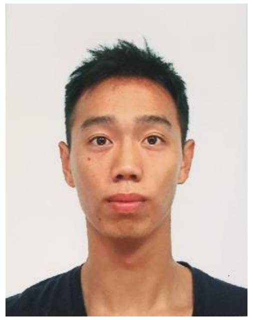
industrial consortium asia pacific

# Autonomous Exploration in ROS

Adriel Ho Jin Liang
Research Engineer



Managed by Advanced Remanufacturing and Technology Centre ARTC

# Background Information: Mapping in Robotics

## What is robotics mapping?

In robotics, mapping is the process of constructing a representation of the environment. A robot would use a range of onboard sensors, such as lidar systems, cameras, proximity sensors, and so on, to do this.

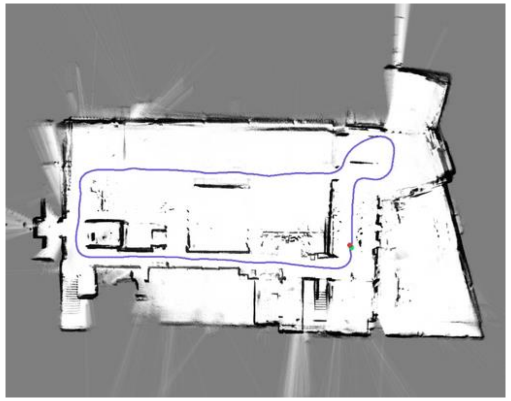


Figure 1: Example of a map for mobile robots [1]

## What is the significance of mapping in robotics?

The mapping procedure is critical to the success of the localization process. The accuracy of the robotic system's perception of the surrounding environment is determined by the precision of the on-board sensors and the reliability of the data post-processing algorithms.

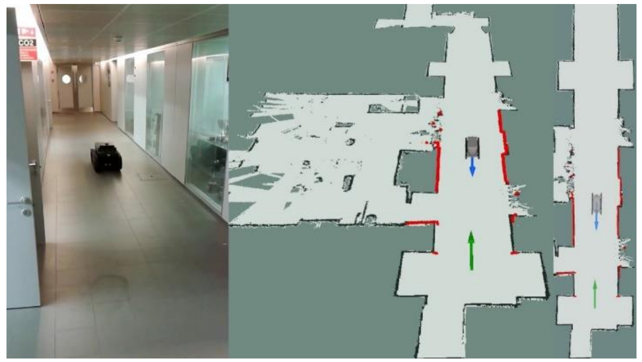


Figure 2: Example of Robot localization [2]

## Background Information: Mapping in Robotics

How is mapping mostly done using ROS in the mobile robotics scene now?

- Manual mapping with SLAM methods provided in ROS (Gmapping, Hector SLAM) using teleoperation
- Autonomous mapping using Frontier-based algorithm and RRT exploration algorithm

## Current Implementations of Mapping in ROS: Manual Mapping

A map of the environment for the robot can be mapped by having an operator control the robot via teleoperations to capture the environment using its on-board sensors. The start and end of the mapping task is dictated by the operator.

