2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

Semantic Topological Descriptor for Loop Closure Detection within 3D Point Clouds In Outdoor Environment

Ming Liao[1], Yunzhou Zhang[1*], Jinpeng Zhang[1], Liang Liang[2],
Sonya Coleman[3], Dermot Kerr[3]

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981965

***Abstract*— Loop closure detection has the potential to correct the drift of trajectories and build a global consistent map in LiDAR SLAM, however it remains a challenging problem in outdoor environment due to the sparsity of 3D point clouds data, large-scale scenes and moving objects. Inspired by the way humans perceive the environment through recognizing objects and identifying their relations, this paper presents a novel descriptor that contains semantic and topological information for loop closure detection. Unlike most existing methods that extract features from the raw point clouds or use all semantic objects, we directly discard point clouds representing pedestrians and vehicles after semantic segmentation. Then, we propose a semantic topological graph representation from the remaining point clouds and convert this graph into a descriptor. Additionally, we propose a two-stage algorithm for matching descriptors to efficiently determine the loop. Our method has been extensively evaluated using the KITTI dataset and outperforms state-of-the-art methods, especially in the challenging situations such as viewpoint changes and dynamic scenes.**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) technology is widely used in autonomous vehicles and robots. Loop closure detection, as an important part of SLAM, can help a robot identify places visited previously, correct the accumulated drift error and build a globally consistent map to provide accurate prior information for moving vehicles.

Vision-based loop closure detection has been investigated for a long time, [1]–[3], using the Bag-of-Words method for encoding image features. However, vision-based methods are susceptible to lighting changes, viewpoint shifts, and dynamic objects. LIDAR-based approaches have received more attention as they can generate high-resolution 3D point clouds, have a larger field of view, and are not affected by illumination.Traditional LIDAR-based methods process raw point clouds directly, using either local descriptors [4]–[6] or global descriptors [7]–[11]. These methods have good geometric descriptiveness for point clouds but are sensitive to occlusion and viewpoint changes. To solve the uncertainty of geometric information, some segment-based approaches [13]–[15] have been proposed. These methods typically use neural networks to extract semantic information from point clouds containing high-level information about the environment. However, they ignore the relationships between semantic objects that can help to describe the environment and are invariant to viewpoint changes.

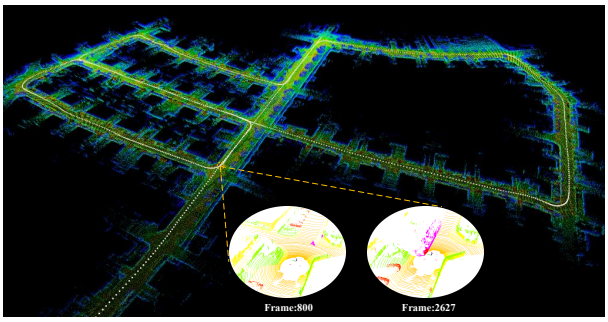


Fig. 1. This is an illustration of our proposed method. The map was created on 05 sequence from the KITTI dataset using LEGO-LOAM [12] and our method. Note that there is a pair of loops between frames 2627 and 880, where a large portion of the scene is obscured by dynamic objects, making it challenging for existing methods.

In this paper, we propose a novel semantic topological descriptor for loop closure detection with a single 3D scan. After semantic segmentation of the point clouds, we discard the point clouds corresponding to the vehicles, pedestrians, roads, and sidewalks to reduce the dynamic effects and computational burden. Then, feature points are obtained from the remaining point clouds and corresponding scores are calculated based on the semantic features and distance distributions. Non-Maximum Suppression (NMS) is performed by bird's-eye projection to extract nodes and construct a semantic topological graph. The graph is converted into a descriptor, and a two-step search strategy is used to find the loop. An illustration of a detected loop is shown in Fig.1. Our main contributions are summarized as follows:

(1) For outdoor scenes, we propose a semantic topological graph representation that incorporates the structural appearance, static semantic information, and topological relationships of 3D point clouds.

(2) We convert the semantic topological graph into a descriptor and compare the similarity in a coarse-to-fine way to complete the loop closure detection.

(3) Comparative evaluation demonstrates that, for viewpoint changes and dynamic scenes, our proposed method outperforms the state-of-the-art loop closure detection methods using the KITTI dataset [16].

*The corresponding author of this paper.
[1]Ming Liao, Yunzhou Zhang and Jinpeng Zhang are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (Email: zhangyunzhou@mail.neu.edu.cn).
[2]Liang Liang is with SIASUN Robot & Automation CO.,Ltd., China.
[3]Sonya Coleman and Dermot Kerr are with School of Computing, Engineering and Intelligent Systems, Ulster University, N. Ireland, UK.

This work was supported by the National Key Research and Development Program of China, National Natural Science Foundation of China (No.61973066) , Major Science and Technology Projects of Liaoning Province(No.2021JH1/10400049), Fundation of Key Laboratory of Aerospace System Simulation(No.6142002200301) and Fundamental Research Funds for the Central Universities (N2004022).

