Proceedings of the 12th International Conference of Control Systems, and Robotics (CDSR 2025)
July 13, 2025 - July 15, 2025 | Imperial College London Conference Center, London, United Kingdom
Paper No. 145
DOI: 10.11159/cdsr25.145

Study of Unknown Environments Exploration Techniques Based on ROS-Enabled Omnidirectional Mobile Robot

Djaidja Djabir[1], Besseghieur Khadir[1], Nemra Abdelkrim[1], Boukouira Zineddine[1], Dellali Rihem[1]
[1]Ecole Militaire Polytechnique
BEB, Algiers, Algeria
djaidja.djabir@gmail.com;

Abstract **-** Autonomous exploration of unknown environments is a critical aspect of robotics, enabling robots to navigate and map uncharted areas without human intervention. This research focuses on various exploration techniques, with particular emphasis on those that allow robots to autonomously generate environmental maps. Among classical methods, frontier-based exploration plays a key role by guiding robots to previously unexplored regions using knowledge of already mapped areas. Additionally, sampling-based approaches such as Rapidly-exploring Random Trees (RRT) offer effective strategies for path planning in complex, high-dimensional spaces. This paper reviews and analyzes these exploration techniques, highlighting their strengths and limitations within the context of autonomous robot navigation. The goal of this work is to compare two prominent exploration techniques-frontier-based exploration and Rapidly-exploring Random Trees (RRT)-in the context of autonomous robotic mapping of unknown environments. By evaluating their respective strengths and limitations, this study aims to clarify the conditions under which each method is most effective and to inform the selection of exploration strategies for practical robotic applications.
Keywords**:** Autonomous exploration, path planning, frontier-based exploration, mapping, random sampling methods, Rapidly-exploring Random Trees, RRT, SLAM.

1. Introduction

Autonomous exploration of unknown environments is a fundamental capability in mobile robotics, particularly for applications involving search and rescue, environmental monitoring, and surveillance. The objective is to enable the robot to incrementally build a representation of the environment while navigating safely and efficiently. This process involves repeatedly updating the map, identifying unexplored regions, and planning efficient trajectories to observe new areas, all while optimizing resource usage such as time, energy, and trajectory length to achieve mission objectives.

Robotic exploration methods can be grouped into several high-level categories based on how they choose where to move to discover unknown space. Frontier-based exploration remains a fundamental strategy in autonomous robotics, guiding robots toward the boundary between known free space and unexplored regions termed frontiers. Frontier-based exploration was introduced by Yamauchi (1997) in[1], and remains widely adopted due to its simplicity and effectiveness in diverse environments. Information-theoretic methods formulate exploration as a decision-making problem where the robot selects actions or goals that maximize expected information gain. These approaches explicitly reason about uncertainty and seek to make exploration decisions that are most informative about the environment,[2].

Reinforcement learning (RL) methods enable robots to learn exploration policies through trial and error, guided by intrinsic or extrinsic rewards that encourage efficient coverage or discovery of novel areas, in [3] Cimurs et al, formulate exploration as a goal-driven DRL task, learning both high-level waypoint selection and low-level navigation without a prior map. Random sampling methods, like Rapidly-exploring Random Trees (RRT), provide a probabilistic framework for exploration, facilitating effective path planning in high-dimensional environments. Sampling-based exploration builds random trees to identify promising exploration paths. These methods can quickly find diverse exploration goals without exhaustively searching the map.

In this work, we present a comparative evaluation of frontier-based and random sampling-based exploration strategies, assessing their performance both in simulation and on real ROS-enabled mobile robots. The structure of the paper is as follows: Section II presents the exploration techniques under study, detailing the principles behind the frontier-based and random sampling strategies. Section III provides implementation details, including the ROS integration and system architecture. Section IV describes the experimental setup, covering both the virtual simulation environment and the real-world deployment using physical robots.

2. Methodology

2.1. Frontier-Based Exploration

Frontier-based exploration is a foundational strategy in autonomous mobile robotics, designed to efficiently map unknown environments. The core idea is to direct the robot toward frontiers-the boundaries between known, explored free space and adjacent unexplored regions. By systematically moving to these frontiers, the robot incrementally expands its map, ensuring efficient coverage and rapid discovery of new areas [1], [4]. The frontier-based method typically involves three main steps, detecting frontier regions, selecting the most promising frontier point, and planning a path to reach it. Over the years, researchers have proposed various enhancements to improve the efficiency, scalability, and robustness of these algorithms. The frontier-based steps are shown in Table 1.