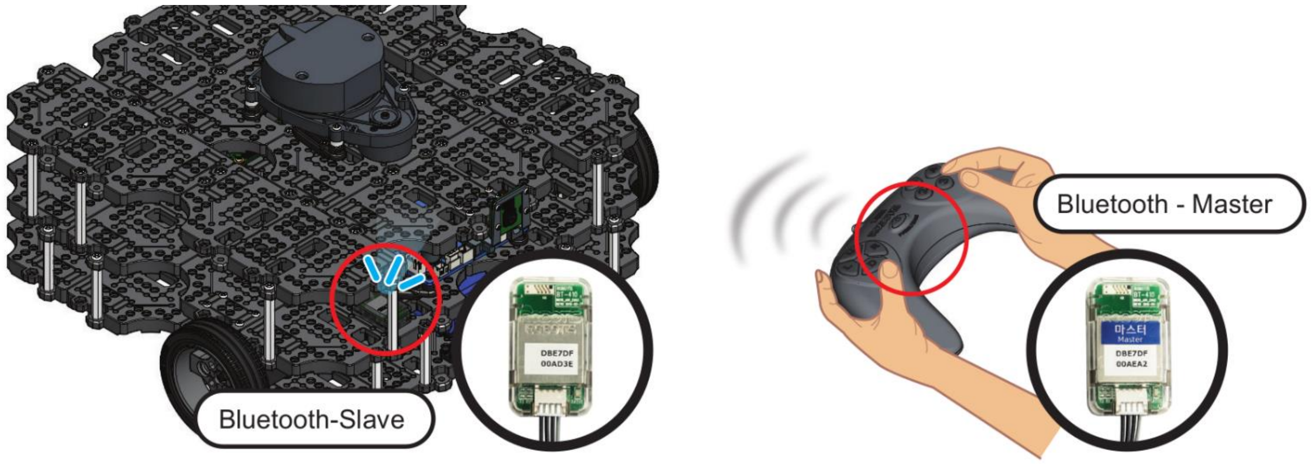


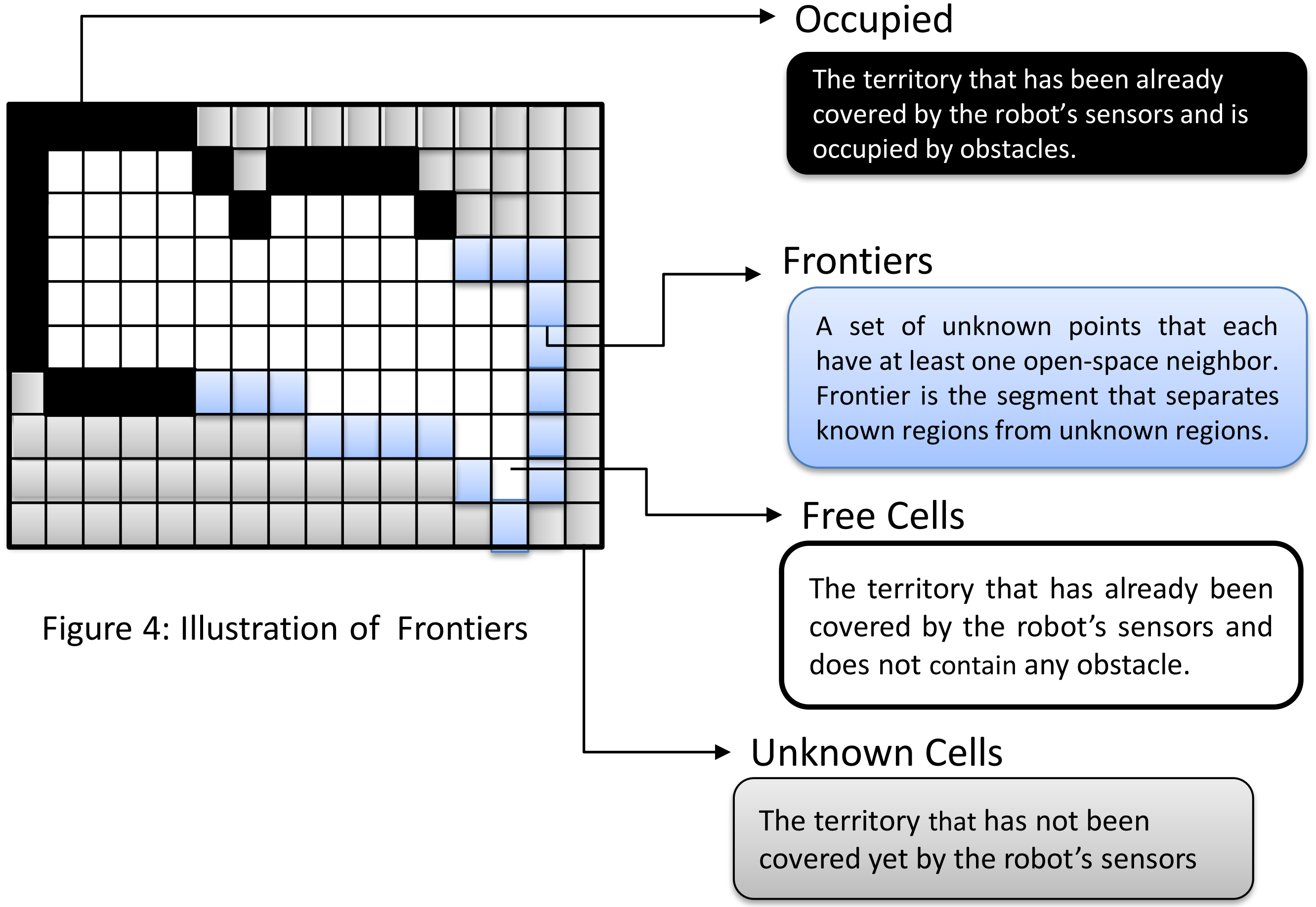
Figure 3 : Teleoperated robot [3]

