Coverage Path Planning for Vacuum Cleaner Robot in ROS Utilizing Turtlebot3

1st Tofigh HOSSEINI
Department of Mechatronics
Erciyes University

***Abstract*—This project presents the development and simulation of an autonomous cleaning robot, integrated within the Robot Operating System (ROS) framework and visualized through the Gazebo simulation environment. Leveraging the capabilities of Turtlebot3, known for its refined Adaptive Monte-Carlo Localization (AMCL) and accurate odometry, the project delineates a two-phase operational paradigm for the robot. Initially, the robot is tasked with autonomously exploring and mapping a defined closed environment to ascertain its layout. Subsequently, it transitions to its primary objective—cleaning. To ensure comprehensive coverage of the area while navigating around obstacles, the robot employs a sophisticated Coverage Path Planning (CPP) algorithm. This study innovates by implementing a grid decomposition approach to CPP, augmented with a dynamic tracking mechanism [1], enabling efficient and thorough cleaning of the environment. This approach not only enhances the robot's operational efficiency but also significantly contributes to the field of autonomous cleaning systems by addressing the complexities of effective area coverage and obstacle avoidance in dynamic environments.**

***Index Terms*—Coverage Path Planner, ROS, Cleaner Robot, Turtlebot3**

I. Introduction

The adoption of robotic solutions for domestic chores has seen a significant upsurge, revolutionizing the way household tasks are approached. Cleaner robots, in particular, have become increasingly prevalent in modern homes, offering an efficient and autonomous means of maintaining cleanliness across various living spaces. This trend extends beyond just vacuuming, with robotics now making inroads into other areas of home maintenance, including lawn mowing and window cleaning. The advent of such technologies signifies a shift towards smarter, more automated homes, where the convenience and efficiency of robotic assistants are leveraged to handle routine but time-consuming tasks, allowing residents to reclaim valuable time and effort.

To accomplish the objective of automated vacuuming using robots, it is imperative to devise a path that effectively covers the entirety of the home environment. Typically, this task is delegated to algorithms specifically designed for generating such paths, known as Coverage Path Planning (CPP) algorithms.

The specifications for an optimal and acceptable coverage path must include the following criteria [2]:

1) The robot must pass every point in the target region, totally covering it.
2) Robot must complete the area without crossing any of the paths twice.
3) Operations must be continuous and sequential, without any route repetition.
4) Robot must avoid obstacles.
5) Under the circumstances, an ideal route is desirable.

II. ROS and SLAM

One of the widely utilized applications within the ROS ecosystem is Simultaneous Localization and Mapping (SLAM). In mobile robotics, SLAM's primary aim is to generate and continually update a map of an unexplored environment using embedded sensors, such as 2D Lidar. The resulting map, established through 2D laser SLAM, manifests as a two-dimensional grid map composed of individual grid cells. These grid cells can hold distinct values denoting various interpretations, as depicted in Figure 3. White signifies a free and traversable region, designated with a stored value of 0. Conversely, black signifies an obstructed and impassable area, assigned a stored value of 100. Gray denotes an area of uncertainty, where the passability of the grid is indeterminate, allocated a stored value of -1. Within the ROS framework, this grid map is commonly referred to as an occupancy map, serving to delineate the occupied and unoccupied regions within a given environment.

III. Coverage Path Planning

In this section, we delve into the pivotal aspect of coverage path planning (CPP), showcasing the methodology adopted in this study: Grid-based Decomposition. This approach ingeniously breaks down the map into discrete grids, each serving as a fundamental unit of analysis. Within this framework, grids are categorized based on their occupancy status, distinguishing between areas that are either occupied or unoccupied. Additionally, a crucial component of this method involves the assignment of weight values to each grid. The process initiates with the selection of a designated start grid, meticulously chosen to anchor the planning process. Initially bestowed with a weight of 0, the start grid serves as the reference point from which subsequent grid cells are evaluated. Moving outward from the start grid, adjacent cells progressively receive weight values incremented by one, effectively quantifying their spatial relationship to the origin. This iterative procedure extends across the entire map until all unoccupied grids are appro-

