中國機械工程學刊第四十四卷第二期第 165~178 頁(民國一百一十二年)
Journal of the Chinese Society of Mechanical Engineers, Vol.44, No.2, pp165~178 (2023)

Optimization and Design of Driverless Vehicle Software System Based on Image Recognition

Bi Li*, Yi-Jui Chiu**, ***, Cai-En Weng****, Yuh-Chung Hu*****, Bing-Rong Zhang**, Ji-Ming Yi**and Sheng-Rui Jian******

Keywords: Intelligent vehicle, Image preprocessing, Edge extraction algorithm, Track condition judgment algorithm, Path tracking planning, Fuzzy-based PID control.

ABSTRACT

An intelligent vehicle is a system that integrates environment perception, path decision planning, automatic driving, and other functions. In order to improve the tracking and motion performance of intelligent vehicles, a system that includes image preprocessing, image processing, path tracking planning, and intelligent vehicle control is designed and optimized. First, the principles and implementation effects of three basic threshold algorithms and the image denoising algorithm are discussed. Second, the traditional edge extraction algorithm and track condition judgment algorithm are improved. Then, a path tracking planning method based on the midline algorithm and an edge fitting algorithm based on the least square algorithm are proposed and simplified. Finally, aiming at solving the shortcomings of the traditional PID algorithm that cannot update the values of K_p, K_i and K_d, an intelligent vehicle control system based on the PID algorithm and fuzzy control is proposed and verified by simulation and experiment. The results show that the designed filtering algorithm can effectively reduce the image noise. The improved edge extraction algorithm has an obvious filtering effect on the abnormal data in the process of intelligent vehicle operation. The difference between the straight and bent track obtained by the improved track condition judgment algorithm is 7.39, which is larger than 1.78 obtained by the traditional algorithm. The improved algorithm is sensitive to the change in the track bending degree and overcomes the problem that the performance of the traditional algorithm decreases with the bending degree. Using the simplified edge algorithm, an edge fitting algorithm based on the least square algorithm is developed. This algorithm is similar to the algorithm with the R-squared greater than 0.994, and the number of edge points used for calculation is reduced from the original 48 points to 2 or 3 points, which greatly improves the operation efficiency of the intelligent vehicle. Using the fuzzy-based PID control algorithm, after the target speed changes, the output curve reaches the target speed at 0.44 s, the maximum excess is approximately 16.4 rpm, and the algorithm becomes stable at the target speed after 7.9 s, which is less than 1.2 s, 56.7 rpm and 12.1 s of traditional PID control respectively. Thus, using the proposed fuzzy-based PID control algorithm, the control performance of the system can be significantly improved. The experimental results of real intelligent vehicle show that the proposed fuzzy PID control algorithm can significantly improve the control effect under high-speed operation.

Paper Received January, 2022. Revised March, 2022. Accepted April, 2022. Author for Correspondence:Yuh-Chung Hu

e-mail: ychu@niu.edu.tw

* *Graduate Student, School of Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen, 361024, Fujian, China.*
** *Professor, School of Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen, 361024, Fujian Province, China.*
*** *Professor, Fujian key laboratory of advanced design and manufacture for bus, Xiamen, 361024, Fujian, China.*
**** *Professor, School of Mechanical Engineering, Sanming University, Sanming, Fujian, China.*
***** *Professor, Department of Mechanical and Electro-Mechanical Engineering, National Ilan University, I-lan Taiwan.*
***** *Professor, Department of Materials Science and Engineering, I-Shou University, Kaohsiung 840, Taiwan.*

INTRODUCTION

Fatigue driving, drunk driving, and other factors are common causes of traffic accidents. With the sustained and stable growth of the world's macro-economy and continuous increase in vehicle ownership, traffic accidents have become more frequent, posing a great threat to people's lives and property safety (Fu et al., 2016; Useche et al., 2017; Zhang et al., 2016). Driverless vehicles can collect real-time traffic information and automatically adjust their speed, complete patrol driving, and other functions, thus avoid traffic accidents caused by human factors, which makes driverless vehicles attract great attention from all walks of life (Konig and Neumayr, 2017).

Rao et al. (2019) proposed an automatic trajectory tracking method based on image processing to overcoming the defect that the traditional trajectory tracking algorithm can easily be interfered by the outside world. The simulation results indicated that the method simplified the traditional method, enhanced the control speed and stability, and had a strong tracking capability when the vehicle was driving automatically at high speed. Li et al. (2020) proposed a feature point extraction algorithm based on a local adaptive threshold to overcome the problem of visual perception and location. The experimental results showed that compared with the existing two advanced algorithms, their algorithm could track feature points more accurately and stably. Xu et al. (2011) proposed an Otsu thresholding algorithm based on a limited gray search range to solve the problem that the Otsu thresholding tends to a class with a large variance. The experimental results indicated that the algorithm had a better image segmentation effect than the Otsu algorithm. Li and Liu (2019) proposed a neural network-based Otsu thresholding algorithm for the problem of difficult distinction between plants and soil in images, and the experimental results showed that the proposed thresholding algorithm can effectively segment the regions of plants and soil. Brettin et al. (2008) proposed a vehicle path tracking algorithm based on conservation law, which used a wavefront tracking scheme to simulate network density first, and then reconstructed the vehicle position and analyzed the influence of its interaction with density wave; finally, using the algorithm and initial data, the exponential uniform convergence of vehicle trajectory was proven. Naranjo et al. (2008) put forward a type of automatic driving system with the overtaking ability. The overtaking was achieved using a fuzzy control algorithm. The experimental results showed that the system could simulate human overtaking at a slower speed according to the GPS location and map information of two vehicles. Milanes et al. (2014) designed a new type of adaptive cruise control system, which adopted a distance sensor and wireless communication technology, and this system was successfully applied to Infiniti m56s vehicle. The experimental results showed that the adaptive cruise control system could improve the response time and transmission stability of the system compared with the production control system and also had a better anti-interference ability. Gaining et al. (2017) proposed a lateral tracking control strategy. The vehicle dynamics model and steering system model were built using the vehicle parameters, and the lateral path tracking control was realized using the control algorithm. The simulation results indicated that the method had good real-time performance and strong robustness. Jeng et al. (2021) proposed a dynamic path planning method based on practical dynamic constraints and performed lane change and turnaround experiments, and the results showed that the proposed method can provide efficient paths in real time in many driving scenarios. Chen et al. (2019) proposed a vision and fuzzy control-based navigation method for a robotic ship and conducted experiments on one, and the results showed that the proposed method can effectively control its operation. Chen et al. (2016) proposed an intelligent driving control strategy based on the coupling control of the throttle and brake for realizing the safe driving of vehicles. Simulink was used to simulate and analyze the control strategy. The simulation results showed that the control strategy could track a vehicle in front actively while maintaining safe distance and constant speed travel. Chen et al. (2019) proposed a speed adaptive control algorithm, which solved the problem that a vehicle is difficult to control the speed accurately, and successfully applied it to the Ford fox. The performances of this algorithm, PID, and fuzzy neural network were compared and analyzed. The results showed that the proposed algorithm could realize accurate control of the vehicle speed, and the mileage deviation was reduced. Jin et al. (2020) proposed an improved PID controller based on fuzzy adaptive control for the low control accuracy of the transplanting manipulator. The simulation results using MATLAB showed that the proposed fuzzy PID control can improve the control accuracy of the transplanting manipulator. Shi et al. (2020) proposed an improved PID control system based on fuzzy adaptive control to improve the reliability of the underwater motion of small bionic robots. The results showed that the proposed control system has better dynamic performance than the PID method. Senapati et al. (2018) proposed a speed control algorithm for smart cars based on fuzzy PID and compared it with PID control algorithms based on different tuning rules. The simulation results showed that the proposed algorithm has better stability and response speed. Guo et al. (2017) proposed a vehicle tracking control system, which uses an adaptive fuzzy control strategy to allocate the required tire longitudinal and lateral coupling forces in real-time according to the optimal tire force distribution law. The simulation results indicated that the control method could effectively improve the tracking accuracy, safety, and driving comfort. Mohammadzadeh et al. (2020) proposed a fuzzy control algorithm for automatic driving based on a non-single instance fuzzy system and a non-stationary fuzzy set. The experiment was conducted under different measurement noise levels. The results showed that the algorithm could be suitable for the path tracking task under various conditions and external influences. Liu et al. (2019) proposed a model-free adaptive control (MFAC) algorithm based on a double successive projection algorithm, which transformed the trajectory tracking problem into a stability problem and was successfully applied to the lateral control of “Ruilong” autopilot vehicle. It was tested in Fengtai, Beijing, China, and satisfactory results were achieved on China’s intelligent vehicle challenge in 2015 and 2016. Almagambetov et al. (2015) proposed a real-time vision-based algorithm to track and detect the signal of vehicle tail lights. The test results showed that the algorithm could track vehicle tail lights and detect alarm signals under most lighting conditions. Chen et al. (2017) proposed a mobile cloud computing framework based on deep learning. The cloud platform was used for training, and mobile devices were used for recognition and data collection; also, the impact of GPU on system performance was examined. Experimental results showed that GPU could increase the computing speed of the system by approximately 20 times, which could be of great significance for deep learning recognition. Drage et al. (2014) realized the control function using a wired system and a navigation control system for a formula car. The test results showed that the proposed system could keep the F1 car running in a straight line, and the steering speed was the same as that of the human driver. Kabzan et al. (2019) proposed a learning-based automatic driving vehicle control algorithm, which uses the Gaussian process enhancement model to improve vehicle performance. The test results on the AMZ driverless vehicle showed that under the premise of ensuring safety, the single lap running time was reduced by 10%, while the driving performance of the vehicle was significantly improved. Etlik (2019) proposed an automobile control system based on fuzzy logic, which combines control theory, fuzzy logic, and image processing to realize automobile automation. The performance of this system was tested in the JavaScript racing game environment. The test results showed that the control algorithm could run successfully on different tracks. Li et al. (2018) proposed a lane detection system based on a camera. The test was conducted using the dataset of the California Institute of Technology and an autonomous vehicle "tuyou". The test results indicated that this method could deal with the complex environment at a low speed and had good robustness and effectiveness. Yu et al. (2019) designed a tracking controller based on front axle reference fuzzy pure tracking control, which uses a fuzzy parameter adaptive algorithm to overcome the defect of the traditional tracking algorithms that can difficultly balance accuracy and stability. The results of the simulation and field experiment indicated that this method enhances the accuracy and robustness of the controller.

