

Point Cloud Segmentation and Detection for Vehicle Based on LIDAR Sensor

Xuan Xiu^{1, a}, Xianglei Zhu^{2, b}, Lianqing You^{2, c}

¹Wuhan University of Technology

²China Automotive Technology and Research Center Co., Ltd

^axiuxuan126@163.com, ^bzhuxianglei@catarc.ac.cn, ^cyoulianqing@catarc.ac.cn

Keywords: intelligent driving, Velodyne LIDAR, 3D point cloud segmentation, vehicle point cloud detection, random sample consensus method, Euclidean segmentation, Hough clustering

Abstract: Currently automatic driving technology is the most rapid research direction in the field of artificial intelligence. Besides, the environment sensor LIDAR is the most significant hardware in system automatic sensor system. This paper concentrates on the cloud data acquired by the LIDAR as the input data and utilizes the point cloud library algorithm to write C++ program. We not only achieve the single vehicle object segmentation and detection in open scenario but realize the same target of the multiple vehicles in parking scene. We obtain a comparatively nice result of vehicle's segmentation and detection. Simultaneously, our C++ program package provides the foundation for further research on point cloud recognition and semantic level segmentation.

1. Introduction

In June 2018, the China Intelligent Driving Challenge, with the theme of “Intelligent Change of the World, Innovation Drives the Future”, was successfully held at the China Automotive Research Center in Tianjin. More than 60 domestic car dealers, unmanned car teams from universities and Internet companies, and more than ten models participated in this traditional smart car championship. On the one hand, this competition is a platform for collective display of domestic autonomous driving technology. On the other hand, we noticed that the fleet of high-threaded lidar sensor equipment in the unmanned smart car competition can almost achieve a higher ranking. It can be seen that the laser radar has the highest detection accuracy. The role played by people and vehicles in the process of intelligent driving and the important components.

Today's unmanned vehicle-mounted sensors generally include a vision sensor camera, a millimeter-wave radar, a laser radar lidar, which are the hardware foundation for intelligent driving. The main process of the smart car to achieve driverless is divided into three modules: through the sensor device pair Perceptual module for surrounding environment information acquisition; chip unit embedded in deep learning enhancement learning algorithm to analyze decision-making module; execution module for executing chip decision through hardware control device. The relationship between the modules is shown in Figure 1. Among them, the advanced level of sensor hardware, which is mainly responsible for sensing and calculating the surrounding environment and object information, determines the level of intelligence of unmanned automatic driving, especially the multi-threaded laser radar equipped, which can not only be no less than 10HZ per second. The frequency detects 360 degrees of all stationary and moving obstacles in the horizontal plane (no blind spots and dead angles) and is able to calculate the exact distance of obstacles within a radius of 200 meters.

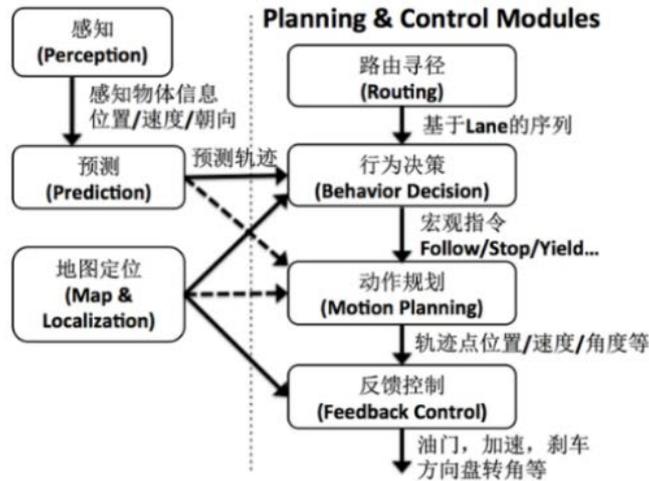


Figure 1. Intelligent driving perception, decision making, control module relationship

2. Advantages of sensing device lidar

The new industry of unmanned autonomous driving technology is undoubtedly in a stage of rapid development and is one of the hottest areas of artificial intelligence. Its technological development and progress are subtly changing the way people think about cars and the way they travel. Although, we need a step-by-step process to achieve the goal of commercialization and mass production of unmanned vehicles that we finally want to achieve, there must be a long process in terms of technological innovation, regulatory policies, consumption promotion and cost savings. A distance needs to go. However, from the perspective of technological innovation, especially the mature laser radar hardware and the main point cloud processing algorithm in this paper have been perfected. The unmanned sensing hardware equipment mainly includes laser radar, vision camera, ultrasonic radar, GPS and other devices. The detection range and detection distance of each device on the driverless car are as shown in Figure 2:

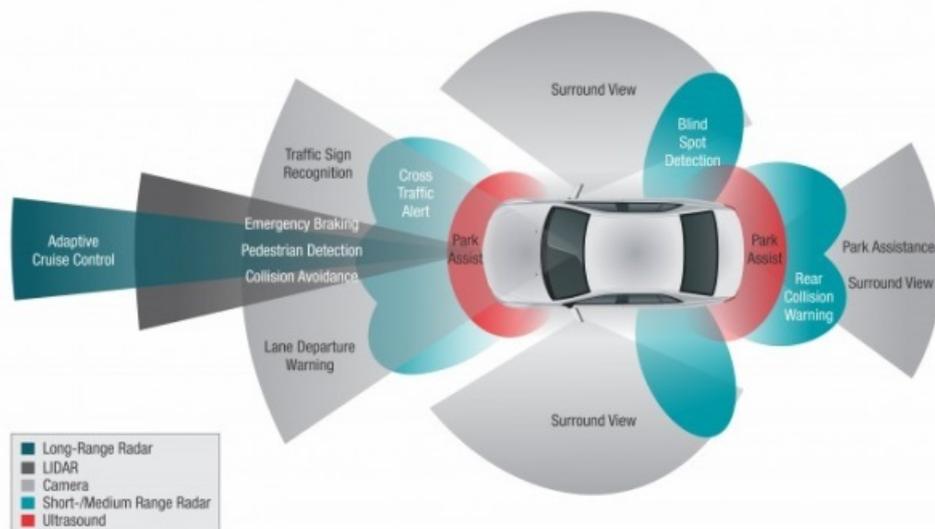


Figure 2 Detection range and distance diagram of intelligent driving sensing device

Image vision sensors and lidar sensors among sensing devices are among the two main types of devices that sense the external environment. Compared with vision sensors (such as binocular cameras), lidar sensors have the advantage of detecting distances and accurately figuring out three-dimensional information about objects. The depth learning method used for image recognition and detection corresponding to the vision sensor is difficult to achieve in real time, but the lidar point

cloud can scan the 3D contour of the obstacle as a whole, so the algorithm is much simpler and clearer than the depth learning. Do it in real time. In addition, the point cloud data of the lidar has the dimension of the laser scanning intensity in addition to the dimension of the three-dimensional information, that is, the density information of the obstacle can be obtained in one step from the point cloud data, which means that the tree is easily distinguished by the point cloud. Semantic information such as leaves, trunks, streetlights, vehicles, etc. The logic of this semantic level is very straightforward and only needs to be based on the intensity spectrum. It is very time consuming to accurately identify through visual learning through visual learning. Speaking of real-time, for the camera.

If you want to process 60FPS frame rate video, then it is equivalent to only 16 milliseconds for each frame. For visual depth learning mode, if there are some frames of image delay processing (equivalent to ~100ms order), then it is possible that the unmanned vehicle moves a few meters away but the delay of the sensing unit causes the decision control unit to have no reaction, which is undoubtedly a safety hazard for the unmanned vehicle. Therefore, visual depth learning is more suitable for analyzing static images, and laser radar is better at dynamic video analysis. We then use the plane fitting method to extract the horizontal plane (equivalent to the horizontal ground) in the section introduced by the algorithm. It is only necessary to call the plane model in the random sample consistency method (RANSAC), which is for visual analysis.

3. Comparison of the development status of laser radar at home and abroad

With the emergence of laser radar in intelligent driving in recent years, domestic research on laser radar technology has been paid more and more attention. Domestic laser radar manufacturers have also rapidly emerged as radar manufacturers such as Radium Intelligent and Siwei Technology. Since the key technology of laser radar cannot be imported from abroad, domestic laser radar manufacturers mainly rely on independent research and development and have obtained relatively mature products. However, compared with the foreign radars such as Velodyne and Quanergy, the gap is quite obvious. In the past two years, Radium Intelligence has launched a 16-beam radar product that can be used for autonomous navigation and obstacle avoidance and car collision avoidance, but the measurement accuracy is not high (for a range of 10 meters, the measurement accuracy is about 15cm). The product can solve the problem of autonomous navigation positioning, but can not draw 3D images.

