Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

FPGA-Based Fuzzy PK Controller and Image Processing System for Small-Sized Humanoid Robot

Yu-Te Su, Chun-Yang Hu
aiRobots Lab., Department of Electrical Eng.
National Cheng Kung University
Tainan 70101, Taiwan, R.O.C
N2895109@mail.ncku.edu.tw

Tzuu-Hseng S. Li
aiRobots Lab., Department of Electrical Eng.
National Cheng Kung University
Tainan 70101, Taiwan, R.O.C
thsli@mail.ncku.edu.tw

***Abstract*—This paper mainly covers the development of a FPGA-based fuzzy controller and image processing system for a small-sized humanoid robot. All the computations are operated on an FPGA board including the real-time image processing and the fuzzy logic controller design for PK event in FIRA RoboWorld cup. At first, the specification of the hardware is introduced. The image processing is then employed for target recognition. The control strategy system for PK event is also developed. Finally the experiments are demonstrated to verify feasibility of the proposed control system.**

I. INTRODUCTION

The PK is one of the most typical events in HuroCup of the FIRA[1]. From the rules of HuroSot league of FIRA, the dimensions of the playing field are at least 220cm by 180 cm. A figure of one possible legal playing field is shown in Fig. 1. The goal is 100cm wide and is placed on the goal side of the playing area with its center along the center line of the playing field. The penalty mark is 75cm away from the goal line. The penalty area is specified by the triangle that extends from the penalty mark to the top left and top right corner points of the goal area. For the task and function consideration, we combines control strategy system and image processing system in one FPGA embedded board. Specifications and features of the constructed humanoid robot are summarized in Table1.

In this paper, the hardware of humanoid robot is discussed firstly. Next, proposed Fuzzy PK controller and image processing system for the humanoid robots are described. Finally, the experiments that have been realized by implementing the proposed method onto our humanoid robots are reported.

II. HARDWARE OF HUMANOID ROBOT

A. Architecture of visual system

As shown in Fig. 2, the vision system of robot is composed by two small servo motors and one CMOS-EYE image sensor. The mainly objective of using two servo motors which can make the head move to any direction is tracking the object for the strategies. The CMOS-EYE image sensor module manufactured by ASAKUSAGIKEN company in Japan. The module is shown in the Fig. 3. The specification of the vision sensor is shown in Table 2. The reason we select this device is small and light. If the vision system is too heavy or too large at the head of robot, it will influence the motion terribly. Besides, the image data is delivered between the module and the central process system by parallel bus, so it takes short time to capture an image and send to the central process system.

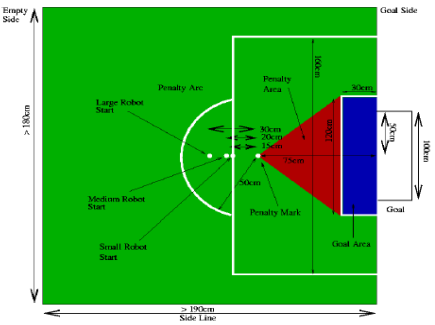


Figure 1. The field of play for penalty kick.

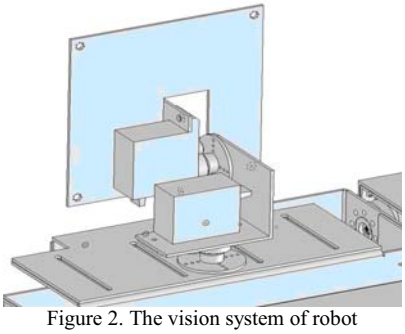


Figure 2. The vision system of robot

TABLE I
THE SPECIFICATION OF OUR ROBOT

Height	50 cm
Width	20 cm
Depth	10 cm
Weight	3 kilogram
Domain of freedom	29 DOFs
Material of structure	A5052 aluminum-magnesium compound metal
Actuator	RC servo motor
Controller	Nios II 1C12
Battery	Li-Po 11.1 voltage

