Pattern Recognition 120 (2021) 108171

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A two-level framework for place recognition with 3D LiDAR based on spatial relation graph[☆]

Yansong Gong[a], Fengchi Sun[a,*], Jing Yuan[b], Wenbin Zhu[b], Qinxuan Sun[b]

[a] College of Software Nankai University, No. 38 Tongyan Road, Jinnan District, Tianjin, 300350 P.R. China
[b] College of Artificial Intelligence Nankai University, No. 38 Tongyan Road, Jinnan District, Tianjin, 300350 P.R. China

ARTICLE INFO

Article history:
Received 26 December 2020
Revised 15 May 2021
Accepted 6 July 2021
Available online 15 July 2021

Keywords:
Place recognition
3D LiDAR
Spatial relation graph
Two-level framework

ABSTRACT

In the field of robotics, due to the complexity of real environments, place recognition using the 3D LiDAR is always a challenging problem. The spatial relations of internal structures underlying the LiDAR data from different places are distinguishable, which can be used to describe the environment. In this paper, we utilize the spatial relations of internal structures and propose a two-level framework for 3D LiDAR place recognition based on the spatial relation graph (SRG). At first, the proposed framework segments the point cloud into multiple clusters, then the features of the clusters and the spatial relation descriptors (SRDs) between the clusters are extracted, and the point cloud is represented by the SRG, which uses the clusters as the nodes and their spatial relations as the edges. After that, we propose a two-level matching model in which two different models are fused for accurately and efficiently matching the SRGs, including the upper-level searching model (U-LSM) and lower-level matching model (L-LMM). In the U-LSM, an incremental bag-of-words model is used to search for candidate SRGs through the distribution of the SRDs in the SRG. In the L-LMM, we utilize the improved spectral method to calculate similarities between the current SRG and the candidates. The experimental results demonstrate that our framework achieves good precision, recall and viewpoint robustness on both public benchmarks and self-built campus dataset.

1. Introduction

Place recognition is a fundamental and critical problem of pattern recognition in the field of robotics. An intelligent robot should have the ability to recognize the place where it is currently located, so as to complete the task of navigation. Meanwhile, place recognition can also be used for loop closure detection in simultaneous localization and mapping (SLAM) [1–3]. Long-term SLAM in large-scale environments cannot avoid error accumulation, which makes the mapping results inconsistent. In this situation, the place recognition method can be utilized to detect the loop closure which is a prerequisite for eliminating accumulated error.

In recent years, with the development of computer vision technology, image-based place recognition has achieved excellent results [4–7]. However, images are sensitive to illumination and the viewpoint of camera. Hence, image-based place recognition may fail in some challenging situations such as the dark environments. The 3D LiDAR can directly obtain the geometric information of the environments with high precision. Compared with the camera, the 3D LiDAR has a wider field of view and is hardly affected by the illumination changes. Therefore, considerable methods on place recognition have been proposed using the 3D LiDAR [8–13].

Inspired by the methods of image retrieval, the traditional methods of place recognition with 3D LiDAR detect the key points and extract the local descriptors from the LiDAR data to describe the environment [8,14]. In order to improve the efficiency of place recognition, some methods utilize the overall distribution of point clouds captured by the LiDAR to extract global descriptors and perform place recognition by measuring the similarities between the global descriptors [12,15]. In recent years, some segmentation-based methods have been proposed, which segment the LiDAR data into multiple clusters, extract the feature from each clutser, and perform place recognition by matching these segmented clusters [16,17].

However, compared with the feature of cluster, the spatial relations between the clusters have not been fully utilized on place recognition. In this paper, we propose a two-level framework which utilizes both the shape characteristics of clusters and the spatial relations between the clusters to perform place recognition.

[☆] This work was supported in part by the National Natural Science Foundation of China under Grant 61873327 and 62073178, the National Key Research and Development Plan under Grant 2018YFB0204304, and the Natural Science Foundation of Tianjin under Grant 20JCYBJC01470.

[*] Corresponding author.
E-mail addresses: 2120190505@mail.nankai.edu.cn (Y. Gong), fengchisun@nankai.edu.cn (F. Sun).

https://doi.org/10.1016/j.patcog.2021.108171
0031-3203/© 2021 Elsevier Ltd. All rights reserved.

The proposed framework contains two phases, i.e., *description* and *searching*. In the *description* phase, the spatial relation descriptor (SRD) is proposed to encode the relative spatial relations between a pair of clusters. Then, the environment is described by the spatial relation graph (SRG), which takes the clusters as the nodes, the features of clusters as the node attributes (describing the shape characteristics of clusters), the relative relations between the clusters as the edges, and the SRDs as the edge attributes. Meanwhile, a two-level matching model including the upper-level searching model (U-LSM) and the lower-level matching model (L-LMM), is proposed to perform a coarse-to-fine matching of the SRGs. In the U-LSM, an incremental BoW model without offline training is used to quickly search the candidate SRGs from historical data. In the L-LMM, the spectral method is improved for calculating the similarities between the SRGs. The contributions of the paper are summarized as follows.

- The proposed framework pays attention to the relative spatial relations between the segmented clusters in the LiDAR data. The SRD is proposed to encode the relative spatial relation between a pair of clusters with high distinguishability. Moreover, the SRD does not make any prior assumptions about the models of clusters, it can describe the general spatial relation between the clusters with irregular shape.
- In the proposed framework, the SRG is proposed to describe the environment, which contains different types of descriptive information (nodes and edges) and organizes them effectively into a unified representation. Moreover, a novel two-level graph matching model is proposed to match the SRGs, which can accurately and efficiently search for the similar SRGs from historical data and calculate the similarities between SRGs. It is worth pointing out that the whole process does not require an offline pre-training process.
- Comprehensive experiments are carried out on multiple datasets such as KITTI, Hannover2 and self-built campus dataset, demonstrating that the proposed SRD is distinguishable and our framework can achieve good results in precision, recall and viewpoint robustness.

The rest of paper is organized as below. The related works are presented in the Section 2. In Section 3, we give a detailed description for the proposed framework. Extensive experimental evaluations are shown in Section 4. Conclusions are presented in Section 5.

2. Related work

Generally, place recognition methods using the 3D LiDAR are divided into four types including scan-matching-based methods, local descriptor-based methods, global descriptor-based methods and segmentation-based methods. The scan-matching-based methods [18,19] align a pair of point clouds through iterative calculation. The most representative scan-matching-based methods are the iterative closest points (ICP) [18] and its variations, such as the point-to-line ICP (PLICP) [19]. However, the scan-matching-based methods may fail without the initial transformation between two scans, which limits its applications in the real scenes.

The local descriptor-based methods [8,20–26] perform the place recognition through matching the local descriptors extracted from the key points in the point cloud. In the point feature histogram (PFH) [20], the geometric information in the neighborhood of a key point was encoded to a histogram, then the place recognition was performed by matching the key points with similar PFHs. The fast point feature histogram (FPFH) [21] improved the efficiency of PFH by reordering the data and caching previously computed values, while retaining most of the descriptive ability of the PFH. The SHOT [22] generated the local descriptor by counting the normal vectors in the neighborhood of a key point. The ISHOT [23] added the laser intensity to the SHOT for enhancing the descriptive ability. In [24] and [25], the point clouds were converted into bearing angle images, and then SURFs [27] and ORBs [28] were extracted to describe the environments, respectively. Similarly, in [26], the point clouds were converted into the range images, and the local descriptors were extracted based on the Laplacian of Gaussian (LoG) method. The work of [8] extended the method in [26] and combined the Normal-Aligned Radial Features (NARFs) extracted from the range images and the bag-of-words (BoW) model to perform the place recognition. Generally, the recognition ability of local descriptor-based methods depends on the number of key points substantially, which makes it difficult to balance the recognition accuracy and efficiency.

The global descriptor-based methods [9–13,15] extract global descriptors from a whole point cloud and perform place recognition by measuring the similarities between the global descriptors. For example, the viewpoint feature histogram (VFH) [15] extended the FPFH for the entire point cloud and computed statistics between the viewpoint and the normal vectors estimated at each point. Since the viewpoint is encoded into the descriptor, the VFH is not suitable for the situation with changing viewpoint. The M2DP [12] used the distribution of the point cloud projected to multiple planes to extract global descriptors, which makes it efficient to be extracted. However, the viewpoint robustness of M2DP is also not outstanding because the distribution of the point cloud will change with the change of viewpoint. The scan context (SC) [13] calculated a global descriptor by dividing the point cloud into multiple bins from the top view and encoding the max height of the points in each bin into a matrix. The SC method can achieve favorable performance on place recognition under the planar motion of the robot. However, if the z-axis of the sensor frame is not invariant w.r.t. the global coordinate system, the SC method cannot obtain good results. In recent years, due to the rapid development of deep learning, some researches have utilized deep learning methods to generate global descriptors for the 3D LiDAR place recognition [29]. PointNetVLAD [9] used deep learning to perform large-scale 3D LiDAR place recognition for the first time. SeqLPD [10] extended the PointNetVLAD by adopting a coarse-to-fine sequence matching strategy. In [11], the point clouds were converted into range images and a convolutional neural network (CNN) was used to extract the global descriptors. OverlapNet [30] utilized a deep neural network to provide overlap area and relative yaw angle estimates between two LiDAR scans, and further performed loop closure detection in a SLAM system. In DiSCO [31], a CNN-based network architecture was proposed to extract global descriptors with rotation invariance. Meanwhile, a differentiable phase correlation estimator is proposed for relative orientation estimation between two scans. The aforementioned deep learning-based methods require sufficient data and time for pre-training, which limits its application in unknown scenes.

The segmentation-based methods [16,17,32–36] segment the point cloud into several local areas, such as planar patches, line segments or irregular clusters, and then perform the place recognition according to the descriptors of local areas or the relations between the local areas. In [32], the planes were segmented from the point cloud, and the geometric relations between neighbouring planar patches are extracted to describe the environments. Similarly, both the plane surfaces and line segments were segmented to describe the environment in [33], and then a robust probabilistic method for selecting the best pose hypothesis was used to match overlapping point clouds. However, the methods in both [32] and [33] require structured or semi-structured scenes to segment the plane patches or line segments stably. In [34], the objects were segmented from the point cloud, and the place recognition was performed by comparing the objects from the new places

