

COLLISION AVOIDANCE MOBILE ROBOT USING VECTOR FIELD HISTOGRAM ON POINT CLOUD KINECT

Devan Yusfa Sukmadya¹ Muhammad Fuad^{2*}

^{1,2}Department of Mechatronics, University of Trunojoyo Madura, Bangkalan, East Java, Indonesia
e-mail: devanyusfa15@gmail.com, fuad@trunojoyo.ac.id^{2*}

Article Information

Article History:

Received : 2 June 2026
Revised : 31 July 2026
Accepted : 18 October 2026
Published : 31 October 2026

*Correspondence:

fuad@trunojoyo.ac.id

Keywords:

collision avoidance, mobile robot, Kinect, vector field histogram

Copyright © 2026 by Author.
Published by Universitas Dinamika.



This is an open access article under the CC BY-SA license.

 10.37802/joti.v8i2.1437

Journal of Technology and Informatics (JoTI)

P-ISSN 2721-4842

E-ISSN 2686-6102

<https://e-journals.dinamika.ac.id/index.php/joti>

Abstract:

Autonomous mobile robots operating in indoor environments must be capable of avoiding humans as dynamic obstacles while maintaining reliable navigation performance. Although the Vector Field Histogram (VFH) algorithm has been widely applied for obstacle avoidance, most previous studies rely on two-dimensional sensors or do not directly integrate three-dimensional point cloud perception with real-time navigation on differential drive mobile robots. This limitation motivates the development of a more robust human-aware navigation system. Therefore, this study proposes a Differential Drive Mobile Robot (DDMR) that integrates Microsoft Kinect V1 point cloud sensing with the Vector Field Histogram (VFH) algorithm for real-time collision avoidance. Environmental depth information is converted into a three-dimensional point cloud, processed into a VFH polar histogram, and used to determine safe navigation directions during robot motion. Experimental evaluation was conducted in dynamic indoor environments involving human obstacles. The proposed system processes point cloud data at approximately 22 FPS, detects humans within a range of 0.8–3.5 m with a measurement error of 0.042 m, achieves an average VFH computation time of 0.05 s, and produces an average avoidance response time of 0.73 s with a collision-free success rate of 93.3%. These results demonstrate that integrating Kinect-based point cloud perception with VFH provides an effective and practical solution for real-time human-aware obstacle avoidance in indoor autonomous mobile robot navigation.

INTRODUCTION

The development of autonomous mobile robots has grown significantly, particularly in environments where robots must operate alongside humans, such as public spaces, hospitals, warehouses, and indoor facilities. These environments are rarely static; the unpredictable movement of humans makes human-aware obstacle avoidance a fundamental requirement for any practical autonomous robot system [1]. The Vector Field Histogram (VFH) algorithm has been extensively investigated for autonomous navigation due to its low computational complexity and real-time obstacle avoidance capability. In terms of algorithmic improvement,

Alkhafaji et al. [2] introduced the VFH+T algorithm to overcome local minimum problems, Mohammed et al. [10] enhanced VFH+ using fuzzy logic for quadcopters, and Kim et al. [16] proposed an improved VFH+ for UAV navigation in dense environments.

The extension of VFH into three-dimensional space has received significant attention: Zong et al. [1] developed a three-dimensional VFH for spatial path planning, Wang et al. [3] combined three-dimensional VFH+ with the Dynamic Window Approach (DWA) for UAV navigation, Zhao et al. [17] proposed a three-dimensional VFH-based method for autonomous aerial vehicles, and Zhang et al. [4] demonstrated VFH-based real-time obstacle avoidance for UAVs in confined spaces. For underwater domains, Zhang et al. [15] developed an intelligent VFH-based collision avoidance method for autonomous underwater vehicles, and Sun et al. [18] proposed an enhanced VFH* algorithm for underwater robot path planning. In the marine surface domain, Lee et al. [5] combined VFH+ with the Velocity Obstacle method for autonomous surface vessel collision avoidance.

Beyond single-robot applications, VFH has been adapted for multi-robot coordination: Sakamoto et al. [8] proposed a probabilistic VFH (p-VFH) for swarm robot navigation, Liang et al. [7] developed a multi-agent VFH+ flocking control method, and Liu et al. [9] and Liu et al. [20] introduced fish evasion behavior-based VFH methods for multi-UAV obstacle avoidance. Recognizing the limitations of purely reactive methods, Qasim and Al-Darraj [6] and Al-Darraj and Qasim [19] explored hybrid approaches integrating VFH with deep reinforcement learning and Soft Actor-Critic reinforcement learning, respectively, for improved adaptive decision-making in dynamic path planning.

In parallel, several studies have explored vector field-based methods that extend beyond the traditional VFH framework. Dahlin and Karayiannidis [11] proposed obstacle avoidance via tunnel-following Model Predictive Control (MPC) with adaptive guiding vector fields, Braquet and Bakolas [12] developed a vector field-based method for obstacles with time-varying elliptical shapes, Becker et al. [13] introduced informed circular fields for robotic manipulators, and Yao et al. [14] proposed guiding vector fields for following occluded paths. While these methods represent important theoretical extensions, they often introduce significant computational complexity that may limit real-time applicability on resource-constrained mobile robot platforms. In the context of ground-based indoor robots, Umam et al. [21] proposed a stereo vision-based obstacle avoidance system for an omni-directional robot, achieving a 93.3% success rate in static environments.