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

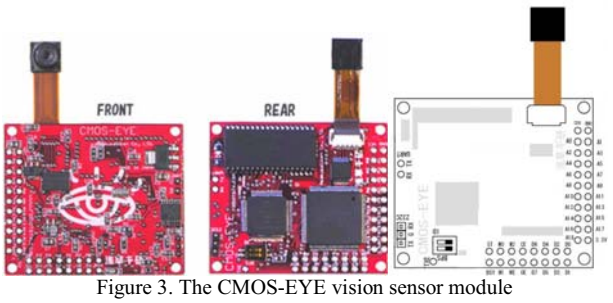


Figure 3. The CMOS-EYE vision sensor module


Figure 4. The ALTERA Nios II EP2C20F484C8 FPGA board.

TABLE 2
THE SPECIFICATION OF THE CMOS-EYE VISION SENSOR

Model	CMOS-EYE
Dimension	39×46×3 mm
Weight	10 g
Resolution	160×120
Voltage	4.8-15 V
Transmission Type	RS232，Parallel Bus

TABLE 3
THE SPECIFICATION OF NIOS II EVALUATION BOARD

LEs		18752
M4K RAM Blocks		52
Embedded Multipliers		26
Total RAM bits		239,616
PLLs		4
Maximum user I/O pins/Available		315/128
Voltage	Internal / IO	1.2V / 3.3V

B. *Central processing unit*

The Central Process Unit is the heart of robot. The ALTERA Nios II EP2C20F324C8 FPGA board as shown in Fig. 4 is used for the central process unit. Table 3 shows the specification of the FPGA board. We design the central process unit module and other peripheral devices such as CMOS image sensor signal controller, sensors and motors. Additionally, the functions of the peripheral devices are written in Verilog HDL. The control strategy and image algorithm are written via the C/C++ language in the ALTERA development platform, the Nios II Integrated Development Environment (IDE). Therefore, the functions of the robot can be operated by downloading the software into the Nios II evaluation board through the JTAG debug interface [2-3].

III. IMAGE PROCESSING SYSTEM

In this section, the target recognition for tracking is introduced. We firstly discuss the digital image processing [4] with our vision system in this section. The effective edge detection method, Laplacian edge detection, is adopted to extract the contours and feature points from the captured colored image.Decoding method. The RCD (Randomized Circle Detection) method is utilized to searching the circle.

C. *Image capture*

CMOS-EYE image sensor module provides two methods for users to control signals and deliver the image data. One is the UART interface, and the other is direct signal control interface. We can send commands to the CMOS-EYE and use the function embedded in the module to process the image and then deliver the image data all through the UART. However, it takes about five seconds to deliver a 160×120 24bits color image. The speed of the transportation is too slow. Thus, we choose another method that is to control the I/O pins, the SRAM (Static Random Access Memory) address and the 8 bits data parallel bus directly to get the image data. The clock of the module runs at 50MHz on the FPGA. The speed which only needs about 0.5 second is increased a lot than before. According to the timing chart in the data sheet of the CMOS-EYE module, we write the Verilog HDL to control the signal sequences of memory addresses and the data bus on the module. The timing between the data transmitted is very important. If we send the signal at the wrong timing, the data may not receive correctly. The data format which stored in the SRAM is the Bayer RGB and the address for 160×120 color image for each 19200 bytes are the red, green, and blue pixels, respectively.

