Fast and Accurate: The Perception System of a Formula Student Driverless Car

HaiLong Gong*
Beijing Institute of Technology
Beijing, China
1120180473@bit.edu.cn

YunJi Feng*
Beijing Institute of Technology
Beijing, China
1120192186@bit.edu.cn

TaiRan Chen*
Nanyang Technological University
Singapore, Singapore
tairanchen@outlook.com

ZuoOu Li
Beijing Institute of Technology
Beijing, China
1120190808@bit.edu.cn

YunWei Li
Beijing Institute of Technology
Beijing, China
1120193558@bit.edu.cn

2022 6th International Conference on Robotics, Control and Automation (ICRCA) | 978-1-6654-8174-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRCA55033.2022.9828892

***Abstract*—To detect the cone of the Formula Student Autonomous China (FSAC) track quickly and accurately, this paper proposed a fast and accurate perception system that fused the 3D LIDAR point cloud data and visual information. First, the vision part spliced the images of the two monocular cameras vertically and adopted the YOLOv3 target detection network with the attention mechanism to obtain the type of candidate targets and their position on the pixel plane. Afterward, the track boundary could be extracted and the drivable area could be segmented using cubic polynomial regression by the center of the bounding box. The LIDAR pipeline used the GPF algorithm and the Euclidean clustering algorithm to process the original point cloud to obtain the three-dimensional information of the cones' coordinates. Finally, the 3D LIDAR point cloud data of reconstructed cones was projected onto the two-dimensional pixel plane and whether the target was detected was determined by judging whether the projected point was in a specific position within the bounding box. Experimental results showed that the proposed method had a larger perception range, faster detection speed, and higher detection accuracy compared with the 2020 season's perception system.**

Keywords—Formula Student Autonomous China (FSAC), object detection, LIDAR cluster, ground segmentation, data fusion)

I. INTRODUCTION

The perception system is one of the most crucial parts of a self-driving system. To improve the accuracy of perception and obtain more environmental information, we always integrate multiple algorithms like object detection and semantic segmentation. Meanwhile, we will combine a variety of sensors like LIDAR and cameras to achieve accurate perception.

In November 2020, the Beijing Institute of Technology formula student driverless team won the championship in Formula Student Autonomous China[1] (hereinafter referred to as FSAC) 2020. The racing cars need to pursue faster lap speed and shorter lap time to achieve better results in this competition. However, we would encounter some difficulties when facing unknown tracks.

On the one hand, the field of view of the monocular camera was limited and the detection speed was determined by algorithms, which increased the difficulty of planning and control [2]. If the turning point in the track was not detected in time, the vehicle would not make a motion plan correctly, thus rushing out of the track and leaving the mission uncompleted. On the other hand, the high-speed movement of the car would lead to a large drift between sensor data of different timestamps, thus directly affecting the data fusion.

Therefore, to achieve high-speed, accurate, and robust detection, we proposed a new binocular perception system. The contributions of this paper would be:

- A fast and accurate cone detection method based on YOLOv3 with attention module and vision-based boundary detection for track cones.
- LIDAR pipeline including fast ground segmentation, point cloud cluster, and cones filter based on cones' shapes.
- A data fusion method that suits the FSAC track.
- FSACOCO, a cone image dataset for FSAC scenarios.

The remainder of this paper was structured as follows. Section II described the cone detection, point cloud cluster, and data fusion. The experimental results and conclusion were presented in Section III and Section IV respectively.

II. METHODOLOGY

In this section, we introduced the perception system based on LIDAR and camera in part 1 and part 2 respectively and our data fusion method after receiving cone positions from both camera and LIDAR pipelines. The architecture of our perception system was shown in Fig. 1.

* The authors contributed equally to this work.