METHOD

This research employed an experimental quantitative approach to evaluate the effectiveness of integrating Kinect V1 point cloud sensing with the Vector Field Histogram (VFH) algorithm for real-time human obstacle avoidance. The study consisted of four sequential stages: system design and hardware development, implementation of Kinect-based point cloud processing and the VFH navigation algorithm, experimental evaluation in dynamic indoor environments with human obstacles, and performance analysis based on predefined evaluation metrics. The experiments were designed to answer the research question by assessing whether the proposed integration improves autonomous navigation reliability in dynamic environments.

Obstacle Avoidance System

Previous works that focused primarily on static obstacles, the authors utilize a 3-wheeled omnidirectional robot to navigate environments with moving pedestrians, identified

via Haar-cascade upper body detection, while tracking a circular target using the Hough Transform. The proposed system combines three core components: a Microsoft Kinect depth camera as the primary perception sensor, the vector field histogram (VFH) algorithm as the obstacle avoidance method, and a differential drive mobile robot (DDMR) kinematic model as the movement framework.

Robot Dimensional and Design

The robot is built on a three-layer PVC frame (each layer $354 \times 330 \times 5$ mm), supported by PLA-printed connectors with heights of 80 mm between the bottom and middle layers, and 180 mm between the middle and top layers. The Kinect sensor is mounted on the middle layer at a height of 167 mm, providing an optimal field of view for point cloud capture and target recognition. The electronics consist of an Arduino R4 Wi-Fi microcontroller (Renesas RA4M1, ARM Cortex-M4, 48 MHz) paired with an BTS7960 motor driver to control two DC PG45 motors (24 V, 440 RPM, 40 kgf-cm torque). Power is supplied by a Li-ion battery (24 V, 35800 mAh, 25–50C). when obstacles are detected within 10 mm. The Kinect connects to a laptop via USB, which runs the main processing software and communicates with the Arduino through a serial port.

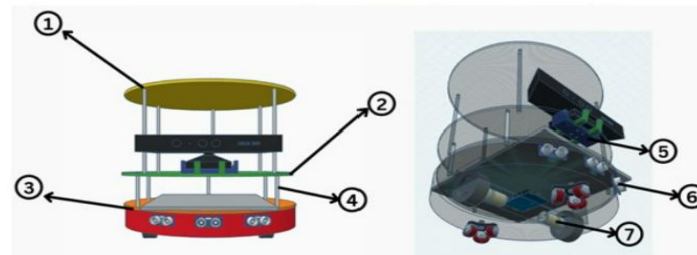


Figure 1. Design of mobile robot

Figure 1 illustrates the overall mechanical design of the autonomous mobile robot that consist of 1) the upper circular plate serves as the protective cover and mounting platform for the robot's upper structure; 2) the vertical support frame connects the upper and lower platforms, providing structural rigidity while accommodating the placement of electronic components; 3) the front lower housing integrates the ultrasonic sensors, which are used to detect nearby obstacles and measure distances during navigation; 4) the middle platform functions as the primary mounting area for the embedded controller and other electronic modules; 5) the Kinect V1 camera sensor is installed at the front of the robot to perform real-time environmental scanning for mapping and obstacle detection; 6) the main controller board, positioned on the lower platform, processes sensor data and executes the robot's navigation and control algorithms; 7) the drive wheel system provides the robot's locomotion, enabling forward, backward, and turning movements during autonomous operation.

Scenario Research and Kinematics

The Target Switching module then manages transitions between waypoints to keep the robot progressing toward its goal in an orderly manner. Finally, these commands are executed by the DDMR kinematic model, translating them into actual differential wheel velocities sent to the drive motors, completing the closed-loop navigation cycle from initial location P_i toward target position P_t . Figure 2 shows DDMR operating as a closed-loop control flow. The Kinect sensor captures two data streams: color image data I_{RGB} , processed by the Target Seeking module to detect the Aruco marker α , and depth image data I_{Depth} (I_{Depth}), processed by the Create Point Cloud module to reconstruct a point cloud data of the surrounding environment

within the camera's field of view.