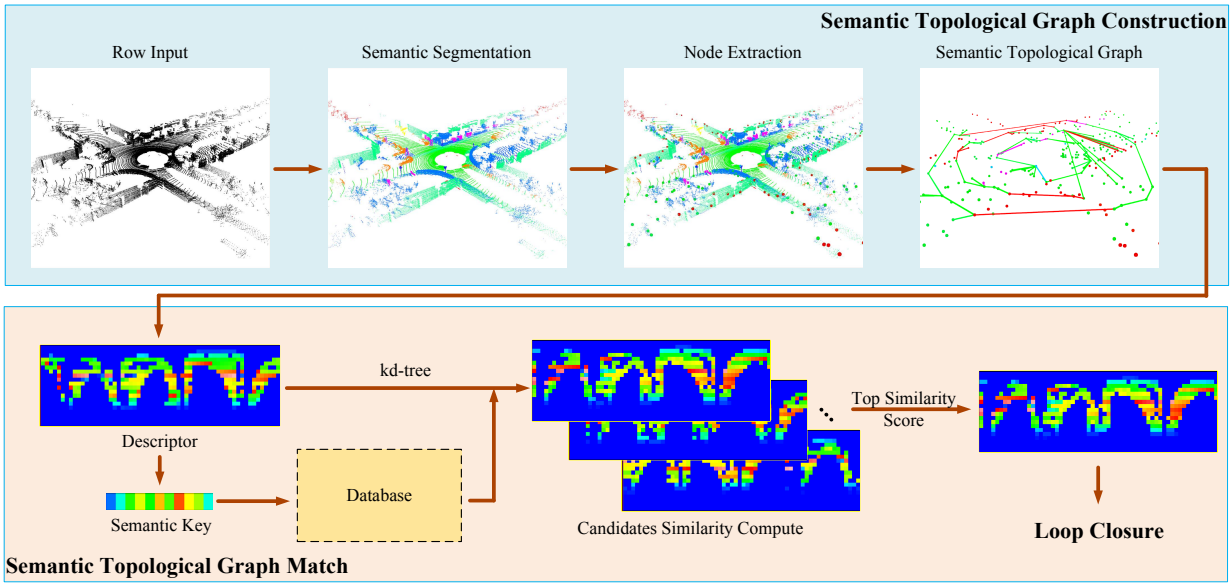


Fig. 2. The framework includes the construction and matching of the semantic topological graph. First, the nodes are extracted from the raw point clouds by semantic segmentation and feature aggregation, and the graph is generated after incorporating the topological information. Then, the semantic topological graph is encoded into a descriptor for matching, where the semantic key is extracted from the descriptor for building the kd-tree to accelerate the search.

II. RELATED WORK

The place recognition methods based on 3D point clouds can be categorized into geometry-based methods, semantic-based methods, and graph-based methods.

A. Geometry-based Methods

A spin image [4] is generated by projecting 3D point clouds of local regions into a 2D grid and matching the point clouds by comparing the 2D image similarity. M2DP [7] projects the 3D point clouds from different viewpoints on to a series of 2D planes, using the first left singular vector and the right singular vector in 2D space as the descriptor. Scan Context [10] projects the 3D point clouds, obtained from a LIDAR scan, into an egocentric 2D matrix, and stores the height value of the highest point in each bin for loop detection. IRIS [11] obtains binary feature images of point clouds by LoG-Gabor filtering and thresholding operations, using the Hamming distance to calculate the similarity for rotation-invariant loop closure detection. Intensity Scan Context [17] uses intensity information to construct a two-dimensional matrix and proposes an efficient binary operation to achieve a fast search. These methods all use low-level information from the environment and can generate descriptors from raw point clouds data quickly, but the performance is limited by the data sparsity.

B. Semantic-based Methods

SegMatch [13] extracts linear, planar, and surface feature vectors from the segmented clusters to construct multiple histograms using shape functions and trains a classifier to match features. SegMap [18] inputs segmented point clouds clusters into the 3D CNN network directly, trains the feature descriptor and performs classification to achieve place recognition. OverlapNet [19] takes the depth, normal vector, intensity, and semantic information of the point clouds as input, uses a neural network to estimate the overlap rate and the yaw angle of LIDAR scans to determine the loop. Semantic Scan Context (SSC) [20] adds semantic information to the Scan Context [10], improves the accuracy of place recognition with a two-step iterative closest point (ICP) method. These approaches use semantic information to express the environment with robustness but do not consider the relationship between semantic objects, which humans naturally use to distinguish scenes.

C. Graph-based Methods

GOSMatch [21] uses semantic segmentation to obtain the point clouds labels, selects the clustering results of static objects as nodes to construct a semantic topological graph. However, the variation in the semantic information is limited in the graph and hence semantic information can not be fully utilized. SPGR [22] selects 12 types of semantic information and input the clustered point clouds into a bespoke neural network for graph similarity matching. Their work represents large-scale objects with one semantic point, which cannot solve the problem when two segments belong to the same class. Locus [23] encodes the topological and temporal information of the point clouds after scene segmentation, aggregates the features and generates a fixed-length global descriptor. The above methods add constraints with the semantic information to achieve a better description of the environment but do not consider the impact of dynamic objects, which are common in outdoor environments.

In this paper, we propose a semantic topological descriptor for outdoor dynamic scenes and apply a loop closure detection by matching similarity from coarse to fine.

III. METHODOLOGY

We present our semantic topological approach for loop closure detection, including semantic topological graph construction and graph similarity computation, as shown in Fig.2.

A. Semantic Topological Graph Construction

Semantic segmentation: RangeNet++ [24] is a neural network used for semantic segmentation of 3D point clouds. It uses the original scan as input and aims to achieve a balance between accuracy and speed. SemanticKITTI [25] provides accurate scan sequence labels based on the KITTI dataset, which includes 19 classes. In the experiments, we use RangeNet++ and SemanticKITTI to generate semantic information for input into the framework. In particular, we discard the dynamic semantic objects and corresponding static semantic objects (for example, a moving car and a parked car) in order to create a static scene.

