Image Processing in Long-distance Race Based on Humanoid Robot

Peng Gao¹, Jia Tian²*, and Jian Fang²

¹Student of Robot major, College of Electrical Engineering, Jilin Engineering Normal University, Changchun 130000, Jilin, China
²College of Electrical Engineering, Jilin Engineering Normal University, Changchun 130000, Jilin, China

*Corresponding author e-mail: 126425@jlenu.edu.cn

Abstract. This article mainly focuses on the path planning problems encountered by robots in long-distance races. In view of the uncertainty of the route of the long-distance running field, the image is processed in black and white according to the actual situation, the camera is reselected, and the program is rewritten. After the above steps, the problem of route recognition by the long-distance running robot is finally solved. And finally achieved good results.

Keywords: Nao Robot, Long-Distance Race, Path Planning

1. Problem Description
In 2019, we participated in the long-distance running robot racing competition organized by the school. During the competition, the color, material, illumination, reflectance and other details of the competition venue were uncertain, and the robot was required to have a certain adaptability. After the referee's whistle sounds, the robot can move. Note that during the movement, the robot must not step on the track with both feet at the same time; otherwise it will be considered a violation. During the movement, the robot cannot be controlled by remote control [1]. The robot must have legs and biped structure. The playing field is a carpeted floor, and the shape of the track is irregular. The detailed specifications are shown in the figure below:

![Figure 1. Robot track specifications](image_url)
2. Key Issues and Solutions
The difficulty of this competition lies in how the robots plan their own trajectories. On the day of the competition, we found that when we communicated with teams from other schools, there were roughly two options: one was to install robots. The external infrared sensor uses the infrared sensor to detect the path to plan a suitable path; the second is to use the robot's camera module to plan the robot's path through vision processing [2]. In this competition, we used the Nao robot, but the Nao robot does not have the conditions to install an external infrared sensor and the Nao robot itself has a camera for use, so we chose to call the camera of Nao and use the visual processing in python Analyze and process the photos taken by the camera to plan an appropriate path.

3. Black and White Processing of Pictures
The following is our program programming idea for this competition: use the camera module of the Nao robot to take pictures. After the picture is taken, the pictures are processed and the pictures taken are subjected to polarization processing. The color photos that will be taken are turned into black and white photos, the unprocessed pictures are shown in Figure 2, and the processed black and white pictures are shown in Figure 3.

![Figure 2. Unprocessed pictures](image1)

![Figure 3. The processed black and white picture](image2)

After the test is successful, we will also process the photos of the track on the green background and white lines into photos of the black background and white lines. The pictures of the track before and after processing are shown in Figure 4 and Figure 5:
After the black-and-white processing of the track is completed, it is the key part of this game, that is, the key part of the program. As we all know, the resolution of the picture can be roughly understood as we can use it after the completion of the shooting [3]. Lines are drawn on it. Both horizontal and vertical lines need to be drawn. The number of horizontal and vertical lines is determined by the resolution of the picture, as shown in Figure 6.
Figure 6. Schematic diagram of picture resolution display
And Nao robot also supports the selection of picture resolution, as shown in Figure 7.

Figure 7. Nao robot's display of picture resolution
In your normal thinking, you will definitely think that the higher the resolution, the better. Indeed, after the resolution is high, the picture will be clearer, but everyone must also know that the resolution is high and the picture is clear, then the corresponding robot the processing speed will be much slower. The scheduling of 5 gears will definitely make the processing speed of the robot very different, but the clarity is very important in the subsequent effects. Therefore, it is very important to choose the right pixel. Important [4].

Some people may be wondering why I have been talking about basic knowledge about pictures for a long time, and talking about a lot of messy things, but still did not get to the point, that is, how do we use the pictures taken by the robot, then the next Part is very important.

