SectionKey: 3-D Semantic Point Cloud Descriptor for Place Recognition

Shutong Jin*, Zhenyu Wu*, Chunyang Zhao, Jun Zhang, Guohao Peng, and Danwei Wang, *Fellow, IEEE*

***Abstract*— Place recognition is seen as a crucial factor to correct cumulative errors in Simultaneous Localization and Mapping (SLAM) applications. Most existing studies focus on visual place recognition, which is inherently sensitive to environmental changes such as illumination, weather and seasons. Considering these facts, more recent attention has been attracted to use 3-D Light Detection and Ranging (LiDAR) scans for place recognition, which demonstrates more credibility by exerting accurate geometric information. Different from pure geometric-based studies, this paper proposes a novel global descriptor, named *SectionKey*, which leverages both semantic and geometric information to tackle the problem of place recognition in large-scale urban environments. The proposed descriptor is robust and invariant to viewpoint changes. Specifically, the encoded three-layers key serves as a pre-selection step and a 'candidate center' selection strategy is deployed before calculating the similarity score, thus improving the accuracy and efficiency significantly. Then, a two-step semantic iterative closest point (ICP) algorithm is applied to acquire the 3-D pose (x, y, θ) that is used to align the candidate point clouds with the query frame and calculate the similarity score. Extensive experiments have been conducted on public Semantic KITTI dataset to demonstrate the superior performance of our proposed system over state-of-the-art baselines.**

I. Introduction

Place recognition is critical for various robotics missions (*e.g.,* loop closure detection in SLAM [1], global localization [2], [3], and collaborative mapping [4]–[6]). By detecting loop pairs, the autonomous robots can eliminate drifting errors and wrong registrations of certain landmarks, after which a globally consistent map can be created [7], [8]. Current studies on place recognition can be roughly categorized into image-based [9]–[12] and LiDAR-based methods [1], [7], [13]–[17]. Due to the intrinsic characteristics of camera, the image-based methods are greatly affected by external factors such as illumination and viewpoint variations [18]–[20], while LiDAR-based methods are relatively robust to appearance changes [21]. In light of this, there has been an increasing interest in LiDAR-based methods recently.

The majority of existing LiDAR-based place recognition algorithms works by encoding geometric features of point clouds into global or local descriptors, and then matching those descriptors. The current methods have been mostly restricted to exploring global or local features such as coordinates [13]–[15], surface characteristics [7], [22], and other low-level geometric features [16]. Currently, there are three challenging issues to be considered for LiDAR-based place recognition methods. First of all, most of the geometric-based solutions fail to build up the connection between human perspective and machine vision, thus leaving semantic information unused [18]. Secondly, the designed descriptors need to be rotational invariant to handle viewpoint changes [14]. Thirdly, the small translation between point clouds should not be neglected since it is likely to have strong impacts on place recognition results [20].

This paper aims to address the aforementioned issues of LiDAR-based place recognition. Although high-level semantic information has proven to be effective in boosting place recognition performances, semantic-based LiDAR place recognition methods are still few [17], [18], [20], [23] and how to use the semantics more effectively remains an open issue [24]. Inspired by how humans identify places, we propose a novel global descriptor named *SectionKey*, which leverages both semantic and geometric information,

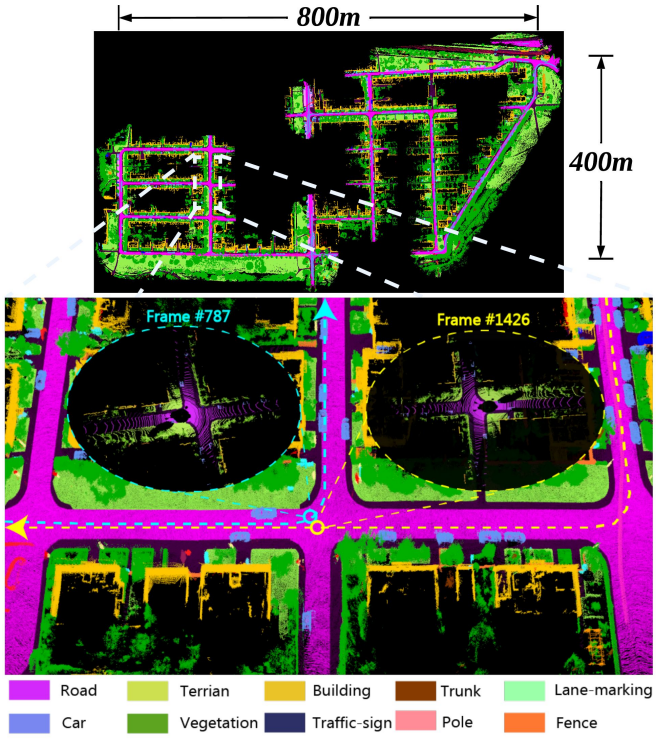


Fig. 1. An illustration of place recognition in urban environments, where a reverse place recognition task in sequence 08 of Semantic KITTI dataset is successfully accomplished by our proposed method. It is worth mentioning that the heading directions of frame #787 and frame #1426 are nearly the exact opposite, which poses great challenges to existing methods.