D. *Noise Reduction*

Because the length of the transport wire is not short enough, we get the image data with noises. The noises cause spots in the vision scene. We need to reduce noises before processing the image data. We deal noises with the MLM (Multi-Level Median) method in smoothing filters. The median method is to sort one sequence from small to large and to choose the value in the median position of the sorted sequence. We have to calculate every pixel value three times with the MLM method because the pixel is composed of three original colors which are red, green, and blue. There are large variations between the pixel value of the noise and the values of the neighbor pixels. With this characteristic, we can filter out noises by substituting the median value for the noise pixel. Fig. 5 is a 3×3 structuring element which defines the operational range. Assuming M_1 is the median value of x_4 , x_5 , and x_6 shown as $M_1 = Med\{x_4, x_5, x_6\}$, $M_2 = Med\{x_1, x_5, x_9\}$,

$M_3 = Med\{x_2, x_5, x_8\}$, $M_4 = Med\{x_3, x_5, x_7\}$. Where x_i is the pixel in the structuring element. Thus, the result Y of the MLM method for the nine gray pixel values in the 3×3 structuring element is $Y = Med\{M_{\min}, M_{\max}, x_5\}$, where $M_{\min} = Min\{M_1, M_2, M_3, M_4\}$, $M_{\max} = Max\{M_1, M_2, M_3, M_4\}$. The noises decrease obviously after the image processing with this algorithm which as shown in Fig. 6

E. Color Detection

As regards identifying the object, we need to know the features of the object. The color of the object is one of the features. When the image is sent to the human-machine interface, we utilize the mouse to click the object. We save the RGB color ranges of the clicked pixel which can be more accurately adjusted by the interface and the interface shows the area in the RGB color ranges with light green.

F. Edge Detection

The color is only one of the features of the object. In order to achieve more accurate, we have to perform the edge detection before identification. Many methods for edge detection have been proposed in the literature [5] for the past decades. One of them is the Laplacian edge filter which is a well known second order operator. However, the disadvantage is too sensitive. In order to against the drawback, we combine the virtue of the Gaussian filter and the Laplacian filter. We utilize the LoG (Laplacian of Gaussian) method which is anti-noise and effective for the edge detection refined from the Laplacian edge filter [6]. The LoG method is to calculate the change of the gradient for an image by coefficient convolution. Before we practice this method, we need to convert the captured color image into the gray level image.

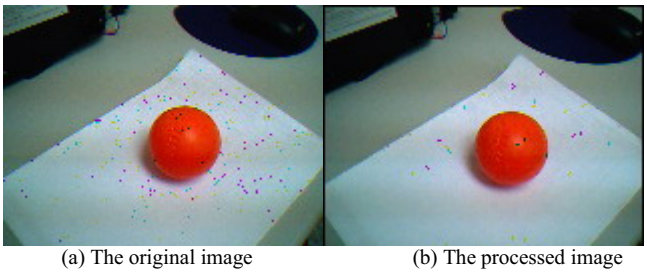
There are two common methods for gray level transformation . One is to take the mean of the RGB values directly and the other is to combine these three values according to different brightness. Considering the speed of the Nios II, we choose the former for the gray level transformation approach. Hence, the gray level value is computed as gray level value = $(R+G+B)/3$. And the result of the gray level process is shown in Fig. 7. The LoG equation is shown as

$$\nabla^2 h(r) = -\left[\frac{r^2-\sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

where $r^2 = x^2 + y^2$, (x, y) is the coordinate of the pixel and σ is the standard variation that determine the blurred level. Fig. 8 shows the diagram of LoG three dimensions curve which σ is 2. Stand on this shape, we choose a 5×5 operator mask which approximate to $\nabla^2 h$ as shown in Fig. 9. Fig. 10 is the LoG structuring element, and the gradient of the pixel Z_{13} called G is computed by the neighbor with 25 pixels according to the Fig. 9. The result of the edge detection with LoG is presented in Fig. 11.

