Journal of Physics: Conference Series

PAPER • OPEN ACCESS

# Real-time Track Obstacle Detection from 3D LIDAR Point Cloud

To cite this article: Chao He *et al* 2021 *J. Phys.: Conf. Ser.* **1910** 012002

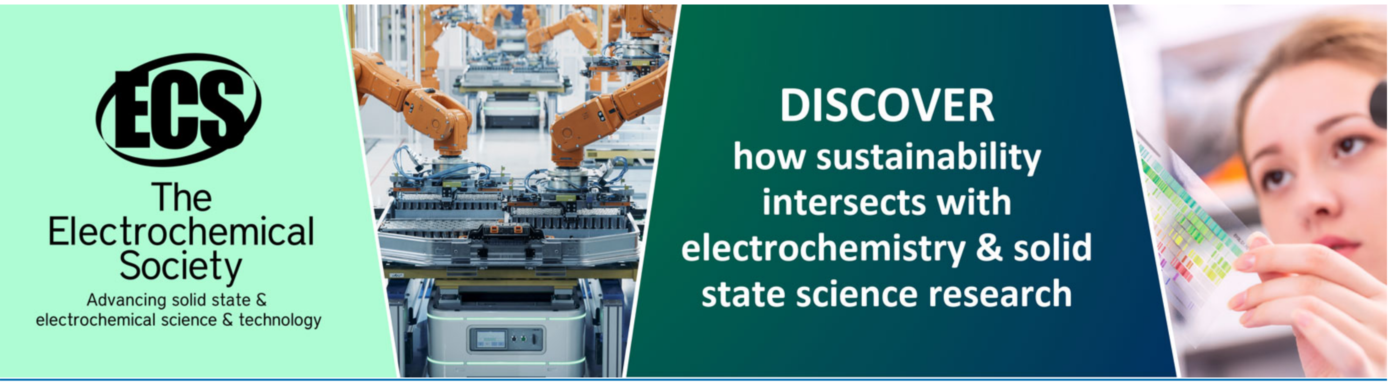
View the article online for updates and enhancements.

You may also like

- A Sensor Array-based Control System Design for the Interior Environment of Driverless Cars
Xinran Li

- Test Methods and Equipment for Assessment of Passenger Compartment Environmental Parameters of Modern and Advanced Driverless Vehicles
A M Saikin, S E Buznikov, A V Zhuravlev et al.

- Obstacle detection in dangerous railway track areas by a convolutional neural network
Deqiang He, Kai Li, Yanjun Chen et al.




This content was downloaded from IP address 13.58.151.231 on 26/04/2024 at 03:32

# Real-time Track Obstacle Detection from 3D LiDAR Point Cloud

**Chao He*, Jiayuan Gong, Yahui Yang, Dong Bi, Jianpin Lan and Leipeng Qie**

Institute of Automotive Engineers, Hubei University of Automotive Technology, Shiyan, China
Email: hcuestc@163.com

**Abstract.** Obstacle detection is one of the important works in the field of driverless vehicle .Camera, millimeter wave radar, LiDAR and other sensors are widely used in the obstacle detection of driverless vehicle. However, cameras and millimeter wave radar are highly dependent on the environment and greatly affected by external factors, especially in the process of high-speed driving, which is easy to produce serious system errors, and even unable to detect obstacles, which has a great potential for the safety of driverless vehicle. LiDAR is used to detect obstacles by launching laser, which is not easy to be interfered by the environment. It has high precision and is suitable for outdoor environment. The main contributions of this paper are as follows: 1) a 16 line LiDAR track recognition method based on driverless equation is proposed. 2) it is easy to get the obstacle coordinates through plane segmentation and clustering for the use of subsequent camera and LiDAR fusion.
**Keywords.** Driverless, Obstacle detection, LiDAR.

## 1. Introduction

In driverless technology, environmental obstacle is one of the most important field. Generally, sensors such as LiDAR, camera, millimeter wave radar and IMU will be installed on the vehicle to identify the surrounding environment such as lane line, vehicle and pedestrian, and these sensors will be used to realize vehicle positioning and mapping [1-3]. However, monocular camera cannot provide accurate location of obstacles.

