Deep Projective 3D Semantic Segmentation

Felix Järemo-Lawin, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, Fahad Shahbaz Khan and Michael Felsberg

Book Chapter

Part of: Computer Analysis of Images and Patterns: 17th International Conference, CAIP 2017, Ystad, Sweden, August 22-24, 2017, Proceedings, Part I. Eds Michael Felsberg, Anders Heyden and Norbert Krüger (eds), 2017, pp. 95-107.
ISBN: 9783319646886 (print), 9783319646893 (online)

Lecture Notes in Computer Science, 0302-9743, No. 10424

DOI: https://doi.org/10.1007/978-3-319-64689-3_8

Deep Projective 3D Semantic Segmentation

Felix Järemo Lawin, Martin Danelljan, Patrik Tosteberg,
Goutam Bhat, Fahad Shahbaz Khan, Michael Felsberg

Computer Vision Lab, Dept. of Electrical Engineering, Linköping University

Abstract. Semantic segmentation of 3D point clouds is a challenging problem with numerous real-world applications. While deep learning has revolutionized the field of image semantic segmentation, its impact on point cloud data has been limited so far. Recent attempts, based on 3D deep learning approaches (3D-CNNs), have achieved below-expected results. Such methods require voxelizations of the underlying point cloud data, leading to decreased spatial resolution and increased memory consumption. Additionally, 3D-CNNs greatly suffer from the limited availability of annotated datasets.
In this paper, we propose an alternative framework that avoids the limitations of 3D-CNNs. Instead of directly solving the problem in 3D, we first project the point cloud onto a set of synthetic 2D-images. These images are then used as input to a 2D-CNN, designed for semantic segmentation. Finally, the obtained prediction scores are re-projected to the point cloud to obtain the segmentation results. We further investigate the impact of multiple modalities, such as color, depth and surface normals, in a multi-stream network architecture. Experiments are performed on the recent Semantic3D dataset. Our approach sets a new state-of-the-art by achieving a relative gain of 7.9%, compared to the previous best approach.

Keywords: Point clouds, semantic segmentation, deep learning, scanning artifacts, hard scape

1 Introduction

The rapid development of 3D acquisition sensors, such as LIDARs and RGB-D cameras, has lead to an increased demand for automatic analysis of 3D point clouds. In particular, the ability to automatically categorize each point into a set of semantic labels, known as semantic point cloud segmentation, has numerous applications such as scene understanding and robotics. While the problem of semantic segmentation of 2D-images has gained a considerable amount of attention in recent years, semantic segmentation of point clouds has received little interest despite its significance. In this paper, we propose a framework for semantic segmentation of point clouds that greatly benefits from the recent developments in semantic image segmentation.

With the advent of deep learning, many tasks within computer vision have seen a rapid progress, including semantic segmentation of images. The key factors for this development are the introductions of large labeled datasets [2] and GPU implementations of Convolutional Neural Networks (CNNs). However, CNNs have not yet been successfully applied for semantic segmentation of 3D point clouds due to several challenges.

In contrast to the regular grid-structure of image data, point clouds are in general sparse and unstructured. A common strategy is to resort to voxelization in order to directly apply CNNs in 3D. This introduces a radical increase in memory consumption and leads to a decrease in resolution. Additionally, labeled 3D data, which is crucial for training CNNs, is scarce due to difficulties in data annotation.

In this work, we investigate an alternative approach that avoids the aforementioned difficulties induced by 3D CNNs. As our first contribution, we propose a framework for 3D semantic segmentation that exploits the advantages of deep image segmentation approaches. The point cloud is first projected onto a set of synthetic images, which are then used as input to the deep network. The resulting pixel-wise segmentation scores are re-projected into the point cloud. The semantic label for each point is then obtained by fusing scores over the different views. As our second contribution, we investigate the impact of different input modalities, such as color, depth and surface normals, extracted from the point cloud. These modalities are fused in a multi-stream network architecture to obtain the final prediction scores.

Compared to semantic segmentation methods based on 3D CNNs [17], our approach has two major advantages. Firstly, our method benefits from the abundance of the already existing data sets for image segmentation and classification, such as ImageNet [2] and ADE20K [28]. This significantly reduces, or even eliminates the need of 3D data for training purposes. Secondly, by avoiding the large memory complexity induced by voxelization, our method achieves a higher spatial resolution which enables better segmentation quality.

