Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 136 (2025) 342–347

www.elsevier.com/locate/procedia

35th CIRP Design 2025

Containerized Pipeline for Standardized and Automated Testing of Global Path Planning Algorithms in ROS 2

Henrik Lurz[a,*], Annika Raatz[a]

[a]*Institute of Assembly Technology and Robotics, Leibniz University Hannover, An der Universität 2, 30823 Garbsen, Germany*
* Corresponding author. Tel.: +49 511 762 12214; *E-mail address:* lurz@match.uni-hannover.de

Abstract

Global path planning algorithms are essential for a mobile robot to reach a target location while finding the shortest, most energy-efficient or smoothest path. However, there is currently no standardized method to evaluate the suitability of a global path planning algorithm for specific applications. For instance, path length may be the most critical factor in logistics, while energy efficiency could be more important for exploring inaccessible terrain. This paper introduces a containerized pipeline for evaluating global path planning algorithms, providing a transferable, repeatable, and comparable framework. The pipeline integrates automated data recording and streamlined visualization, making it accessible even for non-specialists. We conduct a literature-based review of standardized metrics that form the foundation for the pipeline's data analysis. The proposed pipeline utilizes the ROS2 navigation stack and Apptainer containerization, enabling a consistent and flexible testing environment across different platforms. Evaluation results demonstrate the pipeline's capability to reliably assess algorithms across diverse hardware setups, highlighting its potential for cloud-based evaluation of global path planning algorithms. This work offers a promising step toward a standardized approach to global path planning evaluation, with future plans for integrating local path planning.

Keywords: Global Path Planning, Mobile Robotics, Automation

1. Introduction

Modern mobile robots are getting more and more intelligent, not only due to the enhancements in AI research but also the improvements in path planning [1]. This is driven by the rapidly growing global industry of mobile robotics, especially in the logistics and service applications [2]. In these applications mobile robots are relying on the path planning process that allows them to travel to a target location and fulfill their task.

Generally, the path planning of a mobile robot can be separated into global and local path planning. Global path planning uses the current position of the mobile robot, a target location and a known map of the environment to determine a collision-free path to the target location. Local path planning in its simplest form is utilizing a controller to steer the mobile robot along the global path with the least deviation. Both functionalities in the navigation process are important but, in this work, we will focus on the global path planning since this is the basis for a successful navigation. Global path planning has major effects on mobile robot navigation, for example the collision avoidance, the path efficiency and length. However, the properties of global path planning algorithms cannot all be fulfilled simultaneously. This results in various algorithms that are optimized for a specific property. Because mobile robots are used in diverse applications, each with unique requirements, certain global path planning algorithms are better suited to specific tasks than others. The wide variety of global path planning algorithms makes selecting the optimal one for a specific task highly challenging, particularly for someone without specialized expertise. Multiple review papers are trying to present an overview of the existing global path planning algorithms [3–5]. However, a single review paper clearly cannot cover all existing global path planning algorithms or those that will be developed in the future. This makes it necessary to compare algorithms across multiple

10.1016/j.procir.2025.08.060

review papers. Since each review paper uses different evaluation methods and metrics, comparing their findings becomes inconsistent or even unfeasible.

The goal of this research is the development of an open-source testing and evaluation pipeline for global path planning. By employing the proposed pipeline, the evaluation of arbitrary global path planning algorithms becomes more transferable, repeatable, and consequently, more comparable. To achieve this, we give a literature-based review on standardized metrics, which form the foundation for the data analysis performed by the pipeline. The integration of automated, standardized data recording and streamlined visualization further accelerates the evaluation process, enabling non-specialists to independently assess global path planning algorithms effectively.

The paper begins with an overview of existing evaluation processes in the field of mobile robotics, followed by an overview of the metrics used in current review papers. Next, the design of the developed pipeline is explained, along with a presentation of results generated from it. Finally, the paper concludes with a summary and an outlook on the pipeline and the standardized metrics introduced.