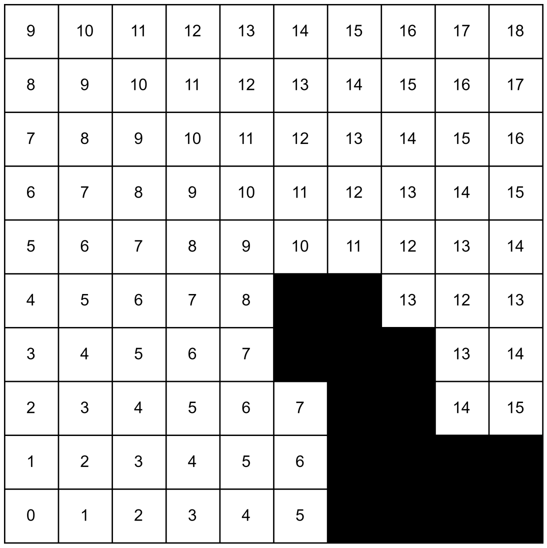

Fig. 1: A Grid map with weights

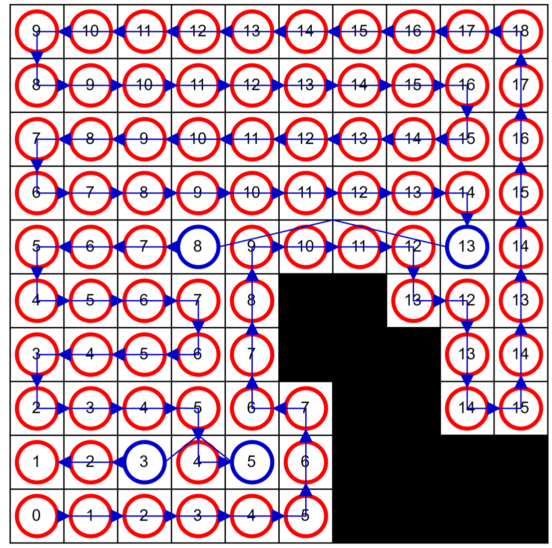

Fig. 2: Grid Decomposition Coverage Path

priately weighted, laying the groundwork for comprehensive coverage path planning.

Furthermore, the Grid-based Decomposition technique offers a systematic approach to path planning, blending simplicity with efficacy. By segmenting the map into manageable grid units, the complexity of navigation is reduced, facilitating efficient decision-making processes for the robot. The assignment of weight values to each grid serves a dual purpose: not only does it provide a spatial hierarchy, guiding the robot towards unexplored territories, but it also enables the creation of a nuanced path that optimizes coverage. This methodological innovation underscores the interdisciplinary nature of robotics, leveraging principles from computational geometry and graph theory to address real-world challenges. In essence, Grid-based Decomposition embodies a paradigm shift in coverage path planning, epitomizing the fusion of theoretical concepts with practical application within the realm of autonomous robotics.

Following the completion of weighting, the algorithm initiates the path planning process by commencing at the start grid, characterized by its weight of 0. From here, the algorithm evaluates the adjacent grids, identifying the grid with the highest weight, and designates it as the subsequent destination. This sequential approach ensures that the robot navigates towards areas of greater significance, prioritizing regions that contribute most substantially to comprehensive coverage. Subsequently, the algorithm iterates this procedure for each successive grid, dynamically adapting its trajectory based on the evolving weight distribution across the map.

In instances where the algorithm encounters a grid surrounded by previously visited cells, thus limiting adjacent options, it seamlessly transitions to an alternative strategy. In such scenarios, the algorithm strategically selects the closest unvisited grid as the next target, thereby ensuring continuous progression through unexplored regions. This adaptive mechanism underscores the algorithm's resilience in navigating complex environments, effectively mitigating potential navigational challenges while maintaining a steadfast focus on achieving optimal coverage. Through this iterative and adaptive approach, the coverage path planning algorithm orchestrates the robot's traversal with precision and efficiency, culminating in thorough and systematic coverage of the designated environment.

A pseudocode representation of the grid decomposition coverage path planning algorithm is presented at Algorithm 1.

