SICE Annual Conference 2010
August 18-21, 2010, The Grand Hotel, Taipei, Taiwan

# An Efficient Object Recognition and Self-Localization System for Humanoid Soccer Robot

Jen-Shiun Chiang, Chih-Hsien Hsia, Shih-Hung Chang, Wei-Hsuan Chang, Hung-Wei Hsu,
Yi-Che Tai, Chun-Yi Li, and Meng-Hsuan Ho

*Department of Electrical Engineering, Tamkang University, Taipei, Taiwan*

*E-mail: chiang@ee.tku.edu.tw; chhsia@ee.tku.edu.tw; 696450682@s96.tku.edu.tw; mynameisoa@hotmail.com; 697450079@s97.tku.edu.tw; 495440322@s95.tku.edu.tw; 698450417@s98.tku.edu.tw; 695450162@s95.tku.edu.tw*

***Abstract*— In the RoboCup soccer humanoid league competition, the vision system is used to collect various environment information as the terminal data to finish the functions of object recognition, coordinate establishment, robot localization, robot tactic, barrier avoiding, etc. Thus, a real-time object recognition and high accurate self-localization system of the soccer robot becomes the key technology to improve the performance. In this work we proposed an efficient object recognition and self-localization system for the RoboCup soccer humanoid league rules of the 2009 competition. We proposed two methods : *1)* In the object recognition part, the real-time vision-based method is based on the adaptive resolution method (ARM). It can select the most proper resolution for different situations in the competition. ARM can reduce the noises interference and make the object recognition system more robust as well. *2)* In the self-localization part, we proposed a new approach, adaptive vision-based self-localization system (AVBSLS), which uses the trigonometric function to find the coarse location of the robot and further adopts the measuring artificial neural network technique to adjust the humanoid robot position adaptively. The experimental results indicate that the proposed system is not easily affected by the light illumination. The object recognition accuracy rate is more than 93% on average and the average frame rate can reach 32 fps (frame per second). It does not only maintain the higher recognition accuracy rate for the high resolution frames, but also increase the average frame rate for about 11 fps compared to the conventional high resolution approach and the average accuracy ratio of the localization is 92.3%.**

***Keywords- RoboCup; Real-Time; Object Recognition; Adaptive Resolution Method; Self-Localization.***

## I. INTRODUCTION

RoboCup [1] is an international joint project to stimulate researches in the field of artificial intelligence, robotics, and related fields. According the rules of RoboCup for 2009 in the humanoid league of kid size [2], the competitions take place on a rectangular field of 600×400 cm$^2$ area, which contains two goals and two landmark poles, as shown in Fig. 1. The objects of the goals and landmark poles are the most critical characteristics in the field, and they are also the key features which we have to pay attention to.

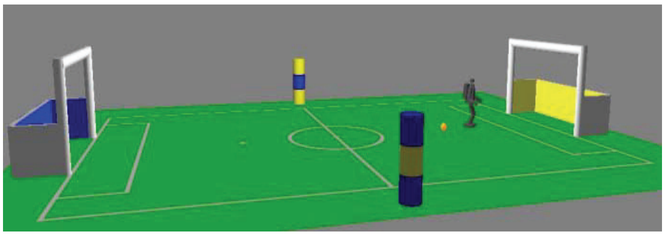


Fig. 1: The field of the competitions [2].

Generally speaking, *1)* object recognition uses object features to extract the object out of the picture frame, and thus color [3]-[4], shape [5]-[6], contour [7]-[9], texture, and sizes of object features are commonly used. Because the object color is distinctive in the contest field, we mainly choose the color information to determine the critical objects. Although this approach is simple, the real-time efficiency is still low. Because there is much information to be processed in every frame for real-time consideration, Sugandi *et al.* [10] proposed a low resolution method to reduce the information. It can speed up the processing time, however the low resolution results in a shorter recognizable distance and it may increase the false recognition rate. In order to improve the mentioned drawbacks, we propose a new approach, adaptive resolution method (ARM), to reduce the computation complexity and increase the accuracy rate. *2)* Self-localization is one of the most important issues of a humanoid soccer robot. In this work we use the landmarks assigned for 2009 RoboCup soccer field for humanoid kid-size soccer robot [2] as the location reference. The humanoid soccer robot has a single camera vision system with pan/tilt motors on the head. As soon as the humanoid soccer robot detects one of the landmarks, the self-localization system starts to localize itself. During the self-localization process, the robot has to measure the distance between the landmark and the robot itself, and the IBDMS (image-based distance measuring system) [11]-[12] technique can be applied to measure the distance. However, there exist distortions in the CCD camera and we need to adjust the coarse data to find a more accurate distance. Based on these concepts, we propose a self-localization system, adaptive vision-based self-localization system (AVBSLS), for the humanoid soccer robot for the 2009 RoboCup contest.

The rest of this paper is organized as follows. Section 2 describes the proposed approaches, ARM and AVBSLS. The experimental results are shown in Section 3. Finally, the conclusions and future works are outlined in Section 4.

## II. The Proposed Method

The proposed object recognition and self-localization system for the humanoid robot consists of the following blocks: image input procedure, coordinate establishment procedure, object recognition procedure, self-localization procedure, and coordinate output procedure. The procedure flow chart is shown in Fig. 2.

The information of the soccer field is displayed by a 2-D manner to facilitate the description of the soccer field. In the detection of the object recognition, it converts the RGB 24-bit color images captured from the camera into the HSV color model information, and uses the adaptive resolution method (ARM) to select one threshold value to find the binary image. It removes noise in the image through the image post-processing to increase the accuracy and completeness of the object recognition. Then, based on the feature of binary image, it can determine whether the condition is in the target site or not. If it is, it starts to estimate the distance between the humanoid robot and landmark. Finally, the estimated information will be displayed by the 2-D coordinate manner and get the location coordinates of the humanoid robot in the contest field.

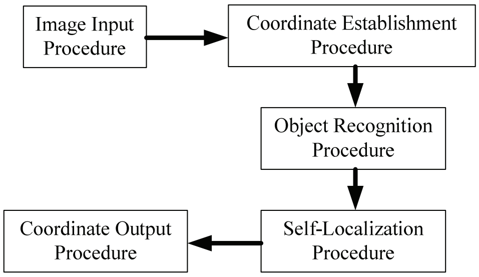


Image Input Procedure
Coordinate Establishment Procedure
Object Recognition Procedure
Self-Localization Procedure
Coordinate Output Procedure

Fig. 2: The AVBSLS flowchart for the humanoid robot.

### A. Image Input and Coordinate Output Procedures

The procedure of the image input and robot coordinates output is shown in Fig. 3. In this procedure, the captured image, RGB 24-bit image, is with resolution of 320 × 240 pixels.

In order to get accurate location results, the camera is set with a fixed focal length, and its value is not much concerned to the positioning system. The image data are thus processed and analyzed by the self-localization procedure to find the current coordinates of the humanoid robot.

