https://doi.org/10.31436/iiumej.v26i3.3554

FRONTIER-BASED DETECTION AND SOCIAL FORCE MODEL FOR AUTONOMOUS ENVIRONMENT MAPPING AND NAVIGATION

ASYAM IRSYAD[1], BIMA SENA BAYU DEWANTARA[2*], SETIAWARDHANA[2]

[1]*Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia*
[2]*Computer Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia*

**Corresponding author: bima@pens.ac.id*

(Received: 29 December 2024; Accepted: 6 August 2025; Published online: 9 September 2025)

ABSTRACT: This paper presents a method for static environment mapping using an autonomous mobile robot by integrating the Social Force Model (SFM) for obstacle avoidance and frontier-based detection for dynamic goal selection. The objective is to enable the robot to autonomously explore unknown environments, avoid obstacles, and generate accurate maps. The system was developed and tested in the Gazebo simulator, with RViz employed to visualize real-time sensor data, including lidar and odometry. Two frontier point selection strategies were evaluated: one based solely on the nearest distance, and another incorporating both distance and orientation. Experimental results demonstrate that the first strategy enabled the robot to complete the mapping task in an average of 10.66 minutes over a distance of 76.37 meters. In contrast, the second strategy, although more directionally aware, resulted in a longer duration of 14 minutes and a travel distance of 104.08 meters. These outcomes highlight the impact of map update delays and suboptimal frontier point selection on overall navigation efficiency. The findings suggest that while the system exhibits promising autonomous navigation capabilities, further improvements in frontier map update speed and goal point optimization are necessary to enhance path efficiency and reduce unnecessary movement.

ABSTRAK: Kertas ini membentangkan kaedah pemetaan persekitaran statik menggunakan robot mudah alih autonomi dengan mengintegrasikan Model Daya Sosial (SFM) bagi mengelak halangan dan pengesanan berasaskan sempadan bagi pemilihan matlamat dinamik. Objektifnya adalah untuk robot meneroka persekitaran yang tidak diketahui secara autonomi, mengelak halangan, dan menjana peta dengan tepat. Sistem ini dibangunkan dan diuji dalam simulator Gazebo, manakala RViz digunakan bagi memvisualisasikan data sensor masa nyata termasuk input lidar dan odometri. Dua strategi pemilihan titik frontier telah dinilai: satu berdasarkan jarak terdekat sahaja, dan satu lagi yang mengambil kira jarak serta orientasi. Dapatan eksperimen menunjukkan bahawa strategi pertama membolehkan robot melengkapkan tugas pemetaan dalam purata masa 10.66 minit dengan jarak perjalanan 76.37 meter. Manakala, strategi kedua, walaupun lebih peka terhadap arah, memerlukan masa yang lebih lama iaitu 14 minit dengan jarak perjalanan 104.08 meter. Dapatan kajian ini menunjukkan bahawa kelewatan dalam mengemas kini peta dan pemilihan titik frontier yang kurang optimum memberi kesan kepada kecekapan navigasi keseluruhan. Walaupun sistem ini menunjukkan kebolehan navigasi autonomi yang baik, penambahbaikan dalam kelajuan kemas kini peta frontier dan pengoptimuman pemilihan titik matlamat diperlukan bagi meningkatkan kecekapan laluan dan mengurangkan pergerakan yang tidak perlu.

KEYWORDS: *Autonomous Navigation, Social Force Model (SFM), Simultaneous Localization and Mapping (SLAM), Frontier Detection, Robot Exploration.*

1. INTRODUCTION

Robotic technology has greatly sped up its development in response to the growing need for more difficult tasks and roles [1-6]. To fulfill the need for human-robot interaction, psychological and social factors are increasingly being considered [7-10]. Along with hardware systems that function well, software and algorithms, such as those used in detection and analysis, navigation, object avoidance, localization, and mapping, are also crucial.

One essential capability that a mobile robot must possess is the ability to navigate within a specific environment. Typically, the environment where the robot will operate can be mapped beforehand. When applied in environments with static obstacles (such as walls and furniture) and dynamic ones (including humans, animals, and other moving objects), a human-operated robot must first map the area [11, 12]. Making maps can also be done by a mobile robot on its own in places where there are only static barriers [13]. Consequently, the majority of robots encounter difficulties navigating unfamiliar environments [14, 25-27].

Furthermore, mapping the area is not difficult for either the robot or the operators in structured and safe environments. However, in unstructured or dangerous areas, a human user might not be present, so the robot needs to be able to map the area on its own. Additionally, the need for a workforce is also something to consider [15]. This study combines Simultaneous Localization and Mapping (SLAM) technology with local navigation tools, particularly the Social Force Model (SFM) [9, 17, 21-23]. This enables a mobile robot to navigate autonomously while identifying its location and interpreting its surroundings. Being able to move independently means that the robot can avoid both static and moving objects while it is working.

SLAM-toolbox was selected in this research due to its robust features and practical advantages [20]. SLAM-toolbox not only simplifies the implementation of SLAM by providing a ready-to-use framework, eliminating the need to develop SLAM algorithms from scratch, which can be highly time-consuming, but it also incorporates occupancy grid mapping. This feature is particularly beneficial as it allows seamless integration with additional algorithms, such as frontier-based exploration, to enhance mapping and navigation capabilities in diverse environments [14, 25-27]. The reason SFM was chosen is that it is easy to use, can accurately model social interactions, can avoid collisions, works well in dynamic environments, and can adapt to various situations [10].

The main contributions of integrating the Social Force Model (SFM) navigation method with SLAM-toolbox and frontier-based exploration are:

- The integration of SFM and SLAM-toolbox enables the robot to autonomously navigate dynamic environments, such as areas with human activity or moving objects, by accounting for social interactions. This ensures that the robot can not only build and maintain an accurate map of the environment but also safely avoid collisions with humans and dynamic obstacles.
- SLAM-toolbox provides a robust framework for real-time map generation and updates using occupancy grids. At the same time, the inclusion of frontier-based exploration allows the robot to identify unexplored areas efficiently. Coupled with SFM, the robot adapts to dynamic environmental changes, maintaining safety and mapping precision during autonomous navigation and exploration.
- The SFM allows the robot to mimic human-like navigation behavior, maintaining safe distances and moving naturally in crowded or socially sensitive environments. This ensures

comfortable interactions and reduces potential disruptions in human-robot interaction scenarios.