We perform qualitative and quantitative experiments on the recently introduced Semantic3D dataset [6]. We show that different modalities contain complementary information and their fusion significantly improves the final segmentation performance. Further, our approach sets a new state-of-the-art performance on the Semantic3D dataset, outperforming both classical machine learning methods and 3D-CNN based approaches. Figure 4 shows an example segmentation result using our method.

2 Related Work

The task of semantic point cloud segmentation has received an increasing amount of attention due to the rapid development of sensors capable of capturing high-quality 3D data. RGB-D cameras, such as the Microsoft Kinect, have become popular for robotics and computer vision tasks. While RGB-D cameras are more suitable for indoors environments, terrestrial laser scanners capture large-scale point clouds for both indoors and outdoors applications. Both RGB-D cameras and modern laser scanners are capable of capturing color in association with the 3D information using calibrated RGB cameras. Besides visualization, this additional information is highly useful for automated analysis and processing of point clouds. While color is not a necessity for our approach, it alleviates the task of semantic segmentation and enables the use of large-scale image datasets.

Most previous works [1,7,11,16,13] in 3D semantic segmentation apply a combination of (i) hand-crafted features, (ii) discriminative classifiers and (iii) spatial smoothness models. In this setting, the construction of discriminative 3D-features (i) is ar-

guably the most important task. Popular alternatives include features based on the 3D structure tensor [7,26,11,1], histogram-based descriptors [7,16,11] such as Spin Images [10] and SHOT [21], and simple color features [26,16,11]. The classifiers (ii) are often based on maximum margin methods [13,1] or employ random forests [7,11,16]. To utilize spatial correlation between semantic labels (iii), many methods apply graphical models, such as the Conditional Random Field (CRF) [26,1,13].

Recently, deep convolutional neural networks (CNNs) have been successfully applied for semantic segmentation of 2D images [15]. Their main strength is the ability to learn high-level discriminative features, which eliminates the need of hand-designed representations. The rapid progress of deep CNNs for a variety of computer vision problems is generally attributed to the introduction of large-scale datasets, such as ImageNet [2], and improved performance for GPU computing.

Despite its success for image data, the application of CNNs to 3D point cloud data [9,20,27] have been severely hindered due to several important factors. Firstly, a point cloud does not have the neighborhood structure of an image. The data is instead sparse and scattered. As a consequence, CNN-based methods resort to voxelization strategies of the underlying point cloud data to enable 3D-convolutions to be performed (3D-CNNs). Secondly, voxelization have several disadvantages, including loss of spatial resolution and large memory requirements. 3D-CNNs are therefore restricted to small volumetric models or processing data in many smaller chunks, which limits the use of context. Thirdly, annotated 3D data is extremely limited, especially for the 3D semantic segmentation task. This greatly limits the power of CNNs for semantic segmentation of generic 3D point clouds.

In contrast, our approach avoids these short comings by projecting the point cloud into dense 2D image representations, thus removing the need for voxelizations. The 2D images can then be efficiently processed using 2D convolutions. Also, performing segmentation in image space allows us to leverage well developed 2D segmentation techniques as well as large amount of annotated data.

3 Method

In this section we present our method for point cloud segmentation. The input is an unstructured point cloud and the objective is to assign a semantic label to each point. In our method we render the point cloud from different views by projecting the points into synthetic images. We render color, depth and other attributes extracted from the point cloud. The images are then processed by a CNN for image-based semantic segmentation, providing a prediction scores for the predefined classes in every pixel. We make the final class selection from the aggregated prediction scores, using all images where the particular points are visible. An overview of the method is illustrated in Figure 1. A more detailed description is provided in the following sections.

3.1 Render views

The objective of the point cloud rendering is to produce structured 2D-images that are used as input to a CNN-based semantic segmentation algorithm. A variety of information stemming from the point cloud can be projected onto the synthetic images. In

