Original Article

An Obstacle Avoidance Approach Based on Naive Bayes Classifier

Peiqiao Shang[1], Wenqian Li[1]*

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

ABSTRACT

Obstacle avoidance plays an important role in mobile robot. However, the traditional methods of obstacle avoidance have difficulty in distinguishing multiple obstacles by edge detection. In this paper, the traditional obstacle avoidance methods are improved to realize the function of multi-obstacle avoidance. Regarding the implementation process, the LiDAR is used instead of the camera, which reduces the difficulty of handling image noise and achieves reliable obstacle detection. It can accurately detect the borders of the nearest obstacle even in complex environments and perform obstacle avoidance. Regarding the obstacle avoidance prediction, the model training is performed through the Naive Bayes classifier based on the three attributes of the velocity of the robot, the left boundary of the obstacle and the right boundary of the obstacle. In the training process, dataset was expanded to enhance the accuracy of classifier model. When the robot goes forward, the improved method enables the robot to move at a higher velocity. The results show the feasibility of advanced obstacle avoidance method by simulation.

Keywords: *Robot Obstacle Avoidance; Naive Bayes; LiDAR; Gazebo Simulation*

ARTICLE INFO
Received: Jul 24, 2020
Accepted: Sep 7, 2020
Available online: Sep 8, 2020

***CORRESPONDING AUTHOR**
Wenqian Li, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; lwq_english@163.com;

CITATION
Peiqiao Shang, Wenqian Li. An Obstacle Avoidance Approach Based on Naive Bayes Classifier. Journal of Autonomous Intelligence 2020; 3(1): 27-37. doi: 10.32629/jai.v3i1.139

1. Introduction

1.1 Research background and significance

With the progress of the times, robots are replacing humans to play a major role in many fields. Compared with traditional robots (such as mechanical arms, etc.), mobile robots are more widely used in industry and daily life because they are capable of environmental awareness and motion control. Therefore, the mobile robot has gradually become the most popular research field of robots at home and abroad. Obstacle avoidance refers to the behavior of the mobile robot, according to the collected status information of the obstacle, to effectively avoid obstacles with a certain method and finally reach the target point when it senses static and dynamic objects that blocks its path[1].

Currently, there are two ways to achieve mobile robots' obstacle avoidance: 1. Command the robot to create a map of its environment, and then navigate and avoid obstacles based on the completed map. 2. Directly use the relevant sensors to detect obstacles, and then issue commands to the robot based on relevant experience to avoid the obstacles encountered. This paper mainly uses the second method, focusing on how to achieve accurate obstacle avoidance without creating an accurate environment map. Finally, the Gazebo platform is used to verify the validity of this research. On the one hand, it can promote the development and application of the Naive Bayes theory. On the other hand, it can achieve intelligent obstacle avoidance of the mobile robots and ensure their safety during operation. This research has both theoretical significance and practical application value.

1.2 Literature review

In 2016, the research of He Ming *et al.*[2] showed that the Naive Bayes Classifier has good classification performance in the case of low entropy, especially in the case that the dataset has completely independent features and function-related features. In 2004, Li Yi and Cai Zixing[3] studied obstacle avoidance of the mobile robot based on the Bayes Classifier and reached the conclusion that the Naive Bayes Classifier has high accuracy of classification in studying obstacle avoidance of the mobile robot. The mobile robot AmigoBot was used to verify the practicality of the Naive Bayes Classifier in robot obstacle avoidance. In 2020, the research of Chen Yi *et al.*[4] showed that in vehicle detection the camera is susceptible to factors such as the light and the detection distance. The LiDAR has the advantages of long detection range, being uninfluenced by the light, and the ability to accurately obtain the target distance information. In summary, robot obstacle avoidance based on LiDAR and Naive Bayes Classifier has strong feasibility and research value.

