Ball Distance Estimation and Tracking System of Humanoid Soccer Robot

Widodo Budiharto, Bayu Kanigoro, Viska Noviantri

Ball Distance Estimation and Tracking System of Humanoid Soccer Robot

Widodo Budiharto, Bayu Kanigoro, and Viska Noviantri

School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
{wbudiharto,bkanigoro}@binus.edu,viskanoviantri@yahoo.com

Abstract. Modern Humanoid Soccer Robots in uncontrolled environments need to be based on vision and versatile. This paper propose a method for object measurement and ball tracking method using Kalman Filter for Humanoid Soccer, because the ability to accurately track a ball is one of the important features for processing high-definition image. A color-based object detection is used for detecting a ball while PID controller is used for controlling pan tilt camera system. We also modify the robots controller CM-510 in order able to communicate efficiently using main controller. The proposed method is able to determine and estimate the position of a ball and kick the ball correctly with the success percentage greater than 90%. We evaluate and present the performance of the system.

1 Introduction

The humanoid soccer robots are popular nowadays for the entertainment or contests such as RoboCup Humanoid League. The important features of humanoid soccer, such as accuracy, robustness, efficient determination and tracking of ball size and location; has proven to be a challenging subset of this task and the focus of much research. With the evolution of robotics hardware and subsequent advances in processor performance in recent years, the temporal and spatial complexity of feature extraction algorithms to solve this task has grown[1].

In the case of Humanoid soccer, vision systems are one of the main sources for environment interpretation. Many problems have to be solved before having a fully featured soccer player. First of all, the robot has to get information from the environment, mainly using the camera. It must detect the ball, goals, lines and the other robots. Having this information, the robot has to self-localize and decide the next action: move, kick, search another object, etc. The robot must perform all these tasks very fast in order to be reactive enough to be competitive in a soccer match. It makes no sense within this environment to have a good localization method if that takes several seconds to compute the robot position or to decide the next movement in few seconds based on the old perceptions[2]. At the same time many other topics like human-machine interaction, robot cooperation and mission and behavior control give humanoid robot soccer a higher level of complexity like no any other robots[3]. So the high speed processor with efficient algorithms is needed for this issue.

One of the performance factors of a humanoid soccer is that it is highly dependent on its tracking ball and motion ability. The vision module collects information that will be the input for the reasoning module that involves the development of behaviour control. Complexity of humanoid soccer makes necessary playing with the development of complex behaviours, for example situations of coordination or differ rent role assignment during the match. There are many types of behaviour control, each with advantages and disadvantages: reactive control is the simplest way to make the robot play, but do not permit more elaborated strategies as explained for example in [4]. On the other side, behaviour-based control are more complex but more difficult to implement, and enables in general the possibility high-level behaviour control, useful for showing very good performances. Intelligent tracking algorithm for state estimation using Kalman filter has been successfully developed [5], and we want to implement that method for ball tracking for humanoid soccer robot.

In this paper we propose architecture of low cost humanoid soccer robot compared with the well known humanoid robots for education such as DarwIn-OP[1] and NAO humanoid robot[6] and test its ability for image processing to measure distance of the ball and track a ball using color-based object detection method, the robot will kick the ball after getting the nearest position between the robot and the ball. The Kalman filter is used here to estimate state variable of a ball that is excited by random disturbances and measurement noise. It has good results in practice due to optimality and structure and convenient form for online real time processing.

2 Proposed System

Humanoid soccer robots design based on the vision involves the need to obtain a mechanical structure with a human appearance, in order to operate into a human real world. Another important feature for modern humanoid robot is the ability to process tasks especially for computer vision. We propose an embedded system that able to handle high speed image processing, so we use main controller based on the ARM7 Processor. Webcam and servo controller are used to track a ball, and the output of the main controller will communicate with the CM510 controller to control the actuators and sensors of the robot as shown in Fig 2.

