Artificial Life and Robotics (2023) 28:217–225
https://doi.org/10.1007/s10015-022-00812-4

ORIGINAL ARTICLE

GPS-based location and path tracking in automatic driving system in a fixed route using fusion algorithm

Yifei Dong[1] · Shigeyuki Tateno[1] · Harutoshi Ogai[1] · Yichen Wang[1] · Shuwei Zhang[1] · Yutian Wu[1]

Received: 14 May 2022 / Accepted: 20 September 2022 / Published online: 21 October 2022

Abstract
To achieve automatic transportation between factories, we make an automatic driving system based on fusion method and state machine that moves the car with a fixed route. Based on the GPS algorithm, we optimize the positioning algorithm by integrating the model calculation algorithm, and optimize the path tracking algorithm by integrating the lane keeping algorithm, which solves the problem caused by the unstable GPS signal to the system and enhances the system robustness. During the design process, we improve the algorithm, and design and optimize system functions with actual test conditions as feedback.

Keywords Location · Algorithm fusion · Path following · System robustness

1 Introduction

This is a research for practical applications. Therefore, in the research process, we have encountered and solved many problems that are not in theoretical research or simulation. This process also helps us understand the algorithm more deeply and optimize it.

During the development process, we gradually tried a variety of control methods, GPS tracking control, lane line keeping control, and magnetic track tracking control. Each control method has its advantages and disadvantages in different scenarios, but a single control method cannot always take into account all driving scenarios. After comprehensive testing and comparing the performance of each method, we finally used a mode that can be freely switched between different control methods according to the scene setting to give full play to the advantages of each control method. In addition, for various reasons, the feedback provided to the control system does not have sufficient accuracy, so many algorithms cannot achieve the performance in the simulation when measured on a real vehicle. We solved this problem to a large extent through the method of algorithm fusion.

The first problem we encounter is positioning. In simulation, we can always get the most accurate position information from the simulator, including the coordinates and yaw angle in a specific coordinate system. However, in practice, we need to obtain location information directly through sensors or calculations. Using lidar for SLAM or mapping can obtain more accurate location information, but unfortunately, we only have traditional sensors, such as RTKGPS and IMU. We can get absolute position via GPS, using the coordinates of the position at different time points can calculate a more accurate body orientation. However, due to the delay of GPS positioning, using only GPS to calculate the yaw angle will result in an output update rate that is too slow and has a large delay. Therefore, we chose to use a model-based calculation method to calculate the yaw angle and correct it based on the distance traveled by the vehicle using the GPS calculated yaw angle. This combination of calculation method and GPS eliminates the cumulative error of the calculation method, compensates for the discontinuity and delay of GPS, and obtains a relatively accurate yaw angle.

Then we also encountered problems in tracking control. At first, we decided to use GPS to track the waypoints recorded in advance. However, due to many objective factors, we cannot obtain stable GPS signals, such as indoors

This work was presented in part at the joint symposium of the 27th International Symposium on Artificial Life and Robotics, the 7th International Symposium on BioComplexity, and the 5th International Symposium on Swarm Behavior and Bio-Inspired Robotics (Online, January 25–27, 2022)

✉ Yifei Dong
dongyifei@akane.waseda.jp

[1] Graduate School of Information, Production and Systems, Waseda University, Shinjuku City, Japan

and under bridges where there is almost no signal, or when close to high-rise buildings, signal insufficiency leads to unstable location. To supplement the stability of the control, we added and integrated the lane line keeping control and the magnetic sensor-based track tracking control in the control algorithm, made evaluation criteria for the data of different sensors, and automatically adjusted the weights and control parameters under different conditions. To ensure that the system can stably control the vehicle to complete path tracking in more scenarios.

Different algorithms have their own advantages and disadvantages. A full understanding of each algorithm and the use of its advantages to complement each other are the main points of this research. Algorithm fusion is also a very effective and practical way to solve practical problems.

2 Location by fusion algorithm

Among the existing positioning methods, it is generally believed that the use of lidar for mapping is a method that can obtain position information stably and accurately. However, due to some objective factors, we did not use lidar in the project, but chose a cheaper GPS as the main sensor for positioning.

In the initial plan, we believe that RTKGPS can directly obtain real-time position coordinates and we can use two shorter time points to obtain two coordinates to calculate a fairly accurate yaw angle. However, the biggest problem we encountered after many tests is that when under a bridge or eaves, or close to a high-rise building, the GPS signal will become weak and disappear due to the obstruction of satellite signals in one direction or multiple directions. This will cause us to completely lose our own location information, and thus cannot perform GPS path tracking.

Figure 1 is a Google satellite map of the test section. There is a beam connecting the two workshops in the red circle in the picture. Figure 2 is the data (utm coordinate system) obtained by RTKGPS recorded during the test in the road section in Fig. 1. It can be seen from Fig. 2 that the closer to the beam, the greater the error of GPS due to signal deterioration. When it is directly below the beam, the GPS signal is completely lost and the current coordinate data cannot be obtained.

