Determining Hit Time and Location of the Ball in Humanoid Robot League

Saeed Javadzadeh[1], Abdolah Chalechale[2*] and Ehsan Zadkhosh[3]

Department of Computer Engineering
Razi University, Kermanshah, Iran
[1]saeed_javadzadeh@yahoo.com, [2]chalechale@razi.ac.ir [3]e_zadkhosh@yahoo.com

[*]*Corresponding Author*

Abstract

Goalkeeper robot in the league of humanoid robots has a key role so that its wrong decision and incorrect jump can cause changes in the result of a competition. This paper presents a method to detect the ball on the ground during the match, determine the coordinates of the ball with respect to the robot, calculating the speed and direction of the ball and eventually the time and place that the ball hits the goal. This can be the basis of making correct decision for the goalkeeper robot. The method is based on chasing the ball and calculating the coordinates of collision point. The results of applying this method on small size humanoid robot was excellent, except for the situations that inadequate rate of imaging lead to losing the ball or mistaken a similar object for the ball by the robot, quite right decisions in other states were made.

Keywords: *Humanoid robot, ball detection, ball coordinates, the collision point.*

1. Introduction

Although it is not a long time that the humanoid soccer robot league is being held in robotic competitions, great progress has been made in this field. This level of progress is clearly obvious in many tournaments in comparison with the previous ones. Currently soccer robots do not have the ability to engage with an opponent while it possesses the ball and usually if the carrier robot has enough capabilities it will have the chance to hit the ball toward the opponent goal. In such circumstances the role of goalkeeper robot in the game is vital and the team that has a good goalkeeper can save lots of balls. In order to save the ball the goalkeeper often jumps toward it after determining the direction and to ensure that it will save the ball the goalkeeper will maintain its position for a while. After that it stands up and throws the ball away and then with localization [1], [2] it goes toward the center of its goal. Since humanoid robots currently have low speed the above steps require a considerable amount of time. So if the robot jumps because of a wrong decision without any ball nearing the goal or jumps in the wrong direction after a shoot, it will lose considerable time of the game. We can conclude that the goalkeeper's correct decision and movement toward the ball play an important role in humanoid robots football matches. In this paper we will present a method to detect the ball direction, determine the speed, acceleration and the exact time and place that the ball hits the goal. This can be the basis of making precise decision for the goalkeeper robot. We implemented our approach on the small size humanoid goalkeeper robots and by its excellent performance we won the third place IranOpen2009 and second position in IranOpen2010.

This paper will have the following sections: in next section we present a review of related work. Section three consists of image processing algorithms to detect the ball on the earth. In the fourth section, method of chasing the ball (Ball Tracking) by the camera of the robot will be explained then the exact coordinates of the ball on the ground will be calculated. Finally we discuss about determining the ball direction, speed, time, and its collision point with the goal. In Section five practical results will be presented and the last section concludes the paper and imposes some new directions for more researches.

2. Related work

Although in recent decades researches in the field of humanoid robots were of lots of importance attention, still in spite of numerous researches there are not similar and sufficient articles about the correct decision of the goalkeeper to jump. Some articles have been found in the similar areas, such as methods to reduce damage in the collision of robot with the ground [3], [4] or localization and vision of robots [2], [5], but a specific discussion that was related to the goalkeeper robots has only found in [6]. The paper briefly reviews determining the coordinates, speed and direction of the ball with respect to the robot. Then the evaluation of the time of the ball hit with the goal is calculated and a method of quick and less harmful jumps toward the ball has been discussed.

