The Knowledge Engineering Review, Vol. 32, e9, 1–11.
doi:10.1017/S0269888916000084

Vision-based obstacle avoidance system with fuzzy logic for humanoid robots

SHU-YIN CHIANG

Department of Information and Telecommunications Engineering, Ming Chuan University, 5 De Ming Road, Gui Shan Distract, Taoyuan City 333, Taiwan;
e-mail: sychiang@mail.mcu.edu.tw

Abstract

This study presents the algorithm for a humanoid robot to accomplish an obstacle run in the FIRA HuroCup competition. It includes the integration of image processing and robot motion. DARwIn-OP (Dynamic Anthropomorphic Robot with Intelligence–Open Platform) was used as the humanoid robot, and it is equipped with a webcam as a vision system to obtain an image of what is in front of the robot. Image processing skills such as erosion, dilation, and eight-connected component labeling are applied to reduce image noise. Moreover, we use navigation grids with filters to avoid the obstacles. Fuzzy logic rules are used to implement the robot's motion, allowing a humanoid robot to access any routes using obstacle avoidance to perform the tasks in the obstacle-run event.

1 Introduction

Research on mobile robots is one of the most popular topics in robotics and intelligent systems. No matter whether the robot is wheeled or a humanoid, it can autonomously navigate in an environment depending on algorithms for object detection and avoidance. Primarily, the movement of the humanoid robot is affected by obstacles that could make it fall and lose its balance. Obstacle avoidance is an important research topic in robotics, but designing an algorithm to let a robot to navigate autonomously in an environment and avoid obstacles is a complicated problem. Many researchers use infrared sensors (Benet *et al*., 2002; Li *et al*., 2004), ultrasonic sensors (Li & Yang, 2002), or lasers (Hancock *et al*., 1998; Soumare *et al*., 2002) to detect barriers. However, these methods have some defects: the search angle and search distance are limited. Moreover, if only a vision sensor is allowed, it is not easy for a robot to avoid obstacles and navigate in an unknown environment due to the planar image.

Some studies (Chao *et al*., 2009; Chen *et al*., 2014) have used wheeled mobile robots with a vision system to navigate an environment without colliding with obstacles. They use either stereo vision (Chao *et al*., 2009) or a Kinect sensor (Chen *et al*., 2014) to obtain the distance from the obstacle to the robot to perform the obstacle avoidance task. As the stability of moving wheeled mobile robots is higher than that of humanoid robots, the methods for wheeled mobile robots may not be applicable. Wong *et al*. (2011) proposed an algorithm for a humanoid robot turning left or right to avoid obstacles, but it seems too complicated for simple humanoid robots to record how many times they have changed directions and turn back to original direction to keep walking toward the destination. In addition, the algorithms (Wong *et al*., 2011) calculating the angles of the obstacle from the robot may not be precise because the vision system is easily affected by the environment. For example, when the robot is walking and its visual angle may vary by head tilt and head moving. Even 5–10° difference obtained in the algorithm may cause errors ranging from 30 to 50 cm.

Computer vision is one of the ways of perceiving the environment, but webcam can only provide a planar image instead of a three-dimensional image with range distances. To obtain the depth of the

obstacle, Budiharto *et al*. (2013) used stereo vision in a humanoid robot to avoid multiple moving obstacles. However, for humanoid robot competitions such as the FIRA HuroCup competition, obstacle avoidance is one of the events, called the obstacle run in the contest. In this competition, humanoid robots that can walk through an environment with many color obstacles such as file holders, paper board, and boundary lines need a real-time obstacle avoidance scheme to perform the task only depending on a vision system. Tradition methods for obstacle avoidance such as ultrasonic sensors and lasers cannot detect the color obstacle mentioned right above. However, in the current robotics market, a commercially manufactured humanoid robot cannot obtain stereo vision or depth information of an obstacle.

Therefore, an algorithm simply based on a planar vision system is proposed to accomplish the obstacle avoidance task and navigate through the environment autonomously. The rest of the paper is structured as follows. Section 2 describes the problem of obstacles and introduces the humanoid robot implemented in this study. An algorithm with fuzzy logic is proposed in Section 3. The experimental results are presented in Section 4 and the paper is concluded in Section 5.

2 Humanoid robot and the problem description

2.1 Humanoid robot

Figure 1 shows the humanoid robot DARwIn-OP, which is an abbreviation for Dynamic Anthropomorphic Robot with Intelligence–Open Platform, and its overall specifications are listed in Table 1. DARwIn-OP was developed by ROBOTIS and comprises 20 Dynamixel MX-28 actuators (Robotis, Korea). The actuators offer 6 d.f. for the legs as well as dual 3 d.f. arm movement and dual 2 d.f. neck movement. The robot is equipped with a USB 2.0 HD Logitech C905 camera (Logitech, Swiss). The camera is the only sensor equipped in the humanoid robot to avoid obstacles in the environment.

