Image Processing Based Robot Soccer: Obtaining Multiple Robots Position and Orientation Using High-Angle Shot of Camera

Hendra Tjahyadi, Giorgy Gunawan, Arnold Aribowo, and David Hareva
Faculty of Computer Science, Universitas Pelita Harapan, Tangerang-15811, Indonesia
Email: {hendra.tjahyadi, giorgy.gunawan, arnold.aribowo, david.hareva}@uph.edu

***Abstract*—This paper proposes an alternate method to obtain position and orientation of multiple soccer robots in global vision system. The proposed method uses high-angle shot of camera as a substitute to bird-eye angle, and employs a specially designed robot markers that contain cues for the detection of the position, orientation, and IDs of the robots. Perspective transformation to normalize the perspective arena image obtained from high-angle shot is utilized in this method. Three testing phase, namely unit testing, integration testing, and system testing are conducted. The results of unit testing are: four arenas are detected from four different placements in the room, the robot detection has an error of 0.86%, and ball detection has an error of 0.4%. Integration testing obtains the global orientation of the robot, which has an error of 1.1%. From the result of system testing, the best camera resolution is 360p with 50% minimum camera brightness. The result of system testing also proves that the application system can be implemented in different rooms and with different camera positions.**

***Index Terms*—robot soccer, global vision, image processing, border tracing, perspective transformation, high-angle shot**

I. Introduction

Robot soccer is a multi-agent system. Three major approaches are common to control the multi agent system: decentralized approach, centralized approach, and the combination between the two of them, which is hybrid approach [1]. Most people came up with a decentralized approach, equiping each robot with a visual feed to detect the ball and then dribble it towards the goal [2]. One of the flaws of this approach is a probability that those robots will collide when all of them are moving towards the ball. To compensate this flaw, a supervisor that oversees the situation and tells the robots to execute the strategy is added. This control approach is known as centralized approach.

To implement the centralized approach, a single visual feed will be processed with image processing and computer vision methods [3]. The image processing algorithm is the key core to this strategy, because a mere image of the arena can not bring out important information such as robot position and orientation, and ball and goal position to be processed as a strategy. After the provision of the position and orientation by image processing is ready, the information then will be processed through strategy algorithm which results in movement commands for each robot. The application of this centralized approach strategy using image processing can enhance the winning rate as it applies strategy and game plan to the robots team.

This paper focuses in acquiring the position and orientation of the robots to support the centralized approach. This objective can be achieved through four following steps: apply perspective transformation onto the soccer robot arena, acquire the positions and orientations of multiple robots which are in the arena, acquire the position of the ball, and make a simple trajectory planning of the movements from the gathered information. For the visual feed positioning. High-angle shot is used as an alternative to bird-eye angle.

The discussion in this paper will continue on in several sections as follows: in Section II some theoretical foundations on image processing are discussed, followed by the system analysis and design in Section III. Implementation and testing is found in Section IV, and in Section V this paper is concluded.

II. Theoretical Foundations

A. Centralized Vision System in Robot Soccer

Vision system in robot soccer is an instrument to obtain an accurate world state of the arena i.e. positions, orientations, and velocities, and the objects of robot soccer [4]. In Fig. 1, it is shown that vision system is a part of centralized robot soccer. As shown in the figure, a camera is required to capture the image from the real world. Generally, the camera is positioned above the arena and commonly known as the term 'overhead camera'. There are 2 kinds of overhead camera angle: high-angle and bird-eye angle. With bird-eye angle, the camera is positioned directly above the object. With high-angle, the camera is positioned higher than the object, but not necessarily directly above.

B. Object Detection in Robot Soccer

Detection of objects in robot soccer vision system usually consists of two steps: cue detection and feature

Manuscript received August 15, 2015; revised November 2, 2015.

doi: 10.18178/joig.4.1.29-35

extraction [4]. Cue detection is a retrieval of certain cue out of an image such as color, pattern, or shape. Feature extraction is a process of acquiring necessary state variables out of the retrieved cue, such as position, orientation, or velocity.