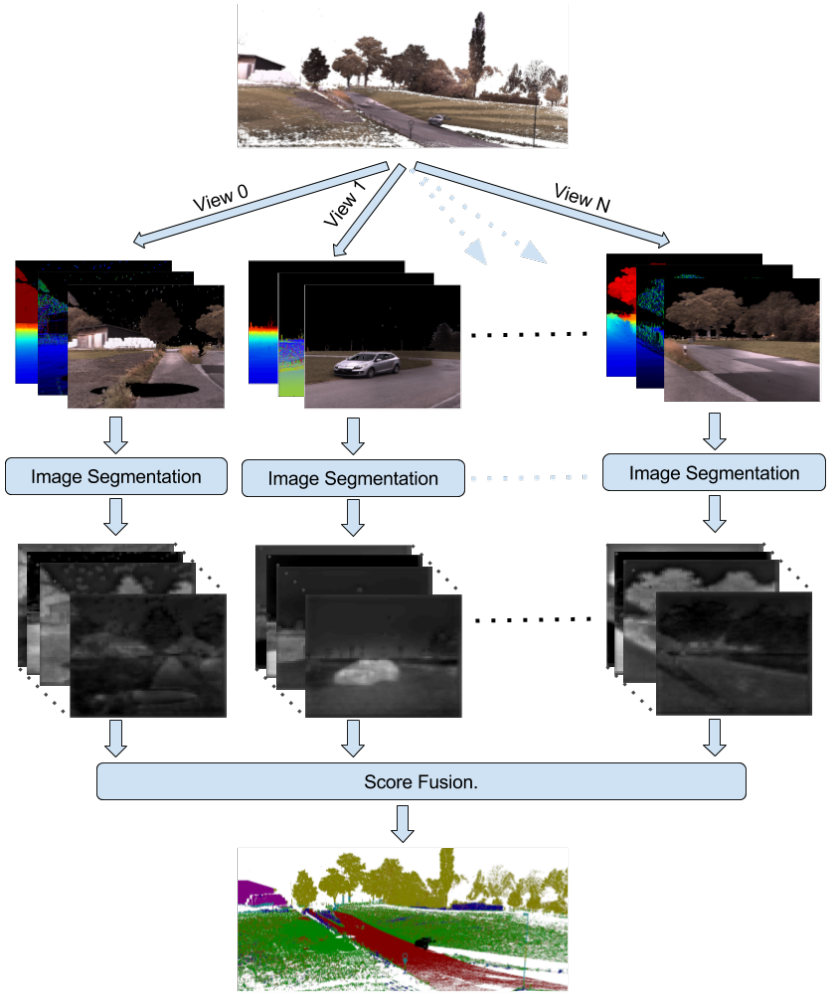

Fig. 1: An overview of the proposed method. The input point cloud is projected into multiple virtual camera views, generating 2D color, depth and surface normal images. The images for each view are processed by a multi-stream CNN for semantic segmentation. The output prediction scores from all views are fused into a single prediction for each point, resulting in a 3D semantic segmentation of the point cloud.

this work we particularly investigate the use of depth, color, and normals. However, the approach can be trivially extended to other features such as HHA [5] and other local information extracted from the point cloud. In order to map the semantic information back to the 3D points, we also need to keep track of the visibility of the projected points.

Our choice of rendering technique is a variant of point splatting [24,29], where the points are projected with a spread function into the image plane. While other rendering

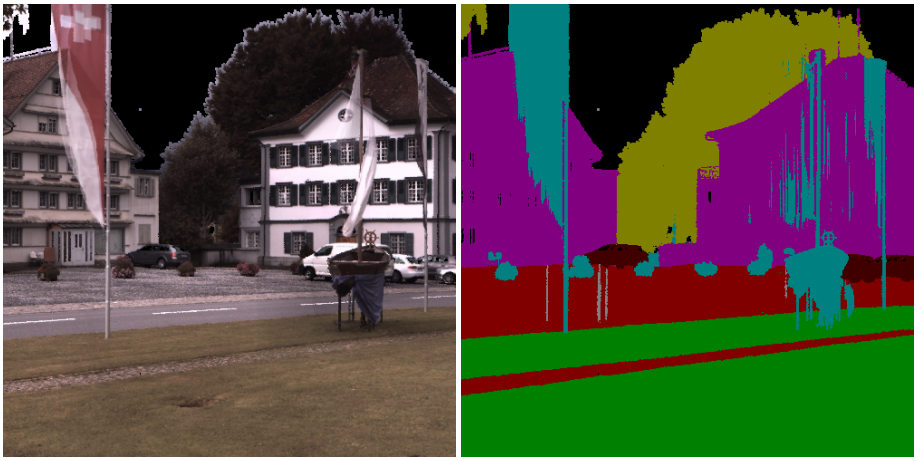

Fig. 2: Example of rendering output. Left: color image. Right: label image.

techniques, such as surface reconstruction as in [12], require demanding preprocessing steps of the point cloud in 3D space, splatting could be completely processed in image space. This further enables efficient and easily parallelizable implementations, which is essential for large-scale or dense point clouds.

