Location and Position Determination Algorithm for Humanoid Soccer Robot

Oei Kurniawan Utomo, Daniel Santoso, Saptadi Nugroho*
Electronic and Computer Engineering Faculty, Satya Wacana Christian University, Jalan Diponegoro 52 – 60 Salatiga 50711, (+62298) 311884
*Corresponding author, e-mail: saptadi_nugroho@yahoo.com

Abstract

The algorithm of location and position determination was designed for humanoid soccer robot. The robots have to be able to control the ball effectively on the field of Indonesian Robot Soccer Competition which has a size of 900 cm x 600 cm. The algorithm of location and position determination uses parameters, such as the goalpost’s thickness, the compass value, and the robot’s head servo value. The goalpost’s thickness is detected using The Centre of Gravity method. The width of the goalpost detected is analyzed using the principles of camera geometry to determine the distance between the robot and the goalpost. The tangent value of head servo’s tilt angle is used to determine the distance between the robot and the ball. The distance between robot-goalpost and the distance between robot-ball are processed with the difference of head servo’s pan angle and compass value using trigonometric formulas to determine the coordinates of the robot and the ball in the Cartesian coordinates.

Keywords: robot, soccer, location determination, robot position

Copyright © 2016 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

There are many techniques used to determine location of the robot and the ball on the field, for example: Voronoi based strategic positioning for robot soccer [1] and Monte-Carlo localization method applied in Multi-Cue localization for soccer playing humanoid robots [2]. The robot’s role on the field determination and its behavior has been researched by D.P. Playne in knowledge-based role allocation [3]. Similarly, Karen Petersen, Georg Stoll, and Oskar von Stryk developed a supporter behavior for soccer playing humanoid robots [4]. Hierarchical reactive control for humanoid soccer robot has been proposed by Sven Behnke and Jorg Stuckler [5]. The self-localization based on monocular vision for humanoid robot determines the robot’s location using a CCD camera [6]. Vision Based Self Localization for Humanoid Robot Soccer was implemented by Nuryono Satya Widodo and Arif Rahman [7] to create a robot soccer localization system. In this system, the robot’s location is used to take a decision whether or not the robot will kick the ball toward enemy team’s goal.

A soccer robot team must be able to play using a good strategy to win the game. Each robot must know the duty as a striker or a defender. The attacker is supposed to be on the attacking region position and scoring goal, while the defender is supposed to guard the defending region. The robot location and position determination becomes the problem in humanoid soccer robot. Therefore, the location and position determination algorithm is required so that the robot can know its duty and its working region.

In this paper we proposed an idea how to determine the coordinate of the robot and the ball in the field using the combination of the Centre of Gravity method and the Camera geometry analysis. The robot’s location in the Cartesian coordinate is used to create a strategy in the team. The goalpost that its coordinate has been known on the field is used as a landmark reference to determine the location of the robot on the field.

2. Research Method

2.1. Robot Control System

The robot control system shown in Figure 1 consists of three main parts: Android Smartphone as the main processor, Kondo-KHR-3HV (RCB4) as the robot’s servo controller,
and ATMega324. The ATMega324 is used as the controller of the head servo. It also gives orders to the KHR-3HV to call the robot's motion [8].

![Figure 1. The robot control system](image)

The Android Smartphone has the orientation sensor which has a yaw axis, pitch axis, and roll axis. The yaw axis is used as 360° compass as shown in Figure 2. The compass value of 0° is always set to the attack direction.

![Figure 2. The Android Smartphone compass](image)

The head servo consists of 2 servos: the pan servo and the tilt servo. The range value of pan and tilt servo is shown in Figure 3. The pan servo and the tilt servo which are used in the robot's head have 1° in precision.

