

A vision-based real-time obstacle avoidance's rules utilising grid-edge-depth map

Budi Rahmani¹, Agus Harjoko², Tri Kuntoro Priyambodo³

¹Department of Informatics, STMIK Banjarbaru, Indonesia

^{2,3}Department of Computer Science, Universitas Gadjah Mada, Indonesia

Article Info

Article history:

Received Oct 17, 2019

Revised Dec 1, 2019

Accepted Jan 14, 2020

Keywords:

GED-map

Navigation

Obstacle avoidance

Robot

Stereo vision

ABSTRACT

This paper presents a new rules-based of a real-time decision system for an autonomous wheeled robot with the holonomic-drive system. The robot uses decisions to avoid collisions with obstacles. The decision rules based on grid-edge-depth map. The grid-edge-depth map represents the obstacle's position and distance in the environment. The generation process of the grid-edge-depth map presented in previous research. The decisions of the first scenario with no destination point are forward, stop, 90o right turn, and 90o left turn. The decisions of the second and third scenarios with a destination point are forward, stop, 90o right turn, 90o left turn, 45o forward to the right, 45o forward to the left, slide to the right, and slide to the left. The proposal tested in a 5x3 meter living environment. Finally, the experiment resulted in 93.3% of navigation's success for all the scenarios.

Copyright © 2020 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Budi Rahmani,

Department of Informatics,

STMIK Banjarbaru, Kalimantan Selatan, Indonesia.

Email: budirahmani@gmail.com

1. INTRODUCTION

An autonomous wheeled robot needs a real-time decision system to navigate and avoid an obstacle, whether indoors or outdoors. Obstacle avoidance is the ability of a robot to detect the presence of objects that obstruct its movement toward the destination. This ability could be based on image processing using a stereo camera. When using a stereo camera, the system will obtain the disparity map based on a pair of rectified images. After that, the system will calculate the approximate distance between objects detected by a calibrated camera [1, 2]. A stereo camera-based navigation system implemented in [3, 4]. The method improved with the detection process and navigation path recognition. When there is an obstacle, the system decides to skip the path. CNN method used to detect the path and the presence of obstacles by [5]. This method provided the training with some stereo images of obstacles. Based on the training, there was a knowledge model that used to detect and distinguish between obstacle paths [6]. In practice, the obstacle faced by robots was not only static but also sometimes had moving objects.

The next model developed using the Sum of Absolute Difference. This method tried to improve the accuracy of the distance measurement process from the stereo camera. In this model, the robot instructed to move back if it finds the obstacle in front of it at less than 50 cm ahead, then direct it to the right or the left by 90o [7]. The other method, called the block matching algorithm, was used to detect the presence of obstacles in front of the camera, the method based on the disparity map. If the image pixel is darker than the others, it means there is more distance to the object from the camera or robot. The reaction of the robot was to move in reverse or turn 90o to the left or the right if the distance was less than 50 cm [8].

Improvement of the obstacle detection model using object imagery training shown in [9], the method is called a hybrid topology map building. In this study, robots trained with a series of image objects stored in a database. The image captured by the camera and matched with the database. The decision was

taken to distinguish the objects as an obstacle or a destination point. This purpose required enough memory capacity to store a series of images in the database [9]. Another method of recognizing the obstacles was to use fuzzy logic with a server to process the image [10]. It has been possible to improve the accuracy of the process of recognizing the obstacle or the destination [11]. The method required a vast hardware resource. The image processing process is given to the server and not processed onboard the robot. Therefore, the fuzzy logic algorithm implemented in [12] to detect the obstacles. In this study, the method tried to separate between the obstacles and the background. The robot will restrict the obstacles at least 90 cm apart.

Another researcher tried to develop a block-based motion estimation model to detect the presence of moving objects using a single camera. The objects are still relatively far from the robot, but the weakness in this model is that there is often a fault detection when there is more than one object that moves and is caught by the camera [13]. Subsequent research with potential tire methods improved deficiencies in [13]. This potential tire method is capable of detecting moving obstacles, and can then predict the direction of movement. The method mimics how humans avoid obstacles when they find them while walking, and in this case, humans will try to move from the obstacles to safer places or positions. This method developed from the artificial potential field navigation method first introduced in [14], which initially used as a method to avoid obstacles and path planning on the robot manipulator. The main idea is to create a virtual, potential force field over an object that is 'viewed' by the camera, from which it will return a virtually determined plane of a safe, passable object in the robot navigation [14]. In this method, at each step or iteration, the process of detecting the possibility of a collision was performed. If there is a possibility of a collision, then the tire potential is described on a grid map. A potential tire is a value of confidence by the robot, which the robot must maneuver around the moving obstacles. The amount of this virtual certainty will tend to increase in the direction of movement of the obstacle, from that value being in the predicted position of the obstacle per 2m step before and after the collision, and calculated by (1) [15]:

$$vc_{(i,j)} = k \frac{C_{max}}{2m} \text{ with } k = 1, 2, \dots, 2m \quad (1)$$

The next method developed to detect the presence of obstacles was Q-Learning [16]. In this method, the robot is trained with a series of environmental images and also many obstacles. The method creates a knowledge model that allows the robot to recognize the arena of an environment where an obstacle does not exist. Also, with this method, the existence of a moving obstacle can be detected, so the next path to be passed can be determined based on the direction of movement of the obstacle.

