Design and Implementation of Autonomous Path Planning for Intelligent Vehicle

Wei Wei[1], Fan Gao[1], Rafał Scherer[2], Robertas Damasevicius[3], Dawid Połap[4]

[1] School of Computer Science and Engineering, Xi'an University of Technology, China
[2] Czestochowa University of Technology Al, Poland
[3] Multimedia Engineering Department, Kaunas University of Technology, Lithuania
[4] Faculty of Applied Mathematics, Silesian University of Technology, Poland
weiwei@xaut.edu.cn, gf502148677@163.com, rafal.schererg@pcz.pl, robertas.damasevicius@ktu.lt, Dawid.Polap@polsl.pl

Abstract

With the development of artificial intelligence, the position of intelligent car in social life has become particularly important. The autonomous path of the smart car is divided into long distance and short distance. In order to improve the autonomous path planning ability of the smart car, we use WIFI to control the mobile end of the smart car, carry out autonomous tracking and autonomous obstacle avoidance in the short distance, and realize the optimization of the shortest path algorithm based on Dijkstra algorithm in the long distance. Based on the Raspberry Pi development board, the environment is installed and configured, the infrared sensor is used as the ranging device, and the optical code plate is used as the directional sensor of the car body bearing reckoning, and the motion control model of the smart car is established. The artificial potential field path planning method is used to realize the autonomous obstacle avoidance, optimal path searching and mobile terminal control of the intelligent car. The experimental results show that our design is efficient. And it has practical application value for fire truck path planning, remote control system, etc.

Keywords: Autonomous obstacle avoidance, Autonomous path planning

1 Introduction

The main algorithms for path planning at home and abroad are as follows artificial potential field method, fuzzy logic algorithm, ant colony algorithm, neural network algorithm, genetic algorithm and A* algorithm.

Artificial potential field method [1] is a robot path planning algorithm proposed by Khatib, and it is a virtual force method. Its advantages are smooth and safe, but there is a local optimal problem. Fuzzy logic algorithm [2] is a planning algorithm based on human prior experience. It simulates the driver's driving experience, combines his physiological perception with action, and obtains the planning information by looking up the table according to the sensor information implemented by the system, so as to realize path planning. Ant colony algorithm [3] is a probabilistic algorithm used to find optimal paths. In 1995, Glasius R applied the Hopfield-type [4] topological organization neural network model with nonlinear simulated neurons to the path planning and obstacle avoidance system. The system can quickly provide an appropriate path from any initial position to any target position, and can avoid static and moving obstacles of any shape. Ismail used genetic algorithm to carry out optimal path planning [5] for mobile robots. Compared with the traditional method based on gradient search, the genetic algorithm can adapt to the complex search space, and verified the feasibility of the algorithm. Ferguson D proposed an extended D* algorithm based on A* algorithm [6], which could effectively plan A* more optimized path with linear interpolation and involve less unnecessary turning, thus solving the problem of low path performance of grid generation.

Our main research is the autonomous path planning [7, 13-19] of intelligent car, Autonomous tracking and obstacle avoidance. In this paper, the second chapter introduces the algorithm principle of each module. The third chapter is our design process. The fourth chapter is the experimental results. The fifth chapter is the conclusion.

2 Intelligent Car Working Principle

2.1 The Working Principle of Mobile Terminal Control Module

The main purpose of the mobile terminal control module is to control the movement of the smart car by connecting to the TCP server built on Raspberry Pi at

*Corresponding Author: Wei Wei; E-mail: weiwei@xaut.edu.cn
DOI: 10.53106/160792642021092205002

the client end or the client APP. In reality, there may be one or more clients connected to the TCP server. Therefore, in this module, we will design a TCP concurrent server for multiple clients to connect, in order to avoid conflicts when multiple clients connect to the TCP server at the same time [20-29].

The data flow diagram of this module is shown in Figure 1. After the server builds the TCP concurrent server and starts the service, the client initiates a request connection to the server, and the server is in the listening state at this time. When the client requests the IP address and port number corresponding to the server, the server will accept the client's connection; At this time, both the client and the server are in the state of sending and receiving data. After the client sends data to the server through the TCP protocol, the server will process the data sent layer by layer, and get the final result. Then, by driving commands to the motor, it can control the movement of the car.

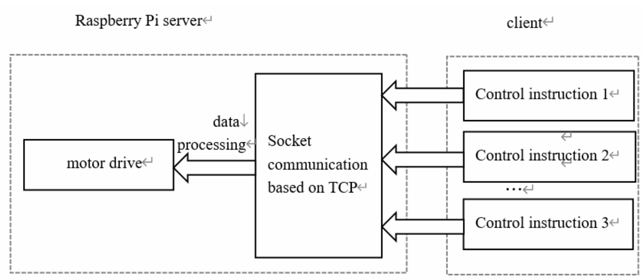


Figure 1. Module data flow diagram

When multiple clients send data to the server, the server will give each client a different address, and process the data sent by each client respectively, in order to control the movement of the car.

The TCP concurrent server ensures the problem of multi-client operation of the car movement. Compared with the general single-threaded server, it has faster response speed and stronger data processing ability.