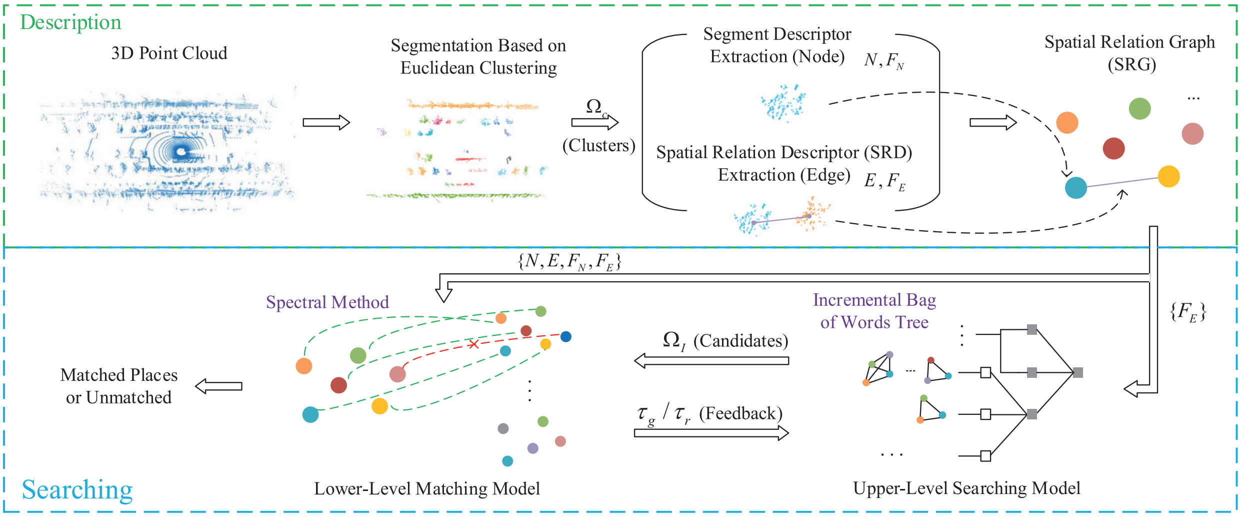

Description
3D Point Cloud
Segmentation Based on Euclidean Clustering
Ω_c
(Clusters)
Segment Descriptor Extraction (Node)
N, F_N
Spatial Relation Descriptor (SRD) Extraction (Edge)
E, F_E
Spatial Relation Graph (SRG)
$\{N, E, F_N, F_E\}$
$\{F_E\}$
Spectral Method
Incremental Bag of Words Tree
Ω_I (Candidates)
τ_g / τ_r (Feedback)
Matched Places or Unmatched
Searching
Lower-Level Matching Model
Upper-Level Searching Model

Fig. 1. System overview.

against the existing objects. This method can only work well with a small number of objects in small scale environments. The Segmatch [16] [17] segmented the point cloud into multiple irregular clusters to describe the environment and performed place recognition according to the similarities of these clusters. Nevertheless, in the Segmatch, a classifier with offline training is required to determine whether the clusters represent the same object. Moreover, the real environments may contain clusters that are extremely similar in shape, which makes it difficult to build correspondences only through the similarities between the clusters. In [35], a semantic graph (SG) representation for 3D point cloud scenes was presented, which captured semantic information and topological relations between objects. And a graph similarity network was proposed for the matching of semantic graphs. In Locus [36], after segmentation, the segment features extracted by a 3D CNN, and then the topological and temporal information related to the segments were aggregated by second-order pooling to obtain a global descriptor of the point cloud. The aforementioned two methods take the relations between segments into account, and achieve good performance in KITTI dataset. However, the SG relies heavily on the results of semantic segmentation, which severely limits the application of the algorithm in diverse environments. Similarly, in Locus, the 3D CNN also needs to be well trained in advance to get the effective feature for each segment.

3. Two-level place recognition framework

In this section, we present the details of the proposed two-level framework for place recognition using 3D LiDAR. First, the overview of our framework is introduced in Section 3.1, which illustrates the overall process of the proposed framework. Then, we present the extraction of SRD and the construction of SRG in Section 3.2. Finally, the two-level matching model is presented in Section 3.3, which fuses two different models to match the SRG accurately and efficiently.

3.1. Overview of framework

Our framework is mainly divided into two phases, including the *description* and *searching*. In the *description* phase, the point clouds are represented by the SRGs to describe the environments. In the *searching* phase, the SRGs are used to search for the similar SRGs and determine whether the corresponding data are collected from the same place. The system overview is shown in Fig. 1 where the notations are listed below.

N	The set of nodes in SRG
F_N	The set of node attributes in SRG
E	The set of edges in SRG
F_E	The set of edge attributes in SRG
Ω_c	The set of segmented clusters
τ_g / τ_r	Growth / Repression factor
Ω_I	The set of the candidate SRGs

Specifically, for the *description* phase, the Euclidean clustering method is utilized to segment a point cloud into a set of clusters Ω_c, and for each cluster, the shape feature is extracted to describe the shape of cluster. Then, the clusters are taken as the nodes of the SRG, and the shape features of clusters are taken as node attributes of the SRG, yielding the set of nodes N and the set of node attributes F_N, respectively. Meanwhile, the edges of the SRG are defined by the relative spatial relations between clusters and the corresponding edge attributes are the extracted SRDs, yielding the set of edges E and the set of edge attributes F_E, respectively. Finally, the SRG is constructed by N, E, F_N and F_E together.

For the *searching* phase, the constructed SRG is fed into the U-LSM to search for the similar SRGs from the historical data, which is based on an established incremental BoW tree and outputs the set of candidates Ω_I. Then, the candidates Ω_I are fed into the L-LMM which utilizes the spectral method to calculate the similarities between the current SRG and the candidates, respectively. Finally, according to the similarities, our framework determines whether the scans are collected from the same places. In addition, a feedback mechanism is applied to improve the searching ability of the U-LSM. Specifically, after receiving the candidates provided by the U-LSM, the growth factor τ_g or repression factor τ_r are calculated in the L-LMM according to the situations of SRG matching. Then, τ_g and τ_r are fed back from the L-LMM to the U-LSM, and the parameters of U-LSM are adjusted adaptively according to τ_g and τ_r, which make the candidates given by the U-LSM more accurate in the future search.

3.2. Construction of SRG

3.2.1. Point cloud segmentation

Before the segmentation of a point cloud, ground removal is a necessary step to make the clusters separated from each other. We adopt the fast segmentation algorithm [37] to efficiently remove the ground in the point cloud. The plane parameters of the ground are fitted heuristically, and then the ground is removed according to the plane parameters. After ground removal, the Euclidean clus-

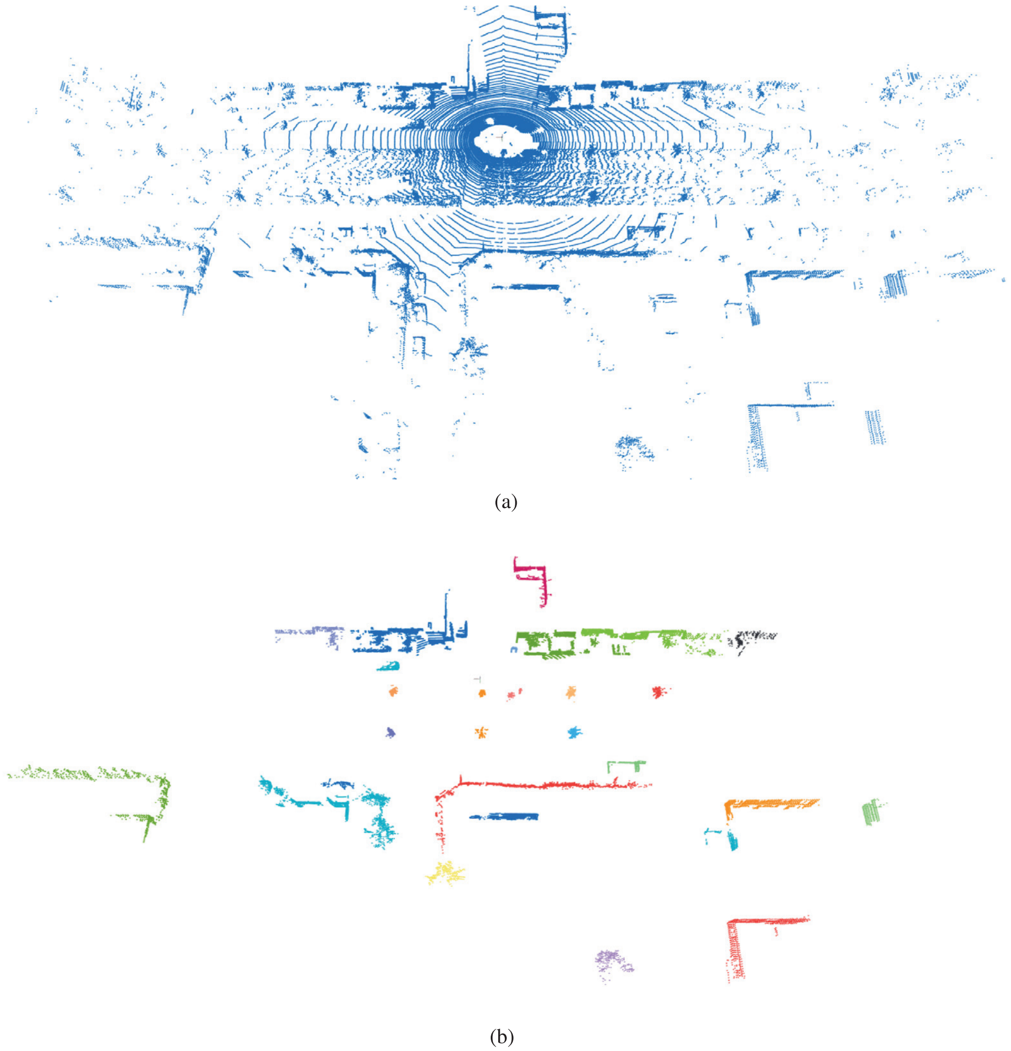


Fig. 2. The visualization of the segmentation results. Fig. 2(a) shows the point cloud before the segmentation, and Fig. 2(b) shows the point cloud after the segmentation, in which the different clusters are shown in different colors.

tering method provided by PCL[1] is used to segment the point cloud into multiple clusters. In order to improve efficiency, 3D grids are established in the space of point cloud, and then the clusters are obtained by clustering the grids and their internal points based on the Euclidean distances. Fig. 2 shows the visualization of segmenting a point cloud captured by the 3D LiDAR. As can be seen, the Euclidean clustering method provided an effective result of segmentation. Specifically, after the segmentation, the ground points and scattered points far away from the laser transmitter are removed, while remaining the clusters which can retain the characteristics of the environment.

3.2.2. *Spatial relation descriptor*

In real environments, especially in outdoor scenes, the relative spatial relations between the segmented clusters are distinguishable. Moreover, compared with the camera, the 3D LiDAR can obtain geometric information in the environment with a wide field of view and high precision, which is convenient for extracting the relative spatial relations. In this section, the SRD is proposed to encode the relative spatial relations between clusters. The computation of the SRD includes two steps, i.e., the relational unit extraction and relational unit encoding.