2. State of the Art

In today's world, every industrially used technology follows a standard. These standards define not only the required specifications for a given technology but also the methods for accurately measuring them. One example is the ISO 9283 [6] which defines the performance criteria of industrial robots. Another example is the BOP-challenge for the 6D pose estimation [7], which, while not an industrial standard, serves as a widely recognized benchmark within the research community. Some standards have already been established for the mobile robotics industry, such as ISO/CD 21423 [8] for communication protocols and ANSI/RIA R15.08-1-2020 [9] for safety requirements. However, currently there is no comparable approach to the ISO 9283 or the BOP-Challenge for defining performance metrics or the evaluation process in mobile robotics, including areas such as global path planning, which is the focus of this paper.

Currently, the evaluation of global path planning algorithms remains largely within the research community, with limited adoption in the industry. Several research papers have introduced benchmark systems [10–13], each tailored to specific aspects of path planning. Sturtevant proposed a repository of maps and path planning problems to unify the benchmarking process [10]. These maps originate from video games, artificial environments and real-world examples. Each map includes predefined start and target positions to ensure consistent testing across different path planners. While this provides a solid foundation for a unified testing environment, it primarily evaluates navigation software as a whole, making it difficult to isolate the advantages of individual components. Moreover, the use of different navigation software stacks can lead to inconsistent results, reducing reproducibility. Heiden et al. introduced a motion-planning benchmark specifically for wheeled mobile robots [11]. Their system presents various scenarios with different map sizes and path planning challenges, addressing the lack of a generic benchmarking tool for mobile robots. However, their benchmarking focusses exclusively on wheeled mobile robots using the Ackermann kinematic model, limiting its applicability to other robot types. Additionally, it relies on custom-developed interface for path planners, making it challenging to integrate with the widely adopted Robot Operating System 2 (ROS2), thereby requiring extra effort to transition existing planners into the benchmarking framework. Althoff et al. proposed modular benchmarks for roadway motion planning [12], introducing the CommonRoad platform, which provides scenarios with unique identifiers containing vehicle dynamics, road networks and obstacles. These scenarios are based on real traffic recordings or hand-crafted edge cases. However, this benchmark is specifically designed for autonomous road vehicles and does not address mobile robots operating in for example a factory environment. Vagale et al. proposed an evaluation method for path planning of surface vehicles emphasizing risk and safety metrics [13]. However, their assessment lacks performance measures such as path length or planning time. Furthermore, the distinct requirements of surface vehicles make direct application to wheeled mobile robots impractical. In summary, existing approaches to standardizing the evaluation of mobile robots and their software are not universally suitable. Currently, no research provides a standardized methodology for evaluating global path planning algorithms. While various studies propose individual metrics, the absence of a unified framework hinders consistent evaluation, making it difficult to compare and assess path planning algorithms effectively.

Numerous review papers on global path planning algorithms have led to a range of available performance metrics. One of the most comprehensive list of performance metrics was proposed by Tsardoulis et al. in 2016 [3]. The included metrics were: Mean execution time, relative standard deviation of execution time, mean path distance, relative standard deviation of path distance, path anomaly, estimated time of traversal and the success rate. Later, they applied the proposed metrics to nine roadmap-based global path planning algorithms. In 2023 Liu et al. released a review paper on path planning which contained over 100 global path planning algorithms [4]. They used seven performance metrics that align with Tsardoulis et al. However, in the publication the authors have not applied the listed performance criteria to all listed global path planning algorithms. As a result, the publication provides no comparison or guidance on the optimal global path planning algorithm based on the performance metrics. Wahab et al. published a comparative review on mobile robot path planning for classic and meta-heuristic methods [14]. In this review the authors used the following set of global path planning metrics: Success rate, number of collisions, number of displacements, execution time, battery consumption, travelled distance and convergence iteration. The used metrics partially align with the presented metrics of Tsardoulis et al. However, some of the metrics used, such as battery consumption, can only be evaluated when incorporating local path planning and are therefore not meaningful when considering global path planning alone.

In summary, global path planning lacks a standardized methodology for consistently evaluating and comparing algorithms across common performance metrics. While numerous metrics have been proposed, no unified framework applies them comprehensively to all path planning algorithms. As a result, there is no clear guidance on the relative strengths or optimality of different approaches, making it difficult for practitioners to choose the most suitable algorithm for specific applications. The proposed pipeline aims to bridge this gap by offering an automated, configurable evaluation process that adapts to various user needs. Additionally, the integrated evaluation scripts will incorporate key performance metrics, enabling automatic visualization and ensuring replicable results.

