

Depth Image Assisted Aim for Scoring Goal in Wheeled Soccer Robot

Hendra Kusuma
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
hendraks@its.ac.id

Tasripan
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
tasripan@its.ac.id

Rudy Dikairono
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
rudydikairono@its.ac.id

Dzulfikar Ahmad Samhan
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
dzulfikar.ahmad.smhn@gmail.com

Abstract— This research addresses the development of an advanced decision-making system for a wheeled soccer robot, focusing specifically on the critical task of determining the optimal aiming angle for goal scoring. The primary objective is to enhance the robot's ability to accurately identify and target unguarded areas of the goal. To achieve this, a novel approach is employed, which integrates depth and RGB image data to predict the position of unguarded spaces within the goal area. This prediction is facilitated by the use of the Intel Realsense D435i depth camera, which captures both RGB and depth images simultaneously.

The methodology involves processing the combined image data to estimate the location of unguarded areas, subsequently converting these locations into precise aiming angles for the robot. The effectiveness of this technique has been rigorously tested across 60 different test points. The results demonstrate a high level of accuracy, with the system successfully predicting the unguarded area in all test cases and achieving an average error rate of only 1.3% in the coordinates predicted.

This research not only proves the feasibility of using integrated image data for robotic decision-making in dynamic sports environments but also sets the groundwork for further improvements in autonomous robotic interactions in complex, real-world settings.

Keywords— *aiming system, depth image, wheeled soccer robot.*

I. INTRODUCTION

Robotic technology has increasingly become an integral part of human lives, enhancing various sectors including medicine [1], [2], social interactions [3],[4], and industrial processes [5],[6]. Recent advancements in artificial intelligence have enabled robots to perform complex tasks traditionally exclusive to humans [3]. This progression has not only improved the capabilities and efficiency of robots but has also sparked excitement among robotics enthusiasts worldwide.

This enthusiasm has led to the creation of various global robotics events, among which ROBOCUP stands out. ROBOCUP features multiple divisions of robot competitions, with the Middle Size League (MSL) being its premier division in robotic soccer. These events not only showcase the cutting-edge developments in robotic technology but also promote innovation and collaboration among experts in the field.

The Middle Size League (MSL) is a competition featuring autonomous wheeled soccer robots. These robots utilize multiple motors as their primary movement mechanism. The autonomous nature of these robots presents significant challenges for robotics enthusiasts who are tasked with developing sophisticated decision-making capabilities. These capabilities enable the robots to determine their actions dynamically, based on real-time environmental inputs.

One of the main challenges in developing these soccer robots is designing an effective goal-scoring mechanism. Key aspects that need to be considered include the shooting mechanism, which enables the robot to kick the ball, and the aiming mechanism, which determines the direction of the shot. Both components are crucial for the robot's performance in accurately targeting and scoring goals, necessitating precise engineering and integration.

This research will focus on the aiming mechanism of soccer robots, which is critical for successful goal scoring. The primary objective of this mechanism is to identify unguarded spaces within the goal area. These identified spaces then serve as references for the system to determine the most effective aiming point. By precisely targeting these unguarded regions, the robot can significantly increase its chances of scoring.

Previous research on goal detection for Middle Size League (MSL) robots has predominantly relied on RGB color detection [7], [8]. While these studies have yielded satisfactory results, their reliance on color can lead to inaccuracies under varying lighting conditions. To address this limitation, this research proposes the use of a depth camera to enhance goal detection accuracy by incorporating depth information. The Intel Realsense D435i has been selected for this purpose, as prior studies have confirmed its effectiveness in accurately perceiving distances [9]–[11]. By integrating depth data with RGB, this approach aims to create a more robust and reliable goal detection system, less susceptible to environmental changes.

The research will utilize a combination of depth and RGB imagery to accurately identify the region of interest within the goal. These images will provide a comprehensive view of the goal area, allowing for detailed analysis. Using this data as a foundation, the system will employ a straightforward elimination process to pinpoint the unguarded areas of the goal. This method systematically excludes regions blocked by

defenders or other obstacles, ensuring that the remaining areas are optimal targets for scoring.

