

Improving the speed of ball detection process and obstacle detection process in ERSOW robot using omnidirectional vision based on ROS

Muhammad Abdul Haq, Iwan Kurnianto Wibowo, Bima Sena Bayu Dewantara

Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Indonesia

Article Info

Article history:

Received Nov 15, 2020

Revised Mar 4, 2021

Accepted Mar 28, 2021

Keywords:

Computer vision

Omnidirectional camera

Optimization

ROS

Threading

ABSTRACT

This paper presents a novel approach for improving the computation speed of the ball detection and obstacle detection processes in our robot. The conditions of obstacle detection and ball detection in our robot still have a slow processing speed, this condition makes the robot not real-time and the robot's movement is hampered. To build a good world model, things to note are obstacle information and real-time ball detection. The focus of this research is to increase the speed of the process of the ball and obstacle detection around the robot. To increase the speed of the process, it is necessary to optimize the use of the OpenCV library on the robot operating system (ROS) system to divide the process into several small processes so that the work can be optimally divided into threads that have been created. Then, minimize the use of too many frames. This information will be sent to the base station and also for obstacle avoidance purposes.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Muhammad Abdul Haq

Department of Informatics and Computer Engineering

Politeknik Elektronika Negeri Surabaya

Kampus PENS, Raya ITS St., Keputih, Sukulilo, Surabaya, Jawa Timur 60111, Indonesia

Email: muhabdulhaq@gmail.com

1. INTRODUCTION

Nowadays, the development of computer technology and robotics are increasingly advanced with proof of some progress. One tangible form of advances in computer technology is the number of robots that are getting smarter has a minimal error, from robots that can facilitate human work to robots that function as entertainment. The number of robot competitions also triggers robot activists to always make updates and add features to the robot so that the robot technology becomes more advanced. Soccer is considered one of the most popular sports of all time in the world [1]. Because of its immense potential, nowadays, automation of soccer has been widely researched, especially in the robotic field [1]. The RoboCup competitions take place once a year and they are divided into different leagues, one of them being the soccer league. Within the soccer league, one of the most dynamic and challenging competition is the middle size league [2]-[5]. RoboCup middle size league is an international level competition that competes for several robots to play soccer from various countries. In Indonesia, there is a similar competition, namely the Indonesian Wheeled Robot Soccer Contest, where the rules are the same as the RoboCup middle size league taking into account conditions and developments in Indonesia such as field size and other factors [6]. Robot soccer is a robot that is designed to be able to play soccer autonomously [7]. So, the robots can play like a soccer game in general, soccer robots must be equipped with features that can later support the running of the robot then the robot can make decision-making conditions. One of the basic features that robots must-have is real-time [8]-[10].

especially in ball detection and obstacle detection around the robot. With these features, the robot can make good strategies and movements, such as detect the ball near and far and can do obstacle avoidance. Being soccer a very dynamic scenario, the robots need to be able to perceive the other robots around them [11]-[14].

ERSOW is a robot team from Politeknik Elektronika Negeri Surabaya (PENS) that competes in the Indonesia Robot Soccer Contest Wheeled [6], [7]. The conditions of the obstacle detection system have a processing speed of 15 frame per second (fps) and the ball detection system still has a processing speed of 24 fps, while the speed of the ERSOW robot is 2 m/s [7]. CAMBADA is the Middle Size League soccer team of the University of Aveiro, Portugal [3]. Looking at the CAMBADA soccer robot which has a processing speed of up to 60 fps in detecting the ball with an average robot moving speed of 3 m/s [3], the ball detection system conditions on the ERSOW robot are still slow, and there are still many frames which are not used but are still being processed so the movement of the robot is not real-time because of the slow reception of object position data.

