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

Liao, M., Zhang, Y., Zhang, J., Cao, Z., Zhao, X., Coleman, S., & Kerr, D. (Accepted/In press). *Semantic Topological Descriptor for Loop Closure Detection within 3D Point Clouds In Outdoor Environment*. 1-8. Paper presented at IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyoto, Japan.

Link to publication record in Ulster University Research Portal

Publication Status:
Accepted/In press: 30/06/2022

Document Version
Peer reviewed version

General rights
Copyright for the publications made accessible via Ulster University's Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact pure-support@ulster.ac.uk.

Download date: 07/01/2023

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

Ming Liao[1], Yunzhou Zhang[1*], Jinpeng Zhang[1], Zhenzhong Cao[1], Xiaoyu Zhao[1],
Sonya Coleman[2], Dermot Kerr[2]

***Abstract*— Loop closure detection can correct the drift of trajectories and build a globally consistent map in LiDAR SLAM, which remains a challenging problem due to the sparsity of 3D point clouds data. In this paper, we propose a novel descriptor that contains semantic and topological information for loop closure detection. Unlike most existing methods, we directly discard point clouds representing people and vehicles during semantic segmentation, whether they are at a standstill or in motion. Then, our method generates a semantic topological graph representation for the static scenes. In addition, We propose a two-stage algorithm for finding the loop efficiently. Our method has been extensively experimented on the KITTI dataset and outperforms the state-of-the-art methods, especially in 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 the robot identify places visited previously, correct the accumulated drift error and build a globally consistent map to provide accurate prior information for autonomous driving.

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 because they can generate high-resolution 3D point clouds, have a larger field of view, and are not affected by illumination. The traditional LIDAR-based methods process the raw point clouds directly, divided into local descriptor [4]–[6] and global descriptor [7]–[11]. Traditional methods have good geometric descriptiveness for point clouds but are sensitive to occlusion and viewpoints change. To solve the uncertainty of geometric information, Some segment-based approaches [13]–[15] have been proposed. These types of methods 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, which can better describe the environment and are invariant to viewpoints change.

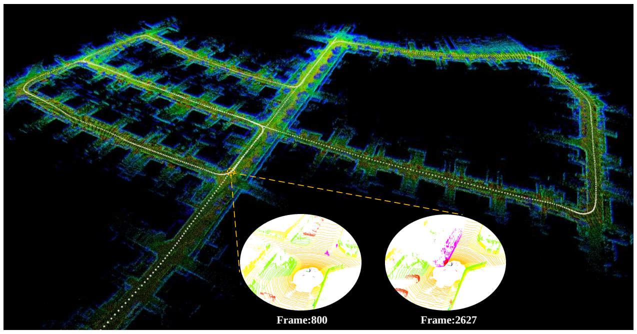


Fig. 1. This is an illustration of our proposed method. The map was created on 05 sequence using LEGO-LOAM [12] and our method. Note that there is a pair of loop between frames 2627 and 880, where a large portion of the scene is obscured by dynamic objects, making it challenging for existing methods.

In this paper, we propose a novel semantic topological descriptor for loop closure detection in dynamic scenes with a single 3D scan. After semantic segmentation of the point clouds, we discard the point clouds of vehicle, person, ground, and sidewalk to reduce the dynamic effects and computational burden. Then, feature points are obtained from the point clouds and the corresponding scores are calculated based on the semantic features and distance distribution. 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 dynamic outdoor scenes, we propose a semantic topological graph representation that incorporates the structural appearance, 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) For viewpoints change and dynamic scenes, our proposed method outperforms the state-of-the-art loop closure methods on the KITTI dataset [16].

II. RELATED WORK

According to the feature encoding methods, loop closure detection can be categorized into geometry-based methods, semantic-based methods, and graph-based methods.

[1] Ming Liao, Yunzhou Zhang, Jinpeng Zhang, Zhenzhong Cao and Xiaoyu Zhao are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (Email: zhangyunzhou@mail.neu.edu.cn).

[2] Sonya Coleman and Dermot Kerr are with School of Computing, Engineering and Intelligent Systems, Ulster University, N. Ireland, UK.

This work was supported by Natural Science Foundation of China (No. 61973066,61471110) , Major Science and Technology Projects of Liaoning Province(No.2021JH1/10400049), Fundation of Key Laboratory of Aerospace System Simulation(No.6142002200301), Fundation of Key Laboratory of Equipment Reliability(No.WD2C20205500306), Open Research Projects of Zhejiang Lab(No.2019KD0AD01/006) and Major Science and Technology Innovation Engineering Projects of Shandong Province(No.2019JZZY010128).

