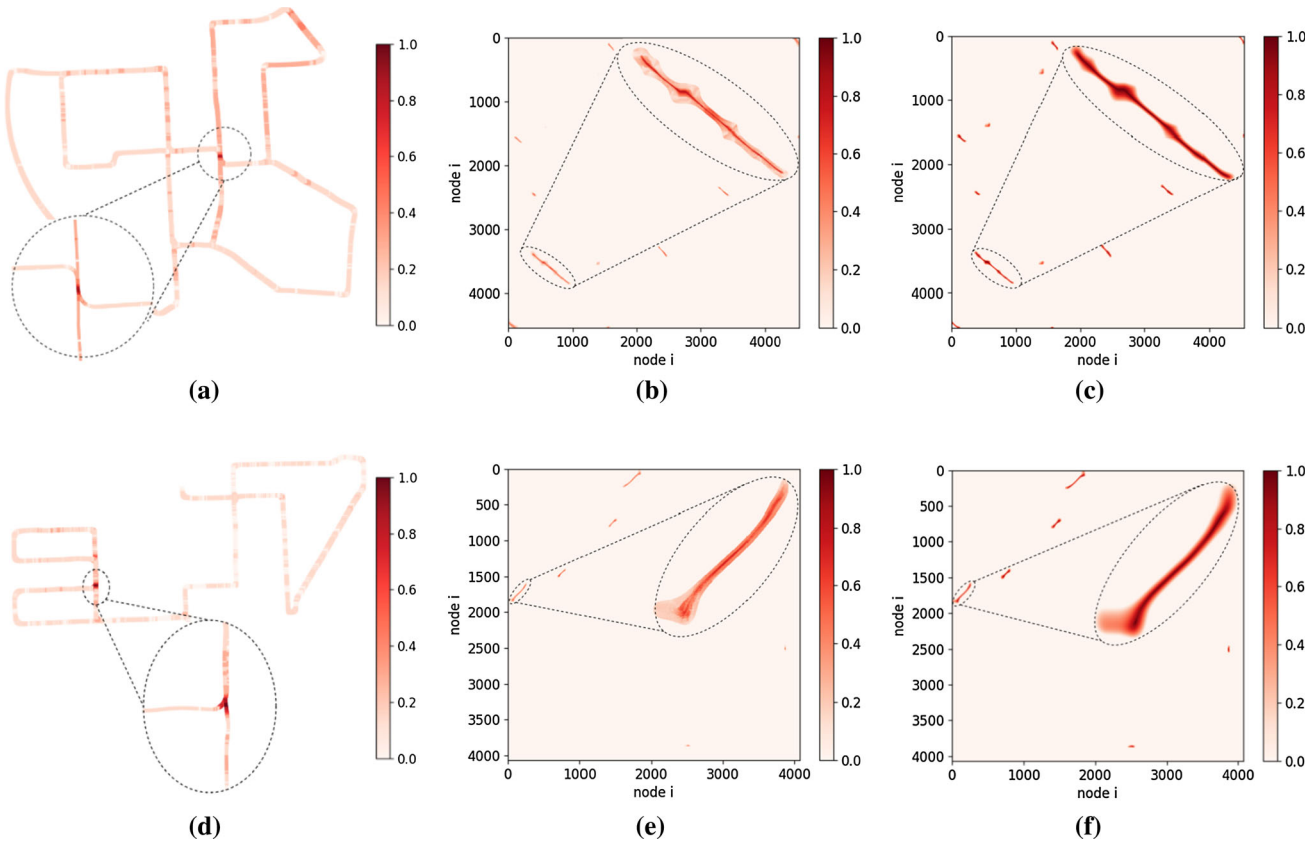


Fig. 11 Similarity visualization. The first row **a**–**c** and the second row **d**–**f** are the visualization results of sequence 00 and sequence 08. For the first column (**a** and **d**), we randomly choose a single scene in each sequence and zoom in similarity scores around this scene. The second (**b** and **e**) and third (**c** and **f**) columns show the corresponding similarity matrix and distance matrix, respectively. The darker the color, the higher the similarity

