EXTRACTING VEHICLES IN POINT CLOUDS OF UNDERGROUND PARKING LOTS BASED ON GRAPH CONVOLUTION

Di Liu[1], Zhipeng Luo[1], Zhenlong Xiao[1], Jonathan Li[1,2]

1 Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics
Xiamen University, Xiamen, Fujian 361005, China
2 Department of Geography and Environmental Management and Department of
Systems Design Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
*Corresponding author: junli@xmu.edu.cn (J. Li)

IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium | 978-1-7281-6374-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/IGARSS39084.2020.9323086

ABSTRACT

Three-dimensional point clouds can describe the shape and position of objects more accurately when compared with 2D images, thereby providing richer information for object recognition, detection, and reconstruction tasks. Extracting vehicles in point clouds of underground parking lots can help autonomous vehicles achieve automatic parking. Camera-based perception algorithms will fail in complicate environments, so it is necessary to study algorithms for extracting targets using point cloud data. In this paper, we designed an effective method to extract the vehicles in the underground parking lot. First, the point clouds belonging to the vehicle will be segmented using a neural network based on graph convolution, and then different vehicles will be separated based on clustering. Finally, the minimum bounding box for each car is calculated. The proposed approach achieved much better results on the point cloud dataset than other state-of-the-art methods. Our method achieves 99.6% in Overall Accuracy and 98.5% in Mean IOU (Intersection over Union).

Index Terms— point cloud, Vehicle extraction, Graph convolution, clustering

1. INTRODUCTION

The environment in which self-driving cars are located is very complicated. There are two challenges in extracting vehicles in an underground parking lot. First, in the point cloud of the underground parking lot, most of the points belong to the ground and the roof, and only a small number of points belong to the vehicles, which causes the problem of imbalance in categories. Second, because of the mutual occlusion between objects, the point cloud of the vehicle is often incomplete. There are many methods for point cloud semantic segmentation. Some methods, such as PointNet [7], directly input point clouds into deep neural networks to extract features, some methods [8] use local features of point clouds to improve the accuracy of segmentation, and some methods [10, 11] explore how to perform efficient convolution on point clouds. However, these methods ignore the use of local geometry information of the point cloud.

Considering that the point cloud of the underground parking lot mainly contains two types of points, one is the point belonging to the vehicle, and the other is the point belonging to the plane (such as the ground and the wall). We divide the vehicle extraction task into two stages: semantic segmentation and clustering. In the first stage, we use Dynamic Graph CNN [12] (DGCNN) as the basic framework and propose a graph convolution network based on dimension features. Because dimensional features can reflect the geometric information around a point, our network can use this information to classify this point. In the second stage, we use a density-based clustering algorithm to cluster the points of each car separately. In order to further improve the accuracy of the semantic segmentation task, we use the information provided by the bounding box of the vehicle to modify the results of the semantic segmentation.

2. RELATED WORK

There are two common point cloud segmentation methods, the traditional method based on manual design features and the deep learning-based method. Traditional methods use hand-designed features to capture the geometric information of the point cloud [1, 2, 3]. Such methods can only solve specific problems, have poor flexibility, and run slowly. Recently, the great success of deep learning in the field of image processing has motivated its applications to point cloud processing, which outperform traditional approaches in various tasks.

Point cloud processing methods based on deep learning can be divided into two types. The first one is to voxelize the point cloud, then use the number of points in each voxel as its characteristic, and then input the voxelized point cloud into a 3D convolutional neural network to extract features [4, 5, 6]. The volumetric representation will inevitably cause the loss of point cloud information, and most voxels do not contain any points. This method needs heavy computation. Another method is based on point sets, e.g. the PointNet [7]. This method directly takes the point cloud as the input of the deep neural network. It uses a symmetric function called max-pooling to solve the disorder of the point cloud, and uses a module called T-Net to increase the network's geometric rotation adaptability to the point cloud. PointNet only uses

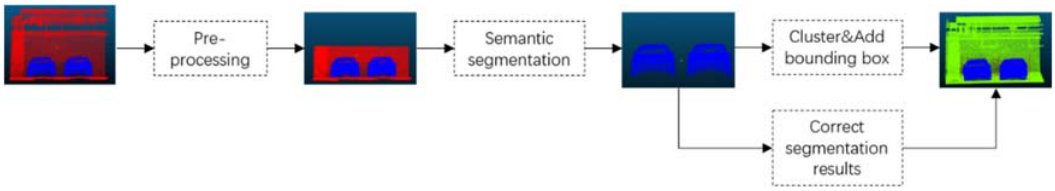

Fig.1 Workflow of our method. We perform semantic segmentation on the processed data to find the point cloud belonging to the vehicle. Then use the clustering method to gather the point clouds belonging to different vehicles separately and calculate the minimum bounding box of each vehicle. Using the information contained in the bounding box, we can revise the results of semantic segmentation to improve its accuracy.

