Semantic-Enhanced Path Planning for Safety-Centric Indoor Robots Navigation

Karameldeen Omer*
Department of Information Engineering
Università Politecnica delle Marche
Ancona, Italy
k.i.m.omer@pm.univpm.it
*Corresponding author

Elena Torta
Department of Mechanical Engineering
Eindhoven University of Technology
Eindhoven, Netherlands
e.torta@tue.nl

Andrea Monteriù
Department of Information Engineering
Università Politecnica delle Marche
Ancona, Italy
a.monteriu@univpm.it

***Abstract*—This paper presents an innovative approach for indoor mobile robot navigation, focusing on environments like Ambient Assisted Living spaces (AAL), smart homes, and factories, where humans and robots coexist. The need for effective space and traffic management in these areas is critical for ensuring safety and mobility. Our novel Semantic Path Planning algorithm improves upon traditional grid mapping by using a multi-dimensional array in the map file, where varying intensities indicate occupancy levels. The planner imposes penalties on free cells near specific static objects based on user preferences, facilitating autonomous robot movement in designated areas without creating rigid barriers. A standout feature of our algorithm is its ability to prioritize specific zones frequented by vulnerable groups like the elderly, or areas designated for rest, pets, or children's play. By enhancing the A* path planner with semantic and geometric data, our approach enables the management of these zones, leading to a comprehensive grid map for optimal path-finding. Moreover, this methodology is promising for multi-robot systems with differing navigation abilities and access rights. It also improves robot localization by focusing on unique environmental landmarks, thereby enhancing tracking accuracy and aiding in swift localization recovery. This approach is ideal for safe, efficient, and context-aware navigation in dynamic shared human-robot environments.**

***Index Terms*—Semantic Mobile Robot Navigation, Semantic Path Planning, Safety-Centric Navigation, Human-Centered Design, Autonomous Robotics, Context-Aware Navigation.**

I. Introduction

In the rapidly evolving field of indoor mobile robot navigation, the coexistence of humans and robots has become a focal point for research and development. This coexistence demands a sophisticated approach to safety, necessitating the integration of advanced technologies and innovative strategies [1]. Central to these advancements are smart sensors, strict zones, and virtual barriers or semantic navigation, which are considered most methods used for the harmonious integration of robots into human-centric environments.

A. Smart Sensors

One of the big steps toward improving indoor navigation by smart sensors. Smart sensors are pivotal in enabling robots to safely navigate complex indoor environments [1]. Equipped with technologies such as LiDAR [2], infrared, ultrasonic, and vision-based sensors, these robots can perceive their surroundings with remarkable precision. This sensory input allows for real-time detection of obstacles, including moving humans, thereby preventing collisions and ensuring seamless navigation.

B. Human-in-the-loop

This approach involves a mobile robot navigating indoors autonomously, with a human operator using a Brain-Computer Interface (BCI) to detect obstacles missed by the robot's sensors [3]. When the operator senses an obstacle, the BCI identifies Error-related Potentials (ErrPs) in the operator's brain signals, triggering a message to create a virtual obstacle for the robot. This virtual obstacle is designed to be easily recognized by the robot's sensors, and the robot's navigation system updates its local cost map accordingly. The mobile robot adjusts its path planning in real-time to avoid the newly detected virtual obstacles, enhancing safety and adaptability in indoor environments [4].

C. Strict Zones and Virtual Barriers for Space Management

Strict zones are designated areas where robot access is heavily restricted or entirely prohibited. These zones are a common topic in literature, with various works exploring space control methods [5], [6]. One particularly relevant contribution is from Omer and Monteriù [7], who not only provide an extensive overview of current methodologies, but also propose a novel approach involving virtual barriers. Their work stands out for its comprehensive coverage and the introduction of this innovative method for managing robot navigation in restricted areas. These zones are crucial in scenarios where safety is paramount, such as in areas frequented by vulnerable groups like children or the elderly. By defining these zones, robots can be programmed to navigate around them, minimizing any potential risk to human occupants. The concept of virtual barriers represents a strategic innovation in managing shared spaces. These invisible boundaries, programmable and adjustable, can be used to direct robot movement, ensuring they avoid specific areas that might be hazardous or require privacy. Virtual barriers offer a flexible solution compared to physical barriers, allowing for dynamic reconfiguration of robot paths as the environment or requirements change.

