Hindawi
Mobile Information Systems
Volume 2022, Article ID 3068614, 10 pages
https://doi.org/10.1155/2022/3068614

*Research Article*

# Obstacle Avoidance Path Planning Algorithm of an Embedded Robot Based on Machine Vision

**Yiyang Shou**

*Robotic Engineering, SHIEN-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 511442, China*

Correspondence should be addressed to Yiyang Shou; wishouyiyang@mail.scut.edu.cn

Received 18 June 2022; Revised 18 July 2022; Accepted 22 July 2022; Published 17 August 2022

Academic Editor: Praveen Kumar Reddy Maddikunta

Due to the lack of accurate modeling information in environment modeling, the traditional path planning algorithm for robot obstacle avoidance is of low accuracy. Therefore, this paper designs an obstacle avoidance path planning algorithm for embedded robot based on machine vision. First, the method of target edge detection is optimized in this paper. The edge detection results are obtained by color space transformation, and the complete target is obtained by edge fusion combined with surrounding pixel attributes. Then, the distance of the obstacle is measured by binocular depth ranging, and the longitudinal positioning of the robot is obtained, and the position of the obstacle is further obtained. Finally, a fuzzy control method for obstacle avoidance path planning is designed to obtain a complete planning scheme. The performance test results of the obstacle avoidance path algorithm show that the obstacle avoidance path planning scheme obtained by the algorithm designed in this paper has better performance in different obstacle avoidance test environments and can successfully avoid obstacles when the robot runs at high speed.

## 1. Introduction

Mobile robots are widely used at present. They can replace manpower for production, detection, farming, and other work so as to save manpower cost. Generally, the movement of robot is mainly through independent identification of working environment, making decisions, and completing work. With the maturity of robot technology, the working space of robot is expanding, which can replace human beings to detect in dangerous environment [1]. To put an embedded robot in a workplace, it first needs to recognize the nearby state of the sports place and have the ability to reach the destination. When encountering obstacles in the site, the safety path can be updated in time [2, 3]. At present, the embedded robot generally uses infrared or ultrasonic for the perception of the surrounding environment. However, such a perception method is greatly affected by the environment. It is easy to misjudge the location of obstacles in path planning, resulting in poor performance of path planning scheme. Machine vision technology is becoming more and more mature. Due to its large amount of information and rich content, it is widely used in different fields, such as defect detection, unmanned technology, and many other fields [4, 5]. In recent years, with the development of deep learning and its visual technology, it has expanded its scope of application. For the obstacle avoidance path planning of embedded robot in practical application, the traditional path planning algorithm lacks complete site modeling information, so the obtained path planning scheme is lack of performance [6]. Therefore, this paper combines machine vision technology with the obstacle avoidance path planning algorithm in order to improve the performance and accuracy of robot obstacle avoidance scheme.

## 2. Obstacle Avoidance Path Planning Algorithm of the Embedded Robot

*2.1. Optimize Target Edge Detection.* In the process of obstacle avoidance route planning, analyzing the movement environment is the basis of obstacle avoidance route planning. In a moving environment, edge detection of target obstacles can lay a foundation for spatial segmentation and obstacle extraction. General object edge detection is to construct a difference operator to detect the edge of gray image. However, the images currently used are generally color images. Therefore, in the original detection operator, we also need to carry out gray conversion. Gray conversion will cause the loss of image information and the waste of color details [7, 8]. Therefore, the edge detection operator is optimized in this paper. For color images, we have to intercept and enlarge a small part of the local image, as shown in Figure 1:

Figure 1: Schematic diagram of edge detection.

In the figure above, a small part of the edge of an object is intercepted and enlarged. It can be seen that the intercepted part is composed of different pixel color bands. Transform the color space of the color image, convert the original RGB color into HSI color, and cooperate with the gray color space for edge detection [9, 10]. The component of the small image in HSI color space can be obtained. According to the obtained component information, edge detection is carried out in $H$, $S$, and $I$ spaces, respectively. The edge detection result corresponding to $H$ space is recorded as $E_H$, the edge detection result corresponding to $S$ space is recorded as $E_S$, and the edge detection result corresponding to $I$ space is recorded as $E_I$. Under the above edge detection results, carry out edge fusion, and the formula is as follows:

$$E_T = (E_H \bigcup E_S) | (E_H \bigcup E_I) | (E_S \bigcup E_I). \quad (1)$$

In the above formula, the obtained $E_T$ is a matrix form, which is recorded as

$$E_T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2)$$

In the matrix of the above formula, 0 represents nonedge pixels and 1 represents edge pixels. Taking the above matrix as an example, the edges are not connected, and the pixels of the edge nodes consist of multiple regions. The pixels that make up nonedge nodes share a region, and there is no real separation between different obstacle targets. Therefore, after expanding the image, we can see that there are many breakpoints in the edge detection results. In this way, the connection of edge lines can be considered [11]. In the original algorithm, the edge fracture will affect the calculation of the surrounding environment and ultimately affect the separation of targets. Therefore, in the target edge monitoring of this paper, we should first detect an isolated edge point. Taking this point as the center and combining with the surrounding pixels, we can form a whole. At this time, we can detect the overall edge. Such a target edge detection method can avoid the isolated detection of pixels and can make a macro judgment from the whole image after combining the surrounding pixel attributes. After the above improvement of edge pixel recognition, the edge breakpoints can be repaired so as to obtain better edge detection effect.

