From Simulation to Reality: Autonomous 3D Exploration with DAEP on Heterogeneous Robots

Emil Wiman[1], Mariusz Wzorek[1], Piotr Rudol[1], Tommy Persson[1] and Mattias Tiger[1]

[1]*Department of Computer and Information Science, Linköping University (LiU) and WARA Public Safety (WARA-PS), Linköping, Sweden*

Abstract

Autonomous 3D exploration in real-world environments remains an open problem in robotics. Exploration in the setting of robotics refers to the concept of a robot being able to independently explore an a priori unknown environment with the goal of building an accurate and complete 3D representation of said environment. Most work in the literature has studied this concept in the static setting, while work in a dynamic context is lacking. However, recent dynamic exploration planners have shown promise in effectively handling complex dynamic environments; nevertheless, most evaluations have been conducted only in simulation. In this work we extend and deploy the Dynamic Autonomous Exploration Planner (DAEP) on multiple robotic platforms in complex real-world environments. We demonstrate that DAEP can be extended to new robotic platforms, making it well-suited for further investigation in collaborative robotics research. We perform real-world experiments to validate our extensions and to provide insights into current advances and limitations.

Keywords

Autonomous exploration, Ground Robots, Sim-to-Real Transfer

1. Introduction

Autonomous robotic systems are becoming increasingly present in daily life [1], moving from controlled industrial settings [2] into the real world. For robots to function effectively in ever-changing environments, they need the ability to build and maintain 3D models of their surroundings. These representations are critical for tasks such as navigation [3], object search [4], search-and-rescue [5], and 3D reconstruction [6]. The process of building such representations is known as exploration planning: an agent explores an a priori unknown environment with the aim to map it as completely as possible, see Figure 1. Formally, consider a bounded 3D volume $V_{total} \subset \mathbb{R}^3$ consisting of free and occupied space. The agent must explore and map V_{total} while avoiding collisions with dynamic obstacles, producing a representation M that matches V_{total} as closely as possible in real time.

Exploration in static environments has been well-studied [7, 8, 9], but less work addresses dynamic settings. Recent approaches such as DAEP [10] incorporate dynamic information into the planning process, yet evaluations have been conducted only in simulation. In simulation there is no need to handle imperfect hardware, sensor noise, complex environmental variability, or communication delays. Bridging this sim-to-real gap is essential for deploying robust exploration capabilities on real robots. The contributions of this work are:

- An extended version of DAEP [10] enabling deployment on ground robots in simulated and real-world environments.
- Simulated experiments in both static and dynamic settings.
- Field experiments on multiple platforms to assess DAEP's performance and identify limitations in real-world environments.

RoSE'26: 8th International Workshop on Robotics Software Engineering, June 1, 2026, Vienna, Austria

emil.wiman@liu.se (E. Wiman); mariusz.wzorek@liu.se (M. Wzorek); piotr.rudol@liu.se (P. Rudol); tommy.persson@liu.se (T. Persson); mattias.tiger@liu.se (M. Tiger)

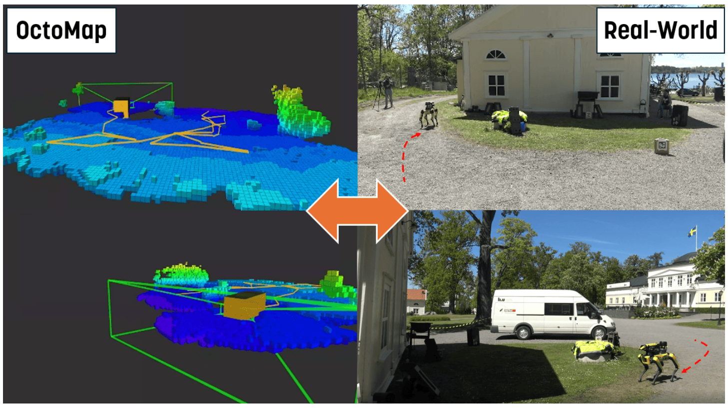


Figure 1: Exploration scenario with Spot in a real-world environment. Left: internal 3D representation. Right: real environment.

2. Related Work

Exploration planning can broadly be divided into frontier-based and sampling-based approaches. Frontier-based methods [11] move the agent towards boundaries between explored and unexplored space, while sampling-based methods use algorithms such as RRT [12] to construct traversable paths and evaluate information gain at sampled viewpoints. The receding-horizon next-best-view planner (RH-NBVP) [7] combined next-best-view sampling with RRT but suffered from premature termination in large environments. The Autonomous Exploration Planner (AEP) [8] addressed this by combining local sampling-based planning with global frontier-based planning, caching unexplored nodes to be explored later during exploration.