Semantic node extraction: We extract feature points from the semantic point clouds as nodes, and large-scale point clouds of objects are represented by multiple nodes. To avoid complex computations when extracting feature points, and inspired by the online topological path optimization method [26], we use the GHPR descriptor [27] which can determine the observability of point clouds through geometric operations. Specifically, the original scan is transformed into a new space as shown in Fig.3, and the distribution of point clouds in the new space is calculated to obtain the convex points concerning the viewpoint. Given the set raw point clouds P_{origin}, we convert it to a new set of point clouds P_{hull} and get the convex point set P_{convex}, where the points are respectively denoted as p_o, p_h, p_c. We transform each origin p_o and viewpoint p_v using the following formulation:

$$p_h = \begin{cases} f\left(\|p_o - p_v\|\right) \cdot \frac{p_o - p_v}{\|p_o - p_v\|}, & p_o \neq p_v \\ p_v, & p_o = p_v \end{cases} \tag{1}$$

$$f\left(\|p_o - p_v\|\right) = \gamma \left(\max_{q \in P_{origin}} \|q - p_v\| - \|p_o - p_v\| \right) \tag{2}$$

where f is a kernel function and the scaling factor $\gamma = 10000$. We select K points p_h^k, where $k = 1 \dots K$, in the neighborhood of point p_h, and if point p_h satisfies the following radial condition then it is chosen as p_c:

$$p_c = \left\{ p_h \mid \|p_h - p_v\| > \frac{1}{K} \sum_{k=1}^{K} \left\| p_h^k - p_v \right\| \right\} \tag{3}$$

To evaluate the stability of points, the distribution of a scan is taken into account. Assuming that the distance of point p_h in a scan from the observation center is d_h, we consider the points close to the average distance $\bar{d}$ to be stable points. We use the variance σ^2 of the current scan and Gaussian kernel function to calculate the geometric score of point p_h:

$$\phi_1\left(p_h\right) = \kappa\left(d_h, \bar{d}\right) = \exp\left(-\frac{\left\| d_h - \bar{d} \right\|_2^2}{2\sigma^2} \right) \tag{4}$$

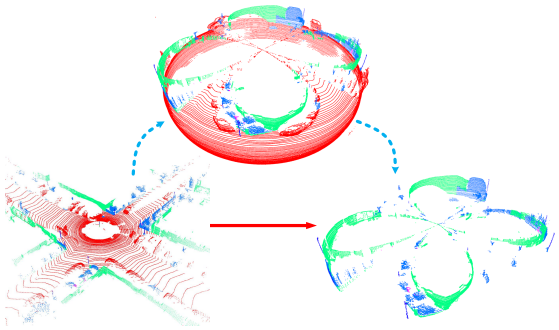


Fig. 3. An example of convex hull transformation. The raw point clouds (left figure bottom) are transformed to the convex hull (figure top), and it can be seen that the raw point clouds become smooth, which is beneficial to evaluate the observability of the point clouds. We discard the roads and vehicles point clouds directly (right figure bottom) to create static point clouds.

Semantics are high-level information in the environment, which are not affected by viewpoint changes. We divide the points into foreground points and background points according to the semantic labels. The foreground points set P_{front} includes labels like "trunk", "pole", "traffic-sign", which have clear shapes and are stable characteristics of the scene, while the background points set P_{back} includes labels like "building", "fence", "vegetation", which have large-scale point clouds and are indispensable for scene descriptions. For different categories of points, we design the following formulation based on the sigmoid function:

$$\phi_2\left(p_h\right) = sigmoid\left(\frac{\alpha}{K} \sum_{k=1}^{K} f\left(p_h, p_h^k\right)\right) \tag{5}$$

where $\phi_2\left(p_h\right)$ represents the semantic score of the point p_h, we use $\alpha = 5$ and $f\left(p_h, p_h^k\right)$ is defined as:

$$f\left(p_h, p_h^k\right) = \begin{cases} -l\left(p_h\right) \oplus l\left(p_h^k\right), & p_h \in P_{back} \\ l\left(p_h\right) \odot l\left(p_h^k\right), & p_h \in P_{front} \end{cases} \tag{6}$$

where $\odot$ indicates that if the labels(l) are the same, the output is 1 and if they are different, the output is 0. $\oplus$ indicates the opposite.

Semantic topological graph: We combine the geometric and semantic scores of points in a scan using:

$$\phi\left(p_h\right) = \phi_1\left(p_h\right) \cdot \phi_2\left(p_h\right) \tag{7}$$

We apply Non-Maximum Suppression to extract the points with the highest score in different regions as feature nodes. Specifically, we first divide a scan into azimuthal and radial bins in the sensor coordinates, L_{max} is the maximum distance from the center, N_s and N_r are the number of sectors and rings respectively. P_{convex} is convex point set and each point p_c is projected into the corresponding bin in the vertical direction:

$$P_{convex} = \bigcup_{i \in [N_r]\ j \in [N_s]} P^{i,j} \tag{8}$$

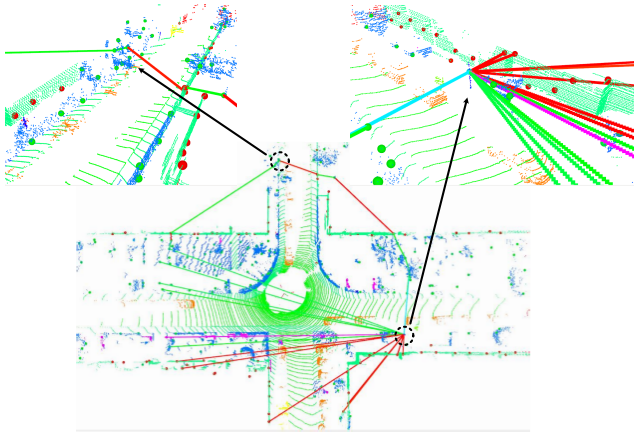


Fig. 4. An illustration of a semantic topological graph. The below figure represents a part of the semantic topological graph constructed by our method. Most of the nodes in a scene will be associated with the foreground nodes (above right), the same way that humans perceive the environment. In addition, the background(above left) nodes are still associated with each other to prevent the descriptor from changing drastically when the foreground nodes are occluded.