- The combination of SFM, SLAM-toolbox, and Frontier-based Detection provides the robot with adaptability in diverse scenarios, whether in static or dynamic environments, and enhances its capability to explore both confined and open spaces with varying population densities.
- This integration significantly increases the robot's autonomy by enabling it to navigate, explore, and interact with dynamic and complex environments without human intervention. The robot can dynamically adjust its behavior while simultaneously updating its map and planning exploration paths to maximize coverage.

The remainder of this paper is organized as follows. Section 2 describes the mobile robot navigation, frontier detection, localization, and mapping system simultaneously. Section 3 delivers the experimental results and discussion. Finally, the conclusion and future work are presented in Section 4.

2. LITERATURE REVIEW

2.1. Social Force Model (SFM)

The Social Force Model (SFM) predicts human movement in dynamic environments [17]. It considers "social forces," which reflect not just physical forces but also the desire to achieve goals while avoiding obstacles and interacting with others. This model is particularly effective for socially aware navigation in robots.

In robotics, the SFM helps robots avoid fixed and moving obstacles by calculating the necessary force to navigate. The total navigation force (F_{nav}) refer to Eq. (1) is the sum of two components: the force to move toward the goal (F_g) refer to Eq. (2) and the force from static obstacles (F_s). Here, v^0 represents the desired velocity of the robot, while v is the current velocity. The sum of physical and social repulsion is a static force, referred to as Eq. (3). This force keeps the robot away from a stationary object.

$$F_{nav} = F_g + F_s \tag{1}$$

$$F_g = m \cdot a = m\frac{v^0 - v}{t} \tag{2}$$

$$F_s = f_{soc}^s + f_{phy}^s \tag{3}$$

$$f_{soc}^s = k^s exp + \left(\frac{r_R - d_R^s}{\psi^s}\right) \cdot \overrightarrow{e_s} \tag{4}$$

$$f_{phy}^s = k^s \cdot (r_R - d_R^s) \cdot \overrightarrow{e_s} \tag{5}$$

The social robot's repulsive force against static obstacles is denoted as f_{soc}^s refer to Eq. (4), while its physical repulsion force is f_{phy}^s as seen as Eq. (5). The robot's radius of the proxemics area is r_R, its distance to the nearest static obstacle is d_R^S, and a gain factor k^s determines future feedback. The effective distance value for avoiding dynamic obstacles is ψ^s, and $\overrightarrow{e_s}$ represents the vector direction of the working force. The vector of SFM is represented in Fig. 1.

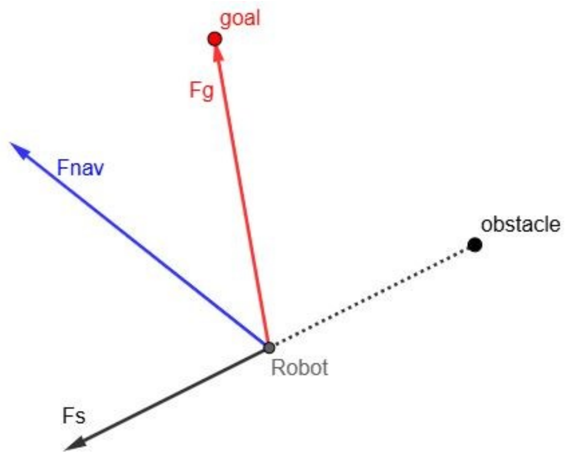


Figure 1. The forces that operate within the SFM framework.

2.2. Simultaneous Localization and Mapping

Simultaneous Localization and Mapping, or SLAM, is a way for robots or self-driving cars to make maps and figure out where they are on the map simultaneously. In this case, SLAM uses algorithms, mathematical models, and sensor data to assist robots in determining their position and actions within unfamiliar environments. Besides that, the SLAM software system can also figure out where the robot is in its surroundings [18].

SLAM-toolbox is a widely used framework in robotics for simultaneous localization and mapping (SLAM), enabling a robot to construct a map of its surroundings while simultaneously determining its position within the environment. SLAM-toolbox is particularly effective for applications requiring high-precision 2D or 3D mapping, especially in structured indoor settings with static features [20]. The key processes and formulations implemented in SLAM-toolbox are detailed in Algorithm 2.

2.3. Robot Model

The Differential Drive Mobile Robot (DDMR) Kinematic function determines the wheel speed of the robot, which requires kinematics, allowing it to calculate the wheel speed difference between the left and right wheels using the robot's coordinate distance to the goal. In the DDMR robot kinematic system with 2 wheels and 1 free-wheel front wheel, to calculate the speed of the robot, the formula for linear speed and angular speed is as Eq. (6 - 9) follows:

$$V_R = \omega_R . r_R \tag{6}$$

$$V_L = \omega_L . r_L \tag{7}$$

$$v = \frac{V_R + V_L}{2} \tag{8}$$

$$\omega = \frac{\omega_R - \omega_L}{2} \tag{9}$$

V_R and V_L show how fast the right and left wheels are moving in a straight line, and R and L show how fast the wheels are turning in a circle. Using the coordinate points (x, y) of the end point to find the wheel speeds, the inverse kinematics of each wheel's speed is used. Here, the input θ, which tells the robot which way to go, will be set separately. The values of x, y and θ are obtained as in Eq. (10 – 12).

$$\dot{x} = v . \cos(\theta) \tag{10}$$

$$\dot{y} = v.\sin(\theta) \tag{11}$$

$$\dot{\theta} = \omega \tag{12}$$

An illustration of equations on the DDMR Robot can be seen in Fig. 2.