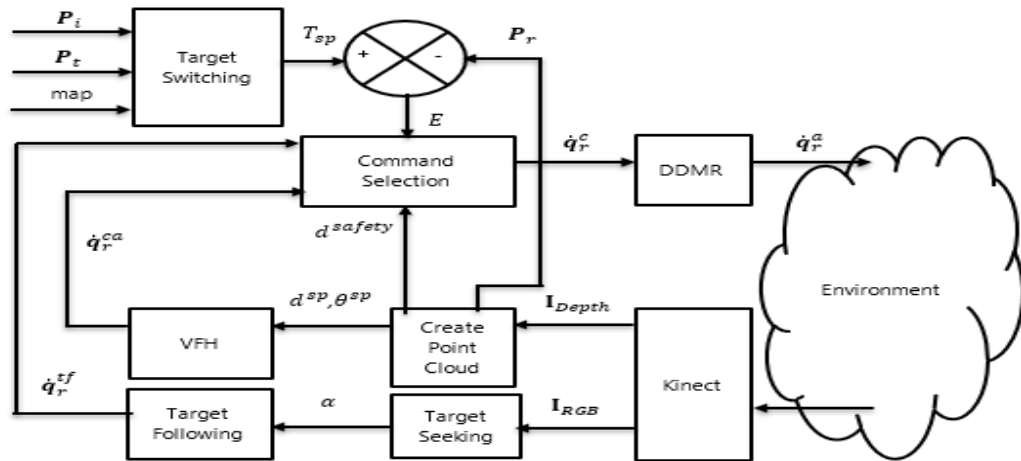


Figure 2. Block Diagram of point cloud-based VFH for mobile robot navigation

The extracted target data α submitted into Target Following module to produce motor command for target following $\dot{q}_r^{tf}$. The point cloud data used to extract distance d^{sp} and angle θ^{sp} passed into VFH to produce motor command for collision avoidance $\dot{q}_r^{ca}$. Command Selection evaluates the best motor command based on safety distance d^{safety} to pass to DDMR as robot command $\dot{q}_r^c$. DDMR moves based on these commands resulting actual robot velocity $\dot{q}_r^a$.

In the Figure 3, the movement of the robot is modeled using differential drive mobile robot (DDMR) kinematics, which describes the relationship between wheel velocities and the robot's position and orientation in a 2D Cartesian coordinate system.

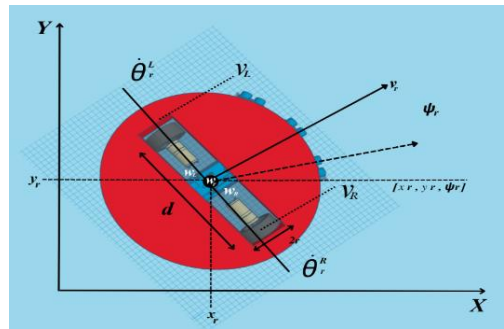


Figure 3. Kinematic model of DDMR

The following describes the equations used in the kinematic model, as shown in equations below.

$$\mathbf{P}_r = [x_r, y_r, \psi_r]^T \quad (1)$$

$$\dot{\mathbf{P}}_r = [\dot{x}_r, \dot{y}_r, \dot{\psi}_r]^T \quad (2)$$

$$\mathbf{q}_r = [\dot{\theta}_r^L, \dot{\theta}_r^R]^T \quad (3)$$

Where the actual robot position $\mathbf{P}_r$ is represented by x_r and y_r and the orientation is represented by ψ_r , $\mathbf{q}_r$ represents the robot's wheel velocity. The configuration transition of the robot from base to coordinate $\dot{\mathbf{P}}_r$ is expressed below.

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\psi}_r \end{bmatrix} = \begin{bmatrix} \cos \psi_r & 0 \\ \sin \psi_r & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (4)$$

The position change using forward kinematics with linear velocity v_r and angular velocity ω_r of the robot, based on the left wheel velocity $\dot{\theta}_r^L$ and right wheel velocity $\dot{\theta}_r^R$ of the mobile robot, is represented in Equation below.

$$\begin{bmatrix} v_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ 1/d & 1/-d \end{bmatrix} \begin{bmatrix} \dot{\theta}_r^L \\ \dot{\theta}_r^R \end{bmatrix} \quad (5)$$

The position change using inverse kinematics with the robot's wheel velocity q_r in terms of linear and angular velocity is represented in Equation below.

$$\begin{bmatrix} \dot{\theta}_r^L \\ \dot{\theta}_r^R \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & d/2 \\ 1 & -d/2 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (6)$$

The error threshold determines when the robot needs to apply correction. Prevents the robot from reacting to insignificant small differences, such as sensor noise or minor fluctuations. Formula used to calculate the error value and compare it against the threshold is expressed in Equation below.

$$E = T_{sp} - P_r \quad (7)$$

Where E is the error, T_{sp} is the target setpoint (target position), and P_r is the actual robot position, as shown in Equation below.

$$|E| > \tau \quad (8)$$

Where τ is the threshold value. If $|E| > \tau$, correction to the direction or velocity is applied. If $|E| \leq \tau$, the robot continues moving without additional correction, as shown in Equation below.

