# REAL-TIME REACHABILITY ANALYSIS FOR SAFE ROBOTIC MANIPULATION IN DYNAMIC HUMAN ENVIRONMENTS

**HARUTO URAKI[1], TAKENORI OBO [1], NAOYUKI KUBOTA [1]**

[1]Department of Mechanical Systems Engineering, Graduate School of Systems Design,
Tokyo Metropolitan University, Hino, Tokyo 191-0065, Japan
E-MAIL: uraki-haruto@ed.tmu.ac.jp

**Abstract:**

**In dynamic environments, safe trajectory planning for robot manipulators requires early determination of whether a goal configuration is reachable from the current configuration for rapid and appropriate decision-making. However, in high-dimensional dynamic environments, obstacle motions change connectivity in configuration space over time, making reachability assessment difficult. This study proposes a method for real-time topology update of configuration space by incrementally reflecting collision information. Reachable regions are identified online using connected component analysis on the graph. Experiments with a six-degree-of-freedom manipulator and dynamic obstacles demonstrate that the proposed method can rapidly and stably evaluate connectivity. The proposed method enables early identification of path-wise unreachable target configurations based on connectivity in configuration space.**

**Keywords:**

**Growing Neural Gas; Voxel Look-Up Table; Reachability Analysis; Topological Connectivity; Dynamic Environments**

## 1. Introduction

Recently, various types of human-like robots that coexist with people have been developed due to the rapid innovations in machine learning technology in artificial intelligence (AI) [1]. For example, physical AI [2, 3, 4] has been proposed to provide daily human support by robotic manipulation. However, hallucination is one of the most crucial problems in physical AI. Therefore, we need a real-time trajectory evaluation system for robotic manipulation in dynamically changing environments.

In robotic manipulation operating in human-coexistence environments, the ability to adapt to dynamically changing workspaces in real-time is essential [5].

For safe trajectory planning, it is crucial to determine early whether a goal posture is reachable from the current posture.

Such early identification helps avoid unnecessary path searches, which can result in delayed responses in time-critical tasks.

Reachability refers to whether a manipulator can move from its current posture to a goal posture without colliding with obstacles. In general, it is difficult for physical AI to confirm the reachability of robotic manipulation in advance, because physical AI is typically used as a black box in end-to-end implementations. Therefore, we need theoretical and mathematical evaluations to evaluate reachability [6]. Reachability is formulated as a connectivity problem on the configuration space (C-space), which represents the set of all possible robot joint angles. Figure 1 illustrates the mapping between the task space and the C-space, along with the corresponding obstacle regions.

Evaluating this connectivity effectively requires an understanding of the free-space structure within the C-space. Traditionally, this involves sampling numerous postures and performing geometric collision checks for each, a process that is computationally expensive for high-degree-of-freedom manipulators. Furthermore, in dynamic environments, the connectivity of the C-space itself fluctuates as obstacles move, making real-time evaluation even more challenging.

Against this background, conventional sampling-based methods struggle to distinguish whether a failed path search originates from structural non-connectivity or simply insufficient sampling. Consequently, resources may be wasted continuing searches toward goal postures that are fundamentally unreachable, causing significant system delays.

To address these issues, this study proposes a method to efficiently update the C-space topology based on obstacle occupancy information and identify reachable regions in real-time through connected component analysis on the graph struc-

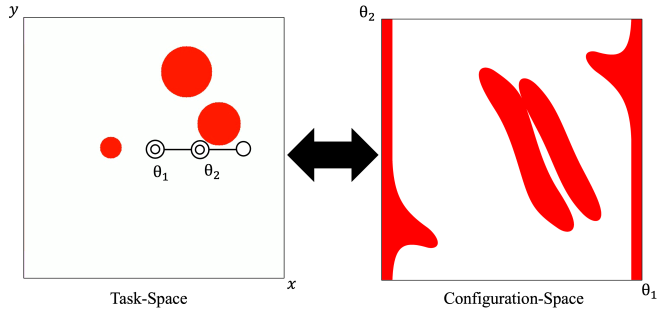



**FIGURE 1.** Correspondence between obstacles in task space and configuration space.

ture. This enables the immediate detection of areas that are structurally unreachable, hereafter referred to as practical non-connectivity, allowing for direct reachability assessment without relying on search failures.

