Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Practical Vision-Based Walking Navigation for the Humanoid Robot NAO in the Maze-like Environment

Li Wei, Zhiwei Wang, Wenchuan Jia*, Jianjun Yuan, Shugen Ma, Long Li
School of Mechatronic Engineering and Automation
Shanghai University
Shanghai, China
*Corresponding author: lovvchris@shu.edu.cn

***Abstract* - The motion features that accompanied by the alternate walking of the humanoid robot, which include body swing, foot slip, and fear of collision, are important factor limiting the indoor applications. In this manuscript, the humanoid robot NAO is taken as the platform, and the reliable pass through the maze-like indoor region is set as specific goal. Firstly, the specific intersection scene types are summarized, and the walking strategies for different intersection scenarios are discussed. Then, the general algorithm of path correction is developed. And the long straight walking, yaw correction, and L-shape intersection steering experiments are carried out. Furthermore, the corresponding optimization methods are discussed. The proposed strategy realizes a comprehensive technical framework for NAO to perform visual perception and stable walking in an unknown environment, which provide practical reference for efficient indoor navigation.**

***Index Terms* - Visual-Based Navigation, Mobile Robotics, Humanoid Robots**

I. Introduction

The maze puzzle is a classical and interesting problem. With the development of the robotics, walking through the maze of humanoid robots has attracted more attention [1]-[8], which also have become the robot competitions.

In the maze environment, the number of intersections is large and similar to each other. To realize the self-localization of humanoid robots, the method of pre-arranging specific road marks is used in many cases, which reduces the requirements for the hardware system of the robot. Common bar-codes are used as road marks by George et al. [1]. The position information returned by the bar-code provides the NAO robot with precise indoor navigation ability. In addition, the process of bar-code recognition is rapid, which improves navigation efficiency. The method of posting the corresponding road marks in different indoor scenes is used by Changyun Wei et al. [2]. In this method, for realizing indoor multi-scene navigation, image feature recognition method based on the sift algorithm is used to distinguish multiple road marks. The above methods rely on the high accuracy of the identification algorithm of the road marks. To optimize the common problems, Camshift, Kalman and PF algorithms are used by Chao Li et al. [3], and the reliability of identification is verified by the actual autonomous walking experiment. To distinguish the wall and the ground in a maze-like environment without using any road marks, the wall-following algorithm is adopted by Julio et al. [4]. However, from the image information returned by the monocular camera, the robot can't measure the distance from the wall. And if the NAO collides with the wall during walking, it would easily make the robot to fall down and lead to the navigation mission fail. To solve this problem and control the NAO robot to walk along the midline of the corridor safely, the visual servo navigation of the wheeled mobile robot is applied to the NAO robot by Faragasso et al. [5]. How to successfully make the NAO robot walk out of the unknown maze, under uncertain of the built-in walking and perception functions of NAO is studied in this paper. This case study can provide direct reference for similar applications of the NAO robots.

II. Visual NavigatIOn Method

A. *Maze-like Environment with NAO*

In this manuscript, the NAO robot with version number 3.2 is used as the humanoid robot platform in the experiment. The NAO robot is an economic biped robot developed by Aldebaran, and the robot has a height of 57.4 cm and a width of 27.5 cm, and has 25 degrees of freedom.

