

PAPER • OPEN ACCESS

Autonomous Driving System Design for Formula Racing

To cite this article: Zheng Zhang *et al* 2020 *J. Phys.: Conf. Ser.* **1651** 012125

View the [article online](#) for updates and enhancements.

You may also like

- [Analysis on the operation condition of Rubber belt CVT of baja Racing car](#)
Tianle Ai, Siying Xiao and Zhanpeng You
- [A novel model for simulating the racing effect in capillary-driven underfill process in flip chip](#)
Wenhui Zhu, Kanglun Wang and Yan Wang
- [Optimization Design and Virtual Prototype Modeling and Simulation Analysis of FSAE Car Steering Trapezium](#)
Bin Chen, Yunxiang Liu and Wei Shi





The
Electrochemical
Society

Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

Autonomous Driving System Design for Formula Racing

Zheng Zhang^{1a}, Peifeng Gao², Zilong He³, Hongbin Chen⁴ and Jianfeng Wang⁵

¹HRT Formula Team, Harbin Institute of Technology at Weihai, Weihai, Shandong Province, 264200, China

^a170120329@stu.hit.edu.cn *Corresponding author's e-mail: lewisonez@163.com

Abstract. This paper mainly introduces automatic driving schemes and algorithms for formula racing car. Formula racing is a good platform for testing the extreme performance of vehicles, but there is a risk of harming the drivers. We propose an automatic driving scheme to avoid such problems. The solution we proposed integrates three modules: environment perception, path planning and vehicle control. Using lidar, camera and GPS/INS integrated navigation as sensors, an innovative data fusion method is proposed. This paper proposes a new path planning method, which uses Delaunay triangulation and topological map-based path search and evaluation to obtain the best path. Finally, a control model based on pure pursuit algorithm is used to control the vehicle precisely. The whole system is developed based on ROS which has a unique loose coupling. This scheme is proved to have high accuracy and good robustness by simulation analysis and multiple vehicle tests. The proposed solution has good portability, and it can also be applied to the field of unmanned passenger cars, robots and drones after appropriate modifications.

1. Introduction

Formula racing provides a platform for testing the extreme performance and stability of vehicles, and is also a vane for the development of the automotive industry. The automatic formula car provides a new solution for the research and testing of autonomous driving. In certain scenarios, high speed and sharp corner pose a challenge to the stability and accuracy of the autonomous driving system. Therefore, the research on the automatic driving formula car is of great significance to the implementation of automatic driving technology.

This paper introduces an automatic driving scheme for formula racing, covering environment perception, path planning, and vehicle control. The obstacles of formula cars are usually cones of different colors. This solution obtains the position of the obstacle by Euclidean clustering processing on the lidar point cloud data. The TensorFlow-based residual network ResNet is used to process the camera image for target detection. The GPS integrated navigation system based on Kalman filter is used to obtain information such as the position and speed of the car. The data of the three sensors are merged into a unified coordinate system to obtain the relationship between the car and the external environment. The obstacle space is divided into regular triangles by the Delaunay triangulation algorithm, and the cost function is set to evaluate the path according to the topological map generated by each side of the triangle to find the best path. Afterwards, a control method based on pure pursuit algorithm is used to control the front wheel angle and speed of the vehicle to complete automatic driving. The whole system is developed based on ROS which has a specific loose coupling and we can flexibly add various modules according to need.



2. Environment perception and data fusion

2.1. Euclidean clustering of radar point cloud

Neural network and Euclidean clustering are two common methods for processing radar point cloud data. The accuracy of the neural network is slightly higher, but it also takes up more computer resources. Since we will use the camera to make more detailed confirmation of the obstacles after the point cloud processing, the Euclidean clustering processing that takes up less resources has a greater advantage.

