Vehicle detection based on point cloud intensity and distance clustering

Zhao Wang¹², Xing Wang³*, Bin Fang¹², Kun Yu¹², Jie Ma¹²

¹ School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
² National key laboratory of multispectral information processing technology, Huazhong University of Science and Technology, Wuhan 430074, China
³ Science and technology on complex system control and intelligent agent cooperation laboratory, Beijing 100074, China

*Corresponding author’s e-mail: wh_ykun@hust.edu.cn

Abstract. In the intelligent transportation system, vehicle detection is one of the essential technologies in obstacle avoidance and navigation, however the existing vehicle detection methods cannot meet the actual needs. This paper presents a vehicle detection method combines the intensity and distance information of point cloud, which improves the segmentation performance of nearby objects. Specifically, the data of point cloud collected by lidar is preprocessed first. Then the processed point cloud is clustered by combining its coordinate and intensity information. Finally, the clustered suspected targets are fed to the random forest classifier. Our method can efficiently detect and classify targets in large-scale disordered 3D point cloud with high accuracy. In the real-scanned Livox Mid-40 Lidar dataset, our proposed method improves the detection accuracy by 31% compared with the traditional Euclidean clustering.

1. Introduction
With the rapid development of science and technology, autonomous driving has attracted more and more attention from people. Vehicle detection in automatic driving system is becoming a hot research direction[1-4]. The sensor Velodyne LiDAR, is often used in autopilot. The radar is expensive and heavy, making it difficult to be widely used. At the same time, the current detection algorithm with good performance is usually the deep learning algorithm, which needs to consume a lot of computing resources, and the computing equipment is not easy to carry. Both of these factors make it difficult to popularize autonomous driving technology. In the traditional point cloud target detection algorithm, the clustering algorithm only uses the coordinate information in the point cloud, and it is difficult to get a good segmentation effect when the target is close to the interferences.

Recently, the Livox series radar launched by DJI not only weighs 1kg but also costs less than 10,000 yuan, which has greatly promoted the civilian use of radar. We adopt DJI Livox Mid-40 radar as the data acquisition module, and designed an embedded vehicle detection system. The detection device is simple and portable, suitable for multi-scene application.

We also improved the clustering algorithm by introducing intensity information. Objects with similar distance can only be classified into one category if their intensity is also similar. This method can effectively improve the detection accuracy. Our detection process is as Figure 1.
Figure 1 The overall pipeline of our proposed vehicle detection method. Firstly, the data of point cloud collected by lidar is preprocessed. Then the point cloud is clustered by combining its coordinate and intensity information. Finally, the clustered suspected targets are fed to the random forest classifier.

2. Detection process

2.1. Target segmentation based on intensity and distance information

2.1.1. Point cloud preprocessing
Raw point cloud data have huge amount of points which will consume huge computing resources. Raw point cloud should be filtered firstly. Passthrough filtering is adopted to filter the sparse point clouds with a distance greater than 60m from the radar. Meanwhile, down-sampling[5] filtering is used to thinning collected point clouds with 0.1m as the unit, Which can simplify the number of point clouds and improve the processing speed.

2.1.2. Ground points removing
In ground scenes, the ground points occupy a large proportion in the point cloud, which seriously affects the subsequent processing. In our work, the ground plane was fitted by the RANSAC (Random Sample Consensus) [6-7], and the plane with the most points was fitted as the ground point for filtering.

2.1.3. Point cloud clustering
After the ground points removing, the point clouds in the space are independent clusters. We use point cloud clustering and gather the close points into the same class. Euclidean clustering[8] and DBSCAN clustering[9] are two common clustering algorithms, these methods cluster the points with close distance into one class, and only the coordinate information of the points is used in the clustering.

In real scenes, there are obstacles such as bushes and trees beside vehicles, which are connected with vehicles, often clustered with vehicles because of their close distance. Vehicles and bushes cannot be separated only according to point cloud coordinate information. We introduce intensity information in clustering[10]. Vehicles and bushes have different reflection intensity due to the big difference in material. We can use this difference to improve the clustering algorithm, and DJI Livox Mid-40 radar saves the intensity information of reflection points in addition to the coordinate information. In our method, Points with close distance (less than 0.2m) can only be grouped into a class if they meet the requirements of similar intensity.