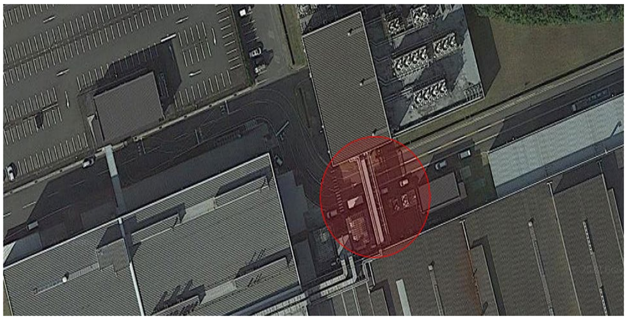


Fig. 1 Google satellite map of the test section

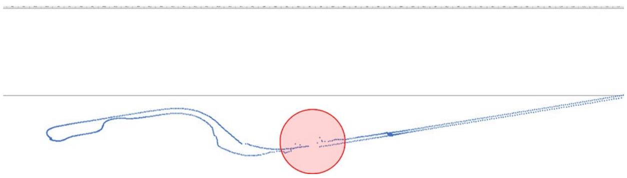


Fig. 2 GPS data in utm coordinate system

To solve the problem of GPS signal instability, we have added an algorithm for calculating the position through the vehicle model to assist RTKGPS positioning. In other words, the two methods are integrated, and each has its own advantages.

This chapter will first introduce the separate calculation logic of the two methods, and then introduce the fusion algorithm.

2.1 RTKGPS location

This time, we are using the open source RTKLIB software under the ubuntu16.04 environment. For ease of use and calculation, we convert the obtained positioning results from latitude and longitude to coordinates in the utm coordinate system. After that, the coordinate calculation of the entire system will be completed in the utm coordinate system.

After the coordinate conversion, we have obtained the coordinates in the position information, and the rest is the calculation of the yaw angle. In theory, we obtain the two coordinates of the vehicle at two moments, and when the two coordinates are infinitely close but not equal to zero, the direction from the coordinate at the previous moment to the coordinate at the next moment is the moving direction of the vehicle. For vehicles, it is approximately equal to the yaw angle.

The yaw angle calculation equation is Eq. (1), where x_t, y_t are the coordinates at time t in the utm coordinate system

$$Yaw_t = arctan\left[\frac{(y_{t_i} - y_{t_{i-1}})}{(x_{t_i} - y_{x_{i-1}})}\right]. \tag{1}$$

This method cannot calculate the yaw angle if the vehicle is not moving between two times. Therefore, we decided to use the distance traveled by the vehicle to determine the time point of obtaining the coordinates, rather than a fixed time. For example, whenever the vehicle travels b meters, we obtain the current coordinates and calculate the yaw angle with the coordinates before b meters. The schematic diagram is shown in Fig. 3.

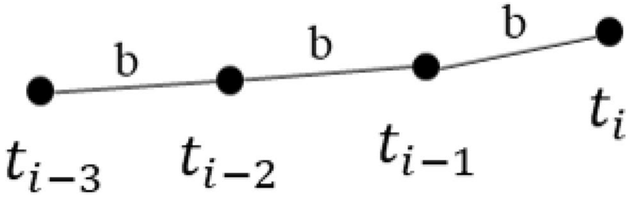


Fig. 3 Schematic diagram

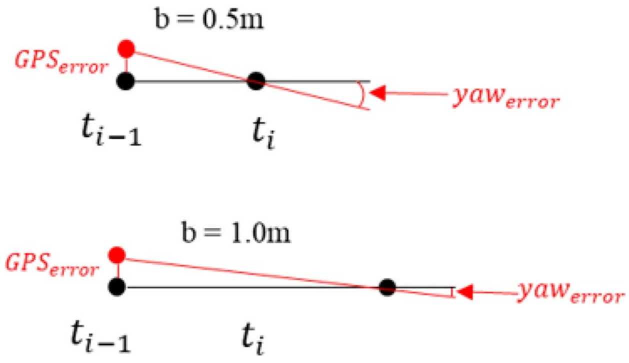


Fig. 4 Error comparison diagram

To calculate the traveled distance, we obtain speed data at a frequency of 50 Hz from the vehicle's canbus. And the driving distance of the vehicle is calculated by Eq. (2). In our case, the Δt is 0.02 s

$$\Delta s = V * \Delta t. \tag{2}$$

In this way, whether the vehicle is in a stopped state or a driving state at different speeds, the yaw angle can be calculated and updated with the accuracy of b meters. In theory, the closer b is to zero, the more accurate the yaw angle, but in fact, we can see from Fig. 4, due to the error of the GPS signal, the smaller the b, the greater the error of the yaw angle will be. However, the larger the setting of b, the slower the update frequency of the yaw angle, as the feedback information of the control system, which will cause the control system to lag.

