Design and Implementation of Double Passing Strategy for Humanoid Robot Soccer Game

Ping-Huan Kuo, Ya-Fang Ho, Te-Kai Wang and Tzuu-Hseng S. Li

Abstract The goal of this paper was to accomplish a technical challenge of double passing soccer game for humanoid soccer robots in RoboCup competition. Using only a vision sensor, the control strategies for the technical challenges of humanoid league in RoboCup are designed and presented. The vision system includes the color space setting, the object recognition, a simplified mean shift algorithm, and the target position derivation. Vision system works on the tasks of object recognition, which includes the goal, landmark poles, and the interval of two black poles. The computational time is reduced greatly by the mean shift algorithm and that time can be utilized to do other control strategies. With the proposed control strategies, humanoid robots can successfully complete the RoboCup double passing task. The successful experiment results demonstrate the feasibility and effectiveness of the proposed foot–eye coordination control scheme.

Keywords Double passing strategy · Foot–eye coordination control scheme · Humanoid robot soccer game

P.-H. Kuo · Y.-F. Ho · T.-K. Wang · T.-H.S. Li (✉)
aiRobots Laboratory, Department of Electrical Engineering,
National Cheng Kung University, Tainan City, Taiwan, ROC
e-mail: thsli@mail.ncku.edu.tw

P.-H. Kuo
e-mail: coll22000@hotmail.com

Y.-F. Ho
e-mail: makinosakuya@hotmail.com

T.-K. Wang
e-mail: greatmutaxx@gmail.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_29

1 Introduction

Many researches of robots have developed in the world for decades of years. Different kinds of robots are designed for corresponding specific uses. It is believed that the development of robots has mainly been in the areas of their interacting with humans and helping them with their needs. So a lot of methods have been proposed for human–robot interaction and communication [1–4]. Since the hardware computational ability improves fast with the size of the devices become smaller and smaller, building a robot for our own is not a difficult thing anymore. The goal of this paper was to design and implement a team (aiRobots-V) of humanoid robots for robot soccer games. A humanoid robot is a highly integrated system including mechanism design, passive sensors, vision system, power supply system, communication system, and algorithms in software programming. The robot soccer game presents a dynamic and complex environment. It could be a challenging platform for multi-agent research, involving many topics such as motion generation [5–9], motion control, image processing algorithm, localization, and collaboration strategy planning. They aim to promote artificial intelligence (AI), robotics, and related fields. The ultimate goal of the RoboCup [10] project is as follows: By the year 2050, develop a team of fully autonomous humanoid robots that can play against the human world champion soccer team and win the game. We attend these competitions in order to enhance and stimulate our technique and development of the humanoid robot. At the same time, we can learn and try to make strides by competing with other teams from different countries. We will also attend the challenge games the 3 on 3 soccer games of RoboCup 2010 humanoid league.

Besides, the robots need to be autonomous and intelligent during the soccer games. Thus, strategies of many cases have to be taken into consideration. Strategies are also programmed in the embedded computer. Robots will communicate with each other via wireless network to accomplish the collaboration strategies. The role of each robot will switch dynamically by itself to reach the best efficiency for offense and defense. The aim of this paper was to develop a fast processing vision system and robust strategy system for not only humanoid robot soccer competition but also challenge games of double passing.

2 Vision System

The vision system used in this paper includes two essential parts: object recognition and simple mean shift algorithm, which are described in the following subsections.

2.1 *Object Recognition*

In the soccer playing field, there are five important objects: ball, landmark pole, goal, black pole, and interval space between the black poles. These need to be recognized for the technical challenge since the goals and landmark poles have been changed in RoboCup 2010, and the interval between the black poles is important for the dribbling event. The goal, landmark poles, and interval detections are presented as follows.

2.1.1 Goal Recognition

After the object with the same color goal is segmented by the YUV lookup table, the following steps are applied to check whether it has the characteristics of a goal.

Step 1 The width and height of the goal must be wider and taller than a set threshold.

Step 2 The goal must be situated on the game field, so the color under the goal probably belongs to the field or the lines. It is necessary to check this condition. Two places will be checked: the lower right corner and the lower left corner. For each corner, there is a 3 × 3 window. Each pixel will be scanned and checked to see whether it is the color of the game field or a white line or something else. An illustration of the goal-checking windows is shown in Fig. 1a. However, the view may be occluded by other robots during the game time, as shown in Fig. 2. Since the robots must be mostly black according to the rules of the RoboCup, the black color also has to be checked in those two windows.

Step 3: Because of the shape of the goal, there should not be any color belonging to the goal. There is a 5 × 5 window in the middle of the goal to scan and check the color in that window to make sure it is not the color of the goal. It is illustrated in Fig. 1b.

