Global Point Cloud Descriptor for Place Recognition in Indoor Environments

Jacek Komorowski, Grzegorz Kurzejamski, Monika Wysoczańska and Tomasz Trzcinski
Warsaw University of Technology, Warsaw, Poland

Keywords: Place Recognition, 3D Point Cloud, RGB-D, Deep Metric Learning.

Abstract: This paper presents an approach for learning-based discriminative 3D point cloud descriptor from RGB-D images for place recognition purposes in indoor environments. Existing methods, such as such as PointNetVLAD, PCAN or LPD-Net, are aimed at outdoor environments and operate on 3D point clouds from LiDAR. They are based on PointNet architecture and designed to process only the scene geometry and do not consider appearance (RGB component). In this paper we present a place recognition method based on sparse volumetric representation and processing scene appearance in addition to the geometry. We also investigate if using two modalities, appearance (RGB data) and geometry (3D structure), improves discriminativity of a resultant global descriptor.

1 INTRODUCTION

Depth-aware sensors, such as time-of-flight cameras or solid state lidars, are becoming more and more affordable and popular. Self-driving cars are frequently equipped with LiDAR scanner which produces a map of the observed environment in the form of a sparse 3D point cloud. In indoor environments, inexpensive time-of-flight cameras, such as the latest generation of Azure Kinect, can generate a representation of an observed scene in the form of an RGB point cloud. Applying deep learning methods to solve 3D computer vision problems based on point cloud representation is an area of active development. A number of methods for classification (Qi et al., 2017a; Qi et al., 2017b), object detection (Qi et al., 2017a; Wang and Jia, 2019), semantic segmentation (Qi et al., 2017a; Choy et al., 2019a) and local (Zeng et al., 2017; Choy et al., 2019b) or global (Angelina Uy and Hee Lee, 2018; Liu et al., 2019) features extraction from 3D point clouds was recently proposed.

We focus our attention on finding a discriminative, low-dimensional 3D point cloud descriptor for place recognition purposes. Such global descriptors are computed for each processed point cloud and stored in the database. Localization is performed by an efficient search for descriptors closest (in Euclidean distance sense) to the query point cloud descriptor. This allows efficiently retrieving the most similar point clouds from the database and reason about the localization of the query point cloud.

In this paper we investigate if using two modalities, appearance (RGB data) and geometry (3D structure), can improve discriminativity of a global point cloud descriptor for place recognition purposes. State-of-the-art place recognition methods based on 3D point clouds, such as PointNetVLAD (Angelina Uy and Hee Lee, 2018), PCAN (Zhang and Xiao, 2019) or LPD-Net (Liu et al., 2019), operate on data acquired in an outdoor environment by a car-mounted LiDAR. They compute a discriminative global descriptor from a raw 3D point cloud, which is then used to find and retrieve the most similar point clouds from the database. These methods are based on a single modality only – geometry. Focusing solely on geometry and neglecting appearance (RGB) component is justified for place recognition in outdoor environments. An appearance of the observed scene can vary drastically due to lighting and seasonal changes. Whereas LiDAR acquired geometry remains relatively constant thorough different times of the day, seasons and weather conditions. In indoor environments there's less variability of appearance component, hence it's reasonable to use both modalities for indoor place recognition task.

Data acquired using LiDAR in an outdoor environment has a different characteristic than data gathered indoor using RGB-D cameras with time-of-flight

Komorowski, J., Kurzejamski, G., Wysoczańska, M. and Trzcinski, T.
Global Point Cloud Descriptor for Place Recognition in Indoor Environments.
DOI: 10.5220/0010340502160224
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 216-224
ISBN: 978-989-758-488-6

sensor. The former creates a sparser point cloud. Surfaces, such as building facades, are relatively far from the observer and mapped with less detail. The latter creates denser point clouds. Observed surfaces are closer to the camera and captured with greater details. Fine-grain structures of objects like like furniture are mapped in detail. Both PointNetVLAD and PCAN methods use PointNet (Qi et al., 2017a) backbone as the first stage of the processing pipeline. While PointNet architecture proved to be successful in many applications, it was originally used to process point clouds representing single objects, not large and complex scenes. The drawback of PointNet architecture is that for the most part each point it processed in isolation. Local features computed separately for each point are aggregated in the last few fully connected layers. As such, it's not well suited to capture local geometric structures of the observed scene. Such structures are more prevalent in indoor scene scans using RGB-D cameras with time-of-flight sensor than in outdoor, LiDAR-based scene scans.

