Perception and Path Planning Algorithm for FSD Electric Racecar Using 3D LiDAR

Dai Kai
Jilin University

Perception and Path Planning Algorithm for FSD Electric Racecar Using 3D LiDAR

MA Fangwu, DAI Kai[1], AN Jingya[1], WANG Bo[2], WU Liang[1]
1. *State key laboratory of automotive simulation and control Jilin University*
2. *Konghui Automotive Technologies (Changchun) Co., Ltd.*

Abstract: Driverless technology has become a significant focus in the automotive industry. This paper proposes a perception and path planning algorithm for formula student driverless (FSD) electric racecar. The proposed algorithm can identify traffic cones on the track and dynamically conduct real-time path planning. As a result, the FSD electric racecar can achieve self-driving on the track without collision. Features of the track are obtained by the 3D LiDAR and then analyzed through filtering and clustering algorithm in the Point Cloud Library (PCL). More than 90% of the features are filtered out while those of the traffic cones are preserved. By computing the coordinates of the geometric center of the traffic cones, a straight or curved path is generated. Subsequently, the coordinates of the points on the path are transmitted to the decision module through CAN bus. Robot Operating System (ROS) is employed to support data processing and transmission herein. The algorithm has been successfully implemented in the racecar with competitive results in China's first FSD competition in 2017. It indicates that the algorithm possesses the abilities to achieve real-time path planning on the track full of traffic cones with high recognition accuracy and strong environmental adaptability.

Key words: FSD, perception, path planning, 3D LiDAR, traffic cone

Introduction

Formula SAE is a long-running annual competition organized by the Society of Automotive Engineers with competition events held in the U.S., Europe, China, Australia and other countries. In 2017, Germany and China held their first FSD competition, respectively. 15 teams participated in the German FSD competition, and 7 teams participated in its Chinese counterpart. More teams from different countries will participate in the upcoming events of this competition.

FSD racecar is similar to the ground driverless vehicle in terms of hardware and software architecture, with the main difference being the driving conditions. FSD racecar runs on fixed tracks without pedestrians or other dynamic obstacles, which leads to a difference in the perception and path planning algorithm. However, there are many technologies on the FSD racecar that can still be ported to driverless ground vehicles. As the influence of the competition increases, research on this subject will attract more people's attention.

At present, some studies have already been presented on FSD racecar. Zeilinger et al. proposed a hardware and software architecture. Ni et al. focused on the dynamics control and proposed a path following controller which was later verified on the racecar. Drage et al. introduced a drive-by-wire system and a navigation control system for the racecar. Moreover, Koppula et al. proposed a CNN-based End-to-End Controller.

FSD racecar is required to drive on tracks full of traffic cones. Therefore, it is of vital importance for the racecar to recognize the traffic cones and achieve path planning. Yong et al. presented a fusion method to detect cones based on vision and radar sensors' information. Moreover, the stereo camera was also used to detect traffic cones. However, cameras are greatly influenced by lighting conditions, which makes it less robust during operation. LiDAR is often used to detect obstacles due to its high accuracy and strong robustness. In this research, a 3D LiDAR was used to detect cones only. It is less likely to be affected by external interferences and more efficient.

As for path planning, static state lattices is an efficient path planning algorithm in static environments. A path planning and tracking framework were presented by Ji et al. to maintain a collision-free path for autonomous vehicles. Schirmer et al. proposed an efficient path planning algorithm which worked well in the robot, but it depends on built-in maps, which is not allowed in this competition. In this study, we concentrate on the problem of path planning under a track full of traffic cones. Through the computation of the geometric centers coordinate of the cones, a straight or curved path is generated.

The paper is organized as follows. In section 1, the hardware and software architecture of the racecar will be introduced. In section 2, the perception algorithm that determines how to use the 3D LiDAR to recognize traffic cones will be described. In section 3, the path planning algorithm under the track full of cones will be presented. In section 4, the effectiveness of the proposed algorithm will be demonstrated through experiment. In section 5, contributions and highlights of the proposed algorithm will be provided.

1 The Architecture of FSD Racecar

In this section, a detailed description of the racecar's hardware and software architecture will be presented.