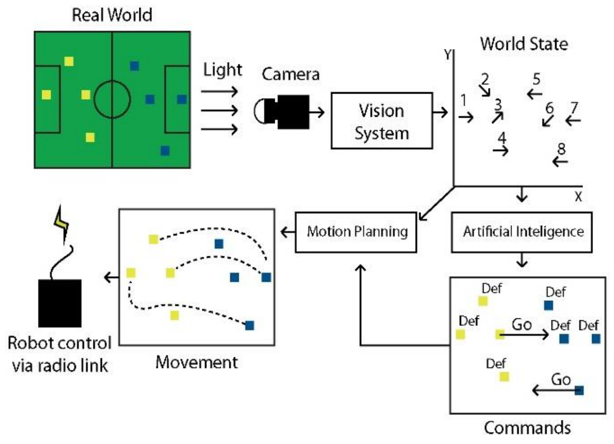

Figure 1. Detailed overall system of centralized approach robot soccer source: Kooij, 2003

To carry out cue detection, several techniques based on image processing operations such as edge detection, background subtraction, and color segmentation can be chosen.

To measure position state variable in feature extraction, center of gravity of the regions found by cue detection is used. While the orientation can be measured by using combination of regions, usually there is a heading region specified on the robot.

C. Contour and Border Tracing

Contour is another representation of boundary. It is a curve which joins all continuous points with the same intensity. There exist 2 kinds of contour: an open contour which indicated as a curve that discontinues at some point, and a closed contour which indicated as a curve that corresponds to region boundary. To trace contour there exist several algorithm such as: Square tracing, Moore-neighbor tracing, radial sweep, Theo pavlidis', and border tracing. In this paper border tracing algorithm is used.

Border tracing algorithm is used to extract the contours of the objects (regions) of an image [5]. When applying the algorithm it is assumed that the image with regions is either binary or those regions has been previously labeled.

D. Perspective Transformation

Perspective transformation is a transformation which maps an arbitrary quadrilateral into another quadrilateral [6]. Perspective transformation preserves parallel lines only when they are parallel to the projection plane. Otherwise, lines converge to a vanishing point [7]. A general form of perspective transformation is denoted as:

$$[x' \quad y' \quad w'] = [u \quad v \quad 1]\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$where\ x = \frac{x'}{w'}\ ; y = \frac{y'}{w'} \qquad (2.1)$$

Using i = 1 to normalize the transformation matrix, the forward mapping functions of perspective transformation can be denoted as:

$$x = \frac{au+dv+g}{cu+fv+1} \qquad (2.2)$$

$$y = \frac{bu+ev+h}{cu+fv+1} \qquad (2.3)$$

By multiplying each side of the equation by the denominators and isolating the independent x and y, the equation can be rewritten as:

$$x = au + dv + g - xuc - svf \qquad (2.4)$$

$$y = bu + ev + h - yuc + yvf \qquad (2.5)$$

Perspective transformation should be conducted by four source points to four reference points, where the points are the corners of the quadrilaterals. Let (u_k, v_k) and (x_k, y_k) for $k = (1,2,3,4)$ be the four points of the source and reference. Applying (2.4) and (2.5) to the four pairs of corresponding point yields 8x8 system of equations shown in (2.6), where $[a\ b\ c\ d\ e\ f\ g\ h]^T$ is the transformation matrix that can be determined by inversion.

$$\begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -v_3x_3 \\ u_4 & v_4 & 1 & 0 & 0 & 0 & -u_4x_4 & -v_4x_4 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3y_3 & -v_3y_3 \\ 0 & 0 & 0 & u_4 & v_4 & 1 & -u_4y_4 & -v_4y_4 \end{bmatrix}\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

(2.6)

E. Polygon Approximation

One of the methods to achieve polygonal approximation is called splitting. Splitting subdivides a segment successively into two parts until a given circumstances is met. For a closed boundary, the best starting points are usually the two farthest points in the boundary, after that the farthest point from the subdivision line should be another point and this part repeats until no more point can be made. Fig. 2 illustrates the splitting technique.