Table 1: Algorithm steps for frontier-based exploration.

Frontier-Based Exploration
1. Initialize map and robot pose
2. Update Continuously map with sensor data (e.g., laser scans).
3. Detect frontiers that are free and adjacent to unknown cells
4. Select next goal frontier to explore next, often the closest or highest information gain frontier.
5. Plan path to selected frontier (e.g., A*, Dijkstra).
6. Navigate to frontier: move the robot along the planned path to the frontier location.
7. **Repeat** the process until no frontiers remain.

Frontier-based exploration is an essential technique for autonomous robots to explore and map unknown environments. Its efficiency and simplicity make it a popular choice for both mono-robot and multi-robot exploration tasks.

2.2. RRT Exploration Method

RRT is an exploration strategy that incrementally builds a tree to cover unknown environments by randomly sampling points and extending the tree toward them. Its inherent bias toward unexplored space makes it well-suited for autonomous robotic exploration, enabling rapid initial coverage. As the robot moves, new branches are added toward randomly selected targets, promoting wide area discovery without prior knowledge of the environment. This randomness allows RRT to handle complex, obstacle-rich spaces effectively. The RRT steps are shown in Table 2.

Table 2: Algorithm steps of Rapidly-exploring Random Trees (RRT).

RRT Exploration
1. **Initialize** Tree with the start configuration.
2. **Sample** a random point in the Robot configuration space.
3. **Find the nearest node** in the tree to this random point.
4. **Extend the tree** from the nearest node toward the random point (limited by step size).
5. **Check for collision** between the new edge and any obstacles.
● If collision-free, add the new node to the tree; otherwise, discard the sample.
6. **Check if the goal is reached** or a stopping criterion is met (e.g., time, iterations).
● If not, return to Step 2 and repeat; if yes, stop the algorithm.

3. Implementation of exploration techniques under ROS

3.1. ROS framework

Implementing of the RRT and frontier-based exploration algorithms requires the integration of several ROS packages to ensure full system functionality. The Robot Operating System (ROS), an open-source framework for robot software development, provides a modular environment where various components can communicate efficiently. Specifically, the ROS Navigation Stack [5] is employed to handle path planning and obstacle avoidance, while a SLAM package is used for real-time mapping of the unknown environment.

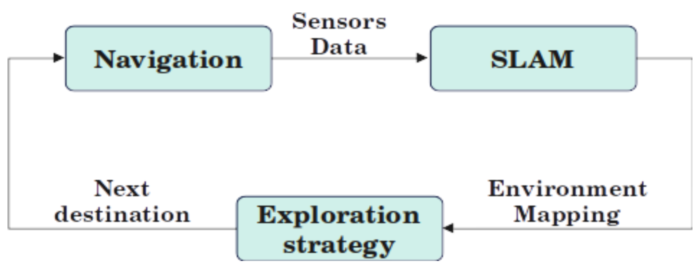

Fig. 1: Autonomous Exploration components.

Robot Operating System (ROS) provides standard libraries and tools for building robot applications. It is not a traditional OS but rather a distributed collection of nodes (independent processes) that communicate to perform sensing, computation, and control. For example, ROS enables seamless integration of different processes (nodes) in a robot system, promoting modularity and code reuse across robots.

In the implementation of both the RRT and frontier-based exploration strategies, several key ROS packages were integrated to support mapping, path planning, and trajectory execution. The "gmapping" package [6] was used to perform Simultaneous Localization and Mapping (SLAM), allowing the robot to incrementally build a 2D occupancy grid map of the unknown environment while localizing itself.

For global path planning, the Dijkstra's algorithm was employed, providing efficient computation of obstacle-free paths to exploration targets. To handle local trajectory generation and real-time obstacle avoidance, the Timed Elastic Band local planner was used, which is well-suited for holonomic robots navigating in dynamic settings. Additionally, the move base package, the core of the ROS Navigation Stack, served as the central interface to coordinate global and local planning components, manage goals, and ensure the execution of safe trajectories toward frontiers.

