remote sensing

MDPI

Article

Unstructured Road Segmentation Based on Road Boundary Enhancement Point-Cylinder Network Using LiDAR Sensor

Zijian Zhu [1], Xu Li [1,*], Jianhua Xu [2], Jianhua Yuan [3] and Ju Tao [2]

[1] School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China; zzzzzzj110@gmail.com

[2] Xinxing Cathay International Group, Beijing 100020, China; xjh2672@163.com (J.X.); ceratj@163.com (J.T.)

[3] Traffic Management Research Institute, Ministry of Public Security, Wuxi 214151, China; Yjh1308@163.com

* Correspondence: lixu.mail@163.com

Abstract: The segmentation of unstructured roads, a key technology in self-driving technology, remains a challenging problem. At present, most unstructured road segmentation algorithms are based on cameras or use LiDAR for projection, which has considerable limitations that the camera will fail at night, and the projection method will lose one-dimensional information. Therefore, this paper proposes a road boundary enhancement Point-Cylinder Network, called BE-PCFCN, which uses Point-Cylinder in order to extract point cloud features directly and integrates the road enhancement module to achieve accurate unstructured road segmentation. Firstly, we use the improved RANSAC-Boundary algorithm to calculate the rough road boundary point set, training in the same parameters with the original point cloud as a submodule. The whole network adopts the encoder and decoder structure, using Point-Cylinder as the basic module, while considering the data locality and the algorithm complexity. Subsequently, we made an unstructured road data set for training and compared it with existing LiDAR(Light Detection And Ranging) semantic segmentation algorithms. Finally, the experiment verified the robustness of BE-PCFCN. The road intersection-over-union (IoU) was increased by 4% when compared with the best existing algorithm, reaching 95.6%. Even on unstructured roads with an extremely irregular shape, BE-PCFCN also currently has the best segmentation results.

Keywords: LiDAR; unstructured road; semantic segmentation; Point-Cylinder; boundary enhancement

check for updates

Citation: Zhu, Z.; Li, X.; Xu, J.; Yuan, J.; Tao, J. Unstructured Road Segmentation Based on Road Boundary Enhancement Point-Cylinder Network Using LiDAR Sensor. *Remote Sens.* **2021**, *13*, 495. https://doi.org/10.3390/rs13030495

Received: 30 December 2020
Accepted: 29 January 2021
Published: 30 January 2021

1. Introduction

The convenience of transportation improves people's lives and promotes economic development. However, it cannot be ignored that the frequent occurrence of traffic accidents has also caused immeasurable economic losses and human losses. Drivers' overspeed behavior, fatigue driving, and mis-operation are the main causes of traffic accidents, and self-driving technology can effectively solve this problem, according to [1]. Over the past few years, there has been a rapid development in the self-driving field. Simultaneously, road segmentation is a crucial step in realizing self-driving. In general driving scenarios, there are mainly two kinds of road. Structured roads usually refer to highways or some roads, which are highly structured with clear road markings, such as boundary lines. Unstructured roads refer to roads with no artificial markings or few road markings [2,3]. Although there are many studies on the road segmentation of structured roads [4–7], few studies on unstructured road segmentation exist. In unstructured road scenes, there are no lane lines, no obvious road boundaries, and even the road's shape is unknown or partially damaged. When compared with structured roads, such unstructured roads are more difficult to segment and are more likely to cause traffic accidents. Simultaneously, it is impossible to directly apply structured road segmentation algorithms to unstructured road scenes. As a result, in response to the complex unstructured road scene, the study of unstructured road segmentation is even more necessary. Moreover, this article mainly solves the road segmentation problem on these kinds of unstructured roads in the urban environment.

Cameras and LiDAR (Light Detection And Ranging) sensors are commonly used sensor devices in self-driving scenarios. There are many methods to use the camera for unstructured road segmentation. However, the color and texture features that are used by the camera sensor are not robust enough [8]. The camera will fail at night, and it will be negatively affected during the day when there is reflection. Unlike cameras, LiDAR can obtain more accurate expressing scene features by directly obtaining point cloud data, and it is an active sensor that is independent lighting conditions. Therefore, LiDAR sensors are widely used in various self-driving scenarios, such as lane marking extraction and lane width estimation [9], positioning system [10], target detection, and semantic segmentation. In the LiDAR coordinate system, the LiDAR sensor itself is the coordinate origin, and the XY-plane is parallel to the road surface. The positive x-direction is the vehicle's forward direction, and the XYZ coordinate system follows the right-handed coordinate system. Some of the methods use LiDAR to perform semantic segmentation in unstructured road scenes. However, most of these methods project the LiDAR point cloud into a bird's-eye view or a spherical projection view, which loses the feature of one dimension [11]. At the same time, most of the current methods only perform feature extraction for a particular unstructured road scene, such as off-road scenes and rural road scenes. Therefore, these methods have poor adaptability. Some algorithms [4,11,12] directly extract point cloud features among the structured road and they have achieved excellent results on the KITTI dataset [13]. However, few studies have applied them directly to unstructured road segmentation. This is because most of these algorithms focus on small-scale point cloud feature extraction without optimizing the features of unstructured roads, resulting in inaccurate segmentation and the inability to guarantee the calculation speed in larger scenes.

In this work, we focus on LiDAR sensors for unstructured road segmentation. We found that, in unstructured road scenes, buildings or vegetation often determine the boundaries of the roads. Therefore, the road boundary must have some features of height changing. This article uses the network to strengthen this feature in order to obtain accurate unstructured road boundary segmentation results. The LiDAR sensor outputs point cloud information of the three-dimensional world, providing such a kind of height information. Consequently, we can directly use LiDAR point cloud data for feature extraction and road semantic segmentation without projecting the point cloud.

This paper proposes a point-wise deep learning network with boundary enhancement, called BE-PCFCN (boundary enhancement Point-Cylinder CNN), which can achieve a good result in an unstructured road semantic segmentation task. When compared with other algorithms for semantic segmentation of unstructured roads, the algorithm shown in this paper fully extracts the point cloud's height features and it has better performance and more robustness in various unstructured road scenes. Whether on unmarked urban roads or irregularly shaped roads, or even partially damaged roads, BE-PCFCN has good segmentation results. In particular, the accurate segmentation of road boundaries provides a guarantee for driving safety.

The contribution of this work mainly lies in three aspects:

1. In unstructured road semantic segmentation, we directly perform feature extraction on point clouds while using the Point-Cylinder module instead of using projection methods. In this way, the point cloud information can be fully utilized, which makes the segmentation result more accurate.
2. We propose a network structure with boundary enhancement. The accuracy of road boundary segmentation can be enhanced by calculating the boundary point cloud above the original point cloud road plane and then putting it into the neural network to compensate for the resulting feature map.
3. The proposed method performs better in unstructured road scenes when compared with some open-source semantic segmentation algorithms.