X_1	X_2	X_3
X_4	X_5	X_6
X_7	X_8	X_9

Figure 5. 3×3 structuring element



(a) The original image (b) The processed image

Figure 6. The results of the MLM algorithm

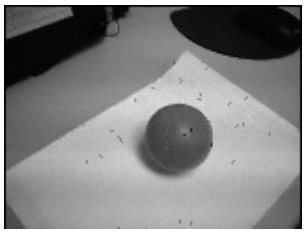


Figure 7. The result of gray level

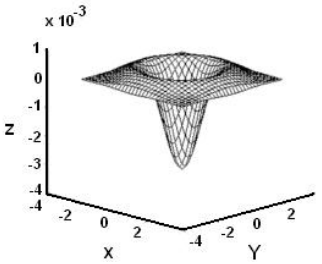


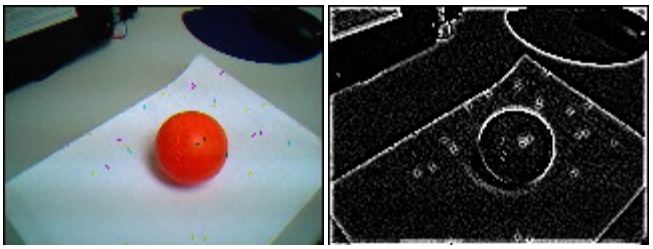
Figure 8. LoG three dimensions curve

0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Fig 9. LoG operator mask

Z_1	Z_2	Z_3	Z_4	Z_5
Z_6	Z_7	Z_8	Z_9	Z_{10}
Z_{11}	Z_{12}	Z_{13}	Z_{14}	Z_{15}
Z_{16}	Z_{17}	Z_{18}	Z_{19}	Z_{20}
Z_{21}	Z_{22}	Z_{23}	Z_{24}	Z_{25}

Fig 10. LoG structuring element



(a) The original image (b) The result of LoG

Fig 11. The results of the LoG edge detection

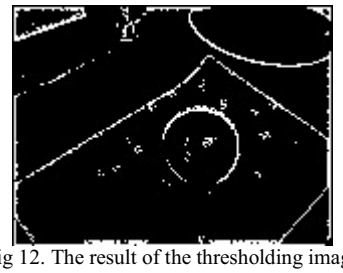


Fig 12. The result of the thresholding image

(a) The original test image (b) The result of the RCD method

Figure 13. The results of the RCD method

G. Thresholding

There are 256 levels brightness in a gray level image. To strengthen the edge feature, we need to convert the gray level image to the binary image. Considering the computation, we simplify the process by setting a fixed value for the threshold T_h shown as

$$T_p = \begin{cases} 1 & , \ pixel \geq T_h \\ 0 & , \ pixel < T_h \end{cases}$$

And the result of the thresholding is shown in Fig 12.

H. Circle Detection with RCD Method

The purpose of this algorithm is to detect the ball in the field of the penalty kick event. In general, the Hough transform method is used to detect the shape of circle in an image. Nevertheless, the time complexity of the Hough transform method takes too much, so we choose another detective method. The method we choose is the RCD (Randomized Circle Detection) [7]. This method can take better accuracy and detection speed into account than previous. The result of this algorithm is shown in Fig. 13. The circle found by the RCD method is drawn in the gray color.

IV. FUZZY PK CONTROLLER

This section describes the tracking strategy for PK event in FIRA HuroCup. The PK event which the kicker needs to kick the ball into the goal guarded by the defender is like the soccer game in the real world. Firstly, robot searches where the ball is. When the ball is found, robot moves and tries to find the best shooting range. Finally, robot shoots the ball into the goal. This event shows that robot has artificial intelligence. Moreover, the fuzzy logic controller for the tracking motors is designed and each component of the controller is introduced.

A. Fuzzy Logic Controller

After getting the target position (x, y), we have to transform this coordinate into the normalized position (x_n, y_n) which is in the range of [-1 1]. This procedure is called the normalization for the FLC input. We calculate with (1) and (2) to normalize the difference between the target and the center of the image plane for the FLC input. Note that the coordinate of the most left-top and the most right-bottom points are (0,0) and (159,119), respectively.

$$x_n = \begin{cases} \dfrac{x-80}{79} & , \ x \geq 80 \\ -\dfrac{80-x}{80} & , \ x < 80 \end{cases} \quad (1)$$

$$y_n = \begin{cases} \dfrac{y-60}{59} & , \ y \geq 60 \\ -\dfrac{60-y}{60} & , \ y < 60 \end{cases} \quad (2)$$