The remainder of this paper is organized as follows: Section 2 provides an overview of the devices and terminology used, as well as a detailed description of the overall system employed in this research. Section 3 presents the results obtained from various tests conducted during the study. Finally, Section 4 offers conclusions drawn from the research findings and discusses potential avenues for future work.

II. METHODOLOGY

A. IRIS Robot

This research will be implemented using a robot developed by the IRIS team from the Sepuluh Nopember Institute of Technology, which competes in the Middle Size League (MSL). The robot as shown in Fig. 1, is equipped with four omnidirectional wheels as its main actuators, allowing it to move in any direction across the ground surface. For shooting, it utilizes a high-torque brushless DC motor, enabling powerful shots. Its frame, constructed from stainless steel, is designed to withstand collisions with other robots, enhancing its durability in competitive settings. Additionally, the robot is equipped with an omnidirectional camera mounted on top of its frame. This camera is essential for capturing the surrounding environment, facilitating robust decision-making processes based on visual data.



Fig. 1. IRIS Robot

For the programming framework, the IRIS team utilizes the Robot Operating System (ROS) to manage all processes within the robot [12]. ROS is a robust software framework designed for robotic applications, capable of supporting simultaneous multi-threaded processes. It comprises a diverse array of tools, which provides extensive functionality to developers. A notable feature of ROS is its multilingual capability, allowing it to support multiple programming languages within a multi-threaded environment. Additionally, as an open-source platform, ROS facilitates easy development and customization by the global research and development community [13].

To effectively comprehend its surroundings, the robot requires the capability to estimate the positions of nearby objects. For this purpose, a fixed coordinate system is essential to serve as a reference. Fig. 2 illustrates the coordinate system adopted by the IRIS robot. In this system, the field length is represented by the Y-axis, while the field

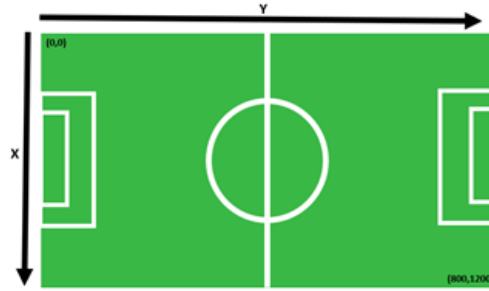


Fig 2. Field Coordinate System

width is represented by the X-axis. Throughout this research, this specific arrangement will be referred to as the field coordinate system.

To enhance the robot's ability to accurately determine the position of objects, it is crucial to estimate its own location within the field. This estimation is achieved using rotary encoders attached to the bottom of the robot. Fig. 3 illustrates the placement of these encoders. Each encoder is paired with an omnidirectional wheel, allowing the robot to calculate its current position based on displacement from an initial reference point. The displacement is derived from the number of rotations recorded by the rotary encoder, as specified in Equation (1).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta + 45^\circ) & \cos(\theta + 135^\circ) \\ \sin(\theta + 45^\circ) & \sin(\theta + 135^\circ) \end{bmatrix} \begin{bmatrix} RE0 \\ RE1 \end{bmatrix} \quad (1)$$

- (x,y) : estimated robot position
- RE0 : displacement of left rotary encoder
- RE1 : displacement of right rotary encoder

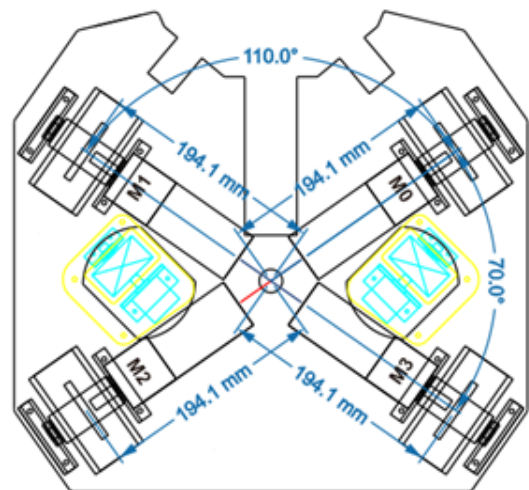


Fig. 3. Rotary Encoder Placement (Green Part)