$$C = K \times E \quad (9)$$

Where C is the correction value, K is a proportional constant, and E is the error value between the target position and the actual position. The following flowchart in Figure 4. The flowchart System Scenario in Figure 4, describes the operational sequence of the robot navigating toward a target setpoint (T_{sp}) while avoiding humans as dynamic obstacles in an indoor environment. The process begins with system initialization, loading the robot's actual position (P_r), target setpoint (T_{sp}), and Kinect sensor data (IRGB, IDepth). The robot then moves toward T_{sp} . The detail process of VFH is described in Flowchart VFH on Figure 5. Figure 5 illustrates the VFH algorithm flowchart that governs the robot's obstacle avoidance decision process. The system starts by reading both depth (IDepth) and color (IRGB) data from the Kinect, which are then used to build the VFH polar histogram. Parameters (P_r , P_r , q_r , q_r). If no obstacle is detected, the system checks whether an aruco marker is visible if detected, the robot navigates toward the traffic cone as the target finish point. If the aruco marker is not detected, the system stops. Once all conditions are resolved, the system outputs the commanded wheel velocities ω_c , ω_c to drive the robot, and the process ends. In Kinect Camera Navigation and obstacle avoidance based on point cloud data obtained from the Kinect sensor. Each point in the point cloud has coordinates (x , y , z) in 3D space. To detect the robot's path and obstacles, each point is projected into the camera coordinate system in Figure 6. To detect the robot's path and

obstacles, each point is projected into the camera coordinate system using Equation (10).