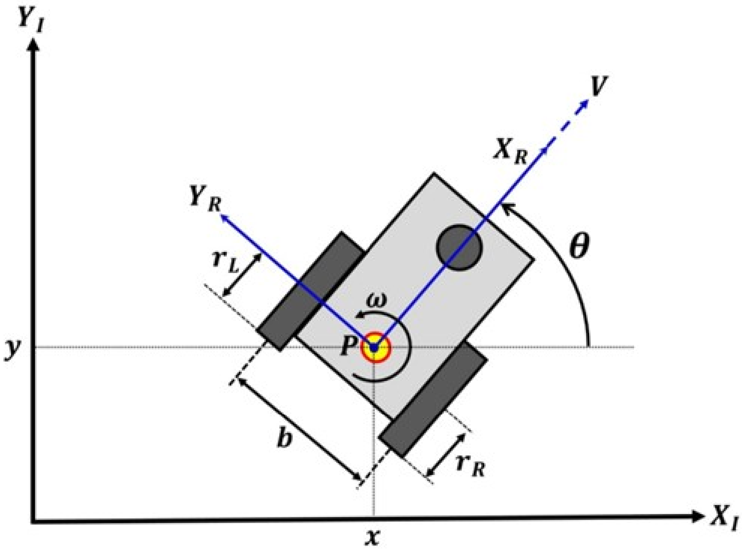


Figure 2. Illustration of Equations on the DDMR Robot [24].

2.4. Frontier Detection

Frontier detection with an occupancy grid is a method for pinpointing cells at the interface between mapped and unmapped regions inside a grid. Frontier cells are crucial for devising exploration routes, as they delineate the boundary between known (blocked or free) and unknown regions [14, 25-27].

Algorithm 1: isFrontierCell

Input: Matrix B, coordinates (x, y)
Output: True if (x, y) is a frontier cell, otherwise False

```
if (x, y) is out of bounds or B[y,x] ≠ −1 then
    return False;
Define neighbors as the 8 surrounding cells
  (including diagonals);
hasZeroNeighbor ← False;
allNeighborsUnknown ← True;
for each (nx,ny) in neighbors do
    if (nx, ny) is within bounds then
        neighborValue ← B[ny,nx];
        if neighborValue == 0 then
            hasZeroNeighbor ← True;
        if neighborValue ≠ −1 then
            allNeighborsUnknown ← False;
return (hasZeroNeighbor AND NOT
  allNeighborsUnknown);
```

Algorithm 2: Detect Frontiers

```
Input: A matrix B of size width × height
Output: List of centroids of strongly connected
        components
Create empty set frontier;
for x ← 0 to B.width − 1 do
    for y ← 0 to B.height − 1 do
        if isFrontierCell(B, x, y) then
            Add (x,y) to frontier;
SCC_list ← Kosaraju_DFS(frontier);
centroids ← [];
for each component in SCC_list do
    if component.size > 8 then
        centroid ← calculateCentroid(component);
        Add centroid to centroids;
return centroids;
```

The algorithm detects frontier cells in a matrix, which are cells with a value of -1 that border empty areas (value 0) and have at least one known neighbor (not -1), as shown in Algorithm 1. It then identifies strongly connected components (SCCs) using Kosaraju's Depth-First Search and computes the centroid of each component with more than 8 cells (Algorithm 2). These centroids represent candidate destinations, and the final goal is selected based on the chosen destination selection strategy.

2.5. Related Works

Several previous studies have explored the use of the Social Force Model (SFM) in combination with various supporting technologies to improve mobile robot navigation. Table 1 summarizes representative research efforts that utilize SFM, highlighting their main objectives and key findings. These studies demonstrate the adaptability and effectiveness of SFM across diverse environments and applications.

Table 1. Summary of related works using SFM for mobile robot navigation

Authors (Year)	Supporting Technologies	Primary Objective	Key Findings / Results
Rifqi et al. (2021)	Fuzzy Inference System (FIS), LiDAR	Adaptive navigation for healthcare robots in social environments	79.9% navigation success; safe behavior via adaptive SFM parameters.
Dewantara & Miura (2016)	Reinforcement Learning (Q-learning)	Generate socially aware guiding behaviors for a guide robot	Reduced over-reactive behavior; achieved smoother, safer, and more comfortable guidance.
Dewantara & Ariyadi (2021)	Fuzzy Inference System (FIS)	Adaptive navigation control for robot soccer in dynamic environments	Improved reactivity and responsiveness; adaptive SFM parameters outperformed constant ones.
Ramadhan et al. (2023)	Fuzzy Inference System (FIS), Genetic Algorithm (GA)	Optimize SFM parameters adaptively for mobile robot navigation	Smoother robot travel (reduced heading deviation); reached goal 1.6 sec faster than fixed parameters.

Inspired by the adaptive use of SFM in the aforementioned studies, this paper proposes a system that combines SFM with frontier-based exploration and SLAM for enhanced autonomous mapping and navigation.

3. METHODOLOGY

3.1. Proposed System

The proposed system design is illustrated in Fig. 3. The lidar sensor is used to measure object distances within the environment. The lidar data is processed to determine distances to static objects. The information from static objects is then fed into the Social Force Model (SFM) module to compute the robot's heading and speed, ensuring safe and human-aware navigation. Simultaneously, the static object data is processed by the SLAM-toolbox module, which localizes the robot and generates an occupancy grid map of the environment. The Frontier-based Detection module further utilizes the resulting map to identify unexplored regions, enabling efficient and autonomous exploration of the environment.

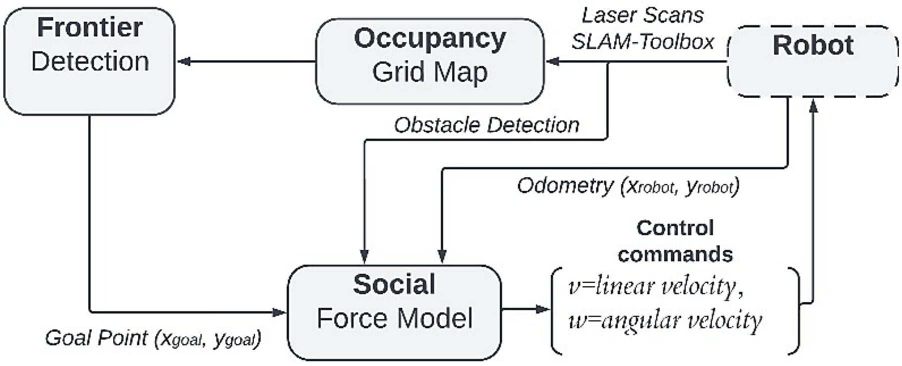


Figure 3. Proposed system

3.2. Simulation

This study is mainly about robot navigation, localization, and mapping. It uses crucial algorithms like the Social Force Model (SFM) for planning local paths, Simultaneous Localization and Mapping (SLAM) for mapping environments, and Frontier Detection for finding goal points. The primary simulation tool for the robot's behavior and environment is Gazebo. RViz is used as the graphical user interface to see how the robot interacts with its environment. In this setting, RViz processes and monitors real-time sensor data, including lidar inputs and odometry readings. This lets people see where the robot is ($R(x, y)$), which way it is facing, and how good the maps made by the SLAM method are. The study concentrates on three primary objectives:

