Research on Autonomous Driving Simulator Control and Decision Algorithms based on Computer Vision Methods

Bolin Zhou1, a, Jihu Zheng1, b, Pei Yin2, c, Chen Chen1, d and Yang Zhai1, e

1 Automotive Data Center, China Automotive Technology and Research Center Co., Ltd, Tianjin 300393, China;
2 School of Automobile, Chang’an University, Shanxi Xi'an 710000, China.

a zhoubolin@catarc.ac.cn, b zhengjihu@catarc.ac.cn, c 13315290320@163.com,
d chenchen2018@catarc.ac.cn, e zhaiyang@catarc.ac.cn

Abstract. Researches on perception and control technologies are the most crucial area in autonomous driving systems and is the core to achieve fully autonomous driving. Simulations, especially hardware-in-loop and algorithms-in-loop could accelerate the testing process. In this article, based on the second World Intelligent Driving Challenge took place in Tianjin, held by CATARC in May 2018, with the combination of computer vision, edge detection, objection detection, Kalman filter and vehicle dynamics methods, constructed in the Automotive Artificial Intelligence Simulator environment, aiming at paving the road to the better research on simulation testing and autonomous driving.

Keywords: Autonomous Driving, Computer Vision Methods, Decision Algorithms.

1. Introduction

As Chinese Industry is experiencing a booming period in diverged areas, including intelligent manufacturing, internet and other industries, electronic information engineer, mobile communication and artificial intelligence are currently the most promising breakthrough in the whole technology developing trends. Highly autonomous driving as a branch of Artificial Intelligent applications, is integrated with multiple disciplines of technology, and has become a new competing arena of enterprises and organizations worldwide. Comparing to a natural driving behavior of an actual human driver, autonomous driving similarly requires a processing sequence of the surrounding information. Firstly, normally referred to as Sensing Process, computing units will do a recognition process of the targets near the vehicle (referred to as ego car afterwards). And then computing unit will process the ranking of the influence of these targets using the fusion results of different sensors equipped on ego car. This step is known as Fusion Process. Finally, a Decision Process will be done to choose a complete behavior that ego car will implement, according to these inferences of those information.

[1]

Several automotive OEM enterprises and R&D companies are conducting research bases on autonomous driving technologies. And tests for Sensing, Fusion and Decision are deployed in corresponding manners. Throughout the whole testing process, how to apply a competitively low-cost and efficient testing solution is a crucial problem. And among the testing subjects, algorithm testing based on simulation is a promising area of research.

In a simulation-based scenario, tests applied in Sensing, Fusion and Decision modules have the advantages of low-cost and reproducibility. Also, such tests are capable for speedup testing, providing sound fundamentals for algorithm tests. In 2018 May, National Development and Reform Commission, Ministry of Science and Technology of PRC, Ministry of Industry and Information Technology of PRC, Office of Central Cyberspace Affairs and Commissions, Chinese Academy of Engineering, China Association for Science and Technology, Tianjin Municipal People's Government jointly held the 2nd World Intelligent Driving Challenge Competition. During the competition, a simulation testbed was provided for the competitors[2].

This paper is based on the methods team ADC used in 2nd World Intelligent Driving Challenge Competition. The method establishes a complete solution for scenario simulation based on computer vision algorithms, mainly including object salient detection, object recognition. And in actual
engineering work, Kalman filter and vehicle dynamic methods were applied for vehicle control. Such methods will provide fundamental research for autonomous driving algorithm testing.

2. Testbed

The research data is obtained from 2nd World Intelligent Driving Challenge Competition simulation testing group. The experiments are based on AAI (Automotive Artificial Intelligence) companies’ simulation testing software. The competing virtual scenarios were constructed according to actual scenes along Jingjin Highway, with camera and 32-channel lidar data acquired virtually. The organizer used HD map information acquired along the highway from Dongli, Tianjin to Binhai New District, Tianjin to reconstruct a high-resolution highway scenario as shown in Fig.1.

