BEVPlace: Learning LiDAR-based Place Recognition using Bird's Eye View Images

Lun Luo[1,2,4], Shuhang Zheng[2], Yixuan Li[2], Yongzhi Fan[2], Beinan Yu[2], Si-Yuan Cao[1,2*],
Junwei Li[1,2], Hui-Liang Shen[1,2,3*]

[1]Ningbo Innovation Center, Zhejiang University
[2]College of Information Science and Electronic Engineering, Zhejiang University
[3]Key Laboratory of Collaborative Sensing and Autonomous Unmanned Systems of Zhejiang Province
[4]HAOMO.AI Technology Co., Ltd.

{luolun, zhengsh, yixuanli, tony_fan, yubeinan, cao_siyuan, lijunwei7788, shenhl}@zju.edu.cn

Abstract

Place recognition is a key module for long-term SLAM systems. Current LiDAR-based place recognition methods usually use representations of point clouds such as unordered points or range images. These methods achieve high recall rates of retrieval, but their performance may degrade in the case of view variation or scene changes. In this work, we explore the potential of a different representation in place recognition, i.e. bird's eye view (BEV) images. We validate that, in scenes of slight viewpoint changes, a simple NetVLAD network trained on BEV images achieves comparable performance to the state-of-the-art place recognition methods. For robustness to view variations, we propose a rotation-invariant network called BEVPlace. We use group convolution to extract rotation-equivariant local features from the images and NetVLAD for global feature aggregation. In addition, we observe that the distance between BEV features is correlated with the geometry distance of point clouds. Based on the observation, we develop a method to estimate the position of the query cloud, extending the usage of place recognition. The experiments conducted on large-scale public datasets show that our method 1) achieves state-of-the-art performance in terms of recall rates, 2) is robust to view changes, 3) shows strong generalization ability, and 4) can estimate the positions of query point clouds. Source codes are publicly available at https://github.com/zjuluolun/BEVPlace .

1. Introduction

Place recognition plays an important role in both the map construction and localization phases of long-term Simulta-
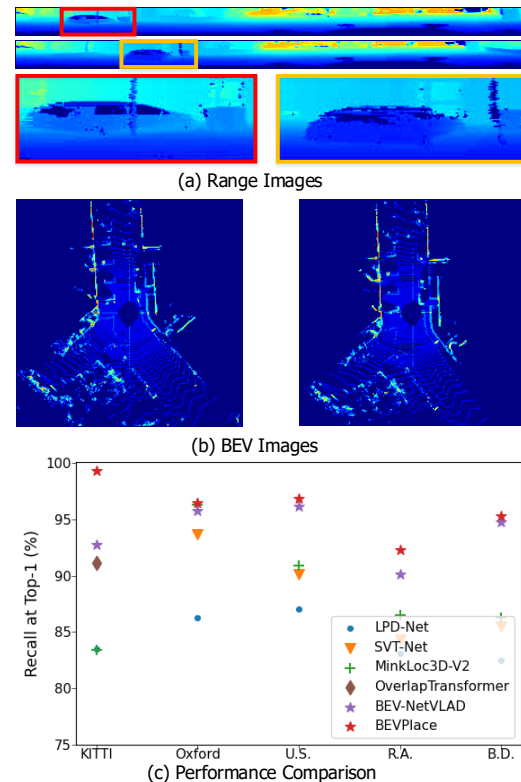

Figure 1. (a) Two range images projected from two point clouds of KITTI that are 5 meters apart from each other. A small point cloud translation will cause distortions such as scale changes and occlusion. (b) The corresponding BEV images. The scale and position distribution of objects on the road almost remains unchanged. (c) Performance on various datasets. A simple BEV-based NetVALD network achieves comparable Top-1 recall to the SOTA methods. Our BEVPlace enhances the baseline further.

neous Localization and Mapping (SLAM) systems [3]. In the map construction phase, it can provide loop closure constraints to eliminate the accumulated drift of the odometry.

*Corresponding author.

In the localization phase, it can re-localize the system when the pose tracking is lost and improve the robustness of the system. In recent years, lots of image-based place recognition methods [8, 2, 23] have been developed and achieved satisfactory performance. However, these methods are vulnerable to illumination changes and view variation due to the imaging mechanism of camera sensors. On the contrary, point clouds of LiDAR sensors are robust to illumination changes due to active sensing. In addition, the availability of precise depth information can help more accurate place recognition [1, 19].

LiDAR-based place recognition can be regarded as a retrieval problem, that is, finding the most similar frame to a query from a pre-built database. The key to solving this problem is to generate a global feature that can model the similarity between point clouds. PointNetVLAD [1] gives the first deep-learning solution to the problem of large-scale LiDAR-based place recognition. It uses PointNet [27] to extract local features from unordered points and NetVLAD [2] to generate global features. There are lots of subsequent methods that follow PointNetVLAD and introduce auxiliary modules such as attentions [36, 30], handcrafted features [19], and sparse convolution [15]. Recently, some methods [4, 22] based on range images have been developed. The range image is the sphere projection of a point cloud. Due to the projection mechanism, the translation of the range image is equivariant to the rotation of the point cloud. Based on this, OverlapTransformer [22] uses a convolution network and a transformer to extract rotation-invariant features from the images. Some methods [13, 14, 4] use similar projections and also achieve place recognition robust to view changes.

Although the aforementioned methods have made great progress, they still have limitations in terms of generalization ability. This is because both unordered points and range images used for place recognition are sensitive to the motions of the LiDAR sensor. Specifically, for unordered points, the point coordinate and the relative positions between points will change severely along with motions of the LiDAR sensor. For range images, the image contents suffer various distortions with translations of point clouds although they are robust to rotations. Current methods [1, 36, 19] force the network to learn these variations of data with data augmentation. However, as pointed out in [17], data augmentation needs the network to be as flexible as possible to capture all the variations, which may result in the large risk of overfitting and poor generalization ability.