The effectiveness of the proposed method is verified through computational cost evaluations and comparative experiments with sampling-based planners. The results demonstrate that the proposed method can identify practical non-connectivity rapidly and distinguish structurally unreachable regions with high efficiency.

## 2. Related Work

Sampling-based planners such as Probabilistic Roadmaps (PRM) [7, 8] and Rapidly-exploring Random Trees (RRT) [9] are widely used for manipulator path generation, and structured intelligence approaches have also been investigated for path planning and learning in redundant manipulators [10]. PRM represents the C-space as a graph, theoretically enabling reachability assessment based on connectivity. However, in dynamic environments, the need for large-scale graph rebuilding makes real-time application difficult .

RRT-based algorithms [9, 11, 12] can find paths quickly for single queries, but their reachability assessment depends on the search results. As noted earlier, it is difficult to distinguish whether a search failure is due to structural non-connectivity or a lack of search time.

Approaches that explicitly represent C-space structure include topology learning methods such as Growing Neural Gas (GNG) [13, 14, 15]. While these can approximate input space distributions as graphs, the framework for applying them to robot C-spaces for reachability assessment is not fully established. Most studies assume static environments and do not consider updating structures while evaluating connectivity under dynamic obstacle changes.

Thus, while search-based reachability evaluation and structure-based space understanding have been investigated separately, a framework that updates the C-space topology and evaluates connectivity in real-time has not been established. This study integrates high-speed geometric indexing via Voxel Look-Up Tables (VLUT) with the topology maintenance of GNG to achieve early detection of unreachable states due to structural non-connectivity.

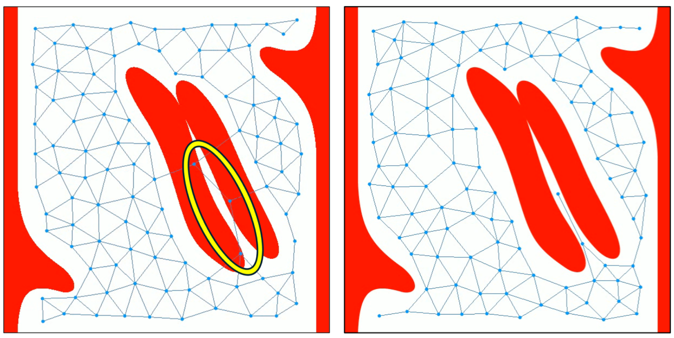
## 3. Proposed Method

This section details the algorithm for real-time C-space topology updates in dynamic environments. The system consists of: (1) Pre-learning the C-space structure using GNG, (2)Constructing a VLUT for high-speed projection of task-space changes into C-space, and (3) Identifying connectivity through connected component analysis.

### 3.1. Growing Neural Gas with Collision Awareness

GNG [13] represents the input domain as a graph consisting of nodes $\mathcal{V}$ and edges $\mathcal{E}$. Each node $i \in \mathcal{V}$ is associated with a reference vector (posture) $w_i \in \mathbb{R}^n$. While GNG can efficiently represent C-space topology with a small number of nodes, standard GNG updates based purely on Euclidean distance may create edges between non-connected regions, ignoring physical constraints and causing topological discontinuities.

We integrate a collision-detection mechanism based on robot geometry into the GNG learning process to ensure transition feasibility between postures. Specifically, we introduce two components:



(a) Standard (b) Proposed

**FIGURE 2.** GNG structure comparison. (a) Standard method with shortcuts, (b) Proposed collision-aware method.

- Collision checking between the input point and the neighbor nodes during winner selection.
- Connectivity verification during edge generation and updates.

The learning process is as follows. Each input $u \in \mathbb{R}^n$ is sampled randomly from the robot's joint range.

