



Vision-based vanishing point detection of autonomous navigation of mobile robot for outdoor applications

Leonard Rusli *, Brilly Nurhalim, Rusman Rusyadi

*Mechatronics Department, Swiss German University
The Prominence Tower Alam Sutera, Tangerang, 15143, Indonesia*

Received 22 October 2021; Accepted 8 December 2021; Published online 31 December 2021

Abstract

The vision-based approach to mobile robot navigation is considered superior due to its affordability. This paper aims to design and construct an autonomous mobile robot with a vision-based system for outdoor navigation. This robot receives inputs from camera and ultrasonic sensor. The camera is used to detect vanishing points and obstacles from the road. The vanishing point is used to detect the heading of the road. Lines are extracted from the environment using a canny edge detector and Houghline Transforms from OpenCV to navigate the system. Then, removed lines are processed to locate the vanishing point and the road angle. A low pass filter is then applied to detect a vanishing point better. The robot is tested to run in several outdoor conditions such as asphalt roads and pedestrian roads to follow the detected vanishing point. By implementing a Simple Blob Detector from OpenCV and ultrasonic sensor module, the obstacle's position in front of the robot is detected. The test results show that the robot can avoid obstacles while following the heading of the road in outdoor environments. Vision-based vanishing point detection is successfully applied for outdoor applications of autonomous mobile robot navigation.

©2021 Research Centre for Electrical Power and Mechatronics - Indonesian Institute of Sciences. This is an open access article under the CC BY-NC-SA license (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

Keywords: Houghline transform (HoughlineP); road lines detection; simple blob detector; vanishing point determination.

I. Introduction

The development of robotics has brought many advantages to human life, such as replacing humans to do complex tasks, dealing with dangerous materials, and even exploring places beyond human capability. A mobile robot is one of the most common examples of this usage of robotics. The navigation of a mobile robot is an essential part of an autonomous system. There has been tremendous progress in autonomous navigation for indoor environments in the past decades. Many types of sensors are used in the indoor autonomous navigation system, such as ultrasonic sensors and a laser range finder (LIDAR) such as ones made by Velodyne. The difficulty with Velodyne systems is their high price due to the lack of volume production. A vision-based approach will be a more economical alternative.

One way to find a robot's heading is to use a memorized route that is learned during the teach-

mode phase of robot preparation [1]. This method requires a robust recognition of the vision features used, which becomes a significant challenge in a dynamic environment with moving obstacles such as pedestrian roads. Road signs can be recognized with digital image processing easier than road appearances [2]. Road appearances can be recognized using color histograms and simplifying the road shape like a triangle. However, the triangle assumption is often no longer valid when road shapes are mixed with road signs and pedestrians.

One way to do road recognition is by color-based contrast and segmentation between the road and the boundary landscape. However, vanishing point detection and voting often delay and lag processing time. As long as the processing time is faster than the camera frame rate (30 Hz), real-time processing is achieved. One way to perform real-time processing is by using regional division and angle limitation [3]. Another effort to speed this up using a contourlet texture detector (CTD) has been attempted to speed up pixel detection and given a voting weight for processing efficiency [4].

* Corresponding Author. Tel: +628111041614; Fax: +62-2129779598
E-mail address: leonard.rusli@sgu.ac.id

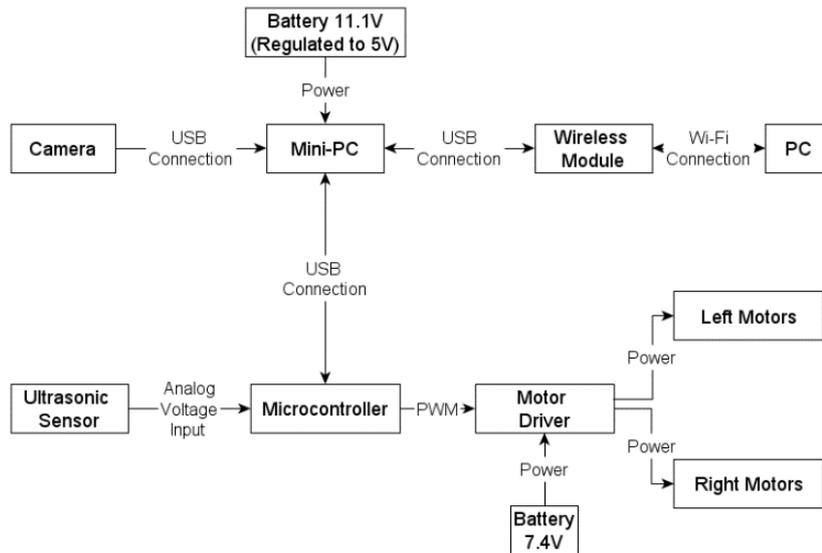


Figure 1. System overview

Vanishing point detection in the unstructured road that lacks road markings is more challenging and requires more complex background suppression and an effective voting strategy [5]. Some previous work tries to circumvent the need for a geometric feature-based detection method by doing motion-based methods of vanishing point detection. Motion is detected by observing optical flows of image points that converge on the focus of expansion [6][7] or alternatively a texture-based method can also be used [8]. This method can be superior for unstructured roads as well. Road segmentation methods using dark channel-based image segmentation also tend to be fast. The accuracy is further improved by combining this with vanishing point voting strategies [9].