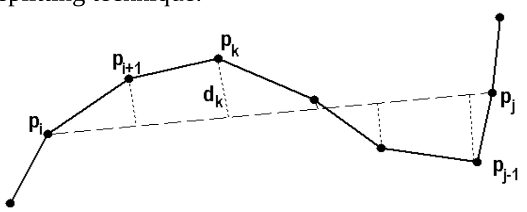

Figure 2. Splitting a curve into segments based on distance source: http://cs.joensuu.fi/~koles/approximation/Ch3_1_files/Image1.gif

III. System Analysis and Design

A. Problem Formulation

Three problems are recognized in the provision of position and orientation of the soccer robots using high-angle shot of camera, namely: camera positioning, perspective distorted arena shape, and objects detection.

To capture the arena environment using high-angle camera shot, camera positioning is important. The camera positioning should consider the angle of view capability which depends on the camera, and also the environmental factor such as lightning and obstructive objects. Hence, a physical setup of the camera and the environment should be established.

With high-angle camera shot, the arena will experience a perspective distortion. To deal with this distortion, the arena should be warped into its normal rectangular shape.

There are two objects that have to be detected: the robots and the ball. The detection consists of two steps, cue detection and feature extraction. The results of feature extraction are state variables such as position and orientation of the objects.

Based on the problem formulation, to achieve the objective, several conditions for physical setup and for application requirements are set as follows: (i) the camera has to be positioned to be able to capture the full image of the arena, (ii) the arena has adequate lighting, (iii) there is no other objects in the camera's field of view besides the arena, the robots and the ball, (iv) the system is able to detect the arena and apply perspective warping onto it, (v) the system is able to detect the cue of the robots and the ball, and to extract position and orientation from the detected cues.

B. Physical Setup Design

A webcam Logitech C525 is used as the camera in the physical setup. The camera has 69 degree diagonal field of view and 54.5 degree vertical field of view. A green carpet of 118×198cm is used as the arena.

The robots will be represented by their patterns. The patterns act as a cue to be detected by cue detection. As there are 5 robots in one team, a pattern to distinguish one robot from another is designed and showed in Fig. 3.

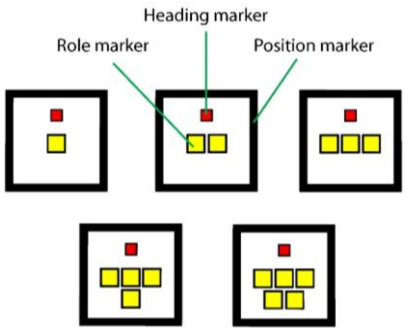

Figure 3. Patterns of the robots

The pattern has 3 features to be extracted: robot position, robot orientation, and robot role. The robot position is determined by the center of gravity of the pattern. The robot orientation is determined by combining the center of the largest square with the center of the red heading marker. The robot role is determined by counting the yellow role markers (the average square). Role marker number one and two are the strikers, number three and four are the backs, and number five is the goal keeper. By utilizing high angle shot of a camera, the position of camera can be varied by the distance from the arena, the height of the camera from the ground, and the depression angle of the camera towards the ground. From experiments, the best view of the arena is achieved when the distance of the camera to the arena is 85 cm with 38 degree depression angle. Fig. 4 shows the illustration of the arena, the camera, and the robots positions.

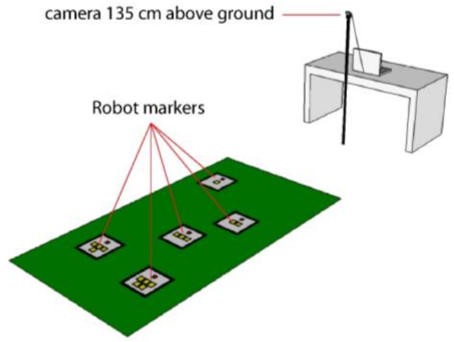

Figure 4. Illustration of the physical setup design

C. Application System Design

From the application requirements, the application should have 4 main processes: arena detection, robot detection, ball detection, and rotation angle and translation trajectory calculation.