2.2. Target Classification based on global and local features

2.2.1. False-alarm
After clustering there are multiple point clouds. In order to reduce false alarm, we conduct preliminary screening according to the size of each point cloud and select point clouds that meet the size of vehicles
for further detection. According to the coordinate information of each point cloud, we can calculate the bounding box of the point cloud, then the length, width and height of the bounding box can be obtained, if it meets the threshold condition, it can be retained and further processed.

2.2.2. Characteristics Calculating

For the point cloud after the false-alarm is removed, we extract a variety of custom typical features and use a classifier for classification. There are not only global features that reflect the overall characteristics of the target, but also local features that are robust to occlusion and deformation. Among them, the global features contain two kinds of features: (1) the geometric attribute feature calculated by the minimum bounding box. (2) the geometric attribute feature obtained by the calculation of covariance feature root. The local typical features is 3D Hog feature based on projection[11].

The bounding box features are divided by length L, width W, and height H, as well as aspect ratio L/W and aspect ratio L/H.

\[
F_{bbox} = \{L, W, H, L/W, L/H\} \tag{1}
\]

In addition to the global geometry size and other features, there are target point cloud space geometric attributes including: linearity, planarity, scattering, variance, curvature, anisotropy, entropy, orientation, these characteristics describe the linear correlation, the closeness to the plane, dispersity, spatial curvature features, can description target point cloud space geometric properties. The specific calculation process is as follows:

First, the mean value of point cloud distribution on the X-axis, Y-axis and Z-axis was calculated as \(\bar{x}, \bar{y}, \bar{z}\) and then the covariance of the X-axis and Y-axis distributions was calculated according to the equation (2):

\[
\text{Cov}(X, Y) = E[(X - E(X))(Y - E(Y))] = \frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y}) \tag{2}
\]

Similarly, the covariance of other coordinates can be obtained, as equation (3):

\[
C = \begin{bmatrix}
\text{Cov}(X, X) & \text{Cov}(X, Y) & \text{Cov}(X, Z) \\
\text{Cov}(Y, X) & \text{Cov}(Y, Y) & \text{Cov}(Y, Z) \\
\text{Cov}(Z, X) & \text{Cov}(Z, Y) & \text{Cov}(Z, Z)
\end{bmatrix} \tag{3}
\]

The eigenvalue solution of covariance matrix by SVD decomposition is as equation (4):

\[CV = EV \tag{4}\]

\(E\) is the eigenvalue vector, \(E = \{\lambda_1, \lambda_2, \lambda_3 (\lambda_1 > \lambda_2 > \lambda_3)\}\), \(V\) consists of three orthogonal eigenvectors, \(V = \{v_1, v_2, v_3\}\), then according to the eigenvectors \(\lambda_1, \lambda_2, \lambda_3\), calculate: Linearity \(K_{\text{linearity}}\), planarity \(K_{\text{planarity}}\), scattering \(K_{\text{scattering}}\), variance \(K_{\text{variance}}\), curvature \(K_{\text{curvature}}\), anisotropy \(K_{\text{anisotropy}}\), entropy \(K_{\text{entropy}}\), orientation \(K_{\text{orientation}}\). Finally, the above features are integrated into the global features of covariance feature roots (equation (5)).

\[
F_{eigen} = \{K_{\text{linearity}}, K_{\text{planarity}}, K_{\text{scattering}}, K_{\text{variance}}, K_{\text{curvature}}, K_{\text{anisotropy}}, K_{\text{entropy}}, K_{\text{orientation}}\} \tag{5}
\]

HOG feature is a feature vector composed of histogram of gradient direction in different local areas of image, and it is robust to geometric and optical deformation. We adopt the projection method to project the spatial point cloud objects on the XY, XZ and YZ planes and then raster the 2D projection. Extract the HOG (Histograms of Oriented Gradients) characteristics of the projected image based on local orientation histogram, The 3D HOG feature is obtained by serial connection of the orientation gradient histogram (equation (6)).

\[
F_{\text{hog3d}} = \{H_{xy}, H_{xz}, H_{yz}\} \tag{6}
\]
Finally, the above bounding box feature, covariance feature root feature and HOG feature are combined to obtain the feature that needs to be sent to the classifier in the end.