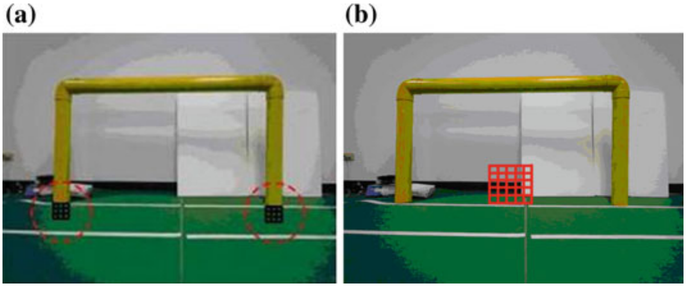


Fig. 1 Illustration of the goal-checking windows of **a** 3 × 3 **b** 5 × 5 pixels

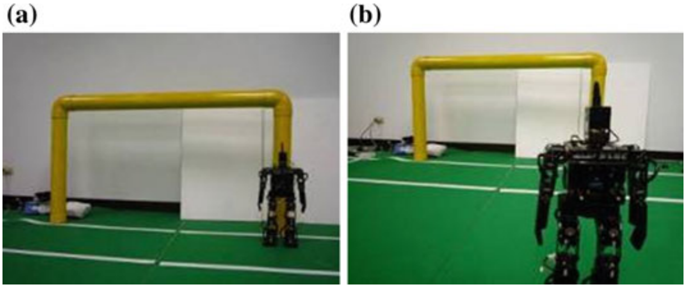


Fig. 2 Examples of a view of a goal occluded by another robot

In steps 2 and 3, every pixel in the window is scanned and a reward function is adopted. If the scanned pixel meets the expectation, the score goes up; otherwise, the score goes down. The reward function can be described as follows:

$$\begin{cases} G_{\text{score}} = G_{\text{score}} + \alpha_G, & \text{if the pixel fits the right color} \\ G_{\text{score}} = G_{\text{score}}, & \text{otherwise} \end{cases} \tag{1}$$

where G_{score} is the total reward and α_G is the reward value; the recognition succeeds if $G_{\text{score}} > \beta$. Otherwise, the recognition fails. β is a threshold defined by the user. The results of the goal recognition are shown in Figs. 3 and 4. Nevertheless, when the robot stands very close to the goal, it can only see part of the goal. Figure 5 gives an example of a partial view of the goal.

In this case, the robot will be asked to raise its head, and then, the view will be clearer. It is able to determine which side of the goal the robot sees by sampling of the goal and calculating the vector, as shown in Fig. 6. After the color of the goal is segmented out, a vertical sampling from the left to the right of the image will be done as shown in Fig. 6a and e show. Then, the vector is calculated by:

$$V_g = \begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} x_R \\ y_R \end{bmatrix} - \begin{bmatrix} x_L \\ y_L \end{bmatrix} \tag{2}$$

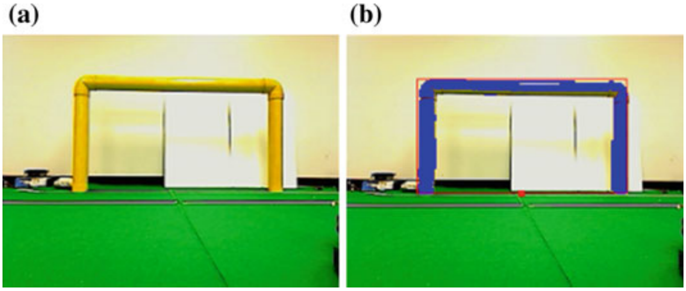


Fig. 3 The result of the goal recognition. **a** The original image, **b** result of the goal recognition

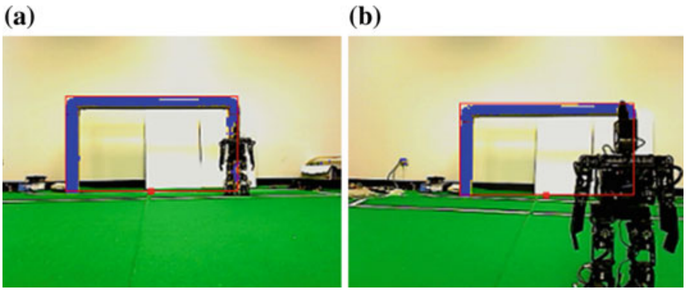


Fig. 4 Goal recognition with occlusion of another robot. **a** Occlusion of a robot (far), **b** occlusion of a robot (near)

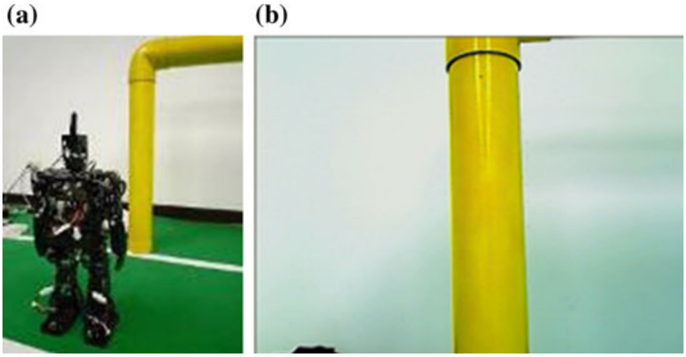


Fig. 5 An example of a partial view of the goal. **a** The relative positions of the robot and goal, **b** view of a robot