An alternative is to use voxelized representation which can be processed using 3D convolutions. Convolutions proved to be very successful in processing 2D visual information as they can effectively capture local structures in the image. However, the naive voxelized representation based on a dense grid of voxels is very inefficient. Most of the voxels are empty and processing entire grid of voxels is computationally very expensive. Recently, an interesting alternative emerged. So called *Minkowski convolutional neural networks* (Choy et al., 2019a; Choy et al., 2019b) are based on a sparse voxelized representation and a very efficient implementation of sparse 3D and higher dimensional convolutions The sparse representation scales linearly with the number of 3D points, without the need to store and process dense 3D voxel grid. The approach proved to be very successful, achieving state-of-the-art methods in different 3D vision tasks, such as semantic segmentation (Choy et al., 2019a) and local features extraction (Choy et al., 2019b). In this paper we compare performance of sparse voxelized point cloud representation and sparse 3D convolutions against unordered point cloud representation and PointNet architecture for place recognition task.

In summary, main contributions of this work are as follows. First, we examined if using two modalities, geometry (3D structure) and appearance (RGB data), can improve place recognition precision in an indoor environment. Are there any advantages from fusing two modalities, or does one dominates the other and there's no gain from using both of them? Second, we experimentally verify if using sparse voxelized representation is advantageous over the popular PointNet architecture based on 'set of unordered points' representation, for place recognition purposes.

2 RELATED WORK

Point Cloud Representation for Deep Learning. Early deep learning methods operating on 3D point clouds use volumetrically discretized representations (Maturana and Scherer, 2015) in the form of a dense grid of voxels. It's a natural extension of 2D image representation as a grid of pixels and 3D convolutions can be applied to process such data. However, such representation is very inefficient. The memory requirements grow cubically as spatial resolution increases, making it inappropriate for processing larger point clouds. Most of the voxels are empty and processing entire grid of voxels is very inefficient and computationally expensive.

(Su et al., 2015) proposed so called multi-view approach, multiple 2D images of a 3D model are first rendered by virtual cameras placed around the object of interest. Each virtually rendered image is processed by 2D convolutional network. Feature maps produced by 2D networks are concatenated and fed into the final classification network.

PointNet (Qi et al., 2017a) was the first deep learning method operating directly on a raw 3D point cloud. An input is organized as an unordered set of points, where each point is described by its X, Y, Z coordinates and optional features, such as normal or RGB. Each point is processed separately by multi-layer perceptrons and point features are aggregated using a symmetric function, such as max pooling. This makes the architecture independent from input points ordering. PointNet learns a set of functions that select interesting and informative key points from a subset of input points, encoding this information in each layers feature vector. The drawback of the architecture is that most of the processing is done separately for each point and the architecture is not well suited to capture local geometric structures. The advantage is it's efficiency, as there's no need to build a costly voxelized representation nor render multiple virtual images.

An alternative approach was recently proposed in (Choy et al., 2019a). So called *Minkowski convolutional neural networks* are based on a sparse voxelized representation and an efficient implementation of sparse 3D and higher dimensional convolutions. This representation joins advantages of both voxelized and 'unordered set of points' representations. As with voxelized representation, 3D convolu-

tions can be used to capture local structures, similarly as 2D convolutions in 2D images. And sparsity allows compact representation and efficient computation.