3. Detection of the ball in the field

The main characteristic of humanoid robots is that they are autonomous. This means that all decisions are made by themselves and no external processor can even exist. Due to these characteristics and because of the equilibrium problem the usage of powerful processors is ruled out. Hence, due to low powered CPUs, robots cannot have high level of image processing. To solve this problem currently detection of objects in the field is done by using colors. For each object that the robot wants to identify a certain color is given to. The field is green, goals are blue and yellow, ball is orange and so on. Therefore the main feature in image processing for these robots is color. Consequently for better conclusions and minimizing the dependency on the light of surrounding environment we made use of HSI space for. To convert RGB image to HSI the following equations have been used [7]:

$$Hue = arctan\left(\frac{\sqrt{3}\,(G-B)}{(R-G)+\,(R-B)}\right) \tag{1}$$

$$Int = \frac{(R+G+B)}{3} \tag{2}$$

$$Sat = 1 - \frac{\min\,(R,G,B)}{Int} \tag{3}$$

In the first step an image is taken as an example and by using a program that is designed for this purpose, desirable spots for the ball will be selected. By transferring these points to the HSI space and calculating the average the target will be determined. By calculating a distance that is based on the environmental conditions, the acceptable set of points for the ball will be selected. This filter can be displayed with the following equation [8]:

$$t_i = \begin{cases} 0 : |r_i - a_i| > d \\ a_i : |r_i - a_i| \le d \end{cases} \quad (4)$$

t_i= output image components
r_i= input image components
a_i= target point component
d= permissible distance from target point

To reduce the effect of environment light, no filter is used on the component *Int*. After extracting the appropriate points, the points will be grouped. In this case neighbor points with distance less than n pixels will place in a group. If the number of pixels in a group is less than *m* that group will be regarded as noise and will be omitted (n and m are considered to be 3 and 100 on our experiments). Among the remaining groups the largest group is considered as the ball. The above process is called *Ball Detection Algorithm*. Figure 1 shows an image of the ball on a ground, taken by the robot camera (left) and the processed image by the Ball Detection Algorithm (right).

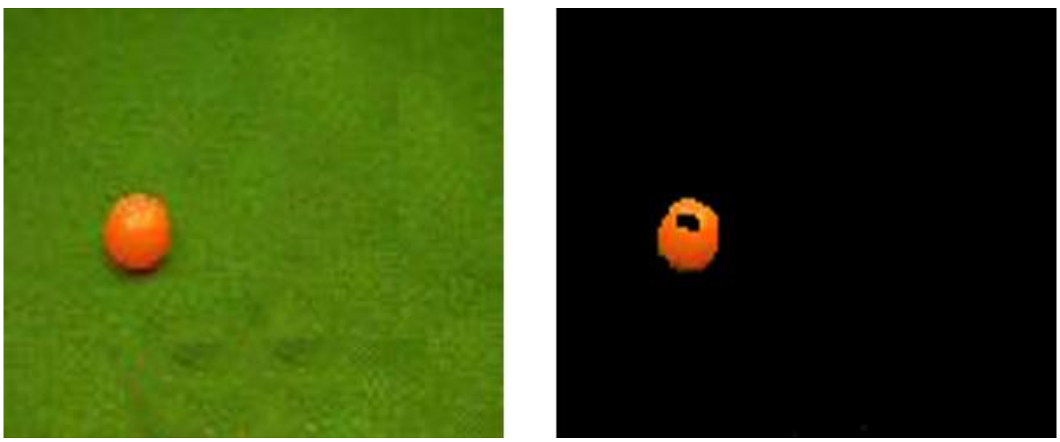

Figure 1. Left: Image of the ball on the ground, taken by the robot camera. Right: the processed image by Ball Detection Algorithm.

Although more advanced methods for segmenting the area based on the color exist [9], considering the fact that the environment is separated in terms of colors and the low-power processors used in the matches, the ball detection algorithm is adequate and provides reasonable results.

4. Extracting the ball properties

4.1. Chasing the ball

During the match, the robot searches different areas to find the ball. The first time when an object with the properties of the ball is found, searching operation ends. At this moment, the robot has an image that part of it is considered as the ball. Now the main objective is to determine the coordinates of the ball with respect to the robot. Since the camera of the robots is placed on two motors that move in horizontal and vertical directions (Pan Tilt), the obtained image should be considered in a spatial coordinate. Extracting the coordinates of the ball just from the image is an imprecise and time consuming action. Therefore, to simplify the computing operations, ball chasing method has been introduced in [10] is adopted. Here, with a proper movement of the head motors the ball will be placed in the middle of the receiving image. If this condition could be satisfied, the ball position toward the robot can be maintained from the angle of the horizontal and vertical motors instead of the calculations on the image. For achieving this, motors should shift the distance of the center of the ball from the middle of the image (in both horizontal and vertical direction) to place the ball at the center of the image. The unit of calculations for image is pixel and for motors the rotation angles. Since the angle of the camera is approximately 70 degrees horizontally and 50 degrees vertically and as the output image has 640 width and 480 height, Hence, each pixel of the image is equivalent to:

$$\frac{70}{640} = \frac{50}{480} \cong 0.1 \qquad (5)$$

So the difference of the center of the ball and the image will be multiplied by 0.1 and will be added to the angle of the head motors in positive or negative directions (depending on the ball placed at the left, right, up or down of the center). With performing these operations if the ball moves, the head motors move quickly based on the new position of the ball, to place the camera along with the ball. Figure 2 shows how the image is shifted to place the ball at the center. By comparing the distance of the ball from the center of image in the horizontal and vertical directions (left image) and applying this distance to head motors the result will be the image on the right.

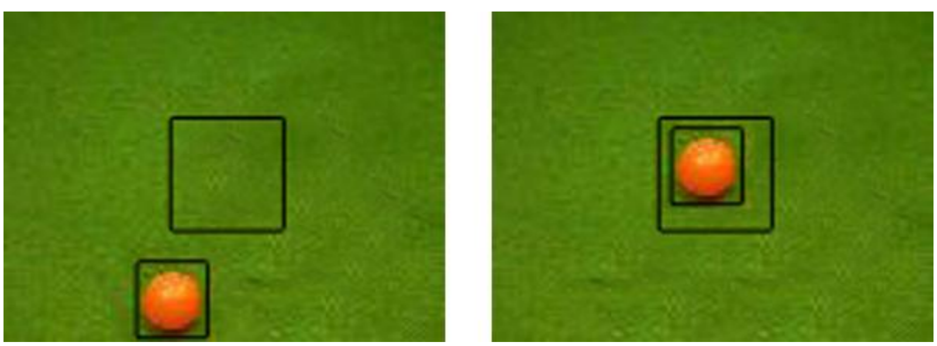

Figure 2. Left: image of the ball on the ground before being transferred to the center. Right: Image of the ball after the operation of chasing and moving the ball to the center.

4.2. Determining the ball coordinates

As the camera is along with the ball, the angle of the horizontal and vertical motors can determine the coordinates of the ball. For this, the robot is considered in the coordinates (0, 0). The distance from the goal in the *Y* axis can be obtained as follows:

$$Y = H \tan(\theta_v) \qquad (6)$$

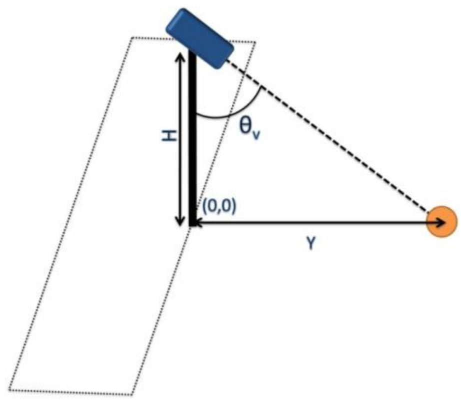

Figure 3. Side view of the robot while the angle of horizontal motor is zero and the angle of vertical motor is θv.

In Figure 3, *H* is robot's tall, *Y* is the distance of the ball from the goal in the *Y* axis and θ_v is the angle of vertical motor. Figure 4 Illustrates upper view of the robot. In this figure *X* is the horizontal distance of the ball from the robot and θ_h is the angle of horizontal motor. Figure 5 exhibits lateral view. Considering Figure 4 and 5, to calculate the horizontal distance of the ball from the robot the following relations are used:

$$X = R \tan(\theta_h) \qquad (7)$$

$$R = \frac{H}{\cos(\theta_v)}$$

Therefore:

$$X = \frac{H \tan(\theta_h)}{\cos(\theta_v)} \qquad (8)$$

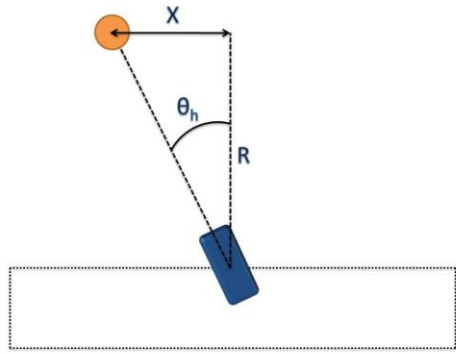

Figure 4. Upper view of robot while the horizontal angle of the motor is θh.