A neural network-based vanishing point recognition method has also been done that significantly boosts the point proposals with iterative optimization [10]. Convolutional Neural Network (CNN) classification problem approach to detect vanishing point is attempted using google street-view image dataset as the training set. The results show that superior accuracy is achieved compared to algorithmic vanishing point detector [11].

The detection of the vanishing point in the image as used by several authors [12][13]. These authors used converging edges and textures to decide the most possible vanishing point. Another author attempted to improve the vanishing point detection using an algorithm that uses long and robust contour segments. The road is modeled using a group of lines instead of rigid triangular lines [14][15]. Straight segments in the edge map are recognized using Hough transform available in OpenCV [16]. This results in a more robust algorithm. This approach is the chosen path to develop in this paper.

To navigate the robot in an outdoor environment, a different method is needed because the outdoor environment has far fewer surrounding walls, reducing the effectiveness of the sensors. One way is to manually steer the robot through predefined waypoints, and the robot will follow the predefined

waypoints while avoiding obstacles. The other way is by detecting the road using a vision system. This method does not require manual driving and is not limited to a predefined path. Some methods in detecting the road are called color histogram and vanishing point detection. This paper focuses on vanishing point detection to navigate the robot in outdoor environments. The overall system is described in Figure 1.

The autonomous mobile robot with a vision-based system for navigation in an outdoor environment receives inputs from camera and ultrasonic sensor. The camera is used to detect vanishing points and obstacles from the road. The vanishing point is used to detect the heading of the road. The detected position of the obstacle is used to steer the robot to the vanishing point without collision. The ultrasonic sensor gives information about the actual distance from the robot to the obstacle by emitting an ultrasonic wave and measuring the time elapsed from the time of sending the signal until the time when receiving it. Information of distance from the ultrasonic sensor is sent to the microcontroller. Next, the data is processed in the mini-PC. The output data from the mini-PC which contains the movement information (forward, backward, turn left, or turn right) for the robot is sent to the microcontroller. The output of the microcontroller is pulse-width modulation (PWM) which is used to drive the motor via H-bridge or motor driver. The mini-PC is monitored using a wireless module via a remote desktop connection.

II. Materials and Methods

As seen in Figure 2 the mechanical hardware is designed for navigation in an outdoor environment. The design comes up with 6 off-road wheels which make the robot able to go through any rough terrains and steep inclines. The design also comes up with a twist suspension to keep every wheel in contact with the ground for maximum traction.



Figure 2. Fully constructed autonomous mobile robot for outdoor navigation

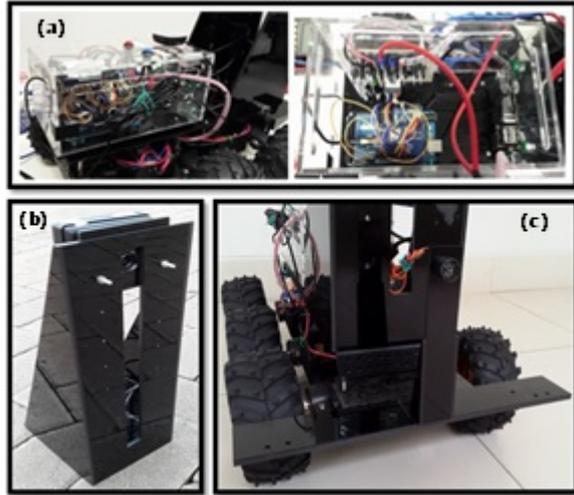


Figure 3. (a) Component box; (b) camera bracket; and (c) ping sensor bracket

To mount all the electrical components on the robot, the robot is also mounted with additional parts such as a camera bracket, components box, and ping sensor bracket which can be seen in Figure 3. All the additional parts are made of acrylic. The reason why acrylic is chosen is that acrylic has less weight compared to steel. The cutting tool used to cut the acrylic is a laser cutting tool which also makes many shapes are easier to cut and also with much lower price.