1.1 Hardware Architecture of FSD Racecar

As shown in Fig. 1, the racecar is equipped with a Drive-by-wire system, while vehicle control is performed by a dedicated dSpace AutoBox which directly communicates with the actuators over the vehicle CAN bus. LiDAR, camera, and GPS are used in the perception system, while perception and planning are performed by an Industrial Personal Computer (IPC) with low latency and high bandwidth for inter-process communication.

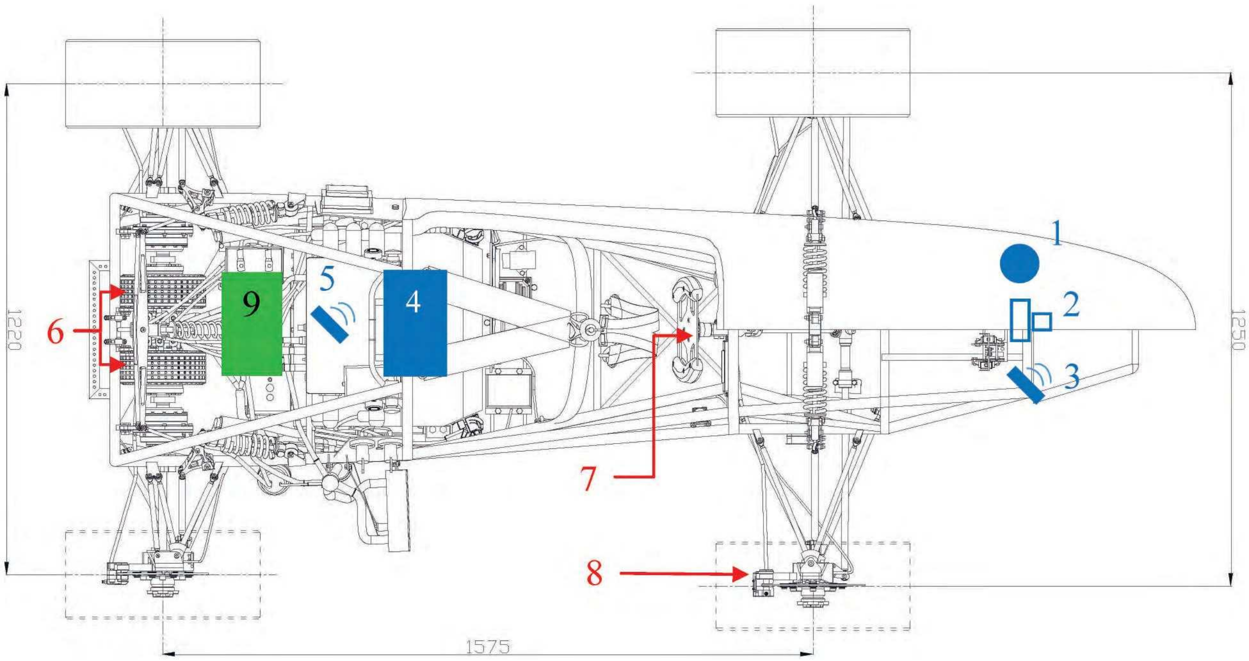

Fig. 1 Hardware Architecture of the FSD Racecar: 1. Velodyne HDL-32E LiDAR, 2. Camera, 3. Front GPS antenna, 4. GPS/INS, 5. Rear GPS antenna, 6. Accelerate-by-wire system, 7. Steer-by-wire system, 8. Brake-by-wire system, 9. dSpace AutoBox

The drive-by-wire system includes the accelerate-by-wire system, the steer-by-wire system, and the brake-by-wire system. Compared with the drive-by-wire system presented by Drage et al., the system is fully electronically controlled with lower latency.

The accelerate-by-wire system consists of two motors and two motor controllers which grants the vehicle with a top speed of 132km/h, and an acceleration time of 2.8 s from 0 to 100km/h. Column electric power steering (C-EPS) is used in the steer-by-wire system, with a rated torque of 2Nm and a transmission ratio of 16.5:1. The brake-by-wire system uses Bosch ESC 9.0 as the hydraulic adjustment unit, with a maximum pressure of 200 bar and a response time of less than 150 ms.