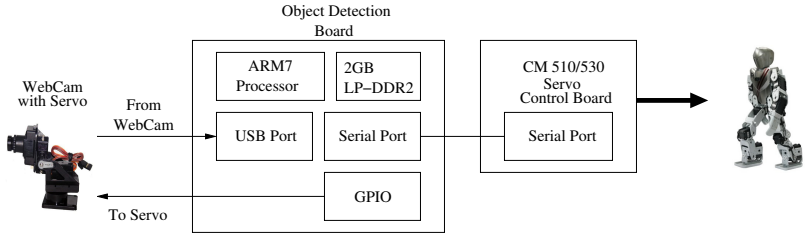


Fig. 1: The architecture of Object Avoiding system

The main controller uses Odroid X2[7] that consist of Cortext-A9 1.7 GHz and sufficient memory and ports to be connected with other devices. The specification of the Odroid X2 is shown in table 1,

Table 1: Odroid X2 Specification

Type	Description
Processor	Exynos4412 Quad-core ARM Cortex-A9 1.7GHz
Memory Capacity	2 GBytes
I/O	6× High Speed USB2.0 Host Port
Network	10/100Mbps Ethernet with RJ-45 LAN Jack

The Firmware of the robot to control the servos is modified from the original one named Robotis Firmware due to the limitation for sending a motion command by serial interface based on Peter Lanius works published in google code[8]. This firmware instead using RoboPlus Task[9] to program the robot controlling its movement but it directly program the AVR Microcontroller inside the CM-510[10] controller using C language. Using this alternative can reduce the size of the program from originally 170KB to 70KB in the memory. By this firmware, the robot can be connected directly to Ball Tracking System using USB Serial Interface to command its motion. Based on this framework, it opens an opportunity to built Real Time Operating System for the robot. The robots control starts with initialization routines of CM-510 controller then move to Wait for Start Button state. In this state, it waits the button to be pressed to change the start_button_pressed variable from FALSE to TRUE then move to Dynamixel servos[11] and Gyro Initialization which send broadcast ping to every Dynamixel servos connected to CM-510. When one or more servos do not respond of the ping then CM-510 will send a message mentioning the failure of a servo to serial terminal. Gyro Initialization does gyro calibration in the robot to get center reference and sends the value to serial terminal. Next state is Waiting Motion Command that waits the command through serial interface, from terminal or tracking module, then check if the command is valid or not. If it does not valid then the state will repeat to Wait Motion Command or continue to next Execute Motion Command state when the command is valid. Execute Motion Command executes a motion command to move a servos based on defined Look-Up-Table (LUT).

For example, when a command says WALKING then the state looks servos values for WALKING stored in the LUT then send it to Dynamixel servo through serial bus. When a motion is completed then it move to preceding state but if there is an emergency which is determined by pressing start button when the servos is moving compared to command input which does not receive stop command, then it move to Dynamixel Torque Disable to disable all the servos torque to save from damage and move to Wait for Start Button state. The

improved system to accept commands from the main controller is shown as the state machine in fig. 2.

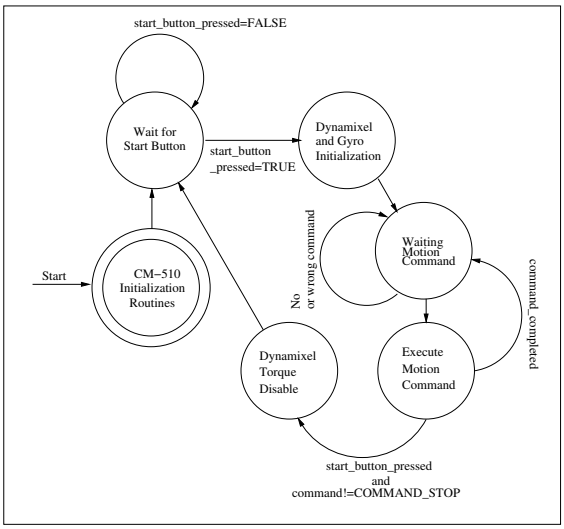


Fig. 2: State machine of the robot's controller

Computer vision is one of the most challenging applications in sensor systems since the signal is complex from spatial and logical point of view. An active camera tracking system for humanoid robot soccer tracks an object of interest (ball) automatically with a pan-tilt camera. We use OpenCV for converting to HSV (Hue Saturation-Value), extract Hue and Saturation and create a mask matching only the selected range of hue value.