2.2 The Working Principle of Tracking Module

In this paper, the tracking module we use is TCRT5000, which has a small volume and high sensitivity, and can also adjust the detection range by rotating the above potentiometer. TCRT5000 sensor of infrared emitting diode to continually emit infrared light to the world, when out of the Ir is not reflected by the external environment, the intensity of reflected back or but is is not big enough, the light activated triode will always is in a state of shut off, the output end of the module at this time for low level, indicating diode would have been out of state;

When the detected object appears in the detected range, the infrared ray will be reflected back and when the intensity is large enough, the photosensitive triode will produce saturation, the output end of the module will be high level, indicating diode will be lit. Because black has a strong absorption ability, so when the infrared ray emitted by the infrared tracking sensor irradiates the black line, the black line will absorb the infrared ray, which leads to the photosensitive transistor on the tracking sensor uniformly in the closed state, at this time, an LED on the sensor will be extinguished.When no black line is detected, the two LED on the sensor remain lit [30-37].

The module uses three tracking sensors to detect the position of the black line and send back data to the Raspberry Pi, so that the car can move along the path of the black line. The three tracking sensors are divided into left, center and right. The infrared sensor on the left is used to detect whether there is no black line shadow on the left side of the black line. If the black line shadow is detected on the left side, the car will turn left until the black line shadow is not detected. An infrared sensor in the center detects black line shadows; The infrared sensor on the right is used to detect whether there is no black line shadow on the right side of the black line. If the black line shadow is detected on the right side, the car will turn to the right until the black line shadow is not detected. The motion stops when all three sensors detect the black line shadow.

2.3 The Working Principle Obstacle Avoidance Module

Infrared obstacle avoidance method use a transmitting tube and a receiving tube, receiving tube is used to detect the external infrared receiving strength to judge the distance from the obstacle. Since the external visible light has a great influence on the infrared, the 250Hz signal is used to modulate the 38KHz carrier in order to reduce some external interference. The receiving tube output TTL level, for the MCU signal processing. Using the principle of infrared emission and reception, the emitted infrared rays will be reflected back when they encounter obstacles, and the reflected signal will be demodulated by the infrared receiving tube, and finally the TTL level will be output.

The infrared obstacle avoidance sensor uses the reflection principle of light. The front end of the sensor has two infrared tubes, one is an infrared transmitting tube, another is an infrared receiving tube. After the sensor is powered on, the infrared emitting tube will continuously emit infrared rays of a certain frequency to the front. When the infrared rays encounter obstacles in front, the infrared rays will return and be received by the receiving tube. At this time, OUT will output a low level. If there is no obstacle ahead and the infrared rays are not reflected back, OUT will output a high level.

In this paper, the module uses two obstacle avoidance sensors, through the detection of the location of obstacles, send back data to the raspberry, so that the car can avoid obstacles smoothly. Two obstacle avoidance sensors are located on the left and right side of the car respectively. The infrared sensor on the left is used to detect whether there is an obstacle on the left side. If the obstacle is detected on the left side, the car will turn right. The sensor on the right is similar to that on the left. The infrared sensor on the right is used to detect whether there is an obstacle on the right. If an obstacle is detected on the right, the car will turn left. When obstacles are detected on both sides, the car will move forward smoothly. If obstacles are detected on both sides, the car will stop, move backwards for a certain position, and then turn left.

3 Autonomous Programming Algorithm

Intelligent car autonomous path planning is the most important work in this paper. In the optimal path planning for the car, it is necessary to find the shortest distance between two points. In this paper, we use Dijkstra [8-9] algorithm to optimize the global path [10-12].

In the autonomous path planning of the intelligent car, on the one hand, the optimal path that the car needs to pass should be calculated; on the other hand, when the car actually passes these paths, it needs to rotate a certain Angle to move in the right direction. Therefore, this module involves two algorithm problems, one is Dijkstra algorithm, another is the turning Angle problem.

3.1 Dijkstra Algorithm

The basic principle of Dijkstra's algorithm is that each time the shortest point is expanded, the distance of its adjacent points is updated. When all edge weights are positive, there will not be an unextended point with a shorter distance, so the distance of this point will never be changed again, thus ensuring the correctness of the algorithm. However, according to this principle, the graph with the shortest circuit obtained by Dijkstra cannot have a negative weighted edge, because the extension to the negative weighted edge will produce a shorter distance, which may destroy the property that the distance of the updated points will not change.

Suppose that each point has a pair of labels (w_j, p_j), where w_j is the length of the shortest path from the point of origin S to the point J (the shortest path from the vertex to itself is zero path (the path without an arc), and its length is equal to zero); p_j is the point before j in the shortest path from s to j. The basic process of solving the shortest path algorithm from the origin point S to the point j is as follows:

(1) Initialization. The source node is set as:

(a) w_s = 0, p_s is null;

(b) All other points: w_i =∞, p_i = ?;

(c) mark the source node s, k= s, and all other points are set as unmarked.

(2) Check the distance from all marked point k to its directly connected unmarked point j, and set:

$$w_j = \min[w_j, w_k + d_{\{kj\}}] \quad \textbf{(1)}$$

In the formula, d{kj} is the direct connection distance from point k to j.