1.2 Software Architecture of FSD Racecar

The software architecture is depicted in Fig. 2. Robot Operation System (ROS) is used to integrate and process sensor data. Point Cloud Library (PCL) is used to process data measured by the LiDAR. Moreover, OpenCV is used to process camera data. Obstacle characteristics can be obtained by the perceptual system and then transmitted to the planning module which generates a driving path. The drive-by-wire system will execute the instructions sent by the decision module to complete the driving process.

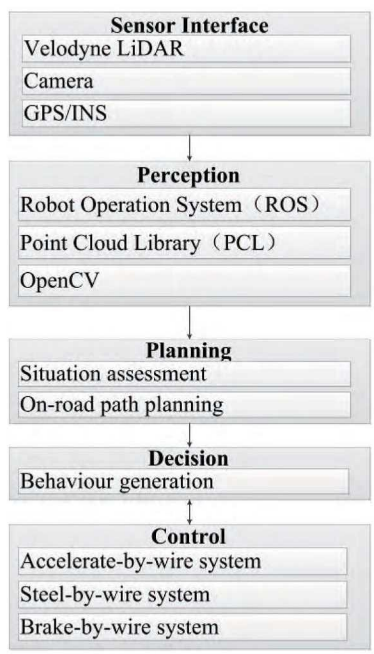

Fig. 2 Software Architecture of the FSD Racecar

A more detailed description of the adopted perception and planning algorithm will be presented in the following section.

2 Perception Algorithm To Recognize Cones

Based on the regulations of FSD, path planning is conducted according to the location of the traffic cones. In this study, a Velodyne HDL-32E LiDAR was used to scan traffic cones, and Point Cloud Library (PCL) is used to process the point clouds generated by the LiDAR. The non-interest regions can be filtered out by using the PassThrough filter in the PCL, and the region where traffic cones located is preserved. Then, the Euclidean clustering algorithm in the PCL is used to cluster the traffic cones. However, some noise will remain after clustering. Therefore, a variance filter is used to filter out the clustered objects with dispersion values higher than the threshold (Fig. 3).

2.1 Filtering Algorithm to Obtain ROI

The PassThrough filter and the variance filter are used in this study, where the PassThrough filter in the PCL is mainly used to obtain the region of interest (ROI). This algorithm can quickly filter out non-interest regions by setting the range of XYZ coordinates, which can significantly reduce the computing resource.

The variance filter calculates the variance of the clustered object. When the variance is greater than a certain threshold, it is considered that the dispersion of this object is too large and does not meet the characteristics of traffic cones and the object is therefore filtered out. Equation (1) and (2) calculate the average value and the variance in the x-direction of the object respectively. The same method can also be used to calculate these values in the y-direction and z-direction. In this way, the accuracy of recognizing the traffic cones is increased.

$$x_{avg} = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{1}$$

$$x_s = \frac{(x_1 - x_{avg})^2 + (x_2 - x_{avg})^2 + \cdots + (x_n - x_{avg})^2}{n} \tag{2}$$

(x_{avg} is the x average value of all points in the x-direction, x_1 is the value of the first point in the x-direction, n represents n points, x_s represents the x-direction variance of the object)

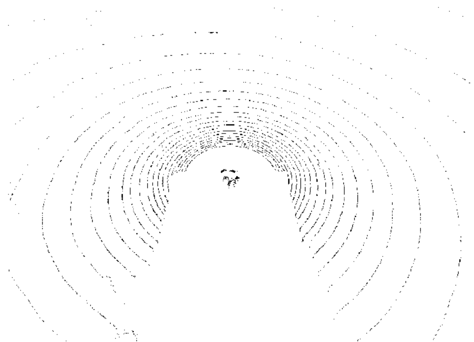

Fig. 3 The raw point cloud data of the track obtained by the Velodyne LiDAR

2.2 Clustering Algorithm to Obtain Coordinates of Cones

Euclidean clustering algorithm in the PCL is used to cluster the traffic cones, and the specific clustering process is as follows (Fig. 4):

Step1: Select k samples from the point cloud
Step2: Traverse all point cloud samples
Step3: Calculate the Euclidean distance between k samples
Step4: Set the parameters of the Euclidean clustering algorithm
Step5: Calculate the centroid for each category
Step6: Get the center coordinates of the clustered objects

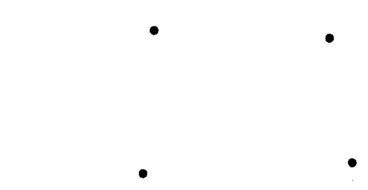

Fig. 4 After the perception algorithm process the raw point cloud data, four traffic cones on the track are identified.