A relational unit can be defined for any pair of points from two different clusters. A relational unit contains three components l_1, l_2 and θ, which represent the sum of distances from the points to the boundaries of the corresponding clusters, the distance between the boundaries of the clusters and the angle between the line connecting the two points and the line connecting two centroids of the clusters, respectively. The relational unit extraction is illustrated in Fig. 3. For a pair of clusters C_j and C_k, their centroids are denoted by p_j^c and p_k^c, respectively. The direction vector from p_j^c to p_k^c is denoted by $\boldsymbol{v}$. Then, for a pair of points p_j^r and p_k^r, which are any pair of points from C_j and C_k, respectively, the direction vector from p_j^r to p_k^r is denoted by $\boldsymbol{\mu}$. Next, we calculate the intersection points p_j^b and p_k^b between the line segment $p_j^r p_k^r$ and the outer boundaries of C_j and C_k, respectively. We define the distance between p_j^r and p_j^b as d_1, and the distance between p_j^b and p_k^b as d_2, the distance between p_k^r and p_k^b as d_3, and the angle between $\boldsymbol{\mu}$ and $\boldsymbol{v}$ as ω. The relational unit $r_u = R_u\left(p_l^r, \quad p_k^r\right) = \{l_1, l_2, \theta\}$ can be calculated by

$$\begin{cases} l_1 = d_1 + d_3 \\ l_2 = d_2 \\ \theta = \begin{cases} \omega \quad if \quad \omega \le 90^\circ \\ 180^\circ - \omega \quad otherwise \end{cases} \end{cases} \tag{1}$$

[1] http://pointclouds.org

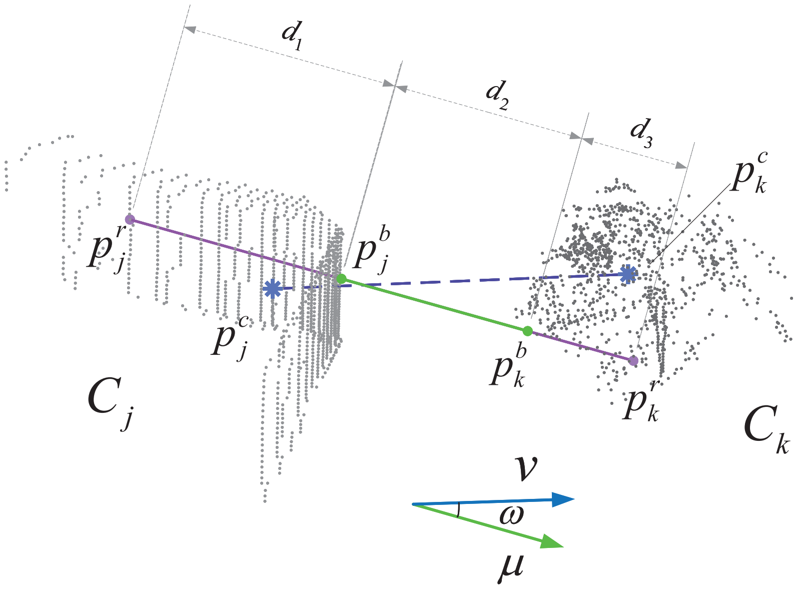


Fig. 3. Diagram of the relational unit extraction.

In the implementation, a grid-based method is designed to calculate d_1 and d_3 efficiently. First, the maximum and minimum coordinates of the clusters in the $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ directions are calculated to measure the scope of the cluster. Then, the space is equally divided into 64 × 64 × 64 grids according to the scope of the cluster. A grid cell is called occupied cell if it contains LiDAR points, and empty cell otherwise. Therefore, it is easy to find the farthest occupied cell from p_j^r along the direction of $\boldsymbol{\mu}$ in C_j, and then we calculate the distance between p_j^r and the center of this occupied cell as d_1. Similarly, d_3 is calculated by the same way. Next, d_2 is obtained by subtracting d_1, d_3 from the distance between the p_j^r and p_k^r.

It is worth pointing out that, in the relational unit, l_1 and l_2 measure the relative shape and distance between two clusters, and θ measures the relative orientation between two clusters. Moreover, in the extraction of relational unit, the boundary information of the clusters is fused into the computation of l_1 and l_2, which makes the relational unit sensitive to the shape of the cluster and increases the distinguishability of SRD encoded by the relational unit. In addition, three principles are followed to extract the SRD and the relational units, i.e., the symmetry, viewpoint robustness and shape sensitivity. The symmetry means that the SRD extracted form C_j to C_k should be completely consistent with the SRD from C_k to C_j. Hence, l_1 is defined by $d_1 + d_3$ in the relational unit to guarantee the symmetry. The viewpoint robustness means that the negative impact of viewpoint changes should be reduced in the extraction process of the SRD. Therefore, the relational unit does not contain components directly related to the viewpoint of robot. The shape sensitivity refers that SRDs extracted from clusters with different shapes should be distinguishable, for this reason, the boundary points p_j^b and p_k^b are selected to calculate the relational units, so as to fit the outer contours between the two clusters.

For the relational unit encoding, given a pair of clusters C_j and C_k, N_r pairs of points are randomly selected from a pair of clusters, denoted by $\{\{p_j^i, p_k^i\}, i = 1, \dots, N_r | p_j^i \in C_j, p_k^i \in C_k\}$. And the relational unit corresponding to each pair of points is extracted, generating the corresponding set of relational units, denoted by $\{r_u^i = R_u(p_j^i, p_k^i) = \{l_1^i, l_2^i, \theta^i\}, i = 1, \dots, N_r\}$. Then, the encoding method in [38] is applied to encode the relational units into a descriptor called SRD, which utilizes the statistical information of the N_r relational units to encode the relative relations from different cluster pairs into a descriptor. The method in [38] is sensitive to the value in the relational units and does not require the prior knowledge of the distribution of the relational units.

Specifically, we first split the space of relational unit into discrete cells along each dimension and l_1, l_2 and θ are divided into the cells according to their values. Then, for each element in the relational unit, two weights are calculated according to the distances of value to the upper and lower boundary of the cells. Therefore, for three elements, there are a total of eight (2^3) combinations of weights. Then, the result of multiplying the weights of the corresponding combination is added to an element of the SRD and the index of the element can be obtained by the indices of the cells along each dimension. The encoding method of the relational units is different from the work [38] by the variables and the value range of variables. The specific encoding process is shown in Algorithm 1.

3.2.3. Spatial relation graph

The SRG is a complete graph, represented by $G_s = \{N, E, F_N, F_E\}$, where N is the set of nodes, E is the set of edges, F_N is the set of node attributes and F_E is the set of edge attributes. The node attribute is defined by $f_n = \{f^l, f^p, f^s, f^a\}$, where f^l, f^p, f^s and f^a quantify the linearity, planarity, scattering and anisotropy of the cluster, respectively, which are calculated by

$$\begin{cases} f^l = (\lambda_1 - \lambda_2)/\lambda_1 \\ f^p = (\lambda_2 - \lambda_3)/\lambda_1 \\ f^s = \lambda_3/\lambda_1 \\ f^a = (\lambda_1 - \lambda_3)/\lambda_1 \end{cases} \tag{2}$$

where λ_1, λ_2 and λ_3 are three eigenvalues of the covariance matrix M_c of the points in the cluster, which satisfy $\lambda_1, \lambda_2, \lambda_3 \in R$ and $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$. The feature f_n describes the shape characteristics of the clusters to distinguish from each other and the extraction of f_n is efficient, because the eigen decomposition of 3 × 3 matrix can be performed very quickly.

The edge attribute of the SRG is denoted by f_e, and the SRDs are adopted as the edge attributes f_e. Although f_n and f_e are two types of feature which have different meanings and dimensions, the SRG

Algorithm 1 Relational unit encoding.

```
Input: The set of relational units $\Omega_r = \{r_u^i | i = 1, 2, \dots, n\}$, $r_u^i = \{l_1^i, l_2^i, \theta^i\}$
Output: SRD $D_s$
Set the value range of the relational units $\left[l_1^{\min}, l_1^{\max}\right]$, $\left[l_2^{\min}, l_2^{\max}\right]$ and $\left[\theta^{\min}, \theta^{\max}\right]$ to [0, 10], [0, 40] and [0, 10π/180], respectively.
Set the number of interval divisions $\eta_{l_1}$, $\eta_{l_2}$ and $\eta_\theta$ to 4, 4 and 4, respectively.
Initialize the SRD $D_s \in \mathbb{R}^{\left(\eta_{l_1}+1\right)\cdot\left(\eta_{l_2}+1\right)\cdot(\eta_\theta+1)}$
for $i = 1, 2, \dots, n$ do
    $k_{l_1} = \left\lfloor \left(l_1^i - l_1^{\min}\right)\eta_{l_1} / \left(l_1^{\max} - l_1^{\min}\right) \right\rfloor$
    $k_{l_2} = \left\lfloor \left(l_2^i - l_2^{\min}\right)\eta_{l_2} / \left(l_2^{\max} - l_2^{\min}\right) \right\rfloor$
    $k_\theta = \left\lfloor \left(\theta^i - \theta^{\min}\right)\eta_\theta / \left(\theta^{\max} - \theta^{\min}\right) \right\rfloor$
    $\omega_{l_1} = \left(l_1^i - l_1^{\min}\right)\eta_{l_1} / \left(l_1^{\max} - l_1^{\min}\right) - k_{l_1}$, $\omega'_{l_1} = 1 - \omega_{l_1}$
    $\omega_{l_2} = \left(l_2^i - l_2^{\min}\right)\eta_{l_2} / \left(l_2^{\max} - l_2^{\min}\right) - k_{l_2}$, $\omega'_{l_2} = 1 - \omega_{l_2}$
    $\omega_\theta = \left(\theta^i - \theta^{\min}\right)\eta_\theta / \left(\theta^{\max} - \theta^{\min}\right) - k_\theta$, $\omega'_\theta = 1 - \omega_\theta$
    $D_s\left[k_{l_1} \cdot \left(\eta_{l_2} + 1\right)(\eta_\theta + 1) + k_{l_2} \cdot (\eta_\theta + 1) + k_\theta\right] += \omega'_{l_1}\omega'_{l_2}\omega'_\theta$
    $D_s\left[k_{l_1} \cdot \left(\eta_{l_2} + 1\right)(\eta_\theta + 1) + k_{l_2} \cdot (\eta_\theta + 1) + (k_\theta + 1)\right] += \omega'_{l_1}\omega'_{l_2}\omega_\theta$
    $D_s\left[k_{l_1} \cdot \left(\eta_{l_2} + 1\right)(\eta_\theta + 1) + \left(k_{l_2} + 1\right) \cdot (\eta_\theta + 1) + k_\theta\right] += \omega'_{l_1}\omega_{l_2}\omega'_\theta$
    $D_s\left[k_{l_1} \cdot \left(\eta_{l_2} + 1\right)(\eta_\theta + 1) + \left(k_{l_2} + 1\right) \cdot (\eta_\theta + 1) + (k_\theta + 1)\right] += \omega'_{l_1}\omega_{l_2}\omega_\theta$
    $D_s\left[\left(k_{l_1} + 1\right) \cdot \left(\eta_{l_2} + 1\right)(\eta_\theta + 1) + k_{l_2} \cdot (\eta_\theta + 1) + k_\theta\right] += \omega_{l_1}\omega'_{l_2}\omega'_\theta$
    $D_s\left[\left(k_{l_1} + 1\right) \cdot \left(\eta_{l_2} + 1\right)(\eta_\theta + 1) + k_{l_2} \cdot (\eta_\theta + 1) + (k_\theta + 1)\right] += \omega_{l_1}\omega'_{l_2}\omega_\theta$
    $D_s\left[\left(k_{l_1} + 1\right) \cdot \left(\eta_{l_2} + 1\right)(\eta_\theta + 1) + \left(k_{l_2} + 1\right) \cdot (\eta_\theta + 1) + k_\theta\right] += \omega_{l_2}\omega_{l_2}\omega'_\theta$
    $D_s\left[\left(k_{l_1} + 1\right) \cdot \left(\eta_{l_2} + 1\right)(\eta_\theta + 1) + \left(k_{l_2} + 1\right) \cdot (\eta_\theta + 1) + (k_\theta + 1)\right] += \omega_{l_1}\omega_{l_2}\omega_\theta$
end for
```

can organize them effectively, yielding a unified representation to describe the environments. Moreover, the SRG is a complete graph and contains all the relative spatial relations between the clusters in one scan. Hence, extensive information is retained in an SRG to describe the environment.

