

Original Paper

Obstacle Detection Based on 3D Lidar Euclidean Clustering

Chen Jinming¹

¹ School of Electrical and Information Engineering, Jiangsu University, Zhenjiang Jiangsu, 212013 China

Received: October 15, 2021 Accepted: November 1, 2021 Online Published: November 8, 2021
doi:10.22158/asir.v5n3p39 URL: <http://doi.org/10.22158/asir.v5n3p39>

Abstract

Environment perception is the basis of unmanned driving and obstacle detection is an important research area of environment perception technology. In order to quickly and accurately identify the obstacles in the direction of vehicle travel and obtain their location information, combined with the PCL (Point Cloud Library) function module, this paper designed a euclidean distance based Point Cloud clustering obstacle detection algorithm. Environmental information was obtained by 3D lidar, and ROI extraction, voxel filtering sampling, outlier point filtering, ground point cloud segmentation, Euclidean clustering and other processing were carried out to achieve a complete PCL based 3D point cloud obstacle detection method. The experimental results show that the vehicle can effectively identify the obstacles in the area and obtain their location information.

Keywords

Environment sensing, LiDAR, PCL, Euclidean clustering

1. Introduction

In recent years, with the development of unmanned driving technology, obstacle detection has become an important research direction of environmental perception (Kostavelis, Nalpantidis, & Gasteratos, 2011). It is of great significance for the safety of intelligent driving to quickly and accurately detect obstacles in the direction of vehicle driving and obtain their location information. At present, obstacle detection methods are mainly based on vision sensor and laser sensor. Visual cameras do not interfere with each other, and can detect obstacles from a wider perspective without scanning. Meanwhile, images contain rich information and high data resolution. Liu, R., Chen, F. X., Chen, K. Y. and Liu, S. N. (2018) proposed an obstacle detection method based on bar graph. By using binocular vision sensor to obtain images for image processing, the image is divided into several histogram areas and the obstacle depth information in each histogram area is extracted. Huang, Zhu, Wang and Liu (2019) proposed to use binocular direct sparse mileage calculation method to construct a three-dimensional

space model, convert the constructed three-dimensional point cloud graph into a two-dimensional grid graph and mark obstacles.

However, the camera is greatly affected by ambient light and dust, and the data processing algorithm takes a long time and has high requirements on the computer, so the real-time performance of the algorithm may not be guaranteed. Lidar is widely used in target detection in various environments because of its high accuracy and strong anti-interference ability (Zong, Wen, & He, 2019).

PCL is a large cross-platform open source C++ programming library established on the basis of absorbing the previous researches on point cloud, realizing a large number of general algorithms and efficient data structures related to point cloud (Zhu, Guo, Su et al., 2012). Based on PCL (Point Cloud Library), this paper designs a Euclian-clustering obstacle detection method, which takes environmental data collected by 3D lidar VLP-16 as input, converts Point Cloud data into data structure under PCL through format conversion. Obstacles were detected by region of interest extraction, voxel filtering sampling, statistical filtering to remove outliers, ground point cloud segmentation and Euclidean clustering. The experimental results show that this method can detect the obstacles on the road effectively and has good real-time performance.

2. Method of Obstacle Detection

The sensor selected in this paper is Velodyne's VLP-16 3D lidar (Figure 1), and the conversion relationship between its polar coordinates and Cartesian coordinates is shown in Figure 2.



Figure 1. Velodyne VLP-16

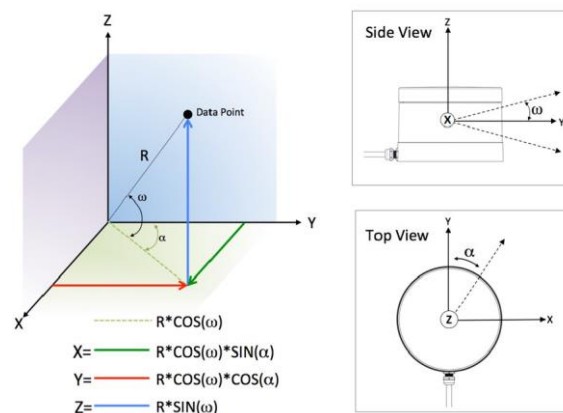


Figure 2. VLP-16 Coordinate System Transformation

Where, (x,y,z) is the Cartesian coordinate value of the laser sensor, R is the distance between the reflected object and the lidar, α is the horizontal rotation angle, ω is the vertical angle.