Figure 4 describes the overview of how the algorithm of the program works. After initialization, first, the road is detected to find the road lines. The detected road lines are used to estimate the vanishing point. The vanishing point is converted to road angle. Next, the obstacle in front of the robot is

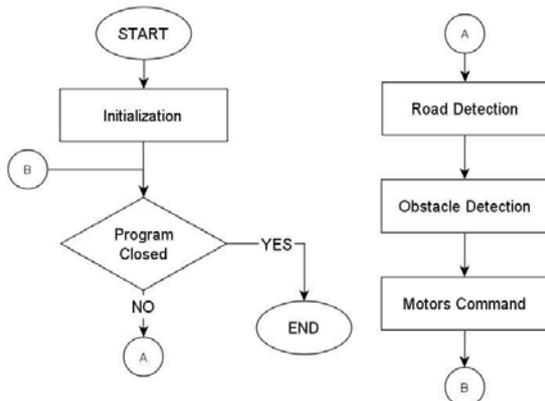


Figure 4. Software design overview

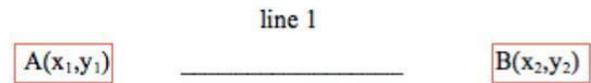


Figure 5. Points of the detected line by using HoughlineP

detected by using a simple blob detector. The distance to the obstacle is detected by using an ultrasonic sensor. The combination of road detection and obstacle detection makes a command to drive the motors. The connection between software at mini-PC to the microcontroller is done through firmata protocol communication.

The vision system first takes the input image from the camera and performs edge detection through the input image. The output of edge detection is a binary image that contains edges. After the edges are detected on the image, the system performs line detection to estimate the possibility of lines created on the image. The line is detected by using Probabilistic Houghline Transform (HoughlineP). The output of HoughlineP points belongs to one line. The illustration can be seen in Figure 5.

Out of the detected lines, lateral lines are filtered out, meaning only lines that are in the longitudinal view will be taken into the next stage of processing. Using the virtual horizon in the image, the lines that intersect with the virtual horizon become the candidate vanishing points. The edges of the road play the most dominant role. Road lanes are not necessary, but they will certainly improve the detection accuracy.

Next, gradient filtering is applied to the detected lines. Only lines that represent the road lines are kept. Other lines such as lines from trees and walls are thrown away. The gradient filtering is illustrated in Figure 6. By calculating the d variable, each of the two lines can be detected whether they are going to intersect each other at some point or not. If d is equal to zero, the lines are parallel and will not intersect each other. On the other hand, if d is not equal to zero, it means that the lines are going to intersect each other at some point. The distance d is calculated from the coordinates x_1, y_1 thru x_4, y_4

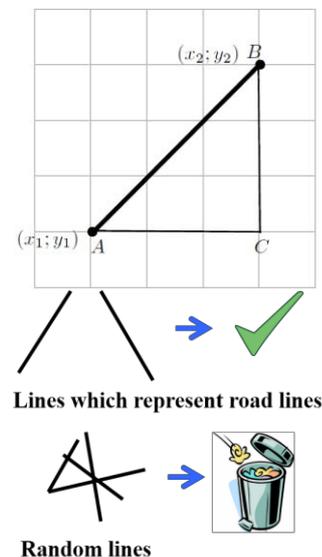


Figure 6. Gradient filtering

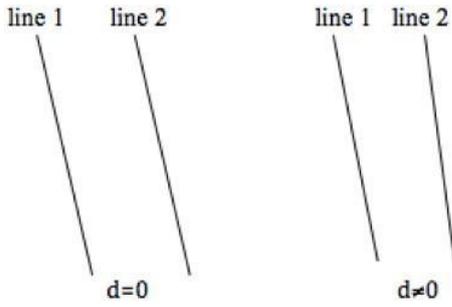


Figure 7. Parallel lines ($d = 0$) and intersecting lines ($d \neq 0$)

which the coordinates of the end points of the lines using equation (1) and illustrated in Figure 7.

$$d = (y_2 - y_1) \cdot (x_4 - x_3) - (y_4 - y_3) \cdot (x_2 - x_1) \quad (1)$$

When d is not equal to zero, the system calculates the intersecting point (x_0, y_0) of two lines by using equation (2). The intersection point represents the vanishing point of the road lines. Knowing where the vanishing point is, the system can estimate the road angle to navigate the robot next. The detected road angle can be seen in Figure 8.

The obstacle in front of the robot is detected by using a simple blob detector and ultrasonic sensor. The input image is converted into grayscale. Next, the input image is thresholded with parameters of `minThreshold` and `maxThreshold` from a simple blob detector. The output of the thresholding is a binary image. The white pixels in the binary image are grouped and called binary blobs. The center of each binary blob is calculated and blobs located closer than `minDistBetweenBlobs` are merged. The center and radius of newly merged blobs are calculated again and returned as key points. Every time there is a blob or obstacle, the simple blob detector calculates the position of the obstacle (x, y) , and combined with an ultrasonic sensor, the distance to the obstacle is known. The algorithm is illustrated in Figure 9.