3 Path Planning Algorithm For Competitions

In this study, the path planning was conducted mainly according to the locations of the traffic cones, which was adopted in two events: The acceleration and the skidpad.

3.1 Path Planning of Acceleration Event

The acceleration event is held on a 75m-long, 5m-wide straight track, with 16 traffic cones on each side. There are two general situations during the event:

Situation 1: When the vehicle is driving forward, the LiDAR scans at least three cones, as shown in Fig. 5, in the triangle ABC. By setting the midpoints of AC and AB as D and E respectively, the demanded line can be acquired by connecting D and E. Starting from E, line EF acts as an extension of ED, with a length of 8m. 20 coordinate points are evenly selected on EF, which will also be updated with the moving of the vehicle.

Situation 2: When the vehicle deviates from the track significantly, the LiDAR is only able to scan two cones on one side of the track, as shown in Fig. 6. Point G, H are the coordinates of the identified traffic cones. By shifting the two points to the track center by 2.5m, the demanded line can be acquired as MN. Starting from M, line MP acts as an extension of MN, with a length of 8m. 20 coordinate points are evenly selected on MP, which will also be updated with the moving of the vehicle (Fig. 7).

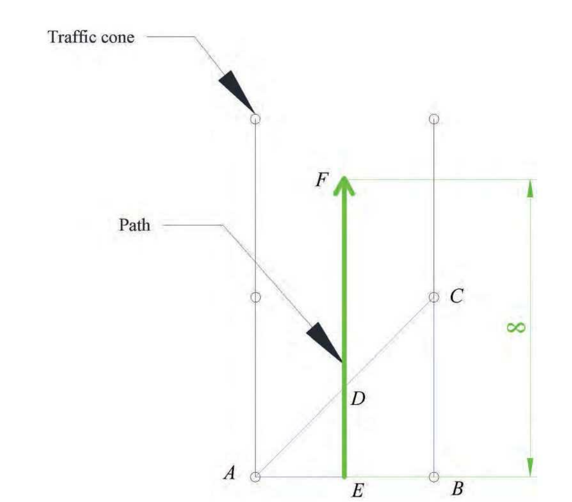

Fig. 5 Acceleration Event Situation 1, LiDAR Scans Three Traffic Cones A、B、C, The Path EF is Planned According to the Position of the Three Traffic Cones.

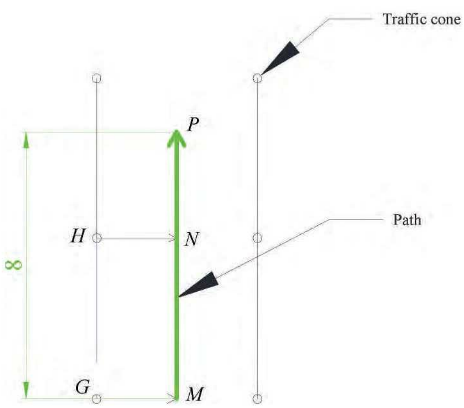

Fig. 6 Acceleration Event Situation 2, LiDAR Scans Two Traffic Cones G、H, The Path MP is Planned According to the Position of the Two Traffic Cones

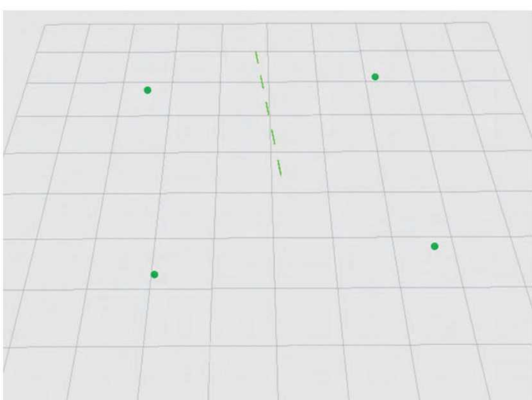

Fig. 7 Path Planning Result of Acceleration Event, Four Green Dots in the Figure are the Identified Traffic Cones, and the Dotted Line is the Planned Path.

