Autonomous Obstacle Avoidance Scheme Using Monocular Vision Applied to Mobile Robots

Yun Yun Song¹, Sheng Yi Yang*², Chao Liu³, Qing Gui Wu⁴

¹School of Data Science and Information Engineering
²Key Laboratory of Pattern Recognition and Intelligent Systems of Guizhou Province, Guizhou Minzu University Guiyang, China
³Key Laboratory of Pattern Recognition and Intelligent Systems of Guizhou Province, Guizhou Minzu University Guiyang, China
⁴School of Mechatronics Engineering, Guizhou Minzu University Guiyang, China

* Corresponding author: yangyshy@163.com

Abstract—Mobile robots cannot move in an unknown environment with static or slow-moving obstacles effectively. We present an enhanced obstacle avoidance strategy using monocular vision to solve this problem. First, we combine Canny and Otsu to extract the barrier feature and find the critical pixel position of obstacles by the monocular vision. Then the image depth estimation algorithm is used to estimate the gaps. With these parameters of barriers, an improved bug algorithm is proposed to avoid the obstacles autonomously. The experiments show that the proposed obstacles avoidance strategy can effectively make a small mobile robot avoid different kinds of obstacles.

1. Introduction
Robotics and automated driving technology overgrow. The autonomous obstacle avoidance of Mobile robots has become a research hotspot in recent years. Some researchers put forward different obstacle avoidance schemes that used various ultrasonic sensors to get obstacle information [1-2]. These methods can meet the requirements in certain circumstances, but there are limitations in practical rescue or exploration. The system’s accuracy and reliability can be improved by integrating ultrasonic and infrared sensor data for ranging. But the infrared sensor is susceptible to ambient light interference, and its sampling period is too long [3-6]. Tang uses laser radar scanning as an avoidance sensor [7]. Its accuracy and real-time performance are improved significantly. But it is limited by the height of the scanning plane. Liu’s mobile robot obstacle avoidance method is based on Q learning. It is hard to apply to nonlinear obstacles [11]. Wang uses binocular vision to detect static barriers [8]. It has high precision and reliability. Li proposed a distance model based on a ground plane to measure obstacles [9]. It has not been applied to a practical obstacle avoidance system. Hao combined obstacle detection and obstacle avoidance algorithm. It is verified by LABVIEW [10]. Benn uses monocular vision to control autonomous navigation in the dynamic environment [12], and this solution is focused on using color segmentation to differentiate obstacles. Hong collected information by infrared sensors, such as modelling human bodies, realizing monocular visual motion target recognition and tracking. Still,
obstacle avoidance needs to be improved when the obstacle is small [14]. Therefore, we present an autonomous obstacle avoidance scheme (AOBAS) using monocular vision to cope with new scenes. It is consists of obstacle feature extraction, image depth estimation, and obstacle avoidance strategy. We apply AOBAS to a small mobile robot. Results show that the robot can detect many obstacles that vary in shapes, sizes, and colors. It can effectively implement autonomous obstacle avoidance.

2. Autonomous obstacle avoidance scheme

2.1. Autonomous Obstacle Avoidance Scheme

Our AOBAS is suitable for small robots moving on flat ground. It is shown in Fig. 1. The core of AOBAS is using monocular vision, and it is reflected in three aspects. First of all, the obstacle feature extraction includes obstacle feature detection and critical pixel point scanning. The obstacle feature detection has adopted the fusion concept to get edges of those obstacles. Using the edge can output coordinate information of the critical pixel of obstacles. Next, image depth estimation can calculate the distance between the test platform and the obstacle in real-time. At last, an improved Bug algorithm judges whether the robot is in a safe area so that it can output the control signal to help the robot in bypassing the obstacle.

![Figure 1](image_url) The scheme of autonomous obstacle avoidance.

2.2. Obstacle Feature Extraction

The image processing of target characteristics is essentially in the autonomous obstacle avoidance system. Using the features detection can get an edge about obstacles and identify the intersecting line between obstacles and ground [14]. According to the intersecting line, the image needs to be divided into security areas and dangerous areas to calculate the critical pixel point.