Since the error of GPS is largely uncontrollable, we choose to solve the problem of discontinuous yaw angle between b meters. The next section is an introduction to how we solve this problem through fusion algorithms.

2.2 Fusion method location

Through the two-wheel kinematics model of the vehicle [1], we can use Eq. (3). to calculate the yaw angle change during the driving process of the vehicle (Figs. 5, 6).

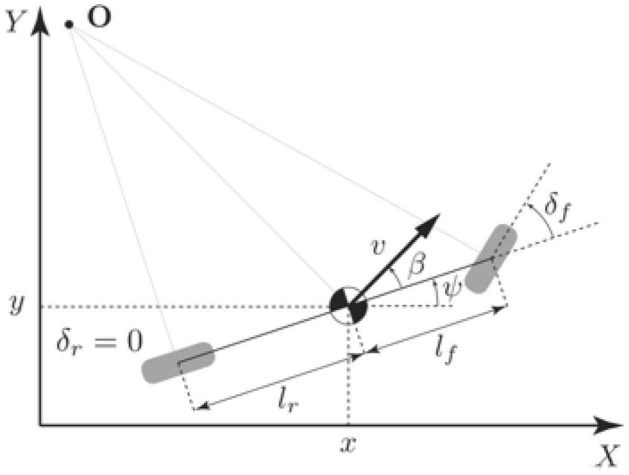


Fig. 5 Vehicle two-wheel kinematics model

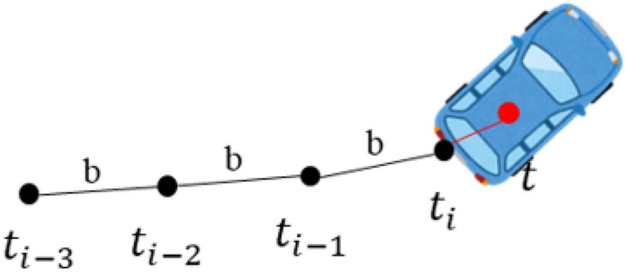


Fig. 6 Schematic diagram

In Eq. (3), β is the speed direction and it is approximately equal to the steering angle of the wheel, l_r is front wheelbase, and v_t is speed

$$\psi_{t+1} = \psi_t + \int_t^{(t+1)} \frac{v_t}{l_r} sin(\beta). \tag{3}$$

When this method calculates the yaw angle, the vehicle speed and wheel steering angle in the integral function are real-time feedback data through sensors. There are certain errors and discontinuities, which cause some data to be lost after integration. As the amount of feedback changes increases, and as time increases, the cumulative error in the result becomes larger.

The yaw angle equation calculated between b meters is added as Eq. (4)

$$Yaw_t = arctan\left[\frac{(y_{t_i} - y_{t_{i-1}})}{(x_{t_i} - y_{x_{i-1}})}\right] + (\psi_t - \psi_{t_i}). \tag{4}$$

Combining Eqs. (3) and (4), the final yaw angle calculation [Eq. (5)] can be expressed as follows:

$$Yaw_t = arctan\left[\frac{(y_{t_i} - y_{t_{i-1}})}{(x_{t_i} - y_{x_{i-1}})}\right] + \int_{t_i}^{t} \frac{v_t}{l_r} sin(\beta). \tag{5}$$

The result of the current yaw angle is only related to the result calculated by the GPS coordinates at the last time t_i and the result calculated by the kinematics model from the time t_i to the current time t. When the car travels more than b meters, the GPS calculation part will calculate the new yaw angle, and the model calculation part will be cleared to recalculate the change.

From a GPS-based perspective, we use the results of the model calculation to supplement the gap before the coordinate calculation updates. From the perspective of model calculation, we use coordinate calculation regularly as feedback, eliminating the cumulative error of model calculation.

So far, we have calculated a relatively stable yaw angle, but in actual use, the problem of GPS loss in a short time has not been resolved. The next section will introduce how we deal with the problem of GPS signal loss.

2.3 Mode switching in the fusion method

Before finding a way to deal with the problem of GPS signal loss, we first need an evaluation method to judge whether the current GPS signal is stable and usable, and then, we can distinguish between normal and unavailable GPS signals.