B. Intel Realsense D435i

The choice of depth camera is critical for the success of this research, as it significantly influences the outcome. The camera must meet several specific requirements to function effectively within the system. First, it must have a fast capture rate; the minimum required is 30 frames per second to ensure synchronization with the robot's primary camera. Second, the camera must be capable of adapting to various



Fig. 4 Intel Realsense D435i

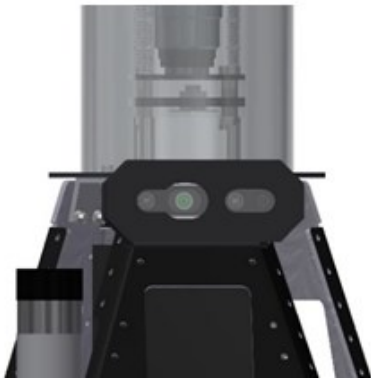


Fig. 5 Depth Camera Placement

lighting conditions. This is crucial due to potential variations between the lighting in the test environment and the actual event venue. Finally, the camera needs to be compact enough to be integrated seamlessly into the IRIS robot's design.

Based on these criteria, the Intel Realsense D435i (Fig. 4) was selected as the depth camera for this research. It offers a high capture rate of up to 90 frames per second, far exceeding the minimum requirement of 30 fps. This capability ensures optimal synchronization with the robot's other cameras. Additionally, its design is well-suited for both indoor and outdoor environments, addressing the need for adaptability under different lighting conditions. Furthermore, its compact chassis is ideal for integration, fitting neatly within the IRIS robot's frame.

This depth camera (Fig. 5) differs from others due to its inclusion of a secondary sensor that enhances its ability to perceive its surroundings. This sensor is a 6-DOF (Degrees of Freedom) Inertial Measurement Unit (IMU), which aids in accurately detecting motion and orientation. The coordinate system utilized by the IMU is detailed in Fig. 6 and will be referred to throughout this research as the camera coordinate system.

C. System Overview

The architecture of the research system encompasses several interconnected processes, collectively depicted in Fig. 7. At the heart of the system lies the image processing block. This crucial component processes both depth and RGB images captured by the depth camera to identify unguarded areas within the goal. The output of this block is the pixel coordinates of these areas. These coordinates are then forwarded to the coordinate converter block, which computes the aiming angle necessary for the robot to target these unguarded zones effectively.

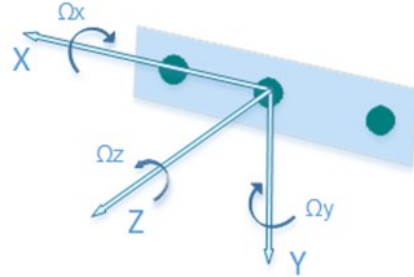


Fig. 6 Intel Realsense D435i Coordinate System [11]

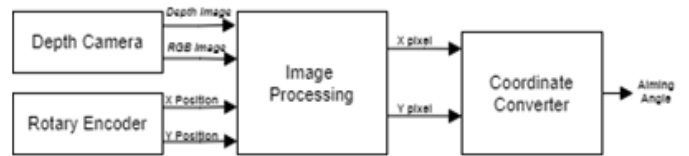


Fig. 7 Block Diagram of System

D. Depth Camera Block

Prior to initiating the image processing algorithm, the raw images captured by the depth camera require preprocessing. This step is necessary due to the differing fields of view between the depth and RGB images. As illustrated in Fig. 8, objects appear smaller in the depth image than in the RGB image, a discrepancy resulting from their varied fields of view. This disparity leads to inconsistent object coordinates across the two image types. To address this issue, the images must be aligned to ensure they share the same field of view. Fig. 9 demonstrates the outcome of this alignment, showing that objects now appear the same size in both images. This alignment facilitates synchronized coordinates between the depth and RGB images, essential for accurate further processing.