This paper designs and optimizes a system for image preprocessing and processing, path tracking planning, and control of an intelligent vehicle. As for image preprocessing, three basic threshold algorithms are discussed, and an improved filtering algorithm based on image binarization and *k*-NearestNeighbor (*k*NN) denoising algorithm is proposed. Image processing is realized using an improved traditional edge extraction algorithm and track condition judgment algorithm. The path tracking planning algorithm based on a midline algorithm and the fitting edge algorithm based on the least square algorithm are proposed, and the edge algorithm is simplified by the Lagrange interpolation. The intelligent vehicle control system based on the PID algorithm and fuzzy control is used to overcoming the defect that the traditional PID control algorithm cannot update the PID parameters in real-time according to the intelligent vehicle requirements. Finally, an intelligent car and MATLAB software are used to verify and simulate the proposed system.

WORKING PROCESS OF INTELLIGENT VEHICLE SYSTEM

The working process of an intelligent vehicle system is presented in Fig. 1. As shown in Fig. 1, after the initialization of the system, the camera collects image information. After image preprocessing (image binarization and image noise reduction) and image processing (edge extraction and track condition judgment), the current position and tracking path of the intelligent vehicle are determined. Fuzzy-based PID control is used to control the steering gear and motor, and finally, automatic driving of the intelligent vehicle is realized.

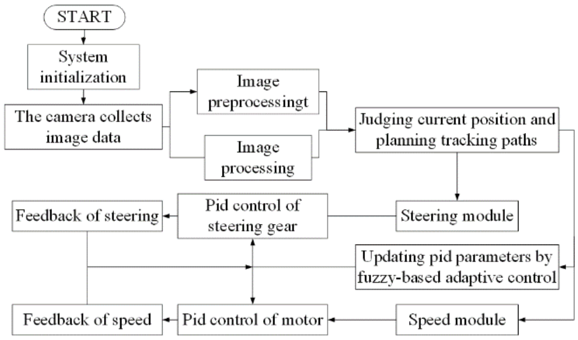


Fig.1. The working process of an intelligent vehicle system

IMAGE PREPROCESSING

IMAGE BINARIZATION

To reduce the amount of Microcontroller Unit (MCU) operation and improve the processing speed, it is necessary to conduct binary processing on the collected image, to reduce the complexity of the image, and to speed up the processing speed without changing the main frequency of MCU. The main part of image binarization is threshold selection. The value of each pixel in the image matrix is compared with a threshold value, and if it is lower than the threshold value, it is considered black; otherwise, it is considered white. Common threshold algorithms include the average threshold algorithm, iterative threshold algorithm, and Otsu threshold algorithm.

Suppose image pixels are set to 0~255 by gray value, the threshold value is set to T, and the number of pixels with the gray value i is set to n_i. The algorithm of related parameters of the threshold algorithm is as follows.

The ratio of the number of background pixels to the total number of pixels, which is denoted as w_0, can be obtained as:

$$w_0 = \frac{\sum_{i=0}^{T} n_i}{\sum_{i=0}^{255} n_i}. \tag{1}$$

The proportion of the number of foreground pixels in the total number of pixels, which is denoted as w_1 can be calculated by:

$$w_1 = \frac{\sum_{i=T}^{255} n_i}{\sum_{i=0}^{255} n_i} = 1 - w_0. \tag{2}$$

The average gray level of background pixels is denoted as u_0 and can be obtained as:

$$u_0 = \frac{\sum_{i=0}^{T} i n_i}{\sum_{i=0}^{T} n_i}. \tag{3}$$

The average gray level of foreground pixels is denoted as u_1 and can be calculated by:

$$u_1 = \frac{\sum_{i=T}^{255} i n_i}{\sum_{i=T}^{255} n_i}. \tag{4}$$

The average gray value can be expressed as:

$$\overline{V} = \frac{\sum_{i=0}^{255} i n_i}{\sum_{i=0}^{255} n_i} = w_0 u_0 + w_1 u_1. \tag{5}$$

The average threshold algorithm is defined by:

$$T = a\overline{V} + b, \tag{6}$$

where a and b are artificially set parameters.

The initial threshold of the iterative threshold algorithm is defined as:

$$T[0] = \overline{V}. \tag{7}$$

The iterative formula is given by:

$$T[k+1] = 0.6(u_0 + u_1). \tag{8}$$

The accuracy requirements are expressed as:

$$\left|T[k+1] - T[k]\right| \le \varepsilon, \tag{9}$$

where ε is a sufficiently small positive number.

The Otsu threshold algorithm finds the most suitable gray value by calculating the variance of front and background pixels. The variance between the foreground and background is given by:

$$g = w_0\left(u_0 - \overline{V}\right)^2 + w_1\left(u_1 - \overline{V}\right)^2. \tag{10}$$

By substituting (2) and (5) into (10), the following relation can be obtained:

$$g = w_0 w_1 \left(u_0 - u_1\right)^2. \tag{11}$$

The three algorithms were compared using MATLAB software. The optimal threshold values of the average threshold algorithm, the iterative threshold algorithm, and the Otsu threshold algorithm were 98.93, 97.33, and 104, respectively.