4. Pixel Processing Problem
We store the black and white photos that have undergone grayscale processing and binarization processing into a numpy array. We specify that all black parts are 1, and all white parts are 0. Then we filter the pixels, leaving only 0 pixels and deleting 1 pixel, then we delete the repeated 0 pixels, and arrange all the 0 pixels. The next point is the key idea of this program: we have already understood the problem of pixels, some people may be curious at that time, what exactly is the role of resolution here, we have explained to you that the resolution can be roughly It is understood that the number of lines drawn on the captured photos is determined by the resolution of the picture we choose, so now I can tell you that the robot judges what state it is in according to the corresponding "line" selected, and then
decides the trajectory of their next actions. After the above processing, the white runway has become a lot of 0 pixels. So how do we choose when facing so many pixels? At this time, the role of the resolution "line" is reflected. After the above processing, a corresponding number of 0 pixels appears on each of the horizontal and vertical "lines" [5-6]. Next, the Nao robot judges it. First of all, the Nao robot is already relatively clear after the above-mentioned pixel processing. Then, the Nao robot performs a secondary analysis on the pixels that have been processed above, that is, all 0s on this "line" Analyze the pixels and find the middle pixel. After finding the middle pixel, the Nao robot will judge it, that is, determine which direction the middle pixel is located in, and then make the corresponding left shift according to the judgment result. Decision to move right or go straight. Figure 8 is a pixel map after selecting the reference line.

The programming ideas for this program are roughly as many as the above. Although the programming ideas are very clear, we have also found many problems in actual use. The first is the problem of image processing itself, although the programming logic is logical It seems very clear, but after a careful taste, we can easily find that in the above programming ideas, we can't have white parts anywhere except the track. Once a robot appears, it will be treated as it should. Part of the track, but this part is not actually part of the track. At this time, the robot will be unable to deal with it, and the overall action will be problematic.

5. The Selection of the Camera is Well Known
In order to debug the robot, we will definitely adopt the control variable method, that is, control the robot's head to be fixed at a reliable angle. If the head is not controlled and locked, then the subsequent debugging process will fail [7]. This is what we don't want to see in a unified situation; secondly, in the process of image collection, due to the aforementioned problem of not being able to see white outside the track, we must adjust Nao’s head to a certain level The angle not only ensures that you do not see the parts other than the track under your feet, but also ensures that you can see the most suitable track area. As shown in Figure 9.
Next is the question of calling the camera. Nao has two cameras for us to call. Take a person as an example. These two cameras are located at the center of the eyebrows and the lips. Although in the strict sense, both cameras can be called, but since the camera itself has a certain wide-angle, just like the human eye, there must be a certain visual blind spot, so how to choose the camera is very important [8]. According to our final actual debugging, we found that the camera at the lips is in the actual shooting process the foot of the robot must be photographed in the middle. We have already said that once the robot photographs the white part except the track, the overall trajectory will be problematic. Unfortunately, the foot of the Nao robot is white, so, the camera on the lips cannot be called. But in the subsequent actual debugging process, we need to make a choice based on the actual debugging effect [9]. Although the camera on the lips will see the hands and feet, we can adjust the tilt angle of the robot's head to adjust it to Where appropriate, the camera at the center of the eyebrow can ensure that the hands and feet of the robot cannot be seen, but its viewing angle is larger, and it is possible to see farther scenes, which may cause the robot to have corresponding problems, as shown in Figures 10 and 11. It shows that the Nao robot is in the same position, and the two sensors take different photo effects. Among them, the camera on the lips captures the foot, and the camera at the center of the eyebrow captures a wider range.
Figure 11. The effect of calling the camera at the center of the brow
Secondly, the timing of taking pictures is also very important. We can imagine that the robot will sway accordingly during the movement, that is, the picture will be greatly tilted, then the judgment of the robot will also have corresponding problems. Figures 12 and 13 also show photos in a steady state and during exercise, respectively.

Figure 12. Photos taken by the robot in a stable standing situation
As shown in the figure, we can see the corresponding distortion in the pictures taken during the movement, which will definitely affect the robot's judgment of the action path. But if it is to pursue stability, let the robot stop, take pictures, judge, and move again every time it advances a corresponding distance, which will have a corresponding impact on the completion efficiency of the game. In summary, the timing of taking pictures is very important [10].

6. Robot Turning Problem
If the above-mentioned problem does not have such a serious impact on the robot, then the next problem is very important. We can imagine a problem, which is also turning. When we face the following two situations, is the situation the same?
As shown in the figure, the above three situations are situations that the Nao robot will definitely encounter during the movement. However, in these three situations, we can intuitively feel that the robot’s forward distance and turning angle must be different. It is easy to understand that if we did not classify and discuss the actual track shape and the position of Nao, then the robot will blindly run forward, there must be problems, so we need to capture the interception the central pixel is classified and judged, that is, it responds differently to the central pixel in different ranges. The distance of advancement and the angle of turning also need to be adjusted accordingly. Here is the uniqueness of python. If elif statement, can help us make multiple judgments on the way forward, and make different responses according to various actual situations, which greatly helps us make appropriate judgments on its progress. It is worth mentioning that, strictly speaking, we need to analyze and process each
horizontal pixel at a different vertical pixel position, but this is not only a waste of time, but it also seems that we are in the corner. Therefore, we chose to classify it with the range value to make the corresponding judgment.

