3D Point Cloud Learning for Large-scale Environment Analysis and Place Recognition

Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Yingtian Liu, Yun-Hui Liu
The Chinese University of Hong Kong

Hesheng Wang
Shanghai Jiao Tong University

Abstract

In this paper, we develop a new deep neural network which can extract discriminative and generalizable global descriptors from the raw 3D point cloud. Specifically, two novel modules, Adaptive Local Feature Extraction and Graph-based Neighborhood Aggregation, are designed and integrated into our network. This contributes to extract the local features adequately, reveal the spatial distribution of the point cloud, and find out the local structure and neighborhood relations of each part in a large-scale point cloud with an end-to-end manner. Furthermore, we utilize the network output for point cloud based analysis and retrieval tasks to achieve large-scale place recognition and environmental analysis. We tested our approach on the Oxford RobotCar dataset. The results for place recognition increased the existing state-of-the-art result (PointNetVLAD) from 81.01% to 94.92%. Moreover, we present an application to analyze the large-scale environment by evaluating the uniqueness of each location in the map, which can be applied to localization and loop-closure tasks, which are crucial for robotics and self-driving applications.

1. Introduction

Autonomous navigation is paramount significant in robotic community such as helping self-driving vehicles [12] and unmanned aerial vehicles [21] achieve full autonomy. A common framework for an intelligent navigation system is to built a 3D map using a vision or LiDAR-based method, plan an optimal trajectory in conjunction with the task, and drive the robots to the destination [4]. Large-scale environment analysis and place recognition represent key challenges of accurate and reliable mapping, localization and planning, especially when a navigation system needs to operate in an unstructured environment [3].

Current solutions for environment analysis and place recognition mainly fall into two categories, image-based and 3D point cloud-based. Many successful image-based approaches were proposed in the literature, due to the feasibility of extracting visual feature descriptors (e.g. SURF [6] or ORB [16]). These visual feature descriptors can be used directly for environment analysis by studying their distribution, or they can be subsequently aggregated into global descriptors for place recognition [18, 19, 10]. Unfortunately, this kind of solutions are challenging due to its non-robustness, for instance, the same scene appears differently under different season or weather conditions, and the same scene place appears different from different viewpoints, which occurs very often during simultaneous localization and mapping (SLAM) process because there is no guarantee that a robot will observe each local scene always from the same viewpoint [23].

Taking into account the limitations of the image-based solution, 3D point cloud-based approach provides an alternative option, which is more robust to external illumination and viewpoint changes. However, compared to feature extraction algorithms for visual images, there is no similar approach designed for point clouds that can reach the same level of maturity [22]. Hence, this manuscript considers 3D point cloud feature learning for their potential to provide reliable environment analysis and robust place recognition.

In this paper, we extract discriminative and generalizable global descriptors from the original 3D point cloud to represent large-scale scenes, aiming for large-scale place recognition and environment analysis applications. Firstly, we extract local distribution features of each point by considering the respective optimal 3D neighborhood. Secondly, by combining the distribution features and the original point position information, we propose a novel learning network to extract the discriminative global descriptor for large-scale scenes. In addition, graph model is introduced to reveal the relations between neighboring points and inductively learn the local structures. Thirdly, based on the proposed global descriptor, we resolve the retrieval task for large-scale place recognition and achieve the environment analysis purpose to facilitate robotics and self-driving applications.
The contributions of this paper can be summarized as follows: 1) We introduce several local distribution features as network inputs instead of only considering position information of each isolated point. These features represent the generalized information in the local neighborhood of each point, which have been successfully applied on different scene interpretation and place matching applications and achieved efficient and effective performances [24, 3]. 2) We design a graph-based neural network (GNN) to reveal the relations between neighboring points and inductively learn the local structures. The GNN is implemented both in the feature space and in the Cartesian space, in order to take into account the statistics information as well as the spatial continuity information. This contributes to extract the local features adequately, reveal the spatial distribution of the point cloud, and find out the local structure and neighborhood relations of each part in a large-scale point cloud, thus improving the discrimination, generalization and classification performance of the proposed global descriptor. 3) We utilize the proposed global descriptor for point cloud-based retrieval tasks to achieve large-scale place recognition, our results (up to 94.92%) outperform the state-of-the-art PointNetVLAD (up to 81.01%) on Oxford RobotCar dataset. We further present an extension to analyze the large-scale environment by evaluating the uniqueness of each location in the map, which can be implemented in several typical tasks, such as the localization and loop closure detection tasks, to facilitate relevant robotics and self-driving applications.