The related works that have been done so far are as follows, ERSOW is a robot team from Politeknik Elektronika Negeri Surabaya (PENS) that competes in the Indonesia robot soccer contest wheeled [6], [7]. The conditions of the obstacle detection system have a processing speed of 15 fps and the ball detection system still has a processing speed of 24 fps, while the speed of the ERSOW robot is 2 m/s [7]. CAMBADA is the robocup middle size league (MSL) soccer team of the University of Aveiro, Portugal [3]. Looking at the CAMBADA soccer robot which has a processing speed of up to 60 fps in detecting the ball with an average robot moving speed of 3 m/s [3], the ball detection system conditions on the ERSOW robot are still fairly slow, and there are still many frames which are not used but are still being processed so that the movement of the robot is not real-time due to the slow reception of ball position data.

The focus in this research is increasing the speed of the ball detection process and obstacle detection using an omnidirectional camera by optimizing the use of the OpenCV library on robot operating system (ROS) and minimizing the use of too many frames so that processing speed can be increased. This system runs using the ROS so that later it will facilitate the division of work systems created programs. After that, the ball and obstacle data obtained will be sent to the base station for further processing. Limitations of this study include the sensor used in this study is a camera with an omnidirectional mirror. The system used uses the robot operating system. Some things that make this study different from other studies are that all programs run simultaneously and are used for real-time cases. This paper is organized as follows. Section 2 describes the whole proposed systems design. Section 3 presents the experimental results and discussion. And finally, the conclusion and future development are written in section 4.

2. RESEARCH METHOD

In this chapter, we explain the system design planning and manufacturing system that consists of the stages carried out in the completion of this research to achieve maximum objectives. The vision system created on the ERSOW robot uses an omnidirectional camera. This omnidirectional camera is capable of capturing images at an angle of 360° around the center of the vertical axis of rotation without the need to move the camera or robot body [3], [15]-[17]. This certainly can speed up the object detection process when compared to using conventional cameras. An omnidirectional camera consists of a camera that faces the mirror. The shape of the mirror will affect the image produced by the camera [17]-[19]. In this research, the camera sensor used is the ELP high-speed camera with a 2.8-12 mm lens that can produce resolutions up to 1920x1080 (60 fps) and 1280x720 (120 fps). This research uses the OpenCV library to facilitate the working of vision systems in the ball detection and obstacle robot ERSOW detection program.

The frames processed in the ERSOW vision system are quite complex so that the frames used in the ball detection program and obstacle detection program are quite a lot. Therefore, the frame that is processed must be ideal and the area of the frame that is processed is not useful. Examples of useless frames in the initial omnidirectional camera are display data outside the field. Areas outside the field are not needed in soccer robot matches, because there are no landmarks outside the field, so displays outside the field are not required.

Figure 1 shows the system to be made as a whole. The robot takes pictures using an omnidirectional camera. The method of tracking the position of the colored balls consists of several sequential processes [6]. Beginning with capturing video with a camera and the other process. Then the robot can observe the image to recognize the ball and obstacle [20]-[22]. By using an image publisher, images taken from an omnidirectional camera will be forwarded to the ball node and obstacle node. To increase the speed of the process the most important thing to do is to break the program so that the processing load will be divided [23]. In this research the program is broken down into 3 nodes, there are omni_publisher, omni_ball, omni_obstacle.

In the ball node, the frame of the omnidirectional camera is converted to hue, saturation, value (HSV) color space. In this HSV image, the color will be distinguished where the orange color for the ball and compared to the contour of the color detection results so that it is known whether the ball is in the field or

outside. In the obstacle node, the image from the omnidirectional camera will also be changed to the HSV color space. In this HSV image, the color of the field will be detected and the results will be given a contouring process, thus producing an obstacle color detection result contour. Then compared with the contour of the color detection results in the field so it is known whether the obstacle is in the field or outside.

