Improving 3D Semantic Segmentation with Twin-Representation Networks

Fabian Duerr

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
fabian.duerr@audi.de

Abstract

The growing importance of 3d scene understanding and interpretation is inherently connected to the rise of autonomous driving and robotics. Semantic segmentation of 3d point clouds is a key enabler for this task, providing geometric information enhanced with semantics. To use Convolutional Neural Networks, a proper representation of the point clouds must be chosen. Various representations have been proposed, with different advantages and disadvantages. In this work, we present a twin-representation architecture, which is composed of a 3d point-based and a 2d range image branch, to efficiently extract and refine point-wise features, supported by strong context information. Additionally, a feature propagation strategy is proposed to connect both branches. The approach is evaluated on the challenging SemanticKITTI dataset [2] and considerably outperforms the baseline overall as well as for every individual class. Especially the predictions for distant points are significantly improved.

1 Introduction

Understanding a 3d environment is one of the key challenges for autonomous vehicles or robots. For this task of 3d scene understanding and interpretation,

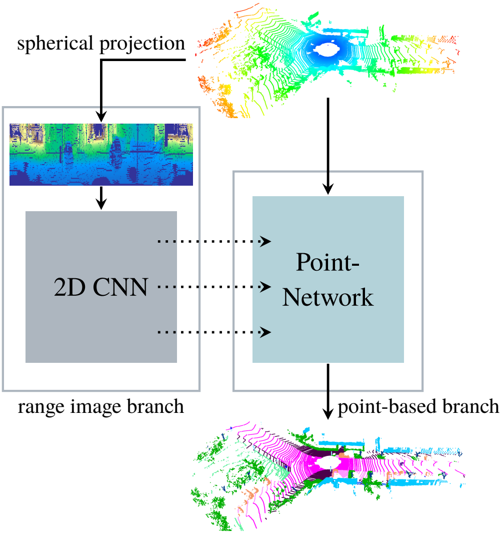

Figure 1.1: The proposed twin-representation architecture, which exploits two different point cloud representations. The 3d point-based branch extracts and refines point-wise features while the 2d range image branch efficiently aggregates context information.

semantic segmentation of images or point clouds, which assigns a class label to every pixel or 3d point, provides valuable information.

The combination of geometric and semantic information provided by 3d semantic segmentation is particularly valuable. To tackle this task with established deep learning approaches, like Convolutional Neural Networks (CNNs), a proper representation of point clouds has to be chosen, to allow their application. Point-based approaches [12, 19] operate directly on the raw point clouds while projection based methods [11, 18] transform them into a regular space, like 2d or 3d grid, to enable convolution operations.

Recently, the combination of voxel and point-based representation showed promising results [9, 17], by exploiting the advantages of both representations. In general, projection based methods, like voxel grids, efficiently aggregate neighborhood information because of the regularity of their data representation. The projection however requires a discretization in most cases, where the choice of resolution is a trade-off between loss of information and memory as well as computational costs. Point-based approaches on the other hand efficiently operate on the original point cloud resolution without information loss, but the aggregation of neighborhood information and context is expensive. Because of these complementary properties, the combination of both representations offers a great potential.

This work follows the general idea of combining projection and point-based representations but focuses on the more efficient range image representation. Therefore, we present a twin-representation architecture, which combines a 2d range image and a 3d point-based branch, see Fig. 1.1. The 2d branch works on range images resulting from a spherical projection and enables the efficient aggregation of local neighborhoods and context. The point-based branch computes point-wise features while preserving the original resolution and is supported by the aggregated information from the 2d branch, to predict the final 3d semantic segmentation. To summarize, our contributions are twofold:

- A twin-representation architecture composed of a 2d range image and 3d point branch, which preserves point-wise features while efficiently aggregating local context.
- A feature propagation strategy for 2d → 3d feature transformation.