Initially the input image of the arena will be loaded, followed by the detection of the arena and the application of perspective transformation onto it, subsequently. After that, robot cue detection and robot feature extraction shall commence. The next step is ball cue detection and ball feature extraction. After the state variables information has been extracted by feature extraction, the information will be stored in robot and ball class, and the rotation angle and translation trajectory are calculated as information for robots movement.

From the physical setup, the camera will capture arena image which will then go through the process of arena detection. The next step is acquiring the position and orientation of the robots in respect to the world coordinate. The world coordinate origin is the origin of MatrixC[x,y]. The position will be represented in coordinates (x, y), and the orientation will be represented in radians. As discussed, robot orientation is acquired by combining 2 coordinates: the center and the heading marker, which means aside from the coordinate of the robot itself, coordinate of the heading marker also needs to be retrieved. The role markers also need to be detected in order to know the role of each robot. Fig. 5 shows the illustration of detecting the robots' position, heading marker coordinate, and detecting the role of the robots. In the figure, the process has three main parts which are robot shape detection, heading marker checking, and Role marker checking.

The subsequent step is ball detection. The cue to detect the ball is color. The ball has an orange color which is distinguishable from other object in the arena. MatrixCHSV[x,y] can be utilized to do color thresholding to obtain the color orange. The result of the thresholding is a binary image which is stored in MatrixCBall[x,y]. Subsequently, MatrixCBall[x,y] will be traced for contour and apply polygon approximation on it to obtain the center coordinate of the ball.

The last step is to calculate the rotation angle and the translation trajectory for the robots to move from one point to another point in the arena. The angle of rotation is the angle needed for the robot to face the ball and the translation trajectory is a set of coordinates that is needed to be passed after the rotation in order to reach the ball. To calculate the angle of rotation Atan2 is used [8], and the trajectory path is limited to a straight path between one point and another point, for example the robot and the ball.

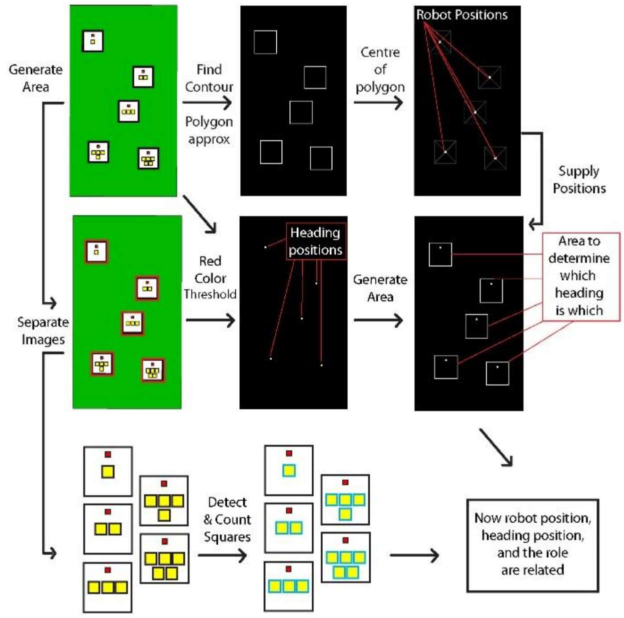

Figure 5. Process of detecting robot position, header position, and role of the robots

IV. Implementation and Testing

The implementation consists of the physical setup implementation and the application implementation. After the implementation, testing phases are conducted and the results are analyzed.