![Figure 3. The robot's head servo: pan and tilt](image)

2.2. Camera Geometry

The principle of Camera Geometry is projecting real object from the 3D dimension world to the 2D dimension image plane. The farther an object from the camera the smaller an object projected on the image plane. The pinhole camera model shown in Figure 4 is the basis in camera image processing [9-10].
A straight line that goes through two points: \( Q_1 = (A_1, B_1, C_1) \) and \( Q_2 = (A_2, B_2, C_2) \) have the real length of \( D_{ab} \). The following perspective equation is used to calculate the projection length of \( d_{ab} \):

\[
d_{ab} = f \frac{D_{ab}}{C}
\]

Where
\( d_{ab} \) = the projection length of the line (cm)
\( f \) = the camera focus (cm)
\( D_{ab} \) = the real length of the line (cm)
\( C \) = the distance between camera focus and the line (cm)

2.3. The Robot Location on the Field

2.3.1. Goal detection

The algorithm of location and position determination uses information obtained from the image of the goalpost, the compass value, and the head pan value. The goalpost is chosen as a landmark reference because it can be seen by the robot from any position on the field. The goalpost detection is calculated using the Center of Gravity (COG) method [11-12] and Android OpenCV Library. The enemy team’s goal area and its own team’s goal area are determined using Android’s compass. The 0° of the compass value shows the direction toward the enemy’s goal area. The 180° of the compass value shows the direction toward the own team’s goal area. The right and the left goalpost are determined using the COG method. Figure 5 shows the left and the right goalpost’s COG value is located on the left side and the right side of the whole goalpost’s COG value respectively.

Figure 4. A line is projected to image plane using pinhole camera model

A straight line that goes through two points: \( Q_1 = (A_1, B_1, C_1) \) and \( Q_2 = (A_2, B_2, C_2) \) have the real length of \( D_{ab} \). The following perspective equation is used to calculate the projection length of \( d_{ab} \):

\[
d_{ab} = f \frac{D_{ab}}{C}
\]

Where
\( d_{ab} \) = the projection length of the line (cm)
\( f \) = the camera focus (cm)
\( D_{ab} \) = the real length of the line (cm)
\( C \) = the distance between camera focus and the line (cm)

2.3. The Robot Location on the Field

2.3.1. Goal detection

The algorithm of location and position determination uses information obtained from the image of the goalpost, the compass value, and the head pan value. The goalpost is chosen as a landmark reference because it can be seen by the robot from any position on the field. The goalpost detection is calculated using the Center of Gravity (COG) method [11-12] and Android OpenCV Library. The enemy team’s goal area and its own team’s goal area are determined using Android’s compass. The 0° of the compass value shows the direction toward the enemy’s goal area. The 180° of the compass value shows the direction toward the own team’s goal area. The right and the left goalpost are determined using the COG method. Figure 5 shows the left and the right goalpost’s COG value is located on the left side and the right side of the whole goalpost’s COG value respectively.

![Figure 5. The detection of the goalpost](image)
2.3.2. Robot-Goalpost Distance Measurement

A goalpost that has cylindrical shape will be projected as a rectangle on the camera’s image plane as shown in Figure 6. If the distance between the robot and the goalpost has a fixed value, the rectangle will have the same width seen by the robot from every position. The rectangle’s width and the distance between the robot and the goalpost can be calculated using the COG method and Android OpenCv.

\[ j_{tg} = f \frac{D_{tg}}{d_{tg}} \]  \hspace{1cm} (2)

Where
- \( j_{tg} \) = the distance between robot and goalpost (cm)
- \( f \) = the camera focus (cm)
- \( D_{tg} \) = the goalpost’s real width (cm)
- \( d_{tg} \) = the projected goalpost’s width on image plane (cm)

The goalpost’s detection process detects the width of the goalpost using pixel units. The projected goalpost’s width on image plane in Equation 2 uses centimeter units. Therefore, the conversion value between centimeter and pixel units is necessary.

The camera resolution which is used as the robot vision is 480 pixels in width. An object detected on the screen with a length of \( d_{ob} \) will use \( p_{ob} \) pixels which can be calculated using the following formula:

\[ p_{ob} = R \frac{d_{ob}}{L} \]  \hspace{1cm} (3)

Where :
- \( p_{ob} \) = the width of the projection of an object in the image plane (pixel)
- \( R \) = the screen resolution (pixels)
- \( d_{ob} \) = the width of objects detected on the screen (cm)
- \( L \) = the screen’s width (cm)