Place Recognition using Learning-based Global Features. PointNetVLAD (Angelina Uy and Hee Lee, 2018) was the first deep network for large-scale 3D point cloud retrieval. It combines PointNet (Qi et al., 2017a) architecture to extract local features and NetVLAD (Arandjelovic et al., 2016) to aggregate local features and produce a discriminative global descriptor.

PCAN (Zhang and Xiao, 2019) enhances PointNetVLAD architecture by adding an attention mechanism to predict significance of each local point feature based on a local context. Local features are extracted using PointNet architecture. Then, features fed to NetVLAD aggregation layer are weighted by their significance. More attention is paid to the localization task-relevant features, while non-informative features are ignored.

To mitigate limitations of PointNet-based architecture in local feature extraction, LPD-Net (Liu et al., 2019) relies on handcrafted features and uses graph neural networks to extract local contextual information. Ten handcrafted features, such as point density or local curvature, are computed for each point. Then, 3D points enhanced with handcrafted features are processed using Point Net architecture and fed to a graph neural network to aggregate neighbourhood features. Finally, global descriptor is computed using NetVLAD (Arandjelovic et al., 2016) layer.

A recent MinkLoc3D (Komorowski, 2020) network has a fully convolutional architecture based on a sparse voxelized representation. The local feature extraction part of the network is modelled after Feature Pyramid Network (Lin et al., 2017) design pattern. Generalized Mean Pooling layer is used to aggregate local features into a global descriptor. Despite its simplicity, the method achieves state-of-the-art results in the outdoor place recognition benchmarks.

Deep Metric Learning. Distance metric learning aims at learning a distance function to measure semantic similarity between data points (Lu et al., 2017). This approach is widely used in many recognition tasks in computer vision domain, such as pedestrian re-identification (Hermans et al., 2017) and image retrieval (Lee et al., 2008). Deep metric learning uses deep neural networks to compute a non-linear mapping from a high dimensional data point space to a low-dimensional Euclidean space, known as a representation or embedding space. The learned mapping preserves semantic similarity between objects. Embeddings of similar data points are closer to each other in a representation space than embeddings of dissimilar objects. Early deep metric learning methods use a Siamese architecture trained with a contrastive loss (Bromley et al., 1994). Latter methods propose more complex loss functions, such as triplet (Hermans et al., 2017) or quadruplet (Chen et al., 2017) loss. Significant attention is put to a selection of an effective sampling scheme to select informative training samples (so called hard negatives mining) (Wu et al., 2017). One of the most popular scheme is *batch hard* negative mining proposed in (Hermans et al., 2017).

3 GLOBAL POINT CLOUD FEATURE DESCRIPTOR FOR PLACE RECOGNITION

In this section we describe our approach for computation of a discriminative, global 3D point cloud descriptor based on two modalities: appearance (RGB data) and geometry (3D structure). We use a deep metric learning approach illustrated in Fig. 2. The embedding network f_w, parametrized by weights vector $\mathbf{w}$, is trained to produce a discriminative, low dimensional descriptor (embedding) of the input point cloud. The network is trained using a triplet loss (Hermans et al., 2017). The aim is to make embeddings of dissimilar point clouds (representing different places) to be further away by a predefined margin than embeddings of the similar point clouds (representing the same place).

We evaluate two architectures of an embedding network, each using a different point cloud representation. One is PointNetVLAD (Angelina Uy and Hee Lee, 2018) method using an unordered set of points representation. It consists of a PointNet-based (Qi et al., 2017a) backbone followed by NetVLAD (Arandjelovic et al., 2016) feature aggregation layer. For details, please refer to (Angelina Uy and Hee Lee, 2018). We modified the original architecture to accept input points clouds with optional RGB features in addition to XYZ coordinates. The network produces a 256 dimensional global descriptor.