2.2.3. Classifier
For classifiers, the field of machine learning provides many classifiers. In this method, we adopt the random forest classifier [12]. Random forest is an algorithm that integrates multiple trees through the idea of integrated learning. Its basic unit is decision tree. Random forest is simple and easy to implement with low computing costs. It can process high-dimensional data without making feature selection. Meanwhile, it performs well for many data sets and is not prone to overfitting.

3. Experimental result
The data acquisition unit of the experimental platform is Livox Mid-40, the core processor of the computer used in the experiment is I7-9700F, the memory is 16GB, the main frequency is 3.00GHz, the graphics card is NVIDIA GeForce GTX 1660, and the implementation tool is VS2013.

For the data collected by Livox Mid-40, we accumulated the points collected within 1s and generated a point cloud as shown in Figure 2(a). Through passthrough filtering and down-sampling, point cloud Figure 2(b) was obtained, and ground points were filtered out by RANSAC algorithm to obtain point cloud Figure 2(c). Then, Euclidean clustering was used to aggregate the nearby points to obtain point cloud Figure 2(d).

![Figure 2](image)

(a) collected point cloud  (b) point cloud after downsamplin  (c) point cloud after ground removing  (d) point cloud after Euclidean clustering

Figure 2 Point cloud processing effect

It can be clearly seen in Figure 2(d) that when the distance between vehicles and the surrounding bushes is close, it is hard to separate them only by distance information. We introduce intensity information to improve segmentation performance.

In order to find an appropriate intensity difference threshold, so that the vehicle and the bush can be separated properly, we conducted a series of comparison experiments. we set different intensity threshold T, and observe the clustering effect. The experimental effect is as Figure 3:

![Figure 3](image)

(a)T=0.5  (b)T=0.8  (c)T=1  (d)T=3  (e)T=4

Figure 3 Different intensity threshold comparison

We can see from Figure 3, when the intensity threshold T is set to be small (T<0.8), the experimental results are overfitting. Even the points on the same vehicle, their intensity difference caused by incident Angle and other factors will be magnified, which results the roof and body being divided into two categories; When the intensity threshold T is set to be large (T>3), the experimental results are
underfitting. With the increase of threshold T, the point cloud around the vehicle gradually gathers into a class with the vehicle. It can be found from the experiment that it is more appropriate to set the threshold between 1-3. In our method, we chose the intensity threshold T=2.

The training data for the Random Forest classifier were derived from the KITTI data sets and vehicle data we collected. After the target of point cloud is processed by the classifier, it can be correctly classified into corresponding categories. We annotate the final segmented vehicles with a bounding box.

In order to verify the effect of our algorithm, we carried out experiments in a variety of environments. The experimental results shows in Figure 4.

![Original(a) Original(b) Original(c)](image)

![Processed(a) Processed(b) Processed(c)](image)

Figure 4 Algorithm run in different environments

In Figure 4, by comparing the original image with the processed image, we can find that in different complex scenes, our algorithm can accurately segment and identify the vehicle targets in the scene, and has strong robustness.

In order to test the timeliness of the algorithm, we counted the processing time of point cloud at each stage in multiple environments. The time consumption is shown in table 1. It can be seen from the table that the clustering process occupies the main time consumption in the whole algorithm, because this algorithm not only traverses the position information between each point, but also traverses the intensity information. 300000 unordered point clouds pass through the whole detection and classification process, which takes less than 1s, showing strong real-time performance.

| Points number | Downsampling (s) | Groundremove (s) | Clustering (s) | Classify (s) | Total (s) |
|---------------|------------------|-----------------|---------------|-------------|----------|
| scene1        | 300000           | 0.029           | 0.043         | 0.601       | 0.069    | 0.742    |
| scene2        | 300000           | 0.023           | 0.023         | 0.551       | 0.074    | 0.671    |
| scene3        | 299900           | 0.027           | 0.178         | 0.595       | 0.084    | 0.884    |
| scene4        | 300100           | 0.027           | 0.102         | 0.598       | 0.074    | 0.801    |
| Average       | 300000           | 0.027           | 0.087         | 0.586       | 0.075    | 0.775    |