2.2 Problem description

In this study, the robot moves from one end of the playing field to the other as quickly as possible following the rules of FIRA's obstacle-run event in the HuroCup. The dimensions of the playing field are at least 300×400 cm. The challenge of the event is that wall obstacles in blue and ground obstacle in yellow cannot be overcome, as shown in Figure 2. Meanwhile, two boundary lines marked in yellow are also avoided by the robot.

3 The proposed algorithm

3.1 Image processing

The camera is the only sensor for the robot and, according to the rules of the obstacle-run event, the color of the obstacles is blue. Hence, the image size of the environment is captured at 320 pixels wide and

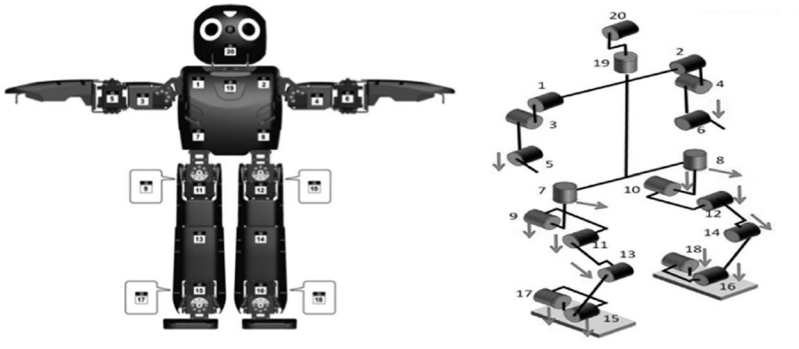

Figure 1 DARwIn-OP (Dynamic Anthropomorphic Robot with Intelligence–Open Platform) servo motor numbering and positions

Table 1 Specification of DARwIn-Op (Dynamic Anthropomorphic Robot with Intelligence–Open Platform)

Categories	Description	Data
Dimension	Height	0.455 m
	Weight	2.8 kg
d.f.	Head	2 d.f.
	Arm	2 × 3 d.f.
	Leg	2 × 6 d.f.
Main controller	CPU	Intel Atom Z530 @ 1.6 GHz
	RAM	DDR2 1 GB
	Disk	Flash Disk 4 GB
	Network	Ethernet/Wi-Fi
	USB port	2 × USB 2.0
Sub controller	CPU	ARM 32-bit Cortex-M3
	Frequency	72 MHz
	Flash memory	512 kB
	SRAM	64 kB
Actuator MX-28	Holding torque	31.6 kgf cm^{-1} @ 12 V
	Speed	67 r.p.m. @ no load
	Position sensor	Magnetic encoder
	Resolution	0.088°
	Command interface	TTL
Sensor	Gyroscope	3-Axis
	Accelerometer	3-Axis
	Pressure-meter	2 × 4 FSR
	Camera	2 MP HD USB
Software	O/S	Linux Ubuntu
	Framework	Open-DARwIn SDK
	Language	C++ /Java
	Compiler	GCC

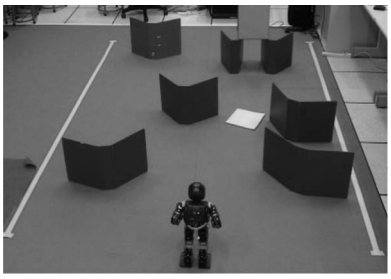

Figure 2 Obstacles and robot

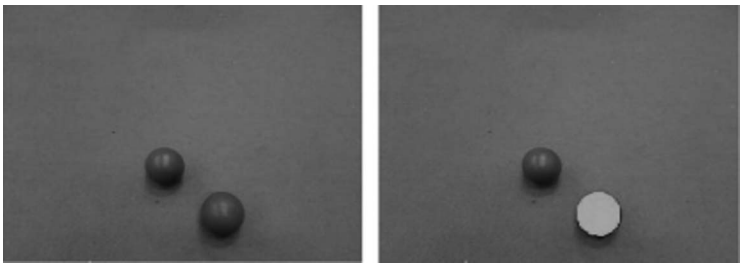

Figure 3 Connected components labeling operation

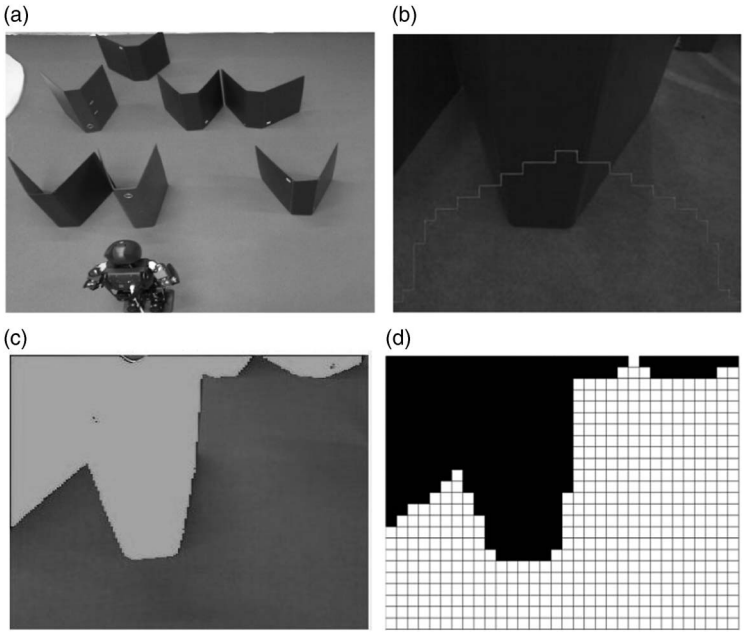

Figure 4 (a) Obstacles and robot from top view. (b) Obstacles from robot's view. (c) Color-coded obstacle from the robot's view. (d) Navigation grid

