A Panoramic Segmentation Network for Point Cloud

Yuanyun Jin¹, Xiangfeng Luo¹,², *, Wenbin Yang², Shaorong Xie², Tong Liu²
¹Shanghai Institute for Advanced Communication and Data Science, China
²School of Computer Engineering and Science, Shanghai University, Shanghai, China

*Corresponding author e-mail: luoxf@shu.edu.cn

Abstract. Scene segmentation mainly consists of semantic segmentation and instance segmentation. The latest research points out that combining the two segmentation methods to achieve panoramic segmentation can understand the current scene better. The point cloud contains rich spatial information, but panoramic segmentation research in this field is rarely discussed. How to use the unified model framework to obtain the results of instance segmentation and semantic segmentation is the key to realize the task of point cloud panoramic segmentation. In this paper, we propose a panoramic segmentation network for point cloud. In feature encoding stage, we introduce the potential correlation information among points to improve the performance of feature extraction. Then, an output module is presented to combine the results of the two decoders which uses objective distance to enhance the semantic and instance segmentation. Experiments show that our model has good performance on the panoramic segmentation task of point cloud.

1. Introduction
Point cloud is a set of points that express the spatial distribution and surface characteristics of a target in the same spatial coordinate system. It has rich spatial information, with which we can accurately locate objects and describe their contours. Point cloud data are widely used in 3D scene segmentation. 3D scene segmentation refers to dividing 3D data into groups of regions with certain specific meanings, marking the regions, and finally obtaining 3d views with information annotation. Nowadays, accurate and efficient scene segmentation is useful in many applications, such as automatic driving, indoor navigation, and virtual reality and reality enhancement. Actually, there are two main segmentation tasks in 3D scene segmentation, semantic segmentation, and instance segmentation. Semantic segmentation mainly studies how to segment diverse objects in the scene. The method of scene segmentation is to assign a class label to each pixel or point. The other is the instance segmentation based on the research of things. The boundary lines are drawn for all objects in the scene, and the mask is used to divide different instances. The method of scene segmentation is usually to generate the bounding box by the object detection and then to segment the points in the bounding box.

But in fact, there is a close relationship between semantic segmentation and instance segmentation, that an object must belong to one class, and the points in the same object belong to the same class. Instance segmentation largely depends on the performance of semantic segmentation. Similarly, using effective instance segmentation to delineate object scope can also react to semantic segmentation and improve its accuracy. Thus, panoramic segmentation is proposed to combine the two segmentation methods. Panoramic segmentation uses a unified network model to predict the category label and instance ID of
each point to achieve a unified and global segmentation. Approaches based on graphical models \[1\], \[2\], \[26\], \[27\] have been proposed to combine the two segmentation methods, but there are rare researches on the panoramic segmentation models proposed based on point cloud.

In recent years, research on point cloud mainly focuses on feature extraction of points and semantic segmentation. Point Net \[7\] proposed a network structure that utilizes MLP to extract features directly from points and achieved good results in 3D object recognition, 3D object parts segmentation and semantic segmentation. Later, Point Net++ \[8\] utilizes metric space distance to learn local features by increasing context scale. Then, algorithms \[9\]–\[12\] were proposed successively and achieved good results. However, the above methods ignore the potential connection between points.

To achieve efficient panoramic segmentation on point cloud data, the following two problems need to be solved: (1) how to efficiently extract the features of point cloud. In fact, both semantic segmentation and instance segmentation are based on point feature extraction. Only effective feature extraction can improve the accuracy. (2) How to combine semantic and instance result output to improve panoramic segmentation accuracy. Experiments on images \[19, 20\] have proved that the combination of semantic and instance segmentation can improve their respective performance and obtain better panoramic segmentation results.