$$N^{i,j} = \left\{ p'_c \;\middle|\; p'_c = \underset{p_c \in P^{i,j}}{\arg\max} \left(\phi \left(p_c \right) \right) \right\} \tag{9}$$

where the point $p_c \in P^{i,j}$ with the highest score represents the bin and becomes a semantic node $N^{i,j}$. Symbol N_s runs from $\{1, 2, ..., N_{s-1}, N_s\}$ and symbol N_r runs from $\{1, 2, ..., N_{r-1}, N_r\}$. We construct the semantic topological graph by connecting semantic nodes with the largest semantic distance in the same radial bin, as shown in Fig.4.

B. Semantic Topological Graph Match

As a robot often has a large number of places to visit, it is not reasonable to search for the loop by brute force. Inspired by the Scan Context [10], we use a two-step search algorithm with semantic information and new cost functions based on the semantic topological graph.

Fast geometric-semantic search: As described in the above section, each bin has a corresponding semantic node, and we convert the 3D semantic topological graph into a 2D descriptor by taking the distance as the value of the bin, as shown in Fig.5. The descriptor is egocentric, so each row is rotation-invariant, and all rows are constructed as a vector by the encoding function for fast searching. The first value of the vector is obtained from the first row, and the following values are obtained from the next rows. The descriptor contains geometric and semantic information, hence our method generates a $N_r + N_o$ dimensional vector k after all rows are encoded, where o is the semantic category:

$$k = \begin{pmatrix} \varphi_g(1), ..., \varphi_g(i), ..., \varphi_g(N_r), \\ \varphi_s(1), ..., \varphi_s(o), ..., \varphi_s(N_o) \end{pmatrix} \tag{10}$$

$$\varphi_g(i) = \frac{1}{N_s} \sum_{j=1}^{N_s} d_g\left(N^{i,j}\right) \tag{11}$$

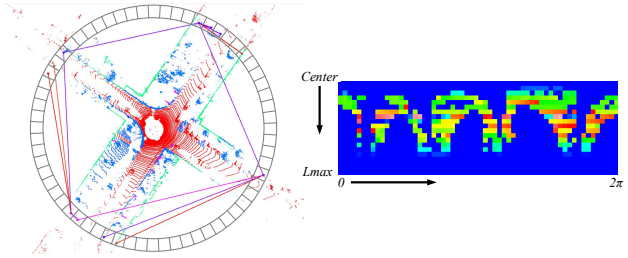


Fig. 5. The left figure is a part of the semantic topology graph, where the ring is transformed into a row in the descriptor. The right figure illustrates a descriptor of the whole semantic topology graph, where each bin corresponds to a node in the descriptor, and the value of the bin is the semantic distance.

$$\varphi_s(o) = \frac{1}{N_s} \sum_{i=1}^{N_r} \sum_{j=1}^{N_s} d_s\left(N^{i,j}\right) \tag{12}$$

φ_g and φ_s are the functions for encoding geometric and semantic information, corresponding to the respective dimensions and d_g and d_s are the functions to calculate the semantic distance, $\%$ represents remainder operation, defined as follows:

$$d_g = \max_{m \in [N_s]} \left(\left(m + N_s - j \right) \% N_s \cdot \phi\left(N^{i,m}\right) \right) \tag{13}$$

$$d_s = \max_{m \in [N_s]} \left(\left(m + N_s - j \right) \% N_s \cdot \phi\left(N^{i,m}\right) \right), l\left(N^{i,m}\right) = o \tag{14}$$

We construct a kd-tree and use the vector k as the key to search for 10 candidates, which are utilised for the similarity calculation.

Similarity calculation: Given the query descriptor ${}^qG$ and candidate descriptor ${}^cG$, we need to calculate the similarity of two places. Due to the values of the descriptor being the topological distances between semantic nodes, we define N^i as the distance vector of row i and use the cosine function to calculate the difference. Semantic similarity is added as a constraint and the function is defined as follows:

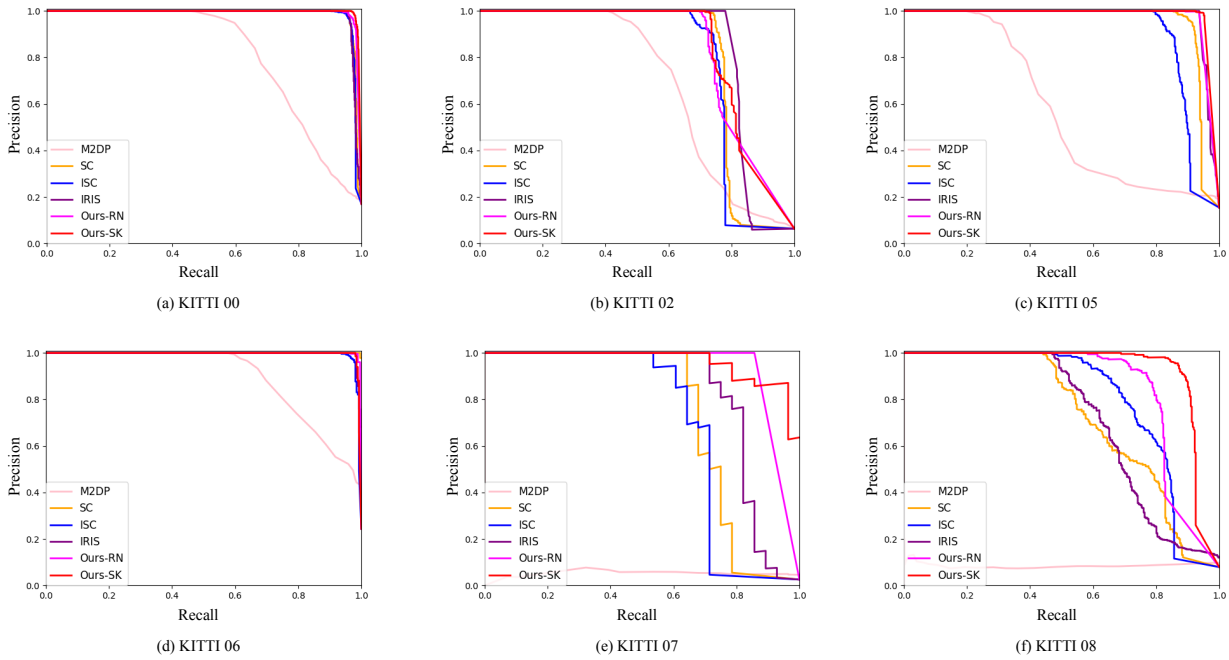
$$d\left({}^qG, {}^cG\right) = 1 - \frac{1}{N_r} \sum_{i=1}^{N_r} \varphi_d\left({}^qN^i, {}^cN^i\right) \tag{15}$$

$$\varphi_d = \frac{{}^qN^i \cdot {}^cN^i}{\left\| {}^qN^i \right\| \cdot \left\| {}^cN^i \right\|}, {}^ql = {}^cl \tag{16}$$

The column vectors of the descriptor may be shifted due to the viewpoint changes. To solve this problem, we extract the maximum distance value and minimize the Hamming distance to correct the shift.

$$R = \left[\max_{i \in [N_r]} d_g\left(N^{i,1}\right), \ldots, \max_{i \in [N_r]} d_g\left(N^{i,N_s}\right) \right] \tag{17}$$

$$d'\left({}^qG, {}^cG\right) = \min_{n \in [N_s]} \left\| {}^qR - {}^cR^n \right\| \tag{18}$$



(a) KITTI 00 (b) KITTI 02 (c) KITTI 05

(d) KITTI 06 (e) KITTI 07 (f) KITTI 08

Fig. 6. Precision-Recall curves on KITTI dataset.

TABLE I
THE INFORMATION ABOUT SEQUENCES.

Seq Id	00	02	05	06	07	08
Distance Threshold (m)	3					
Num of Nodes	4541	4661	2761	1101	1101	4071
Num of True Loops	774	296	425	268	28	321
Route In Loops	Same	Same	Same	Same	Same	Reverse

d' is the loss function of the Hamming distance, and R^n represents the shifted vector. For each candidate, we calculate the rotation n that minimizes d', and shift the descriptor by n columns. The final loop index c^* is determined by the acceptance threshold τ, and c is the indexes of candidates searched from kd-tree.

$$c^* = \arg\min_{c^* \in c} d\left({}^qG, {}^cG\right), \text{ s.t } d < \tau \tag{19}$$

IV. Experiment

A. Dataset and Setting

We evaluate the proposed algorithm using the KITTI dataset, which is a dataset for autonomous driving scenarios and contains complex road scenes acquired with 64-ring LiDAR. We use the sequence 00, 02, 05, 06, 07, 08, and each sequence are summarized in Table I.

In our experiments, the pair of point clouds with an Euclidean distance less than 3m is considered positive, representing a loop pair, and the others are considered negative. The neighboring point clouds have high similarity, and to avoid being judged as a positive pair, we consider a positive pair to be separated by 30s, the number of candidates is 10 and set $L_{max} = 50$, $N_s = 60$, $N_r = 20$.

TABLE II
F_1 MAX SCORES ON KITTI DATASET.

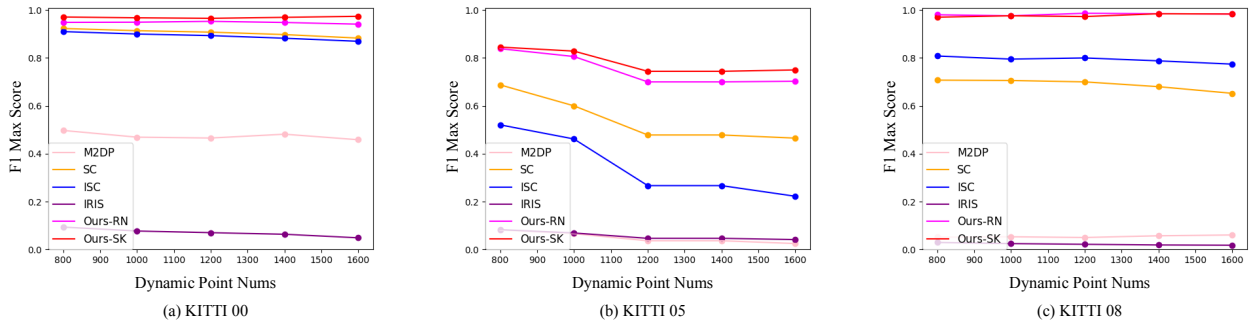
Methods	00	02	05	06	07	08	Mean
M2DP [7]	0.740	0.670	0.520	0.781	0.124	0.179	0.502
SC [10]	0.972	0.849	0.935	**0.998**	0.783	0.654	0.865
ISC [17]	0.970	0.814	0.887	0.976	0.739	0.764	0.858
IRIS [11]	0.975	**0.877**	0.967	0.991	0.833	0.680	0.887
Ours-RN	0.976	0.826	0.964	0.993	**0.923**	0.836	0.920
Ours-SK	**0.984**	0.843	**0.970**	0.991	0.915	**0.897**	**0.933**