E. Image Processing Algorithm

To estimate the position of the unguarded area, the system processes the image data obtained from the depth camera. This processing involves a series of steps, each designed to identify and analyze the relevant sections of the image where no obstacles or defenders are present. The detailed workflow of this image processing sequence is depicted in Fig. 10.

The primary goal of the image processing stage is to enhance the recognition of the goal and the goalkeeper based on color and distance from the robot. To achieve this, the RGB image in Fig. 11 is employed to filter objects by color, identifying those that match the typical colors of the goal and the goalkeeper. The results of this color-based filtering process are depicted in Fig. 12 and Fig. 13. Specifically, Fig. 12 isolates objects in white, which are indicative of the goal,

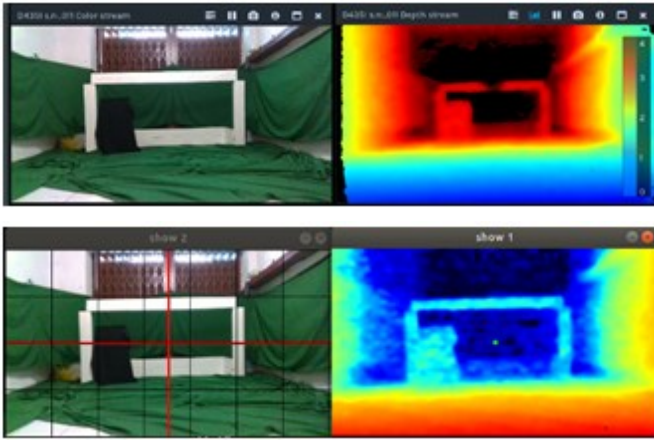


Fig. 11 Diffrence of Field of View

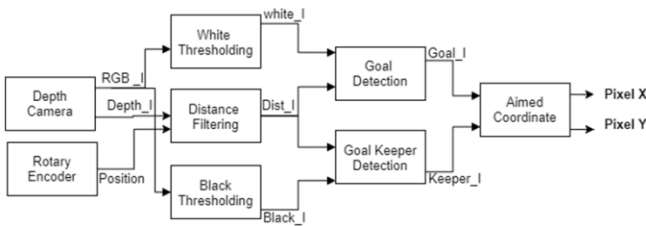


Fig 12 Image Processing Block Diagram

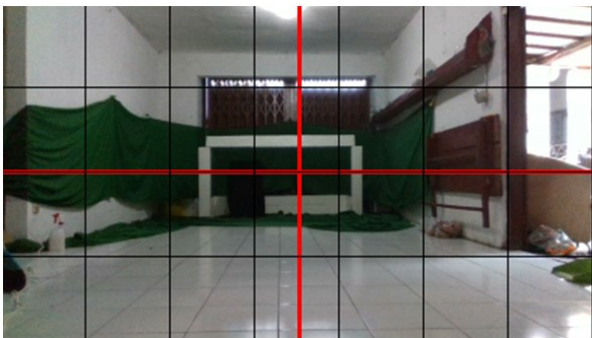


Fig. 13 RGB Goal Image

while Fig. 13 focuses on objects in black, corresponding to the goalkeeper.

As previously noted, this research utilizes both RGB and depth images to enhance goal recognition. The system extracts distance information from the depth image, which complements the color data from the RGB image. Unlike the RGB image, which is filtered by color, the depth image is filtered based on distance values of each pixel. This selective filtering process isolates objects within a specific range of distances, determined by the relative positions of the robot and the goal. The method for calculating this distance range is outlined in Equation (2). Fig. 14 displays the result of this filtering, showing only objects that fall within the predetermined distance range.

$$rg = \sqrt{(xr - xg)^2 - (yr - yg)^2} \pm offset \quad (2)$$

- (xr, yr) : robot coordinate
- (xg, yg) : goal coordinate
- rg : distance refrence
- offset : variable to control range.