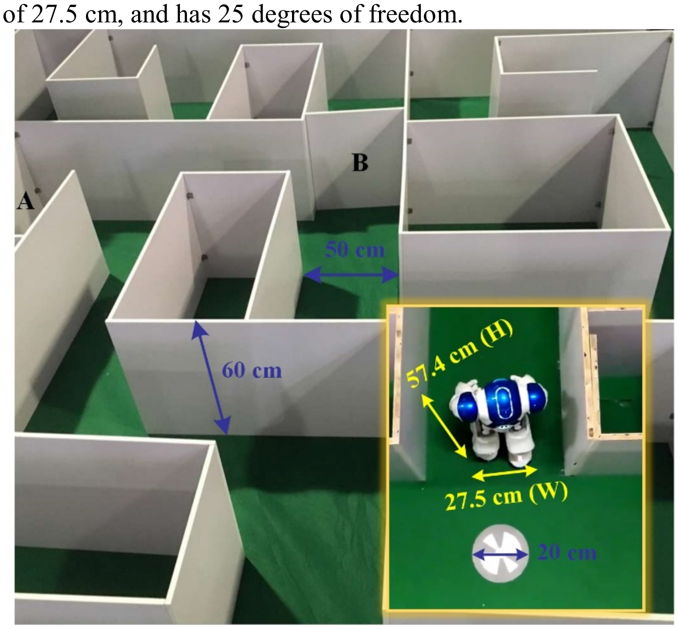


Fig. 1 The maze-like environment

As shown in Fig. 1, the maze environment consists of the ground and unmarked walls. The ground is covered with green short velvet carpet. Circular markers with a diameter of 20cm are placed at the T-shaped intersections. The color of the wall is white and the height of the wall is 60cm. The width of the corridor is 50cm, which is 1.8 times of the width of NAO.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

Multiple entrances and exits can be added to the maze. For example, ports A and B can be freely set as entrances or exits. In addition, there is no auxiliary equipment, such as the global camera at the top of the scene, in the maze. Environmental perception of the robot relies on the built-in camera and sonar sensor. With a clock speed of 499MHz and 128kB cache, the Geode integrated processor developed by AMD provides software interfaces of application development for NAO, which include Python and C++. All navigation programs are executed in real time by the integrated processor of the robot.

The monocular camera on the head of the NAO is used to capture images. Then image processing and image feature extraction are performed on the collected image information. By analyzing the image features, the relative position information of the robot in the maze is obtained and the intersection type in the maze is identified. Finally, according to the intersection type, the corresponding walking strategy is formulated to make the robot through the current intersection.

B. Visual Image Processing and Image Feature Extraction

From the collected image information, visual image processing and image feature extraction are used to extract the boundary lines from the wall and the ground and the circular marker. The boundary line between the wall on the left side of the NAO robot and the road is defined as the *left boundary line*. Similarly, the boundary line between the wall on the right side of the NAO robot and the road is defined as the *right boundary line*, the boundary line between the wall on the front side of the NAO robot and the road is defined as the *front boundary line*. To filter the lines that conform to the boundary line feature from the extracted lines, the slope range of the line is set. The marker contour is detected by the Hough Circles algorithm. The method of boundary line feature extraction is based on OpenCV library. The process is shown in Fig. 2. The Canny algorithm is used to detect the edge from image information to get the contours. Next, using the probability Hough Transform, the blue line segments are extracted from the contours. According to the slope and the y-axis intercept, the line segments are grouped. The line segments whose slope is greater than the set threshold are eliminated. The line segments set that conform to the boundary line feature is obtained. To find the maximum and minimum abscissa values, the abscissa values of the endpoints of the line segments in each set are gain and sort. Finally, the endpoints corresponding to the maximum and minimum values in each set are connected.

C. Yaw Correction

The NAO robot moves in the maze environment by biped walking. Due to friction and collision, the NAO robot tends to deviate from the desired trajectory when moving forward. Moreover, during walking, the swing of the body often makes the sensor unable to accurately perceive the environment. To avoid collision with the wall, it is necessary to realize the autonomous yaw correction during walking. The main process of yaw correction is as follows. Firstly, the position of the NAO robot relative to the wall is judged based on the extracted boundary line features. Then, if the current boundary line features only contain the *left boundary line*, the NAO robot rotates to the right at a small angle. While the current boundary line features only contain the *right boundary line*, the NAO robot rotates to the left at a small angle. If the current boundary line features also contain the *left boundary line* and the *right boundary line*, the NAO robot walks straight ahead to make the robot walk along the midline of the ground.