$$\begin{cases} x_0 = [(x_2 - x_1) \cdot (x_4 - x_3) \cdot (y_3 - y_1) + (y_2 - y_1) \cdot (x_4 - x_3) \cdot x_1 - (y_4 - y_3) \cdot (x_2 - x_1) \cdot x_3] / d \\ y_0 = [(y_2 - y_1) \cdot (y_4 - y_3) \cdot (x_3 - x_1) + (x_2 - x_1) \cdot (y_4 - y_3) \cdot y_1 - (x_4 - x_3) \cdot (y_2 - y_1) \cdot y_3] / (-d) \end{cases} \quad (2)$$

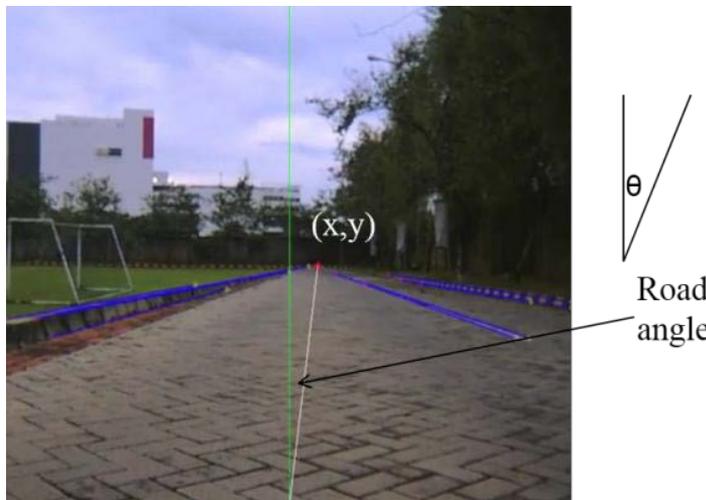


Figure 8. The vanishing point (x, y) is converted to road angle referred to the center of the image

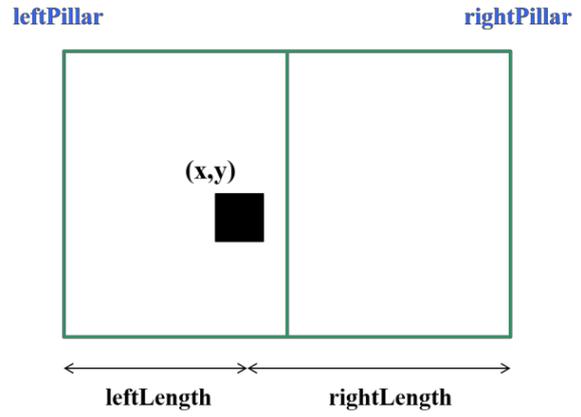


Figure 9. Obstacle detection algorithm

As seen in Figure 9, (x, y) is the position of the obstacle that is detected by the simple blob detector. `leftPillar` and `rightPillar` are the boundaries that indicate the detectable area of the obstacle. Obstacles outside the `leftPillar` and `rightPillar` are not detected as an obstacle. `leftLength` is calculated by $(x - \text{leftPillar})$ and `rightLength` is calculated by $(x - \text{rightPillar})$. Next, the distance to the obstacle is detected by using an ultrasonic sensor. The detectable area from the ultrasonic sensor is from 10 cm to 400 cm. In this paper, the detectable area for the robot to make a turn is limited from 130 cm to 200 cm. If `leftLength` is bigger than `rightLength` and the distance is between 130 cm to 200 cm, it indicates that the obstacle is located on the left area. Therefore, the robot has to go right. Vice versa, if `leftLength` is smaller than `rightLength` and the distance is between 130 cm to 200 cm, it indicates that the obstacle is located in the right area. Therefore, the robot has to go left.

To control the motors, the value of PWM depends on the value of the road angle. The bigger the road angle, the bigger the PWM value. Therefore, the