Fig. 8 Goal Thresholding (White Threshold)

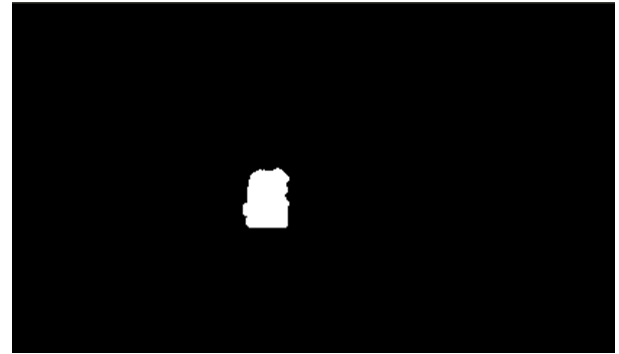


Fig. 9 Goal Keeper Thresholding (Black Threshold)



Fig. 10 Distance Filtering

Although Figures 14 and 12 successfully isolate goal-related objects, they still contain a considerable amount of noise from other detected objects. To mitigate this, the system combines these images to reduce noise, as shown in Fig. 15. This combined image serves as the primary reference for goal recognition, facilitating more accurate detection of the goal position. However, identifying the unguarded area within the goal requires an additional step. This unguarded area is determined through a simple elimination process, which involves subtracting the area occupied by the goalkeeper from the total goal area. This method effectively isolates the unguarded sections of the goal, essential for accurate shot targeting.

The results of the goal recognition process are displayed in Fig. 16. In this image, a blue rectangle outlines the detected goal area, indicating where the goal is located. The goalkeeper is represented by a red object, highlighting its position within the goal. The unguarded area, where no obstacles or goalkeeper are present, is delineated by a green



Fig. 14 Combined Threshold

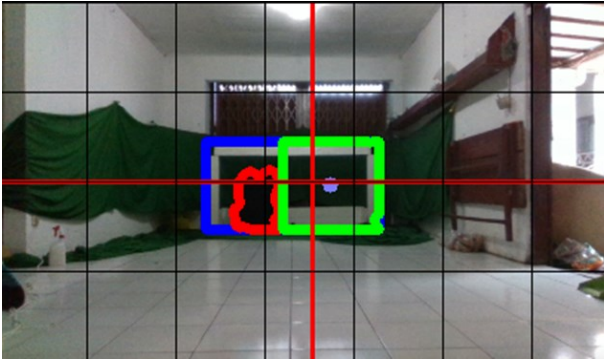


Fig. 15 Final Result

box. This visual representation clearly distinguishes between the goal, the goalkeeper, and the unguarded sections, allowing for accurate assessment and targeting during gameplay.

F. Coordinate Converter Block

The final output from the image processing block is the pixel coordinate of the unguarded area. To effectively utilize this data for targeting, these pixel coordinates must be converted into real-world coordinates. This conversion is achieved using the Intel Realsense function 'deproject pixel to point,' which translates pixel dimensions into spatial measurements.

Although the depth camera has the capability to calculate 3D coordinates, this system utilizes only the 2D coordinates corresponding to the field coordinate system, which comprises the X and Y axes. The values for X and Y are calculated using Equations (3) and (4). Employing these equations ensures that the aiming angle is confined within the goal area, thereby enhancing the accuracy of goal-targeting actions.

$$X = \left(\frac{A-B}{B-C} \right) + 300 \quad (3)$$

$$Y = d \quad (4)$$

(X, Y) : Estimated unguarded position

A : pixel coord. of unguarded area (Point A, Fig. 16)

B : left most part goal area (Point B, Fig. 16)

C : right most part goal area (Point C, Fig. 16)