Splatting-based rendering is performed by first projecting each 3D-point $\mathbf{x}_i$ of the point cloud into the image coordinates $\mathbf{y}_i$ of a virtual camera. The projected points are stored along with their corresponding depth values z_i and feature vectors $\mathbf{c}_i$. The latter can include, e.g., the RGB-color and normal vector of the point $\mathbf{x}_i$. The projection of a 3D-point is distributed by a Gaussian point spread function in the image plane,

$$w_{i,j} = G(y_i - p_j, \sigma^2)\,. \tag{1}$$

Here, $w_{i,j}$ is the contributed weight of point x_i to pixel j in the projected image. It is obtained by evaluating an isotropic Gaussian kernel G with scale σ^2 at the pixel location p_j. In order to reduce computational complexity, the kernel is truncated at a distance r. However, point spread functions, which originate from different surfaces, may still intersect in the image plane. Thus, the visibility of the projected points needs to be determined to avoid contributions of occluded surfaces. Moreover, the sensor data may contain significant foreground noise, such as scanning artifacts, which complicates this task. The challenge is to exclude the contribution from the noise and the occluded surfaces in the rendering process.

In traditional splatting [29], the resulting pixel value is obtained from the weighted average of the point spread functions in an accumulated fashion, using the weights $w_{i,j}$. If the depth of a new point significantly differs from the current weighted average, the pixel depth is either re-initialized with the new value if the point is closer than a specific threshold, or discarded if it is further away [29]. However, this implies that the resulting pixel value depends on both the threshold value and the order in which the

points are projected. Furthermore, noise in the foreground will have significant impact on the resulting images, as it is always rendered.

Similar to the method proposed in [19], we perform mean-shift clustering [24] of the projected points in each pixel with respect to the depth z_i weighted with $w_{i,j}$ using a Gaussian kernel density estimator $G(d, s^2)$, where s^2 denotes the kernel width. Starting from the depth value $d_i^0 = z_i$ for each point $i \in I_j$ that contributes to the current pixel j, $I_j = \{i : \|p_j - y_i\| < r\}$, the following expression is iterated until convergence

$$d_i^{n+1} = \frac{\sum_{i \in I_j} w_{i,j} G(d_i^n - z_i, s^2) z_i}{\sum_{i \in I_j} w_{i,j} G(d^n - z_i, s^2)} \,. \tag{2}$$

The iterative process determines a set of unique cluster centers $\{d_k\}_1^K$ from the converged iterates $\{d_i^N\}_{i \in I_j}$. The kernel density of cluster center d_k is given by,

$$v_k = \frac{\sum_{i \in I_j} w_{i,j} G(d_k - z_i, s^2)}{\sum_{i \in I_j} w_{i,j}} \,. \tag{3}$$

We rank the clusters with respect to the kernel density estimates and the cluster centers,

$$s_k = v_k + \frac{D}{d_k} \,. \tag{4}$$

Here, the weight D rewards clusters that are near the camera. It is set such that foreground noise and occluded points are not rendered. We chose the optimal cluster as $\tilde{k} = \arg\max_k s_k$ and set the depth value of pixel j to the corresponding cluster center $d_{\tilde{k}}$. The feature value is calculated as the weighted average, where the weight is determined by the proximity to the chosen cluster,

$$\mathbf{c}_{\tilde{k}} = \frac{\sum_{i \in I_j} w_{i,j} G(d_{\tilde{k}} - z_i, s^2) \mathbf{c}_i}{\sum_{i \in I_j} w_{i,j} G(d_{\tilde{k}} - z_i, s^2)} \,. \tag{5}$$

Since the indices $i \in I_j$ of the contributing points i are stored, it is trivial to map the semantic segmentation scores produced by the CNN back to the point cloud itself.

An example of the rendering output is shown in Figure 2.

3.2 Deep Multi Stream Image Segmentation

Following the current success of deep learning algorithms we deploy a CNN-based algorithm for performing semantic segmentation on the rendered images. We consider using multiple input modalities, which are combined using a multi-stream architecture [23]. The predictions from the streams are fused in a sum layer, as proposed in [4]. The full multi stream network can thus be trained end-to-end. However, note that our pipeline is agnostic to the applied image semantic segmentation approach.