2. Related Work

Most existing solutions for acquiring point cloud features are achieved by handcraft [17, 18], but they are usually tailored to specific tasks and therefore have poor generalization features. For instance, [17] present FPHF for 2.5D scans, which is a point-wise histogram based 3D feature descriptor. It requires high-density point clouds, so it’s hard to apply to large-scale scenarios. Moreover, CVFH is a point cloud feature designed in [1], it only suitable for detection of partial objects.

In order to solve the above problems, deep neural network was introduced for 3D point cloud feature learning and achieved state-of-the-art performance. Convolution neural network (CNN) has achieved amazing feature learning results for regular 2D image data. However, it is hard to extend the current CNN-based method to 3D point cloud due to their orderless. To this end, on the one hand, some work attempts to alleviate this challenge by converting point cloud input to a regular 3D volume representation, such as the volumetric CNNs proposed in [15] and 3D ShapeNets designed in [25] are suitable for point cloud-based classification and recognition, respectively. Moreover, Vote3D [23] and 3DMatch [29] were designed for local feature learning of small-scale outdoor and indoor environments, respectively. On the other hand, different from volumetric representation, Multiview CNNs [20] projects the 3D data into 2D images so that 2D CNN can be performed. Additionally, until recently PointNet [14] and PointNet++ [13], it became possible to perform feature learning directly based on raw 3D point cloud data. Although PointNet and PointNet++ have achieved superior performance on small-scale shape classification and recognition tasks, they do not scale well for our large-scale environment analysis and place recognition problem. Moreover, PointNet only operates on each point independently, and this independence ignores the relationship between points, leading to the limitation of local feature deletion. Although PointNet++ considers local features, it still only operates each point independently during the local feature learning process.

Traditional point cloud-based environment analysis and place recognition algorithms [4] usually rely on a global, off-line, and high-resolution map, and can achieve centimeter-level localization, but at the cost of time-consuming off-line map registration and data storage requirements. SegMatch [3] proposed a reliable place recognition method based on the matching of 3D segments, where the adopted segments provide a good compromise between local and global descriptions. However, the enough static objects assumption will not always be satisfied for real applications. PointNetVLAD [22] achieves the state-of-the-art place recognition results due to the combination of PointNet and NetVLAD [7], the former is used for feature learning and the latter is responsible for feature aggregation. However, this method does not consider local feature extraction adequately, and does not reveal the spatial distribution of the input point cloud.

3. System Structure

The objective of this paper is to extract discriminative and generalizable global descriptors of an input point cloud, and based on which, to resolve the large-scale place recognition and environment analysis problems. Using the global descriptor, the computational and storage complexity will be greatly reduced, thus facilitating the real-time place recognition and loop closure detection applications.

The system structure of this paper is shown in Figure 1. The original 3D LiDar point cloud is used as system input directly. We design a new deep neural network to extract a global descriptor which uniquely describes the input point cloud and will be stored in a descriptor set. On the one hand, when a new input point cloud is obtained, we can match the global descriptor of the new input cloud with those in the descriptor set to detect that whether the new scene corresponds to a previously identified place, if so, this means that the trajectory has a loop closure and we can update the previously stored descriptor by considering the current information, if not, we can store the new descriptor into the
descriptor set as a new scene. On the other hand, we can analyze the environment by investigating the statistical characteristics of the global descriptors of the already visited places to evaluate the uniqueness of each place and find out several typical places in the environment. This information will greatly facilitate the localization, loop closure detection and mapping tasks in robotics and self-driving applications.