Table 8 Contribution of individual components

Yaw	ICP	Semantic	F_1/EP	Decrease
	√	√	0.896/0.820	3.6%/4.8%
√		√	0.757/0.685	17.5%/18.3%
√	√		0.775/0.762	15.7%/10.6%
√	√	√	0.932/0.868	0.0%/0.0%

evaluate the contribution of our Fast Semantic ICP approach, we directly set Δx and Δy to 0. As shown in Table 8, after removing Yaw, ICP, and Semantic, the average F_1 max score decrease by 3.6%, 17.5%, 15.7%, and the average Extended Precision decrease by 4.8%, 18.3%, 10.6%. Therefore, the following conclusions can be drawn:

- Compared with other methods, our approach can get a more accurate yaw angle and translation.
- As we emphasized, the small translation has a significant impact on scan context-based methods. Simply ignoring the translation will greatly weaken the performance.
- High-level features, like semantics, can bring considerable improvements in the scene description.

Contribution of individual semantic. Our descriptor uses a total of eleven types of static semantic objects (road, parking, sidewalk, other-ground, building, fence, vegetation, trunk, terrain, pole, traffic-sign). We delete each semantic separately to show their contribution. To speed up the experiment, we set α to 10, which means that the number of negative samples is ten times that of positive samples. As show in Table 9, removing any semantics alone will not have a particularly large impact on the results. This shows that our method does not depend on specific semantics and can still work when some semantic information is missing in the environment. Among all semantics, vegetation has the greatest influence, especially in sequences 02 and 08. We found that there is far more vegetation than other objects in these scenes, so a lot of information will be lost if we remove all the vegetation. It can be expected that the more semantic categories we use, the more robust our method will be.

4.8 Efficiency

To evaluate the efficiency, we set α to 1 and compare the average time cost of our method with Scan Context and Intensity Scan Context on sequence 08. As shown in Table 10, the total time cost of our approach is acceptable. As we use the obtained 3-DOF pose to align the point clouds in advance, we don't need to shift the column of descriptors during the matching stage, so our retrieval speed is extremely fast. Our two-step global semantic ICP only takes 2.126

Table 9 Contribution of individual semantic

Discard	road	parking	side-walk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	full
00	0.990↑/0.945↑	0.988↑/0.886↓	**0.977↓/0.830↓**	0.986/0.912↑	0.986/0.918↑	0.985↓/0.893↓	0.982↓/0.896↑	0.985↓/0.914↑	0.985↓/0.884↓	0.986/0.879↓	0.986/0.913↑	0.986/0.895
02	0.973↑/0.833↓	0.961↓/0.837↓	0.970↑/0.848↑	0.966/0.857↑	0.963↓/0.877↑	0.960↓/0.825↓	**0.948↓/0.696↓**	0.966/0.854↑	0.961↓/0.858↑	0.967↑/0.847↑	0.968↑/0.856↑	0.966/0.845
05	0.981↑/0.948↑	0.981↑/0.921↑	0.981↑/0.915↑	0.978/0.906↓	0.976↓/0.896↓	0.980↑/0.894↓	**0.970↓/0.862↓**	0.978/0.909↑	0.973↓/0.892↓	0.978/0.908↑	0.979↑/0.912↑	0.978/0.907
06	**0.981↓/0.973↓**	0.993↓/0.982↓	0.990↓/**0.968↓**	0.993↓/0.984↓	0.993↓/0.989↓	0.994/0.990	0.990↓/0.982↓	0.994/0.988↓	0.992↓/0.981↓	0.993↓/0.981↓	0.994/0.981↓	0.994/0.990
07	0.968↓/0.905↑	0.969/0.821	0.959↓/0.787↓	0.969/0.843↑	0.971↑/0.837↑	0.969/0.849↑	**0.949↓/0.786↓**	0.968↓/0.835↑	0.965↓/0.838↑	0.967↓/0.849↑	0.968↓/0.829↑	0.969/0.821
08	0.950↓/0.924↓	0.963↑/0.934↑	0.956↓/0.931↓	0.963↑/0.935↑	0.962↑/0.933	0.965↑/0.931↓	**0.936↓/0.865↓**	0.957↓/0.935↑	0.965↑/0.925↓	0.965↑/0.934↑	0.963↑/0.928↓	0.961/0.933
Mean	0.974↓/0.921↑	0.976/0.897↓	0.972↓/0.880↓	0.976/0.906↑	0.975↓/0.908↑	0.976/0.897↓	**0.963↓/0.848↓**	0.975↓/0.906↑	0.974↓/0.896↓	0.976/0.900↑	0.976/0.903↑	0.976/0.899
Cmp	-0.002/+0.023	0.000/-0.002	-0.003/-0.019	0.000/+0.008	0.000/+0.010	0.000/-0.002	**-0.013/-0.051**	-0.001/+0.007	-0.002/-0.002	0.000/+0.001	+0.001/+0.005	-