Furthermore, the method of navigation, especially for the process of avoiding the obstacle, is Simultaneous Localization and Mapping (SLAM) [17, 18]. This method has always been developed from the outset [4]. The research developed the SLAM method using stereo cameras - the SLAM method implemented in robots using ultrasonic sensors. The method can map itself to the environment and detect the obstacles. Another SLAM method combined with the Artificial Potential Field (APF) method, the following research uses input from the Laser Range Finder (LRF) sensor to determine its position in the environment and then combine this with the potential impact of an obstacle [14-16]. The method in this study simulated to be able to assist the robot in avoiding obstacles that move [19, 20]. An error found at the time of the simulation was a read error by the LRF sensor that disturbs the process of localization of robots in the environment. Parallel tracking and mapping methods that are part of SLAM also used in research [21]. This method can estimate the position of robots in the environment, as is done by GPS sensors.

In [22], the process of separating obstacles and background images of the outdoor environment described. The method was to build a disparity-map followed by a new method called slope analysis (V-Intercept). V-Intercept, if depicted in XY coordinates, is a hollow-shaped slope resembling the letter V, indicating that in the horizontal position (in an image), there is an obstacle. Based on that information, the direction of the vehicle movement path can then be planned [22].

This study is an extension of [13, 24]. In [13], we have proposed the GED-map building that represents the obstacle's position and distance in the environment. We continue to build rules based-on the GED-map information in real-time. The rules are the exploration result of the best distance and position of the robot to the obstacle. The rules are the new contributions of the present work since we use no other sensor/s except an assembled stereo camera.

2. RESEARCH METHOD

Figure 1 shows our experimental robot in this study. The robot consists of four Omni wheels and four DC motors. They are controlled by an Arduino mega based microcontroller that receives commands from the computer via a USB port. The robot uses a holonomic-drive system that allows the robot to move in

all directions according to a command decision from the system. All the commands coded into an upscale alphabetical code, that is A, B, C, D, E and F, in this case, the robot programmed to go forward (A), backward (B), move to the right (C), move to the left (D), move 45 degrees forward to the right (E), and move 45 degrees forward to the left (F). This study used an assembled stereo camera. The calibration process used a toolbox by Jean-Yves Bouguet [25-27]. Both cameras connected to the computer via a USB port and shown in Figure 1 [3].



Figure 1. A complete assembled robot with a stereo camera [24]

2.1. Grid-edge-depth map-based obstacle avoidance model

The yellow boxes in Figure 2 elucidate all of our proposed model in building a GED-map based obstacle avoidance system. In this study, we added some modules or processes to guide the robot headed for its destination utilizing the GED-map. The modules are yellow lamp detection, rules-based decision, and an Arduino-based motor controller. The algorithm of yellow lamp detection and the rules of decision of the direction will describe in the next subchapter.

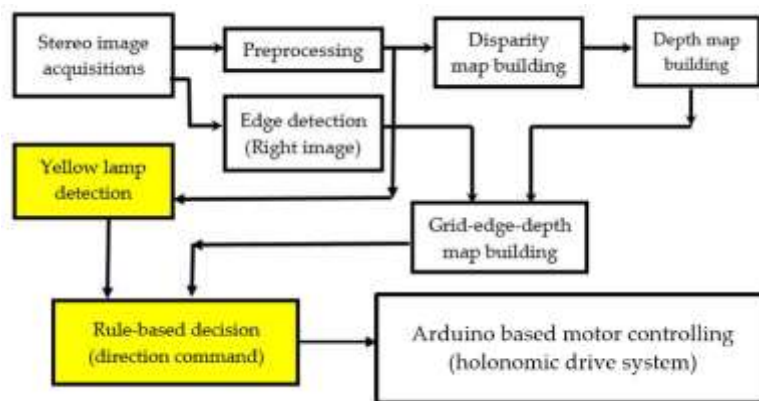


Figure 2. The GED-based obstacle avoidance system

2.2. Yellow lamp detection

In order to fulfill the requirement of the navigation process, so the robot needs to specify its destination. In this case, the destination of the navigation process is the yellow lamp. The first time before the robot moves to the destination, the system will read the actual x and y position of the detected yellow

lamp from the right image. The entire process described in Algorithm 1. In the next rule, the robot will keep its vision to the destination (yellow lamp) while it is reaching the yellow lamp as a final destination.

Algorithm 1 Yellow lamp detection

```

Input: An Equalized right image
Cropping the image with the ROI Position = [1 1 320 120]
Convert to CMYK
Subtract the image to get the yellow channel
Convert to grayscale
Do median filter two by two matrices
Convert to black and white image with threshold 0.3
Remove pixels from BW matrix that less than 10 pixels
Create the label matrix using 8-connected objects
Get the centroid (x and y) of detected yellow lamp

```

2.3. Rule-based decision of direction

GED-map gives a content of distance and relative position information of the obstacle from the robot. In this case, we used only a few grids of GED-map. The grids are *grid_3_1*, *grid_3_2*, *grid_3_3*, *grid_3_4*, and *grid_3_5*. The whole five grids are in the middle position of GED-map (horizontally and vertically). When the obstacle with height is 30-40 cm and 1 m against the robot, it will appear in the middle of a field of view of the robot (GED-map). The height of the obstacle is the reason for choosing the five grids as a base of decision-making. The next research could use the other grids as decision-making for the input.

Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, and Figure 8 respectively show the decision of the robot to detect the presence of the destination point and the process of avoiding the obstacles along the way to the destination. The decision-maker, in this case, is a programmed laptop as a master controller (MC). The MC will send the command to the slave controller (SC); in this case, the SC is a microcontroller (ATmega2560 based).

First of all, Figure 3 shows the flowchart in detecting the destination object (yellow lamp). Once the system started the robot, the MC will read the presence of the yellow lamp. The system reads the right image. In detecting the edge of the object, the system used the right image. After this, MC reads the depth map and GED-map data. The MC will ensure the yellow lamp exists. When MC finds no yellow lamp, the MC commands the SC to stop the robot. When the yellow lamp exists, the MC reads the x-y position of the yellow lamp and understands the depth map value at the same x-y position. The system will get the distance of the yellow lamp from the robot.