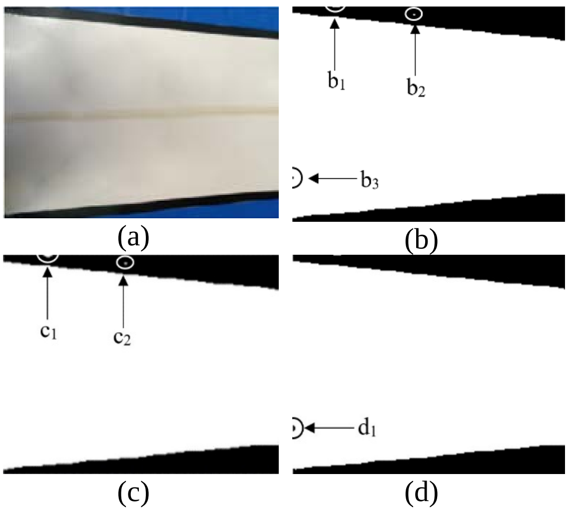


Fig.2. The performance of the threshold algorithm

The results of the three threshold algorithms are presented in Fig. 2. The original image of an intelligent vehicle track is presented in Fig. 2(a); the result of the average threshold algorithm is shown in Fig. 2(b); the result of the iterative threshold algorithm is displayed in Fig. 2(c); the result of the Otsu threshold algorithm is shown in Fig. 2(d). The overall effect of the result presented in Fig. 2(b) is good; there are only three small speckles in the image, which are denoted as b1, b2, and b3. The values of a and b can be manually adjusted to control the output of the average threshold algorithm in order to obtain a satisfactory imaging effect, but for the trajectory environment changing with time, it is needed to update these values constantly. The foreground effect in Fig. 2(c) is better than Fig. 2(b) and Fig. 2(d); namely, only two white speckles c1 and c2 appear in the image background, but they are larger than b1 and b2 in Fig. 2(b). The background effect in Fig. 2(d) is better than Fig. 2(b) and Fig. 2(c); only one black dot d1 appears in the foreground of the image, but it is larger than b3 in Fig. 2(b). The results show that if the recent lines of image data are not used, the effect of Otsu thresholding algorithm is obviously the best, so this paper uses this algorithm for image binarization.

IMAGE DENOISING

Image noise can disturb an intelligent vehicle to obtain the correct image data and can also affect the stability of the intelligent vehicle. A k-NearestNeighbor (kNN) denoising algorithm is selected for intelligent vehicle image denoising.

Suppose that M_0 is a noise, whose coordinates are (x_0, x_0), and M_i is a point adjacent to M_0, and its coordinates are (x_i, x_i); then, the distance between the two points is calculated by:

$$D_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \;. \tag{12}$$

When k is 10, the filtering algorithm can be set based on the kNN algorithm as follows:

$$M_0 = \sum_{i=0}^{k} \frac{M_i}{2^{D_i+2}} \;. \tag{13}$$

To simplify the above filtering algorithm to improve the efficiency of intelligent vehicles, a simple improved filtering algorithm is designed based on the idea of image binarization and kNN algorithm.

The operation principle of the improved filtering algorithm is presented in Fig. 3, where it can be seen that if there are three points with a value of zero around the point with a value of one, the point is judged to be noise and is converted to a value of zero. Fig. 3(a) shows the data before noise reduction, and Fig. 3(b) shows the data after noise reduction.

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

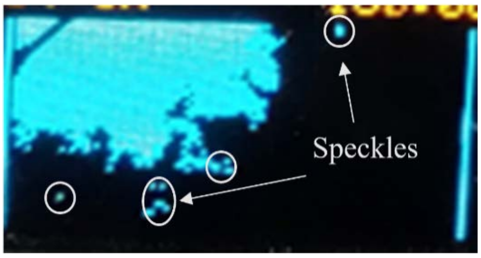
(a)

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

(b)

Fig.3. The operation principle of the improved filtering algorithm

The performance of the improved filtering algorithm is presented in Fig. 4. Figure 4(a) shows the image without using the improved filtering algorithm, and Fig. 4(b) shows the image obtained by the improved filtering algorithm. By comparing images in Figs. 4(a) and 4(b), it can be seen that the image in Fig. 4(a) is relatively scattered, and there are many small speckles in this image, while the image in Fig. 4(b) is relatively concentrated, and there are basically no small speckles like those in Fig. 4(a), so the quality of the image in Fig. 4(b) is better than that in Fig. 4(a). Thus, the improved filtering algorithm can effectively reduce the image noise.



(a)

(b)

Fig.4. The performance of the improved filtering algorithm

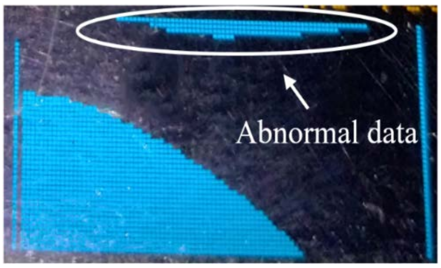
IMAGE PROCESSING

IMPROVED EDGE EXTRACTION ALGORITHM

The traditional edge extraction algorithm is to find the black and white jump points of each line in an image. A jump point from black to white is the left edge, and it is marked as *left*[*nr*] = *nc*, and a jump point from white to black is the right edge, and it is marked as *right*[*nr*]=*nc*. In the experiment, when the traditional edge extraction algorithm was used, there were misjudgments in some scenes.

As shown in Fig. 5, when an intelligent vehicle recognizes the reflection of another track or floor, the traditional edge extraction algorithm makes scene misjudgment. Fig. 5(a) shows the position image of a car, and Fig. 5(b) shows the display image of the OLED screen. As can be seen from Fig.5, the small white area in the upper-right corner of Fig. 5(b) denotes the abnormal data collected by the intelligent vehicle from another track. This paper improves the traditional edge extraction algorithm and proposes a simple and more reasonable edge extraction algorithm named the T-shape edge extraction algorithm that is based on the actual intelligent vehicle image data.

(a)



(b)

Fig.5. The scene misjudgment of the traditional edge extraction algorithm

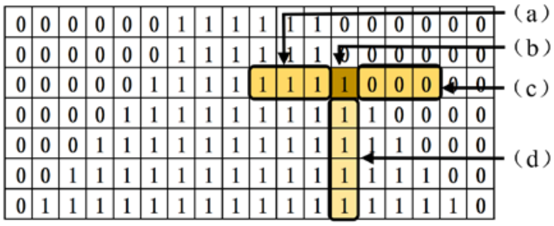
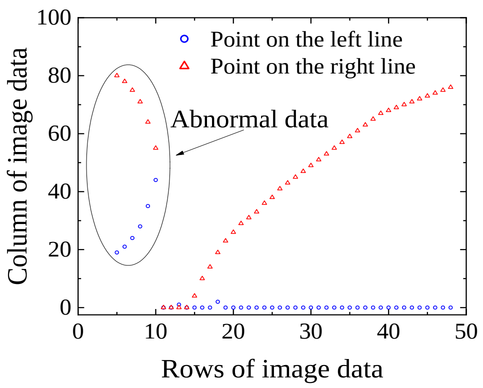


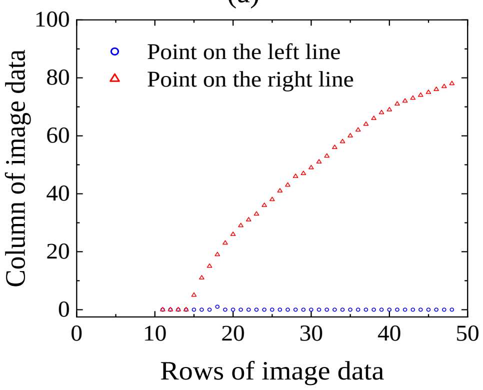
Fig.6. The introduction of rules of T-shaped edge extraction algorithm

The rules of T-shaped edge extraction algorithm in identifying the edge is presented in Fig. 6, where (a) indicates three points to the left of the boundary point, which are all one; (b) is the boundary point; (c) indicates three points to the right of the boundary point, which are all zero; (d) indicates four points below the boundary point, which are all one.