3.3. Two-level matching framework

To perform the place recognition, the SRGs constructed for different places need to be matched. The most widely used graph matching methods are the learning-based methods [39] and the tree search methods [40]. The learning-based methods match the graphs through a pre-training model, which requires the prior knowledge of the graphs, while the tree search method is time-consuming for complete graphs (e.g.,SRGs). To this end, we propose a two-level matching model for the SRGs, including the U-LSM and the L-LMM. In the U-LSM, an incremental BoW model is established using the SRDs in the SRGs from the historical data, and the SRGs similar to the current SRG are searched out via a fast voting strategy as the candidates. Then, in the L-LMM, an improved spectral method is used to perform an exact similarity calculation between the current SRG and the candidates obtained from the U-LSM. In addition, the U-LSM can dynamically adjust its parameters through the feedback from the L-LMM. Hence, our framework continuously optimizes its performance while the robot performs the place recognition, yielding both efficiency and high quality.

3.3.1. Upper-level searching model

As mentioned in Section 3.2.2, the SRD is used to describe the relative relation between a pair of clusters and has powerful distinguishability. Moreover, the SRG is a complete graph, containing all relative relations between the clusters. Hence, the SRDs in an SRG contains rich information which can be used to search for the similar SRGs. In this paper, an incremental BoW model without offline training is used to fit the distribution of SRDs, in which the incremental BoW tree is initialized and maintained by clustering the SRDs. Then, the candidates can be searched out by a fast voting strategy. In the U-LSM, the incremental BoW model includes three steps, i.e., initialization, updating, and voting.

For the initialization, a BoW tree is initialized by clustering the edge attributes (SRDs) of the first SRG into K_{BoW} SRD clusters by K-Means, that is, the root node $N_{\{root}$ of the BoW tree has K_{BoW} children, which are the leaf nodes defined by the SRD clusters, and the attributes of leaf nodes are defined by the centers of SRD clusters. According to the centers of the SRD clusters, the SRDs can retrieve the leaf nodes to which they belong using the BoW tree from-top-to-bottom. Each leaf node N_{leaf}^i has corresponding inverted indexes I_{inv}^i, to record the SRDs belonging to N_{leaf}^i and the SRGs to which the SRDs belongs.

The updating process is to realize the incremental update of the BoW tree. As new SRGs are generated, the SRDs in them are added to the inverted indexes of leaf nodes in the BoW tree. As the data increases, the initial structure of BoW tree cannot fit the new distribution of the SRDs. Instead of re-clustering all the SRDs in the tree, when the size of the inverted indexes I_{inv}^i corresponding to N_{leaf}^i is greater than a threshold Th_s, the SRDs indexed by I_{inv}^i are re-clustered by K-Means, and K_{BoW} new leaf nodes are generated as the children of N_{leaf}^i, which is called the *split* of N_{leaf}^i. Meanwhile, the SRDs in I_{inv}^i are reallocated to the new inverted indexes in the new leaf nodes.

In the voting process, a fast voting strategy is designed to efficiently search for the candidates for an SRG. First, the initial votes of historical SRGs are set to 0. For the current SRG G_s^c, the set of edge attributes F_E^c is sent into the BoW tree. And for each descriptor f_e^c in F_E^c, the leaf node to which f_e^c belongs is retrieved from-top-to-bottom. After that, for every SRG G_s^i indexed by the corresponding inverted indexes, its votes plus t_i, which is calculated by

$$t_i = \frac{1}{(|n_c - n_i|^2 + 1) \cdot \left|F_E^c\right|} \tag{3}$$

where n_c and n_i are the numbers of edge attributes belonging to the inverted indexes in G_s^c and G_s^i, respectively. In the incremental BoW model, not only the types of words are considered in the voting process, but the number of each type is also used to determine the number of votes. For a certain word, the more the numbers of the words in the searching and query SRGs, respectively, vary from each other, the less the number of votes increases.

In addition, The incremental BoW tree is constructed from top to bottom. Thus, the SRG ordering may affect the split order of the leaf nodes during the incremental construction process. However, in the U-LSM of our framework, the impact of SRD or SRG order on the BoW tree is not significant. First, the retrieval performance of the BoW tree mainly relies on the clustering results of the descriptors. Regardless of the specific structure of the BoW tree, the sufficiently similar SRDs always tend to be clustered into the same class. As a result, the performance of SRG retrieval is hardly be affected by the SRG ordering in the construction of incremental BoW trees. Second, for the retrieval time of the BoW tree, as reported in the paper [41], once sufficient data are included in the incremental BoW structure, the tree is supposed to be balanced regardless of the possible different ordering of the SRGs. Therefore, the time consumption of SRG retrieval is hardly be affected by the SRG ordering in the construction of incremental BoW trees.

3.3.2. Lower-level matching model

In order to evaluate the similarities between SRGs, the improved spectral method is applied in the L-LMM. The spectral method [42] is a graph matching method based on matrix spectral decomposition. First, an affinity matrix M_a is established of which the rows and columns express the potential correspondences of graph nodes. The elements of M_a measure the weights of a pair of potential correspondences. Then, through the principal component analysis of M_a, the spectral method utilizes the eigenvector corresponding to the largest eigenvalue to determine the node correspondences of the two graphs. The spectral method has high matching precision and has been successfully applied in many fields. In this paper, the elements of M_a can be calculated by

$$M_a\left(i \cdot n_1 + j, i' \cdot n_2 + j'\right) = \begin{cases} 4.5 - \frac{\left\| f_n^i - f_n^{i'} \right\|^2}{2\sigma_n^2}, & if i = j, i' = j' and \left\| f_n^i - f_n^{i'} \right\| < 3\sigma_n, \\ 4.5 - \frac{\left\| f_e^{ij} - f_e^{i'j'} \right\|^2}{2\sigma_e^2}, & if i \neq j, i' \neq j' and \left\| f_e^{ij} - f_e^{i'j'} \right\| < 3\sigma_e, \\ 0, & otherwise \end{cases} \tag{4}$$

where n_1 and n_2 are the numbers of nodes in two SRGs, i, j, i', j' are the indices of nodes in two SRGs. If $i = j$ and $i' = j'$, the elements of M_a measure how well node i matches the node i'. Assignments that are unlikely to be correct will be filtered out. Similarly, if $i \neq j$ and $i' \neq j'$, the elements of M_a describes how well the relative pairwise relations of two edge (i', j') is preserved after putting them in correspondence with the edge (i, j). σ_n, σ_e are the parameters used to adjust the weights of the potential correspondences. The larger σ_n and σ_e, the more pairwise relations between wrong assignments will get a positive score.

To evaluate the similarity between SRGs, the spectral method is improved from two aspects. First, a filtering threshold $\delta_{sc} (\delta_{sc} > 0)$ is set to avoid overmatching when using the main eigenvector to calculate the correspondences. Second, although the spectral method can calculate the correspondences of nodes in the graph, it cannot evaluate the similarity between two graphs. Therefore, a similarity calculation function is designed to calculate the similarity between two SRGs quickly and accurately, which can be calculated by

$$s = x^T M_a x / (n_1 \cdot n_2) - (n_1 + n_2)\mathcal{R}^2 / 2n_c \tag{5}$$

where x is the correspondences calculated by the spectral method, n_c is the number of correspondences, $\mathcal{R}$ is penalty term which punishes the insufficient correspondences. The process of the improved spectral method is shown in Algorithm 2.

3.3.3. Connection between two levels

The U-LSM and the L-LMM are fused closely in the two-level matching model as illustrated in Fig. 4. The L-LMM receives the candidates provided by the U-LSM to calculate the similarities between the SRGs. Meanwhile, the L-LMM provides the feedback information to update the parameters of the U-LSM, so as to adjust the structure of BoW tree in the U-LSM.

First, the K_{SRG} candidates are provided by the U-LSM to the L-LMM. Theoretically, the results given by the L-LMM considering entire information of the SRG are more accurate than the results given by the U-LSM only considering the statistic information in the SRG. Therefore, Thus, a larger value of K_{SRG} makes the final results given by the whole framework more accurate. However, since the time consumption of the graph matching process in L-LMM is much greater than the graph search process in U-LSM. In order to ensure the efficiency of the framework, K_{SRG} cannot be too large. In the implementation, the K_{SRG} is set to 25, which is a moderate selection to balance the accuracy and the efficiency of the whole framwork.

Algorithm 2 Improved spectral method for evaluating similarities between SRGs.

```
Input: $G_s^c$, $G_s^i$ and their affinity matrix $M_a$, $G_s^c$, $G_s^i$ contain $n_1$ and $n_2$ nodes, respectively.
Output: $s$ (Similarity between $G_s^c$ and $G_s^i$)
Initialize the similarity $s = 0$ and the correspondences vector $x = zeros(n_1 \cdot n_2)$.
Set the penalty term $\mathcal{R} = 1.2$ and filtering threshold $\delta_{sc} = 0.05$.
$[A, V] = Eigen(M_a)$, $A$ and $V$ are the set of eigenvalues and eigenvectors respectively.
$v = V(:, 1)$
while true do
    $[val_m, pos_m] = max(v)$, where $val_m$ is the maximum component in $v$, and $pos_m$ is the index of the $val_m$.
    if $val_m < \delta_{sc}$ then
        break
    end if
    $v[pos_m] = 0,\ x[pos_m] = 1$
    $pos_f = pos_m / n_2,\ pos_s = pos_m \% n_2$
    for $i = 1, \ldots, n_1$ do
        $v[i \cdot n_2 + pos_s] = 0$
    end for
    for $i = 1, \ldots, n_2$ do
        $v[pos_f \cdot n_2 + j] = 0$
    end for
end while
The similarity $s$ between $G_s^c$ and $G_s^i$ is calculated by (5)
```

According to the matching conditions between the SRGs in the L-LMM, the L-LMM transmits the growth factor $\tau_g\left(0 < \tau_g \leq 1\right)$ or the repression factor $\tau_r(\tau_r \geq 1)$ to the leaf nodes of the BoW tree in the U-LSM. Specifically, if the L-LMM matches the SRDs (edge attributes in the SRG) belonging to different leaf nodes, it usually indicates that the two leaf nodes are excessively split during the increment of the BoW tree, thus it is necessary to avoid these two leaf nodes from splitting again. Hence, the split threshold of the leaf nodes is increased by $Th_s = Th_s \cdot \tau_r$, which make the BoW tree in the U-LSM grow slowly in the two leaf node. Similarly, if the SRDs belonging to the same leaf node fail to match in the L-LMM, it usually refers that there exist inconsistent SRDs in this leaf node, and the leaf node needs to be further split to fit the real distribution of SRDs. Hence, the split threshold of the leaf node is decreased by $Th_s = Th_s \cdot \tau_g$, which accelerates the growth of the BoW tree at the leaf node. After applying the feedback mechanism from the L-LMM to the U-LSM, the structure of BoW tree in the U-LSM is adaptively adjusted the tree grows. In the implementation, the we select $\tau_g = 0.98,\ \tau_r = 1.05$, which yields good performance of the framework.

4. Experiments

In this section, the experimental setup is firstly introduced. Second, distinguishability of the SRD is evaluated. Finally, our place recognition framework is comprehensively evaluated by comparison with five state-of-the-art place recognition algorithms, in terms of the precision-recall (P-R) and robustness.

