Smart Cleaning Robot Proposal

A ROS-Based Autonomous Cleaning Robot with Indoor Mapping, Coverage Path Planning, Voice Control, and Expandable Functionality

Pang Liu, Zhenxu Chen

1. Introduction

This project aims to develop a smart autonomous cleaning robot. The robot can quickly explore the environment and record the global map, planning efficient cleaning routes to ensure full coverage purposes, recognize voice commands and execute accordingly, and detect moving obstacles to avoid when cleaning.

The smart mapping system will demonstrate autonomous indoor mapping using the frontier-based exploration (FBE) algorithm and simultaneous localization and mapping (SLAM) to explore and remember all possible rooms and spaces efficiently. The FBE algorithm ensures that the map is constantly updated and can avoid changing obstacles such as pedestrians.

The smart cleaning system custom complete coverage path planning (CCPP) to ensure the robot can cover all room areas.

The intelligent interactive system can recognize voice commands and change the behavior, such as stop cleaning and reset back to the base, continue cleaning, or rebuild the map.

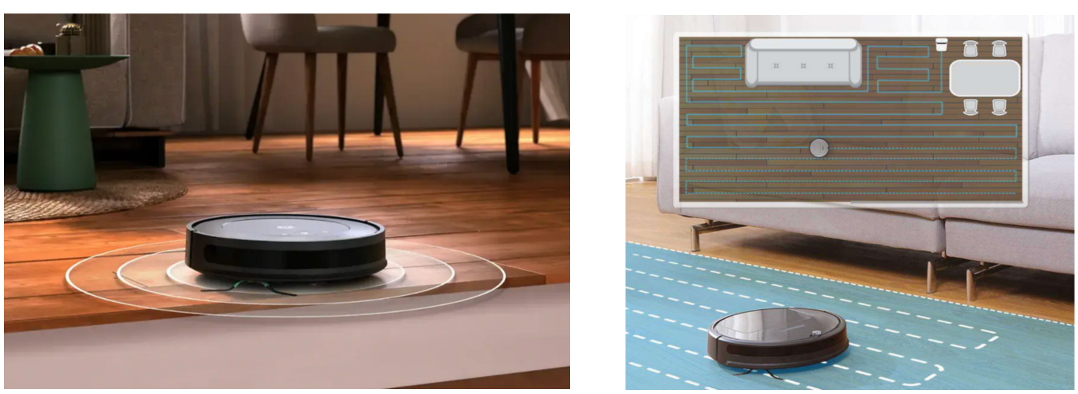

Figure 1. Rendering prototype of the Cleaning Robot

2. Background and Project Objectives

With the rise of smart home technology, users now seek cleaning robots with greater autonomy and interactive capabilities beyond basic cleaning. Traditional robots struggle in dynamic environments and lack precise coverage. This project aims to develop a ROS-based cleaning robot that integrates autonomous exploration, complete coverage path planning, and voice control to provide an intelligent and efficient cleaning experience.

The objectives of this project include:

- Enable autonomous indoor exploration and mapping: Use SLAM and Frontier-Based Exploration (FBE) algorithms to allow the robot to autonomously create a 2D map of an unknown environment.
- Design complete coverage path planning (CCPP): Ensure the robot can cover all accessible areas without missing or re-cleaning spaces.
- Integrate real-time obstacle avoidance: Detect and avoid dynamic obstacles, such as pedestrians, to ensure smooth and safe cleaning operations.
- Build an intelligent interaction system: Adjust cleaning strategies based on user commands, such as stopping and returning to the base, continuing cleaning, or remapping the environment.
- Establish an expandable system architecture: Reserve interfaces for future module expansions (e.g., mobile app control, smart feedback) to make the system scalable.

3. Project Scope and Methodology

(1) Smart Mapping System

The Smart Mapping System enables autonomous indoor mapping using the Frontier-Based Exploration (FBE) algorithm. This algorithm allows the robot to autonomously explore unknown environments and efficiently build a 2D map.