Generally, triangulation and other methods are needed to calculate the real-time location of vehicles. However, this method not only cannot get accurate results, but also has limited detection distance. Although binocular and RGB-D cameras can get the position of obstacles, they are easy to be affected by the environment light and have poor stability, so they are difficult to be well applied outdoors. LiDAR not only has a long detection range, but also has a high stability. Especially in the driverless race, because the camera can only shoot at a fixed angle, some information will be lost when it encounters a curve [4-6]. Therefore, in obstacle detection, LiDAR becomes a good choice.

## 2. Overview

Specifically, to the application scenario of driverless formula racing, our track is made of the same shape cone barrel, the distance between pile bucket is 5 meters, and there are no other obstacles on the track (figure 1). Therefore, we only need to use LiDAR to identify the pile bucket on the left and right sides of the lane line, find the centre coordinates of pile bucket and then fit the curve to get the contour of the track.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

**Figure 1.** Driverless race track composed of pile buckets.

Similar to RGB-D camera, the point cloud data output 4 information, coordinates and reflection intensity of the point cloud. We use the American velodyne-16 line LiDAR, which supports 16 channels, can generate 300000 point cloud data per second, 360 degree horizontal scanning full angle [7-11].

After obtaining the point cloud data of LiDAR, we can achieve our goal by filtering, plane segmentation, clustering, and taking the center point (figure 2).

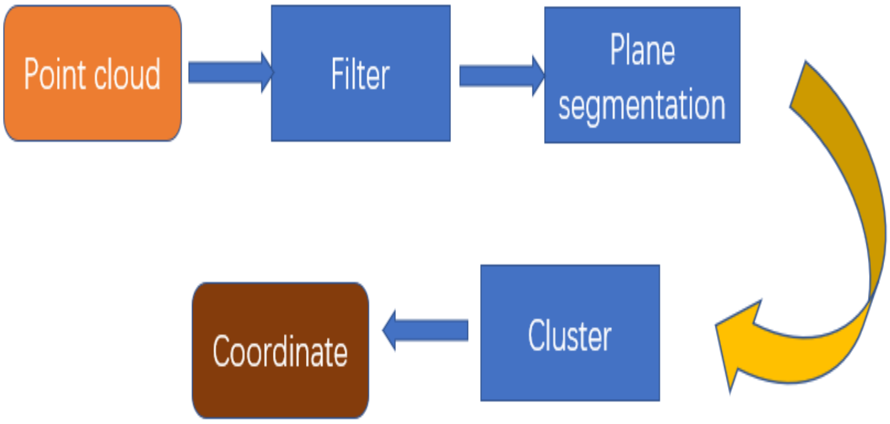



**Figure 2.** LiDAR detection scheme.

## 3. Method

### *3.1. Point Cloud Process*

Because the LiDAR can generate 300000 points cloud data per second, such a large amount of data has a high demand on the computing capacity of vehicles, and because of the impact of equipment accuracy, external environment factors and other factors, the point cloud data we get will inevitably appear noise points. In the actual track, especially on the curve, there will be line of sight occlusion, and there are some outliers in the point cloud. Therefore, we first filter the received point cloud data. On the premise of not destroying the environmental information, we will comprehensively use a variety of filtering algorithms to ensure a small amount of calculation and the effectiveness of the data.

Although the measurement radius of 16 line LiDAR can reach 100 meters, with the increase of distance, the number of points falling on the target will become more and more sparse, which makes it impossible to get the actual shape of the target through clustering, as shown in the figure 3. Therefore, when the distance is large to a certain extent, the point cloud data obtained has no practical significance.

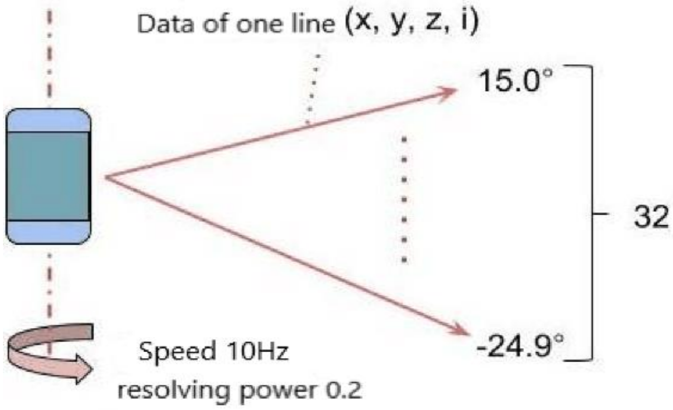



**Figure 3.** LiDAR working principle.

Therefore, similar to the common passes through filter, the point cloud in a certain range along a certain coordinate axis can be removed or retained. We can build a cube V1 with the length, width and height of 10 meters that runs with the car as the origin, and retain the point cloud data in the cube. At the same time, the point cloud data around the car is also useless for us. Let's build a cube V2 with the car as the origin and the length, width and height of 2m. Subtract the point cloud data in V2 from the point cloud data in V1 to get the point cloud we need. This method can greatly reduce the number of point clouds, greatly reduce the amount of calculation, and increase the speed of calculation.