1.3 Related Work

Although the above-mentioned researches have achieved some outcomes in mobile robot obstacle avoidance, there are still shortcomings as follows: 1. the training set of the Naive Bayes Model is small, which fails to address the large number of unknown situations in the actual obstacle avoidance of the robot. 2. camera-based robot obstacle avoidance depends heavily on the environment, and cannot perform well as expected if there is a large amount of noise in the environment. 3. camera-based robot obstacle avoidance cannot achieve the expected effect in the case of overlay imaging of different obstacles near and far. In response to the three major problems, this paper proposes an obstacle avoidance model that uses LiDAR for collection of obstacle information and Naive Bayes Classifier for robot motion decision-making, so as to solve the problem of obstacle avoidance in more complex environments. Finally, the Gazebo platform is used to verify the validity of the research, which has important guiding significance for the application of supervised machine learning in the field of robot obstacle avoidance.

2. Design of the Robot Obstacle Avoidance Scheme

2.1 Detection of obstacle boundary

2.1.1 Based on Roberts operator

The Roberts operator is a partial-differential method of calculating the gradient. The gradient magnitude indicates the edge strength of the obstacle, and the gradient direction is orthogonal to the edge direction of the obstacle[5,8].

The symbols used in the Roberts operator method and their definitions are shown in **Table 1**.

Table 1. Symbols used in the Roberts operator and their meanings

Symbol	Meaning
x, y	Pixel coordinates
$f(x,y)$	Grayscale
$G(x,y)$	The gradient-based operator that computes the sum of squares

The definition of gradient operator is:

$$\mathrm{G}(x,y)=\sqrt{\nabla_x f(x,y)^2+\nabla_y f(x,y)^2} \tag{1}$$

To make the calculation simpler, the gradient operator can be approximated as:

$$\mathrm{G}(x,y)=|\nabla_x f(x,y)|+|\nabla_y f(x,y)| \tag{2}$$

From the above formula, we can conclude that the diagonal Roberts operator for image discretization is:

$$\nabla_x f(x,y)=f(x,y)-f(x-1,y) \tag{3}$$

$$\nabla_y f(x,y)=f(x,y)-f(x,y-1) \tag{4}$$

Analysis of advantages and disadvantages: The advantages of using the Roberts operator to process images are that the calculation is simple and the detection of the image edge is more accurate. However, the Roberts operator is very sensitive to noise, so it is necessary to de-noise in advance.

2.1.2 Based on LiDAR

Since the effect of using the Roberts operator to detect the obstacle boundary does not meet our expectation, we then choose the data of LiDAR returns in Gazebo simulation to calibrate the obstacle boundary.

First read the depth information in the LiDAR return values, and then filter out the depth information in the range of -90°~+90°, i.e. in front of the robot. Find the minimum value of the obtained depth information, search to the left and the right of that minimum value to find the points with abrupt changes in the depth, and then determine the boundary information of the nearest obstacle that is located in front of the robot.

2.2 Obstacle avoidance decision-making based on the Bayes Classifier

The diagram of the Mixed Naive Bayes Classifier is shown in **Figure 1**.

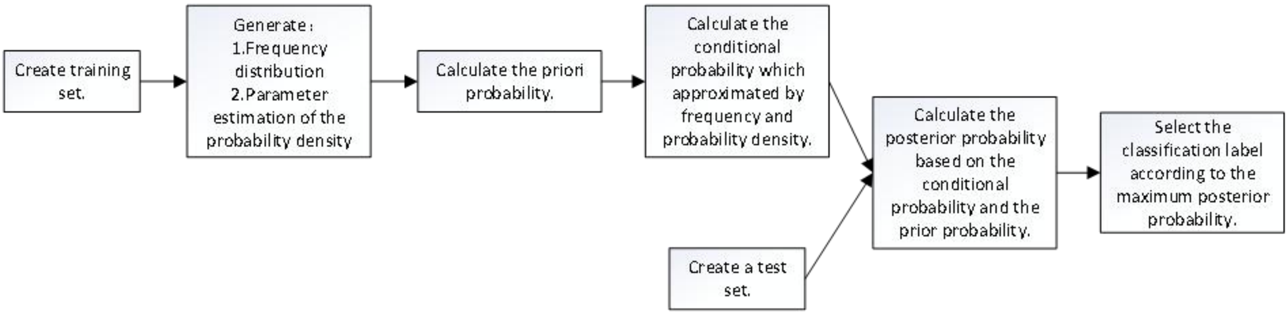
Create training set.
Generate:
1.Frequency distribution
2.Parameter estimation of the probability density
Calculate the priori probability.
Calculate the conditional probability which approximated by frequency and probability density.
Calculate the posterior probability based on the conditional probability and the prior probability.
Select the classification label according to the maximum posterior probability.
Create a test set.