3.2. Frontier-based implementation

We used the "explore_lite" package from the m-explore repository to implement a frontier-based exploration strategy within our robotic setup. This package provides an efficient solution for autonomous exploration by detecting frontiers and navigating to them. To adapt it for the Robotino platform, we customized the package to interface with Robotino's sensors and actuators.

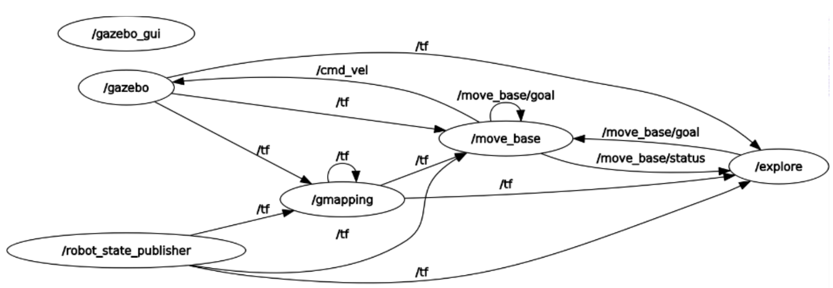

Fig. 2: Frontier-based scheme (explore-llite package).

3.3. RRT implementation

We used the rrt_exploration ROS package developed by Hassan Umari [7], [8], which implements an exploration strategy based on the Rapidly-Exploring Random Tree (RRT) algorithm. This package includes several nodes for global and local frontier detection, frontier filtering and task assignment. Originally designed for Kobuki robots, we adapted it to work with the Robotino by modifying the robot-specific configurations and integrating it with Robotino's control and sensor interfaces. This adaptation enabled the application of the RRT-based exploration strategy in our specific robotic setup.

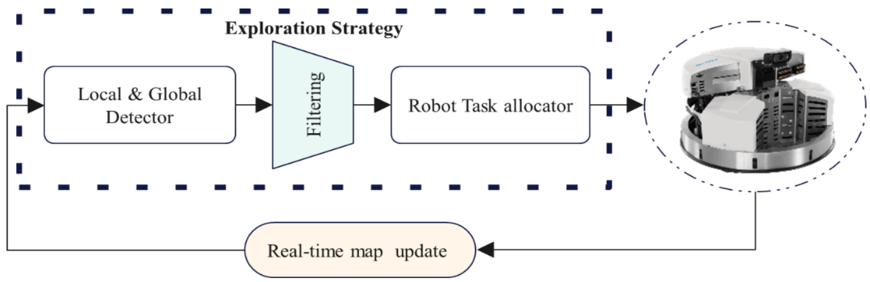

Fig. 3: Frontier-based scheme (explore-llite package).

4. Results and Discussions

4.1. Simulation Setup

Simulations were conducted using the Gazebo simulator, which offers realistic physics-based modelling of robotic motion and sensor behaviour. To evaluate the exploration strategies in a structured indoor environment, the AWS RoboMaker Small House World [9] was utilized. This world provides a detailed simulation of a multi-room house complete with furniture and obstacles, making it suitable for testing autonomous navigation and mapping algorithms in confined and realistic settings.

4.2. Experimental Setup

In our experimental setup, we used the Robotino 3 mobile robot platform, which is equipped with an omnidirectional drive, enabling agile movement in various directions. The robot is integrated with an onboard computer that facilitates real-time data processing and communication. For perception, we employed the Hokuyo URG-04LX laser scanner, a compact 2D LiDAR sensor that offers a 240° scanning range with a maximum detection distance of 4 meters, Figure 4 shows the environment set around exploration.

Fig. 4: Simulation environment and experimental platform.

4.2. Results

The maps obtained during the experiments clearly demonstrate the robot's ability to autonomously navigate and explore its environment. As shown in Figure 5 and Figure 6, The robot, guided by the RRT exploration algorithm or the frontier-based algorithm, successfully identified and navigated to unexplored regions while avoiding obstacles. Both strategies achieved full exploration, each demonstrating different performance characteristics.

