

System Design of Road and Vehicle Detection Based on LiDAR Point

Chun-Ju Huang, Bo-Tai Wu and Yu-Cheng Fan⁺

Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan

Abstract. Autonomous vehicles have been hot topics in recent years. It is very important to sense the environment for autonomous vehicles. There are many sensors can be used to sense the environment including camera, sonar, radar, LiDAR and so on. Especially LiDAR [1-3] is a device with high accuracy, high precision and fast characteristics that can capture the distance and contour of object precisely and quickly. It can get 2.2 million points per second and the scanning range is wide. LiDAR can scan the environment from 0.3m to 100m, the scanning horizontal angle is 0° to 360°, and the horizontal resolution is 0.08°. The vertical angle is -15° to 15°, and the vertical resolution is 0.4°. Therefore, LiDAR is a good choice for autonomous vehicles application. Besides, the technology of neural network is maturing day by day. It is good in object recognition and semantic segmentation. Therefore, we propose a method to use LiDAR, color image, and neural network to detect the environment. The scheme pre-processes point cloud data and color image at first. Then we use neural network to extract drivable roads and cars. Finally, we combine point clouds and neural network outputs to mark the road and car on point clouds by mapping algorithm. Besides, we also propose the hardware design with multiple neural network functions in our system.

Keywords: Autonomous vehicles, Environment detection, LiDAR, neural network

1. Introduction

LiDAR is optical radar that combines many advantages. It has changed the measurement methods in many areas, because it can reconstruct 3D space to get environmental information accurately and quickly. This advantage is suitable for self-driving applications. After using LiDAR to obtain three-dimensional information, how to apply this information is a difficult problem. Therefore, the purpose of this thesis is to use LiDAR's point cloud information combined with deep learning for road planning, and can reach 10 frame/s (LiDAR acquisition speed) to match the speed of the car.

Self-driving is a hot topic in recent years, but in fact, as early as the 1920s, there was a concept of self-driving. With the advancement of technology, such as computer vision, sensors and other fields, the first self-driving car appeared in the 1980s. Carnegie Mellon University promoted two programs of ALV and Navlab in 1984. In 1986, the Navlab1 robot came out, which has the ability of graphics processing, sensor information analysis and path planning capabilities. The sensors include gyroscopes, airborne radars, GPS, etc., which can reach 12km/hr at that time. In 1989, a self-driving car called ALVINN was on the Carnegie Mellon campus. In addition to using computer vision, sensors and other technologies, this car further combines artificial intelligence and completes the automatic driving task. By the time of the 1990s, the speed can reach 70km/hr. This car can be said to be the ancestor of self-driving and the first time uses Artificial intelligence into the self-driving car. Therefore, more and more companies are investing in self-driving research and development. In addition, self-driving is receiving more and more attention from governments. More and more countries are beginning to define themselves. In 2011, Nevada first passed the law that allows self-driving cars to travel on general roads, and some states have passed relevant laws. In addition to the United States, countries such as the United Kingdom and France have laws in place, therefore self-driving is growing faster and faster, and more and more people are investing.

⁺ Corresponding author. Tel.: +81-52-735-7785; fax: +81-52-735-5442.
E-mail address: skystar@ntut.edu.tw

Self-driving almost is a tendency, and most applications are using LiDAR as detection tool. The reason is that LiDAR can reconstruct the surrounding environment to avoid obstacle, therefore it is a very suitable tool for self-driving. However, we usually have various kinds of situation during driving. It is very difficult for us to find an algorithm that can solve all these problems. In this work, we use the convolutional neural network as our proposal. Recently, artificial intelligence is mature technique, and deep learning can be used to solve some abstract problems that general algorithm cannot solve. Therefore, this paper hopes to use deep learning with LiDAR's environmental information to achieve the purpose of finding travelable road for self-driving application.

2. Proposed Method

2.1. Dataset

In order to train point cloud data, we have to prepare a well-marked ground truth dataset. The database [4] provides a wealth of point cloud data and ground truth of road marking information. In addition, KITTI provides a well-calibration formula for matching the point cloud data and color image. The parameter x, y, z is the LiDAR three-dimensional coordinates and u, v , is color image two-dimensional coordinates. Then we match the LiDAR point cloud data and well-marked color image through formula (1) and get marking point cloud data. Next, we use the convolution neural network to train and classify the data.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (1)$$

2.2. System Architecture

The system we propose has two major parts. The first part is the application of LiDAR data. After obtaining the point cloud, we will pre-process the point cloud data. Including data deletion and data format conversion, we delete point cloud data from the point outside the camera's view angle and converted into a form of size $224 \times 224 \times 3$ to match the input of the neural network. This format can also reduce the amount of data processed time, so that we can improve the speed. Next, put the pre-process data into the path of the convolutional neural network, and use the output data of the convolutional neural network to judge the walkable path. The second part is to use the color image to find the car in the point cloud image. At the beginning, it must adjust the color image to match the input size of the neural network, and then use the color image as input of the neural network of the segmentation function. In the end, use the cut out color image of car to match with the coordinate point of LiDAR, and finally perform the matching of car and walkable path on the point cloud data.

2.3. Network

In the part of the neural network, in the past, the splitting began with the folding layer. The proposal starts from the folding layer, and adds the pooling layer to increase the reception field and the speed. However, the pooling layer will cause too much message loss and resulting in distortion in the restoration. The dilated convolution is one of the convolutional deformations, as shown in Fig. 1. While the expansion rate is two, the reception field of original 3×3 filter becomes the same range as the 5×5 filter, but only 9 parameters are used, therefore proposal replaces the traditional neural network convolution layer and the pooling layer part with dilated convolution to enhance its performance. [5-6].

