Iterative Deep Fusion for 3D Semantic Segmentation

Fabian Duerr*‡, Hendrik Weigel*, Mirko Maehlisch*, Jürgen Beyerer†‡

* AUDI AG
Department Sensor Fusion and MapLearning for Automated Driving
Ingolstadt, Germany
firstname.lastname@audi.de

† Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
Fraunhofer Center for Machine Learning
Karlsruhe, Germany
juergen.beyerer@iosb.fraunhofer.de

‡ Vision and Fusion Lab
Karlsruhe Institute of Technology
KIT
Karlsruhe, Germany

***Abstract*—Understanding and interpreting a scene is a key task of environment perception for autonomous driving, which is why autonomous vehicles are equipped with a wide range of different sensors. Semantic segmentation of sensor data provides valuable information for this task and is often seen as key enabler. In this paper, we are presenting a deep learning approach for 3D semantic segmentation of lidar point clouds. The proposed architecture uses a range view representation of 3D point clouds and additionally exploits camera features to increase accuracy and robustness. In contrast to other approaches, which fuse lidar and camera feature maps once, we fuse them iteratively and at different scales inside our network architecture. We demonstrate the benefits of the presented iterative deep fusion approach over single fusion approaches on a large benchmark dataset. Our evaluation shows considerable improvements, resulting from the additional use of camera features. Furthermore, our fusion strategy outperforms the current state-of-the-art strategy by a considerable margin. Despite the use of camera features, the presented approach is also trainable solely with point cloud labels.**

I. Introduction

One of the key challenges of autonomous driving is the understanding of the vehicle's environment. Therefore, autonomous vehicles are equipped with a wide range of sensor modalities, usually including camera, lidar, radar and ultrasonic sensors. With different complementary sensors available, shortcomings of an individual sensor type can be compensated by other sensor types, increasing accuracy and robustness. In this work, we focus on camera and lidar sensors.

Understanding and interpreting a scene is a key task of environment perception for autonomous driving, which makes semantic segmentation of sensor data valuable. For camera images, assigning a class label to every image pixel has been addressed very successfully with Convolutional Neural Networks (CNNs) over the past years, achieving impressive results on road and urban scenes [1]. When dealing with 3D lidar point clouds however, the first challenge is a proper representation, enabling the application of CNNs. One possibility is the lidar's native range view, which has shown promising results [2], [3]. This allows the application of established image segmentation architectures.

Having different sensors available with an overlapping field of view, allows for approaches that fuse the data of different sensors to improve the robustness and overall accuracy. When addressing the fusion of camera and lidar data, some challenges arise. One is a substantial difference in their resolution and another is their considerable difference in measurements. While a camera observes brightness values resulting in an image, a lidar measures the distance to its environment, generating a sparse 3D point cloud. Additionally, different fusion strategies must be considered. Following [4], possible strategies are the fusion of the sensor data (early fusion), the fusion of the predictions for each sensor (late fusion) or the fusion of the feature maps inside a CNN (deep fusion). In this work, we propose a deep fusion approach, depicted in Fig. 1, which is applied to the range view representation of 3D point clouds. It exploits camera and lidar data for semantic segmentation of lidar point clouds. The contributions of this work are twofold:

- First, we propose a new fusion module, which transforms camera and lidar feature maps of arbitrary scale into a common space and fuses them afterwards.
- Second, we propose a fusion architecture building upon the fusion modules and applying them iteratively to camera and lidar feature maps. Instead of fusing camera

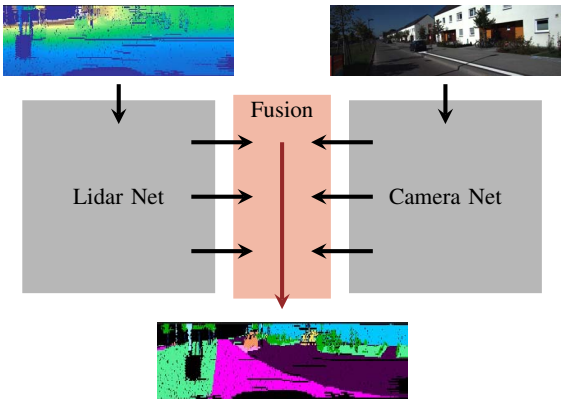


Fig. 1. Basic idea. Feature maps of different scales are iteratively fused to improve the semantic segmentation of a point cloud.

978-1-7281-5237-0/20/$31.00 ©2020 IEEE
DOI 10.1109/IRC.2020.00067

and lidar feature maps once and at one scale, we are fusing the feature maps multiple times at different scales.