Figure 1. Diagram of the Naive Bayes Classifier.

2.2.1 Dataset processing

Since the amount of data in the literature[3] is insufficient, we expand the dataset.

Suppose the maximum range of the LiDAR is $h_{\max}$, the actual width of the robot is d , and a safe distance d_p is reserved on both sides. Then $d+2d_p$ is the width of the robot. Thus, when there is an obstacle within the range of $d+2d_p$ in front of the robot, obstacle avoidance is required. From the maximum range and scanning angle of the LiDAR, it can be known by equation (5).

$$d+2d_p = 2h_{\max}\cos\frac{\alpha}{2} \tag{5}$$

The scanning range of α in front of the robot is equivalent to the front of the robot, where there must be no obstacles, otherwise obstacle avoidance is required. When the obstacle avoidance environment requires a significant change of direction, i.e., when turning, the value β within the range of $(\frac{\alpha}{2},\frac{\pi}{2})$ can be taken as the threshold of turning.

The schematic diagram is shown in **Table 2**.

Table 2. Rules of datasets expansion

Range: $0 < \frac{\alpha}{2} < \beta < \frac{\pi}{2}$

Position of the obstacle center	Boundary of the nearest obstacle	β	Turn
Right	$(\frac{\alpha}{2},\frac{\pi}{2})$	Low velocity β > high velocity β	Go forward
	$(-\beta,\frac{\alpha}{2})$		Left
	$(-\frac{\pi}{2},-\beta)$		Reverse
Left	$(\beta,\frac{\pi}{2})$		Reverse
	$(-\frac{\alpha}{2},\beta)$		Right
	$(-\frac{\pi}{2},-\frac{\alpha}{2})$		Go forward

In addition to expanding the dataset, we also improve and preprocess the dataset. Regarding the preprocessing, we mainly divide the velocities into different gears; $0 \le v < 0.04$ is classified as Gear 0, $0.04 \le v < 0.08$ as Gear 1, and so on (shown in **Table 3**). As for the improvement on the dataset, we mainly take into account the moving velocity of the robot. Combined with the actual situation, the reaction time of the robot varied at different velocities. When the robot is moving at a high velocity, it has very short reaction time. In this case, we design the dataset as such that tends to make the robot reverse. When the robot is moving at a low velocity, it has sufficient reaction time. In this case, we design the dataset as such that tends to make the robot turn to the left or right. Through the improvement on the dataset, we make the moving of the robot more reasonable.

Table 3. The relationship between velocity and gear

Velocity	Gear
<0.04	0
0.04~0.08	1
0.08~0.12	2
0.12~0.16	3
0.16~0.20	4
0.20~0.25	5
>0.25	6

2.2.2 Bayes model theory

The symbols used in the Naive Bayes Classifier[2] and their definitions are shown in **Table 4**.

Table 4. Symbols used in the Naive Bayes Classifier and their meanings

Symbol	Meaning
Y	The event of turning
X	The robot's velocity and the angle of the obstacle's left and right boundaries
c_k	The type of turning of Class k
n_k	The total number of samples with the type of turning of Class k in the sample set
N	Total number of the samples

The principle of the Bayes Classifier:

1. Calculate the priori probability

The formula for calculating the priori probability is shown in equation (6), from which the probability of the robot turning left, turning right, going forward and reversing is calculated respectively.

$$P(Y=c_k)=n_k/N \tag{6}$$

2. Calculate the likelihood

For each test data input, it is necessary to calculate the likelihood of turning left, turning right, going forward and reversing. According to the principle of conditional independence, equation (7) holds.

$$P(X^{(j)}=x^{(j)}\mid Y=c_k)=\prod_{\mathrm{i}=1}^{n}P(X^{(j)}=x^{(j)}\mid Y=c_k) \tag{7}$$

3. Calculate the posterior probability

Calculate the posterior probability of turning left, turning right, going forward and reversing respectively according to formula (8), and select the command corresponding to the maximum value as the output of the naive classifier.

$$P(Y=c_k|X^{(j)}=x^{(j)})=\frac{P(X^{(j)}=x^{(j)}\mid Y=c_k)}{P(X=x)} \tag{8}$$

Through the Bayes Classifier, we can import the results of obstacle detection obtained from the LiDAR depth information into the Bayes Classifier, and then get the next change of the robot's command.