---

**Collision-Aware GNG**

---

**Step 0:** Initialize: $\mathcal{V} = \{w_1, w_2\}$, edge $a_{1,2} = 1$.

**Step 1:** For $u$, identify safe neighbor nodes $\mathcal{V}_{safe} \subseteq \mathcal{V}$. Select winners $s_1, s_2$: $s_1 = \arg\min_{i \in \mathcal{V}_{safe}} \|u - w_i\|$, $s_2 = \arg\min_{i \in \mathcal{V}_{safe} \setminus \{s_1\}} \|u - w_i\|$

**Step 2:** Accumulate error: $E_{s_1} \leftarrow E_{s_1} + \|u - w_{s_1}\|^2$

**Step 3:** Update reference vectors: $w_{s_1} \leftarrow w_{s_1} + \epsilon_b(u - w_{s_1})$ $w_n \leftarrow w_n + \epsilon_n(u - w_n)$

**Step 4:** If no collision, create edge $(s_1, s_2)$.

**Step 5:** Update edge age: $a_{s_1,n} \leftarrow a_{s_1,n} + 1$

**Step 6:** Remove old edges and isolated nodes.

**Step 7:** Every $\lambda$ iterations, insert node $r$: $w_r = 0.5(w_v + w_f)$ $E_v \leftarrow (1-\alpha)E_v$, $E_f \leftarrow (1-\alpha)E_f$, $E_r \leftarrow E_v$

**Step 8:** Decay errors for all nodes: $E_i \leftarrow (1-\beta)E_i$

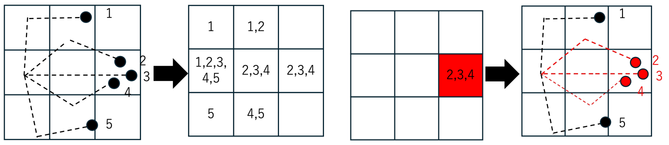
**Step 9:** Repeat from Step 1.

---

### 3.2. Scalable Voxel Look-Up Table (VLUT)

To map task-space geometry directly to the discretized configuration space, we construct a Voxel Look-Up Table (VLUT) offline [16]. The workspace is divided into a 3D voxel grid where each voxel $v$ is mapped to a subset of GNG nodes $M(v)$. During the offline construction phase, the robot posture $w_i$ for each node $i \in \mathcal{V}$ is evaluated using forward kinematics to identify the voxels occupied by the robot body, and $i$ is appended to $M(v)$.

During the online phase, the system identifies colliding nodes by checking the occupancy status of the corresponding voxels through the VLUT. This lookup-based approach bypasses expensive geometric collision checks, enabling high-speed state updates even in complex dynamic environments. Figure 3 shows the conceptual diagram of the offline construction and online update process.




(a) Offline (b) Online

**FIGURE 3.** Concept of VLUT operation.

### 3.3. Connectivity Analysis for Topological Identification

When a node's occupancy status is updated due to obstacle proximity, the graph may become partitioned. We apply a standard connected component analysis based on Breadth-First Search (BFS) to identify the set of nodes $\mathcal{V}_{mainland}$ that have physical continuity from the current posture $n_{start}$ via non-colliding edges in real-time. Isolated node groups not belonging to the Mainland are identified as islands and serve as the basis for judging unreachability.

Figure 4 illustrates this dynamic partitioning process in a simplified 2-DOF planar configuration space over three consecutive time steps (top to bottom). In the left column, the red obstacle moves dynamically in the workspace. The middle column shows the GNG node states in the end-effector space, while the right column visualizes the corresponding C-space topology ($\theta_1$-$\theta_2$). As the obstacle moves, the projected region of colliding nodes (red) and near-collision nodes (yellow) updates incrementally in the C-space. This dynamic motion creates a time-varying blocking barrier that partitions the GNG graph. Depending on the current robot posture, this partition divides the reachable free space (Mainland, green nodes) from the isolated regions (Islands). Since the GNG edge connections intersecting the colliding voxels are deleted in real-time, the BFS traversal dynamically updates the reachable set, allowing the system to immediately detect if a goal posture has become topologically unreachable.

## 4. Experimental Results

To verify the effectiveness of the proposed method, we constructed a simulation environment using a 6-DOF manipulator and a dynamic obstacle arm.