Based on the above analysis, we design the feature extraction network of points. In the encoding layer, we use vector information formed by sampling points and surrounding points as another set of important parameters besides local point feature information to integrate into the single point feature, so as to improve the feature extraction effect of the original point cloud. Aiming at obtaining both semantic and instance results at the same time, ASIS\[3\] inspired us to use two parallel branches for instance segmentation and semantic segmentation respectively, and then fuse the two results as output. However, \[3\] only takes the similar points in feature space, but not Euclidean space. Based on this problem, ASIS is modified in the panorama segmentation output module of our model. Finally, experiments with different data sets demonstrate the superiority of our backbone network in feature extraction. Compared with the original ASIS model, our panoramic segmentation output module can better realize the win-win results of semantic segmentation and instance segmentation. Our model shows certain advantages in panoramic segmentation compared with PointNet++ \[8\] and ASIS \[3\].

2. Related Work

With the application of deep learning in computer vision, semantic segmentation has developed rapidly. In terms of 2D images, great breakthroughs have been made in semantic segmentation \[4\]–\[6\] based on the fully convolutional neural network. In recent years, research on deep learning for point clouds has been increasing. PointNet \[7\] is a pioneering work for the successful application of deep learning in 3D point cloud. Taking original point cloud as the input of deep neural network, PointNet provides a unified architecture for classification, part segmentation, and scene semantic segmentation, but it does not consider the problem of local feature extraction. PointNet++ \[8\] carries out farthest point sampling and region division on point cloud, and extracts local features by using PointNet within the region to improve the feature extraction effect. And then a series of algorithms based on PointNet \[7\] and PointNet++ \[8\] have been proposed. SpiderCNN \[11\] introduced Taylor’s formula in the process of convolution, which further improved the effect of feature extraction. PointConv \[12\] learns the convolution mode of images and constructs a multi-layer deep convolutional network, which can achieve transformation invariance as 2D convolutional network. However, the above method only considers the information about local points but ignores the potential relevance between points.

The deep neural network has also made good progress in image instance segmentation \[13\]–\[15\]. The basic idea is to frame different instances with object detection, and then conduct pixel by pixel marking in different instance regions with method of semantic segmentation. Currently, many deep learning models have been proposed for point cloud object detection \[16\]–\[19\], but there are few researches on point cloud instance segmentation. 3D-SIS \[20\] uses 2D convolution to generate features of RGB-D images and projects the features into 3D grids. 3D RPN (Region Proposed Network) and RoI (Region of Interest) are used to infer target boundaries. SPGN \[21\] uses the similarity matrix to reflect the similarity
of different points in the feature space, so as to estimate the similar area of each point to achieve instance segmentation.

Panoramic segmentation [22] was first proposed jointly by FAIR and the University of Heidelberg in Germany. Panoramic segmentation can achieve both semantic segmentation and instance segmentation. However, it also faces two challenges: Firstly, the target must be non-overlapping. Secondly, not only the target, the pixel or point of background have to be labeled as an instance too. In the aspect of image, we have started the exploration of panoramic segmentation, hoping to realize the reconciliation of things and stuff. Panoptic Segmentation [22] proposes the PQ (Panoptic Quality) index for image segmentation. OANet [2] first proposed an end-to-end algorithm sensitive to occlusion for panoramic segmentation and introduced a novel spatial sorting module to solve the fuzziness of overlapping problem. Panoptic Feature Pyramid Networks [1] designed Panoptic FPN by combining FCN for semantic segmentation with MaskRCNN for instance segmentation, to propose a powerful baseline for panoramic segmentation of images. Through the above research and experiment, it is found that the combination of semantic segmentation and instance segmentation can effectively promote their accuracy. In three-dimensional space, the overlap of objects is smaller than that of images, but the close distance between objects will also make it impossible to accurately segment two different instances. In point cloud, ASIS [3] organically combines the output branches originally used for semantic segmentation and instance segmentation, realizes instance reference in semantic segmentation, and introduces semantic perception in instance segmentation, and provides an effective end-to-end training model. However, [3] only considers the similarity of embedded features of instances in semantic perception but ignores the fact that the points of instances tend to be relatively close in reality. Based on the above research, this work aims to establish a panoramic segmentation model of point cloud to effectively obtain semantic labels and instance ids of points in the scene, so as to better segment the scene.