<u>Drawbacks of manual mapping via Teleoperation</u>

1. Requires an operator
2. Requires a controller (physical/digital)
3. Time consuming process
4. Tedious process

## Current Implementations of Autonomous Mapping in ROS: Frontier Exploration

A mobile robot can perform the action of mapping autonomously until the entire environment has been explored. The exploration goal contains an initial point to start exploration, and a polygonal boundary to limit the exploration scope.



**Occupied**

The territory that has been already covered by the robot's sensors and is occupied by obstacles.

**Frontiers**

A set of unknown points that each have at least one open-space neighbor. Frontier is the segment that separates known regions from unknown regions.

**Free Cells**

The territory that has already been covered by the robot's sensors and does not contain any obstacle.

**Unknown Cells**

The territory that has not been covered yet by the robot's sensors

Figure 4: Illustration of Frontiers

<u>Prominent Packages available in ROS offering this feature</u>

1. explore_lite (ROS1, maintained up till noetic) [4]
2. m-explore-ros2 (ROS2, port from ROS1 explore_lite package) [5]

<u>Drawbacks of frontier exploration</u>

1. Not very customizable
2. Sometimes fail to find the globally optimal solution because they do not consider all the data.
3. Computationally expensive to compute frontier clusters

## Current Implementations of Autonomous Mapping in ROS: RRT Exploration

Points are produced at random and linked to the nearest available node. Every time a vertex is formed, it must be checked to see if it is outside of an obstruction. Furthermore, the vertex must be chained to its closest neighbor while avoiding impediments. When a node is formed within the desired region, or a limit is reached, the process stops.

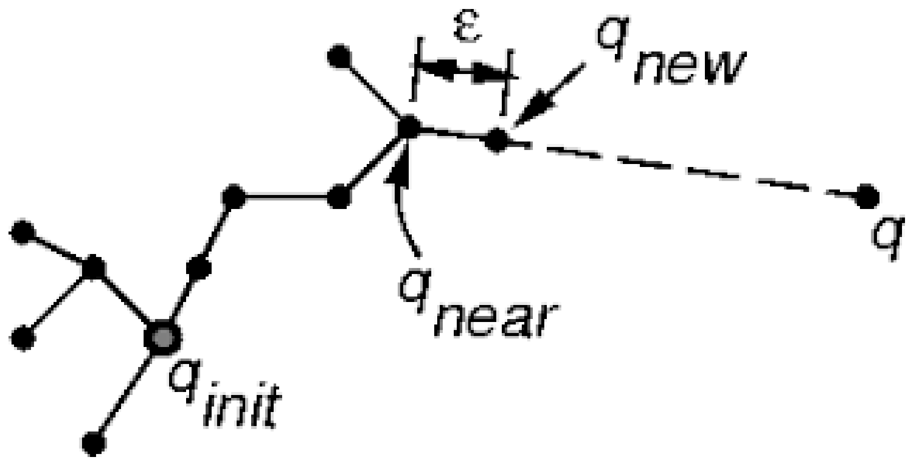


Figure 5: Illustration of Rapid Exploring Random Trees [6]

Prominent Packages available in ROS offering this feature

1. rrt_exploration (ROS1, maintained up till kinetic) [7]