The other approach is based on a sparse voxelized representation. Inspired by (Komorowski, 2020), we designed a 3D convolutional network (called MinkNetVLAD) based on a sparse voxelized representation, shown in Fig. 1. It consists of a fully convolutional local feature extraction block followed by NetVLAD (Arandjelovic et al., 2016) feature aggregation block. Local feature extraction network is

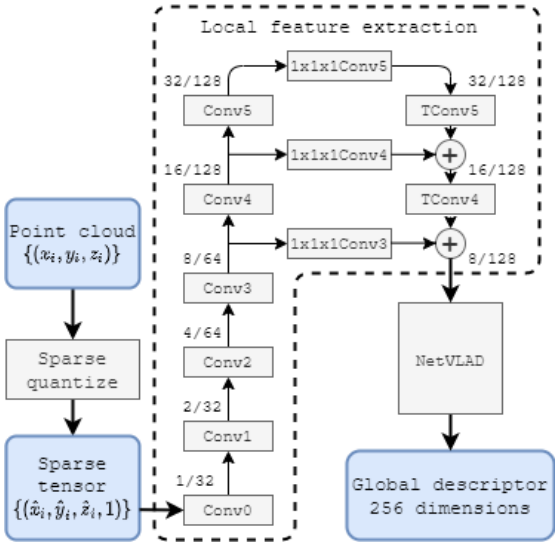

Figure 1: MinkNetVLAD network architecture. The input point cloud is quantized into a sparse 3D tensor. Local features are extracted using a sparse 3D convolutional network. NetVLAD pooling layer is used to pool the resultant 3D feature map and produce a global point cloud descriptor. Numbers in local feature extraction module (1/32, 2/32, ...) show a stride and number of channels of a sparse feature map produced by each block.

modelled after Feature Pyramid Network (Lin et al., 2017) architecture. Bottom-up part part consists of 3D convolutions producing 3D feature maps with decreasing spatial resolution and increasing receptive field. Downsampling of the feature map is achieved using stride 2 convolutions. The top-down part consists of transposed convolutions. Lateral connections (convolutions with 1x1x1 filter) are used to merge features produced by the bottom-up part of the network with the corresponding features from top-down pass. The rationale of using this design, instead of a simple convolutional network, is to increase the receptive field of each feature map element to allow capturing high-level semantic of the input point cloud. Architecture details are given in Tab. 1. Local features are aggregated using NetVLAD layer, producing a compact 256-dimensional global descriptor. The network is implemented using MinkowskiEngine (Choy et al., 2019a) auto-differentiation library for sparse tensors.

To asses impact of each modality on the discriminative power of the resulting global point cloud descriptor, we train the networks using three types of input: geometry (XYZ coordinates) and appearance (RGB component); geometry only; appearance only. When training using geometry only, all points, instead of RGB values, are assigned a dummy one dimensional feature set to 1. When training using appearance only, depth of all 3D points is set to the same dummy value.

Table 1: Details of the local feature extraction block in MinkNetVLAD network. All convolutional layers are followed by BatchNorm and ReLU non-linearity (not listed in the table for brevity).

Block	Layers
Conv0	32 filters 5x5x5
Conv1	32 filters 2x2x2 stride 2
	32 filters 3x3x3 stride 1
	32 filters 3x3x3 stride 1
Conv2	64 filters 2x2x2 stride 2
	64 filters 3x3x3 stride 1
	64 filters 3x3x3 stride 1
Conv3	64 filters 2x2x2 stride 2
	64 filters 3x3x3 stride 1
	64 filters 3x3x3 stride 1
Conv4	128 filters 2x2x2 stride 2
	128 filters 3x3x3 stride 1
	128 filters 3x3x3 stride 1
Conv5	128 filters 2x2x2 stride 2
	128 filters 3x3x3 stride 1
	128 filters 3x3x3 stride 1
1x1x1Conv4	128 filters 1x1x1 stride 1
1x1x1Conv4	128 filters 1x1x1 stride 1
1x1x1Conv4	128 filters 1x1x1 stride 1
Transposed convolutions	
TConv4	128 filters 2x2x2 stride 2
TConv5	128 filters 2x2x2 stride 2

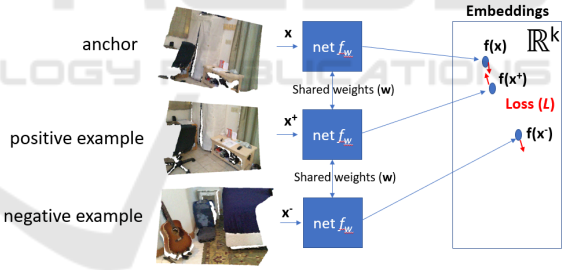

Figure 2: Learning a global point cloud descriptor using a deep metric learning technique with a triplet loss.