3. Our Method

Our network architecture is shown in Fig. 1, consisting of an encoder, two decoders, and a panoramic segmentation output module. Through the combination of encoder and decoder, instance features and semantic features are obtained respectively. Panoramic segmentation output module is used to fuse the two features, to improve the efficiency of instance and semantic segmentation. Finally, the semantic label and instance ID of each point are predicted.

![Figure 1. Model Structure Diagram](image)

3.1. Encoder using potentially connection information

Given input points \( \{ p_i \mid i = 1, 2, ..., n \} \), each point contains coordinates of the point and features such as color, reflectivity, etc. Different from image, in the point cloud, the distance between points is not fixed, so the potential connection information is also an important feature of point cloud, which is ignored by lots of encoders at present. The encoder in this paper is modified on the basis of PointNet++ to effectively improve the feature extraction effect of points.

Firstly, we use farthest point sampling to choose a subset of points \( \{ p_j \mid j \in [1, n] \} \), \( p_j \) means a sampling point, suppose the number of sampling points is no more than \( M \). Then set the radius to \( r \), the ball query selects a fixed number of the neighboring points (including itself) within the radius of each
sampling point. It makes up a local point cloud \( \{ p_{jk} | k = 0, 1, 2, ..., K - 1 \} \). \( p_{jk} \) is the feature of the \( k_{th} \) point around the sampling point \( p_j \). \( K \) is the number of neighboring points. According to the coordinates of each point, we can generate a set of vectors \( \{ \tilde{x}_{jk} | k = 0, 1, 2, ..., K - 1 \} \). \( \tilde{x}_{jk} \) is the vector from \( p_j \) to \( p_{jk} \), used as a potential correlation between points. Similar to PointNet, our model also uses multilayer perceptron (MLP) and symmetric functions to fit edge feature extraction functions. Finally, the point features and edge features are fused to make the feature extraction more effective. It should be added that since the radius query contains the sampling point itself, an empty vector will be included in to represent the line connecting the sampling point to itself, but this has little impact on the model effect. The above process is as follows:

\[
fp = f_1 \left( \left\{ p_{j1}, ..., p_{jk} \right\} \right) \approx g_1 \left( h_1 \left( p_{j1} \right), ..., h_1 \left( p_{jk} \right) \right) \quad (1)
\]

\[
fl = f_2 \left( \left\{ \tilde{x}_{j1}, ..., \tilde{x}_{jk} \right\} \right) \approx g_2 \left( h_2 \left( \tilde{x}_{j1} \right), ..., h_2 \left( \tilde{x}_{jk} \right) \right) \quad (2)
\]

\[
f = h_3 \left( \left[ fp, fl \right] \right) \quad (3)
\]

\( fp \) is the feature formed by local points, \( fl \) is the feature formed by the local vector, \( f \) means the feature fused by \( fp \) and \( fl \). We approximate \( g_1, g_2 \) by a carpooling function and \( h_1, h_2, h_3 \) by a multi-layer perceptron network. The specific structure is shown in Fig. 2.

![Figure 2. Encoder Structure Diagram](image-url)

**Figure 2.** Encoder Structure Diagram

### 3.2 Panoramic segmentation output module

Panoramic segmentation requires that each point be assigned a category label and an instance ID. Two parallel decoders, one for point-level semantic prediction and the other for extracting point-to-point instance relationships, are applied to our network model. Through these two decoders, the semantic labels and instance IDs of the predicted points can be obtained by clustering and classifying the points. Refer to [3], the model can be trained by constructing the following loss function:

\[
L = L_{\text{sem}} + \alpha \cdot L_{\text{ins}} + \beta \cdot L_{\text{reg}} \quad (4)
\]

\[
L_{\text{sem}} = \frac{1}{N} \sum_{i=1}^{l_d} - \log[p_i \cdot p_i^* + (1 - p_i)(1 - p_i)] \quad (5)
\]

\[
L_{\text{ins}} = L_{\text{var}} + L_{\text{dist}} \quad (6)
\]
\[
L_{\text{var}} = \frac{1}{I} \sum_{i=1}^{I} \frac{1}{N_i} \sum_{j=1}^{N_i} ||u_i - e_j||_1 + \delta_v
\]
(7)

\[
L_{\text{dist}} = \frac{1}{I(I-1)} \sum_{i=1}^{I} \sum_{j=1}^{I} [2\delta_d - ||u_i - u_j||_1]^2
\]
(8)

\[
L_{\text{reg}} = \frac{1}{I} \left( \sum_{i=1}^{I} ||u_{si}||_1 \right)^2
\]
(9)