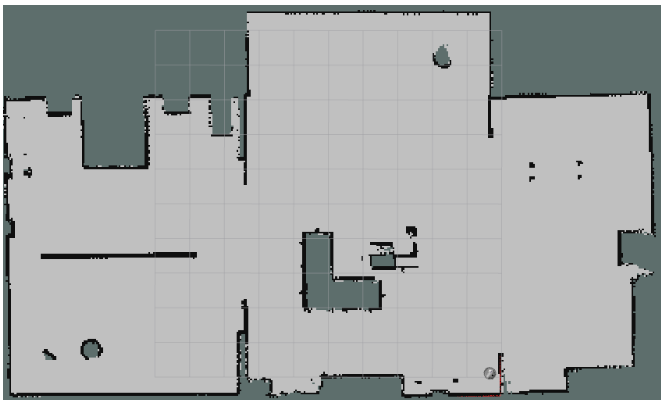

Fig. 5: Map with frontier-based exploration.

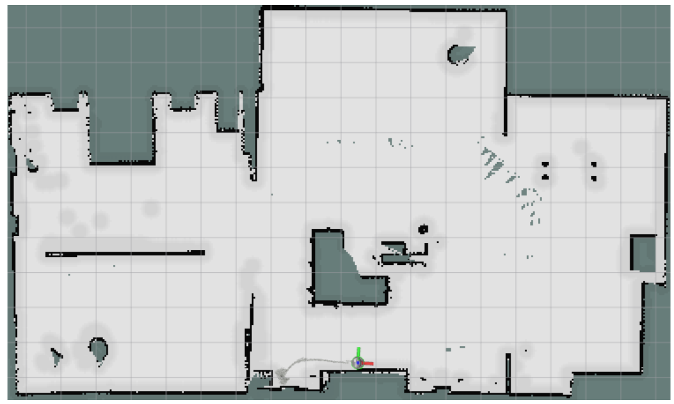

Fig. 6: Generated Map with RRT.

The two maps in Figure 5 and Figure 6, show the results of exploration using the two studied methods. The frontier-based map appears cleaner and more consistent, indicating that the robot followed a more organized exploration strategy. On the other hand, the RRT-based method covered slightly more area and revealed more details around corners, but it suffers from noise due to its probabilistic nature. This highlights a trade-off between the extent of exploration and the overall quality of the generated map.