*2.2. Obstacle Distance Measurement.* For the robot, the premise of path planning is its longitudinal positioning. Therefore, it is necessary to rely on the robot's binocular vision to measure depth. Within the action range of the robot, the robot obtains relevant obstacle distance information from the machine vision shooting area and determines the ordinate of the robot [12] based on this information. In the process of depth estimation, the depth information is mainly obtained by measuring the parallax of obstacles on different imaging planes. If the same object is in different imaging planes at the same time, there will be some parallax. Parallel binocular distance measurement is generally used to measure depth because the measurement value obtained by using this method is more accurate. The principle diagram of parallel binocular distance measurement is shown in Figure 2.

In the above figure, $C_L$ and $C_R$ are the optical centers of two cameras in binocular vision, respectively, and $B$ represents the distance between the two optical centers. The optical center distance is determined through calibration calculation. $O_L$ and $O_R$ represent the center point of the imaging surface, and $f$ is the focal length of the camera [13]. In the measurement and imaging process of binocular vision, we need to get a point $P$ in the normal physical space, and the default is that the point is the idealized coordinate of the obstacle. The position of the obstacle in the aerial coordinate system is expressed as $(x_s, y_s, z_s)$, and it is assumed that the position coordinates on the respective imaging planes of binocular vision are expressed as $P_L(x_L, y_L)$ and $P_R(x_R, y_R)$, respectively. In the actual operation, in order to simplify the operation steps, assuming that the spatial coordinate system is the same as the imaging plane coordinate system, the following can be obtained by using the triangle similarity principle:

$$\begin{cases} \dfrac{z_s - f}{z_s} = \dfrac{a}{a + x_R} \\ \dfrac{z_s - f}{z_s} = \dfrac{B - x_L + x_R + a}{B + x_R + a} \end{cases} \longrightarrow z_s = \frac{fB}{x_L - x_R}. \quad (3)$$

In the above formula, the obtained $x_L - x_R$ is the parallax of binocular vision in the imaging process. During the calibration of obstacles, the measurement of spatial depth is completed through the values of focal length and baseline obtained by calibration [3, 14]. Further processing can obtain the specific location information of the obstacle as

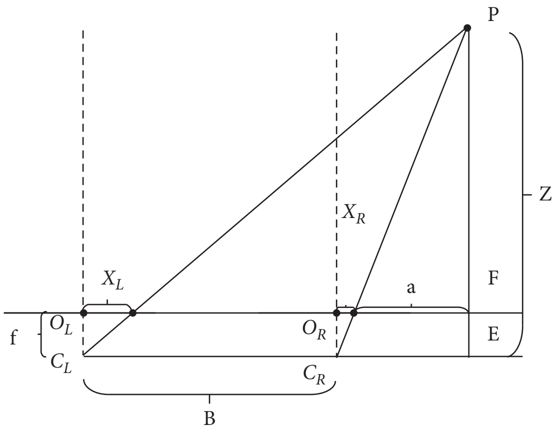

FIGURE 2: Schematic diagram of binocular depth ranging.

$$\begin{cases} x_s = \dfrac{Bx_L}{x_L - x_R}, \\ \\ y_s = \dfrac{By_L}{y_L - y_R}. \end{cases} \tag{4}$$

Under the calculation of the above formula, since the main imaging points of the two cameras are different, the measurement results need to be further revised. The revised equation is

$$\begin{cases} x_s = \dfrac{B(x_L - u_{0L})}{(u_{0L} - u_{0R}) - (x_L - x_R)}, \\ \\ y_s = \dfrac{B(y_L - u_{0L})}{(u_{0L} - u_{0R}) - (y_L - y_R)}, \\ \\ z_s = \dfrac{Bf}{(u_{0L} - u_{0R}) - (z_L - z_R)}. \end{cases} \tag{5}$$

In the above formula, $u_{0L}$ and $u_{0R}$ are the revision coefficients in the two imaging planes, respectively. Based on the above principle of binocular depth distance calculation, it is integrated into the process of obstacle depth estimation.

*2.3. Design Obstacle Avoidance Path Planning Control Method.* For the obstacle avoidance path planning of embedded robot, the core of the algorithm is the design of control algorithm. The designed obstacle avoidance path planning scheme needs to meet the high real-time performance of the embedded robot and also get the path planning scheme quickly and accurately. In the process of designing obstacle avoidance planning, this paper uses the method of fuzzy control to realize efficient data processing. A fuzzy controller is embedded in the path planning algorithm. According to the characteristics of the embedded robot in the walking process, the corresponding fuzzy controller is designed, and its structure is shown in Figure 3.

As can be seen from the above figure, the input variables in the fuzzy controller can be subdivided into two. One is the angle deviation $e$, and the other is the change rate EC of the angle deviation, and the output of the controller is the steering angle $U$ of the robot. After completing the design of the structure of the fuzzy controller, it is also necessary to divide the controller into fuzzy parts. Fuzzification mainly describes the control rules in the process of robot travel through more states. The more the states accumulate, the more flexible and accurate the process of selecting rules. First, the input variable $E$ of the fuzzy controller is fuzzy divided. According to the actual parameters of the robot during driving, the generally selected angle deviation range is set at $-20^\circ \sim 20^\circ$. Taking the motion direction of the robot as the center line, the left side of the center line is negative and the right side is positive. Converting the angle deviation range into quantitative discrete universe, that is, dividing the continuous angle deviation range into $2n$ segments, and the fuzziness coefficient at this time can be expressed as

$$q = \frac{2n}{b - a}. \tag{6}$$

In the above formula, $a$ is the lower limit of the angle deviation range and $b$ is the upper limit of the angle deviation range. The fuzzification coefficient can also be used as a quantization factor here. The fuzzy set corresponding to the angular deviation of state variables in the process of robot walking is $\{\mathrm{NB}, \mathrm{NM}, \mathrm{ZO}, \mathrm{PM}, \mathrm{PB}\}$. The corresponding meanings of fuzzy set are negative large, negative middle, zero, positive middle, and positive large, respectively. The shape of membership function is shown in Figure 4:

The existence of membership function is mainly to quantitatively describe the variables in the process of fuzzy control, so the function used in this paper is triangular function. Combined with the image of membership function, the assignment of angle deviation can be expressed as Table 1.