D. Vision-based Topological Path Planning

Topological path planning, on the other hand, is a method for planning the motion of a mobile robot in an indoor environment based on a topological representation of the space [8]. Instead of relying on a detailed geometric map of the environment, topological path planning uses a high-level abstraction of the environment's connectivity. This abstraction simplifies the planning process and can be more efficient and robust in certain situations, particularly in indoor environments where complex geometries may not be necessary to navigate successfully [9].

E. Semantic Robot Navigation

Semantic robot navigation refers to the method where robots use an understanding of the environment's context and meaning, beyond just spatial dimensions, to navigate more intelligently [10]. This approach enables robots to interpret and interact with their surroundings in a human-like manner, recognizing and responding to the function and significance of objects and areas.

In this paper, we will focus only on Semantic Robot Navigation. More specifically, as main contribution of this work, we propose the Semantic Path Planning that integrates with most robot navigation systems without the need for new sensors or advanced techies. The final objective is to increase the safety and navigation of mobile robots in a shared environment with elder or disabled.

The paper is structured as follows. Section II introduces the proposed approach, namely Semantic Information-based Path Planning. The development and implementation of the proposed approach are detailed in Section III. Simulation results are presented and discussed in Section IV. Finally, Section V concludes the paper, offering insights into future integration and improvements.

II. Proposed Concept: Semantic Information-Based Path Planning

Semantic information-based path planning represents a cutting-edge approach in the field of robotics, particularly in the context of indoor navigation [10], [11]. This advanced methodology extends beyond traditional navigation systems that rely solely on geometric data, by incorporating an understanding of the context and meaning of different elements within the environment.

The core of semantic navigation is the robot's ability to interpret and utilize data about its surroundings in a way that mimics human understanding. For instance, a robot equipped with this technology can differentiate between a stationary obstacle like a wall and a temporary one like a moving person or an open door. This distinction is crucial in dynamic environments where the context and placement of objects can change rapidly.

The integration of semantic information allows robots to navigate more intelligently and adaptively. For instance, in a hospital setting, a robot can recognize areas frequently traversed by humans and choose alternate routes to minimize disruption. In a home environment, it could identify a space as a play area for children and avoid it to prevent potential hazards.

Based on this approach, we propose a new path planning that leverages semantic information about the navigation environment. This approach transcends traditional navigation methods by not just considering the physical layout of the space, but also understanding the contextual and semantic significance of different areas around statically known obstacles.

Key Features of the proposed Path Planner are:

- Semantic Understanding: The path planner interprets environmental data beyond mere geometry. It recognizes and reacts to the semantic properties of objects and spaces, such as identifying a chair as a potential temporary obstacle or a doorway as a transitional point.
- Safety-Centric Navigation: Emphasizing safety, the planner can prioritize routes that minimize human interaction, particularly in areas designated for specific activities like play zones or rest areas.
- Customized Parameters: Users can define specific criteria or constraints based on the unique requirements of their environment, allowing for tailored navigation strategies that align with specific safety and operational goals. computations.

This proposed path planning approach marks a significant advancement in addressing the challenges of human-robot cohabitation. By integrating semantic understanding, the planner ensures more intuitive and human-aware navigation, aligning closely with the nuanced and ever-changing dynamics of indoor spaces. This not only enhances safety but also improves the efficiency and adaptability of robot navigation, paving the way for more sophisticated and harmonious human-robot interactions in shared environments.

A. Path Planner Algorithm

For our research, we employed a customized adaptation of the A* algorithm, as described in the work by Wang *et al.* [12]. The A* algorithm is a widely recognized and extensively used path-finding and graph traversal technique, often in conjunction with Dijkstra's algorithm [13], for global path planning on platforms such as ROS (Robot Operating System) and various other robotic platforms [14], [15]. This algorithm excels at efficiently identifying the shortest path between a designated starting node and a target node within a graph with weighted edges.

The A* algorithm is particularly noteworthy for its ability to merge the advantageous features of Uniform Cost Search and Greedy Best-First Search. It achieves this by striking a delicate balance between the consideration of path cost and heuristic estimates, ensuring an effective and balanced approach to pathfinding.