In this work, we explore the potential of place recognition using bird's eye view (BEV) images. The BEV image is generated by projecting a point cloud to the ground space. In road scenes, the transformations of point clouds are approximately equivariant to the rotations and translations of BEV images [20]. Thus, the contents of BEV images are more robust to sensor motions. As shown in Fig. 1, the translation of a point cloud causes little appearance changes in the BEV image but introduces geometry distortions to the range image. The results shown in Fig. 1 (c) validates that a simple NetVLAD based on the BEV representation achieves comparable performance with the state-of-the art methods. To achieve robustness to viewpoint changes, we design a group convolution [31] network to extract local features from BEV images. Then, we use NetVLAD [2] for global rotation-invariant features extraction. Benefiting from the design of rotation invariance for BEV images, our method has the strong ability of place retrieval in the cases of both viewpoint variations and scene changes. In addition, we observe that the distances of the BEV features correlate well with the geometry distances of point clouds. According to this correlation, we map the feature distance to the geometry distance and then estimate the position of the query cloud, which extends the usage of LiDAR-based place recognition.

We summarize the contributions of this paper as follows:

- We experimentally show that, without any delicate design, a simple NetVLAD network based on the BEV representation outperforms SOTA methods on the KITTI dataset[9] and the benchmark dataset[1].

- We propose a novel LiDAR-based place recognition method called BEVPlace. The method is robust to view changes, has strong generalization ability, and achieves SOTA performance on three large-scale datasets.

- We explore the statistical correlation between the feature distance and the geometry distance of point cloud pairs. To the best of our knowledge, this paper is the first to perform position estimation directly from global descriptors.

2. Related Work

In this section, we briefly review the recent developments in the field of LiDAR-based place recognition. For a more comprehensive overview, the readers may refer to [25]. According to the representations used for feature extraction, we classify the current Lidar-based place recognition methods into two categories, *i.e.,* the methods that utilize 3D points and the methods that use projection images as intermediate representations.

Place recognition based on 3D points. PointNetVLAD [1] leverages PointNet [27] to project each point into a higher dimension feature, and then uses NetVLAD [2] to generate global features. To take advantage of more contextual information, PCAN [36] introduces the point contextual attention network that learns attentions to the task-relevant features. Both PointNetVLAD and PCAN can-

not capture local geometric structures due to the independent treatment for each point. Thus, the following methods focus on extracting more discriminative local features considering the neighborhood information. LPD-Net [19] adopts an adaptive local feature module to extract the handcrafted features and uses a graph-based neighborhood aggregation module to discover the spatial distribution of local features. EPC-Net [10] improves LPD-Net by using a proxy point convolutional neural network. DH3D [6] designs a 3D local feature encoder to learn more distinct local descriptors, and SOE-Net [34] introduces a point orientation encoding (PointOE) module. Minkloc3D [15, 16] uses sparse 3D convolutions in local areas and achieves state-of-the-art performance on the benchmark dataset. Recently, some works including SVT-Net [7], TransLoc3D [35], NDT-Transformer [37], and PPT-Net [11] leverage the transformer-based attention mechanism [32] to boost place recognition performance. However, it was shown that MinkLoc3D outperforms these transformer-based methods with fewer parameters.

Place recognition based on projection images. Steder *et al.* [29] extract handcrafted local features from range images of point clouds and perform place recognition by local feature matching. Kim *et al.* [13] project the point cloud into a bearing-angle image and propose the scan context descriptor. They further introduce the concept of scan context image (SCI) [14] and achieve place recognition by classifying the SCIs using a convolutional network. OverlapNet [4] uses the overlap of range images to determine whether two point clouds are at the same place and uses a siamese network to estimate the overlap. OverlapTransformer [22] further uses a transformer architecture to learn rotation-invariant global features. CVTNet [21] combines range images and BEV images to perform matching. It transforms BEV images into a format similar to range images to achieve rotation-invariance. Different from the aforementioned methods based on the image representations that are built under polar or polar-like projections, BVMatch [20] projects point clouds into BEV images and extracts handcrafted BVFT features from the images. It then uses the bag-of-words model [8] to generate global features. However, it is shown that BVMatch cannot generalize well to unseen environments [20]. Different from BVMatch, we extract rotation-equivariant local features using group convolution [31] and generate global features by NetVLAD [2]. Thanks to the network design, our method can generalize to different scenes while keeping high recall rates.

3. Preliminaries

Let $\mathbf{m}_i$ be the point cloud collected by a sensor at the pose $\mathbf{T}_i = (\mathbf{R}_i, \mathbf{t}_i)$, where $\mathbf{R}_i$ is the rotation matrix and $\mathbf{t}_i$ is the position. The database formed by n point clouds and their associated poses could be denoted as

$\mathcal{M} = \{(\mathbf{m}_i, \mathbf{T}_i)\}_{i=1,2,\ldots,n}$. Given a query point cloud $\mathbf{m}_q$, place recognition aims at finding its most structurally similar point cloud from the pre-built database $\mathcal{M}$. In the problem of LiDAR-based place recognition, two point clouds are usually regarded as structurally similar if they are collected at geometry close places. Towards this goal, we design a network $f(\cdot)$ to map the point cloud to a distinct compact global feature vector such that $\|f(\mathbf{m}_q) - f(\mathbf{m}_i)\|_2 < \|f(\mathbf{m}_q) - f(\mathbf{m}_j)\|_2$ if $\mathbf{m}_q$ is structurally similar to $\mathbf{m}_i$ but dissimilar to $\mathbf{m}_j$. Based on the network f, we perform place retrieval by finding the point cloud with the minimum feature distance to the query point cloud.

In this work, we train our network based on BEV images of point clouds. In addition to place retrieval, we develop an extended usage that estimates the positions of the query point clouds.

4. Method

Our method is formed by two modules as illustrated in Fig. 2. In the BEVPlace network, we project the query point cloud into the BEV image. Then, we extract a rotation-invariant global feature through a group convolution network and NetVLAD [2]. In the position estimator, we retrieve the closest feature of the global feature from a pre-built database. We recover the geometry distance between the query and the matched point clouds based on a mapping model. The position of the query is estimated based on the recovered distances.