The $(\dot{x}_n, \dot{y}_n)$ is defined as

$$\dot{x}_n = x_n - \tilde{x}_n \quad (3)$$

$$\dot{y}_n = y_n - \tilde{y}_n \quad (4)$$

where $\tilde{x}_n$ and $\tilde{y}_n$ represent the previous sample value of x_n and y_n, respectively. The θ_x is the output horizontal angle for the pan motor and the θ_y is the output vertical angel for the tilt motor. The input of the FLC including (x_n, y_n) and $(\dot{x}_n, \dot{y}_n)$ are decomposed into five fuzzy term sets which are NB,NS,ZR,PS and PB. The partitions and the shapes of the membership functions for the input and output are shown in Fig. 15. The fuzzy table is shown in Table 4.

The defuzzification strategy we use in our FLC is the combination of COA (Center of Area) and MOM (Mean of Maximum) methods shown as

$$\theta_x = \frac{\sum_{i=1}^{n} W_{\theta_{x_i}} \times \left(\mu_{x_i} \times \mu_{\dot{x}_i}\right)}{\sum_{i=1}^{n} \left(\mu_{x_i} \times \mu_{\dot{x}_i}\right)} \quad (5)$$

$$\theta_y = \frac{\sum_{i=1}^{n} W_{\theta_{y_i}} \times \left(\mu_{y_i} \times \mu_{\dot{y}_i}\right)}{\sum_{i=1}^{n} \left(\mu_{y_i} \times \mu_{\dot{y}_i}\right)} \quad (6)$$

where n is the number of fuzzy sets for the control input. $W_{\theta_{x_i}}, W_{\theta_{y_i}}$ are the weighting of each fuzzy set i. $\mu_{x_i}, \mu_{\dot{x}_i}, \mu_{y_i}, \mu_{\dot{y}_i}$ are the membership function values of the control input. Nevertheless, the RC servo motors are

controlled by the PWM (Pulse Width Modulation) value. We need to convert the crisp output of the fuzzy which is the angular position to the PWM value. By multiplying a weighting β_x (β_y), the angle can be converted to the corresponding PWM value as

$$PWM_{x\_error} = \theta_x \times \beta_x \quad (7)$$

$$PWM_{y\_error} = \theta_y \times \beta_y \quad (8)$$

Therefore, the PWM value of each motor can be obtained by the fundamental PWM value and the PWM correction is described as

$$PWM_x = PWM_{x\_current} + PWM_{x\_error} \quad (9)$$

$$PWM_y = PWM_{y\_current} + PWM_{y\_error} \quad (10)$$

where $PWM_{x\_current}$ and $PWM_{y\_current}$ represent the current PWM values of pan and tilt motors in horizontal and vertical directions, respectively.

B. *The Strategy for Offense*

When the ball is tracked to the middle of the image, we are going to figure out the distance and the orientation between the ball and robot. Because there is only one eye on robot, the depth calculation of the stereo vision can not be utilized here. Thus, we combine the two motors on the head and the one eye to figure out by geometric mathematics. We can neglect the distance between the motor and the image sensor because the image sensor is flat enough that we can consider the motor and the sensor are in the same plane. The analysis for the tilt motor is illustrated in Fig. 16. The distance between the ball and robot can be calculated by the trigonometric function. Performing some simple mathematical calculations, the distance between robot and the surface of the target denoted s may be estimated as