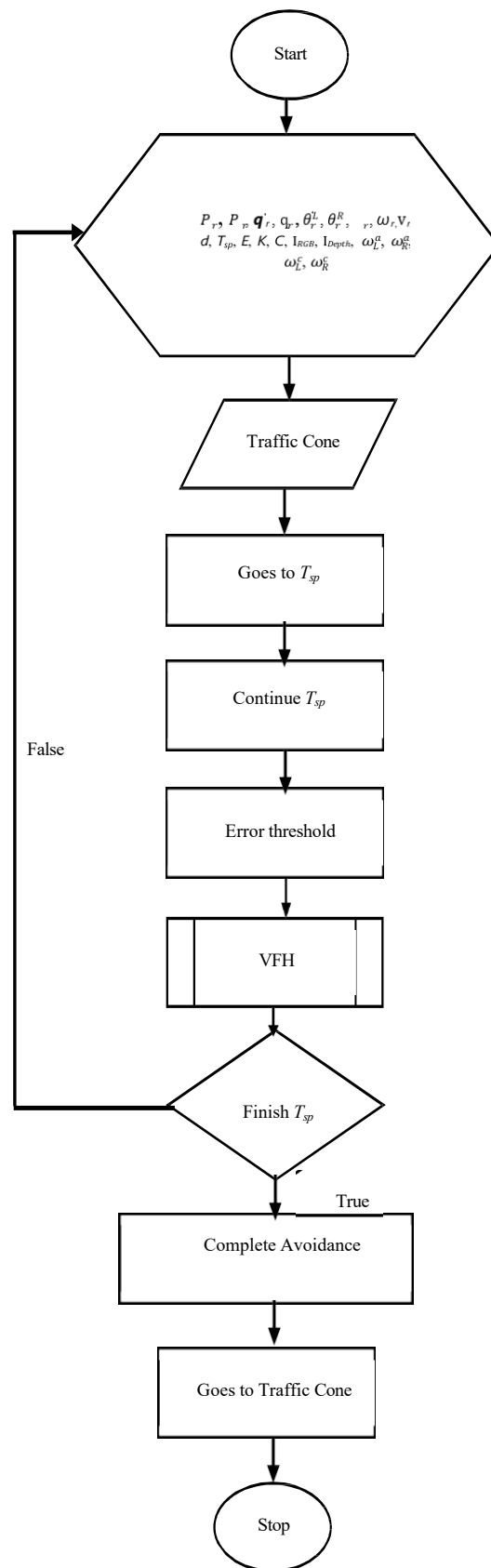


Figure 4. Flowchart system scenario

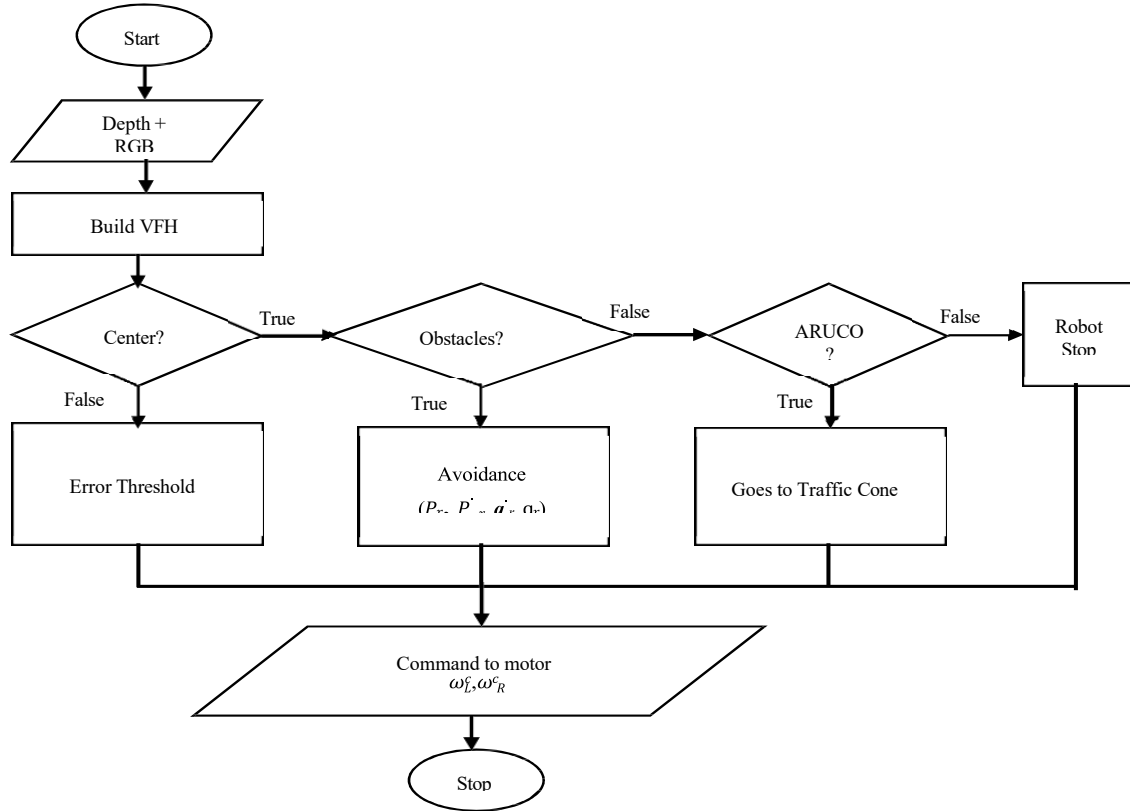


Figure 5. Flowchart VFH

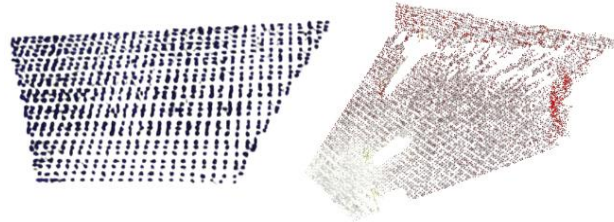


Figure 6. Point cloud data obtained from the kinect sensor

$$X = \frac{(u-c_x).z}{f_x}, Y = \frac{(v-c_y).Z}{f_y}, Z = (u-v)z \quad (10)$$

where:

u, v : Pixel coordinates in the 2D image

$Z(u, v)$: Pixel depth value (from depth Image)

c_x, c_y : Camera principal point (center coordinates)

f_x, f_y : Camera focal lengths in pixels

Sector low indicates that the robot's path is defined as a safe passage with no obstacles, while Sector High indicates that an obstacle is present in the robot's path, as shown in Figure 6. To classify the environment around the robot, each point in the point cloud is evaluated against a threshold to determine whether it belongs to sector low or sector high.

A point is classified as sector low when $y_i \leq 0.01$ m and $z_i \leq 0.18$ m, indicating a safe path as shown in Equation (11).

$$y_i = \leq 0.01 \text{ m, on } z_i = \leq 0.18 \text{ m. In } |x_i - x_{\text{robot}}| \quad (11)$$

Conversely, a point is classified as sector high when $y_i > 0.01 \text{ m}$ and $z_i > 0.18 \text{ m}$, indicating the presence of an obstacle as shown in Equation (12).

$$y_i = > 0.01 \text{ m, on } z_i = > 0.18 \text{ m. In } |x_i - x_{\text{robot}}| \quad (12)$$

After classifying each point into sector low or sector high, the obstacle within sector high is calculated. The higher the number of points classified as sector high in a given sector, the greater the obstacle density. The VFH polar histogram divides the robot's surrounding space into five sectors ranging from 0° to 180° , where each sector represents a specific angular interval around the robot. The sectors are assigned values of $[-1.0, -0.5, 0.0, 0.5, 1.0]$, representing directions from far-left to far-right relative to the robot's forward orientation, as shown in Figure 7.

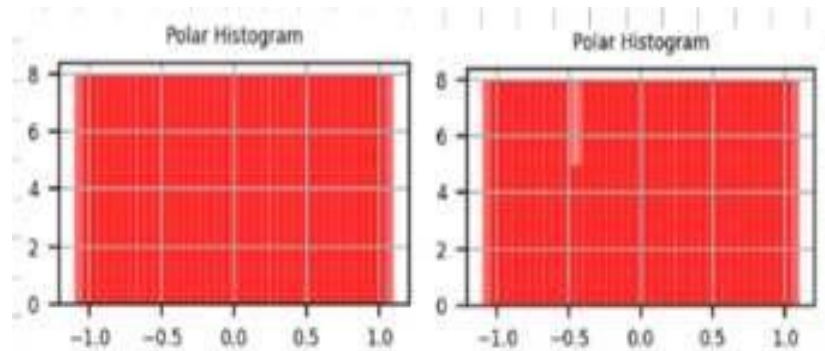


Figure 7. VFH polar histogram on field of view robot

The polar histogram divides the robot's field into five sectors from 0° to 180° , with the center sector at 90° representing the robot's main forward direction. The sector distribution is summarized in Table 1.