Under the above fuzzy control rules, combined with the target edge detection and binocular depth ranging of machine vision technology, the state detection in the robot action environment can be effectively realized. Building an accurate environment model based on the above information is the key of the robot obstacle avoidance path planning algorithm. After the actual physical space is constructed into an abstract space, the global path planning can be carried out. The overall performance of path planning with global coverage has certain advantages compared with local path. In global planning, various traversal methods of local path planning can be combined to improve the performance of path planning scheme. In the process of path finding, the path planning algorithm designed in this paper will use heuristic distance to calculate and update the cost distance. Under the above analysis, the flow chart of the embedded robot obstacle avoidance path planning algorithm based on machine vision is shown in Figure 5:

In the above process, the planning of obstacle avoidance path of the embedded robot based on machine vision can be effectively completed, and the perfect scheme of obstacle avoidance in the process of robot movement can be obtained.

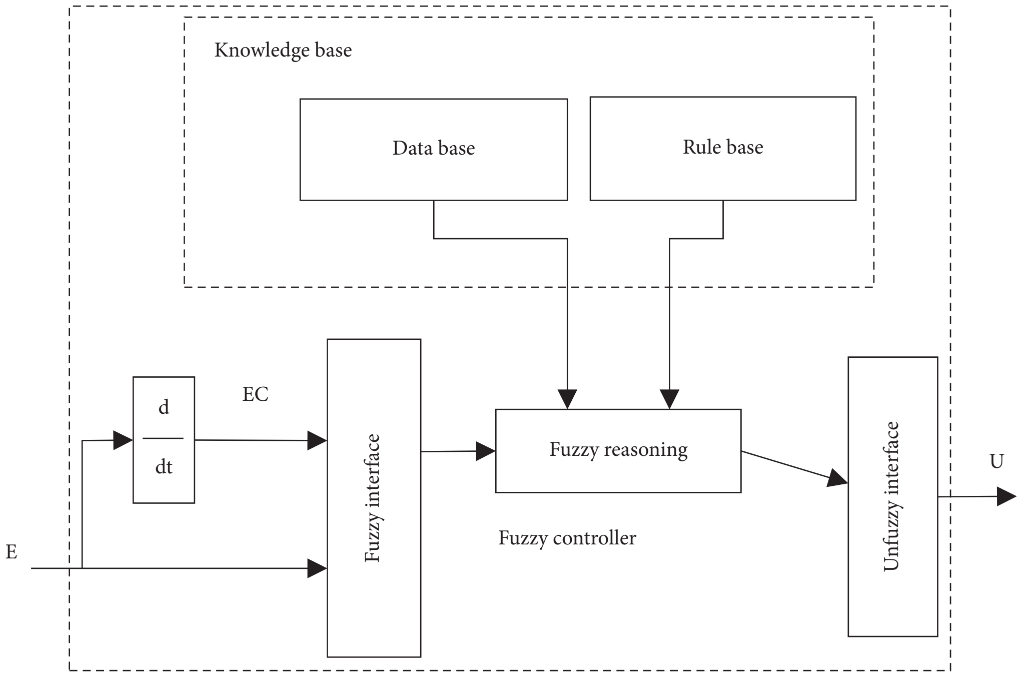

Figure 3: Structure of the algorithm fuzzy controller.

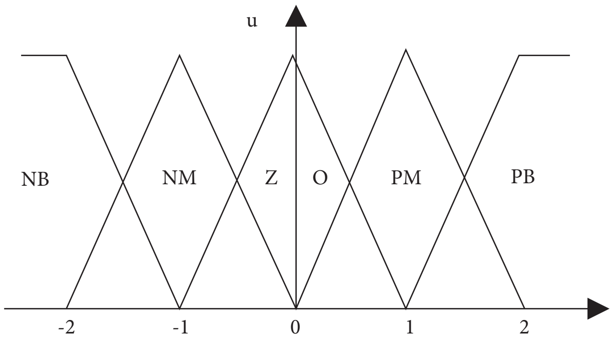

Figure 4: Membership function.

Table 1: Comparison table of angle deviation assignment.

| Quantization threshold | NB | NM | ZO | PM | PB |
|---|---|---|---|---|---|
| −2 | 1 | 0 | 0 | 0 | 0 |
| −1.5 | 0.5 | 0.5 | 0 | 0 | 0 |
| −1 | 0 | 1 | 0 | 0 | 0 |
| −0.5 | 0 | 0.5 | 0.5 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0.5 | 0 | 0 | 0.5 | 0.5 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1.5 | 0 | 0 | 0 | 0.5 | 0.5 |
| 2 | 0 | 0 | 0 | 0 | 1 |

## 3. Algorithm Performance Test and Discussion

According to the above analysis, this paper uses machine vision to optimize the obstacle avoidance path planning algorithm of the embedded robot. In the following experiments, the validity of the proposed algorithm is verified.