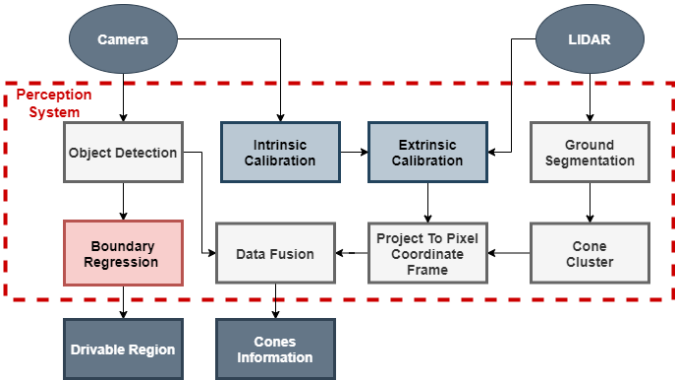


Fig. 1. The architecture of the perception system

A. Binocular Camera Pipeline

1) YOLOv3-Based Cone Detection

Considering the limited field of view (hereinafter referred to as FOV) of a monocular camera, we applied a dual-camera system for cone detection based on YOLOv3[3]. Two cameras shot the same track scene from different viewpoints and scales of cones were quite small compared to background noises. Inspired by [4], we stitched two images from two cameras vertically. Afterward, we sent the stitched image to the detector to get the bounding boxes and classification results. Given that most of the scenes captured by cameras were similar, we trained a single detector for the integrated image through which we could get broader FOV and more accurate detection without increasing the amounts of deep network parameters.

In addition, the colors and edges of cones were quite distinguished against the background, hence we intended to add attention modules[5] to our YOLOv3 model to highlight useful features and suppress the relatively useless ones.

The locations and colors of cones were specified in FSAC, hence we applied the channel attention module and spatial attention module inspired by Convolutional Block Attention Module (CBAM) [6] in order to capture the visual structure better. Channel attention module was constructed after max-pooling and average-pooling, and features would go to a shared fully-connected layer and be summed element-wisely. After a sigmoid layer, we would get the channel attention map. To construct the spatial attention map, we concatenated the results of max-pooling and average-pooling and sent them to a convolutional layer and a sigmoid layer. Moreover, experiments showed that this module is effective in cone detection tasks (FSACOCO).

2) Vision-Based Boundary Regression and Drivable Lane Segmentation

Given the characteristics of the track during the FSAC process, we decided to use three polynomials to regress the center of the bounding boxes (generated by vision cone detection) by minimizing squares. We could predict the position of the invisible cone and at the same time segment the drivable region through the curve calculated by the regression equation.

$$y(x,w) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \cdots + w_m x^m = \sum_{j=0}^{m} w_j x^j \quad (1)$$

The fitted polynomial was evaluated using the mean square error as an error function:

$$E(W) = \frac{1}{2}\sum_{i=1}^{N}(y(x_i,W) - t_n)^2 = \frac{1}{2}(XW-T)^T(XW-T) \quad (2)$$

$$W = (w_0, w_1, w_2, \cdots, w_m)^T \quad (3)$$

$$X = \begin{pmatrix} 1 & x_1 & \cdots & x_1^{m-1} & x_1^m \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^{m-1} & x_n^m \end{pmatrix} \quad (4)$$

Since the error function was a secondary function of the polynomial coefficient W, there was a unique minimum value obtained when the derivative was zero.

$$\frac{\partial E(W)}{\partial W} = X^T X W - X^T T = 0 \quad (5)$$

$$W = (X^T X)^{-1} X^T T \quad (6)$$

Considering the reduction coefficient for image processing and to avoid excessive polynomial coefficient matrix values, we could rewrite the error function and result W.

$$\tilde{E}(W) = \frac{1}{2}\sum_{i=1}^{N}(y(x_i,W) - t_n)^2 + \frac{\lambda}{2}\|W\|^2 = \frac{1}{2}(XW-T)^T(XW-T) + \frac{\lambda}{2}\|W\|^2 \quad (7)$$

$$W = (X^T X + \lambda I_{m+1})^{-1} X^T T \quad (8)$$

$$W_{ori} = [A^T X^T X A]^{-1} A^T X^T (\beta T) = \beta A^{-1} W \quad (9)$$

$A = diag(1, \alpha, \alpha^2, \ldots, \alpha^m)$, $\alpha = \frac{x_{ori}}{x}, \beta = \frac{t_{ori}}{t}$ were the parameters related to image scaling.