The paper is organized, as follows: First, Section 2 introduces the related works. Section 3 is the introduction of methodology and implementation details. Section 4 shows some of the experimental results and comparisons. Finally, we draw a conclusion in Section 5.

2. Related Works

2.1. Structured Road Segmentation Method

Road semantic segmentation is an active research area in the field of self-driving. LiDAR and camera sensors are usually used for road semantic segmentation. Some camera-based algorithms depend on the assumption of global road conditions, such as boundaries [14], lane line [15], or vanish points [16]. Furthermore, some stereo camera-based methods [15] use depth information to help unstructured road drivable area extraction. Despite achieving good performance, camera-based methods are easily affected by changing illumination. LiDAR can effectively solve the shortcomings of the camera. It can be used for higher-precision road semantic segmentation due to its large amount of information in space and the properties of active laser emission not affected by illumination.

The use of LiDAR for road segmentation mainly includes the direct use of point clouds for feature extraction or projection into a 2D projection pseudo image for feature extraction. [4–6] focus on converting the 3D point cloud to 2D grids to enable the usage of 2D Convolutional Neural Networks. SqueezeSeg [7], SqueezeSegv2 [17], and RangeNet++ [6] utilize the spherical projection, which converts the point cloud to a range image and adopts the 2D convolution network on the pseudo image for point cloud segmentation. PolarNet [18] uses bird's-eye view projection for feature extraction.

Although the projection method is fast, it will lose part of the LiDAR point cloud information. PointNet [12], KPConv [11], and RandLA-Net [4] directly use point clouds for feature extraction. Although the accuracy is higher, the problem of slow inference speed is inevitable. Aiming at the efficiency of 3D convolutional networks, increasing numbers of researchers have proposed methods to improve efficiency [16], and Lei et al. [19] proposed reducing the memory footprint of the volumetric representation while using octrees where areas with lower density occupy fewer voxel grids. Liu et al. [20] analyzed point-based and voxel-based methods' bottlenecks and proposed point-voxel convolution. Graham et al. [21] proposed sparse convolution to speed up volume convolution by keeping active sparseness and skipping calculations in inactive regions. SPVNAS [22] presents 3D Neural Architecture Search to efficiently and effectively search the optimal network architecture over this diverse design space.

These methods are based on the encoder and decoder structure [23–26] and they achieve good results in structured road segmentation. However, these algorithms, such as KPConv, RandLA-Net, and SPVNAS, are all aimed at point cloud semantic segmentation. Additionally, there is a lack of relevant experiments to prove whether these algorithms can be applied to unstructured roads directly. On the one hand, because of the lack of relevant data sets; on the other hand, they have not been optimized for the characteristics of unstructured roads.

2.2. Unstructured Road Segmentation Method

The existing road segmentation methods are mainly for structured roads, but the problems in the unstructured road are quite different. When compared with structured roads, unstructured roads have no lane lines, no obvious road boundaries, and no regular shape. Some of the algorithms use traditional feature engineering methods, such as extract radial features and transverse features for road segmentation [27]. However, the mainstream approach is to use deep learning algorithms for road segmentation. Because of the limitation of the dataset, most of these deep learning methods use some techniques to reduce the demand for data. Gao et al. [8] use the driving trajectory for area growth to automatically generate the drivable area for weakly supervised learning to segment road. Although this work dramatically reduces the workload of labeling data, the final result is rough, and it cannot accurately segment the road boundary. Holder et al. [28] use a small number of unstructured road pictures to perform transfer learning that is based on structured roads. However, unstructured roads often have similar texture characteristics to the surrounding environment, so it is not easy to achieve good results in various scenarios.

These algorithms mainly use cameras or projection LiDAR point clouds for road segmentation and propose effective solutions for limited datasets. However, the disadvantage is that the camera will fail at night, and the LiDAR projection will cause information loss. Therefore, current unstructured road segmentation algorithms have limited segmentation accuracy, especially for road boundaries, which cannot be accurately distinguished from the surrounding environment.

This work proposes a method that directly utilizes LiDAR point cloud data with a Point-Cylinder structure, which makes full use of the point cloud features and avoids the camera's shortcomings at night. We also propose a boundary enhancement method that uses the road boundary's height change features to more effectively segment the road.

3. Methods

3.1. Network Overview

Our network that is based on road boundary enhancement with a Point-Cylinder module is called BE-PCFCN. The proposed network input consists of two parts, the original point cloud and the point cloud after the boundary extraction, as shown in Figure 1. There are currently two methods, point-based and voxel-based, in order to directly extract the 3D features of the point cloud. The voxel-based method requires O(n) random memory accesses, in which "n" is the number of points. This method only needs to iterate over all the points once to scatter them to their corresponding voxel grids. While, for the point-based method, gathering all of the neighbor points requires at least O(kn) random memory accesses, in which "k" is the number of neighbors. To conclude, the point-based method has irregularity and poor data locality, and the voxel-based method has high algorithm complexity. We use the Point-Cylinder substructure after considering the advantages and disadvantages of these two methods. This structure reduces the memory consumption while performing the convolutions in voxels in order to reduce the irregular data access. On this basis, it simultaneously transforms the original Cartesian coordinate system into a Cylinder [29] coordinate system. LiDAR point cloud distributes as the closer the more, the far the fewer, so such a structure can make the distribution of points more uniform and reduce the proportion of empty grids, thus greatly reducing the amount of computation.

The network structure adopts the encoder–decoder framework with a skip connection module, which can combine high-level semantics and low-level fine-grained surface information in order to better achieve the segmentation result. The original point cloud passes through the boundary extraction algorithm to obtain the rough road boundary point cloud set. In this way, it can be put into the same encoder–decoder network for feature extraction. After the rough boundary point cloud is passed through the encoder–decoder, the output feature map is superimposed on the original point cloud decoder's output, and the final probability distribution is output through the fully connected layer.