- Navigation: The Social Force Model (SFM) tests the robot's autonomous movement. The local planner tool SFM helps the robot avoid impediments on its way to its destination. Social influences regulate the robot's speed and path to the goal to mimic human movement.
- SLAM/Mapping: This goal combines localization and mapping so the robot can find itself and map its surroundings. Independent navigation requires SLAM to provide the robot with a worldview to plan its subsequent actions.

Frontier detection finds and investigates unexplored environmental areas. Frontier Detection finds known and undiscovered map edges. Recognizing these edges enables the robot to determine where to investigate next, thereby expanding the mapped areas. This function allows the robot to explore new places and avoid previously visited ones, thereby making the map more comprehensive.

The combination of Gazebo and RViz provides a controlled and flexible platform to simulate ideal sensor data in a realistic virtual environment. This approach enables the validation of navigation and SLAM algorithms without the risks and limitations of physical hardware, supporting efficient research in customizable simulation scenarios.

3.3. Experimental Setup

All experiments were conducted on a system with the following specifications: Intel NUC, Core i5-1135G7 (4.2 GHz), 8 GB DDR4 RAM, Iris Xe Integrated Graphics, Ubuntu 22.04 LTS, and a 128 GB NVMe M.2 SSD. The development environment included Visual Studio Code (VSCode), OpenCV 4.9.0 libraries, as well as the Gazebo and RViz 3D simulation tools. Gazebo and RViz were integrated with ROS2 for real-time robotic control and visualization. For experimental purposes, the specified SFM parameters are presented in Table 2.

Table 2. SFM Parameters

Parameters	Value
Gain	3
Robot mass	1.5 kg
Max. speed	0.35 m/sec
Proxemic type	half circle
Proxemic radius	1.3 m
Effective range	1 m

3.4. Experimental Scenario

In this research, using the TurtleBot3 Waffle model, as shown in Fig. 4, which is provided by the community in the ROS 2 framework [28]. This model was chosen because it is well-integrated into the ROS 2 ecosystem and comes equipped with the 360° lidar sensor required for this research. Using the available model can avoid the time-consuming process of designing the robot from scratch, allowing for a greater focus on algorithm development and testing in simulations. In addition, TurtleBot3 is classified as a Differential Drive Mobile Robot (DDMR), which utilizes two independent wheels to drive the robot, providing precise control over its movement and making it an ideal platform for research and development in robotics.

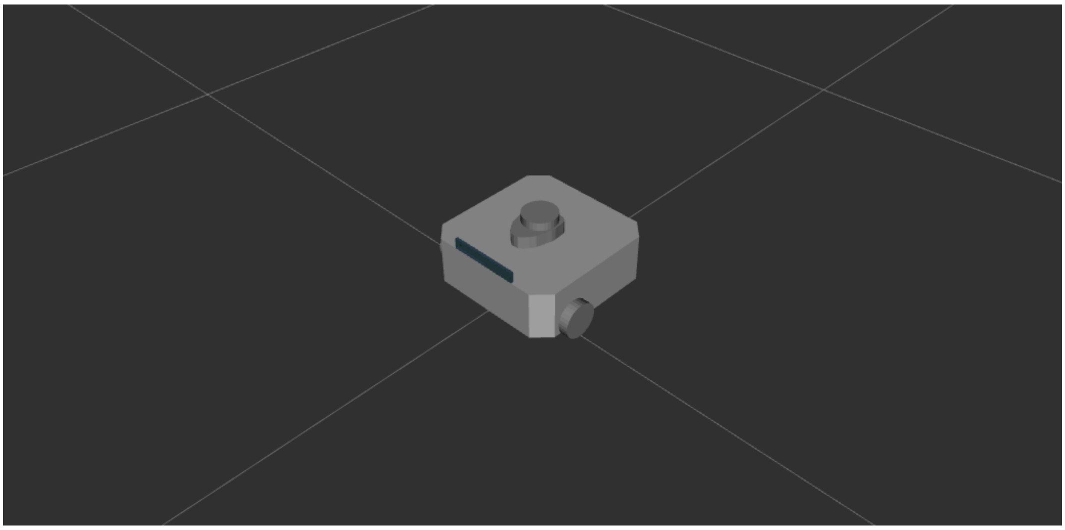


Figure 4. Turtlebot3 model used in the simulation.

https://doi.org/10.31436/iiumej.v26i3.3554

Fig. 5 shows the experimental scenario where the robot is placed in a room, as seen in the image, with an area of approximately 155 m^2. The robot's objective is to map the entire room without colliding with obstacles, then measure the distance traveled to complete the mapping and the time required to perform the mapping.