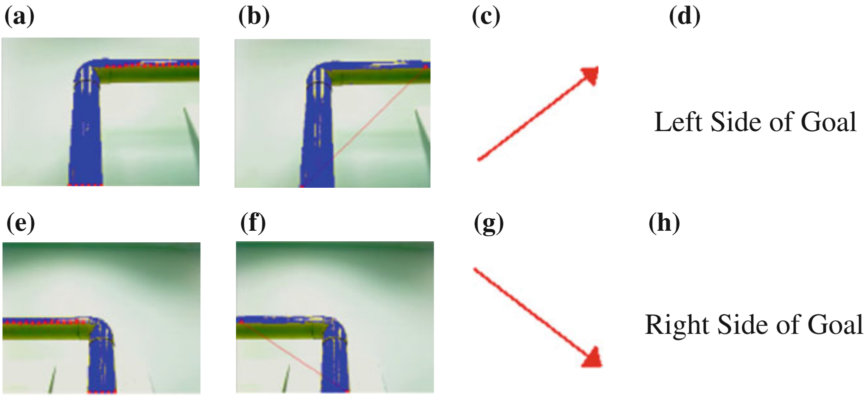


Fig. 6 The recognition result for a partial view of the goal. **a** Sampling points, **b** calculating the vector, **c** the vector, **d** result, **e** sampling points, **f** calculating the vector **g** the vector, **f** result

where (x_R, y_R) is the coordinate of the rightmost sampled point, (x_L, y_L) is the coordinate of the leftmost sampled point, and $V_g(x_g, y_g)$ is the result vector. If the vector points to the upper right corner, this means that the robot sees the left side of the goal. If the vector points to the lower right corner, this means that the robot sees the right side of the goal.

2.1.2 Interval Detection

In the double passing event, there are two more black poles of 30 cm in radius and 90 cm in height. One pass must go through the interval of the two black poles. The interval is about 120 cm in width. Figure 7a shows the two black poles. The detection of the interval is via a sampling method, and the method is divided into several steps.

Step 1 Do the sampling vertically, and the sampling interval is 10 pixels. A group of sample points will be collected. Figure 7b demonstrates the sampling.

Step 2 Check from the first point. If the distance between this point and the next point is long enough and the vector points to the bottom, one black pole is found. Keep scanning the points until there is a vector that orients the opposite way, and then one black pole is detected. All the points will be checked in this step. The distance is given by

$$D_i = |y_i - y_{i-1}|, \quad i = 1 \sim 32 \tag{3}$$

where D_i is the distance between ith point and $(i - 1)$th point and y_i represents the coordinate value of the y-axis. Since the sampling interval is 5 pixels, the difference in the x-axis from every point to the next one will be constant. It is not necessary to take the x-axis into consideration. The computation is reduced in this way. The vector is given by

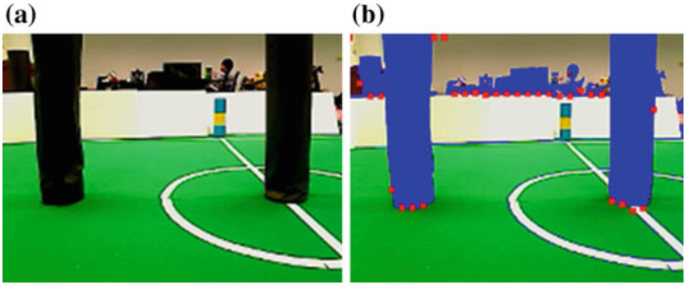
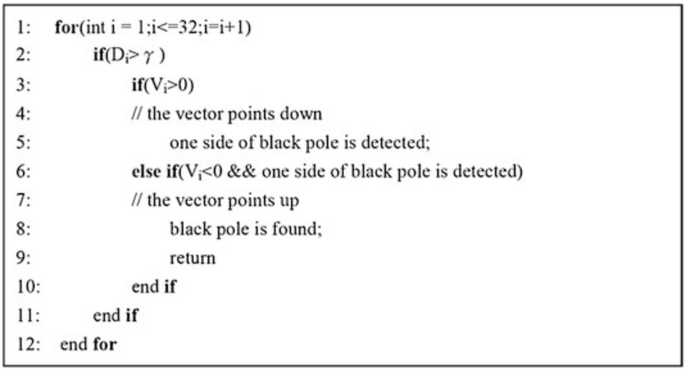


Fig. 7 **a** The position of the black poles in a double passing game. **b** The sampling points of the *black color*



```
for(int i = 1;i<=32;i=i+1)
    if($D_i > \gamma$)
        if($V_i$>0)
        // the vector points down
            one side of black pole is detected;
        else if($V_i$<0 && one side of black pole is detected)
        // the vector points up
            black pole is found;
            return
        end if
    end if
end for
```