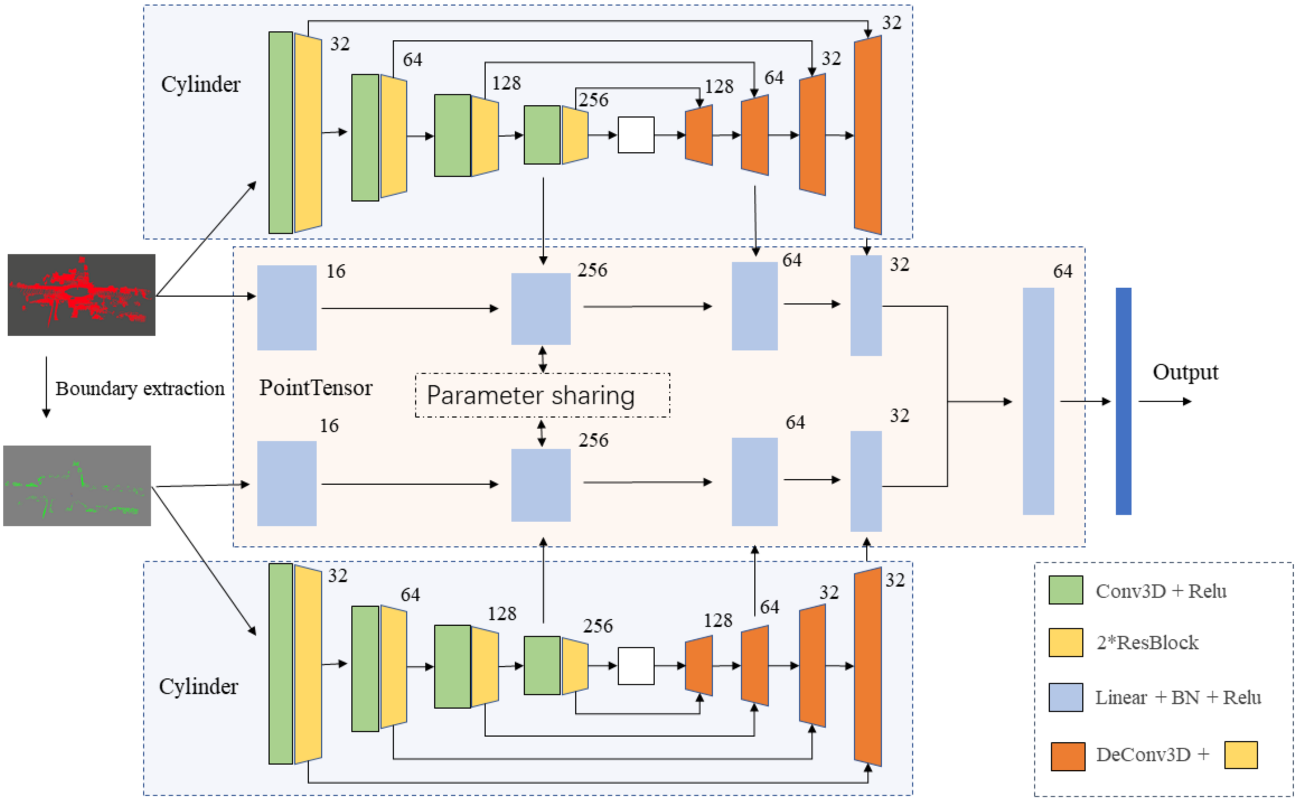

Figure 1. Overall architecture of the proposed boundary enhancement Point-Cylinder CNN (BE-PCFCN). The original point cloud and the point cloud extracted by the boundary enhancement algorithm are input into the network based on Point-Cylinder, respectively. After concatenating the last layer of feature maps, the segmentation results are output through the fully connected layer.

3.2. Road Boundary Enhancement Module

It is necessary to roughly extract the area that may be the road boundary in advance in order to enhance the road boundary in the neural network and to segment the real road.

However, most of the current algorithms for calculating the boundary of point clouds are for small indoor objects, and the calculation efficiency is low in large outdoor scenes. This article mainly focuses on the boundary of the ground, where the height above the ground changes. Therefore, we propose a fast algorithm, which can quickly extract a possible point cloud set of rough ground boundaries for the scene.

Firstly, it is necessary to calculate the plane of the road. Based on the road plane, we can easily find continuous points and whose z value is above this plane. Such a point set can be considered to be a rough road boundary. An assumption is used here: it is assumed that, in the common scene, the ground plane has the largest number of LiDAR points. However, if some conditions make the assumption false, then the point cloud will be enhanced at the unknown position. Because the point cloud label will not change, even if some positions' weight is enhanced, it will not have too much negative impact on the segmentation result. There is no plane larger than the ground in an unstructured road scene, so the plane that is found by the RANSAC (Random Sample Consensus) algorithm [30] can be considered the ground. We can easily extend the RANSAC fitting plane algorithm in three-dimensional (3D) space, according to the RANSAC fitting straight line algorithm. After the plane is obtained, the average z value of the road plane in the LiDAR coordinate system can be calculated. The set of points within a certain threshold above the road plane is considered the possible boundary. The pseudo-code of the improved 3D RANSAC-Boundary algorithm is shown below.

Algorithm 1 3D RANSAC-Boundary algorithm

```
init inliersResult, inliers, maxIterations, diatanceTol, zTol
for i in range(maxIterations):
    while (inliers.size() < 3):
        inliers.append (random points)
    end while
    calculate the plane using the first three point in inliers to get parameters A, B, C, D
    for j in range(cloud.size()):
        if distance (cloud[i], plane) < diatanceTol:
            inliers.append(cloud[i])
        end if
    end for
    if (inliers.size() > inliersResult.size()):
        inliersResult = inliers
    end if
end for
for (point:(cloud-inliersResult))
    if ((the distance between point and last plane in the previous for cycle) < zTol)
        Results.append(point)
    end if
end for
return Results
```

A rough road boundary point cloud collection can be calculated when the algorithm is done. The collection is used as the input of a network branch to enhance the feature map of point cloud segmentation. If the total number of points is N, then the time complexity of plane segmentation is O(kN), where k represents the number of iterations. The time complexity of extracting the boundary part is O(N), and the total time complexity is O(kN), which can run in real-time.

Figure 2 shows the result of the algorithm; all of the point clouds that may be the road's boundary are extracted and placed in the boundary point collection. After that, this rough road boundary point set is used as the input into the network, which shares the same parameters with the original point cloud.

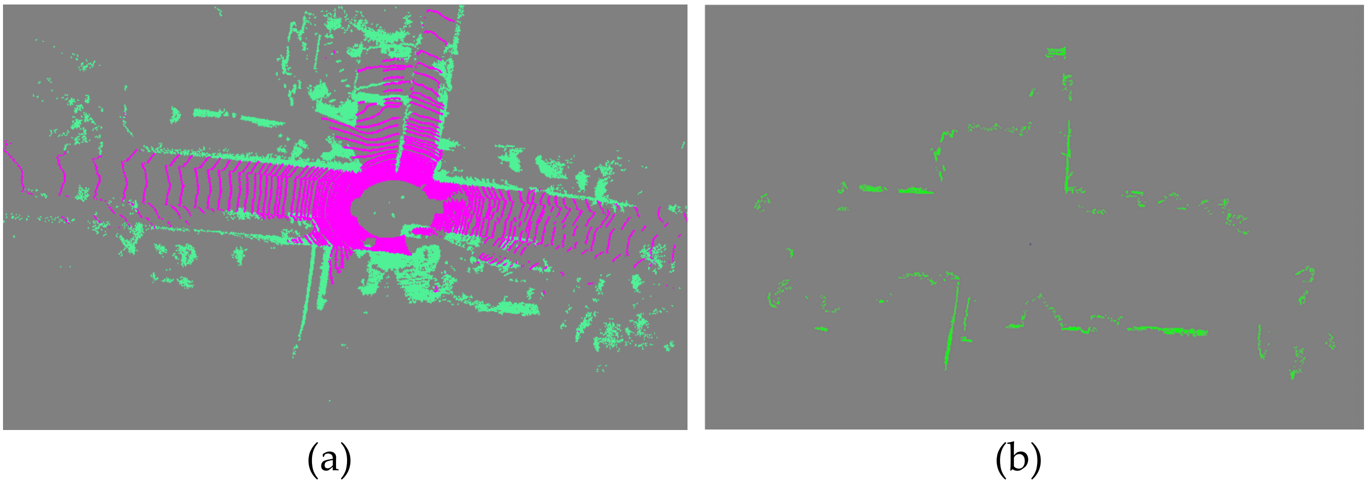

Figure 2. The result of rough road boundary extraction. (**a**) is the original point cloud before boundary extraction, where pink represents roads and green represents others. (**b**) is the point cloud after the boundary extraction algorithm. The green points represent the extracted boundary.