### 4.1. Experimental Setup

For the target 6-DOF manipulator, 10,000 discrete nodes were distributed across the C-space. In the workspace, an-

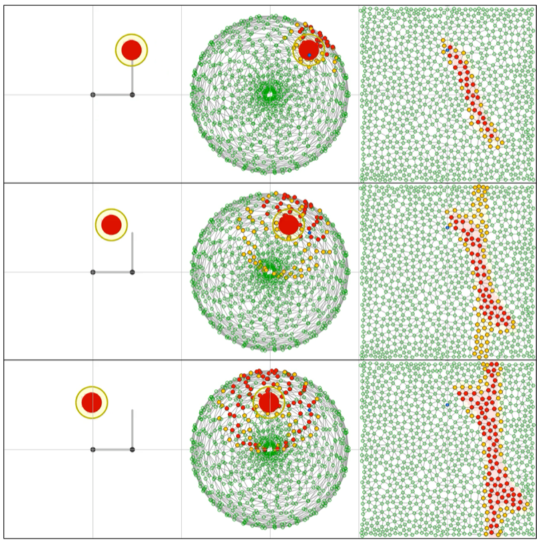


**FIGURE 4.** Conceptual illustration of dynamic configuration space partitioning over three consecutive time steps (top to bottom). The left column shows the task space (red: collision, yellow: danger, green: safe) of a 2-DOF planar manipulator. The center and right columns show the corresponding node states in the end-effector mapping and $C$-space connectivity ($\theta_1$-$\theta_2$), where GNG nodes are partitioned into Mainland and Island components by the blocking barrier of colliding nodes (red).

other 6-DOF manipulator with the same geometric configuration was placed as a dynamic obstacle at base coordinate $\mathbf{p}_{obstacle} = [100, 100, 0]^T$ mm, with the target robot base at $\mathbf{p}_{target} = [0, 0, 0]^T$ mm. The obstacle arm performs periodic waving motions across the target's workspace, dynamically altering the C-space topology. GNG parameters are shown in Table 1, and voxel resolution $\delta$ was set to 20 mm.

**TABLE 1.** GNG Parameter Settings

| Parameter | Symbol | Value |
|---|---|---|
| Total learning iterations | $I_{total}$ | 1,000,000 |
| Maximum node number | $N_{max}$ | 10,000 |
| Maximum edge age | $a_{max}$ | 100 |
| Node insertion period | $\lambda$ | 50 |
| Learning rate (winner) | $\epsilon_b$ | 0.05 |
| Learning rate (neighbor) | $\epsilon_n$ | 0.005 |
| Error decay rate 1 | $\alpha$ | 0.5 |
| Error decay rate 2 | $\beta$ | 0.005 |

The obstacle arm was driven by smooth sinusoidal inputs $q_i(t) = A\sin(\omega t + i\phi)$ with angular velocity $\omega = 1.5$ rad/s and amplitude $A = 1.0$ rad. For the voxel occupancy changes occurring sequentially at approximately 0.1 s intervals, we performed sequential topology analysis using VLUT-based node updates and BFS. The computing environment used for the experiments is shown in Table 2.

**TABLE 2.** Computing environment for experiments

| | |
|---|---|
| Device | MacBook Air |
| CPU | Apple M3 |
| Memory | 16 GB |

## 4.2. Performance of Topological Updates

We evaluated the processing performance of topological updates for the 10,000-node graph. Observations showed that node state updates via VLUT took an average of less than 0.01 ms, confirming that high-speed updates are possible regardless of geometric complexity (Table 3). Subsequent connectivity analysis via BFS took approximately 0.4–0.5 ms, and the total computation time including topology analysis remained stable under 0.6 ms. Comparing with the standard control cycle of robot manipulators (100 Hz, 10 ms), the proposed method maintains an extensive real-time margin.