Fig. 8 Pseudo code of step 2

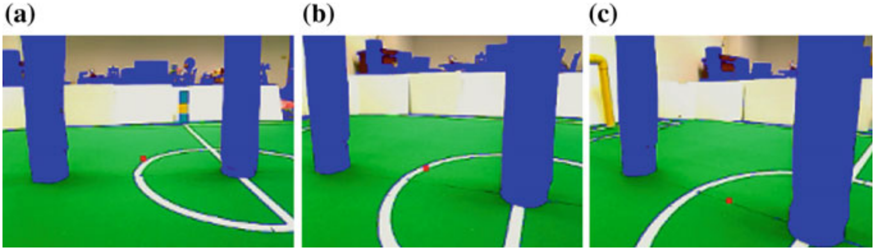


Fig. 9 Results of interval detection of two black poles from different views

$$V_i = y_i - y_{i-1}, \quad i = 1 \sim 32 \tag{4}$$

where V_i is the vector from $(i - 1)$th point to ith point. As denoted in (4), the x-axis is not taken into consideration. The pseudo code of step 2 is shown in Fig. 8.

Step 3 If there are two black poles found in Step 2, the width of their interval can be calculated. The center of the interval is then pointed out. The interval detection is completed, and the result is shown in Fig. 9.

2.2 Simple Mean Shift Algorithm

After the recognitions are done, the next step is to keep tracking the target. There are many papers that discuss different kinds of methods or algorithms for tracking. But actually, the target is not that big, so it is not necessary to always search the whole image. Besides, searching the entire image for a small target is a waste of time and increases the computation of the system. So a mean shift algorithm is adopted to solve this problem. The mean shift algorithm has been widely discussed

[11–15]. The algorithm only searches the region of interest, and the size of the region is adaptive to the variation in size of the target. This section shows how the algorithm works.

2.2.1 Principal Concept of Mean Shift

The main idea and intuitive description of mean shift algorithm can be illustrated as shown in Fig. 10. The red dots shown in Fig. 10 mean the global data distribution. The algorithm aims to find the maximum distribution of the data. In the case of the target tracking in this research, red dots are considered as pixels whose color in YUV channel is similar to the color of the target. The main difference is that the data in our case will be concentrated but not scattered because every target has its distinguishing color organization. For example, if there is only one orange ball on the game field, it is impossible to see the scattered orange color distributed on the field. The data distribution is shown in Fig. 11. Another different point is that, unlike in most cases, the shape of the region is square in this research. The square shape is convenient for searching digital image data.

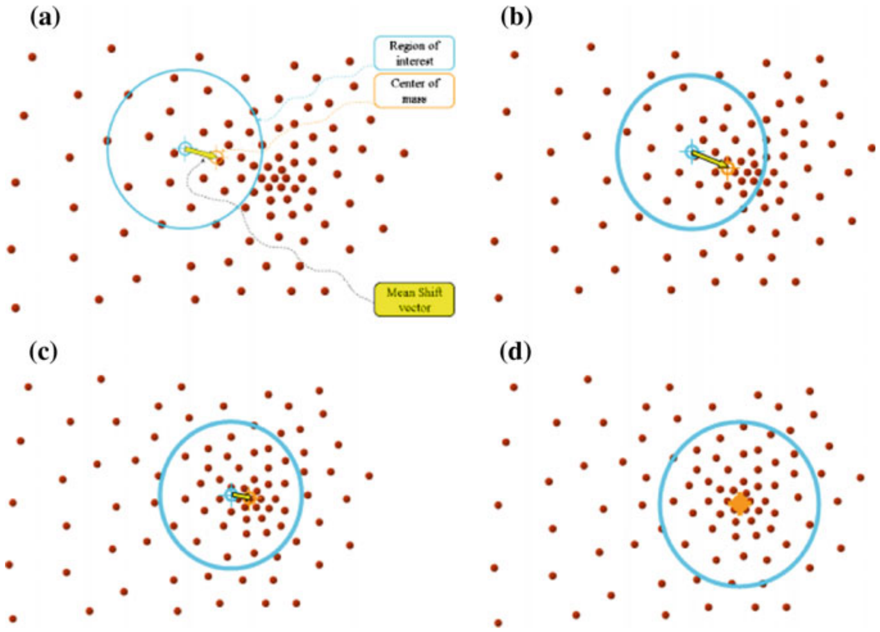


Fig. 10 The principal concept of mean shift algorithm. **a** Calculate the center of mass (COM) of the data in the region of interest, **b** shift the region of interest to the COM and then calculate the new COM, **c** keep shifting the region of interest as in (**b**), **d** the shifting procedure is completed when COM is just the center of the region