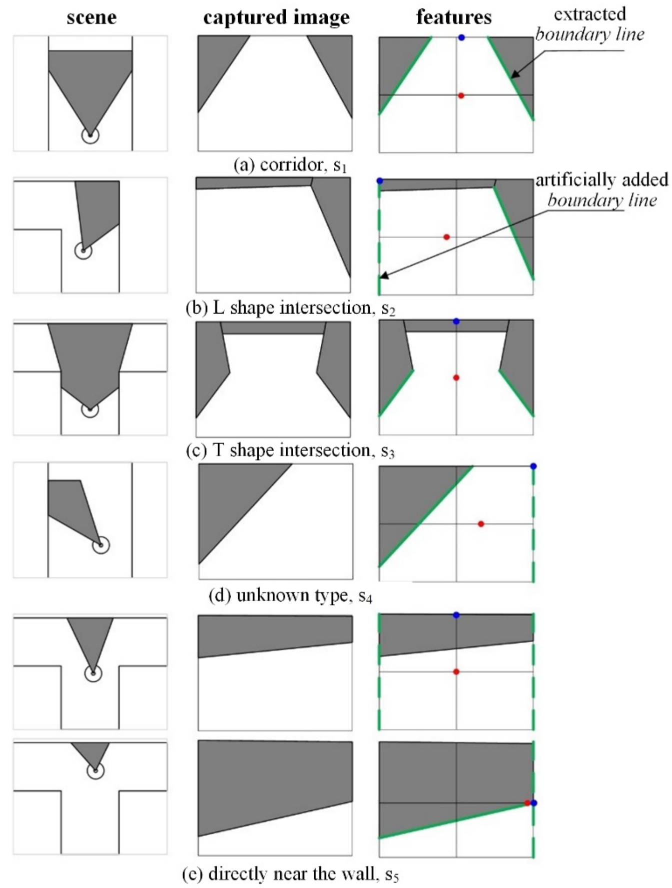


Fig. 2 Image processing and image feature extraction

D. L-shape Intersection

As shown in Fig. 2(b), the NAO robot walks to the corner. At this time, the extracted boundary line features are obtained. Under this circumstance, there will be the following problems when the robot turns at the intersection. When the robot walks to the intersection, the *left boundary line* will get shorter and shorter. Before the robot reaches the intersection, the *left boundary line* features can't be extracted by the image processing. This will trigger the robot to turn ahead, which increases the possibility of collision with the left wall. Therefore, the turning time can't be precisely controlled.

In addition, relative to the volume of the NAO robot, the space of the maze intersection is limited. Considering the turning radius, deviation and slip of the robot, the NAO robot may collide with the wall during turning.

To ensure that the robot keeps a safe distance from the wall when turning at the intersection, a sonar correction algorithm based on the sonar sensor of the robot is proposed in this paper, which controls the distance between the robot body and the front wall and the robot pose. Two sets of sonar sensors are mounted on the chest of the robot. The detection angle range is $120°$ fan-shaped area, the detection distance is 0.2m to 0.8m and the precision is 1cm to 4cm. The smaller the detection distance is, the higher the detection precision will be. According to the above hardware characteristics, the sonar sensors can meet the requirement of the robot to justify its pose during turning.

The sonar correction algorithm is that when the *front boundary line* is extracted from the image information, the sonar sensor will be activated and the distance from the front wall can be measured from the acquired sonar sensor information. If the values of the distance measured by two sets of sonar sensors are both less than the set safety threshold d_1, the robot continues to move forward. If the values of the distance measured by two sets of sonar sensors are greater than d_1, then the distance between the robot and the front wall should be adjusted to the set safety value d_2. When the distance between the robot and the front wall reaches d_2, the difference between the values of the distance measured by two sets of sonar sensors is calculated at this time. If the difference is less than the set threshold d_3, it indicates that the body of the robot is parallel to the front wall and the sonar correction is finished. While the difference is greater than d_3, the robot needs to turn left or right in a small angle until the difference is within d_3.