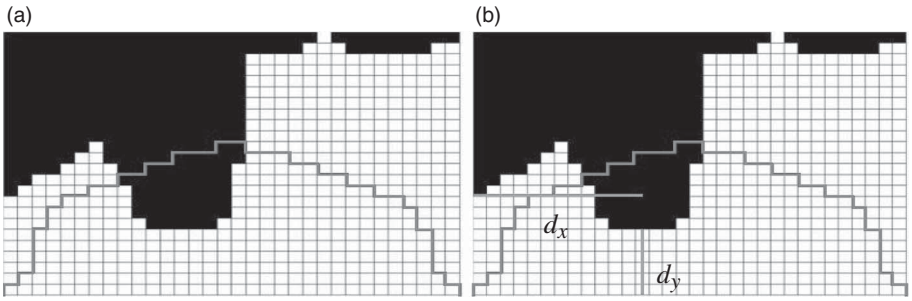

Figure 5 Focus area (a) and the distances (b) of the obstacle d_y and d_x

240 pixels high (320 columns × 240 rows). The image is transformed into a binary image. Then, the connected components labeling operation (He *et al.*, 2009) is used to distinguish the obstacles by assigning a unique label to each maximal connected region of foreground pixels. We use 8-connect components labeling so that a pixel has eight neighbors around it to mark the label. For example, two blue balls are shown in Figure 3. Although they have the same color, the connected components labeling operation marks the closer one in orange. Therefore, the connected components labeling process can be used to distinguish the obstacles of the blue walls.

3.2 Decision making for moving direction

The color-coded model is first built to discover the obstacles. The image employs the connected components labeling method to remove the noise from the image, as shown in Figure 4. Next, the image is transformed into a binary image and image preprocessing techniques such as dilation and elution are used, and the image is reduced into a 32 × 24 grid (Hsia *et al.*, 2012), each grid being a 10 × 10 pixel square. To reduce the noise effect, the grid is marked in black if the pixels of the obstacle are >20 pixels in 10 × 10 (100) pixels. Figure 4 shows the obstacles and robot from top view, obstacles from robot's view, color-coded obstacle from the robot's view, and navigation grids in Figures 4(a)–4(d), respectively.

According to the navigation grid, the depth vector D is a 1×32 dimension vector and is obtained from the first obstacle's location from the bottom row location for each column. Therefore, we have

$$D = [d_1, d_2, \cdots, d_{32}] \tag{1}$$

where d_i is the distance of the first black grid location for the ith column.

To prevent the robot from hitting the obstacle, the focus area that is in front of the robot is defined as shown in Figure 5:

$$F = [f_1, f_2, \cdots, f_{32}] \tag{2}$$

where

$$f_i = \begin{cases} i, & 1 \leqslant i \leqslant 16 \\ 33-i, & 16 < i \leqslant 32 \end{cases} \tag{3}$$

To check whether the obstacles are inside the focus area, define vector V in (4). If V is not equal to 0, then the robot needs to move because it indicates that the obstacle is inside the focus area:

$$V = [v_1, v_2, \cdots, v_{32}] \tag{4}$$

$$v_i = \begin{cases} f_i - d_i, & f_i \geqslant d_i \\ 0, & f_i < d_i \end{cases}, \; i = 1, \; \ldots \; , 32 \tag{5}$$

Define the weighting factor in (6) and (7) to decide the direction in which the robot needs to move. If W_L is greater than W_R, the robot will move left and set the boundary point x_b to be the furthest right point, 319. Otherwise, if W_R is not greater than W_L, the robot will move right and set the boundary point x_b to be the furthest left point, 0. The boundary point x_b is defined in (8):