In our method, each stream is processed using a Fully Convolutional Network (FCN) [15]. However, as previously mentioned, any CNN architecture can be employed. The FCN is based on the popular VGG16 network [22]. The weights in each stream are initialized by pre-training on the ImageNet dataset [2]. In this work, we investigate different combinations of input streams, namely color, depth, and surface normals. While

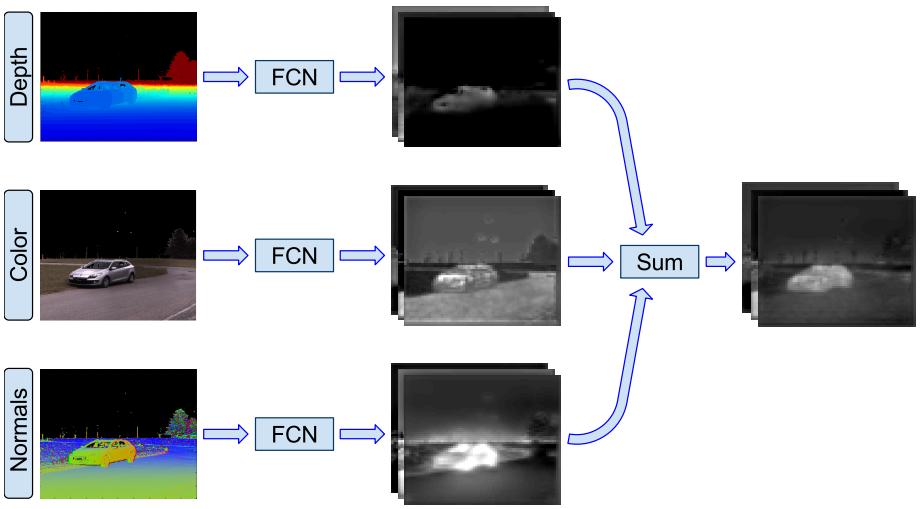

Fig. 3: Illustration of the proposed multi-stream architecture for 2D semantic segmentation. Each input stream is processed by a Fully Convolutional Network[15]. The prediction scores from each stream are summed to get the final prediction.

the RGB-stream naturally benefits from pre-training on ImageNet, this is also the case for the depth stream. Previous work [3] has shown that a 3-channel jet colormap representation of the depth image better benefits from pre-training on RGB datasets, such as ImageNet. Finally, we also consider surface normals as input to a separate network stream. For this purpose, we deploy an efficient algorithm for approximate normals computation, which is based on direct differentiation of the depth map.

3.3 Score fusion

The deep network outputs a prediction score for each class for every pixel in the image. The scores from each rendered view are mapped to the corresponding 3D points using the indices $i \in I_j$ as described in section 3.1. We fuse the scores by computing the sum over all projections. Finally, the points are assigned the labels corresponding to the largest sum of scores.

4 Experiments

4.1 Dataset

We conduct our experiments on the dataset Semantic3D [6], which provides a set of large scale 3D point clouds of outdoor environments. The point clouds were acquired by a laser scanner and include both urban and rural scenes. Colorization was performed using a cube map generated from a set of high-resolution camera images. In total, the

dataset contains 30 separate scans and over 4 billion 3D-points. The points are labeled with 8 different semantic classes: man-made terrain, natural terrain, high vegetation, low vegetation, buildings, hard scape, scanning artifacts, and cars.

4.2 Experimental setup

View selection In order to fully cover the point clouds in the rendered views, we collect images by rotating the camera 360° around a fix vertical axes. For each 360° rotation, we use 30 camera views at equally spaced angles. For each point cloud, we generate four such scans with different pitch angles and translations of the camera, resulting in a total of 120 camera views. To maintain a certain amount of contextual information, we remove images where where more than 10% of the pixels have a depth less than five meters. Furthermore, images with less than 5% coverage were discarded.

Network setup and training For the training we generated ground truth label images by selecting the most commonly occurring label in the optimal cluster from section 3.1. An example is shown in Figure 2. In addition to the 8 provided classes, we also included a 9th background class to label empty pixels, i.e pixels without any intersecting point spread functions. We generated training data from the training set provided by Semantic3D [6], consisting of 15 point clouds from different scenes. Our training data set consists of 3132 labeled images including color, jet visualization of the depth, and surface normals.

We investigate the proposed multi stream approach using color, depth and surface normals streams as input. In order to determine the contribution of each input stream we also evaluate network configurations with a single stream. Since some point clouds may not have color information we also investigate a multi stream approach without the color stream. All network configurations are listed in table 1.