4. Network Design

4.1. The Network Architecture

Our full network architecture is shown in Figure 2. The network directly takes the raw point cloud data composed of a set of 3D points as input, and the output is the aggregated global feature vector. As we mentioned above, most of the existing work is done on small-scale pure object point cloud data (e.g. ModelNet [25] and ShapeNet [27]), but this is not the case for large-scale environments, since the point cloud data at this moment is mainly composed of different objects in the scene and the relationships between the objects. To this end, we have customized for large-scale environments and proposed a network with three main modules, 1) Feature Network, 2) Graph-based Neighborhood Aggregation, and 3) NetVLAD [7]. The NetVLAD is designed to aggregate local feature descriptors and generate the global descriptor vector for an input point cloud. The loss function of the network uses lazy quadruplet loss based on metric learning [22], so that the positive sample distance is reduced during the training process, and the negative sample distance is enlarged to obtain a unique scene description vector. In addition, it has been proven to be permutation invariant [22], thus suitable for 3D point cloud. The reason behind the first network module design is discussed in Section 4.2 and 4.3, and the second module design is discussed in Section 4.4.

4.2. Adaptive Local Feature Extraction

In existing point cloud based networks [14] [13] [22], only the original point position information is considered as network input, local distribution and structure features have not been taken into account. The local features represent the generalized information in the local neighborhood of each point, which have been successfully applied on different scene interpretation and place matching applications and achieved efficient and effective performances [23][2]. Only considering the position information of each point may limit the network ability on extracting the local structures and revealing the spatial distribution of the point cloud [9].

In this paper, we introduce the local distribution features by considering the local 3D structure around each point i. k nearest neighboring points are counted and the respective local 3D position covariance matrix is considered as the local structure tensor. Without loss of generality, we assume that \( \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0 \) represent the eigenvalues of the symmetric positive define covariance matrix. According to [24], \( E_i \) can be used as a measurement to describe the unpredictability of the local structure from the aspect of the Shannon information entropy theory, where

\[
E_i = -L_i \ln L_i - P_i \ln P_i - S_i \ln S_i, \tag{1}
\]

\( L_i = \frac{\lambda_1 - \lambda_2}{\lambda_1} \), \( P_i = \frac{\lambda_2 - \lambda_3}{\lambda_2} \) and \( S_i = \frac{\lambda_3}{\lambda_3} \) represent the linearity, planarity and scattering features of the local neighborhood of each point respectively. These features describe the 1D, 2D and 3D local structure around each point. Since the point distribution in a point cloud is typically uniform, we adaptively choose the neighborhood of each point i by minimizing \( E_i \) across different k values and the optimal neighbor size is determined as

\[
k_{opt}^i = \arg \min_k E_i(k). \tag{2}
\]

Then we select the following ten local features to describe the local distribution and structure information around each point i: Change of curvature \( C_i = \frac{\lambda_2}{\sum_{j=1}^3 \lambda_j} \), Omni-variance \( O_i = \frac{\sum_{j=1}^3 \lambda_j}{\sum_{j=1}^3 \lambda_j^3} \), Linearity \( L_i = \frac{\lambda_1 - \lambda_2}{\lambda_1} \), Eigenvalue-entropy \( A_i = -\sum_{j=1}^3 (\lambda_j \ln \lambda_j) \). Local point
density $D_i = \frac{\zeta_i}{\Pi_{j=1}^{n} \xi_j}$. Vertical component of normal vector $\hat{V}_i$, 2D scattering $S_{i, 2D} = \lambda^1_{2D, 1} + \lambda^1_{2D, 2}$, 2D linearity $L_{i, 2D} = \frac{\lambda^1_{2D, 2}}{\lambda^1_{2D, 1}}$, Maximum height difference $\Delta Z_{i, \max}$, Height variance $\sigma_{Z_i, \max}$. The first six features describe the 3D information of the local point distribution. $S_{i, 2D}$ and $L_{i, 2D}$ are calculated by projecting the 3D point cloud onto the 2D horizontal plane, where $\lambda^1_{2D, 1}$ and $\lambda^1_{2D, 2}$ represent the eigenvalues of the corresponding 2D covariance matrix. The last two local features describe the height distribution characteristics. These features are selected by taking the feature redundancy and discriminability into consideration, which can be used to describe the local spatial distribution and structure information of the point cloud adequately.