The performances of the traditional and improved algorithms in collecting data for scenes similar to that presented in Fig. 5 are displayed in Fig. 7. Fig. 7(a) shows the performance of the traditional edge extraction algorithm, and Fig. 7(b) shows the performance of the T-shaped edge extraction algorithm. As presented in Figs. 7(a) and 7(b), the traditional algorithms cannot process abnormal data similar to that presented in Fig. 5(b), while the improved algorithm has an obviously improved filtering effect on abnormal data and can avoid misjudgment caused by external interference of intelligent vehicle.



(a)



(b)

Fig.7. The performances of the edge extraction algorithms

IMPROVED TRACK CONDITION JUDGMENT ALGORITHM

The current track condition can be judge by calculating the bending degree of the track. Taking the left-bend track as an example, the right edge with relatively complete data is used for analysis. The traditional algorithm to judge the bending degree of the track is to compare the changes in the column values of the 8th, 28th, and 48th rows of data to obtain the track bending degree, which can be expressed as:

$$bend = \frac{right[28]-right[8]}{right[48]-right[28]}. \tag{14}$$

The value of *bend* directly reflects the bending degree of the track. The larger this value is, the greater the bending degree of the track. By setting an appropriate threshold, the current track condition can be judged.

The experimental results have shown that when the intelligent vehicle uses this algorithm, it is prone to understeer in the bent track. The data analysis has shown that the 8th row of data points collected by the intelligent vehicle in the bent track is lost.

To avoid the interference caused by the loss of data in the 8th row, the row-coordinate *ynr* of the farthest row on the right edge collected is used as the initial row for calculating *bend*. At the same time, the current right edge data *nc*=*right*[*nr*] are compared with the right edge data *nc*=*right1*[*nr*] of the ideal straight track to avoid the error caused by the intelligent vehicle system. The relevant algorithm is defined as follows:

$$bend = \left(\sum_{i=ynr}^{48} right1[nr]-right[nr]\right)/(49-ynr) \tag{15}$$

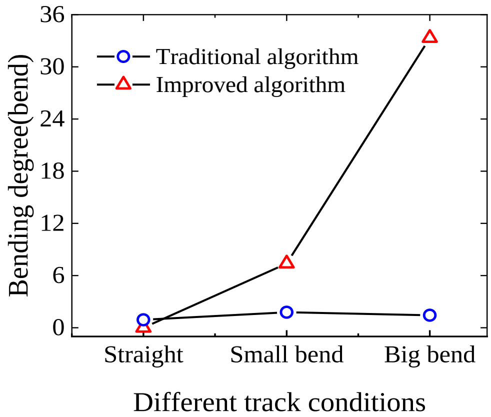


Fig.8. The performances of the two algorithms in the track condition judgment

The performances of the traditional algorithm and the improved algorithm in determining different bending degrees of the track are presented in Fig. 8. As shown in Fig. 8, the difference between the bending degree of the straight track and the small-bend track obtained by the traditional algorithm is only 0.86, which indicates that this algorithm is not sensitive to the change in the track bending. Moreover, since the 8th row of data points of the big-bend track is lost, the obtained bending degree is 1.43, which is less than 1.78 of the small-bend track, indicating that the obtained result is not reasonable. Therefore, at a fast vehicle speed, it could easily happen that the vehicle understeers and rushes out of the big-bend track. The difference between the bending degree of the straight track and the small-bend track obtained by the improved algorithm is 7.39, which shows that this algorithm is sensitive to the track bending change. The obtained bending degree of the big-bend track is 33.32, which is larger than 7.39 of the small-bend track. The obtained result is reasonable; thus, the improved algorithm overcomes the defect of the traditional algorithm. The test results of an intelligent vehicle show that the improved algorithm can enable a vehicle to run more smoothly and overcome the problem that a vehicle could easily rush out of the track at high speed.

PATH TRACKING ALGORITHM DESIGN

Because the track conditions mainly include the straight and bent tracks, this study performs the example analysis using these types of tracks. The middle line algorithm is used to plan the tracking path.

For a straight track, if the left and right edge algorithms are denoted as y_l and y_r, the path tracking algorithm can be expressed as:

$$y=\left(y_l+y_r\right)/2\,. \tag{16}$$

For the left-bend track, due to the limitation of an intelligent vehicle's camera, the left edge data cannot be collected, so the right-edge data denote the complete data that are used to analyze the track's center line. If the right edge algorithm is denoted as y_r, the path tracking algorithm is expressed as:

$$y=y_r\ /\,2\,. \tag{17}$$

The path defined by (17) that is obtained by this method denotes the center line of the image, not the center line of the actual track. In this study, the center line of the actual track is drawn by subtracting a certain value from the right-edge data of each row of the left-bend track.

The specific steps are as follows:

1). Collect the right-edge data *right1*[*nr*] of the ideal straight track;
2). Using the data obtained in the previous step, the right edge algorithm of the ideal straight track can be defined as:

$$y_r=0.6x+64\,. \tag{18}$$

3). The difference between the ideal right edge and centerline *MAX_COL/2* is expressed as:

$$y_r=0.6x+64-MAX\ \_COL\,/\,2\,. \tag{19}$$

4). The actual left-bend tracking algorithm is defined as:

$$y=y_r-y_{r1}\,. \tag{20}$$

Obviously, the key part of the path tracking algorithm design is the design of the edge algorithm.

EDGE ALGORITHM DESIGN

The edge data obtained by the T-shape edge extraction algorithm consist of discrete points. Using the least square fitting algorithm and Lagrange interpolation algorithm in combination with the actual analysis of intelligent vehicles, the track edge extraction algorithm of intelligent vehicles is designed and optimized.

Assuming that the data point is (x_i, y_i) where i = 0, 1, …, n, the polynomial of degree m can be obtained as:

$$y(x)=\sum_{j=0}^{m}a_j x^j\,, \tag{21}$$

where a_j denotes the undetermined coefficient.

Fitting the normal polynomial equation by the least square algorithm, the following relation is obtained:

$$\begin{pmatrix} n+1 & \sum_{i=0}^{n}x_i & \cdots & \sum_{i=0}^{n}x_i^{\ m} \\ \sum_{i=0}^{n}x_i & \sum_{i=0}^{n}x_i^{\ 2} & \cdots & \sum_{i=0}^{n}x_i^{\ m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n}x_i^{\ m} & \sum_{i=0}^{n}x_i^{\ m+1} & \cdots & \sum_{i=0}^{n}x_i^{\ 2m} \end{pmatrix}\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix}=\begin{pmatrix} \sum_{i=0}^{n}y_i \\ \sum_{i=0}^{n}x_i y_i \\ \vdots \\ \sum_{i=0}^{n}x_i^{\ m}y_i \end{pmatrix} \tag{22}$$

The value of $\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix}$ can be obtained from (22) and then substituted into (21) to obtain the edge algorithm.

An intelligent vehicle is often not required to obtain a particularly accurate edge algorithm. In order to simplify the edge algorithm, this paper uses multiple keypoints to develop the edge algorithm.