After the sonar correction is finished, the NAO robot will be in the center of the intersection. To collect the scene images on the left and right sides of the body, the robot turns the head 90° to the left and right. Using the mean hash algorithm, the hash values of two images are calculated and compared. If the hamming distance exceeds the set threshold, it indicates that the images contain a non-wall region. The gray values of the images need to be further calculated. The non-wall region background has many mottled colors. Thus, the gray value of the non-wall region is smaller than that of the solid color wall, which can be used to identify the non-wall side. To pass through the L-shaped intersection, the robot turns 90° to the non-wall side by adjusting its gaits.

E. Closed intersection

When the intersection before the robot is predicted as a closed intersection, the robot can directly turn away to reduce the adjustment of the pose and increase the walking efficiency.

After the robot recognizes the *front boundary line*, the sonar correction algorithm is used to adjust the relative distance and orientation between the robot and the front wall. After the adjustment, to collect the scene images on the left and right sides of the body, the robot turns the head 90° to the left and right. The mean hash algorithm is used to compare the similarity of two images. If the Hamming distance is less than the set threshold, it can be judged that the left and right sides of the robot are walls. Then the robot directly turns $180°$ to leave the closed intersection.

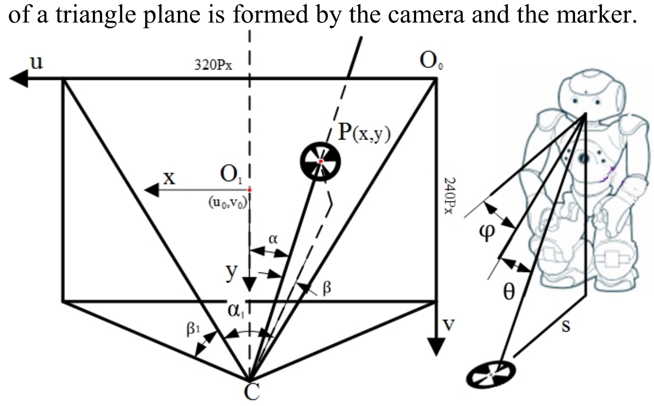
F. T-shape Intersection

In case that the NAO robot can't identify the T-shaped intersection, the circular markers are posted at the T-shaped intersection. After the robot recognizes the contour of the marker, the position information of the marker is used by the robot to adjust the body pose, so that the head and the body of the robot can point to the center of the marker. To control the robot to walk directly above the marker, the monocular ranging model is used to calculate the distance between the projection of the robot on the ground and the marker. Subsequently, the camera is used to capture images of scenes of the front, left and right sides of the robot respectively. The method of recognizing the wall at the L-shaped intersection is used to judge whether there is a wall in front of the robot. If there is a wall, the robot turns right. If there is no wall, the robot continues to move on. The principle of the above monocular ranging is as follows: firstly, the feedback position information of the marker is calculated. The position information is the horizontal angle α and the vertical angle β of the robot camera relative to the center of the marker. Using the provided application programming interface, the values of α and β can be directly obtained. They can also be calculated according to the coordinate of the marker centre. As shown in Fig. 3(a), using the Hough Circles algorithm, the centre coordinate $P(x, y)$ can be obtained. The point C represents the camera lens. The horizontal viewing angle α_1 and the vertical viewing angle β_1 of the lens are $60.97°$ and $47.64°$ respectively. The origin of the image coordinate system O_1 is at the center of the pixel coordinate system O_0. The coordinate of the center O_1 origin is (u_1, v_1) in O_0. The captured image size is 320px × 240px, then the coordinate of the O_1 origin is $(160,120)$. Using the triangle transformation, the values of α and β are calculated by

$$\alpha/\alpha_1 = x/320 - 0.5 \quad (1)$$

$$\beta/\beta_1 = y/240 - 0.5 \quad (2)$$

According to the values of α and β, the robot adjusts the pose of the head to make the camera optical axis pass through the marker center. Finally, as shown in Fig. 3(b), when the adjustment is finished, a monocular ranging model consisting of a triangle plane is formed by the camera and the marker.