Canny and Otsu are combined to realize the edge feature detection and separation of obstacles. Otsu is based on probability statistics, and Canny is to perform convolution on the image. Otsu is more accurate when the object is not different from the background, but its color is bright. Otsu cannot be separated from the obstacle beyond a safe distance when obstacles occupy a small part of the image. But Canny can separate obstacles effectively when they differ significantly from the background. When the obstacles’ pixels increase, obstacle avoidance will fail if the distance between the obstacle and the robot is less than the given safe distance. Therefore, we find out the image’s standard deviation in experiments when Otsu cannot be separate obstacles to improve the obstacle feature detection to prevent image information loss caused by edge detection of grey-scale images. The larger the standard deviation of the grey-scale image is, the more discrete the number of the image’s grey-scale level is. We get the result by many experiments that the separation effect is better when the standard deviation of each grey level of the grey image is set as 35. The flow chart is shown in Fig. 2.
The pixel coordinate of the obstacle’s nearest point plays an important role in our scheme. Feature extraction can get the intersection line between obstacles and ground. Then the nearest point from the camera can be found on this intersection line. This point is called the critical pixel point. This point is determined mainly by the pixel coordinate point scanning. Scanning method is scanned column by column and line by line in binary image. The input parameters of our image depth estimation algorithm can be provided by this critical pixel coordinates.

The obtained \( m \times n \) binary pixel image is denoted as the \( m \times n \) matrix, as in (1),

\[
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{bmatrix}
\tag{1}
\]

Where \( a_{ij} \) is an image pixel value in \( A_{m \times n} \). If there are elements \( a_{ij} \) that are suitable for (2),

\[
a_{ij} = 255(0 < i \leq m, 0 < j \leq n).
\tag{2}
\]

We write the subscripts \( i \) of these elements to the set \( I_1 \) and the subscripts \( j \) of these elements to the set \( I_2 \), and these need to be suitable for (3),

\[
I_1 = \{i \in \mathbb{N}^+ \mid 1 \leq i \leq n\}, \quad I_2 = \{j \in \mathbb{N}^+ \mid 1 \leq j \leq m\}.
\tag{3}
\]

Then we can calculate the maximum of \( I_1 \) and \( I_2 \). These maximums are denoted as \( x_{\text{max}} \) and \( y_{\text{max}} \). Using the maximums, we can get the critical pixel point \( a_{x_{\text{max}}, y_{\text{max}}} \).

2.3. Image Depth Estimation

Image depth estimation is mainly to judge the robot’s current state utilizing critical pixel coordinate. The essence of visual depth estimation is that it can establish the geometric proportion relationship.
between the test point and the conversion relationship between coordinate systems and camera imaging model [9]. The image depth estimation model is shown in Fig. 3. The focal length of the camera is perpendicular to the virtual image plane. The farther the obstacle is from the camera, the closer the nearest point is to the top of the image. Therefore, the distance of the obstacle can be estimated when it nears the bottom of the image.

![Figure 3 Model of the image depth estimation.](image)

As shown in Fig. 3, $O_w - X_wY_wZ_w$ is world coordinate system, $O_c$ is the camera origin point, the critical pixel point $M(X'_o, Y'_o, 0)$ is located in the plane $X_oY_o$, $M'$ is a projection where $M$ is located in $Y_o$, $m$ and $m'$ are the corresponding points of them in the coordinate system [13]. $\theta$ is the angle between $O_cC$ and $Y_w$, and $f = |O_cC'|$ is the camera focus, $S = |O_oM|$ is the distance between camera and obstacle, $u_m$ is an abscissa of the pixel spot $m$, $h$ is the camera height. According to this geometric relationship, the following relations can be obtained. As in (5), $u_0$ is the image resolution. $d_x$ is the physical length of the pixel. And its unit is mm.

$$\beta = \frac{\pi}{2} - \theta + \phi,$$

$$X'_o = \frac{|O_oM'|}{|O_cM|} |mm'| = \frac{h \cos \phi}{f \cos \beta} (u_m - u_o)d_x,$$

$$Y'_o = |O_oM'| = h \tan \beta.$$

As in (7), the distance between the test platform and the critical pixel point of obstacle $M$ can be solved.

$$S = \sqrt{(X'_o)^2 + (Y'_o)^2}.$$  

2.4. Obstacle Avoidance Strategy

Our AOBAS is designed to solve the problem of the robot effectively avoids obstacles. The obstacle avoidance strategy aims to achieve autonomous obstacle avoidance. Considering the Bug pathfinding [13], we present an effective avoidance strategy. It can generate control signals based on the judgment results whether the robot is at a safe distance. The robot can perform corresponding avoidance actions to bypass obstacles. Our strategy is shown in Fig. 4.
Step1: robot is obtained the coordinates of obstacle distance estimation called \( S \) and the critical pixel point called \( O \).
Step2: starting loop 1.
Step3: comparing the relationship between \( S \) and the given safety distance, When \( S \) is too large, the mobile robot moves forward in a straight line.
Step4: otherwise, running loop 2.
Step5: comparing distance between the point \( O \) and the image.
Step6: robot is turned the steering gear left if the point \( O \) is on the right side of the image.
Step7: otherwise, robot is turned the steering gear right.
Step8: ending loop 2.
Step9: robot is gone forward at low speed.
Step10: running loop 1 until the task is ended.