D : nearest depth value

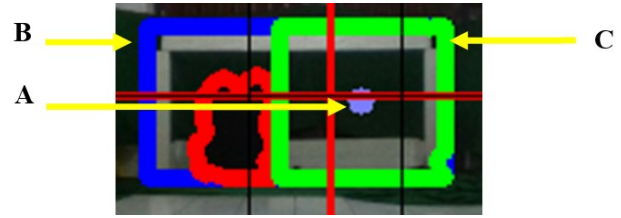


Fig. 16 Goal Point of Interest

III. TESTING AND RESULT

A. Coordinate Detection Test

The research includes several tests to verify the accuracy of the system, conducted in distinct phases. The initial phase involves a coordinate detection test. During this test, an object measuring 43 cm in width and 80 cm in height is placed at various locations within the testing room in front of the depth camera. The Intel Realsense camera then estimates the coordinates of the object, which are subsequently compared with the actual, measured coordinates to assess accuracy. Figs 18 and 19 display these comparisons between the real-world and estimated coordinates.

The results of this test indicate that the average error in the X-axis is 5.7%, and in the Y-axis, it is 1.49%. Both errors fall within the acceptable error margin of 10%, demonstrating the system's capability to reliably estimate positions within a tolerable range of accuracy.

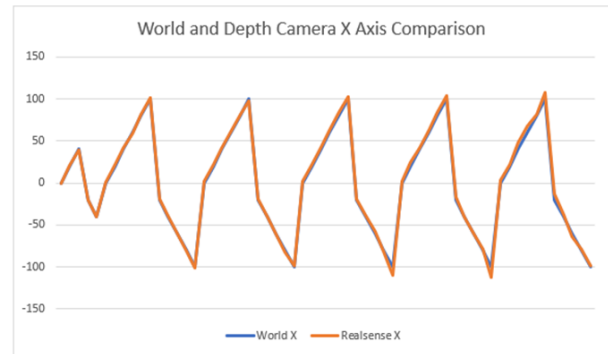


Fig. 17 Comparison Graph for X Axis

Fig. 19 demonstrates a noticeable increase in error for objects positioned more than 300 cm away from the camera. This observation is consistent with the specifications provided by the Intel Realsense D435i [14]. The increased error at greater distances aligns with the known limitations of the camera's sensing capabilities, as detailed in the manufacturer's documentation.

B. Width Measurement Test

As detailed in Part F of the methodology section, to accurately estimate the real-world coordinates of the unguarded goal area, the system compares the distances between the leftmost and rightmost points of the goal area. To validate the accuracy of this method, a specific test is conducted. In this test, the depth camera estimates the distance between the rightmost and leftmost coordinates of a single object, similarly to the coordinate detection test. The actual width of the object used in this test is 43 cm. Fig. 21 presents a comparison between the actual width and the width

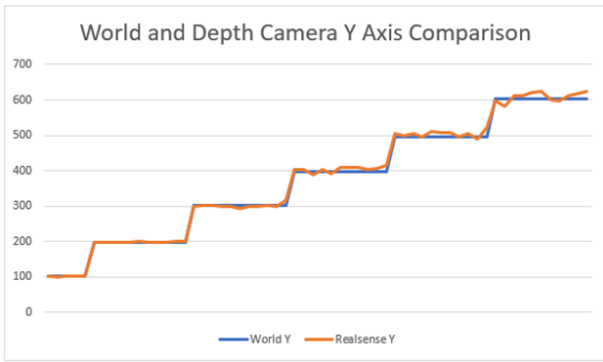


Fig. 18 Comparison Graph for Y Axis

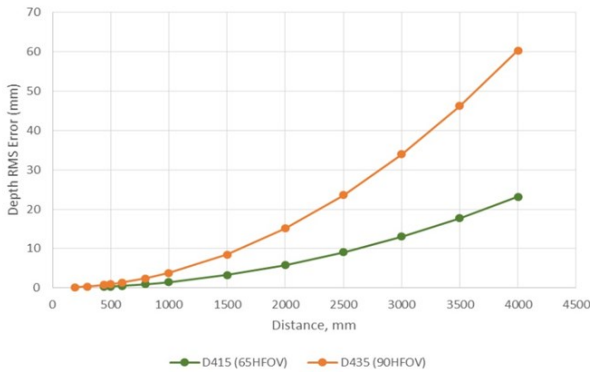


Fig. 19 RMS Error to Distance Graph [15]