Every time the vehicle travels b meters, we obtain the coordinates before and after b meters. By calculating the distance between the two coordinates, the theoretical result should be b meters. Therefore, when the result is not equal to b meters, it means that there is a problem with the GPS signal. When the result is less than b meters, it may be that the signal stays in front of b meters after the GPS signal is interrupted; when the result is greater than b meters, it may be that the GPS signal is interfered and caused a large error. According to this logic, we designed a GPS abnormal signal "G", the logic is shown in Eq. (6)

$$G = \begin{cases} \sqrt{[(y_{t_i} - y_{t_{i-1}})^2 + (y_{t_i} - y_{t_{i-1}})^2]} < 0.75 * b & G = 1 \\ \sqrt{[(x_{t_i} - x_{t_{i-1}})^2 + (x_{t_i} - x_{t_{i-1}})^2]} < 0.75 * b & G = 1 \\ else & G = 0. \end{cases} \tag{6}$$

We design a quarter of b meter as the allowable error. In fact, the allowable error needs to be adjusted according to the equipment used and experience. In this article, the RTK-GPS which we use has a theoretical maximum error of 0.1 m when obtaining a fix solution, and we generally set b to 0.5 m during normal testing, so we think that when the error of a point exceeds a quarter of b (0.125 m), the current GPS signal is judged to be unavailable, and the output G is equal to 1.

When $G = 1$, we still set a conservative distance Dis, and expect the vehicle to continue driving and wait for the GPS signal to recover. When driving in this conservative distance, the positioning information is only used by the kinematics model to continue to calculate the yaw angle change. It can be understood that the b meter in the GPS calculation is set to a longer Dis at this time. When the driving distance exceeds Dis and G is still equal to 1, we believe that the current system has no reliable positioning information, to ensure safe execution of the stop command.

Figure 7 is the state flowchart. In the figure, D_t is the total distance traveled by the vehicle at time t.

After obtaining the location information based on the above method, the next chapter introduces the path tracking control algorithm based on GPS and lane detection.

3 Path following by GPS and lane

We initially selected the GPS path following algorithm. The actual test has the same problem as the GPS positioning, the loss of the GPS signal. Therefore, we added lane line keeping control to assist GPS path following control.

This chapter first introduces the GPS path following algorithm, and then introduces the following algorithm of fusion lane line keeping control.

3.1 GPS path following control

We are a fixed route following control, so we manually drive to record the GPS coordinates of the complete route beforehand as a reference route.

The logic of the follow-up control is to select a suitable coordinate point from the reference path as the target point according to the coordinates of the vehicle position. The first is the target point's selection.

We use the vehicle's own coordinates as the center of the circle and set a search range with a radius of 5 m. Because the target point needs to be in front of the car, we start from the last coordinate point of the reference path and calculate

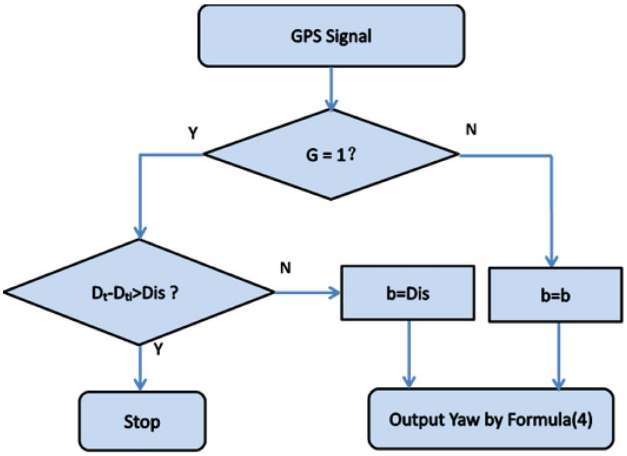


Fig. 7 Yaw output flowchart

the distance between each coordinate point and the vehicle position in turn. When the result of the first distance is less than 5 m, set this point as the current target point, and output the number of the point in the reference path "*No.tp*". Figure 8 is a schematic diagram.

Based on the target point coordinates $(x_{No.tp}, y_{No.tp})$ and the vehicle's own coordinates (x_t, y_t), with the vehicle yaw angle *Yaw*, we can calculate the target angle θ (Eq. (7)).

$$\theta = Yaw_t - arctan\left[\frac{(y_{No.tp} - y_t)}{(x_{No.tp} - x_t)}\right]. \tag{7}$$

Figure 9 is a schematic diagram of Eq. (7).

Finally, we use the target angle as the input for PID control, and the wheel angle as the output for steering control. And, according to the number of the target point, we distinguished the curve and the straight in advance, and set different target speeds.

3.2 Fusion with lane keeping control

Similarly, when there is a problem with the GPS signal, we added a lane line recognition detection based on deep learning [2], and made lane line keeping control based on the detection results.

This paper proposes a deep convolution neural network-based lane detection method, which consider the lane detection task as a pixel level segmentation of the lane markings. It also propose an automatic training data generating method, which can significantly reduce the effort of the training phase.

After image processing, we can get the coordinates of the lane line in the image. Then, use the difference between this coordinate and the target position (red line) in the image to correct the steering angle to achieve lane keeping.

First, we use the camera to record some target lane line material videos. Then, cut out the picture from the material videos and train the model after calibrating the lane lines in the picture. Finally, use the model to identify the lane lines in the video from the camera in real time.