(a) (b)

Fig. 3 The monocular ranging model

Using the programming interface, the distance H of the camera from the ground and the pitch angle φ of the robot head can be obtained. The angle θ between the camera and the horizontal direction is $39.7°$. Thus, the distance S between the projection of the robot and the marker is calculated by

$$S = \frac{H}{tan(\theta \times \pi \div 180° + \varphi)} \tag{3}$$

G. *The Maze Navigation Algorithm*

Considering all the above intersection types, the robot can be placed in any position of the maze. If the exit position is detected by the robot, the navigation will be finished. The maze navigation algorithm consisting of walking strategies at different intersections is shown in Fig. 4, which is used to guide the robot to walk safely in the maze.

According to the navigation strategy, the behavior of the robot and the corresponding state transition are further obtained, as shown in Fig. 5, and the navigation states are illustrated in Fig. 2.

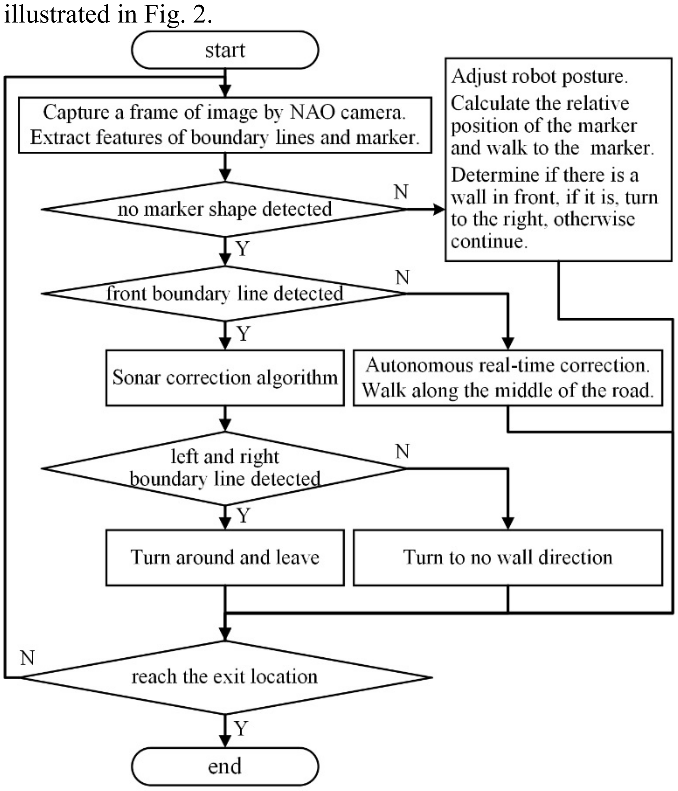


Fig. 4 The maze navigation algorithm

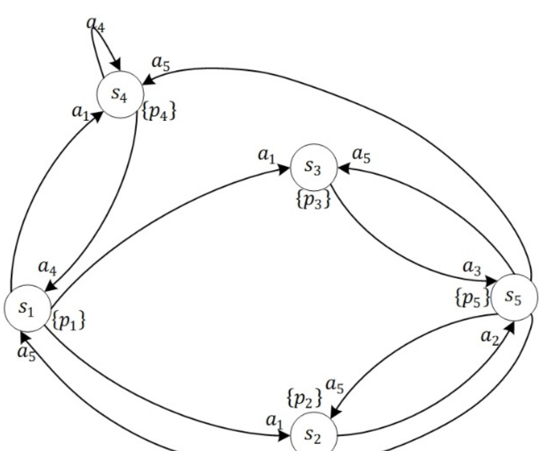


Fig. 5 State transition diagram

Specific behaviors include a_1, a_2, a_3, a_4, a_5. Behavior a_1 represents the strategy of straight corridor, in this situation, the robot keeps moving forward, reconstructing visual features using the identified left and right boundary lines.

Behavior a_2 represents the strategy of L shape. The robot adjusts the orientation to the left or right. The visual feature point is constructed by selecting an identified *left boundary line* or *right boundary line*, adding a vertical artificial boundary line in the image coordinate system, and discarding the identified *front boundary line* feature.

