Scientific Research Publishing

Advances in Pure Mathematics, 2021, 11, 502-523
https://www.scirp.org/journal/apm
ISSN Online: 2160-0384
ISSN Print: 2160-0368

WADE-Net: Weighted Aggregation with Density Estimation for Point Cloud Place Recognition

Ke Liu[1*], Xin Wang[1], Yaxin Peng[1*], Zhen Ye[2], Chaozheng Zhou[2]

[1]College of Science, Shanghai University, Shanghai, China
[2]Shanghai Electric Central Research Institute, Shanghai, China
Email: *kkliu1996@shu.edu.cn, xinwang@shu.edu.cn, *yaxin.peng@shu.edu.cn, zhye1985@aliyun.com, zhouchzh2018@aliyun.com

How to cite this paper: Liu, K., Wang, X., Peng, Y.X., Ye, Z. and Zhou, C.Z. (2021) WADE-Net: Weighted Aggregation with Density Estimation for Point Cloud Place Recognition. *Advances in Pure Mathematics*, **11**, 502-523.
https://doi.org/10.4236/apm.2021.115035

Received: April 28, 2021
Accepted: May 24, 2021
Published: May 27, 2021

Abstract

Point cloud based place recognition plays an important role in mobile robotics. In this paper, we propose a weighted aggregation method from structure information adaptively for point cloud place recognition. Firstly, to preserve the prior distributions and local geometric structures, we fuse learned hidden features with handcrafted features in the beginning. Secondly, we further extract and aggregate adaptively weighted features concerning density and relative spatial information from these fused features, named Weighted Aggregation with Density Estimation (WADE) module. Then, we conduct the WADE block iteratively to group the latent manifold structures. Finally, comparison results on two public datasets Oxford Robotcar and KITTI show that the proposed approach exceeds the comparison approaches on recall rate averagely 7% - 8%.

Keywords

Point Cloud, Place Recognition, Deep Learning, Feature Extraction

1. Introduction

Large-scale place recognition plays a significant role in robotics and automatic driving, since it usually enhances localization and mapping optimization [1]-[6]. Vision data based large-scale place recognition has been investigated, and some successful solutions are presented in several surveys [7] [8] [9]. However, they are sensitive to season and illumination variations. Meanwhile, with the help of spatial-aware feature information, 3D points based methods are relatively robust to these changes [10] [11] [12] [13] [14]. In **Figure 1**, an outdoor scene shows that 3D point cloud could better describe the different spatial distributions of

*Corresponding author.

Figure 1. Different point distributions in an outdoor scene. Three enlarged dashed circles are an advertising board, the rear of one car, and afforestation, respectively.

different local regions. As a consequence, place recognition from point cloud data is becoming an increasingly attractive research topic. The main challenge of point cloud recognition lies in how to extract effective features and generate a discriminative representation. Various feature extraction strategies based on handcrafted or deep learning methods for 3D point clouds emerge gradually [15] [16].

Traditionally, some 3D recognition methods pay attention to handcrafted local feature extraction, including normal orientation, curvature and distribution histogram [17] [18] [19] [20] [21]. Specifically, [19] [20] generate histograms based on geometric attribution to obtain local information. These methods, however, often consume much time and more computational resources. By contrast, some works [18] [21] try to improve the efficiency of local feature extraction methods. They can extract local point cloud features with lower computation cost. However, they do not perform well at sparse space locations. Moreover, these approaches only concentrate on some local views, but not from a global perspective.

Therefore, methods for extracting global descriptors are proposed gradually [22] [23] [24] [25] [26]. Ensemble of Shape Functions (ESF) [22], Normal Aligned Radial Feature (NARF) [23] and Viewpoint Feature Histogram (VFH) [24] are able to filter out some local locations with sparse distribution of point cloud. The conception of projecting 3D point cloud to 2D image is utilized [5] [27] [28]. He *et al.* propose a global descriptor, Multiple 2D Planes (M2DP) [28], for place recognition and loop detection. It projects the 3D point cloud into multiple 2D planes and generates a descriptor vector for place representation. Following the M2DP, Kim *et al.* propose Scan-Context [5] to handle place rec-

ognition on the basis of 3D point database. Scan-Context separates the whole point cloud into many bins by radius and azimuth, and defines the maximum height of points in each bin as the feature value. In general, these traditional works, based on prior knowledge, obtain the handcrafted spatial feature of 3D data and have made contributions on many tasks. However, some latent features may be neglected due to the disadvantage of these handcrafted methods.

Fortunately, deep learning based methods have powerful feature extraction capability relying on numerous data fittings. They have received high attention and have been widely utilized to extracting high-dimension features from order-less point clouds [29]-[38]. Generally speaking, there are three ways of feature extraction for point cloud. Firstly, some works [29] [30] [31] [32] [33] represent the input point cloud as regular 3D gridding or voxel, but this operation may cause complex pre-process and high computational cost [12]. Secondly, motivated by CNN in 2D images, some works [34] [35] consider projecting 3D points into 2D images and using multi-view to analyze point clouds roundly. Finally, PointNet [36] and PointNet++ [37] make it possible to input raw point cloud into a network directly, but they are designed to handle small object classification and indoor scene segmentation. PointNet extracts point-wise learning features, while PointNet++ enriches them by grouping neighbor points for local information. However, it does not consider features in global perspective.

Furthermore, PointNetVLAD [10] uses the NetVLAD [39] block to generate global descriptors. Recently, various modifications of PointNetVLAD emerge [11] [12]. Specifically, PCAN [11] introduces an attention mechanism into the NetVLAD block. However, these methods may ignore the prior information of input data. LPD-Net [12] considers using traditional features to enrich input data, and adds a graph-based neighborhood aggregation module to improve the feature extraction of the network. However, it does not consider the structure information such as density and normal in local regions.

Overall, the existing methods have two main disadvantages: not considering both prior structure and latent manifold structure from data manifold. Therefore, in this paper we propose a traditional feature fusion module for prior structure extraction and a Weighted Aggregation with Density Estimation (WADE) module for iteratively extracting latent structure, respectively. Our contributions include the following three aspects:

- We fuse point coordinates with handcrafted features and neural network learned features to enrich the input information of deep network.
- We provide an iterative WADE module for local structure encoding. Specifically, the WADE introduces a weighted density into local points relative relationships.
- We conduct experiments on two benchmark datasets Oxford Robotcar [40] and KITTI [41] to demonstrate the superiority of WADE-Net over other state-of-the-art methods. Our approach exceeds most of the comparison methods on the recall rate at least 10% at TOP 1.

The rest of this paper is organised as follows. In Section 2, we introduce two

mostly related methods, and the proposed method is based on their framework. In Section 3, our WADE-Net approach is explained in detail. In Section 4, we report the comparison results and some ablation experiments. In Section 5, we draw the conclusions.

2. Related Works

Traditional feature based methods. Usually, handcrafted traditional feature extraction is designed according to prior knowledge of human beings. There are several works for traditional point cloud feature extraction. Spin image (SI) [27] projects 3D points within a cylinder onto a 2D spin image. The 3D shape context [17], Point Feature Histogram (PFH) [20] and Signature Histogram of OrienTation (SHOT) [19] leverage geometric attribution to obtain local features. Fast Point Feature Histogram (FPFH) [21] and 3D Scale-Invariant Feature Transform (SIFT) [18] are proposed to extract local point cloud features with lower computation cost. Subsequently, Ensemble of Shape Functions (ESF) [22], Normal Aligned Radial Feature (NARF) [23] and Viewpoint Feature Histogram (VFH) [24] are proposed to generate a global descriptor for point cloud representation. Multiple 2D Planes (M2DP) [28] projects the 3D point cloud into multiple 2D planes and finally generates a descriptor vector. Kim *et al.* propose Scan-Context [5] to separate the whole point cloud into many bins by radius and azimuth, and a global feature map is obtained. Yan *et al.* [42] propose a sparse semantic map building method and utilize the semantic map to generate special texture features for scene recognition. LiDAR-Iris [25] generates a global descriptor based on a binary signature image obtained from the point cloud. DELIGHT [26] leverages LiDAR intensity information and encodes the information into a representative descriptor. In conclusion, the traditional feature based methods make many contributions to point cloud recognition, but few works fuse them into a learning framework.

Learning feature based methods. Several deep learning based point cloud feature extraction methods have been proposed in recent years. 3D ShapeNets [29], Vote3Deep [30], VoxelNet [31], 3D Generative Adversarial Network (3D-GAN) [32], and Volumetric CNN [33] transform point cloud inputs into regular 3D gridding or voxel representations, which may cause complex pre-processing operations and high computational cost [12]. Multi-View based Convolutional Neural Network (MVCNN) [34] and Group-View CNN (GVCNN) [35] project 3D points into 2D images and use multi-view to analyze point cloud roundly.