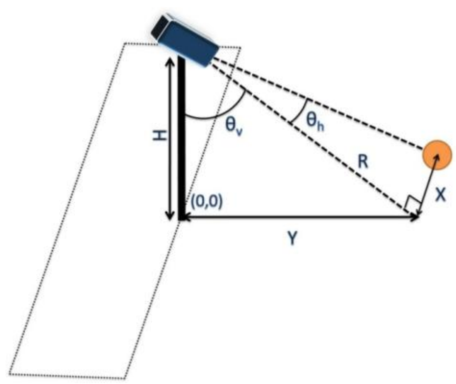

Figure 5. Side view of the robot while the horizontal motor angle is θh and the vertical angle is θv. Thus the coordinate (x, y) of the ball can be calculated at any time.

4.3. Calculation of the motion parameters of the ball

Now the parameters needed to make a right decision for the goalkeeper can be calculated from the coordinates of the ball. The most important parameter is the collision point of the ball with the goal. Goalkeeper robot with this parameter can choose the most appropriate jump (among different predefined type of jumps) and leaps toward the ball. In each image frame, the new coordinates of the ball obtains. By using the previous coordinates and the new coordinates the linear equation that the ball moves on it can be gained. Like what is seen in Figure 6 by crossing the acquired line with the line $y = 0$ (goal line) and using below equations collision point of the ball with the goal can be determined.

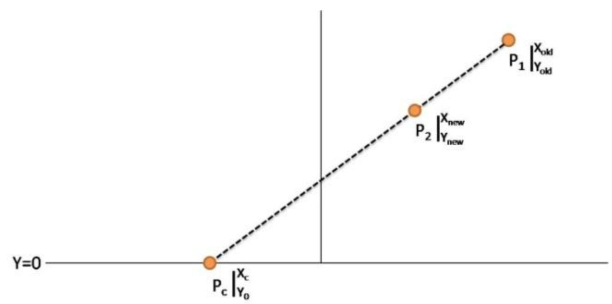

Figure 6. With calculating the coordinates of the ball in two points P1 and P2 the collision point Pc can be calculated.

$$Y - Y_{new} = m(x - x_{new}); \; m = \frac{y_{old} - y_{new}}{x_{old} - x_{new}} \tag{9}$$

$$x_{contact} = \frac{-y_{new}}{m} + x_{new} = \frac{(x_{new} - x_{old})y_{new}}{y_{new} - y_{old}} + x_{new}$$

The next important parameters are speed and acceleration of the ball movement. These two parameters can be calculated absolutely (based on the interval of two frames) or calculated relatively. To reduce computation and increasing the speed and since there is no difference in the result, we have used the relative values. Speed can be computed by using two points and three points are needed to calculate the acceleration. In addition, useful parameters are speed and acceleration of the ball in y direction (in the direction of approaching the goal) which is calculated as bellow:

$$v_y = y_{old} - y_{new} \tag{10}$$

$$a_y = v_{old} - v_{new}$$

Now based on these two parameters it can be estimated whether the ball that is moving toward the goal will reach the goal or not, and if it reaches the exact time of hit can be calculated as follows.

$$V = at + V_0 \xrightarrow{V=0} t_{stop} = \frac{-V_0}{a} \tag{11}$$

$$y = \frac{1}{2}at^2 + V_0 t + y_0 \xrightarrow{t=t_{stop}} L = y - y_0 = \frac{-V_0^2}{2a}$$