3.3. Network Structure

The basic structure of the network is the encoder and decoder with skip connections of two branches, which is the original point cloud stream and the rough road boundary stream. The encoder and decoder deal with a sub-module, called Point-Cylinder. This design can extract point cloud features from two angles, point-wise and cylinder-wise, while considering memory consumption and point cloud locality. At the end of the network, the two branches' feature maps are concatenated to generate a total feature map. The final segmentation result comes after the fully connected layer. Figure 1 shows the overall structure. Additionally, the following will explain in detail from three perspectives: Point-Cylinder substructure, two-branch 3D encoder–decoder structure, and Loss function.

3.3.1. Point-Cylinder Substructure

How to encode the point cloud for feature extraction is the first consideration for deep learning using LiDAR point cloud data directly. The original data of the point cloud are an unordered sequence, and each element of the sequence contains four dimensions of (x, y, z, i). Therefore, the algorithm complexity of calculating directly on the point cloud will not be too high, but there will be problems of poor spatial locality and irregularity. Our network needs this kind of spatial locality to obtain the features of point cloud z-value changes and segment the road and the surrounding environment. In this regard, voxelizing the point cloud will effectively solve this problem. However, the voxel-based representation requires very high resolution in order to not lose information, which will cause huge computational overhead. Therefore, this paper adopts the point-voxel [22] idea to design a substructure to extract features. Simultaneously, a cylinder is used instead of the voxel to solve voxel sparsity, which makes the point cloud distribution more legitimate with the Velodyne LiDAR data, effectively reducing the number of empty voxels and unnecessary calculations.

Figure 3 shows the Point-Cylinder substructure that was proposed in this article. The input point cloud is applied to Multi-layer Perception for feature extraction. Simultaneously, the point cloud is converted from Cartesian coordinate into a cylinder coordinate and voxelize to the cylinder [29]. The origin of the cylinder coordinate system is consistent with the LiDAR coordinate system's origin. The polar axis is parallel to the x-axis, and the z-axis is the same as the LiDAR coordinate system. Note that the point features $\{f_k\}$ remain unchanged during the transformation. We denote the cylinder coordinates as $\{\hat{p}_k\}$. We use the following formula to transform the point cloud $\{(f_k, \hat{p}_k)\}$ into Cylinder grid $\{Cy_{u,\varepsilon,v}\}$.

$$Cy_{u,\varepsilon,v,c} = \frac{1}{N_{u,\varepsilon,v}} \sum_{k=1}^{n} \left[\!\left[floor(\hat{\ell}_k \times r) = u, floor(\hat{\theta}_k \times \alpha) = \varepsilon, floor(\hat{z}_k \times r) = v \right]\!\right] \times f_{k,c} \tag{1}$$

where r denotes the voxel resolution, α denotes the angular resolution, $[\![\cdot]\!]$ is 0-1 operator indicating whether the coordinate $\hat{p}_k$ belongs to the voxel grid, $f_{k,c}$ denotes the cth channel feature corresponding to $\hat{p}_k$, and $N_{u,\varepsilon,v}$ is the number of points that fall in the voxel grid $Cy_{u,\varepsilon,v}$. In the implementation of the algorithm in this paper, the resolution r is set to 0.05 and the angular resolution α is set to $\pi/180$, so the angle can be directly rounded in the calculation.

Subsequently, we use a sequence of residual Sparse Convolution blocks [31] to extract features and apply the batch normalization and the nonlinear activation function after that. More specifically, the Sparse Convolution Tensor for Cylinder can be written as

$$C = \begin{bmatrix} u_1 & \varepsilon_1 & z_1 & t_1 & b_1 \\ & & \vdots & & \\ u_N & \varepsilon_N & z_N & t_N & b_N \end{bmatrix}, F = \begin{bmatrix} f_1^T \\ \vdots \\ f_N^T \end{bmatrix} \tag{2}$$

where b_i and f_i are the batch index and the feature associated to the i-th coordinate. Subsequently, we do the Sparse Convolution on feature Tensor F:

$$F_u^{out} = \sum_{i \in N^D(u, C^{in})} W_i F_{u+i}^{in}, u \in C^{out} \tag{3}$$

where $N^D(u, C^{in}) = \{i | u + i \in C^{in}, i \in N^D\}$ is a set of offsets from the current Cylinder center u that exist in C^{in}. C^{in} and C^{out} are the input and output coordinates of Sparse Tensor. W_i is the convolution kernel weight of i-th coordinate.

Finally, we use trilinear interpolation to transform the Cylinder grids into point cloud feature format and then concatenate the features extracted from the two parts to obtain the final output.

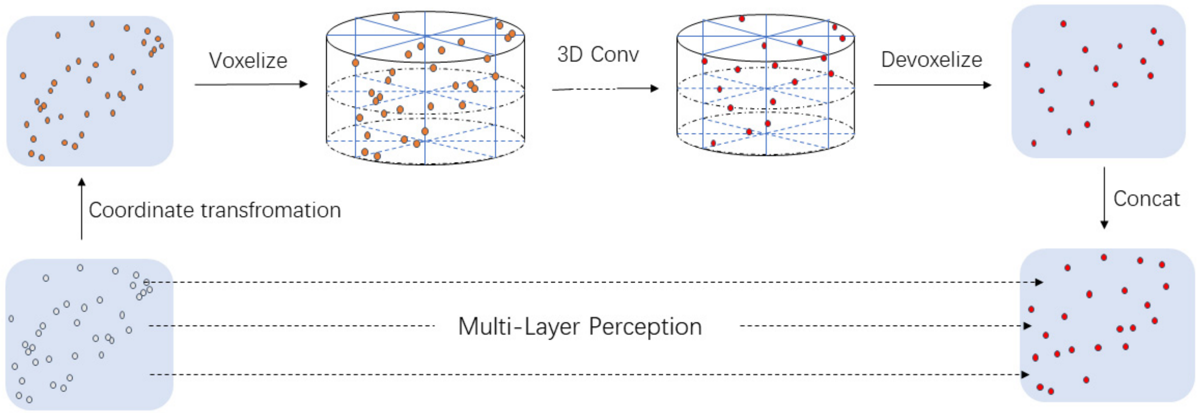

Figure 3. The structure of Point-Cylinder substructure. In this module, the original point cloud is first converted from the Cartesian coordinate system to the cylinder coordinate system, and the Cylinder partition is performed. Based on this, three-dimensional (3D) convolution operations are performed to extract the features. Finally, trilinear interpolation is used to transform the Cylinder grids into point cloud feature format and concatenate the features extracted from the original point cloud.