PointNet [36] and PointNet++ [37] have the ability to extract point-wise features from a raw point cloud. Inspired by them, networks such as PointCNN [43], Frustum Pointnet [44], SO-Net [45] and Splatnet [46] are proposed for point cloud feature extraction.

Furthermore, PointNetVLAD [10] proposes a new point cloud place recognition method via a global descriptor module. Recently, LPD-Net [12] and PCAN [11] improve PointNetVLAD to recognize places efficiently. However, PCAN may ignore the prior information of input data and it leads to high cost in the

proposed attention module. SeqLPD [4] and LPD-AE [47] utilize LPD-Net as a place recognition module to implement environment construction. Moreover, [15] projects input point cloud into cylindrical coordinates and converts 3D point cloud to 2D image for place recognition. MinkLoc3D [48] uses a 3D feature pyramid network [49] to extract local features, and then it introduces Generalized-Mean (GeM) [50] pooling for global descriptor generation. Locus [51] considers fusing the segmentation, topological and temporal information for point cloud representation. In this paper, we try to enhance the important structure information, including density and spatial relationship.

3. Methodology

For fusing and aggregating meaningful structure and features from point cloud, our network framework is composed of three modules: the prior feature fusion $\mathcal{F}$ (green-dashed block), the iterative WADE $\mathcal{I}$ (yellow-dashed block), and the global descriptor generation module $\mathcal{G}$ (red-dashed block), as shown in Figure 2. The network maps the input raw point cloud $P=\left\{p_n \mid p_n \in \mathbb{R}^3\right\}_{n=1}^{N}$ into a high-dimension feature space for place representation. For a certain place $P \in \mathcal{P}$, a descriptor $\mathcal{M}(P) \in \mathcal{G}$ is generated by

$$\begin{aligned} &\mathcal{M}(\cdot): \mathcal{P} \to \mathcal{X} \\ &P \mapsto \mathcal{M}(P)=\mathcal{G}\left(\mathcal{I}\left(\mathcal{F}(P)\right)\right) \end{aligned} \tag{1}$$

3.1. Traditional Feature Fusion Module

In this part, we fuse the point coordinates with handcrafted features and the learned features for prior information enhancement (green-dashed block in Figure 2).

The extracted handcrafted features including range value, density feature and normal description are shown in Figure 3.

- **Range value** $R_A=\sqrt{x_A^2+y_A^2+z_A^2}$ has the capability to record the relative distance between the target point $A=\left(x_A, y_A, z_A\right)$ and the original of coordinates.
- **Density value** can indicate some local distribution information of each point, and can be formulated by $D_B=\dfrac{1}{\frac{1}{|\mathcal{N}|}\sum_{B_i \in \mathcal{N}} d_{B_i}}$, where $\mathcal{N}$ represents the

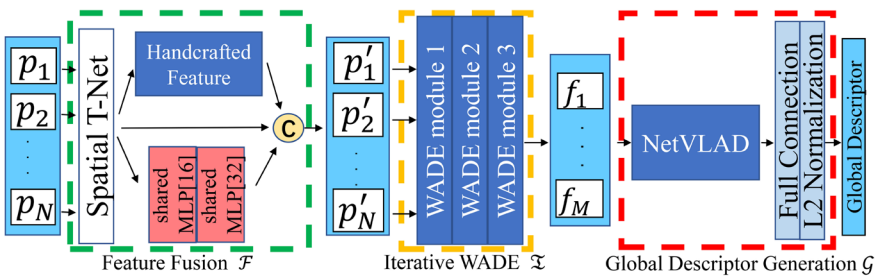


Figure 2. Network architecture of the proposed method.

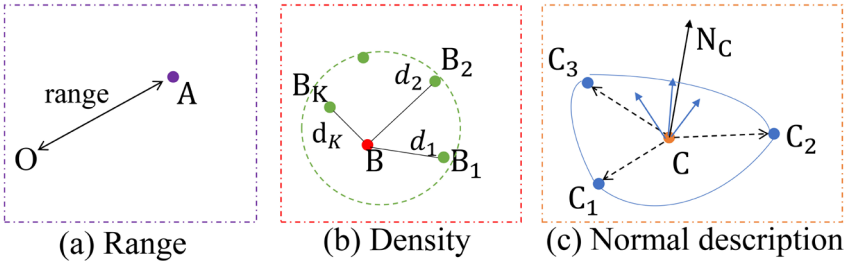


Figure 3. Traditional handcrafted feature extraction. It is composed of range, density and normal vector.

set of neighbor points of the target point, and d_{B_i} is the Euclidean distance between neighborhood B_i and the target point B (the red point). The nearer neighbor points are, the larger density value is.

- **Normal vector** N_C is depicted in **Figure 3(c)**, which can be approximated via $N_C = \sum_{C_j \in \mathcal{N}} \left(\overrightarrow{CC_j} \times \overrightarrow{CC_{j+1}} \right)$, where $\times$ is the cross product of vector, and C_j is the neighboring point.

We concatenate these three handcrafted features to get the local prior features with size $N \times 5$ in module $\mathcal{F}$, which is different from the local feature extraction block of LPD-Net [12]. The cross contrast experiments of the two handcrafted features are shown in Section 4.

Simultaneously, we use a two-layer MultiLayer Perceptron (MLP) [52] to extract learned point-wise features. After concatenating the point coordinate with traditional features and the learned features, we get the high dimension features $P' = \{p_i'\}_{i=1}^{N}$. This feature fusion block makes good use of both latent and structure features. However, due to the non-uniform distribution in a point cloud, the significance of the local structure of different points may be different. We need some adaptive sampling and weighting during feature integration.

3.2. WADE Module

In the iterative WADE module $\mathcal{I}$, we further consider the weighted density distribution adaptively for feature extraction and aggregation. **Figure 4**, the following Sampling and Grouping (SG) operation and Feature Encoding steps describe the one WADE module.

1) Sampling and Grouping (SG) Operation. Assuming that the inputs of the WADE module are a point cloud P^{in} with size $N^{in} \times 3$ and its corresponding features F^{in} with size $N^{in} \times C^{in}$. Notice that the input features of the first WADE module is P'.

By utilizing Farthest Point Sampling (FPS) [53] we get a sampling subset $P^S = \{p_1, p_2, \cdots, p_{N^S}\} \subset P^{in}$ with size $N^S \times 3$, as shown in **Figure 5**.

Then, for each sampled point $p_s \in P^S$, the ball query is used to find its $\bar{K}$ neighbor points $\{p_s^1, p_s^2, \cdots, p_s^{\bar{K}}\}$ from the input point set P^{in}. For observing their relative spatial locations and local geometric patterns, the x-y-z coordinates and features of all neighbouring points are grouped as $p_s^G = \{p_s^1, p_s^2, \cdots, p_s^{\bar{K}}\}$

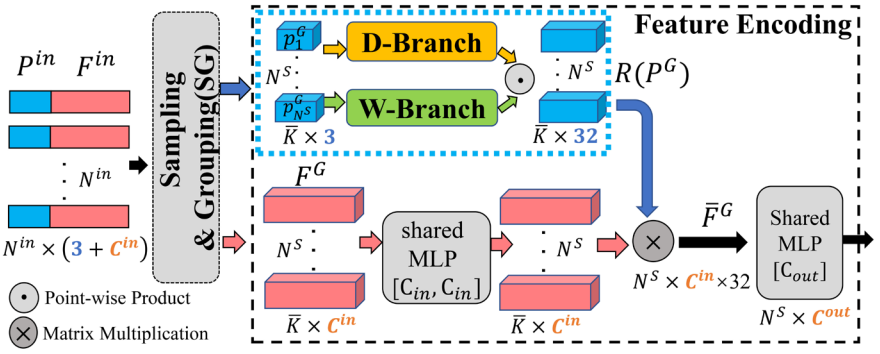


Figure 4. Flow chart of one WADE module. N^S means the number of sampled points. $\bar{K}$ represents the neighbor number for grouping. C_{in} and C_{out} are dimensions of input and output features.