$$W_R = \sum_{i=1}^{32} (33-i) \times v_i \tag{6}$$

$$W_L = \sum_{i=1}^{32} i \times v_i \tag{7}$$

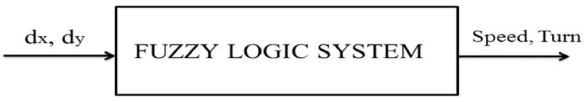

Figure 6 Fuzzy system block diagram

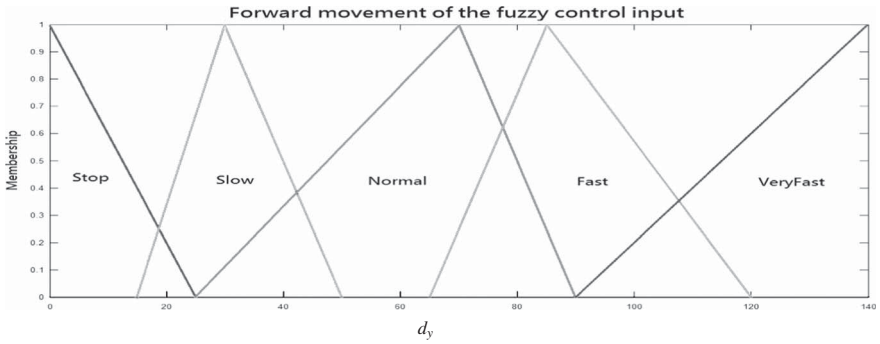

Figure 7 Membership functions for input d_y

$$x_b = \begin{cases} 0, & W_L \leqslant W_R \\ 319, & W_L > W_R \end{cases} \tag{8}$$

3.3 Obstacle avoidance by fuzzy logic control

To obtain the minimal distance from robot to obstacle, we have the boundary distance from the y direction d_y, defined in (9) as the shortest distance from the depth vector D. The distance from the center point of x_c defined in (10) to the boundary point x_b from the x direction d_x, shown in (11). The d_y and d_x shown in Figure 5 are the shortest distance of obstacles in a vertical direction and the distance from the

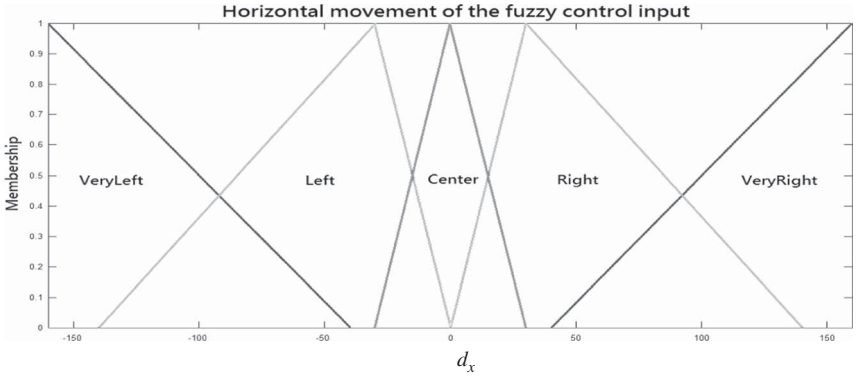

Figure 8 Membership functions for input d_x

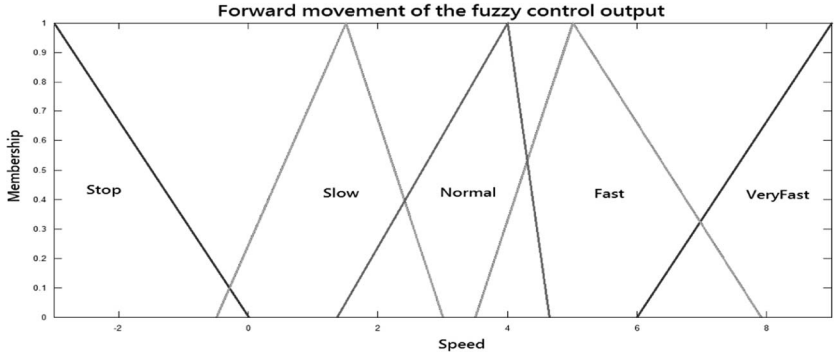

Figure 9 Membership functions for output *speed*

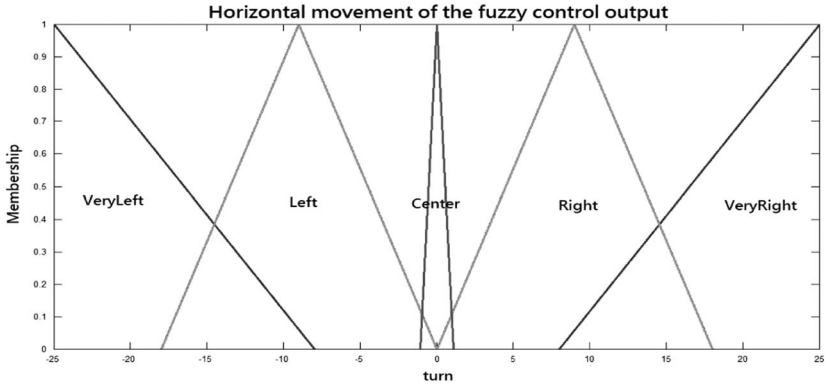

Figure 10 Membership functions for output *turn*

center of the obstacle to the boundary, respectively. These two parameters are used in the fuzzy logic control system to decide the movement of the robot to avoid the obstacles:

$$d_y = \min\{d_i\} \tag{9}$$

$$x_c = \left(\sum_{i=1}^{n} x_i\right) / n, \quad (x_i, y_i) \in A \tag{10}$$

where A is a region with obstacles inside the focus area, and n grids are assumed in the focus area.

$$d_x = x_c - x_b \tag{11}$$

The proposed scheme involves a fuzzy system that approximates any nonlinear function to arbitrary accuracy with only a small number of fuzzy rules (Mendel, 1995). According to the fuzzy system, it has three steps: fuzzification, fuzzy rule, and defuzzification. The fuzzy system diagram is illustrated in Figure 6.

