Real-Time Segmentation of Sparse 3D Point Clouds on a TX2

Yuchen Liu, Hongsai Zhu, Ben Zhang, Yan Su

ABSTRACT

In this paper, we propose a real-time 3D object detection method, which from sparse 3D point clouds and based on NVIDIA Jetson TX2. This effective method first uses the external parameter matrix to transform the coordinates of the point clouds data, then supplements the missing data in the original data through the interpolation algorithm, and then uses the RANSAC algorithm to remove the ground, uses the improved DBSCAN clustering algorithm, which clustering segmentation is performed for the point clouds fusion partition distance threshold and angle threshold of different distance intervals. Finally, we regress the obstacle boundary to the mini-box by gradient descent, output for the result is visualized in the form of cuboid detection. Compared to traditional DBSCAN algorithm, our method reduces the missed detection rate and improves the real-time performance. Above all, we implements our approach in C++ and ROS, and the experimental results show that our method produces high quality object detection by using NVIDIA Jetson TX2 online.

KEYWORDS

NVIDIA jetson TX2, object detection, interpolation, RANSAC, improved DBSCAN clustering.

INTRODUCTION

As one of the modern technology products have the strongest impact on human society, autonomous vehicles play an important part in the development of the social, economic, technology and national defense construction. The recent activity in the area of autonomous vehicles navigation has initiated a series of reactions that stirred the automobile industry, pushing for the fast commercialization of this technology which, until recently, seemed futuristic[1]. Perception is key for autonomous driving technologies, which determines it’s ability directly, and perception is also the primary guarantee for the safety and intelligence[2]. The task of environmental perception is to identify the surrounding data through on-board sensors, so as to provide vital information.

Yuchen Liu, Hongsai Zhu, Ben Zhang, Yan Su, School of Mechanical Engineering Nanjing University of Science and Technology Nanjing, China.
support for the navigation and positioning, path planning, and control of autonomous vehicles and mobile devices, 3D object detection is the most important 3D perception tasks[3].

Object detection based on LiDAR sensors is one of the main methods currently applied to the detection of objects in autonomous vehicles[4]. In case of LiDAR sensors with the ability to capture 360 degrees of information, the data is represented as a set of 3D points called a point clouds which is organized in layers. The points in each layer are also organized in an elliptical fashion and the starting points of all elliptical layers are considered to share the same orientation. Known for the ability to portray this information, LiDAR sensors are popular among academic research teams for their incomparable advantages such as long range, satisfactory accuracy, strong directionality and anti-interference ability, recently the development of hardware expected to bring lower cost and smaller size, which has aroused extensive attention.

The 64-beam LiDAR or even more beams are generally large and expensive, which is not convenient for mobile devices and increase the difficulty of mass-production commercial. Our method selects a 16-beam LiDAR as the only sensor and creates a reasonable algorithm system to detect objects from sparse 3D point clouds, which builds a objects detection system performances well in detection accuracy and time cost. The 16-beam LiDAR has smaller amount of data in a single frame, this paper propose a real-time 3D object detection method that from sparse 3D point clouds and based on NVIDIA Jetson TX2, the overall cost of the perception system is relatively low, which is conducive to the commercialization process in mass-production of autonomous vehicles in specific scenarios and other mobile devices[5]. Online objects detection research with mobile CPU and low cost sensors, has great application value for the perception of autonomous vehicles and other related intelligent mobile devices[6].

This paper mainly describes a TX2-based real-time objects detection method by using sparse 3D point clouds, by filling the blank of invalid point clouds data segments through the interpolation algorithm, combining multi-threaded programming and using the range-data to optimization the segmentation algorithms, playing this effective low-cost 3D objects detection system on the mobile platform—NVIDIA Jetson TX2.

RELATED WORK

The basic process of the objects detection method applied to 3D point clouds is roughly the same: 1) Calibrate the external parameters of the 3D LiDAR relative to the mobile devices, and get the original point clouds. 2) Preprocess the original point clouds, including coordinate conversion and smoothing. 3) Fit and remove the reference plane according to the feature of the point clouds. 4) Perform clustering and segmentation of objects point clouds, and extract the effective part to regress mini-box.
Fitting and removing the reference plane is a frequently used preprocessing step, which has an important auxiliary effect on the segmentation of 3D point clouds. Plane fitting, continuous straight line, grid height difference, and feature extraction method are the most commonly used methods for reference plane extraction and removal. The most classic method is to use the RANSAC to fit the reference plane and remove the point clouds nearly.

Segmentation is the key point of objects detection. Objects detection is usually segmented only by a single laser scan of the 3D point clouds has three categories: clustering and other related improvement algorithms, raster map, and machine learning separation. Among them, Ester Martin et al. proposed the clustering algorithm DBSCAN, is the most classic density-based clustering method[7], which can effectively solve the irregular problem of data. And recently 3D point cloud segmentation’s algorithms are still mostly based on DBSCAN to improve and optimize[8].

**METHODOLOGY**

The following paragraphs describe in detail the complete methodology for the real-time segmentation of sparse 3D point clouds on an TX2, which is mainly composed of three parts: point clouds preprocessing, clustering segmentation, and output for the result.