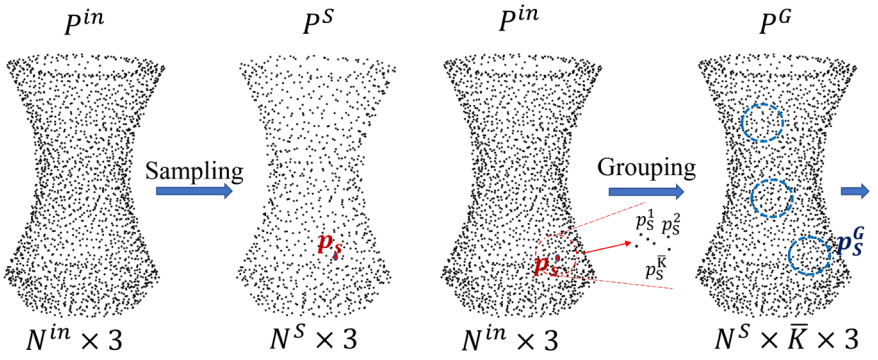


Figure 5. Schematic diagram of Sampling and Grouping (SG). P^S is the sampled points set, and P^G is the grouped 3D point set.

and $f_s^G = \left\{ f_s^1, f_s^2, \cdots, f_s^{\bar{K}} \right\}$ respectively. $G = \left\{1, 2, \cdots, \bar{K}\right\}$ is the index set of neighbourhood point. So, the outputs of SG operation include the grouped 3D point set $P^G = \left\{ p_s^G \right\}_{s=1}^{N^S}$ with size $N^S \times \bar{K} \times 3$ and its corresponding grouped features $F^G = \left\{ f_s^G \right\}_{s=1}^{N^S}$ with size $N^S \times \bar{K} \times C^{in}$.

2) Feature Encoding.

To aggregate the feature concerning density and relative spatial information, the grouped point set P^G and its corresponding grouped features F^G are put into the following three branches: D-branch, W-Branch and feature aggregation, as shown in Figure 4.

D-Branch. This branch is about the generation of density factor, since it can represent the important structure of distribution and is proportional to the significance of the sampled point. In Figure 6, for each sampled point $p_s \in P^S$, the Gaussian kernel function is used to calculate the density from the raw point cloud P, which is formulated by

$$d\left(p_s\right) = \frac{1}{\left|\mathcal{N}\left(p_s\right)\right| \cdot h^3 \cdot \sqrt{2\pi}} \sum_{p_s^k \in \mathcal{N}(p_s)} \exp\left(-\frac{\left(p_s - p_s^k\right)^2}{2h^2} \right), \tag{2}$$

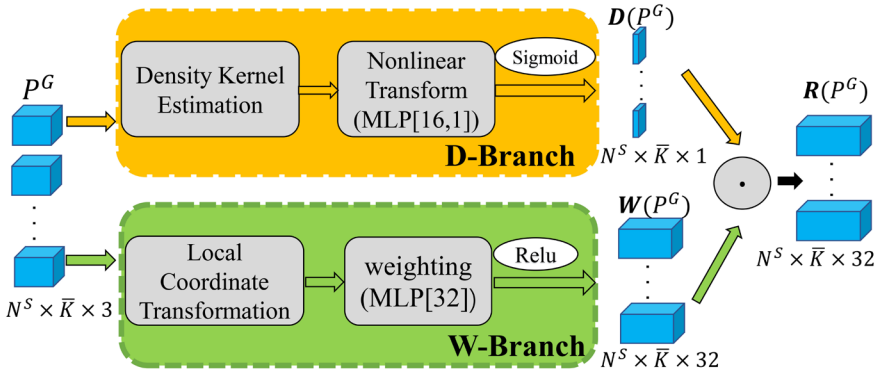


Figure 6. Detailed description of D-Branch and W-Branch.

where h is the bandwidth of the kernel function, and $p_s^k \in \mathcal{N}(p_s) \subset P$. Then we normalize the estimated density by $d(p_s) := d(p_s) \Big/ \max\left\{d\left(p_s^k\right)\right\}_{k=1}^{\bar{K}}$.

After the MLP encoding, we choose *sigmoid* as the nonlinear activation function of last layer to compute the density value $D\left(p_s^G\right)$ for each grouped point p_s^G. The reason for choosing *sigmoid* is that the generated density factor should be close to the binary choice mechanism. Moreover, the reason to use the nonlinear transform is for deciding adaptively whether to use the density value. Mathematically, The $D\left(p_s^G\right)$ is formulated as

$$D\left(p_s^G\right) = Sigmoid\left(MLP\left(d\left(p_s^1\right) \oplus \cdots \oplus d\left(p_s^{\bar{K}}\right)\right)\right),$$

where $\oplus$ is the concatenation operation. So, we obtain the density factor $D\left(P^G\right)$ with the size $N^S \times \bar{K} \times 1$ for the grouped 3D point set P^G.

W-Branch. Considering that the relative spatial relationships can reflect contributions of one point to the surroundings structure, we learn a position relation of grouped points. In Figure 6, for every grouped points p_s^G, we transform neighbor grouped $\bar{K}$ points $\left\{p_s^1, p_s^2, \cdots, p_s^{\bar{K}}\right\}$ into the local coordinate system of p_s, to get relative local coordinates $\left\{p_s - p_s^1, p_s - p_s^2, \cdots, p_s - p_s^{\bar{K}}\right\}$. Moreover, weights $W\left(p_s^G\right)$ are generated by MLP with *Relu* activation, which is the same as other MLPs of the proposed network. The reason for choosing *Relu* is that it helps stabilize the output of network, and it screens parameters of the network. Mathematically, for the $\bar{K}$ neighbor points $\left\{p_s^1, \cdots, p_s^{\bar{K}}\right\}$, we compute the $W\left(p_s^G\right)$ as

$$W\left(p_s^G\right) = Relu\left(MLP\left(p_s^1 - p_s \oplus \cdots \oplus p_s^{\bar{K}} - p_s\right)\right),$$

where $\oplus$ is the concatenation operation. Finally, we can obtain the relative spatial factor $W\left(P^G\right) = \oplus_s W\left(p_s^G\right)$ with size $N^S \times \bar{K} \times 32$, where $\oplus_s$ is the concatenation operation for all $s = 1, \cdots, N^s$.

Feature Aggregation. We propose a weighted density based feature aggregation in this block, as shown in the black dashed box of Figure 4. Before feature aggregation, a weighted density ratio $R\left(P^G\right)$ is acquired by integrating the density factor $D\left(P^G\right)$ and spatial relation factor $W\left(P^G\right)$ via point-wise

product

$$R\left(P^G\right)=D\left(P^G\right)\odot W\left(P^G\right), \tag{3}$$

where $\odot$ is the point-wise product in each group, and the size of $R\left(P^G\right)$ is $N^S\times\overline{K}\times 32$. The details are depicted in Figure 6.

In the beginning of feature encoding, the grouped feature points F^G is fed into the shared MLP to obtain the point-wise feature extraction $MLP\left(F^G\right)$ with size $N^S\times\overline{K}\times C^{in}$, as shown in Figure 4.

Then, we conduct the feature aggregation via a matrix multiplication between the weighted density ratio $R\left(P^G\right)$ and features $\overline{F}^G$ as follows

$$\overline{F}^G=R\left(P^G\right)\otimes MLP\left(F^G\right), \tag{4}$$

where $\otimes$ is matrix multiplication. Furthermore, the output of feature encoding $MLP\left(\overline{F}^G\right)$ is generated by one MLP with C^{out} output channels.

To get an efficient structure constrained local feature aggregation, we conduct the aforementioned WADE module iteratively. Finally, we obtain the output from the iterative WADE module $\mathbf{F}=\left\{f_1,\cdots,f_M\mid f_i\in\mathbb{R}^D\right\}_{i=1}^M$ with size $M\times D$.

3.3. Global Descriptor and Metric Learning

Applying NetVLAD block [39], we aggregate the local features into a discriminative global descriptor for each point cloud. The NetVLAD block will learn K_c cluster centers $\left\{\boldsymbol{c}_1,\cdots,\boldsymbol{c}_j,\cdots\mid\boldsymbol{c}_j\in\mathbb{R}^D\right\}_{j=1}^{K_c}$ and get their correlations $\left\{\boldsymbol{w}_j,b_j\right\}, j=1,\cdots,K_c$ to feature points $\left\{\boldsymbol{f}_i,i=1,\cdots,M\right\}$ by softmax operation. The representation of each cluster center $\boldsymbol{c}_j$ can be expressed as the following feature vector

$$\boldsymbol{V}_j\left(\mathbf{F}\right)=\sum_{i=1}^{M}\frac{\mathrm{e}^{\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{f}_i+b_j}}{\sum_j\mathrm{e}^{\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{f}_i+b_j}}\left(\boldsymbol{f}_i-\boldsymbol{c}_j\right),\quad j=1,\cdots,K_c \tag{5}$$

where $\left\{\boldsymbol{w}_j\right\}$ and $\left\{b_j\right\}$ are learned weights and biases that determine the contribution of the correlation between feature points and each cluster center $\boldsymbol{c}_j$.