In order to further reduce the amount of point cloud data, we can also use voxel mesh method to achieve filtering. The principle of this method is to create a three-dimensional cube grid from the input point cloud data, and all points in the voxel can be represented by the center of gravity of the cube.

### *3.2. Plane Segmentation*

In the segmentation plane, we use the RANSAC (random sample consensus) method. The fundamental is getting the model we need from a group of point clouds with errors and defects. In our data, figure 4 displays the algorithm. First, we get three points (x, y, z) from the point cloud to fit a plane model, and then judge whether the distance from the fourth point (x4, Y4, Z4) to this plane is less than the tolerance we set. If the distance from the point to the plane is less than the threshold T, then the point is the interior point.

RANSAC has good robustness [12-14]. When there is a big error in our data, we can still estimate a good model. But the number of iterations required is higher, which may take up more time cost.

**Algorithm** :RANSAC for planner segmentation

**Input**: $I_{iteration}$, $T_{threshold}$, $P_{clouds}$

**Output**: $P_{inliers}$

$P_{inliers}$=0;

**while**($I_{iteration}$ − −)

**for**(size_t i= 0;i < $P_{clouds}$.size( );i++)

$(X_1,Y_1,Z_1),(X_2,Y_2,Z_2),(X_3,Y_3,Z_3) \longleftarrow P_{clouds}$;

$P_1 = (X_1 - X_2) * (Y_1 - Y_3) - (Y_2 - Y_1) * (X_3 - X_1)$ ;

$P_2 = (Z_2 - Z_1) * (X_3 - X_1) - (X_2 - X_1) * (Z_3 - Z_1)$ ;

$P_3 = (Y_2 - Y_1) * (Z_3 - Z_1) - (Z_2 - Z_1) * (Y_3 - Y_1)$ ;

$P_d = (P_1 * X_1 + P_2 * Y_1 + P_3 * Z_1)$;

$d = fabs(X_4 * P_1 + Y_4 * P_2 + Z_4 * P_3$;

**if** (d < $T_{threshold}$)

**return** $P_{inliers}$;

**Figure 4.** RANSCA algorithm for planner segmentation.

### *3.3. Clustering*

Through RANSAC we segment the track plane, then we will find the pile bucket on both sides of the track by clustering, and finally fit the available track.

Because our track is composed of cones, there are no other complex obstacles, so we can use Euclidean algorithm to cluster pile bucket. First, take a certain point as the target point, use K-D tree to find the nearest n points, and calculate the distance from these n points to the target point. If the distance is less than the threshold we set, these points will be divided into one class. Repeat this process until no point cloud can be divided to complete clustering.

In order to prove the practicability of this method, we try to test the above method on the open dataset. The results show that our method can also identify vehicles on urban roads. The following figure 5 shows the test results using Baidu dataset. In the box is the vehicle identified by LiDAR. We can see that our method accurately detects two vehicles on the current road.