**TABLE 3.** Computational Cost of Topology Updates ($N = 10,000$)

| Processing Item | Avg. Time [ms] |
|---|---|
| Node State Update (VLUT Core) | $< 0.01$ |
| Connectivity Analysis (BFS) | 0.4–0.5 |
| **Total (Topology Analysis)** | $< 0.6$ |

## 4.3. Reachability Verification via RRT-Connect Comparison

To verify that detected island nodes are indeed structurally unreachable, we conducted reachability tests using RRT-Connect [11]. We extracted $N = 1,000$ island nodes at random during the obstacle's motion. Path searches were attempted with two timeout limits: 10 ms, representing the real-time responsiveness required within a single 100 Hz control cycle, and 50 ms, representing a condition with increased computational resources. The RRT-Connect step size was set to 0.05 rad.

## 4.4. Analysis of Results

Results are shown in Table 4. The low success rates (0.8% and 3.1% under 10 ms and 50 ms timeouts, respectively) confirm that the detected island nodes represent regions of practical non-connectivity. This confirms that the proposed method can effectively identify regions that are virtually unreachable within standard planning cycles, enabling immediate and reliable judgement of unreachability without relying on exhaustive search failures.

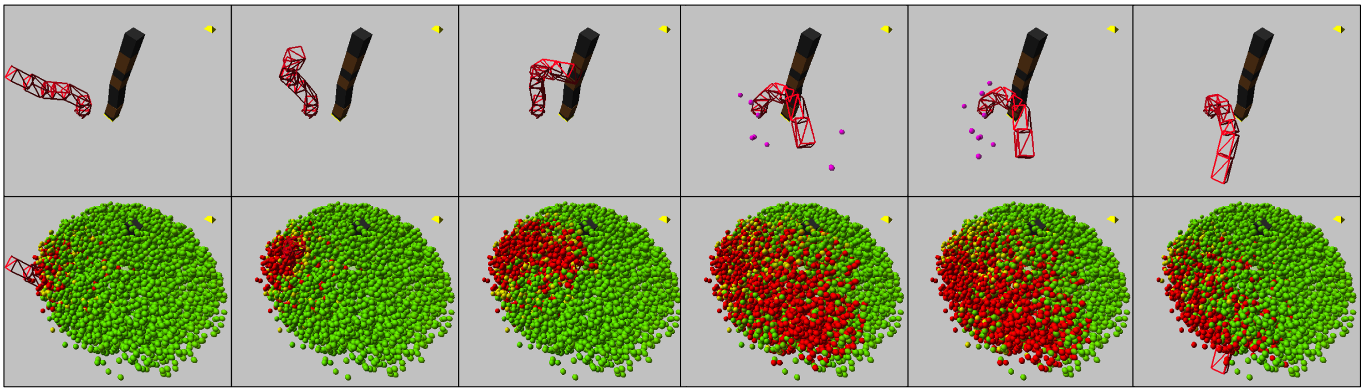


**FIGURE 5.** Temporal transition of configuration space topology and robot arm behavior during dynamic obstacle avoidance. The top row shows reactive motions and detected Island nodes (Magenta spheres), while the bottom row visualizes the corresponding connectivity changes in the GNG graph (free space is Green, occupied is Red).

**TABLE 4.** RRT-Connect Reachability to Island Nodes ($1,000$ trials)

| Planner Configuration | Success Rate [%] |
|---|---|
| RRT-Connect (10 ms timeout) | 0.8 |
| RRT-Connect (50 ms timeout) | 3.1 |

## 5. Discussion

The voxel resolution $\delta$ involves a trade-off between geometric accuracy and computational efficiency. While a smaller resolution increases the fidelity of the collision model, it also increases memory consumption and processing time. In this study, $\delta = 20$ mm was selected based on the physical scale of the manipulator, specifically chosen to be smaller than the robot's link thickness. This configuration allows the discretized representation to capture the essential robot geometry while maintaining an extensive real-time performance margin.