3.2 Path Planning of Skidpad Event

The skidpad track consists of two congruent circles, touching externally. The racecar enters the track through a path tangent to the circles at their contact point, covering the right circle twice followed by the left circle twice before leaving via the tangent path again, as shown in Fig. 8.

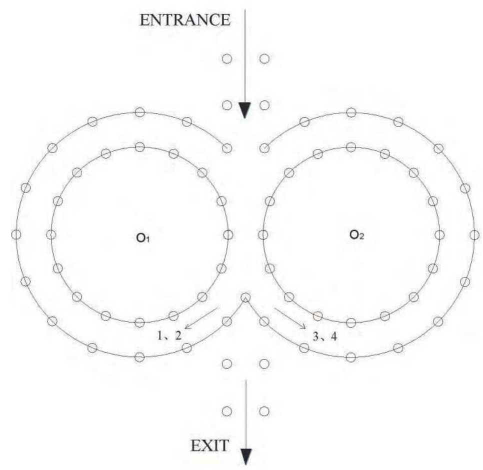

Fig. 8 Skidpad Track

When the vehicle is driving in a circle, the LiDAR can scan at least three cones, and the three points form a triangle. A total of four different types of triangles will be encountered while driving in the skid pad track, as shown in Fig. 9.

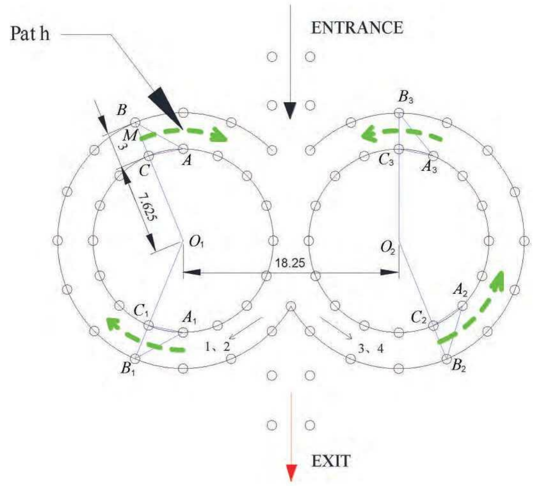

Fig. 9 Path Planning Diagram of Skidpad Event, Determine the Direction of the Curve According to the Position of the Traffic Cones on the Track, and Dynamically Plan the Path

In the triangle ABC, as shown in Fig. 10, there is a line that crosses the C point and is parallel to the x-axis. This line intersects the line AB at point P. If $C_{.x} > P_{.x}$, it is judged that the current curve is a right turn. Similarly, in the triangle $A_1B_1C_1$, $C_{1.x} > P_{1.x}$, it is judged that the current curve is a right turn. On the contrary, as shown in Fig. 11, if $C_{2.x} < P_{2.x}$ or $C_{3.x} < P_{3.x}$, it is judged that the current curve is a left turn.

As shown in Fig. 9, it is known that $BC = 3\text{m}$, $O_1C = 7.625\text{m}$. With the coordinates of point B and C, equation (3) can be used to solve the coordinates of point O_1.