Still in Figure 3, when the position of the yellow lamp is correct, and the distance is ≤ 65 cm and the value of *grid_3_3* is ≤ 65 cm then the MC commands the SC to stop or break the robot. When the distance of yellow lamp is > 65 cm and the value of *grid_3_2* is > 65 cm, *grid_3_3* is > 65 cm, and *grid_3_4* > 65 cm, the decision of system is shown in Figure 4.

When the system finds no obstacle in front of it, the decision or command from the MC to the SC is forward to the yellow lamp. However, if the system finds a very close to the obstacle or its distance is < 65 cm, then the decision will be related to Figure 6. The very close obstacle is when the length from the robot is < 65 cm, and if the distance is > 65 cm, the obstacle is not too close to the robot yet.

The system also preserves the x (horizontal) position of yellow lamp greater or equal than 135 pixels, smaller or equal than 185 pixels, and the position of y position smaller than 130 pixels ($135 > x \leq 185$). Based on the exploration, these x-y positions enable the system to adjust the robot position to the yellow lamp not too frequently. At first the value of x was $150 \geq x \leq 170$ pixel. Then the x value was enlarged to 50 pixels ($135 > x \leq 185$) as mentioned before. The y value was set to < 130 pixels so as not to detect the shadow of the yellow lamp on the floor.

Figure 5 shows the decision of the system to preserve the position of the robot. The system will decide to turn the robot to the right, or the left one degree when the x position of the yellow lamp is ≥ 135 and x position of the yellow lamp is ≤ 185 horizontally. Turn the robot one degree to the left or the right moves the robot more smoothly.

Figure 6 shows the first condition when the robot is too close to the obstacle. The state is the continuation of Figure 4. If the MC found an obstacle and the distance to the robot is less than 65 cm at the left front of it (*grid_3_2* < 65 cm and *grid_3_3* < 65 cm) while the distance to the yellow lamp is more than 65 cm, then MC will command the SC to move the robot to the right about 12 cm (in 2.5 seconds). If the distance of the yellow lamp is less than 65 cm, it means the robot reached the goal, and the MC will command to the SC to stop the robot.

Figure 7 shows the condition when the robot is not too close to the obstacle (the distance is more than 65 cm). The state is the continuation of Figure 6. When the MC of the robot detects the obstacle to the left front of it (*grid_3_2* < 65 cm, *grid_3_3* > 65 cm, *grid_3_4* > 65 cm and *grid_3_5* > 65 cm), then the

system checks the distance to the yellow lamp. If the distance from the yellow lamp to the robot is < 65 cm, then the MC commands the SC to stop or brake the robot; but if the distance from the yellow lamp to the robot is > 65 cm, then the MC commands the SC to 45 degrees right forward about 5 cm (in one second). If the system finds another condition of the obstacle's distance and relative position, the decision will show in Figure 8.

Figure 8 shows the second condition when the robot is too close to the obstacle. When the obstacle found in the left front of the robot ($grid_3_1 < 65$ cm, $grid_3_2 < 65$ cm, $grid_3_3 < 65$ cm, and $grid_3_4 > 65$ cm) and the distance of the yellow lamp is more than 65 cm, the MC commands the SC to shift to the right 5 cm (in 1 second); but if the distance of the yellow lamp is less than 65 cm, the MC commands the SC to stop or brake the robot. When the obstacle is found to the right front of the robot while the distance of the yellow lamp is more than 65 cm, the MC commands the SC to shift to the left 5 cm (in 1 second), or the MC commands the SC to stop the robot if the distance of the yellow lamp is less than 65 cm. The end of every decision will be back to terminal B to loop the process of detecting the goal of navigation (the yellow lamp).

As mentioned before, GED-map informs about the distance and the position of the obstacles. In this case, we utilize some of the grid information to decide the direction of the robot when moving in the environment. The evaluation of decision rules in some scenarios based on the experiment described in the next chapter.