IV. Simulations

In the simulation section, I demonstrate the functionality of the vacuum cleaner robot through two distinct closed environments, each tailored to highlight specific aspects of its performance. The diversity of environments aims to provide a comprehensive evaluation of the robot's capabilities, showcasing its adaptability across varied scenarios. You can examine these enclosed settings, which replicate indoor household environments, depicted in Figures 3 and 4.

The utilization of the ROS explore_lite [3] package significantly streamlined the mapping process, enabling the robot to efficiently navigate through unexplored areas while effectively capturing the environment's layout. This approach not only enhances the robot's autonomy but also contributes to the overall effectiveness of the cleaning operation by ensuring thorough coverage of the designated space. The results of this simulation and cleaning can be seen in Figures 5 and 6 where the red lines showcase the areas the robot has cleaned.

Algorithm 1: Grid Decomposition Coverage Path

```
Data: map (occupancy map), footprint (robot base
      radius), height, width
Result: coverage_path
begin
    map ← Adjust resolution to match the robot's
      footprint;
    Assign a weight to every grid within the map;
    current_grid ← start_grid;
    coverage_path ← (start_grid);
    while all grids are not visited do
        adjacent_grids ← current_grid adjacent girds;
        while adjacent_grids is not empty do
            next_grid ← heightest weight grid in
              adjacent_grids;
            if next_grid not visited then
                coverage_path ← (coverage_path,
                  next_grid);
                current_grid ← next_grid;
                break;
            else if next_grid visited then
                remove next_grid from adjacent_grids
                  list;
            if adjacent_grids is empty then
                next_grid ← find the closes unvisited
                  grid;
                coverage_path ← (coverage_path,
                  next_grid);
                current_grid ← next_grid;
    Return coverage_path;
```

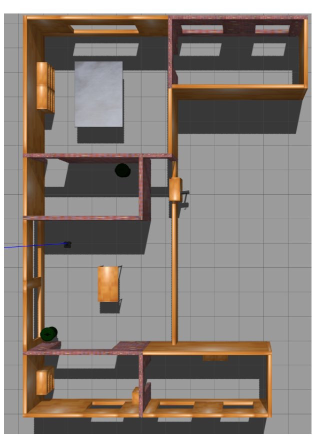

Fig. 3: Simulation 1 environment

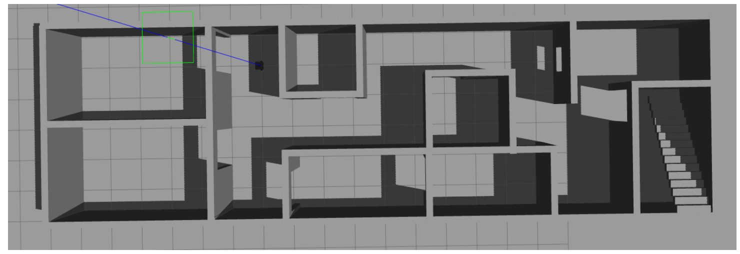

Fig. 4: Simulation 2 environment

Fig. 5: Simulation 1 results

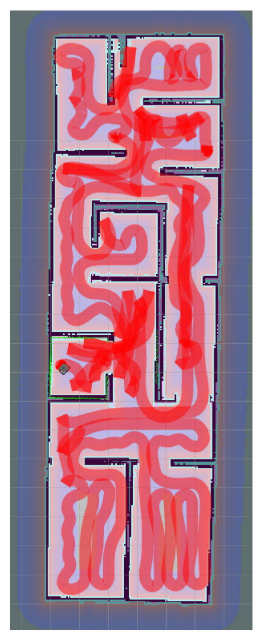

Fig. 6: Simulation 2 results

REFERENCES

[1] Zhao, Shengmin, and Seung-Hoon Hwang. ”Complete coverage path planning scheme for autonomous navigation ROS-based robots.” ICT Express 10.1 (2024): 83-89.

[2] Cao, Zuo Llang, Yuyu Huang, and Ernest L. Hall. ”Region filling operations with random obstacle avoidance for mobile robots.” Journal of Robotic systems 5.2 (1988): 87-102.

[3] Explore_lite - ROS wiki, 2023, Available online: http://wiki.ros.org/explore_lite/. (Accessed 28 January 2023).