Main Steps of the A* Algorithm

1) **Initialization**: Start with an open list containing only the start node and a closed list that is empty.
2) **Loop**: Repeat the following steps until the open list is empty:

a) **Node Selection**: Choose the node with the lowest

$$f(n) = g(n) + h(n) \tag{1}$$

from the open list, where $g(n)$ is the cost from the start node to n, and $h(n)$ is the heuristic estimated cost from n to the goal.

b) **Goal Check**: If the chosen node is the goal, backtrack through the parents to recover the path.

c) **Successors**: Generate the node's successors. For each successor:

   i) **Successor Cost**: Calculate $f(n)$ for each successor.

   ii) **Open or Closed**: Check if the successor is in the open or closed list with a lower $f(n)$. If it is, skip it.

   iii) **Add to Open List**: Add the successor to the open list and set its parent to the current node.

d) **Move Node**: Move the current node from the open list to the closed list.

3) **Failure**: If the open list is empty and the goal was not reached, return failure.

The heuristic function $h(n)$ is problem-specific. It estimates the cost to reach the goal from node n. A good heuristic is essential for the efficiency of A*. Common choices are:

- **Manhattan Distance**: Used for grid-based maps.
- **Euclidean Distance**: Used for direct distance measurements.

B. *Semantic A* Path-Planner Algorithm*

The modification will be by introducing a new penalty value for each cell in the given map. This penalty is calculated for each cell based on the cell distance from the given list of static obstacles defined by their semantic information. After calculating all distances the penalty will be calculated based on the user-defined parameters then the A* adds these values to the heuristic cost and the process continue for given start point to the given goal. Flowchart shown in Figure 1 gives the summarized description.

1) Map processing: This process is considered the cornerstone of the proposed path planner which is a critical process that initiates with the loading of the map file along with its associated metadata. The metadata, contained in a YAML file, provides essential details such as the map's size and resolution, the types of static obstacles represented as occupied cells, and the threshold differentiating these obstacles. Additionally, it includes specific penalty values assigned to each static object on the map and parameters for the penalty function.

Once this data is fully loaded, the planner executes a function that identifies and separates all occupied cells, creating masks. These masks are then used to index vectors in a subsequent function. This function is responsible for calculating the distance from every free cell to the listed obstacles, storing these values for future reference. These stored distances play a crucial role in the penalty calculation process, contributing to the planner's ability to make informed navigation decisions.

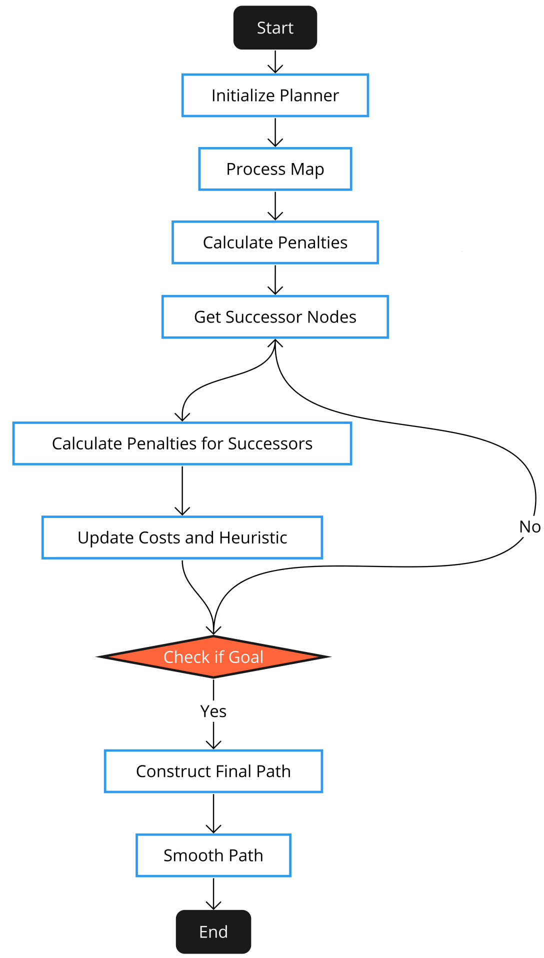

Fig. 1. Flowchart for the implemented and modified A* algorithm.