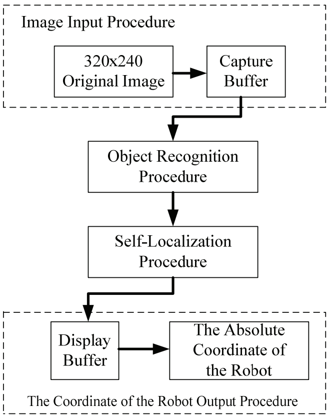


Image Input Procedure
320x240 Original Image
Capture Buffer
Object Recognition Procedure
Self-Localization Procedure
Display Buffer
The Absolute Coordinate of the Robot
The Coordinate of the Robot Output Procedure

Fig. 3: The architecture flow chart.

### B. Coordinate Establishment Procedure

Before processing the localization procedure, we must establish two appropriate coordinate systems, "absolute coordinate system" on the field and "relative coordinate system" in the image. It needs four steps to establish the absolute coordinate system: *1)* to estimate the sizes of the field and robot; *2)* to find the interested position in the soccer field; *3)* according to the proportion of the robot in the field to adjust the value in each block; *4)* to divide the field into several blocks with the same size and assign the interested position as the center block. Through these coordinate systems, the location of the robot, landmark, and goal can be located explicitly.

### C. Object Recognition Procedure

#### C.1. Color Based Object Recognition Method

The flow chart of a traditional color recognition method is shown in Fig. 4. The RGB color model [13] comes from the three additive primary colors, red, green, and blue. The RGB color model can describe all colors by different proportion combinations. Because the RGB color model is not explicit, it can be easily influenced by the light illumination and make people select error threshold values.

An HSV color model [13] relates the representations of pixels in the RGB color space, which attempts to describe the perceptual color relationships more accurately than RGB. Because the HSV color model describes the color and brightness component respectively, the HSV color model is not easily influenced by the light illumination. The HSV color model is therefore extensively used in the fields of color recognition. The HSV transform function is shown in equations (1)-(3) as follows:

$$H = \begin{cases} \left(6 + \dfrac{G-B}{C_{MAX} - C_{\min}}\right) \times 60°, \textit{if } R = C_{MAX} \\ \left(2 + \dfrac{B-R}{C_{MAX} - C_{\min}}\right) \times 60°, \textit{if } G = C_{MAX} \\ \left(4 + \dfrac{R-G}{C_{MAX} - C_{\min}}\right) \times 60°, \textit{if } B = C_{MAX} \end{cases} \quad (1)$$

$$S = C_{MAX} - C_{\min} / C_{MAX} \quad (2)$$

$$V = C_{MAX} \quad (3)$$

In equations (1)-(3), H is hue, and its range is 0°~360°; S means saturation, and its range is 0~1; V represents value, and its range is 0~255. The RGB values are confined by (4):

$$C_{MAX} = MAX(R,\ G,\ B);\ C_{\min} = \min(R,\ G,\ B) \quad (4)$$

where $C_{MAX}$ is the maximum value in the RGB color components, and $C_{min}$ is the minimum value in the RGB color components. Hence, we can directly make use of H and S to describe a color range of high environmental tolerance. It can help us to obtain the foreground objects mask M(x, y) by the threshold value selection in (5).

$$M(x,y) \begin{cases} 1,\ \textit{foreground} \\ \quad \textit{if } T_L < H < T_H S > Thd_S \\ \\ 0,\ \textit{background} \\ \quad \textit{otherwise} \end{cases} \quad (5)$$

$$T_L = Thd_H - R_{H,}\ T_H = Thd_H + R_H$$

where $Thd_H$, $Thd_S$, and $R_H$ are the threshold of hue, threshold of saturation, and the range of hue respectively by manual setting. The foreground object mask usually accompanies with the noise, and we can remove the noise by the simple morphological methods, such as dilation, erosion, opening, and closing. It needs to separate the objects by labeling when many objects with the same colors are existed in the frame. The following procedures are the operation flow for labeling:

Step 1: Scan the threshold image M(x,y).
Step 2: Give the value $\text{Label}^{i}_{color}$ to the connected component Q{n} of pixel(x,y).
Step 3: Give the same value $\text{Label}^{i}_{color}(x,y)$ to the connected component of Q{n}.
Step 4: Until no connected component can be found.
Step 5: Update $\text{Label}^{i}_{color}$ , i = i+1. Then go to Step 1 and repeat Steps 2-4.
Step 6: Completely scan the image.

By using the procedure mentioned above, the objects can be extracted. Although this method is simple, it is only suitable for low frame rate sequences. For a high resolution or noisy sequence, this approach may need very high computation complexity.

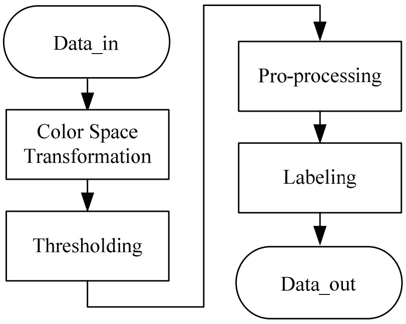


Fig. 4: The flow chart of the traditional color recognition method.

### *C.2. Low Resolution Method*

For overcoming the above-mentioned problems, several approaches of low resolution method were proposed [10][14]. The flow chart of a general low resolution method is shown in Fig. 5. Several low resolution methods, such as the approach of applying 2-D Discrete Wavelet Transform (DWT) and the using of 2×2 average filter, were discussed. Cheng *et al.* [14] applied 2-D DWT for detecting and tracking moving objects and only the LL3-band image is used for detecting motion of the moving object. Because noises are preserved in high-frequency, it can reduce computing cost for post-processing by using the LL3-band image. This method can be used for coping with noise or fake motion effectively; however the conventional DWT scheme has the disadvantages of complicated calculation when an original image is decomposed into the LL-band image. Moreover if it uses an LL3-band image to deal with the fake motion, it may cause incomplete moving object detecting regions. Sugandi *et al.* [10] proposed a simple method by using the low resolution concept to deal with the fake motion such as moving leaves of trees. The low resolution image is generated by replacing each pixel value of an original image with the average value of its four neighbor pixels and itself as shown in Fig. 6. It also provides a flexible multi-resolution image like the DWT. Nevertheless, the low resolution images generated by using the 2×2 average filter method are more blurred than that by using the DWT method, as shown in Fig. 7. It may reduce the preciseness of post-processing (such as object detection, tracking, and object identification), because the post-processing depends on the correct location of the moving object detecting and accuracy moving object.

In order to detect and track the moving object more accurately, we propose a new approach, ARM, which is based on the 2-D integer symmetric mask-based discrete wavelet transform (SMDWT) [15]. It does not only retain the features of the flexibilities for multi-resolution, but also does not cause high computing cost when using it for finding different subband images. In addition, it preserves more image quality of the low resolution image than that of the low resolution method [10].