B. Precision Recall Evaluation

We compare the proposed method with the state-of-the-art methods, including M2DP [7], SC [10], ISC [17] and IRIS [11]. We use the maximum F_1 score to evaluate the performance defined as:

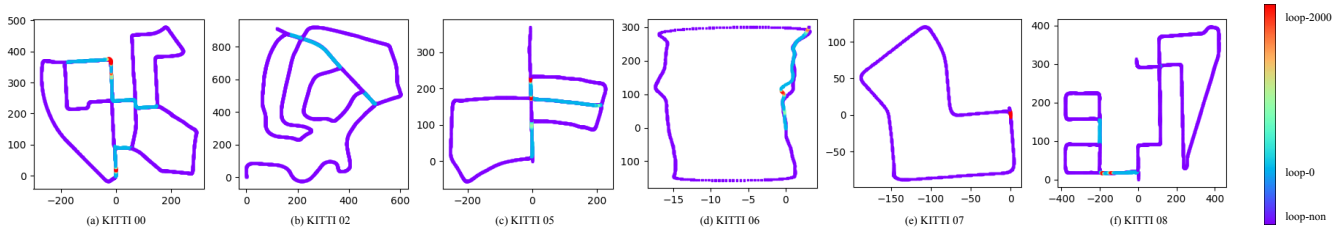
$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{20}$$

The results are shown in Fig.6 and Table II, and our method achieves most of the maximum F_1 max scores compared with other methods. The 08 sequence has a large number of reverse loops and the M2DP, using normal vector projection, degrades in this scene. The distribution of height and intensity information between different scenes is similar, which makes SC and ISC perform poorly, and IRIS is also affected by the lack of features. Our method considers the semantic and topological information of scenes, which can distinguish between scenes with similar feature distributions and achieves significant advantages in the 08 sequence. The 07 sequence contains a few loops, and the experimental result demonstrates that our method has a better ability to distinguish a loop. There is a long narrow road in the 02 sequence, and our method uses a bird's eye view for



(a) KITTI 00 (b) KITTI 05 (c) KITTI 08

Fig. 7. F_1 max scores in different dynamic scenes.



(a) KITTI 00 (b) KITTI 02 (c) KITTI 05 (d) KITTI 06 (e) KITTI 07 (f) KITTI 08

Fig. 8. Loop visualization. Depending on the number of dynamic point clouds, we show the non-loop(loop-non), static loop(loop-0), and dynamic loop(from loop-0 to loop-2000) for 00, 02, 05, 06, 07, and 08 sequences on the KITTI dataset.

TABLE III

AVERAGE YAW ERROR ON KITTI DATASET.

Methods	Dynamic							Static+Dynamic						
	00	02	05	06	07	08	Mean	00	02	05	06	07	08	Mean
SC(rad)	0.184	0.000	2.129	**0.125**	1.552	2.018	1.001	0.256	1.180	0.551	**0.025**	1.552	1.287	0.809
ISC(rad)	0.160	0.000	3.152	0.274	2.000	1.499	1.181	0.296	1.475	0.755	0.071	2.000	1.396	0.999
Ours-RN(rad)	0.124	0.000	1.835	0.275	0.673	**0.727**	0.606	0.241	1.233	0.561	0.048	0.673	1.599	0.725
Ours-SK(rad)	**0.103**	0.000	**1.585**	0.273	**0.199**	0.870	**0.505**	**0.226**	**1.083**	**0.492**	0.070	**0.199**	**1.204**	**0.546**

projection, resulting in lost information. IRIS achieves better performance in the 02 sequence because it has a higher column resolution, but this results in the matching time increasing rapidly.

As presented in the table, the performance of Ours-RN (using RangeNet++) is lower than Ours-SK (using SemanticKITTI) but still satisfactory. This shows that the proposed approach can be applied to real-world scenarios and that better semantic segmentation results can contribute to higher accuracy. The results indicate that our method is effective for loop closure detection.

C. Dynamic Scenes Performance

loop visualization is shown in Fig.8. We choose sequences 00, 05, and 08, which contain a large number of dynamic point clouds in loops, as the evaluation sequences for this section. According to the number of dynamic point clouds, we extract the dynamic scenes and calculate the corresponding F_1 max score for the candidates. If the algorithm has a fast search strategy, candidates are used directly. Otherwise, we construct candidates for each scene, which are composed as follows: if the scene has loops, half of the candidates are selected from loops randomly, and the rest are selected randomly from the previous scenes. If the scene does not have a loop, then all candidates are chosen randomly from the previous scenes.

Dynamic objects cause movement and occlusion of point clouds in the environment, and extracting features directly from the raw point clouds will generate the wrong descriptor and lead to matching failure. As shown in Fig.7, our method has the highest F_1 scores compared with other methods in each of dynamic point clouds, indicating that our method performs better in dynamic scenes. In the 00 and 08 sequences, the dynamic point clouds are composed of multiple small objects, which attenuate the dynamic influence, hence the curve decreases slowly. In the 05 sequence, the dynamic point clouds include large objects that cause significant occlusion and interference. The scores of all methods decrease as the number of dynamic point clouds increases, but our method decreases more slowly and has better resistance to dynamic objects. We believe that having no point clouds is better than having the wrong point clouds for loop closure detection, and our method demonstrates this with its robustness to dynamic scenarios.