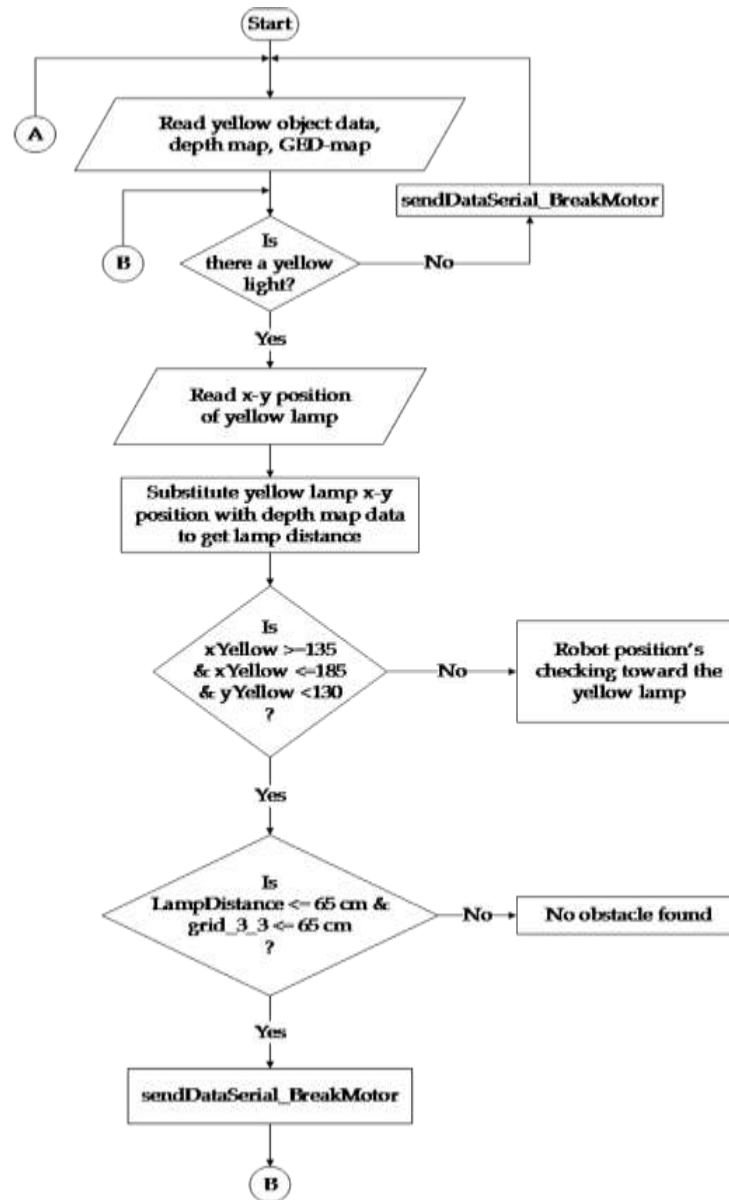


Figure 3. Flowchart of destination detection



Figure 4. Flowchart of the decision when finding no obstacle

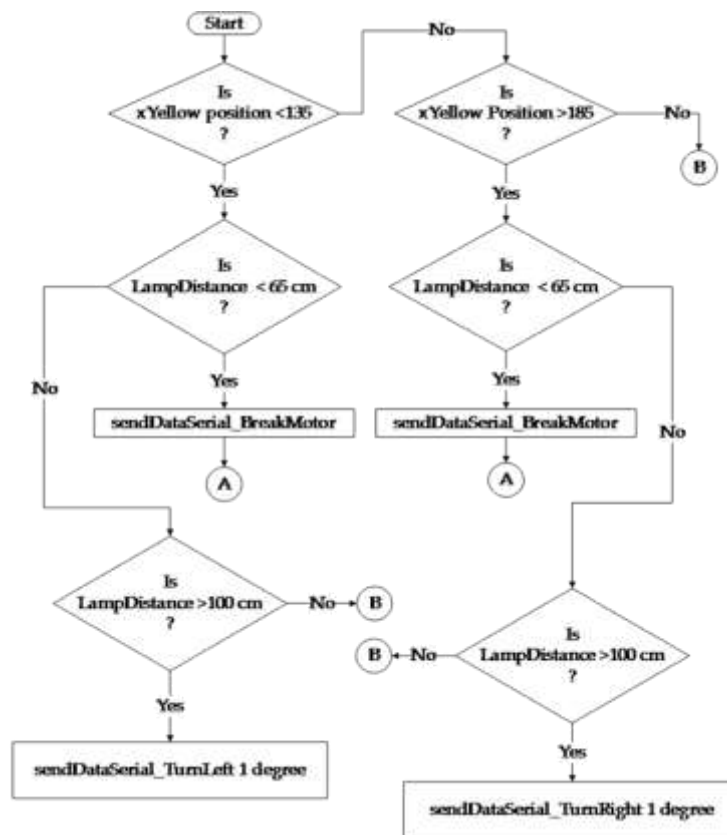


Figure 5. Checking and preservation position of the robot toward the yellow lamp

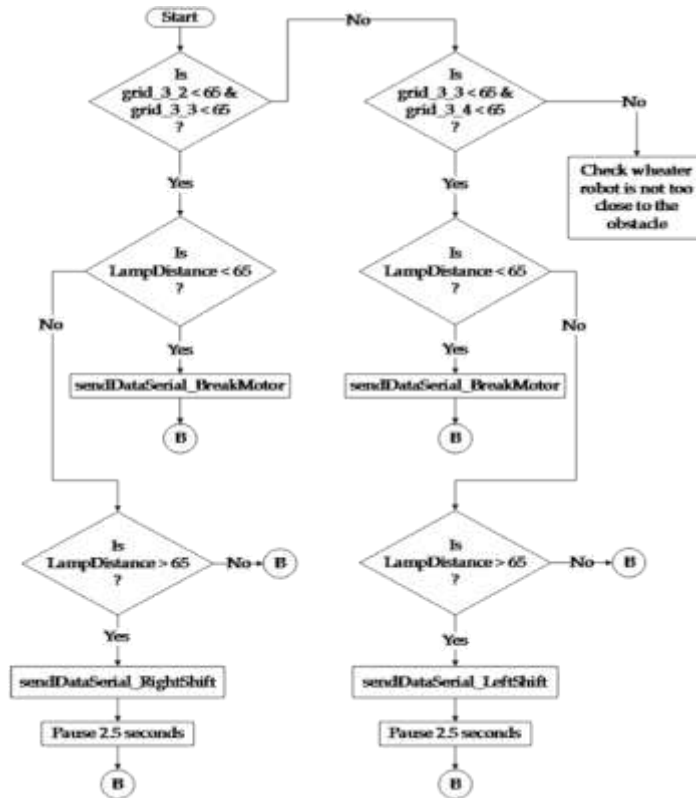


Figure 6. The decision when the robot is too close to the obstacle (1)