the global features of the point cloud and ignores the local features. PointNet++ [8] increases the ability of the network to utilize local features. It can extract local features of the point cloud at different scales, and get deep features through the multilayer network structure. Inspired by the outstanding 2D shape descriptor SIFT, Jiang et al., proposed a module called PointSIFT [9] that encodes information of different orientations and is adaptive to scale of shape. PointSIFT can be embedded in any PointNet-based network to improve their performance.

Because point cloud data is unordered, it is important to study a convolution method for point clouds. Some researchers have investigated how to perform effective convolutions on point clouds. PointCNN [10] uses X-Conv layer to learn certain canonical ordering of points. PointConv defined the convolution on a non-uniformly sampled 3D point cloud [11], which can build convolutional neural networks on 3D point clouds, and be used to compute translation-invariant and permutation-invariant convolution on any point set in the 3D space. The Dynamic Graph CNN [12] (DGCNN) combines the local and global features of the point cloud to achieve good results based on a new grouping method, KNN, instead of ballquery used in PointNet++ [8]. Superpoint graph (SPG) [13] was proposed to effectively describe the contextual relationships between object parts.

3. METHOD

The workflow of our proposed method is shown in Fig 1. The workflow consists of four parts: data preprocessing, semantic segmentation, clustering, and modifying semantic segmentation results.

3.1. Point Cloud Segmentation

The point cloud of an underground parking lot may contain 20 million points, processing all points will take a lot of time, and most of the points are not what we want. Only about 20% of the points in an underground parking lot's point cloud belong to the vehicle, which causes category imbalance in the data.

In order to solve the above two problems, we preprocess the point cloud before inputting it into the semantic segmentation network. Considering that there is a large distance between the top of the vehicle and the top of the underground parking lot, we can easily cut off the roof. Specifically, we filter out all the points with a height of more than two meters. This can greatly reduce the amount of calculations and alleviate the imbalance of categories.

After preprocessing, only half of the original point cloud is left. We use the DGCNN [12] network to implement semantic segmentation of the point cloud and make some modifications to make it more suitable for underground parking lot scenarios. Because the dataset used in the experiment is small, we simplified the network by reducing the number of layers and channels to prevent over-fitting during the training process. At the same time, it can reduce the amount of calculation and increase the inference speed. The specific network structure is shown in Fig. 2. Through the analysis of underground parking lot data, we found that most of the point clouds belong to the wall or the ground, which is a planar structure, and some point clouds belong to the vehicles, which is a three-dimensional curved structure. Based on the above analysis, we have added dimensional features to the network to improve its segmentation ability. We select the nearest 20 points for each point to calculate its covariance matrix COV. Because COV is a positive definite matrix, it must be able to perform eigenvalue decomposition as follows:

$$COV = [e_1 \quad e_2 \quad e_3]\begin{bmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{bmatrix}\begin{bmatrix}e_1\\ e_2\\ e_3\end{bmatrix} \quad (1)$$

Where: e_i is the eigenvector of the matrix, and λ_i is the eigenvalue of the matrix. The size relationship between a, b, and c can reflect the shape information around this point. When $\lambda_1 >> \lambda_2 \approx \lambda_3$, the object is a linear shape, when $\lambda_1 \approx \lambda_2 >> \lambda_3$, the object is a flat structure, and when $\lambda_1 \approx \lambda_2 \approx \lambda_3$, the object is a 3D surface. In fact, we don't need to compare the magnitude relationship between these eigenvalues, but combine them with the features extracted by the network. The network will automatically learn these features and use them in the segmentation task.