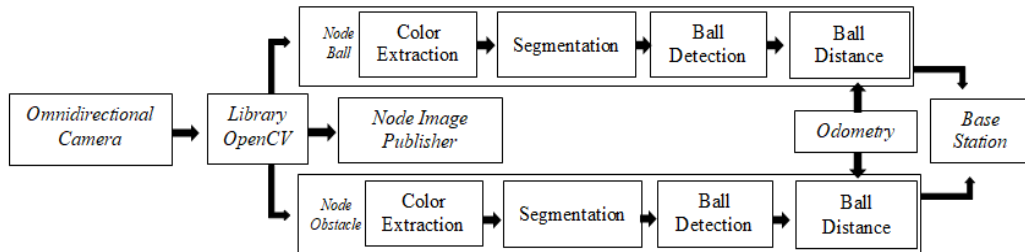


Figure 1. Diagram of the system

After the object is detected, the pixel distance data is changed to the actual distance. The actual distance is added to the odometry value of the robot so that the position of the object against the field can be known. The value of the position of the object is sent to the base station for further processing. This research is implemented on the ERSOW robot using the robot operating system (ROS). By implementing the ROS, the management of programs in the ERSOW robot becomes more organized and easier to troubleshooting. Before the implementation of ROS, the ball and obstacle detection program was still a unitary program. After the implementation of ROS, the ball detection program and obstacle detection program become separate and can facilitate program management.

The multi-node ROS capability can control many of the robot's operational functions. Robots that have a large task of carrying out many functions and sophisticated, the application of traditional methods will consume researchers' time and energy [24], [25]. With multi-node, the ROS property will provide efficient robot flow control. Furthermore, the process of building robot software does not take long because it can be done as a team.

In this system, three nodes are created namely the publisher node, the ball node, and the obstacle node. The division into 3 nodes is intended so that the process of the program does not burden the computer and also the memory management is evenly distributed. The function of the publisher node is to take pictures from an omnidirectional camera with a frame size of 1280x720, then the frame size is cut smaller to 640x640 so that computer vision processing becomes faster because the frame processed is not large. The output of this node is the frame that has been cut to 640x640 and sent with the message publisher on ROS. To increase the speed of the process of ball and obstacle detection, threading is one way for the program and its functions to run at the same time [23].

3. RESULTS AND ANALYSIS

ROS has a service as well as an operating system in general, including hardware abstraction, lower-level device control, implementation of commonly used functions, delivery of messages/data between processes, and management packages. ROS also provides tools and libraries that enable us to get, build, program, and run programs through multiple computers. By implementing the robot operating system, program management in the ERSOW robot becomes more organized and easier for troubleshooting. Before the implementation of ROS, the ball and obstacle detection program was still a unitary program. After the implementation of ROS, the ball detection program and obstacle detection program become separate and can facilitate program management.

That is because of the publisher's features on ROS, this publisher can send initial frame data from the camera to multiple nodes. In the case of the vision program in the ERSOW robot, there are 3 nodes including the omni_publisher, omni_ball, and omni_obstacle. The function of the omni_publisher is to take camera data into the initial frame, then send using the publisher to the omni_ball node and the omni_obstacle node. The omni_ball and omni_obstacle nodes receive data using the subscriber facility provided by ROS to receive data from the publisher. After implementing ROS in the ERSOW robot, the speed of the ball detection process increased by 36 fps, while the processing speed of the obstacle detection increased by 4.5 fps. For system testing, there are several parts which are:

3.1. Testing of ball and obstacle tracking

Before we did this research, we had researched ball detection using local binary pattern (LBP) [7]. However, the output provided is not satisfactory, for example, the FPS value on ball detection using the LBP method produces about 20 fps when used for a long time. It's different when we implement the color thresholding method, the value of the speed of the process can be increased faster. When compared to the speed of the process, ball detection using the color thresholding method is faster than the LBP method. The basic object that must be detected in robot soccer is ball and obstacles. The black robot with an orange ball is located on a green field with a white line. So, colors are the easiest instructions to process for segmenting objects. Color extraction is done in HSV color space. 3 colors must be detected, there are orange (ball), green (field), and black (obstacle).

Figure 2 is a display of the obstacle detection program, the increase in the speed of this process does not affect the obstacle detection results. So that it does not confuse the obstacle detection results of the ERSOW robot system. The segmentation process in HSV color space is done through the color threshold of each object. In the ERSOW vision system, the method used to detect the ball is to use color thresholding. The program detects the color orange, then the detection results are compared with the results of thresholding the field, whether the ball is on the field or not. For obstacle detection programs, there is only 1 color required for calibration, namely green (field). The segmentation process in the ros color space is carried out through a green threshold.