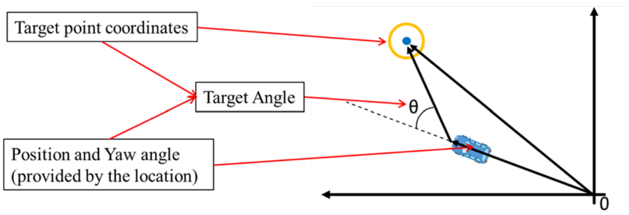


Fig. 9 Schematic diagram

In Fig. 10, the length and width are pixels of the screen, with 1 pixel as 1 unit, and the lower left corner of the screen is the far point to establish a coordinate system. The target position is measured in advance.

Lane line keeping can also be used as a separate control, but because there is no area information, it is impossible to accurately distinguish between a curve and a straight road. It is conservative to keep driving at a low speed during its independent operation. When the GPS signal is unstable, a certain weight of the angle of the lane line control output is added to the total angle output. When GPS signal is interrupted, switch to separate lane line control and reduce driving speed.

Figure 11 is a flowchart in the fusion algorithm. θ_{keep} is the angle output of the lane line keeping control, and θ_{GPS} is the angle output of the gpg path following control.

4 Experiment and result

To visually show the difference between the fusion algorithm and the separate algorithm, we run the fusion algorithm and the separate GPS algorithm at the same time during the test, and obtain the yaw angle calculation results of the two algorithms at the same time for comparison. And record the coordinate data in the utm coordinate system

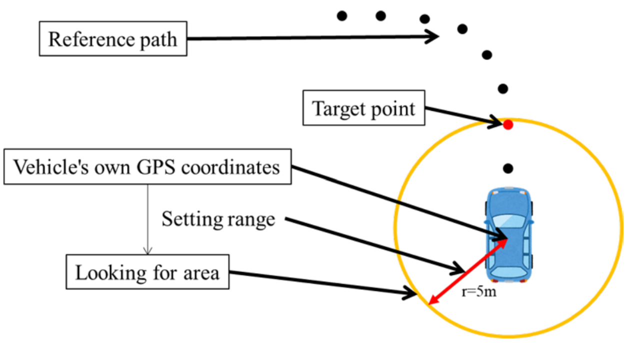


Fig. 8 Schematic diagram

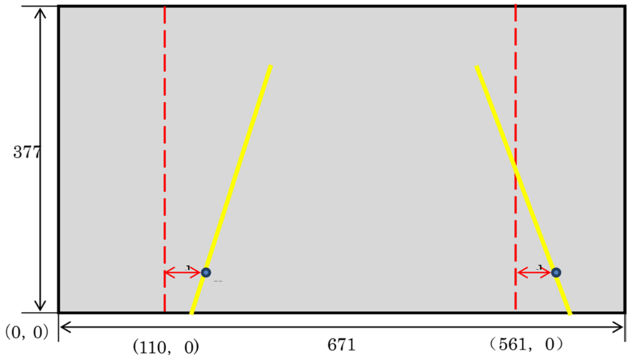


Fig. 10 Lane detection result

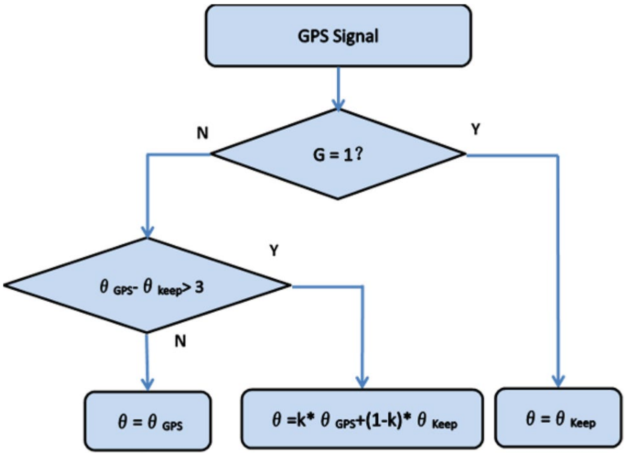


Fig. 11 Angle output flowchart

directly obtained by GPS. It is convenient to see the degree of loss of GPS data.

In the test, the parameter b is set to 0.5 m. To see the GPS signal condition through the density, the test vehicle speed is set to a constant speed of 5km/h throughout the whole process.

Figure 12 shows the result of test1. The blue point in Fig. 12 is the yaw angle calculated by the fusion algorithm, and the orange point is the yaw angle result of the GPS algorithm. It can be seen from Fig. 13 that the GPS signal is basically stable throughout the whole process. The results of fusion algorithm and GPS algorithm are basically the same in GPS steady state.