4.1. Experimental setup

The KITTI [43], Hannover2 [44], and self-built campus dataset in Nankai University (NKU-MC) are employed to carry out the

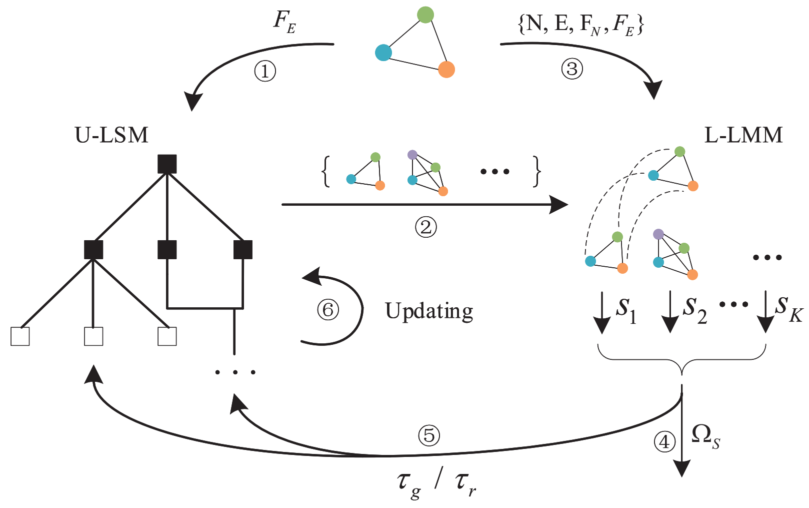


Fig. 4. The connection between the two levels. ① The SRG sends its edge attributes to the U-LSM. ② The U-LSM searches for the similar SRGs as the candidates and sends them to the L-LMM. ③ The L-LMM receives the SRG with its nodes, edges, node attributes and edge attributes. ④ The L-LMM calculates the similarities between the SRG and the candidates. ⑤ The L-LMM sends back the growth or repression factor to the U-LSM. ⑥ The U-LSM updates its internal parameters and adjusts its structure.

experiments. The KITTI dataset is collected in streets by a car equipped with a Velodyne HDL-64H LiDAR. In the experiments, the sequences 00, 05, 06, 07 and 08 are selected, which contain 4541, 2671, 1101, 1101 and 4071 frames of point cloud, respectively. Each point cloud in the KITTI contains about 120,000 points with max range of 100m. The Hannover2 dataset is collected in the campus environment of the Universitt Hannover by a rotating SICK LMS sensor. 922 frames of point cloud are obtained in the Hannover2, and each point cloud contains approximately 16,600 points with max range of 30m. The NKU-MC is collected by a P3D-X robot equipped with a Velodyne HDL-32 LiDAR. In the NKU-MC, the robot travels around the campus and collects 2454 frames of point cloud, each of which contains about 60,000 points with max range of 120 m.

In all the experiments, two frames of data are considered to be collected from the same place if their Euclidean distance is less than 6.0 m. In order to avoid performing place recognition with the adjacent frames, 50 adjacent frames before the current frame are excluded from the searching scope. Under this condition, there are 817, 513, 271, 86, 404, 289 and 360 positive samples in KITTI 00, 05, 06, 07, 08, Hannover2 and NKU-MC sequence, respectively. Moreover, the thresholds for segmenting point clouds are set to 1.2 m, 1.0 m and 1.0 m on the KITTI, Hannover2 and NKU-MC, respectively. All the experiments are carried out on a unified hardware platform, with Intel i7-8700 CPU with a clock speed of 2.6GHz, 8GB memory, and Ubuntu 18.04 operating system.

4.2. SRD quality evaluation

First of all, the stability of SRD should be demonstrated. To this end, an experiment is carried out to illustrate the influence of the number of random point pairs N_r to the stability of SRD. In the experiment, 5000 pairs of clusters are selected from the KITTI 00 sequences. For each pair of clusters, the SRD is extracted for 10,000 times for a specific value of N_r. Then, the centroid and covariance of the 10,000 SRDs are calculated. Next, the maximum distance from the SRD to the centroid and the maximum eigenvalue of the covariance are calculated. Finally, the maximum distances and maximum eigenvalues of the covariances are averaged over the 5000 pairs of clusters. The results are shown in Fig. 5, in which the x-axis represents the value of N_r, and the y-axis represents the average maximum distance and maximum eigenvalue for Fig. 5(a) and (b), respectively. As can be seen, when the N_r is larger than 1000, the distribution of the computed SRDs is relatively invariant w.r.t. the value of N_r. Therefore, in the following experiment, in order to balance the stability and computational efficiency of SRD, N_r is set to 1500.

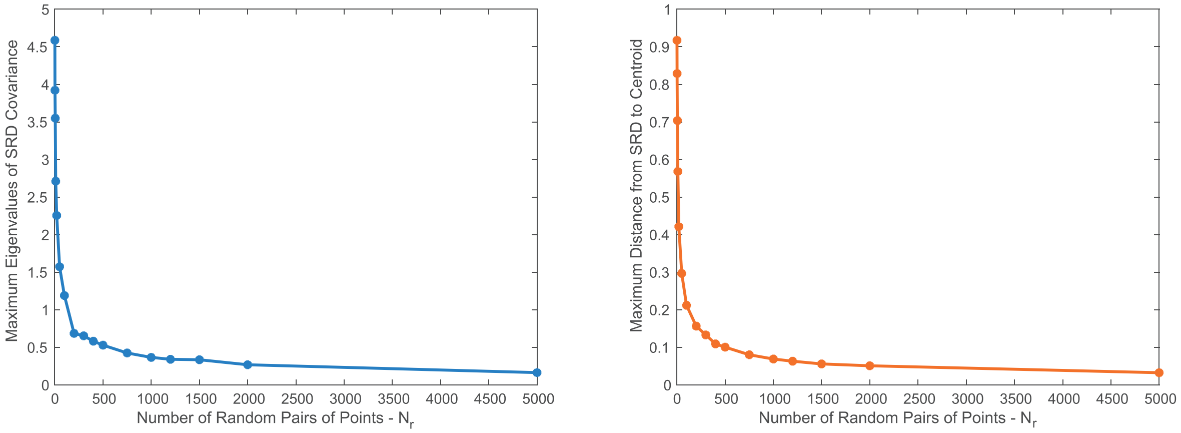
To the best of our knowledge, the proposed SRD is the first descriptor to describe the general relative relations between a pair of irregular clusters. Therefore, the SRD is compared with the features which are also extracted from a pair of clusters. Given a pair of clusters $C_j = \{p^i_j, i = 1, 2, \ldots, n_j\}$ and $C_k = \{p^i_k, i = 1, 2, \ldots, n_k\}$ where n_j and n_k are the numbers of points in C_j and C_k. Given a function $F(\cdot)$ of which the input is a set of points, and the output is a feature to describe the characteristic of the point set. Define the features f^{jk}_+ and f^{jk}_- as

$$f^{jk}_+ = F(C_j \cup C_k) \tag{6}$$

$$f^{jk}_- = D(F(C_k), F(C_k)) \tag{7}$$

In (7), $D\left(f_j, f_k\right) = \{\left|f^i_j - f^i_k\right|, i = 1, \ldots, dim\left(f_j\right)\}$ where the features f_j and f_k corresponding to the C_j and C_k are generated by $F\left(C_j\right)$ and $F(C_k)$, $dim(\cdot)$ is a function to calculate the dimension of a feature, and f^i_j and f^i_k represent the ith components in the f_j and f_k, respectively. The features f^{jk}_+ (equal to f^{kj}_+) describe the holistic characteristics of the pair of clusters C_j and C_k, f^{jk}_- (equal to f^{kj}_-) represent the differences between C_j and C_k in feature space. In this paper, the ESF [45], VFH and globally aligned spatial distribution (GASD) [46], which are usually used for object recognition, are selected to perform the function $F(\cdot)$ and extract the features f^{jk}_+ and f^{lj}_- from C_j and C_k. In the experiment, f^{jk}_+ generated by the ESF is nominated as ESF$_+$, and f^{jk}_- generated by the ESF is nominated as ESF$_-$. Similarly, the VFH$_+$, VFH$_-$, GASD$_+$ and GASD$_-$ are defined in the same way. Therefore, the SRD is compared with the ESF$_+$, ESF$_-$, VFH$_+$,VFH$_-$, GASD$_+$ and GASD$_-$, which are the features to describe a pair of clusters just like the SRD.

In order to show the distinguishability of SRD comprehensively, we extract the SRD, ESF$_+$, ESF$_-$, VFH$_+$,VFH$_-$, GASD$_+$ and GASD$_-$ from multiple pairs of clusters in four sequences from three dataset, i.e. KITTI 00, KITTI 08, Hannover2 and NKU-MC, respectively. From each sequence, 5000 positive samples (the same



(a) (b)

Fig. 5. The influence of the number of random point pairs to the stability of SRD.

Table 1
AUC of the features extracted from a pair of clusters.

	ESF_+	ESF_-	VFH_+	VFH_-	GASD_+	GASD_-	SRD
KITTI 00	0.9597	0.7767	0.6468	0.6134	0.8842	0.7351	**0.9912**
KITTI 08	0.9282	0.8502	0.8397	0.7175	0.9395	0.7873	**0.9989**
Hannover2	0.9886	0.8822	0.8614	0.8342	0.9037	0.8971	**0.9969**
NKU-MC	0.9338	0.8586	0.7260	0.7212	0.9129	0.8360	**0.9974**