4.1. BEVPlace Network

In road scenes, a LiDAR sensor on a car or a robot can only move on the ground plane. Since we generate BEV images by projecting point clouds into the ground plane, the view change of the sensor will result in a rotation transformation on the image. To achieve robust place recognition, we aim at designing a network f to extract rotation-invariant features from BEV images. Denoting the rotation transformation $\mathbf{R} \in SO(2)$ on the BEV image $\mathbf{I}$ as $\mathbf{R} \circ \mathbf{I}$, the rotation invariance of f can be represented as

$$f(\mathbf{R} \circ \mathbf{I}) = f(\mathbf{I}). \tag{1}$$

A straightforward approach to achieve such invariance is to train a network with data augmentation [17]. However, data augmentation usually requires that the network has a larger group of parameters to learn the rotations and may not generalize to the combination of rotations and scenes not occurring in the training set. In this work, we use the cascading of a group convolution network and NetVLAD to achieve rotation invariance. Our BEVPlace has strong generalization ability since the network is designed inherently invariant to rotations.

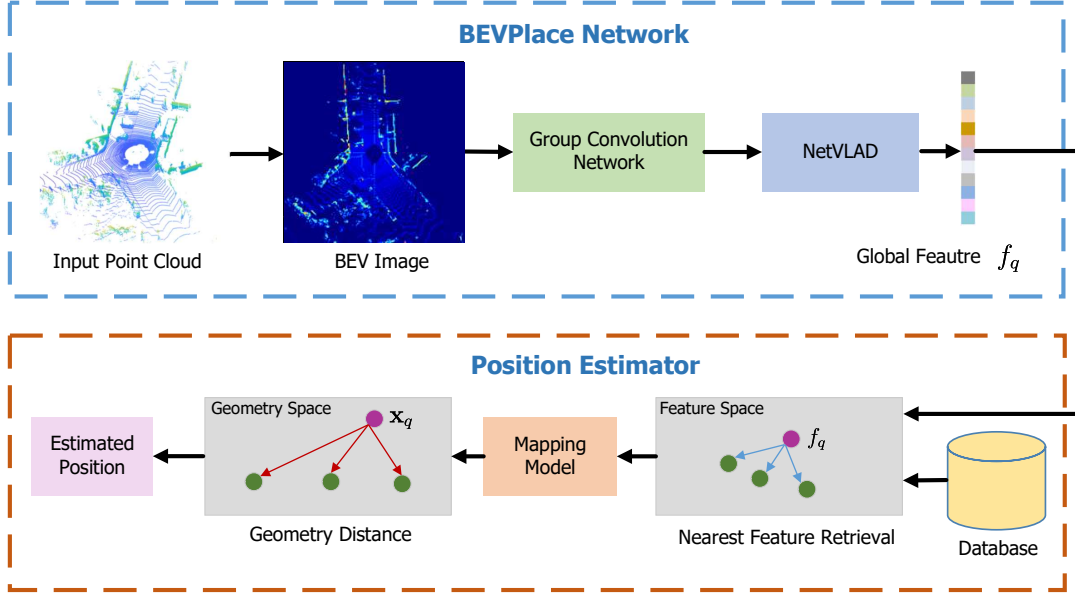**Bird's Eye View Image Generation.** We follow BVMatch [20] and use the point density to construct images.

Figure 2. Two modules of our method. In the BEVPlace network, we project point clouds into BEV images and extract rotation-invariant global features. In the position estimator module, we recover geometry distances from feature space and estimate positions of query point clouds.

We discretize the ground space into uniform grids with a grid size of 0.4 m. For a point cloud $\mathbf{m}$, we compute the number of points in each grid and use the normalized point density as the pixel intensity of the BEV image $\mathbf{I}$.

Group Convolution Network. Group convolution treats the feature map as functions of the corresponding symmetry-group [31]. Considering the 2D rotation group $SO(2)$, applying group convolution f_{gc} on a BEV image $\mathbf{I}$ results in rotation-equivariant features, which can be written as

$$f_{gc}(\mathbf{R} \circ \mathbf{I}) = \mathbf{R}' \circ f_{gc}(\mathbf{I}). \qquad (2)$$

That is, transforming the input $\mathbf{I}$ by a rotation transformation $\mathbf{R}$ and then passing it through the mapping f_{gc} should give the same result as first mapping $\mathbf{I}$ through f_{gc} and then transforming the feature with $\mathbf{R}' \in SO(2)$. Usually, f is designed such that $\mathbf{R}' = \mathbf{R}$.

Group convolution has been well-developed for a few years, and there are some mature group convolution designs [31, 18, 33]. We implemented our network based on GIFT [18]. GIFT is originally designed for image matching and can produce distinct local features. Our main modification to GIFT is to remove the scale features since there is no scale difference between BEV images. More details of our network implementation are appended in the supplementary materials.

Rotation invariant global features. According to Eq. 2, the contents of the feature map of group convolution keep the same for rotated images and are only transformed by a rotation $\mathbf{R}'$. Thus, we can use a global pooling opera-

tion to extract rotation-invariant global features. To capture more information about the statistics of local features, we use NetVLAD [2] for feature aggregation. We achieve rotation invariance by cascading the group convolution network and NetVLAD, which is,

$$\begin{aligned} \text{NetVLAD}\left(f_{gc}(\mathbf{R} \circ \mathbf{I})\right) &= \text{NetVLAD}\left(\mathbf{R}' \circ f_{gc}(\mathbf{I})\right) \\ &= \text{NetVLAD}\left(f_{gc}(\mathbf{I})\right). \end{aligned} \qquad (3)$$

Loss function. There are some loss functions [1, 34] for LiDAR-based place recognition problem. In this work, we train our network with the simple commonly used lazy triplet loss [1], formulated as

$$\mathcal{L} = \max_{j}([m + \delta_{\text{pos}} - \delta_{\text{neg}j}]_+), \qquad (4)$$

where $[\ldots]_+$ denotes the hinge loss, m is the margin, δ_{pos} is the feature distance between an anchor point cloud $\mathbf{m}_a$ and its structurally similar ("positive") point cloud, $\delta_{\text{neg}j}$ is the feature distance between $\mathbf{m}_a$ and its structurally dissimilar ("negative") point cloud. We follow the training strategy in [1, 19, 34] and regard two point clouds are structurally similar if their geometry distance is less than ϵ meters.