As Figure 1 shows, the preprocessing part performs coding conversion, coordinate conversion and missing point clouds difference complementation, convert the original input point cloud into rang data then preprocessed point cloud in turn; the segmentation part performs the fitting and removal of the reference plane, segmentation with the objects point clouds, get point cloud without plane and segmented one after another; the last step output part includes regress the mini-box and visualization.

Figure 1. The system flow chart, shows the details of the method. The three dashed boxes represent three major steps respectively.
Point Clouds Preprocessing

The mechanical LiDAR detects the surrounding scene by receiving the transmitted signal returned by scanning to the surroundings. The laser probe on each layer emits laser light at equal angular intervals during the 360° rotation and calculates the distance to the nearest obstacle point in the current direction through the flight time. So theoretically the amount of data in each frame is the same, is layers Multiplied by the number of launch points per layer, and the point cloud data in the cartesian coordinate system can be converted into range data by follows:

\[
dis = \sqrt{x^2 + y^2 + z^2}
\]
\[
r = 16 - (\sin^{-1}(\frac{z}{dis}) + 17)/2\)
\[
c = \cos^{-1}(\sqrt{\frac{dis^2 - z^2}{}}).
\]

(1)

Perform conversion, where \(r\) and \(c\) represent the number of layers and angle coordinates in the corresponding code, respectively. In this experiment, the LiDAR is used in the 16 layers and 2016 point clouds each layer, so that the point clouds data in the rectangular coordinate system is converted into the range data, to facilitate post-processing.

The point cloud data is collected by regarding LiDAR as the reference coordinate. The rotation matrix \(R\) and the translation matrix \(T\) of the external parameters of the LiDAR are calibrated by:

\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = R \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + T
\]

(2)

The point clouds set is converted into a point cloud set with the mobile devices as the reference coordinate system[9], so that the objects detection result is also based on the mobile devices.

Affected by the measured environment, the original lidar has a lot of missing data. This paper judges the left and right effective values of the missing point cloud segment in each harness, selects missing point clouds whose missing distance is within the threshold range to fill in the difference, and accelerates this preprocessing process through multi-threading. Figure 2 shows the effective of integration, the red parts is the missing data, while the other gray value represents different distance.
Figure 2. Supplement the missing data in the original data through the interpolation algorithm. Top: The raw range data. Bottom: The data after interpolation.

Cluster Segmentation

The point clouds in the space are usually connected by the ground or other reference, which is why the fitting of the reference plane and the removal of related point clouds are always part of the cluster segmentation[10].