Figure 3 is a display of the ball detection program, the increase in the speed of this process does not affect the ball detection results. So as not to confuse the ball detection results from the ERSOW robot system. In the process of detecting the ball, it is checked on the nine field points around the ball. These points are obtained from the spherical coordinates detected. Detection is intended to find out whether the object is really in the field or not. In the obstacle detection program, only field colors, namely green, are used to calibrate, thus saving frame usage and can increase the speed of the obstacle detection process. The method used is the color thresholding by taking the contour from green, then the binary results from the green color are inverted using the bitwise function in the OpenCV library. The result is that colors other than green will turn white so that obstacles in the field that don't have green will be detected as obstacles. The obstacle is also compared to the result of the field threshold itself so that the robot knows it only detects obstacles in the field area.

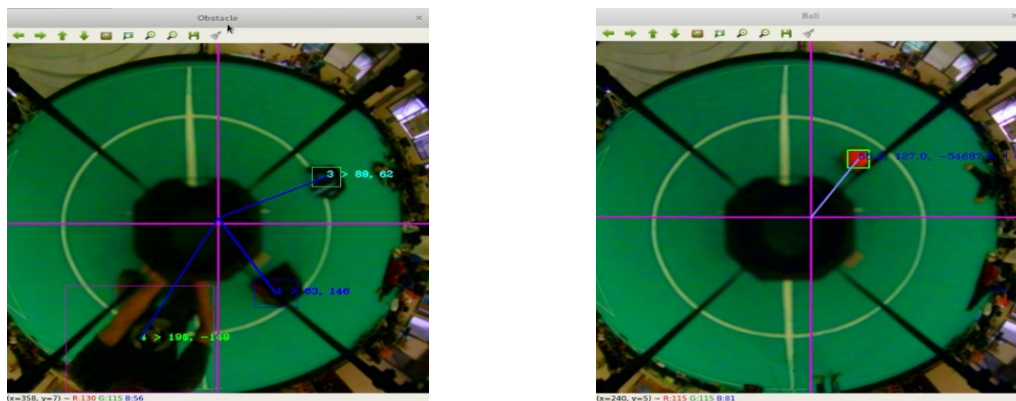


Figure 2. Obstacles detection program of ERSOW robot Figure 3. Ball detection program of ERSOW robot

3.2. Testing of ball detection's and obstacle detection's speed process

Time execution taking of some programs is depending on the algorithm of the program and statement time take [26]. At this stage, it is testing the system after optimizing the speed of the process for the ball detection program as well as the obstacle detection program. The following is a graph of increasing the speed of each program.

The graph in Figure 4 shows the effects of the ROS implementation process, the use of Region of Interest, and also the optimization of frame usage. In the implementation of ROS, an increase in the speed of 6 fps was obtained, for the use of region of interest (ROI), an increase in the speed of the ball detection process was 13.4 fps, while to optimize the use of the frame an increase in the speed of the ball detection process was 16.6 fps.

The graph in Figure 5 shows the effects of the ROS implementation process, the use of region of interest, binary inversion, and also the optimization of frame usage. In the implementation of ROS, an increase in the speed of 2 fps was obtained, for the use of ROI, an increase in the speed of the obstacle

detection process was 7.8 fps. For the effect of binary inverse using bitwise obtained an increase in the speed of the obstacle detection process by 7.6 fps while for the optimization of the use of frames obtained an increase in the speed of the obstacle detection process by 12.6 fps.