4.2. Position Estimator

The lazy triplet loss forces the network to learn a mapping that preserves the adjacency of point clouds in the geometry space. Although there isn't an explicit mapping function that reveals the relationship between the feature space and the geometry space, we observe that the distance of global features and the geometry distance of point

clouds are inherently correlated. Based on this property, we recover the geometry distance between the query and the match and then use it for position estimation.

Statistical correlation between the feature and geometry distances. To reveal the relationship between the feature space and the geometry space, we train our method on the sequence "*00*" of the KITTI dataset [9]. We then plot the feature distances and the geometry distances of all point cloud pairs in different sequences of the dataset. As shown in Fig. 3, for all the sequences, the feature distance approximately monotonically increases with the geometry distance and saturates when the point clouds are far away from each other. This phenomenon is intuitive since two point clouds are more similar if they are geometry closer, and consequently the feature distance is smaller. It can be seen that the mean curve and the standard deviation differ in different sequences since sequences are collected in diverse scenes. Despite this, the mean curves have similar shapes and can be depicted using a function based on the generalized Gaussian kernel [24], which is

$$\|f(\mathbf{m}_i - f(\mathbf{m}_j)\|_2 = \alpha \left(1 - \exp(-\frac{\|\mathbf{t}_i - \mathbf{t}_j\|_2^\gamma}{\beta}) \right), \quad (5)$$

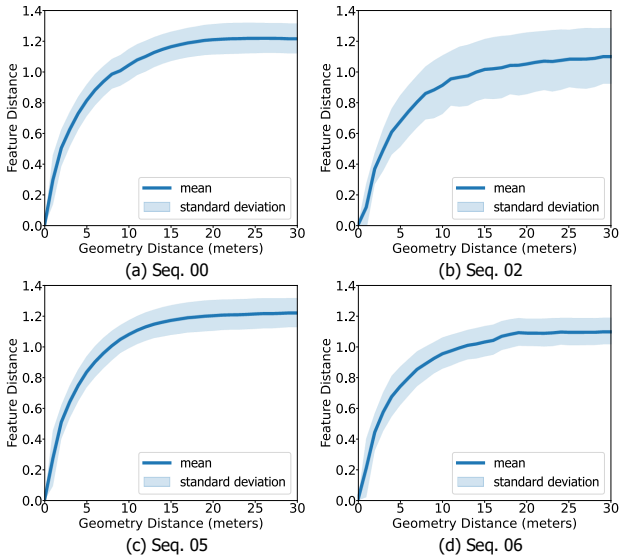where α is the max feature distance, γ and β control the curve shape.

in the database $\mathcal{M}$. Specifically, we compute its feature and geometry distances to all the other point clouds in $\mathcal{M}$. We then fit the curve with Eq. 5 and compute the parameters α_i, β_i, and γ_i. After this, we can recover the geometry distance of a query point cloud $\mathbf{m}_q$ to $\mathbf{m}_i$ according to Eq. 5, that is

$$\|\mathbf{t}_q - \mathbf{t}_i\|_2 = \left(-\beta_i \log(1 - \frac{\|f(\mathbf{m}_q) - f(\mathbf{m}_i)\|_2}{\alpha_i}) \right)^{\frac{1}{\gamma_i}}.$$
$$(6)$$

Position Recovery. Since the positions of the point clouds in the database are given, we can compute the position of the query point cloud $\mathbf{m}_q$ if we know its geometry distances to at least three reference point clouds. To this end, we first follow the place recognition procedure and find the most similar point cloud $\mathbf{m}_r$ of $\mathbf{m}_q$. We choose the reference point clouds as $\mathbf{m}_r$ and the point clouds that are less than ϵ meters away from $\mathbf{m}_r$. Denoting $\Omega = \{k\big|\|\mathbf{t}_r - \mathbf{t}_k\|_2 < \epsilon\}$ and d_k as the recovered geometry distance between $\mathbf{m}_q$ and $\mathbf{m}_k$, the position of $\mathbf{m}_q$ can be easily computed by solving the following minimization problem,

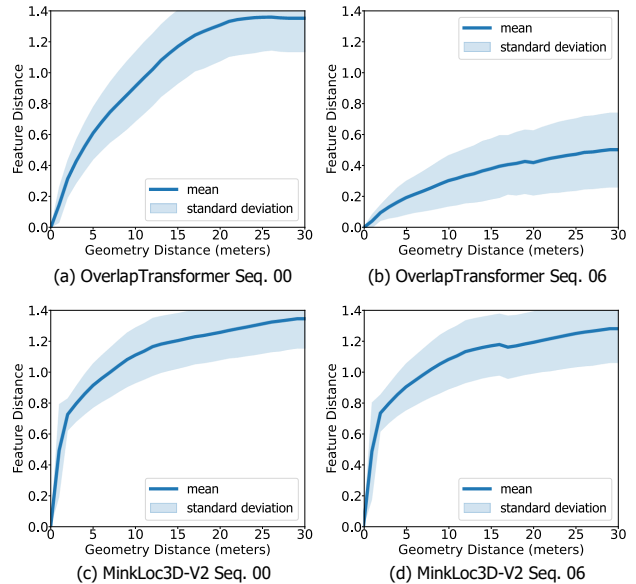$$\mathbf{t}_q = \arg\min_{\mathbf{t}} \sum_{k \in \Omega} \left(\|\mathbf{t} - \mathbf{t}_k\|_2 - d_k \right)^2. \quad (7)$$

Figure 3. Geometry distance and feature distance relationship of the point clouds in different sequences of the KITTI dataset.

Figure 4. Geometry distance and feature distance relationship of the point clouds in the sequence "*00*" and "*06*" of KITTI of two methods.