2 Related work

The growing importance of autonomous vehicles and robots also raised the importance of 3d semantic segmentation. Supported by an increasing number of available indoor [1] and outdoor datasets [2, 23, 3] considerable progress has recently been achieved. A crucial and recurring question when addressing 3d semantic segmentation with CNNs is the representation of 3d point clouds. Many different representations have been proposed in recent works, which can generally be grouped into two categories.

Point-based methods, like PointNet [12] and its successor PointNet++ [13], directly process the raw point clouds. PointNet applies a shared multilayer perceptron (MLP) pointwise and a symmetric operation performs global feature aggregation. While this is very efficient, a single global feature aggregation greatly limits the ability to capture spatial relations. Therefore, PointNet++ was proposed, which applies individual PointNets to local regions and aggregates them in a hierarchical fashion. While being one of the first approaches, many others [8, 7, 20, 19] followed.

Projection based methods can further be divided into subcategories based on the chosen regular space, like voxel grids [27, 18], permutohedral lattice [15, 16] or bird's eye view [26]. Another possibility is a spherical projection, which results in a so called range image. SqueezeSeg [21] was one of the first approaches building upon range images for a road segmentation task. Improved versions were released in [22] and [24]. The latter targets full semantic segmentation and proposed Spatially-Adaptive Convolutions (SAC) to deal with spatially-varying feature distributions, induced by the spherical projection. Another approach is RangeNet++ [11], which builds upon the DarkNet53 backbone [14] and presented a label projection strategy from range image space to 3d point clouds. [10] proposed LaserNet, based on deep layer aggregation [25], for 3d object detection, while one intermediate result is a semantic segmentation.

Recently, first attempts were made to exploit the advantages of multiple representations in one architecture. PVCNN [9] combined a shared MLP for point-wise feature extraction with 3d convolutions in voxel space for context aggregation. It is therefore able to extract point features in full resolution while extracting and aggregating neighborhood information in a coarse voxel space. It's successor SPVCNN [17] replaced the dense 3d convolutions by its spare counterparts, which allows for a higher voxel resolution and therefore more preserved information. While sparse 3d convolutions already improve the performance and possible resolution, 2d convolutions are still more efficient with similar or less information loss. Therefore, our proposed segmentation architecture combines a 2d range image branch with a point-based branch and relies on a novel feature propagation strategy from 2d range image space to 3d point clouds.

3 Twin-Representation Network

The goal of the presented approach is the exploitation of two different input representations, range images and 3d point clouds, to improve 3d semantic segmentation. Fig. 3.1 shows the overall architecture, which consists of three main components. The range image backbone provides 2d feature maps of different stages and resolution while a feature propagation step transforms the 2d features back to their corresponding 3d points. Thereby, both components together efficiently provide aggregated neighborhood and context information for each individual point. These are used by the third component, a 3d point network. In the following, we provide details for each individual component.

Range Image Backbone Range images and the corresponding spherical projection are motivated by a lidar's internal structure, which usually consists of a vertical stack of lasers spinning around their vertical axis. As a result, the measurements can be described by an azimuth angle ϕ, an elevation angle θ and measured distance r and intensity e. We follow [4] for the conversion of the point clouds to range images of shape $6 \times h \times w$, with channels r, x, y, z, e and an occupancy flag.

The chosen 2d network architecture is based on deep layer aggregation [25] and closely related to LaserNet [10]. We reduced the number of Residual Units [6] in the first two feature extractors to four and five. Additionally, the downsampling in the first feature extractor was omitted. The backbone provides 2d feature maps of three different stages, see Fig. 3.1. Because of the underlying deep layer aggregation, all three stages are at full resolution while still representing features of different context stages. Full resolution feature maps have the advantage, that a distinctive feature vector can be provided for every 3d point, expect for colliding points [4].