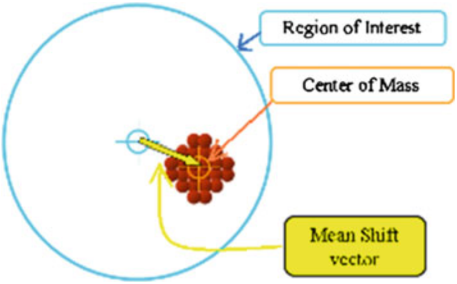


Fig. 11 The data distribution in the case of the ball target

2.2.2 Simplified Mean Shift Algorithm

The first step is to initialize the region of interest. In the beginning, scanning all the pixels of an image is needed because the data only gather together somewhere but are not scattered in the whole image. We can say that the initial region of interest is the whole image. When the target is found, the region is then determined and will keep shifting with the target until the target is lost. If the target disappears in the region of interest, the procedure goes back to search the entire image. A diagram of a simplified mean shift algorithm is shown in Fig. 12. The center of mass $C(x_c, y_c)$ is computed by

$$X_c = \frac{\sum_{i=1}^{n} x_i}{n}, \quad \forall x_i \neq 0 \tag{5}$$

$$Y_c = \frac{\sum_{i=1}^{n} y_i}{n}, \quad \forall y_i \neq 0 \tag{6}$$

where x_i and y_i are the coordinate values of the pixels of the target on the image plane and n is the total number of pixels of the target. Then, the mean shift vector $V_m(x_m, y_m)$ is given by

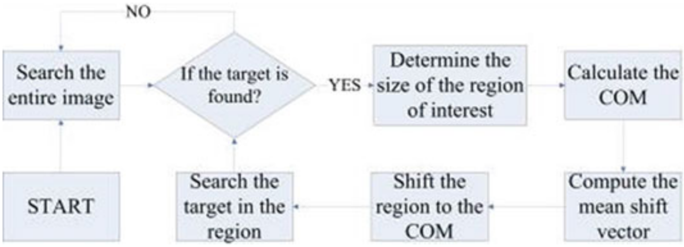

NO
Search the entire image
If the target is found?
YES
Determine the size of the region of interest
Calculate the COM
START
Search the target in the region
Shift the region to the COM
Compute the mean shift vector

Fig. 12 Diagram of a simplified mean shift algorithm

$$V_m = \begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} X_c - X_{\text{region}} \\ Y_c - Y_{\text{region}} \end{bmatrix} \tag{7}$$

where $\left(X_{\text{region}}, Y_{\text{region}}\right)$ is the center of the region of interest. So the new center of region $\left(X'_{\text{region}}, Y'_{\text{region}}\right)$ is as follows:

$$\begin{bmatrix} X'_{\text{region}} \\ Y'_{\text{region}} \end{bmatrix} = \begin{bmatrix} X_{\text{region}} + x_m \\ Y_{\text{region}} + y_m \end{bmatrix} \tag{8}$$

The size of a target that is shown on the captured image varies when the distance from the robot and target changes. A fixed size of a searching region is not reasonable. So the size of the region or, more specifically, the side length of the square will be adaptive to the size of the target. The side length is adjusted according to the size of the target. The adaptive mechanism is given by

$$L_{\text{region}} = k \times S_{\text{target}} \tag{9}$$

where L_{region} is the side length of the region and the S_{target} is the size information of the target. S_{target} may be different from other targets. For instance, S_{target} is the radius in the case of the ball and the width in the case of the pole and goal. k is a related coefficient. With the adaptive region, the algorithm is more efficient and time-saving. The results are shown in Fig. 13. The tracked target in this case is a ball. There are two squares in every picture in Fig. 13.

The outer square is the region of interest, and the inner square is the boundary of the target after the recognition is successful. It is obvious that the side length of the region of interest varies according to the area of the target. When the ball is far from the robot, the region becomes smaller; otherwise, it becomes bigger. The efficiency is raised in this way.

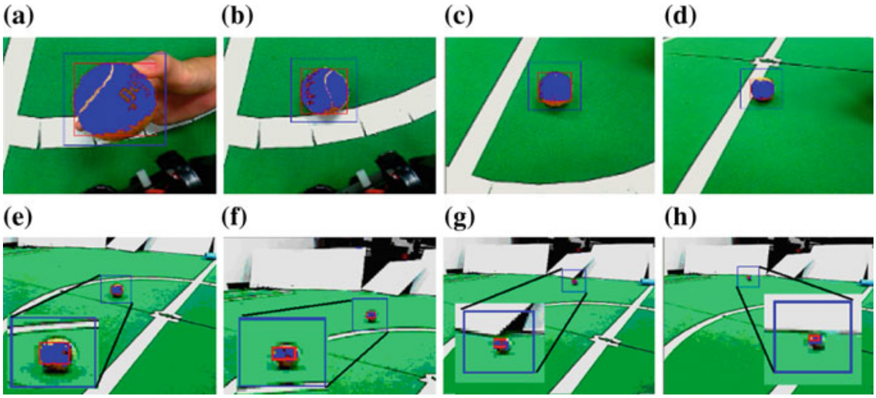


Fig. 13 The result of a simplified mean shift algorithm

3 Control Strategy

The control strategy is divided into two independent processes because there are two players and their tasks are a little different. In order to achieve the strategy, target tracking by the head of the robot is required because many procedures in the strategy need to track the target, walk toward it, and calculate the distance from it. So, once the current target is found, keeping it in the image all the time is important. Here, the fuzzy logic controller (FLC) implemented in [16] is adopted to solve the problem. The FLC is done by controlling the two motors as pan and tilt on the head. In this way, the following strategies can run smoothly. The flowchart strategy of robot Player A is shown in Fig. 14a. As shown in Fig. 14a, the process of a control strategy for Player A is partitioned into three parts. Process 1 is to make the first pass to Player B and then walk toward the left pole until the distance from Player A is less than a certain length and turn back to face the game field to wait for the ball to be passed back. Process 2 is to pass the ball through to the interval between the two black poles, walk toward the goal until the distance from the goal is less than a certain length, and then wait for the ball to be passed back. One of the steps in process 2 is to dribble the ball in the direction of the goal. This step can make the second pass of Player A easier according to the many times of experiments. The third process is to make a goal when the ball is passed back from Player B, just as in a normal soccer game.