The principle of Euclidean clustering is to calculate the distance between two points in the point cloud data, set the points whose distance is less than the set threshold as one class, and iteratively calculate until the distance between all the points outside the class and the points in the class is greater than the threshold. The nearest neighbor query algorithm based on KD-Tree is an important preprocessing method to accelerate the Euclidean clustering algorithm. The algorithm flow is shown in Figure 1.

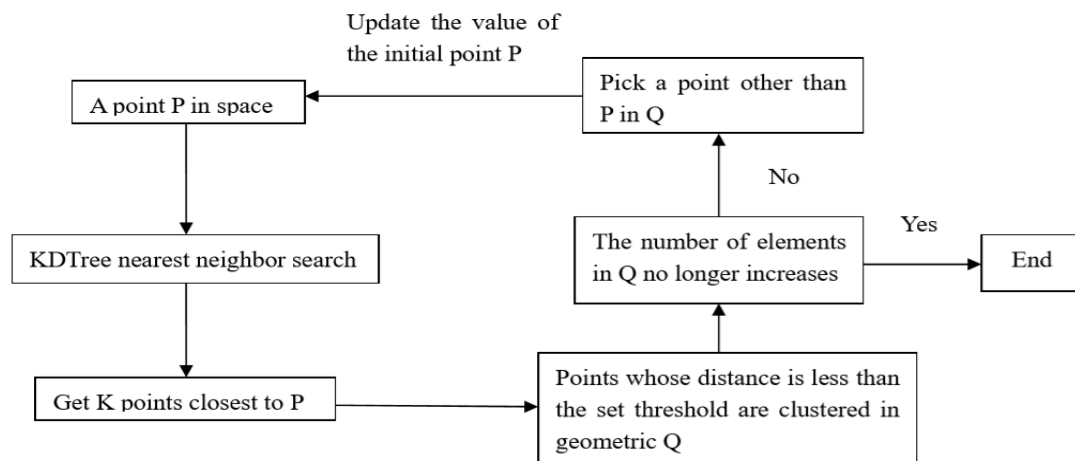


Figure 1. Flow chart of Euclidean clustering principle.

Since Euclidean clustering cannot reflect the geometric characteristics of objects, and it will be interfered by ground points during the clustering process, which increases the difficulty of camera recognition. Thus, it is necessary to remove ground points by random sampling algorithm before Euclidean clustering.

2.2. Target recognition based on deep learning

2.2.1. ResNet principle

Repeatedly superimposing the neural network will not increase the accuracy of the network, but will decrease its accuracy, as shown in Figure 2. Due to the existence of the degradation problem, the output result of too deep network is even worse than that of shallow network. The residual network can solve this problem. The deeper the residual network, the better the effect on the training set.

The residual is the difference between the actual observation and the estimated value (fitting value) in mathematical statistics. If the regression model is correct, we can regard the residual as the error observation. For example, if you want to find an x such that $f(x)=b$, given an estimated value of x x_0 , the residual is $b-f(x_0)$, and the error is $x-x_0$.

ResNet has become easier to optimize by adding shortcut connections. Several layers of networks containing a shortcut connections are called a residual block, as shown in Figure 3.