The usual reference plane fitting and removal are done by the Random Sample Consensus (RANSAC), which has a very good plane extraction ability, but based on the characteristics of the point clouds, this part is optimized in the encoding format data. This paper uses a threshold distance point clouds near the installation height of the LiDAR to fit the RANSAC plane with a reference random seed, and then selects the 9-16 line (the emission angle is inclined downward) to search, removes the point clouds nearly, and speed up this process by multithread programming. Algorithm 1 shows the programs’ details of the algorithm process.

```
Algorithm 1: Reference plane fitting and removal.

Results: label: mark the reference point cloud as -1
1. Initialization:
2. \( P \): input point cloud as range data
3. \( P_f \): point clouds near the installation height of the LiDAR
4. \( N_{it} \): number of iterations
5. \( T_{thr} \): threshold distance from the plane
6. Main Loop:
7. for \( i = 1 \) : \( N_{it} \) do
8. model = RANSAC \((P_f)\);
9. if Evaluation(model) > score then
10. score = Evaluation(model);
11. \[ a, b, c, d = \text{model}; \]
12. end
13. end
14. for \( r = 9 \) : 16 do
15. for \( c = 1 \) : 2016 do
16. if Distance\( (P_f[r][c]) < T_{thr} \) then
17. \[ \text{label}[r][c] = -1; \]
18. end
19. end
20. end
21. Distance\( (k) \):
22. return \( \frac{\text{abs}(a \cdot k_1 + b \cdot k_2 + c \cdot k_3 - d)}{\sqrt{a^2 + b^2 + c^2}} \);
Clustering segmentation is the most important part of this methodology. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is the most commonly used algorithm in point clouds objects detection. However, due to its segmentation effect, some improvements are usually made during use, such as modifying the distance calculation format of clustering[11].

The 16-beam LiDAR used in this experiment has a large vertical gap and a large geometric distance between adjacent points in the distance. The traditional DBSCAN is very inapplicable, so in our method, the comprehensive segmentation judgment is fusing the angle threshold and the distance of the segmented range data. First get the distance by:

$$d_{i1} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + 0.3 \times (r_1 - r_2)^2},$$

(3)

The distance threshold is divided into different number, Then compute the angle using trigonometric rules as follows:

$$d_1 = \sqrt{x_2^2 + y_2^2 + z_2^2}$$
$$d_2 = \sqrt{x_1^2 + y_1^2 + z_1^2}$$
$$\beta = 360^\circ \times |r_1 - r_2|/2016$$
$$d_{i2} = \tan^{-1}\left(\frac{d_2 \sin \beta}{d_1 - d_2 \cos \beta}\right),$$

(4)

When the distance and the angle threshold both reach the clustering standard, it is considered to belong to the same object point clouds set.

Searching through the whole point clouds is a waste of time obviously when finding other point clouds of one object. In addition, the distance calculation method in the method in this paper is closely related to the encoding position of the range data, so the search classification judgment is also performed by the domain BFS method. For each unlabeled point cloud, perform the search and distance calculation of the surrounding point clouds, add the point clouds within the threshold range to the current set and mark them with the same label’s number, until no new point cloud is added to the current object point clouds set, search for the next point clouds which is unlabeled. The algorithm process is shown in Algorithm 2:
The point clouds set after clustering is divided into independent objects point cloud sets. Figure 3 shows the segmentation result by bird view, while the yellow rectangle represents the projection part of the car body, the green point clouds represents the reference plane point clouds, and the different colors represent the point clouds that are segmented into different objects point cloud sets.
Figure 3. The segmentation result by bird view. The yellow rectangle represents the projection part of the car body, the green point clouds represent the reference plane point clouds, and the different colors represent the point clouds that are segmented into different objects point cloud sets.

Output for the Result

As shown in Figure 4, the gradient descent method is used in the bird view perspective to perform the regression of the minimum area rectangle for each segmented objects point clouds sets, and the distance from the highest point in the target point cloud collection to the ground point corresponding to the position is the qualified object.

Figure 4. The bird view perspective of the result. Each yellow rectangle represents a object’s projection in bird view.

RESULTS & CONCLUSIONS

In order to verify the effectiveness and real-time performance of the method in this paper, we have examined our methodology based on the NVIDIA Jetson TX2 [14], connected to the 16-beam LiDAR by robosense, which is placed on the top of the vehicle, and used the wire-controlled electric vehicle as the mobile platform as shown in Figure 5, the outdoor scene is tested, and the interception is within 25m of the car. The point cloud data is used to detect the objects of the mobile devices.
The method proposed in this paper has a higher recall rate and a higher accuracy rate than the traditional DBSCAN algorithm for objects detection, which reduces the missed detection and false merging. Figure 6 certificate our method improved distinctly, some cars and trees can be detected correctly while missed in traditional DBSCAN.

In our experimental evaluation, we used our default parameter angle threshold $\theta_{d2} = 10^\circ$, and set the distance threshold by interval as follows:

$$
\theta_{d1} = \begin{cases} 
0.5 & 0 \rightarrow 10 \\
1 & 10 \rightarrow 20 \\
1.5 & 20 \rightarrow \infty 
\end{cases}
$$

The method proposed in this paper reduces the time complexity by optimizing the recognition of the reference plane and the clustering search algorithm. The data processing time of the 16-beam LiDAR by robosense on the NVIDIA Jetson TX2 in the experiment is shown in Figure 7. The 16-beam LiDAR acquisition frequency is 10HZ, the segmentation average time is 59.29ms, and the all preprocessing is 77.25ms to meet the real-time requirements.
Figure 7. Timings obtained on the experimental platform. The x-axis depicts the index of individual point clouds while the y-axis shows the processing time in ms.

ACKNOWLEDGEMENTS

Corresponding author: Yan Su

REFERENCES

1. Zermas D, Izzat I, Papanikolopoulos N. Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications[C]// 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017.
2. Wang S F, Dai X, Xu N. Overview on Environment perception technology for unmanned ground vehicle[J]. Journal of Changchun University of Science and Technology (Natural Science Edition), 2017, 40(1):1-6.
3. Liu J. Research on key technologies in unmanned vehicle driving environment modelling based on a 3D LiDAR[D]. Hefei: University of Science and Technology of China, 2016.
4. Lu X H, Yao J, Tu J G, et al. Pairwise linkage for point cloud segmentation[J]. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016, III-3:201-208.
5. Wang X Y. Autonomous vehicles into the reality [J]. Financial Technology Time, 2016(7): 87-87.
6. J. Zhao, X. Huang, and Y. Massoud. “An efficient real-time fpga implementation for object detection,” in New Circuits and Systems Conference (NEWCAS), 2014 IEEE 12th International. IEEE, 2014, pp. 313–316.
7. Ester M, Kriegel H P, Xu X. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise[C]. International Conference on Knowledge Discovery and Data Mining. AAAI Press, 1996: 226-231.
8. Luo Z, Habibi S, Mohrenschildt M. LiDAR based real time multiple vehicle detection and tracking[J]. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2016, 10(6):1125-1132.
9. Ye G. Multi-target detection and tracking algorithm for autonomous driving car based on a 3D lidar in urban traffic environment[D]. Beijing:Beijing Institute of Technology, 2016.
10. Chu P, Cho S, Sim S, et al. A fast ground segmentation method for 3D point cloud[J]. Journal of Information Processing Systems, 2017, 13(3):491-499.
11. Bogoslavskyi I, Stachniss C. Efficient online segmentation for sparse 3D laser scans[J]. IAPG-Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 2017, 85(1):41-52.