3. Simulation Experiment

3.1 Design of the Gazebo simulation experiment

3.1.1 Design of the robot

The Turtlebot3 robot is a small, low-cost, programmable mobile robot based on ROS and has the camera and LiDAR required for obstacle detection discussed in Chapter 2. In addition, the official Turtlebot3[7] provides a Gazebo simulation model with precise physical properties. Using this robot as a simulation object can restore the reality to the best extent possible. From **Table 5**, we select the Waffle model as the robot for the simulation experiment.

Table 5. Specifications of Turtlebot3 robot models (excerpt)[7]

Items	Burger	Waffle	Waffle PI
Maximum translational velocity	0.22m/s	0.26m/s	0.26m/s
Maximum rotational velocity	2.84rad/s	1.82rad/s	1.82rad/s
Camera	-	Intel® Realsense™ R200	Raspberry Pi Camera Module v2.1
LDS	360 Laser Distance Sensor LDS-01	360 Laser Distance Sensor LDS-01	360 Laser Distance Sensor LDS-01

This robot model has important nodes such as velocity control and LiDAR scanning. Through these two nodes, you can obtain the speed of the robot's response and the LiDAR ranging information. At the same time, it provides feedback based on the processing of the two kinds of data and changes the direction of the robot's path, so as to achieve obstacle avoidance. The ROS message types of velocity and LiDAR are shown in **Figure 2** and **Figure 3** respectively.

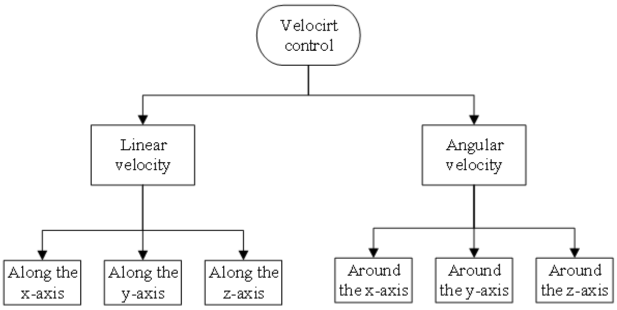

Figure 2. ROS message type of velocity.

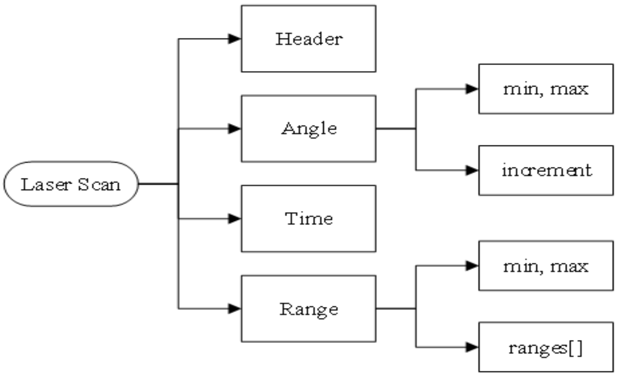

Figure 3. ROS message type of LiDAR.

3.1.2 Design of the robot simulation environment

We use the official turtlebot3_world simulation environment (see **Figure 4**), which is a closed area surrounded by hexagons and with 9 cylinders inside. Although the types of the obstacles in this environment are monotonous, but the obstacles' number is large and the arrangement is complicated. Therefore, this environment can be used to validate the feasibility of the above-mentioned obstacle avoidance method.