(3) Pick the next point. From all unmarked nodes, select the smallest i in w_j:

$$\text{w\_i} = [\text{w\_j, all unmarked points j}] \quad \textbf{(2)}$$

Point i is then selected as a point in the shortest path and set to marked.

(4) Find the point before point i. From the marked points, find the point j* directly connected to point I, as the previous point, and set: i = j*.

(5) Mark point i. If all the points are marked, the algorithm is fully deduced; otherwise, k= i, go to 2) and continue.

3.2 Steering Angle Problem Algorithm

The basic process of car steering Angle calculation is as follows:

(1) Because the starting point of the car is different from other points and has its own particularity, the starting point is calculated separately from other points here. Two coordinate points are used to calculate the included Angle of the starting point, and three points are used to calculate the remaining points. The Angle between the starting point: The Angle between the starting point is calculated using the starting point coordinates (x1, y1) and the coordinates of the next point to be reached (x2,y2). The starting point, the next point that is about to arrive and the X axis can form a right triangle. The tangent value of this Angle, Tanseta, can be obtained by using the starting point and the point that is about to arrive, and the included Angle, Seta, can be obtained by using the arctangent function:

$$tanseta = (y2 - y1)/(x2 - x1) \quad \textbf{(3)}$$

$$seta = arctan(transeta) \quad \textbf{(4)}$$

The Angle between other points: The Angle between other points is calculated using the previous point (x1, y1), the current point (x2,y2), and the next point to pass through (x3,y3). After passing the starting point, the coordinate axis changes with the direction of the car's movement, so the Angle of the current point in the irregular triangle surrounded by these three points is calculated, that is, the Angle is obtained by using the law of inverse cosines, and then the Angle is subtracted by 180 degrees to get the desired turning Angle:

$$seta = arccos(*b * b - a * a - c * c)/(-2 * a * c)) \quad \textbf{(5)}$$

Where, a, b, and c are the lengths of the sides corresponding to the three points in the triangle.

(2) Calculate the direction of car rotation (turn left or turn right)

If the y-coordinate of the current point is greater than the y-coordinate of the next approaching point, turn right. Otherwise, turn left.

(3) Calculate the time required for car rotation: According to the solution of rotation Angle, car speed and car radius (r=14.2cm) to calculate the car rotation time, using the following formula:

$$time = seta * r / v \quad \textbf{(6)}$$

4 Software Design

4.1 Server Software Design

After the smart car is powered on, the Raspberry Pi server opens the service function to the outside, constantly listens for the operation instructions sent by the client, and controls the car to move. This module is developed with C language and adopts a structured design method. Its software flow design is shown in Figure 2.

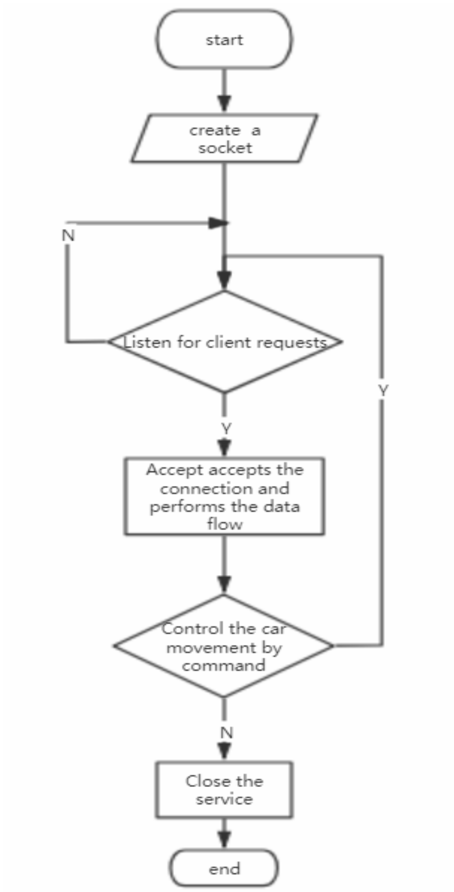


Figure 2. Server software flow design

The realization of the module is first on the server to create a streaming socket, through the bind function will be the server's IP address and port to bind to a socket file descriptors, and then use the listen function to monitor whether the client sends a connection request, and if listening to the client connection request, and then through the accept function to receive the client's connection, through the recv and the send function can transmit or receive data, corresponding data processing, so as to control the smart car in motion.

At the same time, this module uses multi-thread thinking to process, and realizes multi-client control of the smart car at the same time.

4.2 Client Software Design

The flow chart of the client software of the mobile terminal control module is shown in Figure 3.

