SSC: Semantic Scan Context for Large-Scale Place Recognition

Lin Li[1], Xin Kong[1], Xiangrui Zhao[1], Tianxin Huang[1], Wanlong Li[2], Feng Wen[2], Hongbo Zhang[2] and Yong Liu[1,*]

***Abstract*— Place recognition gives a SLAM system the ability to correct cumulative errors. Unlike images that contain rich texture features, point clouds are almost pure geometric information which makes place recognition based on point clouds challenging. Existing works usually encode low-level features such as coordinate, normal, reflection intensity, etc., as local or global descriptors to represent scenes. Besides, they often ignore the translation between point clouds when matching descriptors. Different from most existing methods, we explore the use of high-level features, namely semantics, to improve the descriptor's representation ability. Also, when matching descriptors, we try to correct the translation between point clouds to improve accuracy. Concretely, we propose a novel global descriptor, Semantic Scan Context, which explores semantic information to represent scenes more effectively. We also present a two-step global semantic ICP to obtain the 3D pose (x, y, yaw) used to align the point cloud to improve matching performance. Our experiments on the KITTI dataset show that our approach outperforms the state-of-the-art methods with a large margin. Our code is available at: `https://github.com/lilin-hitcrt/SSC`.**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) has rapidly developed in recent decades as critical technologies for autonomous vehicles and robots. Place recognition 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 [1]. Place recognition is usually conducted by using images or point clouds. 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.

Most existing works on LiDAR-based place recognition 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 [2]–[6], normal [7], reflection intensity [7]–[10], etc. In recent years, with the development of point cloud deep learning, many LiDAR-based object detection [11] and semantic segmentation [12], [13] 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 [7], [14], [15].

For place recognition, when a robot passes through a place visited before, it does not mean that the two poses are the same. Instead, the robot may walk through the original area from any direction, and there may be a small amount of translation from the original position. Many existing works consider the robot's orientation, namely rotation, and realize the invariance of rotation [3], [4], [10], [14]. They may think that the small translation will not strongly impact the recognition result and therefore ignore it. However, we find that simply ignoring the translation for the scan context-based methods will greatly reduce the similarity of the positive samples, making them difficult to identify.

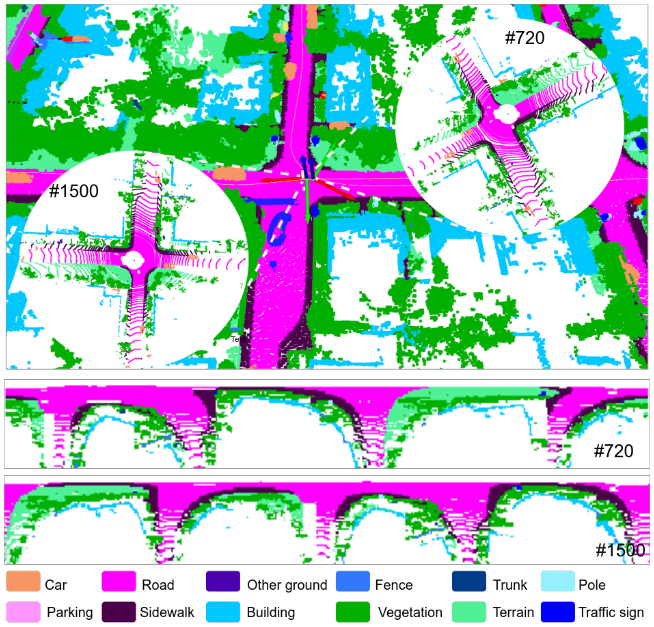


Fig. 1: An example of place recognition using semantic scan context. It is a partial map of the KITTI sequence 08, where the frames 720 and 1500 form a reverse loop. The lower part of the figure is the semantic scan context corresponding to the two frames. Since the directions of them are opposite, the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

In this paper, we propose a novel global descriptor named Semantic Scan Context (SSC), which explores semantic information to enhance the expressive power of descriptors. We also propose a two-step global semantic ICP that can produce reliable results regardless of the pose initialization, to obtain the 3D pose (x, y, yaw) of the point cloud. The pose is then used to align the point clouds to reduce the influence of rotation and translation on the similarity of the descriptors. Furthermore, it can also provide good initial values for 6D ICP algorithms to refine the global pose further. Fig. 1 is a demonstration of our results. The main contribution is summarized as follows:

- We propose a novel global descriptor for LiDAR-based

[1] Lin Li, Xin Kong, Xiangrui Zhao, Tianxin Huang and Yong Liu are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, P. R. China. (*Yong Liu is the corresponding author, email: yongliu@iipc.zju.edu.cn).