This research was supported in part by the National Research Foundation, Singapore under its Medium Sized Centre for Advanced Robotics Technology Innovation (CARTIN).

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. E-mail: shutong.jin1999@gmail.com; zhenyu002@e.ntu.edu.sg

*Co-first authorship and corresponding author

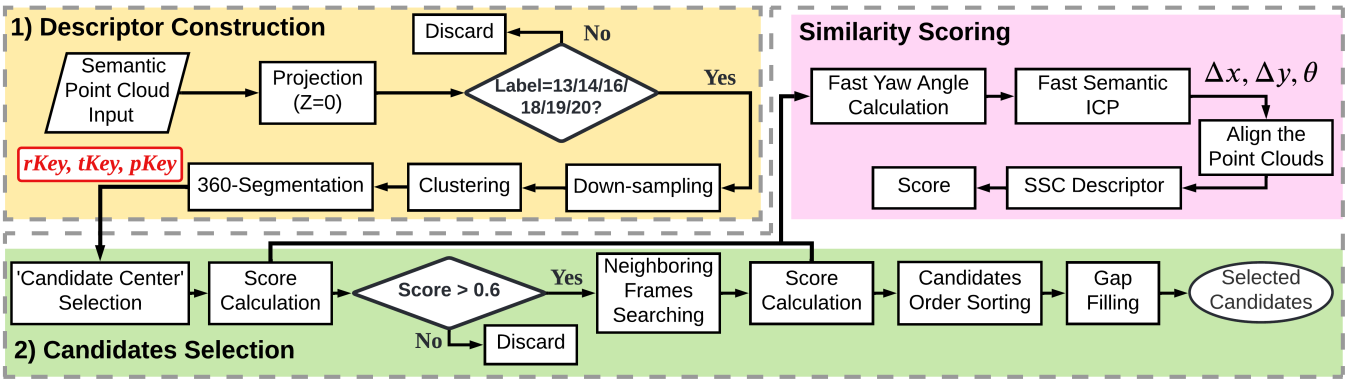


Fig. 2. Flowchart of the proposed system. The proposed system mainly consists of two submodules: 1) Descriptor Construction: construct the novel three-layers global descriptor *SectionKey* (*rKey, tKey, pKey*) with both semantic and geometric information. Label 13, 14, 16, 18, 19, 20 denotes the class of building, fence, trunk, pole, traffic-sign, and lane-marking, respectively; 2) Candidates Selection: select the candidate locations. The Similarity Scoring part denotes the semantic scan context score calculation module adopted from [20].

to address the problem of place recognition in large-scale urban environments. Prior to the encoding, we select points with certain labels and project them to the x-y plane, then perform the clustering. After the clustered point cloud being segmented azimuthally, we will encode them in both radical and peripheral way that eventually ending up with a rotation-invariant three-layers key, which serves perfectly as the pre-selection procedure. The main contributions of this paper are listed as follows:

- We propose the *SectionKey*, a novel rotation-invariant global descriptor with three-layers which exploits both semantic and geometric information.
- We formulate a novel place recognition framework, which comprises a *Descriptor Construction* submodule with a three-layers key and a *Candidates Selection* submodule with a 'candidate center' strategy.
- We perform extensive experiments on the Semantic KITTI dataset. Overall evaluations confirmed that the proposed method outperforms state-of-the-art baselines.

II. Methodology

In this paper, we propose a novel semantic global descriptor named *SectionKey* and the workflow of the proposed system is shown in Fig. 2. In the first *Descriptor Construction* part, we encode both the spatial and semantic information of point clouds to form a three-layers key, which serves perfectly as a pre-selection step. For the second *Candidates Selection* part, we perform an efficient 'candidate center' selection strategy, where the score calculation module from SSC [20] is adopted as an auxiliary scoring module.