Feature Propagation The fusion of feature maps $\boldsymbol{F}$ and point features $\boldsymbol{f}^{\text{point}}$ requires a transformation of 2d features back to their corresponding 3d points $\boldsymbol{p}$. One possible strategy is the assignment of a 2d feature to the 3d point belonging

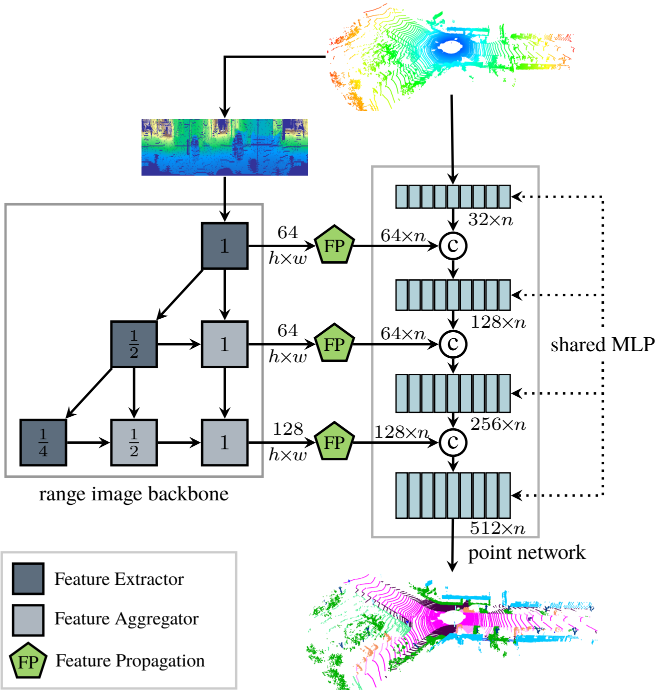

Figure 3.1: The proposed segmentation architecture. The range image backbone efficiently aggregates 2d context information and provides them via a feature propagation step to the point network, which itself computes point-wise features and combines them with the provided context information.

to its pixel position

$$\boldsymbol{f}_{\boldsymbol{p}}^{\text{twin}} = \boldsymbol{f}_{\boldsymbol{p}}^{\text{point}} \oplus \boldsymbol{F}[u_{\boldsymbol{p}}, v_{\boldsymbol{p}}], \tag{3.1}$$

where u and v are the 2d coordinates resulting from the spherical projection and $\oplus$ denotes concatenation. One possible disadvantage of this strategy occurs for colliding points, because all of them get the same feature vector assigned, even if they are far apart in 3d. For solving the related challenge of label back-projection, [11] proposed a KNN-based approach. The 3d labels are chosen by a majority vote among the approximated k-nearest-neighbors, weighted by their euclidean distance. Although we want to back-project intermediate feature vectors, instead of simple labels, we yet pick up the general idea of using the 2d neighborhood of a pixel as nearest-neighbor candidates. Instead of a majority vote, we compute the weighted sum over the feature vectors:

$$\boldsymbol{f}_{\boldsymbol{p}}^{\,\text{twin}} = \boldsymbol{f}_{\boldsymbol{p}}^{\,\text{point}} \oplus \sum_{\tilde{\boldsymbol{p}} \in \mathcal{N}_k(\boldsymbol{p})} w_{(\boldsymbol{p},\tilde{\boldsymbol{p}})} \cdot \boldsymbol{F}[u_{\tilde{\boldsymbol{p}}}, v_{\tilde{\boldsymbol{p}}}], \quad w_{(\boldsymbol{p},\tilde{\boldsymbol{p}})} = \frac{1}{||\boldsymbol{p} - \tilde{\boldsymbol{p}}||^2}, \tag{3.2}$$

with $\mathcal{N}_k$ being the $k \times k$-neighborhood of $\boldsymbol{p}$. Therefore, features are aggregated based on the point distribution in 3d space.

Point Network Motivated by the original PointNet, the point network stacks multiple shared MLPs to extract and refine point features. After each stage, the propagated features from the 2d branch, which provide the aggregated neighborhood and context information, are concatenated with the point features, see Eq. 3.1 and 3.2. The point network operates on the original point cloud resolution over all stages, so no information are lost. The shared MLPs are implemented by 1×1-convolutions and their feature channel depth increases with network depth.