Then we do intra-normalization of each vector $\boldsymbol{V}_j$ firstly, and concatenate them, followed by L_2 normalization to get the global feature vector $\boldsymbol{V}_{global}$ with size $K_c\times D$. At last, a fully connected layer is utilized to reduce the dimension of global descriptor to 512, and then L_2 normalization guarantees that the learned global descriptor $\boldsymbol{x}$ is unit length.

The learned global descriptor $\boldsymbol{x}$ represents one point cloud in the descriptor space. For recognition assignment, the similarity or dissimilarity of two point clouds should be considered, and the metric constraint is chosen. The metric constraint can balance the similarity between intra-class and inter-class via triplet constraint. The process and purpose of metric constraint can be explained in Figure 7. Each mark means the global descriptor of a specific place. Blue disks

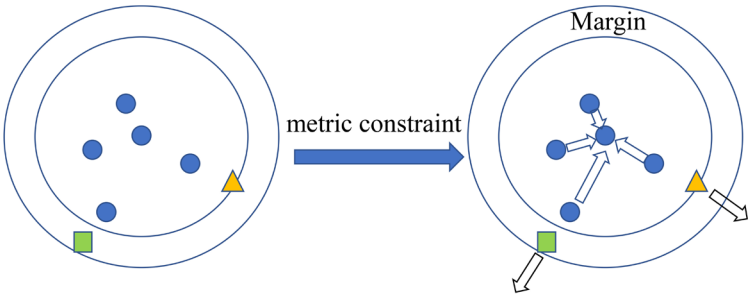


Figure 7. A simple example showing the goal of metric constraint of the whole framework. Each mark means the global descriptor of a specific place. The blue disc shape belong to one place, and the green square and yellow triangle respectively represent another place.

are similar descriptors, and the green square and yellow triangle are both dissimilar with blue disks. Intuitively, a proper metric constraint should make blue disks closer and maintain a margin between blue disk and other categories. Therefore, a more discriminative metric constraint pushes similar descriptors closer, and away from dissimilar ones.

Generally, a set of triplet tuples from the training dataset is obtained with supervised position information (GPS). We introduce a traditional triplet constraint [54] [55] in an intuitive way, denoted:

$$\mathcal{T} = \left\{ \left(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k \right) : \delta_{ij} \le \delta_{ik} \right\}, \tag{6}$$

where $\boldsymbol{x}_i$ is similar to $\boldsymbol{x}_j$, and dissimilar to $\boldsymbol{x}_k$. δ_{ij} denotes the distance between similar samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and δ_{ik} denotes the distance between dissimilar samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$. It means that the distance of samples representing different places should be as large as possible, while as small as possible in the same place. Traditionally, hinge loss function [56] mainly be used for the triplet constraint

$$\mathcal{L}_H = \min \sum_{\mathcal{T}} \left[\alpha + \delta_{ij} - \delta_{ik} \right]_+, \tag{7}$$

where $\left[\cdot \right]_+$ is the hinge loss, which means $\left[m \right]_+ = m$ if $m \ge 0$ and $\left[m \right]_+ = 0$ otherwise. α is the margin value to clear the edge between similar ones and dissimilar ones.

Moreover, due to the complexity of outdoor environment, we introduce Lazy Triplet Loss [10] to augment the relation of similarity and dissimilarity. Mathematically, the Lazy Triplet constraint is calculated as

$$\mathcal{L}_{Lazy}\left(\mathcal{T} \right) = \min \left(\left[\alpha + \max \delta_{pos} - \min \delta_{neg} \right]_+ \right), \tag{8}$$

where $\max \delta_{pos}$ means the maximum in δ_{pos}, $\min \delta_{neg}$ means the minimum in δ_{neg}, δ_{pos} is the Euclidean distance between descriptor of query (current descriptor) and that of similar place, and δ_{neg} means the distance between current and dissimilar one. This loss can learn a more discriminative and robust mapping in order to optimize parameters in the network.

4. Experiments

In this section, we evaluate the proposed approach with some traditional and deep learning based methods as follows: VFH [24], ESF [22], Scan-Context (SC) [5], M2DP [28], PNMAX, PNSTD [36], PNVLAD [10], PCAN [11] and LPD-Net [12]. The computer is equipped with an NVIDIA GTX 1080Ti GPU with 11GB memory, Xeon(R) CPU E5-2650-v4, 64 RAM. The code is implemented with Ubuntu 16.04 operating system with TensorFlow. For the super parameters of our network, we set batch size = 2, $n_{pos}=2$, $n_{neg}=18$, margin $\alpha=0.5$ and maximum iteration = 15, which follow the comparison methods. The cluster center numbers of VLAD, K_c is set to 32, and the dimension of the output global descriptor D is set to 512. All the comparison experiments is tested in the same machine, and the input point cloud number is set to 1024 uniformly, so the fairness is guaranteed.

4.1. Benchmark Datasets

The comparison experiment conducts on two public outdoor large-scale datasets Oxford Robotcar dataset [40] and KITTI dataset [41]. The processing of these two datasets is described as follows and showed in **Figure 8**.

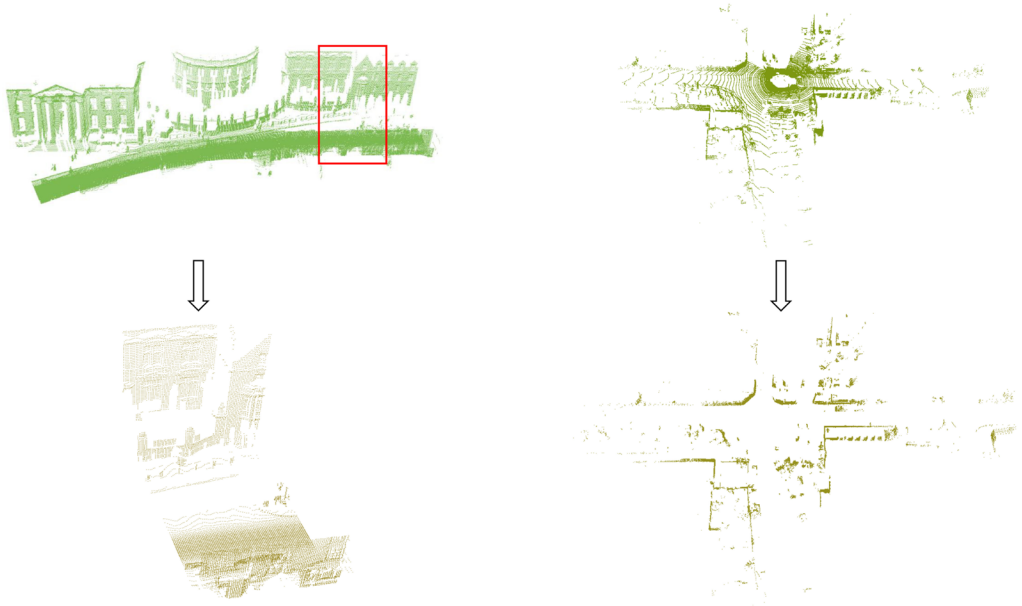
Oxford Robotcar dataset [40]: It includes 21,711 3D point cloud submaps made up of point clouds within the car's 20 m (meters) trajectory for training, and 3030 submaps for testing. The data were collected in different seasons, times and weathers. For evaluation details, each submap is tagged with a Universal Transverse Mercartor (UTM) coordinate. Point clouds are defined as positive pairs if they are at most 10 m apart and negative pairs if they are at least 50 m apart. In the evaluation process, the retrieved point cloud is regarded as a correct match if the distance is within 25 m between the retrieved point cloud and the query point cloud.

KITTI dataset [41]: It captures real-world traffic situations and ranges from freeways over rural areas to urban scenes with many static and dynamic objects. We choose 11 scenes named KITTI 00 to KITTI 10 for training and testing, since they supply accurate odometry ground truth information. For each scene, we utilize the reduplicative frames of places that are passed more than twice as testing samples, and other frames as training. Limitation of positive pairs and negative pairs are set to 5 m and 50 m respectively. On evaluation stage, the relative distance of correct match is 5 m and we choose 4 scenarios primarily used by researchers for evaluation. The ground points are removed using the method in [57].