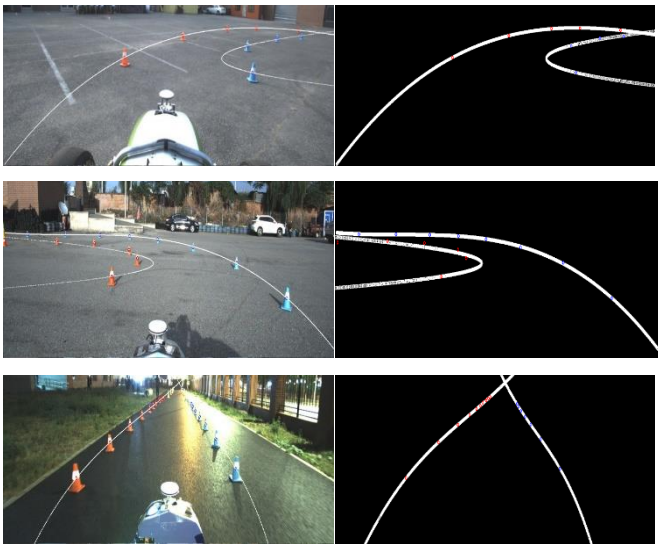


Fig. 2. Cone track boundary regression and drivable region segmentation on FSACOCO

B. LIDAR Cone Detection

1) Fast ground segmentation

LIDAR could get 3D information of the track, allowing the perception system to get the distance between the vehicle and the cones. Inspired by the[7], our goal was to realize a fast and robust ground segmentation and point clouds cluster algorithm to obtain a high-definition map of the track.

The fast ground segmentation algorithm consisted of six parts, i.e. the original point cloud, the point cloud filter, the initial seeds, the plane estimation, and the point cloud classification. The analysis of the point cloud of each frame combined with the track characteristics made it clear that the lowest point was more likely to be a point on the ground surface. Therefore, we considered the entire track map and the height of the LIDAR on the vehicle to set the height threshold Th_{height}. We set up the Th_{height} and combined it with Th_{height} to compute the initial seeds to estimate the plane model.

We used the simple linear model to estimate the ground:

$$ax + by + cz + d = 0 \tag{10}$$

$$\boldsymbol{n} \cdot \boldsymbol{p} = -d \tag{11}$$

a, b, c were constants and $\boldsymbol{n} = [a, b, c]$, $\boldsymbol{p} = [x, y, z]^T$ and x, y, z represented the coordinates of the points.

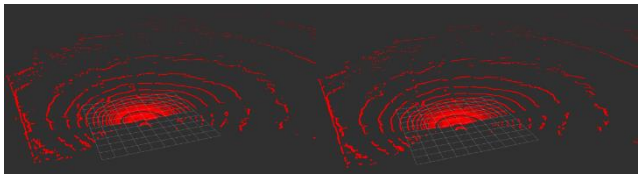
Singular vectors were calculated using singular value decomposition (SVD) in this paper, which represented the main distribution of the track's po int clouds. Among the calculated singular vectors, the direction of the minimum variance was the normal vector perpendicular to the ground. We could solve the normal vector n through the covariance matrix C by calculating the singular vector corresponding to the minimum singular value.

$$C = \sum_{i=1}^{|S|} (S_i - \hat{S})(\hat{S} - S_i)^T \tag{12}$$

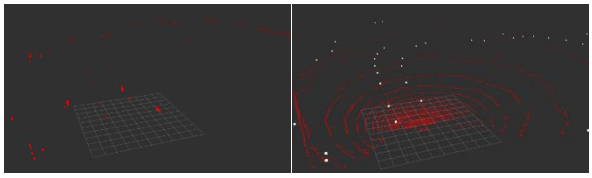
$|S|$, $\hat{S} \in R^3$, $S_i \in S$ represented the total number, the mean value of all points, and the i-th point in the point set respectively.