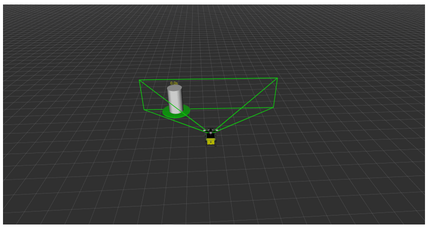
Most prior work considers only static environments. The Dynamic Exploration Planner (DEP) [13] introduced a PRM-based system for reactive obstacle avoidance but scales poorly in large dynamic environments. Dynamic frontiers [14] cluster dynamic obstacles for temporary avoidance, primarily in 2D. DAEP [10] extended AEP to the dynamic setting by incorporating time-of-arrival estimation and dynamic information gain into the RRT expansion, outperforming DEP in both small and large-scale environments. However, all DAEP experiments were conducted in simulation. In this work, we bridge the sim-to-real gap by deploying DAEP on real ground robots.

3. DAEP Background

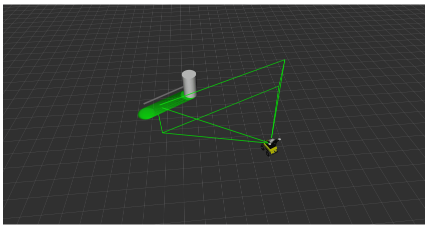
DAEP [10] consists of a local planner that expands a time-based RRT tree, estimating the agent's future time-of-arrival (TOA) in each node. By comparing TOA against predicted dynamic obstacle motion, collision-free paths can be constructed. Each node is scored using a dynamic score that combines dynamic information gain, travel cost, and a dynamic frequency map that prioritizes areas historically occupied by moving obstacles. The local planner executes the first node of the best-scoring branch and re-plans in a receding-horizon fashion. When no local branch exceeds a score threshold, a global planner searches for cached frontier nodes using a time-based RRT* tree. A Kalman filter [15] with a constant velocity motion model provides predictions of dynamic obstacle positions. DAEP has previously only been evaluated on aerial platforms in simulation, with no real-world experiments conducted. For full algorithmic details, see [10].

3.1. Extensions for Ground Robotics

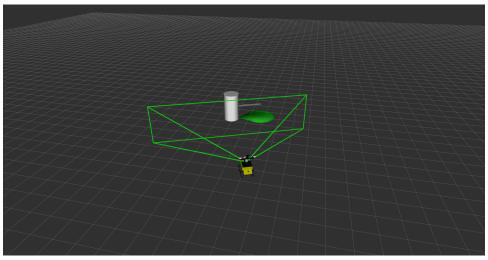
We extend DAEP from aerial to ground platforms in several ways. The RRT expansion is constrained to a 2D plane at the robot's height. The collision bounding volume is changed from a sphere to a cylinder to avoid false collisions with the ground. We also introduce a perception stack for detecting dynamic obstacles (see Figure 2d) in simulation: YOLOv8 [16] provides pedestrian detections that are tracked by



(a) Random walk prediction



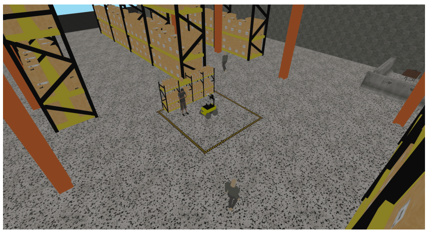
(b) Linear prediction



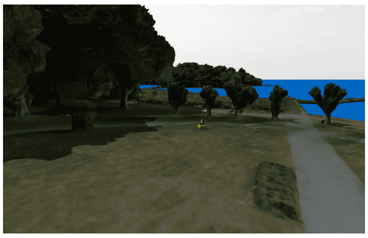
(c) Kalman Filter prediction

(d) Detection and tracking of pedestrian in simulation environment.

Figure 2: Prediction modes (Fig. (2a)-(2c)) in the predictor component. Detection and tracking (Fig (2d)) from the perception stack.



(a) *Warehouse*



(b) *Castle* (Simulation)

(c) *Castle* (Real)

Figure 3: Simulated worlds in benchmark (Fig. (3a)-(3b)) and real environment (Fig. 3c)

OC-SORT [17] and fed into the prediction component, replacing the simulation assumption of known obstacle positions. Additionally, we introduce multiple prediction modes (see Figure 2) to provide flexibility across use cases [18].