A simplified algorithm based on Lagrangian interpolation algorithm is proposed. Suppose the Lagrangian basis function L_i is defined as:

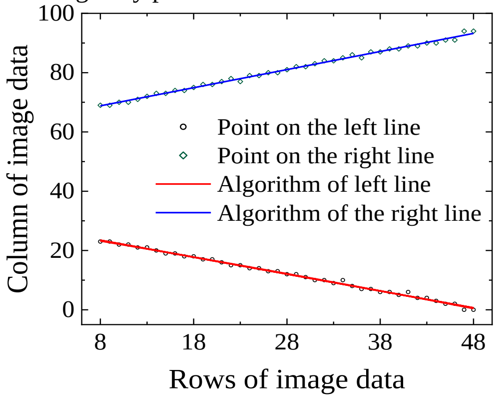
$$L_i=\prod_{j=0,\,j\neq i}^{n}\frac{X-x_j}{x_i-x_j}\,. \tag{23}$$

Then, the Lagrange interpolation polynomial can be expressed as:

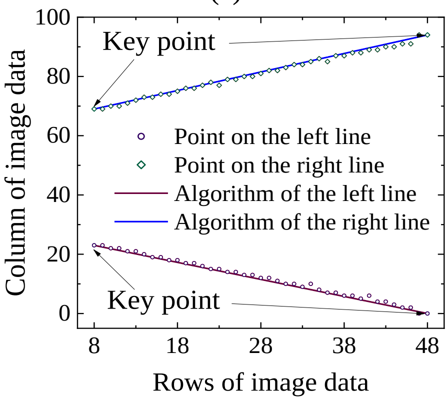
$$Y_{n+1}=\sum_{i=0}^{n}y_i L_i\,. \tag{24}$$

By substituting the coordinates of appropriate edge key points into (23) and (24), a simplified edge algorithm can be obtained.

Generally, the farther the distance between the used points is, and the more the points used, the better the edge algorithm performance. The edge of a straight track is similar to a straight line, and there is an axiom of a straight line, which shows that at least two key points are needed to determine a straight line. The edge of a bent track is similar to a circular arc, and it is known from the axiom of a circle that at least three key points that are not co-linear are required to determine a circle. Thus, take edge points of the straight track in the first and last rows as edge key points, and edge points of the bent track in the first, middle, and last rows as edge key points.



(a)



(b)

Fig.9. The performances of the least square fitting algorithm and the simplified algorithm on the straight track

The performances of the least square fitting algorithm and the simplified algorithm on the straight track are presented in Fig. 9. Fig. 9(a) shows the performance of the least square fitting algorithm on the straight track, and Fig. 9(b) shows the performance of the simplified algorithm on the straight track. As shown in Fig. 9, the value of the R-squared fitted by the least square algorithm is larger than 0.994, which indicates that the fitting degree and reliability between the fitting line and the actual data are high.

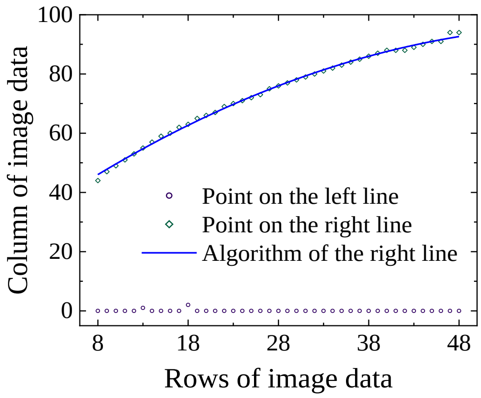
The current straight-track edge algorithm by the least square fitting algorithm is defined as:

$$y = 63.92 + 0.611x . \tag{25}$$

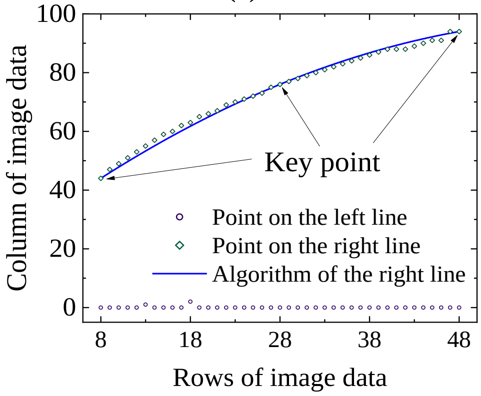
The current straight-track edge algorithm obtained by the simplified algorithm is defined as:

$$y = 64 + 0.625x . \tag{26}$$

Obviously, the simplified algorithm can achieve a similar performance as that presented in Fig. 9(a). The number of edge points used in the calculation is reduced from 48 to 2, thus greatly reducing the calculation cost and improving the operation efficiency of intelligent vehicles.



(a)



(b)

Fig.10. The performances of the least square fitting algorithm and the simplified algorithm on the bent track

The performances of the least square fitting algorithm and the simplified algorithm on the bent track are presented in Fig. 10. Fig. 10(a) shows the performance of the least square fitting algorithm, and Fig. 10(b) shows the performance of the simplified algorithm. As shown in Fig. 10 the value of the R-squared fitted by the least square algorithm is larger than 0.994, which indicates that the fitting degree and reliability between the fitting line and the actual data are high.

The current bended-track edge algorithm by the least square fitting algorithm is defined as:

$$y = 27.28 + 2.23x + 0.0175x^2 . \tag{27}$$

The current bended-track edge algorithm obtained by the simplified algorithm is defined as:

$$y = 30 + 2.11x + 0.0168x^2 . \tag{28}$$

Obviously, the simplified algorithm can achieve a similar effect as that presented in Fig. 10(a). The number of edge points used in the calculation is reduced from 48 to 3, thus greatly reducing the calculation cost and improving the operation efficiency of intelligent vehicles.

FUZZY-BASED PID CONTROL ALGORITHM

The traditional PID control algorithm cannot update the values of K_p, K_i and K_d in real-time. Therefore, in this study, the fuzzy-based PID control algorithm is proposed to update the PID parameters in real-time to meet the requirements of intelligent vehicles.

The main differences between this paper and previous studies are as follows:

1). The research object is an intelligent vehicle control system.
2). The proposed control algorithm is verified by MATLAB simulation experiment and real intelligent vehicle experiment.

The schematic diagram of the fuzzy-based PID control system is displayed in Fig. 11. The system is composed of the fuzzy-based adaptive control and the traditional PID control. According to the changes of E and EC and the fuzzy control rules, the values of K_p, K_i and K_d are updated in real time to ensure the good performance of the control system.

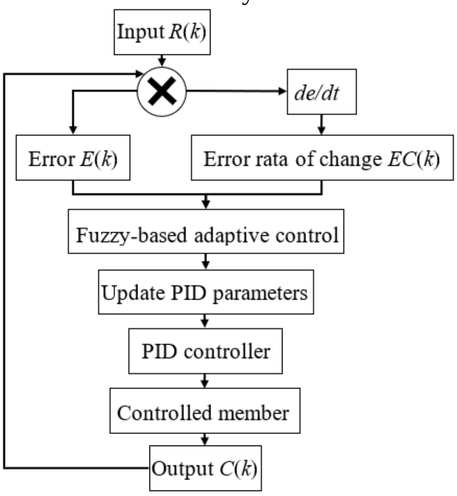


Fig.11. The schematic diagram of the fuzzy-based PID control system

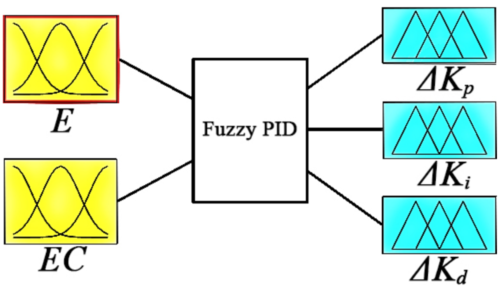


Fig.12. The framework of the fuzzy-based PID control system

The framework of the fuzzy-based PID control system with E, EC, and ΔK_p as inputs and ΔK_i and ΔK_d as outputs is presented in Fig. 12. E, EC, ΔK_p, ΔK_i, and ΔK_d are fuzzified into the following seven sets: *NB* (negative big), *NM* (negative middle), *NS* (negative small), *ZO* (zero), *PS* (positive small), *PM* (positive middle), and *PB* (positive big). The change range is set to be in the range of [-6, 6]. The membership functions of E and EC use the Gaussian curve function, and the membership functions of ΔK_p, ΔK_i, and ΔK_d use the triangle functions.