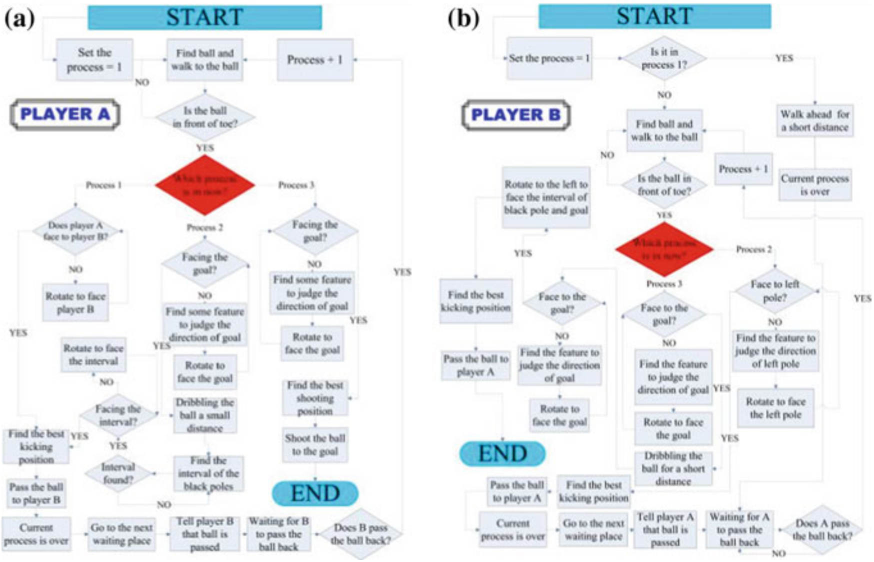

(a)
START
Set the process = 1
Find ball and walk to the ball
Process + 1
NO
PLAYER A
Is the ball in front of toe?
YES
Process 1
Process 3
Process 2
Does player A face to player B?
NO
Rotate to face player B
YES
Facing the goal?
NO
Find some feature to judge the direction of goal
YES
Rotate to face the goal
Dribbling the ball a small distance
Find the interval of the black poles
Facing the goal?
NO
Find some feature to judge the direction of goal
YES
Rotate to face the goal
Find the best shooting position
Shoot the ball to the goal
END
YES
Rotate to face the interval
NO
Facing the interval?
YES
YES
Interval found?
NO
Find the best kicking position
Pass the ball to player B
Current process is over
Go to the next waiting place
Tell player B that ball is passed
Waiting for B to pass the ball back
Does B pass the ball back?
(b)
START
Set the process = 1
Is it in process 1?
YES
NO
PLAYER B
Walk ahead for a short distance
Find ball and walk to the ball
NO
Process + 1
Current process is over
Is the ball in front of toe?
YES
Rotate to the left to face the interval of black pole and goal
YES
Process 2
Process 3
Find the best kicking position
Face to the goal?
NO
Find the feature to judge the direction of goal
Rotate to face the goal
Pass the ball to player A
Face to the goal?
NO
Find the feature to judge the direction of goal
YES
Rotate to face the goal
YES
Dribbling the ball for a short distance
Face to left pole?
NO
Find the feature to judge the direction of left pole
Rotate to face the left pole
YES
END
Pass the ball to player A
Find the best kicking position
Current process is over
Go to the next waiting place
Tell player A that ball is passed
Waiting for A to pass the ball back
Does A pass the ball back?
NO

Fig. 14 Flowchart for the double passing strategy of the robots. **a** Player A and **b** Player B

4 Experiment Results

The full procedure for double passing is demonstrated in Figs. 15, 16, 17, 18, and 19. Figure 15 shows the first pass from robot Player A to robot Player B. The first kick is very slight because the two players are very close to each other. Figure 16