3. Containerized Pipeline Concept

The previous chapter highlighted several key gaps in the current state of the art. To effectively address these issues, a set of specific requirements must be met. These requirements are essential for creating a comprehensive solution that overcomes the previously identified limitations and advances the field:

- The global path planning algorithm under evaluation must be exchangeable
- Mobile robot environment must be interchangeable
- Experiments must be reliably replicable
- Resulting data must be readily accessible or automatically plotted
- Pipeline must be easy to use

To address the mentioned requirements, our proposed pipeline leverages two key components: ROS2 navigation stack and containerization for virtualization. Firstly, the ROS2 navigation stack, a widely adopted framework for mobile robot navigation, is used to handle the core navigation tasks, ensuring the pipeline's compatibility with standard robotics systems. Secondly, containerization provides a lightweight virtualization solution that avoids the need to emulate a full physical computer or its low-level hardware components. This allows for efficient and consistent execution of the pipeline across different environments. Container virtualization involves packaging and isolating the software components required to run a specific application. There are multiple providers that develop platforms for using virtualized containers. In our case, we have decided to use Apptainer [15]. Apptainer supports the use of other popular container providers and allows containers to be executed by unprivileged users. This is especially important when using a container on a decentralized computer like a computer cluster, where a user does not have superuser rights. To facilitate easy distribution and ensure consistency across different environments like a computation cluster or a typical office computer, we created an Apptainer definition file that includes all necessary applications and dependencies to build and run the global path planning testing and evaluation pipeline. Defining specific software versions for applications and dependencies is essential, as it ensures that all users operate with the same foundational setup, thereby producing consistent and reliable results. This approach enhances replicability in testing.

Fig. 1 illustrates the proposed pipeline, with green-outlined boxes marking components where user interaction is needed or optional, and blue-outlined boxes indicating the pipeline's outputs. The pipeline is organized into four primary sections. On the left side of Fig. 1 are the essential user interactions preparing a global path planning test. This preparation includes specifying software and system requirements in a definition file. Additional settings - such as the mobile robot model, global path planning algorithm, and testing environment - are configured in a user-defined configuration file. This feature allows users to either utilize the default global path planner included in the ROS2 navigation stack or incorporate a custom global path planning algorithm into the pipeline for evaluation against standardized path planning algorithms. Once these parameters are set, the user can initiate either the container build process or execute the container to start the testing pipeline.

In the upper-middle section of Fig. 1, the container-building process is illustrated. Based on the definition and configuration files, Apptainer generates an executable container file. For this setup, Ubuntu 22.04 with ROS2 Humble has been selected. The resulting container is shown in the lower middle of Fig. 1. Although users cannot modify preinstalled programs within the container, they can specify which processes should be executed inside. The definition of these processes is part of the pipeline and therefore predefined. The user only initiates execution by running an Apptainer command, which launches the container with all settings defined in the configuration file. This action triggers a complete testing cycle, including a specified number