As a result, fused feature maps at a shallower level are refined while propagated through the consecutive fusion blocks. Thereby, more information of the camera image is used to support the 3D semantic segmentation, improving overall results.

II. Related Work

A. 2D Semantic Segmentation

The success of deep learning applied for scene parsing and semantic segmentation [5]–[7] is closely related to its success in classical image classification [8]–[10]. One widely used approach are Fully Convolutional Neural Networks (FCNN) [5], which calculate a pixel-wise prediction for a given image in an end-to-end fashion. [5] replaced the fully connected layers of common classification architectures with 1x1-convolutions, thereby replacing the original image classification with a pixel-wise classification.

One main challenge, recent works have focused on, is the loss of spatial resolution while aggregating information. It is of great importance to capture the global context of a scene as well as fine local structures. DeepLabv3 [11], [12] addresses this by 'atrous' convolutions, which increase the size of the receptive fields without reducing resolution or increasing filter sizes. 'Atrous' convolutions with different rates are employed in parallel to exploit context at different scales. In [13], an aggregation architecture is presented, which the authors call deep layer aggregation (DLA), also targeting the challenge of extracting meaningful semantic features while preserving spatial information. PSPNet [14] combines local and global context by a pyramid pooling module, which aggregates the global context at different scales and appends it to the original feature maps. OcNet [15] adapts the idea of the pyramid pooling module and multiscale 'atrous' convolutions by introducing an object context module, which exploits object context at different scales, instead of spatial context.

B. 3D Semantic Segmentation

When addressing semantic segmentation of 3D point clouds with CNNs, the first thing to consider is the representation of the point clouds. In recent works, multiple different representations are proposed. PointNet [16] uses the raw and unstructured point clouds directly as input by applying point-wise 1x1-convolutions and a symmetric operation for feature aggregation. Because a single global feature aggregation limits the ability to capture spatial relations, the authors proposed PointNet++ [17], which applies individual PointNets to local regions and aggregates the resulting local features in a hierarchical fashion. [18] converts the point clouds into a voxel grid and applies a 3D-FCNN, followed by a Conditional Random Field (CRF) to refine the results. A bird's eye view (BEV) with the vertical axis as feature channel is used by [19] to retrieve a 2D representation of the point clouds. Having a 2D representation, they are using the U-Net architecture [6], known from image segmentation.

When working with point clouds generated by a lidar sensor, the range view is another possibility of representation. SqueezeSeg [20] was one of the first works using the range view for a segmentation task. Their goal was the segmentation of road objects, with an improved version released in [21]. Another approach is RangeNet++ [3], which employs the DarkNet53 backbone [22] for full semantic segmentation. [23] proposed LaserNet, which uses the range view as input for object detection, while one of their intermediate results is a semantic segmentation of the input. Their architecture is based on deep layer aggregation. Transforming the point cloud into a 2D representation and applying established 2D image segmentation architectures mostly outperforms other forms of representations while being faster. Therefore, our work also builds upon the range view as 2D representation.

C. Multimodal 3D Semantic Segmentation

Multi-sensor fusion architectures using camera and lidar mostly focus on object detection [2], [4], [24]–[26]. Only [2] also tackles the task of 3D semantic segmentation, using the range view as input representation. Camera image feature maps, extracted by three ResNet blocks [10], and extracted lidar feature maps from the range view are concatenated and passed to a LaserNet, which serves as DLA for the semantic segmentation. In contrast to applying early fusion and fusing the RGB values with the range view, this approach aggregates camera image information first, using the original usually much higher resolution of the camera image. This deep fusion allows for more information being preserved and exploited for the semantic segmentation of the lidar point cloud. While considerably improving the mean Intersection over Union over all classes (mIoU) on distant content $(+5.19)$, the overall improvements are rather small $(+0.25)$. We are also using deep layer aggregation and the full camera image resolution for deep fusion of camera and lidar. In contrast to [2], which fuses the features before applying their DLA network (LaserNet), we are applying a separate network to both, the lidar range view and the camera image but fuse both networks following iterative deep aggregation [13]. As a result, our deep fusion approach is able to aggregate and use more information of the camera for the semantic segmentation of the lidar point cloud and achieves better results.

III. Iterative Deep Fusion and Aggregation

In this section, we present our range view input representation, our fusion module and the network architecture, used for the fusion of the lidar and camera input.