The fuzzy control rules proposed generally meet the adaptive principles. When $E = NB$ and $EC = NB$, to speed up the response speed of the system and prevent the differential supersaturation caused by the instantaneous increase in the initial deviation from exceeding the allowable range, it is set that $\Delta K_p = PB$, $\Delta K_i = NB$, and $\Delta K_d = PS$. When EC or E is large, ΔK_p is changed from large to small, ΔK_i is changed from small to large, and ΔK_d is changed from large to small and then again to large. When $E = ZO$ and $EC = ZO$, to reduce the overshoot and ensure a certain response speed, it is set that $\Delta K_p = ZO$ and $\Delta K_i = ZO$. At the same time, ΔK_d should be small ($\Delta K_d = NS$) to reduce the influence of K_d on the system performance. When E = PB and $EC = PB$, to reduce the overshoot, it is set that $\Delta K_p = NB$, $\Delta K_i = PB$, and $\Delta K_d = PB$.

Table 1. Fuzzy control rules

ΔK_p/ΔK_i/ΔK_d (E \ EC)	NB	NM	NS	ZO	PS	PM	PB
NB	PB/NB/PS	PB/NB/NS	PM/NM/NB	PM/NM/NB	PS/NS/NB	ZO/ZO/NM	ZO/ZO/PS
NM	PB/NB/PS	PB/NB/NS	PM/NM/NB	PS/NS/NM	PS/NS/NM	ZO/ZO/NS	ZO/ZO/ZO
NS	PM/NM/ZO	PM/NM/NS	PS/NS/NM	PS/NS/NM	ZO/ZO/NS	NS/PS/NS	NS/PS/ZO
ZO	PM/NM/ZO	PM/NM/NS	PS/NS/NS	ZO/ZO/NS	NS/PS/NS	NM/PM/ZO	NM/PM/ZO
PS	PS/NS/ZO	PS/NS/ZO	ZO/ZO/ZO	NS/PS/ZO	NS/PS/ZO	NM/PM/ZO	NM/PB/PS
PM	ZO/ZO/PB	ZO/ZO/NS	ZO/PS/PS	NM/PS/PS	NM/PM/PS	NB/PB/PS	NB/PB/PB
PB	ZO/ZO/PB	ZO/ZO/PM	NS/PS/PM	NM/PM/PS	NM/PM/PS	NB/PB/PS	NB/PB/PB

The fuzzy control rules designed according to the adaptive principle are given in Table 1. The fuzzy control rules are written into the fuzzy logic designer to complete the construction of the fuzzy-based adaptive controller.

The output curved surfaces of ΔK_p, ΔK_i, and ΔK_d processed by the fuzzy controller are presented in Fig. 13. The output curved surfaces of ΔK_p, ΔK_i, and ΔK_d are displayed in Figs. 13(a), 13(b), and 13(c), respectively. The analysis of the graphic characteristics of the output surfaces in Fig. 13 is as follows. When EC or E increases, the overall trend of the output surface of ΔK_p changes from high to low, that of ΔK_i changes from low to high, and that of ΔK_d changes from high to low, and then again to high. It can be considered that the output graph changes of ΔK_p, ΔK_i, and ΔK_d meet the above adaptive principle and have obvious gradient distributions, which indicates that the fuzzy mapping from E and EC to ΔK_p, ΔK_i, and ΔK_d of the designed fuzzy control has a good match with the theoretical design.

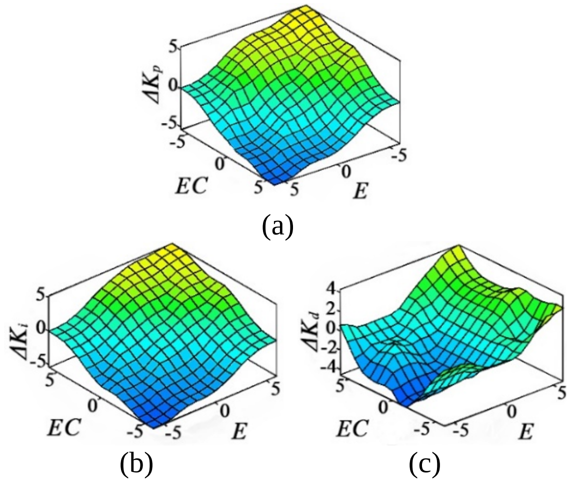


Fig.13. The output curved surface of ΔK_p, ΔK_i, and ΔK_d

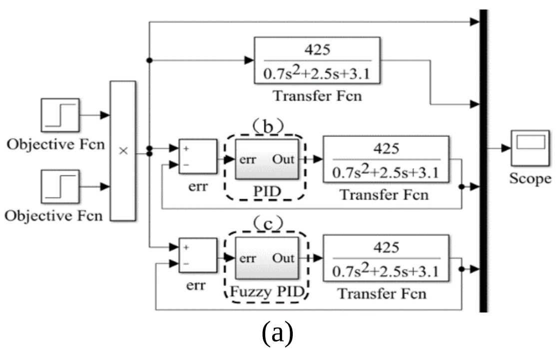
The related parameters of the motor of an intelligent vehicle are presented in Table 2. The motor speed transfer function can be expressed as:

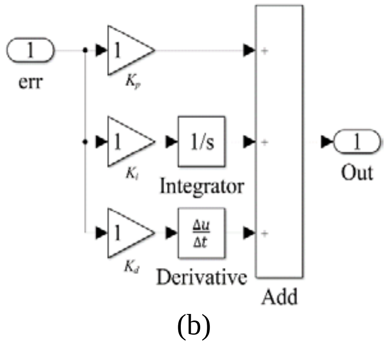
$$\begin{aligned} H(s) &= \frac{K_r}{L_a J s^2 + \left(L_a B + R_a J\right)s + K_e K_r + R_a B} \\ &= \frac{425}{0.7s^2 + 2.5s + 3.1} \end{aligned}. \quad (29)$$

Table 2. The related parameters of an intelligent vehicle

Motor	Armature inductance: L_a	0.19mH
	Armature resistance: R_a	0.23Ω
	Torque constant: K_r	4.25×10^{-3}N·m·A^{-1}
	Damping coefficient: B	8.5×10^{-5}N·m/(rad·s^{-1})
	Back-EMF coefficient: K_e	2.6×10^{-3}V/(rad·s^{-1})
	Total moment of inertia: J	3.8×10^{-5}kg·m^2

The simulation model of the control system realized on the Simulink platform is presented in Fig. 14. Fig. 14(a) shows the overall simulation model of the control system, Fig. 14(b) displays the simulation model of the traditional PID controller, and Fig. 14(c) presents the simulation model of the fuzzy-based PID controller.