A. Descriptor Construction

Preparations A frame of point cloud $\mathbf{C}$ can be defined as $\{c_1, c_2, ..., c_n \ | c_i \in \mathbf{C}\}$, which comprises a series of points within it. For point c_i, it is defined as (x_i, y_i, z_i, l_i), with geometric reading $[x_i, y_i, z_i]$ and semantic label l_i, as shown in Fig. 3(a). Then we will project the points with label $l_i = 13(building), 14(fence), 16(trunk), 18(pole), 19(traffic-sign), 20(lane-marking)$ to the x-y plane, as shown in Fig. 3(b), then convert them into polar coordinate as we only want to select the points that maintain static structural information. Suppose c_k is one of the projected points, we can obtain:

$$c_k = [x_k, y_k, \rho_k, \theta_k, l_k] \tag{1}$$

$$\rho_k = \sqrt{{x_k}^2 + {y_k}^2} \tag{2}$$

$$\theta_k = \arctan \frac{y_k}{x_k} \tag{3}$$

where $\rho_\mathbf{k}$ and $\theta_\mathbf{k}$ denote the polar diameter and polar angle, respectively. Then we will perform KD-Tree clustering for points with different labels respectively to make it unaffected by partial missing points. The clustering parameters set for different labels are listed in Tab. I. Note that in order to accelerate the process of clustering, points with label 13 and label 14 undergo a downsampling procedure before the clustering. As we mentioned before, we want to keep the crucial geometric information as well as reduce the computational load. Therefore, our algorithm preserves landmarks (pole, traffic-sign, lane-marking) as much as possible. This is why items like traffic-sign and lane-marking are easier to form a cluster and each cluster is represented by its average center. The frame after clustering is shown in Fig. 3(c), which will be further divided into 360 pieces, denoted as $s_1, s_2, ..., s_{360}$. The segmented frame is shown in Fig. 3(d).

Generation of *rKey* After the preparations, the first layer of encoding begins. The value assigned for each segment follows:

$$s_i = \begin{cases} 1, & \text{if there exists a cluster in this segment} \\ 0, & \text{else} \end{cases} \tag{4}$$

We name the first layer of descriptor as *rKey*, where r stands for radical. It is the summation of the value of each segment

TABLE I
Pameters of KD-Tree clustering for different labels

label	13	14	16	18	19	20
item	building	fence	trunk	pole	traffic-sign	lane-marking
tolerance	0.25	0.25	0.25	0.25	1	1
minSize	10	10	10	10	10	10
maxSize	5000	5000	5000	5000	5000	5000

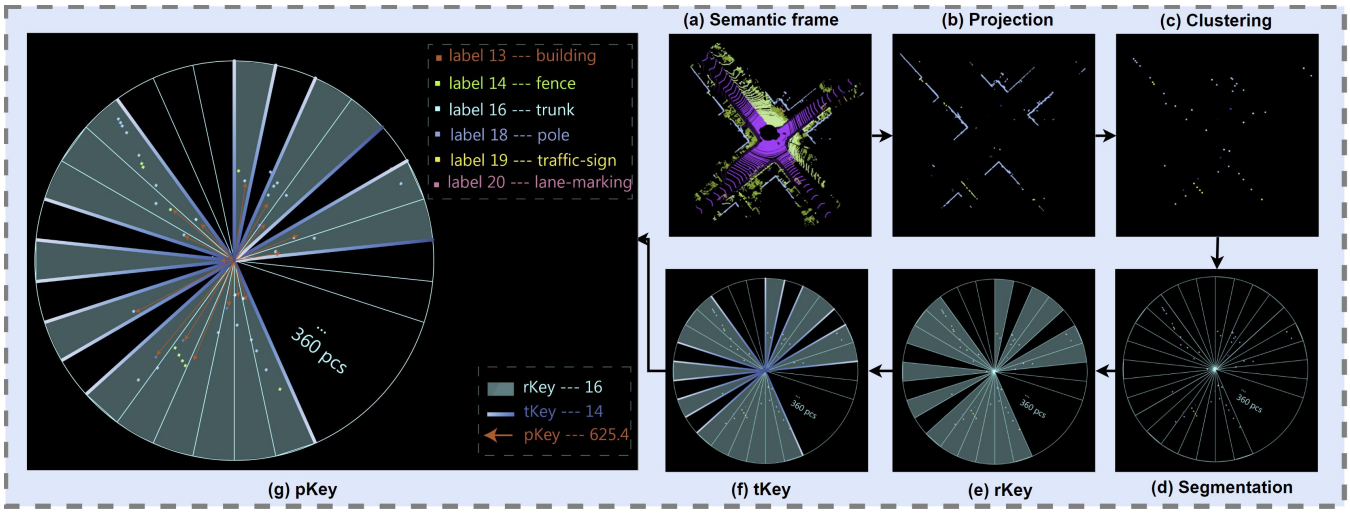


Fig. 3. Flowchart of the construction of *SectionKey*. (a) is a input frame of point cloud with semantic information. (b) is the projection of (a) with certain labels. (c) is (b) after the KD-Tree clustering. (d) is (c) being equally segmented into 360 pieces. (e) is an illustration of the *rKey*. (f) is a combined illustration of the *rKey* and *tKey*. (g) is a combined illustration of the *rKey*, *tKey*, and *pKey*. Here, all three keys are scalar.