4 Experiments

4.1 SemanticKITTI

We evaluate our approach on the challenging, large-scale SemanticKITTI dataset [2, 5], which provides point-wise annotations for $360°$-Velodyne-HDL-64E scans. The annotations contain 19 classes for the single scan benchmark. 22 labeled sequences of varying length, recorded at $10\,\text{Hz}$, add up to just over $43,000$ scans. Sequences 0-10 are provided with labels for training and validation while sequences 11-21 without published labels form the test split. The official recommendation is to use sequence 08 for validation, but we use a larger validation split for our ablation studies, consisting of sequences $02, 06, 10$, for more significant conclusions. We follow the official evaluation metric and report the mean Intersection-over-Union (mIoU).

Table 4.1: Observed improvements when adding the point network (PN) and KNN feature propagation, compared to a single range image backbone (RB).

RB	PN	KNN	mIoU (%)
✓			51.4
✓	✓		53.7
✓	✓	✓	$\mathbf{54.8}$

4.2 Implementation Details

The implementation is based on PyTorch and all experiments are trained in mixed precision mode using distributed data parallel training on four Tesla V100 GPUs.

Class-balanced cross entropy loss is optimized by Adam with a weight decay of 0.0005 for $100k$ iterations. The learning rate starts with 0.001 and is then multiplied with $e^{-5 \cdot 10^{-5} \cdot i}$ after every iteration i. To counteract overfitting, we randomly flip the range images horizontally with a probability of $p = 0.5$ and rely on random crops of size 64×1024 during training.

First, solely the range image backbone is trained with a batch size of 32. Building upon this, we train the entire network, also with a batch size of 32.

4.3 Results

Our evaluation starts with an investigation of the influence of the individual components, with the results being depicted in Table 4.1. The range image backbone, as a common 2d range image approach, is our baseline and achieves a mIoU of 51.4%. The presented twin-representation architecture, which is composed of the backbone and a point network, significantly outperforms the baseline by $+2.3\%$. Replacing the simple propagation strategy by the proposed KNN feature propagation further improves the results to 54.8%.

In the next step, we investigate the results restricted to the distance intervals $0-20$m, $20-40$m and $>$40m. Table 4.2 shows an overall performance increase of our approach for all chosen intervals. However, especially the results for distant

Table 4.2: Comparison of the mIoU (%) for different distance intervals.

Approach	0−20m	20−40m	>40m
RB	52.7	43.6	33.9
RB+PN	54.9	46.7	36.7
RB+PN+KNN	**55.6**	**47.2**	**39.2**

points are significantly improved by $+5.3\%$, which is particularly challenging because of the declining point density with increasing distance. For the other two intervals, a smaller but still considerable improvement of $+2.9\%$ and $+3.6\%$ is achieved.

Finally, we evaluate the results for the individual classes. Looking at Table 4.3, especially the classes motorcycle, truck, person and bicyclist experience a significant improvement by using the combination of range image backbone and point network. Likewise, the results for the classes car, other-vehicle, trunk and pole improved. In general, while no significant improvements for greater static classes can be observed, small classes greatly benefit from our approach. Adding KNN feature propagation further improves the results for most classes, without any bias regarding a special group of classes. One class to emphasize however is motorcyclist, which is improved by $+11.2\%$.

5 Conclusion

In this work, we presented a twin-representation architecture to combine a 3d point-based branch with a 2d range image branch, to improve 3d semantic segmentation. While the first computes and refines point-wise features over multiple stages, the latter supports the 3d branch with an efficient aggregation of neighborhood and context information. A feature propagation step connects both branches. The evaluation showed a significant overall improvement, considering that our approach outperforms the baseline for every individual class. Additionally, especially distant points experience a significant improvement. To summarize, combining the two input representations enables the exploita-

Table 4.3: Overview over the improvements for the individual classes. The presented approach outperforms the baseline for every single class. Values are given as IoU (%).