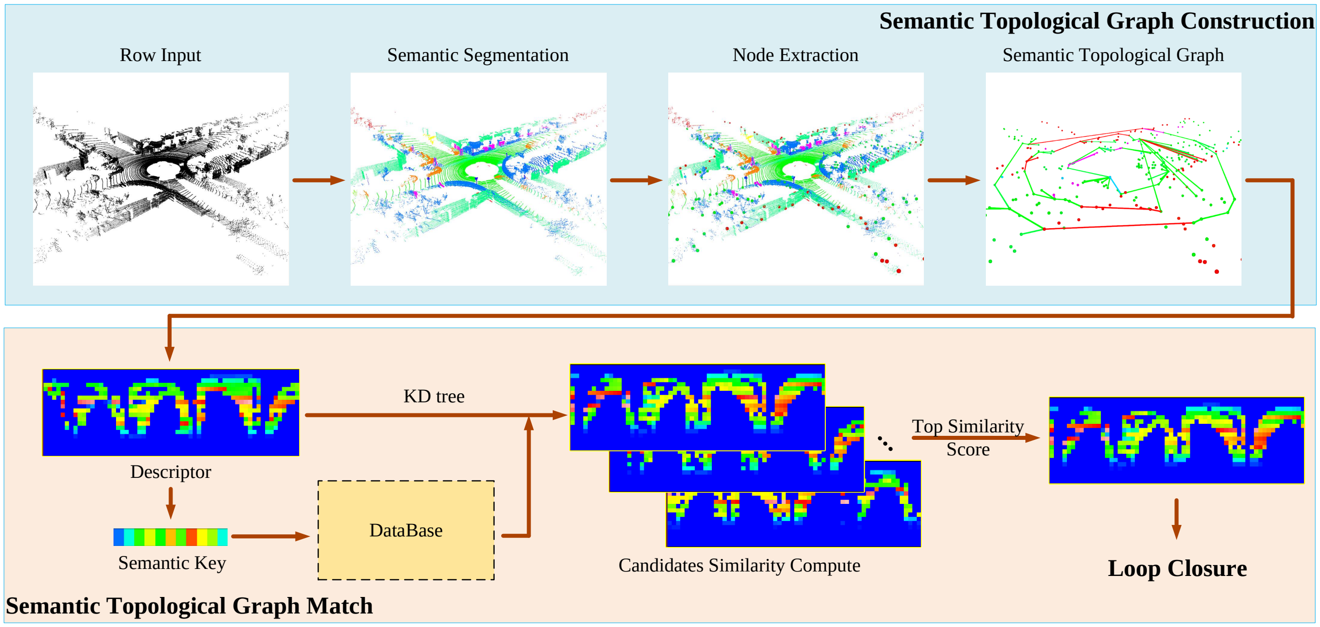

Semantic Topological Graph Construction
Row Input
Semantic Segmentation
Node Extraction
Semantic Topological Graph
KD tree
Descriptor
Semantic Key
DataBase
Top Similarity Score
Candidates Similarity Compute
Loop Closure
Semantic Topological Graph Match

Fig. 2. The framework includes the construction and match 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.

A. *Geometry-based methods*

Spin Image [4] projects the 3D point clouds of local regions into a 2D grid and matches the point clouds by comparing the 2D image similarity. SHOT [5] constructs a local reference frame (LRF) from the feature points and uses a local histogram for matching based on the normal vector. M2DP [7] projects the 3D point clouds from different viewpoints onto 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 round of LIDAR scanning into an egocentric 2D matrix, and stores the height value of the highest point in each bin for detecting loop. Iris [11] obtains binary feature images of point clouds by LoG-Gabor filtering and thresholding operations, uses 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 fast search.

The above methods use low-level information of the environment and can generate descriptor from raw point clouds data quickly, but the performance is limited by the sparsity of the data.

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 the neural network to estimate the overlap rate and yaw angle of LIDAR scans for determining loop. SSC [20] adds semantic information to the Scan Context [10], constructs global descriptor with semantic features, and improves the accuracy of place recognition by a two-step ICP method.

The above approaches use semantic information to express the environment with better robustness but do not consider the relationship between semantic objects, which is in the perception of humans distinguishing scenes.