Dataset. We conduct our experiments using ScanNet (Dai et al., 2017) dataset. ScanNet is a large, richly-annotated dataset with 3D reconstructions of indoor scenes. It contains 2.5 million views (RGB-D images) in more than 1500 locations, annotated with 3D camera poses and surface reconstructions.

We split the dataset into three separate parts: training set, validation set to choose training hyperparameters and test set for final performance evaluation. The training set contains 993 thousand point clouds reconstructed from RGB-D images taken at 616 distinct locations. The validation set contains 65 thousand point clouds and 45 locations. The test set contains 253 thousand point clouds and 176 locations. Fig. 3 shows

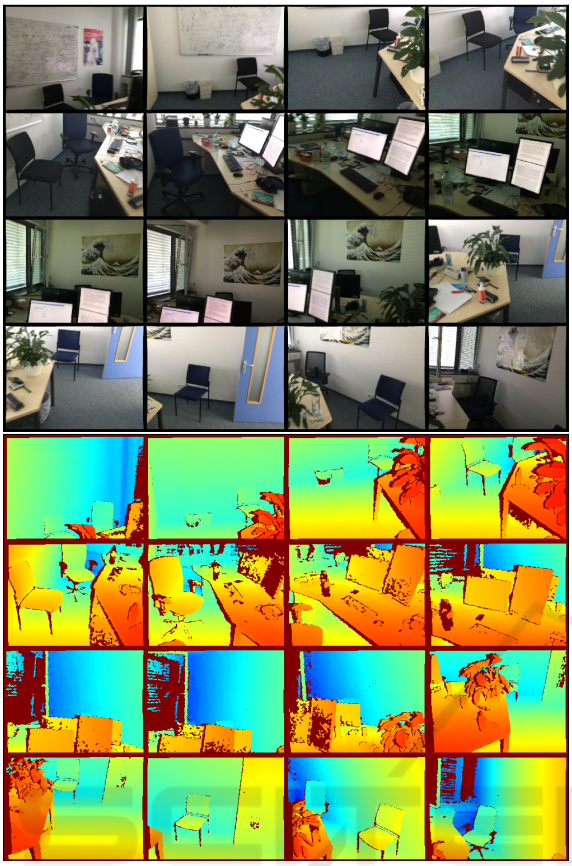

Figure 3: Exemplary RGB-D images from one location in ScanNet dataset. RGB images on the top and corresponding depth maps at the bottom.

exemplary RGB-D images from one location. On the top, there are RGB images and on the bottom, corresponding depth maps. Point clouds are constructed from RGB-D images in the dataset by backprojecting each pixel in the RGB image using known camera intrinsics and depth. An example of a reconstructed point clouds is shown in Fig. 4. The point clouds are fed into an embedding network to compute a global descriptor.

Deep distance metric learning methods, such as methods based on triplet networks, require information on semantically similar and dissimilar data points. In our case, similar elements are point clouds representing largely overlapping parts of the scene, and dissimilar elements are point clouds representing different places. Such information is not available in the ScanNet dataset and needs to be computed to prepare sufficiently large training dataset. Using solely camera pose ground truth to asses visible scene overlap of two RGB-D images is problematic. Spatially distant cameras may show the same place from different angles, and the corresponding point clouds

Figure 4: An exemplary point cloud generated from one RGB-D image.

should be considered similar. To solve this problem we developed] an efficient method to compute a view overlap between two point clouds. It is based on calculating a percentage of points co-visible on RGB-D images that are used to construct point clouds.