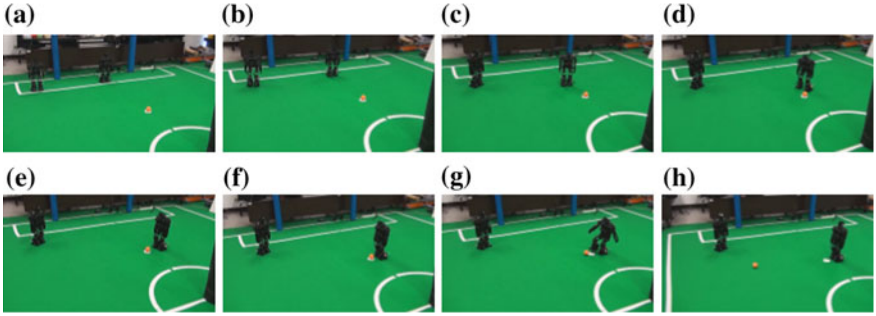


Fig. 15 The first part of the result of the double passing strategy

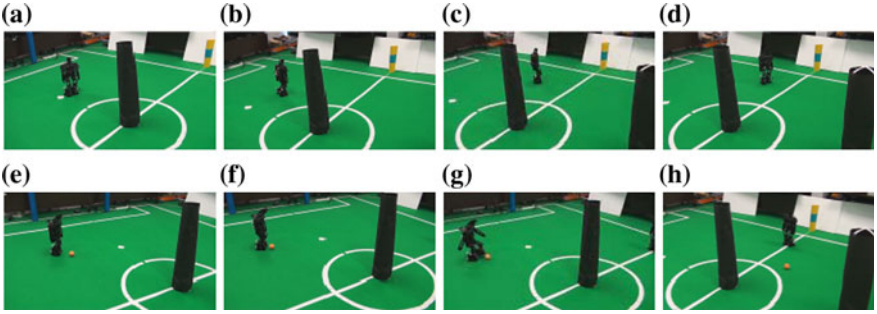


Fig. 16 The second part of the result of the double passing strategy

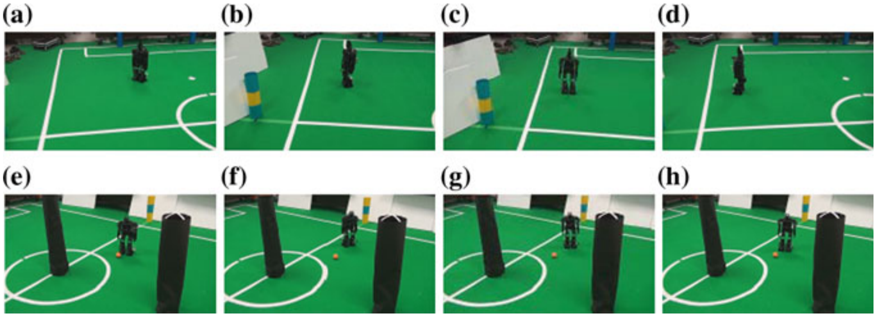


Fig. 17 The third part of the result of the double passing strategy

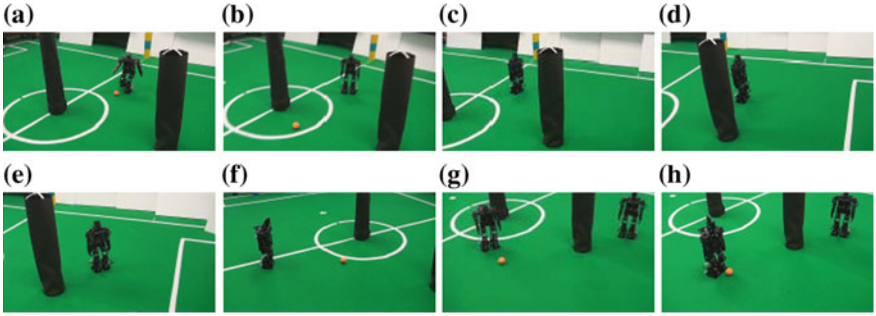


Fig. 18 The fourth part of the result of the double passing strategy

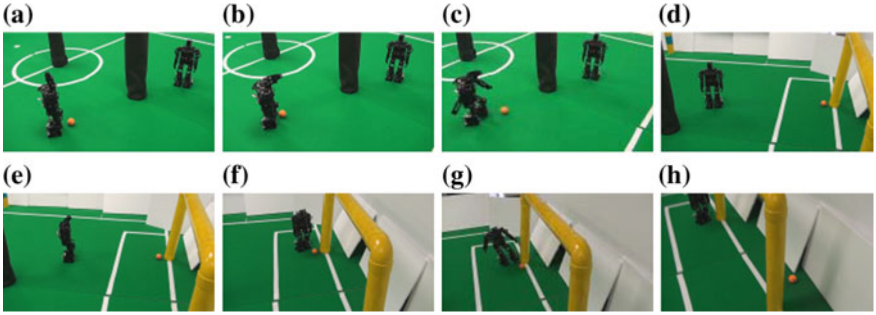


Fig. 19 The last part of the result of the double passing strategy

shows the second pass from Player B to A. The strength of the kick is stronger than in the first pass. The strength is not as strong for shooting the goal because the ball must be controlled in the field. Figures 17 and 18 demonstrate the third pass from Player A to B. The strength of the kick is like the second pass to maintain the ball inside the field. Figure 19 shows the last task of the double passing. Player A shoots the ball into the goal. After the ball is in the goal, the two robots will stay at their position. Finally, the double passing is then completed after the last shot.