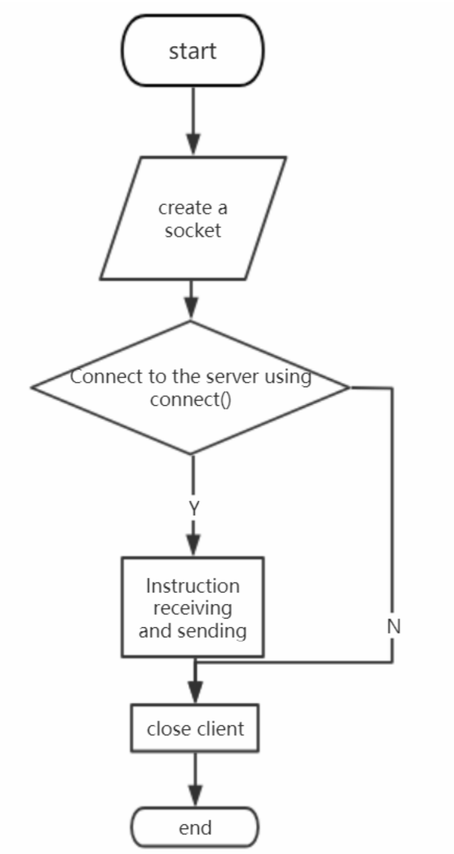


Figure 3. Client software flow

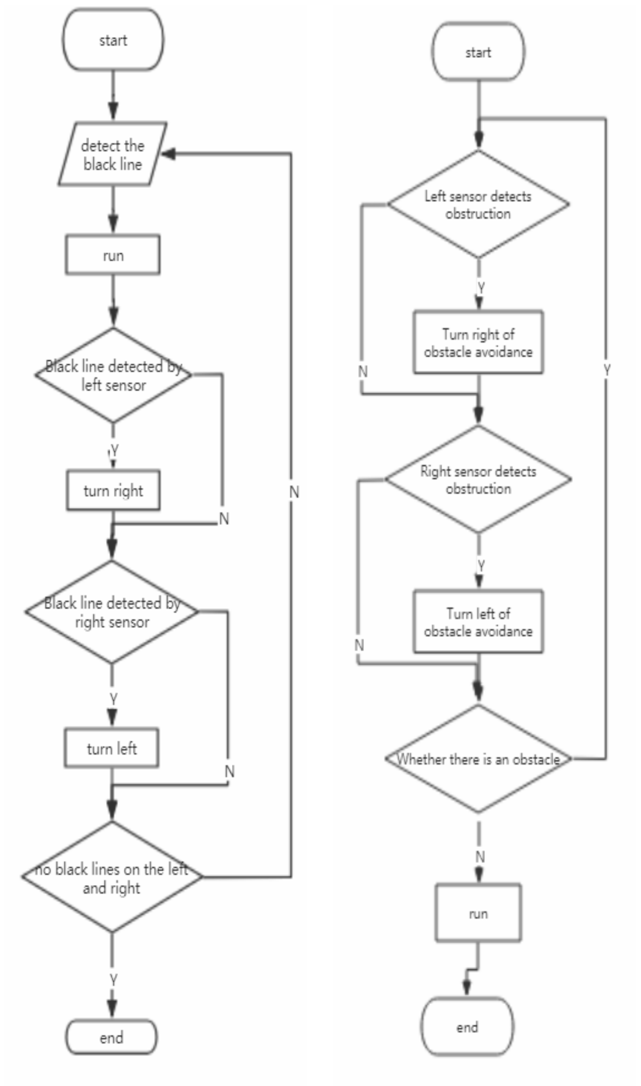
The module created the first create a streaming socket, through the bind function will be the server's IP address and port to bind to a socket file descriptors, and then use the connect function to connect the server, if the server has been open to the specified port, the client can connect the server, then through the recv and send function to receive or send instruction, and then to a server-side processing.

4.3 Track Module Software Design

The intelligent car starts to run from the starting point, and the three tracking sensors at the bottom of the car detect the predetermined track. When the black line is collected by the sensor, the data will be quickly transmitted to the development board for judgment, and the motion state of the car will be adjusted appropriately. Car in the absence of black line will be free to move, undisturbed. The tracking module is developed using Python language, and its software design flow chart is shown in Figure 4(a).

4.4 Obstacle Avoidance Module Software Design

After the smart car starts, a pair of obstacle avoidance sensors on the car will start working. When the sensor detects an obstacle to the car's movement, the data will be sent to the development board for judgment. After the data is processed, the direction of the car's movement will be adjusted. The car will move freely in the absence of obstacles and will not be disturbed. The obstacle avoidance module is developed in Python language, and its software design flow chart is shown in Figure 4(b).



(a) Tracing module software design flow chart

(b) Obstacle avoidance module software design flow chart

Figure 4.

4.5 Global Path Planning Process

Intelligent car global path planning is the top priority in this paper. After starting the function of global path planning, the system needs to store the matrix of global path information for the car according to the path set by the car in advance, and then dynamically input the starting point and end point of the car through the terminal, and use the pre-designed Dijkstra algorithm to calculate the optimal path that the car needs to pass. The trolley traverses each node in the optimal path. For the starting node, the car will use the current node and the next node to arrive to calculate the rotation Angle and rotation direction of the car; For other nodes, the car will use the current node, the previous node and the next node to calculate the rotation Angle and rotation direction of the car. In the program, the car will continuously make cyclic judgment on the optimal path node, and the car will not stop moving until the end of the cyclic traversal of the optimal path node. See Figure 5.