[2] Wanlong Li, Feng Wen and Hongbo Zhang are with Huawei Noah's Ark Lab, Beijing, China.

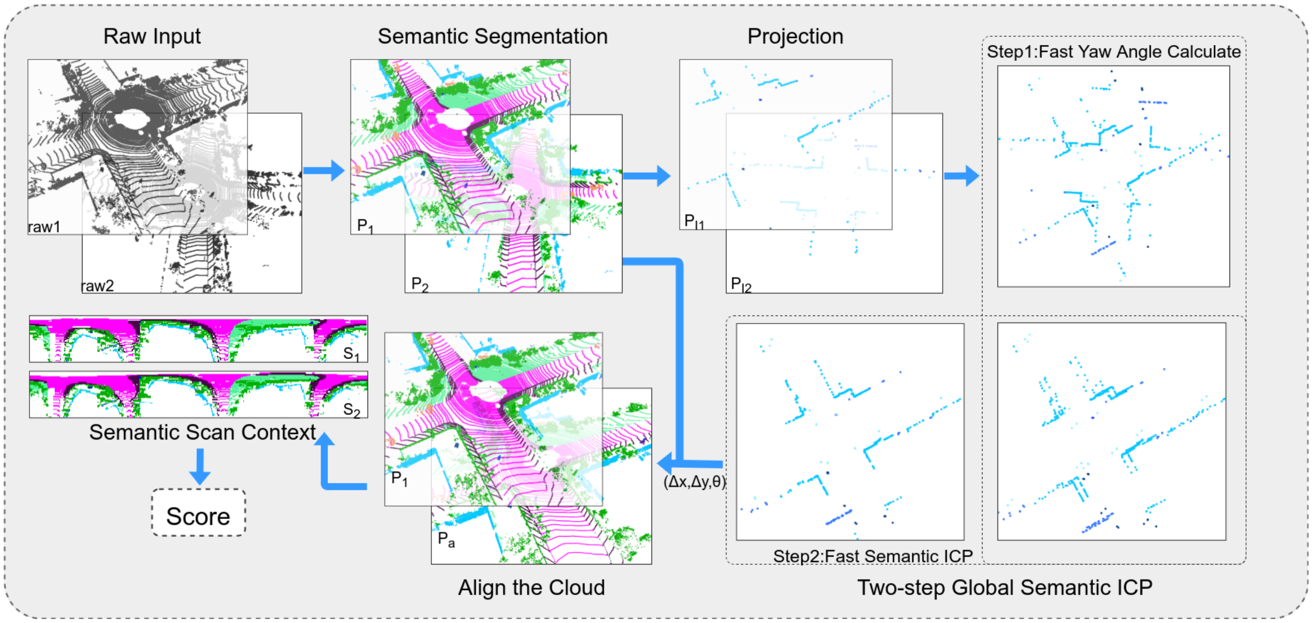


Fig. 2: The pipeline of our approach. It mainly consists of two parts: two-step global semantic ICP and Semantic Scan Context. First, we conduct semantic segmentation on the raw point cloud. Then we use semantic information to retain representative objects and project them onto the x-y plane. The two-step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

place recognition, which exploits semantic information to encode the 3D scenes effectively.

- We propose a two-step global semantic ICP, which doesn't require any initial values, to obtain the 3D pose (x, y, yaw) of the point clouds.
- We align point clouds with the obtained 3D poses to eliminate the influence of rotation and translation error on the similarity of the descriptors, which can also further benefit the SLAM system as good initial poses.
- Exhaustive experiments on the KITTI odometry dataset show that our approach achieves state-of-the-art performance both in place recognition and pose estimation.