3.3.2. Two-Branch 3D Encoder–Decoder Structure

This paper uses the two-branch structure to perform feature extraction and feature concatenate on the original point cloud and the point cloud after boundary extraction. The two-branch network uses the same parameters to enhance the boundary of the road on the feature map. In this way, the road's boundary will be strengthened during the training process in order to obtain greater weight. This boundary information is a kind of prior value that can help the network better distinguish the adjacent objects. We train the original point cloud and the boundary point cloud in two branches and perform a feature-level fusion, improving the network's response to the road boundary on the fused feature map. Especially in the unstructured road scene, the two-branch structure is more necessary to enhance the road boundary, because it is easily confused with the surrounding environment.

The network adopts the encoder and decoder structure, while using the original point cloud and the extracted rough boundary point cloud as input. Based on the Point-Cylinder substructure that was proposed in the previous section, skip connections are added when encoding and decoding the cylinder, which is shown in Figure 1. This can combine high-level semantics and low-level fine-grained surface information to better achieve the segmentation result. The encoder and decoder adopt an asymmetrical design. For PointTensor, the up-sampling stage requires more information from the cylinder feature map to ensure spatial locality. Therefore, we only concatenate the features of PointTensor and Cylinder once in the down-sampling and concatenate twice in the up-sampling phase.

For the input of the extracted rough road boundary point cloud, the parameters (kernel weights and biases) of the Conv layer of two streams are shared at the training stage. For the sharing parameters, the road boundary point cloud must be performed, as follows: a) expand the PointTensor to the same shape as the original point cloud; b) zero the element at the expanded position.

For the encoder part of the cylinder, we use 3D sparse convolution and two-layer ResBlock as the basic unit, which can effectively perform feature extraction. DeConv3D and two-layer ResBlock are used for upsampling with skip connection to combine high-level semantics and low-level fine-grained surface information in the decoder part. For the point cloud data, the fully connected layer, Batch Normalization, and ReLu are used for feature extraction. Based on this, the feature map that is generated by the corresponding layer of the cylinder is concatenated to obtain the feature vector of the point cloud data.

Finally, the two networks' output feature maps are concatenated and, after the fully connected layer, the final segmentation result can be obtained.

3.3.3. Loss Function and Optimizer

For network optimization, weighted cross-entropy loss [32] and Lovász-Softmax [33] are combined as the loss function to maximize the IoU (intersection-over-union) score for classes. Cross-entropy loss is suitable for multi-classification problems, which is the semantic segmentation of LiDAR point clouds in this article. Lovasz loss is a Lovaze extension of IoU Loss, which performs better. It is suitable for multi-class segmentation tasks using IoU as the evaluation index. Therefore, combining the two loss functions can make the results of the entire network more accurate. The two parts have the same weight, and the final loss function can be expressed as:

$$L_{total} = -\sum_{i} 1/\sqrt{v_i} P(y_i) \log P(\hat{y}_i) + 1/|C| \sum_{c \in C} J(e(c)) \tag{4}$$

where v_i is the frequency of each class, $\mathrm{P}(\hat{y}_i)$ and $\mathrm{P}(y_i)$ are the corresponding predicted and ground truth probability, J is the Lovász extension of IoU, $e(c)$ is the vector of errors for class c, and p is the number of pixels considered. The model uses SGD (Stochastic gradient descent) as the optimizer, and the initial learning rate is set to 1e-4.

4. Experiments and Results

In the experimental part, we first introduce the dataset that was used for training and testing and the metrics that we used to evaluate the model. Subsequently, the promotion of adding road boundary enhancement to the network is verified. Simultaneously, we compare the results of BE-PCFCN and some typical algorithms on the KITTI [13] test set and our unstructured road test set.

4.1. Dataset

We collected some unstructured scene data and labeled them for training to evaluate the segmentation performance of the proposed network. In our work, irregular outdoor parking lots and small roads lacking lane lines are collected as a dataset of unstructured road scenes for labeling. We build a typical unstructured road dataset while using the vehicle with a Velodyne VLP-32C LiDAR and a front-view monocular camera. LiDAR data are used for labeling and training, and camera data are only used for visualization. During data collection, the vehicle is driven by a human, and all of the devices are synchronized in time. Finally, we selected 1000 frames of LiDAR point cloud data in two scenes for labeling. 70% of them are used for training, 10% are used as a validation set, and 20% are used as a test set.

Structured roads and unstructured roads have some elements that are the same, such as vehicles, people, and vegetation. Therefore, we combined our dataset and KITTI dataset for training and validation. The KITTI test set and our own labeled test set are separately tested. Specifically, according to the characteristics of the elements in the scene and the number of corresponding point clouds, we divide all of the points into the following categories: road, buildings, vegetation, vehicles, people, and others. We did not make any changes to the KITTI dataset for separate training. In the fusion data set, we map 'car', 'truck', 'bicycle', 'motorcycle', and 'other vehicle' to 'vehicles'; map 'bicycle', 'motorcyclist', and 'person' to 'people'; map 'road', 'parking', 'sidewalk', 'other ground' to 'road'; map 'fence', 'vegetation', 'trunk', and 'terrain' to 'vegetation'; 'building' remains unchanged; and, map other points to 'others'. This makes the data set to match the data set that we labeled. All of the experiments were performed on GeForce RTX 3090 GPU.

4.2. Metrics

For evaluating the algorithm performance, we adopt the following criteria:

$$IoU = \frac{T_p}{T_p + F_p + F_N} \tag{5}$$

$$recall = \frac{T_p}{T_p + F_N} \tag{6}$$

$$precision = \frac{T_p}{T_p + F_p} \tag{7}$$

where T_p is the number of ground points detected correctly, F_N is the ground points not detected, and F_p is the ground points falsely detected. We use IoU to judge the difference between the segmented result and the ground truth, $recall$ to analyze the missed detection rate, and $precision$ to analyze the false detection rate.

4.3. Experimental Results

We selected two scenarios to verify the adaptability of the method proposed in this paper: general unstructured roads with clear boundaries and unstructured roads with irregular edges for testing. We tested BE-PCFCN, BE-PCFCN without boundary enhancement in this paper and some representative semantic segmentation algorithms. All of the training parameters of the two networks are the same. SGD optimizer is used with the initial learning rate of 0.001, batch size of 4, distanceTol of 0.05, zTol of 1.0, training epochs of 50, and the same training set and validation set are used.