Fuzzification. The first step in fuzzification is to determine inputs and outputs, and choose an appropriate membership function for input and output that, for simplicity in implementation, selects a triangular membership function in this study. First, let us define the membership function for the inputs d_y and d_x, as

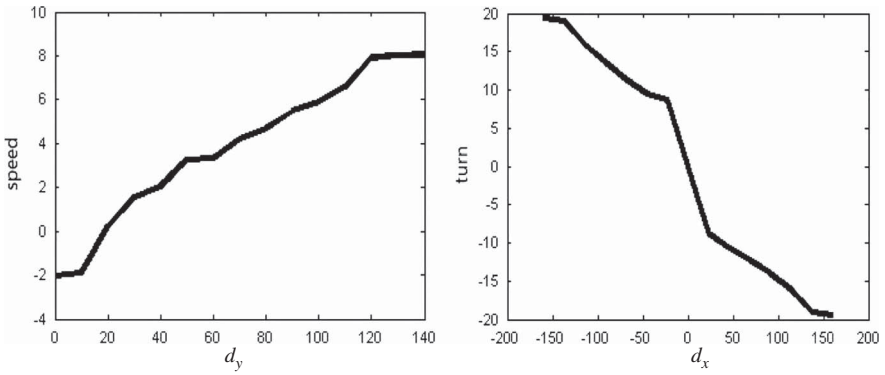

Figure 11 Defuzzification of the output *speed* and *turn*

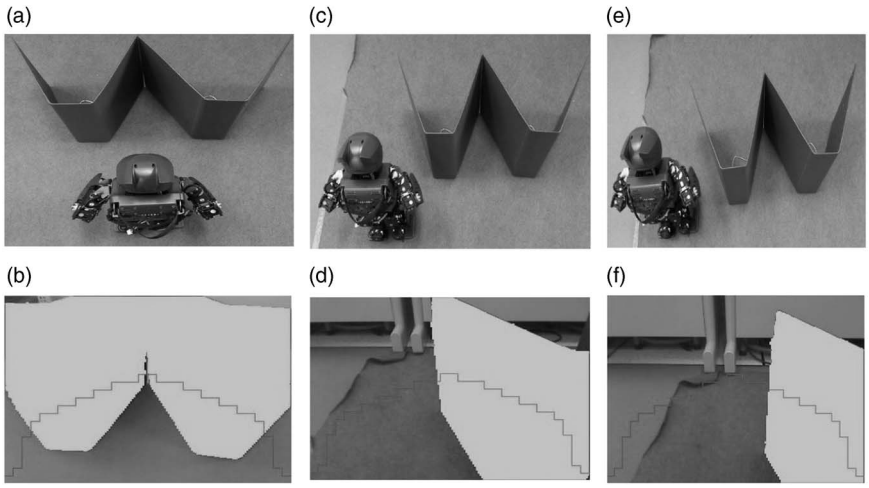

Figure 12 Experiment for testing the proposed scheme to avoid obstacles. The robot moves from (a), (c) to (e) and the corresponding vision is (b), (d) and (f), respectively

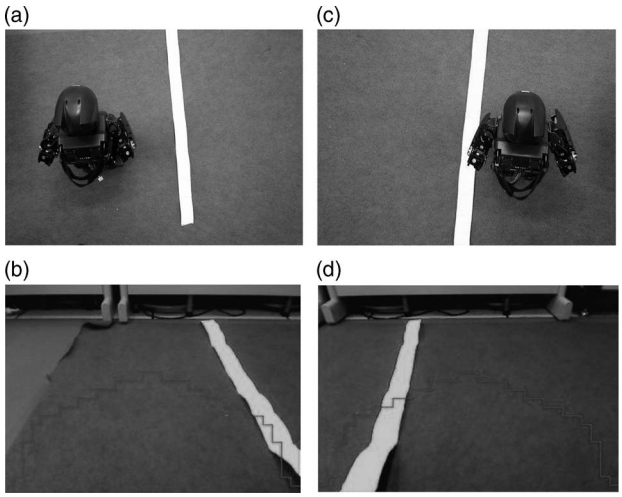

Figure 13 Experiment for testing the proposed scheme for boundary lines. The robot avoid both boundary line in (a) and (c) and the corresponding vision is (b) and (d)