\(L_{\text{sem}}\) is the loss of semantic segmentation, with classic cross-entropy loss to oversee the semantic segmentation branch, \(p_i\) means ground truth, \(p_i^\ast\) means predict labels; \(L_{\text{ins}}\) is the loss of instance segmentation, we adopt the class-agnostic instance embedding learning strategy refer to ASIS [3], \(N_i\) represent the number of points contained in instance \(i\), \(\mu_i\) the mean embedding of instance \(i\), \(e_j\) is an embedding of a point; \(L_{\text{reg}}\) represent regular loss, \(u_{si}\) means the \(i\)th point feature of \(F_{\text{ins}}\), \(\alpha\) is the ratio of semantic segmentation with instance segmentation performance. During the test, the final instance tag can be obtained by using the average embedding in the instance, and the semantic tags of the points in the same instance can be identified as their categories. The model structure of panoramic segmentation has been realized. ASIS [3] proves that there is a potential relationship between instance segmentation and semantic segmentation. An instance object must have its category tag, and the category tag of the same instance interior point is consistent. Through the effective combination of instance features and semantic features, the accuracy of both results can be improved, so as to improve the efficiency of the final panoramic segmentation. However, when combining instances with semantics, ASIS [3] only considers the similarity of instance features to build semantic features of points. After feature learning, points with similar features are not necessarily close in spatial, which is inconsistent with the objective fact that points with the same instance are close in spatial. Therefore, this paper takes spatial location relationships as an important factor of feature combination to improve the final segmentation result.

KNN is used to search for a fixed number of neighboring points for each point in the instance feature space. According to the index matrix in the semantic feature matrix \(F_{\text{sem}}\) find their semantic characteristics and grouped respectively, then fuse semantic features of each point by a carpooling. The ball query is used to search the neighboring points in the spatial distance, and the carpooling is also used to fuse their semantic features. After combining the semantic features of two different search methods, it generates the final semantic features. This process can be expressed as:

\[
F_{\text{isem}1} = \text{Max} (\text{KNN}(F_{\text{ins}}))
\]
(10)

\[
F_{\text{isem}2} = \text{Max} (\text{queryball}(F_{\text{sem}}))
\]
(11)

\[
F_{\text{isem}} = [F_{\text{isem}1}, F_{\text{isem}2}]
\]
(12)

4. Experiment

The experiment is divided into the following parts. Firstly, the performance of encoder and decoder in our model is verified by different 3d recognition tasks. Secondly, the performance of our panoramic segmentation output module is demonstrated on the S3DIS data set. Finally, the structural characteristics of the model are analyzed through control experiments. All our experiments are implemented by Tensor flow on GTX 2080Ti GPU.

4.1. Backbone network

In order to verify the performance of our point cloud encoder network, we compared the point-based classification and segmentation methods. The data sets used mainly include ModelNet40 [24], ShapeNet [25].

Experiment on classification: ModelNet40 contains 12, 311 CAD models of 40 different types, which are divided into 2 parts, 9, 843 for training and 2,468 for testing. The model was uniformly sampled to
1024 points according to the configuration of PointNet [7]. We randomly rotate the point cloud along the $z$-axis in the training process and use the Gaussian noise with the mean value of 0 and standard deviation of 0.02 to flutter each point. In the training process, we used the same Adam optimizer for all models and trained for 250 epochs. As shown in TABLE I, our encoder performs better than the original PointNet++ [8]. In the comparison of this experiment, our model only adds vector information in the coding layer. Therefore, it can be seen that we introduce edge information into the features of the point is effective. Compared with some other algorithms, our encoder has some shortcomings in performance. Next, we will try to introduce edge features into these algorithms and experiment to see if we can improve their performance.