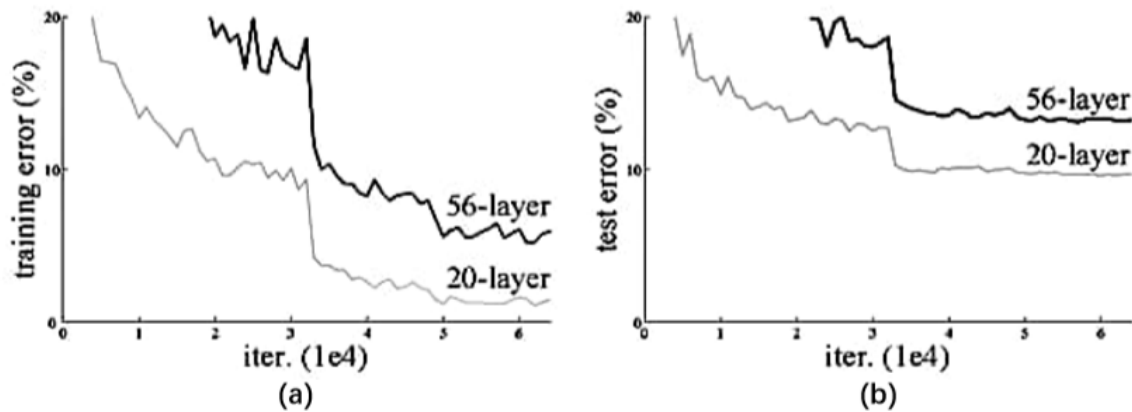


Figure 2. Training error (a) and test error (b) of the "simple" network with 20 and 56 layers on CIFAR-10.

In fact, ResNet is not the first model to use shortcut connections. Highway Networks has introduced gated shortcut connections. These parameterized gates control the amount of information flowing through the shortcut. ResNet can be considered as a special case of Highway Network, so the solution space of Highway Network contains ResNet, so its performance should be at least as good as ResNet. However, in fact, the performance of Highway Network is not better than that of ResNet, which shows that maintaining the smooth flow of these gradient highways is more important than obtaining a larger solution space. According to this idea, through a pre-activated variant of the residual block, the gradient can reach the previous layer through the fast connection in the model without any obstacle, as shown in Figure 4.

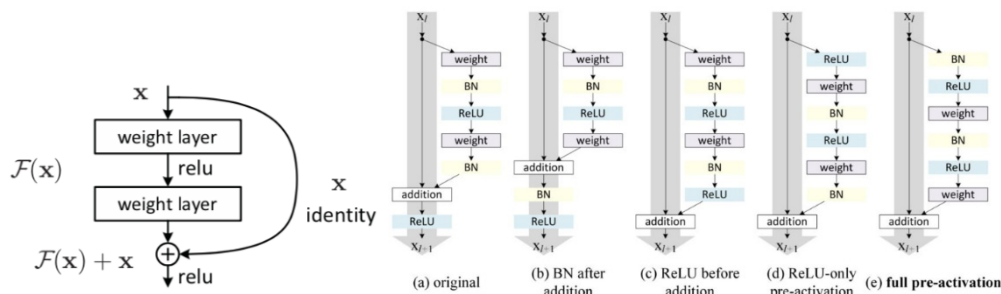


Figure 3. Residual block.

Figure 4. Pre-activated variant of residual block.

The advantage of ResNet is that it is easy to build, convenient to design a network suitable for various visual recognition tasks in various situations, and it overcomes the degradation problem caused by the deepening of the network layer. It greatly improves the recognition accuracy of the neural network, so our driverless formula car chooses ResNet as the vision algorithm.

2.2.2 Neural network training and detection

We use a large number of obstacle pictures in different environmental backgrounds and different weather conditions, and expand the training data by rotating, transforming, and distorting the images. Classify them according to their characteristics to complete the production of the data set. After training, we get the weight of ResNet. The neural network reads this weight and recognizes the input image, determines the color of the obstacle and determines whether it is a cone. We verify the network structure and the weights obtained, and the results show that the recognition accuracy rate reaches 96%, which meets our design requirements.

2.3. Sensor data fusion

2.3.1 Joint calibration of radar and camera

Regarding the joint calibration of radar and camera, it is mainly divided into two parts, internal parameter determination and external parameter determination. The external parameter calibration needs to use the camera internal parameters. The camera internal parameters are given by the projection matrix and distortion parameters. We use the MATLAB camera calibration toolbox to calculate them.