A. Range View

Commonly used lidar sensors usually observe their environment by spinning a set of vertically stacked lasers around their vertical axis. The position of a laser in this stack is often referred to as channel, corresponding to an elevation angle. The Velodyne HDL-64E, used to record the SemanticKitti dataset [27], [28], has 64 channels, an azimuth resolution of approximately 0.17° and an elevation resolution of $1/3^\circ$ for

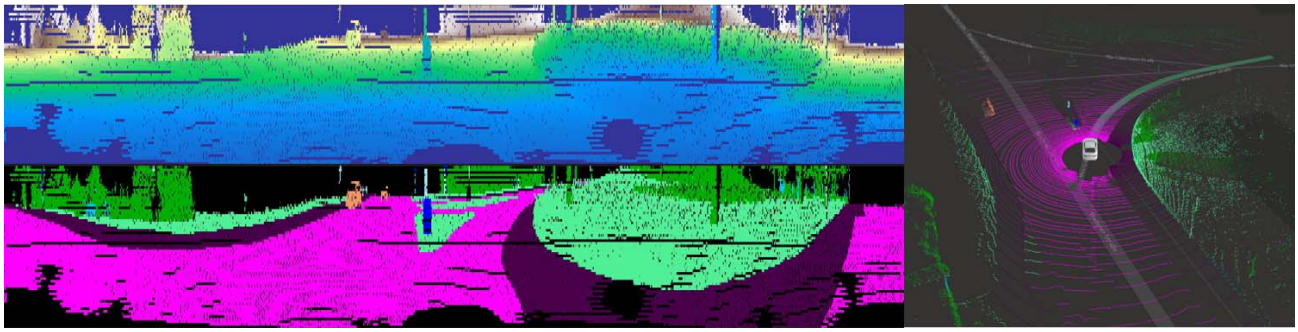


Fig. 2. Range view (top left) showing the lidar depth measurements and the projected ground truth (bottom left) of one sample of the SemanticKitti dataset [27]. The original point cloud is also shown on the right.

the upper and $^1/_2{}^\circ$ for the lower half of the lasers. The sensor provides measurements $\boldsymbol{o}_i = (c_i, \phi_i, r_i, e_i)$, with channel id c_i, azimuth angle ϕ_i, measured distance r_i and reflectance e_i. The corresponding 3D points are

$$\boldsymbol{p}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} r_i \cos(\theta_i) \cos(\phi_i) \\ r_i \cos(\theta_i) \sin(\phi_i) \\ r_i \sin(\theta_i) \end{pmatrix}, \tag{1}$$

omitting correction factors. The elevation angle θ_i is derived from the sensor configuration and the channel id c_i.

We generate a range view by mapping every point or measurement to a row and column index. Having measurements from a Velodyne HDL-64E, the row and column indices are calculated by using the channel as row index and discretizing the azimuth angle. If only the 3D points $\boldsymbol{p}_i$ are provided, the azimuth and elevation angle are given by

$$\phi_i = -\text{arctan2}(y_i, x_i) \quad \text{and} \quad \theta_i = \arcsin\left(\frac{z_i}{r_i}\right). \tag{2}$$

Finally, for a range view resolution of $h \times w$, image coordinates $\boldsymbol{u}_i^{\text{li}} = (u_i^{\text{li}}, v_i^{\text{li}})$ are

$$u_i^{\text{li}} = \begin{cases} \left\lfloor 0.5 \cdot h \cdot \frac{\theta_i - \theta_{\text{up}}}{\theta_{\text{mid}} - \theta_{\text{up}}} \right\rfloor & \theta_i \geq \theta_{\text{mid}} \\ \left\lfloor 0.5 \cdot h \cdot \left(1 + \frac{\theta_i - \theta_{\text{mid}}}{\theta_{\text{down}} - \theta_{\text{mid}}}\right) \right\rfloor & \theta_i < \theta_{\text{mid}} \end{cases}, \tag{3}$$

$$v_i^{\text{li}} = \left\lfloor 0.5 \cdot \left(1 + \frac{\phi_i}{\pi}\right) \cdot w \right\rfloor, \tag{4}$$

with a vertical field of view $\theta_{\text{fov}} = \theta_{\text{up}} - \theta_{\text{down}} = 2^\circ - (-24.8^\circ) = 26.8^\circ$ and the border angle between the two vertical resolutions $\theta_{\text{mid}} = -^{26}/_3{}^\circ$. Following this, we are mapping the input measurements r, e, x, y and z to the 2D range view, receiving a $5 \times h \times w$ input tensor $\mathbf{R}$. The depth channel (r) is visualized in Fig. 2.

Ego motion, uncertainty and non-uniformity of the angles can lead to mapping collisions. As a result, more than one point is mapped to the same range view pixel. This implies not only a loss of information but also missing predictions for the shadowed points. The latter isn't an issue for object detection, for semantic segmentation however, it has to be considered. Therefore, a post-processing step based on the labeled points is required to compute class labels for the shadowed points. Following the simplest one, we assign the same label to all measurements projected on the same range view pixel. Another approach is based on k-nearest neighbor [3]. We will investigate the post-processing step in future work. In this work, we are focusing on the feature fusion.