as:

$$\mathbf{rKey} = \sum_{i=1}^{360} s_i \tag{5}$$

An illustration is shown in Fig. 3(e), where the shaded segments mean there exists at least one cluster in this area.

Generation of *tKey* The second layer of descriptor is named as *tKey*, which focuses on encoding the transition relation within the clustered point cloud. If there exists a difference between value of neighboring segments, namely there is one transition. And the value of *tKey* is the total number of transitions, which can be represented as:

$$\mathbf{tKey} = |s_1 - s_{360}| + \sum_{i=2}^{360}(|s_i - s_{i-1}|) \tag{6}$$

An illustration is shown in Fig. 3(f), where segment borderlines marked in bright blue denote where transition happens.

Generation of *pKey* To make the descriptor closer towards humanoid perception, the third layer of descriptor — *pKey*, is proposed to depict the relative topological distance. Suppose d_i is the distance between the center point and its nearest cluster in this segment. If there is no cluster within this segment, d_i is set to zero. The value of k depends on how many clusters are within this segment.

$$d_i = \begin{cases} 0, \text{ if there is no cluster in this segment} \\ min(\rho_1, ..., \rho_k), \text{ else} \end{cases} \tag{7}$$

$$\mathbf{pKey} = |d_1 - d_{360}| + \sum_{i=2}^{360}(|d_i - d_{i-1}|) \tag{8}$$

This three-layers global descriptor *SectionKey* not only enjoys robustness and rotation invariance that preserves the crucial geometric information, but also serves as a perfect pre-selection in the candidate selection part which will be detailed subsequently.

B. Candidates Selection

The candidate selection steps are summarized as follows:

1) **Set a highly selective threshold to find the potential 'candidate center'.** As mentioned earlier, a frame of point cloud could be encoded into three keys — *rKey*, *tKey*, and *pKey*. Only when all three keys of the frame accord with the threshold, it is recognized as a 'candidate center' as illustrated in Fig. 4. For the adjustable threshold, it is calculated based on the distribution of the keys of previously visited places.
2) **Check the validity of the 'candidate center'**. We will check the validity of those potential 'candidate centers' from two aspects. First of all, the temporal difference check. We check whether the 'candidate center' belongs to the adjacent frames of the query frame. Since the sensor feedback is time-continuous in a SLAM system [25], a single loop closure occurrence often indicates high similarity on the adjacent LiDAR scans. Therefore, if there is no temporal difference between the query frame and 'candidate center', then it is not qualified. Apart from this, we also adopt the similarity scoring module from SSC [20] to check the similarity between the query frame and 'candidate center'. Only when the similarity score is larger than 0.6, which is a empirically determined threshold, then the second requirement is considered satisfied. This is because the 'candidate center' stands for the starting point of the searching interval indicated in Step 3 below.
3) **Search the neighboring frames of the qualified 'candidate center'.** As the neighboring frames possess high similarity with the 'candidate center', they are also considered as the highly potential matching candidates of the query frame. In our algorithm, the number of candidates it will have is 0 if no loop closure is detected. Each query frame can have at most 25 candidate frames. And the candidates are sorted in a descending order based on their similarity score.