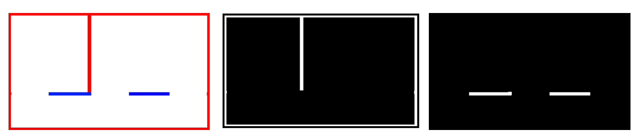

Fig. 2. From left to right, the navigation map red walls, and the blue glass partition, which are then separated by the planner as masks: the first mask (in the middle) represents the wall cells, while the other one (to the right) represents the glass cells.

2) Penalties Calculation: The penalty for empty cell is a function of the distance (d) from each close wall or obstacle. This could be a continuous or piecewise function, which will be customized for all static obstacles by predefined parameters. These parameters will be kept in a metadata file that is updated by user as required. The piecewise penalty function results as follows:

$$\text{Penalty}(d) = \begin{cases} \beta_1 & \text{if } d < \alpha_1 \\ \beta_2 & \text{if } \alpha_1 \leq d \leq \alpha_2 \\ \gamma & \text{if } d > \alpha_2 \end{cases} \tag{2}$$

with $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma \geq 0$.

The penalty function could be also calculated as a continuous penalty function as below:

$$\text{Penalty}(d) = A \times e^{-k_1(d-\mu_1)^2} + C \times \log((d - \mu_2 + k_2)) \quad (3)$$

where:

- A, C: scaling constants for the two components of the function.
- k_1: determines the rate of decrease of the penalty near the wall.
- μ_1: denotes distance offset for the first component; typically set to 0.
- μ_2: represents the distance at which the logarithmic penalty begins to have a notable effect.
- k_2: offset to ensure the value inside the logarithm is always positive.

Figure 3 represents how a penalty value changes with the distance from an element or obstacle. The x-axis, labeled "Distance", varies based on the element or the obstacle in consideration. The y-axis represents the "Penalty Value" applied in the context of a path-finding algorithm like A*.

The function shown in the graph has several key features:

- Slopes α_1 and α_2 represent the rate at which the penalty decreases as the distance from the obstacle increases.
- The threshold β_1 indicates a distance beyond which the penalty sharply drops.
- The threshold β_2 represents the distance beyond which no penalty is applied.
- The line γ represents the minimum penalty value that is applied once the distance is beyond β_2.

This penalty function is particularly useful in scenarios where a robot must navigate around obstacles, discouraging paths that pass too close to potentially hazardous elements. By adjusting the parameters α_1, α_2, β_1, β_2, and γ, the path planner can balance between a safe and a conservative path, optimizing the overall route for efficiency and safety.

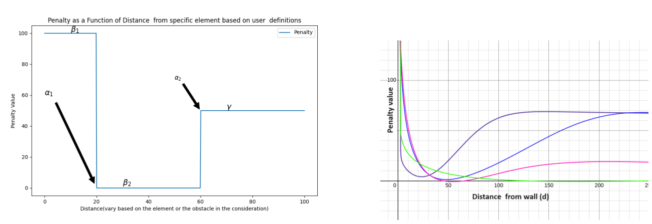

Fig. 3. Penalty as a distance function: as a piecewise function on the left, as a continuous function on the right.

III. Implementation

A. Simulation Platform

The proposed method was developed and implemented within a simulated environment by using the Robot Operating System (ROS) as the primary framework. ROS is a comprehensive collection of software tools and libraries, specifically designed to support the development of autonomous robotics [16]. The versatility and robustness of ROS, documented in various sources [14], [17], [18], [19], make it indispensable in robotics, enabling the integration of diverse functions necessary for advanced robotic applications. Within this framework, our path planner was adeptly integrated as a plugin into the ROS navigation stack, aligning with the best practices as outlined in [20]. This integration highlights the adaptability and effectiveness of our approach within the established ROS ecosystem.