Figure 14 shows the results of test2. It can be seen from Fig. 15 that the GPS signal of some road sections is seriously lost during the test process. From Fig. 14, it can be seen that the GPS algorithm of the corresponding road section cannot calculate the yaw angle result, and the output is 0. The fusion algorithm can still calculate the yaw angle of the entire road section normally.

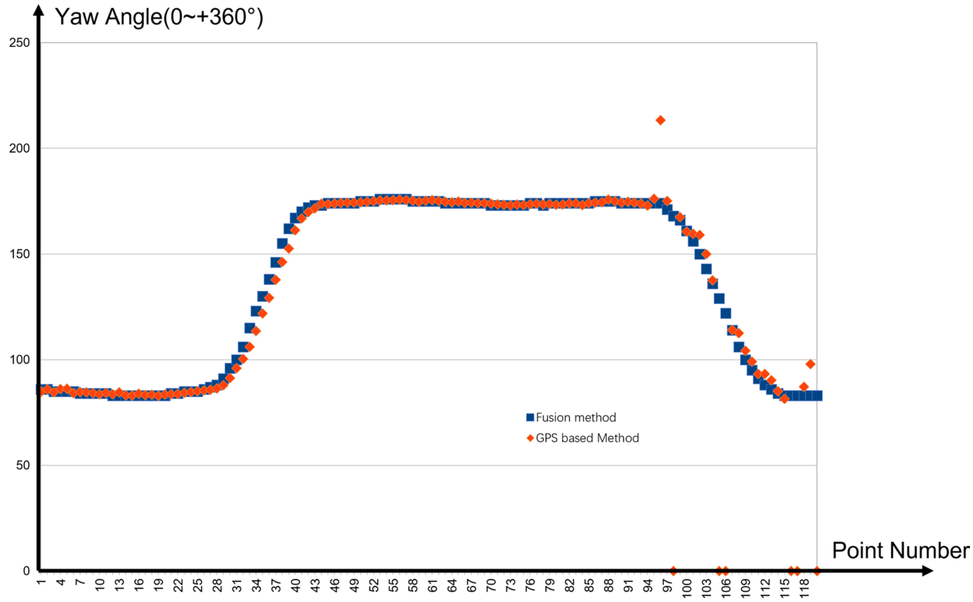
Figure 17 shows the results of test3. In this test, we added a curve. From the results in the Fig. 16, it can be seen that the fusion algorithm can basically maintain the stability of the whole process when the continuous steering is performed (Fig. 17).

5 Conclusion

Different algorithms have advantages and disadvantages. A good understanding of each algorithm and exploiting its strengths complement each other is the focus of this study. Algorithm fusion is also a very effective and practical method to solve practical problems. We use the vehicle model calculation method to fill in the instability of the GPS algorithm, and make a robust positioning algorithm. At the same time, it can be regarded as the accuracy of the GPS to eliminate the cumulative error of the model calculation algorithm.

Real vehicle testing is different from simulation testing. Often, you cannot get the same accurate feedback as the simulator, and many unexpected situations will occur. To deal with these issues, we need to think about more than just the simulation process. This research was born when simulation algorithms were put into practical use. In the simulated environment, we do not need to consider the problem of signal error or loss, or the misrecognition in

Fig. 12 Yaw calculation result (test1)



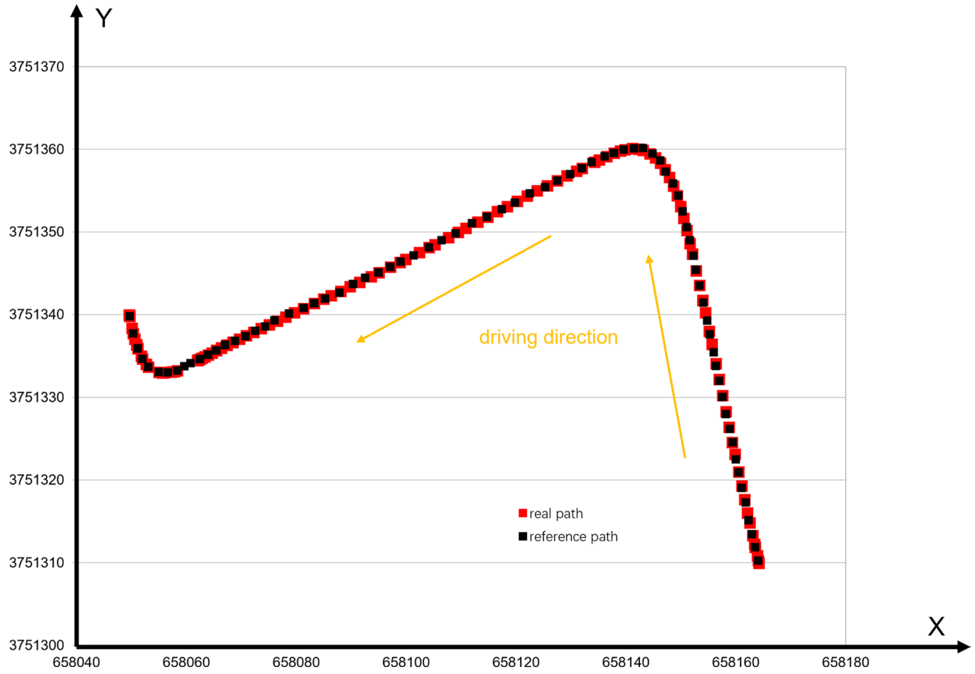


Fig. 13 GPS data in utm coordinate (test1)