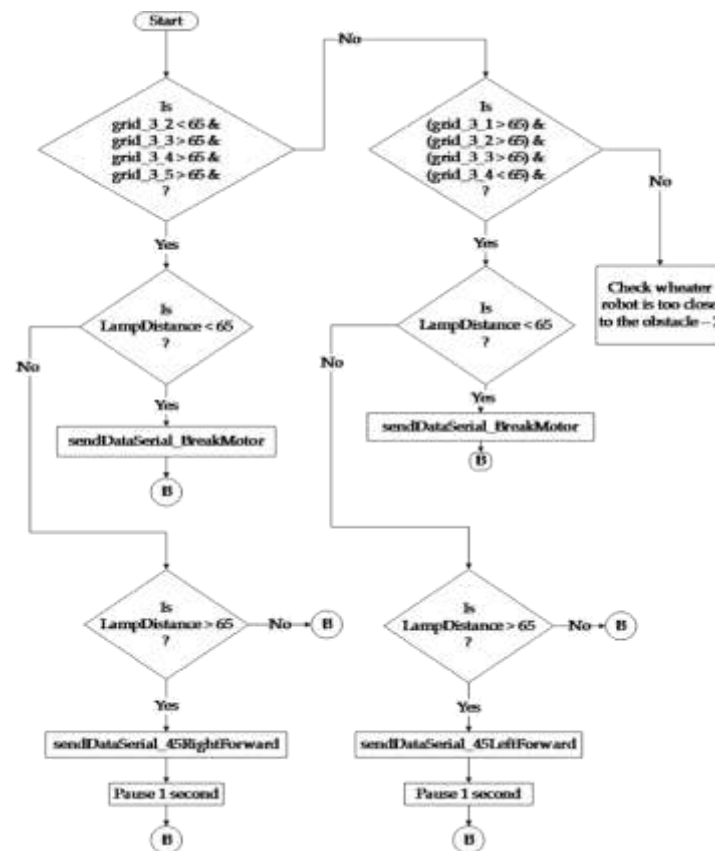


Figure 7. The decision when the robot is not too close to the obstacle and its illustration

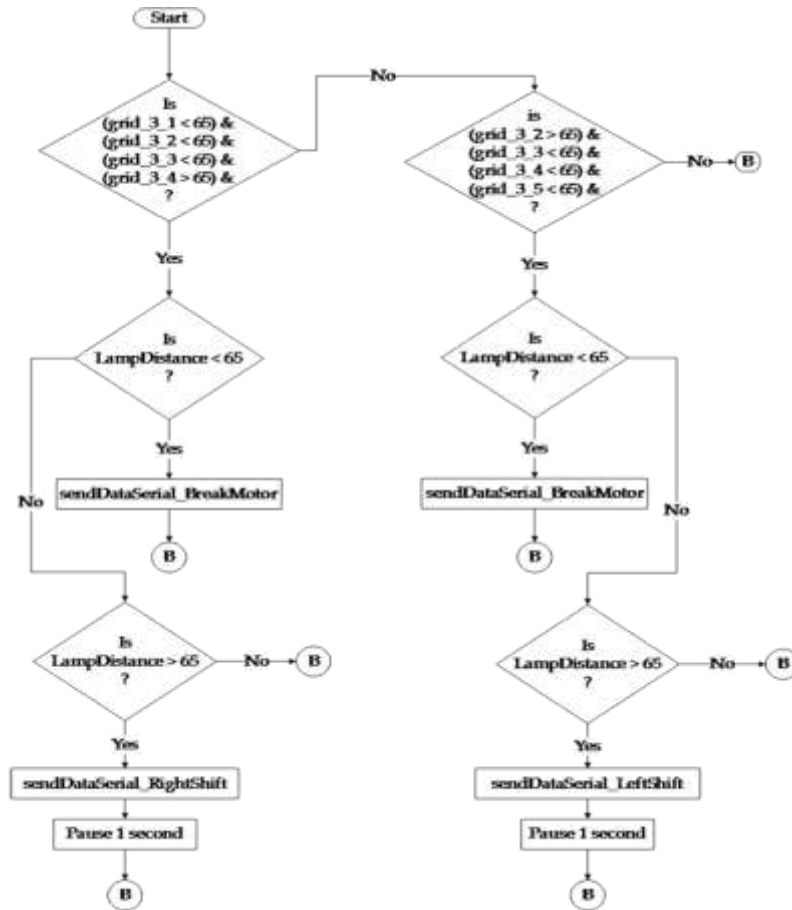


Figure 8. The decision when the robot is too close to the obstacle (2)

2.4. The evaluation scenarios

In order of evaluation of the proposed method, the experiment conducted in 3 scenarios. The first, the second, and the third scenarios, will examine the consistency of rule-based decision of directions. Figure 9 shows the first scenario in avoiding an obstacle at 210 cm distance. The obstacle is a box 37 x 12 x 33 cm in size. The distance of the robot to the goal is about 390 cm. The goal is a yellow lamp on the 30 cm height of the table. Figure 10 shows the second scenario in avoiding a pair of obstacles at 130 cm distance. The obstacles are a pair of 25 x 25 cm chairs. The gap between the chairs is 60 cm. Figure 11 shows the third scenario. The robot must go to the destination while avoiding three obstacles. The obstacles in the first and second scenarios will combine in the third scenario. For all the scenarios, the robot should achieve the goal while the distance toward the goal is ≤ 50 cm. After that, the traveling time of the robot will be equal since its velocity is 5 cm/second.