Behavior a_3 represents the strategy of T shape. The robot keeps moving forward or stops waiting for artificial motion commands. If keep moving forward, the visual feature are constructed using the identified *left* and *right boundary lines*, and the identified *front boundary line* features are discarded.

Behavior a_4 represents the strategy when no features are recognized. The robot adjusts the orientation to the left or right. The visual features are constructed by using an identified *left* or *right boundary line*, and a vertical artificial boundary line is added to the image coordinate system.

Behavior a_5 represents the strategy of wall avoidance. After the *front boundary line* is recognized, the robot advances, turns left, or turns right according to the slope of the *front boundary line*, and the intersection of the *front boundary line* and the vertical axis of the image coordinate system.

IV. Analysis Of Experimental Results

A. *Yaw Correction Experiment*

The camera beside the mouth position of NAO is used in the experiment. The initial yaw angle of the head is set to 0° and the pitch angle is set to 23°. The slope extraction range of the line segment in the *left boundary line* is set to -8.0 ~ -2.0. The slope extraction range of the line segment in the *right boundary line* is set to 1.5 ~ 5.5. The walking process of the robot in the corridor is shown in Fig. 6. The corridor length is 2.4m. As shown in Fig. 6(1) and Fig. 6(2), the robot immediately rotates in the opposite direction when the robot leans to one side of the wall. Fig. 6(3) and Fig. 6(4) show that the robot is walking along the midline of the corridor.

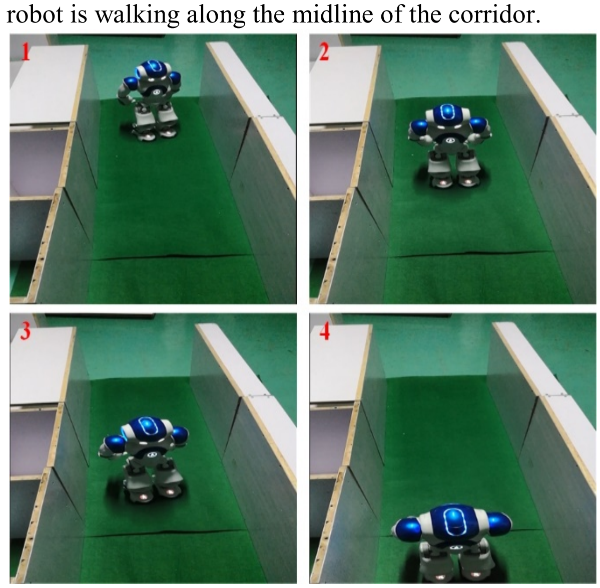


Fig.6 Yaw correction during walking in the corridor

B. *Wall Recognition Experiment*

The forehead camera of the NAO is used in the experiment, which takes images with and without the wall at the L-shaped, T-shaped and closed intersections. The images with the wall are solid color images, while the images without the wall have many mottled colors. Then the hash value and the gray value of the image are calculated. Two basic experiments are performed. One is the comparison of the parameter values of images with the wall. The other is the comparison of the parameter values of images with and without the wall. As shown in Fig. 7, multiple groups of results are obtained after several experiments. If the taken images both contain the wall, the value of hamming distance between two images is less than 13. In addition, the gray value of the image with the wall is greater than that of the image without the wall. The comparison based method can be used to recognize the wall at the intersection. As shown in Fig. 8, the robot turns to the non-wall side at the L-shaped intersection.