Table 1: Network configurations with input streams in the left column

	RGB	**D**	**N**	**RGB+D+N**	**D+N**
Color	X			X	
Depth jet		X		X	X
Surface normals			X	X	X

All network configurations were trained using the same training parameters. We trained for 45 epochs with a batch size of 16. The initial learning rate was set to 0.0001 and divided by two every tenth epoch. Following the recommendations from [14], we used a momentum of 0.99. The networks were trained using MatConvNet [25].

4.3 Results and Discussions

We evaluated our method for the different network configurations on the reduced test set provided by Semantic3D. The test set consists of four point clouds, containing 80

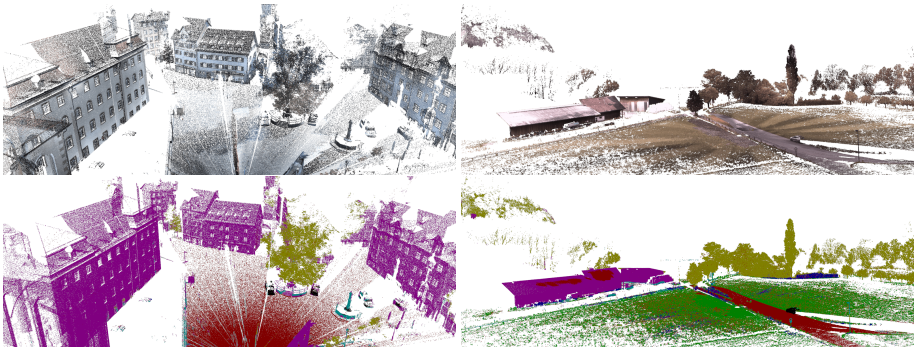

Fig. 4: Qualitative results. Top: input point clouds. Bottom: Segmentation output using our proposed **RGB+D+N** network.

million points in total. All points are assigned a class label j, which is compared to the ground truth label i. A confusion matrix C is constructed, were each entry c_{ij} denotes the number of points with the ground truth label i that are assigned the label j. The quantitative measure provided by the benchmark [6] is the intersection over union for each class i, given by

$$\text{IoU}_i = \frac{c_{ii}}{c_{ii} + \sum_{j \neq i} c_{ij} + \sum_{k \neq i} c_{kj}} \,. \tag{6}$$

The over all accuracy is also provided and is given by

$$\text{IoU} = \frac{\sum_i c_{ii}}{\sum_j \sum_{jk} c_{jk}} \,. \tag{7}$$

The evaluation results are shown in table 2. The single-stream network with RGB and surface normals as input performs significantly better than the single-stream depth network. However, the three streams seem to provide complementary information, and give a significant gain in performance when used together. Our best multi-stream approach significantly improves over the previous state-of-the art method [8]. Also our multi-stream approach without the color stream obtains results comparable to the previous state-of-the, showing that our method is applicable even if color information is absent. Interestingly, even our single-stream approaches with only RGB or surface normals as input achieves a remarkable gain compared to the 3D-CNN based VoxNet [6]. Figure 4 shows some qualitative results on the test set using our multi-stream **RBG+D+N** network.

Note that we are using a simple heuristic for generating camera views, and a basic segmentation network trained on limited data. Yet, we obtain very promising results. Replacing these blocks with better alternatives should improve the results even further. However, this is outside the scope of this paper.

Table 2: Benchmark results on the reduced test set in Semantic3D [6]. IoU for categories (1) man-made terrain, (2) natural terrain, (3) high vegetation, (4) low vegetation, (5) buildings, (6) hard scape, (7) scanning artefacts, (8) cars.

	Avg IoU	**OA**	**IoU1**	**IoU2**	**IoU3**	**IoU4**	**IoU5**	**IoU6**	**IoU7**	**IoU8**
TML-PCR[18]	0.384	0.740	0.726	0.730	0.485	0.224	0.707	0.050	0.000	0.150
DeepNet[6]	0.437	0.772	0.838	0.385	0.548	0.085	0.841	0.151	0.223	0.423
TLMC-MSR[8]	0.542	0.862	**0.898**	0.745	0.537	0.268	0.888	0.189	**0.364**	0.447
Ours RGB	0.515	0.854	0.759	0.791	0.720	**0.335**	0.857	0.209	0.123	0.326
Ours D	0.262	0.662	0.281	0.468	0.395	0.179	0.763	0.006	0.001	0.000
Ours N	0.511	0.846	0.815	0.622	0.679	0.164	0.903	**0.251**	0.186	0.470
Ours RGB+D+N	**0.585**	**0.889**	0.856	**0.832**	**0.742**	0.324	0.897	0.185	0.251	**0.592**
Ours D+N	0.543	0.872	0.839	0.736	0.717	0.210	**0.909**	0.153	0.204	0.574