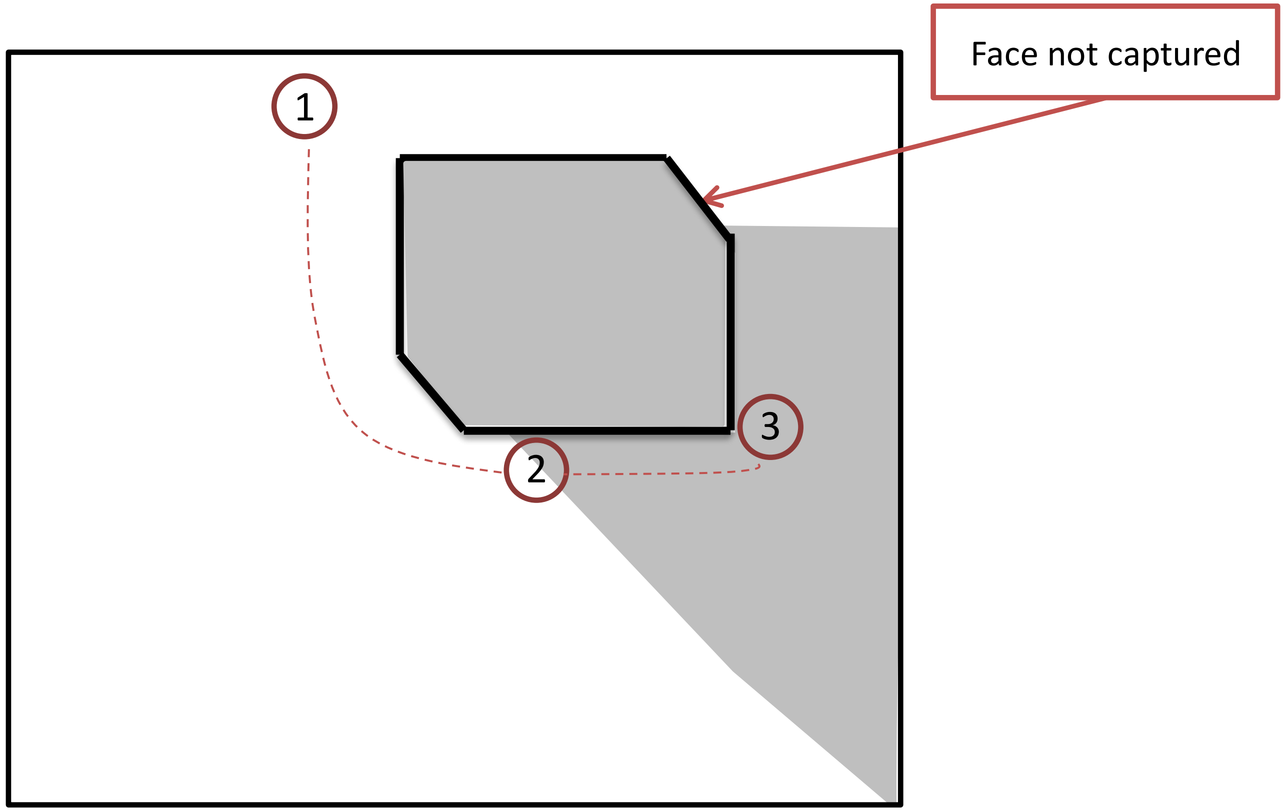
Drawbacks of rrt exploration

1. Not very customizable
2. Only supported for Gmapping

## Current Implementations of Autonomous Mapping in ROS: Use-Case Limitation

Limited customizability = Limited use-cases

Use-case Example: Environment Reconstruction



## Problem and Solution

### Problem Statement

- Autonomous Mapping Package in ROS2
- Supported for various SLAM features in ROS2
- Highly customizable for end-users to suit their various use-cases

### Our Proposed Method

The proposed algorithm completes the autonomous exploration task when all regions in the map is "visited" by the mobile robot instead of completion due to a lack of frontiers or points to go to.

The points selected by the robot to explore takes into account information like topological data and contents of the search grid etc.