We conducted comparative experiments to verify the accuracy of our algorithm. In segment part, we collected 38 point cloud pictures of different scenarios, including a total of 56 vehicles. according to the clustering algorithm with distance information, 38 vehicles segmented, segmentation precision was 67.9%, some vehicle connected with bushes and cannot be separated. After the introduction of intensity
information, 50 vehicles segmented, segmentation rate was 89.3%, effects improved by 31%, as shown in Table 2.

| Table 2 | The comparison of segment result with different cluster methods |
|---------|---------------------------------------------------------------|
|          | Total number of vehicle | Cluster with distance information | Cluster with distance & intensity information |
| Number of vehicle | 56 | 38 | 50 |
| Segmentation precision | 67.9% | 89.3% |

4. Conclusion

Based on the characteristics of DJI Livox Mid-40 Lidar, this paper presents a vehicle detection method, which mainly contains three steps: First, raw point clouds are downsampled and ground points are removed to obtain objects above the ground. Then, a improved Euclidean clustering algorithm combines coordinate information and intensity information to promote the clustering accuracy. Finally, suspected targets are classified by the random forest. In the DJI Livox Mid-40 dataset, our method achieved 89.3% precision in the vehicle detection experiment. Compared with the traditional Euclidean clustering, our method improves the detection accuracy by 31%. The clustering accuracy can be improved when intensity information clustering is introduced. However the intensity of the reflection point cloud is also affected by the color of the object, sunlight, and so on. In future work, RGB and other information will be considered to improve detection accuracy.

References

[1] Yang B, Luo W, Urtasun R. Pixor: Real-Time 3D Object Detection From Point Clouds[C]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018: 7652-7660.
[2] Shang E, An X, Wu T, et al. LiDAR Based Negative Obstacle Detection for Field Autonomous Land Vehicles[J]. Journal of Field Robotics, 2016, 33(5):591-617.
[3] Zeng Y, Hu Y, Liu S, et al. RT3D: Real-Time 3-D Vehicle Detection in LiDAR Point Cloud for Autonomous Driving[J]. IEEE Robotics and Automation Letters, 2018, 3(4):3434-3440.
[4] Wang M L, Li Y, Yang Y, et al. Localization and mapping in urban area based on 3D point cloud of autonomous vehicles[J]. Journal of Beijing Institute of Technology, 2016, 25(4):473-482.
[5] Ester M. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[J]. Proc.int.conf.knowledg Discovery & Data Mining, 1996.
[6] Fischler M, Bolles R. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. communications of the ACM, 981, 24(6):381-395.
[7] Zhang Y, Wang J, Wang X, et al. Road-Segmentation-Based Curb Detection Method for Self-Driving via a 3D-LiDAR Sensor[J]. IEEE Transactions on Intelligent Transportation Systems, 2018.
[8] Wen L, He L, Gao Z. Research on 3D Point Cloud De-distortion Algorithm and Its Application on Euclidean Clustering[J]. IEEE Access, 2019, PP(99):1-1.
[9] A T N T, B K D, B M D. Revised DBSCAN algorithm to cluster data with dense adjacent clusters[J]. Chemometrics and Intelligent Laboratory Systems, 2013, 120(2):92-96.
[10] Reymann C, Lacroix S. Improving LiDAR Point Cloud Classification using Intensities and Multiple Echoes[C]// IEEE/RSJ International Conference on Intelligent Robots & Systems. IEEE, 2015:5122-5128.
[11] Angelini F, Fu Z, Velastin S A, et al. 3D-Hog Embedding Frameworks for Single and Multi-Viewpoints Action Recognition Based on Human Silhouettes[J]. 2018:4219-4223.
[12] Liaw A, Wiener M. Classification and Regression by randomForest[J]. R News, 2002, 23(23).