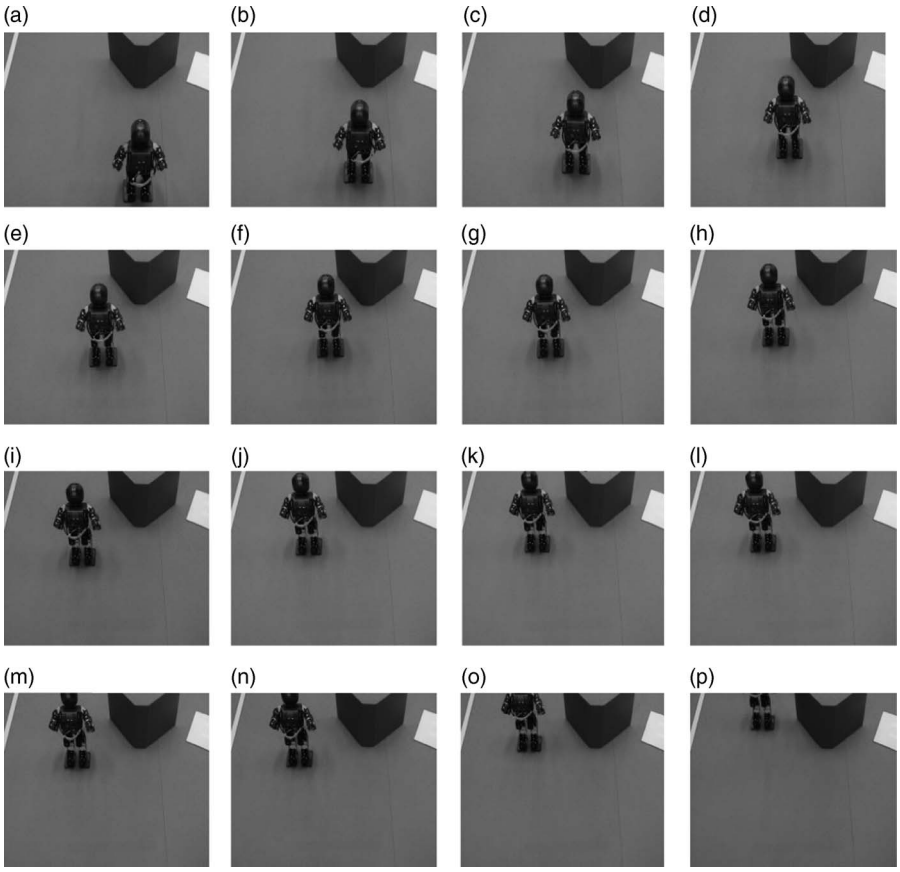

Figure 14 Continuous movement for the proposed scheme to avoid the obstacle. The robot moves from (a) to (p) in alphabetical order to avoid the obstacle

shown in Figures 7 and 8, respectively. Here, d_y is the forward movement and d_x the horizontal movement of the fuzzy control input.

Then, we define the membership function for the output *speed* and *turn*, as shown in Figures 9 and 10, respectively. Here, *speed* is the forward movement speed and *turn* the horizontal movement of the fuzzy control output.

Fuzzy rule. If… then… is used to implement the fuzzy system. The rules are as follows:

Rule 1:	If d_y is Stop,	then *speed* is Stop.
Rule 2:	If d_y is Slow,	then *speed* is Slow.
Rule 3:	If d_y is Normal,	then *speed* is Normal.
Rule 4:	If d_y is Fast,	then *speed* is Fast.
Rule 5:	If d_y is Very Fast,	then *speed* is Very Fast.
Rule 6:	If d_x is Very Left,	then *turn* is Very Left.
Rule 7:	If d_x is Left,	then *turn* is Left.
Rule 8:	If d_x is Center,	then *turn* is Center.
Rule 9:	If d_x is Right,	then *turn* is Right.
Rule 10:	If d_x is Very Right,	then *turn* is Very Right.

Defuzzification. This is used to transform a fuzzy set to a crisp set. Therefore, the input for defuzzification is the aggregate output, and the output of it is a crisp number. Based on fuzzy methodology, the center of gravity method is used in this study. Then, we have the nonlinear relationship between input and output for both pairs of (d_y, *speed*) and (d_x, *turn*), as shown in Figure 11.

4 Performance analysis

To illustrate the performance of the proposed algorithm, the experimental environment is established, as shown in Figure 12. First, we evaluate the situation where an obstacle is in front of the robot, as shown in Figure 12(a), and the robot's corresponding vision is presented in Figure 12(b). Then, the robot moves