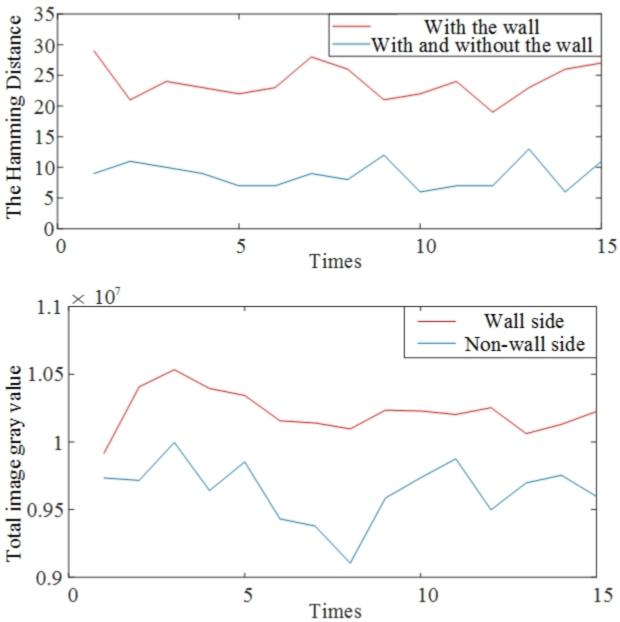


Fig. 7 Multiple groups of parameter values after several experiments

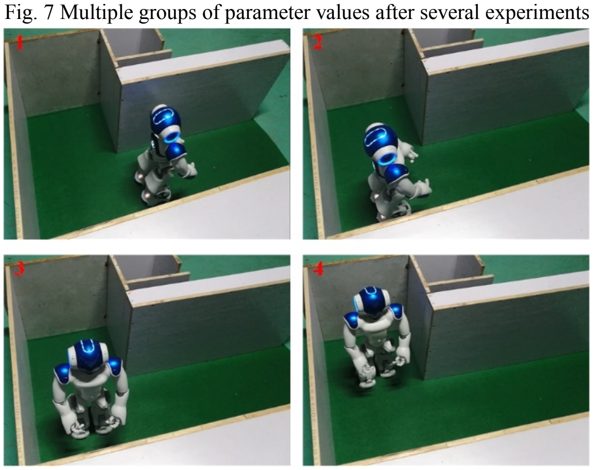


Fig. 8 The walking process of NAO at the L-shaped intersection

C. *Marker-based Monocular Ranging*

The camera beside the mouth of the NAO is used in the experiment, and the distance of that camera from the ground in standard pose is 33cm. The initial deflection angle of the robot head is set to 0° and the pitch angle is set to 23°. During the experiment, the pose of the robot is maintained all the time. The marker is placed in a position in front of the robot. The image coordinate system model and the built-in application programming interface are used to respectively calculate the relative distance between the marker and robot. The distance value measured and calculated by the two methods is recorded in Table 1. When the robot is far away from the marker, the distance value calculated by the image coordinate system model is closer to the actual value. While the distance value S that calculated by α and β measured by the application programming interface is more accurate, when the robot is close to the marker. The maximum error value between the calculated value of two methods and the actual value is 1.4cm, which meets the requirements of the monocular ranging.

TABLE 1
THE RESULTS OF THE MONOCULAR RANGING EXPERIMENT

The marker coordinate	The monocular ranging experiment results						
	The API method			The monocular ranging method			The actual value
	α	β	s	α	β	s	
(20,-5)	0.104	-0.032	0.203	0.098	0.037	0.217	0.206
(20,0)	-0.007	0.027	0.195	-0.015	-0.042	0.192	0.200
(20,5)	-0.103	0.035	0.208	-0.105	0.037	0.203	0.206
(25,-5)	0.090	0.100	0.245	0.098	-0.080	0.246	0.255
(25,0)	-0.019	-0.071	0.239	-0.002	-0.057	0.237	0.250
(25,5)	-0.131	-0.105	0.249	-0.123	-0.067	0.241	0.255
(30,-5)	0.099	-0.189	0.330	0.105	-0.150	0.336	0.341
(30,0)	-0.015	-0.206	0.299	-0.035	-0.112	0.289	0.300
(30,5)	-0.115	-0.146	0.327	-0.104	-0.157	0.335	0.341

D. *Maze Environment Navigation Experiment*

Considering all the above intersection types, the maze environment is constructed. The above maze navigation algorithm is used to make the robot walk from A to B. The actual walking process is shown in Fig. 9.