D. Pose Align Accuracy

The proposed approach is egocentric and uses topological distances, with the ability to estimate the relative transformation of the yaw angle while detecting the loop. We compare our method with SC and ISC, which are also able to correct the offset angle. We select the loops that are correctly detected and calculate the average angular error between

TABLE IV

F_1 MAX SCORES AND THE COMPARISON WITH VIEWPOINT CHANGES.

VPC (x,y,yaw)	Methods	FMSc							RAT						
		00	02	05	06	07	08	Mean	00	02	05	06	07	08	Mean
(0m,0m,180deg)	SC	0.972	**0.849**	0.935	**0.998**	0.783	0.654	0.865	0.000	**0.000**	0.000	**0.000**	0.000	0.000	0.000
	ISC	0.970	0.814	0.887	0.976	0.739	0.764	0.858	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Ours-RN	0.976	0.826	0.964	0.993	**0.923**	0.836	0.920	0.000	0.000	0.000	0.000	**0.000**	0.000	0.000
	Ours-SK	**0.984**	0.843	**0.970**	0.991	0.915	**0.897**	**0.933**	**0.000**	0.000	**0.000**	0.000	0.000	**0.000**	**0.000**
(1m,0m,0deg)	SC	0.957	**0.841**	0.928	**0.996**	0.792	0.637	0.859	0.016	0.010	**0.008**	**0.002**	0.012	0.026	0.012
	ISC	0.949	0.821	0.870	0.952	0.766	0.762	0.853	0.021	0.008	0.019	0.024	0.036	**0.003**	0.018
	Ours-RN	0.963	0.806	0.945	0.989	0.893	0.823	0.903	0.014	0.024	0.019	0.004	0.033	0.016	0.018
	Ours-SK	**0.974**	0.839	**0.956**	0.987	**0.923**	**0.873**	**0.925**	**0.010**	**0.004**	0.014	0.004	**0.009**	0.027	**0.011**
(-1m,0m,0deg)	SC	0.961	**0.849**	0.928	**0.994**	0.792	0.645	0.862	0.012	**0.001**	**0.007**	**0.004**	**0.012**	0.014	**0.008**
	ISC	0.953	0.815	0.874	0.966	0.636	0.762	0.834	0.017	0.001	0.015	0.010	0.139	**0.003**	0.031
	Ours-RN	0.967	0.810	0.953	0.987	**0.857**	0.808	0.897	**0.009**	0.019	0.012	0.006	0.071	0.033	0.025
	Ours-SK	**0.974**	0.822	**0.957**	0.987	0.852	**0.877**	**0.912**	0.010	0.024	0.013	0.004	0.069	0.022	0.024
(0m,1m,0deg)	SC	0.937	0.722	0.883	**0.996**	0.711	0.653	0.817	0.036	0.150	0.056	0.002	0.091	**0.001**	0.056
	ISC	0.922	0.608	0.782	0.983	0.605	0.755	0.776	0.049	0.253	0.118	0.008	0.182	0.012	0.104
	Ours-RN	0.946	0.780	0.940	0.987	0.809	0.778	0.873	0.030	**0.056**	**0.025**	0.004	0.124	0.069	0.051
	Ours-SK	**0.968**	**0.793**	**0.943**	0.991	**0.923**	**0.838**	**0.909**	**0.016**	0.058	0.028	**0.000**	**0.009**	0.067	**0.030**
(0m,-1m,0deg)	SC	0.960	0.781	0.938	**0.996**	0.863	0.762	0.883	0.012	0.081	**0.003**	**0.002**	0.102	0.165	0.061
	ISC	0.893	0.254	0.861	0.918	0.816	0.836	0.763	0.079	0.688	0.028	0.058	0.104	0.093	0.175
	Ours-RN	0.969	0.859	0.948	0.989	**0.963**	0.846	0.929	0.007	0.040	0.016	0.004	0.043	**0.012**	**0.020**
	Ours-SK	**0.978**	**0.871**	**0.950**	0.981	0.943	**0.919**	**0.940**	**0.007**	**0.033**	0.021	0.009	**0.031**	0.025	0.021

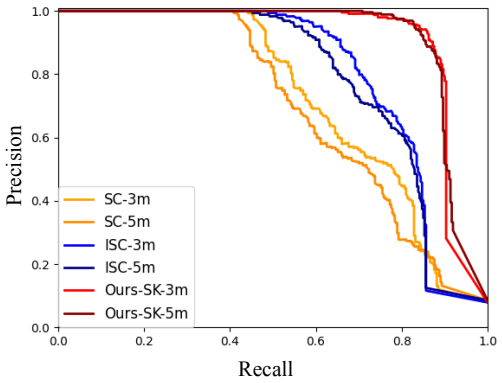


Fig. 9. Precision-Recall curves on KITTI08 with different distance thresholds.

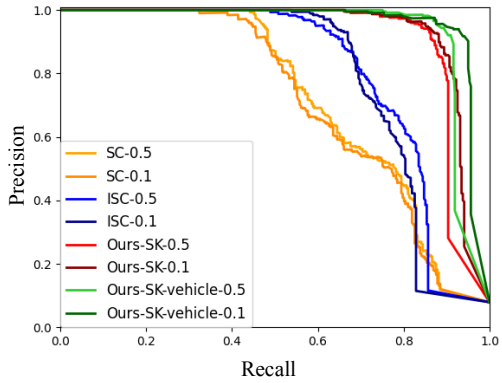


Fig. 10. Precision-Recall curves on KITTI08 with different semantics and resolution.

the estimates and the ground truth. We also select loops of dynamic scenes for evaluating angular errors, and the result shows that our method still achieves the best performance, as shown in Table III.