C. *Graph-based methods*

GOSMatch [21] uses semantic segmentation to get the labels of point clouds, selects the clustering results of the parked vehicle, trunks and poles as nodes to construct a semantic topological graph, uses the distance histogram to search and get a 6-DOF initial pose estimate. However, there are a few semantic varieties in the graph and semantic information is not fully utilized. SPGR [22] and SA-LOAM [23] select 12 kinds of semantic information, input the clustered point clouds into a specially designed neural network for graph similarity matching. Their work represents large-scale objects with one semantic point, which cannot solve the problem that two segments of the same class. Locus [24] encodes the topological and temporal information of the point clouds after scene segmentation, aggregates the features using second-order pooling and generates 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 caused by 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. METHOD

In this section, 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: The raw 3D point clouds contain the geometric structure of the environment, from which semantic segmentation can extract high-level information to provide more robust constraints for loop closure detection. Rangenet++ [25] is a neural network for semantic segmentation of 3D point clouds. It uses the original scan as input and balances accuracy and speed. SemanticKITTI [26] provides accurate scan sequence labels based on the KITTI dataset, which includes 19 classes.

In the experimental, we use RangeNet++ and SemanticKITTI as semantic information input. In particular, we integrate the dynamic semantic labels into the corresponding static labels(for example, a moving car becomes a parked car) and discard them to create a static scene.

Semantic node extraction: Previous works use the centroids of point clouds as semantic graph nodes, which are not robust to large-scale scenes. In our approach, we extract feature points from the semantic point clouds as nodes, and large-scale point clouds of objects will be represented by multiple nodes. To avoid complex computation for extracting feature points, inspired by the online topological path optimization method [27], we use GHPR descriptor [28] which can determine the observability of point clouds by 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 raw point clouds set P_{origin}, we convert it to a new point clouds set P_{hull} and get the convex point set P_{convex}, the points are denoted as p_o, p_h, p_c. We transform each point p_o and viewpoint p_v using the following formulation:

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

where $(*)_{par}$ is the coordinates of the forward direction and f is a kernel function:

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

where scaling factor $\gamma = 10000$. We select K points p_h^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:

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

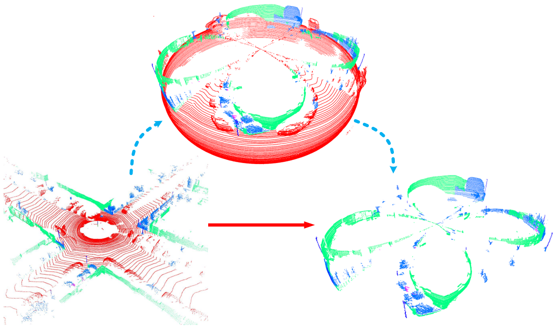


Fig. 3. An example of convex hull transformation. The raw point clouds (left figure below) are transformed to the convex hull (figure above), and it can be seen that the raw point clouds become smooth, which is beneficial to evaluate the observability of the point clouds. We discard the ground and vehicle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

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 d_{mean} 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:

$$\phi_1\left(p_h\right) = K\left(d_h, d_{mean}\right) = \exp\left(-\frac{\|d_h - d_{mean}\|_2^2}{2\sigma^2}\right) \quad (4)$$

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

$$\phi_2\left(p_h\right) = sigmoid\left(F_c\left(p_h, p_h^k\right)\right) \quad (5)$$

$$F_c\left(p_h, p_h^k\right) = \frac{\alpha}{K}\sum_{k=1}^{K} f\left(p_h, p_h^k\right) \quad (6)$$

$\phi_2\left(p_h\right)$ 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:

$$f\left(p_h, p_h^k\right) = \begin{cases} -l\left(p_h\right) \oplus l\left(p_h^k\right), & p_h \in P_{back} \\ l\left(p_h\right) \odot l\left(p_h^k\right), & p_h \in P_{front} \end{cases} \quad (7)$$

where $\oplus$ indicates that the same label is 1 and the difference is 0, and $\odot$ indicates that the different label is 0 and the same is 1.