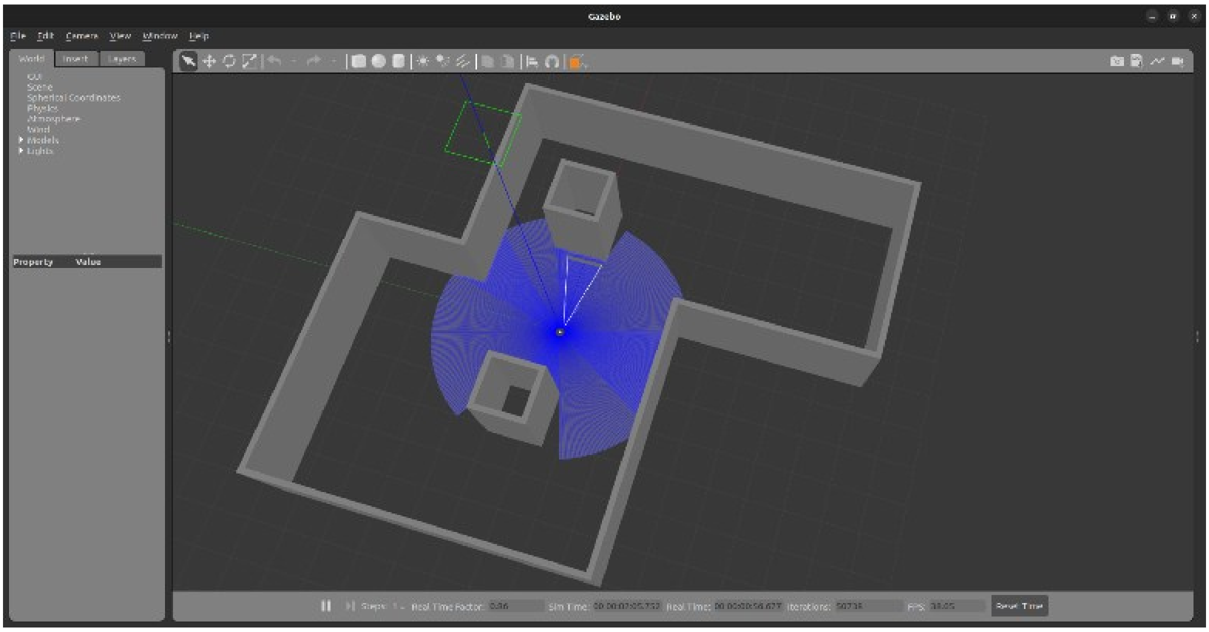


Figure 5. Experiment layout room scenario.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Frontier detection produces candidate frontier points that represent potential goal locations for the robot. In the scenario illustrated in Fig. 6, the selection strategy relies solely on identifying the nearest point from the robot's current location. While this approach is computationally simple, it often results in inefficient trajectories, as the robot consistently chooses the closest frontier point without considering direction or overall path efficiency. This can cause the robot to perform frequent orientation changes, including abrupt turns of up to 180 degrees. After reaching a goal, the map is updated and new frontier points are generated, prompting the robot to re-evaluate and select the next closest target. Due to this greedy behavior, the resulting path is sometimes unnecessarily convoluted.

The outcomes across three trials under this strategy demonstrate variations in performance. The robot traveled 77.53 m, 61.33 m, and 90.26 m, with corresponding mapping times of 13, 9, and 10 minutes, respectively. On average, the travel distance was 76.37 m, and the mapping time was 10 minutes and 40 seconds.

https://doi.org/10.31436/iiumej.v26i3.3554

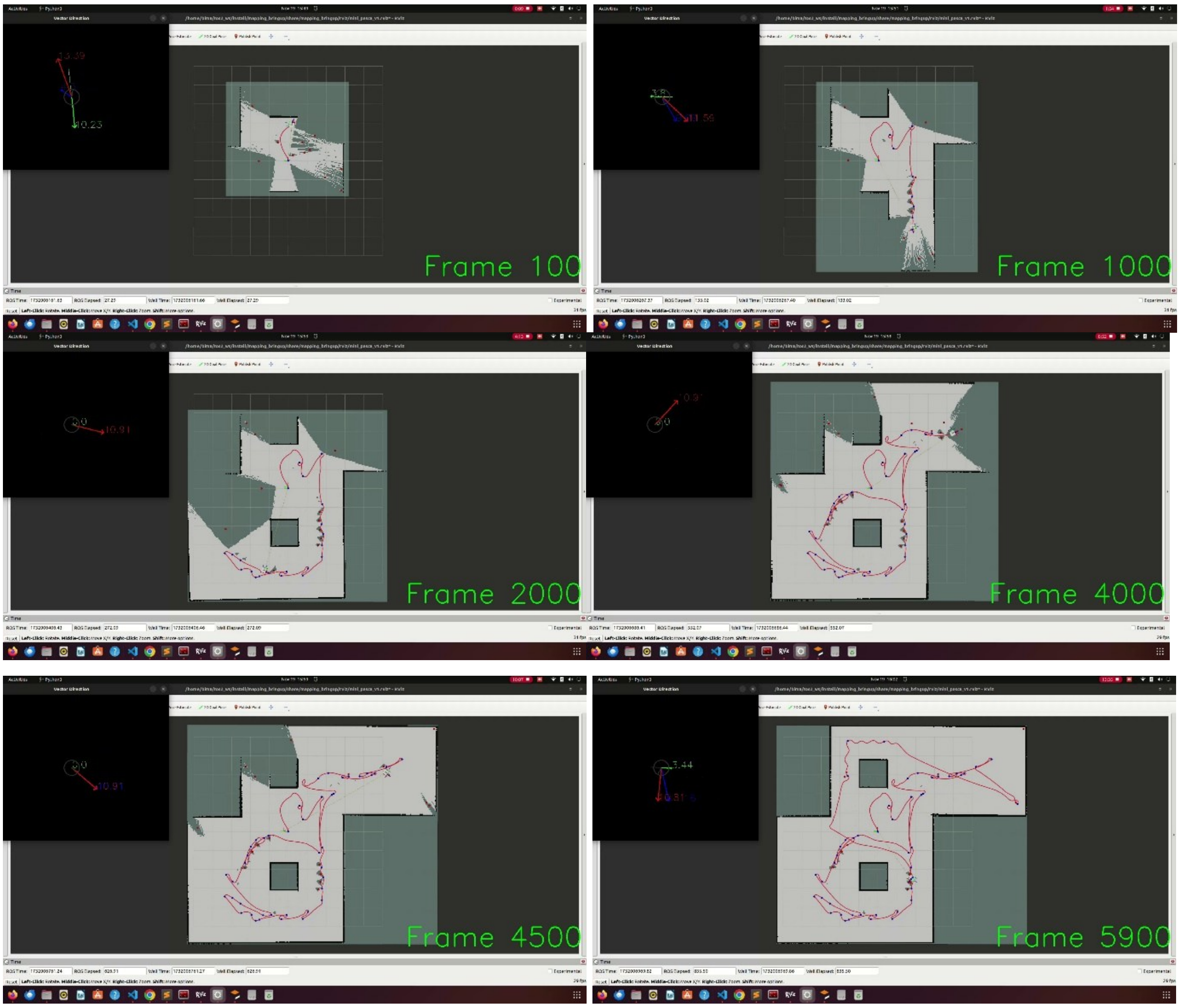


Figure 6. Mapping a room using the nearest distance of frontier points.