### 4.3. Feature Network

Existing point cloud network ([14], [22]) takes the spatial position of each point as input, which is difficult to extend to large-scale applications, since this neglect the fact that the target object’s feature is determined by the neighborhood relationship of the point and the relation between the local and the overall aggregation. Therefore, our feature network is designed to introduce the distribution characteristics of points and replace simple position information by aggregating to form powerful and effective features.

The input to the feature network is the raw point cloud data, which is simultaneously transported to the Input Transformation Net and Adaptive Local Feature Extractor, then map the two kinds of features to the high-dimensional space through a two-layer MLP, and finally make the output of the Feature Network invariant to the spatial transformation through Feature Transformation Net. Experimental results in Section 5 show that the proposed method can fully exploit the local structure information in large environments with occlusion.

![Figure 2. Network Architecture.](image)

The network takes the raw point cloud data as input, applies Feature Network to obtain the point cloud distribution and the enhanced point cloud features, which is aggregated in the feature space and Cartesian space through the graph neural network. The resulted feature vectors are then inputted to NetVLAD [17], and the features are further aggregated to generate a global descriptor of the overall environment.

![Figure 3. Graph formulation.](image)

This formulation. Note that the receptive field of each point corresponds to a local neighborhood in the original point cloud, since the feature network has introduced the local point distribution characteristics and Local structure into the feature of each point. Then the Graph Neural Network can be used to aggregate and extract the description vector of the whole point cloud.

### 4.4. Graph-based Neighborhood Aggregation

Point cloud-based large-scale place recognition has good robustness since it is not affected by illumination, seasons, and the like. A large-scale point cloud environment consists of 3D structures (such as planes, corners, shapes, etc.) of surrounding objects and their distribution relationships (such as relative distance, orientation, etc.). Similar structures in different environments have similar properties (such as structure, size, etc.), which are the main judgments for scene recognition using point cloud data. This is called relational reasoning in the Graph Neural Network, which uses a structured representation to get the composition and the relationship between them. Specifically, we represent the composition of the scene as entities and relations in the graph model, and represent their intrinsic relationships and...
generate unique scene description vectors through Graph Neural Network relational reasoning.

4.4.1 Neighborhood Structure and Neighbor Relation

In the above Feature Network, we have merged the neighborhood structure information into the feature vector of the neighborhood center point, that is, each point can be regarded as the feature description of the surrounding neighborhood. As shown in Figure 3, we use the neighborhood structure represented by each point as the vertex of the graph network, and the edge formed between the nodes as the adjacency relationship between the neighborhoods, and then the Graph Neural Network can be used to aggregate and extract the description vector of the whole environment. To this end, we apply the dynamic graph model on the feature space, as shown in the upper subfigure of Figure 4. In this way we can overcome the Cartesian distance limit and aggregate the neighborhood nodes with the same characteristics to generate more effective local feature vectors. In addition to using graph neural networks in feature spaces, we also extend them to Cartesian space, as shown in the bottom of Figure 4 because the spatial distribution of features is also important for large scene applications. Therefore, we also update the graph model based on the neighbor vertex in the Cartesian space to better record the spatial distribution of the point cloud features. Similar to convolution operation in CNN, the features are extracted step by step from the local to the global, and finally a global description vector is generated by combine each neighborhood vertex and the adjacency edge. Experimental results in Section 5 show that the proposed method using the Cartesian space and feature space graph model has achieved state-of-the-art results.