In order to verify the effectiveness of the embedded robot obstacle avoidance path planning algorithm based on machine vision in practical application, experiments are needed. The embedded robot is used to carry the path planning algorithm designed in this paper and the traditional path planning algorithm, respectively. It is debugged under the existing laboratory test conditions, and the path planning schemes of the robot obtained by different algorithms are analyzed.

*3.1. Performance Test Environment.* The main running environment of obstacle avoidance path planning test of the embedded robot is the laboratory paved with white tiles. In such a laboratory environment, there is an obstacle environment in the area where the robot runs. The test platform of the embedded robot used is shown in Figure 6.

In the test platform above, set the relevant parameters and required equipment during the experiment, as shown in Table 2.

Under the experimental equipment and relevant parameters in the above table, in order to reduce the obstacle avoidance delay, use the multimeter, oscilloscope, and other experimental instruments to debug the module of the robot, and then complete the power on of the embedded robot, and debug to ensure that each module of the robot can be used normally in the test process. In the test process, the state changes of the robot in the test movement, such as driving speed, turning angle, and other parameters, will be obtained through the signal coding feedback device installed on the embedded robot. In the experiment, different robot velocity gradients are set up, obstacles are set up and tested in the laboratory site, and the experimental results are observed and analyzed.

*3.2. Robot Testing in Different Obstacle Testing Environments.* In the test of obstacle avoidance path planning algorithm in this paper, the test sites under single obstacle, multidirectional

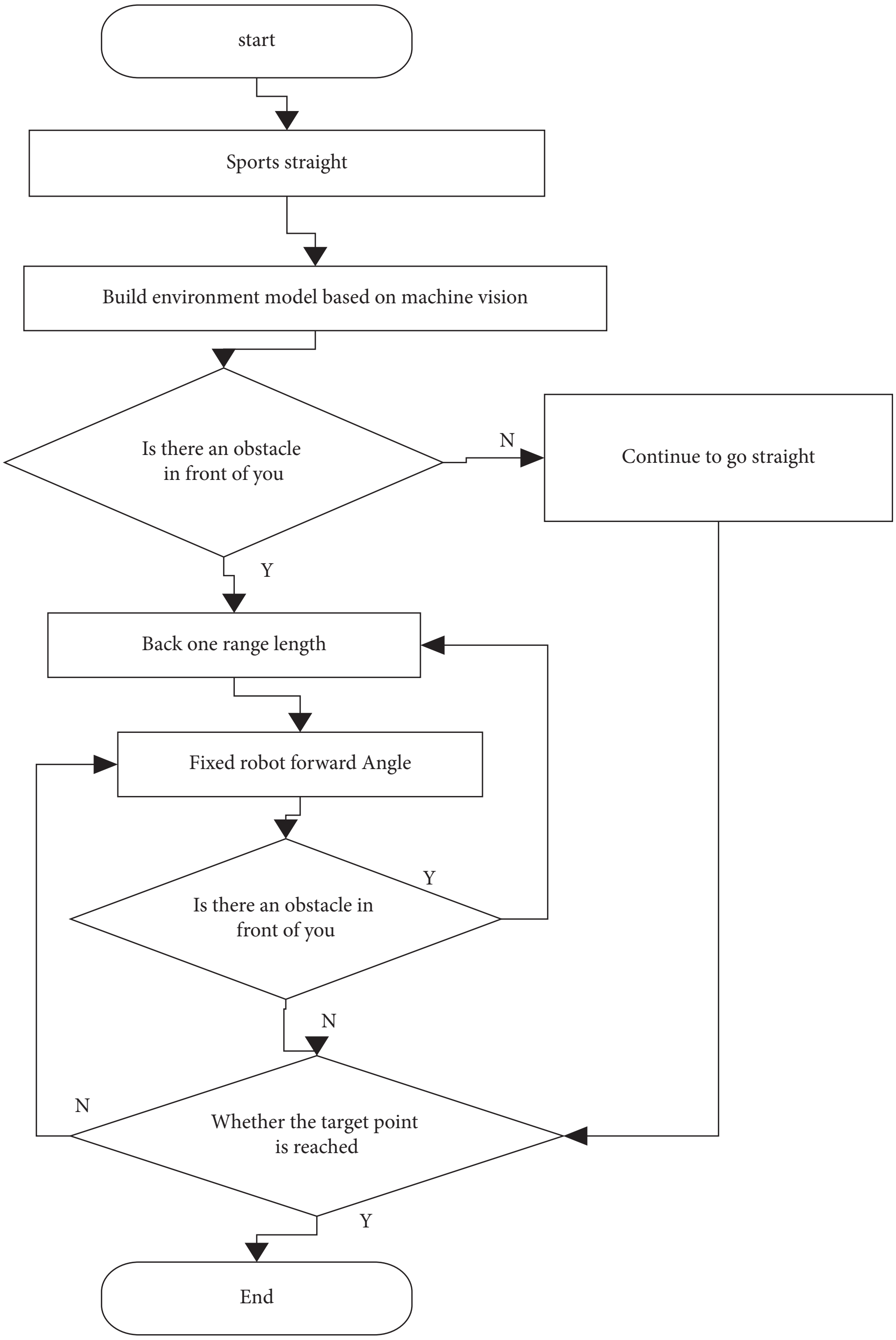
start
Sports straight
Build environment model based on machine vision
Is there an obstacle in front of you
N
Continue to go straight
Y
Back one range length
Fixed robot forward Angle
Y
Is there an obstacle in front of you
N
N
Whether the target point is reached
Y
End

Figure 5: Algorithm flow chart.

obstacle, and complex obstacle environment are set up, respectively. After many experiments in a single obstacle test environment, the obstacle avoidance planning path of the robot under the algorithm in this paper is shown in Figure 7:

In the test above, the position of obstacles placed on the laboratory floor changes a lot. The obstacles are around the moving direction of the robot. Under the above test environment, change the starting position of the robot and carry out multiple tests. In order to increase the contrast, the traditional robot obstacle avoidance path planning algorithm is used for many tests in the same experimental environment, and the experimental results are recorded.