Figure 4 Obstacle avoidance strategy.

3. Experiments & results

3.1. Test Platform Construction
Our robot testbed is shown in Fig. 5. OpenMV3 is chosen as the vision sensor. The height of this OpenMV3 is 19cm, and its pitch angle is 20° downward.

Figure 5 Robot testbed.

3.2. Obstacle Feature Extraction
Fig. 6 (a) and Fig. 7 (a) are the grey image of two various scenarios. Their results are shown in Fig. 6 (b) and Fig. 7 (b). Results show that our algorithm can be effectively divided into security area and dangerous area in an image. It’s convenient to find the critical pixel point.

Figure 6 Results of obstacle feature extraction.

Figure 7 Results of obstacle feature extraction.

3.3. Image Depth Estimation
The test ranges are from 16cm to 70cm. The results are shown in Table 1. Due to the influence of lens distortion, the farther the distance is, the greater the relative error between the measurement and the real value will be. When the distance is less than 30cm, the relative error will be smaller.
Table 1 Results of Image Depth Estimation.

| Times | Corresponding Point Pixel Coordinates (u, v) | Monocular Estimated Value (cm) | Actually Measured Value (cm) | Relative Error (%) |
|-------|---------------------------------------------|--------------------------------|-----------------------------|--------------------|
| 1     | (50,117)                                    | 19.9                          | 18.1                        | 9.94               |
| 2     | (77,112)                                    | 19.1                          | 17.8                        | 7.30               |
| 3     | (60,105)                                    | 20.4                          | 20.5                        | -0.49              |
| 4     | (67,85)                                     | 22.9                          | 26.4                        | -13.2              |
| 5     | (91,64)                                     | 29.4                          | 33.9                        | -13.27             |
| 6     | (92,47)                                     | 40.6                          | 46.2                        | -15.12             |
| 7     | (91,64)                                     | 46.7                          | 52.7                        | -11.39             |
| 8     | (67,85)                                     | 48.9                          | 54.8                        | -10.77             |
| 9     | (60,105)                                    | 51.8                          | 58.4                        | -11.30             |
| 10    | (77,112)                                    | 52.3                          | 57.6                        | -9.20              |
| 11    | (50,117)                                    | 61.4                          | 73.6                        | -16.57             |

Before starting the mobile robot, it needs to measure the distance between the camera and the obstacle, which was regarded as the given value of safe distance called d in the program. If the mobile robot approaches the obstacle, when the calculated value of the image depth estimation algorithm is less than d, the distance between the mobile robot and the obstacle is measured. The results of the estimated distance of the camera and the measured values during obstacle avoidance are shown in Table 2. Due to the existence of measurement error, through the analysis of experimental data, when the safety distance is between 20cm and 40cm, the value of relative error is smaller, and the turning radius of the mobile robot is 25cm. If less than 25, the robot will be increased the risk of hitting obstacles. To ensure safety, 35cm is selected as the obstacle avoidance distance.

Table 2 Camera Distance During Obstacle Avoidance.

| Times of Measurement (times) | The Actual Distance from the Obstacle (cm) | Given Safe Distance (cm) | Obstacle Avoidance Action Distance (cm) | Relative Error (%) |
|-------------------------------|--------------------------------------------|--------------------------|----------------------------------------|--------------------|
| 1                             | 47.9                                       | 35                       | 38.7                                   | -9.56              |
| 2                             | 55.4                                       | 35                       | 39.3                                   | -10.94             |
| 3                             | 53.5                                       | 30                       | 34.5                                   | -13.04             |
| 4                             | 56.4                                       | 30                       | 35.6                                   | -15.73             |
| 5                             | 58.5                                       | 25                       | 24.6                                   | 1.62               |
| 6                             | 48.7                                       | 25                       | 27.5                                   | -9.09              |

3.4. Obstacle Avoidance test

The results of obstacle avoidance in problematic situations are shown in Fig. 8. The obstacle attributes and positions are placed randomly. What using the fixed camera is recorded the experimental video can analyze the results of obstacle avoidance of the mobile robot. The experimental images are stacked frame by frame to show the obstacle avoidance trajectory. When obstacles don’t be detected, the path of the mobile robot is a straight line. When obstacles are detected, and the distance is less than a safe distance, the mobile robot will take measures to avoid obstacles. The mobile robot in the figure does not carry out path planning, merely performs detection and obstacle avoidance actions on the path. Through experimental tests, the mobile robot can avoid obstacles with different properties on the path. To ensure safety, the distance between obstacles should be greater than the turning radius of the mobile robot.