B. Feature Transformation and Fusion

A crucial part of our work is the feature fusion, which fuses the lidar and camera features. We are choosing the range view as our reference system and project camera features into it. The inverse projection, from lidar to camera, is mathematically given by the equation

$$\begin{pmatrix} u_i^{\text{cam}} \\ v_i^{\text{cam}} \\ 1 \end{pmatrix} = \mathbf{K} \cdot \mathbf{T}_{\text{li2cam}} \cdot \begin{pmatrix} \boldsymbol{p}_i \\ 1 \end{pmatrix}, \tag{5}$$

with the camera matrix $\mathbf{K}$ and transformation matrix from lidar to camera $\mathbf{T}_{\text{li2cam}}$. The calculated pixel indices define the correspondence between 3D points and camera pixels. For this correspondence being still valid after scaling the range view by β or the camera image by α, the following extensions are made

$${}^\alpha\boldsymbol{u}_i^{\text{cam}} = \begin{pmatrix} \lfloor u_i^{\text{cam}} \cdot \alpha \rfloor \\ \lfloor v_i^{\text{cam}} \cdot \alpha \rfloor \end{pmatrix} \quad \text{and} \quad {}^\beta\boldsymbol{u}_i^{\text{li}} = \begin{pmatrix} \lfloor u_i^{\text{li}} \cdot \beta \rfloor \\ \lfloor v_i^{\text{li}} \cdot \beta \rfloor \end{pmatrix}, \tag{6}$$

with $\alpha, \beta \in [0, 1]$. Given scalable projection indices, we are now able to project camera features $\mathbf{I}^\alpha$ into the range view $\mathbf{R}^\beta$, following

$$\mathbf{R}^\beta[{}^\beta\boldsymbol{u}_i^{\text{li}}] \coloneqq \mathbf{I}^\alpha[{}^\alpha\boldsymbol{u}_i^{\text{cam}}]. \tag{7}$$

This is a fixed, geometrically motivated mapping, considering only one location per 3D point in the camera feature maps. To capture more context and to compensate errors in the calibration, we apply a learnable function $F_{\boldsymbol{w}}$ before performing the fixed projection, resulting in

$$\mathbf{I}_F^\alpha = F_{\boldsymbol{w}}(\mathbf{I}^\alpha) \quad \text{and} \quad \mathbf{R}_{\boldsymbol{w}}^\beta[{}^\beta\boldsymbol{u}_i^{\text{li}}] \coloneqq \mathbf{I}_F^\alpha[{}^\alpha\boldsymbol{u}_i^{\text{cam}}]. \tag{8}$$

The fusion module shown in Fig. 3 builds upon this to implement the camera feature transformation. We are using two 3x3 convolutions followed by Batch Norm [29] and ReLu as learnable function $F_{\boldsymbol{w}}$. The projected camera features and the lidar features are concatenated and fused by ResNet blocks.

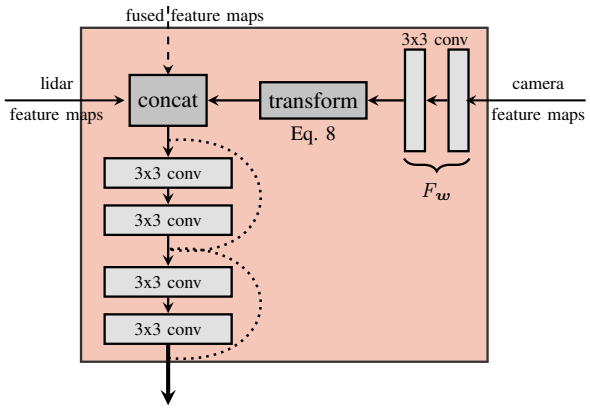


Fig. 3. The main building block of our architecture. The fusion module transforms the camera features into the lidar range view. Afterwards, lidar feature maps, camera feature maps and optionally fused features maps from the stage before are fused.