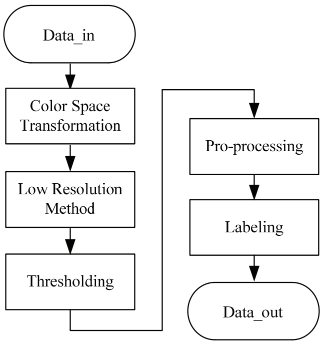


Fig. 5: The flow chart of a general low resolution method.

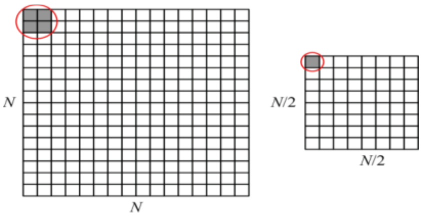
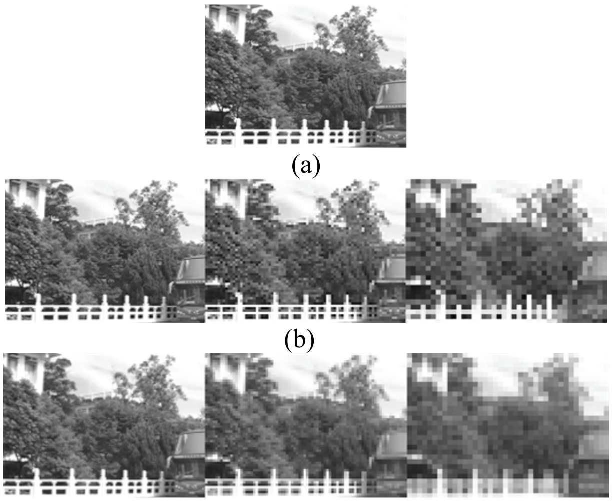


Fig. 6: Diagram of the 2×2 average filter method.



(a)

(b)

(c)

Fig. 7: Comparisons of low resolution images. (a) Original image (320×240). (b) Each subband image with DWT from left to right as 160×120, 80×60, and 40×30, respectively. (c) Each resolution image with the 2×2 average filter method from left to right as 160×120, 80×60, and 40×30, respectively.

### *C.3. Adaptive Resolution Method (ARM)*

ARM takes the advantage of the information obtained from the camera and motor of the robot to know the object distance and chooses the most proper resolution. The operation flow chart is shown in Fig. 8. After HSV color transformation, ARM has two operation modes, manual mode and tracking mode. The manual mode can let us manually choose the resolution. The high resolution approach brings a better recognizable distance but with a slower running speed. On the other hand, the low resolution approach brings a lower recognizable distance but with a faster running speed. When we get the information from the motor angle, we can convert it as the "sel" signal through the adaptive selector to choose the appropriated resolution. The "sel" condition is shown in (6).

$$\begin{cases} \text{sel} = 0 \text{ (Do Nothing)}, & \text{if } D_{ball} > D_{thd2} \cup f_b = 1 \\ \text{sel} = 1 \text{ (1-Level DWT)}, & \text{if } D_{thd1} > D_{ball} \geq D_{thd2} \\ \text{sel} = 2 \text{ (2-Level DWT)}, & \text{if } D_{ball} \leq D_{thd1} \end{cases} \qquad (6)$$

The relationship between the resolution and the distance of the ball is described in Table 1. According to Table 1, we can conclude a distance equation as follows:

$$D_{ball} = H_{cam} \times \tan \theta_m \qquad (7)$$

where $H_{cam}$ is the height of the camera place, and $\theta_m$ is the information of the motor angle. In (6), $D_{thd1}$ and $D_{thd2}$ are the threshold values for the recognizable distance and are set to 0.6 and 2.5, respectively. $D_{ball}$ is the distance between the robot and ball that obtained from (7). In order to obtain more accurate $D_{ball}$, we have to keep the ball in the center of the frame to reach the function of ball tracking. If the ball disappears in the frame, the flag $f_b$ is set to 1. At the same time, the frame changes into the original size to have a higher probability to find out the ball. Since the sizes of the other critical objects (such as goal and landmark) in the field are larger than the ball, they can be recognized easily. Fig. 9 shows the results of different resolutions after the threshold processing.

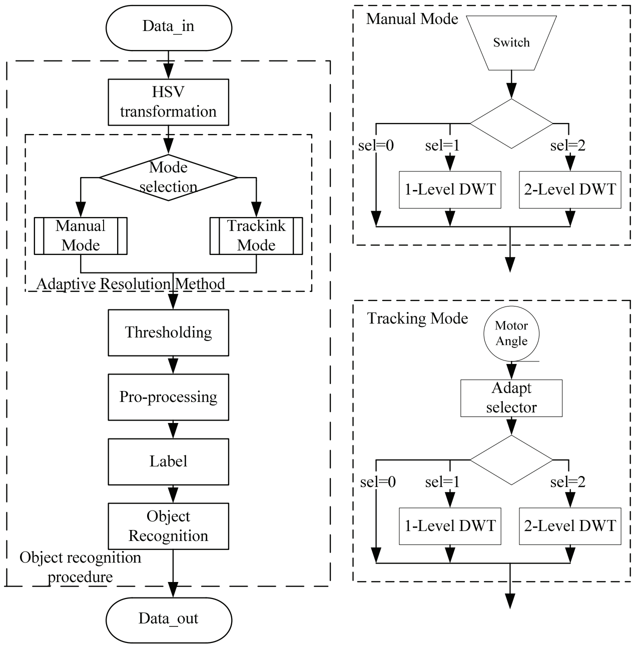


Fig. 8: The flow chart of ARM.

Table 1: The relationship between the resolution and the distance of the ball.

| *Resolution* | *Run time (sec)* | *Frame rate (fps)* | *Recognizable distance(m)* |
|---|---|---|---|
| 320×240 | 0.072 | 13.8 | 3.8 |
| 160×120 | 0.041 | 24.4 | 2.6 |
| 80×60 | 0.034 | 29.4 | 0.8 |

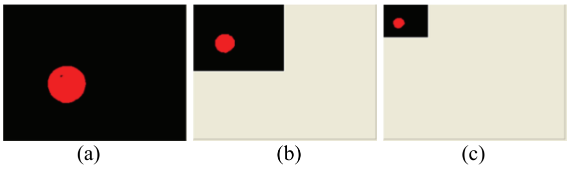


Fig. 9: The results after the threshold-processing under the resolution (a) 320×240. (b) 160×120. (c) 80×60.