![Virtual scene test diagram](image)

Fig. 1 Virtual scene test diagram

In addition, point cloud, simulated by feedback of lidar rays hit on different objects and materials, is used to test the effectiveness of this simulation method. The highway scenarios used in AAI competition have following specific scenarios: narrowing in roads under repair, highway on-ramp, highway off-ramp. The test scenarios ended at the toll station in Binhai New District, with a scenario of driving off high-way. Plus, the testbed also provides traffic simulation, random traffic data generation, to further the simulation of actual highway situations. The test subjects include front-car intersection, front-car braking, following and emergency tests.

3. Research in Object Recognition and Decision Algorithm based on Sensors

This research used a c930e Logitech camera as video input device. Parameters of it are 90 degrees FOV, 30 FPS and 2.1 mega-pixel, with the resolution of 1920*1080 pixels. To complete a whole loop of object detection by camera, and to perform automatic obstacle-avoidance, some modules required to be accomplished, including camera imaging process, lane salient detection, vehicle recognition and decision-making based on vehicle communication.

3.1 Computation of Imaging Matrices

Due to the reason that 3-dimentional object data is projected in a 2-dimentional imaging plane, some distance inference methods are needed to make-up the loss of data in far-near dimension. Meanwhile, convex lens is normally used in cameras, resulting in aberrance in the image. Thus, imaging matrices need to be computed to transform the distorted image in a planar form [3]. The expression of camera imaging matrices can be expressed as:

\[ x = [R \ t ]X, \quad x \in P^2, X \in P^3 \]  \hspace{1cm} (1)

In the function above, \( P^2 \) is denoted as the 2D projection coordinate system, and \( P^3 \) as 3D projection coordinate system. Supposing zone \((u, v)\) have bias denoted as \((x, y)\), and sum of square error is:

\[ S(x, y) = \sum_u \sum_v w(u, v)(l(u + x, v + y) - l(u, v))^2 \]  \hspace{1cm} (2)
By applying a Taylor Expansion on \( I(u + x, v + y) \), it can be transformed as:

\[
I(u + x, v + y) \approx I(u, v) + I_x(u, v)x + I_y(u, v)y
\]

And an approximation can be applied as:

\[
S(x, y) = \sum_u \sum_v w(u, v)(I_x(u, v)x + I_y(u, v)y)^2
\]

To modify the expression using matrix,

\[
S(x, y) = (x \ y)A \begin{pmatrix} X \\ Y \end{pmatrix}
\]

In the formation, A is the structure tensor of the image, expressed as:

\[
A = \sum_u \sum_v w(u, v)\begin{bmatrix} I_x(u, v)^2 & I_x(u, v)I_y(u, v) \\ I_x(u, v)I_y(u, v) & I_y(u, v)^2 \end{bmatrix}\begin{bmatrix} \langle I_x^2 \rangle \\ \langle I_xI_y \rangle \\ \langle I_y^2 \rangle \end{bmatrix}
\]

By minimizing the error S, the camera can be calibrated to its correct pose.

![Fig. 2 Checkerboard edge detection](image1)

By deducting differences of checkerboard’s scale with ground truth, the imaging matrices and pixel aberrance matrices can be calculated, and corresponding conversion can be accomplished. Then the forward camera viewport can be converted into an overlooked view, as shown in pic.3,

![Fig. 3 Imaging matrix conversion effect diagram](image2)

The converted pixel coordinate in forward camera viewport can represent the distance to ego car in ground truth coordinate system. Such a deduced distance can be referenced and used for Lane Keeping Assist System.

### 3.2 Lane Detection Algorithm

Lane Detection is a crucial part in autonomous vehicle application. Using Lane Detection can distinguish between drivable zone and undrivable zone. Also, an advance Lane Detection can allow more complex decision based on the lane type. For instance, a positive feedback of a full line or a dotted line will give different instructions to the decision about whether to change a track or not.