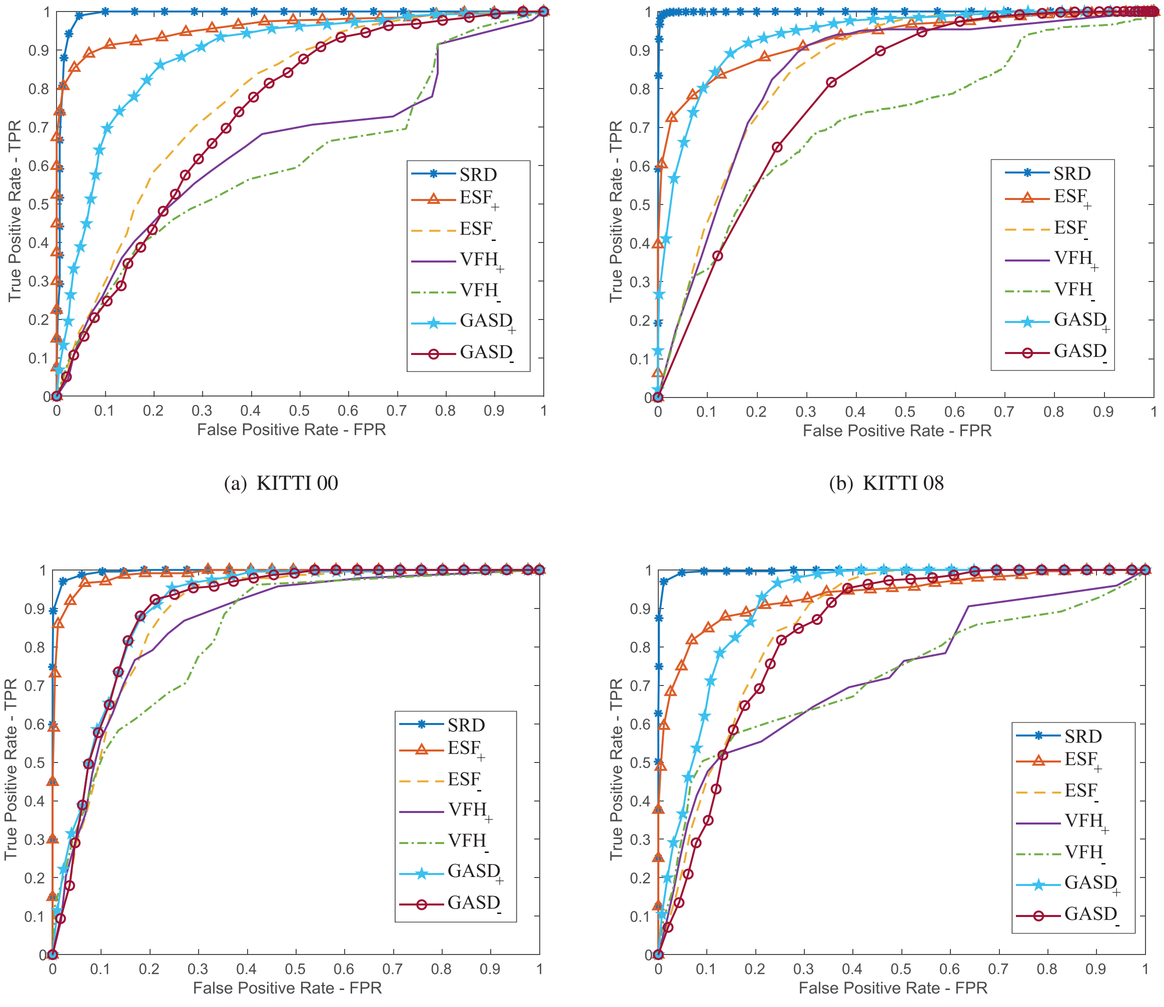
pair of clusters from the real world) and 5000 negative samples (a different pair of clusters) are randomly selected. The Euclidean distance is used to measure the similarity between the features. When the Euclidean distance between the features is less than a threshold th_e, the features are considered to be extracted from the same pair of clusters, which are then compared with the ground-truth. As th_e changes, a set of the false positive rates (FPRs) and the true positive rates (TPRs) is computed, and the curve of receiver operating characteristic (ROC) is drawn to evaluate the distinguishability of the features. The area under curve (AUC) is the area enclosed by the ROC curve and the coordinate axis, which also reflects the distinguishability of the features quantitatively. The VFH, ESF and GASD are implemented by point cloud library (PCL) to generate ESF_+, ESF_-, VFH_+,VFH_-, GASD_+ and GASD_-. In the VFH, the radius for calculating the normal vectors is set to 0.05m. The default parameters provided by PCL are used for ESF and GASD. The ROC curves are shown in Fig. 6, and the AUC is shown in Table 1.

Compared with the other features, the proposed SRD focuses on the spatial relations between the clusters, which are usually distinguishable in real environments. Moreover, since the boundaries of clusters are considered in the process of extracting SRD, two pairs of clusters with similar relative distances can also be distinguished well by the SRD. As can be seen from the ROC curve and the AUC for the four sequences, the proposed SRD achieves better results than the other features. in each sequence, the AUC of SRD is greater than 0.99, demonstrating that the SRD has high distinguishability, and also proving that the SRG constructed by the SRDs contains sufficient information to distinguish the different places.

4.3. Precision-recall evaluation

To evaluate the P-R performance, our framework is compared with five state-of-the-art place recognition algorithms, including the SC [13], M2DP [12], SHOT [22], ESF [45] and VFH [15]. The SC encodes a whole point cloud into a matrix as the global descriptor. The M2DP is a global descriptor which describes the environment according to the distribution of point cloud. The SHOT is a histogram-based local descriptor which encodes the statistics of normal vectors into a descriptor. The ESF and VFH are the global descriptors which are also based on histograms. For the implementation of the SC and M2DP, we use the open-source MATLAB code released by the authors. As for the SHOT, ESF and VFH, a C++ version implemented in the PCL is used. The M2DP and ESF use the default parameters given by their authors. And for the SC, a sub-descriptor called key ring is required for K-nearest-neighbor search, and thus the number of neighbors K_{SC} is an important parameter of the SC. According to [13], we choose $K_{SC} = 10$ and $K_{SC} = 50$ in the SC, respectively, which are denoted by SC-10 and SC-50. The radius for calculating the normal vectors in the SHOT and VFH is set to 0.05 m. For the parameters of our framework, the candidate frame K_{SRG} is selected as 25, σ_n and σ_e in the L-LMM are set to 0.03 and 0.08, respectively.

The P-R curves for the six methods are shown in Fig. 7. It can be seen that our framework outperforms the other five algorithms in all the sequences. Specifically, the SHOT, ESF and VFH have poor performance in all sequences, because they are based on histograms and can hardly distinguish the places with similar structures. The SC and M2DP achieve outstanding results in the most sequences, but they perform poorly in the Hannover2 and KITTI08, respectively. For the Hannover2, due to the limited range of SICK LMS, which makes the point clouds in the Hannover2 contain less environmental information than the other datasets, the results of the comparison algorithms are all not outstanding in this dataset. In contrast, the proposed framework achieves good results in the Hannover2, because the high distinguishability of the SRD can remedy the defect caused by insufficient information. Moreover, for KITTI 08, the robot revisits the same place from different directions when collecting the data. As a result, the M2DP fails to perform the place recognition in the KITTI 08. The reason is that the M2DP depends on the distribution of point cloud to perform place recognition, it cannot keep stable when the viewpoint of the robot changes. In contrast, the proposed framework also achieves



(a) KITTI 00

(b) KITTI 08

(c) Hannover2

(d) NKU-MC

Fig. 6. The ROC curves of the SRD and the comparison features in (a) KITTI 00, (b) KITTI 08, (c) Hannover2 and (d) NKU-MC, respectively.

Table 2
Recall at 100% precision in each evaluation sequence.

	SC-50	SC-10	M2DP	SHOT	VFH	ESF	Ours
KITTI 00	0.8444	0.8580	0.8641	0.8335	0.1261	0.0147	**0.9119**
KITTI 05	**0.8677**	0.8424	0.6304	0.7254	0.2943	0.0468	0.7992
KITTI 06	0.9333	0.9398	0.9188	0.7601	0.3875	0.1070	**0.9815**
KITTI 07	0.3918	0.3299	0.2404	0.5342	0.1146	0.0312	**0.7396**
KITTI 08	0.3276	0.3267	0.2005	0	0	0.0074	**0.3614**
Hannover2	0	0	0.0801	0.0941	0.0329	0.1202	**0.5017**
NKU-MC	0.2927	0.3122	0.1293	0.1972	0.0861	0.0167	**0.5333**

good results in the KITTI 08, because our framework is based on the graph matching which is robust to the viewpoint of robot.

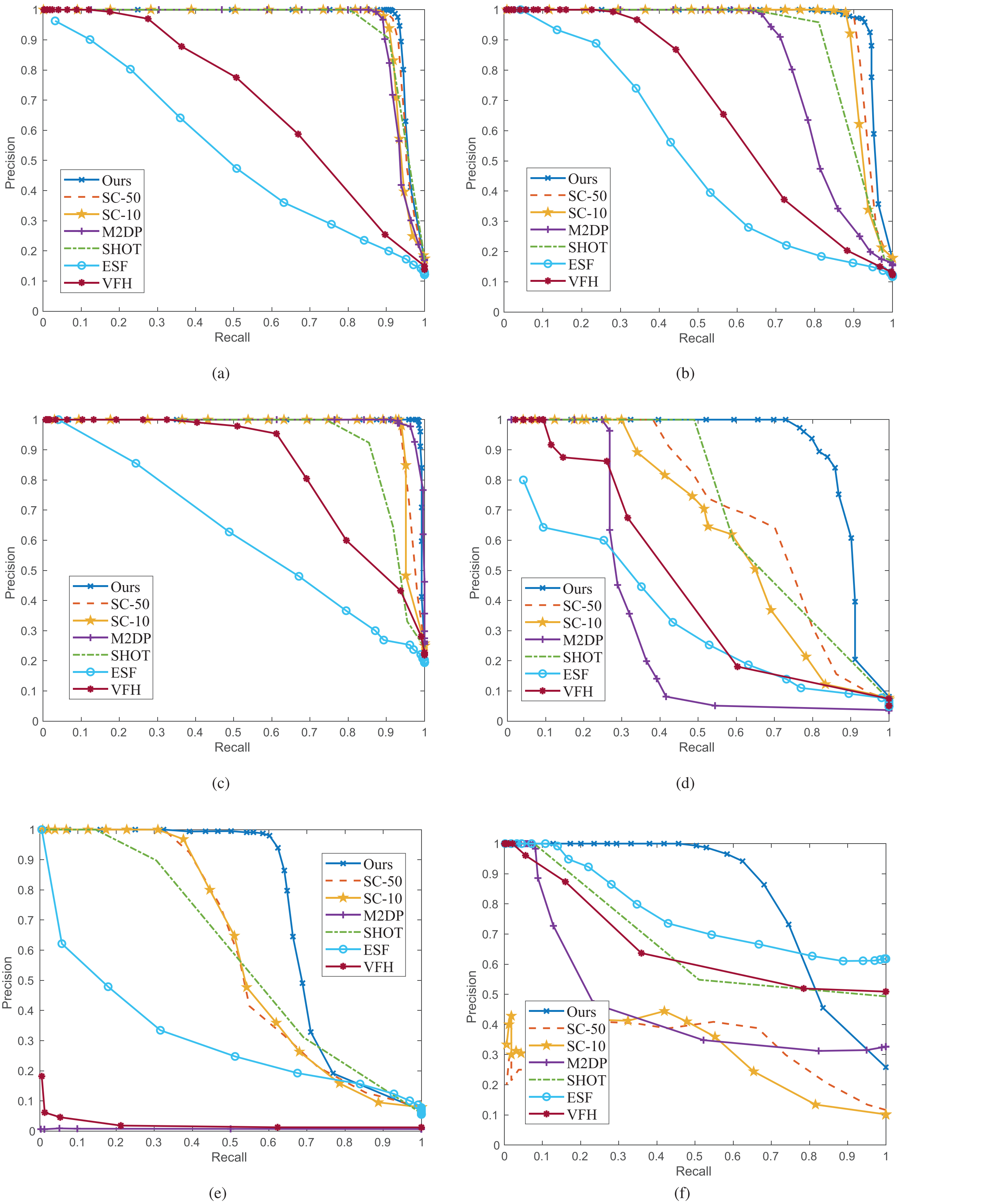
The results in terms of the recall at 100% precision are listed in Table 2. Our framework achieves the best results on six sequences, and the recalls on the KITTI 07, Hannover2 and NKU-MC are 20.54%, 40.05% and 22.11% better than the second-best method, respectively. These quantitative results demonstrate that our framework guarantees high recall with no misrecognition. In addition, the maximum F1-scores for the methods listed in Table 3, and the F1-score is calculated by

$$F1 = 2 \cdot pr \cdot rc/(pr + rc) \tag{8}$$

where *pr* and *rc* represent the precision and recall. For the maximum F1-scores, our framework also achieves the best results in all sequences, which further proves the superiority of the presented framework.