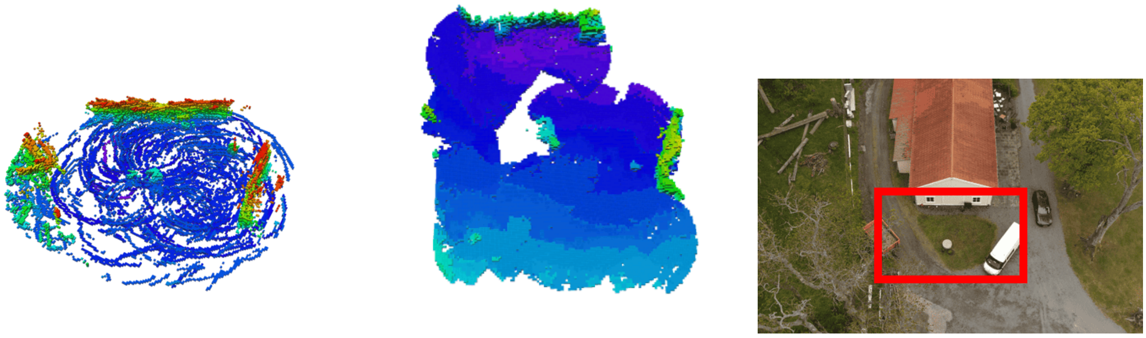
4. System and Implementation

We deploy DAEP on two common research platforms: the ClearPath Robotics Husky A201 and Boston Dynamics Spot. The Husky offers high payload and long battery life but limited terrain agility, while Spot provides exceptional mobility over rough terrain. The Husky is equipped with a Velodyne LiDAR (FoV $60° \times 30°$, 5 m range) and the Spot uses its two front stereo depth cameras (FoV $90° \times 20°$, 5 m range). Both platforms run DAEP on an external computer communicating via Wi-Fi. Localization uses SLAM with a pre-built map for the Husky and built-in GraphNav for Spot; these maps are not accessible to DAEP itself.

The environment is represented using OctoMap [19] at 0.1 m resolution, integrated with ROS2 [20]. The simulation benchmark uses Gazebo Harmonic [21] with the Husky A201 configuration, containerized with Docker [22] for reproducibility. Two simulated worlds are used: *Warehouse* and *Castle* (a 3D scan of a real environment), each in static and dynamic variants, see Figure 3. Spot was excluded from simulation due to lack of an officially released simulation environment.

Table 1
Performance metrics under static and dynamic scenarios (mean ± std).

	Coverage (%)			Time (s)			Path length (m)			Planning time (s)		
Scenario	μ	$\pm$	σ	μ	$\pm$	σ	μ	$\pm$	σ	μ	$\pm$	σ
Static	91.48	$\pm$	4.34	592.63	$\pm$	39.38	81.22	$\pm$	6.31	17.95	$\pm$	4.06
Dynamic	89.79	$\pm$	6.54	533.67	$\pm$	104.23	72.35	$\pm$	13.20	21.26	$\pm$	7.01



(a) OctoMap from Husky. (b) OctoMap from Spot. (c) Exploration area (red boundary).

Figure 4: Real-world exploration results. OctoMaps from Husky and Spot in the Castle environment, with the exploration area shown for reference.

5. Results

Each simulated experiment is repeated five times and reported as $\mu \pm \sigma$. The robot starts at a fixed location and drives forward for 5 seconds before DAEP begins. Exploration continues until the planner terminates or a 10-minute time limit is reached.

5.1. Simulated Experiments

Table 1 summarizes the simulation results. In static scenarios, DAEP achieves $91.48\% \pm 4.34$ coverage with an average exploration time of 593 s. In dynamic scenarios, coverage remains high at $89.79\% \pm 6.54$, though exploration time and path length decrease slightly, suggesting that dynamic obstacles may block certain pathways leading to earlier termination. Planning time increases in the dynamic setting, indicating more re-planning effort when paths are invalidated by moving obstacles.

5.2. Real-World Experiments

Field experiments were conducted in the *Castle* environment (Figure 4c) with both platforms. The resulting OctoMaps (Figure 4a, 4b) show that both robots capture the overall structure. Spot produces a denser OctoMap due to its front-facing depth cameras, while the Husky's forward-oriented LiDAR captures details further up walls. Exploration times were 480 s for Husky and 225 s for Spot. The shorter simulation times are likely due to sensor imperfections causing earlier termination. Spot's ability to rotate while walking also reduces travel time compared to the Husky, as both robots must follow the structure of the RRT tree. In the current implementation, the Husky stops and rotates at the tree's nodes before driving along the edges, whereas Spot can rotate while moving along the edges between nodes. However, both platforms succeed in mapping the environment, confirming that DAEP generalizes across environments and platforms. Note that the field experiments were conducted in a static setting.