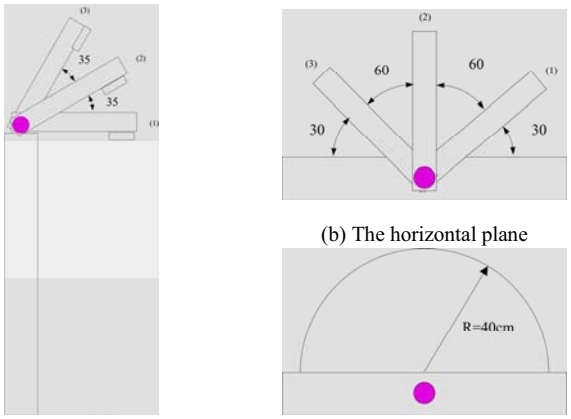
$$s = L\cos\theta_y + (L\sin\theta_y + H)\tan\theta_y \quad (7)$$

where θ_y denotes the rotation angle of the tilt motor, L denotes the length of the head and H denotes the height of the body. The analysis for the pan motor is also illustrated in Fig. 17. The orientation of the ball can be calculated by the pan motor. The rotation angle θ_x can be obtained from the output of the FLC. After figuring out the information of the ball, robot starts to approach the ball according to the distance and the orientation. robot checks if the ball is near the tiptoe. When the distance between the ball and robot is short enough, robot rotates back to face the goal. Then, robot determines the best orientation for shooting the ball. robot analyzes the position of the goal and the goalkeeper and calculate the distance as shown in Fig. 18. d_L and d_R are the distance estimated from the corner of the goal to the edge of the goalkeeper.. In the middle case, the corner of the goal is absent to robot, so d_L and d_R are estimated from the edge of the scene to the edge of the goalkeeper. The best orientation for shooting is to face the largest area of the goal. If d_L is too small as in the case 1, robot turns right an appropriate angle for shooting. Similarly does the case 3. The case 2 is a special case that the corner is absent to robot. However, if d_L is smaller than d_R, robot turns right a small angle, otherwise, robot turns left a small angle.

After robot fixes the orientation, the ball may not close the right foot which is for shooting. Therefore, robot tracks the ball again. By shifting left or shifting right and walking forward, the ball is located at the right foot.

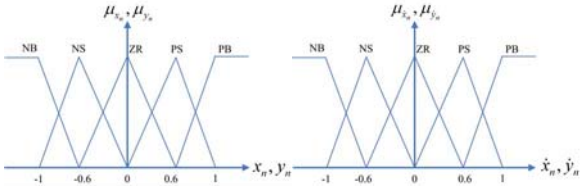
TABLE 4 THE FUZZY RULE TABLE

$\dot{x}_m(\dot{y}_m)$ \ $x_m(y_m)$	NB	NS	ZR	PS	PB
NB	NB	NB	NB	NS	ZR
NS	NB	NS	NS	ZR	PS
ZR	NB	NS	ZR	PS	PB
PS	NS	ZR	PS	PS	PB
PB	ZR	PS	PB	PB	PB



(a) The vertical plane (b) The horizontal plane (c) The field of view

Figure 14. The searching directions



(a) The inputs (x_n, y_n) (b) The inputs $(\dot{x}_n, \dot{y}_n)$

(c) The membership functions of outputs (θ_x, θ_y)

Figure 15. The membership functions of the input and output

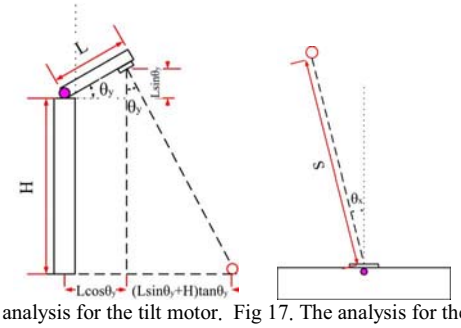
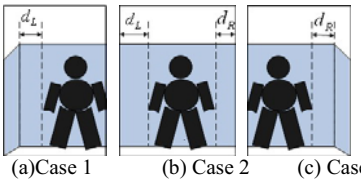