Mapping Model. The mapping function above inspires us to recover the geometry distance from the feature distance, and further estimate positions of the query point clouds. However, this mapping relationship may differ slightly in local areas due to the appearance changes of point clouds. For more accurate geometry distance recovery, we build a mapping function for each point cloud $\mathbf{m}_i$

Discussion. In fact, the monotonicity of the mapping from feature distance to the geometry distance also holds for other methods. Fig. 4 plots the relationship between the feature and the geometry spaces of two state-of-the-art methods Minkloc3D-V2 [15] and OverlapTransformer [22] on the sequence "*00*" and "*06*" of KITTI. Although the mappings of the methods have quite different shapes, they

all can be approximately depicted by Eq. 5 with specific parameters and thus positions can also be estimated based on the mapping model. In the experiment, we will compare their position estimation accuracy with our method.

5. Experiments

We compare our method with the state-of-the-art place recognition methods including Scan Context [13], BV-Match [20], PointNetVLAD [1], LPD-Net [19], SOE-Net [34], MinkLock3D-V2 [16], and OverlapTransformer [22], among which the last 5 methods are deep learning ones. The open-sourced codes of compared methods are used for evaluation. For our method, we set the triplet margin $m = 0.3$, and the number of clusters of NetVLAD as 64. In the training phase, we choose 1 positive point cloud and 10 negative point clouds in calculating loss functions.

We test the methods in terms of place retrieval with metrics of recall at Top-1 and recall at Top-%1. For a more comprehensive evaluation, we also compare the loop closure detection performance with the metric of Precision-Recall (PR) curve. In addition, we test the position estimation accuracy using the absolute translation error (ATE).

5.1. Datasets

We conduct experiments on three large-scale public datasets, i.e. the KITTI dataset [9], the ALITA dataset [26], and the benchmark dataset [1].

KITTI dataset contains a large number of point cloud data collected by a Velodyne 64-beam LiDAR under low viewpoint variation. We select the sequences "00", "02", "05", and "06" under the Odometry subset for evaluation since these sequences contain large revisited areas. We split the point clouds of each sequence into database frames and query frames for place retrieval. The partition of each sequence is summarized in Table 1. For our method, we crop each point cloud with a $[-20 \text{ m}, 20 \text{ m}]$ cubic window and downsample it into 4096 points. We then generate BEV images from the downsampled point clouds. For Point-NetVLAD, LPD-Net, MinkLock3D-V2, we normalize the point values to fit their input. For OverlapTransformer, we use full point clouds since its performance is sensitive to the point density.

Table 1. Dataset Partition of the KITTI dataset.

Sequence	00	02	05	06
Database	0-3000	0-3400	0-1000	0-600
Query	3200-4650	3600-4661	1200-2751	800-1100

ALITA dataset is a dataset for long-term place recognition in large-scale environments. It contains point cloud data of campus and city scenes under different illuminations and viewpoints. In this work, we use its subset released in the General Place Recognition Competition [1]. We evaluate the generalization ability of the methods on its validation set and its test set. Note that the evaluation result of the test set is automatically calculated by the server once we upload the global features to the website. The point clouds of the dataset have been cropped into a $[-20 \text{ m}, 20 \text{ m}]$ cubic window and downsampled into 4096 points. Similar to the process in the KITTI dataset, we generate BEV images with the downsampled point clouds for our method. We normalize the points to fit the input of PointNetVLAD, LPD-Net, and MinkLock3D-V2. We do not evaluate OverlapTransformer on this dataset as it cannot adapt to such sparse point clouds.

Benchmark dataset is broadly used by the recent place recognition method based on unordered points. It is a dataset set consisting of four scenarios: an outdoor dataset Oxford RobotCar, three in-house datasets of a university sector (U.S.), a residential area (R.A.), and a business district (B.D.). It provides normalized point clouds of 4096 points, which can be directly used by PointNetVLAD, LPD-Net, and MinkLock3D-V2. For our method, we multiply the point values by 20 and then project the point cloud into the BEV image for feature extraction. Note that the recovered point cloud is not of the actual scale since we do not know the exact coordinate range of the point clouds. In spite of this, our method can adapt to such scale variation thanks to the convolution network design.

For the KITTI dataset and the ALITA dataset, we regard a retrieval as true positive if the geometry distance between the query and the match is less than $\epsilon = 5$ meters. For the benchmark dataset, we set $\epsilon = 25$ meters following the configurations in [1, 19, 34, 16].

5.2. Place Recognition

We train the methods with the point clouds in the database of sequence "00" of the KITTI dataset. During the training phase, we apply data augmentation by randomly rotating the point clouds around the z-axis (which is perpendicular to the ground) within the interval of $[-\pi, \pi)$.

Ablation on the BEV representation and Group Convolution Network. To validate the claim that a NetVLAD network based on BEV images can achieve comparable performance to the SOTA methods, we implemented two NetVLAD networks with ResNet34 and ResNet18 as backbones, respectively. To explore the influence of rotation-invariance designs to the unordered points based methods, we additionally implement a network VN-PointNetVLAD by replacing the backbone of PointNetVLAD with Vector Nureon [5]. We observe the results shown in Table. 2 and find that,

- Without any delicate design, both NetVLAD networks achieves comparable recalls to OverlapTransformer
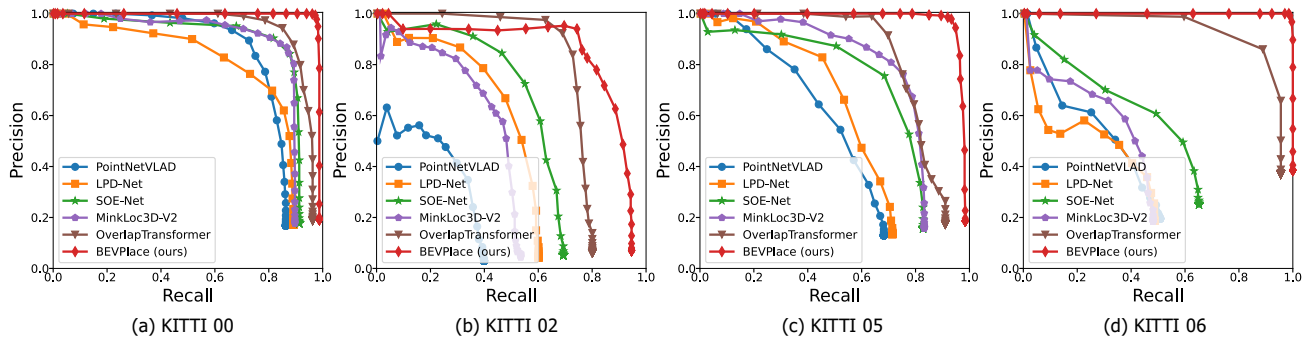
Figure 5. Precision-recall curves for the sequences of the KITTI dataset.