B. Robot Specification and Workspace

For the simulation, as a robot platform, we used the ROSbot 2R model in the Gazebo environment has a size of 198 x 228 x 218 mm [L x W x H] equipped with a camera, LiDAR (Light Detection And Ranging), and IMU (Inertial Measurement Unit). This robot running in a simulated environment constructed as a simple building layout consisting of two rooms connected by a hallway, with a glass partition separating one of the rooms from the hallway. The main goal of the simulation is to evaluate the robot's navigation algorithm in a specific environment, focusing on identifying the most efficient path between two rooms. In situations where the optimal route involves passing through a glass partition, the algorithm is expected to ensure the robot maintains a safe distance from the glass while favoring routes closer to the walls for more safety. This simulation scenario is adaptable and can be further extended to include additional complexities, such as avoiding navigating over thick carpets in the middle of a room or adhering to no-stop zones in corridors, enhancing the algorithm's applicability and robustness in diverse indoor settings. The navigation stack is supplied with a map in PNG format, accompanied by a YAML file that includes all the semantic interpretations derived from the map. As the map is in standard grayscale, ranging from 0 to 255 [21], both the map and the planner are afforded a broad spectrum for incorporating new static elements. Each element can be coded with a unique gray value, enabling precise and distinct representation.

Follows an example of the YAML file:

image: map_file.png
resolution: 0.0105
origin: [-3.8, -2.4, 0.0]
occupied_thresh: 89
free_thresh: 205
wall_thresh: 0
glass_thresh: 65
wall_penalty:
    close_threshold: α_1
    mid_threshold: α_2
    close_penalty: β_1
    mid_penalty: β_2
    far_penalty: γ
glass_penalty:
    close_threshold: α'_1
    mid_threshold: α'_2
    close_penalty: β'_1

`mid_penalty:` β_2'
`far_penalty:` γ'
`negate: 0`

Furthermore, the final penalty for each cell is computed based on user-defined criteria, allowing customization of the robot's behavior. Users can specify how to combine the penalties of each cell (see Figure 4). This flexible approach ensures that the robot's navigation is not only efficient but also aligns with specific safety requirements and desired behaviors in different environments. This method underlines the practical implementation of our path planning algorithm, demonstrating its adaptability and effectiveness in controlled scenarios.

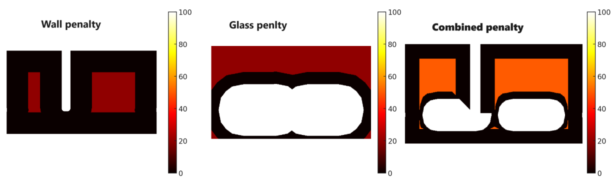

Fig. 4. Computed penalty for each cell in the map, from all present obstacles.

IV. Results and discussion

The simulation results for our path planning methodology, which integrates semantic map data within a geometric map, have successfully validated the effectiveness of our concept in a controlled environment. This test environment was designed as a simplified space, mimicking small buildings with features such as walls and glass partitions. The primary objective of the simulation was to evaluate the robot's ability to maintain a safe distance from walls while simultaneously avoiding glass partitions, by predefined penalty parameters.

The outcomes of the simulation were promising. In most scenarios, the robot demonstrated efficient navigation, closely adhering to the walls and effectively avoiding areas with glass partitions. This positive performance in a basic setup is a strong indication of the method's capability to fulfill specific, user-defined navigation requirements. It also highlights the system's adaptability in tailoring path planning to suit the unique navigation abilities of different robots. This successful application in a simplified scenario offers a glimpse into the potential of this approach in more complex and dynamically changing environments. The initial findings from our path planning system, though still in the stages of refinement, have demonstrated significant potential and represent a considerable step forward in the realm of sophisticated, context-aware robotic navigation. A critical insight gained from our simulation tests is the paramount importance of precise penalty value settings. The effectiveness of the navigation strategy is heavily relay on these values; inaccuracies or imprecise configurations can result in less-than-ideal navigational choices, as clearly illustrated in Figure 5. This figure illustrates the planned path using A* (on the left), and the generated path by using our semantic path planner (on the right).

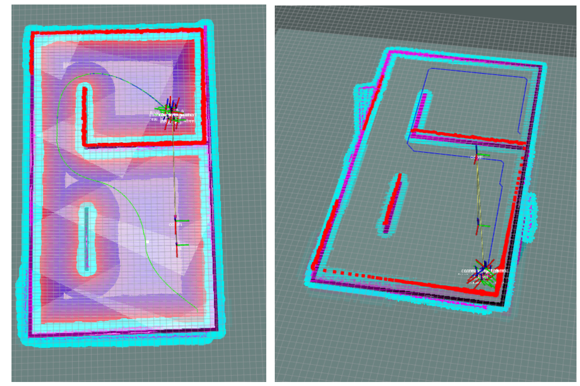

Fig. 5. Different goals and starting point path planning: on the left with normal A*, on the right using semantic A*.