Three testing phases are conducted: unit testing, integration testing, and system testing. The purpose of unit testing is to examine the functionality and the accuracy of robot detection and ball detection. The purpose of integration testing is to examine the accuracy of the combined information, such as the global orientations of each robot, the angle of rotation needed to face the ball position, and the distance from the robot to the ball. The purpose of system testing is to examine the effect of environmental and hardware changes such as: change of resolution, room substitution, change of lighting, and change of camera position.

A. Physical Implementation

Fig. 6 shows the implementation of the physical setup. The camera is positioned 135cm above the ground and 85cm from the arena, the arena is 118×198cm and the printed robot patterns are 18×18cm. The computer runs the robot position and orientation detection application which uses the camera to capture the arena image. The arena is placed in a 11×15m^2 room with 3 neon lamps of 36 watt.

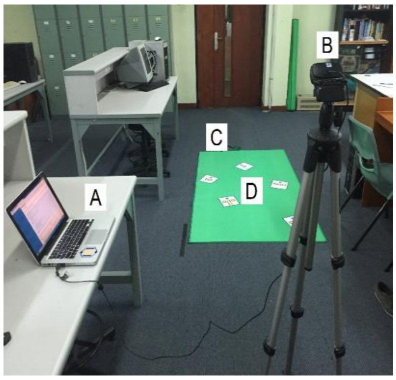

Figure 6. Implementation of physical setup (A) the computer, (B) the camera 135cm from the ground and 85cm from the arena, and (C) the arena with (D) the robot patterns on it.

B. Application Implementation

The application is implemented using C++ programing language as a console application. After implementing Robot detection and Ball detection, the data about positions of the robots and the orientations of the robots (both global and ball orientation) are acquired. Table I and Table II list the gathered data and Fig. 7 illustrates the robots, the ball, and the gathered data.

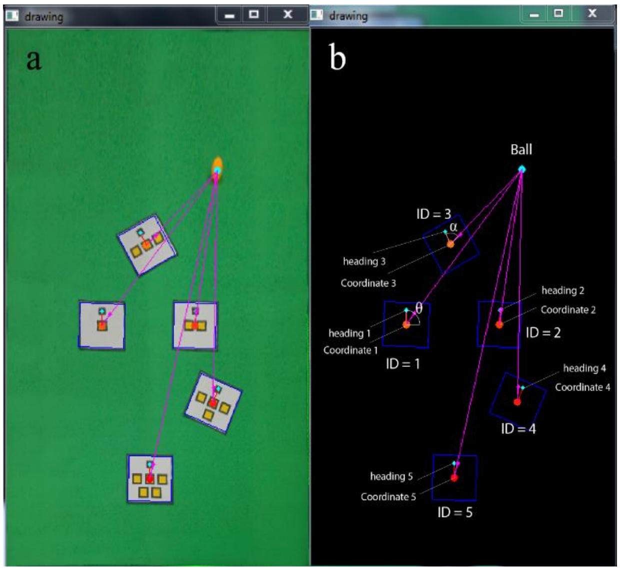

Figure 7. Gathered data after the implementation (a) illustrated over quad (b) illustrated on zero matrix

TABLE I. List of Gathered Data

Id	Coordinate of robot	Coordinate of heading	Global orientation (θ)
1	(105,309)	(105 , 294)	90.000
2	(208,309)	(209 , 294)	86.186
3	(154.5,225.5)	(148 , 212)	115.709
4	(228.5,389.5)	(234 , 375)	69.227
5	(158.5,469)	(148 , 212)	91.909

TABLE II. List of Gathered Data

Angle of rotation (α)	Direction
38.313	Clockwise
4.958	Clockwise
70.709	Clockwise
19.709	Counter Clockwise
14.936	Clockwise

C. Application Testing

Application testing is accomplished by following three testing phases of white box testing. They are unit testing, integration testing, and system testing.

Unit testing examines the functionality and the accuracy of the functions of the application in arena detection, robot detection, and ball detection. For arena detection, only the functionality is tested. Arena detection is proven functional if it can detect arena in given arena image.