In contrast, Fig. 7 shows the result of using a more refined selection strategy that considers both the nearest distance and the closest orientation relative to the robot's current heading. This method aims to reduce unnecessary rotation and improve navigational smoothness. However, in practice, this strategy still suffers from inefficiencies due to timing mismatches between goal arrival and frontier detection updates. Specifically, when the robot reaches a goal point, the system may not yet finish updating the frontier points based on the new map, as the algorithm performs a complete scan of all cells. Consequently, the robot may select a suboptimal goal, one that appears optimal based on outdated map information resulting in a longer and more winding path.

In the three experiments conducted under this method, the robot traveled 99.47 m, 88.75 m, and 124.02 m, with mapping times of 17, 13, and 12 minutes, respectively. The average travel distance was 104.08 m, and the average mapping time was 14 minutes.

https://doi.org/10.31436/iiumej.v26i3.3554

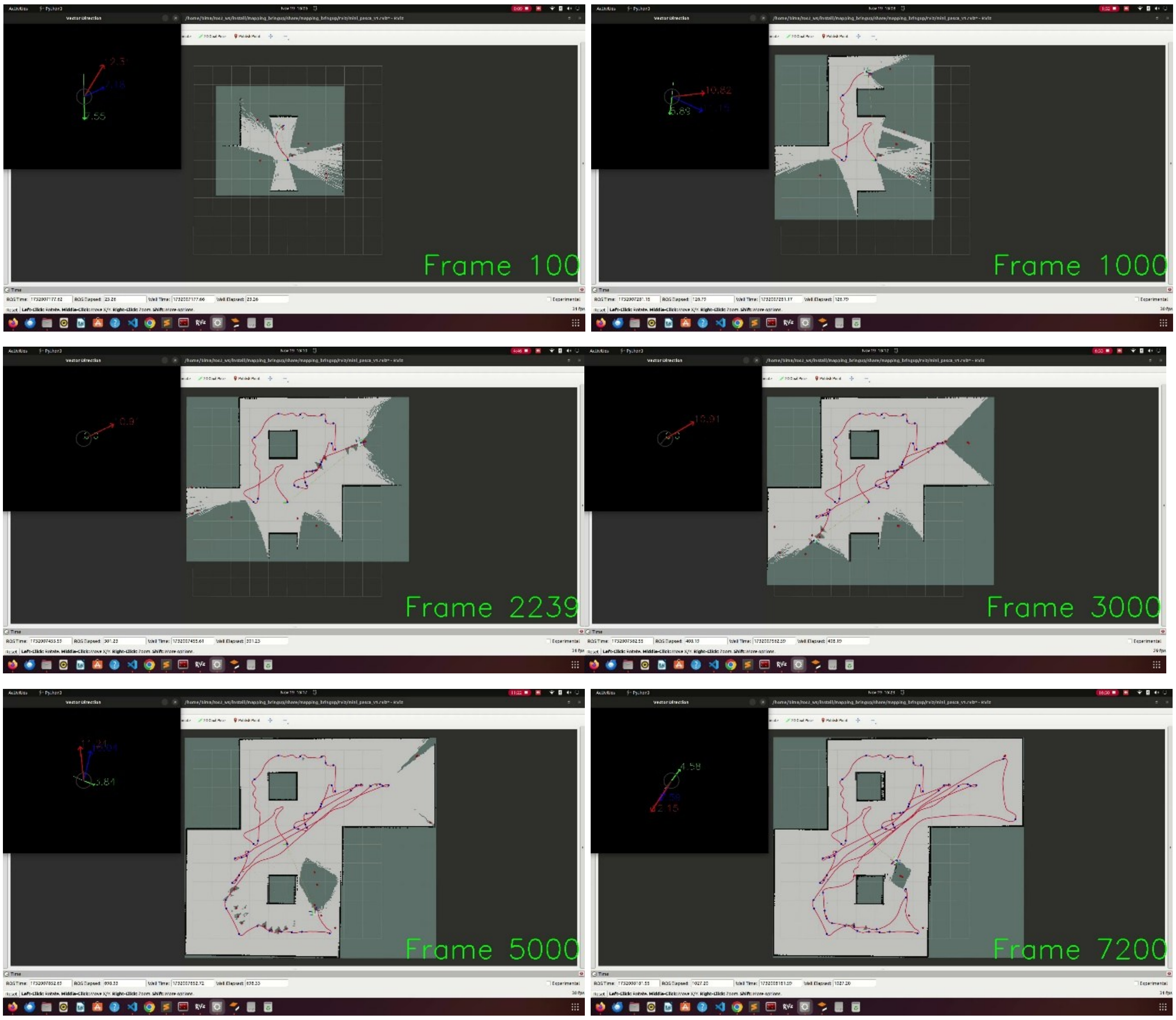


Figure 7. Mapping a room using the nearest distance and nearest orientation of frontier points.

A comparison of the two frontier point selection strategies is summarized in Table 3. The first strategy selects the frontier point based only on the shortest distance to the robot, while the second strategy additionally considers the orientation of the robot toward the point. Each approach was tested in three different experimental runs.