*C.4. Sample Object Recognition Method*

According to the above-mentioned color segmentation method, it can fast and easily extract the orange ball in the field, but it is not enough to recognize the goals and landmarks. The colors of the goals and landmarks are yellow and blue, and by color segmentation the extraction of goals and landmarks may not be correct as shown in Fig. 10. Therefore we have to use more features and information to extract them. Since the contest field is not complicated, a simple recognition method can be used to reduce the computation complexity. The landmark is a cylinder with three colors. One of them the upper and bottom layers are yellow, and the center layer is blue; this one is defined as the YBY-landmark. The color combinations of the other one are in contrast to the previous one, and the landmark is defined as the BYB-landmark. The labels of the BYB landmark can be calculated by (8):

$$\begin{aligned}&Landmark_{BYB}(x,y)=Label_B^i(x,y)+Label_B^j(x,y)+Label_Y^K(x,y)\\&if\left|Label_B^j(X_{\min})-Label_B^i(X_{\min})\right|<\alpha\\&\cap\left|Label_B^j(X_{\max})-Label_B^i(X_{\max})\right|<\alpha\\&\cap Label_B^i(Y_{\max})<Label_Y^K(Y_c)<Label_B^j(Y_{\min})\end{aligned}\tag{8}$$

where $\text{Label}^{\text{i}}_{\text{color}}(\text{x,y})$ is the pixel of the i-th blue component in a frame, $\text{X}_{\text{min}}$ the minimum value for the object i at the x direction in the frame, $\text{X}_{\text{max}}$ the maximum value, $\text{Y}_{\text{min}}$ and $\text{Y}_{\text{max}}$ the minimum value and the maximum value at y direction respectively, and $\text{Y}_{\text{c}}$ the center point of the object at the vertical direction. The threshold value α is set as 15. The YBY landmark is in the same manner as the BYB landmark. The landmark is composed of two same color objects in the vertical line, and the center is in different color. If it can find an object with this feature, the system can treat this object as the landmark. (9) is used to define the label of the ball:

$$\begin{aligned}&Ball(x,y)=Label_0^3(x,y)\\&\text{if the size of } Label_0^3(x,y) \text{ is maximum of } Label_0\\&\cap\beta_1<\frac{Label_0^3(X_{\max})-Label_0^3(X_{\min})}{Label_0^3(Y_{\max})-Label_0^3(Y_{\min})}<\beta_2\end{aligned}\tag{9}$$

where $\text{Label}^{\text{i}}_{\text{color}}(\text{x,y})$ is the pixel of the s-th orange component in a frame. Since the ball is very small in the picture frame, in order to avoid the noise, the ball is treated as the maximum orange object and with a shape ratio of height to width approximately equal to 1. Here $\beta_1$ and $\beta_2$ are set to 0.8 and 1.2, respectively. The goal recognition is defined in (10).

$$\begin{aligned}&Goal_B(x,y)=Label_B^m(x,y)\\&if\,\frac{Label_B^m(X_{\max})-Label_B^m(X_{\min})}{Label_B^m(Y_{\max})-Label_B^m(Y_{\min})}>\gamma\end{aligned}\tag{10}$$

where $Label_B^m(x,y)$ is the pixel of the *m*-th blue component in a picture frame. Since the blue goal is composed of the blue object and the shape ratio of the height to the width is greater than 1.2, the parameter γ is set as 1.2. The yellow goal is in the same manner as the blue goal.

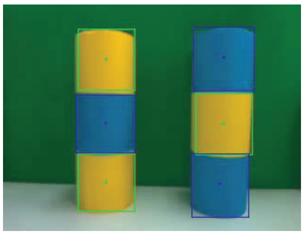


Fig. 10: False segmentation of the landmark.

*D. Self-Localization Procedure*

For self-localization, we proposed a new approach, adaptive vision-based self-localization system (AVBSLS), and AVBSLS consists of six steps. The operation flow of this self-localization mechanism is shown in Fig. 11. The details of the self-localization operations are described in the following subsections.

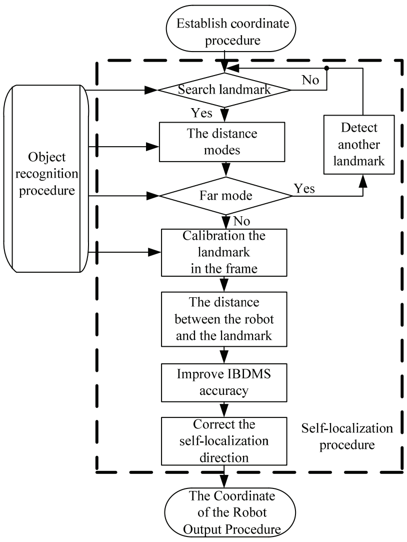

Establish coordinate procedure
Search landmark
No
Yes
The distance modes
Detect another landmark
Object recognition procedure
Far mode
Yes
No
Calibration the landmark in the frame
The distance between the robot and the landmark
Improve IBDMS accuracy
Correct the self-localization direction
Self-localization procedure
The Coordinate of the Robot Output Procedure

Fig. 11: The flowchart of self-localization procedure.

*D.1. Landmark Searching*

In the initialization of the orientation, the robot keeps searching one of the landmarks until finding it. Finally, it will mark five feature points, upper left $(X_1,Y_1)$, upper right $(X_2,Y_2)$, lower left $(X_3,Y_3)$, lower right $(X_4,Y_4)$, and center $(X_C,Y_C)$, for the landmark in the image, as shown in Fig. 12. According to the five feature points, the horizontal length $F_H$ and vertical length $F_V$ for the landmark in the frame can be found. To get robust $F_H$ and $F_V$, the landmark shape will be completely displayed in the frame, as shown in Fig. 13(b). Figs. 13(a) and 13(c) show the incomplete landmarks with partial landmark displayed in the frame. In order to insure the completeness of the landmark, the pixel values between the edge of the landmark and the edge of the frame should be greater than some appropriate values.

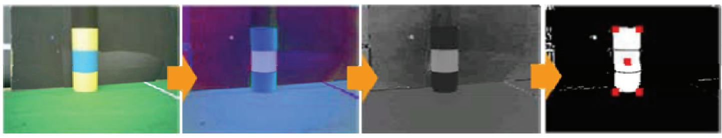


Fig. 12: The process to mark the upper left, upper right, lower left, lower right,and center of the landmark.

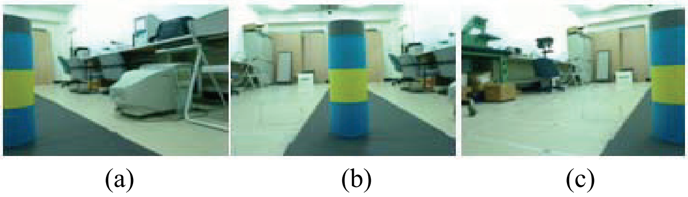


Fig. 13: Landmark searching (a) incomplete landmark, (b) complete landmark, (c) incomplete landmark.