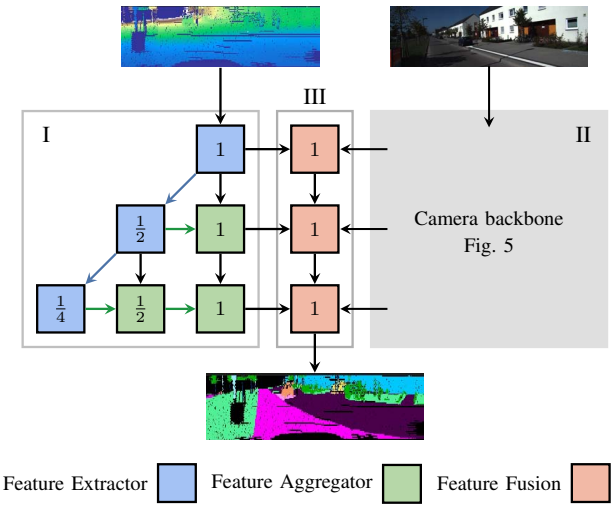
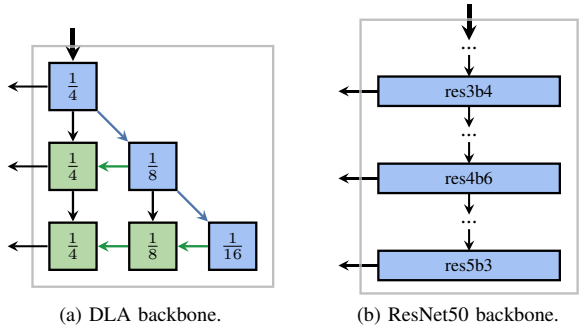


Fig. 4. Our proposed fusion architecture Fusion3DSeg, which fuses the lidar and camera features iteratively, following the idea of iterative deep aggregation [13]. The numbers indicate the size ratio of the output feature maps of the individual blocks with regard to the original network input. In contrast to the camera image, the lidar range view is only resampled horizontally, so its height stays fixed.



(a) DLA backbone. (b) ResNet50 backbone.

Fig. 5. The two backbones for camera feature extraction used in this work. The DLA network and its scale factors are shown in a), the layers of ResNet50, which are fused, are shown in b).

C. Network Architecture

Our proposed network architecture is shown in Fig. 4 and has three main components. First, a DLA network called Lidar3DSeg (I) for processing the lidar range view and computing lidar features. It follows the proposed architecture of [23], which itself is based on [13]. However, we reduced the number of ResNet blocks inside the feature extractors, resulting in 4, 5 and 6 blocks. We have also removed the downsampling in the first block. By using a DLA architecture, we ensure to efficiently aggregate multi-scale lidar features.
The second component is a backbone network (II) for extracting camera image features. Generally, an arbitrary image network can be used, which feats image features of different scale to the fusion blocks. We are evaluating two different backbones in this work, shown in Fig. 5. The DLA backbone, depicted in Fig. 5a, has the same architecture then Lidar3DSeg, except that we doubled the number of feature channels. We also downsample the camera images before applying the DLA network. The resolution of the camera images is much higher than of the lidar images, so the induced loss in spatial information is small, whereas the aggregated semantic information are considerably improved. We follow the ResNet architecture and downsample the camera image with a strided convolution and max pooling by a factor of four. This also decreases the run time and memory requirements. The second backbone is a ResNet50 [10], shown in Fig. 5b, with dilated convolutions instead of the fourth and fifth downsampling stride of 2.
The last component are the stacked fusion blocks (III), which apply the previously presented feature transformation and fusion. They follow the idea of a feature aggregator except that they transform and aggregate features of different sensors instead of different scales of one sensor. Thereby, the fused features are interatively refined as they are aggregated and propagated through the stacked fusion blocks. Because we use DLA for the Lidar-Net, we need no classical decoder. The fusion blocks work on the original resolution of the lidar range view and finally output the semantic segmentation.

IV. Experiments

A. SemanticKitti

We evaluate our approach on the SemanticKitti dataset [27], [28], which contains labels for 19 classes for the single scan benchmark. A total of 22 labeled sequences result in $43,552$ labeled scans. The official split allocates sequences 0–10 for training and sequences 11–21 for testing, for which the labels haven't been published. However, the official benchmark doesn't support the usage of the camera images, meaning for our evaluation, only the sequences with published labels 0–10 can be used. Therefore, we exclude sequences 02, 06 and 10 from training and validation and use them only in the end for testing. This results in 6963 frames for testing and $16,238$ for training and validation. We follow the official evaluation metric and report the mean Intersection-over-Union (mIoU). For our approach, only the lidar scan parts overlapping with

Approach	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic sign
Lidar3DSeg (*Ours*)	47.3	81.7	23.2	39.8	**42.0**	29.0	35.6	**11.8**	**2.4**	93.1	**56.1**	76.8	3.4	67.1	57.2	78.0	58.1	67.2	36.4	39.9
Fusion3DSeg (*Ours*)	**49.8**	**82.0**	**26.4**	**43.1**	37.8	**29.1**	**42.7**	11.0	0.3	**93.2**	55.9	77.0	0.4	**74.0**	**59.4**	**81.4**	**65.8**	**72.0**	**49.6**	**45.6**
LaserNet++ [2]	47.6	81.5	23.8	40.0	36.2	27.6	37.1	10.1	0.2	92.8	55.2	**77.2**	**4.1**	69.2	57.0	80.5	63.9	68.4	41.9	38.2

TABLE I
COMPARISON OF THE RESULTS OF THE PRESENTED DEEP FUSION ARCHITECTUR WITH LASERNET++ AND THE PURELY LIDAR BASED LIDAR3DSEG. VALUES ARE GIVEN AS IOU.

the camera's field of view in the front of the car are used for evaluation.