5 Conclusions

This paper has proposed a complete and systematic vision system and foot–eye coordination control for humanoid soccer robots. The major objects on the field can all be recognized by different methods. A simplified mean shift algorithm is proposed for target tracking, and it searches the region of interest instead of always searching the entire image. In addition, the area of the region is adaptive according to a change in size of the target. It can save more computation time than the original mean shift algorithm by the adaptive searching region. Finally, the foot–eye coordination control strategies are presented for the double passing challenges. The control strategies allow the robots to successfully implement the technical challenge of the humanoid league of the RoboCup. For the dribbling challenge, the proposed control scheme provides a complete and effective strategy for robots to accomplish the tasks. Additionally, the proposed foot–eye coordination control scheme is used in humanoid soccer robots, which entered the humanoid league of the RoboCup in 2010 and won second place in the technical challenge of RoboCup 2010.

Acknowledgment This work supported by Ministry of Science and Technology of Taiwan, R.O. C, under Grants MOST 103-2221-E-006-252, MOST 104-2221-E-006-228-MY2, and the aim for the Top University Project to the National Cheng Kung University (NCKU) is greatly appreciated.

References

1. Stiefelhagen, R., Ekenel, H.K., Fugen, C., Gieselmann, P., Holzapfel, H., Kraft, F., Nickel, K., Voit, M., Waibel, A.: Enabling multimodal human-robot interaction for the Karlsruhe humanoid robot. IEEE Trans. Robot. **23**, 840–851 (2007)
2. Spexard, T.P., Hanheide, M., Sagerer, G.: Human-oriented interaction with an anthropomorphic robot. IEEE Trans. Robot. **23**, 852–862 (2007)
3. Lee, J.M., Choi, J.S., Lim, Y.S., Kim, H.S., Park, M.: Intelligent and active system for human-robot interaction based on sound source localization. In: Proceedings of International Conference on Control, Automation and Systems, ICCAS, pp. 2738–2741 (2008)
4. Rivera-Bautista, J.A., Ramirez-Hernandez, A.C., Garcia-Vega, V.A., Marin-Hernandez, A.: Modular control for human motion analysis and classification in human-robot interaction. In: Proceedings of 5th ACM/IEEE International Conference on Human-Robot Interaction, HRI, pp. 169–170 (2010)
5. Su, Y.-T., Chong, K.-Y., Li, T.-H.S.: Design and implementation of fuzzy policy gradient gait learning method for walking pattern generation of humanoid robots. Int. J. Fuzzy Syst. **13**(4), 369–382 (2011)
6. Li, T.-H.S., Su, Y.-T., Lai, S.-W., Hu, J.-J.: Walking motion generation, synthesis, and control for biped robot by using PGRL, LPI and fuzzy logic. IEEE Trans. Syst. Man, Cybern. B **41**(3), 736–748 (2011)
7. Kuo, P.-H., Ho, Y.-F., Lee, K.-F., Tai, L.-H., Li, T.-H.S.: Development of humanoid robot simulator for gait learning by using particle swarm optimization. In: Proceedings of 2013 IEEE International Conference on System, Man, Cybernetics, pp. 2683–2688 (2013)
8. Hong, Y.-D., Park, C.-S., Kim, J.-H.: Stable bipedal walking with a vertical center-of-mass motion by an evolutionary optimized central pattern generator. IEEE Trans. Ind. Electron. **61** (5), 2346–2355 (2014)
9. Li, T.-H.S., Kuo, P.-H., Ho, Y.-F., Kao, M.-C., Tai, L.-H.: A biped gait learning algorithm for humanoid robots based on environmental impact assessed artificial bee colony. IEEE Access. **3**, 13–26 (2015)
10. RoboCup. http://www.robocup.org/
11. Pooransingh, A., Radix, C.A., Kokaram, A.: The path assigned mean shift algorithm: a new fast mean shift implementation for colour image segmentation. In: Proceedings of 15th IEEE International Conference on Image Processing, ICIP, pp. 597–600 (2008)
12. Wang, H., Yang, B., Tian, G., Men, A.: Object tracking by applying mean-shift algorithm into particle filtering. In: Proceedings of 2nd IEEE International Conference on Broadband Network and Multimedia Technology, IC-BNMT, pp. 550–554 (2009)

13. Li, Y.H., Pang, Y.G., Li, Z.X., Liu, Y.L.: An intelligent tracking technology based on Kalman and mean shift algorithm. In: Proceedings of Second International Conference on Computer Modeling and Simulation, ICCMS, pp. 107–109 (2010)
14. Yafeng, Y., Hong, M.: Adaptive mean shift for target-tracking in FLIR imagery. In: Proceedings of Wireless and Optical Communications Conference, WOCC, pp. 1–3 (2009)
15. Liu, Y., Peng, S.: A new motion detection algorithm based on snake and mean shift. In: Proceedings of Congress on Image and Signal Processing, CISP, pp. 140–144 (2008)
16. Su, Y.T., Hu, C.Y., Li, T.H.S.: FPGA-based fuzzy PK controller and image processing system for small-sized humanoid robot. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, SMC, pp. 1039–1044 (2009)