*D.2. Distance Mode Selection*

According to the distance between the robot itself and the landmark, it is classified into three modes, near mode, mid mode, and far mode. During the self-localization process the pan motor can move randomly but the tilt motor is fixed in the center position. Fig. 14 shows the landmark images when the robot is very close to the landmark. In this situation, both $F_V$'s of the landmarks are indistinguishable and $F_H$ can be used to find the distance between the robot itself and the landmark, and this range is classified as the near mode. If the distance between the robot itself and the landmark is far excessively (more than 400cm for example), $F_H$'s and $F_V$'s for different landmarks are indistinguishable and Fig. 15 shows the situation. This distance range is classified as the far mode. If the distance between the robot itself and the landmark is in between the near mode and far mode, it is classified as the mid mode. Fig. 16 shows the images of the landmarks in the mid mode. In the mid mode situation, $F_H$'s are indistinguishable but $F_V$'s are distinguishable, and therefore $F_V$ can be used to find the distance between the robot itself and the landmark.

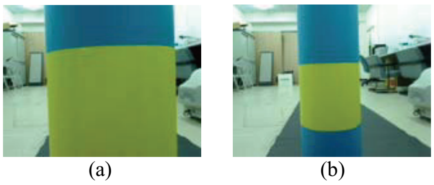


Fig. 14: The land mark images with distances of (a) 30cm and (b) 50cm. The ($F_H$, $F_V$) are equal to (a) (104,238), (b) (155,238) pixels in the frame.

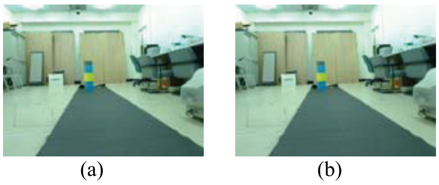


Fig. 15: The landmark images with distances of (a) 410cm and (b) 450cm. The ($F_H$, $F_V$) are equal to (a) (14,38), (b) (14,38) pixels in the frame.

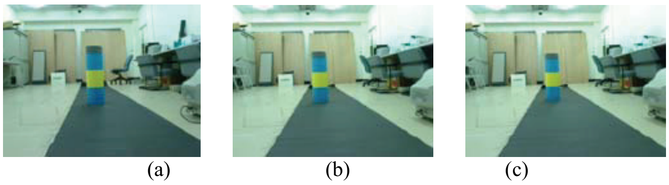


Fig. 16: The images of the landmark with distances of (a) 200cm, (b) 210cm,and (c) 250cm. The ($F_H$, $F_V$) are equal to (a) (28,80), (b) (28,77), and (c) (27,71) pixels in the frame.

*D.3. Landmark Calibration in the Frame*

Because the pan motor can move arbitrarily, the landmark may appear in any position in the frame. However, the nonlinear factors, such as the lens distortion of the CCD camera and the influence of the brightness of light may cause the divergence of the captured images. With the same distance between the robot itself and the landmark but the landmark in different positions of the frame, the pixel sizes of the landmarks are different and Figs. 17 and 18 show the features. Here we use the statistical approach to calibrate the size of the captured landmark.

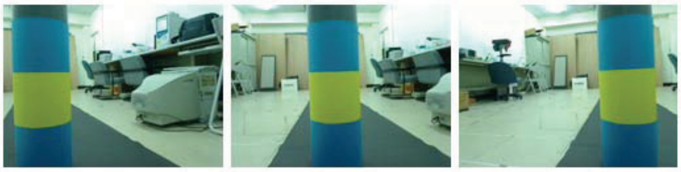


Fig. 17: With the same distance from the robot to the landmark, the ($F_H$,$F_V$) of the landmark at left, middle, and right positions of the frame are (86,227), (80,219), (87,223) pixels, respectively.

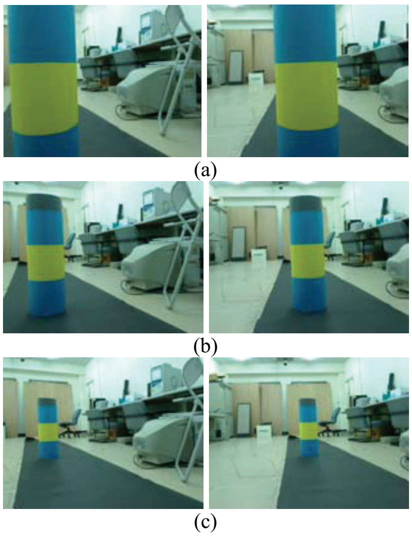


Fig. 18: The landmarks at the same position in the frame with different distance of (a) 50cm, (b) 100cm, and (c) 200cm have variations of $F_H$ between the left and right frames of (a) 9 pixels, (b) 5 pixels, and (c) 1 pixel.

*D.4. The Distance between the Robot and the Landmark*

The measurement of the distance between the robot itself and the landmark is accomplished by the modified IBDMS [12]-[13] approach. The details of the distance measurement are described in the following subsections.

### *D.4.1. Intrinsic Parameters of the CCD*

Here we try to find the intrinsic parameters of the CCD cameras regardless of the CCD makers. The relationship of the distance between the CCD camera and the object and some parameters are shown in Fig. 19. In Fig. 19, *OP* is the optical origin; $\theta_H$ and $\theta_V$ are the horizontal and vertical viewpoints, respectively. $D_{OP}$ is the distance between *OP* and the edge of the CCD lens.

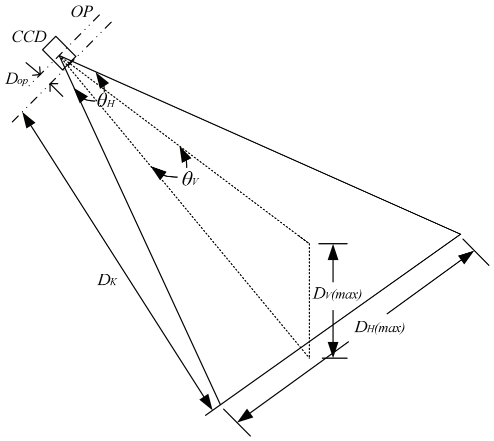
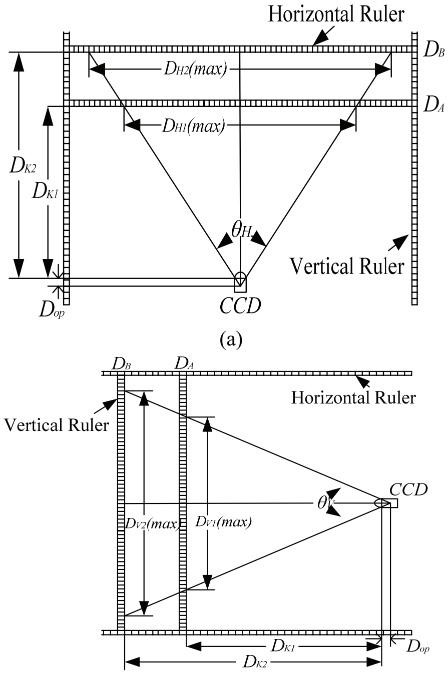


Fig. 19: CCD camera internal parameters diagram.