Figure 6 shows how varying the penalty values can drastically alter the robot's chosen path toward the same navigation target. In instances where these values are not accurately calibrated, the robot might opt for routes that are unnecessarily longer or less efficient. This observation underscores the necessity of precisely adjusting these penalty parameters. Fine-tuning them is crucial for ensuring that the robot's navigation is not only effective but also optimizes the route in terms of distance and efficiency. This aspect of our path planning system highlights the delicate balance between algorithmic precision and practical navigation outcomes.

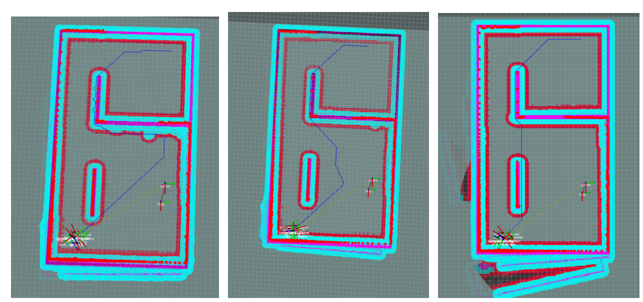

Fig. 6. Effect of using different heuristic scale factors and penalty costs on the path planning

V. Conclusion

The integration of semantic map data into our path planning system marks a significant advancement in robotic navigation, enhancing both safety and efficiency. This approach, which focuses on generating safe path plans, plays a vital role in ensuring secure robot navigation. Importantly, it also improves localization accuracy by emphasizing landmarks and adopting a human-centered design philosophy. This strategy is especially advantageous for robots with diverse capabilities, making their navigation not only safer but also more intuitive and attuned to human environments.

From a human-centered design perspective, our system gives precedence to safe interactions between robots and humans in shared environments. By semantically understanding and interpreting surroundings, robots are equipped to navigate in a way that is more in tune with human presence and activities [10]. This capability is particularly crucial in settings such as healthcare facilities, warehouses, and public areas, where smooth and safe interactions between humans and robots are paramount. The semantic approach ensures that robots can coexist with humans, adapting to and respecting their needs and behaviors in these shared spaces.

One notable potential of this path planner is its seamless integration with Building Information Modeling (BIM) [22] as digital twins. This integration enables the generation of maps from live building data [23], enriching the robot's map with comprehensive semantic information about building elements, furniture, and equipment. The system effectively mirrors the physical environment in a virtual model, augmented with semantic details. This real-time updating capability minimizes the likelihood of navigational errors and substantially boosts safety in dynamic environments, as detailed in [24]. The ability to accurately mirror and adapt to the physical world in this manner is a substantial advancement in robotic path planning, underlining the importance of integrating advanced digital twin technology in autonomous navigation systems.

Additionally, the customization of navigation paths tailored to the unique skills of different robots is a key feature of this system. This customization ensures that each robot operates within its optimal capacity, adhering to stringent safety protocols. This adaptability is especially crucial in complex environments where a variety of robots with different functionalities coexist, necessitating a flexible and responsive navigation system.

In conclusion, semantic path planning represents not just a technological advancement in robotic navigation but a significant step toward the development of more responsive and safer robotic systems. One of the most notable aspects of this advancement is its ability to enhance navigation without relying on expensive additional sensors. This makes the technology broadly accessible and easily integrated with most existing robotic platforms, without the need for high-tech upgrades. By aligning with the progressive concepts of digital twins and human-centered design, semantic path planning is setting a new standard in the field of robotics, pushing toward more efficient, safer, and user-friendly robotic systems.

References

[1] W. R. Faiza Gul and S. S. N. Alhady, "A comprehensive study for robot navigation techniques," *Cogent Engineering*, vol. 6, no. 1, p. 1632046, 2019.

[2] Y. K. Tee and Y. C. Han, "Lidar-based 2d slam for mobile robot in an indoor environment: A review," in *2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, 2021, pp. 1–7.