2) Cone cluster

The point set P_{free} was obtained after the ground points P_{grd} were extracted from the original points. P_{free} still had noises from obstacles different from the cones around the racecar such as vehicles, pedestrians, railings, etc. To reduce these obstacles' interference and get a more accurate track map, we divided the cone clustering into two steps, i.e. Euclidean clustering and shape filtering, as shown in Fig. 3.



(a) Raw point cloud (b) Ground point cloud



(c) Non-ground point cloud (d) Cones clustering

Fig. 3. Ground segmentation and cone clustering renderings

The standardized cone of the FSAC track was 30cm high and 20cm long and wide. For each cluster after Euclidean clustering, we calculated its centroid and filtered out the clusters that met the track rules according to the centroids and length of each side.

TABLE I. TRACK IDENTIFICATION CONE INFORMATION TABLE

Cone type			
Cone shape	20 x 20 x 30	20 x 20 x 30	20 x 20 x 30
Cone color	Red	Blue	Yellow

C. Data Fusion

The accuracy of object detection was improved by fusing the detection results of LIDAR and camera. We used the method in [8] to calibrate to obtain the transform matrix between the lidar coordinate frame and the camera coordinate frame and applied[9] to make the camera intrinsic calibration.

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & u_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = KPI \tag{13}$$

Utilizing the coordinate transformation matrix $[\boldsymbol{R}_{cl} \quad |\boldsymbol{t}_{cl}]$, we projected the 3D points to the camera coordinate frame and the pixel coordinate frame through the camera intrinsic matrix K.

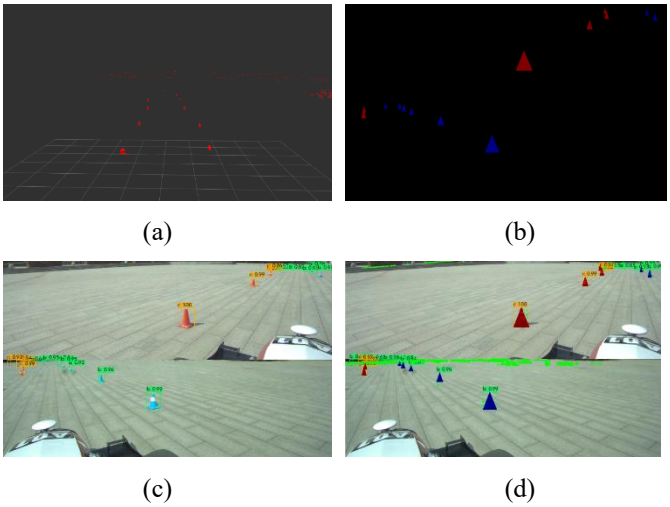
$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \boldsymbol{R}_{cl} \begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix} + \boldsymbol{t}_{cl} = [\boldsymbol{R_{cl}} \quad | \boldsymbol{t_{cl}}] \begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix} \tag{14}$$

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & u_0 \\ 0 & 0 & 1 \end{bmatrix} [\boldsymbol{R_{cl}} \quad |\boldsymbol{t_{cl}}] \begin{bmatrix} x_l \\ y_l \\ z_l \\ 1 \end{bmatrix} \tag{15}$$

$\boldsymbol{R}_{cl}$ was the rotation matrix and $\boldsymbol{t}_{cl}$ was the translation matrix.

It was difficult to judge the color of the cones by the intensity of points due to the size limitation of the cones and the influence of the track characteristics. As shown in Fig. 4(d), we projected the reconstructed cones' points into the pixel coordinate frame and combined the class provided by the visual object detection. We would consider the cone to be detected if one of its points was located in the triangle (whose upper vertex was the midpoint of the bounding box and the bottom edge coincided with the

bounding box's) inside the bounding box. The process of data fusion could be seen in Fig. 4.