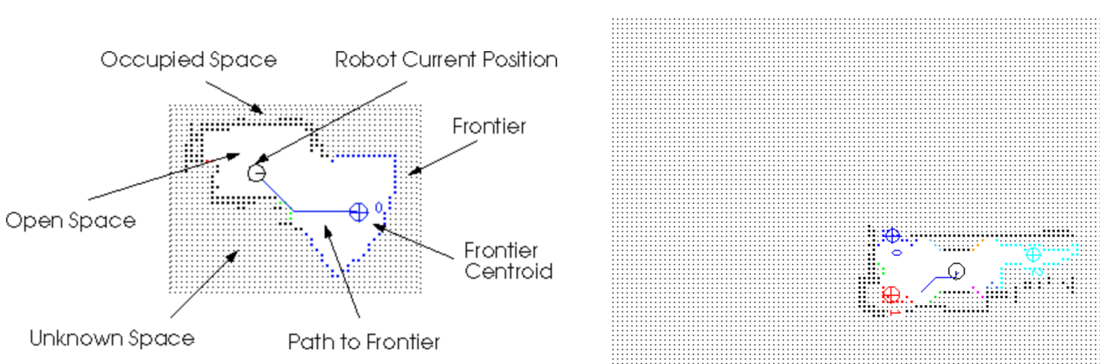

Figure 3-1a. Demo of Frontier Based Explore (a)

We chose the FBE algorithm because it is very efficient and can generate maps rapidly. Also, its computational cost is not large and adapts well to dynamic changes in the environment. So, the FBE algorithm is ideal for cleaning robots, which require high coverage and quick exploration, aligning perfectly with this project's needs.

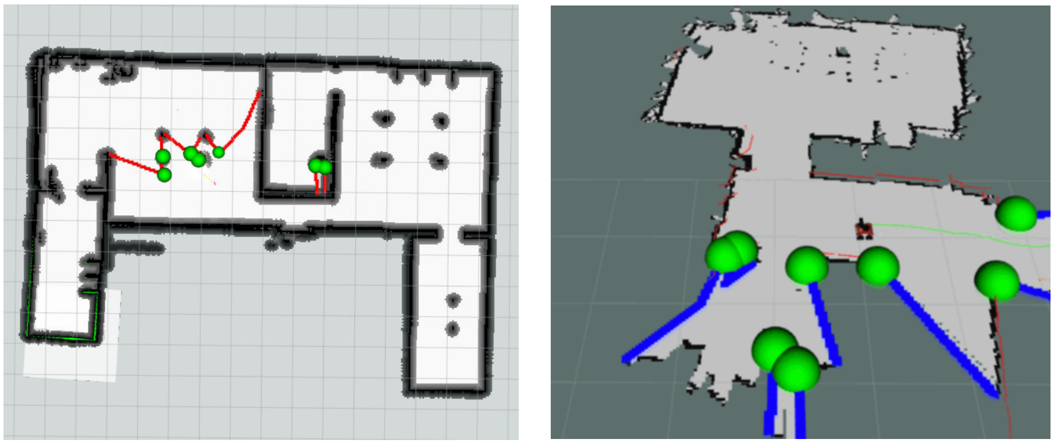

Figure 3-1b. Demo of Frontier Based Explore (b)

(2) Smart Cleaning System

Once the map is generated, the robot will use the CCPP (Complete Coverage Path Planning) algorithm to ensure it covers all accessible areas for cleaning. In detail, we use the Boustrophedon Decomposition algorithm to cover every area of the room.

We chose this algorithm because it divides the space into smaller subregions, allowing the robot to follow a "back-and-forth" path within each section. It can effectively reduce redundant paths, provides high coverage, and makes it ideal for household cleaning robots.