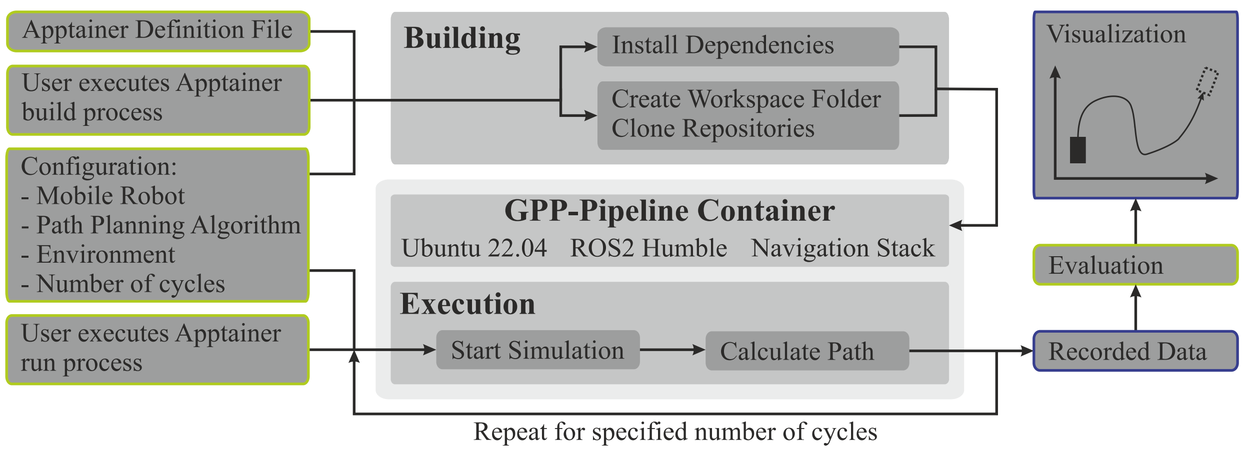

Fig. 1: Overview of the GPP-Pipeline container displaying the building and execution processes while showing inputs and outputs of the pipeline.

of path planning attempts. Each cycle executes the path planner for the defined number of iterations. Each cycle in the sequence begins with launching the simulation environment and the ROS2 navigation stack, followed by positioning a mobile robot at a designated starting point. A target position is then sent to the ROS2 navigation stack, which calculates a path using the selected global path planning algorithm. This path is stored for later evaluation. Once all data is saved, all processes close, and the cycle repeats until the specified number of cycles is reached.

Upon completion of execution and data recording, the data can be evaluated, as shown on the right side of Fig. 1. Users can either use the provided evaluation scripts to analyze performance with the standard global path planning metrics outlined in Chapter 2, or extend the pipeline's functionality by creating custom evaluation scripts. All evaluation scripts access the recorded data, generating plots that display metrics from individual tests or comparisons across multiple algorithms. This setup provides a concise overview of the critical performance metrics of the path planning algorithms under analysis. Furthermore, the pipeline allows enough customization to be used as an efficient evaluation tool for various performance tests in global path planning.

4. Evaluation of Global Path Planning Metrics

The proposed pipeline is executed across various properties to assess its feasibility as a standardized evaluation procedure. Since this study focuses on evaluating the pipeline itself rather than comparing multiple global path planning algorithms in depth, we emphasize the transferability and repeatability of results across different hardware.

The first property tested is the selection of global path planning algorithms. For this, we use the three default algorithms from the ROS2 navigation stack: NavFN planner, Smac planner, and Theta Star planner. All three planners are derived from foundational Dijkstra or A* path-planning algorithms, utilizing grid-based search techniques to optimize pathfinding. These algorithms produce consistent and repeatable results, making them ideal for testing the pipeline's transferability and reliability.

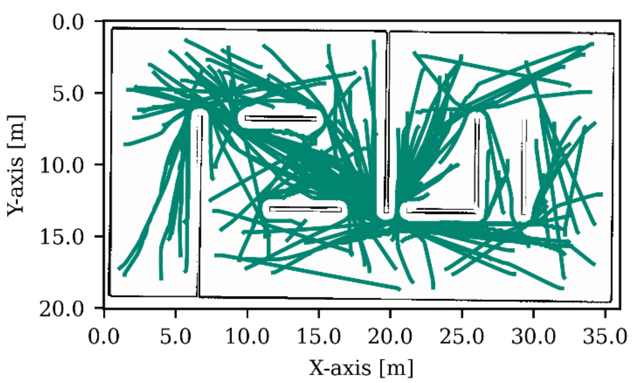

Fig. 3: 2D-map of the environment with all obstacles displayed in black. The green lines show the 100 paths planned by the NavFN planner.

The second property is the simulated testing environment visualized in Fig. 4, which we custom-designed to include multiple passageways with varying widths. This environment provides multiple possible paths to the target, allowing each algorithm to demonstrate its approach to path optimization.

The third property involves start and target positions, between which the algorithms calculate paths. For consistency, we use a set of 100 randomly generated start and target pairs across all experiments.