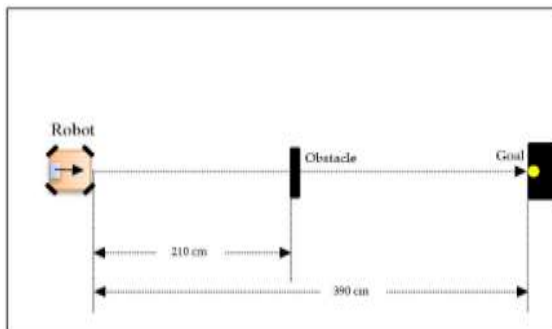


Figure 9. Scenario 1

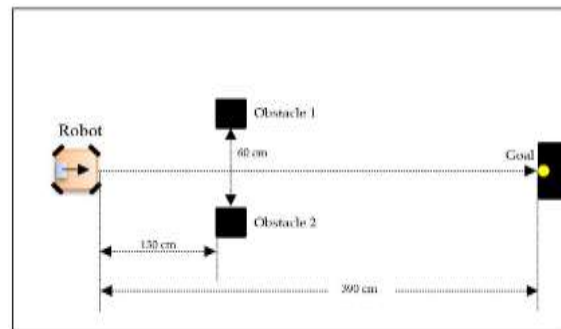


Figure 10. Scenario 2

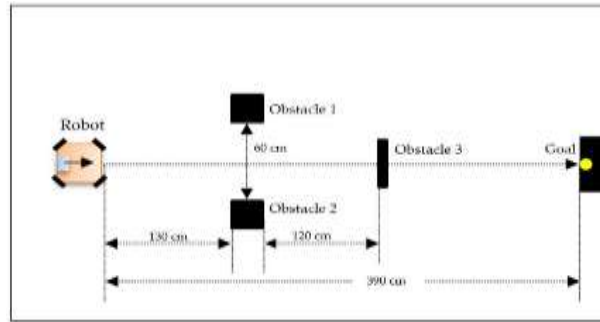


Figure 11. Scenario 3

3. RESULTS AND ANALYSIS

The proposed method is applied to build a collision-avoidance system utilizing the GED-map and stereo vision system. The following paragraphs describe the results of experiments of this study. The description of GED-map’s generation process could found in [28, 29].

3.1. Real-time in-building collision avoidance evaluation

Table 1, Table 2, and Table 3, respectively, show the experimental result of each scenario. Every scenario had fifteen-time trials. The distance of the robot’s start position and the goal (the yellow lamp) is about 390 cm. The distance when the robot reaches the target is ≤ 55 cm as the previous research has done. Afterwards, if the minimum distance to the goal achieved and there is no collision with the obstacles, then the status of the trial is ‘OK’. When it found any collision and or the distance is more than 55 cm, the trial status will be ‘FAILED’. The documentation when the robot was avoiding an obstacle depicted in Figure 12 as scenario 1. Figure 13 showed the condition when the robot was avoiding two obstacles (scenario 2). Figure 14 showed the documentation when the robot was avoiding three obstacles (scenario 3).

Table 1. The result of scenario 1

Trial number	Ground Truth (cm)	Traveled distance (cm)	Robot’s distance to the goal (cm)	Time traveled (second)	Trial status
1	390	350	40	48	OK
2	390	355	35	49	OK
3	390	355	35	56	OK
4	390	354	36	54	OK
5	390	354	36	46	OK
6	390	345	45	58	OK
7	390	344	46	57	OK
8	390	337	53	49	OK
9	390	354	36	53	OK
10	390	351	39	50	OK
11	390	350	40	49	OK
12	390	350	40	43	OK
13	390	331	59	50	FAILED
14	390	335	55	50	OK
15	390	340	50	47	OK

Table 2. The result of scenario 2

Trial number	Ground Truth (cm)	Traveled distance (cm)	Robot’s distance to the goal (cm)	Time traveled (second)	Trial status
1	390	351	39	43	OK
2	390	351	39	49	OK
3	390	349	41	45	OK
4	390	353	37	43	OK
5	390	350	40	58	OK
6	390	331	59	40	FAILED
7	390	352	38	43	OK
8	390	340	50	47	OK
9	390	340	50	46	OK
10	390	349	41	47	OK
11	390	347	43	46	OK

Trial number	Ground Truth (cm)	Traveled distance (cm)	Robot's distance to the goal (cm)	Time traveled (second)	Trial status
12	390	348	42	43	OK
13	390	335	55	40	OK
14	390	352	38	44	OK
15	390	348	42	43	OK

Table 3. The result of scenario 3

Trial number	Ground Truth (cm)	Traveled distance (cm)	Robot's distance to the goal (cm)	Time traveled (second)	Trial status
1	390	340	50	43	OK
2	390	340	50	49	OK
3	390	350	40	45	OK
4	390	348	42	43	OK
5	390	345	45	48	OK
6	390	331	59	40	FAILED
7	390	350	40	43	OK
8	390	350	40	47	OK
9	390	348	42	46	OK
10	390	356	34	47	OK
11	390	346	44	46	OK
12	390	352	38	43	OK
13	390	345	45	45	OK
14	390	340	50	48	OK
15	390	345	45	48	OK



Figure 12. Collision avoidance in scenario 1 (one obstacle)



Figure 13. Collision avoidance in scenario 2 (two obstacles)



Figure 14. Collision avoidance in scenario 3 (three obstacles)

3.2. Discussion

Grid-Edge-Depth map built-in 0.09-0.14 second and enabled the real-time decision. The system decides while the robot is moving toward its goal position. As inputs of decision, the system uses the information of object's distance from grid_3_1, grid_3_2, grid_3_3, grid_3_4 and grid_3_5. In this study, we found the ideal distance to detect the presence of object or obstacle and field of view positioning of the robot's vision to the yellow lamp as a goal. The ideal distance to detect the presence of the obstacle is

65-100 cm. When the distance of obstacle is between 80-100 cm, the decision of direction to avoid the collision are 45 degrees to the left front and or 45 degrees to the right front. When the distance of the obstacle is less than 65 cm, the decision of direction to avoid the collision is a shift to the right or shift to the left.

In order to keep the position of the robot's field of view to the yellow lamp, we found the best positioning system of it. It is $135 \geq x$ position of the yellow lamp ≤ 185 , and y position of the yellow lamp is < 130 . The size of the acquired image is 320x240 pixels. By this positioning, we found the robot was not moving too much to keep its field of view toward the yellow lamp. Related to the obstacles, we try out some obstacles with a different color in the experiment. The obstacle is little kids' black chairs, 30 cm height tables, and a 40x10x40 cm white box.