| Method            | Input                  | Accuracy (%) |
|-------------------|------------------------|--------------|
| PointNet[7]       | 1024 points            | 89.2         |
| PointNet++[8]     | 1024 points            | 90.2         |
| SpiderCNN[11]     | 1024 points+normal     | 92.4         |
| PointConv[12]     | 1024 points+normal     | 92.5         |
| Ours              | 1024 points            | 90.8         |

Experiment on part segmentation: we used an encoder and a decoder to form a point cloud segmentation model, and conducted experiments on ShapeNet [25] data sets. The ShapeNet [25] dataset contains 16,881 3D shapes of 16 different types and contains 50 different parts. We divided the dataset into two parts, 14,006 for training and 2,874 for testing. We compared with other network models, and the results were shown in TABLE II. Through analysis, it can be found that our model is improved based on PointNet++, but it also shows the performance not inferior to the current advanced algorithms. In this experiment, in addition to replacing the coding layer, we also used the iterative algorithm in the decoding layer. The result is a significant increase over PointNet++, which proves that our design ideas are useful.

### 4.2. Panoramic segmentation results

The experiment of panoramic segmentation is based on the S3DIS [23] dataset. During data preprocessing, the room was still divided into several overlapping blocks according to the ground 1m × 1m, and each overlapping block contained 4096 points. The ASIS source code is then used to attach semantic labels and instance ids to each point in the S3DIS dataset. In order to be fair to compare with ASIS, when training model will also be \( \delta_v \) set to 0.5, \( \delta_d \) set to 1.5, for instance, the fusion of KNN search, K set to 30, training 100 epochs, use Adam optimizer, basic vector is set to 0.001, every 300000 steps in half. For the test, the bandwidth was set to 0.6 for the average shifted cluster. The result of instance segmentation is shown in TABLE III. Based on the skeleton of PointNet++, comparing our output module with ASIS, mCov increased by 0.6 and mRec increased by 1.0, achieving better instance segmentation effect. In terms of semantic segmentation, as shown in TABLE IV, our output modules mAcc and mIoU have both been improved. Finally, referring to [1], we used the PQ as the result index of panoramic segmentation, as shown in TABLE V. Compared with the original model based on PointNet++ [7] and ASIS, our model improved by 0.3 on the PQ and achieved a good panoramic segmentation result. In order to better demonstrate the splitting Effect of our network, we present the final result in the form of semantic segmentation and instance segmentation, as shown in Fig. 3.
Table 2. Segmentation results on ShapeNet part dataset

| Method         | Class avg (%) | Instance avg (%) |
|----------------|---------------|------------------|
| PointNet[7]    | 80.4          | 83.7             |
| PointNet++[8]  | 81.9          | 85.1             |
| SpiderCNN[11]  | 82.4          | 85.3             |
| PointConv[12]  | 82.8          | 85.7             |
| Ours           | 82.0          | 84.9             |

Table 3. Instance segmentation results on S3DIS dataset.

| Backbone       | Method | mCov  | mWCov | mPrec | mRec |
|----------------|--------|-------|-------|-------|------|
| PN++[8]        | ASIS[3]| 45.2  | 48.1  | 53.3  | 42.4 |
|                | Ours   | 45.8  | 48.6  | 53.7  | 43.4 |
| Ours           | ASIS[3]| 46.3  | 49.2  | 55.3  | 45.3 |
|                | Ours   | 47.0  | 49.8  | 56.5  | 46.5 |
| Test on 6-fold CV |     |       |       |       |      |
| PN++[8]        | ASIS[3]| 51.2  | 55.1  | 62.9  | 47.5 |
|                | Ours   | 52.6  | 56.8  | 63.5  | 48.2 |
| Ours           | ASIS[3]| 55.3  | 57.4  | 64.3  | 48.9 |
|                | Ours   | 55.7  | 58.0  | 65.4  | 49.6 |