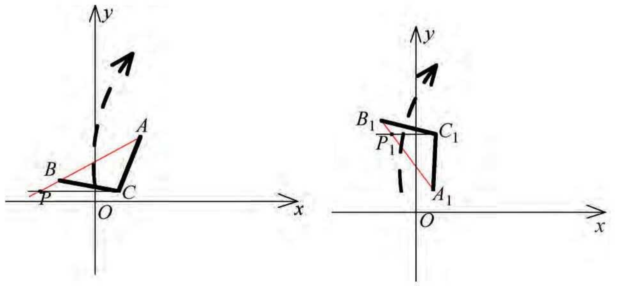

Fig. 10　Using Three Traffic Cones to Judge the Right Turn Condition

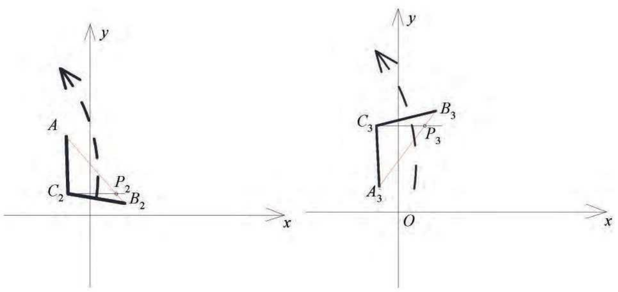

Fig. 11　Using Three Traffic Cones to Judge the Left Turn Condition

$$\frac{BC}{O_1C}=\frac{3}{7.625} \quad (3)$$

M is set as the midpoint of BC and $O_1M=9.125\text{m}$. By using equation (4), the demanded curve can be acquired. Starting from M, a section of the arc is selected on the curve, with a length of 8m. 20 coordinate points are evenly selected on the arc, which will also be updated with the moving of the vehicle.

$$(x-O_{1.x})^2+(y-O_{1.y})^2=O_1M^2 \quad (4)$$

The following selection process is adopted to select the path that matches the requirements of the competition. According to the algorithm as described above, when the vehicle travels to the junction of the left and right circle, there will be two arc paths, as shown in Fig. 12, the right turn path should be chosen for the first and second laps, the left turn path should be chosen for the third and fourth laps (Fig. 13).

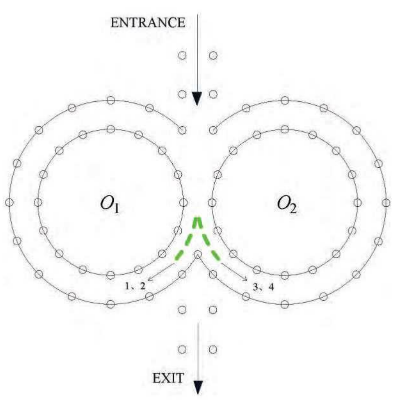

Fig. 12　When the Racecar Drives to the Junction of the Left and Right Circle, There Will be Two Arc Paths

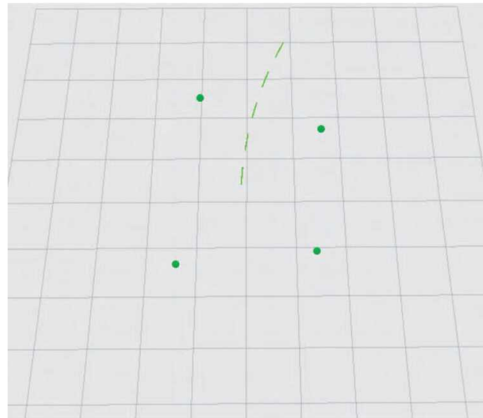

Fig. 13　Path Planning Result of Skid Pad Event, Four Green Dots in the Figure are the Identified Traffic Cones, and the Curved Dotted Line is the Planned Path.

4　Experimental Results

This algorithm was developed with Robot Operating System (ROS) in Ubuntu 14.04 LTS. The system operated on an Industrial Personal Computer (IPC) with Core i7, 3.1GHz, 16GB RAM. The Rviz tool and the Rqt tool were employed in ROS for performance checking, as shown in Fig. 14. Moreover, a PCIe CAN card was embedded in the IPC to communicate with the VCU.