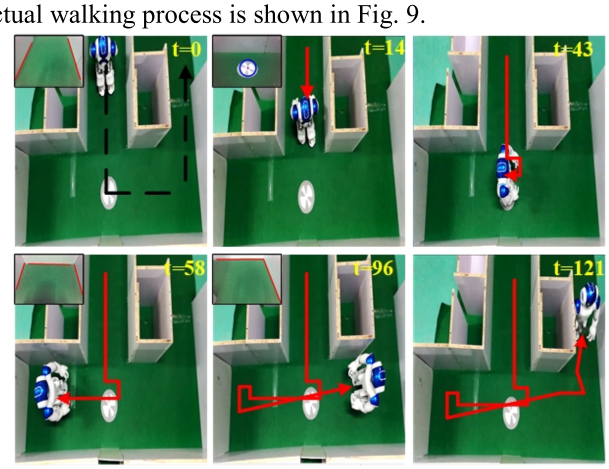


Fig. 9 The navigation experiment of NAO in the maze-like environment

The desired trajectory of the robot is marked by the black dotted lines, and the actual walking trajectory is marked by the red lines. The upper left corner of each part in Fig. 9 is the intersection features identified by real-time image processing. The whole walking process takes about 121s. During the whole walking, no human factors are used to adjust the robot motion. The robot always keeps a safe distance from the wall and never falls down for touching the wall. It shows that the navigation algorithm can be used to realize the autonomous and reliable navigation movement of the robot in the maze-like environment.

V. CONCLUSION

A practical method of the maze feature recognition and a visual navigation method in the maze environment are proposed in this paper. In this paper, the NAO robot is selected as the platform of the experiment. Using the visual, sonar and marker information, the position information of the robot relative to the maze can be obtained. Furthermore, according to the information, the type of intersection can be recognized. According to different intersection types, the corresponding walking strategies are formulated. When the robot deviates from the desired trajectory, the visual navigation method can be used to realize the yaw correction of the robot, which enables the robot to successfully pass through the intersection and realize the maze navigation. The validity of this method is verified by the practical maze navigation experiment.

The future work direction is as follows: firstly, when the robot walks continuously on the carpet, to prevent the robot from falling down, the gaits of the robot need to be improved for enhancing the robot walking ability. Secondly, the navigation strategy of the crossroad intersections needs to be added.

REFERENCES

[1] George Laurent, Mazel Alexandre, "Humanoid robot indoor navigation based on 2D bar codes: Application to the NAO robot," *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 329-335.

[2] Changyun Wei, Junchao Xu, Chang Wang, "An approach to navigation for the humanoid robot NAO in domestic environments," *Lecture Notes in Computer Science*, v 8069 LNAI, 2014, pp. 298-310.

[3] Chao Li, Xin Wang, "Visual localization and object tracking for the NAO robot in dynamic environment," *IEEE ICIA 2016*, pp. 1044-1049.

[4] Julio Delgado-Galvan, Alberto Navarro-Ramirez, "Vision-based humanoid robot navigation in a featureless environment," *Lecture Notes in Computer Science*, v 9116, 2015, pp. 169-178.

[5] Angela Faragasso, Giuseppe Oriolo, "Vision-based corridor navigation for humanoid robots," *IEEE International Conference on Robotics and Automation*, 2013, pp. 3190-3195.

[6] Padhy, Ram Prasad, Feng Xia, Suman Kumar Choudhury, Pankaj Kumar Sa, Sambit Bakshi, "Monocular vision aided autonomous UAV navigation in indoor corridor environments," *IEEE Transactions on Sustainable Computing*, 2019, Vol.4, No.1, pp. 96-108.

[7] Ido, Junichi, Yoshinao Shimizu, Yoshio Matsumoto, Tsukasa Ogasawara, "Indoor navigation for a humanoid robot using a view sequence," *International Journal of Robotics Research*, 2009, vol. 28, no.2, pp. 315-25.

[8] Pasteau, Francois, Alexandre Krupa, Marie Babel, "Vision-based assistance for wheelchair navigation along corridors," *IEEE International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp.4430-4435.