The algorithm flow of VLP-16 for euclidean clustering obstacle detection is shown in Figure 3: (1) The data is collected by lidar and converted into cartesian coordinate representation under PCL by data type; (2) In order to reduce the interference of distant obstacles and reduce the number of point clouds, regions of interest (ROI) were extracted from the original data; (3) In order to focus on object detection

on road surface, ground point cloud is segmented for point cloud data and ground point cloud is filtered; (4) In order to improve the accuracy and real-time performance of clustering, sampling under voxel filtering and statistical filtering were used to remove outliers respectively. (5) Kd-tree accelerated euclidean clustering was performed on the pre-processed data, and the clustering results were obtained.



Figure 3. Flow Chart of Euclidean Clustering Algorithm

2.1 Lidar Data Preprocessing

Because of its high precision and strong anti-interference ability, lidar is widely used in environmental perception in various occasions, scanning objects in the environment to obtain the location information of targets. The rich point cloud data of 3D lidar improves the accuracy of detection, but it also brings some problems to algorithm processing. 3D lidar can generate hundreds of thousands or millions of point cloud data per second. If these point clouds are directly processed, the speed of data processing will be seriously reduced, and the real-time performance of the algorithm is difficult to meet the requirements of detection. At the same time, the accuracy of target detection will be affected by outliers caused by tree branches and leaves and noise caused by distant tree rows.

Therefore, it is necessary to preprocess the original laser point cloud data before point cloud clustering to identify the adjacent rows of fruit trees in the orchard.

2.1.1 Region of Interest Refers

Region of Interest (ROI) refers to the effective area where the vehicle is driving and the road segment with a certain distance ahead. Accurate regional data extraction can filter out point cloud data generated by distant obstacles, which can greatly reduce the amount of data and focus on the target to be detected. The top view of the vehicle and lidar is shown in Figure 4. To filter out interference from distant obstacles, thresholds were set in the vehicle forward (X) direction, the vehicle on both sides (Y) direction and the vehicle vertical (Z) direction respectively, and the point cloud data that did not meet the following conditions were put into the Extract Indices filter in PCL for filtering, so as to extract the point cloud subset of the region of interest from the original point cloud data.

$$\begin{cases} X_{\min} < x_i < X_{\max} \\ Y_{\min} < y_i < Y_{\max} \\ Z_{\max} > z_i \end{cases} \quad (1)$$

In the formula

x_i, y_i, z_i — coordinate value of lidar at point i

X_{\min}, X_{\max} — the distance threshold in the forward direction(X) of vehicle

Y_{\min}, Y_{\max} — the distance threshold on the both sides direction(Y) of vehicle

Z_{min}, Z_{max} ——the distance threshold of vertical direction(Y) of vehicle

According to the above formula (1), point cloud data of vehicles in the effective driving area can be extracted. However, since the 3D lidar itself has a certain angle field of view (Velodyne VLP-16: $-15^{\circ} \sim +15^{\circ}$), the collected point cloud data may contain information of the vehicles itself. The data should not affect subsequent detection results and should also be regarded as noise points for filtering.

$$d_i = \sqrt{x_i^2 + y_i^2} \quad (2)$$

In formula (2), d_i is the distance between the detected object and the lidar. When the reflection distance d_i is less than the set threshold D, the corresponding point cloud data is filtered out.