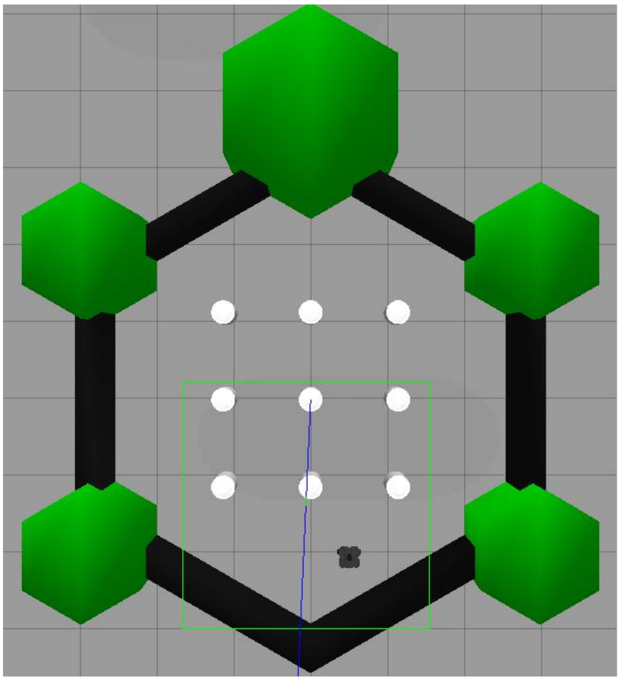

Figure 4. Robot simulation environment.

3.1.3 Design of the nodes and communication model

After data acquisition and motion control are realized, the trained Bayes model can be connected to the communication system to realize data processing and motion control prediction.

Figure 5 shows the designing graph of using ROS to implement robot obstacle avoidance. The circles represent nodes and the boxes represent topics[9,10].

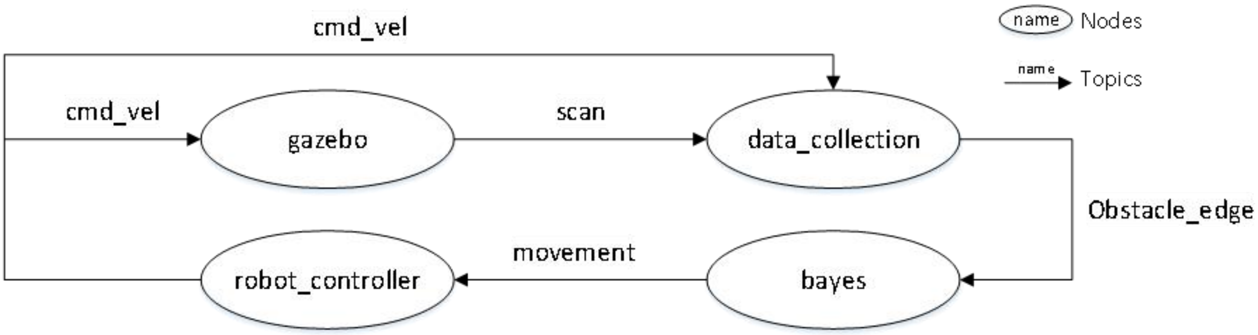

Figure 5. Diagram of the designing of robot obstacle avoidance.

The data composition of the LiDAR scanning node scan is more complicated. However, because the place right ahead of the robot is fixed, it only needs to take the LiDAR ranging information in that direction to meet the needs of obstacle detection[6].

The node cmd_vel is the velocity control command, which mainly receives the control command from prediction and transmits the current moving velocity to data_collection, which is the data collection node.

Gazebo are mainly used for the simulation of robot models. The scan topic is LiDAR scanning, which is used to transmit the depth information directly in front of the robot to data_collection[7].

Data_collection is not only used to obtain the original data, but also undertakes the task of data preprocessing. Through the gradient method, the left and right boundaries of the nearest obstacle are selected from the depth information, and then combined with the current velocity of the robot to form a sample, which is passed to the bayes node, i.e. the Bayes Classifier, and performs obstacle avoidance prediction.

The node bayes is the Naive Bayes Classifier. By receiving boundary attributes of the obstacle and velocity of the robot, it invokes the trained model to perform prediction of the direction of moving.

The node robot_controller receives the prediction result of the Bayes Classifier and compares it with the previous prediction result. In the case that the robot receives a command of going forward when it is going forward, its velocity will increase. Otherwise, the robot will move at a low velocity. The node also forms control commands to be passed to cmd_vel for robot simulation.

3.2 Experimental results and discussion

3.2.1 Results and Analysis of Boundary Detection

Figure 4 shows the initial position of the robot in the simulation environment. **Figure 6** shows the two-dimensional image taken by the Camera under specific lighting conditions when the robot is in the initial position. **Figure 7** shows the edge image processed by the Roberts operator. **Figure 8** shows the distance information of the obstacles within the range of -90°~+90° in front of the robot detected by the LiDAR in the same position.