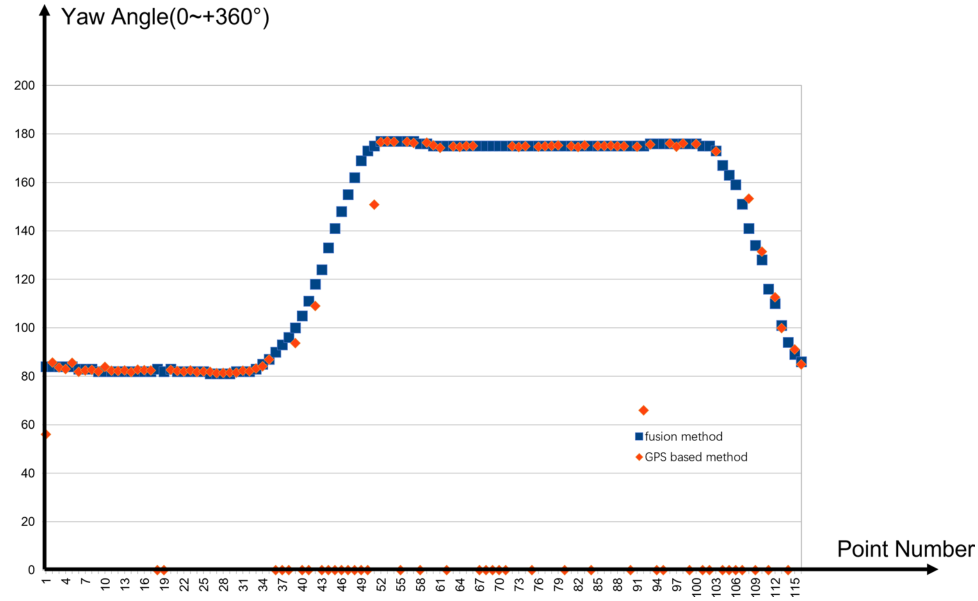


Fig. 14 Yaw calculation result (test2)

the recognition algorithm and the inability to recognize due to external factors such as backlight.

We think how to enhance the robustness of the system in practical applications is a very worthy question. In the follow-up research, we will also try to add sensors such as radar to the system, expand the function of the system to adapt to more road conditions, and try to run the system on non-fixed routes.

Fig. 15 GPS data in utm coordinate (test2)

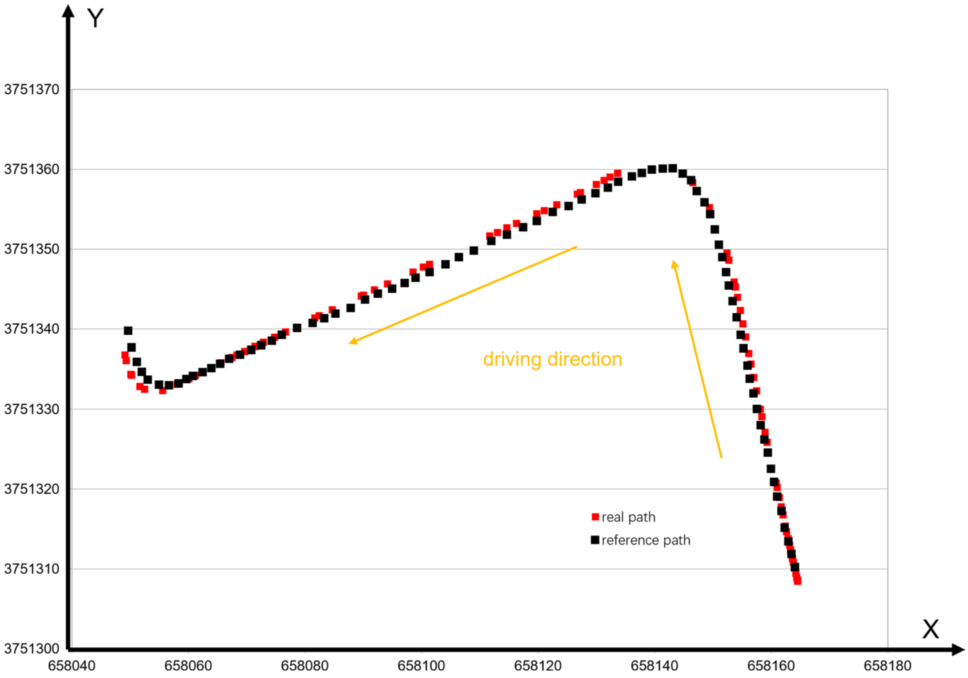


Fig. 16 Yaw calculation result (test3)