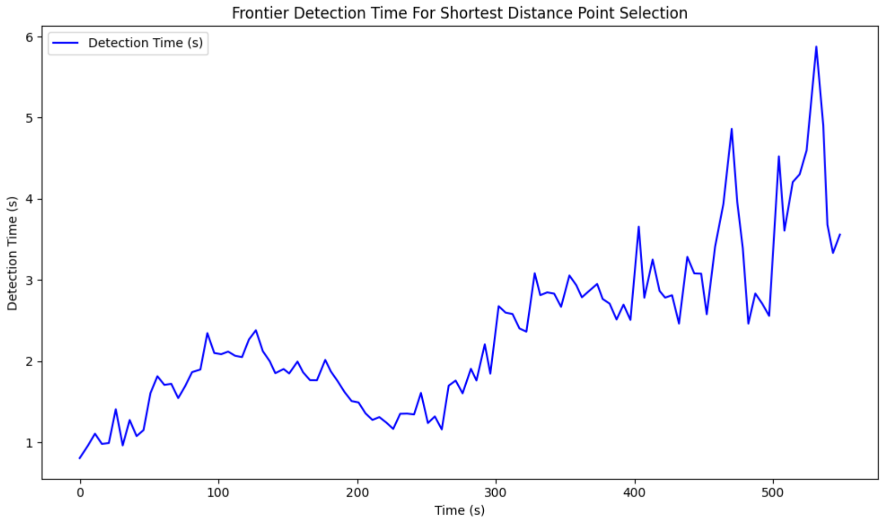
Table 3. Comparison of travel distance and mapping time for different point selection strategies.

Point Selection	**Experiment (n)**	**Travel Distance (m)**	**Mapping Time (minutes)**
Shortest distance	1	77.53	13
	2	61.33	9
	3	90.26	10
	Avg	**76.37**	**10.66**
Shortest distance & closest orientation	1	99.47	17
	2	88.75	13
	3	124.02	12
	Avg	**104.08**	**14**

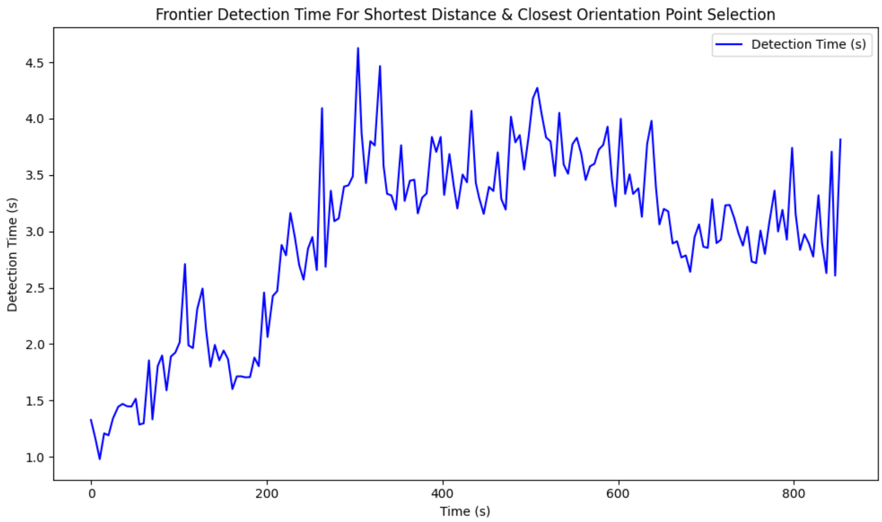
From the table above, it is evident that the strategy considering both distance and orientation tends to result in a longer travel distance and mapping time. Although this approach aims to align the robot's direction with the target, the delayed frontier update occasionally causes the robot to select less optimal points. In contrast, the shortest distance only strategy yields more efficient performance in terms of time and distance, albeit with less smooth trajectories.

The computation time required for frontier detection is also critical. Fig. 8 illustrates the detection time performance throughout the exploration. In the shortest distance scenario (Fig. 8a), the average frontier detection time is 2.34 seconds, with a maximum of 5.87 seconds and a minimum of 0.80 seconds. Meanwhile, in the shortest distance and closest orientation scenario (Fig. 8b), the average detection time increases to 2.95 seconds, with a maximum of 4.62 seconds and a minimum of 0.97 seconds.

These results indicate that as the map grows and the number of cells increases, the time needed to detect frontiers becomes longer, especially when the algorithm scans all grid cells (as described in Algorithm 1). Consequently, the robot may select a suboptimal or outdated frontier point before the detection algorithm completes processing the updated map.



(a) Shortest distance point selection scenario.



(b) Shortest distance and closest orientation point selection scenario.

Figure 8. Detection time for the frontier detection algorithm over time during exploration.

5. CONCLUSION

This study integrated the Social Force Model, SLAM Toolbox, and frontier-based detection to enable autonomous map exploration by a differential drive mobile robot. The system was evaluated in a controlled environment using the Gazebo simulator. Two frontier point selection strategies were tested: (1) based solely on the nearest distance, and (2) based on the nearest distance with orientation consideration. Experimental results show that the first strategy enabled the robot to complete the mapping task in an average of 10.66 minutes over a distance of 76.37 meters, while the second strategy required an average of 14 minutes and a distance of 104.08 meters. Although the second strategy incorporated orientation alignment, it resulted in a longer and more convoluted path. This was primarily due to two factors: limited frontier point selection logic and delay in frontier map updates, which occasionally caused the robot to choose outdated or inefficient target points. These findings indicate the need for improvements in both the frontier detection update speed and the frontier point selection strategy. Enhancing the update frequency of the frontier map would allow the robot to make more informed and timely decisions. Furthermore, optimizing the point selection mechanism by balancing distance, orientation, and predicted travel efficiency could significantly improve overall navigation performance and reduce unnecessary movements.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Directorate General of Vocational Education, Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia, for funding this research through the Master's Thesis Research Grant, as stated in Decree Number 67/SPK/D.D4/PPK.01.APTV/III/2024. The authors also extend their appreciation to the Center of Research and Community Service (P3M) Politeknik Elektronika Negeri Surabaya (PENS) for facilitating the research through Contract Number 516/PL14/PT.01.05/III/2024.

Special thanks are also extended to all members of the Social Robotics and Smart System Applications Laboratory (SRSSA Lab) for their valuable support and collaboration during the study. Finally, the authors gratefully acknowledge the constructive feedback and insightful suggestions provided by the anonymous reviewers, which significantly improved the quality of this manuscript.