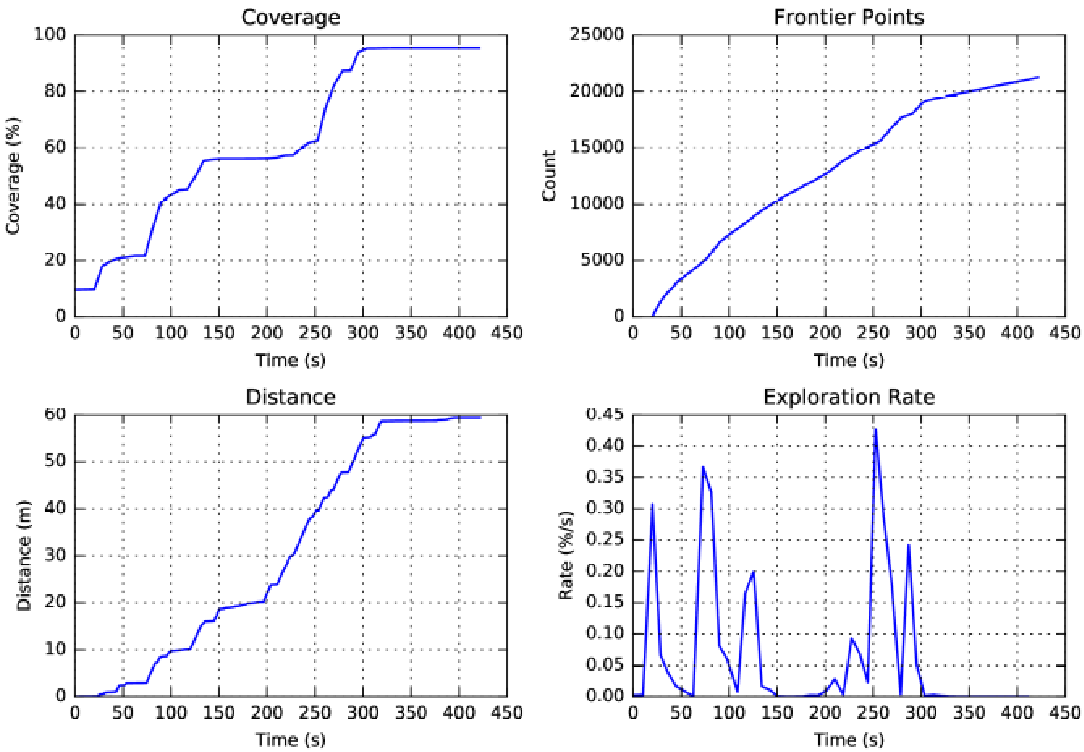

Fig. 7: RRT Exploration performances.

From Figure 7 RRT-based exploration algorithm demonstrates steady coverage growth and effective initial expansion into unknown areas. However, the noticeable fluctuations in the exploration rate suggest that while RRT can make rapid progress at times, it also faces challenges. Due to its random sampling nature, RRT may not always find the most efficient path or plan effectively in complex areas, leading to reduced efficiency. These issues become more evident near the end of the exploration, when the robot needs to navigate small and hard-to-reach areas. Despite these challenges, the algorithm successfully covers the entire map, as indicated by the consistent behavior in the distance traveled and frontier count curves. This demonstrates RRT's effectiveness in quickly exploring large areas initially. However, its performance could be enhanced by integrating refinement strategies to ensure smoother and more consistent progress, especially in the later stages of the mission.

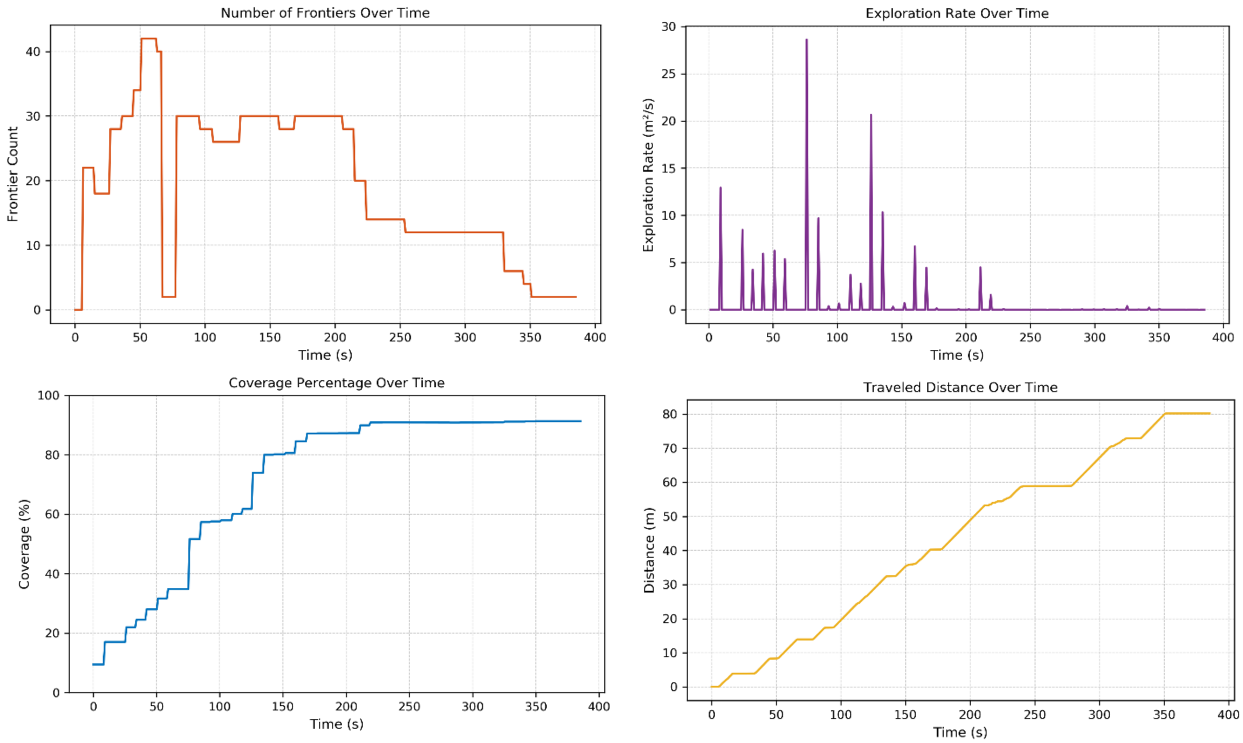

Fig. 8: Frontier-based exploration performances.