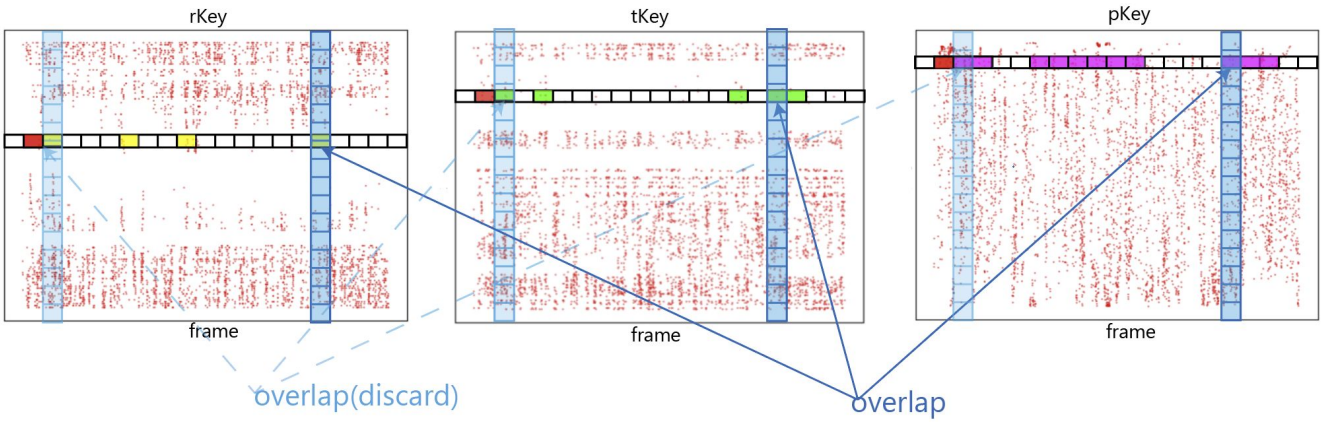


Fig. 4. An illustration of the candidate selection process in KITTI sequence 00. The horizontal X-axis direction stands for the sequence of the frames, where it is from 0 to 4540 in KITTI 00. The vertical Y-axis direction stands for the value of the Keys, which is an integer for both *rKey* and *tKey*, and a floating number for *pKey*. The horizontal red square stands for the query frame. Each dot represents the frame's value of the specific key, so there are 4541 dots in a graph for KITTI 00. Since the squares represent the frames, the number of squares should be equal to the number of dots. In this figure, the number of squares are largely reduced for better visualization. The horizontal squares filled in yellow, green, and purple denote the frames that accord with the threshold in the three Keys, respectively. Note that we only consider the overlapped frame which similarity score is higher than 0.6 and temporal difference is higher than 50 as the 'candidate center'. The vertical squares denote the search in the neighboring frames, which is conducted around the qualified 'candidate center'. The light blue candidate is discarded due to that it fails to meet the temporal difference requirement.

4) **Fill the gap.** As three selective thresholds and two check are set previously, the standard for becoming a qualified 'candidate center' is relatively high. Some query frames with ground-truth candidates may end up having no 'candidate center' selected. Due to the time consistency property, it is therefore necessary and reasonable to fill the gap between intensive qualified 'candidate centers'. To save computational time, the candidates picked for the frames in the 'gap' are from its neighboring 'candidate centers'. Here, the definition for 'gap' is the difference of the sequence between two adjacent 'candidate centers', which should be less than or equal to five. If the difference of the sequence is larger than five, then there is no need to fill it because the 'candidate centers' are not intense here.

C. Similarity Scoring Calculation

For the Similarity Scoring Module, we adopt it from SSC [20] due to its high precision on calculating the similarity between a pair of point cloud frames. The key points of SSC's mechanism for calculating similarity can be concluded as follows:

1) **Fast yaw angle calculation** Given one pair of point clouds $(\mathbf{C_1}, \mathbf{C_2})$ with semantic information, the representative objects are filtered out and they are converted into polar coordinates. Each converted point cloud is then segmented to N_s sectors by yaw angle, and only the smallest polar diameter in each segment is used to form 1-D vectors $\mathbf{P_1}$ and $\mathbf{P_2}$. The point cloud pair $(\mathbf{C_1}, \mathbf{C_2})$ now is transformed into a 1-D vector pair $(\mathbf{P_1}, \mathbf{P_2})$. Similar to Scan Context [14], the shift of column vector and yaw angle are obtained through:

$$\mathbf{shift} = \underset{i, i\in[0,N_s]}{\operatorname{argmin}} \Psi(\mathbf{P_1}, \mathbf{P_2^i}) \tag{9}$$

$$\theta = \mathbf{360} - \frac{\mathbf{360} \times \mathbf{shift}}{\mathbf{N}} \tag{10}$$

where $\mathbf{N}$ indicates the number of segments, $\mathbf{P_2^i}$ is $\mathbf{P_2}$ shifted by the i^{th} element and Ψ is defined as:

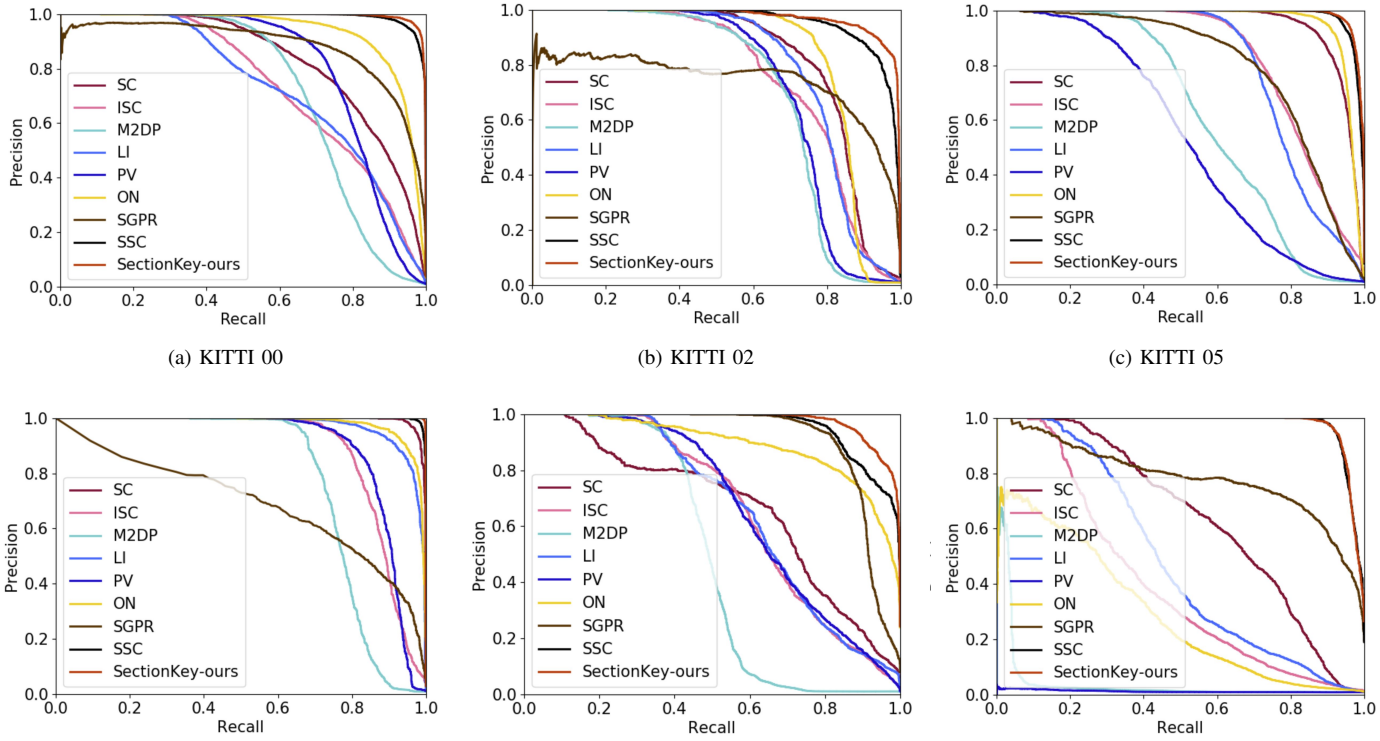
$$\Psi(\mathbf{P_1}, \mathbf{P_2^i}) = \left\| \mathbf{P_1} - \mathbf{P_2^i} \right\|_1 \tag{11}$$

2) **Translation calculation** Use the obtained yaw angle θ to align $\mathbf{P_2}$ to the same direction with $\mathbf{P_1}$, then the semantic ICP algorithm [20] can be performed to acquire the translation that minimize the difference between $\mathbf{P_1}$ and $\mathbf{P_2}$.
3) **Similarity scoring** The obtained yaw angle and translation are used to align two point clouds. Given aligned point cloud pair $(\mathbf{C_1}, \mathbf{C_a})$, divide them azimuthally and radially. The value for each small patch is the output of an encoding function. Finally, the output similarity score would be equal to the percentage of matched patches.

III. Experiments

A. Experimental Setting

The experiments were conducted on public Semantic KITTI Dataset [26], which contains 11 sequences (from 00 to 10) collected by a 64-line LiDAR with manual semantic annotation and ground-truth poses. The ground-truth poses are used to test the correctness of the place recognition candidates that each algorithm generated. To make our comparisons more convincing, the point cloud pair with a relative distance less (greater) than 3m (20m) is regarded as a positive (negative) sample, which is same to SSC [20]. In our experiments, the sequences with loop closure scenarios (*i.e.,* sequences 00, 02, 05, 06, 07, and 08) were chosen to evaluate the performance of our algorithm. Among them, only the sequence 08 has reverse loops, while other sequences only have loop events with the same direction. The dataset contains 28 classes which include classes to distinguish non-moving and moving objects, and we only select six classes (*i.e.,* building, fence, trunk, pole, traffic-sign, and lane-marking) that contain the most unique and static geometric information. All tests



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

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

Fig. 5. Precision-Recall curves of different methods on pairs selected from Semantic KITTI dataset, with negative : positive = 100 : 1.

were performed on a laptop with Intel Core i7-10750H CPU @ 2.60GHz and 16 GB RAM.