4.2. Evaluation Results

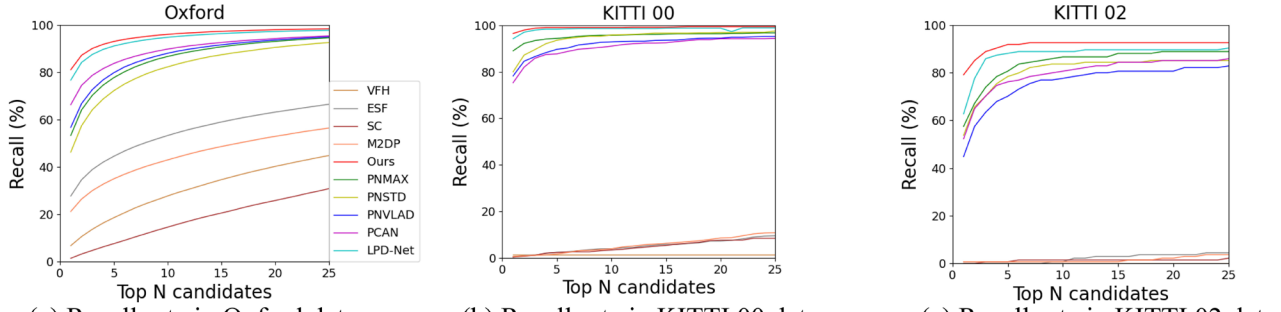
The evaluation results are given in **Figure 9** and **Table 1**. TOP 1 (@1) represents that the similar place of current frame is recognized the first time among candidate places. TOP 1% (@1%) means that the correct area is retrieved within 1% frame number of current scene.

Figure 9 shows that the proposed approach performs better than other networks in different datasets. The evaluation curve generated by our approach is, on the whole, numerically higher than these comparison ones. In KITTI 06 an 07, the advantage of the proposed method cannot be reflected fully because of simplicity of this scene.

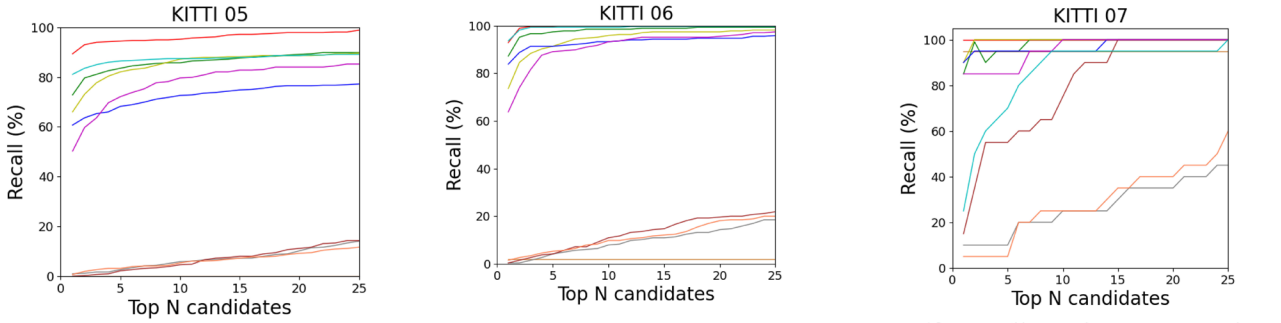


(a) Oxford Dataset (b) KITTI Dataset

Figure 8. Datasets visualization and pre-processing.



(a) Recall rate in Oxford data (b) Recall rate in KITTI 00 data (c) Recall rate in KITTI 02 data



(d) Recall rate in KITTI 05 data (e) Recall rate in KITTI 06 data (f) Recall rate in KITTI 07 data

Figure 9. Comparison of the recall rate for different methods. (a) Oxford dataset. (b)-(f) different scenes in KITTI dataset.

Table 1. Comparison of retrieval recall rate (%) at TOP 1(@1) and TOP1% (@1%) among different networks.

	Recall Rate @1/@1%					
	Oxford	KITTI 00	KITTI 02	KITTI 05	KITTI 06	KITTI 07
VFH [24]	6.79/16.19	1.33/1.33	0.0/0.0	0.0/0.0	1.89/1.89	95.00/95.00
ESF [22]	0.40/41.86	0.0/16.33	79.10/8.21	0.97/14.56	0.38/8.30	10.00/25.00
Scan-Context [5]	1.45/6.21	0.40/15.01	0.0/5.22	0.0/15.05	0.38/21.89	15.00/85.00
M2DP [28]	21.21/32.56	0.66/16.20	0.75/7.46	0.73/12.14	1.51/20.00	5.00/25.00
PNMAX [36]	53.28/74.27	88.98/97.34	57.46/88.81	72.82/90.53	87.17/98.49	85.00/100.0
PNSTD [36]	46.24/68.39	79.95/98.27	53.73/85.82	66.02/89.81	73.58/96.23	90.00/100.0
PNVLAD [10]	56.69/76.47	78.20/96.40	44.80/84.30	60.70/78.20	83.80/93.60	90.00/95.00
PCAN [11]	66.26/81.30	75.30/95.09	52.24/88.06	50.24/85.68	63.77/93.58	85.00/100.0
LPD-Net [12]	76.62/89.45	94.16/99.07	62.69/91.04	81.07/89.32	93.58/99.62	90.00/100.00
Ours	**82.22/92.66**	**99.07/99.60**	**87.31/97.76**	**95.39/98.30**	**97.36/100.0**	**100.00**/100.00

Table 1 shows that our approach exceeds most of the comparison ones on the recall rate at least 10% at TOP 1 and TOP 1% on Oxford dataset. Compared with LPD-Net, we have almost 2% - 3% increase in retrieval results at both TOP 1 and TOP 1%. At the comparison experiment in KITTI dataset, our network performs much better than the best of the other comparison methods at TOP 1, which means that it is more possible to recognize the passing place all at once. What is more, at TOP 1%, our network has at least 1% - 2% increase to the best of others. Considering that the TOP 1% candidates number has relation with the frame number of the outdoor scene, there may be little difference in results at TOP 1%.

Additionally, Table 1 shows that the first two traditional methods, VFH and ESF, cannot perform as well as other learning based approaches. Empirically, traditional methods rely on prior knowledge and they may have little ability to view surroundings roundly, especially in outdoor environment. For traditional place loop detection algorithms, e.g., Scan-Context and M2DP, they do not perform well as some other methods. Analytically speaking, the point number of each point cloud affects the performance of these two methods.

4.3. Analysis and Discussion

Iteration number of WADE module. The proposed WADE module is considered as a feature extraction layer, and the proposed WADE-Net iterates it for obtaining multi-scale features. On account of point cloud number difference, parameters of WADE module have different settings in each iteration.

Table 2 shows the settings in the iterative WADE modules. Parameter N denotes the output point number of each iteration in WADE module, r and h denote the radius of ball query in grouping operation of SG operation and the bandwidth of Gaussian kernel function in Equation (2), respectively, and K is the number of neighbor points of $\mathcal{N}$ in Equation (2). As Table 2 shows, r, h and C_{out} increase gradually with the point number decreases during each itera-

tion. It is natural that the query ball radius r should enlarge as the sampled number decrease. The change of h aims to keep balance in terms of the range of density kernel. The C_{out} is related to features from different scales as N decreases.

Moreover, Table 3 illustrates that the WADE-Net can perform the best when the iteration number is set to 3. Obviously, as the iteration number increase, the evaluation recall rates will decrease.

Ablation results of different modules. The effectiveness of different modules used in our network, *i.e.*, 3D point coordinates, traditional features (TF) and iterative WADE module, is described in Table 4. As we can see from rows 1 - 2, PNVLAD + WADE performs much better than the baseline method, having average 9% recall rate increase for most of the data. The rest of rows reflects that taking traditional local feature and point coordinates into consideration is reasonable.

Ablation results of different handcrafted features. Table 5 shows the effectiveness of handcrafted feature extraction strategies used in LPD-Net and the proposed method. If the traditional features is replaced by those in LPD-Net, the recognition results are worse than ours.

Table 2. Parameters of iterative WADE module in the proposed network.

Iteration 1	$N = 1024$, $r = 1.0$, $h = 0.1$, $K = 32$, $C_{in} = 32$, $C_{out} = 64$
Iteration 2	$N = 512$, $r = 2.0$, $h = 0.2$, $K = 32$, $C_{in} = 64$, $C_{out} = 128$
Iteration 3	$N = 256$, $r = 4.0$, $h = 0.4$, $K = 32$, $C_{in} = 128$, $C_{out} = 256$
Iteration 4	$N = 128$, $r = 8.0$, $h = 0.8$, $K = 32$, $C_{in} = 256$, $C_{out} = 512$

Table 3. The comparison results of the iteration number about WADE module.

