

Research on White-Line-tracking Walking Technique of NAO Robot

Wanlong Peng^{1,2}, Yizhun Peng^{1,2*}, Yuheng Zhang^{1,2}, Tianye Jian^{1,2}

¹College of Electronic Information and Automation,
Tianjin University of Science and Technology, China

²Advanced Structural Integrity International Joint Research Centre,
Tianjin University of Science and Technology, China

E-mail: * pengyizhun@tust.edu.cn
www.tust.edu.cn

Abstract

Line-tracking walking technique is an essential part of intelligent robot technology. This paper selected NAO robot as the research platform to study the line-tracking walking¹ problem. Firstly, NAO white shell increased the difficulty of white line recognition, this is different from other robot. A method based on image preconditioning² is presented, sorting white edge with cut-point which is non-differential or derivative is zero, then designed the controller based on the improved digital incremental PID(proportion integration differentiation)algorithm and determined the controller parameters through the concise test method on Simulation. Finally, it was realized by programming in Python language.

Keywords: Humanoid robot, Edge classification, Image preconditioning, Simulation

1. Introduction

In recent years, the research and application of intelligent humanoid robots have gradually emerged. Line-tracking robot with certain autonomous decision-making and perception capabilities have quietly become an industry trend. Intelligent robot with line-following walking function can replace or help humans to complete some high-complexity, high-risk and high-workload inspections³ under complex conditions in some harsh environments, to find and eliminate equipment abnormalities in time. Minimize the loss of personnel and property. Therefore, the research on intelligent robots with line-following walking function has great significance in the industrial field. The NAO robot represent by the humanoid robot is a biped humanoid robot used for scientific research, competition and entertainment, with 25 degrees of freedom and 36 sensors. Possess certain environmental perception and the ability to make corresponding decisions. This paper uses the NAO robot as a platform to study the image perception processing of robot. The main work content of this research includes: Propose a color-segmentation method to increase the speed and reliability of the line recognition algorithm⁴;The robot's built-in camera collects

images every 30s, and converts the image information into reference input signals through technical means; Find the difference between the potential and the reference signal as the controller input; Write the program in python and carry out the line-tracking simulation experiment in the software.

2. White line processing

We takes the following steps to processing the white line:



Fig1. white line identification flowchart

2.1 Image preprocessing

Through the color segmentation of the image the points outside the white line will generally become the fulcrum, so that the image boundary is easy to find. The original image format provided by the NAO robot is YUV422, but the original image is generally used directly to avoid the overhead of format conversion.

The color table is a three-dimensional array, each dimension represents Y, U, and V

respectively. The array value stores the color type, such as 0 for unknown color, 1 for red, etc. Set the dimension to 64,64 and 64 respectively enough to meet the requirements. The color table needs to be established by manually collecting pixels. This is established offline during robot debugging. The established color table is used in the competition.

The image segmented by the color table generally has some color points that are different from the actual ones. For example, there are green points on the robot, and sometimes white or unknown colors may appear on the ground that should be green. This phenomenon is difficult to remove, but it can use some noise processing methods to filter out some salt and pepper noise.

3 White edge recognition

The traversal of white points is an edge growth process, and the recognition process is as follows: First, scan the image in order, find the first white edge point, and record all adjacent points of this point. If the image is scanned, turn to enter; Second, take one of the neighboring points as the starting point, start from this point, traverse along the edge, and record these edge points, turn to third, turn to traverse, if the neighboring points are traversed, return first to perform the next scan; Third, calculate the relative slope information of the current point and the previous recorded point every few points to determine whether it is a split point, if it is not a split point, continue to traverse, otherwise go to last; Last, when the split point appears, calculate the slope information of the point recorded in second, determine whether it is a long straight line, a short straight line or an arc, and record the result, and return to second; Next, The result recorded in the comprehensive analysis last Information, output results.

3.1 Scan and traverse edge points

Scan the image first to find the white edge points, and click the grid scan here. So as to ensure the effectiveness of scanning and speed up the algorithm at the same time. Scan the image from top to bottom in a column. If white dots and non-white dots appear alternately in the same row or in the same column of the 4 dots during scanning, edge dots appear, and then scan row by row or column by row to locate the edge point.

There is the problem of whether the path is unique during traversal. In addition to the

multiple paths that may appear in the outer scan for the first time, there are also two cases where there are multiple path options, as shown in Figure 5 and Figure 6, other points are normal, There is only one way to traverse.