Regarding scalability, extending the framework to higher-dimensional systems, such as 7-DOF redundant arms or dual-arm manipulators, presents a trade-off between discretization resolution and topological connectivity. While increased dimensionality naturally leads to a lower node density for a fixed number of nodes, it is not yet clear whether this drop in resolution fundamentally hinders coarse reachability assessment. In fact, increased redundancy may provide more routing alternatives in the C-space, potentially compensating for lower resolution and allowing for stable connectivity analysis. Since the online VLUT update cost scales with the number of changed task-space voxels rather than C-space dimensionality, the primary practical bottleneck remains the BFS-based connectivity analysis. Further verification is required to determine the minimum required resolution for effective topological identification in such high-DOF systems.

Finally, the primary advantage of our method is the early detection of structural non-connectivity. Unlike conventional planners that search until a timeout is reached, the proposed method identifies Mainland and Island regions in less than 1 ms. This allows high-level task planners to immediately abort futile motions and switch to alternative safe tasks, significantly enhancing responsiveness in dynamic human-robot interaction.

## 6. Conclusions

In this paper, we proposed a method for real-time detection of topological changes by integrating GNG discretized structures of large-scale C-spaces with high-speed VLUT geometric indexing. Experimental results for a 10,000-node system demonstrated that separating Mainland and Island regions is possible in less than 1 ms. Furthermore, reachability verification using RRT-Connect demonstrated that detected island nodes effectively represent regions of practical non-connectivity, with over 96.9% of these nodes remaining unreachable within practical computation times. This validates the proposed method's ability to immediately identify regions that are virtually unreachable within standard control cycles.

The proposed method enables real-time identification of structurally unreachable regions in dynamic environments. This allows for avoiding futile searches toward unreachable goal postures, efficient use of computational resources, and stabilization of calculation times for reachability assessment. While traditional methods focus on local safety judgements based on instantaneous collision status, our approach provides an explicit grasp of reachable free-space regions continuously from the current posture. By providing this rigorous mathematical safety layer, our method can complement physical AI to effectively reduce hallucinations, ensuring the topological feasibility of planned trajectories. This contribution is expected to enhance safe and stable robot manipulation in human-involved environments. Future work includes verification with real-world manipulators, extending the framework to predictive risk evaluation, and developing a hybrid planning architecture that integrates physical AI with real-time configuration space topology analysis.

## Acknowledgements

This work was supported by the Japan Science and Technology Agency (JST) under the Moonshot Research and Development Program (Grant Number JPMJMS2034). The authors would like to express their sincere gratitude for this support.

## References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[2] Aslan Miriyev and Mirko Kovaç. Skills for physical artificial intelligence. *Nature Machine Intelligence*, 2(11):658–660, 2020.

[3] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, et al. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] Naoyuki Kubota. Learning and adaptation for human-friendly robot partners in informationally structured space. In *Advances in Autonomous Mini Robots*, pages 11–26. Springer, 2012.

[6] Patrick Holmes, Mark Leather, Carmelo Sferrazza, Alex Leake, Masoud Mazaheri, David Paster, and Ram Vasudevan. Reachable sets for safe, real-time manipulator trajectory design. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[7] L. E. Kavraki, P. Svestka, J-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[8] R. Bohlin and L. E. Kavraki. Path planning using lazy prm. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 521–528, 2000.

[9] Steven M. LaValle and James J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Dimensions*, pages 303–307. A. K. Peters, 2001.

[10] Naoyuki Kubota, Takemasa Arakawa, and Toshio Fukuda. Trajectory planning and learning of a redundant manipulator with structured intelligence. *Journal of the Brazilian Computer Society*, 4(3):14–26, 1998.

[11] James J. Kuffner and Steven M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, pages 995–1001, 2000.

[12] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, 2011.

[13] Bernd Fritzke. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, 1995.

[14] Y. Toda, T. Matsuno, and T. Minato. Growing neural gas based 3d space object tracking for autonomous mobile robot. In *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 1–6, 2024.

[15] Naoki Doteguchi and Naoyuki Kubota. Topological tracking for mobility support robots based on multi-scale batch-learning growing neural gas. In *Proceedings of the 10th International Conference on Mobile Wireless Middleware, Operating Systems, and Applications (MOBILWARE 2021)*, EAI/Springer Innovations in Communication and Computing, pages 17–31. Springer, 2022.

[16] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.