II. RELATED WORK

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

Geometry-based methods: Spin image [2] 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 [16] proposes a shape descriptor that combines angle, point-distance, and area to boost the recognition rate. M2DP [5] 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 [3], [4] 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 [6] 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 [17] extracts a binary signature image for each point cloud then uses the Hamming distance of two corresponding binary signature images as the similarity. Seed [18] 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. It can be expected that 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 extracted by neural networks. Such 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 [9] and ISC [10] exploit the intensity information of the point cloud for place recognition. SegMatch [19] and SegMap [20] cluster a point cloud into segments. Then they extract features for each segment and use the kNN algorithm to identify corresponds. PointNetVLAD [21] combines PointNet [22] and NetVLAD [23] to extract global descriptors from the 3D point clouds end-to-end. L^3-Net [24] selects key points from the given point cloud then uses a PointNet to learn local descriptors for each key point. OREOS [25] projects the 3D point cloud into a 2D range image and proposes a convolutional neural network to extract the global descriptor. DH3D [26] 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 [27] 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 [28] uses a sparse voxelized point cloud representation and sparse 3D convolutions to compute a discriminative 3D point cloud descriptor. SeqSphereVLAD [29] projects the point cloud onto a spherical view, extracts features on it and sequences those features to form a descriptor. SpoxelNet [30] 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 [14] represents the scene as a semantic graph then uses a graph similarity network to score the similarity of the graphs. GOSMatch [15] 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 estimation. 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 [7] 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.

III. 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 3D pose space. 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.

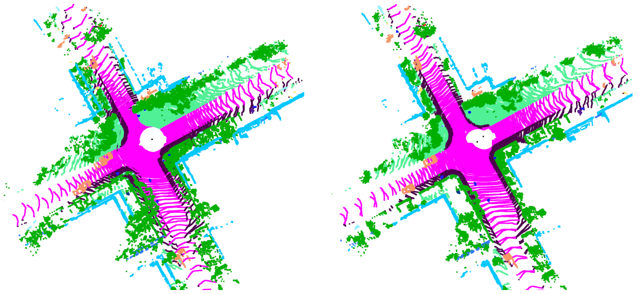

(a) Yaw Aligned
(b) Translation Aligned

Fig. 3: An illustration of the two-step global semantic ICP.

A. Global Semantic ICP

It is known that the general ICP algorithm based on local iterative optimization is susceptible to local minimums [31]. For place recognition, 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 representative objects such as buildings, tree trunks, and traffic signs based on semantic information. Then we convert the filtered clouds to polar coordinate in the x-y plane:

$$\begin{aligned} p_i &= [r_i, \varphi_i, x_i, y_i, \eta_i] \\ r_i &= \sqrt{x_i^2 + y_i^2} \\ \varphi_i &= arctan(\frac{y_i}{x_i}) \end{aligned} \tag{1}$$

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

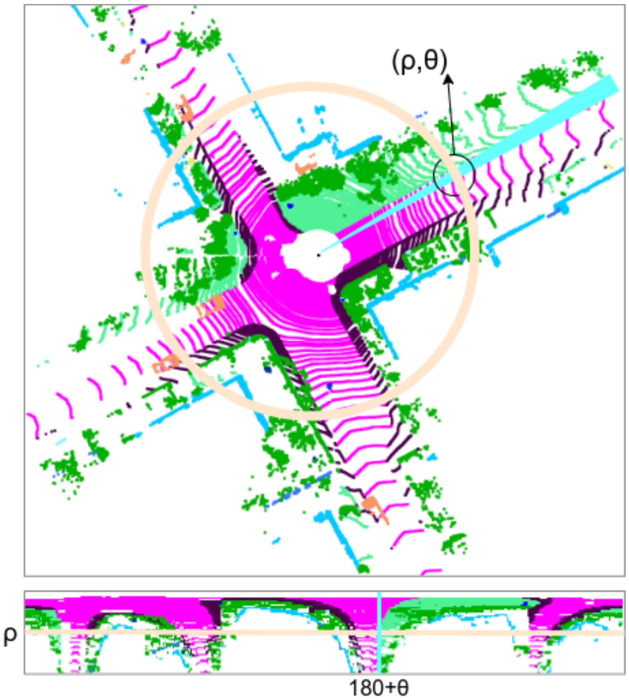


Fig. 4: An example of generating SSC. ρ and θ represent the polar diameter and polar angle, respectively. A sector corresponds to a descriptor column, while a ring corresponds to a row of the descriptor.