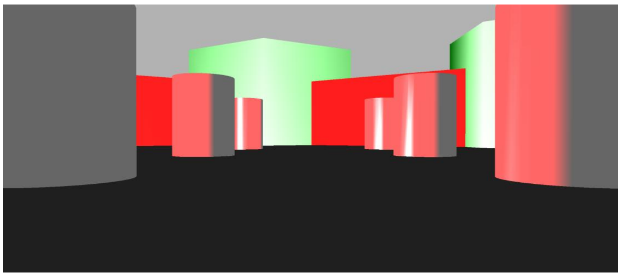

Figure 6. The two-dimensional image taken by the Camera under specific lighting conditions.

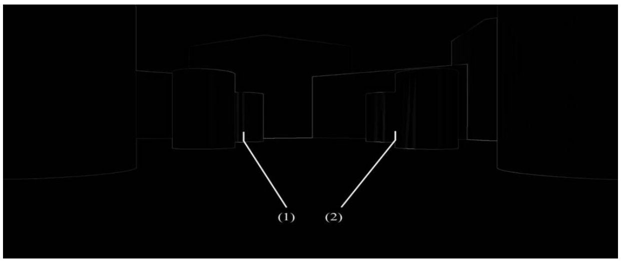

Figure 7. The edge image processed by the Roberts operator.

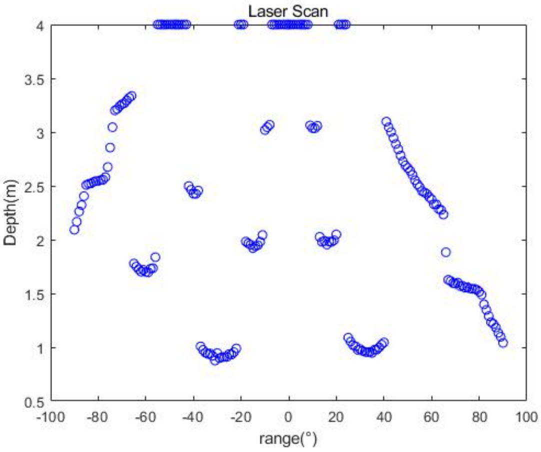

Figure 8. The distance information of obstacles detected by the LiDAR.

It can be seen from **Figure 7(1)** that the lighting conditions affect the detection results of the Roberts method, resulting in cases where non-boundaries may be detected as obstacle boundaries. **Figure 7(2)** shows the case where the Roberts method cannot detect the boundary of the obstacle due to the overlapping of obstacles. However, the obstacle detection of the LiDAR in **Figure 8** will not be affected. (Shown in **Table 6**)

In terms of the data volume, the Camera is presented in the form of 60FPS 1280*720, while the LiDAR is presented in the form of 60FPS 2*360. When we compare the size of the collected data, it can be seen that the size of data in one frame collected by the LiDAR is much smaller than that of the Camera. The LiDAR has lower requirement of the processor's performance and higher efficiency of computing. (Shown in **Table 6**)

Table 6. Comparison of the Camera and the LiDAR

	Camera	**LiDAR**
Extent of being affected by the environment	Large	Small
Effect of detecting complex obstacles	Poor	Good
Data size (one frame)	2700KB	2.39KB

3.2.2 Results and analysis of the naive bayes classifier

Because Li Yi's[3] dataset with a large number of unknown conditions leads to wrong commands to the robot, the training results obtained through the Naive Bayes Classifier are not ideal, with an accuracy rate of 10 samples only 80%. After the dataset is expanded, a large number of obstacle avoidance scenarios are supplemented, and the accuracy rate is increased to 90%. (Shown in the second line of **Table 7**)