The camera focus of \( f \) has 7.2 cm in length. The goalpost’s real width (\( D_{tg} \)) is 10 cm. The screen resolution (\( R \)) is 480 pixels. The screen width (\( L \)) is 6.4 cm. The \( j_{tg} \) value can be calculated using Equation 2 and Equation 3 by substituting \( d_{ob} \) and \( p_{ob} \) with \( d_{tg} \) and \( p_{tg} \) respectively. The following equation is the result of substituting Equation 2 and Equation 3:

\[ j_{tg} = f \frac{D_{tg}}{(p_{tg} L / R)} = f \frac{D_{tg} R}{p_{tg} L} = \frac{5400}{p_{tg}} \]  \hspace{1cm} (4)
The robot’s Cartesian coordinate \((x_{rg} \text{ and } y_{rg})\) relative to a goalpost as shown in Figure 7 is calculated using the trigonometric equation. The coordinate of the robot position on the field can be calculated by translating the robot’s relative coordinate based on the detected goalpost.

![Diagram of robot's Cartesian coordinate relative to a goalpost](image)

Figure 7. The distance between the ball and the robot

2.4. The Ball Location on the Field

The distance between the ball and the robot is calculated to determine the position of the ball on the field as shown in Figure 8 and Figure 9. The distance between the robot and the ball \((j_{br})\) is calculated using the following tangent equation:

\[
j_{br} = \frac{h_r}{\tan(\gamma)} \quad \text{(robot's height is 47 centimeters)}
\]

\[
j_{br} = \frac{47}{\tan(\gamma)}
\]

Where

\(j_{br}\) = the distance between the ball and the robot (cm)

\(h_r\) = the robot’s height (cm) = (47 cm)

\(\gamma\) = the tilt value (degree)

![Diagram of ball location on the field](image)

Figure 8. The distance between the ball and the robot
The coordinate of the ball position on the field can be calculated by translating the ball’s relative coordinate using robot’s coordinate on the field as shown in Figure 9.

2.5. The Positioning Algorithm for Attacker and Defender Robot
The striker and the defender robots require location and position determination algorithms. The attacker robot is supposed to be on the attacking region position and scoring goal, while the defender robot is supposed to guard the defending region. This algorithm is supported by a communication process between the robots in a team which contained information about which robot is holding the ball. The attack condition is indicated when the striker robot holds the ball. The defense condition happens when the goalkeeper holds the ball.

2.5.1. The Positioning Attack Algorithm of Striker Robot
In attack condition the striker robot will move forward up to the half of the goalpost-ball distance. The robot’s movement is drawn using yellow arrow as shown in Figure 10.

2.5.2. The Positioning Algorithm of Striker Robot and Defender Robot
In defense condition the striker robot will retreat up to the center line of the field. This algorithm prevents striker robot loses the ball at the front of the field. In attack condition the defender robot will stay on the behind of the field’s center line. If the defender robot surpasses the line, it will retreat backward. Figure 11 shows the robots configuration.
3. Results and Analysis

3.1. The Location Determination Algorithm of the Robot

Table 1 shows that there are error values at coordinate determination of the robot. The integer pixel values ($p_{tg}$) and the error of $\alpha$ values cause the error of coordinate determination. Table 2 shows the robot-goalpost distance according to the pixel value calculated using Equation 4. The farther the ball from the robot the bigger error value of $j_{rg}$ obtained.

<table>
<thead>
<tr>
<th>Table 1. Robot’s Coordinate on The Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error value of $\alpha$ (°)</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>Average</td>
</tr>
<tr>
<td>Standard deviation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. The $j_{rg}$ and the pixel value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel value ($p_{tg}$)</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>24</td>
</tr>
</tbody>
</table>

Compared to field which has 900 cm x 600 cm in size, the error value of the robot’s coordinate is about 9.72% in x coordinate and 9.65% in y coordinate.

3.2. The Location Determination Algorithm of The Ball

Table 3 shows that there are error values on coordinate determination of the ball. This is happened because of the integer value of the tilt $\gamma$ and the errors of $\beta$ values. Table 4 shows robot-ball distance according to the tilt value calculated using Equation 8. The farther the ball from the robot the bigger error value of $j_{br}$ obtained.