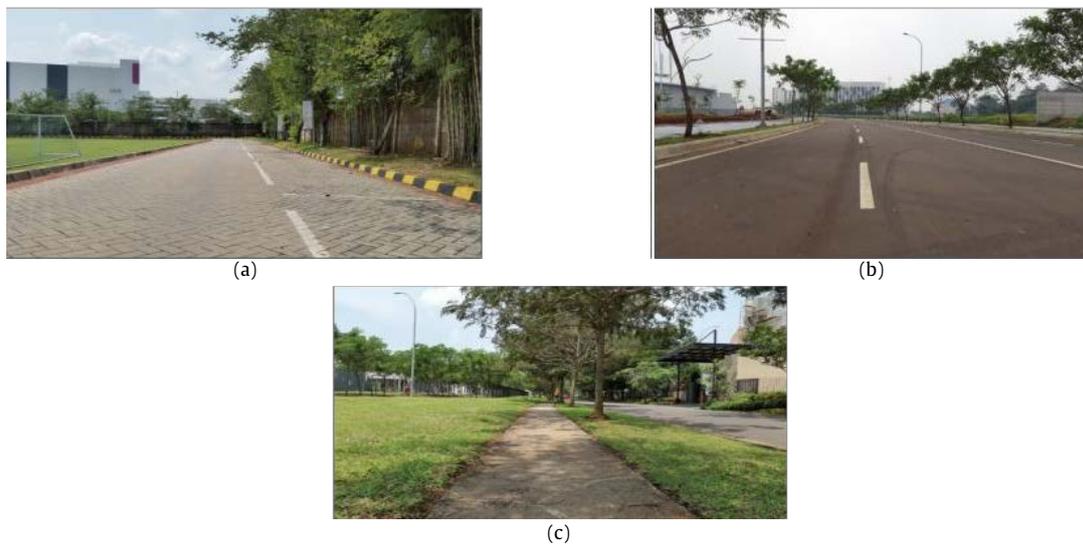


Figure 10. Roads that are chosen to run the robot; (a) the road beside soccer field at SGU Campus; (b) the road in front of ICE; and (c) the pedestrian road in front of Glacido Cluster

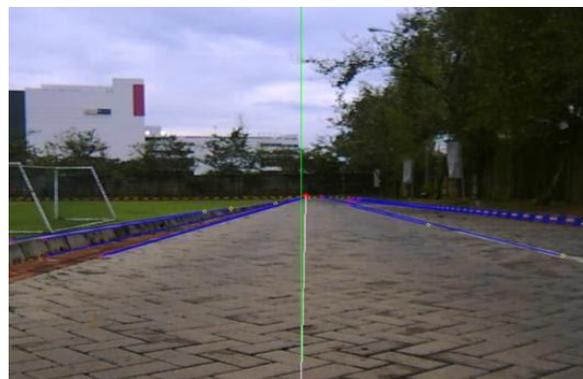


Figure 11. Detected vanishing point on environment 1

robot will turn left or turn right depending on the heading of the road. The sign from the road angle (minus or plus) indicates which direction the robot has to go. If the road angle has minus value (from 0 to -90 degrees), it means the road is now heading to the left and the robot has to go to the left. Vice versa, if the road angle has positive value (from 0 to 90 degrees), it means the road is now heading to the right and the robot has to go to the right. The obstacle detection has the highest priority to command the motors. If there is an obstacle detected in the left region, the robot will turn right at high speed. Vice versa, if there is an obstacle detected in the right region, the robot will turn left at high speed. For high-speed turning (left or right), the PWM value is already set manually. After the obstacle is avoided, the robot will go back following the heading of the road.

III. Results and Discussion

Figure 10 describes the roads that are used to test the robot. The road beside soccer field at SGU Campus has similar characteristics to the road in Frederick D. Fagg park [15]. They have red boundaries on the left and right sides of the road. The difference is the road beside soccer field at SGU Campus has road marking in the middle of the road. These red boundaries and road marking will be detected as road lines by the camera.

The road in front of ICE has similar characteristics to the Hellman Why Road. They are a little bit curved and have boundaries on the left and right sides of the road. On the Hellman Why Road, the boundaries are colored red. On the other hand, the boundaries are formed from the road marking and colored white on the road in front of ICE.

The pedestrian road in front of Glacido Cluster has similar characteristics to the road in front of Leavey Library. They do not have any explicit lines. In the pedestrian road in front of Glacido Cluster, the road lines are formed from different contrast between the color of the grass (green) and the color of the pedestrian (gray). The vision system will try to detect the road lines that are formed from this different contrast.

A. Driving the robot in the selected environments without obstacle

After the selected roads are tested to detect lines. The lines are used to detect the vanishing point. The detected vanishing points are converted to road angle to estimate the heading of the road. When the road angle changes, it means that the heading of the robot has to change also because the robot has to follow the heading of the road.

As seen in Figure 11, the lines detected on this road are formed from red boundaries on the left and the right sides of the road, from road markings located on the right side of the road. The road beside

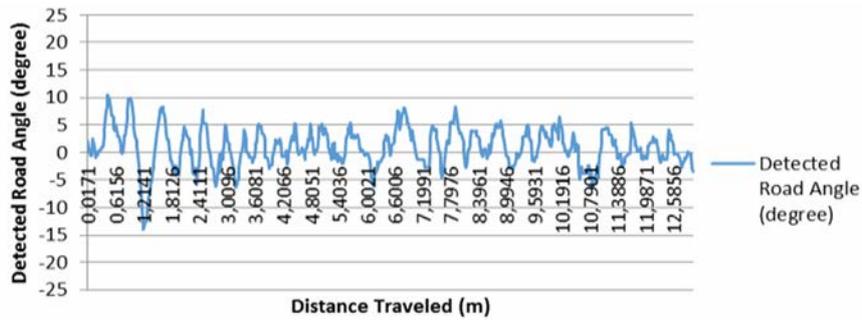


Figure 12. Graph of detected road angles in environment 1, the road beside soccer field at SGU Campus

soccer field at SGU Campus side A is straight, therefore the road angle detected has to be around zero. When the robot starts to move, the detected road angles are plotted.