In order to find the distance between the CCD center and object, $\theta_H$, $\theta_V$, and $D_{OP}$ must be found first. We can capture the width of an object, $D_{H1}(max)$, with a pre-defined position, $D_A$, and the width of the same object, $D_{H2}(max)$, with a pre-defined position, $D_B$. The two captured frames can be combined to form Fig. 20(a).



(a)

(b)

Fig. 20: The intrinsic parameters measuring system (a) horizontal viewpoint, (b) vertical viewpoint.

Similarly, we can capture $D_{V1}(max)$ and $D_{V2}(max)$ with known pre-defined positions of $D_A$ and $D_B$ in two frames, respectively. These two captured frames can be combined to form Fig. 20(b). Due to the characteristics of the CCD camera, there is a $D_{OP}$ difference in between $D_A$ and $D_{K1}$, and there is also a $D_{OP}$ difference in between $D_B$ and $D_{K2}$. Same thing happens in the vertical viewpoint as shown in Fig. 20(b). Together with Figs. 20(a) and 20(b) and the trigonometric functions we can derive $\theta_H$ and $\theta_V$ in (11) and (12).

$$\theta_H = 2\cot^{-1}\left(2\times\frac{D_{K2}-D_{K1}}{D_{H2}(max)-D_{H1}(max)}\right) \tag{11}$$

$$\theta_V = 2\tan^{-1}\left(\frac{1}{2}\times\frac{D_{V2}(max)-D_{V1}(max)}{D_{K2}-D_{K1}}\right) \tag{12}$$

By the theorem of similar triangles and Figs. 20(a) and 20(b), the horizontal and vertical $D_{OP}$'s can be found as shown in (13) and (14).

$$\frac{D_{OP(H)}+D_{K1}}{D_{OP(H)}+D_{K2}} = \frac{D_{H1}(max)}{D_{H2}(max)} \tag{13}$$

$$\frac{D_{OP(V)}+D_{K1}}{D_{OP(V)}+D_{K2}} = \frac{D_{V1}(max)}{D_{V2}(max)} \tag{14}$$

$D_{OP}$ can be found by averaging $D_{OP(V)}$ and $D_{OP(H)}$ as shown in (15):

$$D_{op} = \frac{1}{2}\times\left(\frac{D_{K2}D_{H1}(max)-D_{K1}D_{H2}(max)}{D_{H2}-D_{H1}}+\frac{D_{K2}D_{V1}(max)-D_{K1}D_{V2}(max)}{D_{V2}-D_{V1}}\right) \tag{15}$$

The above approach can be applied to find the intrinsic parameters for any kind of CCD cameras.

### *D.4.2. Image-Based Distance Measurement*

By the IBDMS approach for calculating the distance between the CCD camera and landmark, $F_H$ and $F_V$ must be converted to $D_H(max)$ (the maximum horizontal width) and $D_V(max)$ (the maximum vertical width). The relationships of $F_H$ and $D_H(max)$ and $F_V$ and $D_V(max)$ are depicted in Fig. 21, and the conversion equations are shown in (16) and (17).

$$D_H(max) = \frac{F_H(max)}{F_H}\times D_{SH} \tag{16}$$

$$D_V(max) = \frac{F_V(max)}{F_V}\times D_{SV} \tag{17}$$

For a 320×240 picture frame, $F_H(max) = 320$, $F_V(max) = 240$, $D_{SH} = 20$cm, and $D_{SV} = 60$cm. $F_H$ and $F_V$ are the lengths in pixels for horizontal and vertical lengths of the landmark in the picture frame as shown in Fig. 22.

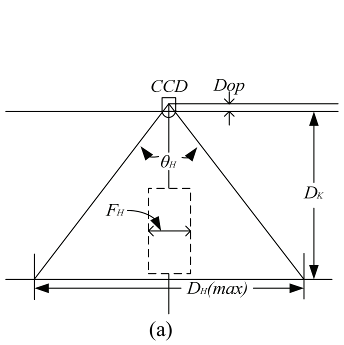


Fig. 21: Three-dimension distance measurement. (a) horizontal measurement, (b) vertical measurement.

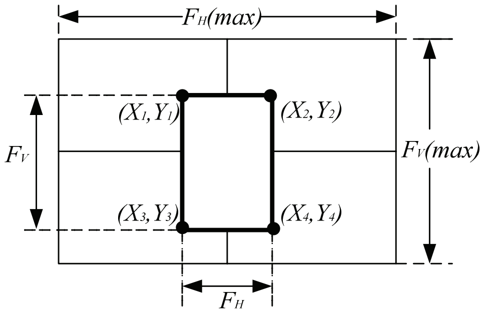


Fig. 22: The horizontal and vertical information of the landmark in the frame.

By the IBDMS approach, the distance $D_K$ (photo-distance), between the CCD camera and the landmark, by the horizontal view can be found from (18), and the vertical view can be found from (19).

$$D_K = \frac{1}{2} \times D_H(max) \times \cot(\frac{\theta_H}{2}) - D_{OP} \tag{18}$$

$$D_K = \frac{1}{2} \times D_V(max) \times \cot(\frac{\theta_V}{2}) - D_{OP} \tag{19}$$

Due to the non-ideality of the CCD lens and the brightness of light, the measurement of $D_K$ may be not accurate enough. It can be fine-tuned by the artificial neural network technique. The neural network technique is described in the following subsection.

### D.5. The Improved IBDMS

So far several neural network methods have been proposed, such as back propagation neural (BPN) network, self-organizing neural network, etc. Because the BPN network has the advantages for higher learning precision and fast recall speed [16]. Here we use the technique of BPN network to find a more accurate distance between the robot and landmark. There are seven steps to improve the distance precision by the BPN network and the procedures are shown in Fig. 23 [17].

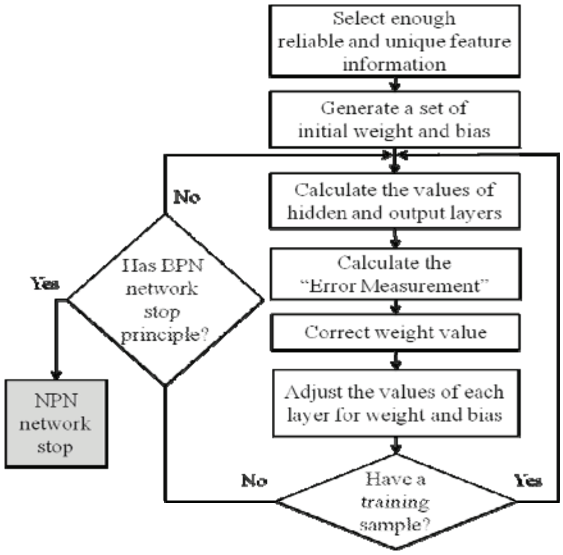

Select enough reliable and unique feature information
Generate a set of initial weight and bias
Calculate the values of hidden and output layers
Calculate the "Error Measurement"
Correct weight value
Adjust the values of each layer for weight and bias
Have a training sample?
No
Yes
Has BPN network stop principle?
No
Yes
NPN network stop

Fig. 23: The procedure for improving precision.