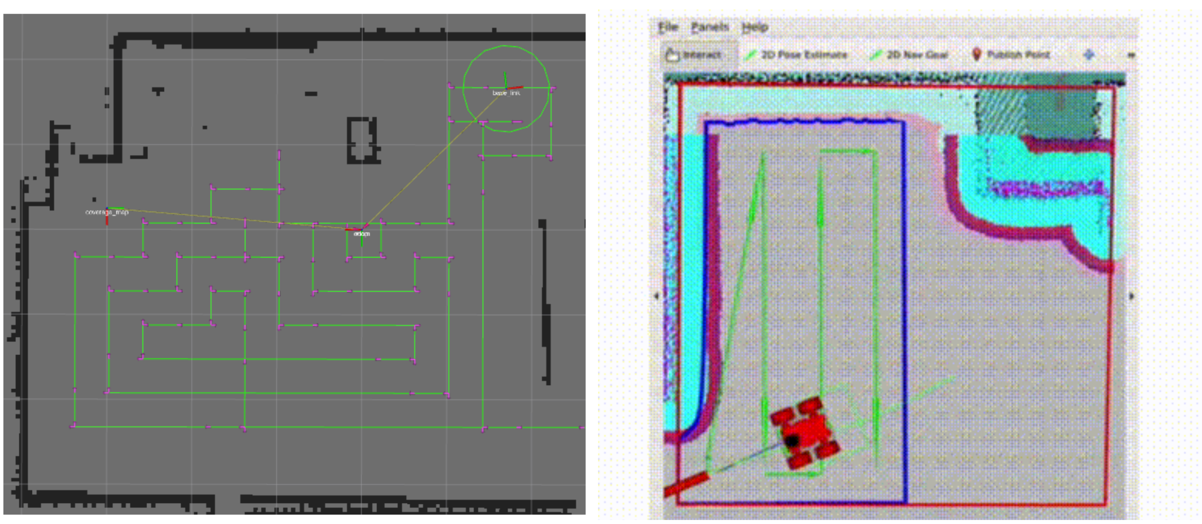

Figure 3-2. Demo of CCPP

(3) Intelligent Interactive System

In the intelligent interaction system, we added an offline voice control module to the robot and planned to use the PocketSphinx module for keyword recognition. Through offline processing, the robot can recognize common commands (such as "start", "stop" or "return") and perform corresponding cleaning actions. Offline keyword recognition not only ensures privacy and stability but also enables the robot to flexibly respond to users' voice commands in an offline environment, improving the user experience.

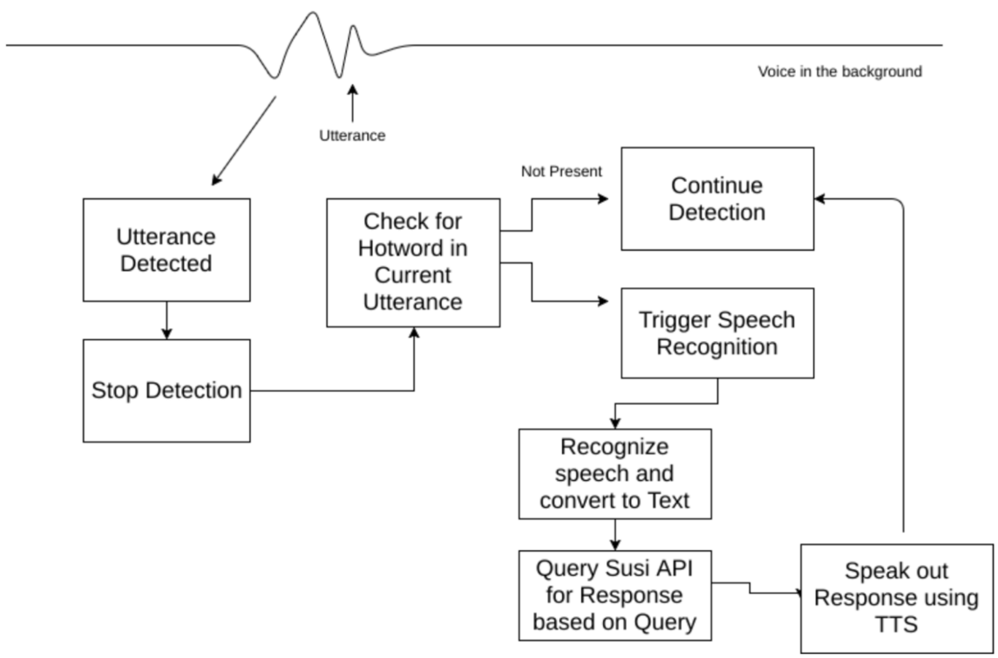

Figure 3-3. Demo of recognizing keywords in the background

4. Technology Implementation

(1) Software System

We use Linux, ROS, and Python language to implement all required codes and algorithms. Also, we use Gazebo Simulation to implement tests in a virtual environment.

(2) Hardware System

We use TurtleBot3 to implement all needed sensors, including:

- LiDAR: For distance sensing and mapping.
- Wheel encoders: For odometry.
- IMU: For better localization and motion tracking with Kalman Filter.

5. Running Stages

(1) Mapping Mode

The robot is placed in an unfamiliar indoor environment (e.g., a room with walls, furniture, and obstacles). Upon starting, it uses the open-source explore_lite package to autonomously map the entire room. The robot will move around, avoiding obstacles, scanning the environment, and creating a 2D map. Once the mapping process is complete, the robot will send a "mapping complete" signal and stop moving.