By comparing with the classic machine learning algorithm BP neural network (shown in **Table 4**), after adjusting the parameters of the neural network, it is found that when the neural network is over fitted, that is, when the training accuracy is high, some serious prediction errors (i.e. the confusion between going forward and reversing) will occur. When the degree of fitting is reduced, although there are less serious prediction errors, the overall accuracy also drops to 88%. To summarize the comparison between the Naive Bayes Classifier and the BP neural network, although the accuracy of the BP neural network is slightly higher than that of the Naive Bayes Classifier in the case of no or very few serious prediction errors, we still prefer the Naive Bayes Classifier in order to avoid fatal errors and reduce the complexity of the algorithm. (Shown in the third and fourth lines of **Table 7**)

Table 7. Comparison of algorithms

	Naive Bayes[3] **(Camera, with an insufficient data set)**	**Naive Bayes** **(LiDAR, with a sufficient data set)**	**BP NNs** **(LiDAR, with a sufficient data set)**
Total samples	A large number of unknown conditions	1308	1308
Accuracy (10 samples)	80%	90%	90%
Test samples	---	392	392
Accuracy (Test samples)	---	87.8%	88.0%

3.2.3 Results and analysis of the robot obstacle avoidance experiment

After training the Naive Bayes Classifier, the trajectory diagram of the robot obstacle avoidance simulation is shown in **Figure 9**.

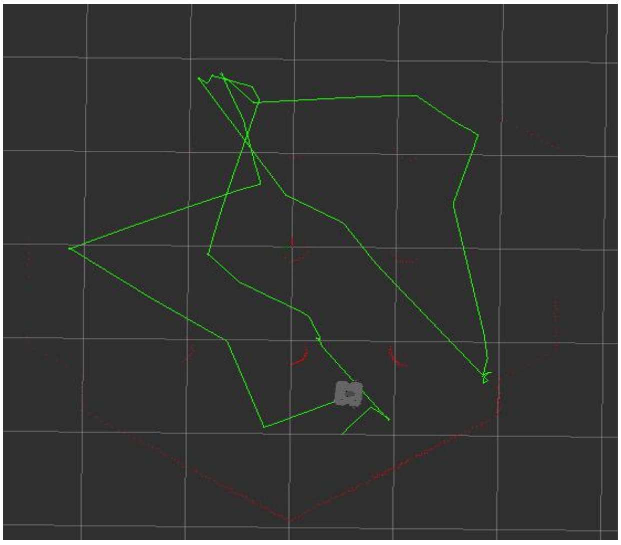

Figure 9. Trajectory diagram of the robot obstacle avoidance simulation.

Compared with camera method, the LiDAR-based method of obstacle detection can sense the relative positions between the robot and the nearest obstacle very well. The robot can sense which part of the range of -90°~+90° is occupied by the nearest obstacle, and the results of this part will be used as input data to the Bayes Classifier[3,6].

The red point cloud in **Figure 9** represents the obstacle detected by the robot LiDAR. The green line represents the path that the robot has traversed. **Figure 9** shows that the Waffle robot can slow down in advance and make the desired obstacle avoidance action when encountering an obstacle. Besides, when there is no obstacle, it can gradually accelerate to the maximum velocity (0.26m/s). The simulation experiment proves that the robot obstacle avoidance algorithm proposed by this paper is quite successful in most cases.

However, the situation shown in **Figure 10** may appear during the experiment: the robot collides with the obstacle. There is a main reason for the serious collision: The Naive Bayes Classifier issues an incorrect command, causing the robot to hit an obstacle.

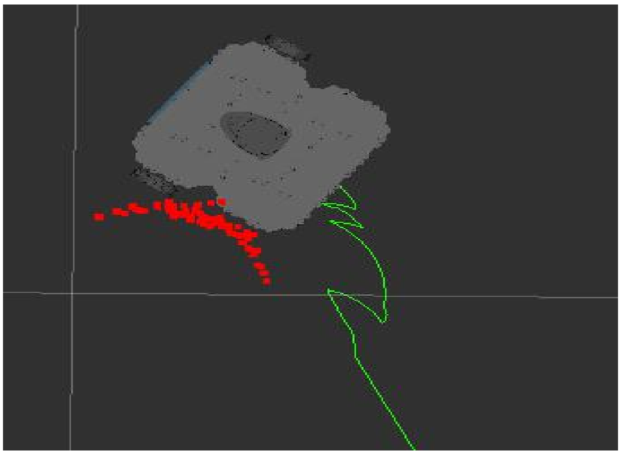

Figure 10. The collision between the robot and the obstacle.