Currently, SPVNAS is the best algorithm for semantic segmentation among open source codes [22]. Accordingly, we compare BE-PCFCN with it. Because SPVNAS initially used the weight trained on the KITTI dataset, which is different from our dataset. Therefore, we use the default training parameters to train SPVNAS on our dataset. Based on the obtained new weight file, it is compared with the BE-SPCNN that is proposed in this paper.

We first selected some simple and more common road scenes for comparison. Although all of the algorithms have good results, the BE-PCFCN can more accurately segment the road, as shown in Figure 4. Especially for road boundaries, the enhancement algorithm we proposed can make the network learn boundary features more effectively and reduce the misclassification of roads and surrounding environments. Simultaneously, BE-PCFCN has advantages in road segmentation when compared with other LiDAR point cloud semantic segmentation algorithms. With the boundary enhancement module, the network can effectively learn the road boundary feature, reducing road misclassification.

We compared BE-PCFCN with other LiDAR semantic segmentation algorithms on the KITTI test set to illustrate the adaptability of our algorithm in road segmentation, which is shown in Table 1. Because we cannot obtain the codes of some papers, all of the comparison data are based on their papers. We use buildings and vegetation to enhance the road's boundary, and their test IoU are also compared.

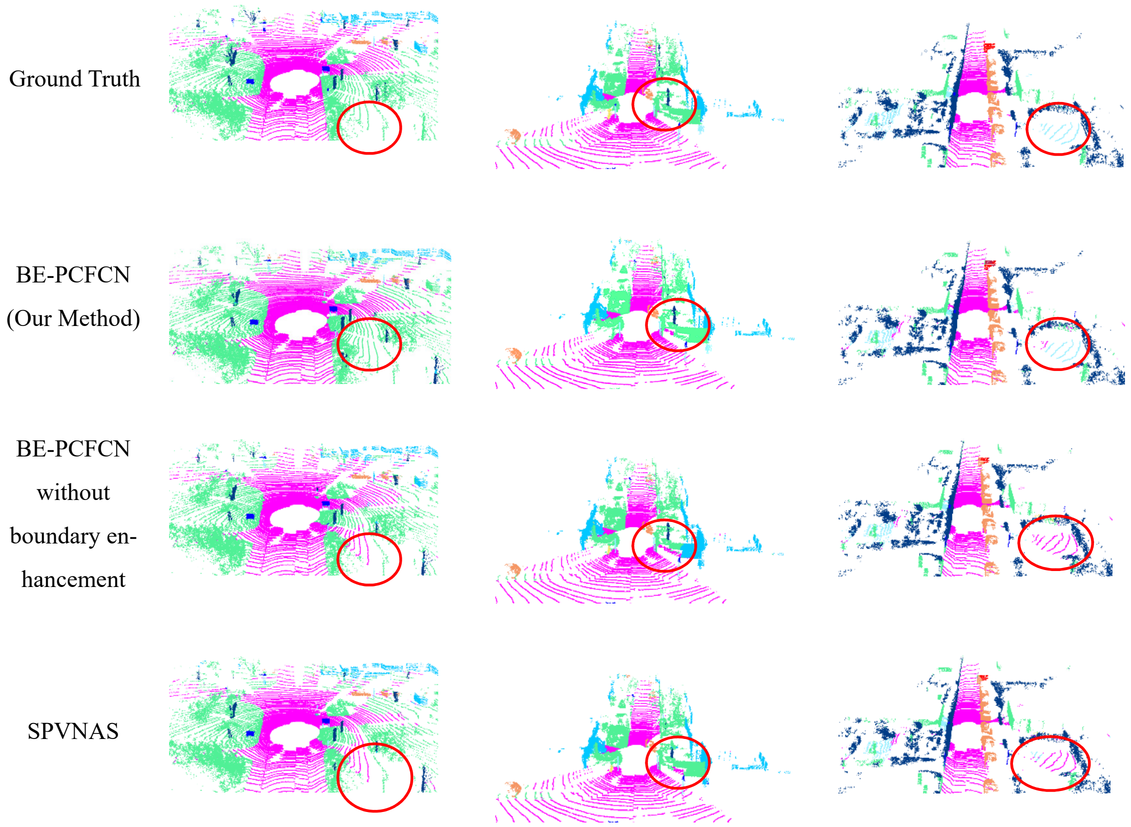

Figure 4. Some outputs of different models on simple unstructured road scenes. Among them, magenta represents roads, green represents vegetation, and blue represents buildings.

Table 1. Building intersection-over-union (IoU), Road IoU, and Vegetation IoU of different models on KITTI dataset.

	Road IoU	**Buildings IoU**	**Vegetation IoU**
PointNet [12]	61.6	41.4	31.0
PointNet++ [34]	72.0	62.3	41.5
RandLA-Net [4]	90.4	86.9	81.7
Kpconv [11]	88.8	90.5	84.8
SalsaNext [5]	91.7	90.2	81.8
TORANDONet [35]	89.7	91.3	85.6
SPVNAS [22]	90.2	**91.6**	**86.1**
Cylinder3D [29]	91.4	91.0	85.4
BE-PCFCN without boundary enhancement	90.4	90.8	80.7
BE-PCFCN	**95.6**	88.4	80.3

Table 1 shows that BE-PCFCN has significantly improved the road segmentation result, which is 4% higher than the previous best result of road segmentation. However, the road enhancement module inevitably weakens the network's learning of other parts, resulting in a decline in both Buildings IoU and Vegetation IoU. When we further looked at the Fp in the calculation process, we found that falsely detected building points were more likely to be classified as vegetation. Nevertheless, in the actual driving scene, when the road is accurately segmented as the drivable area, the vehicle can safely drive on the road. At this time, it does not matter whether the surrounding environment is vegetation or buildings. Simultaneously, BE-PCFCN without boundary enhancement has good results, and the road IoU also exceeds 90%, even when there is no significant height difference between the surrounding environment and the road.

In the previous simple road segmentation test, BE-PCFCN has good performance. In order to verify the robustness of BE-PCFCN, the following tests will be carried out in more complex unstructured road scenes. We selected roads with irregular shapes, uneven roads, and roads with many obstacles for testing. The KITTI dataset uses a 64-beams LiDAR sensor to verify whether our algorithm can maintain excellent segmentation results with less information; we also tested it with the 32-beams LiDAR sensor.

On the more complex unstructured road test set, BE-PCFCN can also have good road segmentation results. The example shown in Figure 5 covers a scene where the road has no obvious boundaries or is partially damaged and it includes a 32-beams LiDAR as input. It can be seen that in complex unstructured roads, BE-PCFCN is more accurate and will not segment the outside of vegetation and buildings into road areas. However, SPVNAS will have this kind of mis-segmentation error. When the input is changed to a 32-beams LiDAR, all of the algorithms' segmentation results are significantly worse due to the limited information and sparse point cloud. However, BE-PCFCN has the best result with more continuous and complete road surface segmentation and clearer road boundary. Furthermore, the mis-segmentation is lower than that of the SPVNAS.