The graph in Figure 12 shows the detected road angle while the robot runs. The result comes up with a detected road angle from 0 to 10 degrees. The standard deviation of the detected road angle is 3.48

degrees. As seen in Figure 13, the lines detected on the road in front of ICE are formed from road markings on the left and right sides of the road. On the other hand, the lines detected on the pedestrian road in front of Glacido Cluster are formed from the difference of color, between the color of the grass and the color of the pedestrian road. The robot runs in environments 2 and 3 can be seen in Figure 14.

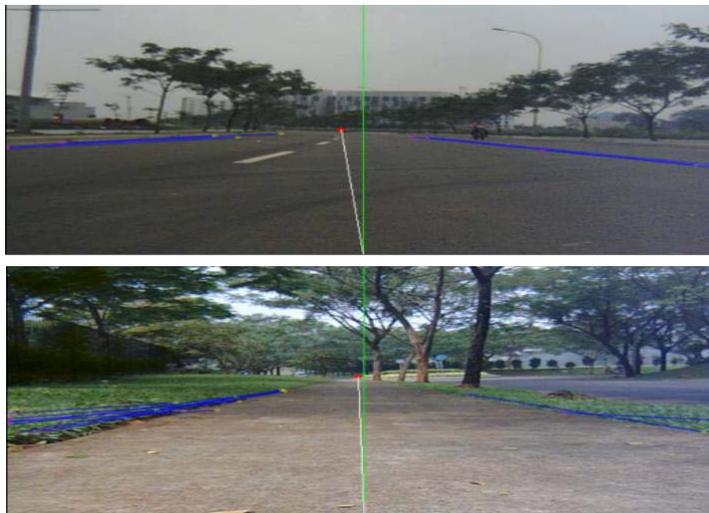


Figure 13. Detected vanishing point on environment 2 (top) and 3 (bottom)



Figure 14. The robot runs in environment 2 (top) and environment 3 (bottom)