Semantic topological graph: We combine the geometric and semantic score of points in a scan, which is defined by:

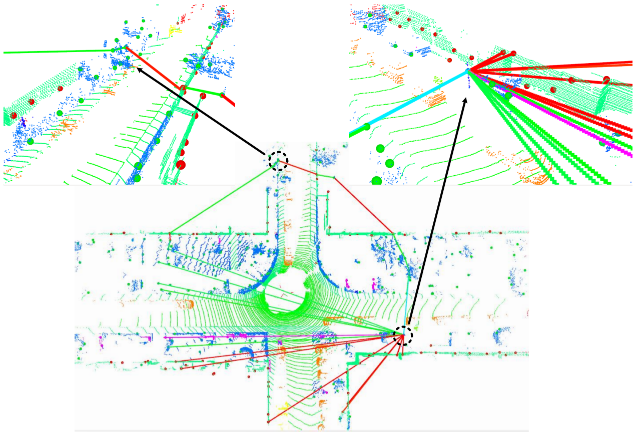


Fig. 4. An illustration of a semantic topological graph. The below figure represents a part of the semantic topological graph constructed by our method. Most of the nodes in a scene will be associated with the foreground nodes (above right), the same way that humans perceive the environment. In addition,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

$$\phi\left(p_h\right)=\phi_1\left(p_h\right)\cdot\phi_2\left(p_h\right) \tag{8}$$

We use the 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 coordinate, L_{max} is the maximum distance from the center, N_s and N_r are the number of sectors and rings. Each point p_c is projected into the corresponding bin in vertical direction, where the point $p^{i,j}\in P^{i,j}$ with the highest score represents the bin and becomes a semantic node $N^{i,j}$.

$$P_{convex}=\bigcup_{i\in[N_r]\; j\in[N_s]} P^{i,j} \tag{9}$$

$$N^{i,j}=\left\{p_c \,\middle|\, p_c=\max_{p_c\in P^{i,j}}\left(\phi\left(p_c\right)\right)\right\} \tag{10}$$

where symbol N_s is equal to $\{1,2,...,N_{s-1},N_s\}$ and symbol N_r is equal to $\{1,2,...,N_{r-1},N_r\}$. We construct the semantic topological graph by connecting semantic nodes with the largest semantic distance in the same radial, as shown in Fig.4.

B. Semantic Topological Graph Match

With a large number of places being visited, it is not reasonable to search 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 search. The first value of the vector is obtained from the first row, and the following values are obtained from the next row. The descriptor contains geometric and semantic information, so our method generates a N_{r+s} dimensional vector k after all rows are encoded:

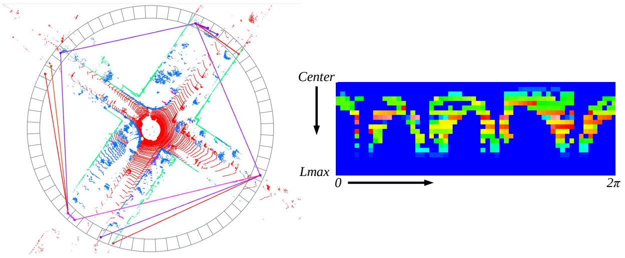


Fig. 5. The left figure is a part of the semantic topology graph, and the ring will be transformed into a row in the descriptor. The right figure is a descriptor of the whole semantic topology graph, a bin corresponds to a node in the descriptor, and the value of the bin is the semantic distance.

$$k=\begin{pmatrix}\varphi_g\left(1\right),...,\varphi_g\left(i\right),...,\varphi_g\left(N_r\right), \\ \varphi_s\left(1\right),...,\varphi_s\left(s\right),...,\varphi_s\left(N_s\right)\end{pmatrix} \tag{11}$$

$$\varphi_g\left(i\right)=\frac{1}{N_s}\sum_{j=1}^{N_s} d_g\left(N^{i,j}\right) \tag{12}$$

$$\varphi_s\left(s\right)=\frac{1}{N_s}\sum_{i=1}^{N_r}\sum_{j=1}^{N_s} d_s\left(N^{i,j}\right) \tag{13}$$

φ_g and φ_s are the functions for encoding geometric and semantic information, corresponding to the respective dimensions. d_g and d_s are the functions to calculate the semantic distance, defined as follows:

$$d_g=\left\{d \,\middle|\, d=\max_{m\in[N_s]}\left(\left|m-j\right|\cdot\phi\left(p^{i,m}\right)\right)\right\} \tag{14}$$