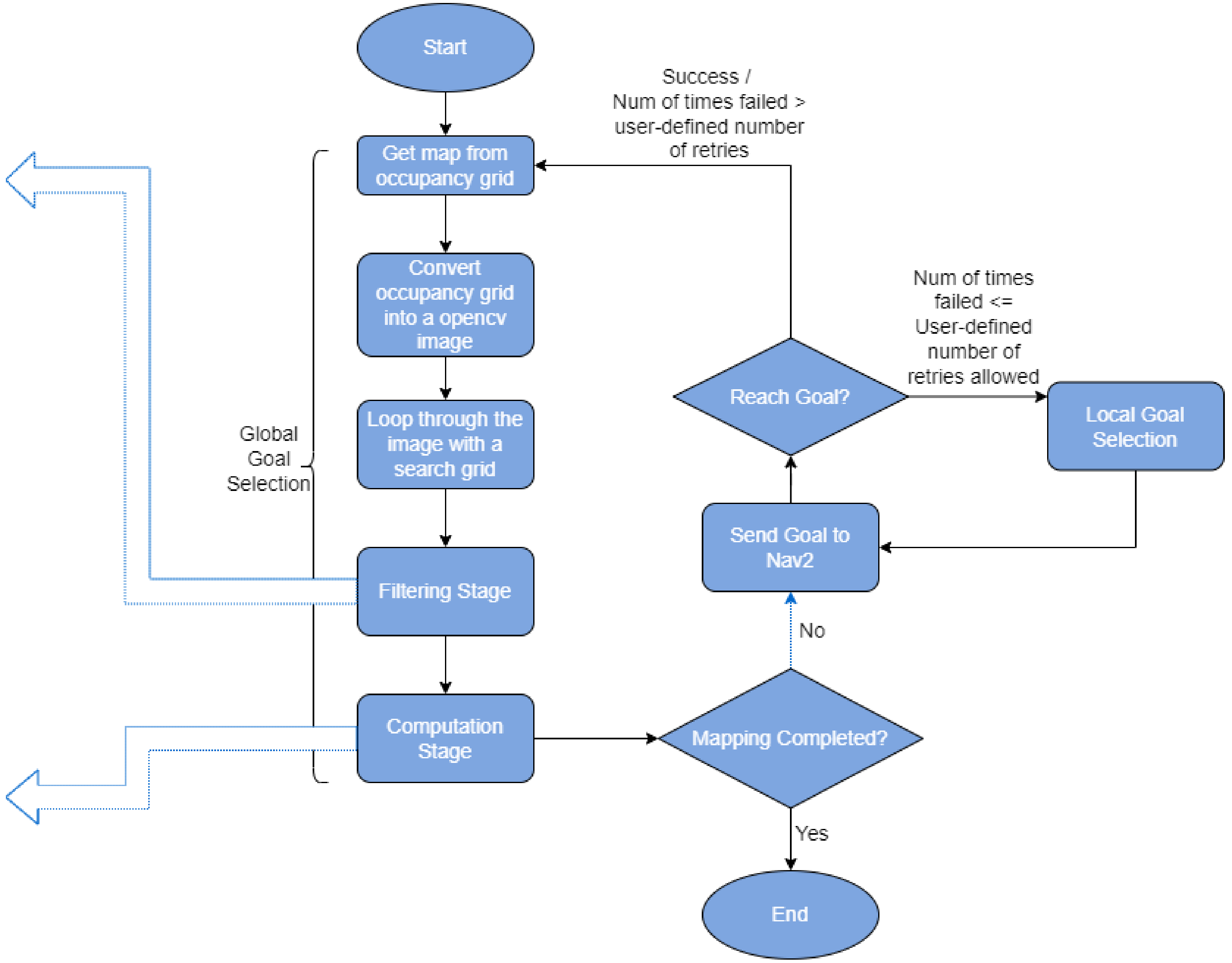
These information are tied to adjustable weights, which allows the users to decide the way in which the robot will execute the autonomous mapping task.

# Exploration Algorithm Flow

The Filtering Stage is the stage where irrelevant search grids are discarded before any computation is done to improve efficiency.

The Computation Stage is the stage relevant search grid are awarded a preference number based on its information gain that is computed via a cost function.

The cost function will feature parameters which the user can change easily to suit their applications and use cases.



Start
Get map from occupancy grid
Convert occupancy grid into a opencv image
Loop through the image with a search grid
Filtering Stage
Computation Stage
Global Goal Selection
Success / Num of times failed > user-defined number of retries
Reach Goal?
Num of times failed <= User-defined number of retries allowed
Local Goal Selection
Send Goal to Nav2
No
Mapping Completed?
Yes
End

Managed by

## Key Benefits, Possible Use-Cases and Limitations

### Advantages over previous implementations

- Will be supported for any ROS-based SLAM algorithms. Currently supported for: cartographer, gmapping and slam toolbox
- Unexplored areas have more context than simply frontier size and distance information
- Users have greater control of the robot during the autonomous mapping process