The fourth property is the computing system on which the pipeline is executed. We tested the pipeline on three different systems: (1) a Dell Precision 3551 laptop with an Intel i5-10400H processor, integrated Intel UHD graphics, and 16GB of RAM; (2) a desktop with an Intel i5-11600 processor, a Nvidia GTX 4060 GPU, and 16GB of RAM; and (3) the Leibniz University of Hannover's computing cluster, featuring an Intel Haswell Xeon E5-2630 v3 processor. For tests on the cluster, we allocated four cores and 16GB of RAM, with no dedicated GPU.

After defining these four properties, we begin the evaluation by testing the repeatability of the pipeline. This is conducted through executing the same number of experiments (100 times) with the same global path planner (NavFN) on the same computing system (desktop computer) five times. Fig. 2 shows the path length for all 100 different start and target

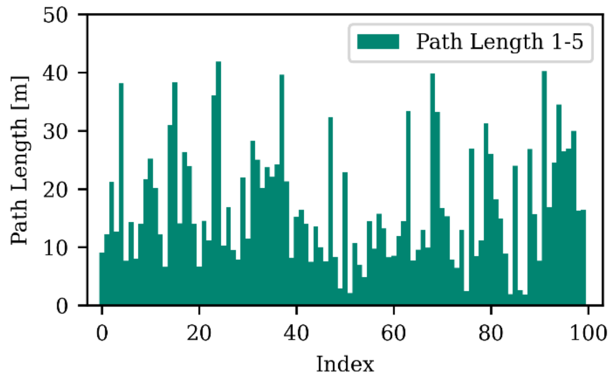

Fig. 2: Shown are 100 path planning tasks, each executed five times. For each task, the path lengths are consistent across all five executions whenever the planning algorithm successfully generated a solution.

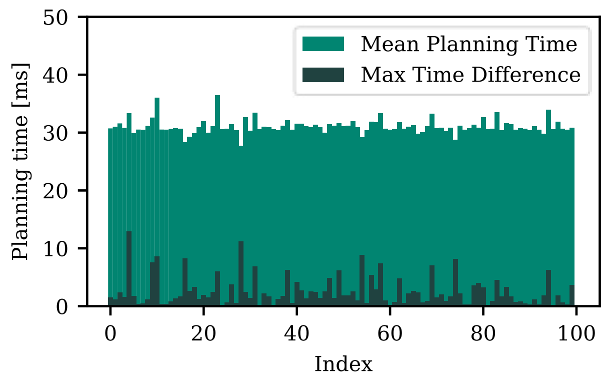

Fig. 4: 100 path planning problems are displayed, with light green indicating the mean planning time across five cycles and dark green bars showing the maximum deviation to the mean value.

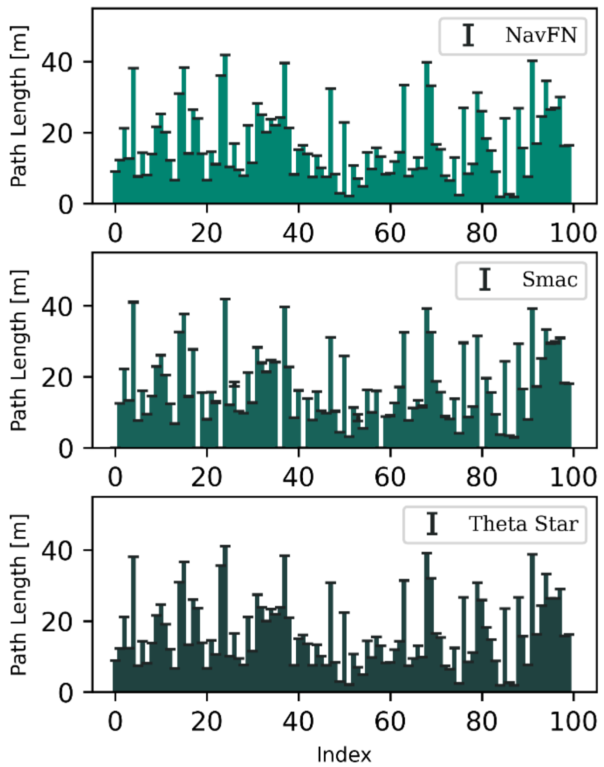

Fig. 5: Shown are 100 path planning tasks for the three path planners. The average path length is plotted with the maximal difference being the shown as the error bar.

positions and the five planning attempts. As expected, the path length of the five cycles for each start and target position set shows no deviation. This is due to the fact that a grid-base and not randomized path planning algorithm will always output the same path for a path planning problem.