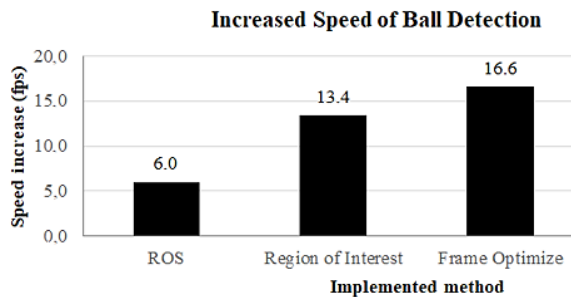


Figure 4. The graph of increasing speed process of ball detection in milliseconds

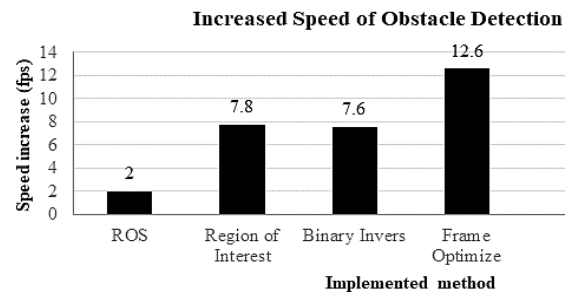


Figure 5. The graph of increasing speed process of obstacle detection in milliseconds

Figure 6 shows the difference in process speed in ball detection between before and after the research, for the stability of the processing speed obtained in the ball detection program, which is stable between the values of 58 and 57 fps. So that the speed of the ball detection process can be said to be stable with a value of around 58 fps. After that, there is also a graph of the process improvement from the obstacle detection program on the ERSOW robot.

Figure 7 shows the difference in processing speed in obstacle detection between before and after the research, for the stability of the processing speed obtained in the obstacle detection program, which is stable between the values of 45 fps and 46 fps. So that the speed of the process in obstacle detection can be said to be stable with a value of around 45 fps. In the previous ERSOW robot program, every time we run the program, the laptop temperature always rises from 60 degrees Celsius to 98 degrees Celsius because a lot of laptop memory is used. After increasing processing speed as well as optimizing frame usage, the laptop temperature may drop.

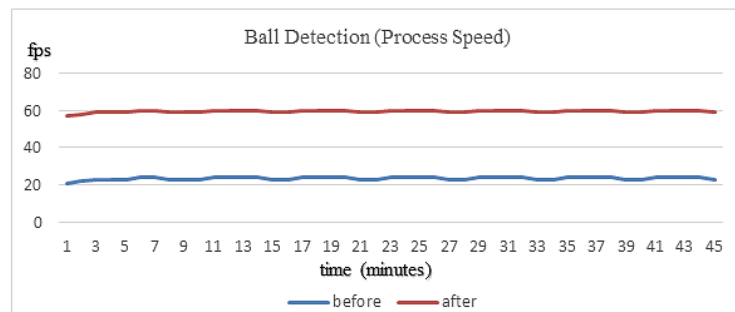


Figure 6. The difference in processing speed in ball detection between before and after the research

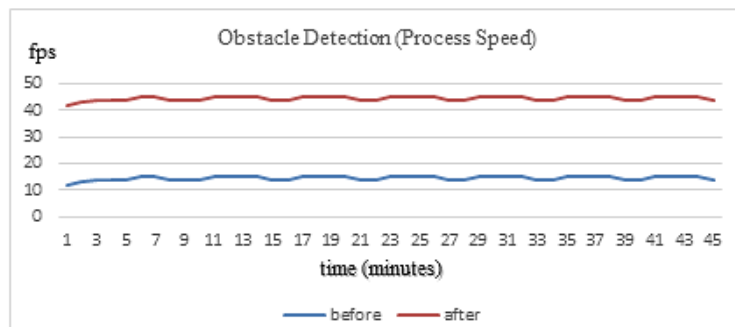


Figure 7. The difference in processing speed in obstacle detection between before and after the research

4. CONCLUSION

The focus in this research is increasing the speed of the ball detection process and obstacle detection using an omnidirectional camera by optimizing the use of the OpenCV library on ROS and minimizing the use of too many frames so that processing speed can be increased. Based on the results of tests and analyzes that have been done, it can be concluded that the speed of the process of detecting ball which was previously 24.3 fps earlier became faster that is 58.8 fps, while the speed of the process of detecting obstacles which were previously 14.9 fps became faster that is 45.4 fps. Also, the ball detection and obstacle detection programs have been successfully applied to the ERSOW robot using the robot operating system (ROS) and information from the ball and obstacle detection results can be sent to the base station. This research has been implemented in real-time on the original robot, and the speed of the ball detection and obstacle detection is good. For future works, the development of improving the speed of the ball detection process and obstacle detection will always be increased so the robot's movement is close to real-time. The scope that can be done in the future is about managing the use of memory resources used by the ERSOW robot.