(a) (b)

(c) (d)

Fig. 4. Camera and LIDAR data fusion renderings

III. Dataset and Experiment

A. FSACOCO Dataset

FSACOCO is the first open-source cone detection dataset for FSAC events in China. To facilitate each racing team to train their cone detection models and improve the perception level of a racecar in this event, we collected and annotated the standard track cone images. FSACOCO contains two object categories — red cones and blue cones, 56696 labeled objects, and 8399 pictures. For each object, we labeled it in the YOLO format which included class_index, mid_x, mid_y, width, and height representing the class, midpoint coordinates, width, and height of the ground truth bounding box. So far, more than seven racing teams have benefited from FSACOCO and participated in the promotion of this dataset.

TABLE II. Summary of the Cones Label in FSACOCO Dataset

Dataset	Red	Blue	Yellow	Total
FSACOCO	23226	24680	8202	56696

B. Experiments

In the following experiments, we made the evaluation and tested the visual cone detection on FSACOCO. Moreover, we tested the LIDAR pipeline in the racetrack scene.

1) Vision cone detection

a) Train details

All of our models were trained on FSACOCO, which was split into training and testing set by a ratio of 7:3. In our experiments, we used an industrial personal computer configured with Intel i7-8700 processors and an NVIDIA Tesla T4 GPU. The model of LIDAR was Hesai 40M and the model of our camera was Basler ace. All the sensors were fitted to the specific locations on the racecar and the LIDAR and the camera were placed in the front of the racecar and the main ring respectively. The horizontal distance between the center of the two cameras was 1.880 m and the vertical distance was 0.787 m.

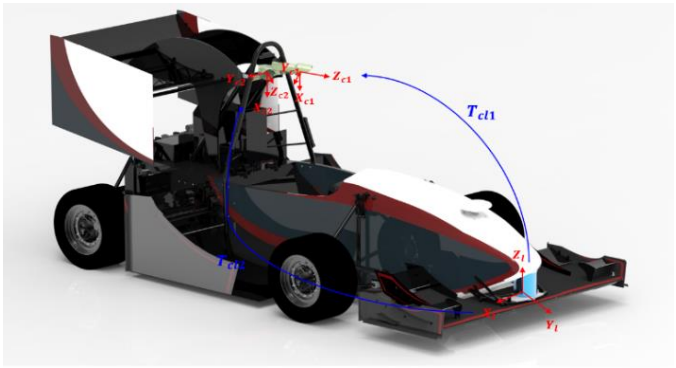


Fig. 5. The perception system on the "Smart Shark V" of Beijing Institute of Technology, Camera and LIDAR coordinate frames, and their conversion relationships

b) Results analysis

To investigate whether the attention mechanism could motivate the performance of vision cone detection, we conducted experiments that later proved our thoughts. In this experiment, we trained YOLOv3 in different configurations, the results showed in Table III: The YOLOv3 combining with CBAM module performances better in learning where and what to emphasize or suppress and refines intermediate features, with 93.3% precision, 87.5% recall and 89.8% mAP50 on FSACOCO. Finally, we applied this module in our vision cone detection system.

TABLE III. Summary of Each Model's Performance on FSACOCO

Algorithm	Precision	Recall	mAP50
YOLOv3[1]	92.1%	84.1%	87.5%
YOLOv3-tiny	94.3%	71.4%	75.7%
YOLOv3-spp	92.5%	87.3%	88.4%
YOLOv3 + CBAM	**93.3%**	**87.5%**	**89.8%**
YOLOv3-spp + CBAM	92.8%	86.7%	89.2%

2) Cone cluster

To explore the general performance of our cone cluster algorithm in terms of speed and accuracy, we conducted a modular test on cone clustering in the racetrack scene. Compared with the combination of RANSAC and Euclidean clustering, our algorithm performed faster and more accurately.

[1] See https://github.com/ultralytics/yolov3

Using the VoxelGrid filter in PCL to adjust the size of the point cloud input into the module, we got the relationship between the point cloud size and cost time, as shown in Table IV.

TABLE IV. PERFORMANCE COMPARISON OF TWO CONE CLUSTERING ALGORITHM

Point sizes	Average time (ms)		Accuracy (%)	
	RnEC*	**Ours**	RnEC*	**Ours**
2000	23	18	70.0	85.3
2500	28	**21**	68.4	**86.8**
3000	35	24	67.2	86.0
3500	42	29	72.3	88.1
4000	51	35	71.8	89.2

(RnEC* stands for the combination of RANSAC and Euclidean clustering).