The technical functions of the fully functional Advanced Driver Assistance System (ADAS), as well as the intelligent completion of the unmanned vehicle, such as deceleration, lane change, overtaking, scene switching and other driving behaviors are inseparable from the use of high-precision multi-threaded laser radar. . Most of today's driverless R&D companies use laser radars from companies such as Velodyne and Quanergy. The main parameters of Velodyne's top-of-the-line product HDL-64E, such as detection range and detection accuracy, are shown in Table 1. Google has been a staunch user of the laser radar industry giant Velodyne. The large Internet company headed by Google has gained extremely advanced experience and technology based on the accumulation of massive driving data. Up to now, Google has completed more than 4 million kilometers of self-driving car mileage, and has obtained a large number of laser radar point cloud data. After Tesla's serious driverless traffic accident in 2015, Marx, the company's chief executive who said he did not need to use lidar to collect point cloud data, gradually changed his mind to the unmanned car in September 2016. The lidar sensor is also loaded in the road test. In the domestic scope, both traditional automobile companies and Internet companies with strong technical funds are increasingly aware of the market development prospects and realization possibilities of the unmanned vehicle industry, and have joined the new technology. Revolutionary cross-border cooperation and technological competition come. In order to greatly reduce the production and purchase cost of laser radar, Baidu announced that it will invest USD 150 million to jointly invest in laser radar system manufacturer Velodyne. After the acquisition of Uber brand, business, and data, Didi Travel Company is also very likely to use Uber's self-developed laser radar products.

Table 1 HDL-64E lidar parameter table

Safety Level 1	Eye Safety	Angular resolution	0.09°
Wavelength	905nm	Scanning frequency	>1.333Mpoints/second
Number of lasers	64	Working environment	-10° C ~ 50° C
Horizontal field of view	360°	Detection distance	120m car or tree
Vertical field of view	26.8°	Pavement range	50m
Detection distance accuracy	<2cm	Weight	13.2KG

4. Algorithm Application of Lidar Point Cloud Data

3.1 Point cloud data extraction and format transcoding

For this experiment, the raw data used by the point cloud data detection algorithm is the point cloud data collected by the Velodyne 16-thread lidar sensor in the second Drip Algorithm Competition in March 2017. One is scene detection of a single vehicle target on an open road surface, and the other is scene detection of multiple vehicles in a parking lot. The input data format is the ROS bag type, and the output data format is a coded file of the pcd type. We first need to interface the point cloud data to the message mechanism of the ROS operating system, define the callback function (Callback function) and the message acceptor (Subscriber) to achieve the extraction of point cloud data.

After storing the point cloud data as a file in pcd format, we need to decode the data in this file format. Because the pcd file is a binary file, its advantage is 5-10 times smaller than the normal string format file storage space, but it is not readable. Therefore, before processing, it is converted into a document file of type string.

3.2 Filtering and extracting point cloud data

After completing the decoding of the point cloud data, the data can be initially filtered and extracted. The main purpose of filtering and extracting point cloud data is to reduce errors and eliminate errors. Because the ideal point cloud data is characterized by no noise and dead pixels, it meets the processing requirements of real-time data, and the collected object point clouds are evenly distributed. However, the point cloud data measured by the sensor is always accompanied by statistical errors, as well as weather conditions, the measurement accuracy of the instrument itself, and the systematic error that the measurement personnel will bring. At the same time, the outliers generated during the measurement process are the wrong information. Wrong point cloud information will result in poor detection and even failure results for subsequent point cloud local and global feature estimation, as well as other algorithms.

The noise reduction algorithm we use here is the Statistical Outlier Removal Algorithm to remove point cloud noise and remove outliers. Such a noise reduction algorithm expresses a very straightforward idea. Its main principle is equivalent to two nested for statement loops. The outer layer first traverses each point cloud data point, and the inner layer traverses the point cloud point. The distance of the nearest N points. Statistically, the large number theorem can guarantee that the distance distribution of these N points satisfies the Gaussian distribution, then we consider the point that deviates from the center point of the Gaussian distribution by n times σ distance as noise. The number N of point clouds around a given point, and the setting of the $n\sigma$ parameter boundary are implemented by the following C++ statement:

```
noise.setMeanK(10)
noise.setStddevMulThresh(0.2)
```

The parameter size setting depends on whether the shape of the output point cloud geometry is smoother than that displayed by the input point cloud, and there is no feature information that loses its point cloud shape. The geometry information of the output point cloud during the noise reduction

process is not sensitive to the adjusted parameter changes, so the parameter convergence speed is faster. Here we take the parameter values of 10 and 0.2 respectively.