B. Place Recognition Performance on Pairs

1) Dataset: For fairness, our results are based on the same evaluation samples provided by SSC [20]. The list of evaluation samples selected by SSC are all positive samples, and some randomly-selected negative samples are based on a fixed ratio for evaluation. In this experiment, the pairs are from Semantic KITTI 00, 02, 05, 06, and 08 sequences. Based on the ratio of positive pairs over negative pairs, we divide the experiments into two categories: 1) neg_10: the number of negative pairs is 10 times of positive pairs; 2) neg_100: the number of negative pairs is 100 times of positive pairs. And in this section, we compare our system with state-of-the-art baselines, which include: 1) Scan Context (SC) [14]; 2) Intensity Scan Context (ISC) [7]; 3) M2DP [13]; 4) LiDAR Iris (LI) [16]; 5) PointNetVLAD (PV) [27]; 6) OverlapNet (ON) [17]; 7) SGPR [18]; and 8) SSC [20]. Similar steps are taken to process the OverlapNet (ON) [17] and SGPR [18] methods in accordance with SSC [20].

2) PR Curve: The performance of *SectionKey* for negative ratio equal to 100 is analyzed using the precision-recall curve as shown in Fig. 5. It is clear that our proposed *SectionKey* method outperforms all the comparative methods in all sequences and achieves large margin especially in sequence 02 and 07. SSC has comparable performances with our method in sequence 05 and 08. For OverlapNet, it is easy to find that its performance is severely degraded in sequence 08, which can be explained by its incapability to robustly handle reverse loop closures.

3) F_1 and EP: Apart from PR curve, we also introduce two more indicators, *i.e.,* the maximum F_1 score and Extended Precision [28], to evaluate the performance of our proposed *SectionKey* method, together with the aforementioned comparative methods. The definition if F_1 score and Extended Precision (EP) are described as follows:

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{12}$$

$$\text{EP} = \frac{1}{2}(P_{R0} + R_{P100}) \tag{13}$$

where F_1 is the harmonic mean of P and R, and Extended Precision is a specific metric designed for place recognition algorithms. P and R denote the Precision and Recall, respectively. Normally, precision rate goes down while recall rate increases due to the increasing threshold. Therefore, F_1 indicates the integrated optimal performance that one method can achieve. P_{R0} is the precision at the minimum recall, while R_{P100} is the maximum recall at 100% precision.

The F_1 score and Extended Precision for above mentioned methods are shown in Table. II. We can see from this table that our method achieves an overall leading performance. And especially in KITTI 00 and 07, where other methods' performance are not ideal and our proposed *SectionKey* still maintains an advantageous performance.

IV. Conclusion

In this paper, we presented a novel rotation-invariant global descriptor named *SectionKey*, which exploits both se-

TABLE II

F_1 MAX SCORES AND EXTENDED PRECISION ON SELECTED PAIRS (NEG_100) FROM KITTI SEQUENCES 00, 02, 05, 06, 07, AND 08.

Methods	KITTI 00	KITTI 02	KITTI 05	KITTI 06	KITTI 07	KITTI 08	Mean
SC [14]	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 [7]	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 [13]	0.708/0.616	0.717/0.603	0.602/0.611	0.787/0.681	0.560/0.586	0.073/0.500	0.575/0.516
LI [16]	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 [27]	0.779/0.641	0.727/0.691	0.541/0.536	0.852/0.767	0.631/0.591	0.037/0.500	0.595/0.538
ON [17]	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 [18]	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
SSC [20]	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
SectionKey-Ours	**0.996/0.882**	**0.918/0.749**	**0.956**/0.900	**0.995/0.992**	**0.911/0.818**	**0.940**/0.892	**0.948/0.872**

[1] 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.

mantic and geometric information to significantly boost place recognition performance in large-scale urban environments. The proposed system consists of a *Descriptor Construction* submodule with a three-layers key and a *Candidates Selection* submodule with a 'candidate center' strategy. Extensive experimental results on public Semantic KITTI dataset have demonstrated the superiority of the proposed system over existing methods in terms of accuracy and efficiency. Future work will be focusing on integrating the semantic and relative topological distance information, and achieving intelligent selection of geometric labels for different scenarios.