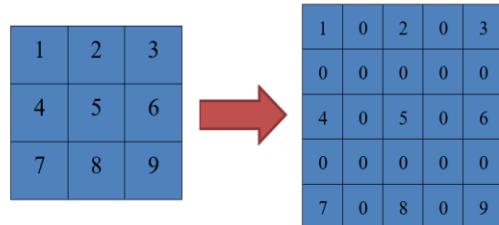


Fig. 1: Dilated convolution.

In addition to the dilated convolution, we refer to the pyramid scene parsing network to improve our architecture, also known as PSP Net. This architecture was released in 2017, which improved the architecture of the previous FCN-8s. It shows that FCN-8s restores the pooling layer 3, the pooling layer 4 and the pooling layer 5 to the same size as the original image by means of up sampling, and then we can obtain previous information by this method. FCN-8s has better results than FCN in semantic segmentation. On the other hand, PSP Net is different from the network with FCN-8s. It replaces the addition of pooling layers with a stack and passes through a layer of convolution to the output. Experiments have shown that this network is better than the previous FCN.

In the experimental data, both the dilated convolution network and the PSP Net have very good result, therefore we combine the both method in our neural network architecture. Initially, we adjust the input image to match the size of the neural network, and then use the image as input to the proposal neural network. There are many network architectures available in the CNN section, including Alex Net, VGG, Google Net, Res Net. In our proposal, we adopt a structure similar to VGG, and make some deletions in the architecture. In addition, proposal neural network removes the original pooling layer and leaves only one pooling layer to increase the speed of network. This strategy can also increase the reception field, but avoid losing too much information.

Next, we add ReLU after each convolution layer, it can increase the variability of the entire network, and avoid the negative network from affecting the accuracy of the entire circuit. After passing through the neural network output, it enters the PSP Net architecture. Here, proposal changes the pooling layer of PSP Net to use the method of dilated convolution. The advantage of this method is that it can expand its reception field. When rate is equal to 1, the reception field is 3×3 . When rate is equal to 2, the reception field is 7×7 . When input is next layer, the reception field is 15×15 . Therefore, we use the dilated convolution of three consecutive rate increases the reception field, and then superimpose the result of the dilated convolution with the feature map of the last layer of CNN. By stacking each of the different scales, we can obtain better global information of input. Finally, the three layers of convolutional layer and ReLU are gradually decoded, and then output of network obtain the marked feature map.

3. Experiment Results

In this section, we will compare the proposal neural network results in our paper with those of other papers. Our main comparison objects are the architectures of FCN-8s and FCN-16s [7]. Table I shows the comparison list of speed. Here we compare the same hardware conditions that the GPU is the NVIDIA GeForce GTX 1080 Ti, and the dataset is the Cityscapes Valid Dataset. The input image size and output image size are all in the same condition, 224×224 , and under this condition, the FCN-8s architecture can handle 11 images per second, and the FCN-16s has 12 images. In our architecture, 16 images per second can be processed and converted. The former handles about 0.09s and 0.083s per image, while the latter has a structure of about 0.0625s. If we add the time of the mapping, only our method can achieve 11 processing speeds per second, which can meet operating speed of LiDAR.

Table I: Car Split Comparison

	Input & Output Image size	Data Set	GPU	Speed	Speed with Mapping
FCN-8s	224×224	Cityscapes Valid	NVIDIA GeForce GTX 1080 Ti	11 frames/s	8 Frames/s
FCN-16s	224×224	Cityscapes Valid	NVIDIA GeForce GTX 1080 Ti	12 frames/s	9 frames/s
Proposed	224×224	Cityscapes Valid	NVIDIA GeForce GTX 1080 Ti	16 frames/s	11 frames/s

4. Conclusion

This paper proposes a method of road and vehicle detection based on LiDAR point. The result is that we have 99% pixel accuracy in cutting cars and 92% in road detection. In terms of speed, we can achieve 16 frames per second on our device. And with the time of Mapping, we can still achieve 10 frames per second, which is in line with LiDAR data acquisition speed. The system mainly uses 3D point cloud mapping color image to find the roads and cars in the point cloud, so that you can get the depth information from the objects, which is good for applications of self-driving.

5. Acknowledgements

This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 106-2221-E-027-135-. The authors gratefully acknowledge the Chip Implementation Center (CIC), for supplying the technology models used in IC design.

6. References

- [1] LiDAR. [Online]. Available: <https://baike.baidu.com/item/Lidar>.
- [2] Velodyne LiDAR. [Online]. Available: <http://velodynelidar.com/news.php>.
- [3] Velodyne. [Online]. Available: <https://read01.com/zh-tw/E5Gg5J.html#.WgVxu2iCy70>.
- [4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision Meets Robotics: The KITTI Dataset," International Journal of Robotics Research, vol. 32, no. 11, pp. 1231-1237, 2013.
- [5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, No. 4, pp. 834-848, Apr. 2018.
- [6] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in 2017 IEEE Conference on Computer Vision and Pattern Recognition, HI, USA, July, pp. 6230-6239, 2017.
- [7] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, No. 4, pp. 640-651, Apr. 2017.
- [8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in 2017 IEEE Conference on Computer Vision and Pattern Recognition, HI, USA, July, pp. 6230-6239, 2017
- [9] F. Yu, V. Koltun, "Multi-scale Context Aggregation by Dilated Convolutions," in International Conference on Learning Representations, San Juan, Puerto Rico, May 2-4, 2016