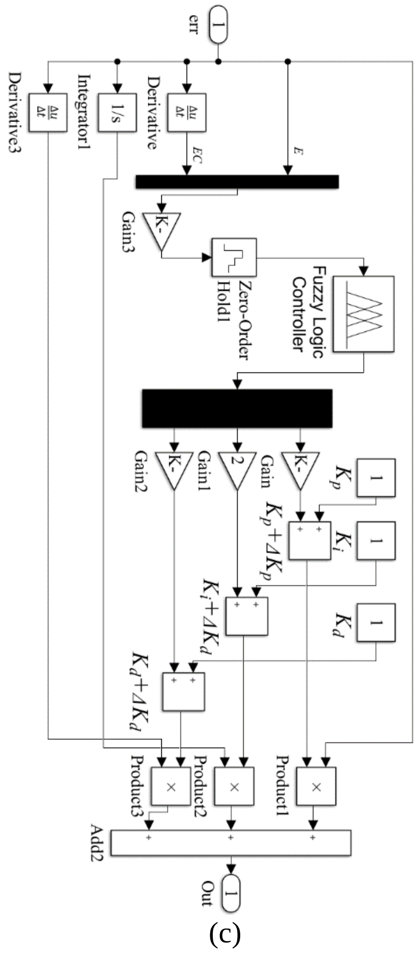


Fig.14. The simulation model of the control system

The simulation results of the proposed control system are presented in Fig. 15. As shown in Fig. 15, when the target speed increased from 0 rpm to 200 rpm at t = 1 s, the output curve of the no-control case reached the target speed at $t \approx 1.06$ s and then deviated from the target speed rapidly with the simulation time. The output curve of the PID control reached the target speed at $t \approx 2.2$ s, and the maximum excess was approximately 56.7 rpm, which became stable at the target speed after $t \approx 13.1$ s. The output curve of the fuzzy-based PID control reached the target speed at $t \approx 1.44$ s, and the maximum excess was approximately

16.4 rpm, which was stable at the target speed after $t \approx 8.9$ s. When the target speed increased from 200 rpm to 400 rpm at $t = 14$ s, the output curve of the PID control reached the target speed at $t \approx 15.2$ s, and the maximum excess was about 58.9 rpm, and it was stable at the target speed after $t \approx 26.1$ s. The output curve of the fuzzy-based PID control reached the target speed at $t \approx 14.44$ s, and the maximum excess was about 16.7 rpm, and it was stable at the target speed after $t \approx 22$ s. Compared with the no-control case, using the PID control could reduce the response speed, but it could make the control output curve quickly tend to the objective function. Compared with the PID control, the response speed of the output curve of the fuzzy-based PID control was significantly faster, the excess was smaller, the vibration amplitude was smaller, and it was faster and more stable than the target value, which significantly improved the control performance of the system.

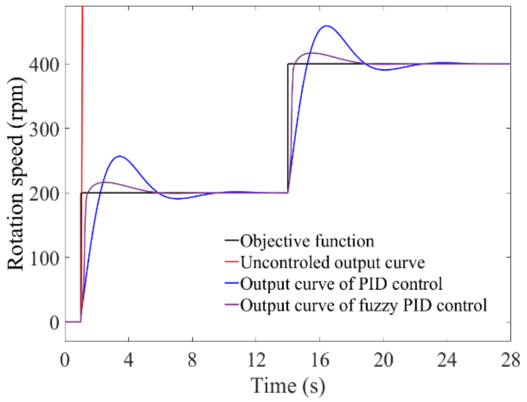


Fig.15. The simulation results of the control system

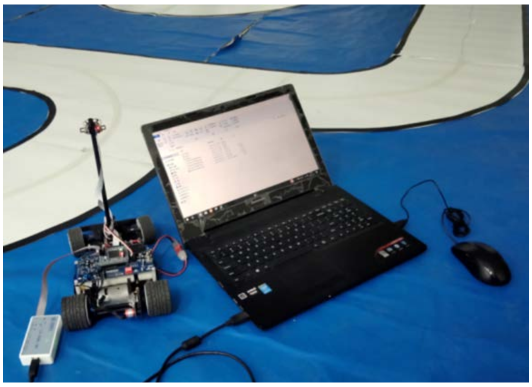


Fig.16. Experimental environment

The proposed control algorithm was experimentally verified on an intelligent vehicle. Programming and debugging were performed using a computer and IAR software, and the experiments were conducted under indoor lights in order to avoid interference from external environmental changes, and the experimental environment is shown in Figure 16. In the experiment, the smart cars with different algorithms are allowed to run 20 tracks, and the success rate of intelligent vehicles with different algorithms passing through the whole track is analyzed to highlight the advantages of the improved control algorithm. The experimental results are shown in Table 3.

Table 3. Success rate of the intelligent vehicle at different average vehicle speeds

Algorithm	Low speed (0.5 m/s)	Medium speed (1.5 m/s)	High speed (2.5 m/s)
PID control	95%	85%	70%
Fuzzy-based PID control	100%	95%	90%

As can be seen from Table 3, the higher the speed, the lower the success rate of the intelligent vehicle running a complete lap of the track; when the speed is around 2.5 m/s, the success rate of the intelligent vehicle with PID control is only 70%, while the success rate of the intelligent vehicle with fuzzy-based PID control is 90%, so the control algorithm proposed in this experiment has a better control effect compared with PID control algorithm.

CONCLUSIONS

An intelligent vehicle including a single camera is used as a research object in this study. The main contributions of this work are as follows:

(1). The algorithms and implementation effects of three basic threshold algorithms are discussed. The experimental results show that, compared with the other two thresholding algorithms, the Otsu thresholding algorithm achieves the best performance without using the recent lines of image data. Using the idea of image binarization and *k*NN algorithm, an improved filtering algorithm is designed. The image data collected by a smart car's camera show that the improved filtering algorithm can effectively reduce the noise.

(2). A T-shape edge extraction algorithm based on the actual situation of a smart car is proposed. The experimental results show that the improved algorithm can filter the abnormal data that may be collected by using the traditional algorithm and can avoid the misjudgment caused by external interference. The improved algorithm based on the track bending degree is proposed to judge the track conditions. The experimental results show that the sensitivity of the improved algorithm is more than eight times that of the traditional algorithm, and the improved algorithm can also overcome the problem that the performance of the traditional algorithm decreases for the high bending degree of a track.

(3). The path tracking is realized using the midline algorithm and edge algorithm, and the edge algorithm is simplified by the Lagrange interpolation. The experimental results show that using the simplified edge algorithm, an edge algorithm similar to the least square fitting algorithm with the R-squared greater than 0.994 can be obtained. The number of edge points used in the calculation is reduced from 48 to 2 or 3, which effectively reduces the amount of operation and improves the operation efficiency of intelligent vehicles.

(4). The fuzzy-based adaptive algorithm is proposed to update PID parameters in real-time. The simulation results show that after the target speed changes, the output curve of the traditional PID control algorithm reaches the target speed in 1.2 s, while the fuzzy-based PID control algorithm reaches the target speed in 0.44 s. The maximum excess of the traditional PID control algorithm is about 16.4 rpm, while that of the fuzzy-based PID control algorithm is 56.7 rpm. Also, the traditional PID control algorithm becomes stable in 12.1 s at the target speed, and the fuzzy-based PID control algorithm becomes stable after 7.9 s. Thus, the performance of the control system is obviously improved when the fuzzy-based PID control algorithm is used. The experimental results of the real intelligent vehicle showed that the proposed fuzzy-based PID control algorithm has a good control effect.

ACKNOWLEDGMENT

This entire study works are sustained by Fujian Industrial Technology Development and Application Project No. 2021I0024.

REFERENCES

Almagambetov A., Velipasalar S., and Casares M., “Robust and Computationally Lightweight Autonomous Tracking of Vehicle Taillights and Signal Detection by Embedded Smart Cameras,” IEEE Transactions on Industrial Electronics, Vol.62, No.6, pp. 3732-3741 (2015).

Bretti G. and Piccoli B., “A Tracking Algorithm for Car Paths on Road Networks,” Siam Journal on Applied Dynamical Systems, Vol.7, No.2, pp. 510-531 (2008).

Chen Y. H, Chen Y. Y., Huang J. M., “Guidance Law Design of Unmanned Robot Boats Based on Vision Navigation,” Journal of the Chinese Society of Mechanical Engineers, Vol.40, No.3, pp. 248-255 (2019).

Chen C. H., Lee C. R., and Lu W. C. H., “Smart in car camera system using mobile cloud computing framework for deep learning,” Vehicular Communications, Vol.10, No.10, pp. 84-90 (2017).

Chen G., Zhang W., Li X., and Yu B., “Adaptive speed control algorithm for electromagnetic direct drive vehicle robot driver based on fuzzy logic,” Measurement and Control, Vol. 52, No.10, pp. 1344-1353 (2019).