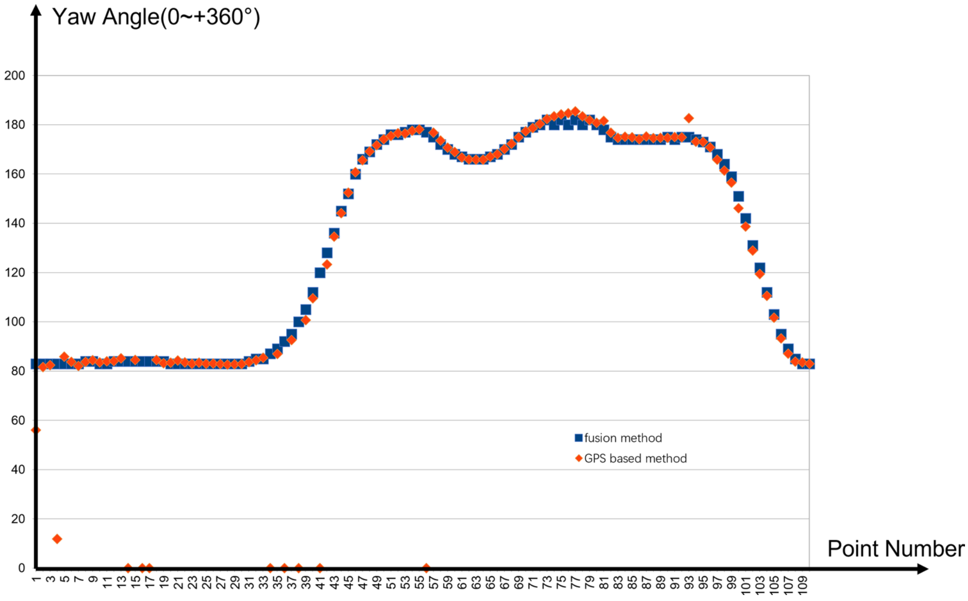
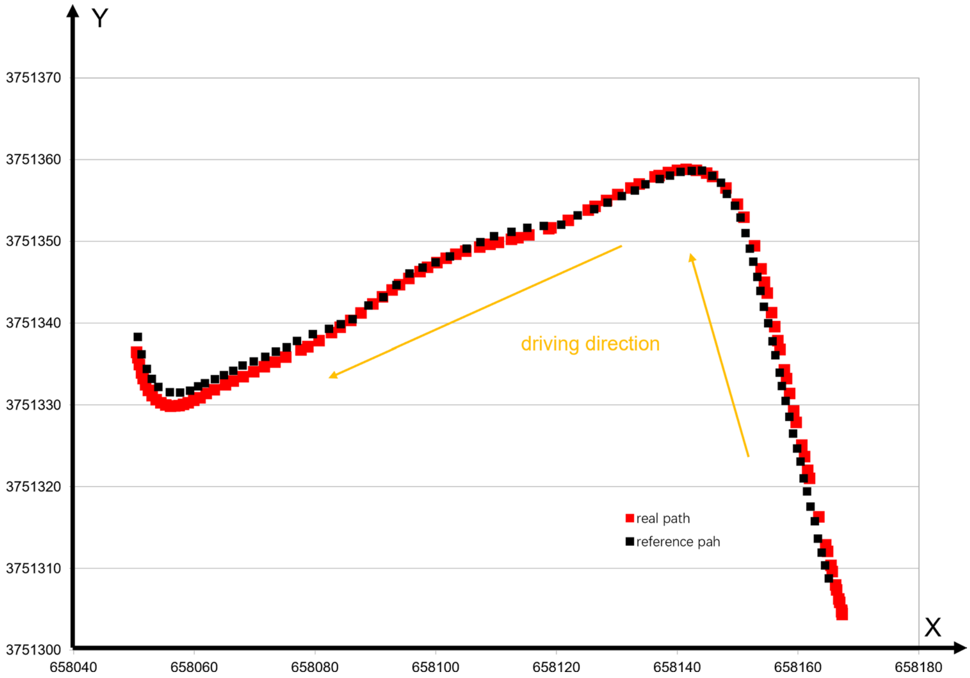


Fig. 17 GPS data in utm coordinate (test3)



References

1. Jianwei G, Yan J, Wei X (2014) Model predictive control of driverless vehicles. Beijing Institute of Technology Press, Beijing
2. Xun P, Harutoshi O (2019) Fast lane detection based on deep convolutional neural network and automatic training data labeling [J]. IEICE Trans Fundam Electron Commun Comput Sci 102(3):566–575

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.