REFERENCES

[1] H.I. Krebs and N. Hogan, "Therapeutic Robotics: A Technology Push: Stroke rehabilitation is being aided by robots that guide movement of shoulders and elbows, wrists, hands, arms and ankles to significantly improve recovery of patients," *Proc IEEE Inst Electr Electron Eng.*, vol. 94, no. 9, pp. 1727 – 1738, 2006.

[2] Passler J. Tilley, "Automation, robotics, and the factory of the future," The Great Re-Make: Manufacturing for Modern Times.

[3] D.F. Yépez-Ponce, J.V. Salcedo, P.D. Rosero-Montalvo, and J. Sanchis, "Mobile robotics in smart farming: current trends and applications," Front Artif Intell., vol. 6, pp. 1213330, 2023.

[4] I. Kubasáková, J. Kubánová, D. Benco, and N. Fábryová, ``Application of Autonomous Mobile Robot as a Substitute for Human Factor in Order to Increase Efficiency and Safety in a Company," Applied Sciences, vol. 14, no. 13, pp. 5859, 2024.

[5] A. Tuomi, I.P. Tussyadiah, and J. Stienmetz, "Applications and Implications of Service Robots in Hospitality," Cornell Hospitality Quarterly, vol. 62, no. 2, pp. 232 – 247, 2021.

[6] L. Wijayathunga, A. Rassau, and D. Chai, "Challenges and Solutions for Autonomous Ground Robot Scene Understanding and Navigation in Unstructured Outdoor Environments: A

Review," Applied Sciences, vol. 13, no. 17, pp. 9877, 2023.The Contiki operating system. Available: http://www.contiki-os.org/index.html.

[7] H.M. Mu'allimi, B.S.B. Dewantara, D. Pramadihanto, and B.S. Marta, "Human Partner and Robot Guide Coordination System Under Social Force Model Framework Using Kinect Sensor," The 22nd International Electronics Symposium (IES), 2020.

[8] A.T. Rifqi, B.S.B. Dewantara, D. Pramadihanto, and B.S. Marta, "Fuzzy Social Force Model for Healthcare Robot Navigation and Obstacle Avoidance," 2021 International Electronics Symposium (IES), 2021.

[9] B.S.B. Dewantara and J. Miura, "Generation of a Socially Aware Behavior of a Guide Robot Using Reinforcement Learning," IEEE International Electronics Symposium (IES), 2016.

[10] B.S.B. Dewantara and B.N.D. Ariyadi, "Adaptive Behavior Control for Robot Soccer Navigation Using Fuzzy-based Social Force Model," Smart Science, vol. 9, no. 1, pp. 14 – 29, 2021.

[11] A.C.C. Meng, M. Wand, and V.S.S. Hwang, "A Methodology of Map-Guided Autonomous Navigation with Range Sensor in Dynamic Environment," Proc. SPIE 1007, Mobile Robots III, 1989.

[12] C. Chen and Y. Cheng, "Research on Map Building by Mobile Robots," 2008 Second International Symposium on Intelligent Information Technology Application, Shanghai, China, 2008, pp. 673 – 677, 2008.

[13] J. Achterhold, S. Guttikonda, J.U. Kreber, H. Li, and J. Stueckler, "Learning a Terrain- and Robot-Aware Dynamics Model for Autonomous Mobile Robot Navigation," Cornell University, arXiv:2409.11452, 2024.

[14] B. Yamauchi, "A frontier-based approach for autonomous exploration," Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97, pp. 146 – 151, 1997.

[15] M. Sasaki, Y. Tsuda, and K. Matsushita, "Development of Autonomous Mobile Robot with 3DLidar Self Localization Function using Layout Map," Electronics, vol. 13, pp. 1082, 2024.

[16] A. Ramadhan, B. Dewantara, and S. Setiawardhana, "Optimization of Fuzzy Social Force Model Adaptive Parameter using Genetic Algorithm for Mobile Robot Navigation Control," Jurnal Rekayasa Elektrika, vol. 19, no. 1, 2023, doi: 10.17529/jre.v19i1.28330.

[17] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," Phys Rev., vol. E51, pp. 4282 – 4286, 1995.

[18] H. Taheri and Z.C. Xia, "SLAM; definition and evolution," Engineering Applications of Artificial Intelligence, vol. 97, 2021.

[19] A.A. Housein and G. Xingyu, "Simultaneous Localization and Mapping using differential drive mobile robot under ROS," Journal of Physics: Conference Series, vol. 1820, 2021 International Conference on Mechanical Engineering, Intelligent Manufacturing and Automation Technology (MEMAT), 2021.

[20] S. Macenski and I. Jambrecic, "SLAM Toolbox: SLAM for the dynamic world," JOSS, vol. 6, no. 61, p. 2783, May 2021.

[21] G. Ferrer et al., "Robot companion: a social-force based approach with human awareness-navigation in crowded environments," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1688 – 1694, 2013.

[22] F. Zanlungo et al., "Social force model with explicit collision prediction," EPL (Europhysics Letters), vol. 93, pp. 68005-p1 – 68005-p6, 2011.

[23] M. Luber et al., "People tracking with human motion predictions from social forces," IEEE International Conference on Robotics and Automation, pp. 464 – 469, 2010.

[24] N. Chindakham et al., "Simultaneous Calibration of Odometry and Head-Eye Parameters for Mobile Robots with a Pan-Tilt Camera," Sensors, vol. 19, no. 16, p. 3623, Aug. 2019.

https://doi.org/10.31436/iiumej.v26i3.3554

[25] M. Keidar and G. A. Kaminka, “Efficient frontier detection for robot exploration,” The International Journal of Robotics Research, vol. 33, no. 2, pp. 215 – 236, Feb. 2014.

[26] P. Quin, A. Alempijevic et al., “Expanding wavefront frontier detection: An approach for efficiently detecting frontier cells,” Australasian Conference on Robotics and Automation (ACRA), 2014.

[27] Y. Sun and C. Zhang, “Efficient and Safe Robotic Autonomous Environment Exploration Using Integrated Frontier Detection and Multiple Path Evaluation,” Remote Sensing, vol. 13, no. 23, p. 4881, Dec. 2021.

[28] ROBOTIS-GIT, “turtlebot3_simulations,” GitHub repository}, available at: https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git, accessed on: November 2024.