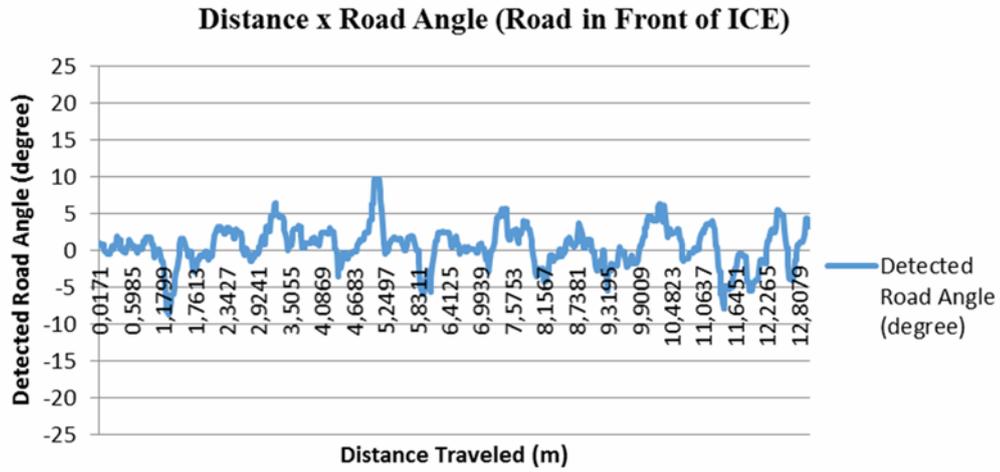


Figure 15. Graph of detected road angles in environment 2

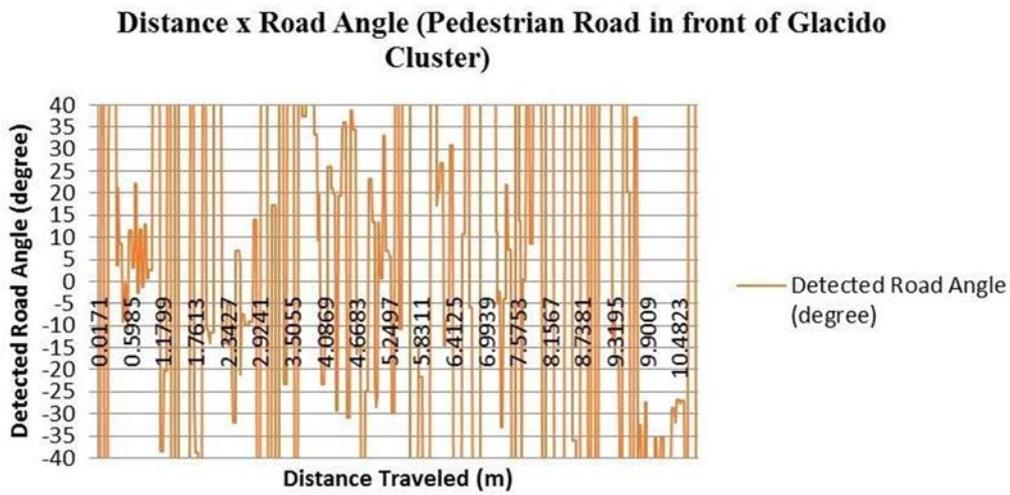


Figure 16. Graph of detected road angles in environment 3

The detected road angles in the road in front of ICE as shown in Figure 15 have values from 0 to 5 degrees. The standard deviation of the detected road angle is 2.73 degrees. The road is heading to the left therefore the robot is going to the left following the heading of the road.

On the other hand, the detected road angles in the pedestrian road in front of Glacido Cluster as shown in Figure 16 have values from 0 to more than 40 degrees. The standard deviation of the detected road angle is 53.1 degrees. The road is straight,

therefore a value of 53.1 degrees does not represent the actual heading of the road. This is caused by the high texture of the grass which causes disturbance to the line detection. The output of edge detection which indicates the high texture of the grass is shown in Figure 17.

The inaccurate detection of road angle causes the robot to follow the wrong direction. Therefore, during three testing times, the robot never reaches the end of the road and falls off from the road as shown in Figure 18.



Figure 17. Output of canny edge detection in environment 3



Figure 18. The robot falls from the road because of an inaccurate road angle

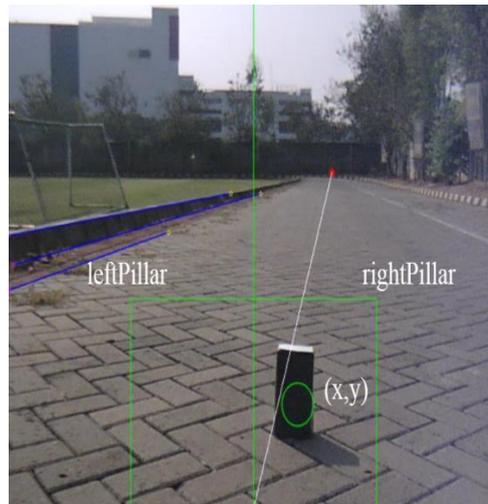


Figure 19. The robot avoids obstacle while following the heading of the road

obstacle position=	256 , 336	distance=	148.59
obstacle position=	248 , 334	distance=	137.16
obstacle position=	235 , 342	distance=	135.89
obstacle position=	229 , 347	distance=	124.46
obstacle position=	222 , 354	distance=	130.81
obstacle position=	211 , 357	distance=	101.6

Figure 20. Screenshot of a program when detecting the position and distance to the obstacle

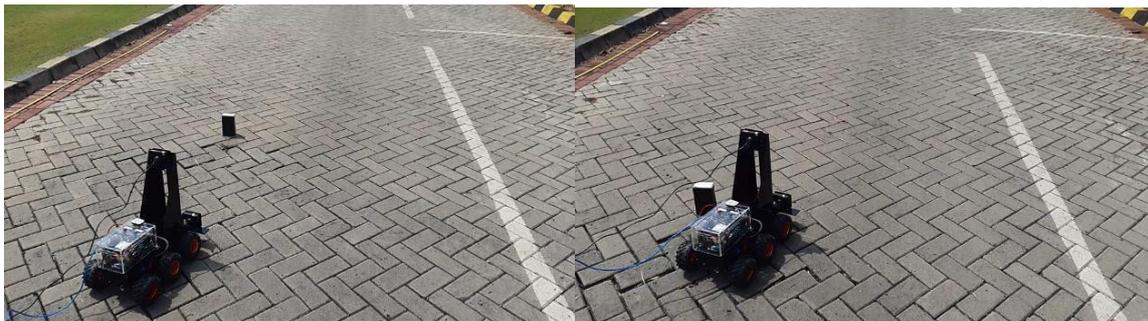


Figure 21. The robot detects an obstacle and avoids it

B. Obstacle detection and avoidance

When the robot is following the heading of the road, the robot also avoids obstacle that was detected by the camera. The result of the obstacle detection using a simple blob detector and ultrasonic sensor can be seen in Figure 19.

As seen in Figure 19, the obstacle will only be detected if the obstacle is located inside the detectable area (green box). If the obstacle is located in the left region, then the robot will turn right. And if the obstacle is located in the right region, then the robot will turn left. The vision system will only

detect the position of the obstacle (x,y) if it is located from distance between 130 cm to 200 cm from the robot. Next, after the position of the obstacle is known. The robot calculates the distance to the obstacle by using the ultrasonic sensor. The information about the position and distance to the obstacle can be seen in Figure 20.