3.3 Scene point cloud data downsampling

Denised point cloud data generally cannot be segmented and detected. Otherwise, it will consume longer calculation time. Therefore, the point cloud data needs to be further thinned to meet our unmanned intelligent driving vehicle. Real-time requirements necessary. Downsampling is required to satisfy both the acquisition of fewer points and the basic features of the guaranteed point cloud without loss. Among them, we use the Voxel Grid Filter algorithm. Its main algorithm is to divide the point cloud space into small cube 3D grids, calculate the centroid points of all point clouds in each cube grid, and replace the grid with this centroid. All point clouds inside to achieve the purpose of downsampling. As shown in Figure 5 below, the convenience of this downsampling algorithm is that we only need to set the parameter side length of the cube mesh. We only need to perform one-dimensional scanning on this parameter to get the convergence result very quickly. The parameter value is 0.01downsample.setLeafSize (0.01f, 0.01f, 0.01f);

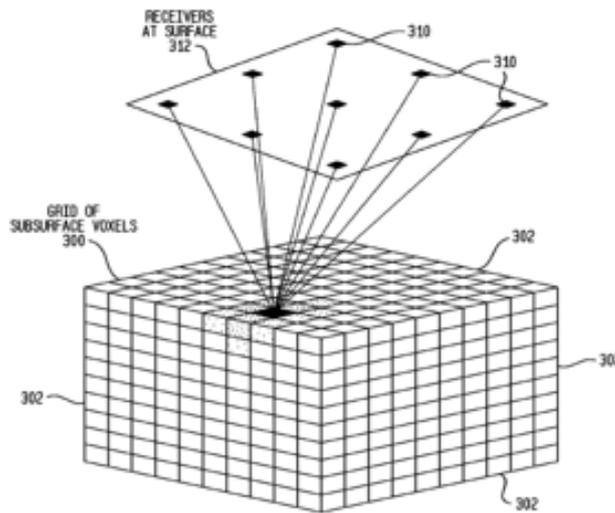


Figure 3 voxel grid algorithm takes the lattice centroid for downsampling

3.4 Segmentation of scene point clouds

3.4.1 Random sample consistency method fits the extraction plane

After the elimination of the point cloud excess noise data and the downsampling of the high point density, we can segment the point cloud data to the point cloud level. The purpose of the segmentation is to give all the point clouds of each segmented cluster. point. The first problem encountered here is that the points of all obstacles (including pedestrians, vehicles, and tree vegetation) scanned by the lidar are not independent from the point cloud map, they are generated during the process of transmitting and receiving signals with the lidar. The circular planes in the horizontal direction are connected. These circular point clouds generally do not belong to the target range we are studying. They are redundant information in the process of segmentation detection of vehicle targets, so we must first remove them. The method usually used to remove the circle point cloud point includes using its color information, or its spatial structure information. Here our point cloud data does not include RGB information, so we need to use a plane model (SACMODEL_PLANE) to fit this. The plane circle point cloud, the fitting algorithm used is random sample consensus (RANSAC),

3.4.2 European Segmentation Algorithm for Scene Segmentation

In the point cloud map after fitting the extracted plane, we will use the point cloud segmentation algorithm to segment all obstacles in the graph at the data point level, including vehicles, pedestrians, and trees in the parking lot. We use the European segmentation algorithm to segment the point cloud data. Each cluster we split is called cluster. The parameters that need to be adjusted in the C++ program include the shortest distance between clusters, the minimum number of point clouds allowed for clusters, and the maximum number of point clouds allowed for clusters.

```
pcl::EuclideanClusterExtraction<pcl::PointXYZ>eu;  
eu.setClusterTolerance (0.5);  
eu.setMinClusterSize (20);  
eu.setMaxClusterSize (25000);
```

When scanning parameters, the three parameters are not independent of each other. Therefore, the convergence speed is slow, and the final values are 0.5m. The best effect is 20, 25000.