Fig. 3 illustrates the average planning time for the five pipeline-execution cycles in light green, with dark green bars representing the maximum deviation between individual runs and the mean value. The planning time remains relatively stable across different path planning problems, showing little fluctuation regardless of path length. Although minor, there is some deviation between the five cycles, likely due to parallel execution of the pipeline alongside other processes on the same processor. Overall, the graphs demonstrate strong repeatability on the same computing system, with only a few outliers.

To assess cross-system consistency, we examine the repeatability of path planning results across different computing systems. For this, the three systems presented (computing cluster, desktop computer, and laptop) each execute three global path planning algorithms (NavFN, Smac, and Theta Star) using the same 100 start and target position sets. Fig. 5 presents three subplots, each depicting the path lengths generated by one of the global path planning algorithms: NavFN, Smac, and Theta Star. Each bar in the charts shows the average path length calculated for a specific path planning instance, with black error bars indicating the maximum deviation between the average path length and individual results across the different hardware. The error bars demonstrate that differences in path length across systems are negligible. Only the Smac planner shows a few outliers, with

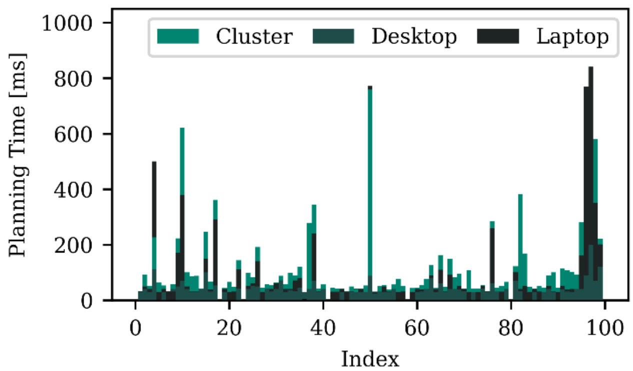

Fig. 6: Planning Time for the 100 different path planning tasks for the Smac planner. The heighest bar shows the slowest planner while a time of 0ms a failed planning attempt represents.

variations under 0.5 meters for paths exceeding 20 meters. These slight variations can be attributed to the tolerance settings within Smac's curve optimization. Overall, the path length data supports the pipeline's capability to reliably evaluate path planning algorithms across diverse hardware setups.

Fig. 6 presents the planning time results, with color-coded bars representing each hardware system. Here, the Smac path planning algorithm is shown as an example due to its higher fluctuations in planning time compared to the other algorithms. Each color in the bar chart corresponds to a specific hardware setup, with the lowest color indicating the fastest planning times. The graph highlights the differences in planning time across computing systems, revealing two main insights. Firstly, planning times vary with the computational complexity of a path planning problem like path length. Secondly, hardware performance characteristics greatly influence planning time, as demonstrated by the desktop computer consistently achieving the fastest calculations, confirming its superior processing power. In contrast, the computing cluster showed the slowest planning times, which can be attributed to the limited resources allocated for this test. Furthermore, the cluster features a processor with a lower clock speed compared to the desktop computer and laptop, as well as slower file connections due to the separation of processors and file storage.

Overall, the results show that path length comparisons across different devices are consistent and reliable within the standardized global path planning pipeline. The data collected by the proposed pipeline demonstrates strong performance across multiple platforms, including cluster computing systems. By leveraging the proposed pipeline, cloud computing can be utilized for evaluating global path planning algorithms. However, hardware-dependent factors significantly influence metrics like planning time, making it unsuitable as a standardized metric. That said, it can still serve as a valuable benchmarking tool to identify the optimal hardware for a specific path planner, particularly when minimizing execution time is crucial.