3.2. Cluster and add bounding boxes

The semantic segmentation network can only extract all points that belong to vehicles, but these points are still scattered. Because there must be a distance of tens of

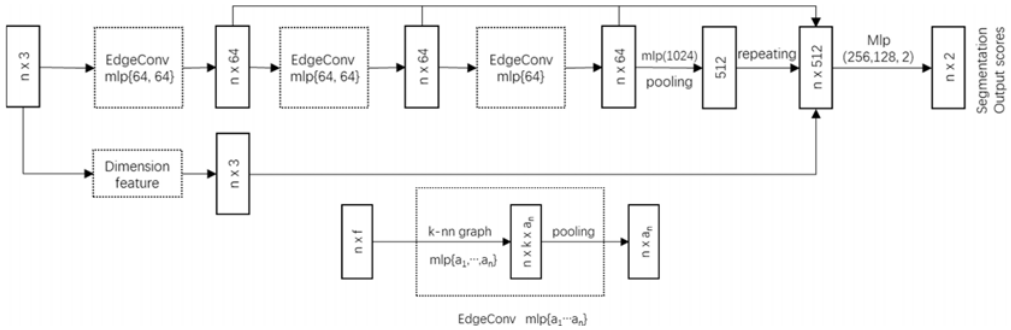

Fig.2 Structure of the network. The EdgeConv block takes as input a tensor of shape n × f, computes edge features for each point by applying a multi-layer perceptron, and generates a tensor of shape n × a_n. The dimension feature block calculates the dimensional characteristics of each point and adds it to the classification network to improve the accuracy of classification.

centimeters between different vehicles, we can use clustering methods to gather the points belonging to different vehicles separately.

We use a clustering method called Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to gather points that belong to different cars. The DBSCAN algorithm does not need to set the number of clusters in advance, and the number of eventually generated clusters is also uncertain. It only needs to set two parameters in advance, one is epsilon and the other is minimum points. Epsilon is the maximum distance between two samples for them to be considered as in the same neighborhood. Minimum points is the number of points in a neighborhood for a point to be considered as a core point. Specifically, we set epsilon to 0.2 meter and minimum points to 10 in the experiment, and achieved good clustering results.

After clustering, we can get multiple clusters, but not all clusters are the cars we want, and some clusters are noise. Some points that belong to other categories such as people and walls are also classified as vehicles, and these noises will form their own clusters after clustering. We calculate a minimum bounding box for each cluster, because the number of noise points is very small, their minimum bounding box size will be small. In order to remove noise, we designed a filter, which only allows the bounding box whose length, width, and height are greater than 1 meter. Because the length, width, and height of the vehicle must be greater than 1, the vehicle can pass this filter and the noise will be removed.

3.3. Revise semantic segmentation results

Although our semantic segmentation network has achieved very good results, there are still some points that are misclassified. For example, a few points of a car are predicted as the ground, but because most points of the car are predicted correctly, we can still accurately calculate its minimum bounding box, and the bounding box can completely cover the car. All points surrounded by this bounding box should belong to this car. Using the information provided by the bounding box, we can easily correct points that belong to the car but are predicted as ground. Fig. 3 shows an example of using the bounding box to modify the semantic segmentation result.

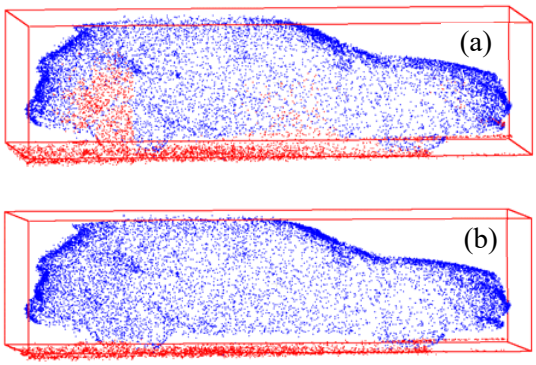

Fig.3 An example of using bounding box to modify the results of semantic segmentation. As shown above, (a) is the original semantic segmentation result. Although the points near the tire of the vehicle are misclassified, the minimum bounding box of the vehicle can still be accurately calculated. (b) is the result of modified semantic segmentation. Using the information provided by bounding box, the misclassified points are reclassified as vehicle (as show).

4. RESULTS AND DISSCUSSION

We prove the effectiveness of the method proposed in this paper on our own dataset. This dataset collected by a backpack lasers scanning (PLS) system with two Velodyne VLP-16 laser scanners. It contains 120 cars in two underground parking lots. We use 50% of the dataset as the training set and the rest as the test set. To augment the dataset, we also extracted some scenes with vehicles from the outdoor road dataset and added them to the training set. We have semantically labeled each point in the dataset, and all points are divided into two categories: vehicle or other.

In order to achieve better performance, we pre-trained our model on Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [14]. S3DIS contains 3D point cloud data of 272 rooms in 6 areas. Each point belongs to one of 13 categories—e.g. board, ceiling, chair and clutter. We re-labeled this dataset, classifying objects with planar structures such as walls, floors, and blackboards into the flat category, and the rest of the objects into the clutter category. On our own dataset, we consider the points that belong to the vehicle as the clutter category and the rest as the flat category.