Iterations (Iter)				Recall Rate @1/@1%					
Iter1	Iter2	Iter3	Iter4	Oxford	KITTI 00	KITTI 02	KITTI 05	KITTI 06	KITTI 07
√	√			81.37/92.35	98.54/99.60	83.58/96.27	94.17/98.79	97.36/100.0	100.0/100.0
√	√	√		**82.22/92.66**	**99.07/99.60**	**87.31/97.76**	**95.39/98.30**	**99.25/100.0**	100.0/100.0
√	√	√	√	81.02/91.69	96.41/99.47	79.10/92.54	88.83/97.09	92.83/100.0	100.0/100.0

Table 4. Ablation results of different modules by retrieval recall rate (%).

Modules				Recall Rate @1/@1%				
PNVLAD	WADE	3D	Traditional Features	Oxford	KITTI 00	KITTI 02	KITTI 05	KITTI 06
√				56.69/76.47	78.20/96.40	44.80/84.30	60.70/78.20	83.8/93.60
√	√			73.76/87.49	93.76/98.80	67.91/92.54	85.92/96.12	88.68/100.0
√	√	√		77.44/89.39	95.62/99.07	79.10/92.54	88.83/97.09	92.83/100.0
√	√		√	74.92/88.18	95.88/99.47	71.64/92.54	89.08/97.82	92.45/100.0
√	√	√	√	**81.02**/91.69	**96.41/99.47**	**79.10/93.82**	**89.32/98.79**	**92.83/100.0**

Table 5. Comparison on the effectiveness of handcrafted traditional features (TF) extraction part between LPD-net and our method.

	Recall Rate @1%					
	Oxford	KITTI 00	KITTI 02	KITTI 05	KITTI 06	KITTI 07
LPD-Net	89.45	99.07	91.04	89.32	99.62	95.00
LPD-Net + our TF	89.94	99.34	91.79	91.26	100.0	100.0
Ours + TF of LPD-Net	90.00	99.07	93.28	97.09	100.0	100.0
Ours	**91.69**	**99.47**	**93.28**	**98.79**	**100.0**	100.0

Ablation studies about hyper-parameters.

Table 6 shows the ablation experiments for the hyper-parameters K_c and D. The table illustrates that if the K_c is set 32, and D is set to 512, the WADE-Net performs better.

Moreover, Equation (8) represents the constraint condition to discriminate the relationship of descriptors in positive pairs and negative pairs. In order to balance the distances of positive pairs and negative pairs, α is put forward and its ablation experiment results are depicted in Figure 10. It shows that it is able to get a moderately better result as $\alpha = 0.5$. In this paper, a strict mechanism is considered to focus on the most dissimilar positive sample and the most similar sample, so the margin value should be intuitively decreased. The ablation experiment testifies this idea.

Time and resources consumption. In Table 7, we list the average inference time and computational resources among the deep learning based methods. The process of inference time represents that the input point cloud in inputted into the network and a global descriptor is generated. Parameters in Table 7 means the learned parameters w and b in network framework. GFLOPs means 1 billion floating-point numbers. The smaller the result value is, the more efficient the approach is.

From Table 7, we can analysis that the parameters and GFLOPs of WADE-Net are smaller than the others. However, it does not perform well at the inference time because the TF and feature fusion module is conducted online and is based CPU.

4.4. Visualization Results

Figure 11 shows the sampled points of each iteration stage, and the first iterative stage (stage 1) has the same point number with the input point cloud. It illustrates that the sampling algorithm can keep the scene structure, and the maintained points can be considered significant or presenting local relation information.

Figure 12 gives the low dimension manifold visualization of place descriptors in the road trajectory from KITTI dataset. Each point of the sub-figure describes the global descriptor, and different colors represent different places. Figure 12 illustrates that the proposed method can generate more discriminative descriptor and retains similar topology structure of the road trajectory.

Table 6. Experiment recall results about cluster center number (K_c) and output dimension of global descriptor (D) in Feature Aggregation Module.

		Recall rate @1	Recall rate @1%
$D = 256$	$K_c = 32$	79.58	91.14
	$K_c = 48$	78.70	91.32
	$K_c = 64$	79.91	91.29
$D = 512$	$K_c = 32$	**82.22**	**92.66**
	$K_c = 48$	79.02	90.86
	$K_c = 64$	79.85	91.33

Table 7. The comparison results of inference time (ms) and computational resources. In the comparison methods, the deep learning based approaches are chosen.

	Recall Rate @1/@1%		
	time (ms)	Parameters	GFLOPs
PNMAX	10.26	3.052M	5.291
PNSTD	10.97	3.306M	5.291
PNVLAD	11.44	10.584M	6.431
PCAN	31.73	11.034M	7.545
LPD-Net	20.29	10.604M	6.859
Ours	23.10	2.009M	2.453

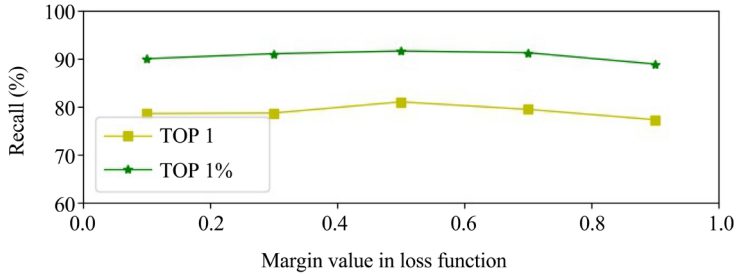
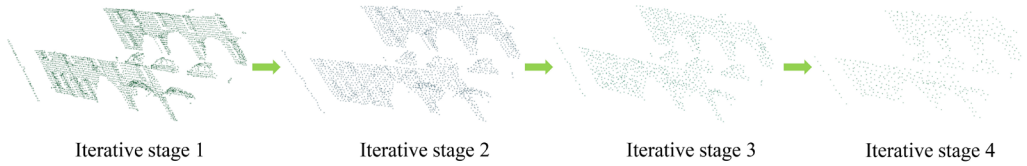


Figure 10. The ablation experiment results in Oxford dataset for the margin value α of loss function.



(a) Sampled points change in one submap of scene in Oxford Dataset

(b) Sampled points change in one frame of scene in KITTI Dataset

Figure 11. Visualization of the extracted sampled points of each iteration in WADE module in one scene. (a) is for one scene in Oxford Dataset. The shown scene id is 2014-11-14-16-34-33 (b) is for one scene in KITTI Dataset. The frame id of shown scene is KITTI-02-900.

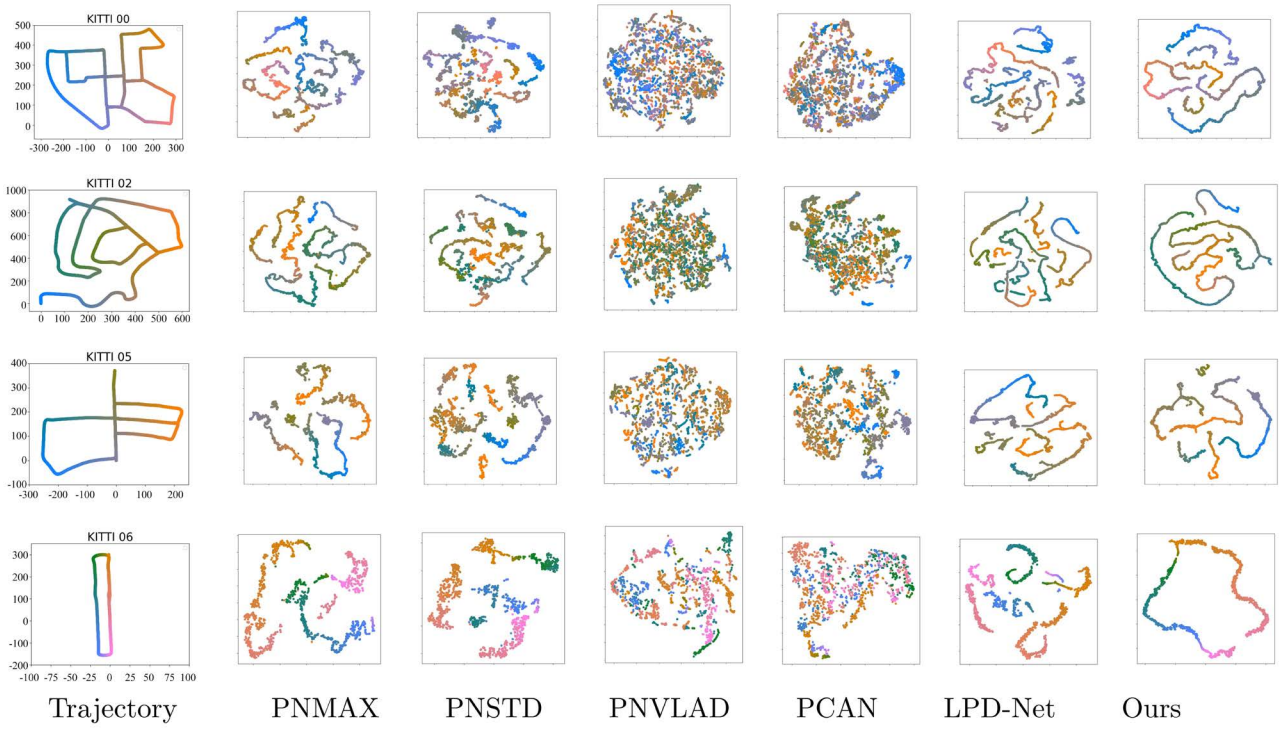


Figure 12. t-SNE visualization of descriptors generated by deep learning methods for KITTI dataset. Every color point in each subfigure represents the global descriptor of one frame. Row 1-4 means that visualization of different methods is conducted on KITTI 00, 02, 05, 06 respectively. Column 1 is the trajectory of each scene, which is colored by the accurate position of frames. Columns 2-7 represent different approaches including ours.