The result of arena detection unit testing is shown in Fig. 8 using Hough transform and Fig. 9 using contour and polygon approximation. Hough transform is found inefficient in this testing as it needs a lot of parameter alteration.

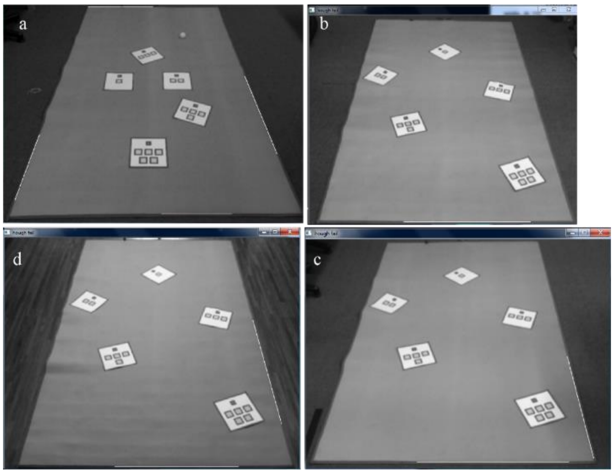

Figure 8. Testing arena detection using probabilistic Hough line transform only (a) has 4 lines detected, while (b), (c), and (d) fails to detect 4 lines.

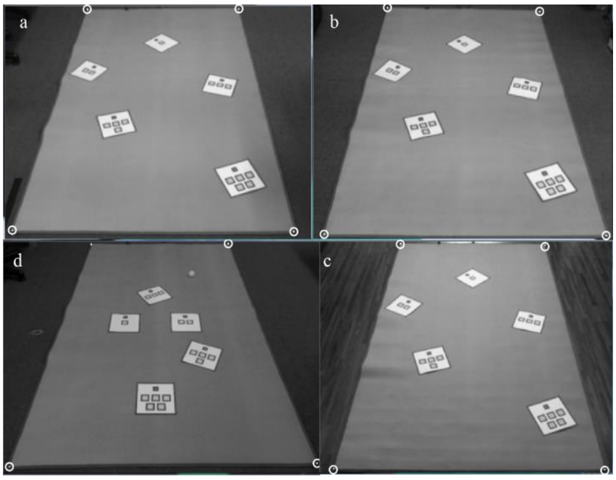

Figure 9. Testing arena detection using polygon approximation. The white point shows the detected. The white circles are made to show where the white points are. All images have their corners detected.

The result of robot detection unit testing is shown in Table III and Table IV, both tables show the comparison of the detected coordinate and real world (measured) coordinate.

The result of ball detection unit testing is shown in Table V, the table shows the comparison of the detected coordinate and real world (measured) coordinate.

From the result of unit testing, it is found that contour and polygon approximation is preferred for arena detection compared to Hough transformation, robot detection has maximum error of 0.86% in center position and 0.562% in heading position and ball detection has a maximum error of 0.4%.

TABLE III. COMPARISON OF DETECTED ROBOT COORDINATE AND REAL WORLD COORDINATE

ID	Robot Coordinate			
	Detected (px)	Real World (cm)	X error	Y error
1	(105 , 309)	(37 , 110)	1.492%	0.855%
2	(208 , 309)	(73.5 , 109)	0.318%	0.053%
3	(154.5 , 225.5)	(54.5 , 79.5)	0.145%	0.110%
4	(228.5 , 389.5)	(80.5 , 137.5)	0.016%	0.0213%
5	(158.5 , 469)	(56 , 165.5)	0.303%	0.017%

TABLE IV. COMPARISON OF DETECTED HEADING COORDINATE AND REAL WORLD HEADING COORDINATE

ID	Robot Heading			
	Detected (px)	Real World (cm)	X error	Y error
1	(105 , 294)	(37 , 103.5)	0.040%	0.0562%
2	(209 , 294)	(73.5 , 103.5)	0.160%	0.0562%
3	(148 , 212)	(52 , 74.5)	0.252%	0.234%
4	(234 , 375)	(82.5 , 132)	0.092%	0.067%
5	(148 , 212)	(52 , 74.5)	0.252%	0.234%