Table 1. Sector Distribution Visualization

Angle (degree)	Description
0	$[-1.0]$ Indicates far-left
45	$[-0.5]$ Indicates center-left
90	
135	
180	$[1.0]$ Indicates far-right

Each sector corresponds to a 45° angular interval within the Kinect field of view. Sector 0° represents the far-left region, followed by 45° (center-left), 90° (forward), 135° (center-right), and 180° (far-right). Obstacle density within each sector is calculated from point cloud measurements acquired within the effective sensing distance of $0.8\text{--}3.5 \text{ m}$ as shown in Figure 8.

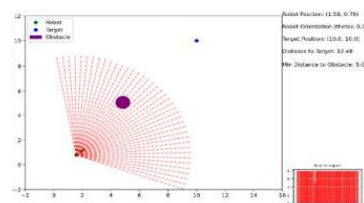


Figure 8. Candidate valley on show in simulation

VFH algorithm prior to physical implementation, a simulation was conducted to observe the robot's obstacle avoidance behavior in a controlled environment. The simulation demonstrates how the polar histogram responds to detected obstacles and how the candidate valley is selected to determine the robot's navigation direction, as shown in Figure 9.

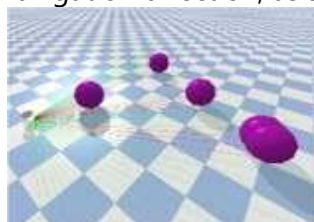


Figure 9. VFH collision avoidance scenario on pybullet simulation

The simulation is conducted to verify the VFH algorithm behavior before physical implementation on the robot. Simulation models the complete VFH process including point cloud, sector classification, obstacle density calculation, and candidate valley selection.

RESULTS AND DISCUSSION

1. Target Following (Aruco Marker)

First test evaluates the ability of the robot to detect and follow the Aruco marker as the navigation target using IRGB data from the Kinect camera. Show on Figure 10.



Figure 10. Target Following on Mobile Robot

Results show that the robot successfully detects and follows the Aruco marker as the navigation target in real time. The Kinect camera captures IRGB data and correctly identifies the marker position, allowing the robot to orient itself and move toward the target setpoint (Tsp) as shown result with encoder read on Figure 11.

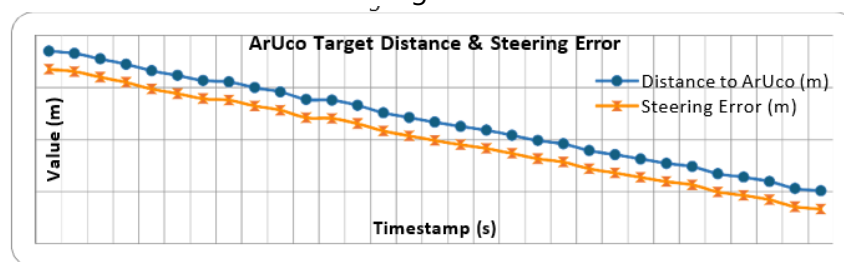


Figure 11. Aruco Target Distance & Steering Error from csv

2. VFH Polar Histogram with Point Cloud

Performance of the VFH algorithm in processing point cloud data from the Kinect sensor to generate a polar histogram for obstacle detection and avoidance direction selection. The polar histogram divides the robot's field of view into five sectors from 0° to 180° , where each sector is classified as Sector Low (safe) or Sector High (obstacle) based on the density of detected points within that sector. The results of this test demonstrate how the robot responds to human presence as a dynamic obstacle in real time, as shown in Figure 12.



Figure 12. Real-Time VFH Polar Histogram with Point Cloud from Kinect

When the Kinect sensor detects a human as an obstacle within the robot's path, the point cloud data is immediately processed and mapped into the polar histogram. The affected sector is classified as sector high, indicating the presence of an obstacle, and the VFH algorithm identifies the candidate valley — the nearest sector with the lowest obstacle density — as the safest navigation direction. The robot then adjusts its wheel velocities accordingly to steer away from the detected obstacle, as shown in Figure 13.



Figure 13. Obstacle shown on right side of robot in point cloud

3. COLLISION AVOIDANCE MOBILE ROBOT USING VECTOR FIELD HISTOGRAM ON POINT CLOUD KINECT

The results demonstrate that the robot successfully detects the human using point cloud data from the Kinect sensor and responds directly by selecting the appropriate candidate valley through the VFH algorithm, allowing the robot to steer around the obstacles and continue moving toward the target without collision, as shown in Figure 14.



Figure 14. Robot avoid human obstacles when detected on point cloud

Visual odometry is used to estimate the robot's position and movement by analyzing the change in point cloud data captured by the Kinect sensor across consecutive frames. This allows the robot to track its own displacement and orientation in real time without relying on external positioning systems, supporting the accuracy of navigation toward the target setpoint, as shown in Figure 15.