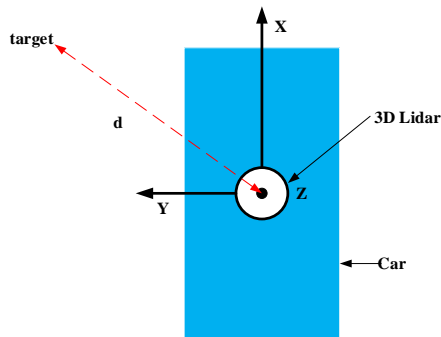


Figure 4. Top Down View

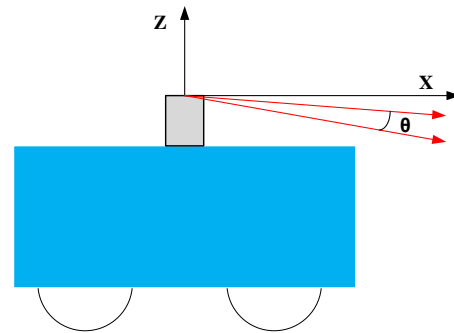


Figure 5. Side View

2.1.2 Ground Point Cloud Segmentation

Environmental sensing based on lidar is usually only interested in targets on the road surface. In order to quickly and accurately detect the location of obstacles, ground and non-ground points need to be separated before cluster detection (Miao & Tseng, 2004; Reitberger, Schnoerr, Krzystek, et al., 2009; Gamal, Wibisono, Wicaksono, et al., 2020). In this paper, the Ray Ground Filter method is adopted for ground filtration, and the flow chart of this method is as follows:

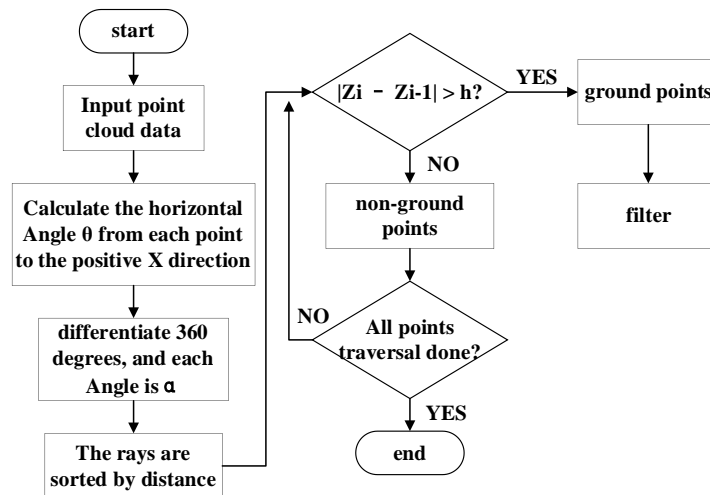


Figure 6. Flow chart of Ray Ground Filter

In Figure 6, angle α is the horizontal angular resolution of the laser radar VLP-16. When the frequency of VLP-16 is 10Hz, $\alpha = 0.2^\circ$; Z_i and Z_{i+1} are the values in the Z direction (altitude direction) of point i and point $i+1$ sorted by ray distance respectively, and h is the pre-set slope threshold. First of all, the point cloud after ROI extraction is input, and angle differentiation is performed for 360° , with each angle α being 0.2° . After angle differentiation, $360/0.2=1800$ rays can be obtained. Secondly, ascending order is carried out according to ray distance. Finally, ground point cloud is judged by comparing the relationship between height difference and slope threshold h of the two point clouds before and after. When the height difference is greater than the slope threshold h , it is judged to be ground point cloud and filtered. Otherwise, non-ground point cloud is considered. The calculation of slope between Angle differential and point cloud is shown in Figures 7 and 8.

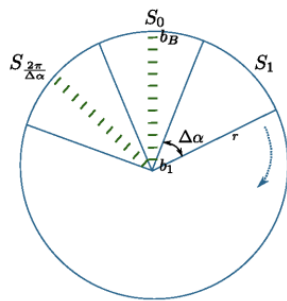


Figure 7. Angle Differentiation

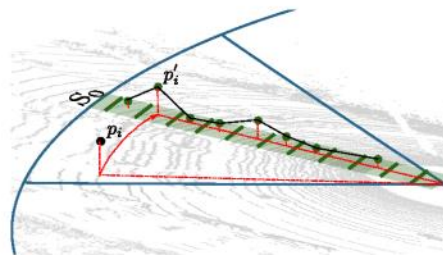


Figure 8. Slope Calculation

2.1.3 Sampling under Voxel Filtering

After the original data was extracted by ROI and ground cloud point segmentation, the amount of point cloud data was greatly reduced, but the data of lidar was still large. In order to reduce the number of reduced points and maintain the shape characteristics of the point cloud, this paper adopts the voxel