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:

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

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. Fig. 3 shows the result of Fast Yaw Angle Calculate.

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 can't 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:

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

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:

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

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, shown in Fig. 3.

B. 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.

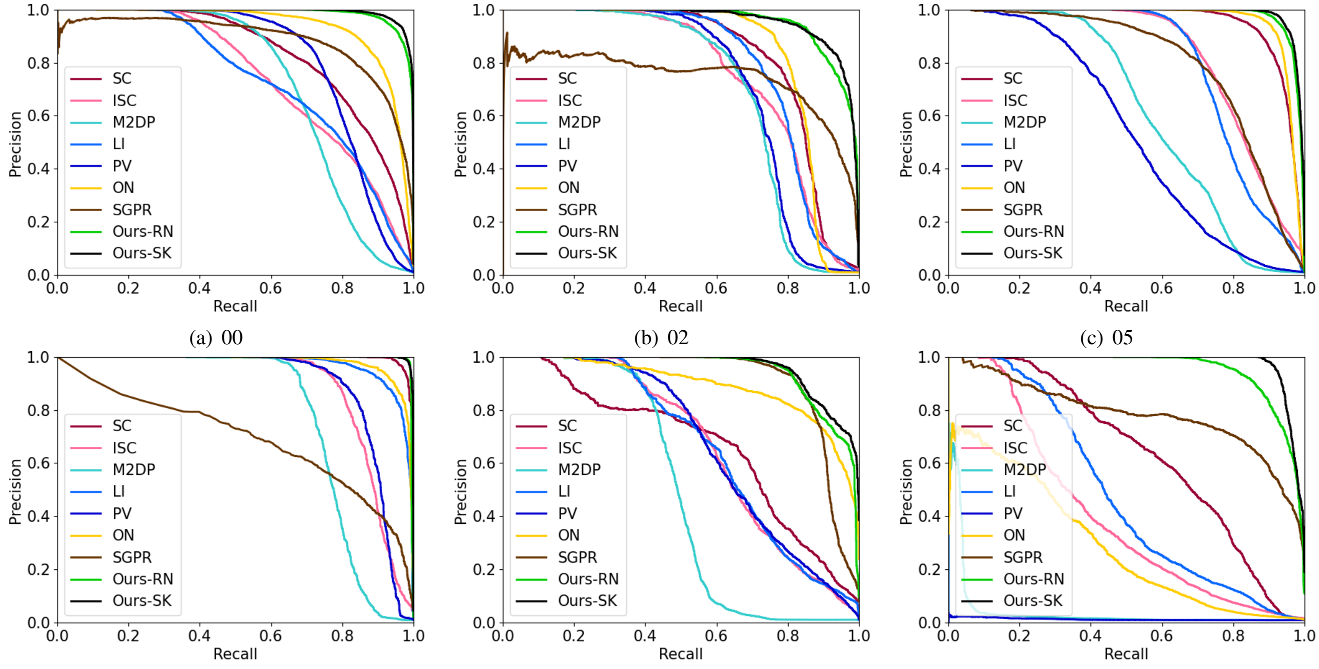
Descriptor definition. Given a point cloud P, we first convert it to the polar coordinate system as we did in Section III-A. Then, like scan context, we divide the point cloud into $N_s \times N_r$ blocks along the azimuthal and radial directions. Each block is represented by:

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

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:

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

f is an encoding function to encode features of B_{ij}. Note that if $B_{ij} = \varnothing$, $f(B_{ij}) = 0$. 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



(a) 00

(b) 02

(c) 05

(d) 06

(e) 07

(f) 08

Fig. 5: Precision-Recall curves on KITTI dataset.

TABLE I: F_1 max scores and Extended Precision on KITTI dataset

Methods	00	02	05	06	07	08	Mean
SC [3]	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 [10]	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 [5]	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 [17]	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 [21]	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 [7]	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 [14]	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.

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:

$$score = \frac{\sum\limits_{1 \le i \le N_r} \sum\limits_{1 \le j \le N_s} I(S_1(i,j) = S_2(i,j))}{\sum\limits_{1 \le i \le N_r} \sum\limits_{1 \le j \le N_s} I(S_1(i,j) \neq 0 \ or \ S_2(i,j) \neq 0)} \tag{9}$$

where I is the indicator function, defined by:

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

Fig. 4 shows Semantic Scan Context creation.