ACKNOWLEDGEMENTS

Thank everyone who helped me to finish this research, especially the PENS Robotics Team for helping with this research from beginning to end. I hope this research can be useful for the development of middle size league robot research and the other field.

REFERENCES

- [1] N. Fitriana, K. Mutijarsa, and W. Adiprawita, "Color-based segmentation and feature detection for ball and goal post on mobile soccer robot game field," *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, pp. 1-4, 2016, doi: 10.1109/ICITSI.2016.7858232.
- [2] A. J. R. Neves, A. Trifan, P. Dias, and J. L. Azevedo, "Detection of Aerial Balls in Robotic Soccer Using a Mixture of Color and Depth Information," *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, pp. 227-232, 2015, doi: 10.1109/ICARSC.2015.13.
- [3] R. Dias, *et al.*, "CAMBADA'2016: Team Description Paper," *RoboCup 2016 Middle Size League*, 2016.
- [4] F. M. W. Kanters, *et al.*, "Tech united eindhoven team description 2011," in *RoboCup 2011 Middle Size League*, 2011.
- [5] U. P. Kappeler, *et al.*, "RFC Stuttgart Team Description," *RoboCup 2011 Middle Size League*, 2011.
- [6] M. A. Haq, *et al.*, "The Development of EEPIS RoboSoccer on Wheeled (ERSOW) 2019," *The Indonesian Symposium on Robot Soccer Competition 2019*, 2019.
- [7] I. K. Wibowo, M. A. Haq, M. M. Bachtiar, B. S. B. Dewantara, and F. L. H. Ihsan, "Ball Detection using Local Binary Pattern in Middle Size Robot Soccer (ERSOW)," *2019 2nd International Conference of Computer and Informatics Engineering (IC2IE)*, pp. 29-32, 2019, doi: 10.1109/IC2IE47452.2019.8940835.
- [8] J. Michels, A. Saxena, and A. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," *Proceedings of the 22nd international conference on Machine learning*, pp. 593-600, 2005, doi: 10.1145/1102351.1102426.
- [9] S. Lenser and M. Veloso. "Visual sonar: fast obstacle avoidance using monocular vision," *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)* (Cat. No.03CH37453), 2003, pp. 886-891, doi: 10.1109/IROS.2003.1250741.
- [10] H. Koyasu, J. Miura and Y. Shirai, "Real-time omnidirectional stereo for obstacle detection and tracking in dynamic environments," *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 31-36, doi: 10.1109/IROS.2001.973332.
- [11] R. Dias, N. Lau, J. Silva, and G. H. Lim, "Multi-object tracking with distributed sensing," *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 564-569, 2016, doi: 10.1109/MFI.2016.7849548.
- [12] A. Ahmad and P. Lima, "Multi-robot cooperative spherical-object tracking in 3D space based on particle filters," *Rob. Auton. Syst.*, vol. 61, no. 10, pp. 1084-1093, 2013, doi: 10.1016/j.robot.2012.12.008.
- [13] J. Silva, N. Lau, A. J. R. Neves, J. Rodrigues, and J. L. Azevedo, "Obstacle detection, identification and sharing on a robotic soccer team," *Progress in Artificial Intelligence*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 350-360, 2009, doi: 10.1007/978-3-642-04686-5_29.
- [14] M. Jamzad, "Object detection and localization using omnidirectional vision in the RoboCup environment," *Sci. iran.*, vol. 14, no. 6, 2007.
- [15] B. Cunha, J. Azevedo, N. Lau, and L. Almeida, "Obtaining the inverse distance map from a non-SVP hyperbolic catadioptric robotic vision system," *RoboCup 2007: Robot Soccer World Cup XI, Berlin*, 2008, pp. 417-424, doi: 10.1007/978-3-540-68847-1_43.