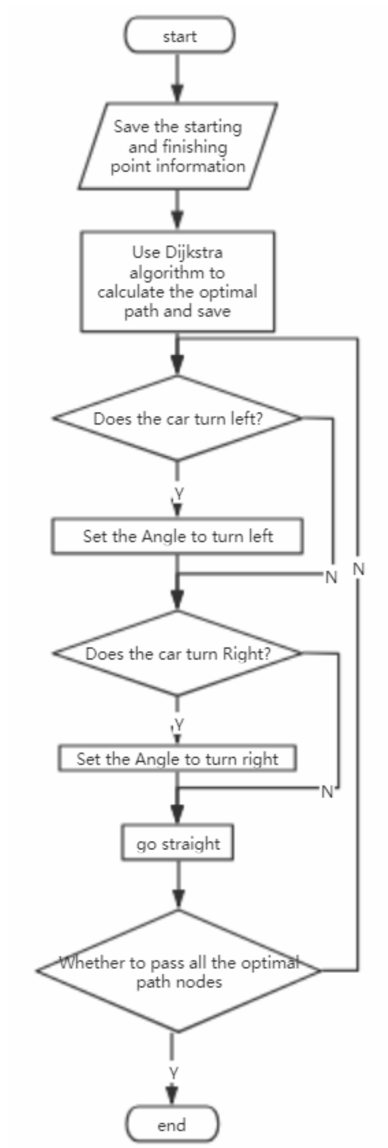


Figure 5. Path planning process design flow chart

5 Experimental Result

The software interface of controlling the Raspberry

Pi car with the mobile terminal software is shown in Figure 6.

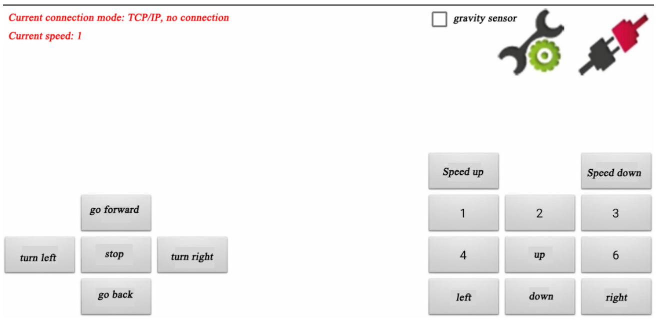


Figure 6. Mobile terminal control interface diagram

The mobile terminal first connects to the WIFI hotspot of the Raspberry Pi development board. After opening the software, set the WIFI icon (as shown in Figure 7), set the control address TCP/IP and control port of the server, and click Save to control the basic movement of the smart car.

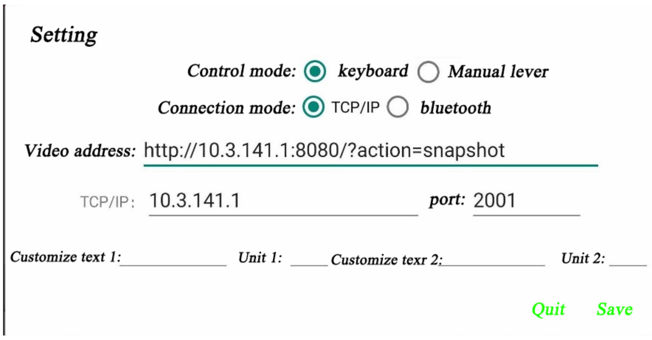


Figure 7. Setting up TCP/IP and ports

As shown in Figure 8, the connection is successful, and the current connection mode in the interface will display "TCP/IP", Connected. After clicking gravity sensing, gravity sensing can also be used to control the movement of the smart car.

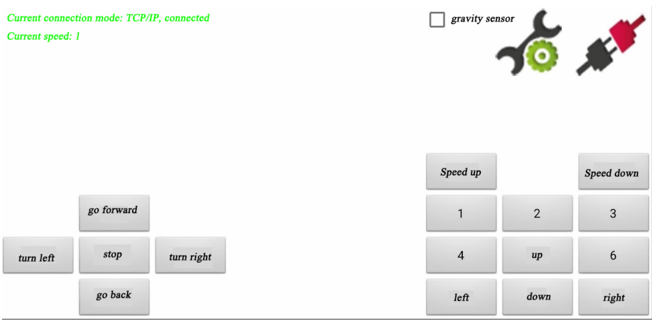


Figure 8. Successfully connected Raspberry Pi

5.1 The Car Tracking

The intelligent car tracking module is to control the car movement by detecting the black line through three tracking sensors of the car.

The result is shown in Figure 9.

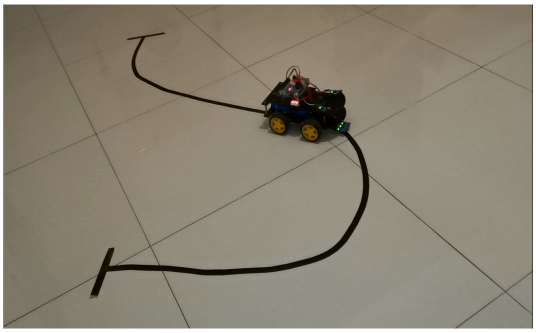


Figure 9. Vehicle tracking path

Through the experimental test, the intelligent car can drive normally through the detection of the black line and completely pass the whole black line. When the car encounters the T-intersection, the three sensors detect the black line and stop moving. The experimental test results reach the design goal. There are also some errors in the experiment: if the car is given a higher speed, it may be because the sensor just detected the black line, but the car rushed out of the black line because of the inertial velocity, so that the car can not move according to the predetermined route; In addition, if the tracking sensor is not sensitive, it will also lead to the car can not move according to the predetermined route.