F_1 max scores and Extended Precision: F_1 max scores / Extended Precision. We remove one semantic each time to verify its effect. The most affected data in each sequence is marked in bold

Table 10 Average time cost on KITTI 08

Methods	Size	Description	Retrieval	ICP	Total
SC	20×60	4.825	0.158	–	4.983
ISC	20×90	3.094	0.800	–	**3.894**
Ours	50×360	**2.563**	**0.066**	2.126	4.755

The unit of time in the table is milliseconds.
The best results are marked with bold

milliseconds on average. This algorithm is fast due to the following reasons. Firstly, since we only keep N_a (360 taken in our experiments) points, the computational cost is greatly reduced compared to the original point cloud (about 120,000 points). Secondly, We divide the algorithm into two steps, first calculate the yaw angle, and then iteratively calculate Δx and Δy, which simplifies the algorithm and speeds up the calculation. Thirdly, when calculating Δx and Δy, we use the yaw angle to align the input clouds in advance. Therefore we don't need to traverse the entire point cloud when looking for the corresponding points. Instead, we can find them near the corresponding positions, which greatly reduces the number of searches.

5 Conclusion

This paper addressed loop-closure detection for LiDAR SLAM. We propose a semantic-based place recognition method called Semantic Scan Context to estimate the similarity between pairs of LiDAR scans. At the same time, our method can also give a rough 3-DOF pose, eliminating the influence of rotation and translation on descriptor matching. Based on Semantic Scan Context, we add geometric verification and other operations to reduce mismatches further, thus obtaining a complete loop-closure detection module. Finally, we combine the proposed loop-closure detection module with the well-known LOAM to build a full LiDAR SLAM system. Extensive experiments on the KITTI and KITTI-360 datasets prove that our method is competitive to the state-of-the-art methods, robust to the environment, and has good generalization ability.

However, in practical applications, we have also found some shortcomings, one of which is the cost of redundant calculation. Since we need to align the source point cloud to the target point cloud, the descriptor corresponding to the source point cloud always needs to be calculated online. In our future work, we will try to solve this problem.

References

Angeli, A., Filliat, D., Doncieux, S., & Meyer, J. (2008). Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics, 24*(5), 1027–1037.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J .(2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5297–5307).

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *Computer Vision - ECCV 2006* (pp. 404–417). Berlin: Springer.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., & Gall, J. (2019). Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 9297–9307).

Chang, MY., Yeon, S., Ryu, S., & Lee, D. (2020). Spoxelnet: Spherical voxel-based deep place recognition for 3d point clouds of crowded indoor spaces. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 8564–8570).