To find pairs of similar and dissimilar point clouds for network training, we randomly sample 500 RGB-D images from each location. For each sampled image, we compute its view overlap with a different set of 500 images sampled from the same location. This generates view overlap information for $500 \cdot 500 = 250$ thousand pairs in each location. To construct the validation and test set, we sample 100 RGB-D images from each scene and compute view overlap between each pair of sampled images. This produces view overlap data for $100 \cdot 100/2 = 5$ thousand pairs. View overlap between two RGB-D images is calculated as a percentage of points co-visible on both images. This is done as follows. We sample a set of 500 random points in the first image. Using depth data, camera intrinsics and relative pose between two images (given in the dataset groundtruth) we re-project a point $\mathbf{p}$ in the first image onto the second image, obtaining a point $\mathbf{p}'$. If the point falls outside the second image area then it's not co-visible on both images. Otherwise, we re-projected point $\mathbf{p}'$ back onto the first image, obtaining a point $\mathbf{p}''$. If the Euclidean distance between an original point $\mathbf{p}$ and its re-projection $\mathbf{p}''$ is below a given threshold (4 pixels in our implementation) we consider the point $\mathbf{p}$ co-visible on both images. To make our view overlap measure symmetric, we compute it in two directions: first by projecting points from the first image into the second and from the second to the first; and then by projecting points from the second image to the first and back to the second. The final overlap measure between two RGB-D images, is taken as a minimum of these two results. Such overlap measure can be effectively computed using a vectorized implementation operating on an array

of points sampled in one image. Results on the view overlap calculations can bee seen in Fig. 6.

For network training and evaluation purposes we consider two point clouds similar, if they are constructed from two RGB-D images taken at the same location and having a view overlap above the threshold (30% in our implementation). Otherwise, two point clouds are considered dissimilar.

Network Training. The embedding network is trained using a stochastic gradient descent approach with a triplet loss (Hermans et al., 2017). Mini-batches contain triplets consisting of an anchor, a positive and a negative element. A positive element is a point cloud similar to the anchor cloud, with the view overlap above the threshold (30% in our implementation). A negative element is a point cloud showing a different place than an anchor. A randomly chosen negative element would often depict a scene that is very different, both in appearance and geometry, from an anchor element. In the presence of such easy cases, the network will quickly learn how to produce sufficiently different embeddings and the training will stagnate. To improve effectiveness of the training process we use *batch hard* negative mining scheme to construct triplets, as proposed in (Hermans et al., 2017). Each triplet is constructed using the hardest negative example found within a batch. The hardest negative example for each anchor is a dissimilar point cloud that has the closest embedding to the anchor embedding, computed using current network weights.

We use a popular triplet loss formulation as defined in (Hermans et al., 2017):

$$L(a_i, p_i, n_i) = \max\{d(a_i, p_i) - d(a_i, n_i) + margin, 0\},$$

where $d(x,y) = ||x-y||_2$ is an Euclidean distance between embeddings x and y; a_i, p_i, n_i are embeddings of an anchor, a positive and a negative point cloud in i-th triplet and *margin* is set to 0.4. The loss function is minimized using a stochastic gradient descent approach with Adam optimizer. We train the the network for 16 epochs with an initial learning rate set to 0.001, which is decreased to 0.0001 after eight epochs.

To increase variability of the training data and decrease overfitting, we apply on-the-fly data augmentation. It includes photometric distortions, random rotation, translation and resizing of the point cloud. Additionally we adapted a random erasing augmentation (Zhong et al., 2017) to operate on 3D point clouds. A fronto-parallel cuboid with a random size and position is randomly generated, and all points lying within the cuboid are removed.