The core method for Lane Detection is a kind of salient detection algorithm named as Canny Edge Detection, which is a commonly used one.\[4\] Firstly, a Gaussian Blur Filter is applied to smooth the image and blur pepper noise with ground truth signal, making the noise in images less influential. It is applicable because pepper noise is high frequency signal in the frequency domain and are highly probable to be classified as edge. Gaussian Filter is expressed as:
By tuning Gaussian Filter kernel function, images will be classified into different blur effects and will contribute to the validity of edge detection [5]. After Gaussian Filter is applied, a Non-maximum Suppression is applied to calculate local optimal value and to satisfy the function:

$$
G = \sqrt{G_x^2 + G_y^2}, \theta = \arctan2(G_y, G_x) \tag{7}
$$

By applying Gaussian Filter and Non-maximum Suppression, a lane detection can be completed, as shown in Fig.4

![Fig. 4 Lane line edge detection effect map](image)

By limiting detection range, the detection results can be optimized for the current track. Then the algorithm will project the line into world coordinate system, and finally fit a linear function to output a segment fitting the actual lane, as shown below:

![Fig. 5 Limited range lane line detection effect map](image)

The above algorithm can detect lane accurately, and distinguish between full line and dotted line, supporting relevant work in vehicle decision modules for drivable zone, direction and routes.

Due to the reason that other curves or straight lines might interfere recognition task, the recognized results might severely fluctuate even with gradient suppression and locale optimization method, it is necessary to apply Kalman filter for lane detection task to filter out odd points and acquire a steady detection result. [6] The prerequisite of Kalman linear stochastic differential is:

$$
x_k = F_k x_{k-1} + B_k u_k + w_k \tag{8}
$$

$F_k$ denotes transform matrix of time k referring to time k-1, $x_{k-1}$ denotes the ground truth position of lane border at time k-1, $B_k$ denotes actual control signal model in time k, $u_k$ denotes the control vector in time k, $w_k$ denotes multivariant normal distribution noise. Due to the reason that ground truth cannot be directly monitored, we assume the actual edge detection results based on observation of time k as:
\[ z_k = H_k x_k + v_k \] (9)

Here, \( H_k \) denotes the transform matrix of ground truth status to observed status, \( v_k \) is noise in observed status. In this research, we can improve lane recognition accuracy by limiting range of feature points on lanes and assumed. We applied a Kalman filter to avoid target lost when light condition suddenly changes, and lane outline is blocked in a simulated or true scenario. And this method can also apply in other scenarios when the lane sight is lost due to other possible reasons. In AAI simulation scenarios, lane is fitted using segments. Using 1-st order Kalman Filter for segments can avoid lane detection failure when vehicle is in a turn.

3.3 Image-based Vehicle Recognition Algorithm

By applying above algorithms, we can make better decisions to lane information and driving directions in the scenarios. However, to accomplish simulated competition tests, we must recognize different kinds of obstacles on roads. We used YOLO[7], which is a state-of-art target detection algorithm based on deep learning vision methods, to accomplish this task. YOLO has a 106-layer network, it can accomplish bounding box range decision, target recognition and feature extraction.

Bounding box can outline the boundaries of recognized and unrecognized objects. It is a crucial part in the image plane to determine the distance from sensor to target. We acquired approximate outlines of the target object using tuned network model and calculated the center of the object. Suppose the prior height and width of the object is predicted as \( p_h, p_w \), and coordinate \( t_x, t_y, t_w, t_h \), together with the relationship between predicted zone and origin of the image \( c_x, c_y \), we can describe the bounding box value as:

\[
b_x = \sigma(t_x) + c_x \] (10)
\[
b_y = \sigma(t_y) + c_y \] (11)
\[
b_w = p_w e^{t_w}, \quad b_h = p_h e^{t_h} \] (12)

Deep learning is by using manual labeled image data to compare with similar areas in testing images to classify different objects in the image, together with the predicted positions. YOLO is a based on a 53-layer network named Darknet-53. Through iterating multiple convolution layer, residual layer, pooling layers and Softmax classifier to do a recognition process. YOLO is capable to identify a relatively small pixel value, it can have a good effect on objects in long-distance objects, as shown in the following picture:

![Fig 6. Schematic diagram of image vehicle recognition algorithm](image)

We used object recognition algorithm to accomplish an LKA method and avoid pileup, enabling vehicle to keep a safe distance automatically in real time, and can accomplish the safe driving in the competition.