[3] K. Omer, F. Ferracuti, A. Freddi, S. Iarlori, A. Monteriù, and C. Porcaro, "Human-in-the-loop approach for enhanced mobile robot navigation," in *2022 IEEE International Conference on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)*. IEEE, 2022, pp. 416–421.

[4] F. Ferracuti, A. Freddi, S. Iarlori, A. Monteriù, K. I. M. Omer, and C. Porcaro, "A human-in-the-loop approach for enhancing mobile robot navigation in presence of obstacles not detected by the sensory set," *Frontiers in Robotics and AI*, vol. 9, p. 909971, 2022.

[5] D. Sprute, K. Tönnies, and M. König, "A study on different user interfaces for teaching virtual borders to mobile robots," *International Journal of Social Robotics*, vol. 11, no. 3, pp. 373–388, 2019.

[6] D. Sprute, P. Viertel, K. Tönnies, and M. König, "Learning virtual borders through semantic scene understanding and augmented reality," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4607–4614.

[7] K. Omer and A. Monteriù, "An effective method for creating virtual doors and borders to prevent autonomous mobile robots from entering restricted areas," in *2023 9th International Conference on Automation, Robotics and Applications (ICARA)*, 2023, pp. 192–196.

[8] A. Mai, "Novel solution of topological recognition of indoor objects based on optical flow and planar attributes," *Advances in Military Technology*, vol. 17, no. 2, pp. 383–396, 2022.

[9] R. Barber, J. Crespo, C. Gómez, A. C. Hernámdez, and M. Galli, "Mobile robot navigation in indoor environments: Geometric, topological, and semantic navigation," in *Applications of Mobile Robots*. IntechOpen, 2018.

[10] L. Naik, S. Blumenthal, N. Huebel, H. Bruyninckx, and E. Prassler, "Semantic mapping extension for openstreetmap applied to indoor robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3839–3845.

[11] M. Peavy, P. Kim, H. Oyediran, and K. Kim, "Integration of real-time semantic building map updating with adaptive monte carlo localization (amcl) for robust indoor mobile robot localization," *Applied Sciences*, vol. 13, no. 2, p. 909, 2023.

[12] H. Wang, X. Qi, S. Lou, J. Jing, H. He, and W. Liu, "An efficient and robust improved a* algorithm for path planning," *Symmetry*, vol. 13, no. 11, p. 2213, 2021.

[13] J. Smith and J. Jones, "Dijkstra's algorithm for robotic path planning," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 873–882, 2016.

[14] B. Cybulski, A. Wegierska, and G. Granosik, "Accuracy comparison of navigation local planners on ros-based mobile robot," in *2019 12th International Workshop on Robot Motion and Control (RoMoCo)*, 2019, pp. 104–111.

[15] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[16] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3.2. Kobe, Japan, 2009, pp. 1–6.

[17] O. Robotics, "ROS planners," http://wiki.ros.org/navigation/Planning, 2022, accessed: 2022-12-20.

[18] K. Zheng, "Ros navigation tuning guide," *Robot Operating System (ROS) The Complete Reference (Volume 6)*, pp. 197–226, 2021.

[19] O. Robotics, "ROS map servers," http://wiki.ros.org/map_server, 2022, accessed: 2022-12-20.

[20] R. Wiki, "Writing a global path planner as plugin in ros," 2015.

[21] T. Kumar and K. Verma, "A theory based on conversion of rgb image to gray image," *International Journal of Computer Applications*, vol. 7, no. 2, pp. 7–10, 2010.

[22] R. Volk, J. Stengel, and F. Schultmann, "Building information modeling (bim) for existing buildings—literature review and future needs," *Automation in construction*, vol. 38, pp. 109–127, 2014.

[23] S. Karimi, I. Iordanova, and D. St-Onge, "An ontology-based approach to data exchanges for robot navigation on construction sites," *arXiv preprint arXiv:2104.10239*, 2021.

[24] P. Staczek, J. Pizon, W. Danilczuk, and A. Gola, "A digital twin approach for the improvement of an autonomous mobile robots (amr's) operating environment—a case study," *Sensors*, vol. 21, no. 23, p. 7830, 2021.