5 Conclusion

We propose an approach for semantic segmentation of 3D point clouds that avoids the limitations of 3D-CNNs. Our approach first projects the point cloud onto a set of synthetic 2D-images. The corresponding images are then used as input to a 2D-CNN for semantic segmentation. Consequently, the segmentation results are obtained by re-projecting the prediction scores to the point cloud. We further investigate the impact of multiple modalities in a multi-stream deep network architecture. Experiments are performed on the Semantic3D dataset. Our approach outperforms existing methods and sets a new state-of-the-art on this dataset.

References

1. Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A.Y.: Discriminative learning of markov random fields for segmentation of 3d scan data. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA. pp. 169–176 (2005)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
3. Eitel, A., Springenberg, J.T., Spinello, L., Riedmiller, M., Burgard, W.: Multimodal deep learning for robust rgb-d object recognition. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 681–687. IEEE (2015)
4. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 1933–1941 (2016), http://dx.doi.org/10.1109/CVPR.2016.213
5. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: European Conference on Computer Vision. pp. 345–360. Springer (2014)
6. Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M.: Semantic3d. net: A new large-scale point cloud classification benchmark. arXiv preprint arXiv:1704.03847 (2017)

7. Hackel, T., Wegner, J.D., Schindle, K.: Fast semantic segmentation of 3d point clouds with strongly varying density. In: ISPRS Annals - ISPRS Congress, Prague (2016)
8. Hackel, T., Wegner, J.D., Schindler, K.: Fast semantic segmentation of 3d point clouds with strongly varying density. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, Czech Republic 3, 177–184 (2016)
9. Huang, J., You, S.: Point cloud labeling using 3d convolutional neural network. In: International Conference on Pattern Recognition (ICPR) (2016)
10. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans. Pattern Anal. Mach. Intell. 21(5), 433–449 (1999)
11. Kähler, O., Reid, I.D.: Efficient 3d scene labeling using fields of trees. In: IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013. pp. 3064–3071 (2013)
12. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Transactions on Graphics (TOG) 32(3), 29 (2013)
13. Kim, B., Kohli, P., Savarese, S.: 3d scene understanding by voxel-crf. In: IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013. pp. 1425–1432 (2013)
14. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. arXiv preprint arXiv:1506.04579 (2015)
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440 (2015)
16. Martinovic, A., Knopp, J., Riemenschneider, H., Gool, L.J.V.: 3d all the way: Semantic segmentation of urban scenes from start to end in 3d. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. pp. 4456–4465 (2015)
17. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 922–928. IEEE (2015)
18. Montoya-Zegarra, J.A., Wegner, J.D., Ladickỳ, L., Schindler, K.: Mind the gap: modeling local and global context in (road) networks. In: German Conference on Pattern Recognition. pp. 212–223. Springer (2014)
19. Ogniewski, J., Forssén, P.E.: Pushing the limits for view prediction in video coding. In: 12th International Conference on Computer Vision Theory and Applications (VISAPP'17). Scitepress Digital Library, Porto, Portugal (February-March 2017)
20. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 5648–5656 (2016)
21. Salti, S., Tombari, F., di Stefano, L.: SHOT: unique signatures of histograms for surface and texture description. Computer Vision and Image Understanding 125, 251–264 (2014)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
23. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 568–576 (2014), http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos
24. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer-Verlag New York, Inc. (2010)

25. Vedaldi, A., Lenc, K.: Matconvnet – convolutional neural networks for matlab. In: Proceeding of the ACM Int. Conf. on Multimedia (2015)
26. Wolf, D., Prankl, J., Vincze, M.: Fast semantic segmentation of 3d point clouds using a dense CRF with learned parameters. In: IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015. pp. 4867–4873 (2015)
27. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. pp. 1912–1920 (2015), http://dx.doi.org/10.1109/CVPR.2015.7298801
28. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
29. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Surface splatting. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 371–378. ACM (2001)