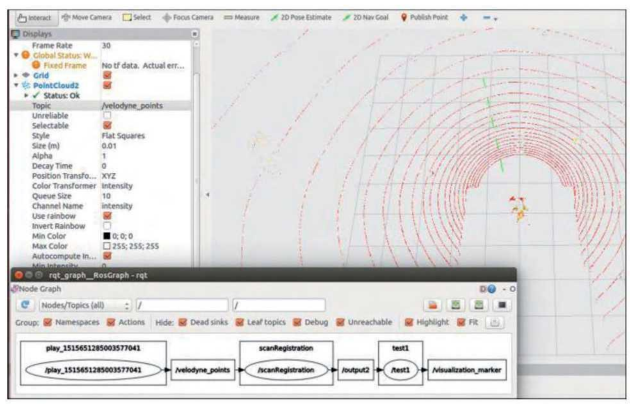

Fig. 14　ROS Interface

The algorithm has been successfully employed in the vehicle with competitive results in China's first FSD competition in 2017 (Fig. 15). It indicates that the algorithm possesses the abilities to achieve real-time path planning on track full of traffic cones with

Fig. 15　The FSD Electric Racecar Developed by the Authors

high recognition accuracy and strong environmental adaptability. Hence, the efficiency and accuracy of the algorithm for driverless vehicles proposed in this article have been verified and demonstrated through practice.

5 Conclusion

The hardware architecture and software architecture was proposed in this paper for the FSD electric vehicle, which will help FSD teams to design and build vehicles quickly. The proposed algorithm for perception and path planning has been validated on the vehicle. This algorithm can realize traffic cone recognition by using 3D LiDAR and achieve path planning based on the coordinates of cones without built-in maps, which can significantly improve the autonomous driving ability of the vehicle on an unfamiliar track.

This work will provide reference and guidance for the application of FSD racecar. This perception and path planning algorithm under certain conditions will contribute to the development of driverless ground vehicles.

References

[1] M. Zeilinger, R. Hauk, M. Bader, and A. Hofmann. Design of an Autonomous Race Car for the Formula Student Driverless (FSD). *Oagm & Arw Joint Workshop*, pp. 57-62, 2017.

[2] C. Stiller and J. Ziegler. 3D Perception and Planning for Self-Driving and Cooperative Automobiles [C]. presented at the International Multi-Conference on Systems, Signals and Devices (SSD 2012) 9th, Chemnitz, Germany, 2012.

[3] JunNi and JibinHu. Dynamics control of autonomous vehicle at driving limits and experiment on an autonomous formula racing car [J]. *Mechanical Systems and Signal Processing*, vol. 90, pp. 154-174, 2017.

[4] J. Ni and J. Hu. Path following control for autonomous formula racecar Autonomous formula student competition [J]. presented at the Intelligent Vehicles Symposium (IV), 2017 IEEE, Los Angeles, CA, USA, 2017.

[5] T. Drage, J. Kalinowski, and T. Braunl. Integration of Drive-by-Wire with Navigation Control for a Driverless Electric Race Car [J]. *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 23-33, 2014.

[6] S. Koppula. Learning a CNN-based End-to-End Controller for a Formula Racecar [J]. *arXiv preprint arXiv: 1708.02215*, 2017.

[7] H. Yong and X. Jianru. Real-time Traffic Cone Detection for Autonomous Vehicle [C]. presented at the Control Conference (CCC), 2015 34th Chinese, Hangzhou, China, 2015.

[8] L. Zhou, H. Wang, D. Wang, L. Xie, and K. P. Tee. Traffic Cone Detection and Localization in TechX Challenge 2013 [C]. presented at the International Conference on Control, Automation, Robotics & Vision 13th, Singapore, 2014.

[9] S. Kuthirummal, A. Das, and S. Samarasekera. A Graph Traversal based Algorithm for Obstacle Detection using Lidar or Stereo [C]. presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, 2011.

[10] M. Pivtoraiko and A. Kelly. Efficient constrained path planning via search in state lattices [C]. presented at the 8th International Symposium on Artifical Intelligence, Robotics and Automation in Space (i-SAIRAS 2005) 8th, Munich, Germany, 2005.

[11] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang. Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints [J]. *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952-964, 2017.

[12] R. Schirmer, P. Biber, and C. Stachniss. Efficient Path Planning in Belief Space for Safe Navigation [C]. presented at the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017.

[13] A. Ng *et al.*. ROS: an open-source Robot Operating System [C]. *Icra Workshop on Open Source Software*, 2009.

[14] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke. Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D [J]. *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 110-124, 2015.

[15] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek. A brief introduction to OpenCV [C]. presented at the 35th International Convention MIPRO 2012, Opatija, Hrvatska, 2012.