TABLE V. THE DETECTED BALL COORDINATE COMPARED WITH THE REAL WORLD COORDINATE

Ball coordinate			
Detected (px)	Real World (cm)	X error	Y error
(194 , 123)	(68.5 , 43.5)	0.242%	0.401%

Integration testing examines the accuracy of data acquired from combined information such as global orientation of robots which is calculated from the combination of robot coordinate and heading coordinate; angle of rotation to face the ball which is calculated from the combination of robot coordinate, heading coordinate, and ball coordinate; and robot to ball distance which is calculated from the combination of robot coordinate and ball coordinate. The comparison of detected global orientation, ball orientation, and ball distance with the real world measurement are shown in Table VI, Table VII, and Table VIII, respectively.

TABLE VI. COMPARISON OF DETECTED GLOBAL ORIENTATION WITH MEASUREMENT IN REAL WORLD

ID	Global orientation (θ)		
	Detected (degree)	Real World (degree)	Error
1	90.0002105	90	0.0002 %
2	86.1860304	86	0.2163 %
3	115.709973	116	0.2500 %
4	69.2276256	70	1.103 %
5	91.9093058	91	0.999 %

TABLE VII. COMPARISON OF DETECTED ANGLE OF ROTATION TO FACE THE BALL WITH MEASUREMENT IN REAL WORLD

ID	Angle of rotation to face the ball (α)		
	Detected (degree)	Real World (degree)	Error
1	38.3131148 CW	38 CW	0.823 %
2	4.95861167 CW	5 CW	0.827 %
3	70.7098674 CW	70 CW	1.014 %
4	19.7092325 CCW	20 CCW	1.453 %
5	14.9362076 CW	15 CW	0.425 %

TABLE VIII. COMPARISON OF DETECTED ROBOT TO BALL DISTANCE WITH MEASUREMENT IN REAL WORLD

ID	Ball distance				
	Detected (px)	Detected (cm)	Detected times scale	Real World (cm)	Error
1	206.465	5.462719	72.86999	73.1	0.315 %
2	163.918	4.336997	57.85341	58.1	0.424 %
3	111.016	2.937298	39.18211	39.2	0.046 %
4	242.542	6.417257	85.60305	86.1	0.577 %
5	330.506	8.744637	116.6491	116.7	0.044 %

From the result of integration testing, it is found that the detected global orientation has maximum error of 1.1%, detected ball orientation has a maximum error of 1.45%, and detected ball distance has a maximum error of 0.57%.

System testing examines the effect of environmental and hardware changes on the accuracy and performance. Functionality failure found during this test will automatically be repaired by code modifications. The items to be tested are:

1) The effects of different camera resolutions on accuracy and performance.
2) The effects of different rooms on detection accuracy.
3) The effects of different environment lighting conditions on functionality and detection accuracy.
4) The effects of different camera position on detection accuracy.

From the result of system testing, the limitations of the developed application are found. The limitations consist of several things, such as: (i) 50% camera brightness is the minimal lighting condition in order for the algorithm to run smoothly, (ii) The heading detection will need adjustment in color segmentation parameter when there is a change in ambience or lighting condition in the environment, and (iii) If the floor is tiled, the arena detection needs readjustment in canny edge detection parameter as the parameter is not suitable for tiled floors. Moreover, the application has been tested to work best in low camera resolution (360p), and can be applied with four different camera positions

V. CONCLUSION

A method that utilizes high angle shot of a camera is proposed to detect multiple soccer robots positions and orientations. This method is an alternative way to bird eye angle method. To obtain the robots positions and orientation with this approach a series of process which includes camera positioning, perspective transformation, arena detection, robot detection, ball detection, and rotation angle and translation trajectory calculation is designed and implemented. Experiment results based on the designed physical arrangement demonstrate that the proposed method is able to obtain the robot positions and orientation with reliable success rate. The positions and orientations detected by the developed application of the proposed method can then be used in soccer robot to support hybrid control approach. With these acquired informations, strategy in soccer robot can be actualized.