4 EXPERIMENTAL RESULTS

This section describes experimental evaluation results of global point cloud descriptors performance for place recognition purposes in indoor environment. Evaluation is done on a subset of ScanNet dataset contains 253 thousand point clouds gathered at 176 locations that are different from locations used for training. The evaluation is done using the following procedure. First, the test set is split randomly into the query set, containing 10% of elements, and the database containing remaining 90%. Then, global descriptors of all point clouds are computed using a trained embedding network. Finally, for each point cloud in the query set, we search for $k = 20$ most similar point clouds in the database. This is done by finding point clouds in the database with the closest, in Euclidean distance sense, descriptor to the descriptor of the query point cloud. If the view overlap, calculated using a procedure detailed in the previous section, between the query point cloud and retrieved point cloud is above the threshold (we set threshold to 30%) we declare a match (true positive). Otherwise we declare a false positive. We use Precision@k as the evaluation metric, averaged over all query set elements. Precision@k is defined as the percentage of correctly retrieved elements (true positives) within the first k elements. Fig. 6 visualizes point cloud retrieval results using descriptors calculated with MinkNetVLAD network trained with geometry and appearance modality.

Fig. 5 shows performance of PointNetVLAD (Angelina Uy and Hee Lee, 2018) and MinkNetVLAD network architectures trained using three different modalities: both geometry (XYZ) and appearance (RGB); only appearance; and only geometry. Numerical results are shown in Tab. 2.

Table 2: Evaluation of MinkNetVLAD and PointNet network architectures and different modalities on point cloud retrieval task.

Base network	Modality	Precision @1	Precision @10
PointNetVLAD	geometry	0.855	0.383
PointNetVLAD	RGB	0.979	0.666
PointNetVLAD	RGB+geom.	0.986	0.681
MinkNetVLAD	geometry	0.939	0.542
MinkNetVLAD	RGB	0.976	0.662
MinkNetVLAD	RGB+geom.	**0.992**	**0.670**

When using only geometry modality, MinkNetVLAD outperforms PointNetVLAD by a large margin. The former has 0.939 (0.542) and the latter 0.855 (0.383) precision@1 (precision@10).

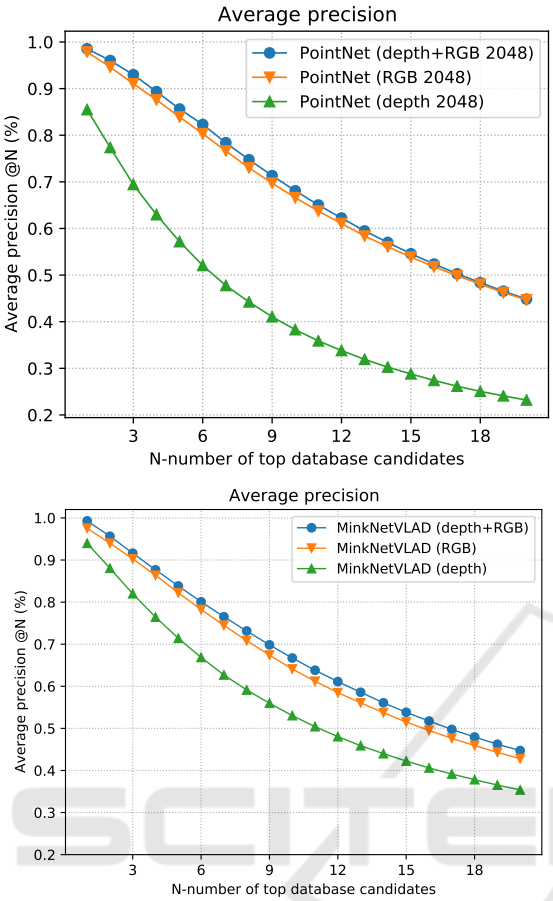

Figure 5: Point cloud retrieval results using PointNetVLAD architectures and different modalities.