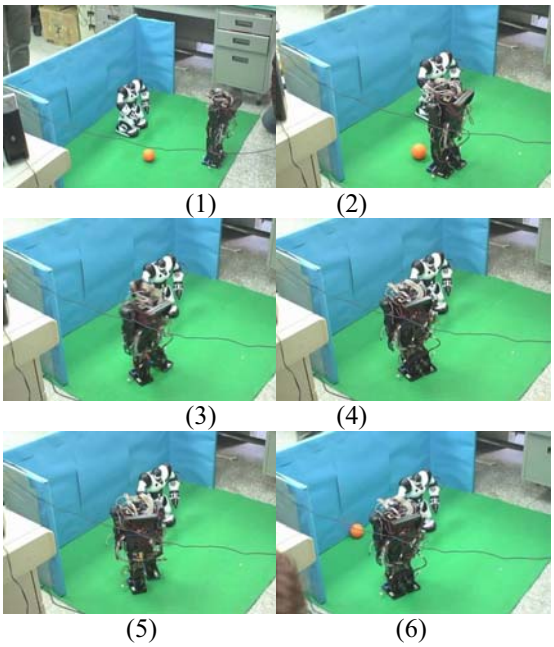
Fig 16 The analysis for the tilt motor.

Fig 17. The analysis for the pan motor



(a)Case 1 (b) Case 2 (c) Case 3

Figure 18. The analysis for the goal and the goalkeeper



(1) (2)

(3) (4)

(5) (6)

Figure 19. The results of the strategy of the PK

Finally, the ball is near the right tiptoe and is kicked by the right foot of robot. In ideal circumstances, robot has a strong probability to accomplish the offense mode successfully in the PK event through above steps.

V. EXPERIMENT

The environment of the PK event is based on the PK rule of FIRA. We placed a goalkeeper to be the defender. The ball is the orange plastic ball. The color of the door is light blue. The following images which are shown in Fig. 19. Robot firstly scans the environment for searching the orange ball from left to right. Then, aiRobot-2 has tracked and found the position of the ball and rotates left and starts to walk straightly to approach the ball. Robot will stop walking in front of the ball and check if the ball is near the tiptoe. Next, robot-2 is going to analyze the goal situation by raising his head. After that, aiRobot-2 rotates right to face the best shooting orientation. Finally, aiRobot-2 kicks the ball and the ball is shot into the goal. Text heads organize the topics on a relational, hierarchical basis.

VI. CONCLUSION

The development of a FPGA-based fuzzy PK controller and image processing system for a small-sized humanoid robot is presented in this paper. All the image processing and control system are operated on one FPGA board without other computation system. Finally, the experiment results show the capability, the efficiency and validity of whole robot system.

ACKNOWLEDGMENT

This work was supported by National Science Council of Taiwan, R.O.C, under Grant NSC99-2221-E006-006-382-MY3.

REFERENCES

[1] FIRA, http://www.fira.net

[2] C.-L. Hsu, et. al., T-H S. Li, "Design and Implementation of an SOPC-Based Small Sized Humanoid Robot," in 2007 CACS Int. Automatic Control Conference, Nov. 2007, Taiwan..

[3] T-H S. Li, J.-J. Hui, et al., "Improvement and Development of Humanoid Soccer Robot NCKU-II," in 2006 FIRA RoboWorld Congress, July, German..

[4] Y.-T. Su, C.-Y. Hu, M. F. Lu, C.-M. Chang, S.-W. Lai, S.-H. Liu, and T-H S. Li, "Design and implementation of SOPC based image and control system for HuroCup," J. of Harbin Institute Tech. (New Series), Vol. 15, pp.41-46, September 2008 China.

[5] X. Wang, "Laplacian operator-based edge detectors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, Issue 5, pp. 886-890, May 2007.

[6] S. C. Pei and J. H. Horng, "Circular arc detection based on Hough transform," *Pattern Recognition Letters*, vol. 16, Issue 6, pp. 615-625, Jun. 1995.

[7] T. C. Chen and K. L. Chung, "An efficient randomized algorithm for detecting circles," *Computer Vision and Image Understanding, Academic Press*, pp. 172-191, 2001.