The proposed method, however, has several limitations. The perspective transformation and image resizing cause slight pixel shifting which increases the error rate. Moreover, the algorithm takes longer time than bird eye angle method due to the process of arena detection and arena perspective transformation.

ACKNOWLEDGMENT

This research is funded by Indonesian Directorate of Higher Education, Kopertis Wilayah III, No.108/K3/KM/2015, and Faculty of Computer Science, Universitas Pelita Harapan, No.027/LPPM-UPH/III/2015.

REFERENCES

[1] M. A. Demetriou, “Centralized and decentralized policies for the containment of moving source in 2D diffusion processes using sensor/actuator network,” in *Proc. American Control Conference*, June 10-12, 2009.

[2] C. A. C. Parker and H. Zhang, “Collective unary decision-making by decentralized multiple-robot systems applied to the task-sequencing problem,” *Swarm Intelligent*, vol. 4, pp. 199-220, 2010.

[3] P. Dutkiewicz and M. Kielczewski, “Vision feedback in control of a group of mobile robots,” in *Proc. 7th International Conference Clawar*, 2005.

[4] N. Koiij, *The Development of a Vision System of Robotic Soccer*, Master’s thesis, University of Twente, 2003.

[5] Iowa University. (October 27, 2003). 55:148 digital image processing. [Online]. Available: http://user.engineering.uiowa.edu/~dip/LECTURE/Segmentation2.html #tracing

[6] L. Gool, M. Proesmans, and A. Zisserman, “Planar homologies as a basis for grouping and recognition,” *Image and Vision Computing*, vol. 16, pp. 21-26, 1998.

[7] G. Wolberg, “Perspective transformation,” in *Digital Image Warping*, Los Alamitos, Calif.: IEEE Computer Society Press, 1994.

[8] T. Yoshikawa, *Foundations of Robotics Analysis and Control*, Cambridge, Mass.: MIT Press, 1990.

Hendra Tjahyadi was born in Bandung, Indonesia in 1969. He received bachelor degree from Electrical Engineering Department, Maranatha Christian University, Indonesia, and Master Degree from Physics Engineering, Institute Technology Bandung, Indonesia in 1993 and 1996, respectively. He was graduated as Ph.D. in control engineering from Flinders University, Australia in 2006. He joined University Pelita Harapan in 2013. His research interests are adaptive control and robotics.

Giorgy Gunawan was born in Jakarta, Indonesia in 1993. He received the S.Kom degree from the University of Pelita Harapan (UPH), Indonesia, majoring in computer systems. He is currently an assistant lecturer in UPH. His research interests are image processing and robotics.

Arnold Aribowo was born in Semarang, Indonesia in 1976. He graduated from Electrical Engineering Department (Computer Engineering and Informatics Study Program), Universitas Diponegoro, Semarang, Indonesia when he received Bachelor Degree in 1999. He obtained Master Degree from Electrical Engineering (Computer and Informatics study program), Universitas Gadjah Mada, Yogyakarta, Indonesia in 2001.

He joined Universitas Pelita Harapan, Indonesia since 2002. His research interest is Artificial Intelligence. He teaches Discrete Mathematics, Artificial Intelligence System, and other courses.

David Hareva was born in Cimahi, Indonesia in 1972. He obtained under graduate from Computer Sciences of Universitas Padjajaran Bandung, Indonesia and continued his study in Bio-medical Engineering and Informatics at Okayama University, Okayama, Japan and received Master of Health Sciences (2006), and Doctor of Health Sciences (2009).

He worked at Universitas Pelita Harapan, Indonesia since 2011 with research interests in Computer Graphics, Medical Image processing, and Mobile Health Support System.