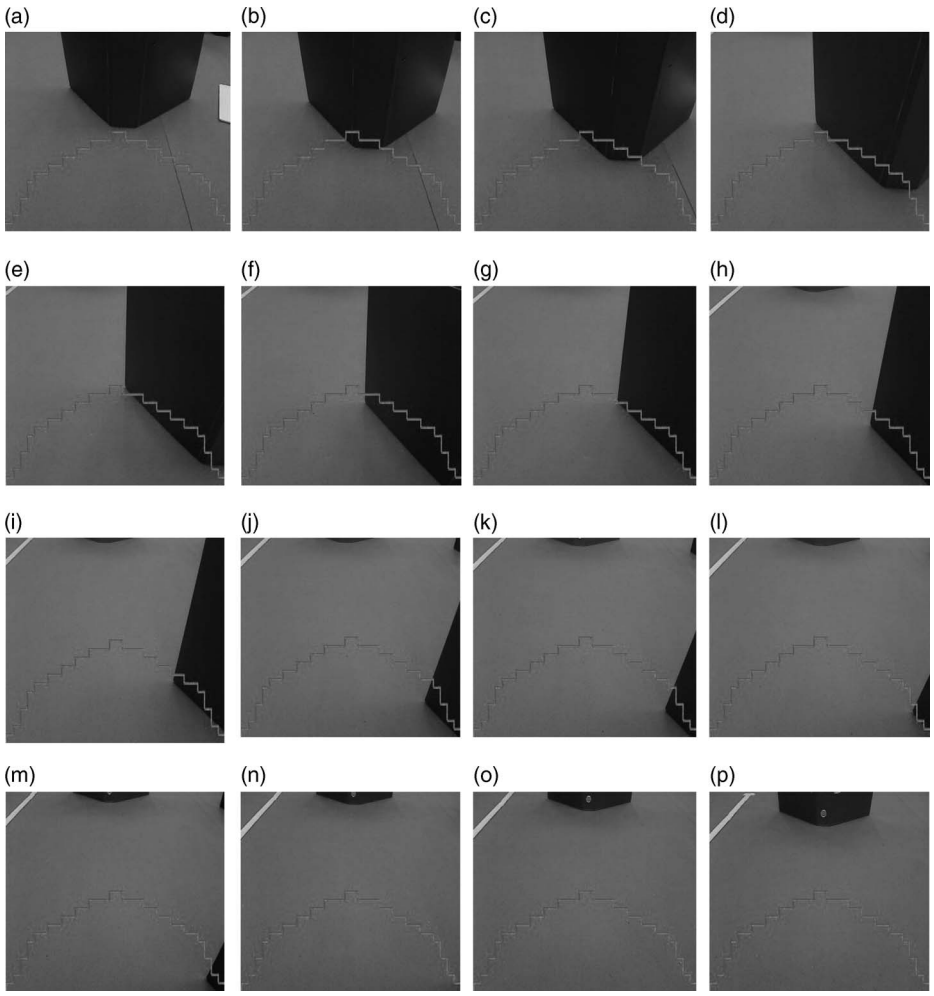

Figure 15 Corresponding vision from (a) to (p) in alphabetical order from the robot for the experiment in Figure 14 (a) to (p), respectively

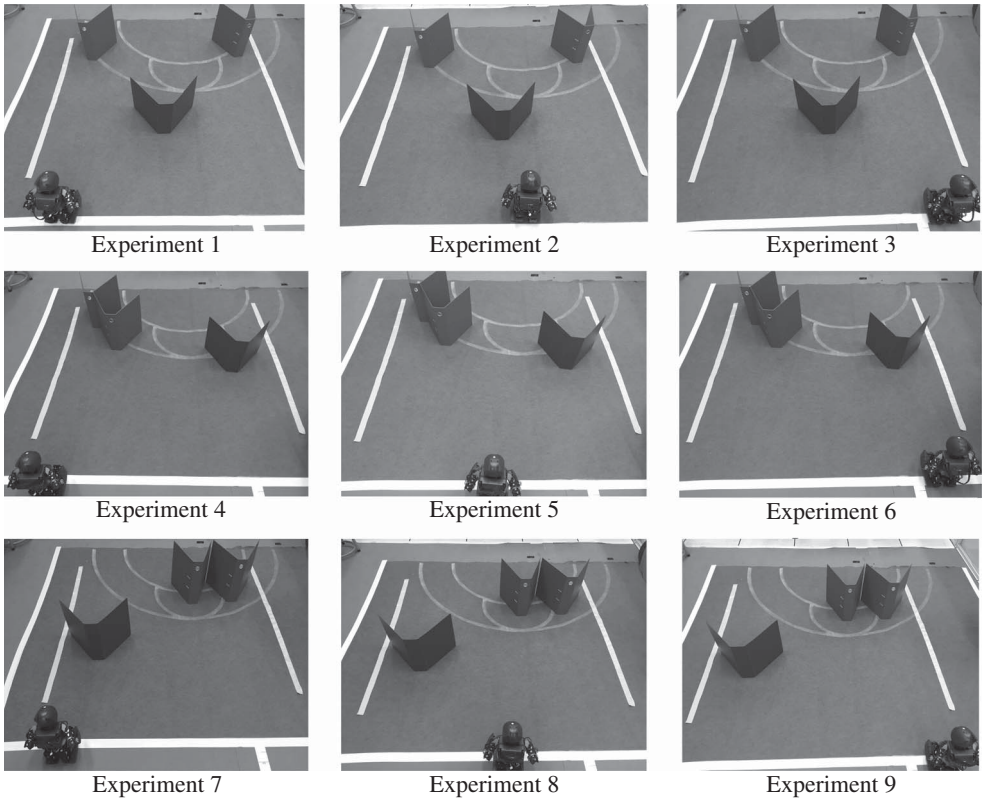

Figure 16 Nine test experiments for the robot

Table 2 Results of test experiments

Experiment no.	Test times	Successful pass times
1	3	3
2	3	3
3	3	3
4	3	3
5	3	3
6	3	3
7	3	2
8	3	3
9	3	2
Average successful rate		92.6%

left as per the proposed algorithm, as shown in Figure 12(c), and its vision is shown in Figure 12(d). The robot finds that the obstacle is still in the focus area; therefore, it follows the proposed scheme to move left more, as shown in Figure 12(e), and the vision for the robot is shown in Figure 12(f).