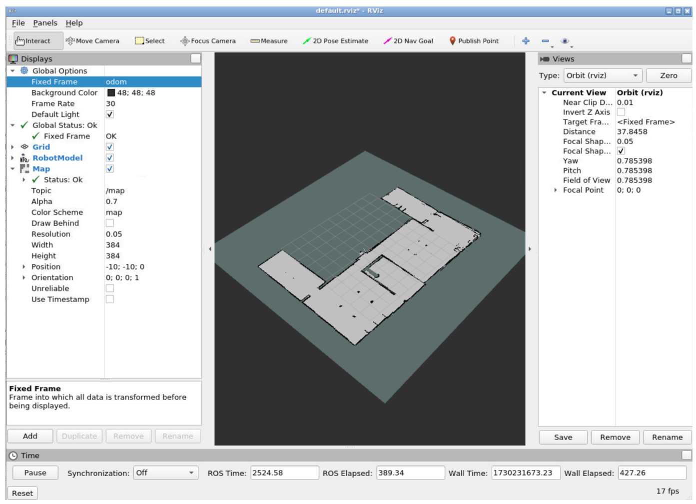

Figure 5-1. Demo of Mapping Complete result

(2) Coverage Path Generation

After the map is completed, the system will automatically transition to the path-planning phase which might take some time using the map generated in stage 1. The robot will generate a complete coverage path to ensure it can reach all areas within the mapped environment. A visual representation of the path will be shown, demonstrating how the robot will navigate the room to clean all reachable spaces.

(3) Autonomous Navigation

If the robot is placed in a random location within the room, a “start cleaning” command is sent. The robot will begin navigating the room, following the previously generated path, ensuring that all areas are covered. The robot will move along the path, avoiding moving obstacles like walking people by using Local Path Planning (e.g. DWA) and efficiently covering the room until the task is complete or a “stop” command is sent.

Also, if the robot recognizes keywords such as “Stop”, then the robot will stop cleaning. If the robot stops over several seconds, the robot will autonomously navigate back to its base. The robot will remember the point before it back to the base.

6. Test Environment

The robot will operate in a typical indoor environment containing walls and furniture (or obstacles). Also, the entire room should be closed, no matter how many small rooms are in it.

Also, a random moving person is included in the environment to test the function of auto-avoid.

Figure 6. Examples of tests in simulated environment and real environment

7. Challenges and Risk Management

The major risks in this project include:

- SLAM Accuracy: The quality of the map generated using explore_lite directly impacts the coverage path. Errors in mapping or localization could lead to incomplete coverage.
- Path Planning Robustness: The complete coverage path planning algorithm needs to be efficient and ensure that all areas are covered without leaving gaps or causing the robot to repeat areas unnecessarily.
- Obstacle Avoidance: Ensuring that the robot can handle moving obstacles and re-plan the path during the cleaning process.
- Voice Detecting accuracy: To make sure the voice can be detected correctly.

8. Optional Extensions

Some modules will be considered for integration if time permits, while others will be long-term objectives. During system development, interfaces or extension points will be reserved to accommodate the potential addition of the following modules.

(1) App Control Module (iOS or Android)

To enhance user interaction with the robot, an Android application or an iOS app might be developed to control and monitor the cleaning process. The app will be built using Android

Studio, and ROSBridge or any other ROS Android libraries will be used to establish communication between the app and TurtleBot3.

The app will include the following features:

1. **Robot Control**: Users will be able to initialize (mapping first), start, stop, or pause the robot's cleaning task using the app. This will be done by sending commands to the robot via Wi-Fi or Bluetooth, depending on the available connection.
2. **Real-Time Monitoring**: The app will display the room map generated by the robot, allowing users to track the robot's progress and current location within the mapped environment. The cleaned area will be shaded, giving users a clear understanding of the areas that have been cleaned.
3. **Interactive Feedback**: The app will provide real-time feedback on the robot's performance, including notifications for task completion, battery status, and any potential issues.
4. **LLM Application:** After reviewing Professor Pito's reflections, an LLM agent could be developed and applied. A distilled or quantization LLM, suitable for mobile applications, could be used to recognize user commands and interact with the interface for controlling the robots.

(2) Room Divided and Select Cleaning

To further enhance the robot's cleaning functionality, the system could incorporate a room division and selective cleaning feature. Once the robot generates a complete map, it will use room segmentation techniques to distinguish between different rooms based on boundaries and layout. Users could then select specific rooms or areas for targeted cleaning via the app or voice command. This feature would make the robot more flexible and efficient by allowing users to prioritize cleaning in high-traffic or specific areas without covering the entire space. Techniques such as image processing on the map and clustering algorithms can be used to implement this room division effectively.

(3) Chatting and Complex Voice Recognition

To enable a more interactive and responsive user experience, the robot could incorporate complex voice recognition and conversational AI capabilities. By integrating with an online AI service, the robot would understand and respond to more complex and varied commands beyond basic keywords. This feature would allow users to engage in natural, multi-step conversations with the robot for more personalized control, such as asking about cleaning status, scheduling tasks, or even selecting specific cleaning modes. Using tools like GPT-based models or similar AI services, the robot can process and generate context-aware responses, enhancing user engagement and interaction.