Chen, X., Läbe, T., Milioto, A., Röhling, T., Behley, J., & Stachniss, C .(2021). OverlapNet: a siamese network for computing LiDAR scan similarity with applications to loop closing and localization. Autonomous Robots

Chen, X., Läbe, T., Milioto, A., Röhling, T., Vysotska, O., Haag, A., Behley, J., & Stachniss, C .(2020). OverlapNet: Loop Closing for LiDAR-based SLAM. In *Proceedings of Robotics: Science and Systems (RSS)*.

Cop, K.P., Borges, P.V.K., & Dubé, R .(2018). Delight: An efficient descriptor for global localisation using lidar intensities. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3653–3660).

Du, J., Wang, R., & Cremers, D. (2020). Dh3d: Deep hierarchical 3d descriptors for robust large-scale 6dof relocalization. In *European Conference on Computer Vision* (pp. 744–762).

Dubé, R., Cramariuc, A., Dugas, D., Sommer, H., Dymczyk, M., Nieto, J., Siegwart, R., & Cadena, C. (2019). Segmap: Segment-based mapping and localization using data-driven descriptors. The International Journal of Robotics Research p 0278364919863090.

Dubé, R., Dugas, D., Stumm, E., Nieto, J., Siegwart, R., & Cadena, C. (2017). Segmatch: Segment based place recognition in 3d point clouds. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5266–5272).

Fan, Y., He, Y., & Tan, U.X. (2020). Seed: A segmentation-based egocentric 3d point cloud descriptor for loop closure detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5158–5163).

Ferrarini, B., Waheed, M., Waheed, S., Ehsan, S., Milford, M. J., & McDonald-Maier, K. D. (2020). Exploring performance bounds of visual place recognition using extended precision. *IEEE Robotics and Automation Letters, 5*(2), 1688–1695.

Galvez-López, D., & Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics, 28*(5), 1188–1197.

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

Guo, J., Borges, P. V. K., Park, C., & Gawel, A. (2019). Local descriptor for robust place recognition using lidar intensity. *IEEE Robotics and Automation Letters, 4*(2), 1470–1477.

Han, F., Wang, H., Huang, G., & Zhang, H. (2018). Sequence-based sparse optimization methods for long-term loop closure detection in visual slam. *Autonomous Robots, 42*(7), 1323–1335.

He, L., Wang, X., & Zhang, H. (2016). M2dp: A novel 3d point cloud descriptor and its application in loop closure detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 231–237).

Ji, X., Zuo, L., Zhang, C., & Liu, Y. (2019).Lloam: Lidar odometry and mapping with loop-closure detection based correction. In *2019 IEEE International Conference on Mechatronics and Automation (ICMA)* (pp. 2475–2480).

Johnson, A. E., & Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*(5), 433–449.

Kim, G., & Kim, A. (2018). Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4802–4809).

Kim, G., Park, B., & Kim, A. (2019). 1-day learning, 1-year localization: Long-term lidar localization using scan context image. *IEEE Robotics and Automation Letters, 4*(2), 1948–1955.

Komorowski, J. (2021). Minkloc3d: Point cloud based large-scale place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1790–1799).

Kong, X., Yang, X., Zhai., G., Zhao, X., Zeng, X., Wang, M., Liu, Y., Li, W., & Wen, F. (2020). Semantic graph based place recognition for 3d point clouds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 8216–8223).

Liao, Y., Xie, J., & Geiger, A. (2021) .KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. arXiv:2109.13410.

Liu, Z., Zhou, S., Suo, C., Yin, P., Chen, W., Wang, H. Li, H. & Liu, Y.H. (2019). PD-NET: 3d point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2831–2840).

Lu,W., Zhou, Y., Wan, G., Hou, S., & Song, S. (2019). L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6389–6398).

Milioto, A., Vizzo, I., Behley, J., & Stachniss, C. (2019). Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4213–4220).

Muhammad, N., Fuentes-Perez, J. F., Tuhtan, J. A., Toming, G., Musall, M., & Kruusmaa, M. (2019). Map-based localization and loop-closure detection from a moving underwater platform using flow features. *Autonomous Robots, 43*(6), 1419–1434.

Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics, 33*(5), 1255–1262.