Fig. 8 shows the qualitative visualization of the proposed method corresponding to the recall at 100% precision, where the true positives are marked by red and the false negatives by black. In the KITTI 00, 05, 06, 07 and NKU-MC sequences, our framework recognizes most of places correctly, demonstrating that our framework performs well in most scenarios. In the KITTI 08 and Hannover2, though there exist some false negatives, the robot still keeps retrieving the true positives along the trajectory, which provides frequently detected loop closures for the localization and mapping.

Moreover, it is worth pointing out that our framework obviously outperforms the comparison methods in the environments containing multiple similar places (e.g., KITTI 07). Because in our framework, the relative spatial relations between the clusters are fully exploited which makes the places with similar structures distinguishable from each other. For example, the scans of frame 796 and frame 846 in KITTI 07 are collected at different places marked in Fig. 9(a). The two frames of data were collected on a narrow street with abundant parked vehicles, the appearances and structures of the two frames are similar to each other, which is very challenging for distinguishing the different places, as can be seen in Fig. 9(b). In our framework, the differences between the two



(a)

(b)

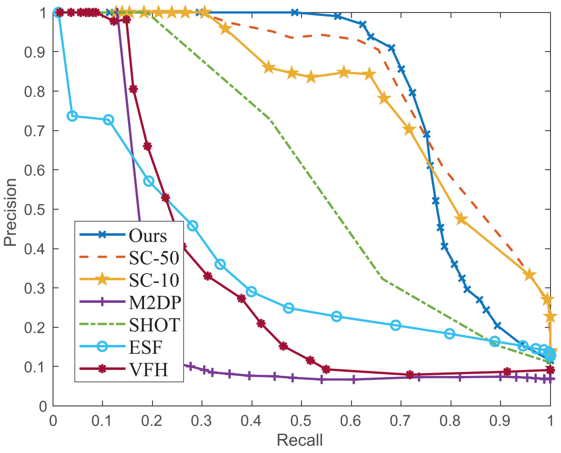
(c)

(d)

(e)

(f)

Fig. 7. The P-R curve for each evaluation sequence. Fig. 7(a), (b), (c), (d), (e), (f) and (g) show P-R curves in KITTI 00, KITTI 05, KITTI 06, KITTI 07, KITTI 08, Hannover2 and NKU-MC, respectively.

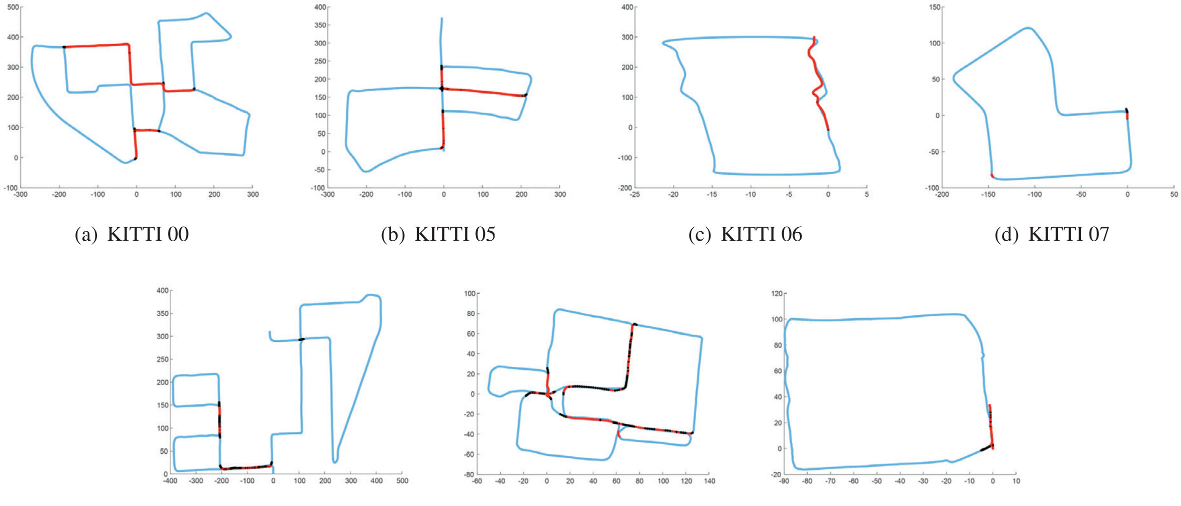


(g)

Fig. 7. Continued

Table 3
Maximum F1-scores in each evaluation sequence.

	SC-50	SC-10	M2DP	SHOT	VFH	ESF	Ours
KITTI 00	0.9459	0.9347	0.9335	0.9332	0.6454	0.5025	**0.9543**
KITTI 05	0.9408	0.9363	0.8000	0.8856	0.6125	0.4939	**0.9481**
KITTI 06	0.9655	0.9690	0.9704	0.8928	0.7566	0.5675	**0.9907**
KITTI 07	0.6769	0.6178	0.4194	0.6939	0.4490	0.3953	**0.8670**
KITTI 08	0.5886	0.5856	0.0174	0.4688	0.0540	0.3483	**0.7519**
Hannover2	0.5120	0.4498	0.4920	0.6606	0.6747	0.7638	**0.7665**
NKU-MC	0.7613	0.7365	0.2648	0.5556	0.3570	0.3324	**0.7806**



(a) KITTI 00
(b) KITTI 05
(c) KITTI 06
(d) KITTI 07
(e) KITTI 08
(f) Hannover2
(g) NKU-MC

Fig. 8. Qualitative performance visualization of our framework at 100% precision (0 false positives) along the trajectory. The true positives are marked by red, the false negatives are marked by black and the true negatives are marked by blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

frames can be described by the relative spatial relations between the segmented clusters, as is illustrated in Fig. 9(c), which makes the presented framework perform place recognition correctly.

In order to further show the P-R performance of our framework, two state-of-the-art algorithms, i.e., SG [35] and Locus [36], are compared with our framework. In SG, the point clouds are segemted by a semantic segmentation method and the places are represented by the semantic graphs. Then, the similarities between the graphs are estimated by a SimGNN-based network. The Locus segments the point cloud and extracts the segment features by a 3D CNN. Both the segment features and the topological information related to the segments are encoded into a global descriptor to evaluate the similarities between two places. In the experiment, the same experimental setup and data sequences are used as in [36] and [35], and the comparison results in term of the maximum F1-score are listed in Table 4, where the results of SG and Locus are reported in [35] and [36], respectively. Our framework achieves the best results on most sequences. For the average maximum F1-scores on the six sequences, our framework performs 4.0% better than the second-best method. It is worth pointing out that

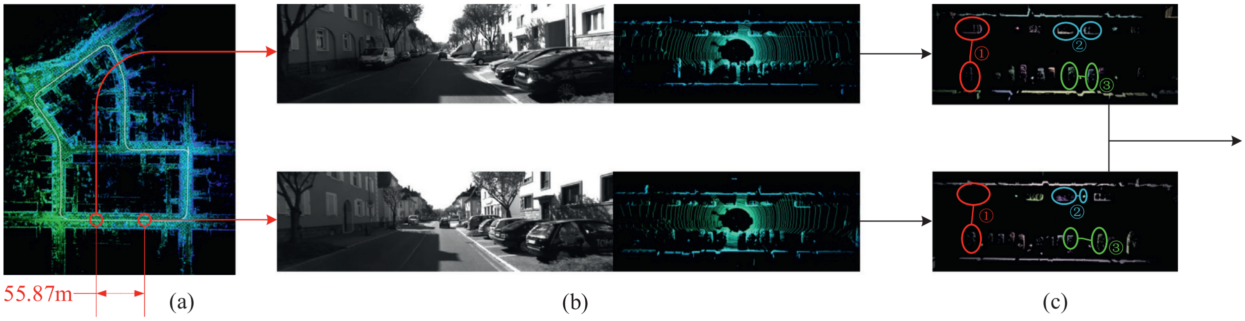


Fig. 9. The illustration of the recognition process at frame 796 and frame 846 in KITTI 07. (a) The locations of the vehicle with the LiDAR sensor at the two frames, respectively, are marked in the map, and the distance between the two loactions is 55.87 m. (b) The images and point clouds taken at the two loactions, respectively. And the appearances and structures of the two places are extremely similar. (c) The clusters generated by the segmentation method. The differences between the two frames can be identified by analyzing the spatial relations between the segmented clusters. For example, ① there is no object in the same relative position of a car. ② There is an object with different shapes in the same relative position of a car. ③ Although the shapes of the cars are similar, their relative spatial relations are different.

Table 4
Maximum F1-scores in each evaluation sequence.

	KITTI 00	KITTI 02	KITTI 05	KITTI 06	KITTI 07	KITTI 08	Mean
SG-RN	0.960	0.859	0.897	0.944	0.984	0.783	0.904
SG-SK	0.969	0.891	0.905	0.971	0.967	0.900	0.934
locus	0.983	0.762	0.981	0.992	**1.0**	0.931	0.942
Ours	**0.995**	**0.943**	**0.994**	**1.0**	0.993	**0.966**	**0.982**

Table 5
Retrieval recall with different SRG ordering in BoW tree.

	KITTI 06	KITTI 07	Hannover2	NKU-MC
Positive Order	1.0	0.9896	0.9100	0.9111
Reverse Order	1.0	0.9896	0.9192	0.9067

Table 6
Average retrieval time with different SRG ordering in BoW tree (ms).

	KITTI 06	KITTI 07	Hannover2	NKU-MC
Positive Order	4.235	4.559	2.744	4.228
Reverse Order	4.314	4.348	2.812	4.167

both the SG and Locus require a complex offline training process. In comparison, our framework does not need offline training and all the steps can be run without any priori information of the environment.

In addition, the proposed framework can ensure reasonable calculation efficiency. The runtimes of the *description* phase and *searching* phase of our framework are computed on the KITTI 00, which are 0.2124s and 0.2355s on average, respectively. Therefore, the place recognition performed by our framework can be run in real-time at approximately 2–3 Hz.

4.4. Evaluation on robustness of U-LSM against SRG ordering

To demonstrate that the impact of SRG ordering on the U-LSM is not significant, we experimentally verify the influence on the SRG ordering on retrieval performance and retrieval time of the incremental BoW tree. The experimental setups and the parameters of our framework are the same as those in the P-R experiment in Section 4.3. For the parameters of the U-LSM, the number of retrievals K_{SRG} is set to 25. The retrieval performance is evaluated by the retrieval recall of the U-LSM. In order to analyze the influence of the SRG ordering on the retrieval performance, the SRGs are inputted into the tree in positive order and in reverse order, respectively. The recalls of retrieval are calculated and compared on four sequences KITTI 06, KITTI 07, Hannover2 and NKU-MC from three datasets, respectively, as shown in Table 5. It can be seen that, on the same sequences, the retrieval recalls of BoW tree are approximative for different ordering of SRGs. Especially for the KITTI 06 and 07, the retrieval recalls are equal for the positive and reverse order. The results demonstrate that the retrieval performance of BoW tree is hardly impacted by the SRG ordering.

Table 7
Recall at 100% precision when the viewpoint change of robot is greater than 30°.