IV. EXPERIMENTS

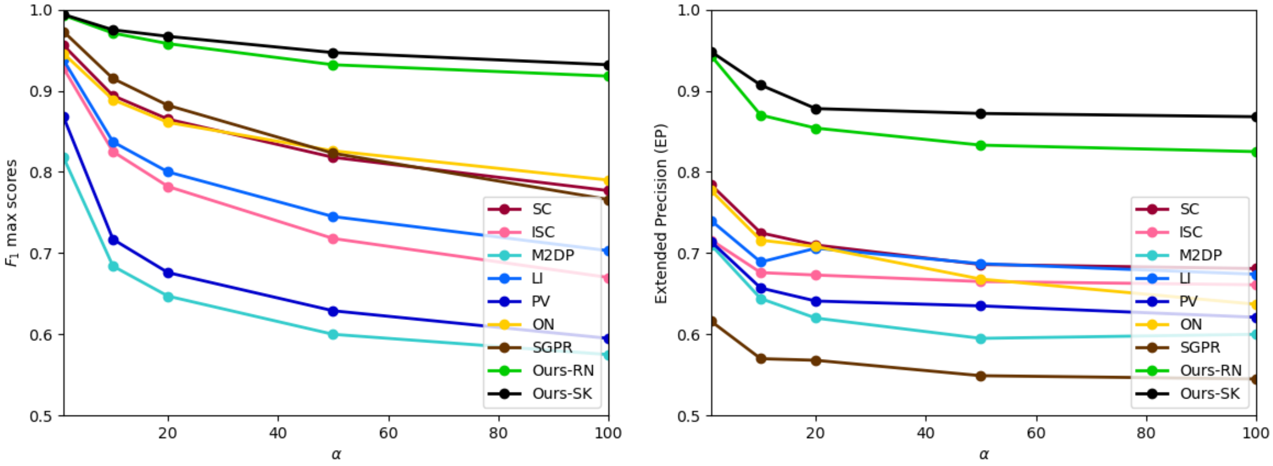
A. Experiment Setup

We conduct experiments on the KITTI odometry dataset [32] collected by a 64-ring LiDAR, which 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. Similar to SGPR [14], we regard the point cloud pair with a relative distance less (greater) than 3m (20m) as a positive (negative) sample. Since there are too many negative samples, we only select a part of the negative samples for evaluation. Specifically, if there are N_p positive samples 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.

The ground-truth semantic labels are from the SemanticKITTI dataset [33]. We also test our method with the semantic segmentation algorithm (RangeNet++ [34]) to prove that our method can be applied to noisy predictions in real situations. 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.

B. Place Recognition Performance

As mentioned in Section IV-A, we use both ground-truth semantic labels (Ours-SK) and predicted semantic labels



(a) Average F_1 max scores

(b) Average EP

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

(Ours-RN) for testing. We compare our approach with the state-of-the-art methods, including Scan Context [3] (SC), Intensity Scan Context [10] (ISC), M2DP [5], LiDAR Iris [17] (LI), PointNetVLAD [21] (PV), OverlapNet [7] (ON), and SGPR [14]. 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 $100N_p$. Fig. 5 shows the precision-recall curve of each method. Additionally, we also use the maximum F_1 score and Extended Precision [35] (EP) shown in Tab. I to analyze the performance. The F_1 score is defined as:

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{11}$$

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:

$$EP = \frac{1}{2}(P_{R0} + R_{P100}) \tag{12}$$

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 place recognition algorithms.

As shown in Fig. 5 and Tab. I, 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 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.

TABLE II: 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**

Change α. In this experiment, we change the value of α to analyze the influence of the number of negative samples on those algorithms. Fig. 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 place recognition, 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.