B. Implementation Details

Training starts with an initial learning rate of 0.0001, which is then multiplied during the training by $10^{\frac{-2 \cdot \text{it}}{\text{it}_{\max}}}$. Thereby, the learning rate exponentially decreases by $1/100$ during training. We train our network for $50k$ iterations using the Adam optimizer with a batch size of 32. To improve generalizability and reduce overfitting, we are using random crops of the whole $360°$ lidar scan for training the lidar net. Although the crop is random, it follows the constraint, that the overlapping field of view with the camera has to be fully inside the crop of size 44×1536. The fusion modules finally crop the resulting lidar feature maps exactly to the overlapping field of view. Additionally, we apply random flipping horizontally to the lidar and camera images. For the camera backbones we are evaluating a ResNet50, which is pretrained on ImageNet [8] and an untrained DLA.

To counteract the class imbalance, we are using the same class-balanced cross entropy loss as [3]. It is applied to the final output as well as to an auxiliary loss. The latter is used on the final feature map of the Lidar-Net. Following the proposed settings of PSPNet [14], we are weighting the auxiliary loss by 0.4.

For comparison, we also implement the approach proposed in [2] on the SemanticKitti dataset. We follow their architecture and training setup but use the same class-balanced cross entropy loss we used for our approach. Additionally, we use our input channels (r, x, y, z, e), whereby we experienced better results.

C. Quantitative Results

To evaluate our approach, we conduct different experiments and compare to different baselines. With exception of the ablation study, a pretrained ResNet50 is used as camera backbone in our experiments. We start with evaluating the improvements gained by the fusion of lidar and camera data compared to using only lidar data. The next step is an investigation of the benefits of the presented fusion strategy compared to another state-of-the-art strategy [2]. Finally, a different backbone and training strategy is evaluated, to show, that our fusion approach is unrelated to a specific camera backbone and is also trainable without camera image labels.

Approach (Backbone)	mIoU
Fusion3DSeg (ResNet50)	**49.8**
Fusion3DSeg (DLA)	49.4

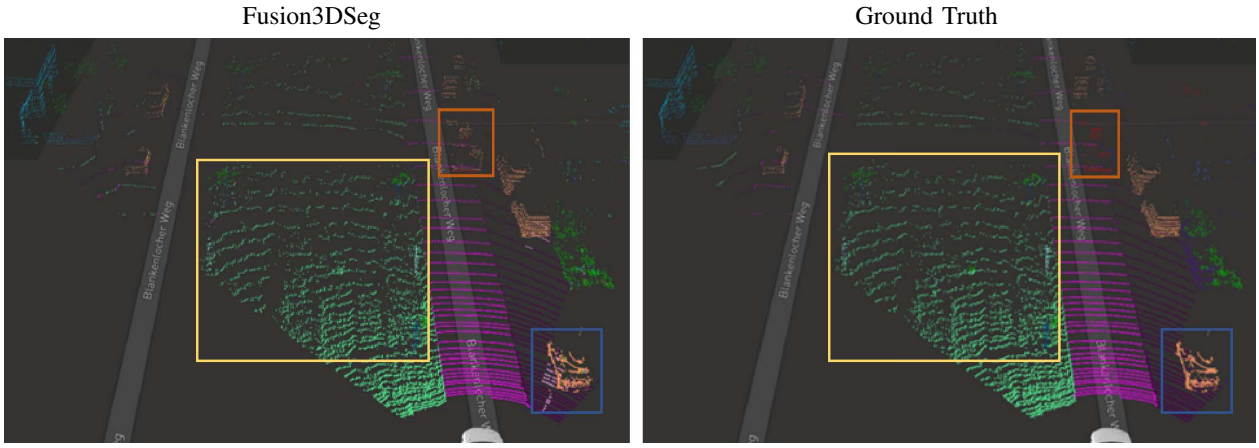
TABLE II
RESULTS WITH DIFFERENT CAMERA BACKBONES. THE RESNET50 IS PRETRAINED ON IMAGENET, WHILE THE DLA WASN'T PRETRAINED.