Chen X. W., Zhang J. G., and Liu Y. J., “Research on the Intelligent Control and Simulation of Automobile Cruise System Based on Fuzzy System,” Mathematical Problems in Engineering, Vol. 2016, No.5, pp. 1-12 (2016).

Drage T., Kalinowski J., and Braunl T., “Integration of Drive by Wire with Navigation Control for a Driverless Electric Race Car,” IEEE Intelligent Transportation Systems Magazine, Vol. 6, No.4, pp. 23-33 (2014).

Etlik U. B., Korkmaz B., Beke A., and Kumbasar T., “A fuzzy logic-based autonomous car control system for the JavaScript Racer game,” Transactions of the Institute of Measurement and Control, Vol. 43, No.5, pp. 1028-1038 (2019).

Fu R., Wang H., and Zhao W., “Dynamic driver fatigue detection using hidden Markov model in real driving condition,” Expert Systems with Applications, Vol. 63, No.11, pp. 397-411 (2016).

Guo J., Luo Y., and Li K., “Integrated adaptive dynamic surface car following control for nonholonomic autonomous electric vehicles,” Science China Technological Sciences, Vol. 60, No.8, pp. 1221-1230 (2017).

Han G., Fu W., Wang W., and Wu Z., “The Lateral Tracking Control for the Intelligent Vehicle Based on Adaptive PID Neural Network,” Sensors, vol. 17, no. 6, pp. 1244 (2017).

Jeng S. L., Chieng W. H., Wang Y. C., “Real-Time Heuristic Motion Planning for Autonomous Vehicle Driving,” Journal of the Chinese Society of Mechanical Engineers, Vol.42, No.2, pp. 187-196 (2021).

Jin X., Chen K., Zhao Y., “Simulation of hydraulic transplanting robot control system based on fuzzy PID controller,” Measurement, Vol. 163, pp. 108023 (2020).

Kabzan J., Hewing L., Liniger A., and Zeilinger M. N., “Learning Based Model Predictive Control for Autonomous Racing,” IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 3363-3370 (2019).

Konig M. and Neumayr L., “Users' resistance towards radical innovations: The case of the self-driving car,” Transportation Research Part F: Traffic Psychology and Behaviour, vol. 44, no. 1, pp. 42-52 (2017).

Li M. Q. and Liu Z. F., “Study on Navigation Line Detection in L. barbarum Garden Based on Genetic Algorithm,” Journal of the Chinese Society of Mechanical Engineers, vol. 40, no. 6, pp. 683-692 (2019).

Li H., Yang H., and Chen K., “Feature Point Extraction and Tracking Based on a Local Adaptive Threshold,” IEEE Access, vol. 8, no. 3, pp.

44325-44334 (2020).

Li Q., Zhou J., and Li B., "Robust Lane-Detection Method for Low-Speed Environments," Sensors, vol. 18, no. 12, pp. 4274 (2018).

Liu S., Hou Z., Tian T., Deng Z., and Li Z., "A Novel Dual Successive Projection-Based Model-Free Adaptive Control Method and Application to an Autonomous Car," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3444-3457 (2019).

Mohammadzadeh A. and Taghavifar H., "A robust fuzzy control approach for path-following control of autonomous vehicles," Soft Computing, vol. 24, no. 8, pp. 3223-3235 (2020).

Milanes V., Shladover S. E., Spring J., Nowakowski C., Kawazoe H., and Nakamura M., "Cooperative Adaptive Cruise Control in Real Traffic Situations," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 1, pp. 296-305 (2014).

Naranjo J. E., Gonzalez C., Garcia R., and Pedro T., "Lane-Change Fuzzy Control in Autonomous Vehicles for the Overtaking Maneuver," IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 3, pp. 438-450 (2008).

Rao Y. and Yang F., "Research on Path Tracking Algorithm of Autopilot Vehicle Based on Image Processing," International Journal of Pattern Recognition and Artificial Intelligence, vol. 34, no. 5 (2019).

Shi L., Hu Y., Su S., Guo S., Xing H., Hou X., and Liu Y., "A fuzzy PID algorithm for a novel miniature spherical robots with three-dimensional underwater motion control," Journal of Bionic Engineering, vol. 17, no. 5, pp. 959-969 (2020).

Senapati A., Maitra A., Mondal S., "Speed Control of Smart Car using Fuzzy Logic Controller," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 7, no. 3, pp. 1121-1129 (2018).

Useche S. A., Viviola G., and Cendales B. E., "Stress-related psychosocial factors at work, fatigue, and risky driving behavior in bus rapid transport (BRT) drivers," Accident Analysis & Prevention, vol.15, no. 12, pp. 335-339 (2017).

Xu X., Xu S., Jin L., and Song E., "Characteristic analysis of Otsu threshold and its applications," Pattern Recognition Letters, vol.32, no. 7, pp. 956-961 (2011).

Yu L., Yan X., Kuang Z., Chen B., and Zhao Y., "Driverless Bus Path Tracking Based on Fuzzy Pure Pursuit Control with a Front Axle Reference," Applied Sciences, vol.10, no. 1, pp. 230 (2019).

Zhang G., Yau K. K. W., Zhang X., and Li Y., "Traffic accidents involving fatigue driving and their extent of casualties," Accident Analysis & Prevention, vol.87, no. 2, pp. 34-42 (2016).

NOMENCLATURE

n_i	: The number of pixels with the gray value i
g	: The variance between the foreground and background
D_i	: The distance between the ith point and noise
$bend$	: The track bending degree
$right$	: The right edge data
$right1$	: The ideal straight track
y	: The tracking path
y_l	: The left edge
y_r	: The right edge

基於圖像識別的無人駕駛汽車軟體系統最佳化設計

李筆 張炳榮 易際明
廈門理工學院機械與汽車工程學院

邱亦睿
廈門理工學院機械與汽車工程學院
福建省客車先進設計與製造重點實驗室

翁才恩
三明學院機械工程學院

胡毓忠
國立宜蘭大學機械與機電工程學系

简賸瑞
義守大學材料科學與工程學系

摘 要

智能車輛是集環境感知、路徑決策規劃、自動駕駛等功能於一體的系統。為提高智能車輛的跟蹤和運動性能，設計並優化了包括圖像預處理、圖像處理、路徑跟蹤規劃和運動控製在內的智能車輛系統。首先，討論了三種基本閾值算法和圖像去噪算法的原理和實現效果。其次，對傳統的邊緣提取算法和路況識別算法進行了改進。然後，提出並簡化了基於中線算法和基於最小二乘算法的邊緣擬合算法的路徑跟蹤規劃方法。最後，針對傳統PID算法無法更新K_p、K_i和K_d值的缺點，提出了一種基於PID算法和模糊控製的智能車輛控製系統。實驗結果表明，所提出的濾波算法能有效地降低圖像噪聲。改進的邊緣提取算法對智能車運行過程中的異常數據具有明顯的濾波效果。改進的路況識別算法得到的直、彎路況差值為7.39，大於傳統算法得到的1.78，改進算法對軌道彎曲度變化敏感，克服了傳

統算法性能隨彎曲度變化而下降的問題。在簡化邊緣算法的基礎上，提出了一種基於最小二乘算法的邊緣擬合算法，該算法與R^2大於0.994的算法相似，計算所用的邊緣點數量由原來的48個點減少到2或3個點，大大提高了智能車的運行效率。使用模糊PID控製,在目標速度的變化後,輸出曲線在0.44s達到目標速度,最大超出量約16.4rpm, 目標速度在7.9秒後穩定,分別小於傳統PID的1.2s, 56.7rpm, 12.1s。因此，采用所提出的基於模糊PID控製算法，可以顯著提高系統的控製性能。真實智能車的實驗結果表明，所提出的模糊PID控製算法能顯著提高高速運行時的控製效果。