In the multidirectional obstacle testing environment, obstacles are set in different directions on the experimental site, and different path planning algorithms are used for testing. The expected route and obstacle position of the embedded robot under the path planning algorithm in this paper are shown in Figure 8.

In the multidirectional obstacle environment, that is, in the test process of robot path planning algorithm, there are

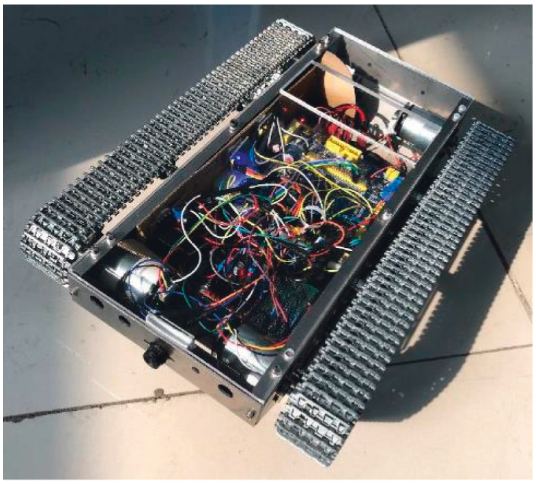

Figure 6: Embedded robot test platform.

Table 2: Experimental parameters and equipment.

| Experimental equipment | Parameter description | Unit |
|---|---|---|
| AC servo motor driver | Input voltage | 58 V |
| | Send PWM signal frequency | 375 KHz |
| | Duty cycle | 50% |
| Robot sensor | Scanning cycle | 4000 ms |
| PC | System | win7 |
| | Memory | 2G |
| Serial port extension cable | 7 | *m* |

more than one obstacle that affects the normal travel of the robot. As shown in the above figure, multiple obstacles are at different positions in the forward direction of the robot. Under the above test environment, change the starting position of the robot and conduct multiple tests. In order to increase the contrast, the traditional robot obstacle avoidance path planning algorithm is used for many tests in the same experimental environment, and the experimental results are recorded.

In order to verify the usability of the planning algorithm, the complex and common obstacle environment in the process of robot travel is simulated in the laboratory environment, and the obstacle avoidance test is carried out in this environment. The path of the robot under the algorithm in this paper is shown in Figure 9.

In the above environment, the robot platform equipped with this model and the traditional model is used for multiple tests, the number and times of robot collision with obstacles are recorded, the average value is obtained, and the experimental results are compared.

In the above experiments, the obstacle avoidance path planning of the robot is carried out for many times, and the test results of the two algorithms are shown in Table 3:

According to the test results, the two obstacle avoidance algorithms show good decision-making performance of obstacle avoidance path planning in a simple obstacle test environment. For a single obstacle, there is an accurate ranging and prediction, and change the original route in time to replan the route. In the case of multiple tests, it shows a good performance. In the multidirectional obstacle environment, the average number of collision obstacles and the average number of collision obstacles in the traditional obstacle avoidance path planning algorithm begin to rise slightly. Under this algorithm, there is little change. In the complex environment with a large number of obstacles, the number of collision obstacles under the traditional algorithm increases, and the effect of modified path planning after collision is poor, resulting in multiple collisions with unified obstacles. Based on the above experimental data, it can be concluded that the performance of the path scheme obtained in the obstacle avoidance path planning algorithm designed in this paper is better, and the average number and average times of collision obstacles are reduced compared with the traditional methods.

*3.3. Analysis of Obstacle Avoidance Results.* Four obstacles are placed in a row in the direction of robot movement, and the obstacle avoidance performance of the robot at different speeds is tested. For the robot, its obstacle avoidance route is shown in Figure 10:

In the test environment shown in the figure above, in order to avoid obstacles in the middle, the robot will generally make the obstacle avoidance path planning, as shown in Figure 10. In the test, this algorithm and the traditional algorithm are used to test many times under different initial running speeds of the robot. The test results are shown in Table 4:

In the obstacle avoidance path planning test in this paper, the same robot was tested repeatedly for 100 times at