7. Program Writing
First of all, it is the fixation of the head of the Nao robot, as shown in the 11th line of the code, 0.35 in the figure is the control of the tilt angle of the robot's head. We call the tilt angle and camera selection to achieve the choice of optimal control of the image.

![Figure 17. The program segment of robot head tilt angle control](image1.png)

Secondly, it is the selection and judgment of the central pixel. Figure 18 is the large-scale control area that we have not modified. From the above description, we can know that its accuracy is low and the effect is poor; the modified program is shown in the figure. As shown in 19, we have refined the range, distance, and angle to achieve better results.

![Figure 18. Large-scale area control program](image2.png)
Figure 19. Control program after scope refinement

The last is a description of the situation of taking pictures in sports. The above problems are indeed difficult to solve at first glance, but after in-depth thinking and practical experiments, we have also obtained a solution. First, we have modified the logic, that is, every time the robot takes pictures and moves at the same time, so what about taking pictures under shaking conditions? We chose to take a photo just before the robot has moved the corresponding distance, which neither affects the continuous advancement of the robot nor affects the shooting of the photo.

8. Conclusion
In addition to the speed requirements in the robot long-distance race, the recognition of the route is also important, because if you run the wrong route, no matter how fast the speed is, the effect of not reaching the end will occur. It needs to follow the prescribed route. The traditional detection method using sensors is inefficient, because more sensors are used, which affects the operating efficiency of the processor. With fewer sensors, the robot cannot walk in a straight line, so we finally decided to use the camera directly. For visual processing, the path is directly photographed and processed, so that the robot directly knows how to go, instead of taking one step. At the same time, the program is modified accordingly, which solves the problem of robot recognition of the route in the long-distance race. In the end, our robot won the second place in the long-distance race.

Acknowledgments
Source of the project1: 2019 Project of the “13th Five-Year Plan” for Education Science in Jilin Province
Item Number: GH19322
Source of the project2: School-level scientific research project of Jilin Normal University of Engineering and Technology
Item Number: XYBK202008
Team support: In-school innovation team of professional educational robots

References
[1] Tawseef Ahmed Teli, M. Arif Wani. A fuzzy based local minima avoidance path planning in autonomous robots [J]. International Journal of Information Technology, 2020 (prepublish).
[2] Robotics; Studies Conducted at Baylor University on Robotics Recently Reported (Robot Path
Planning Using Analog Circuit Phase Delay) [J]. Journal of Robotics & Machine Learning, 2020.

[3] Abhijeet Ravankar, Ankit A. Ravankar, Michiko Watanabe, Yohei Hoshino, Arpit Rawankar. Multi-robot path planning for smart access of distributed charging points in map [J]. Artificial Life and Robotics, 2020 (prepublish).

[4] Navid Toufan, Aliakbar Niknafs. Robot path planning based on laser range finder and novel objective functions in grey wolf optimizer [J]. SN Applied Sciences, 2020, 2(8).

[5] Ishita Mehta, Geetika Singh, Yogita Gigras, Anuradha Dhull, Priyanka Rastogi. Robotic Path Planning Using Flower Pollination Algorithm [J]. Recent Advances in Computer Science and Communications, 2020, 13(2).

[6] Hamed Fazlollahtabar. Comparative Simulation Study for Configuring Turning Point in Multiple Robot Path Planning: Robust Data Envelopment Analysis [J]. Robotica, 2020, 38(5).

[7] Robotics; Studies from Beihang University Update Current Data on Robotics (The Experience-memory Q-learning Algorithm for Robot Path Planning In Unknown Environment) [J]. Robotics & Machine Learning, 2020.

[8] Robots; Findings from Nanjing University of Posts and Telecommunications Broaden Understanding of Robotics (A New Global Best Guided Artificial Bee Colony Algorithm With Application In Robot Path Planning) [J]. Journal of Engineering, 2020.

[9] Fatin H. Ajeil, Ibraheem Kasim Ibraheem, Mouayad A. Sahib, Amjad J. Humaidi. Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm [J]. Applied Soft Computing Journal, 2020, 89.

[10] Petar Ćurković, Lovro Čehulić. Diversity Maintenance for Efficient Robot Path Planning [J]. Applied Sciences, 2020, 10(5).