5.2 Vehicle Obstacle Avoidance

The obstacle avoidance module of intelligent car detects obstacles through two obstacle avoidance sensors of the car so as to control the car's movement. The effect is shown in Figure 8. After the experiment test, the intelligent car can avoid all obstacles to move smoothly. When a sensor on the left detects an obstacle, the car rotates to the right; When a sensor on the right detects an obstacle, the car rotates to the left; When obstacles are detected on both sides, the car will retreat one after another, and then rotate to the left until all obstacles are successfully avoided. The experimental test results meet the design requirements and have reached the expected design objectives. This module design also has some errors: for example, when the car encounters the corner, the car may not be able to avoid the corner due to the limited detection range of the infrared sensor and the car body is too long.

The result is shown in Figure 10.

Figure 10. Obstacle avoidance of trolley

5.3 Autonomous Path Planning

The autonomous path planning of the intelligent car utilizes Dijkstra algorithm to optimize the path of the car. After calculating the optimal path that the car needs to run, it saves it and moves according to the optimal path. Here we take node A as the starting point and node D as the end point. By comparing the results calculated by the algorithm program with the results calculated by hand, the results are completely consistent.

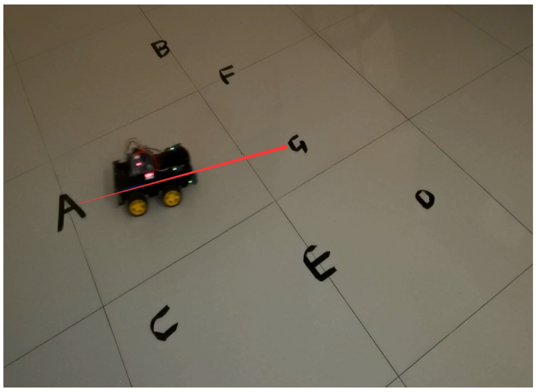
The final path optimization result from node A to node D is:

$$[<0,0,0,0,>,<11.0,-4.0>,$$
$$<7.0,-9.0>,<12.0,-9.0>]$$

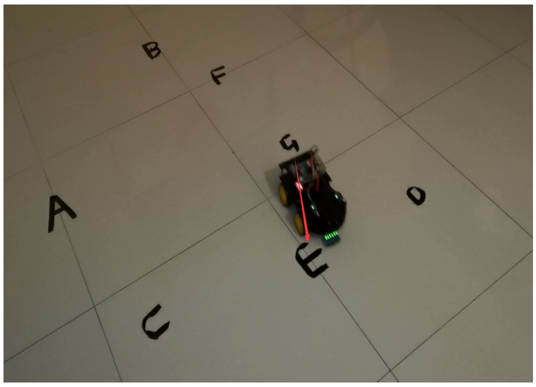
Its time complexity is O(n2), where n represents the number of nodes in the graph. The physical test of the path optimization of the smart car is as follows Figure 11.

After testing, the car can pass the optimal path from point A to point D correctly according to the design goal. The system test results meet the design requirements and achieve the expected goal.

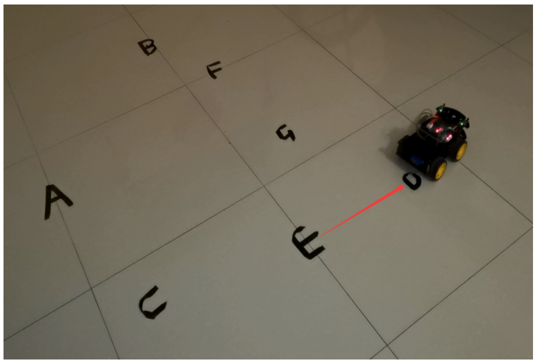
Similarly, some problems have also been found in the process of system implementation and testing: The smoothness of the ground may affect the running position of the car, and relative deviation may occur.



(a) Starting point of path optimization



(b) Optimizing path



(c) Path optimization end point

Figure 11.

6 Conclusion

Through the independent design and test of each module of the intelligent car, the ideal results have been achieved. This paper mainly introduces each module and its working principle, and at the same time expounds the algorithm of autonomous path planning and its algorithm idea. At the same time, according to the requirements of the functional modules to be realized in the intelligent car system, the flow chart of each module is designed respectively, and the corresponding software is developed according to the flow chart. And the realization and test of each module function of the car, and its error is analyzed Through the research on the global path planning of intelligent car, it can make great contributions to the application scenarios of fire truck path planning, remote control system and so on.

Acknowledgments

This job is supported by Natural Science Foundation of Shaanxi Province of China (2021JM-344) and the Key Research and Development Program of Shaanxi Province (No. 2018ZDXM-GY-036) and Shaanxi Key Laboratory of Intelligent Processing for Big Energy Data (No. IPBED7).