Subsequently, we use our own 32-beams LiDAR dataset as the training set, train BE-PCFCN and SPVNAS, test on our unstructured road test set, and calculate Road IoU, Road Recall, and Road precision. Table 2 shows the final results.

Table 2. Road IOU, Road Recall, and Road precision of SPVNAS and BE-PCFCN on our unstructured road data set.

	Road IOU	**Road Recall**	**Road Precision**
SPVNAS	75.0	96.0	77.3
BE-PCFCN	77.2	97.8	78.5

The result of road segmentation decreased significantly in a difficult scene. After further analysis, it is found that the **precision** is significantly reduced. That means real road points can be segmented, but non-road points are misclassified into road points. When 32- beams LiDAR is used as input, the ground point cloud is too sparse, the curbs between the road and the grass are sometimes not scanned by the LiDAR. In this case, the grass and road will be considered on the same continuous plane, which reduces segmentation precision. When the LiDAR point cloud scans enough information, the segmentation results will become accurate.

When comparing the results shown in Table 2 with Table 1, with 32-beams LiDAR as input, the algorithm in this paper cannot very accurately segment the road, and the result is not as good as 64-beams LiDAR. The segmentation result is still better than the other algorithm. The 32-beams LiDAR can be used to assist driving in combination with cameras and other sensors for self-driving tasks.

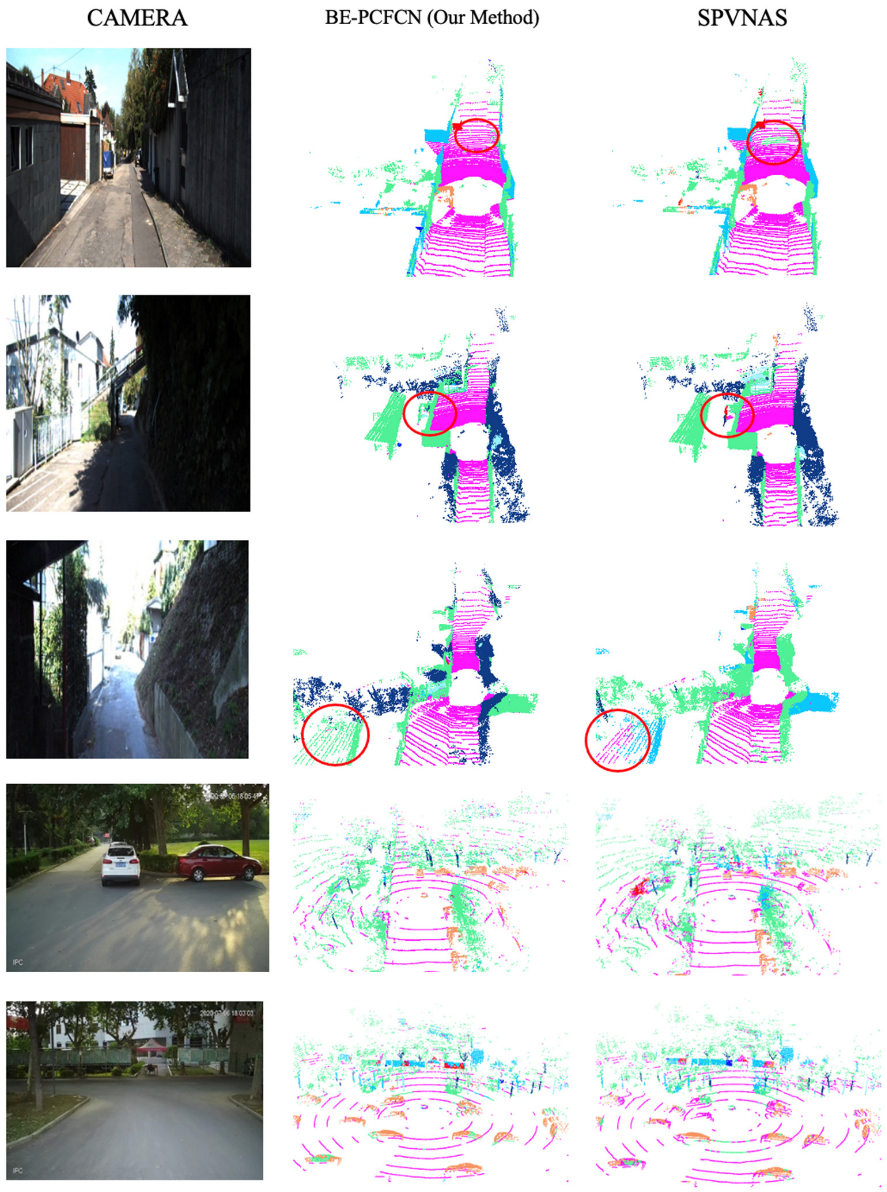

Figure 5. Some output of BE-PCFCN and SPVNAS on hard unstructured road scenes under 32-line LiDAR as input.

4.4. Summary

Based on the experiment above, we can draw the following conclusions:

(1) The BE-PCFCN model performs well in the task of road segmentation. The IoU of road segmentation exceeds the current best algorithm by 4% on the KITTI dataset.

(2) In the simple unstructured road scenes, BE-PCFCN can accurately segment the road and environment around the road with the boundary enhancement module.

(3) In the complex unstructured roads, BE-PCFCN has obvious advantages over other algorithms. However, when the input data are a 32-beams LiDAR point cloud, the point cloud on the ground will become sparse. Sometimes the road boundary feature cannot be obtained, which will result in poor segmentation results. However, once enough road boundary features are obtained, the network will still have an excellent output.

To sum up, BE-PCFCN has the best segmentation results on different unstructured road scenes, which has no lane lines, no obvious road boundaries, and the road's shape is even unknown or partially damaged. When using a 64-beams LiDAR, it has strong robustness, and it is suitable for scenes that require high road segmentation accuracy. When conditions are limited and only 32-beams LiDAR can be used, BE-PCFCN can also provide higher segmentation accuracy than other algorithms.

5. Conclusions

This paper has presented a highly robust method, called BE-PCFCN, to achieve reliable and accurate performance in unstructured road segmentation. In our work, the road boundary enhancement module and Point-Cylinder sub-module are designed to adapt to various unstructured road segmentation scenarios because of the shortcomings in the current unstructured road segmentation algorithm.