Approach	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist
RB	51.4	83.9	31.7	35.9	33.4	31.1	45.4	23.2	2.4
RB+PN	53.7	85.0	**32.3**	44.1	42.5	34.7	**53.6**	**29.0**	3.1
RB+PN+KNN	**54.8**	**86.5**	29.7	**45.9**	**44.4**	**36.2**	53.0	28.4	**14.3**

Approach	road	sidewalk	parking	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic sign
RB	91.2	80.2	58.8	8.6	76.7	58.0	82.8	63.8	70.1	49.5	49.6
RB+PN	90.8	80.0	59.0	**8.8**	76.5	58.3	82.9	**66.2**	70.9	**52.0**	50.4
RB+PN+KNN	**91.3**	**80.5**	**61.0**	7.3	**78.1**	**60.2**	**83.4**	65.6	**71.7**	49.3	**52.8**

tion of their different strengths, which considerably improves 3d semantic segmentation.

References

[1] Iro Armeni et al. “3D Semantic Parsing of Large-Scale Indoor Spaces”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[2] Jens Behley et al. “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2019.

[3] Holger Caesar et al. “nuScenes: A multimodal dataset for autonomous driving”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[4] Fabian Duerr et al. “Iterative Deep Fusion for 3D Semantic Segmentation”. In: *IEEE International Conference on Robotic Computing (IRC)*. 2020.

[5] 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 (CVPR)*. 2012.

[6] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[7] Qingyong Hu et al. “RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[8] Yangyan Li et al. “PointCNN: Convolution On $\mathcal{X}$-Transformed Points”. In: *Advances in Neural Information Processing Systems*. 2018.

[9] Zhijian Liu et al. “Point-Voxel CNN for Efficient 3D Deep Learning”. In: *Advances in Neural Information Processing Systems*. 2019.

[10] Gregory P. Meyer et al. “LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[11] A. Milioto et al. “RangeNet++: Fast and Accurate LiDAR Semantic Segmentation”. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*. 2019.

[12] Charles Ruizhongtai Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[13] Charles Ruizhongtai Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: *Advances in Neural Information Processing Systems*. 2017.

[14] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *ArXiv*. Vol. abs/1804.02767. 2018.

[15] Radu Alexandru Rosu et al. “LatticeNet: Fast Point Cloud Segmentation Using Permutohedral Lattices”. In: *Robotics Science and Systems (RSS)*. 2020.

[16] Hang Su et al. “SPLATNet: Sparse Lattice Networks for Point Cloud Processing”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[17] Haotian Tang et al. “Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution”. In: *IEEE European Conference on Computer Vision (ICCV)*. 2020.

[18] Lyne P. Tchapmi et al. “SEGCloud: Semantic Segmentation of 3D Point Clouds”. In: *International Conference on 3D Vision (3DV)*. 2017.

[19] Hugues Thomas et al. “KPConv: Flexible and Deformable Convolution for Point Clouds”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2019.

[20] Shenlong Wang et al. “Deep Parametric Continuous Convolutional Neural Networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[21] Bichen Wu et al. “SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017.

[22] Bichen Wu et al. “SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018.

[23] Jun Xie et al. “Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[24] Chenfeng Xu et al. “SqueezeSegV3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation”. In: *European Conference on Computer Vision (ECCV)*. 2020.

[25] Fisher Yu et al. “Deep Layer Aggregation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[26] Chris Zhang, Wenjie Luo, and Raquel Urtasun. “Efficient Convolutions for Real-Time Semantic Segmentation of 3D Point Clouds”. In: *International Conference on 3D Vision (3DV)*. 2018.

[27] Yang Zhang et al. “PolarNet: An Improved Grid Representation for Online LiDAR Point Clouds Semantic Segmentation”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.