When an object is imaged on a picture, the object in the radar coordinate system in the real world is first rotated and translated, then transferred to the camera's 3D coordinate system, and then the object is projected onto the picture through the camera's internal parameters. Equation (1) is the rotation matrix and the translation matrix.

$$\begin{bmatrix} r_1 & r_2 & r_3 & x_t \\ r_4 & r_5 & r_6 & y_t \\ r_7 & r_8 & r_9 & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \quad (1)$$

After introducing the distortion model to correct the entire imaging process, it can be described by equation (2).

$$\begin{cases} r_2 = u^2 + v^2 \\ u' = u(1 + k_1 r^2 + k_2 r^4) + [2p_1 uv + p_2(r^2 + 2u^2)] \\ v' = v(1 + k_1 r^2 + k_2 r^4) + [p_1(r^2 + 2v^2) + 2p_2 uv] \end{cases} \quad (2)$$

According to the pinhole model and distortion model of the camera, the radar point cloud is projected onto the picture, and the external camera parameters can be obtained by observing the overlap of the point cloud on the picture with the background and continuously adjusting the parameters in the rotation and translation matrix. After testing, the external parameters meet the accuracy requirements.

2.3.2. Joint calibration of radar and GPS integrated navigation

Due to the difference in the installation positions of the two sensors, the translation and rotation matrices in equation (1) are used to unify the information of the two into a coordinate system. However, because the orientation of the two sensors has a spatial angle w , the position error of obstacles will accumulate regularly over time due to the existence of the angle w during the racing process, as shown in Figure 6. We propose a method of optimization calculation based on MATLAB. We record the location information of obstacles during driving, and establish an error evaluation function based on the least square method. After that, we will substitute w into the evaluation function and calculate within a certain angle range. When the accumulation of position error is the smallest, w at this time can be considered as the correct spatial angle.

3. Path planning

3.1. Delaunay triangulation and path evaluation

First, we use the triangulation algorithm to discretize the obstacle space. Then, the tree of possible paths is grown through discretization features. Finally, all paths and track boundaries are evaluated and calculated through the cost function, and the path with the lowest cost is selected. Due to the limited range of the sensors, we only plan a certain length of path each time, and update the path according to the new obstacle observation results and the frequency of sensor data fusion. In order to avoid wasting computer resources, we trim the paths that obviously do not meet the requirements when looking for a path. The planning result is shown in Figure 6.

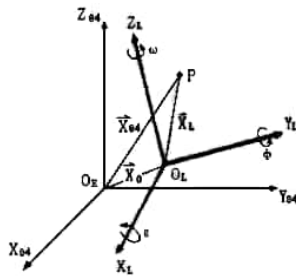


Figure 5. The space angle between two coordinate systems.

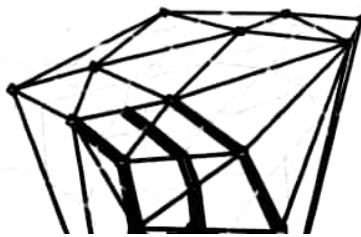


Figure 6. Delaunay Triangulation.

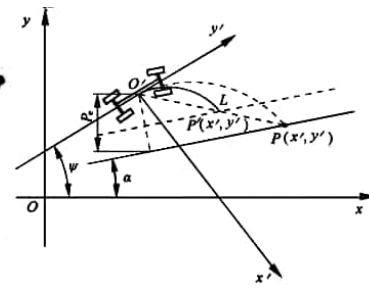


Figure 7. Geometric analysis of pure pursuit model.

3.2. Bowyer-Watson algorithm

First, construct a super triangle, including all the scattered points, and put it into the triangle linked list. After that, the scattered points in the point set are sequentially inserted, and the triangle whose circumcircle contains the insertion point is found in the triangle linked list, which is called the influence triangle of the point. Delete the common edge of the influence triangle, and connect the insertion point with all the vertices of the influence triangle, thus the insertion of a point in the Delaunay triangle linked list is completed. According to the optimization criterion, the newly formed triangle is optimized. Put the formed triangle into the Delaunay triangle list. Repeat until all the scattered points are inserted. This is the simplest and most widely used algorithm for Delaunay triangulation.