At any moment Y that is the ball distance from the goal, will be compared with *L* that is the distance that the ball traverse until it stops. If L> Y the ball will hit the goal and the goalkeeper should jump towards the ball. To calculate the time that the ball hits the goal, the following equation is in used:

$$t_{contact} = \frac{-V_0 \pm \sqrt{\frac{V_0}{-2ay}}}{a} \tag{12}$$

The significance of this time is in the fact that the permitted height for robots in small size robot competition could be up to 60 cm, but generally most teams make use of robots with a height between 40 and 50 cm. On the other hand the goal width is 150 cm in this tournament. So when the goalkeeper robot jumps from the middle of the goal to both sides it may not be able to cover all parts of the goal. Hence, the collision time, provides the chance for the goalkeeper to take a few steps if the ball goes to the end points of the goal. This can be used as a solution to the problem of small height of the robot to catch the ball.

5. Practical Results

The aforementioned procedure was implemented in two different systems on a robot. In the first system we used kind of camera (Havimo) that does not require an additional processor and it will deliver data of images in separated form and as matrixes on the serial line. In this way we can drive it with microcontrollers. Because this camera has no need to additional processors, the robot can keep its balance easier. On the other hand its resolution is low (120 x 160 pixels) and its imaging rate is up to 5 frames per second. Therefore, because of the low capability of the camera the number of errors, like missing the ball or pick up a noise instead of the ball will increase. Results of 50 practical tests which in them the ball has been shot from different locations of the field and with different speeds towards the robot in the goal are shown in Table 1.

Table 1 Results of 50 throws towards the goalkeeper robot with the first system.

Total jumps	*Lost balls*	*wrong object detections*	*wrong jumps*	*correct jumps*
50	8	4	1	37

In the second system we improved the robot and with lessening the balance problem, using more powerful imaging and image processing systems were in hand. In this system we used Genius VGA webcam and ICOP Vega86-6270 processor. Its related codes are written in C#. To speed up processing and imaging rate, codes are written as unsafe codes (direct access to memory) [11], which doubled the image rate. Results of 50 test of this method are in Table 2.

Table 2 Results of 50 throws towards the goalkeeper robot with the second system

Total jumps	*Lost balls*	*wrong object detections*	*wrong jumps*	*correct jumps*
50	3	2	0	45

Figure 7 shows Images of one of these jumps. As it is seen goalkeeper robot with correct diagnosis attempted to jump and prevent the ball from entering the goal

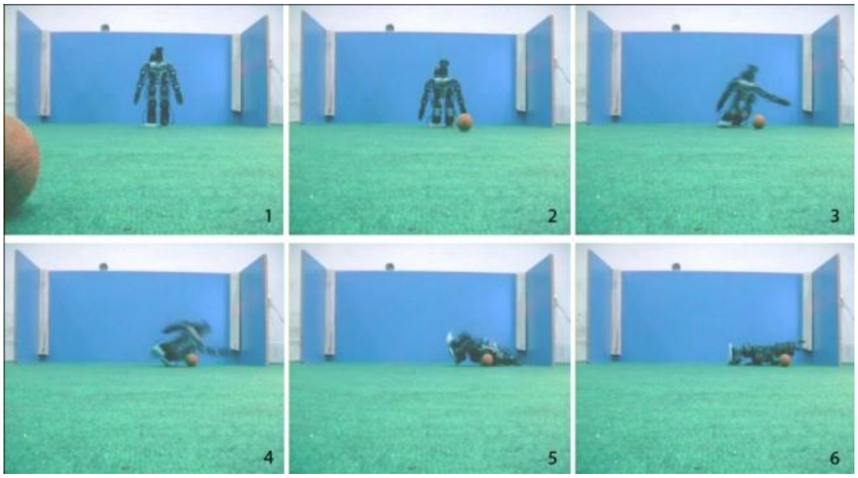

Figure 7. Goalkeeper has detected the direction of the ball and jumped towards it

In the first method, the ball tracking algorithm is not in use and instead of it by analyzing the output data of the camera the coordinates of the ball has been gained. The other steps are exactly the same as what we mentioned in the paper.

As mentioned, by using this method for goalkeeper we won third place in IranOpen2009 and second place in IranOpen2010.