Next, we test the boundary lines to check whether the robot will avoid hitting the boundary lines. The experiment is illustrated in Figures 13(a)–13(d). The robot follows the algorithm to find both sides of the boundary lines and moves left or right to avoid stepping on the boundary lines in the event.

Finally, the continuous movement of the robot is evaluated in the experiment. According to the proposed algorithm, the robot obtains d_x and d_y from (1) to (11). Then, the fuzzy logic system for robot movement is established to get *turn* and *speed*. From Figure 14, we have the robot adjusting its steps to avoid the obstacles in Figures 14(a)–14(p). The corresponding visions obtained from the robot are shown

in Figures 15(a)–15(p).We find that when the obstacle is in the focus area, the robot avoids it using the proposed algorithm until there is no obstacle in the focus area.

To show the performance of the algorithm, the repeatability test is carried out. The test experiments are shown in Figure 16. In Figure 16, the obstacles are located in the left, center, and right with equal and non-equal distribution and the robot stands on three different positions: left, center, and right. Each experiment is tested for three times and the result is summarized in Table 2 and it shows that about 92.6% of test experiments can pass the obstacles by the proposed algorithm.

5 Conclusions

This study describes an obstacle avoidance system for a humanoid robot employing just a webcam as its vision system. The proposed algorithm uses the image obtained from the robot to calculate navigation grids. We define a focus area that is the obstacle region that the robot needs to avoid. Therefore, the distances from *y* and *x* directions are calculated by the filter processes. Then, a fuzzy logic system is used to decide the *speed* and *turn* parameters for the robot. The experiment results show that the humanoid robot can avoid the obstacles and the boundary lines in an experiment and that it can access any route with its obstacle avoidance ability to implement the task of the obstacle-run event.

Acknowledgment

This work is supported by Ministry of Science and Technology of Taiwan under Grants: MOST 104-2221-E-130-012.

References

Benet, G., Blanes, F., Simó, J. E. & Pérez, P. 2002. Using infrared sensors for distance measurement in mobile robots. *Robotics and Autonomous Systems* **40**(4), 255–266.

Budiharto, W., Moniaga, J., Aulia, M. & Aulia, A. 2013. A framework for obstacles avoidance of humanoid robot using stereo vision. *International Journal of Advanced Robotic Systems* **10**, 1–7.

Chao, C.-H., Hsueh, B.-Y., Hsiao, M.-Y., Tsai, S.-H. & Li, T.-H. S. 2009. Fuzzy target tracking and obstacle avoidance of mobile robots with a stereo vision system. *International Journal of Fuzzy Systems* **11**(3), 183–191.

Chen, C.-Y., Chiang, S.-Y. & Wu, C.-T. 2014. Path planning and obstacle avoidance for omni-directional mobile robot based on Kinect depth sensor. In *National Symposium on System Science and Engineering*, June.

Hancock, J., Hebert, M. & Thorpe, C. 1998. Laser intensity-based obstacle detection intelligent robots and systems. In *Proceedings of the IEEE Conference on Intelligent Robotic Systems*, **3**, 1541–1546.

He, L., Chao, Y., Suzuki, K. & Wu, K. 2009. Fast connected-component labeling. *Pattern Recognition* **42**(9), 1977–1987.

Hsia, C.-H., Chang, W.-H. & Chiang, J.-S. 2012. A real-time object recognition system using adaptive resolution method for humanoid robot vision development. *Journal of Applied Science and Engineering* **15**(2), 187–196.

Li, H. & Yang, S. X. 2002. Ultrasonic sensor based fuzzy obstacle avoidance behaviors. In *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, **2**, 644–649.

Li, T.-H. S., Chang, S.-J. & Tong, W. 2004. Fuzzy target tracking control of autonomous mobile robots by using infrared sensors. *IEEE Transactions on Fuzzy Systems* **12**(4), 491–501.

Mendel, J. M. 1995. Fuzzy logic systems for engineering: a tutorial. *Proceedings of IEEE* **83**(3), 345–377.

Soumare, S., Ohya, A. & Yuta, S. 2002. Real-time obstacle avoidance by an autonomous mobile robot using an active vision sensor and a vertically emitted laser slit. In *Intelligent Autonomous Systems*, 301–308.

Wong, C.-C., Hwang, C.-L., Huang, K.-H., Hu, Y.-Y. & Cheng, C.-T. 2011. Design and implementation of vision-based fuzzy obstacle avoidance method on humanoid robot. *International Journal of Fuzzy Systems* **13**(1), 45–54.