- [16] S. Lange, A. Voigtländer, M. A. Riedmiller, and M. Lauer, "Real-time 3D ball recognition using perspective and catadioptric cameras," *EMCR*, 2007.
- [17] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of the 7th international joint conference on Artificial intelligence*, vol. 2, pp. 674-679, 1981.
- [18] A. Ahmad, J. Xavier, J. Santos-Victor, and P. Lima, "3D to 2D bijection for spherical objects under equidistant fisheye projection," *Comput. Vis. Image Underst.*, vol. 125, pp. 172-183, 2014, doi: 10.1016/j.cviu.2014.04.004.
- [19] A. Neves, A. Pinho, and D. Martins, "A hybrid vision system for soccer robots using radial search lines," *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions Portuguese Robotics Open-ROBOTICA '2008*, 2008, pp. 51-556.
- [20] A. Rahman and N. S. Widodo, "Colored ball position tracking method for goalkeeper humanoid robot soccer," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 11, no. 1, pp. 11-16, 2013, doi: 10.12928/telkomnika.v11i1.877.
- [21] 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)*, pp. 432-437, 2016, doi: 10.1109/ELECSYM.2016.7861045.
- [22] D. Scaramuzza, S. Pagnottelli, and P. Valigi, "Ball detection and predictive ball following based on a stereoscopic vision system," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1561-1566, 2006, doi: 10.1109/ROBOT.2005.1570336.
- [23] A. K. Das, R. Fierro, V. Kumar, B. Southall, J. Spletzer, and C. J. Taylor, "Real-time vision-based control of a nonholonomic mobile robot," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 2002, doi: 10.1109/ROBOT.2001.932858.
- [24] O'Kane, Jason M, "A Gentle Introduction to ROS," *Columbia: University of South Carolina*, 2018.
- [25] A. Mahtani, L. Sanchez, E. Fernandez, and A. Martinez, "Effective robotics programming with ROS-third edition," 3rd ed. Birmingham, England: Packt Publishing, 2016.
- [26] Z.S. Al-Hashami, "Execution of C++ Programs in Shortest Possible Time (Nanoseconds or Miliseconds)," *International Journal of Technology Enhancements and Emerging Engineering Research*, vol 2, no. 5, 2016.

BIOGRAPHIES OF AUTHORS



Muhammad Abdul Haq, S.Tr.T, completed his undergraduate degree in Computer Engineering from Electronic Engineering Polytechnic Institute of Surabaya, Indonesia, in 2020. Currently, He is master student of the Electrical Engineering and Computer Science Area at Tokyo Metropolitan University. His research is often related to computer vision and machine learning. He has always participated in the Indonesian robot contest wheeled soccer division from 2017 until 2019 and won championships at the regional and national levels.



Iwan Kurnianto Wibowo, S.ST., M.T, is a lecturer in the D4 Computer Engineering Study Program - Engineering from Electronic Engineering Polytechnic Institute of Surabaya (PENS). Focus on research in the field of Embedded Systems and Robotics from 2011 until now. The author is also still active in developing student-level educational robots since 2010.



Bima Sena Bayu Dewantara received the B.Eng. degree in Information Technology from Electronic Engineering Polytechnic Institute of Surabaya, Indonesia, and the M.Eng. degree in Electrical Engineering from Sepuluh Nopember Institute of Technology, Indonesia, in 2004 and 2010, respectively. He obtained the Dr.Eng degree from Toyohashi University of Technology, Japan, in 2016. In 2005, he joined the Department of Electronic Engineering at Electronic Engineering Polytechnic Institute of Surabaya, Indonesia, as a lecturer. He moved to the Department of Informatics and Computer Science in 2007. Currently, He is an Assistant Professor in the major field of autonomous intelligent systems which covers pattern recognition, computer vision, machine learning and robotics system. He received The Best Achievement Award from Toyohashi University of Technology, Japan, in 2016 and also received some Best Paper Awards in 2016, 2018, 2019 and 2020.