The average travel time of scenario 1 (all trials) is 50.6 seconds, while it is 50.8 seconds for 'OK' only status. The average distance traveled by the robot is 347 cm for all trials and 350 cm for 'OK' only status in scenario 1. The success rate of scenario 1 in reaching the goal without colliding or nudging the obstacle is 93.3%. Hereafter, the average travel time of scenario 2 (all trials) is 45.1 seconds, and 45.9 seconds for 'OK' only status. The average distance traveled by the robot is 346 cm for all trials and 348 cm for 'OK' only status in scenario 2. The success rate of scenario 2 in reaching the goal without nudging the obstacles is 93.3%. The last scenario (scenario 3) has an average travel time of 45.4 seconds for all trials, and 45.8 seconds for 'OK' only status. The average distance traveled by the robot is 346.3 cm for all trials and 346.9 cm for 'OK' only status. The success rate of scenario 3 in reaching the goal without nudging any of the obstacles is 93.3%. This achievement because of the GED-map based rule decision equipped with position estimation, while [30-32] were not. So, it has been more accurate than the previous method in deciding the robot's movement. show the documentation of experiments using three scenarios in this study. The 'FAILED' status found when the distance of the robot reach to the goal 59 cm in the thirteenth trial, the sixth trial, and the sixth trial of the first, second, and the third scenarios respectively - this 9 cm based on the maximum difference between real distance measurement and system measurement of the GED-map.

4. CONCLUSION

We have discussed the evaluation of this study. The rule-based decisions of the robot utilizing the GED-map have succeeded in anticipating the collision with the obstacle in real-time. The processing time since the image acquisition process to the decision process is 0.09 – 0.14 seconds. The success rates of the first, second, and third scenarios in reaching the goal without colliding or nudging the obstacle are 93.3%, 93.3%, and 93.3% respectively. When the robot found the obstacles and the position of the obstacle was in front of the robot, we found a consistency of decision-making by the system in avoiding it.

ACKNOWLEDGEMENTS

This work supported by a research grant from the Ministry of Research, Technology, and Higher Education of the Republic of Indonesia (2018, 2019, and 2020).

REFERENCES

- [1] S. Tijmons, G. C. H. E. de Croon, B. D. W. Remes, C. De Wagter, and M. Mulder, "Obstacle Avoidance Strategy using Onboard Stereo Vision on a Flapping Wing MAV," in *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 858-874, August 2017.
- [2] B. Rahmani, A. Harjoko, T. K. K. Priyambodo, and H. Aprilianto, "Research of Smart Real-time Robot Navigation System," in *AIP Conference Proceedings*, vol. 1707, no. 1, pp. 1-8, February 2016..
- [3] M. Papoutsidakis, "Intelligent Design and Algorithms to Control a Stereoscopic Camera on a Robotic Workspace Z-error Z-Error," *International Journal of Computer Applications*, vol. 167, no. 12, pp. 32-35, June 2017.
- [4] K. N. Al-Mutib, E. A. Mattar, M. M. Alsulaiman, and H. Ramdane, "Stereo Vision SLAM Based Indoor Autonomous Mobile Robot Navigation," *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, Bali, pp. 1584-1589, 2014.
- [5] L. Ran, Y. Zhang, Q. Zhang, and T. Yang, "Convolutional Neural Network-Based Robot Navigation Using Uncalibrated Spherical Images," *Sensors*, vol. 17, no. 6, pp. 1341, June 2017.
- [6] Y. Zhao, W. Cheng, L. Jia, and S. Ma, "The Obstacle Avoidance and Navigation Based on Stereo Vision for Mobile Robot," *2010 International Conference on Optoelectronics and Image Processing*, Haikou, pp. 565-568, 2010.
- [7] S. Solak and E. D. Bolat, "Distance Estimation using Stereo Vision for Indoor Mobile Robot Applications," *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)*, Bursa, pp. 685-688, 2015.
- [8] R. A. Hamzah, H. N. Rosly, and S. Hamid, "An Obstacle Detection and Avoidance of A Mobile Robot with Stereo Vision Camera," *2011 International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, Kuala Lumpur, pp. 104-108, 2011.