grid method to realize downsampling. A 3D voxel grid was created through the input point cloud data extracted from ROI, and then the barycenter of all points in the voxel was used to approximate the other points in each voxel. The outliers generated by occlusion of branches and leaves are filtered by statistical filter in PCL open source library, calculating the distance distribution between the midpoint of input data and adjacent points. For each point, calculate the average distance \bar{d} between the point and all adjacent points. Assuming that the result obtained is a gaussian distribution, the points whose average distance \bar{d} is outside the standard range are defined as outliers and filtered from the data.

2.2 Point Cloud Clustering Based on Euclidean Distance

Kd tree (short for K-dimensional tree) is a data structure that divides k-dimensional data space. It is mainly used to search key data in multi-dimensional space. Based on PCL's Nearest Neighbor open Source library (FLANN), this paper conducts fast nearest neighbor retrieval for 3D point cloud data to improve the clustering speed to meet the real-time requirements of perception.

Euclidean clustering is a kind of clustering method which uses Kd-tree to search the nearest neighbor and takes euclidean distance as reference (Gamal, Wibisono, Wicaksono, et al., 2020). The algorithm compares the distance between two points with the clustering radius ε , and the distance smaller than the clustering radius ε is regarded as the same class. The calculation formula of euclidean clustering is as follows:

$$dist(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3)$$

Where, (x_i, y_i) is the cartesian coordinate of the data at point i .

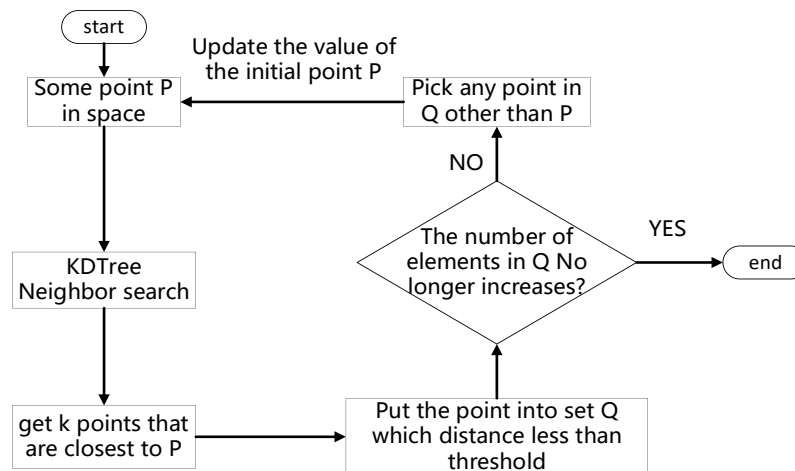


Figure 9. Flow Chart of Euclidean Distance Clustering

For a certain point P in the space, k points closest to P are obtained according to Kd-tree nearest neighbor search, and the Euclidean distance between two points is calculated by formula (3) and compared with the clustering radius ε . When the Euclidean distance is less than the set cluster radius ε , the point is added to the cluster set Q. When the number of elements in Q continues to increase, points other than P are selected from set Q and the above process is repeated. Otherwise, the clustering ends.

3. Test Results and Analysis

3.1 Region of Interest Extraction Test

The area of interest of vehicles is usually the road surface with a certain width and a distance in the forward direction. The area threshold set in this paper is $X[0m,20m]$, $Y[0.4m,2m]$, $Z[0m,3m]$. The effect of region of interest extraction is shown in the figure below, where the number of point clouds changes from 20358 to 1832, which can effectively reduce the number of point clouds while filtering distant obstacle.