To ensure that the presented improvements are significant, a statistical test is conducted. Following [30], a randomization test is performed on the compared approaches. With $\alpha = 0.05$, all of the subsequently presented improvements are significant.

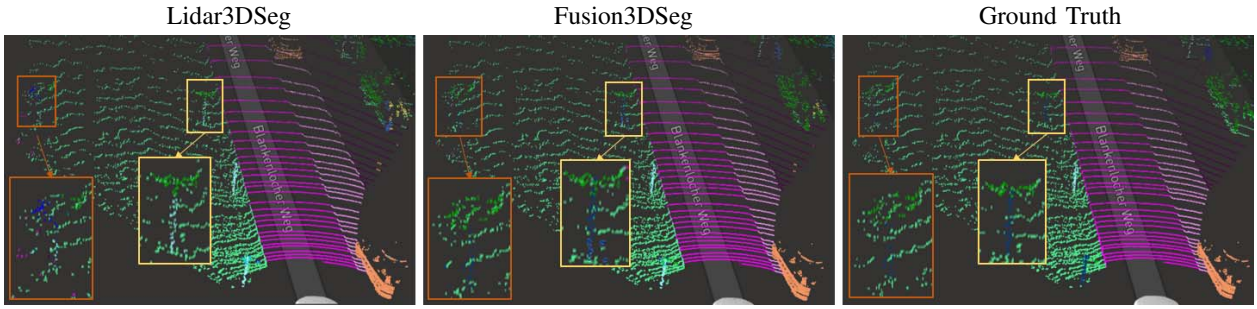
1) Lidar and Camera Fusion: We evaluate the improvements gained by the fusion of camera and lidar data for lidar semantic segmentation. Therefore, we use only the Lidar3DSeg part of our full network as baseline, which uses only lidar data, and compare it to our fusion approach. As stated in Sec. III-C, Lidar3DSeg is closely related to LaserNet [23] but with some modifications and targeted for semantic segmentation only. Table I shows, that Fusion3DSeg considerably outperforms Lidar3DSeg. Looking at the individual classes, most of them benefit from the fusion. Especially, the results of the classes person, traffic sign, pole and trunk greatly improved due to their small size, which makes it difficult to get them handled by lidar only. The additional higher resolution camera information clearly helps to identify these classes.

2) Iterativ Deep Fusion: The next evaluation step is the investigation of the presented iterative deep fusion strategy compared to the state-of-the-art fusion strategy of [2]. The results are shown in Tab. I. Our approach also outperforms [2] by a noticeable margin. The difference for the classes person, pole and trunk are smaller compared to Lidar3DSeg, which underlines the value of the camera features for these classes. However, our approach achieves better results for all of them, demonstrating that it is capable of exploiting more camera features. For the traffic sign class only our approach is able to exploit the camera features at all and improve the results. Generally, our iterative deep fusion strategy outperforms [2] for mostly all classes on the SemanticKitti dataset.

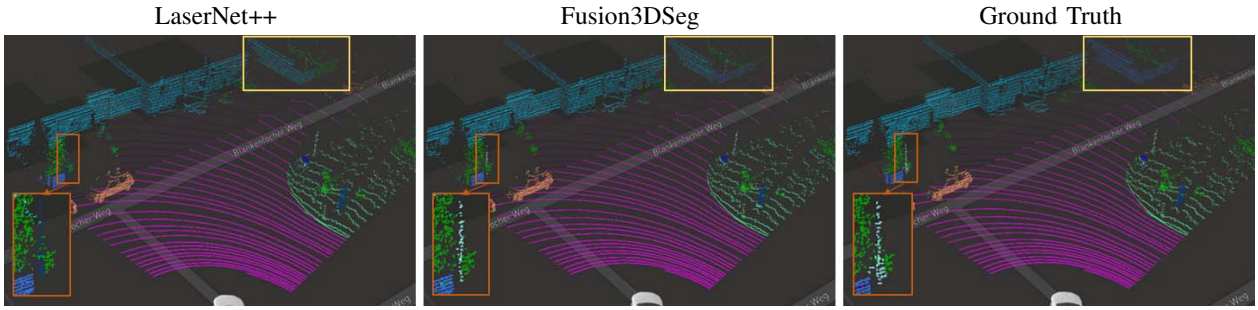
3) Ablation Study: Camera Backbones: While our previous evaluations are based on a pretrained ResNet50 as camera backbone, we also evaluate an untrained DLA network as



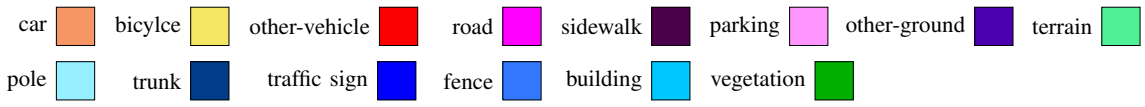
(a) Qualitative results of Fusion3DSeg.