To prove that the probability of collision is extremely low, we did a supplementary experiment.

Table 8. Collision analysis

	Value
Obstacle avoidance experiment	466
Collision caused by classification error	1
Collision probability	<0.22%

It can be seen from **Table 8** that although collisions caused by the classification error of the Naive Bayes Classifier are inevitable, a large number of obstacle avoidance experiments show that the probability of collision caused by the classification error is extremely low. Therefore, the method proposed in this paper is effective and feasible.

To avoid collides, the distance between the robot and the obstacle can be detected in the robot control node. If it is less than the safe turning radius, then no matter what output the Bayes Classifier issues, the robot must be commanded to reverse in order to ensure a safe turn.

4. Conclusions

To conclude, this paper focuses on the research on obstacle recognition and robot strategy. The main research methods are theoretical research and program design. It proposes the theory of LiDAR-based collection of data on obstacle boundary and Naive-Bayes-Classifier-based robot behavior decision-making. The validity of the theory is verified on the simulation platform, which shows that tasks of robot obstacle avoidance have been completed well. This paper mainly solves the previous methods' shortcomings of low accuracy and failure in obstacle avoidance in complex environment. It proposes a simpler method of obstacle boundary detection, which reduces the amount of calculation and increases the velocity of obstacle avoidance when the robot is performing real-time obstacle avoidance. To address the insufficiency of the dataset, it provides a method of expanding the datasets, which greatly improves the accuracy of the Bayes Classifier and enables the robot to make more successful decisions on obstacle avoidance. It improves on the traditional camera-based Bayes method of obstacle avoidance.

Although this paper only uses simulation experiment to verify the Naive-Bayes-Classifier-based method of mobile robot obstacle avoidance, the Gazebo is very similar to the real environment faced by the robots, since it has a large number of robot models (such as Turtlebot3) and scenarios (such as lights) as well as rich sensor resources (such as the Camera and the LiDAR). Besides, it is very simple and convenient to transplant the Gazebo-based simulation experiment method and code to the real robot. Therefore, the experimental method used in this paper is very relevant to practical applications.

The Naive-Bayes-Classifier-based method of mobile robot obstacle avoidance can be applied to the following scenarios: obstacle avoidance of domestic robots with low processor performance, obstacle avoidance of disaster relief robots that do not rely on maps, and the stress response function of most robots, etc. Therefore, this method can be widely used.

Conflict of interest

The authors declare no potential conflicts of interest.

Acknowledgement

We would like to express our sincere and deep gratitude to Dr. Jing Huang, for her inspiring instructions, earnest encouragement and great patience.

References

1. Chang J, Wu C, Li B. A survey of obstacle avoidance methods for mobile robots. Proceedings of the Chinese Eighth National Conference on information acquisition and processing 2010.
2. He Ming, Sun Jianjun, Cheng Ying. A review of the text classification based on the Naive Bayes[J]. Information Science 2016; 34(07): 147-154.
3. Li Y, Cai Z. Obstacle avoidance of mobile robot based on Bayes classifier. Control Engineering of China 2004; 20(4): 332-334, 359.
4. Chen Yi, Zhang Shuai, Wang Guiping. Vehicle detection algorithm based on the information fusion of the LiDAR and the camera[J]. Machinery & Electronics 2020; 38(01): 52-56.
5. Qizhen H, GuoLong Z. On image edge detection

method. International Conference on Electrical and Control Engineering, Yichang 2011; pp: 262-265.

6. Hong W, Bartels M. Unsupervised segmentation using gabor wavelets and statistical features in lidar data analysis. Amsterdam: North-Holland Pub 2006.
7. Lim D. Turtlebot3-Melodic. Received from http://wiki.ros.org/turtlebot3 2018.
8. Mohini Pandey． Different operator using in edge detection for image processing[J]． International Journal of Computer Science Engineering and Information Technology Research 2014; 4(1): 57-61.
9. YoonSeok P, HanCheol C, RyuWoon J, et al. ROS robot programming. Seoul: Robotis 2017.
10. Woodall W, Liebhardt M, Stonier D, et al. ROS topics: capabilities [ROS Topics]. IEEE Robotics & Automation Magazine 2014; 21(4): 14-15.