After the obstacle is avoided, next the robot will follow the vanishing point again to go back to the road. As seen in Figure 21, the robot detects the obstacle in front of it by using a simple blob detector and measures the distance using an ultrasonic sensor.

IV. Conclusion

The road detection algorithm has made a good output of vanishing points. The vanishing point detected represents the heading of the road referred to the center of the image. From testing in 3 different roads (road beside soccer field at SGU Campus, road in front of ICE, and pedestrian road in front of Glacido Cluster), the heading of the road is detected by the vision system with plus-minus of 0 to 10 degrees and standard deviation of fewer than 5 degrees. Except for the road angles in the pedestrian road in front of Glacido Cluster which is inaccurate because of noises on detecting the road lines. The noises are caused by the high texture of the grass. For further recommendations, the implementation of IMU can be used to estimate the horizon line. The horizon line is used for vanishing point voting which results in better vanishing point filtering. The implementation of the encoder can be used to estimate the absolute heading of the robot and a color detection algorithm can be applied to navigate the robot in an environment where the road lines are hard to be detected.

Acknowledgement

The authors would like to thank Swiss German University for supporting the project with internal research funds.

Declarations

Author contribution

L. Rusli is the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no conflict of interest.

Additional information

Reprints and permission information is available at <https://mev.lipi.go.id/>.

Publisher's Note: Research Centre for Electrical Power and Mechatronics - Indonesian Institute of Sciences

remains neutral with regard to jurisdictional claims and institutional affiliations.

References

- [1] A. Cherubini, F. Spindler, and F. Chaumette, "A new tentacles-based technique for avoiding obstacles during visual navigation," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [2] R. A. Dasari, *Automatic Driving System by Recognizing Road Signs Using Digital Image Processing*. ProQuest Dissertations Publishing, 10196473, pp. 1-28, 2016.
- [3] Y. Kondo, M. Numada, H. Koshimizu, and I. Yoshida, "A Study on Fast and Robust Vanishing Point Detection System Using Fast M-Estimation Method and Regional Division for In-vehicle Camera," *J. of Electrical Engineering*, 6, 2018.
- [4] G. Yang, Y. Wang, J. Yang, and Z. Lu, "Fast and Robust Vanishing Point Detection Using Contourlet Texture Detector for Unstructured Road," *IEEE Access*, vol. 7, pp. 139358-139367, 2019.
- [5] J. Han, Z. Yang, G. Hu, T. Zhang, and J. Song, "Accurate and Robust Vanishing Point Detection Method in Unstructured Road Scenes," *J. Intell. Robot. Syst.*, 94, 143-158, 2019.
- [6] C. N. Khac, Y. Choi, J. H. Park, and H. Jung, "A Robust Road Vanishing Point Detection Adapted to the Real-world Driving Scenes," *Sensors*, 21, 2133, 2021.
- [7] Z. Yu and L. Zhu, "Roust Vanishing Point Detection Based on the Combination of Edge and Optical Flow," in *Proc. of the 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, Japan, pp. 184-188, 2019.
- [8] W. Yang, B. Fang, and Y. Y. Tang, "Fast and Accurate Vanishing Point Detection and Its Application in Inverse Perspective Mapping of Structured Road," *IEEE Trans. Syst. Man Cybern. Syst.*, 48, 755-766, 2018.
- [9] Y. Li, W. Ding, X. Zhang, and Z. Ju, "Road detection algorithm for Autonomous Navigation Systems based on dark channel prior and vanishing point in complex road scenes," *Robotics and Autonomous Systems*, 85, 2016.
- [10] S. Liu, Y. Zhou, and Y. Zhao, "VaPiD: A Rapid Vanishing Point Detector via Learned Optimizers," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12859-12868, 2021.
- [11] C. Chang, J. Zhao, and L. Itti, "DeepVP: Deep Learning for Vanishing Point Detection on 1 Million Street View Images," in *Proc. International Conference on Robotics and Automation (ICRA)*, pp. 4496-4503, 2018.
- [12] H. Kong, J. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2211-2220.
- [13] O. Miksik, "Rapid vanishing point estimation for general road detection," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [14] C. Siagian, C. Chang, R. Voorhies, and L. Itti, "Beobot 2.0: Cluster architecture for mobile robotics," *Journal of Field Robotics*, vol. 28, no. 2, pp. 278-302, March/April 2011.
- [15] C. Siagian, C. Chang, and L. Itti, "Mobile robot navigation system in outdoor pedestrian environment using vision-based road recognition," in *Proc. International Conference on Robotics and Automation*, 564-571, 2013.
- [16] N. Tokuda, T. Funahashi, M. Numada, and H. Koshimizu, "The Line Detection by Hough Transform for Vanishing Point Detection," Presented at *VIEW2012*, IS1-C6, Dec. 2012.