As mentioned earlier, PointNet-based architecture is not well suited to capture local geometric structures which adversely affects the quality of the resultant descriptor. MinkNetVLAD network can extract more discriminative features using 3D convolutions and sparse voxelized representation. When using RGB modality, both architectures show similar performance: PointNetVLAD achieves 0.979 (0.666) precision@1 (precision@10) and MinkNetVLAD 0.976 (0.662). The results using solely scene appearance (RGB modality) are significantly better, compared to geometry. This can be understood, as appearance of scenes in our evaluation dataset exhibits limited variability. Image acquisition conditions are not affected by environmental factors, lighting is usually constant, and only differences are to a viewpoint change. Fusing two modalities, appearance and geometry, improves discriminability of the resultant global descriptor. For MinkNetVLAD architecture, precision@1 increases from 0.976 (RGB only) to 0.992 (RGB+geometry). For PointNetVLAD, it improves from 0.979 to 0.986. However, it must be noted that the improvement is moderate, 1.6 p.p. in the first case, and 0.7 p.p. in the second case.

MinkNetVLAD architecture consistently outperforms PointNetVLAD (Angelina Uy and Hee Lee, 2018) method. It performs significantly better using geometry only (0.939 vs 0.855 precision@1); comparable using scene appearance only (0.976 vs 0.979 prevision@1) and slightly better using both modalities (0.992 vs 0.986).

5 CONCLUSIONS

Our experiments show, that in indoor environments, scene appearance is much more informative that scene geometry for place recognition purposes. Both evaluated architectures trained with RGB data yielded significantly better results compared to training using solely scene geometry modality. Fusing two modalities, scene appearance and geometry, improved discriminativity of the resultant global descriptor by a small factor (0.6-1.2 p.p.). RGB component dominates over geometry, and there's little gain from using both of them in indoor environment. When using only scene geometry, MinkNetVLAD architecture, based on sparse voxelized representation and using 3D convolutions, yields significantly better results compared to PointNetVLAD (Angelina Uy and Hee Lee, 2018) method, based on PointNet (Qi et al., 2017a) architecture.

For future work, we plan to examine more sophisticated approaches for fusing scene appearance and geometry modalities. One idea is to use a pretrained 2D convolutional network to extract features from RGB image and link them with 3D points, before feeding to the global descriptor extraction network.

ACKNOWLEDGEMENTS

The project was funded by POB Research Centre for Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative Program - Research University (ID-UB).

Figure 6: Visualization of point cloud retrieval results using embeddings calculated with MinkNetVLAD architecture and RGB+geometry modality. Each row shows a query RGB point cloud (on the left) and its five nearest neighbours retrieved from the database (on the right). *Distance* is an Euclidean distance between a query and a database point cloud embedding. Different scene names correspond to different locations.

REFERENCES

Angelina Uy, M. and Hee Lee, G. (2018). Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4470–4479.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE*

conference on computer vision and pattern recognition, pages 5297–5307.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1994). Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744.

Chen, W., Chen, X., Zhang, J., and Huang, K. (2017). Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412.

Choy, C., Gwak, J., and Savarese, S. (2019a). 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084.

Choy, C., Park, J., and Koltun, V. (2019b). Fully convolutional geometric features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8958–8966.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

Hermans, A., Beyer, L., and Leibe, B. (2017). In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

Komorowski, J. (2020). Minkloc3d: Point cloud based large-scale place recognition. *arXiv preprint arXiv:2011.04530*.

Lee, J.-E., Jin, R., and Jain, A. K. (2008). Rank-based distance metric learning: An application to image retrieval. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Liu, Z., Zhou, S., Suo, C., Yin, P., Chen, W., Wang, H., Li, H., and Liu, Y.-H. (2019). Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2831–2840.

Lu, J., Hu, J., and Zhou, J. (2017). Deep metric learning for visual understanding: An overview of recent advances. *IEEE Signal Processing Magazine*, 34(6):76–84.

Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953.

Wang, Z. and Jia, K. (2019). Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1742–1749. IEEE.

Wu, C.-Y., Manmatha, R., Smola, A. J., and Krahenbuhl, P. (2017). Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848.

Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., and Funkhouser, T. (2017). 3dmatch: Learning the matching of local 3d geometry in range scans. In *CVPR*, volume 1.

Zhang, W. and Xiao, C. (2019). Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12436–12445.

Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. (2017). Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*.