By the neural network method, we can get a more accurate distance between the robot and landmark.

### D.6. Calibration of the Self-Localization Direction

Here we use the rotation angle of the pan motor to decide the robot direction. It assumes that the rotation angle, *α'*, of the pan motor to be 0 degree when the center of the landmark is at the center of the frame. The pan motor can rotate left and right 25 degrees, respectively. However, the pan motor angle is *α* instead *α'* as shown in Fig. 24. In order to find *α'*, we must measure the horizontal pixel numbers, $X_C$, of the center of the landmark. From Fig. 24, when *α'* is found, we can find *β* and then the absolute coordinate *x'* and *y'* can be calculated as follows:

$$\begin{cases} x' = x + r\cos\beta \\ y' = y - r\sin\beta \end{cases}, \quad -90^\circ \le \beta \le 90^\circ \tag{20}$$

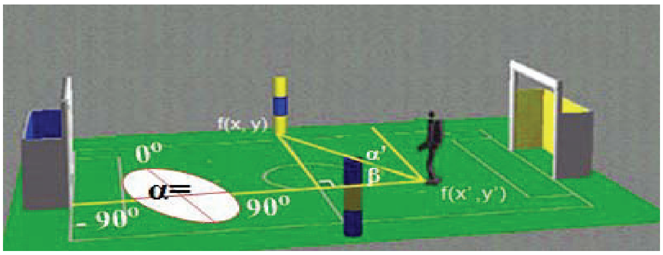


Fig. 24: The direction of the robot in the soccer field.

## III. Experimental Results

The experiment is based on the feature of the competition field for the 2009 RoboCup soccer humanoid league. The resolution is 320×240 pixels, and the frame rate is 30 fps. The field contains two goals and two landmark poles. Because the width of the robot shoulder is 26cm, we set the unit length of the coordinate to be 30cm in length and the field can be divided into 29×17 blocks as shown in Fig. 25. The experimental robot vision module comprises a single CCD camera and pan/tilt motors as shown in Fig. 18. The CCD camera is the Logitech QuickCam® Pro [18] for Notebooks, and the pan/tilt motors are ROBOTIS Dynamixel AX-12 [19].

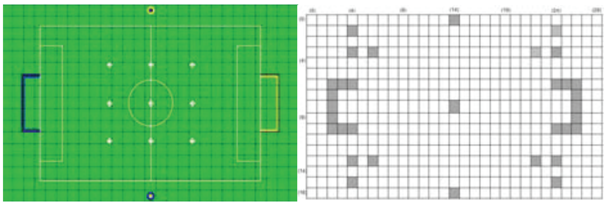

(a) (b)

Fig. 25: The RoboCup soccer field. (a) the original field with 29×17 blocks, (b) the coordinate of the soccer field [3].

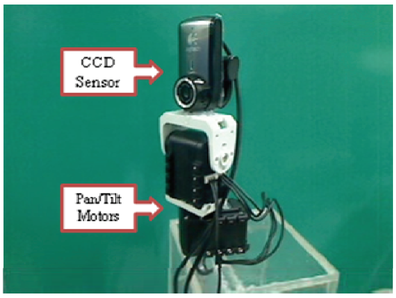


Fig. 26: The robot vision module.

### A. Room- in and room-out of the picture frame

In this experiment, the camera is set in the center of the field. The scene simulates that the robot kicks ball into the goal and the vision system will track the ball. The experimental results of the accuracy rate and average fps under different resolutions and ARM are shown in Table 2. "False Positive" means the error misdiagnosis. "False Negative" means that it does not recognize the object. According to Table 2 we find that even though the 320×240 resolution had high accuracy rate, the process speed is slow. The 80×60 resolution has the highest processing speed, but it has low accuracy rate. By this approach, it gets high accuracy rate only when the object is close to the camera. On the other hand, the proposed ARM does not only have high accuracy rate, but also keep high processing speed.

Table 2: The experimental results of accuracy rate and average fps under different resolutions and ARM.

| *Resolution* | *Total Frame* | *Object Frame* | *False* positive | *False Negative* | *Accuracy Rate* | *Average fps* |
|---|---|---|---|---|---|---|
| 320×240 | 124 | 87 | 0 | 3 | 86.55% | 14.2 |
| 160×120 | 230 | 164 | 2 | 43 | 72.56% | 24.3 |
| 80×60 | 242 | 181 | 1 | 110 | 38.67% | 29.1 |
| ARM | 135 | 97 | 1 | 2 | 96.91% | 20.5 |

### B. The function of object recognition

In this experiment, several scenes are simulated: Scene 1, it closes the ball slowly. Scene 2, the camera turns left to see the BYB landmark and keeps turning until the BYB landmark disappears. Scene 3, the camera turns up to see the goal and turns right and then turns left until the goal disappears. Scene 4, The YBY landmark is always in the frame and the ball enters from the bottom of the frame and then the camera turns left to see the similar color object. Scene 5, the camera turns left to see the YBY landmark, ball, and goal respectively. The experimental results of these scenes are shown in Table 3.

Table 3: The experimental results of the several kinds of scene simulation.

| *Scene* | *Total Frame* | *Object Frame* | *False positive* | *False Negative* | *Accuracy Rate* |
|---|---|---|---|---|---|
| (1)ball | 691 | 691 | 0 | 7 | 98.99% |
| (2)landmark | 290 | 191 | 3 | 21 | 87.43% |
| (3)goal | 232 | 212 | 0 | 13 | 93.87% |
| (4)ball&landmark | 753 | 753 | 0 | 18 | 97.61% |
| (5)ball&goal&landmark | 616 | 616 | 12 | 68 | 87.01% |
| Total | 2582 | 2463 | 15 | 127 | 94.23% |

### C. Obtaining the Camera Intrinsic Parameters

According to Section D4.1, we set the distance step as 10cm and start from $D_K$=30cm to measure $D_H(max)$ and $D_V(max)$ until $D_K$=400cm. The values of $\theta_H$, $\theta_V$ and $D_{OP}$ can be obtained accordingly. The averaged intrinsic parameters of $\theta_H$, $\theta_V$ and $D_{OP}$ can be found by (21), (22), and (23), respectively.

$$\cot\frac{\theta_H}{2}=\frac{1}{N-1}\sum_{i=1}^{N-1}\cot\frac{\theta_H}{2}(i), i=1\sim(N-1)\Rightarrow\cot\frac{\theta_H}{2}=1.701 \quad (21)$$

$$\tan\frac{\theta_V}{2}=\frac{1}{N-1}\sum_{i=1}^{N-1}\tan\frac{\theta_V}{2}(i), i=1\sim(N-1)\Rightarrow\tan\frac{\theta_V}{2}=0.444 \quad (22)$$

$$D_{op}=\frac{1}{N-1}\sum_{i=1}^{N-1}D_{op}(i), \qquad i=1\sim(N-1) \Rightarrow D_{op}=3.0 \qquad (23)$$

### D. The Analysis for the Actual Measuring Distance

This work uses IBDMS and improved IBDMS techniques to measure the distance between the robot and landmark from 30cm to 400cm. Fig. 27 shows the errors between the measuring distance and actual distance, and the results are listed in Table 4. In the distance from 30cm to 400cm, the average and maximum errors for the IBDMS are 7.08cm and 15.0cm, respectively. On the other hand, those of the improved IBDMS are 0.82cm and 6.0cm, respectively. The proposed improved IBDMS approach improves the accuracy significantly.