3) The perception system's performance

This experiment aimed to investigate the general performance of the entire perception system. We test our binocular perception system on the main three missions (Acceleration, Skidpad and AutoCross) in FSAC. Compared with the monocular perception system of the 2020 competition season, the object detection speed of the system proposed in this paper was increased by 50%, the range was expanded by 30%, and the power consumption was reduced by 66%. As shown in Table V, in the acceleration mission, the overall time-consuming of the sensing system was shortened to 18 ms from 23ms and the accuracy reached 85.3%. In Skidpad mission, the accuracy was greatly improved from 24% to 64.0%. Moreover, the new system performed better in the autocross mission with an accuracy of 66.6%.

TABLE V. PERFORMANCE COMPARISON OF PERCEPTION SYSTEM IN 2020 SEASON AND 2021 SEASON

Mission	Average detection time (ms)		Accuracy (%)	
	2020	**2021**	2020	**2021**
Acceleration	23	**18**	70	**85.3**
Skidpad	24	**21**	24	**64.0**
AutoCross	30	**26**	25	**66.6**

IV. CONCLUSION

This paper introduced a new perception system to detect the track in FSAC, which was fast, accurate, and robust. The vision cone detection including the attention mechanism could effectively detect cones of the FSAC track and perform continuous semantic segmentation of the track. Meanwhile, the cone clustering in the LIDAR pipeline used the GPF algorithm and cones filter based on the shape of cones to provide 3D coordinate information for the fusion module, which output the final coordinates for the motion planning module. We hope that our work will inspire the racing teams in FSAC and contribute to their development in perception systems.

REFERENCES

[1] Hanqing & Tian, Jun & Ni, JiBin & Hu. (2018). Autonomous driving system design for formula student driverless racecar. 1-6.

[2] Chen Tairan,Gao Xinyu,Huang Chenrui,Li Xiang,Yang Shaokun,Gong Hailong,Feng Yunji. Real-time Motion Planning and Control for a Formula Student Driverless Car. Proceedings of China SAE Congress 2020: Selected Papers. 203-219.

[3] Redmon, Joseph & Farhadi, Ali. (2018). YOLOv3: An Incremental Improvement.

[4] Chen Y , Zhang P , Li Z , et al. Dynamic Scale Training for Object Detection[J]. 2020.

[5] Mnih, Volodymyr & Heess, Nicolas & Graves, Alex & Kavukcuoglu, Koray. (2014). Recurrent Models of Visual Attention. Advances in Neural Information Processing Systems. 3.

[6] Woo, Sanghyun & Park, JongChan & Lee, Joon-Young & Kweon, Inso. (2018). CBAM: Convolutional Block Attention Module.

[7] D. Zermas, I. Izzat and N. Papanikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications," 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 5067-5073, doi: 10.1109/ICRA.2017.7989591.

[8] Dhall, Ankit & Chelani, Kunal & Radhakrishnan, Vishnu & Krishna, K.. (2017). LiDAR-Camera Calibration using 3D-3D Point correspondences.

[9] Zhang, Zhengyou. (2000). A Flexible New Technique for Camera Calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 22. 1330 - 1334. 10.1109/34.888718.

[10] Wang, Fei & Jiang, Mengqing & Qian, Chen & Yang, Shuo & Li, Cheng & Zhang, Honggang & Wang, Xiaogang & Tang, Xiaoou. (2017). Residual Attention Network for Image Classification. 6450-6458. 10.1109/CVPR.2017.683.

[11] Ba, Jimmy & Mnih, Volodymyr & Kavukcuoglu, Koray. (2014). Multiple Object Recognition with Visual Attention.

[12] Wang, Fei & Jiang, Mengqing & Qian, Chen & Yang, Shuo & Li, Cheng & Zhang, Honggang & Wang, Xiaogang & Tang, Xiaoou. (2017). Residual Attention Network for Image Classification. 6450-6458. 10.1109/CVPR.2017.683.