- [9] J. Kim, C. Park, and I. S. Kweon, "Vision-based Navigation With Efficient Scene Recognition," *Intelligent Service Robotics*, vol. 4, no. 3, pp. 191–02, 2011.
- [10] P. Benavidez, "Mobile Robot Navigation and Target Tracking System," *2011 6th International Conference on System of Systems Engineering*, Albuquerque, NM, pp. 299-304, 2011.
- [11] T. H. Nam, J. H. Shim, Y. I. Cho, "A 2.5D map-based mobile robot localization via cooperation of aerial and ground robots," *Sensors*, vol. 17, no. 12, pp. 2730, December 2017.
- [12] M. Chen, Z. Cai, Y. Wang, "A method for mobile robot obstacle avoidance based on stereo vision," *IEEE 10th International Conference on Industrial Informatics*, Beijing, pp. 94-98, 2012.
- [13] J. Kim and Y. Do, "Moving Obstacle Avoidance of a Mobile Robot Using a Single Camera," *Procedia Engineering*, vol. 41, pp. 911-916, 2012.
- [14] Q. Zhang, D. Chen, and T. Chen, "An Obstacle Avoidance Method of Soccer Robot Based on Evolutionary Artificial Potential Field," *Energy Procedia*, vol. 16, pp. 1792-1798, 2012.
- [15] F. A. Yaghmaie, A. Mobarhani, H. D. Taghirad, "Study of Potential Ban Method for Mobile Robot Navigation in Dynamic Environment," *4th Annual International Power Electronics, Drive Systems and Technologies Conference*, Tehran, pp. 535-540, 2013.
- [16] Y. Zhang, C. W. de Silva, D. Su, and Y. Xue, "Autonomous Robot Navigation with Self-learning for Collision Avoidance with Randomly Moving Obstacles," *2014 9th International Conference on Computer Science & Education*, Vancouver, BC, pp. 117-122, 2014.
- [17] K. Sharma, "Improved visual SLAM: a novel approach to mapping and localization using visual landmarks in consecutive frames," *Multimedia Tools and Applications*, vol. 77, pp. 7955-7976, 2018.
- [18] C. Brand, M. J. Schuster, H. Hirschm, and M. Suppa, "Stereo-Vision Based Obstacle Mapping for Indoor / Outdoor SLAM," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, pp. 1846-1853, 2014.
- [19] W. G. Aguilar, V. P. Casaliglla, and J. L. Pólit, "Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles," *Electronics*, vol. 6, no. 10, pp. 1-23, January 2017.
- [20] M. A. Martínez, J. L. Martínez, and J. Morales, "Motion Detection from Mobile Robots with Fuzzy Threshold Selection in Consecutive 2D Laser Scans," *Electronics*, vol. 4, pp. 82-93, January 2015.
- [21] S. Wangsiripitak, "In-building navigation system using a camera," *2014 6th International Conference on Knowledge and Smart Technology (KST)*, Chonburi, pp. 35-40, 2014.
- [22] T. Cao, Z. Y. Xiang, and J. L. Liu, "Perception in Disparity: An Efficient Navigation Framework for Autonomous Vehicles With Stereo Cameras," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2935-2948, Oct. 2015.
- [23] S. Iizuka, T. Nakamura, and S. Suzuki, "Robot Navigation in Dynamic Environment Using Navigation Function APF with SLAM," *2014 10th France-Japan/ 8th Europe-Asia Congress on Mechatronics (MECATRONICS2014-Tokyo)*, Tokyo, pp. 89-92, 2014.
- [24] B. Rahmani, A. Harjoko, and T. K. Priyambodo, "The Accuracy Improvement of Object's Distance Measurement Based on Grid-Edge-Depth Map in the Determination of Wheeled Robot's Decision of Direction," Universitas Gadjah Mada, 2019.
- [25] J.-Y. Bouquet, "Camera Calibration Toolbox for Matlab," 2015. [Online]. Available: http://www.vision.caltech.edu/bouquetj/calib_doc/.
- [26] Y. Zhao and J. Lei, "A Camera Self-Calibration Method Based on Plane Lattice and Orthogonality," *TELKOMNIKA*, vol. 11, no. 4, pp. 767–774, December 2013.
- [27] B. O. Kennedy, "Stereo Camera Calibration," University of Stellenbosch, 2002.
- [28] B. Rahmani, A. Harjoko, and T. K. Priyambodo, "Grid-edge-depth map building employing sad with sobel edge detector," *International Journal on Smart Sensing and Intelligent Systems*, vol. 10, no. 3, pp. 551–566, September 2017.
- [29] B. Rahmani, A. Harjoko, T. K. Priyambodo, and H. Apriyanto, "Early Model of Vision-Based Obstacle Mapping Utilizing Grid-Edge-Depth Map," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 2, pp. 4519–4523, December 2019.
- [30] M. Khan, S. Hassan, S. I. Ahmed, and J. Iqbal, "Stereo-vision-based real-time obstacle detection scheme for Unmanned Ground Vehicle with steering wheel drive mechanism," *2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, Islamabad, pp. 380-385, 2017.
- [31] J. Jin, Y. G. Kim, S. G. Wee, D. H. Lee, and N. Gans, "A Stable Switched-System Approach to Collision-Free Wheeled Mobile Robot Navigation," *Journal of Intelligent and Robotic Systems*, vol. 86, no. 3–4, pp. 599–616, 2017.
- [32] N. F. Mustamin, "Relative distance measurement between moving vehicles for manless driving," *2017 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, pp. 1-4, 2017.

BIOGRAPHIES OF AUTHORS

Budi Rahmani received his bachelor's of Electrical Engineering, Master of Informatics, and Doctor of Computer Science from Yogyakarta State University, Dian Nuswantoro University, and Universitas Gadjah Mada in 2003, 2010 and 2019 respectively. He was interested in Computer Vision, Embedded System, and Robotics. Currently, his research focused is computer vision and control system for the robot. His other research interests include decision support system using an artificial neural network. His contact email: budirahmani@gmail.com <http://budirahmani.wordpress.com>



Agus Harjoko received his bachelor's degree in 1986 Universitas Gadjah Mada, Indonesia; His master degree in 1990 from University of New Brunswick, Canada; and his doctoral degree in 1996 from University of New Brunswick, Canada. Since 1987 he has been a lecturer and a researcher at the Department of Computer Science and Electronics, Universitas Gadjah Mada. His research interests lie in the field of Digital Image Processing, Machine Vision, Sensor Network, Multimedia IR (Image, Audio, Video Processing). His contact email: aharjoko@ugm.ac.id <http://acadstaff.ugm.ac.id/aharjoko>



Tri Kuntoro Priyambodo currently he is an Associate Professor at Department of Computer Science and Electronics Gadjah Mada University. He is a member of IEEE. He also holds a position as a Secretary of Satellite and Aerospace Electronics Research Group, Universitas Gadjah Mada. During 2010-2013 he was INSPIRE National Project Leader. His responsibility was coordinating the development of the First Indonesia Inter-University Satellite (IiNUSAT-1), a project funded by the Directorate General of Higher Education, Ministry of Education and Culture. He actively joins to Small Satellite Standardization group in Kitakyushu Institute of Technology. Also, active in APSCO's Small Student Satellite Project Meeting. Currently, he conducts research in the field of nanosatellite and Autonomous Unmanned Systems. His other research interests include Computer Network Security eGovernment Systems. His contact email: mastri@ugm.ac.id <http://acadstaff.ugm.ac.id/mastri>