The results of the frontier-based exploration strategy as shown in Figure 8 and in Table 3 demonstrate its ability to provide smooth, efficient, and consistent exploration performance. As seen in the Coverage Percentage Over Time and Explored Area Over Time plots, the robot achieved a steady and rapid increase in map coverage, reaching high percentages early in the mission. This indicates the strategy's effectiveness in continuously identifying informative frontiers and directing the robot efficiently. The Travelled Distance Over Time graph supports the method's efficiency, showing continuous movement with minimal pauses or detours.

Table 3: Algorithm steps for frontier-based exploration.

	Frontier-based	RRT
Total Exploration Time	384 s	421.82 s
Average Exploration Rate	0.36 m²/s	0.38 m²/s
Total Coverage Area	155.27 m²	162.32 m²
Total Coverage Percentage	95.16 %	95.48 %
Total Travelled Distance	80.22 m	59.35 m

Overall, Frontier-based exploration is faster and smoother than random strategies like RRT, especially in structured environments. It keeps consistent progress and avoids major navigation issues, making it a strong choice for real-world use.

However, it can struggle in cluttered or dynamic environments where frontiers are hard to find or change often. On the other hand, RRT is better for quickly covering open or complex spaces and is good at handling irregular layouts. But it often has unstable performance in tight or obstacle-rich areas, which can slow down exploration. In summary, the frontier-based method is faster but involves more movement, while RRT covers slightly more area with better path efficiency but takes more time.

5. Conclusion

In this work, we compared two widely used exploration strategies—frontier-based and RRT-based—on a ROS-enabled omnidirectional mobile robot. Both methods allowed the robot to successfully explore and map unknown environments, each showing different performance characteristics. The frontier-based method produced more structured and consistent maps, while the RRT-based strategy provided faster initial coverage and better performance in complex areas.

This study highlights the strengths and limitations of each approach and provides a basis for further research. Future work could focus on extending these strategies to multi-robot systems to improve efficiency and scalability. Another important direction is the exploration of three-dimensional environments, which presents new challenges in perception, planning, and coordination. These perspectives will contribute to more advanced and robust autonomous exploration systems.

References

[1] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. "Towards New Computational Principles for Robotics and Automation,"* Monterey, CA, USA: IEEE Comput. Soc. Press, 1997, pp. 146–151. doi: 10.1109/CIRA.1997.613851.

[2] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ International Conference on Intelligent Robots and System*, Lausanne, Switzerland: IEEE, 2002, pp. 540–545. doi: 10.1109/IRDS.2002.1041446.

[3] R. Cimurs, I. H. Suh, and J. H. Lee, "Goal-Driven Autonomous Exploration Through Deep Reinforcement Learning," Sep. 09, 2021, *arXiv*: arXiv:2103.07119. doi: 10.48550/arXiv.2103.07119.

[4] A. Buriboev, A. J. Choi, and H. S. Jeon, "Optimized Frontier-Based Path Planning Using the TAD Algorithm for Efficient Autonomous Exploration," *Electronics*, vol. 14, no. 1, p. 74, Dec. 2024, doi: 10.3390/electronics14010074.

[5] "ROS/Concepts - ROS Wiki." Accessed: May 09, 2025. [Online]. Available: https://wiki.ros.org/ROS/Concepts

[6] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007, doi: 10.1109/TRO.2006.889486.

[7] H. Umari, *hasauino/rrt_exploration*. (Apr. 28, 2025). Python. Accessed: May 09, 2025. [Online]. Available: https://github.com/hasauino/rrt_exploration

[8] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC: IEEE, Sep. 2017, pp. 1396–1402. doi: 10.1109/IROS.2017.8202319.

[9] *aws-robotics/aws-robomaker-small-house-world*. (Apr. 18, 2025). CMake. AWS Robotics. Accessed: May 09, 2025. [Online]. Available: https://github.com/aws-robotics/aws-robomaker-small-house-world