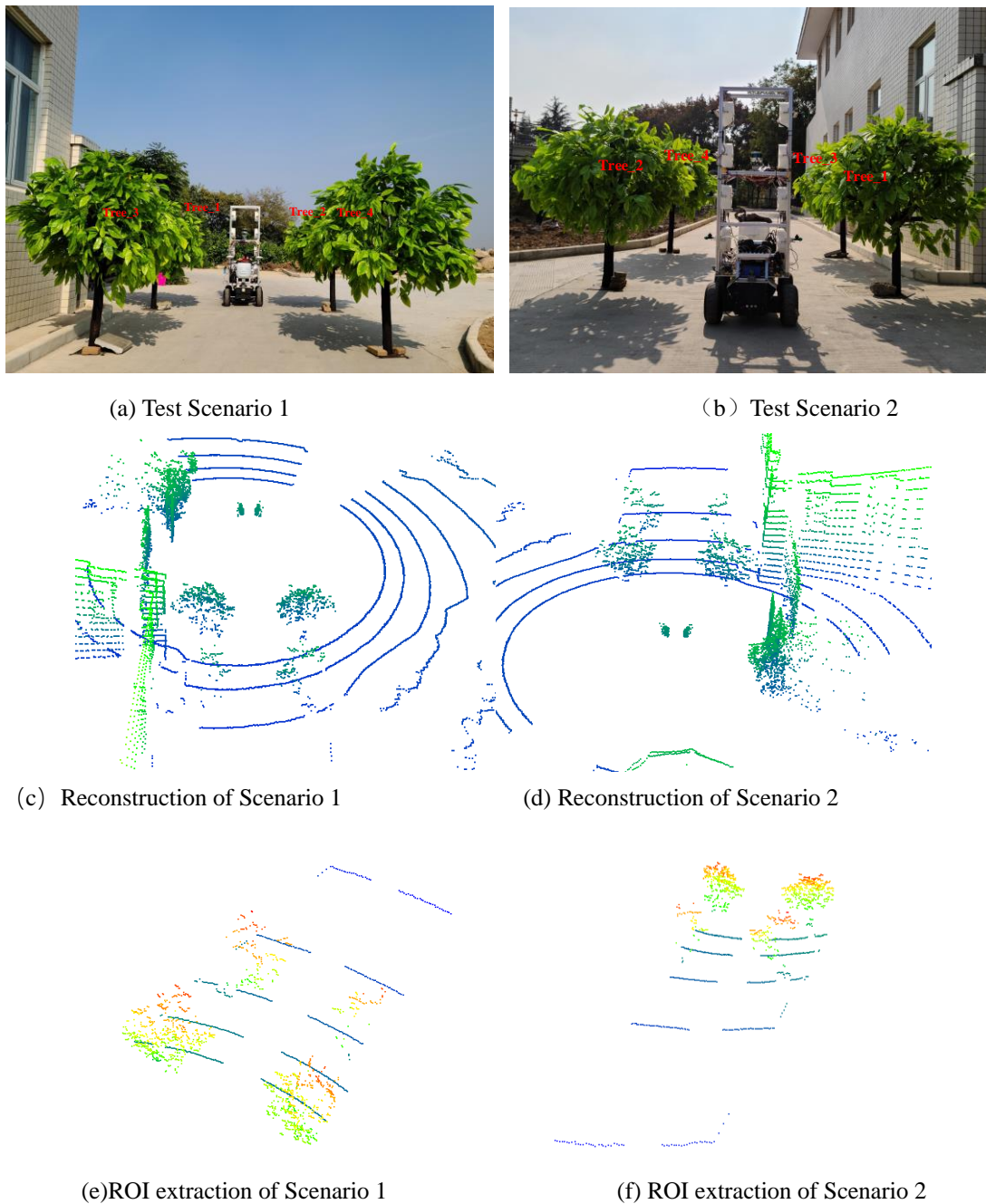


Figure 10. ROI Extraction

As can be seen from Figure 10 (c and d), noise points are generated when the laser scans the vehicle itself during scanning, but the noise points are filtered out by formula 2. After filtering out the noise points generated by the vehicle itself and the point clouds in the non-area of interest, its effect is shown in Figure 10 (e and f).

3.2 Ground Point cloud Segmentation Test

When an intelligent vehicle is on the road, it only pays attention to the situation on the ground. Point clouds on the ground do not contribute to obstacle detection, and these point clouds also affect the accuracy of obstacle detection on the road. In this thesis, we use the Ray Ground Filter method to distinguish the ground point cloud and non-ground point cloud. After segmentation, the number of point clouds changed from 1831 to 1303, and its effect is shown in Figure 11.