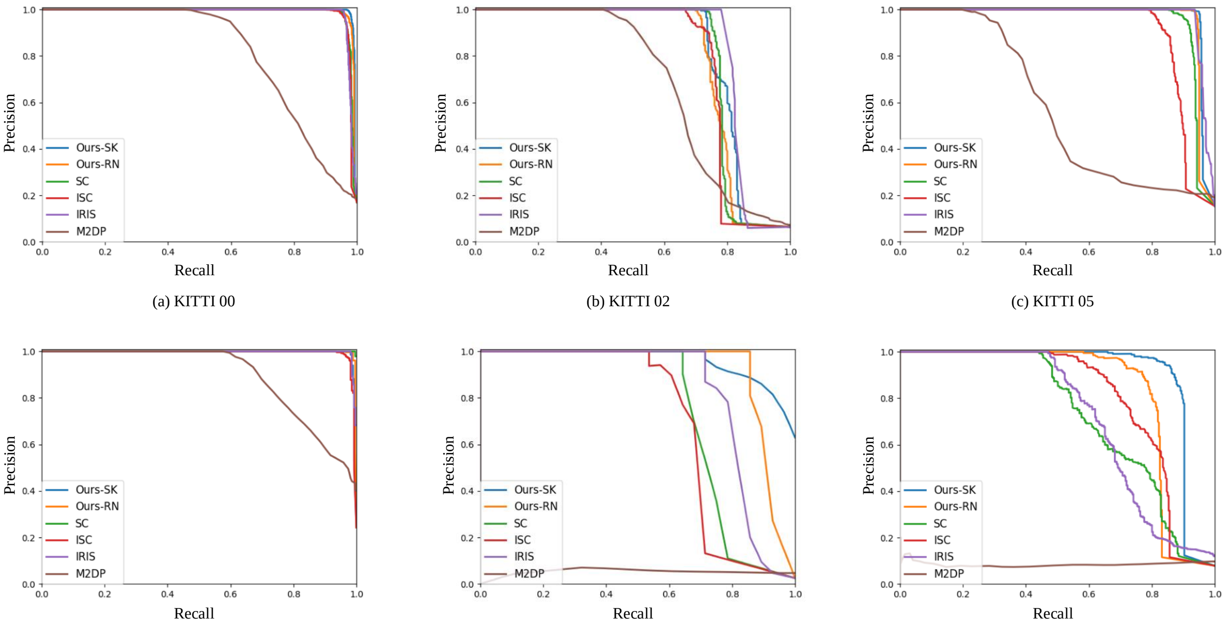
$$d_s=\left\{d \,\middle|\, \begin{array}{c} d=\max\limits_{m\in[N_s]}\left(\left|m-j\right|\cdot\phi\left(p^{i,m}\right)\right), \\ l\left(p^{i,m}\right)=s \end{array}\right\} \tag{15}$$

We construct a kd-tree and use vector k as the key to search out 10 candidates, which is provided to similarity calculation.

Similarity calculation: Given the query descriptor ${}^{q}G$ and candidate descriptor ${}^{c}G$, we need to calculate the similarity of two places. Due to the values of the descriptor being the topological distances between semantic nodes, we use the cosine function to calculate the difference. Semantic similarity is added as a constraint and the function is defined as follows:

$$d\left({}^{q}G,{}^{c}G\right)=\frac{1}{N_r}\sum_{i=1}^{N_r}\sum_{j=1}^{N_s}\varphi_d\left({}^{q}N^{i,j},{}^{c}N^{i,j}\right) \tag{16}$$

$$\varphi_d=\begin{cases}\frac{d_g\left({}^{q}N^{i,j}\right)\cdot d_g\left({}^{c}N^{i,j}\right)}{\left\|d_g\left({}^{q}N^{i,j}\right)\right\|\cdot\left\|d_g\left({}^{c}N^{i,j}\right)\right\|}, & {}^{q}l={}^{c}l \\ 0, & {}^{q}l\neq{}^{c}l\end{cases} \tag{17}$$



(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

The column vectors of the descriptor may be shifted in the same position due to the viewpoints change. To solve this problem, we extract the maximum distance value and minimize the Hamming distance to correct the shift.

$$D\left({}^{q}G,{}^{c}G\right)=\min_{n\in[N_s]} d'\left({}^{q}G,{}^{c}G\right) \tag{18}$$

$$d'\left({}^{q}G,{}^{c}G\right)=\ \min\left\|{}^{q}G'-{}^{c}G'\right\| \tag{19}$$

$$G'=\left[\max_{i\in[N_r]} d_g\left(N^{i,1}\right),\ldots,\max_{i\in[N_r]} d_g\left(N^{i,N_s}\right)\right] \tag{20}$$

d' is the loss function of Hamming distance, G' represents the rotated descriptor. The final loop is determined as:

$$c^{*}=\min_{c^{*}\in c} D\left({}^{q}G,{}^{c}G\right) \tag{21}$$

IV. Experiment

A. Dataset and Setting

We evaluate the proposed algorithm with 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.