Negre Carrasco, P. L., Bonin-Font, F., & Oliver-Codina, G. (2016). Global image signature for visual loop-closure detection. *Autonomous Robots, 40*(8), 1403–1417.

Qi, C.R., Su, H., Mo, K., & Guibas, L.J. (2017) .Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652–660).

Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics, 34*(4), 1004–1020.

Röhling, T., Mack, J., & Schulz, D. (2015). A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 736–741).

Rozenberszki, D., & Majdik, A.L. (2020). Lol: Lidar-only odometry and localization in 3d point cloud maps <sup> *<sup>. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4379–4385).

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision* (pp. 2564–2571).

Schaupp, L., Bürki, M., Dubé, R., Siegwart, R., & Cadena, C. (2019). Oreos: Oriented recognition of 3d point clouds in outdoor scenarios. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3255–3261).

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., & Li, H. (2020). PV-RCNN: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10529–10538).

Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., & Han, S. (2020). Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision* (pp. 685–702).

Uy, MA., & Lee, G.H. (2018). Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4470–4479).

Wang, Y., Sun, Z., Xu, C.Z., Sarma, S.E., Yang, J., & Kong, H. (2020b). Lidar iris for loop-closure detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5769–5775).

Wang, H., Wang, C., & Xie, L. (2020a). Intensity scan context: Coding intensity and geometry relations for loop closure detection. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2095–2101).

Wohlkinger, W., & Vincze, M. (2011). Ensemble of shape functions for 3d object classification. In *2011 IEEE International Conference on Robotics and Biomimetics* (pp. 2987–2992).

Yang, J., Li, H., Campbell, D., & Jia, Y. (2016). Go-ICP: A globally optimal solution to 3d ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(11), 2241–2254.

Yin, H., Tang, L., Ding, X., Wang, Y., & Xiong, R. (2018). Locnet: Global localization in 3d point clouds for mobile vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)* (pp. 728–733).

Yin, P., Wang, F., Egorov, A., Hou, J., Zhang, J., & Choset, H. (2020). Seqspherevlad: Sequence matching enhanced orientation-invariant place recognition. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5024–5029).

Zhang, J., & Singh, S. (2017). Low-drift and real-time lidar odometry and mapping. *Autonomous Robots, 41*(2), 401–416.

Zhu, Y., Ma, Y., Chen, L., Liu, C., Ye, M., & Li, L. (2020). Gosmatch: Graph-of-semantics matching for detecting loop closures in 3d lidar data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5151–5157).

Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., & Lin, D. (2021). Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9939–9948).

Lin Li received his B.Eng. degree in Detection Guidance and Control Techniques from Harbin Institute of Technology, Harbin, China, in 2020. He is currently working towards the M.S. in Control Science and Engineering from the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China. His latest research interests include SLAM and 3D computer vision.

Xin Kong received his B.Eng. degree in Automation from Harbin Institute of Technology, Harbin, China, in 2018. He is currently working towards the M.S. in Control Science and Engineering from the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China. His latest research interests include robotic perception, SLAM and 3D computer vision.

Xiangrui Zhao received his B.Eng. degree in Automation from Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently a Ph.D. candidate in Control Science and Engineering from the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China. His latest research interests include robotics vision and SLAM systems.

Tianxin Huang received his B.Eng. degree in Mechanical Engineering from Xi'an Jiaotong University, Xi'an, China, in 2017. He is currently a Ph.D. candidate in Control Science and Engineering from the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China. His latest research interests include 3D visions and graphics.

Yong Liu (M'11) received the B.S. degree in computer science and engineering and the Ph.D degree in computer science from Zhejiang University, Zhejiang, China, in 2001 and 2007, respectively. He is currently a professor of Institute of Cyber-Systems and Control at Zhejiang University. His main research interests include: intelligent robot systems, robot perception and vision, deep learning, big data analysis, and multi-sensor fusion. He has published over 30 research papers on machine learning, computer vision, information fusion, and robotics.