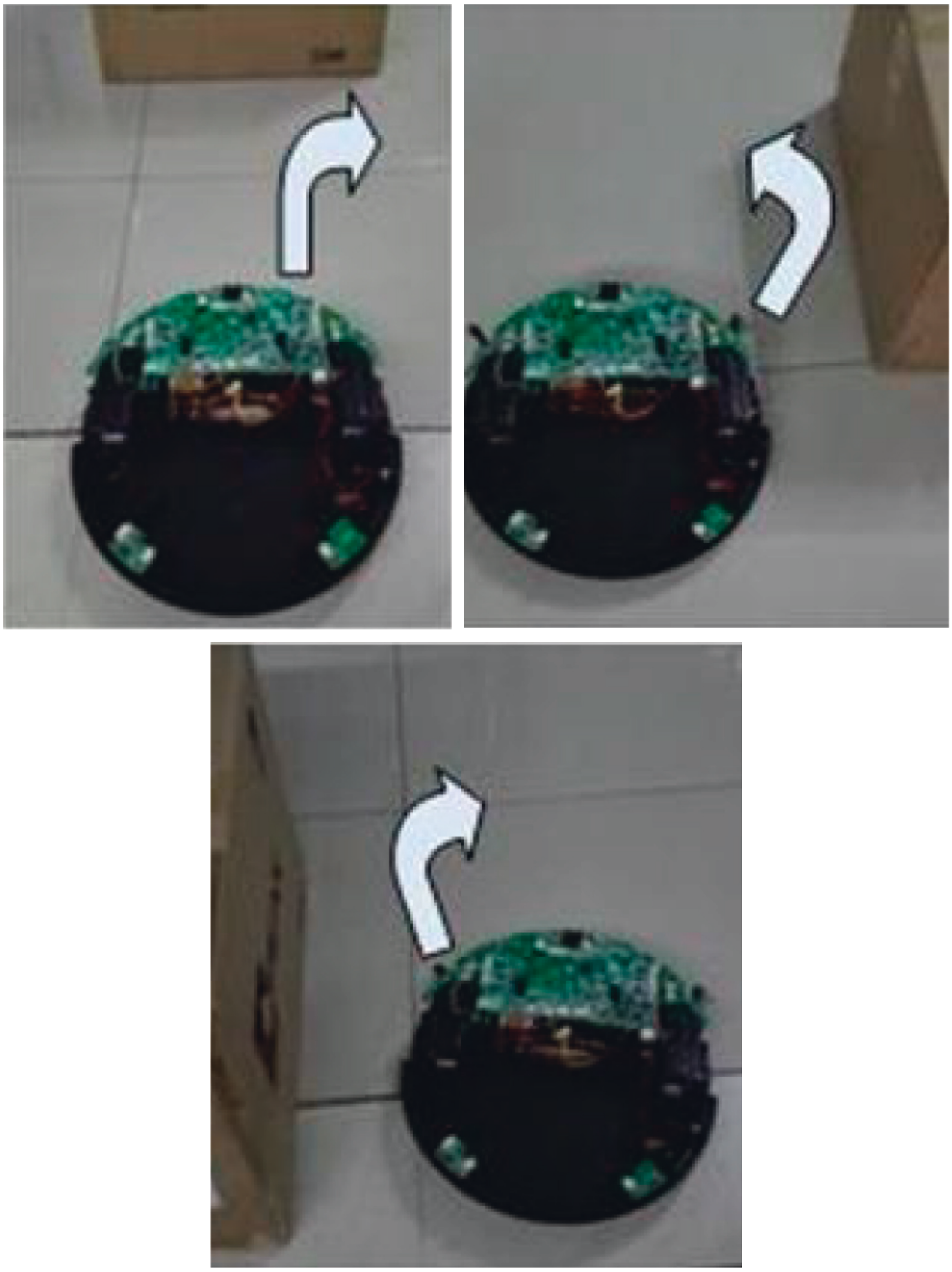

Figure 7: Robot path planning trend under single direction obstacle environment.

the initial running speed. Generally speaking, when the number of successful obstacle avoidance in the test results reaches more than 85 times, the path planning scheme obtained by the path planning algorithm is considered to be effective. As can be seen from the data in the above table, when the running speed of the robot is lower than 55 cm/s, both path planning algorithms can effectively avoid obstacles. When the running speed of the robot is 55–70 cm/s, the successful obstacle avoidance times of the traditional algorithm are reduced to 80 times, and the obstacle avoidance is not effective. However, the obstacle avoidance path scheme based on the algorithm designed in this paper can still achieve effective obstacle avoidance. With the increase of the initial speed of the robot, the number of successful obstacle avoidance of the two algorithms cannot reach the standard of successful obstacle avoidance. To sum up, the embedded robot obstacle avoidance path planning algorithm based on machine vision designed in this paper