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	0.565	0.781	0.124	0.567
IRIS [11]	0.968	**0.877**	0.967	0.991	0.833	0.680	0.886
ISC [17]	0.968	0.814	0.887	0.976	0.739	0.758	0.857
SC [10]	0.972	0.849	0.935	**0.998**	0.783	0.654	0.865
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**

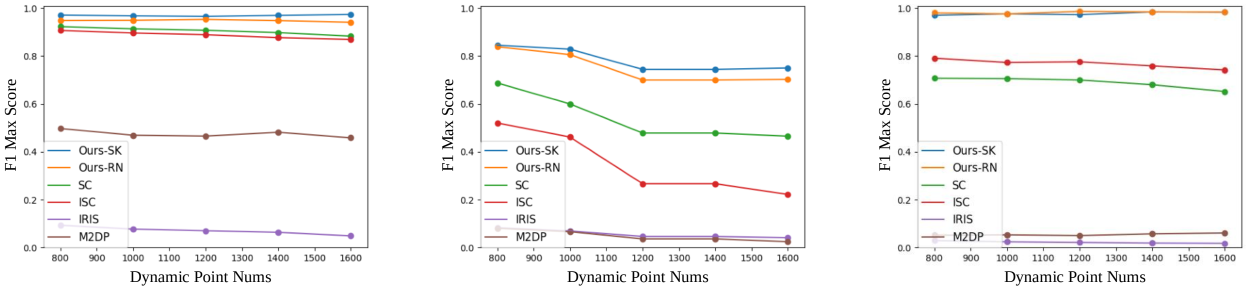
In our experiments, the pair of point clouds with Euclidean distance less than 3m is positive, representing a loop pair, and the others are negative. The neighboring point clouds have high similarity, and to avoid their being judged as a positive pair, we consider that a positive pair should be separated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

B. Precision Recall Evaluation

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

$$F_1 = 2\times\frac{precision\times recall}{precision+recall} \tag{22}$$

The result is shown in Fig.6 and Table II, our method achieves most of the best F_1 max score 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



(a) KITTI 00

(b) KITTI 05

(c) KITTI 08

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

TABLE III

AVERAGE YAW ERROR ON KITTI DATASET.

Sequences	SC(rad)	ISC(rad)	Ours(rad)
00	2.503	2.538	**2.444**
02	4.675	4.612	**4.580**
05	1.921	1.955	**1.872**
06	1.979	2.004	**1.932**
07	1.843	1.843	**1.801**
08	0.647	**0.612**	0.646

and topological information of scenes, which can distinguish scenes with similar feature distribution and achieves significant advantages in the 08 sequence. The 07 sequence contains a few loops, and the experiment result demonstrates that our method has a better ability to distinguish loop. There is a long narrow road in the 02 sequence, and our method uses a bird's eye view for projection, making the information lost. IRIS can achieve better performance in the 02 sequence because it has a higher resolution, but this can make the matching time increase rapidly.

As presented in the table, Ours-RN performance is lower than Ours-SK but still satisfactory. This shows that our work can be applied to real-world scenarios and that better semantic segmentation results can bring higher accuracy to our work. The results indicate that our method is effective for loop closure detection.

C. Dynamic scenes performance

We choose sequences 00, 05, and 08, which contain a large number of dynamic point clouds in loops, as the evaluation sequences in this section. According to the number of dynamic point clouds, we extract the dynamic scenes and calculate the corresponding F_1 max score with 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 cloud will generate the wrong descriptor and lead to matching failure. As shown in Fig.7, our method has the highest F_1 scores than other methods in each number 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, so 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 is robust to dynamic scenarios.

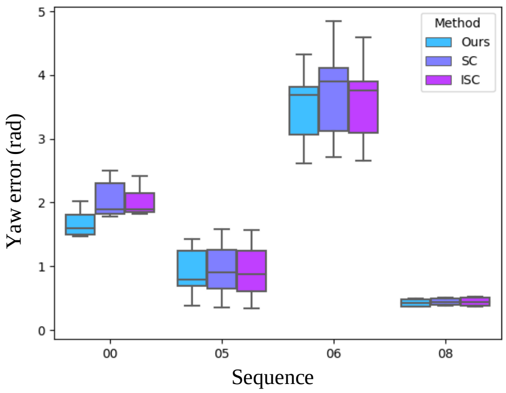


Fig. 8. Yaw error on KITTI dataset in different dynamic scenes.