C. Pose Accuracy

As described in Section III-A, 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:

$$\begin{gathered} T = T_1^{-1} T_2 \\ (\Delta x, \Delta y, \theta) = (T(1,3), T(2,3), arctan(\frac{T(2,1)}{T(1,1)})) \end{gathered} \tag{13}$$

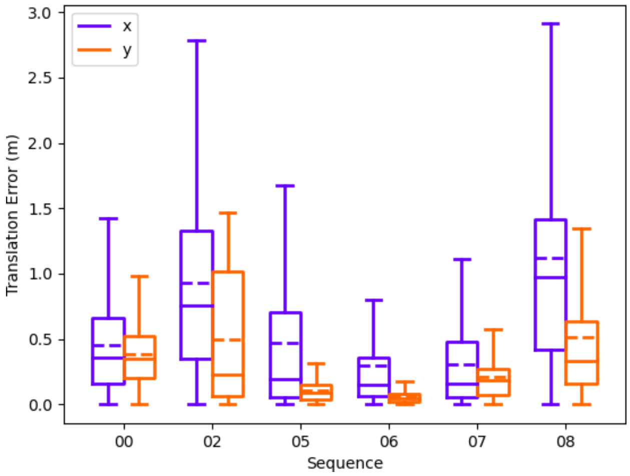


Fig. 7: Translation error.

TABLE III: 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%

where $T_1 \in SE(3)$ and $T_2 \in SE(3)$ 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.

Tab. II 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 Section IV-B, OverlapNet performs poorly due to its inability to handle reverse loops.

Fig. 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 3D 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.

D. Ablation Study

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 evaluate the contribution of our Fast Semantic ICP approach, we directly set Δx and Δy to 0. As shown in Tab. III, 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.

TABLE IV: 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.

E. 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 Tab. IV, the total time cost of our approach is acceptable. As we use the obtained 3D 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 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.

V. CONCLUSION

In this paper, we propose a novel semantic-based global descriptor for place recognition. We propose a two-step global semantic ICP to obtain the 3D pose (x, y, yaw) of the point cloud pair, aligning the point clouds to improve the descriptor matching accuracy. In addition, it can provide good initial values for point cloud registration. We achieve leading performance on the KITTI odometry dataset compared to the state-of-the-art methods.

Our method also has some limitations. Like most place recognition methods, our method does not consider pitch angle and roll angle. Therefore, our method may fail in some extreme scenarios.

In the future work, we will try to solve the above problems and further explore the application of semantic information in LiDAR-based SLAM systems.

References

[1] A. Angeli, D. Filliat, S. Doncieux, and J. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1027–1037, 2008.

[2] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[3] 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)*, pp. 4802–4809, 2018.

[4] G. Kim, B. Park, and A. Kim, "1-day learning, 1-year localization: Long-term lidar localization using scan context image," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1948–1955, 2019.

[5] 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)*, pp. 231–237, 2016.

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

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

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

[9] J. Guo, P. V. K. Borges, C. Park, and A. Gawel, "Local descriptor for robust place recognition using lidar intensity," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1470–1477, 2019.

[10] 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)*, pp. 2095–2101, 2020.

[11] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[12] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," *arXiv preprint arXiv:2011.10033*, 2020.

[13] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European Conference on Computer Vision*, 2020.

[14] 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)*, pp. 8216–8223, 2020.

[15] 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)*, pp. 5151–5157, 2020.

[16] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *2011 IEEE International Conference on Robotics and Biomimetics*, pp. 2987–2992, 2011.

[17] 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)*, pp. 5769–5775, 2020.

[18] Y. Fan, Y. He, and U. X. Tan, "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, 2020.

[19] 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)*, pp. 5266–5272, IEEE, 2017.

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

[21] 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*, pp. 4470–4479, 2018.

[22] 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*, pp. 652–660, 2017.

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

[24] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "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, 2019.

[25] L. Schaupp, M. Bürki, R. Dubé, R. Siegwart, and C. Cadena, "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, 2019.

[26] J. Du, R. Wang, and D. Cremers, "Dh3d: Deep hierarchical 3d descriptors for robust large-scale 6dof relocalization," in *European Conference on Computer Vision*, pp. 744–762, Springer, 2020.

[27] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y.-H. Liu, "Lpd-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, 2019.

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

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

[30] M. Y. Chang, S. Yeon, S. Ryu, and D. Lee, "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, 2020.

[31] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2241–2254, 2016.

[32] 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.

[33] 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 International Conference on Computer Vision*, pp. 9297–9307, 2019.

[34] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[35] B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. J. Milford, and K. D. McDonald-Maier, "Exploring performance bounds of visual place recognition using extended precision," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1688–1695, 2020.