4.4.2 Graph Neural Network Structure

In the feature space, as shown in Figure 4, we build a dynamic graph $G_{i,d}$ for each point $i$ through the multiple kNN iteration. In this dynamic graph model, the global point cloud feature will be updated with multiple times through updating each point feature $p_i$ in different graphs, thus achieving the point cloud feature extraction. More specifically, in each iteration, the output feature vector of the previous iteration is used as network input and a kNN aggregation is conducted on each point by finding $k$ neighbors with the nearest feature space distances. Each vertex $v_{j,m}$ in graph $G_{i,d}$ represents the point feature $p_{j,m}^i$ and each edge $e_{ij,m}$ represents the feature space relation between point $i$ and its feature space neighbors point $j^m (j^m \in \{j^1, \ldots, j^k\}$ represents the $k$ feature space neighbors of point $i$). In this paper, we define $e_{ij,m} = p_i - p_{j,m}^i$. The mlp network is used to update neighbor relations and the max pooling operation is used to aggregate $k$ edge information into a feature vector to update the point feature $p_i$. In each iteration, the graph $G_{i,d}$ will be updated by considering the current feature distribution in the feature space and the point feature $p_i$ will be updated by aggregating the feature of neighbors in current graph $G_{i,d}$. The feature space kNN network achieves the clustering operation in the feature space, this implies that two points with large Cartesian space distance can also be aggregated to capture similar semantic structures. In contrast, contextual neighborhood information should be more concerned on Cartesian space, and the point cloud feature extraction is also implemented through the neighborhood graph neural network, the vertex and edge update strategies are also the same as in the feature space. But the difference is that the network graph model is constructed only by Euclidean distance kNN clustering on Cartesian space. This is similar to CNN to achieve multi-scale feature learning.

4.4.3 Feature Aggregation Structure

We believe that the Graph Neural Network model in feature space and Cartesian space can aggregate neighborhood features and spatial distribution information separately. In order to combine the advantages of the two graph network aggregation point cloud features, to better extract the intrinsic relationship for scene representation, we designed three feature aggregation methods for exploration: Parallel-Concatenation structure (Figure 5(a)): Cascading the neighborhood feature vectors of the feature space and Cartesian space network outputs, and merging the dual-dimensional information through MLP to aggregate the point cloud features. Parallel-Maxpooling structure (Figure 5(b)): directly integrates the neighborhood feature vectors of the feature space and Cartesian space network outputs through the max pooling layer, and updates the feature neighborhood and spatial neighborhood information in different spaces respectively, taking the maximum value of the feature vector as the unified description vector. Series-FC structure (Figure 5(c)): The feature space network first
learns to aggregate the neighborhoods with the same semantic features, and then inputs the features into Cartesian space network so that the features of the neighborhood and spatial relationships are aggregated. The experimental results show that the third structure has the best feature aggregation properties, please refer to Section 5 for more details.

5. Experiment

5.1. Implementation Details

In this subsection, we describe the implementation issues of the overall network. The network takes the raw point cloud data as input, applies Feature Network (FN) to obtain the point cloud distribution and the enhanced point cloud features, which is aggregated in the feature space (FSN) and Cartesian space (CSN) through the graph neural network. We evaluated three methods for performing feature aggregation: Parallel-Concatenation (PC), Parallel-Maxpooling (PM), Series-FC (SF), and the resulted feature vectors are then inputted to NetVLAD, and the features are further aggregated to generate a global descriptor of the overall environment. The loss function of the network uses lazy quadruplet loss based on metric learning \[22\], so that the positive sample distance is reduced during the training process, and the negative sample distance is enlarged to obtain a unique environment description vector.