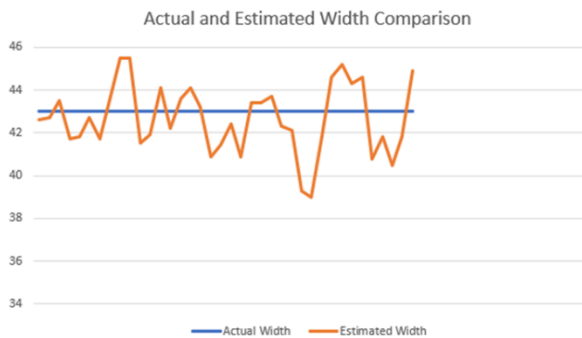


Fig. 20 Actual and Estimated Width Comparison

estimated by the camera. The results indicate that the average error in this test is 3.14%.

C. Unguarded Area Detection Test

This final test is crucial for confirming the overall functionality and accuracy of the system. In this test, the system is positioned at various locations around the testing area to detect the unguarded goal area. The unguarded areas detected by the system are then compared with the actual unguarded areas to compute the overall system accuracy. Additionally, this test evaluates whether the system successfully estimates the aiming angle. Success is defined by the estimated coordinates falling within the actual range of the unguarded area. To provide a clear representation of the test outcomes, Table 1 displays a selection of test data along with the results. The comprehensive data indicates that the average error across all tests is 1.3%, confirming that the system consistently and accurately identifies the position of unguarded areas from all test points.

Table 1 Final Test Result

Pos x	Pos y	Estimated X	real x	Left goal	right goal	Error (%)	Success
400	600	362.185	365	300	430	0.77	1
490	630	355.856	365	300	430	2.50	1
310	630	365.625	365	300	430	0.17	1
400	600	343.333	350	300	400	1.90	1
490	630	339.669	350	300	400	2.95	1
310	630	350.406	350	300	400	0.11	1
400	600	343.333	350	300	400	1.90	1
400	600	440.678	435	370	500	1.30	1
490	630	439.2	435	370	500	0.96	1
310	630	446.341	435	370	500	2.60	1
400	690	441.429	435	370	500	1.47	1
400	600	454.622	450	400	500	1.02	1
490	630	454.098	450	400	500	0.91	1
310	630	460.976	450	400	500	2.43	1

Pos x : System position in X axis
 Pos Y : System position in Y axis
 Estimated X : Estimated unguarded position in X axis
 Real X : Actual unguarded position in X axis
 Left/Right Goal : Unguarded area range
 Error : Diff. for actual and estimated position

III. CONCLUSION

The first series of tests validated the ability of the Intel Realsense D435i depth camera to estimate positions with an average error of 5.7% on the X-axis and 1.49% on the Y-axis, well below the acceptable error margin of 10%. These results affirm the reliability of the Intel Realsense D435i for positional estimation within the scope of this research.

The width measurement test assessed the reliability of the method used to calculate the unguarded area of the goal. With an average error of 3.14% in estimated width, the Intel Realsense D435i demonstrated high reliability, supporting the conclusion that the method for calculating the unguarded goal area is dependable and suitable for application.

The final test aimed to measure the accuracy of the entire system, evaluating both the error in estimated coordinates and the system's success rate. The results revealed an impressively low average error of 1.3% in coordinate estimation and a 100% success rate. The perfect success rate can be attributed to the relatively wide unguarded area, which ensures that even with some coordinate estimation errors, the targeted coordinates still fall within the actual range. These findings confirm that the system operates effectively as intended.

In summary, the conducted tests demonstrate that the overall system is highly accurate and reliable for its intended application, confirming the efficacy of the Intel Realsense D435i and the methodologies employed in this research.