(b) Qualitative results of Lider3DSeg and Fusion3DSeg.



(c) Qualitative results of LaserNet++ and Fusion3DSeg.



(d) Colors of the occurring classes.

Fig. 6. Qualitative result of the approaches Lidar3DSeg, Fusion3DSeg and LaserNet++ [2] on the SematicKitti dataset [27].

another backbone. Thereby, we demonstrate that our approach works with different backbones and can also be trained end-to-end with lidar labels only. Independently of the used backbone, our approach outperforms the Lidar3DSeg and LaserNet++, demonstrating that our approach also works without any labeled images. The results for both backbones and training strategies are shown in Tab. II.

D. Qualitative Results

In addition to the presented quantitative results, we also discuss qualitative results of our approach. In Fig. 6a, a segmentation result of Fusion3DSeg is shown. A first observation, highlighted in yellow, is the correct segmentation of the trees as trunk and vegetation, and the poles, with no confusion between the trunk and pole class. Also road and sidewalk are well segmented. Two challenges for our approach, as well as the other two, are highlighted in orange and blue. The other-vehicle class is sometimes confused with the car class, and the ground below cars is sometimes classified as parking area, even if there isn't one.

The next figure, Fig. 6b, underlines some advantages of the deep fusion approach Fusion3DSeg compared to Lidar3DSeg.

The highlighted tree (orange) is correctly classified by Fusion3DSeg as trunk and vegetation but not by Lidar3DSeg, which classifies the tree trunk mostly as pole and the crown as mixture of traffic sign and vegetation. Similarly, for the other highlighted tree (yellow), Lidar3DSeg confuses the trunk with a pole, while Fusion3DSeg correctly classifies the points. This underlines the presented quantitative improvements for these classes and the benefits of the higher resolution camera information.

The last qualitative comparison is shown in Fig. 6c for LaserNet++ and Fusion3DSeg. The latter correctly segments the highlighted pole, in contrast to LaserNet++, which confuses it with the trunk of a tree. In the highlighted yellow area, Fusion3DSeg doesn't segment the fence completely right, but parts of it, while LaserNet++ classifies all the points as building. This also corresponds to the quantitative results, demonstrating that our approach is better capable of exploiting additional information from the camera image.

V. Conclusion

In this work, we have presented a deep learning approach for semantic segmentation of 3D lidar point clouds. Our approach uses a range view representation of the lidar scans, enabling the application of established image segmentation approaches. Furthermore, we use camera image features of different scales, extracted by a camera backbone, and iteratively fuse them inside our network with the lidar feature maps. Our experiments underline the advantages of our deep fusion approach, which outperforms a lidar-only approach by a considerable margin. The presented fusion strategy also outperforms the current state-of-the-art fusion strategy by a noticeable margin.

References

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, 2016.

[2] G. P. Meyer, J. Charland, D. Hegde, A. Laddha, and C. Vallespi-Gonzalez, "Sensor fusion for joint 3d object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[3] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6526–6534, 2016.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. Vol.9351, pp. 234–241, 2015.

[7] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *International Conference on Learning Representations (ICLR)*, 2015.

[12] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 2016.

[13] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2403–2412, 2017.

[14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, 2016.

[15] Y. Yuan and J. Wang, "Ocnet: Object context network for scene parsing," *ArXiv*, vol. abs/1809.00916, 2018.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NIPS*, 2017.

[18] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," *International Conference on 3D Vision (3DV)*, pp. 537–547, 2017.

[19] C. Zhang, W. Luo, and R. Urtasun, "Efficient convolutions for real-time semantic segmentation of 3d point clouds," *International Conference on 3D Vision (3DV)*, pp. 399–408, 2018.

[20] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1887–1893, 2017.

[21] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4376–4382, 2018.

[22] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.

[23] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "Lasernet: An efficient probabilistic 3d object detector for autonomous driving," *ArXiv*, vol. abs/1903.08701, 2019.

[24] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 918–927, 2018.

[25] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, 2017.

[26] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *ArXiv*, vol. abs/1608.07916, 2016.

[27] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences." in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ArXiv*, vol. abs/1502.03167, 2015.

[30] M. D. Smucker, J. Allan, and B. Carterette, "A comparison of statistical significance tests for information retrieval evaluation," in *CIKM*, 2007.