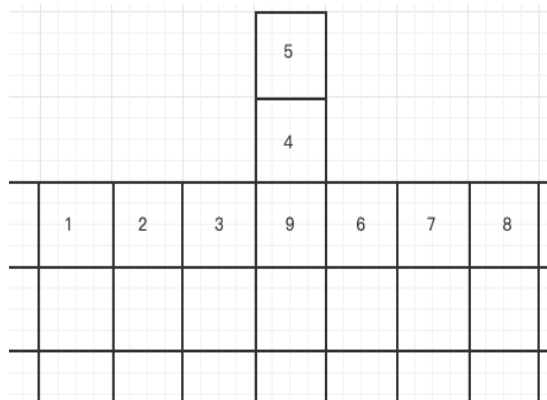


Fig.2 Convex edge

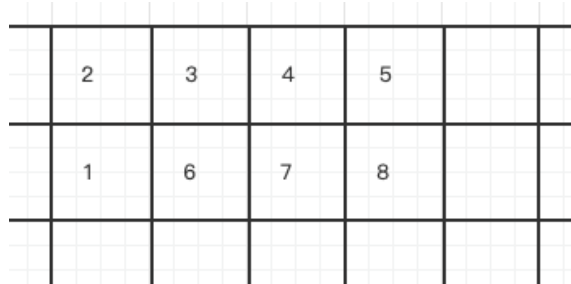


Fig.3 Edge with width 2

The numbers 1 to 8 in Figure 2 are all white edges. When traversing to 4, there are two paths of 5 and 6 in different directions. There are two ways to solve this convex situation: (1) Use the stack data structure to save Such different paths are then traversed by depth-first traversal; (2) Consider this situation during image preprocessing, and directly filter out the points 4 and 5 that cause this convex edge. Another situation where different paths appear is shown in Figure 3. After research, it is found that when the traversal priority is the same, the traversed paths will not intersect between the edges. This solves the ambiguity path problem that appears when traversing along the white edge.

3.2 Calculate the split point

As shown in Figure 4, there are several straight lines and curves on the plane, so how to classify these lines based on the situation of the NAO stadium? This article uses dividing points to divide them. Intuitively, the intersection of the straight lines should be divided Based on this idea, the dividing point in this paper is the

non-derivable point and the point where the derivative is 0.

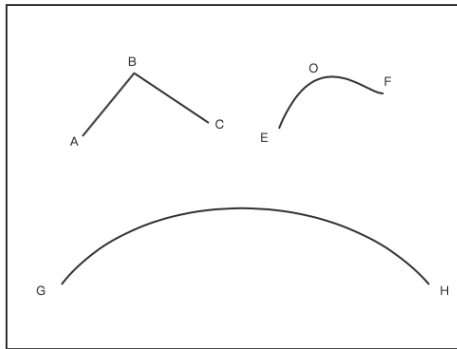


Fig.4 Line classification

There is a polygonal line ABC in 4, and its function is $f(x)$. When accessing from point A to point C along the polygonal line ABC, calculate the left and right limits of each point P on the path. If the left and right limits of P exist and are equal, the formula (1) is established, then continue to the next point. When visiting point B, it is found that the left limit of point B is equal to the slope of the straight line AB, and its right limit is equal to the slope of the straight line BC, that is, formula (1) is not true, then ABC must not be on a straight line, and point B is the dividing point. Record the path from A to B, then start from point B and continue to visit point C in the same way, thus dividing the polyline ABC into two segments AB and BC. When the straight line and the abscissa are perpendicular, the points on the straight line can be guided but the slope does not exist, and the included angle is 90 degrees. It is worth mentioning that in order to reduce the impact of large changes in the slope of adjacent edge points caused by unsmooth edges⁷, you can calculate the angle of the vector every N points along the edge, which may bring about the phenomenon of "turning back" when traversing the edge with a small width. In this way, pay attention to the direction change of the traverse point when using the angle instead of the slope.

$$\lim_{x \rightarrow p^+} f(x) = \lim_{x \rightarrow p^-} f(x) \quad (1)$$

The curve EOF is exactly the deformation of ABC, and these can be treated in the same way. The curve GH in Figure 4 is a circular arc with a large curvature. When traversing GH, there is a point on which the derivative is zero, but when calculating the slope, it is found that the slope changes very little and must be within the threshold range. There is no split point.

3.3 Result classification

When a segmentation point⁸ appears or the image boundary is reached during traversal, a series of points just traversed need to be classified. These points can be classified into long straight lines, short straight lines and arcs. First of all, the number of points can be used to judge whether the current points are short straight lines. The others are long straight lines or circular arcs. Some simple calculations can be used to judge whether they are long straight lines or circular arcs. Obviously, the slope of the start point and the end point of the long straight line should not change much, while the arc should change more. The method used in this experiment is to compare the slopes of the front N points and the back N points of a series of points to be classified to judge, that is, use the difference between slope1 of formula (2) and slope2 of formula (3) to measure, And then judge whether formula (4) is true or not, this method is simple to calculate without losing validity.