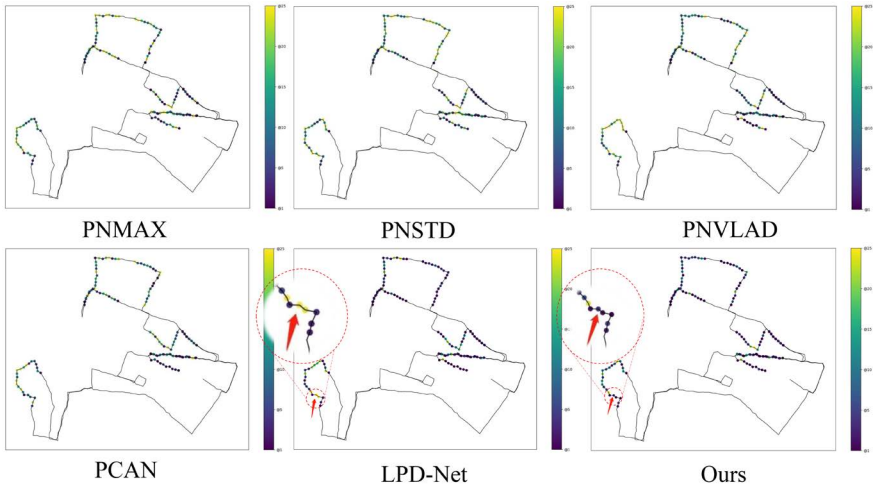


Figure 13. Retrieved trajectory map of comparison methods on Oxford dataset. Each point of the retrieved trajectory is colored. The darker the color is, the better the recognition result is.

Figure 13 depicts the retrieved trajectory map of comparison approaches mentioned in **Figure 12** in Oxford data. Each trajectory point in test region is colored and the brightness of the color corresponds to TOP N candidates number of correct place. For each colored point, the darker the color is, the better the recognition result is. This visualization illustrates that our approach gets a more

accurate result than most of other comparison methods. When compared with LPD-Net, we mark out one local region in LPD-Net and ours. The enlarged regions from **Figure 13** show that our approach outperforms LPD-Net.

5. Conclusion

In this paper, we have proposed new point cloud representation framework via an iterative weighted density aggregation method. It enhances the input prior information for traditional feature fusion module. Then the network extracts the important structure information, including density and spacial relationship, via iterative WADE module. At last, we compare our approach with some off-the-shelf methods on two public datasets with different kinds of outdoor scenes. Experiments and visualization results show that our network has competitiveness and performs better than the others.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 11771276, 12026416 and 11871327, and in part by Shanghai Science and Technology Innovation Action Plan 18441909000. Moreover, this work is supported by shanghai science and technology innovation action plan (18441909000) and the capacity construction project of local universities in Shanghai (No. 18010500600).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Mur-Artal, R., Montiel, J.M.M. and Tardos, J.D. (2015) ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, **31**, 1147-1163. https://doi.org/10.1109/TRO.2015.2463671

[2] Mur-Artal, R. and Tardos, J.D. (2017) ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, **33**, 1255-1262. https://doi.org/10.1109/TRO.2017.2705103

[3] Lin, J. and Zhang, F. (2019) A Fast, Complete, Point Cloud Based Loop Closure for LiDAR Odometry and Mapping.

[4] Liu, Z., Suo, C., Zhou, S., Xu, F., Wei, H., Chen, W., Wang, H., Liang, X. and Liu, Y. (2019) SeqLPD: Sequence Matching Enhanced Loop-Closure Detection Based on Large-Scale Point Cloud Description for Self-Driving Vehicles. 2019 *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), Macau, 4-8 November 2019, 1218-1223. https://doi.org/10.1109/IROS40897.2019.8967875

[5] Kim, G. and Kim, A. (2018) Scan Context: Egocentric Spatial Descriptor for Place Recognition within 3D Point Cloud Map. 2018 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, 1-5 October 2018, 4802-4809. https://doi.org/10.1109/IROS.2018.8593953

[6] Qiao, Y., Cappelle, C., Ruichek, Y. and Yang, T. (2019) ConvNet and LSH-Based

Visual Localization Using Localized Sequence Matching. *Sensors*, **19**, 2439. https://doi.org/10.3390/s19112439

[7] Lowry, S., Sunderhauf, N., Newman, P., Leonard, J.J., Cox, D., Corke, P. and Milford, M. (2015) Visual Place Recognition: A Survey. *IEEE Transactions on Robotics*, **32**, 1-19. https://doi.org/10.1109/TRO.2015.2496823

[8] Yu, J., Zhu, C., Zhang, J., Huang, Q. and Tao, D. (2019) Spatial Pyramid-Enhanced NetVLAD with Weighted Triplet Loss for Place Recognition. *IEEE Transactions on Neural Networks and Learning Systems*, **31**, 661-674. https://doi.org/10.1109/TNNLS.2019.2908982

[9] Chen, L., Jin, S. and Xia, Z. (2021) Towards a Robust Visual Place Recognition in Large-Scale vSLAM Scenarios Based on a Deep Distance Learning. *Sensors*, **21**, 310. https://doi.org/10.3390/s21010310

[10] Angelina, U.M. and Hee, L.G. (2018) PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4470-4479.

[11] Zhang, W. and Xiao, C. (2019) PCAN: 3D Attention Map Learning Using Contextual Information for Point Cloud Based Retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, 15-20 June 2019, 12436-12445. https://doi.org/10.1109/CVPR.2019.01272

[12] Liu, Z., Zhou, S., Suo, C., Yin, P., Chen, H., Wang, W., Li, H. and Liu, Y. (2019) LPD-Net: 3D Point Cloud Learning for Large-Scale Place Recognition and Environment Analysis. *Proceedings of the IEEE International Conference on Computer Vision*, Seoul, 27-28 October 2019, 2831-2840. https://doi.org/10.1109/ICCV.2019.00292

[13] Ao, S., Guo, Y., Gu, S., Tian, J. and Li, D. (2020) SGHs for 3D Local Surface Description. *IET Computer Vision*, **14**, 154-161. https://doi.org/10.1049/iet-cvi.2019.0601

[14] Yan, L., Liu, K., Belyaev, E. and Duan, M. (2020) RTL3D: Real-Time LIDAR-Based 3D Object Detection with Sparse CNN. *IET Computer Vision*, **14**, 224-232. https://doi.org/10.1049/iet-cvi.2019.0508

[15] Cao, H., Zhan, R., Ma, Y., Ma, C. and Zhang, J. (2020) LFNet: Local Rotation Invariant Coordinates Frame for Robust Point Cloud Analysis. *IEEE Signal Processing Letters*, **28**, 209-213. https://doi.org/10.1109/LSP.2020.3048605

[16] Cui, W., Liu, J., Du, S., Liu, Y., Wan, T., Han, M., Mou, Q., Yang, J. and Guo, Y. (2020) Individual Retrieval Based on Oral Cavity Point Cloud Data and Correntropy-Based Registration Algorithm. *IET Image Processing*, **14**, 2675-2681. https://doi.org/10.1049/iet-ipr.2019.1420

[17] Frome, A., Huber, D., Kolluri, R., Bulow, T. and Malik, J. (2004) Recognizing Objects in Range Data Using Regional Point Descriptors. In: *European Conference on Computer Vision*, Springer, Berlin, 224-237. https://doi.org/10.1007/978-3-540-24672-5_18