To have a good estimation, the object must be in the centre of the image, i.e. it must be tracked. Once there, the distance and orientation are calculated, according to the necks origin position, the current neck's servomotors position and the position of the camera in respect to the origin resulting of the design [12]. We considered method for distance estimation of the ball by centering the ball on the camera image, using the head tilt angle to estimate the distance to the ball.

Region growing algorithms are also used to locate the ball color blobs that have been identified by region growing and are useful and robust source for further image processing, as demonstrated by [13]. The ball will be tracked based on the color and webcam will track to adjust the position of the ball to the center of the screen based on the Algorithm 1.

The Kalman Filter is a state estimator which produces an optimal estimate in the sense that the mean value of the sum (actually of any linear combination) of the estimation errors gets a minimal value. In other words, The Kalman Filter gives the following sum of squared errors:

Algorithm 1 Ball Tracking and Kick the ball

Require: Object to be tracked
1: Get input image from the camera.
2: Convert to HSV (Hue-Saturation-Value)
3: Extract Hue & Saturation
4: Create a mask matching only for the selected range of hue
5: Create a mask matching only for the selected saturation levels
6: Find the position (moment) of the selected regions
7: **if** ball detected **then**
8: Object tracking using Kalman Filter
9: Centering the position of the ball
10: Move robot to the ball
11: **if** ball at the nearest position with the robot **then**
12: Kick the ball
13: **end if**
14: **end if**

$$E[e_x^T(k)e_x(k)] = E[e_{x_1}^2(k) + \dots e_{x_n}^2(k)] \tag{1}$$

a minimal value. Here

$$e_x(k) = e_{est}(x) - x(k) \tag{2}$$

is the estimation error vector.

By assuming discrete-time state space model as system model of Kalman Filter then,

$$x(k+1) = f[x(k), u(k)] + Gw(k) \tag{3}$$

where x is the state vector of n state variables, u is the input vector of m input variables, f is the system vector function, w is random noise vector, G is the process noise gain matrix relating the process noise to the state variables. It is common to assume that $q = n$, making G square. In addition it is common to set the elements of G equal to one. Assuming that non-linear then the measurement model is,

$$y(k) = g[x(k), u(k)] + Hw(k) + v(k) \tag{4}$$

where y is the measurement vector of r measurement variables, g is the measurement vector function, H is a gain matrix relating the disturbances directly to the measurements. It is however common to assume that H is a zero matrix of dimension $(r \times q)$,

$$H = \begin{bmatrix} 0\,0 & 0 & 0 \\ 0\,0 & \ddots & \vdots \\ 0\,0 & \cdots & H_{rq} \end{bmatrix} \tag{5}$$

and v is a random (white) measurement noise vector.

The calculation of Kalman Filter is to be done by following these steps:

1. Calculate the initial state estimation x_p,

$$x_p(0) = x_{init} \tag{6}$$

   where x_{init} is the initial guess of the state.
2. Calculate the predicted measurement estimate y_p from the predicted state estimation:

$$y_p(k) = g[x_p(k)] \tag{7}$$

3. Calculate innovation variable as the difference between the measurement $y(k)$ and the predicted measurement $y_p(k)$:

$$e(k) = y(k) - y_p(k) \tag{8}$$

4. Calculate corrected state estimate x_c:

$$x_c(k) = x_p(k) + Ke(k) \tag{9}$$

   where K is the Kalman Filter gain
5. Calculate the predicted state estimate for the next time step, $x_p(k+1)$:

$$x_p(k+1) = f[x_c(k), u(k)] \tag{10}$$

The estimated position (x, y) from Kalman Filter is used as an input to PID controller. PID controller calculates an error value as the difference between a measured (input) and a desired set point to control high speed HS-85[14] servos. The controller attempts to minimize the error by adjusting (an Output). The generic model of PID Controller shown in figure 3.