	KITTI 00	KITTI 05	KITTI 08
M2DP	0.05333	0	0
SC-50	0.34722	0.51563	0.33838
SC-10	0.28070	0.54688	0.33333
ESF	0	0.01563	0.00754
VFH	0	0	0
SHOT	0.01333	0.15625	0.19849
Ours	**0.65333**	**0.59375**	**0.36432**

Similarly, for the influence of the SRG ordering on the retrieval time of the incremental BoW tree, the SRGs are inputted into the tree in positive and reverse order, respectively. The time consumption of retrieval are compared on the sequences KITTI 06, KITTI 07, Hannover2 and NKU-MC, respectively, as shown in Table 6.

4.5. Evaluation on robustness against viewpoint

When a place is revisited by the robot from a different viewpoint, the spatial distribution of the point cloud may change significantly, which is substantially challenging for the place recognition. However, our framework benefits from the graph matching which is robust against changes in viewpoint. Hence, the changes of viewpoint have less negative impact on our framework. In sequences KITTI 00, 05 and 08, there are multiple places that are revisited by the robot from different viewpoints. The recall at 100% precision on the three sequences are calculated when the robot revisits a place with a viewpoint change larger than 30°. As shown in Table 7, compared with other algorithms, our framework achieves the highest recall at 100% precision, demonstrating that our framework achieves high viewpoint robustness.

5. Conclusion

This paper has proposed a two-level framework for 3D LiDAR place recognition based on the SRG that captures the spatial relations between clusters segmented from the environments. First, the effective ground removal and segmentation methods have been applied to segment the point cloud into multiple independent clusters. Then, the SRDs between the clusters have been extracted and the point cloud is represented by the SRG to describe the environment. Finally, two effective models have been fused into a two-level model for matching the SRGs. In the U-LSM, an incremental BoW model is utilized to quickly search the candidates through the distribution of the SRDs in the SRG, and then the improved spectral method is used to calculate the similarities between the current SRG and the candidates in the L-LMM. The two models have a bi-directional information exchange, which improves the performance of the overall model. Extensive experiments have been conducted on both the public datasets and the self-built dataset. The results demonstrate that the SRGs contains sufficient information to describe the environments and the proposed framework performs place recognition with high precision, recall and viewpoint robustness.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Jiang, J. Yuan, X. Zhang, X. Zhang, Dvio: an optimization-based tightly coupled direct visual-inertial odometry, IEEE Trans. Ind. Electron. (2020), doi:10.1109/TIE.2020.3036243.

[2] Q. Sun, J. Yuan, X. Zhang, F. Duan, Plane-edge-slam: seamless fusion of planes and edges for slam in indoor environments, IEEE Trans. Autom. Sci. Eng. (2020), doi:10.1109/TASE.2020.3032831.

[3] R. Muoz-Salinas, M.J. Marn-Jimenez, R. Medina-Carnicer, Spm-slam: simultaneous localization and mapping with squared planar markers, Pattern Recognit. 86 (2019) 156–171, doi:10.1016/j.patcog.2018.09.003.

[4] X. Zhang, L. Wang, Y. Su, Visual place recognition: a survey from deep learning perspective, Pattern Recognit. (2020) 107760, doi:10.1016/j.patcog.2020.107760.

[5] S. Lowry, N. Sünderhauf, P. Newman, J.J. Leonard, D. Cox, P. Corke, M.J. Milford, Visual place recognition: a survey, IEEE Trans. Rob. 32 (1) (2015) 1–19.

[6] J. Yuan, W. Zhu, X. Dong, F. Sun, X. Zhang, Q. Sun, Y. Huang, A novel approach to image-sequence-based mobile robot place recognition, IEEE Trans. Syst. Man Cybern. (2019).

[7] N. Piasco, D. Sidib, C. Demonceaux, V. Gouet-Brunet, A survey on visual-based localization: on the benefit of heterogeneous data, Pattern Recognit. 74 (2018) 90–109, doi:10.1016/j.patcog.2017.09.013.

[8] B. Steder, M. Ruhnke, S. Grzonka, W. Burgard, Place recognition in 3d scans using a combination of bag of words and point feature based relative pose estimation, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 1249–1255.

[9] M. Angelina Uy, G. Hee 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.

[10] Z. Liu, C. Suo, S. Zhou, F. Xu, H. Wei, W. Chen, H. Wang, X. Liang, Y.-H. Liu, Seqlpd: Sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2019, pp. 1218–1223.

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

[12] L. He, X. Wang, 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.

[13] G. Kim, 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.

[14] S.A. Scherer, A. Kloss, A. Zell, Loop closure detection using depth images, in: 2013 European Conference on Mobile Robots, IEEE, 2013, pp. 100–106.

[15] R.B. Rusu, G. Bradski, R. Thibaux, J. Hsu, Fast 3d recognition and pose using the viewpoint feature histogram, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 2155–2162.

[16] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, 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.

[17] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, C. Cadena, Segmap: segment-based mapping and localization using data-driven descriptors, Int. J. Rob. Res. 39 (2–3) (2020) 339–355.

[18] P.J. Besl, N.D. McKay, Method for registration of 3-d shapes, in: Sensor Fusion IV: Control Paradigms and Data Structures, volume 1611, International Society for Optics and Photonics, 1992, pp. 586–606.

[19] A. Censi, An ICP variant using a point-to-line metric, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 19–25.

[20] , Aligning point cloud views using persistent feature histograms", author="rusu, radu bogdan and blodow, nico and marton, zoltan csaba and beetz, michael, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2008, pp. 3384–3391.

[21] R.B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3d registration, in: 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009, pp. 3212–3217.

[22] F. Tombari, S. Salti, L. Di Stefano, A combined texture-shape descriptor for enhanced 3d feature matching, in: 2011 18th IEEE International Conference on Image Processing, IEEE, 2011, pp. 809–812.

[23] J. Guo, P.V. Borges, C. Park, A. Gawel, Local descriptor for robust place recognition using lidar intensity, IEEE Rob. Autom. Lett. 4 (2) (2019) 1470–1477.

[24] Y. Zhuang, N. Jiang, H. Hu, F. Yan, 3-D-laser-based scene measurement and place recognition for mobile robots in dynamic indoor environments, IEEE Trans. Instrum. Meas. 62 (2) (2012) 438–450.

[25] F. Cao, Y. Zhuang, H. Zhang, W. Wang, Robust place recognition and loop closing in laser-based slam for UGVs in urban environments, IEEE Sens J 18 (10) (2018) 4242–4252.

[26] B. Steder, G. Grisetti, W. Burgard, Robust place recognition for 3d range data based on point features, in: 2010 IEEE International Conference on Robotics and Automation, IEEE, 2010, pp. 1400–1405.

[27] H. Bay, T. Tuytelaars, L. Van Gool, Surf: speeded up robust features, in: European conference on computer vision, Springer, 2006, pp. 404–417.

[28] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: 2011 International Conference on Computer Vision, Ieee, 2011, pp. 2564–2571.

[29] X. Song, S. Jiang, L. Herranz, C. Chen, Learning effective RGB-D representations for scene recognition, IEEE Trans. Image Process. 28 (2) (2018) 980–993.

[30] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, C. Stachniss, F. Fraunhofer, OverlapNet: loop closing for LiDAR-based SLAM, in: Proc. of Robotics: Science and Systems (RSS), 2020.

[31] X. Xu, H. Yin, Z. Chen, Y. Li, Y. Wang, R. Xiong, Disco: differentiable scan context with orientation, IEEE Rob. Autom. Lett. 6 (2) (2021) 2791–2798.

[32] E. Fernández-Moral, P. Rives, V. Arévalo, J. González-Jiménez, Scene structure registration for localization and mapping, Rob. Auton. Syst. 75 (2016) 649–660.

[33] R. Cupec, E.K. Nyarko, D. Filko, A. Kitanov, I. Petrović, Place recognition based on matching of planar surfaces and line segments, Int. J. Rob. Res. 34 (4–5) (2015) 674–704.

[34] R. Finman, L. Paull, J.J. Leonard, Toward object-based place recognition in dense rgb-d maps, ICRA Workshop Visual Place Recognition in Changing Environments, Seattle, WA, volume 76, 2015.

[35] X. Kong, X. Yang, G. Zhai, X. Zhao, X. Zeng, M. Wang, Y. Liu, W. Li, F. Wen, Semantic graph based place recognition for 3D point clouds, arXiv preprint arXiv:2008.11459 (2020).

[36] K. Vidanapathirana, P. Moghadam, B. Harwood, M. Zhao, S. Sridharan, C. Fookes, Locus: liDAR-based place recognition using spatiotemporal higher-Order pooling, arXiv preprint arXiv:2011.14497 (2020).

[37] D. Zermas, I. Izzat, N. Papanikolopoulos, Fast segmentation of 3d point clouds: a paradigm on lidar data for autonomous vehicle applications, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 5067–5073.

[38] H. Gao, X. Zhang, J. Yuan, J. Song, Y. Fang, A novel global localization approach based on structural unit encoding and multiple hypothesis tracking, IEEE Trans. Instrum. Meas. 68 (11) (2019) 4427–4442.

[39] T.S. Caetano, J.J. McAuley, L. Cheng, Q.V. Le, A.J. Smola, Learning graph matching, IEEE Trans. Pattern Anal Mach. Intell. 31 (6) (2009) 1048–1058.

[40] G.-J. Wen, J.-j. Lv, W.-x. Yu, A high-performance feature-matching method for image registration by combining spatial and similarity information, IEEE Trans. Geosci. Remote Sens. 46 (4) (2008) 1266–1277.

[41] D. Filliat, Interactive learning of visual topological navigation, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 248–254.

[42] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, volume 2, IEEE, 2005, pp. 1482–1489.

[43] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3354–3361.

[44] robotic, [3D scan repository, Hannover, Germany], 2009[Online; accessed 10-August-2009]. [Online]. Available: http://kos.informatik.uni-osnabrueck.de/3Dscans/.

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

[46] J.P.S. do Monte Lima, V. Teichrieb, An efficient global point cloud descriptor for

object recognition and pose estimation, in: 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), IEEE, 2016, pp. 56–63.

Yansong Gong received the B.Sc. degree in software engineering from Nankai University, Tianjin, China, in 2019, where he is currently pursuing the M.Sc. degree in software engineering. His current research interests include place recognition and SLAM.

Fungchi Sun received the B.Sc. degree in industrial automation and the M.Sc. degree in automation from the Shandong University of Science and Technology, Qingdao, China, in 1994 and 1998, respectively, and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2003. He is currently an Associate Professor with the College of Software, Nankai University. His current research interests include autonomous mobile robots and embedded systems.

Jing Yuan received the B.Sc. degree in automatic control and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2002 and 2007, respectively. He was with the College of Computer and Control Engineering, Nankai University from 2007 to 2018, where he is currently a Professor with the College of Artificial Intelligence. His current research interests include motion planning, SLAM, and target tracking.

Wenbin Zhu received the B.Sc. degree in software engineering from Nankai University, Tianjin, China, in 2019, where he is currently pursuing the M.Sc. degree in control science and engineering. His current research interests include place recognition and target tracking.

Qinxuan Sun received the B.Sc. degree in electronic information engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2013, and the M.Sc. degree in control science and engineering from Nankai University, Tianjin, China, in 2016, where she is currently pursuing the Ph.D. degree. Her current research interests include mobile robot navigation and simultaneous localization and mapping.