We divided the test scenarios into the simple unstructured road and the complex unstructured road in order to test separately in the experiment. We compared BE-PCFCN, BE-PCFCN without boundary enhancement, and other LiDAR point cloud semantic segmentation algorithms. In a simple unstructured road scene, BE-PCFCN has excellent performance, which is 4% higher than the previous best result of road segmentation on the road IoU. In more complex unstructured road scenarios, BE-PCFCN also has better performance than other algorithms. Although BE-PCFCN emphasizes the surrounding environment's contribution as the edge of the road, which sometimes confuses the surrounding environment's specific categories, BE-PCFCN is still a robust algorithm and it has higher accuracy than other algorithms.

However, this paper has the limitation that the 3D RANSAC-Boundary algorithm will fail in vertically curved road surfaces. From another perspective, the boundary algorithm will also fail if there is no significant height difference between the road boundary and the environment. In these cases, BE-PCFCN will degenerate into a network without road enhancement.

In the future, we will continue our work, so that the algorithm can segment the road well in more possible unstructured scenarios.

Author Contributions: Conceptualization, X.L. and Z.Z.; methodology, Z.Z.; software, Z.Z.; validation, Z.Z. and J.X.; formal analysis, X.L. and J.Y.; investigation, Z.Z.; resources, Z.Z.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, X.L., Z.Z. and J.T.; supervision, J.X. project administration, X.L. and J.X.; funding acquisition, X.L. and J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R & D Program of China, grant number 2019YFC1511505, the Key R & D program of Jiangsu Province, grant number BE2019106, in part by the National Natural Science Foundation of China, grant number 61973079 and 41904024, and in part by the Program for Special Talents in Six Major Fields of Jiangsu Province, grant number 2017 JXQC-003.

Acknowledgments: The authors would like to thank Peizhou Ni and Zhiyong Zheng for useful suggestions for writing skills.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, H.; Zhao, Y.; Ma, X. Critical Factors Analysis of Severe Traffic Accidents Based on Bayesian Network in China. *J. Adv. Transp.* **2020**, *2020*, 8878265. [CrossRef]
2. Huang, J.; Kong, B.; Li, B.; Zheng, F. A New Method of Unstructured Road Detection Based on HSV Color Space and Road Features. In Proceedings of the 2007 International Conference on Information Acquisition, Jeju, Korea, 9–11 July 2007.
3. Liu, Y.-B.; Zeng, M.; Meng, Q.-H. Unstructured Road Vanishing Point Detection Using Convolutional Neural Networks and Heatmap Regression. *IEEE Trans. Instrum. Meas.* **2021**, *70*. [CrossRef]
4. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 11108–11117.
5. Cortinhal, T.; Tzelepis, G.; Aksoy, E.E. SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving. *arXiv* **2020**, arXiv:2003.03653.

6. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019.
7. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In Proceedings of the ICRA, IEEE International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018; pp. 1887–1893.
8. Gao, B.; Xu, A.; Pan, Y.; Zhao, X.; Yao, W.; Zhao, H. Off-Road Drivable Area Extraction Using 3D LiDAR Data. *arXiv* **2020**, arXiv:2003.04780.
9. Cheng, Y.-T.; Patel, A.; Wen, C.; Bullock, D.; Habib, A. Intensity Thresholding and Deep Learning Based Lane Marking Extraction and Lane Width Estimation from Mobile Light Detection and Ranging (LiDAR) Point Clouds. *Remote Sens.* **2020**, *12*, 1379. [CrossRef]
10. Liu, H.; Ye, Q.; Wang, H.; Chen, L.; Yang, J. A Precise and Robust Segmentation-Based Lidar Localization System for Automated Urban Driving. *Remote Sens.* **2019**, *11*, 1348. [CrossRef]
11. Thomas, H.; Qi, C.R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6411–6420.
12. Qi Charles, R.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
13. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9297–9307.
14. Yuan, Y.; Jiang, Z.; Wang, Q. Video-based road detection via online structural learning. *Neurocomputing* **2015**, *168*, 336–347. [CrossRef]
15. Broggi, A.; Cardarelli, E.; Cattani, S.; Sabbatelli, M. Terrain Mapping for Off-road Autonomous Ground Vehicles Using Rational B-Spline Surfaces and Stereo Vision. In Proceedings of the Intelligent Vehicles Symposium, Gold Coast, Australia, 23–26 June 2013.
16. Wang, P.-S.; Liu, Y.; Sun, C.; Tong, X. Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes. *ACM Trans. Graph.* **2018**, *37*, 1–11. [CrossRef]
17. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In Proceedings of the IEEE International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019; pp. 4376–4382.
18. Zhang, Y.; Zhou, Z.; David, P.; Yue, X.; Xi, Z.; Gong, B.; Foroosh, H. Polarnet: An improved grid representation for online lidar point clouds se-mantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 9601–9610.
19. Lei, H.; Akhtar, N.; Mian, A. Octree Guided CNN With Spherical Kernels for 3D Point Clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9623–9632.
20. Liu, Z.; Tang, H.; Lin, Y.; Han, S. Point-Voxel CNN for Efficient 3D Deep Learning. In Proceedings of the Conference and Workshop on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
21. Graham, B.; Engelcke, M.; van der Maaten, L. 3D Semantic Segmentation With Submanifold Sparse Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9224–9232.
22. Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; Han, S. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
23. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 801–818.
24. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
25. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer Assisted Intervention Society, Munich, Germany, 5–9 October 2015; Springer: New York, NY, USA, 2015; pp. 234–241.
26. Çiçek, Ö.; Abdulkadir, A.; Lienkamp, S.S.; Brox, T.; Ronneberger, O. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. *arXiv* **2016**, arXiv:1606.06650.
27. Liu, T.; Liu, D.; Yang, Y.; Chen, Z. Lidar-based Traversable Region Detection in Off-road Environment. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019.
28. Holder, C.J.; Breckon, T.P.; Wei, X. From On-Road to Off: Transfer Learning Within a Deep Convolutional Neural Network for Segmentation and Classification of Off-Road Scenes. In *Lecture Notes in Computer Science*; Springer Science+Business Media: Berlin/Heidelberg, Germany, 2016; pp. 149–162.
29. Zhou, H.; Zhu, X.; Song, X.; Ma, Y.; Wang, Z.; Li, H.; Lin, D. Cylinder3D: An Effective 3D Framework for Driving-scene LiDAR Semantic Segmentation. *arXiv* **2020**, arXiv:2008.01550.

30. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
31. Choy, C.; Gwak, J.Y.; Savarese, S. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
32. Zhang, Z.; Sabuncu, M.R. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems, Proceedings of the 2018 Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018*; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2018; pp. 8778–8788.
33. Berman, M.; Triki, A.R.; Blaschko, M.B. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4413–4421.
34. Qi, C.R.; Li, Y.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems, Proceedings of the 2017 Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2018; pp. 5099–5108.
35. Gerdzhev, M.; Razani, R.; Taghavi, E.; Liu, B. Tornado-net: Multiview total variation semantic segmentation with diamond inception module. *arXiv* **2020**, arXiv:2008.10544.