As shown in Figure 2, our designed network is divided into three modules: FN, Graph Network (RC, PM, and SF) and NetVLAD. In FN, the network configuration can be described as \( P_{1N}(3) - (IT(3), ANFE(10)) - C(13) - MLP(64, 64) - FT(64) - P_{FN}(64) \), where \( P_{1N}(3) \) represents the raw point cloud data \((N \times 3)\), \( P_{FN}(64) \) denotes the feature vector \((N \times 64)\) output by FN. \((IT(3), ANFE(10))\) means that the Input Transformation Net and Adaptive Neighborhood Feature Extractor are parallel, and the feature vectors of \( N \times 3 \) and \( N \times 10 \) are obtained, respectively. \( C(13) \) indicates that the 13-dimensional feature vector is obtained by connecting the channels in series, and \( FT(64) \) represents the input vector to the Feature Transformation Net. NetVLAD is the same as the version stated in \[22\], and lazy quadruplet loss is used with \( \alpha = 0.5, \beta = 0.2, P_{pos} = 2, P_{neg} = 18 \). In the Graph Network, PC’s network configuration is \( P_{FN}(64) - (FSN(64), CSN(64)) - C(128) - MLP(64, 128, 1024) - P_{GN}(1024) \). More specifically, the network configuration of FSN is \( kNN_F(20 \times 64) - GB(20 \times 128) - MLP(64, 64) - M(64) \), where \( kNN_F(k \times f) \) means kNN with cluster number \( k \) in the feature space, and the dimension of the vertex is \( f \)-dimensional. \( GB(k \times 2f) \) represents the update process of the graph model, and \( M() \) represents the maxpooling in the neighborhood. \( kNN_C(20 \times 64) - GB(20 \times 64) - MLP(64, 64) - M(64) \) is the network configuration of the CSN, and \( kNN_C(k \times f) \) means kNN is carried out in the Cartesian space. Similarly, PM’s network configuration is \( P_{FN}(64) - (FSN(64), CSN(64)) - C(128) - MLP(64, 128, 1024) - P_{GN}(1024) \) with max pooling for aggregation, and SF’s network configuration is \( P_{FN}(64) - FSN(64) - CSN(64) - MLP(64, 128, 1024) - P_{GN}(1024) \). All experiments are conducted with a 1080Ti GPU on TensorFlow, showing great potential for real-time applications.

5.2. Dataset

We train and evaluate the network on the Oxford Robotcar dataset \[11\]. For the application of lidar point cloud data in large-scale environments, we also directly transplant the model to the KITTI dataset \[5\] for evaluation and verify its generalization ability. Specifically, The Oxford RoboCar Dataset is obtained by vertical scanning of the SICK LMS-151 2D LiDAR mounted on the car. The 3D point cloud submap is made up of point clouds within the car’s 20m trajectory. We use its 44sets, 21,711 training submaps to train our network, use its 3030 testing submaps for evaluation. The point cloud of each submaps contains 4096 points and is normalized to the range \([-1, 1]\).

We evaluated the network trained by the Oxford dataset on the KITTI Odometry dataset. Unlike the Oxford dataset, the point cloud data of the KITTI dataset was obtained by the Velodyne-64 LIDAR in the outdoor scan. The scanned data per frame is about 100k points. We removed the ground from KITTI and downsampled the original point cloud, tak-
ing a range of 25 meters long and 10 meters wide, using the voxel grid filter and random sampling downsampling to 4096 points, and normalized to the range of [-1,1].

### 5.3. Place Recognition Results

![Figure 6. Average recall under different networks.](image)

Table 1. Comparison results of the average recall (%) at top 1% (@1%) and at top 1 (@1) under different networks.

|                     | Ave recall @1% | Ave recall @1 |
|---------------------|----------------|---------------|
| PN STD              | 46.52          | 31.87         |
| FN-VLAD             | 73.87          | 54.16         |
| FN-VLAD baseline    | 81.01          | 62.76         |
| FN-VLAD refine      | 80.77          | 63.33         |
| FN-PM-VLAD (our)    | 79.21          | 61.96         |
| FN-PC-VLAD (our)    | 89.77          | 75.79         |
| FN-SF-VLAD (our)    | 91.20          | 78.77         |
| FN-SF-VLAD (our)    | 92.27          | 81.41         |
| FN-SF-VLAD (our)    | 94.92          | 86.28         |