D. Pose align accuracy

In the real world, the viewpoints may change even after arriving at the same place. 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 the estimates and the ground truth, and the results are shown in Table III.

SC uses the average height values and ISC uses the geometry distributions to find 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

TABLE IV

F1 MAX SCORES AND THE COMPARISON WITH VIEWPOINTS CHANGE.

VPC	Methods	FMS$_c$		COM			RAT	
(x,y,yaw)		00	02	05	06	07	08	Mean
(0,0,180)	Ours-SK	**0.984 0.000 0.000**	0.843 0.000 0.000	**0.970 0.000 0.000**	0.991 0.000 0.000	0.915 0.000 0.000	**0.897 0.000 0.000**	**0.933 0.000 0.000**
	Ours-RN	0.976 0.000 0.000	0.826 0.000 0.000	0.964 0.000 0.000	0.993 0.000 0.000	**0.923 0.000 0.000**	0.836 0.000 0.000	0.920 0.000 0.000
	SC	0.972 0.000 0.000	**0.849 0.000 0.000**	0.935 0.000 0.000	**0.998 0.000 0.000**	0.783 0.000 0.000	0.654 0.000 0.000	0.865 0.000 0.000
	ISC	0.970 0.000 0.000	0.814 0.000 0.000	0.887 0.000 0.000	0.976 0.000 0.000	0.739 0.000 0.000	0.764 0.000 0.000	0.858 0.000 0.000
(1,0,0)	Ours-SK	**0.974 0.010 0.010**	0.839 **0.004 0.004**	**0.956** 0.014 0.014	0.987 0.004 0.004	**0.923 0.008 0.009**	**0.873** 0.024 0.027	**0.925** 0.011 **0.011**
	Ours-RN	0.963 0.014 0.014	0.806 0.020 0.024	0.945 0.019 0.019	0.989 0.004 0.004	0.893 0.030 0.033	0.823 0.013 0.016	0.903 0.017 0.018
	SC	0.957 0.015 0.016	**0.841** 0.009 0.010	0.928 **0.007 0.008**	**0.996 0.002 0.002**	0.792 0.009 0.012	0.637 0.017 0.026	0.859 **0.010** 0.012
	ISC	0.949 0.021 0.021	0.821 0.007 0.008	0.870 0.017 0.019	0.952 0.023 0.024	0.766 0.027 0.036	0.762 **0.002 0.003**	0.853 0.017 0.018
(-1,0,0)	Ours-SK	**0.974** 0.010 0.010	0.822 0.021 0.024	**0.957** 0.012 0.013	0.987 **0.004 0.004**	0.852 0.063 0.069	**0.877** 0.020 0.022	**0.912** 0.022 0.024
	Ours-RN	0.967 **0.009 0.009**	0.810 0.016 0.019	0.953 0.011 0.012	0.987 0.006 0.006	**0.857** 0.066 0.071	0.808 0.028 0.033	0.897 0.023 0.025
	SC	0.961 0.011 0.012	**0.849 0.000 0.001**	0.928 **0.007 0.007**	**0.994** 0.004 0.004	0.792 **0.009 0.012**	0.645 0.009 0.014	0.862 **0.007 0.008**
	ISC	0.953 0.017 0.017	0.815 0.001 0.001	0.874 0.013 0.015	0.966 0.009 0.010	0.636 0.103 0.139	0.762 **0.003 0.003**	0.834 0.024 0.031
(0,1,0)	Ours-SK	**0.968 0.016 0.016**	**0.793** 0.049 0.058	**0.943** 0.027 0.028	0.991 **0.000 0.000**	**0.923 0.008 0.009**	**0.838** 0.060 0.067	**0.909 0.027 0.030**
	Ours-RN	0.946 0.030 0.030	0.780 **0.046 0.056**	0.940 **0.024 0.025**	0.987 0.004 0.004	0.809 0.115 0.124	0.778 0.058 0.069	0.873 0.046 0.051
	SC	0.937 0.035 0.036	0.722 0.128 0.150	0.883 0.053 0.056	**0.996** 0.002 0.002	0.711 0.071 0.091	0.653 **0.001 0.001**	0.817 0.048 0.056
	ISC	0.922 0.048 0.049	0.608 0.206 0.253	0.782 0.104 0.118	0.983 0.008 0.008	0.605 0.134 0.182	0.755 0.009 0.012	0.776 0.085 0.104
(0,-1,0)	Ours-SK	**0.978 0.007 0.007**	**0.871 0.028 0.033**	**0.950** 0.020 0.021	0.981 0.009 0.009	0.943 **0.028 0.031**	**0.919** 0.022 0.025	**0.940** 0.019 0.021
	Ours-RN	0.969 0.007 0.007	0.859 0.033 0.040	0.948 0.016 0.016	0.989 0.004 0.004	**0.963** 0.040 0.043	0.846 **0.010 0.012**	0.929 **0.018 0.020**
	SC	0.960 0.012 0.012	0.781 0.069 0.081	0.938 **0.003 0.003**	**0.996 0.002 0.002**	0.863 0.091 0.102	0.762 0.108 0.165	0.883 0.048 0.061
	ISC	0.893 0.077 0.079	0.254 0.560 0.688	0.861 0.025 0.028	0.918 0.057 0.058	0.816 0.077 0.104	0.836 0.071 0.093	0.763 0.145 0.175