SC uses the average height values and ISC uses the geometry distributions to determine the optimal angle offset, but they both use statistical data, an approach that is not effective when the distribution of environmental information is similar. We use the topological distance between semantic objects for correction, which can include unique semantic information about the environment and better correct for angular errors.

E. Robustness Test

Viewpoints change: The viewpoint may differ even when arriving at the same place. To test the effectiveness of the approach with viewpoint changes(VPC), we rotate and translate the matched point clouds in the x(m), y(m), and yaw(deg) directions and recalculate the similarity. The results are shown in Table IV. We use FMS_c, and RAT to analyze the performance of the methods, where FMS_c is the F_1 max score(FMS) with the viewpoint changes and RAT is the ratio of the difference, defined as:

$$RAT = \frac{FMS - FMS_c}{FMS} \tag{21}$$

The result shows that all methods are invariant to a single rotation of the point clouds, but our method has a higher F_1 max score. In addition, SC and ISC divide the point clouds into different regions and express them with a single geometric value, which leads to confusion and loss of information after translating the point clouds in the x and y directions. Our method uses semantic and topological nodes to better express the features of point clouds in different regions and reduce the information overwritten. Overall, our method has the higher F_1 max score and the smaller RAT, making it more competitive in scenarios with viewpoint changes.

Distance threshold change: The distance threshold to determine whether the loop is positive or not affects the performance of the algorithm. We use distance thresholds of 3m and 5m and implement comparison experiments with SC and ISC using the 08 sequence, as shown in Fig.9. It can be seen that the curves of SC and ISC have large variations,

while our method has no significant changes, indicating that our method is less affected by the distance threshold.

Semantics and resolution change: In this experiment, considering the possibility that the vehicles are stationary, we add their point clouds to our method. In addition, we adjust the resolution of point clouds by downsampling using $0.5m^3$ and $0.1m^3$, and test the performance using the 08 sequence, as shown in Fig.10. We find that a small resolution is beneficial in reducing the degree of semantic confusion, allowing our method to achieve better performance. In the 08 sequence, there is a large number of parked vehicles, which can provide more complete information to the loop and make our method perform better with more semantic information.

F. Runtime

The average runtime is evaluated on sequence 00 with an Intel i7-8750 with 2.2 GHz and an NVIDIA GeForce RTX 2060 Super. The runtime for processing a scan consists of 83ms for semantic segmentation (using RangeNet++), 18ms for calculating descriptor and 45ms for searching loop, making it applicable in real-time robotics systems. It should be noted that the runtime is influenced by the semantic segmentation module heavily, so a fast semantic segmentation algorithm can greatly improve the efficiency of our method.

V. CONCLUSIONS

In this paper, we propose a novel descriptor for semantic topological graph based on 3D point clouds, design an efficient method for searching candidates and for similarity calculations to accomplish loop closure detection. Unlike previous works, we add topological information between objects at the semantic level, which allows a better representation of the uniqueness of the environment. In addition, we analyze and constrain dynamic objects and repetitive textureless point clouds during loop closure detection. Exhaustive evaluations demonstrate the accuracy and robustness of our method, especially in viewpoint changes and dynamic scenes.

REFERENCES

[1] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "Pl-slam: A stereo slam system through the combination of points and line segments," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 734–746, 2019.

[4] A. E. Johnson, "Spin-images: a representation for 3-d surface matching," 1997.

[5] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[6] M. Bosse and R. Zlot, "Place recognition using keypoint voting in large 3d lidar datasets," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2677–2684.

[7] L. He, X. Wang, and H. Zhang, "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)*. IEEE, 2016, pp. 231–237.

[8] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *2011 IEEE international conference on robotics and biomimetics*. IEEE, 2011, pp. 2987–2992.

[9] N. Muhammad and S. Lacroix, "Loop closure detection using small-sized signatures from 3d lidar data," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 2011, pp. 333–338.

[10] G. Kim and A. Kim, "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)*. IEEE, 2018, pp. 4802–4809.

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

[12] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

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

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

[15] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4470–4479.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

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

[18] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, "Segmap: 3d segment mapping using data-driven descriptors," *arXiv preprint arXiv:1804.09557*, 2018.

[19] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss, "Overlapnet: Loop closing for lidar-based slam," *arXiv preprint arXiv:2105.11344*, 2021.

[20] L. Li, X. Kong, X. Zhao, T. Huang, and Y. Liu, "Ssc: Semantic scan context for large-scale place recognition," *arXiv preprint arXiv:2107.00382*, 2021.

[21] Y. Zhu, Y. Ma, L. Chen, C. Liu, M. Ye, and L. Li, "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)*. IEEE, 2020, pp. 5151–5157.

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

[23] K. Vidanapathirana, P. Moghadam, B. Harwood, M. Zhao, S. Sridharan, and C. Fookes, "Locus: Lidar-based place recognition using spatiotemporal higher-order pooling," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5075–5081.

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

[25] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9297–9307.

[26] P. Huang, L. Lin, K. Xu, and H. Huang, "Autonomous outdoor scanning via online topological and geometric path optimization," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[27] S. Katz and A. Tal, "On the visibility of point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1350–1358.