References

[1] R. Ma, Z. Guan, Summarization for Present Situation and Future Development of Path Planning Technology, *Modern Machinery*, No. 3, pp. 22-24+27, June, 2008.

[2] G. Zhang, X. Hu, J. Chai, L. Zhao, T. Yu, Summary of Path Planning Algorithm and Its Application, *Modern Machinery*, No. 5, pp. 85-90, October, 2011.

[3] Z. Xu, Current Status and Development of Path Planning Technology for Mobile Robots, *Technology Innovation and Application*, No. 3, pp. 43, March, 2016.

[4] M. Xing, Research on Cleaning *Robot System Development and Path Planning*, Beijing Jiaotong University, July, 2007.

[5] S. Cao, *Research on Path Planning Algorithm of Mobile Robot in Dynamic Environment*, Harbin Institute of Technology,

December, 2019. doi: 10.27061/d.cnki.ghgdu. 2019.000580

[6] J. Fan, R. Liu, F. Han, B. Wu, Research on Intelligent Robot Control Based on Raspberry Pi, *Information Technology and Informatization*, No. 4, pp. 142-144, April, 2018.

[7] K. Qin, Classical Shortest Path Algorithm and Its Implementation, *China New Telecommunications*, Vol. 21, No. 19, pp. 137-139, October, 2019.

[8] Q. Meng, D. Zhang, Research on Optimization and Application of Dijkstra Shortest Path Algorithm, *E-Business Journal*, No. 12, pp. 60-61, December, 2014.

[9] B. Zhang, H. Zhou, S. Liu, Research on improved Dijkstra algorithm in public transportation, *Internet of Things Technologies*, Vol. 8, No. 11, pp. 45-48, November, 2018.

[10] Y. Zhang, Optimization of Dijkstra Algorithm, *Journal of Nanchang Institute of Technology*, Vol. 25, No. 3, pp. 30-33, August, 2006.

[11] G. Rossum, Python Programming Language, *Proceedings of the USENIX Annual Technical Conference*, Santa Clara, CA, USA, 2007.

[12] X.-Y. Liu, Y.-L. Chen, Application of Dijkstra Algorithm in Logistics Distribution Lines, *Third International Symposium on Computer Science and Computational Technology (ISCSCT'10)*, Jiaozuo, China, 2010, pp. 48-50.

[13] D. Zhao, H. Yu, X. Fang, L. Tian, P. Han, A Path Planning Method Based on Multi- Objective Cauchy Mutation Cat Swarm Optimization Algorithm for Navigation System of Intelligent Patrol Car, *IEEE Access*, Vol. 8, pp. 151788-151803, August, 2020.

[14] V. Lefkopoulos, M. Menner, A. Domahidi, M. Zeilinger, Interaction-Aware Motion Prediction for Autonomous Driving: A Multiple Model Kalman Filtering Scheme, *IEEE Robotics and Automation Letters*, Vol. 6, No. 1, pp. 80-87, January, 2021.

[15] W. Wei, B. Zhou, D. Polap, M. Wozniak, A regional adaptive variational PDE model for computed tomography image reconstruction, *Pattern Recognition*, Vol. 92, pp. 64-81, August, 2019.

[16] W. Wei, X. Xia, M. Wozniak, X. Fan, R. Damasevicius, Y. Li, Multi-sink distributed power control algorithm for Cyber-physical-systems in coal mine tunnels, *Computer Networks*, Vol. 161, pp. 210-219, October, 2019.

[17] W. Wei, H. Song, W. Li, P. Shen, A. Vasilakos, Gradient-driven parking navigation using a continuous information potential field based on wireless sensor network, *Information Sciences*, Vol. 408, pp. 100-114, October, 2017.

[18] W. Wei, Q. Xu, L. Wang, X. H. Hei, P. Shen, W. Shi, L. Shan, GI/Geom/1 queue based on communication model for mesh networks, *International Journal of Communication Systems*, Vol. 27, No. 11, pp. 3013-3029, November, 2014.

[19] Q. Ke, J. Zhang, W. Wei, D. Polap, M. Wozniak, L. Kosmider, R. Damasevicius, A neuro-heuristic approach for recognition of lung diseases from X-ray images, *Expert Systems with Applications*, Vol. 126, pp. 218-232, July, 2019.

[20] Q. Ke, J. Zhang, W. Wei, R. Damasevicius, M. Wozniak, Adaptive Independent Subspace Analysis of Brain Magnetic Resonance Imaging Data, *IEEE Access*, Vol. 7, pp. 12252-12261, January, 2019.

[21] Q. Ke, J. Zhang, H. Song, Y. Wan, Big data analytics enabled by feature extraction based on partial independence, *Neurocomputing*, Vol. 288, pp. 3-10, May, 2018.

[22] Q. Ke, J. Zhang, M. Wozniak, W. Wei, The Phase and Shift-Invariant Feature by Adaptive Independent Subspace Analysis for Cortical Complex Cells, *Information Technology And Control*, Vol. 48, No. 1, pp. 58-70, March, 2019.