[18] Scovanner, P., Ali, S. and Shah, M. (2007) A 3D SIFT Descriptor and Its Application to Action Recognition. In: *Proceedings of the* 15*th ACM International Conference on Multimedia*, ACM, New York, 357-360. https://doi.org/10.1145/1291233.1291311

[19] Salti, S., Tombari, F. and Di, S.L. (2014) SHOT: Unique Signatures of Histograms for Surface and Texture Description. *Computer Vision and Image Understanding*, **125**, 251-264. https://doi.org/10.1016/j.cviu.2014.04.011

[20] Rusu, R.B., Blodow, N., Marton, Z.C. and Beetz, M. (2008) Aligning Point Cloud Views Using Persistent Feature Histograms. 2008 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, 22-26 September 2008, 3384-3391. https://doi.org/10.1109/IROS.2008.4650967

[21] Rusu, R.B., Blodow, N. and Beetz, M. (2009) Fast Point Feature Histograms (FPFH) for 3D Registration. 2009 *IEEE International Conference on Robotics and Automation*, Kobe, 12-17 May 2009, 3212-3217. https://doi.org/10.1109/ROBOT.2009.5152473

[22] Wohlkinger, W. and Vincze, M. (2011) Ensemble of Shape Functions for 3D Object Classification. 2011 *IEEE International Conference on Robotics and Biomimetics*, Karon Beach, 7-11 December 2011, 2987-2992. https://doi.org/10.1109/ROBIO.2011.6181760

[23] Steder, B., Rusu, R.B., Konolige, K. and Burgard, W. (2010) NARF: 3D Range Image Features for Object Recognition. *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), Volume 44.

[24] Rusu, R.B., Bradski, G., Thibaux, R. and Hsu, J. (2010) Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. 2010 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, 18-22 October 2010, 2155-2162. https://doi.org/10.1109/IROS.2010.5651280

[25] Wang, Y., Sun, Z., Xu, C., Sarma, S., Yang, J. and Kong, H. (2019) LiDAR Iris for Loop-Closure Detection. https://doi.org/10.1109/IROS45743.2020.9341010

[26] Cop, K.-P., Borges, P.-V. and Dube, R. (2018) Delight: An Efficient Descriptor for Global Localisation Using LiDAR Intensities. 2018 *IEEE International Conference on Robotics and Automation* (*ICRA*), Brisbane, 21-25 May 2018, 3653-3660. https://doi.org/10.1109/ICRA.2018.8460940

[27] Johnson, A.E. (1997) Spin-Images: A Representation for 3D Surface Matching.

[28] He, L., Wang, X. and Zhang, H. (2016) M2DP: A Novel 3D Point Cloud Descriptor and Its Application in Loop Closure Detection. 2016 *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), Daejeon, 9-14 October 2016, 231-237. https://doi.org/10.1109/IROS.2016.7759060

[29] Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S. and Su, H. (2015) ShapeNet: An Information-Rich 3D Model Repository.

[30] Engelcke, M., Rao, D., Wang, D.Z., Tong, C.H. and Posner, I. (2017) Vote3deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks. 2017 *IEEE International Conference on Robotics and Automation* (*ICRA*), Singapore, 29 May-3 June 2017, 1355-1361. https://doi.org/10.1109/ICRA.2017.7989161

[31] Zhou, Y. and Tuzel, O. (2018) VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4490-4499. https://doi.org/10.1109/CVPR.2018.00472

[32] Wu, J., Zhang, C., Xue, T., Freeman, B. and Tenenbaum, J. (2016) Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In: *Advances in Neural Information Processing Systems*, MIT, Cambridge, 82-90.

[33] Qi, C.R., Su, H., Niener, M., Dai, A., Yan, M. and Guibas, L.J. (2016) Volumetric and Multi-View CNNs for Object Classification on 3D Data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30

June 2016, 5648-5656.

[34] Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E. (2015) Multi-View Convolutional Neural Networks for 3D Shape Recognition. *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 945-953. https://doi.org/10.1109/ICCV.2015.114

[35] Feng, Y., Zhang, Z., Zhao, X., Ji, R. and Gao, Y. (2018) GVCNN: Group-View Convolutional Neural Networks for 3D Shape Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 264-272. https://doi.org/10.1109/CVPR.2018.00035

[36] Qi, C.R., Su, H., Mo, K. and Guibas, L.J. (2017) PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 652-660.

[37] Qi, C.R., Yi, L., Su, H. and Guibas, L.J. (2017) PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: *Advances in Neural Information Processing Systems*, MIT, Cambridge, 5099-5108.

[38] Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L. and Tai, C.-L. (2020) D3feat: Joint Learning of Dense Detection and Description of 3D Local Features. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, 13-19 June 2020, 6359-6367. https://doi.org/10.1109/CVPR42600.2020.00639

[39] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T. and Sivic, J. (2016) NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 5297-5307. https://doi.org/10.1109/CVPR.2016.572

[40] Maddern, W., Pascoe, G., Linegar, C. and Newman, P. (2017) 1 Year, 1000 km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research*, **36**, 3-15. https://doi.org/10.1177/0278364916679498

[41] Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. (2013) Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, **32**, 1231-1237. https://doi.org/10.1177/0278364913491297

[42] Yan, F., Wang, J., He, G., Chang, H. and Zhuang, Y. (2020) Sparse Semantic Map Building and Relocalization for UGV Using 3D Point Clouds in Outdoor Environments. *Neurocomputing*, **400**, 333-342. https://doi.org/10.1016/j.neucom.2020.02.103

[43] Li, Y., Bu, R., Sun, M., Wu, W., Di, X. and Chen, B. (2018) PointCNN: Convolution on X-Transformed Points. In: *Advances in Neural Information Processing Systems*, MIT, Cambridge, 820-830.

[44] Qi, C.R., Liu, W., Wu, C., Su, H. and Leonidas, G.J. (2018) Frustum Pointnets for 3D Object Detection from RGB-D Data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 918-927.

[45] Li, J., Chen, B.M. and Lee, G.H. (2018) So-Net: Self-Organizing Network for Point Cloud Analysis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 9397-9406. https://doi.org/10.1109/CVPR.2018.00979

[46] Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.H. and Kautz, J. (2018) Splatnet: Sparse Lattice Networks for Point Cloud Processing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 2530-2539. https://doi.org/10.1109/CVPR.2018.00268

[47] Suo, C., Liu, Z., Mo, L. and Liu, Y. (2020) Lpd-ae: Latent Space Representation of

Large-Scale 3D Point Cloud. *IEEE Access*, **8**, 108402-108417. https://doi.org/10.1109/ACCESS.2020.2999727

[48] Komorowski, J. (2021) Minkloc3d: Point Cloud Based Large-Scale Place Recognition. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1790-1799.

[49] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017) Feature Pyramid Networks for Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 2117-2125. https://doi.org/10.1109/CVPR.2017.106

[50] Radenovic, F., Tolias, G. and Chum, O. (2018) Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **41**, 1655-1668. https://doi.org/10.1109/TPAMI.2018.2846566

[51] Vidanapathirana, K., Moghadam, P., Harwood, B., Zhao, M., Sridharan, S. and Fookes, C. (2020) Locus: Lidar-Based Place Recognition Using Spatiotemporal Higher-Order Pooling.

[52] Rosenblatt, F. (1957) The Perceptron, a Perceiving and Recognizing Automaton Project Para. Cornell Aeronautical Laboratory.

[53] Moenning, C. and Dodgson, N.A. (2003) Fast Marching Farthest Point Sampling. Technical Report, University of Cambridge, Computer Laboratory, Cambridge.

[54] Weinberger, K.-Q., Blitzer, J. and Saul, L.-K. (2006) Distance Metric Learning for Large Margin nearest Neighbor Classification. In: *Advances in Neural Information Processing Systems*, MIT, Cambridge, 1473-1480.

[55] Ying, S., Wen, Z., Shi, Y., Peng, Y., Peng, J. and Qiao, H. (2018) Manifold Preserving: An Intrinsic Approach for Semi-Supervised Distance Metric Learning. *IEEE Transactions on Neural Networks and Learning Systems*, **29**, 2731-2742.

[56] Hoffer, E. and Ailon, N. (2015) Deep Metric Learning Using Triplet Network. In: *International Workshop on Similarity-Based Pattern Recognition*, Springer, Berlin, 84-92. https://doi.org/10.1007/978-3-319-24261-3_7

[57] Himmelsbach, M., Hundelshausen, F.V. and Wuensche, H.-J. (2010) Fast Segmentation of 3D Point Clouds for Ground Vehicles. 2010 *IEEE Intelligent Vehicles Symposium*, La Jolla, 21-24 June 2010, 560-565. https://doi.org/10.1109/IVS.2010.5548059