and outperforms the unordered points based methods. While BVMatch also leverages BEV images, it generates global descriptors using the bag-of-words model [8], which leads to inferior performance compared to the deep learning-based BEV methods.

- Our BEVPlace achieves higher recalls than the NetVLAD networks, validating the significance of the rotation invariance design.

- Among the rotation-invariant methods based on different representations, i.e. VN-PointNetVLAD based on unordered points, OverlapTransformer based on range images, and BEVPlace based on BEV images, our BEVPlace achieves higher recall and better generalization ability than the non-BEV networks, further indicating the importance of the BEV representation in place recognition.

Table 2. Recall at Top-1 on the KITTI dataset. * denotes that method is designed rotation-invariant.

Sequence	00	02	05	06	Mean
Scan Context [13]	89.7	73.9	77.0	86.7	81.8
PointNetVLAD[1]	91.6	62.3	76.9	77.8	77.2
LPD-Net [19]	95.7	72.3	83.6	82.2	83.5
SOE-Net [34]	95.0	65.5	84.8	69.6	78.7
MinkLoc3D-V2 [15]	95.9	72.3	86.4	80.4	83.4
*BVMatch [20]	93.8	78.2	90.2	93.8	89.0
*VN-PointNetVLAD [1, 5]	94.3	66.5	87.5	84.3	82.9
*OverlapTransformer[22]	96.7	80.1	91.9	95.6	91.1
ResNet18+NetVLAD	95.9	83.2	90.3	98.5	92.0
ResNet34+NetVLAD	96.3	84.1	92.2	98.5	92.8
*BEVPlace (ours)	**99.7**	**98.1**	**99.3**	**100.0**	**99.3**

Robustness to view changes. In the testing phase, we randomly rotate the point clouds of KITTI along z-axis to simulate view changes. As shown in Table 3, our method shows much higher recall rates than ResNet18-NetVLAD and ResNet34+NetVLAD. It is also noted that VN-PointNetVLAD performs better than PointNetVLAD. These validate the significance of the rotation invariance de-

Table 3. Recall at Top-1 on the rotated KITTI dataset. * denotes that method is designed rotation-invariant.

Sequence	00	02	05	06	Mean
Scan Context [13]	89.7	73.9	77.0	86.7	81.8
PointNetVLAD[1]	86.1	41.0	69.7	51.5	62.1
LPD-Net [19]	89.6	61.9	72.2	48.9	68.2
SOE-Net [34]	93.1	63.5	82.8	65.5	76.2
MinkLoc3D-V2 [15]	89.4	48.7	83.0	48.1	67.3
*BVMatch [20]	93.5	77.8	89.1	92.6	88.6
*VN-PointNetVLAD	93.2	62.3	85.2	82.9	80.9
*OverlapTransformer[22]	96.7	80.1	91.9	95.6	91.1
ResNet18-NetVLAD [28]	92.3	64.1	89.8	89.9	84.0
ResNet34+NetVLAD [2]	93.1	64.2	90.7	90.4	84.6
*BEVPlace (ours)	**99.6**	**93.5**	**98.9**	**100.0**	**98.0**

sign to view variations. However, VN-PointNetVLAD cannot generalize well to sequences "00", "02", "05", and "06". On the other hand, our BEVPlace shows much higher recall and better generalization ability than all the other methods.

Loop closure detection. Loop closure detection is an important application of place recognition. For a query point cloud, we accept its Top-1 match as positive if the feature distance is less than a threshold. By setting different thresholds, we compute the precision-recall curves and plot them in Fig. 5. It can be seen that our method outperforms the compared methods. It is worth noting that although our method is only trained on a part of the point clouds of the sequence 00, it generalizes much better to the other sequences than the other methods. We believe our method can be deployed by LiDAR SLAM systems [12] and help globally consistent mapping building.

Table 4. Recall rates on the ALITA dataset.

	Val Set		Test set	
	@1	@1%	@1	@1%
PointNetVLAD[1]	42.3	55.4	39.8	-
LPD-Net [19]	51.2	72.7	49.6	-
SOE-Net [34]	66.6	92.8	59.5	-
MinkLoc3D-V2 [15]	55.6	82.8	55.3	-
BEVPlace (ours)	**96.7**	**99.2**	**91.7**	-

Table 5. Recall rates on the benchmark dataset.

	Oxford		U.S.		R.A.		B.D		Mean	
	AR@1	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1	AR@1%
PointNetVLAD [1]	62.8	80.3	63.2	72.6	56.1	60.3	57.2	65.3	59.8	69.6
LPD-Net [19]	86.3	94.9	87.0	96.0	83.1	90.5	82.5	89.1	84.7	92.6
NDT-Transformer [37]	93.8	97.7	-	-	-	-	-	-	-	-
PPT-Net [11]	93.5	98.1	90.1	97.5	84.1	93.3	84.6	90.0	88.1	94.7
SVT-Net [7]	93.7	97.8	90.1	96.5	84.3	92.7	85.5	90.7	88.4	94.4
TransLoc3D [35]	95.0	98.5	-	-	-	-	-	-	-	-
MinkLoc3Dv2 [16]	96.3	98.9	90.9	96.7	86.5	93.8	86.3	91.2	90.0	95.1
ResNet34+NetVLAD [16]	95.8	**99.2**	96.2	99.0	90.1	**99.4**	94.8	**100.0**	94.2	**99.4**
BEVPlace (ours)	**96.5**	99.0	**96.9**	**99.7**	**92.3**	98.7	**95.3**	99.5	**95.3**	99.2

Generalization performance on ALITA. We test the place recognition performance of the methods on the ALITA dataset based on the model trained on the KITTI dataset. Table 4 shows the recall rates on the validation set and the test set. It can be seen that our method generalizes well in ALITA. On the other hand, the recall rates of the compared methods degrade much.