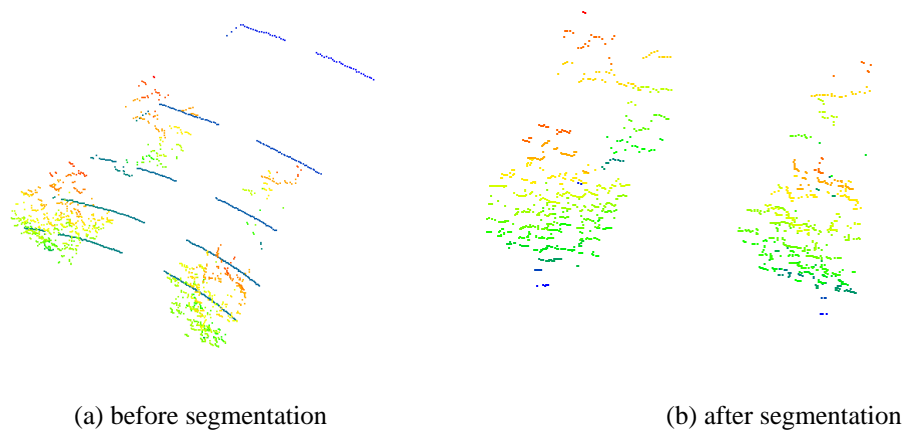


Figure 11. Ground Point Cloud Segmentation

3.3 Point Cloud Filtering Test

According to the mesh method, it is found that the filtering effect is best when the cube with the side length of the subsampled voxel is 0.2 cm. The number of point clouds changed from 1303 to 972, which kept the characteristic shape of the object while reducing the number of point clouds. For outliers generated by object occlusion, the point cloud processed by downsampling is put into PCL statistical filter for outlier removal, and the effect is shown in the Figure 12.

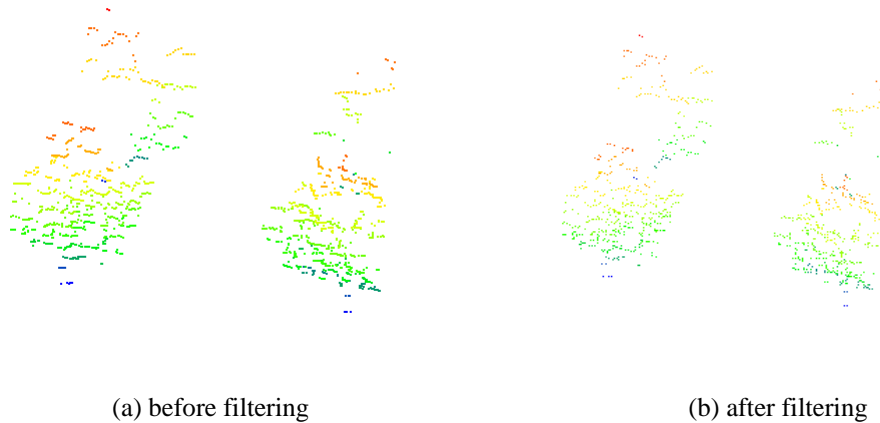


Figure 12. Point Cloud Filtering

3.4 Euclidean Clustering Test

After a series of point cloud data preprocessing such as region of interest extraction, ground point cloud segmentation and point cloud filtering, this paper conducts clustering operation based on Euclidean distance for the processed point cloud. After several experiments and parameters debugging, it is found that the clustering effect is better when the clustering radius threshold is 0.05 and the minimum number of clustering points threshold is 30. After writing the clustering effect into the point cloud file and performing visual operation, the effect is shown in Figure 13.