Our results demonstrate that our network has superior advantages for point cloud-based place recognition in a large-scale environment, far exceeding PointNetVLAD and reaching the state-of-art. Even if the model trained on Oxford dataset is directly migrated to KITTI, it can achieve good results without training, which shows that our algorithm can realize place recognition and loop-closure in generalized large-scale environments. In the evaluation process, we randomly selected four scenes from the 44 sets data of the Oxford dataset for evaluation. The data collection of 44 sets is in different seasons, different times and different weather, and we querying the same scene in these sets for place recognition. Such place recognition with large time span and light changes is not possible with images. Specifically, we use the network model to generate an environment description vector from the point cloud, and query the scene closest to the Euclidean distance of the test scene vector to determine whether it is the same scene. We use Recall to evaluate the ability of place recognition to see if there is a real scene in the top N scenes closest to it. We compare it with Average Recall@N and Average Recall@1.

We compare our approach to the original PointNet architecture with the maxpool layer (PN MAX), and PointNet trained for object classification in ModelNet (PN STD) to study whether the model trained on small-scale dataset can be scaled to large-scale environments. Moreover, we also compare our network with the state-of-the-art PN-VLAD baseline, and PN-VLAD refine [22] to show the performance of our algorithm. We train the PN STD, PN MAX, PN-VLAD baseline, and PN-VLAD refine using only the Oxford training dataset. The network configurations of PN STD and PN MAX are set to be the same as [14], and network configurations of PN-VLAD baseline and PN-VLAD refine are set to be the same as [22].

Comparison results with PointNetVLAD are shown in Figure 6 where FN-PM-VLAD, FN-PC-VLAD, and FN-SF-VLAD are our network with three different feature aggregation structure PM, PC, and SF, FN-VLAD represent the network without graph-based neighborhood aggregation (See Section 4 for more details). Results show that due to the application of feature distribution statistics and graph neural network model, our four network structures are better than PointNetVLAD. The best experimental results are obtained by FN-SF-VLAD, and its recall accuracy in Top 5 is higher than 95%. The accuracy of such scene recognition can meet the requirements of current unmanned vehicle positioning and SLAM closed loop tasks. This recall accuracy has been able to meet the requirements of self-driving vehicle positioning and SLAM loop-closure tasks.

Additionally, we also designed NeuralNeighborVLAD network (NNVLAD), which uses KNN clustering and MLP instead of ANFE. The output of the network is also 10 dimensions neighborhood information, but the parameters are obtained through network learning. The comparison results are shown in the table 1. According to the results, we can analyze that among the three aggregation methods of FSN and CSN, SF is the most accurate. In SF, the graph neural network learns the neighborhood structure features of the same semantic information in feature space, and then further aggregates in Cartesian space. This method can learn the spatial distribution characteristics of neighborhood features, which can be introduced into the following network and learned as a spatial coordinate relationship. However, the training and reasoning speed of FN-SF-VLAD is slower than that of FN-PM-VLAD and FN-PC-VLAD, since the network is serial, and it is necessary to continuously update the graph model. In addition, FN-PC-VLAD is better than FN-PM-VLAD with faster convergence speed and higher recall accuracy, since it can save more information for subsequent aggregation.

Figure 7 shows some of the successfully matched and recognized point clouds in KITTI dataset, the left sub-figures show the measured point cloud and the right sub-figures show the correctly retrieved one. It can be seen that our network has learned to ignore irrelevant noise during the matching process. An example of the place discrimination
result in the KITTI dataset is given in Figure 8. Although there are two very similar point clouds that correspond to different locations, our proposed network recognizes these two point clouds as different places successfully.

5.4. Application to Environment Analysis

We first evaluate the similarity of each point cloud with all the other point clouds in the whole environments. The similarity between two point clouds is calculated from the distance between two corresponding global descriptors. Then for a given point cloud, we can calculate the similarity index of this point and plot a similarity map. Figure 9 shows an example of the similarity evaluation result. We can find that the similarity between point cloud (b) and the point clouds with the near location is much larger than those with long location distances. With normalization, the sum of the similarity with all the other point clouds in the whole environment can be used to evaluate the uniqueness of the given point cloud (the given place). Figure 10 shows the uniqueness evaluation results of the whole environment. This uniqueness analysis will greatly facilitate the localization, loop closure detection and mapping tasks in relevant robotics and self-driving, for example, we can choose the point clouds with the largest uniqueness as the key frames in the long-term localization tasks to reduce the redundancy, or we can design an active loop closure and mapping algorithm which conducts the map closure operation in the place with the largest uniqueness in the whole environment to avoid mis-matchings and increase accuracy. These applications will be presented in our future work.