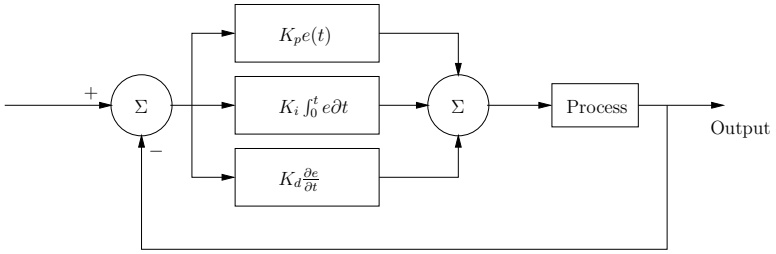


Fig. 3: A Generic PID Controller

The output of a PID controller, equal to the control input to the system, in the time-domain is as follows:

$$u_c(t) = K_p e(t) + K_i \int_0^t e\partial t + K_d \frac{\partial e}{\partial t} \tag{11}$$

Measuring distance between the robot and the ball can be accomplished by using trigonometric function. Assume h_{robot} is a height of the robot, then d_{ball} which is a distance between the robot and the ball can be approximately measured by,

$$\tan \alpha = \frac{d_{ball}}{h_{robot}} \tag{12}$$

$$d_{ball} = \tan \alpha \times h_{robot} \tag{13}$$

The angle α shown in equation 12 and 13 is the angle between robot's body and camera which is placed on the robot. The best angle of the camera is around $15^\circ - 20^\circ$ when the robot will kick the ball.

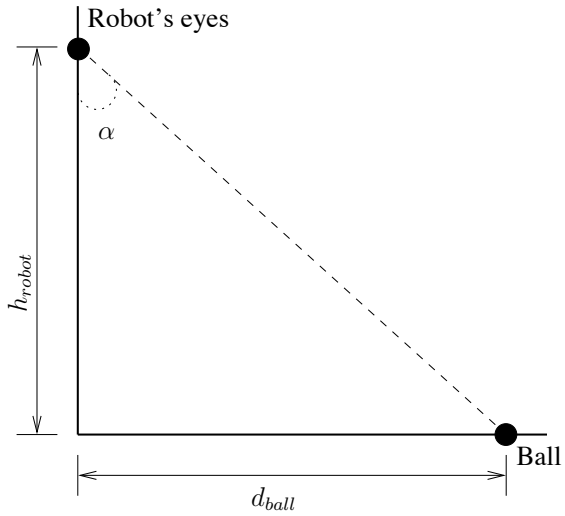
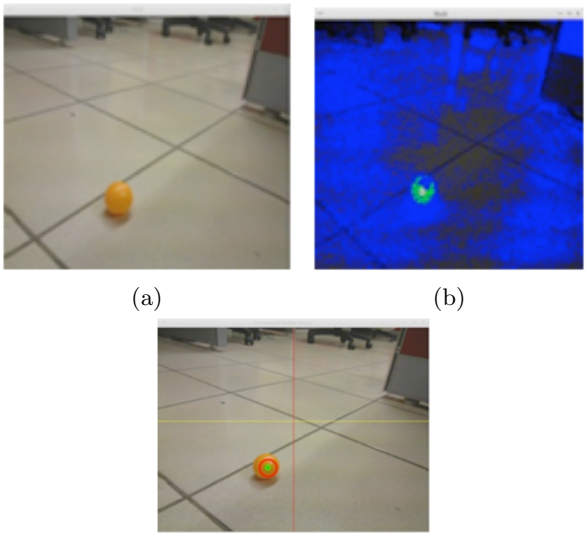