Performance on benchmark datasets. Following the previous works, we train our method using only the Oxford RobotCar training dataset and test the method on the test set. The details of the dataset partition can be found in [1]. For a more comprehensive comparison, we also compare our method with the state-of-the-art transformer-based methods, including NDT-Transformer [37], PPT-Net [11], SVT-Net [7], and TransLoc3D [35]. For all the compared methods, we directly use the results from their papers. Table 5 shows that the ResNet34+NetVLAD network achieves comparable recall to the state-of-the-art method minklock3D-v2 and shows better generalization ability. Our BEVPlace outperforms other methods including the transformer-based ones with large margins.

5.3. Position Estimation

We first recovery the geometry distances between the query and the matches and then estimate the global position of point clouds. In the following, we evaluate the performance of these two phases.

Accuracy of the recovery distances. We compute the errors of the recovered distances on the sequence "00" of the KITTI dataset. Fig. 6 shows the fitting distribution of the distance errors of the methods. It can be seen that our method can recover the geometry distance more accurately. This will lead to more accurate position estimation results since the estimation is based on the recovered distances.

Position estimation. Fig. 7 (a), (b), (c), and (d) show the cumulative distribution of the translation error on different sequences of the KITTI datasets. Our method and OverlapTransformer, both of which are based on projection images, achieve more accurate position estimation than the compared methods. To validate the performance under view changes, we randomly rotate the point clouds in the testing
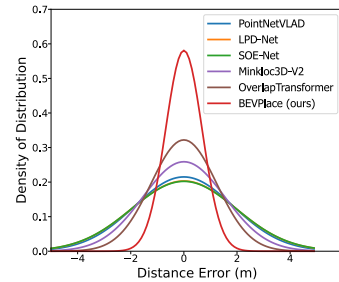

Figure 6. Distance estimation error distribution.

phase. Fig. 7 (e), (f), (g), and (h) show that our method and OverlapTransformer perform well since they are designed to be rotation invariant. On the other hand, the other methods shows poor robustness and their performance degrades much.

5.4. Running Time

We evaluate the runtime of BEVPlace on a desktop with an RTX3090 GPU. For each query, place recognition costs about 30 ms, and position estimation costs less than 1 ms. Our method can realize real-time localization as most Li-DAR sensors operate at 10 Hz.

6. Conclusions

In this work, we explore the potential of LiDAR-based place recognition using BEV images. We designed a rotation invariant network called BEVPlace based on group convolution. Thanks to the use of BEV images and the rotation invariance design, our method achieves high recall rates, strong generalization ability, and robustness to viewpoint changes, as shown in the experiments. In addition, we observe that the geometry and feature distance are correlated, and we model the correlation for position estimation. This model can adapt to other place recognition methods, but our BEVPlace gives more accurate estimation results. In our future work, we will try to encode the rotation information into global features and estimate 6-DoF pose of point clouds.
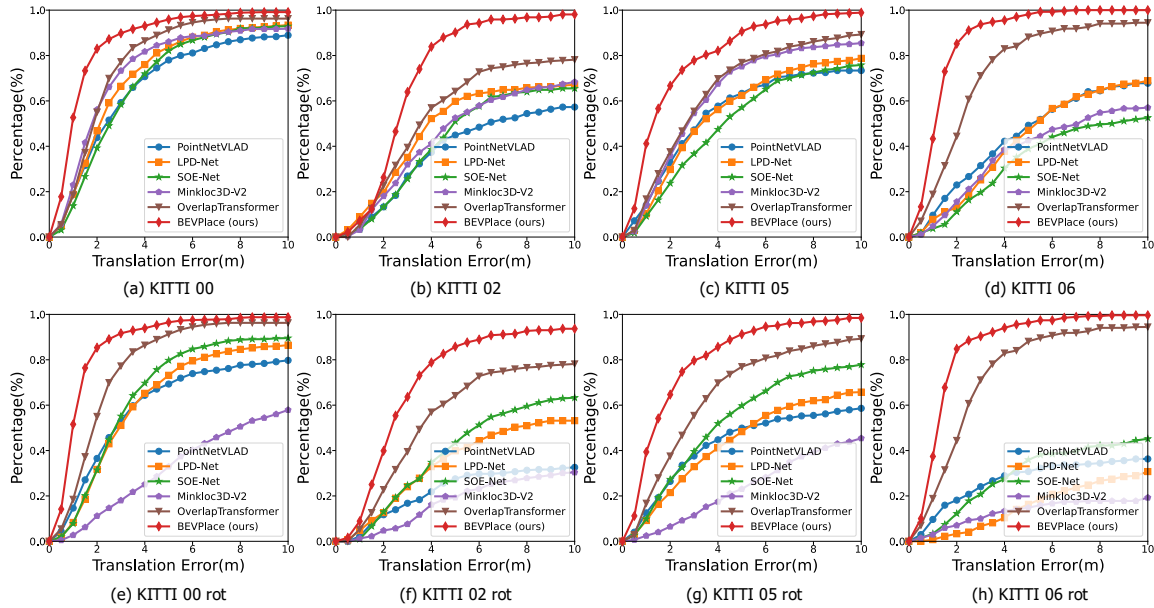
Figure 7. Accumulative translation error distribution on the KITTI dataset with and without rotations.