IV. FUTURE RESEARCH

The research has employed a novel method to calculate the position of the unguarded area within the goal, which has proven to be relatively effective. However, there is considerable potential for enhancement. A primary limitation of the current approach is its dependence on accurately detecting the rightmost and leftmost parts of the goal to estimate the position of the unguarded area. This dependency could lead to inaccuracies under certain conditions where these extremities are obscured or not distinctly visible. To

advance the reliability and robustness of the research, developing a new method that does not rely on the goal's range is essential. Such an approach could potentially yield more consistent and accurate results across a variety of scenarios, thereby improving the overall effectiveness of the system.

REFERENCES

- [1] S. E. Butner and M. Ghodoussi, "Transforming a surgical robot for human telesurgery," *IEEE Trans. Robot. Autom.*, vol. 19, no. 5, pp. 818-824, Oct. 2003, doi: 10.1109/TRA.2003.817214.
- [2] A. A. Morgan, J. Abdi, M.A.Q. Syed, "Robots in Healthcare: a Scoping Review." *Current Robotics Reports* 3, 271–280 (2022). <https://doi.org/10.1007/s43154-022-00095-4>
- [3] K. Severinson-Eklundh, A. Green, and H. Hüttenrauch, "Social and collaborative aspects of interaction with a service robot," *Robot. Auton. Syst.*, vol. 42, no. 3-4, pp. 223-234, Mar. 2003, doi: 10.1016/S0921-8890(02)00377-9.
- [4] M. J. Matarić, "Socially assistive robotics: Human augmentation versus automation". *Sci. Robot.*, *eaam5410* (2017). DOI :10.1126/scirobotics.aam5410
- [5] B. Singh, N. Sellappan, and P. Kumaradhas, "Evolution of Industrial Robots and their Applications," *Int. J. Emerg. Technol. Adv. Eng.*
- [6] P. Cano, Y. Tsutsumi, C. Villegas, and J. Ruiz-del-Solar, "Robust Detection of White Goals," in *RoboCup 2015: Robot World Cup XIX*, Cham, 2015, pp. 229-238.
- [7] M. Hägele, K. Nilsson, J.N. Pires, R. Bischoff. "Industrial Robotics". In: Siciliano, B., Khatib, O. (eds) *2016 Springer Handbook of Robotics*. Springer Handbooks. Springer, Cham. https://doi.org/10.1007/978-3-319-32552-1_54
- [8] A. K. Mulya, F. Ardilla, and D. Pramadihanto, "Ball tracking and goal detection for middle size soccer robot using omnidirectional camera," in *2016 International Electronics Symposium (IES)*, Denpasar, Indonesia, Sep. 2016, pp. 432-437, doi: 10.1109/ELECSYM.2016.7861045.
- [9] M. S. Ahn et al., "Analysis and Noise Modeling of the Intel RealSense D435 for Mobile Robots," in *2019 16th International Conference on Ubiquitous Robots (UR)*, Jeju, Korea (South), Jun. 2019, pp. 707-711, doi: 10.1109/URAI.2019.8768489.
- [10] J. D. Mejia-Trujillo et al., "KinectTM and Intel RealSenseTM D435 comparison: a preliminary study for motion analysis," 2019, p. 4.
- [11] S. Sayyar-Roudsari, S. A. Hamoush, T. M. V. Szeto, and S. Yi, "Using a 3D Computer Vision System for Inspection of Reinforced Concrete Structures," in *Advances in Computer Vision*, vol. 944, K. Arai and S. Kapoor, Eds., Cham: Springer International Publishing, 2020, pp. 608-618.
- [12] A. R. Tinkar et al., "Team Description Paper: IRIS Team 2020," 2020, p. 8.
- [13] M. Quigley et al., "ROS: an open-source Robot Operating System," p. 6.
- [14] "How to Getting IMU Data from D435i and T265," Intel RealSense, [Online]. Available: <https://www.intelrealsense.com/how-to-getting-imu-data-from-d435i-and-t265/>.
- [15] A. Grunnet-Jepsen, J. N. Sweetser, and J. Woodfill, "Best-Known-Methods for Tuning Intel® RealSense™ D400 Depth Cameras for Best Performance," p. 10.