Table 4. Semantic segmentation results on S3DIS dataset

| Backbone       | Method | mAcc  | mIoU  | oAcc  |
|----------------|--------|-------|-------|-------|
| PN++[8]        | ASIS[3]| 60.9  | 52.6  | 86.8  |
|                | Ours   | 61.3  | 53.0  | 86.6  |
| Ours           | ASIS[3]| 61.2  | 52.8  | 86.9  |
|                | Ours   | 62.0  | 53.6  | 87.0  |
| Test on 6-fold CV |     |       |       |       |
| PN++[8]        | ASIS[3]| 69.8  | 59.3  | 85.8  |
|                | Ours   | 71.1  | 60.4  | 86.5  |
| Ours           | ASIS[3]| 72.2  | 61.5  | 86.8  |
|                | Ours   | 73.1  | 62.3  | 87.0  |

Table 5. Panoptic segmentation results on S3DIS dataset

| Backbone       | Method | PQ   |
|----------------|--------|------|
| PN++[8]        | ASIS[3]| 68.7 |
|                | Ours   | 68.9 |
| Ours           | ASIS[3]| 68.6 |
|                | Ours   | 69.0 |
| Test on 6-fold CV |     |      |
| PN++[8]        | ASIS[3]| 72.3 |
|                | Ours   | 73.1 |
| Ours           | ASIS[3]| 72.5 |
|                | Ours   | 73.1 |
5. Conclusion

This paper proposes a model framework for panoramic segmentation of point cloud. In order to better capture the features of points, we introduce the correlation information of points in the coding layer. In the output module of panoramic segmentation network, we consider the effect of objective distance to improve the segmentation performance. Experiments prove that the efficiency of our model is better than many similar models. Our encoder and decoder can also be applied to other different feature extraction networks. We hope that our work can contribute to the panoramic segmentation of point cloud, and we also hope that there will be more relevant researches in the future to achieve panoramic segmentation of point cloud as efficient as image panoramic segmentation.

Acknowledgments

The research reported in this paper is supported in part by the National Natural Science Foundation of China (No. 91746203 and No. U1813217), and the project of the Intelligent Ship Situation Awareness System.

References

[1] A. Kirillov, R. Girshick, K. He, and P. Dollar, “Panoptic feature pyramid networks,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[2] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang, “An end-to-end network for panoptic segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[3] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, “Associatively segmenting instances and semantics in point clouds,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[4] G. Lin, A. Milan, C. Shen, and I. Reid, “Refinenet: Multi-path refinement networks for high-resolution semantic segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[5] L. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” CoRR, vol. abs/1706.05587, 2017.
[6] G. Lin, C. Shen, A. van den Hengel, and I. Reid, “Exploring context with deep structured models for semantic segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 6, pp. 1352–1366, June 2018.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3d classification and segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5099 – 5108.

[9] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, “Pointcnn: Convolution on x-transformed points,” in Advances in Neural Information Processing Systems 31, S. Bengio, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 820 – 830.

[10] M. Jiang, Y. Wu, and C. Lu, “Pointsift: A sift-like network module for 3d point cloud semantic segmentation,” CoRR, vol. abs/1807.00652, 2018.

[11] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, “Spidercnn: Deep learning on point sets with parameterized convolutional filters,” in The European Conference on Computer Vision (ECCV), September 2018.

[12] W. Wu, Z. Qi, and L. Fuxin, “Pointconv: Deep convolutional networks on 3d point clouds,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

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

[14] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[15] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully convolutional instance-aware semantic segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[16] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[17] Y. Yan, Y. Mao, and B. Li, “Second: Sparsely embedded convolutional detection,” Sensors, vol. 18, no. 10, 2018.

[18] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: Fast encoders for object detection from point clouds,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[19] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum PointNets for 3d object detection from rgb-d data,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[20] J. Hou, A. Dai, and M. Niessner, “3d-sis: 3d semantic instance segmentation of rgb-d scans,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[21] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun, “Deep parametric continuous convolutional neural networks,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[22] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, “Panoptic segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[23] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, “3d semantic parsing of large-scale indoor spaces,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[24] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3d shapenets: A deep representation for volumetric shapes,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[25] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M.
Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “Shapenet: An information-rich 3d model repository,” CoRR, vol. abs/1512.03012, 2015.

[26] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In CVPR, 2012.

[27] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In CVPR, 2013.