$$\text{slope1} = (\sum_{i=0}^N (P_i P_{i+1}) / N \quad (2)$$

$$\text{slope2} = (\sum_{i=0}^N (P_{n-i} - 1 P_{n-i}) / N \quad (3)$$

$$|\text{slope1} - \text{slope2}| = \text{threshold} \quad (4)$$



Fig.5 import result

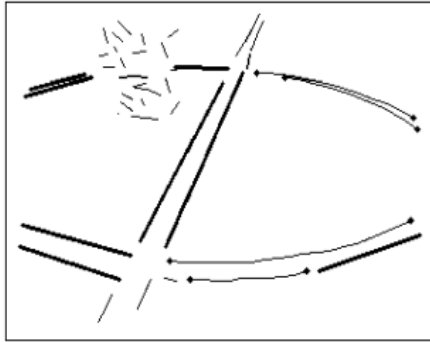


Fig. 6 Classification result

It can be seen from the classification results that the arcs on the court were originally classified as straight lines. It is necessary to consider the arcs and straight lines in combination with the actual situation of the court. In fact, many recognition algorithms directly transform the arc into a short straight line, and then connect, after comprehensive consideration. This can be achieved by the principle of Hough line transformation, but the element considered in Hough line transformation is a point. And here is a straight line. Also pay attention to the parallel features of the lines, and join together short straight lines, long straight lines, and arcs.

4. Experimental results and analysis

The experiment used is 3. The 3 version of the NAO robot, the main hardware parameters9: CPU:AMD GEODE x86 500MHz; Synchronous Dynamic Random Access Memory (SDRAM): 256MB, Flash: 2G; Camera: 2 COMS cameras, providing up to 30 frames/s, resolution of 640 × 480 images; Open CV library comes with the operating system. The criterion of experimental judgment is to identify the main straight lines and arcs. The results are shown in Table 1.

Table 1. Experimental data

distance light	Light 1	Light 2
Far away (more than 4m)	63%	20%
Medium (0.5m to 3m)	96%	71%
Near (less than 0.3m)	50%	50%

The color table of the experiment in Table 1 is configured under light 1, and the white line that is too far away cannot be recognized basically (when the distance is very far, the recognized court boundary can be used to give an estimated value), and it can be recognized when the distance is gradually reduced. Beyond the boundary, there is a good recognition rate around 2m. When the distance is too close, although the image is very clear, the robot’s field of view is very small, and it can be easily recognized when there is no interference. When NAO is used to eliminate interference, the robot basically blocks the white line, so the formal recognition rate is 50%. Using the color table configured under light 1 to split the color of the court under light 2, the experimental results are much worse than that under light 1, but the close distance is basically not affected. In fact, most of the recognition results are used by the upper layer in the robot positioning algorithm. Most of the positioning methods used are Monte Carlo filtering or generalized Kalman filtering, so that the program only needs to identify the image. The main lines and arcs. his method is used for the calculation of the two-dimensional array of the same price of the three-dimensional image after color segmentation. It also uses grid scanning. The main reason is that the recognition method is simple to calculate. The white line recognition program runs generally in 9ms to 12ms, which satisfies the robot competition. Requirements for straight line recognition. Compared with simple line recognition, the method in this paper pays more attention to the overall efficiency of robot image processing. The image preprocessing in it makes subsequent recognition algorithms more effective.

References

1. Thomas Rofer, Tim laue, Particle-filter-based Self-localization using Landmarks and Directed Lines Robot Soccer World Cup IX.2005,4020,pp.608-615.
2. Xiangkan, Sun, Detection and tracking of NAO based on machine learning[J]. Journal of Changchun university of Science and Technology(Natural Science Edition), 2016, 39(2), pp.116-119.
3. Rong-Jong Wai. Tracking control based on neural network strategy for robot manipulator[J]. Neurocomputing, 2002.
4. D.M. Dawson, Z. Qu, F.J. F. Dorsey. Robust control for the tracking of robot motion[J]. International Journal of Control, 1990.
5. Shen Zhang, Research on target recognition and localization of NAO robot [D]. Anhui University, 2013, pp.20-23.
6. Kun Zhao, Research on the visual navigation method of intelligent inspection robot for substation [D]. North China Electric Power University, 2014, pp.43-49.
7. Xuefeng Bai, Bin Yang, Based on NAO robot target recognition and localization algorithm [J]. Journal of Chengdu University of Information Technology, 2014, 29(6), pp.625-629.
8. Yanhong Wang, Research on moving target detection and tracking algorithm based on OpenCV [D]. Hangzhou: Hangzhou Dianzi University, 2013.
9. Xin Wang, Hongliang Yu, Research on diesel engine fault diagnosis based on the fusion of Bayesian and genetic neural network [J]. Ship engineering, 2012, 34(1), pp.32-35.