![Image of obstacle avoidance](image_url)
4. Conclusion

The obstacle feature detection algorithm in this paper has mainly combined the Otsu with the Canny to detect the edge of obstacles. This way can improve the obstacle separation effect further. The position information of the critical pixel point can be obtained by the pixel scanning of the critical pixel point algorithm. The distance between the test platform and the obstacle can be estimated by the image depth estimation algorithm so that the mobile robot can avoid obstacles efficiently. The obstacle avoidance system in this paper has the characteristics of conciseness, effectiveness and real-time. The obstacle avoidance strategy can effectively enhance the obstacle avoidance effect of the mobile robot. This system can be applied to some small mobile robots on flat ground in an unknown environment.

Acknowledgments

The authors gratefully acknowledge the contribution of the Machine Intelligence Group and Dr. Yang Yonghuai of Guizhou Minzu University.

Science and Technology Foundation of Guizhou Province, Grant NO. [2017] 1088 and Grant NO. [2018] 5773-YB02.

References

[1] P. H. Wu, D. X. Lu, Z. H. Luo, L. Mo, S. H. Xiao, “Development and application of ultrasonic obstacle avoidance technology for intelligent robots,” Machine Tool & Hydraulics, vol. 47, no. 9, pp. 46-50+57, 2019.

[2] R. D. Yang, L. J. Yang, Y. F. Wang, H. Y. Zhang, “Design of intelligent obstacle avoidance vehicle based on android system,” Sensors and Microsystems, vol. 37, no. 3, pp. 81-84, 2018.

[3] P. Ma, Y. Zhang, P. J. Su, S. W. Liu, “Design of obstacle avoidance system based on ARM embedded AGV,” Machine Tool & Hydraulics, vol. 47, no.5, pp. 61-64+71, 2019.

[4] H. M. Ge, X. H. Xu, Z. H. Gu, J. L. Zhang, “Design of obstacle avoidance system of intelligent car based on Arduino,” Modern Electronics Technology, vol. 37, no. 11, pp. 118-120, 2014.

[5] Y. E. Zhao, Z. Q. Wu, “Design and implementation of dual-mode Intelligent obstacle avoidance vehicle system based on Arduino,” Modern Electronic Technology, vol. 40, no. 21, pp. 94-97, 2017.

[6] T. K. Xia, M. Yang, R. Q. Yang, “Advances in navigation algorithms for mobile robots based on monocular vision,” Journal of Control and Decision, vol. 25, no. 1, pp. 1-7+19, 2010.

[7] W. X. Tang, H. Yan, “Design and implementation of obstacle avoidance system based on two-dimensional lidar,” Computer Measurement and Control, vol. 24, no. 10, pp.202-204+208, 2016.

[8] Z. Wang, X. Zhao, H. J. She, H. H. Liu, Y. W. Zhao, “Obstacle detection and obstacle avoidance of AGV based on binocular vision,” Computer Integrated Manufacturing System, vol. 24, no. 2, pp.400-409, 2012.

[9] Q. Li, “Monocular vision real-time ranging algorithm research,” Harbin Institute of Technology, no.3, 2014.

[10] H. Q. Hao, “Obstacle avoidance research based on monocular vision mobile robot”. Taiyuan University of Technology, no.3, 2016.

[11] H. Liu, J. Wang, J. F. Li, J. N. Li, “Research on robot obstacle avoidance design in unknown environment,” Mechanical Design and Manufacturing, no.10, pp.236-238, 2013.

[12] W. Benn and S. Lauria and Z. Wang. "Robot Navigation Control Based on Monocular Images: An Image Processing Algorithm for Obstacle Avoidance Decisions," Mathematical Problems in Engineering, pp. 732-748, 2012.

[13] Y. Peng, L. Z. Bao, D. Qu, Y. M. Xie, “Multi-bug Global Path Planning Algorithm Research,” Journal of agricultural machinery, vol.51, no.6, pp.375-384, 2020.

[14] Y. C. Hong, W. Wei, S. S. Du, L. T. Zhu, “Research on infrared obstacle avoidance and monocular vision tracking of mobile robots,” Mechanical and Electrical Engineering, no.06, pp.60-62, 2006.