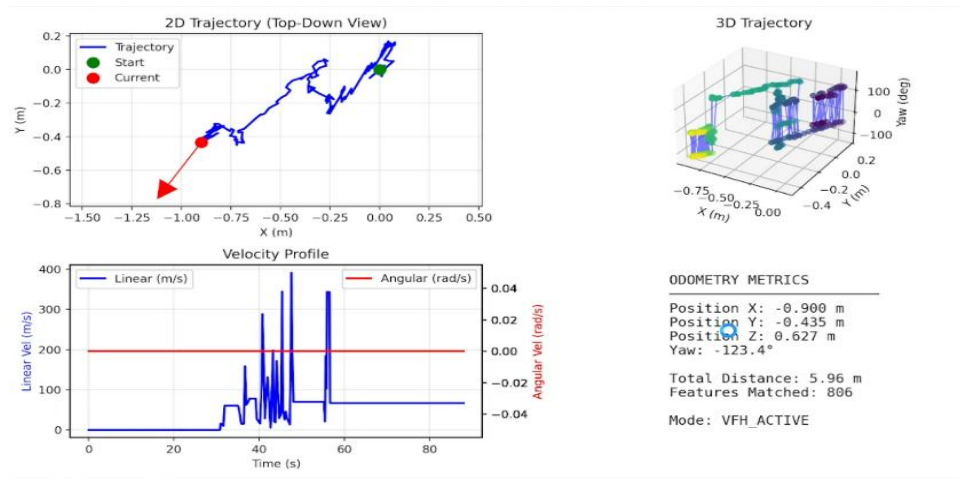


Figure 15. Visual odometry shown when robot moving

4. Comparison.

The proposed method with the work of Umam (2023), Obstacle Avoidance Based on Stereo Vision Navigation System for Omni-directional Robot. Both studies achieved the same obstacle avoidance success rate of 93.3%. However, Umam (2023) utilized stereo vision for obstacle detection, whereas the proposed system employed Kinect V1 point cloud data that directly provides depth information.

Table 2. Comparison

Parameter	Umam (2023) [21]	Proposed Method (this paper)
Method	Stereo Vision Navigation	Vector Field Histogram (VFH)
Sensor	Stereo Vision Camera	Kinect V1
Environment Data	Stereo Image	3D Point Cloud
Robot Type	Omni-directional Robot	DDMR
Obstacle Type	Static Obstacle	Dynamic Human Obstacle

Table 2 summarizes the comparison between the proposed Kinect-VFH navigation framework and the stereo vision-based obstacle avoidance approach proposed by Umam et al. [21]. The comparison highlights differences in sensing modality, robot platform, obstacle type, and navigation strategy. The evaluation considered obstacle detection capability, collision occurrence, avoidance response time, path deviation, and overall navigation success. The detailed experimental results are summarized in Table 3.

Table 3. Experimental Evaluation of Human Obstacle Avoidance Performance

Trial	Human Position	Detection	Collision	Avoidance Time (s)	Success
1	Front	Yes	No	0.71	Yes
2	Left	Yes	No	0.69	Yes
3	Right	Yes	No	0.75	Yes
4	Front-Left	Yes	No	0.73	Yes
5	Front-Right	Yes	Yes	0.76	Yes

CONCLUSIONS AND SUGGESTIONS

This study addressed the problem of achieving reliable real-time human obstacle avoidance for Differential Drive Mobile Robots operating in indoor environments through the integration of Kinect V1 point cloud. Despite the promising results, several limitations were identified during this study. The Kinect sensor is limited to indoor environments with controlled lighting conditions, as direct sunlight or reflective surfaces can interfere with depth data quality. The effective detection range of 0.8–3.5 meter also restricts the system's applicability in larger open spaces. For future development. Out of 15 experimental trials, 14 were successfully completed without collision, resulting in an overall collision-free success rate of 93.3%. The average avoidance response time was 0.73 s, while the average path deviation remained 0.12 m, demonstrating the effectiveness of the proposed Kinect–VFH navigation system.

ACKNOWLEDGMENTS

Thanks also to the Department of Mechatronics at the University of Trunodjoyo Madura for providing the facilities and support that made this research possible.