6. Conclusion and Future Work

In this work we bridged the sim-to-real gap for DAEP by extending it to ground robots and deploying it in real-world environments on both the Husky and Spot platforms. Simulated experiments show high coverage in both static and dynamic settings, and field experiments confirm that the system generalizes to real sensor data. We identified several limitations: DAEP conflates exploration and motion planning, with its RRT-based traversal lacking kinodynamic feasibility, leading to inaccurate time estimates in real-world dynamic conditions. Additionally, sensor sweeping must be slow enough for accurate mapping, which conflicts with the maneuverability needed in dynamic settings. Future work will address these issues by integrating a dedicated motion planner and investigating collaborative multi-robot exploration.

Acknowledgments

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Declaration on Generative AI

During the preparation of this work, the author(s) used Cursor 2.5.17 in order to transfer the text from the original RA-L template to the CEUR-WS template used for this workshop. After using this tool, the author(s) reviewed the material to ensure that the content was not altered and take full responsibility for the publication's content.

References

[1] J. K, A comprehensive survey on robotics and automation in various industries, in: 2022 7th International Conference on Communication and Electronics Systems (ICCES), 2022, pp. 192–196. doi:10.1109/ICCES54183.2022.9835730.

[2] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall Press, USA, 2009.

[3] M. Harms, M. Kulkarni, N. Khedekar, M. Jacquet, K. Alexis, Neural control barrier functions for safe navigation, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2024, pp. 10415–10422. doi:10.1109/IROS58592.2024.10802694.

[4] W. Ge, C. Tang, H. Zhang, Commonsense scene graph-based target localization for object search, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2024, pp. 13318–13325. doi:10.1109/IROS58592.2024.10801656.

[5] A. Romero, C. Delgado, L. Zanzi, R. Suárez, X. Costa-Pérez, Cellular-enabled collaborative robots planning and operations for search-and-rescue scenarios, in: 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 5942–5948. doi:10.1109/ICRA57147.2024.10611179.

[6] J. Zeng, Y. Li, J. Sun, Q. Ye, Y. Ran, J. Chen, Autonomous implicit indoor scene reconstruction with frontier exploration, in: 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 18041–18047. doi:10.1109/ICRA57147.2024.10611382.

[7] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, Receding horizon "next-best-view" planner for 3d exploration, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 1462–1468. doi:10.1109/ICRA.2016.7487281.

[8] M. Selin, M. Tiger, D. Duberg, F. Heintz, P. Jensfelt, Efficient autonomous exploration planning of large-scale 3-d environments, IEEE Robotics and Automation Letters 4 (2019) 1699–1706. doi:10.1109/LRA.2019.2897343.

[9] B. Zhou, Y. Zhang, X. Chen, S. Shen, Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning, IEEE Robotics and Automation Letters 6 (2021) 779–786. doi:10.1109/LRA.2021.3051563.

[10] E. Wiman, L. Widén, M. Tiger, F. Heintz, Autonomous 3d exploration in large-scale environments with dynamic obstacles, in: 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 2389–2395. doi:10.1109/ICRA57147.2024.10610996.

[11] 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', 1997, pp. 146–151. doi:10.1109/CIRA.1997.613851.

[12] S. M. LaValle, Rapidly-exploring random trees: a new tool for path planning, The annual research report (1998).

[13] Z. Xu, D. Deng, K. Shimada, Autonomous uav exploration of dynamic environments via incremental sampling and probabilistic roadmap, IEEE Robotics and Automation Letters 6 (2021) 2729–2736. doi:10.1109/LRA.2021.3062008.

[14] V. Cavinato, T. Eppenberger, D. Youakim, R. Siegwart, R. Dubé, Dynamic-aware autonomous exploration in populated environments, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 1312–1318. doi:10.1109/ICRA48506.2021.9560933.

[15] R. E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME–Journal of Basic Engineering 82 (1960) 35–45.

[16] G. Jocher, J. Qiu, A. Chaurasia, Ultralytics YOLO, 2023. URL: https://github.com/ultralytics/ultralytics.

[17] J. Cao, J. Pang, X. Weng, R. Khirodkar, K. Kitani, Observation-centric sort: Rethinking sort for robust multi-object tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 9686–9696.

[18] T. Kruse, A. K. Pandey, R. Alami, A. Kirsch, Human-aware robot navigation: A survey, Robotics and Autonomous Systems 61 (2013) 1726–1743. doi:10.1016/j.robot.2013.05.007.

[19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Autonomous Robots (2013). doi:10.1007/s10514-012-9321-0, software available at https://octomap.github.io.

[20] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, Robot operating system 2: Design, architecture, and uses in the wild, Science Robotics 7 (2022) eabm6074. doi:10.1126/scirobotics.abm6074.

[21] N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), volume 3, 2004, pp. 2149–2154 vol.3. doi:10.1109/IROS.2004.1389727.

[22] D. Merkel, Docker: lightweight linux containers for consistent development and deployment, Linux journal 2014 (2014) 2.