[23] W. Wei, X. Fan, H. Song, X. Fan, J. Yang, Imperfect information dynamic stackelberg game based resource allocation using hidden Markov for cloud computing, *IEEE Transactions on Services Computing*, Vol. 11, No. 1, pp. 78-89, January-February, 2018.

[24] W. Wei, J. Su, H. Song, H. Wang, X. Fan, Cdma-based anti-collision algorithm for epc global c1 gen2 systems, *Telecommunication Systems*, Vol. 67, No. 1, pp. 63-71, January, 2018.

[25] W. Wei, Z. Sun, H. Song, H. Wang, X. Fan, X. Chen, Energy Balance-Based Steerable Arguments Coverage Method in WSNs, *IEEE Access*, Vol. 6, pp. 33766-33773, 2018.

[26] W. Wei, H. Song, H. Wang, X. Fan, Research and Simulation of Queue Management Algorithms in Ad Hoc Networks under DDoS Attack, *IEEE Access*, Vol. 5, pp. 27810-27817, March, 2017.

[27] W. Wei, Y. Qiang, J. Zhang, A Bijection between Lattice-Valued Filters and Lattice-Valued Congruences in Residuated Lattices, *Mathematical Problems in Engineering*, Vol. 2013, Article No. 908623, July, 2013.

[28] W. Wei, X. L. Yang, B. Zhou, J. Feng, P. Y. Shen, Combined energy minimization for image reconstruction from few views, *Mathematical Problems in Engineering*, Vol. 2012, Article No. 154630, October, 2012.

[29] W. Wei, X. L. Yang, P. Y. Shen, B. Zhou, Holes detection in anisotropic sensornets: Topological methods, *International Journal of Distributed Sensor Networks*, Vol. 8, No. 10, Article No. 135054, October, 2012.

[30] W. Wei, H. M. Srivastava, Y. Zhang, L. Wang, P. Shen, J. Zhang, A local fractional integral inequality on fractal space analogous to Anderson's inequality, *Abstract and Applied Analysis*, Vol. 2014, Article No. 797561, June, 2014.

[31] W. Wei, X. Fan, M. Wozniak, H. Song, W. Li, Y. Li, P. Shen, $H\infty$ Control of Network Control System for Singular Plant, *Information Technology And Control*, Vol. 47, No. 1, pp. 140-150, February, 2018.

[32] W. Wei, X. Fan, H. Song, H. Wang, Video tamper detection based on multi-scale mutual information, *Multimedia Tools & Applications*, Vol. 78, No. 19, pp. 27109-27126, October, 2019.

[33] W. Wei, M. Wozniak, R. Damaevicius, X. Fan, Y. Li, Algorithm Research of Known-plaintext Attack on Double Random Phase Mask Based on WSNs, *Journal of Internet Technology*, Vol. 20, No. 1 , pp. 39-48, January, 2019.

[34] W. Wei, Q. Yong, Information potential fields navigation in wireless Ad-Hoc sensor networks, *Sensors*, Vol. 11, No. 5, pp. 4794-4807, May, 2011.

[35] W. Wei, S. Liu, W. Li, D. Du, Fractal Intelligent Privacy

Protection in Online Social Network Using Attribute-Based Encryption Schemes, *IEEE Transactions on Computational Social Systems*, Vol. 5, No. 3 pp. 736-747, September, 2018.

[36] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, M. Wozniak, Accurate and fast URL phishing detector: A convolutional neural network approach, *Computer Networks*, Vol. 178, Article No. 107275, September, 2020.

[37] W. Wei, E. S. L. Ho, K. D. McCay, R. Damasevicius, R. Maskeliunas, A. Esposito, Assessing facial symmetry and attractiveness using augmented reality, *Pattern Analysis and Applications*, pp. 1-17, March, 2021.

Biographies

Wei Wei is an associate professor of School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China. He is a senior member of IEEE, CCF. He received his Ph.D. and M.S. degrees from Xian Jiaotong University in 2011 and 2005, respectively.

Fan Gao received her B.Sc. from Xi'an Shiyou University, China, in 2020. He is currently a postgraduate student at Xi'an University of Technology, in China and will receive his M.S. in 2023. His research interests are image processing, recognition and parsing.

Rafał Scherer received his MSc degree in computer science from the Czestochowa University of Technology, Poland, in 1997 and his PhD in 2002 from the same university. Currently, he is an associate professor at Czestochowa University of Technology. His present research interests include machine learning and neural networks for image processing, computer system security, prediction and classification.

Robertas Damasevicius graduated at the Faculty of Informatics, Kaunas University of Technology (KTU) in Kaunas, Lithuania in 1999, where he received a B.Sc. degree in Informatics. He finished his M.Sc. studies in 2001 (cum laude), and he defended his Ph.D. thesis at the same University in 2005. Currently, he is a Professor at Software Engineering Department, KTU and lectures robot programming and software maintenance courses.

Dawid Polap is with Institute of Mathematics of the Silesian University of Technology in Poland, where he received his diplomas in computer science and applied mathematics (with honors), in 2016 and 2017 receptively. In his career, he was awarded by the Ministry of Science and Higher Education with a "Diamond Grant" for the most talented students.