3.3. Cost function design

In order to guide the algorithm to find the best path, we designed multiple cost items and added constraints such as vehicle kinematics and track rules. The five cost items are normalized, squared, weighted and finally added to get the path cost. Then we obtain the path and its corresponding track boundary estimate. The path estimation algorithm based on this cost function has been extensively tested under various track layout conditions. The test results show that only when the car deviates from the real track seriously will there be 4.3% prediction errors. The result clearly shows the robustness of the algorithm.

4. Control model based on pure pursuit algorithm

Pure pursuit algorithm is a calculation method based on geometric principles. It can be used to calculate the arc trajectory that the vehicle travels from the current position to the target point in motion. The principle is shown in Figure 7.

$$\theta = \arctan \frac{2H(P_e \cos \psi_e - \sqrt{L^2 - P_e^2} \sin \psi_e)}{L^2} \quad (3)$$

After calculation, the front wheel angle of the pure pursuit model is shown in equation (3). It shows that when the vehicle deviates from the planned path by a certain distance, the pure pursuit method determines the next expected steering angle through the current lateral deviation, the current heading angle and the forward-looking distance. From the derivation of the pure pursuit model, the forward looking distance L is related to the vehicle speed v , the control period T , the heading error and the position error, and the vehicle body speed v and the control period T have the greatest influence.

5. Conclusion

In this paper, we detailed the design scheme of the self-driving formula car and provided a new method for research and testing of unmanned driving. In the environmental perception part, we use radar, camera and GPS integrated navigation for data fusion; in the path planning part, we obtain a

reasonable path through Delaunay triangulation and path evaluation; in the control part, we use the control model based on pure pursuit algorithm to control the vehicle accurately. Simulation analysis and multiple vehicle tests have confirmed the feasibility of the scheme. Based on the current results, afterwards, we will focus on more accurate sensor data matching, reduce the delay of the perception system, and determine more accurate relationship between forward-looking distance and vehicle speed in the pure pursuit model.

Acknowledgments

We would like to thank our colleagues for completing the chassis and electrical work and providing us with a test platform. At the same time, we thank Teacher Wang for providing us with research opportunities.

References

- [1] Hu, F., Hu, C. S., Wang, X. S., & Jiao, H. W. (2011). Calibration of external relation between imaging laser radar and camera. *Opt. Precis. Eng*, 19, pp.938-943.
- [2] Wu, Z., Shen, C., & Van Den Hengel, A. (2019). Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90, pp.119-133.
- [3] Wang, J., Shi, F., Zhang, J., & Liu, Y. (2008). A new calibration model of camera lens distortion. *Pattern recognition*, 41(2), pp.607-615.
- [4] Thompson, J.N. (1984) Insect Diversity and the Trophic Structure of Communities. In: *Ecological Entomology*. New York. pp. 165-178.
- [5] Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., & Frenkel, A. (2011, May). On the segmentation of 3D LIDAR point clouds. In: *2011 IEEE International Conference on Robotics and Automation* . pp. 2798-2805.
- [6] Oh, J., Kim, K. S., Park, M., & Kim, S. (2018). A comparative study on camera-radar calibration methods. In: *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)* .pp. 1057-1062.
- [7] Park, M. W., Lee, S. W., & Han, W. Y. (2014). Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm. In: *14th International Conference on Control, Automation and Systems (ICCAS 2014)* . pp. 1443-1447.
- [8] Gao, D., Duan, J., Yang, X., & Zheng, B. (2010, July). A method of spatial calibration for camera and radar. In: *2010 8th World Congress on Intelligent Control and Automation* .pp. 6211-6215.
- [9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In: *European conference on computer vision*. Springer, Cham. pp. 630-645
- [10] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770-778.
- [11] Chen, L., & Xu, J. C. (2004). Optimal delaunay triangulations. In: *Journal of Computational Mathematics*.pp.299-308.