References

[1] Mikaela Angelina Uy and Gim Hee Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4470–4479. IEEE, 2018. 2, 4, 6, 7, 8

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 2, 3, 4, 7

[3] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. 1

[4] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, Cyrill Stachniss, and FKIE Fraunhofer. OverlapNet: Loop closing for LiDAR-based SLAM. In *Proc. of Robotics: Science and Systems*, 2020. 2, 3

[5] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for SO(3)-equivariant networks. In *IEEE International Conference on Computer Vision*, pages 12200–12209, 2021. 6, 7

[6] Juan Du, Rui Wang, and Daniel Cremers. DH3D: Deep hierarchical 3D descriptors for robust large-scale 6DoF relocalization. In *European Conference on Computer Vision*, 2020. 3

[7] Z. Fan, Z. Song, H. Liu, Z. Lu, J. He, and X. Du. SVT-Net: A super light-weight network for large scale place recognition using sparse voxel transformers. In *AAAI Conference on Artificial Intelligence*, 2022. 3, 8

[8] D Galvez-Lpez and J. D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. 2, 3, 7

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 2, 5, 6

[10] Le Hui, Mingmei Cheng, Jin Xie, Jian Yang, and Ming-Ming Cheng. Efficient 3D point cloud feature learning for large-scale place recognition. *IEEE Transactions on Image Processing*, 31:1258–1270, 2022. 3

[11] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid point cloud transformer for large-scale place recognition. In *IEEE International Conference on Computer Vision*, pages 6078–6087, 2021. 3, 8

[12] Zhang Ji and Singh Sanjiv. LOAM: LiDAR odometry and mapping in real-time. In *Proceedings of Robotics: Science and Systems*, 2014. 7

[13] Giseop Kim and Ayoung Kim. Scan Context: Egocentric spatial descriptor for place recognition within 3D point cloud map. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4802–4809. IEEE, 2018. 2, 3, 6, 7

[14] Giseop Kim, Byungjae Park, and Ayoung Kim. 1-day learning, 1-year localization: Long-term LIDAR localization using scan context image. *IEEE Robotics and Automation Letters*, 4(2):1948–1955, 2019. 2, 3

[15] Jacek Komorowski. MinkLoc3D: Point cloud based large-scale place recognition. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1789–1798, 2021. 2, 3, 5, 7

[16] Jacek Komorowski. Improving point cloud based place recognition with ranking-based loss and large batch training. In *IEEE International Conference on Pattern Recognition*, 2022. 3, 6, 8

[17] Dmitry Laptev, Nikolay Savinov, Joachim M. Buhmann, and Marc Pollefeys. TI-POOLING: Transformation-invariant pooling for feature learning in convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 289–297, 2016. 2, 3

[18] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. GIFT: Learning transformation-invariant dense visual descriptors via group cnns. In *Conference and Workshop on Neural Information Processing Systems*, 2019. 4

[19] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis. In *IEEE International Conference on Computer Vision*, pages 2831–2840. IEEE, 2019. 2, 3, 4, 6, 7, 8

[20] Lun Luo, Si-Yuan Cao, Bin Han, Hui-Liang Shen, and Junwei Li. BVMatch: Lidar-based place recognition using bird's-eye view images. *IEEE Robotics and Automation Letters*, 6(3):6076–6083, 2021. 2, 3, 6, 7

[21] Junyi Ma, Guangming Xiong, Jingyi Xu, and Xieyuanli Chen. CVTNet: A cross-view transformer network for place recognition using lidar data. *arXiv preprint arXiv: 2302.01665*, 2023. 3

[22] Junyi Ma, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. OverlapTransformer: An efficient and yaw-angle-invariant transformer network for LiDAR-based place recognition. *IEEE Robotics and Automation Letters*, 7(3):6958–6965, 2022. 2, 3, 5, 6, 7

[23] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2017. 2

[24] Saralees Nadarajah. A generalized normal distribution. *Journal of Applied Statistics*, 32(7):685–694, 2005. 5

[25] Yin Peng, Zhao Shiqi, Cisneros Ivan, Abuduweili Abulikemu, Huang Guoquan, Milford Micheal, Liu Changliu, Choset Howie, and Scherer Sebastian. General place recognition survey: Towards the real-world autonomy age. In *arXiv preprint arXiv:2209.04497*, 2022. 2

[26] Yin Peng, Zhao Shiqi, Ge Ruohai, Cisneros Ivan, Fu Ruijie, Zhang Ji, Choset Howie, and A. Scherer Sebastian. ALITA: A large-scale incremental dataset for long-term autonomy. In *arXiv preprint arXiv:2105.11605*, 2022. 6

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2015. 7

[29] Bastian Steder, Michael Ruhnke, Slawomir Grzonka, and Wolfram Burgard. Place recognition in 3D scans using a combination of bag of words and point feature based relative pose estimation. In *IEEE International Conference on Intelligent Robots and Systems*, 2011. 3

[30] Qi Sun, Hongyan Liu, Jun He, Zhaoxin Fan, and Xiaoyong Du. DAGC: Employing dual attention and graph convolution for point cloud based place recognition. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 224–232, 2020. 2

[31] Cohen Taco and Welling Max. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 2990–2999, 2016. 2, 3, 4

[32] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference and Workshop on Neural Information Processing Systems*, pages 5998–6008, 2017. 3

[33] Maurice Weiler and Gabriele Cesa. General $e(2)$-equivariant steerable cnns. In *Conference and Workshop on Neural Information Processing Systems*, 2019. 4

[34] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. SOE-Net: A self-attention and orientation encoding network for point cloud based place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11343–11352, 2021. 3, 4, 6, 7

[35] T. X. Xu, Y. C. Guo, Y. K. Lai, and S. H. Zhang. TransLoc3D : Point cloud based large-scale place recognition using adaptive receptive fields. *arXiv preprint arXiv:2105.11605*, 2021. 3, 8

[36] Wenxiao Zhang and Chunxia Xiao. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[37] Zhicheng Zhou, Cheng Zhao, Daniel Adolfsson, Songzhi Su, Yang Gao, Tom Duckett, and Li Sun. NDT-Transformer: Large-scale 3D point cloud localisation using the normal distribution transform representation. In *IEEE International Conference on Robotics and Automation*, pages 5654–5660, 2021. 3, 8