6. Conclusions and Future Work

In this paper a method to detect the ball on the ground, determining its coordinates to the robot, speed and direction of the ball and finally determining when and where the ball hits the goal was presented. The proposed method is based on finding and chasing the ball. In this method, if the motor in the head of the robot could be fast enough and the imaging and image processing rate be adequate we can highly trust on the correct results. There are lots of techniques to improve the results, for example setting a coefficient for changing the angle of head motors equal to the coefficient which is matched with the speed of the ball in both horizontal and vertical directions. In this case, if the ball speed is high, increasing the rotation angle of the motors will decrease the lost probabilities.

References

[1] Sridharan, M., Kuhlmann, G., Stone, P.: Practical vision-based Monte Carlo localization on a legged robot. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA) (2005).

[2] Minakata, Hayashibara, H., Ichizawa, Y., Horiuchi, K., Fukuta, T., Fujita, M., Kaminaga, S., Irje, H., Sakamoto, K., Chiba, H.:A method of single camera robocup humanoid robot localization using cooperation with walking control. In: 10th IEEE International Workshop, pp. 5--55. Trento (2008).

[3] Ruiz-del-Solar, J., Palma-Amestoya, R., Marchant, R., Parra-Tsunekawa, I., Zegers, P.: Learning to fall: Designing low damage fall sequences for humanoid soccer robots. In: science, Robotics and Autonomous Systems, Vol. 57, No. 8, pp. 796--807, (2009).

[4] Ruiz-del-Solar, J., Palma-Amestoy, R., Vallejos, P., Marchant, R., Zegers, P.: Designing Fall Sequences That Minimize Robot Damage in Robot Soccer. In: RoboCup 2008, LNCS, vol 5399, pp. 271—283. Springer, Berlin / Heidelberg (2009).

[5] Ude, A., Shibata, T., Atkeson, C.G.: Real-time visual system for interaction with a humanoid robot. In: Robotics and Autonomous Systems, Vol. 37, pp. 115--125, (2001).

[6] Wilken, T., Missura, M., Behnke, S.: Designing falling motions for a humanoid soccer goalie. In: a workshop of the 2009 IEEE-RAS Intl. Conf. On Humanoid Robots (Humanoids 2009), pp. 79-84, France (2009).

[7] Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J.: Color image segmentation: advances and prospects. In: Science, Pattern Recognition 34, pp. 2259--2281, (2001).

[8] Gonzales, R.C., Woods, R.E.: Digital Image Processing. New Delhi-110001, pp. 431--432, (2008)

[9] Lucchese, L., Mitra, S.K.: Colour Image Segmentation: A State-of-the-Art Survey. PINSA, 67, No. 2, pp. 207--221,(2001).

[10] Kim, S., Lee, S., Kim, S., Lee, J.: Object Tracking of Mobile Robot using Moving Color and Shape Information for the aged walking. In: International Journal of Advanced Science and Technology, Vol. 3, pp. 59--68,(2009).

[11] The Microsoft MSDN Website, http://msdn.microsoft.com/en-us/library/aa288474(l=en-us,v=VS.71).aspx.

.

Authors

Saeed Javadzadeh received his BS degree in electrical engineering from Razi university, Kermanshah, Iran in 2009 and he is doing his MSc degree at Razi University in computer engineering. He is a member of the robotic lab of Razi university. His research interest focuses on autonomous robots, hardware design and image processing.

Abdolah Chalechale born in Kermanshah, Iran, received his BS and MSc degrees in Electrical Eng (Hardware) and Computer Eng. (Software) from Sharif University of Technology, Tehran, Iran, respectively. He received his PhD degree from Wollongong University, NSW, Australia in 2005 and currently is with Razi University, Kermanshah, Iran. His research interests include image processing, machine vision and human-machine interactions.

Ehsan Zadkhosh born in Mashhad , Iran, in 1985. He received his BS degree in computer engineering from Azad university, Iran in 2008 and he is doing his MSc degree at Razi University.His research interest includes SoC, NoC, network, performance modeling and fault tolerance of artificial neural networks.