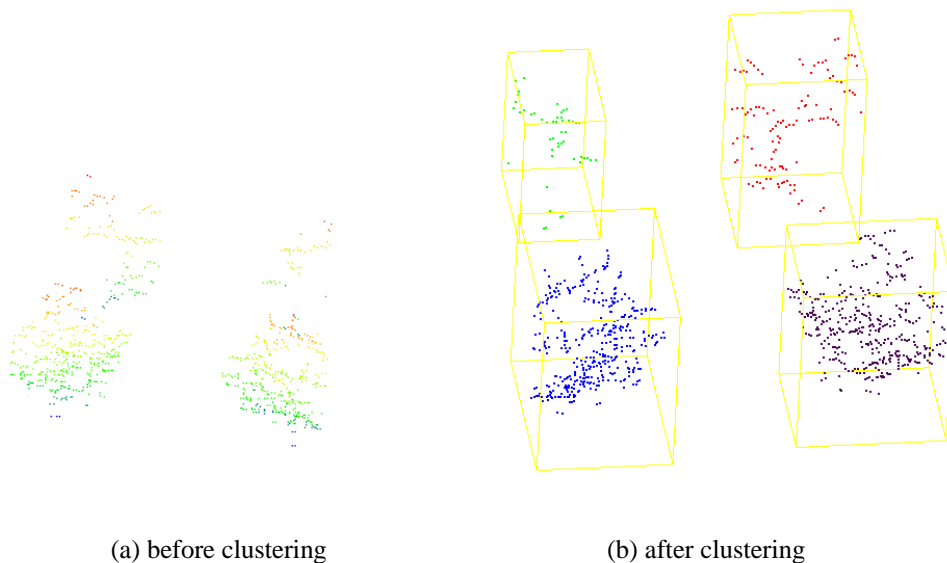


Figure 13. Clustering Rendering of Euclidean Distance

4. Conclusion

In this paper, a PCL based Euclidean clustering obstacle detection method is implemented, and experimental verification is carried out on the experimental platform, finally, the following conclusions are obtained:

(1) PCL point cloud library has excellent point cloud data processing ability and can operate point cloud data directly, which lays a good foundation for subsequent detection and algorithm design.

(2) After the point cloud data is processed by region of interest extraction, ground point cloud segmentation and point cloud filtering, the number of point clouds is greatly reduced and interference such as noise points and outliers are removed.

(3) The Euclidean clustering based on KD-tree acceleration can get better clustering effect, and the algorithm has better real-time performance.

(4) The effect of Euclidean clustering is greatly affected by parameters. In order to get better clustering effect, it is necessary to debug parameters according to the actual situation. The subsequent research will build an adaptive clustering radius based on the size of the distance to improve the robustness and applicability of the algorithm.

References

- Gamal, A., Wibisono, A., Wicaksono, S. B., et al. (2020). Automatic LIDAR building segmentation based on DGCNN and euclidean clustering. *Journal of Big Data*, 7(1). <https://doi.org/10.1186/s40537-020-00374-x>
- Huang, Y.-T., Zhu, X.-H., Wang, J.-H., & Liu, R. (2019). Obstacle detection and map construction of unmanned ship based on binocular vision. *Computer measurement and control*, 27(08), 193-197.
- Kostavelis, I., Nalpantidis, L., & Gasteratos, A. (2011). Supervised traversability learning for robot navigation. *Proceedings of the 12th Annual Conference on Towards Autonomous Robotic Systems*, pp. 289-298. https://doi.org/10.1007/978-3-642-23232-9_26
- Liu, R., Chen, F. X., Chen, K. Y., & Liu, S. N. (2018). Research on obstacle detection method based on binocular vision. *Computer measurement and control*, 26(12), 67-71.
- Miao, W., & Tseng, Y. H. (2004). LIDAR data segmentation and classification based on octree structure. *Parameters*.
- Reitberger, J., Schnoerr, C., Krzystek, P., et al. (2009). *3D segmentation of single trees exploiting full waveform LIDAR data* (pp. 561-574). <https://doi.org/10.1016/j.isprsjprs.2009.04.002>
- Zhu, D. H., Guo, H., Su, W., et al. (2012). *Point cloud library PCL Learning Tutorial*. Beijing: Beihang University Press.
- Zong, C. F., Wen, L., & He, L. (2019). Three-dimensional lidar obstacle detection technology based on Euclidean clustering algorithm. *Journal of Jilin University (Engineering and Technology Edition)*, 6.