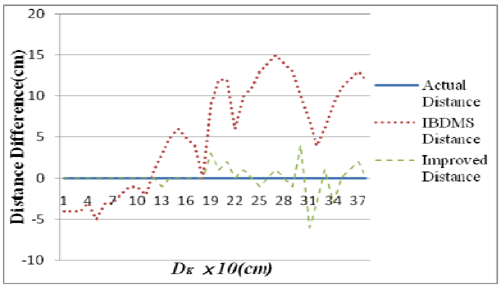


Fig. 27: The distance differences for the IBDMS, improved IBDMS, and actual distance.

Table 4: Comparisons of the maximum and average error for different Methods.

| *Total experimental point = 130* | | | |
|---|---|---|---|
| Situation | Correct | Incorrect | Accuracy rate |
| AVBSLS (IBDMS) | 92 | 38 | 70.8% |
| AVBSLS (Improved IBDMS) | 120 | 10 | 92.3% |

### *E. The Results for the Self-Localization in the Contest Field*

The robot position was measured by IBDMS and AVBSLS in the actual field. Since the left and right sides of the field are with the same situation (as shown in Fig. 26), without loss of generality this experiment focuses on the right side of the field. Fig. 28 shows the measurement results of various locations for the robot by AVBSLS, where the stars indicate the various locations of the robot.

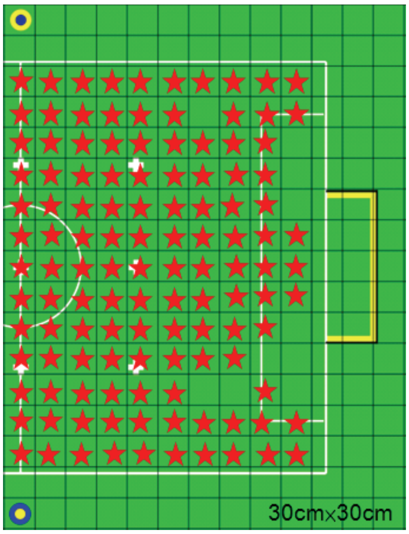


Fig. 28: The positions of the improved architecture. (The stars are the robot locations on the correct positions at that moment.)

## IV. Conclusion

This work proposes an efficient approach AVBSLS of self-localization and object recognition for humanoid robot. The major mechanism is a single CCD camera and pan motor on the robot head. This research accomplishes self-localization and object recognition for humanoid robot within an image and the landmark can be located at any position in the frame. The humanoid robot can use any type of CCD camera. Through the camera intrinsic parameters and the proposed AVBSLS, the position of the humanoid robot can be found. The distance between the robot and the landmark is measured by the improved IBDMS method. Compared with IBDMS, the improved IBDMS approach has less average error. Therefore, the accuracy for self-localization can be increased significantly. Due to the simple processing operation by ARM approach, the processing speed can be as high as 15 fps.

## Acknowledgment

This work was supported by the National Science Council of Taiwan, R.O.C. under grant number NSC-98-2218-E-032-003.

## References

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, “Robocup: The robot world cup initiative,” *IJCAI-95 Workshop on Entertainment and AI/ALife*, pp. 19-24, 1995.

[2] RoboCup Soccer Humanoid League Rules and Setup for the 2009 competition. http://www.robocup2009.org/153-0-rules.

[3] N. Herodotou, K. N Plataniotis, and A. N. Venetsanopoulos, “A color segmentation scheme for object-based video coding,” *IEEE Symposium on Advances in Digital Filtering and Signal Processing*, pp.25-29, Jun. 1998.

[4] O. Ikeda, “Segmentation of faces in video footage using HSV color for face detection and image retrieval,” *International Conference on Image Processing*, vol. 3, pp. 913-6, Sep. 2003.

[5] F. Chaumette, “Visual servoing using image features defined on geometrical primitives,” *IEEE Conference on Decision and Control*, pp. 3782-3787, Dec. 1994.

[6] J.-H. Jean and R.-Y. Wu, “Adaptive visual tracking of moving objects modeled with unknown parameterized shape contour,” *IEEE Conference Networking, Sensing and Control*, pp. 76-81, Mar. 2004.

[7] S. J. Sun, D. R. Haynor, Y. M. Kim, “Semiautomatic video object segmentation using V snakes,” *IEEE Transactions on Circuits System Video Technol.* vol. 13, no. 1, pp.75-82, Jan. 2003.

[8] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: active contour models,” *International Journal of Computer Vision.*, vol. 1, pp.321–331, Jan. 1988.

[9] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, Nov. 1986.

[10] B. Sugandi , H. Kim, J. K. Tan, and S. Ishikawa, “Tracking of moving objects by using a low resolution image,” *International Conference on Innovative Computing, Information and Control,* pp. 408-408, Sep. 2007.

[11] C. C. Chen, M. C. Lu, C. T. Chuang, and C. P. Tsai. “Vision-Based Distance and Area Measurement System,” *IEEE Sensors Journal*, vol. 6, no. 2, pp. 495-503, April 2006.

[12] C. C. Hsu, M. C. Lu, W. Y Wang, and Y. Y. Lu. “Distance Measurement Based on Pixel Variation of CCD Images,” *ISA Transactions*, vol. 48, issue 4, pp. 389-395, October 2009.

[13] R. C. Gonazlez and R. E. woods, *Digital Image Processing*, 2$^{nd}$ edition, Addison-Wesley, 1992.

[14] F.-H. Cheng and Y.-L. Chen, “Real time multiple objects tracking and identification based on discrete wavelet transform,” *Pattern Recognition*, vol. 39, no. 3, pp. 1126-1139, Jun. 2006.

[15] C.-H. Hsia, J.-M. Guo, and J.-S. Chiang, “Improved low-complexity algorithm for 2-D integer lifting-based discrete wavelet transform using symmetric mask-based scheme,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 8, pp. 1-7, Aug. 2009.

[16] L.-H. Chiang, *Neural Network—Application of MATLAB*, Gau-Lih Publish, Seventh Edition, July 2005.

[17] F.-J. Chang and L.-C. Chang, *Artificial Neural Network*, Tun-Ghua Publish, Third Edition, August 2007.

[18] Logitech QuickCam® Pro for Notebooks. http://www.logitech.com/index.cfm/home/&cl=us.

[19] AX-12 Manual (English). http://www.robotis.com/zbxe/software_en