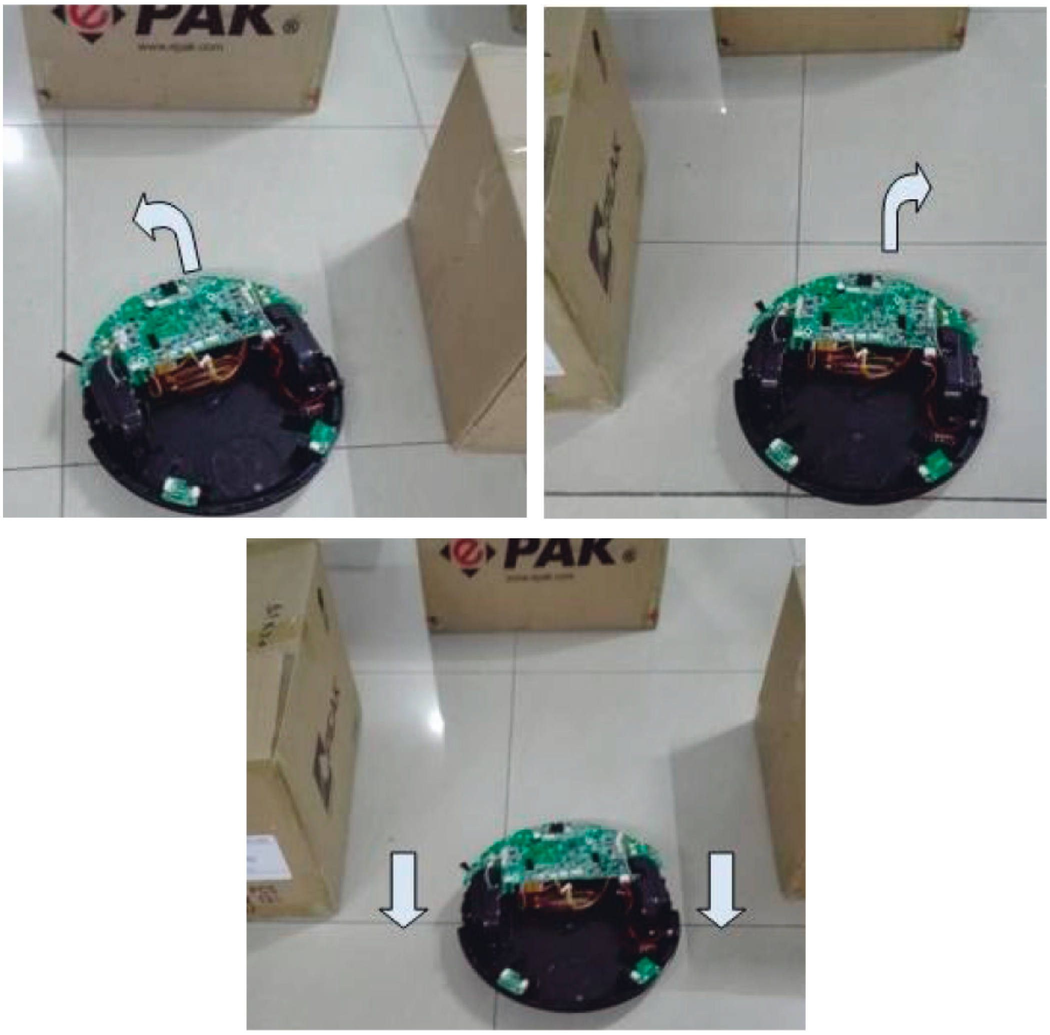

Figure 8: Robot path planning trend under the multidirectional obstacle environment.

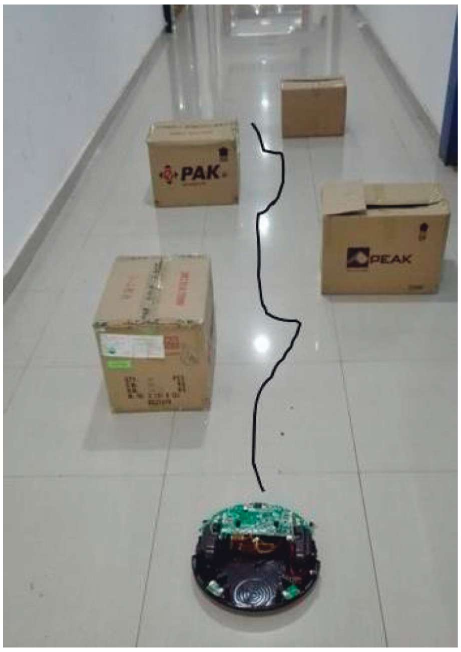

Figure 9: Obstacle environment in the complex environment robot path planning trend under this algorithm.

Table 3: Comparison of experimental results of two embedded robot obstacle avoidance path planning algorithms.

| Testing environment | Traditional obstacle avoidance path planning algorithm | | The obstacle avoidance path planning algorithm in this paper | |
|---|---|---|---|---|
| | Average number of collision obstacles | Average number of collisions with obstacles | Average number of collision obstacles | Average number of collisions with obstacles |
| Single obstacle environment | 0.11 | 0.29 | 0 | 0 |
| Multidirectional obstacle environment | 0.73 | 1.23 | 0.07 | 0.15 |
| Complex obstacle environment | 2.13 | 5.62 | 0.32 | 1.36 |

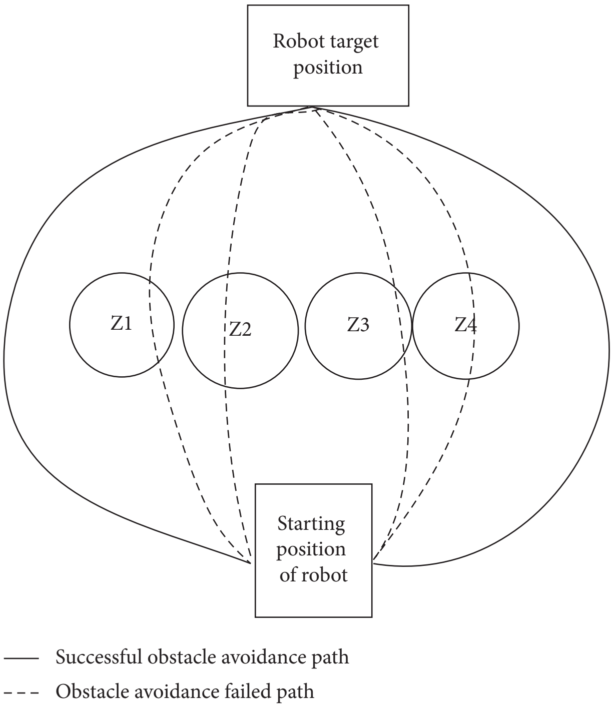
Robot target position
Z1
Z2
Z3
Z4
Starting position of robot
—— Successful obstacle avoidance path
- - - Obstacle avoidance failed path

Figure 10: Different path classification of the obstacle avoidance test.