REFERENCES

- [1] Zong, C., Du, Q., Chen, J., Shan, Y., Wu, Y., & Sha, Z., 2024. "A Design of Three-Dimensional Spatial Path Planning Algorithm Based on Vector Field Histogram*." *Sensors Journal*, Volume 24 Issue 17, Article 5647. DOI:10.3390/s24175647
- [2] Alkhafaji, A., Al-Samarraie, M., & Abdullah, H., 2024. "Optimizing Autonomous Navigation in Unknown Environments: A Novel Trap Avoiding Vector Field Histogram Algorithm (VFH+T)." *Results in Engineering*, Volume 24. DOI:10.1016/j.rineng.2024.102625
- [3] Wang, X., Cheng, M., Zhang, S., & Gong, H., 2023. "Multi-UAV Cooperative Obstacle Avoidance of 3D Vector Field Histogram Plus and Dynamic Window Approach." *Drones Journal*, Volume 7 Issue 8. DOI: 10.3390/drones7080504
- [4] Zhang, Y., Li, J., & Sun, K., 2023. "The Control Method of Autonomous Flight Avoidance Barriers of UAVs in Confined Environments." *Sensors Journal*, Volume 23 Issue 13. DOI: 10.3390/s23135896
- [5] Lee, H.G., Kim, J.H., & Park, S.K., 2024. "Collision Avoidance Algorithm of Autonomous Surface Vessel Using Vector Field Histogram and Velocity Obstacle Method." *Journal of Advanced Navigation Technology*, Volume 31 Issue 2. DOI:10.3390/app14219693
- [6] Qasim, M.H., & Al-Darraj, S., 2026. "A Hybrid Approach Using Vector Field Histogram and Deep Reinforcement Learning for Dynamic Path Planning." *Iraqi Journal for Electrical and Electronic Engineering*, Volume 22 Issue 1. DOI: 10.37917/ijeee.22.1.34
- [7] Liang, W., Chen, Y., & Zhao, T., 2025. "Multi-Agent VFH+ Flocking Control with Escaping Dynamic Equilibrium under Complex Obstacles." *Complex & Intelligent Systems Journal*. DOI: 10.1016/j.jfranklin.2024.107181
- [8] Sakamoto, K., Tanaka, H., & Fujimoto, Y., 2025. "Collision-Based Probabilistic VFH (p-VFH) Obstacle Avoidance Algorithm for Swarm Robots Navigation in Unknown Environment." *Advanced Robotics Journal*, Volume 39 Issue 4. DOI:10.1080/01691864.2025.2530515
- [9] Liu, W., Zhang, Y., & Wang, H., 2025. "A Fish Evasion Behavior-Based Vector Field Histogram Method for Obstacle Avoidance of Multi-UAVs." *Aerospace Science and*

- Technology, Volume 158. DOI: 10.1016/j.ast.2025.109974
- [10] Mohammed, K., Aliedani, A., & Al-Ibadi, A., 2024. "Adaptive Vector Field Histogram Plus (VFH+) Algorithm using Fuzzy Logic in Motion Planning for Quadcopter." *Journal of Robotics and Control (JRC)*, Volume 5 Issue 2. DOI:10.18196/jrc.v5i2.21540
 - [11] Dahlin, A., & Karayiannidis, Y., 2023. "Obstacle Avoidance in Dynamic Environments via Tunnel-Following MPC with Adaptive Guiding Vector Fields." *IEEE Robotics and Automation Letters*, Volume 8 Issue 4. DOI:10.48550/arXiv.2303.15869
 - [12] Braquet, M., & Bakolas, E., 2022. "Vector Field-Based Collision Avoidance for Moving Obstacles with Time-Varying Elliptical Shape." *IEEE Control Systems Letters*, Volume 6. DOI: 10.1016/j.ifacol.2022.11.246
 - [13] Becker, M., Schillinger, P., & Buss, M., 2022. "Informed Circular Fields for Global Reactive Obstacle Avoidance of Robotic Manipulators." *Robotics and Autonomous Systems*, Volume 157. DOI: 10.3389/frobt.2024.1447351
 - [14] Yao, W., Li, T., & Zhang, J., 2022. "Guiding Vector Fields for Following Occluded Paths." *IEEE Access*, Volume 10. DOI:10.1007/s40789-026-00898-2
 - [15] Zhang, G., Liu, X., & Chen, Z., 2022. "Intelligent Vector Field Histogram Based Collision Avoidance Method for Autonomous Underwater Vehicles." *Ocean Engineering*, Volume 270. DOI:10.1016/j.oceaneng.2022.112525
 - [16] Kim, J., Lee, S., & Park, D., 2023. "Improved VFH+ for Real-Time UAV Navigation in Dense Environments." *Drones Journal*, Volume 7. DOI: 10.18196/jrc.v5i2.21540
 - [17] Zhao, Y., Wang, X., & Chen, L., 2023. "Three-Dimensional VFH-Based Local Obstacle Avoidance for Autonomous Aerial Vehicles." *Sensors Journal*, Volume 23 Issue 13. DOI:10.1155/2021/8819618
 - [18] Sun, H., Zhao, M., & Li, K., 2024. "Enhanced VFH* Algorithm for Underwater Robot Path Planning." *Sensors Journal*, Volume 24 Issue 17. DOI: 10.3390/s24175647
 - [19] Al-Darraj, S., & Qasim, M.H., 2026. "Soft Actor-Critic Reinforcement Learning Integrated with VFH for Dynamic Robot Navigation." *Iraqi Journal for Electrical and Electronic Engineering*, Volume 22 Issue 1. DOI:10.3390/jsan14050087
 - [20] Liu, W., Wang, H., & Zhang, Y., 2025. "Dynamic Multi-UAV Obstacle Avoidance Using Fish-Evasion-Based VFH Navigation." *Aerospace Science and Technology*, Volume 158. DOI: 10.3390/jmse12030412
 - [21] Umam, F., Fuad, M., Suwarno, I., Ma'arif, A., & Caesarendra, W., 2023. "Obstacle Avoidance Based on Stereo Vision Navigation System for Omni-directional Robot." *Journal of Robotics and Control (JRC)*, Volume 4 Issue 2. DOI: 10.18196/jrc.v4i2.17977