3.5 Vehicle target detection

The purpose of point cloud detection is to mark the target object with a rectangular frame. Here we set the target object we want to detect as a stationary vehicle, and use the algorithm of correspondence grouping to detect the vehicle. Due to the angle of the target vehicle scanning, the overall geometric outline of the car body cannot be completely scanned by the cloud point, but we noticed that the vehicle point cloud distribution characteristics in the parking lot are mainly divided into "L" shape and "one" The shape of the word. Therefore, we separately extract the "L"-type point cloud vehicles with dense contours in the parking lot and the "one" type point cloud vehicles with sparse contours as our point cloud input model (point cloud input data), as shown in the figure. Then, using the 2D, 3D and statistical features of the point cloud to geometrically summarize the vehicle contours, use the correspondence clouds pair to find the corresponding points in the parking lot scene, and finally use the Hof 3D clustering algorithm to find the corresponding points.

First we need to calculate the normal vector of each point cloud point. The calculation of the normal vector of a point cloud is a process of taking the limit. Set the n points closest to the cloud point of the point, and then fit a plane based on the n+1 points, and take a minimum limit approximation to the plane. The direction of the normal vector of the result of the approximate approximation is The normal vector of the point cloud point. From this normal vector we can calculate the local descriptor of each point cloud point. This so-called local descriptor represents the local geometric features of the point. The specific parameters that need to be set in the C++ program are as follows

The local descriptors are used to find the corresponding point cloud pairs. The main parameters that satisfy the point cloud pairing are the point cloud in the model and the point cloud in the scene. The geometric distance in the local description subspace is less than 0.25.

Hover 3D clustering algorithm is used to cluster the corresponding points found. The principle of Hough clustering is mainly based on the algorithm of Hough voting, and the corresponding local coordinate system needs to be calculated for each point cloud point. The parameters set include:

```
reference.setInputCloud (model_keypoints);  
reference.setInputNormals (model_normals);  
reference.setSearchSurface (model);  
reference.compute (*model_reference)  
cluster.setHoughBinSize (10);  
cluster.setHoughThreshold (5);  
cluster.setInputCloud (model_keypoints);  
cluster.setInputRf (model_reference);  
cluster.setSceneCloud (scene_keypoints);  
cluster.setSceneRf (scene_reference);
```

Finally, the target vehicle detection result in the open scene is obtained.

The target vehicle detection result in the parking lot scene can also be obtained. The original point cloud image of Figure 10 and the comparison result of Figure 11 are shown:

5. Conclusion

The main content of this paper is based on the point cloud data acquired by the environmental sensing device lidar of the unmanned vehicle, which realizes obstacle segmentation and vehicle detection for single vehicle scenes and multi-vehicle scenes. Previously, the point cloud 2D projection and 2.5D projection algorithm will lose some dimension information of the point cloud. In contrast, the point cloud algorithm analysis in the 3D range without losing any dimension information is relatively better. Good segmentation and detection effects. The research directions of other point cloud data include the recognition algorithm of the scene target, the segmentation algorithm at the semantic level, and the image data fusion algorithm provide reference and reference. And for the future individuals to continue to study the smart car environment perception link algorithm provides the basis of the formed program framework.

References

- [1] Point cloud library official website
- [2] Fu Yanjun, Yan Wei, et al. Real-time vehicle monitoring based on multi-feature fusion. *Computer Simulation* 2007, 10
- [3] Zhan Qingming, Liang Yubin, et al. Comparative study of three laser point cloud segmentation recognition algorithms. *The first national Lidar Earth Observation Advanced Symposium*, 2010, 7
- [4] M. Himmelsbach, A. Müller, T. Lüttel and H.-J. Wünsche: LIDAR-based 3D Object Perception. 2010
- [5] A. Aldoma, Z. Marton., F. Tombari: Three-Dimensional Object Recognition and 6DoF Pose Estimation. 2012
- [6] M. Lehtomäki, A. Jaakkola, J. Hyypä: Object Classification and Recognition From Mobile Laser Scanning Point Clouds in a Road Environment. 2016
- [7] Z. Xu, X. Chen, W. Jie: Vehicle recognition and Classification Method Based on Laser Scanning Point Cloud Data. 2015
- [8] S. Awan, M. Muhamad, K. Kusevic, P. Mrstik, and M. Greenspan: Object class recognition in mobile urban lidar data using global shape descriptors, in *Proc. Int. Conf. 3D Vis.*, Seattle, WA, USA, 2013, pp. 350–357.