6. Conclusion

In this paper, we propose a novel deep neural network that solves point cloud based place recognition and environment analysis for large-scale applications. We applied adaptive local feature extractor and graph-based neighborhood aggregation for our network to learn discriminative and generalized point cloud features. Experimental results for place recognition increased the existing state-of-art result (PointNetVLAD) from 81.01% to 94.92%. We also present an extension to analyze the large-scale environment by evaluating the uniqueness of each location in the map.
References

[1] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 585–592, 2011.

[2] B. Axelrod, L. P. Kaelbling, and T. Lozano-Pérez. Provably safe robot navigation with obstacle uncertainty. The International Journal of Robotics Research, 2018.

[3] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena. Segmatch: Segment based place recognition in 3d point clouds. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 5266–5272, 2017.

[4] J. Fossel, K. Tuyils, B. Schmieders, D. Claes, and D. Hennes. Noctoslam: Fast octree surface normal mapping and registration. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 6764–6769, 2017.

[5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: the kitti dataset. The International Journal of Robotics Research, 32(11):1231–1237, 2013.

[6] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, pages 404–417, 2006.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Proceedings of the Conference on Neural Information Processing Systems, 2012.

[8] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using local affine regions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8):1265–1278, 2005.

[9] J. Li, B. M. Chen, and G. H. Lee. So-net: Self-organizing network for point cloud analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[10] S. Lowry, N. Snderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. IEEE Transactions on Robotics, 32(1):1–19, 2016.

[11] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000km: the oxford robotcar dataset. The International Journal of Robotics Research, 36(1):3–15, 2017.

[12] T. Ort, L. Paull, and D. Rus. Autonomous vehicle navigation in rural environments without detailed prior maps. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 2040–2047, 2018.

[13] C. R. Qi, L. Li, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Conference on Neural Information Processing Systems, 2018.

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[15] C. R. Qi, H. Su, M. Niessner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5648–5656, 2016.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In Proceedings of the IEEE International Conference on Computer Vision, pages 2564–2571, 2011.

[17] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 3212–3217, 2009.

[18] S. A. Sadat, K. Chutkoff, D. Jungic, J. Wawerla, and R. Vaughan. Feature-rich path planning for robust navigation of mavs with mono-slam. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 3870–3875, 2014.

[19] S. Soatto. Actionable information in vision. In Proceedings of the IEEE International Conference on Computer Vision, pages 2138–2145, 2009.

[20] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE International Conference on Computer Vision, pages 945–953, 2015.

[21] K. Sun, K. Mohta, B. Pfommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar. Robust stereo visual inertial odometry for fast autonomous flight. IEEE Robotics and Automation Letters, 3(2):965–972, 2018.

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

[23] D. Z. Wang and I. Posner. Voting for voting in online point cloud object detection. In Proceedings of the Robotics: Science and Systems, 2015.

[24] M. Weimann, J. Boris, and M. clément. Semantic 3d scene interpretation: a framework combining optimal neighborhood size selection with relevant features. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(3):181, 2014.

[25] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1912–1920, 2015.

[26] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1912–1920, 2018.

[27] L. Yi, V. G. Kim, D. Ceylan, L.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3d shape collections. In Proceedings of the SIGGRAPH Asia, 2016.

[28] P. Yin, L. Xu, Z. Liu, L. Li, H. Salman, Y. He, W. Xu, H. Wang, and H. Choset. Stabilize an unsupervised feature learning for lidar-based place recognition. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2018.

[29] A. Zeng, S. Song, M. Niener, M. Fisher, and J. Xiao. 3dmatch: Learning the matching of local 3d geometry in
range scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.