REFERENCES

[1] G. Kim, S. Choi, and A. Kim, "Scan Context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Trans. Robot. (TRO)*, pp. 1–19, 2021.

[2] X. Chen, T. Läbe, L. Nardi, J. Behley, and C. Stachniss, "Learning an overlap-based observation model for 3D LiDAR localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2020, pp. 4602–4608.

[3] Z. Wu, Y. Yue, M. Wen, J. Zhang, J. Yi, and D. Wang, "Infrastructure-free hierarchical mobile robot global localization in repetitive environments," *IEEE Trans. Instrum. Meas. (TIM)*, vol. 70, pp. 1–12, 2021.

[4] Y. Yue, C. Zhao, Z. Wu, C. Yang, Y. Wang, and D. Wang, "Collaborative semantic understanding and mapping framework for autonomous systems," *IEEE/ASME Trans. Mechatron. (TMECH)*, vol. 26, no. 2, pp. 978–989, 2020.

[5] Q. Zhang, M. Wang, Y. Yue, and T. Liu, "LCR-SMM: Large convergence region semantic map matching through expectation maximization," *IEEE/ASME Trans. Mechatron. (TMECH)*, pp. 1–12, 2021.

[6] Y. Deng, M. Wang, Y. Yang, and Y. Yue, "HD-CCSOM: Hierarchical and dense collaborative continuous semantic occupancy mapping through label diffusion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, to be published, 2022.

[7] H. Wang, C. Wang, and L. Xie, "Intensity Scan Context: Coding intensity and geometry relations for loop closure detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2020, pp. 2095–2101.

[8] Y. Yue, C. Zhao, M. Wen, Z. Wu, and D. Wang, "Collaborative semantic perception and relative localization based on map matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2020, pp. 6188–6193.

[9] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 5297–5307.

[10] X. Tang *et al.*, "Place recognition using line-junction-lines in urban environments," in *Proc. IEEE Int. Conf. Cyb. Intelli. Syst. (CIS) and IEEE Int. Conf. Robot., Autom. Mechatron. (RAM)*. IEEE, 2019, pp. 530–535.

[11] G. Peng, J. Zhang, H. Li, and D. Wang, "Attentional pyramid pooling of salient visual residuals for place recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, October 2021, pp. 885–894.

[12] G. Peng, Y. Huang, H. Li, Z. Wu, and D. Wang, "LSDNet: A Lightweight Self-Attentional Distillation Network for Visual Place Recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, to be published, 2022.

[13] L. He, X. Wang, and H. Zhang, "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2016, pp. 231–237.

[14] G. Kim and A. Kim, "Scan Context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2018, pp. 4802–4809.

[15] Y. Fan, Y. He, and U.-X. Tan, "Seed: A segmentation-based egocentric 3D point cloud descriptor for loop closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2020, pp. 5158–5163.

[16] Y. Wang *et al.*, "LiDAR Iris for loop-closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2020, pp. 5769–5775.

[17] X. Chen *et al.*, "OverlapNet: Loop closing for LiDAR-based SLAM," *arXiv preprint arXiv:2105.11344*, 2021.

[18] X. Kong *et al.*, "Semantic graph based place recognition for 3D point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2020, pp. 8216–8223.

[19] G. Peng, Y. Yue, J. Zhang, Z. Wu, X. Tang, and D. Wang, "Semantic reinforced attention learning for visual place recognition," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2021, pp. 13 415–13 422.

[20] L. Li *et al.*, "SSC: Semantic scan context for large-scale place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2021, pp. 2092–2099.

[21] Z. Wu, Y. Yue, M. Wen, J. Zhang, G. Peng, and D. Wang, "MSTSL: Multi-sensor based two-step localization in geometrically symmetric environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2021, pp. 5245–5251.

[22] K. P. Cop, P. V. Borges, and R. Dubé, "Delight: An efficient descriptor for global localisation using lidar intensities," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2018, pp. 3653–3660.

[23] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2020, pp. 5151–5157.

[24] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++: Efficient lidar-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2019, pp. 4530–4537.

[25] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2012, pp. 1643–1649.

[26] J. Behley *et al.*, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 9297–9307.

[27] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4470–4479.

[28] B. Ferrarini *et al.*, "Exploring performance bounds of visual place recognition using extended precision," *IEEE Robot. Autom. Lett. (RA-L)*, vol. 5, no. 2, pp. 1688–1695, 2020.