Fig. 4: Measuring distance between the ball and the robot

3 Experimental Result

The approach proposed in this paper was implemented and tested on a humanoid Robot named Humanoid Robot Soccer Ver 2.0 based on Bioloid Premium Robot. By modify the robots controller (CM-510) in order to accept serial command from the main controller, this system able to communicate efficiently. Detecting

several colors means creating several binary image maps, as shown in figure 5. The tracking system is able to track a ball with the maximum speed of 6 cm/s. The ball will be kicked by the robot when it be detected, tracked, and determined if nearest to the robot by using equation 12 and 13. The result of estimation



(a) (b)

(c)

Fig. 5: The original image (5a), the mask (5b) and ball detected and tracked using Kalman Filters in the green circle (5c).

of position of ball using Kalman filter is shown in figure 6, it shows that the estimated point able to follow and estimate the position of the ball.

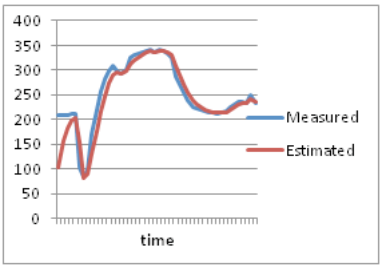


Fig. 6: True measurement versus estimation using Kalman Filter

4 Conclusion

In this paper, we introduced the hardware architecture implemented on our humanoid robot soccer. They are based on Odroid X2 that has powerful ability

for high speed image processing. We propose the simple way to estimate distance and track a ball based on the color, and kick the ball after getting the nearest position of the robot from the ball. The Kalman filter is a robust method to track a ball in the real situation. For future work, we want to use Extended Kalman Filter and shape-based object tracking and defining intelligent behavior for the humanoid robot soccer.

References

1. Ha, I., Tamura, Y., Asama, H., Han, J., Hong, D.W.: Development of open humanoid platform darwin-op. In: SICE Annual Conference (SICE), 2011 Proceedings of, IEEE (2011) 2178–2181
2. Martín, F., Aguero, C., Cañas, J.M., Perdices, E.: Humanoid soccer player design. (2010)
3. Blanes, F.: Embedded distributed vision system for humanoid soccer robot. Journal of Physical Agents **5**(1) (2011) 55–62
4. Behnke, S., Rojas, R.: A hierarchy of reactive behaviors handles complexity. In: Balancing reactivity and social deliberation in multi-agent systems. Springer (2001) 125–136
5. Noh, S., Park, J., Joo, Y.: Intelligent tracking algorithm for manoeuvering target using kalman filter with fuzzy gain. Radar, Sonar & Navigation, IET **1**(3) (2007) 241–247
6. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., Maisonnier, B.: Mechatronic design of nao humanoid. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE (2009) 769–774
7. Hardkernel: Odroid X2. `www.hardkernel.com/main/products/prdt_info.php?g_code=G135235611947` Accessed: 2013-09-30.
8. Lanius, P.: Bioloidccontrol. `http://code.google.com/p/bioloidccontrol/` Accessed: 2013-09-30.
9. Robotis: RoboPlus. `http://support.robotis.com/en/` Accessed: 2013-09-30.
10. Robotis: CM-510 controller. `http://support.robotis.com/en/` Accessed: 2013-09-30.
11. Robotis: Dynamixel AX-12A robot actuator. `http://www.robotis.com/xe/dynamixel_en` Accessed: 2013-09-30.
12. Maggi, A., Guseo, T., Wegher, F., Pagello, E., Menegatti, E.: A light software architecture for a humanoid soccer robot. In: Workshop on Humanoid Soccer Robots of the IEEE-RAS International Conference on Humanoid Robots (Humanoids' 06), Genoa, Italy. (2006)
13. Ghanai, M., Chafaa, K.: Kalman filter in control and modeling (2010)
14. Hitec: HS-85BB Premium Micro Servo. `http://hitecrcd.com/products/servos/micro-and-mini-servos/analog-micro-and-mini-servos/hs-85bb-premium-micro-servo/product` Accessed: 2013-09-30.