We first trained 100 epochs with a batch size of 8 on the S3DIS dataset, then trained 60 epochs with a batch size of 8 on our own dataset, and finally achieved excellent results. Table 1 shows the experimental results of semantic segmentation task. We choose PointNet [7] and DGCNN [12] as the baseline. As shown in the table, our method achieves excellent results among the selected methods. Compared with DGCNN, our improved network has a 2.6% improvement in overall accuracy and an 8.5% improvement in mean IOU. Fig. 4 shows the results of vehicle extraction. As shown in the figure, our method can completely extract the vehicles in the underground parking lot, and even the incomplete vehicles can be accurately extracted.

Table1. Results of semantic segmentation. Metric is mean IOU and classification accuracy calculated on points.

	Mean class accuracy (%)	Overall Accuracy (%)	Mean IOU (%)
PointNet	90	94.9	83.8
DGCNN	90	97	90
ours	99.6	99.6	98.5

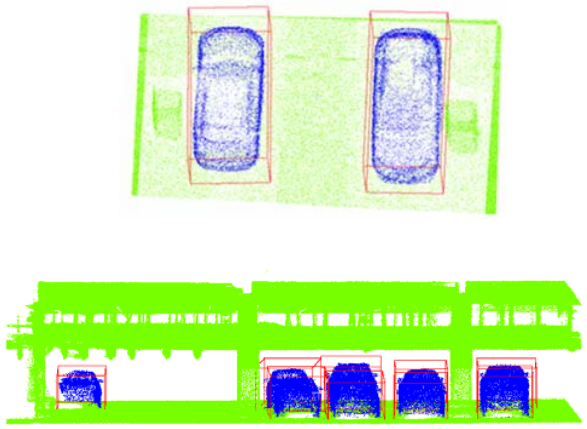

Fig.4 Results of vehicle extraction in parking.

5. CONCLUSION

In this paper, we propose an effective method to extract vehicles in an underground parking lot. We added dimensional features to the graph convolutional network to improve its segmentation performance, and we also used the information provided by bounding box to modify the segmentation results, which further improved the accuracy of the semantic segmentation task. Experimental results prove that our method can effectively extract vehicles in underground parking lots. In future work, we will improve this algorithm so that it can be applied to more scenarios.

6. ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grants 1471379, 41871380, 61371144 and U1605254 and the Natural Sciences and Engineering Research Council of Canada under Grant 50503-10284.

7. REFERENCES

[1] M. Lu, Y. Guo, J. Zhang, Y. Ma, and Y. Lei. “Recognizing objects in 3d point clouds with multi-scale local features”. Sensors, vol.14, no.12, 24156–24173, 2014.

[2] R. B. Rusu, N. Blodow, and M. Beetz. “Fast point feature histograms (fpfh) for 3D registration”. ICRA, pp. 3212-3217, 2009

[3] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. “Aligning point cloud views using persistent feature histograms”. IROS, pp. 3384-3391, 2008.

[4] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. “3D shapenets: A deep representation for volumetric shapes”. CVPR, pp. 1912–1920, 2015

[5] D. Maturana and S. Scherer. “VoxNet: A 3D convolutional neural network for real-time object recognition”. IROS, pp. 922-928, 2015

[6] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. “Volumetric and multi-view cnns for object classification on 3d data”. CVPR, pp. 5648-5656, 2016

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. “Pointnet: Deep learning on point sets for 3D classification and segmentation”. CVPR, pp. 652-660, 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”. NIPS, pp. 5099-5108, 2017.

[9] M. Jiang, Y. Wu, and C. Lu, “Pointsift: A siftlike network module for 3D point cloud semantic segmentation”, arXiv preprint arXiv:1807.00652, 2018.

[10] Y. Li, R. Bu, M. Sun, W. Wu, X. Di and B. Chen. “PointCNN: Convolution On X-Transformed Points”, NeurIPS, pp. 820-830, 2018

[11] W. Wu, Z. Qi and F. Li, “PointConv: Deep Convolutional Networks on 3D Point Clouds”, CVPR, pp. 9621-9630, 2019

[12] Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M., Solomon, J. “Dynamic graph CNN for learning on point clouds”. ACM Trans. Graph, 38(5), 1-12, 2019.

[13] L. Landrieu, M. Simonovsky. “Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs”, CVPR, pp. 4558-4567, 2018

[14] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. “3D semantic parsing of large-scale indoor spaces”. CVPR, pp. 1534-1543, 2016.