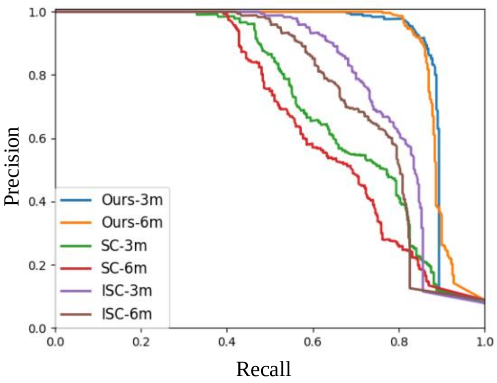


Fig. 9. Precision-Recall curves on KITTI08 with different distance thresholds.

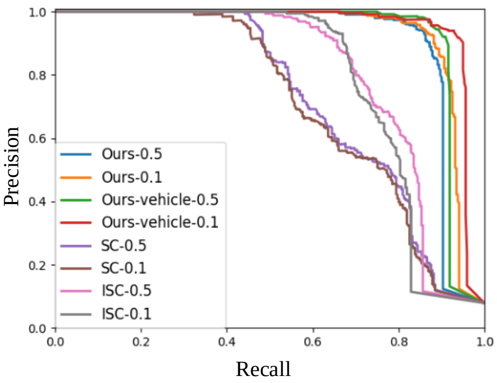


Fig. 10. Precision-Recall curves on KITTI08 with different semantics and resolution.

correction, which can include unique semantic information about the environment and better correct for angular errors.

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 Fig.8.

E. Robustness Test

Viewpoints Change: The viewpoints may change even when arriving at the same place. To test the effect of viewpoints change, we rotate and translate the matched point clouds in the x(m), y(m), and yaw(°) directions, recalculate the similarity, and the results are shown in Table IV. We use FMS_c, COM, and RAT to analyze the performance of the methods, where FMS_c is the F1 max score with the viewpoints change, COM is the absolute value of the difference between F1 max score, and RAT is the ratio of COM to F1 max score, is defined as follows:

$$RAT = \frac{COM}{FMS} = \frac{FMS - FMS_c}{FMS} \tag{23}$$

The result shows that all methods are invariant to a single rotation of the point clouds, but our method has a higher F1 max score. In addition, SC and ISC divide the point cloud 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 remains a higher F1 max score and has a smaller COM and RAT, making it more competitive in scenarios with viewpoints change.

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 6m and implement comparison experiments with SC and ISC in 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 vehicle is stationary, we add their point clouds to our method. In addition, we adjust the resolution of point clouds downsampling by using 0.5 m and 0.1 m, and test the performance in the 08 sequence, as shown in Fig.10. All methods are insensitive to the resolution, and 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 are a large number of parked vehicles, which can provide more complete information to the loop and make our method work better with more semantic information.

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 similarity calculation 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 in loop closure detection. Exhaustive evaluations demonstrate the accuracy and robustness of our method, especially in 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] L. Li, X. Kong, X. Zhao, W. Li, F. Wen, H. Zhang, and Y. Liu, "Sa-loam: Semantic-aided lidar slam with loop closure," *arXiv preprint arXiv:2106.11516*, 2021.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.