5. Conclusion and Future Work

This work introduces a structured and replicable solution for evaluating global path planning algorithms in mobile robotics by developing a standardized testing and evaluation pipeline. The proposed pipeline is open-source and accessible on Github [16]. The pipeline's containerized design, utilizing Apptainer, ensures ease of distribution and enables consistent software versions, repeatable experiments across various environments and computing systems. This feature addresses the current limitations in the field, where differing evaluation methodologies in review papers have hindered fair comparisons of algorithms. By providing a unified approach that can be adapted to different robots, environments, and algorithms, our pipeline enables a reliable basis for benchmarking path planning algorithms using clearly defined metrics. The evaluation confirmed the pipeline's suitability for consistent path planning performance testing, demonstrating reliable repeatability on the same computing system and comparability across systems for path-dependent metrics. However, as expected, certain metrics like planning time are influenced by hardware variations and not suited to be included in a standardized comparison.

The current pipeline focuses solely on testing and evaluating global path planning algorithms. In future iterations, we plan to expand its capabilities to fully leverage the Gazebo simulation environment, enabling full navigation simulation with a virtual mobile robot. This enhancement would integrate local path planning, allowing comprehensive evaluation of the entire navigation process. Additionally, we aim to improve user interaction with the pipeline. At present, users must adjust multiple configuration files to apply settings. While this method suits remote cluster computing, a user interface would streamline setup on typical desktop computers, reducing configuration time and enhancing accessibility.

Acknowledgement

The authors gratefully acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG – German Research Foundation) – Project no. 541021498. The authors would like to thank the DFG for the support within the SPP 2433 – Measurement technology on flying platforms.

References

[1] Loganathan A, Ahmad NS. A systematic review on recent advances in autonomous mobile robot navigation. Engineering Science and Technology, an International Journal 2023;40:101343.
[2] Mobile Robots Revolutionize Industry. IFR International Federation of Robotics n.d. https://ifr.org/news/mobile-robots-revolutionize-industry (accessed February 7, 2025).
[3] Tsardoulias EG, Iliakopoulou A, Kargakos A, Petrou L. A Review of Global Path Planning Methods for Occupancy Grid Maps Regardless of Obstacle Density. J Intell Robot Syst 2016;84:829–58.
[4] Liu L, Wang X, Yang X, Liu H, Li J, Wang P. Path planning techniques for mobile robots: Review and prospect. Expert Systems with Applications 2023;227:120254.
[5] Sánchez-Ibáñez JR, Pérez-del-Pulgar CJ, García-Cerezo A. Path Planning for Autonomous Mobile Robots: A Review. Sensors 2021;21:7898.
[6] ISO 9283 - Manipulating industrial robots - Performance criteria and related test methods 1998.
[7] Hodaň T, Michel F, Brachmann E, Kehl W, Buch AG, Kraft D, et al. BOP: Benchmark for 6D Object Pose Estimation. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer Vision – ECCV 2018, Cham: Springer International Publishing; 2018, p. 19–35.
[8] ISO/CD 21423 - Autonomous mobile robots for industrial environments — Communications and interoperability n.d.
[9] R15.08-1-2020 - Industrial Mobile Robots - Safety Requirements - Part 1: Requirements for the Industrial Mobile Robot 2020.
[10] Sturtevant NR. Benchmarks for Grid-Based Pathfinding. IEEE Transactions on Computational Intelligence and AI in Games 2012;4:144–8.
[11] Heiden E, Palmieri L, Arras K, Sukhatme G, Koenig S. Experimental Comparison of Global Motion Planning Algorithms for Wheeled Mobile Robots. ArXiv 2020.
[12] Althoff M, Koschi M, Manzinger S. CommonRoad: Composable benchmarks for motion planning on roads. 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, p. 719–26.
[13] Vagale A, Bye RT, Osen OL. Evaluation of Path Planning Algorithms of Autonomous Surface Vehicles Based on Safety and Collision Risk Assessment. Global Oceans 2020: Singapore – U.S. Gulf Coast, 2020, p. 1–8.
[14] Wahab MNA, Nefti-Meziani S, Atyabi A. A comparative review on mobile robot path planning: Classical or meta-heuristic methods? Annual Reviews in Control 2020;50:233–52.
[15] Apptainer - Portable, Reproducible Containers n.d. https://apptainer.org/ (accessed February 7, 2025).
[16] GPP-Benchmark 2025. https://github.com/match-MobRob2/GPP-Benchmark.