Table 4: Statistics of the robot obstacle avoidance test results under different initial speeds.

| Initial motion speed of the robot (cm/s) | Total number of tests | The number of successful obstacle avoidance under this algorithm | Number of successful obstacle avoidance under the traditional algorithm |
|---|---|---|---|
| 10 | 100 | 100 | 100 |
| 15 | 100 | 100 | 100 |
| 20 | 100 | 100 | 100 |
| 25 | 100 | 100 | 99 |
| 30 | 100 | 100 | 97 |
| 35 | 100 | 98 | 94 |
| 40 | 100 | 97 | 94 |
| 45 | 100 | 95 | 90 |
| 50 | 100 | 93 | 88 |
| 55 | 100 | 92 | 86 |
| 60 | 100 | 90 | 84 |
| 65 | 100 | 88 | 82 |
| 70 | 100 | 85 | 80 |
| 75 | 100 | 82 | 78 |
| 80 | 100 | 80 | 75 |
| 85 | 100 | 80 | 72 |
| 90 | 100 | 78 | 70 |
| 95 | 100 | 73 | 67 |
| 100 | 100 | 70 | 61 |

has a certain improvement in performance compared with the traditional algorithm; that is, the validity of the algorithm constructed in this paper is verified.

## 4. Conclusion

Aiming at the shortcomings of obstacle avoidance path planning of the embedded robot, this paper designs an obstacle avoidance path planning algorithm of the embedded robot based on machine vision. It mainly uses machine vision technology to perceive and model the surrounding environment more quickly and accurately, designs the obstacle avoidance path planning fuzzy control method in the planning algorithm, and obtains the completed planning process. Experiments show that the path planning algorithm designed in this paper has a certain improvement in performance. Although the research of this paper has achieved some results, there are still many details to be improved in the follow-up study and research. For example, how to use machine vision for rapid modeling to improve the running speed of the robot is the key content to be studied in the future.

## Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Conflicts of Interest

The author(s) declare no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

## References

[1] X. Zhu, Y. Liang, H. Sun, X. Wang, and B. Ren, "Robot obstacle avoidance system using deep reinforcement learning," *Industrial Robot: The International Journal of Robotics Research and Application*, vol. 49, no. 2, pp. 301–310, 2021.

[2] Y. Liu, "Intelligent robot obstacle avoidance perception method using a high-precision laser distance measuring system," *Journal of Nanoelectronics and Optoelectronics*, vol. 16, no. 3, pp. 471–478, 2021.

[3] A. M. Abdulkadium, "A robot obstacle avoidance method based on random forest HTM cortical learning algorithm," *Webology*, vol. 17, no. 2, pp. 788–803, 2020.

[4] T. Xu, H. Zhou, S. Tan, Z. Li, X. Ju, and Y. Peng, "Mechanical arm obstacle avoidance path planning based on improved artificial potential field method," *Industrial Robot: The International Journal of Robotics Research and Application*, vol. 49, no. 2, pp. 271–279, 2021.

[5] N. Magadevi and V. J. S. Kumar, "Energy efficient, obstacle avoidance path planning trajectory for localization in wireless sensor network," *Cluster Computing*, vol. 22, no. S5, pp. 10529–10535, 2019.

[6] M. K. P. So, "Robo-advising risk profiling through content analysis for sustainable development in the Hong Kong financial market," *Sustainability*, vol. 13, no. 3, p. 1306, 2021.

[7] Z. Wang, T. Xin, S. Wang, and H. Zhang, "Depth-embedded instance segmentation network for urban scene parsing," *Journal of Intelligent and Fuzzy Systems*, vol. 42, no. 3, pp. 1269–1279, 2022.

[8] K. Yan, H. Wang, S. Bu, L. Yang, and J. Li, "Scene parsing for very high resolution remote sensing images using on attention-residual block-embedded adversarial networks," *Remote Sensing Letters*, vol. 12, no. 7, pp. 625–635, 2021.

[9] B. J. Erickson, "Basic artificial intelligence techniques: machine learning and deep learning," *Radiologic Clinics of North America*, vol. 59, no. 6, pp. 933–940, 2021.

[10] W. Gao, Q. Tang, B. Ye, Y. Yang, and J. Yao, "An enhanced heuristic ant colony optimization for mobile robot path planning," *Soft Computing*, vol. 24, no. 8, pp. 6139–6150, 2020.

[11] H. Yixiong, L. Zhaoran, M. Junmin, H. Weixuan, L. Liling, and W. Bing, "Collision-avoidance path planning for multi-ship encounters considering ship manoeuvrability and COLREGs," *Transportation Safety and Environment*, vol. 3, no. 2, pp. 103–113, 2021.

[12] W. Gouda and R. J. B. Chikha, "NAO humanoid robot obstacle avoidance using monocular camera," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 1, pp. 274–284, 2020.

[13] W. Zheng, H. B. Wang, Z. M. Zhang, N. Li, and P. H. Yin, "Multi-layer feed-forward neural network deep learning control with hybrid position and virtual-force algorithm for mobile robot obstacle avoidance," *International Journal of Control, Automation and Systems*, vol. 17, no. 4, pp. 1007–1018, 2019.

[14] M. Duguleana and G. Mogan, "Neural networks based reinforcement learning for mobile robots obstacle avoidance," *Expert Systems with Applications*, vol. 62, pp. 104–115, 2016.