3.4 Decision based on Recognition Results and Vehicle Dynamic Algorithm

During the competition, ECU equipped on vehicle send instructions to AAI server via UDP. The transmission rate is 10 FPS. In each loop, the algorithm will complete message exchange, vehicle control and accomplish the validation process in real-time.
AAI can receive different dimensions of data, including offset from the central line, acceptable lateral movement range, pitch, yaw, roll angles. We comprehensively use lane detection, control algorithm on ECU to adjust deviation from central line and adapt yaw angle, to complete a turning action. By identifying target vehicle and keeping a safe distance, we can fulfill the collision avoidance function and allow speed adjustment.

In a highway simulation scenario, the vehicle speed is limited to 100 km/h. We used central point of the screen pixels, along with mean of dots along the detected lane, to compute the signed difference of them, checking if the lane is at a turn. If the lane is on a right turn, mean of dots would be larger than central pixel, indicating a positive difference. Correspondingly, if the lane is on a left turn, the difference would be negative. If the lane is straight, allowing a 20 pixels range on both left and right, we considered this situation to be a straight driving lane, these conditions can be interpreted as:

\[
yaw = \begin{cases} 
0.3(mid - lane) & \text{if } x < |20| \\
0.6(mid - lane) & \text{if } |20| \leq x < |60| \\
0.9(mid - lane) & \text{if } |60| \leq x 
\end{cases}
\]  

(13)

Using such a definition of bias, we can set the steering of the vehicle accordingly. Bias is computed using central of the camera and heading yaw angle of the vehicle. Then steering angle is computed using the equations above. After all ranges are set, we used a PID (Proportional-Integral-Derivative) algorithm [8] to accomplish the vehicle control. PID algorithm is expressed as:

\[
u(t) = K_p e(t) + K_i \int_0^t e(t') dt' + K_d \frac{de(t)}{dt}
\]  

(14)

\(K_p, K_i, K_d\) are three non-negative control parameters. By using a PID algorithm, we can filter control signal of previous moment, and increase the stability of the current moment. And finally, by setting a safe range of distance as 100m. By combining speed control and angle control we completed the speed control of the vehicle, fulfilling a safe driving test.

4. Conclusion

Applying research contents above, CATARC team completed all the competing subjects, with a valid driving distance of 10 miles. In comprehensive scoring subjects including obstacle avoidance, driving distance, deviation, CATARC team obtained 3rd position among 16 competitors. During the competition, the team gained experiences in algorithms and technologies of simulation platform, ECU communication, lane detections, traffic sign recognition, and vehicle-dynamic-based controlling. The experiences provide research basis for the Vehicle-in-loop and simulation tests, helping with further research on performance evaluation in simulation environments and testing in real scenarios. In the future, the testing work will be the foundations for developing real vehicle testing algorithm, scenarios testing regulations and standards in virtual test, supporting autonomous driving technologies’ developments in the future.

References

[1]. X Xiong, J Wang, F Zhang, K Li, Combining Deep Reinforcement Learning and Safety Based Control for Autonomous Driving[J]. arXiv: 1612.00147, 2016.

[2]. Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, Vladlen Koltun, Carla: An Open Urban Driving Simulator[C]. Proceedings of the 1st Annual Conference on Robot Learning 2017: 1-16.

[3]. Z Zhang. A Flexible New Technique for Camera Calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence[M], 22(11):1330-1334, 2000.
[4]. Canny, J., A Computational Approach to Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

[5]. Mallat S, Zhong S. Characterization of Signals from Multi scale Edges [J]. IEEE Trans on PAMI, 1992, 14 (7):710-732.

[6]. R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems. ASME Journal of Basic Engineering[J], Vol. 82: 34-45, 1960.

[7]. Joseph Redmon, Ali Farhadi, Yolov3: An Incremental Improvement[J]. arXiv: 1804.02767, 2018.

[8]. Bequette, B. Wayne. Process Control: Modeling, Design, and Simulation. Prentice Hall PTR, 2010.