<table>
<thead>
<tr>
<th>Table 3. The Ball’s Coordinate on The Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error value of $\beta$ (°)</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>Average</td>
</tr>
<tr>
<td>Standard deviation</td>
</tr>
</tbody>
</table>

Compared to field which has 900 cm x 600 cm in size, the error value of the ball’s coordinate is about 8.65% in x coordinate and 9.65% in y coordinate.
Table 4. The $j_{br}$ and the tilt value

<table>
<thead>
<tr>
<th>Tilt value (γ)</th>
<th>$j_{br}$ (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1345.9</td>
</tr>
<tr>
<td>4</td>
<td>672.13</td>
</tr>
<tr>
<td>6</td>
<td>447.17</td>
</tr>
<tr>
<td>8</td>
<td>334.42</td>
</tr>
<tr>
<td>10</td>
<td>266.55</td>
</tr>
<tr>
<td>12</td>
<td>221.11</td>
</tr>
<tr>
<td>14</td>
<td>188.5</td>
</tr>
<tr>
<td>16</td>
<td>163.9</td>
</tr>
<tr>
<td>18</td>
<td>144.65</td>
</tr>
<tr>
<td>20</td>
<td>129.13</td>
</tr>
</tbody>
</table>

The error value of the ball’s coordinate is about 10.39% in x coordinate and 10.13% in y coordinate compared to field size.

3.3. The Positioning Algorithm for the Striker and Defender Robot

Table 5, table 6, and table 7 show the position determination algorithm error values of the striker and the defender robot. The positioning attack algorithm of the striker robot shown in Table 5 is affected by the robot location determination error values shown in Table 1 and the ball location determination error values shown in Table 3. The robot location determination error values shown in Table 1 affect the positioning attack and positioning defense algorithm of the striker and the defender robot shown in Table 6 and Table 7.

Table 5. The Positioning Attack Algorithm of Striker Robot

<table>
<thead>
<tr>
<th>Error value of coordinate (cm)</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>24.6</td>
<td>27.64</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>13.45</td>
<td>9.91</td>
</tr>
</tbody>
</table>

Table 6. The Positioning Defense Algorithm of Striker

<table>
<thead>
<tr>
<th>Error value of coordinate (cm)</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>26.7</td>
<td>54.02</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>17.32</td>
<td>20.98</td>
</tr>
</tbody>
</table>

Table 7. The Positioning Algorithm of Defender

<table>
<thead>
<tr>
<th>Error value of coordinate (cm)</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>25.94</td>
<td>14.08</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>38.12</td>
<td>16.18</td>
</tr>
</tbody>
</table>

3.4. Computational Time

The computational time for executing the algorithm is measured using Android Logcat as shown in table 8 and table 9. The algorithm is run on 1 GHz processor and 512 RAM Android Smartphone.

Table 8. Computational Time for executing Robot Location Determination Algorithm

<table>
<thead>
<tr>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
</tr>
<tr>
<td>Standard deviation</td>
</tr>
</tbody>
</table>
4. Conclusion

The humanoid soccer robot uses the algorithm of location and position determination to determine location and position of the robot and the ball on the field. The algorithm has the error tolerance values which are caused by the integer of pixel value, the integer of tilt value, and the error values of the difference between the compass and the pan value. The farther the ball from the robot the bigger error value of $j_{rg}$ and $j_{br}$ obtained. The location determination's error values affect the position determination algorithm.

The location and position determination algorithm has 15% error value in average of robot-goalpost distance, while the robot-ball distance has 27.94% error value in average because the pixel values and the tilt values are integer. The average error value of robot-goalpost’s pan-compass difference is $5.74^\circ$. The average error value of robot-ball’s pan-compass difference is $5.48^\circ$. Compared to field size, the error value of the robot’s coordinate is about 9.72% in x coordinate and 9.65% in y coordinate. The error value of the ball’s coordinate is about 10.39% in x coordinate and 10.13% in y coordinate compared to field size. The computational time required to determine the location of the robot has 85.33 milliseconds in average. The average of computational time required to determine the location of the ball is 72.67 milliseconds.

The number of pixels which represent the projected goalpost’s real width in image plane depends on the resolution of the camera. If the camera has high resolution then the tolerance error value of the robot and the goalpost distance will be reduced because there are many pixels used. The distance between the ball and the robot is represented by the tilt value. The servo which has higher precision of encoder will reduce the tolerance error value of the ball and the robot distance.

References