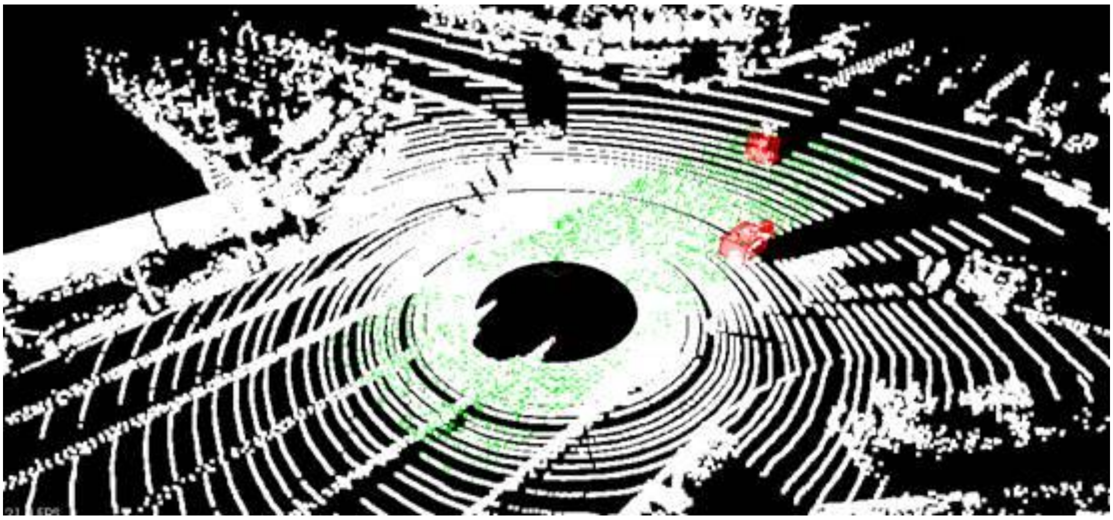


**Figure 5.** Detection result, red box are obstacles.

## 4. Conclusion

In this paper, a method of using LiDAR to detect obstacles is proposed. It can be seen from the experiment that the 16 line LiDAR can quickly and accurately detect the size and location of obstacles when the surrounding environment is not too complex. Compared with the camera, the LiDAR can output more accurate and stable results where the light changes obviously, especially in the corners where the camera has blind field of vision, the LiDAR has more advantages.

## Acknowledgments

This work was supported in part by in part by Research Project of Hubei Provincial Department of Education under Q20181804, Natural Science Foundation of Hubei Province under 2017CFC853, and partially supported by Doctor's Research Fund of HUAT under BK201604.

## References

[1] Chen T T, Dai B, Liu D X, et al. 2011 3D LiDAR based ground segmentation *The First Asian Conference on Pattern Recognition* 446-450

[2] Davison Andrew J, Reid I D, Molton N D and Stasse O 2007 Real-time single camera SLAM *IEEE Transactions on Pattern Analysis and Machine Intelligenc* 1052-1067

[3] Raúl M A, Montiel J M M and Tardós J D 2015 ORB-SLAM: A versatile and accurate monocular SLAM system *IEEE Transactions on Robotics* 1147-1163

[4] Raúl M A and Tardós J D 2017 Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras *IEEE Transactions on Robotics* 1255-1262

[5] Li Q, Zhang L, Mao Q, Zou Q, Zhang P and Feng S W 2014 Ochieng motion field estimation for a dynamic scene using a 3d LiDAR *Sensors* (Basel: Switzerland) **14** 16672-16691
[6] Asvadi A, Premebida C, Peixoto P and Nunes U 2016 3D LiDAR basedstatic and moving obstacle detection in driving environments: An approach based on voxels and multiregion ground planes *Robotics and Autonomos Systems* **83** 299-311
[7] Steinhauser D, Ruepp O and Burschka D 2008 Motion segmentation and scene classification from 3d LiDAR data *2008 IEEE Intelligent Vehicles Symposium* 398-403
[8] Zhang M, Morris D D and Fu R 2015 Ground segmentation based on loopy belief propagation for sparse 3d point clouds *IEEE 2015 International Conference on 3D Vision* 615-622
[9] Ruan X Y and Liu B L 2020 Review of 3d point cloud data segmentation methods *International Journal of Advanced Network Monitoring and Controls* **5** (1) 66-71
[10] Ma J W and Czerniawski T 2020 Semantic segmentation of point clouds of building interiors with deep learning *Augmenting Training Datasets with Synthetic BIM-based Point Clouds* **113**
[11] Chen D, Zhang L Q and Wang Z and Deng H 2013 A mathematical morphology-based multi-level filter of LiDAR data for generating *Information Sciences* **56** 144-157
[12] Chen C F, Li Y Y, Li W and Dai H L2013 A multiresolution hierarchical classification algorithm for filtering airborne LiDAR *ISPRS Journal of Photogrammetry and Remote Sensing* **82** 1-9
[13] Bartels M and Wei H 2010 Threshold-free object and ground point separation in LiDAR data *Pattern Recognition Letter* **30** (10)
[14] Young J and Simic M 2015 LiDAR and monocular based overhanging obstacle detection *Procedia Computer Science* **60** 632-638