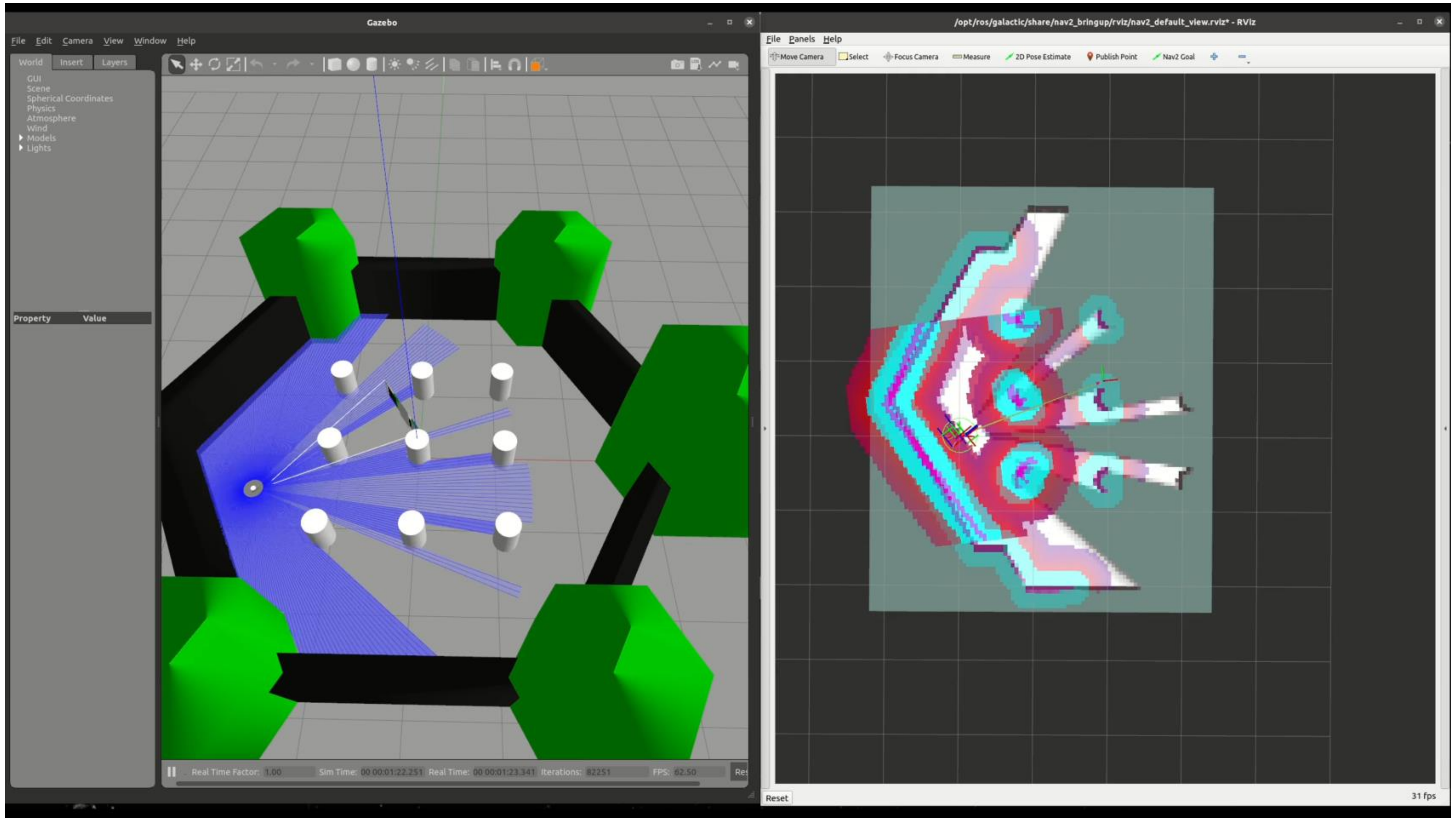
### Suitable use-cases

- Robot deployment in limited / inaccessible areas.
- Robot deployment for large operational spaces.
- Autonomous mapping with minimum human intervention
- Coupled with data-capturing equipment, it can replace the manual labour for thorough scene reconstruction of indoor environments

### Possible Limitations

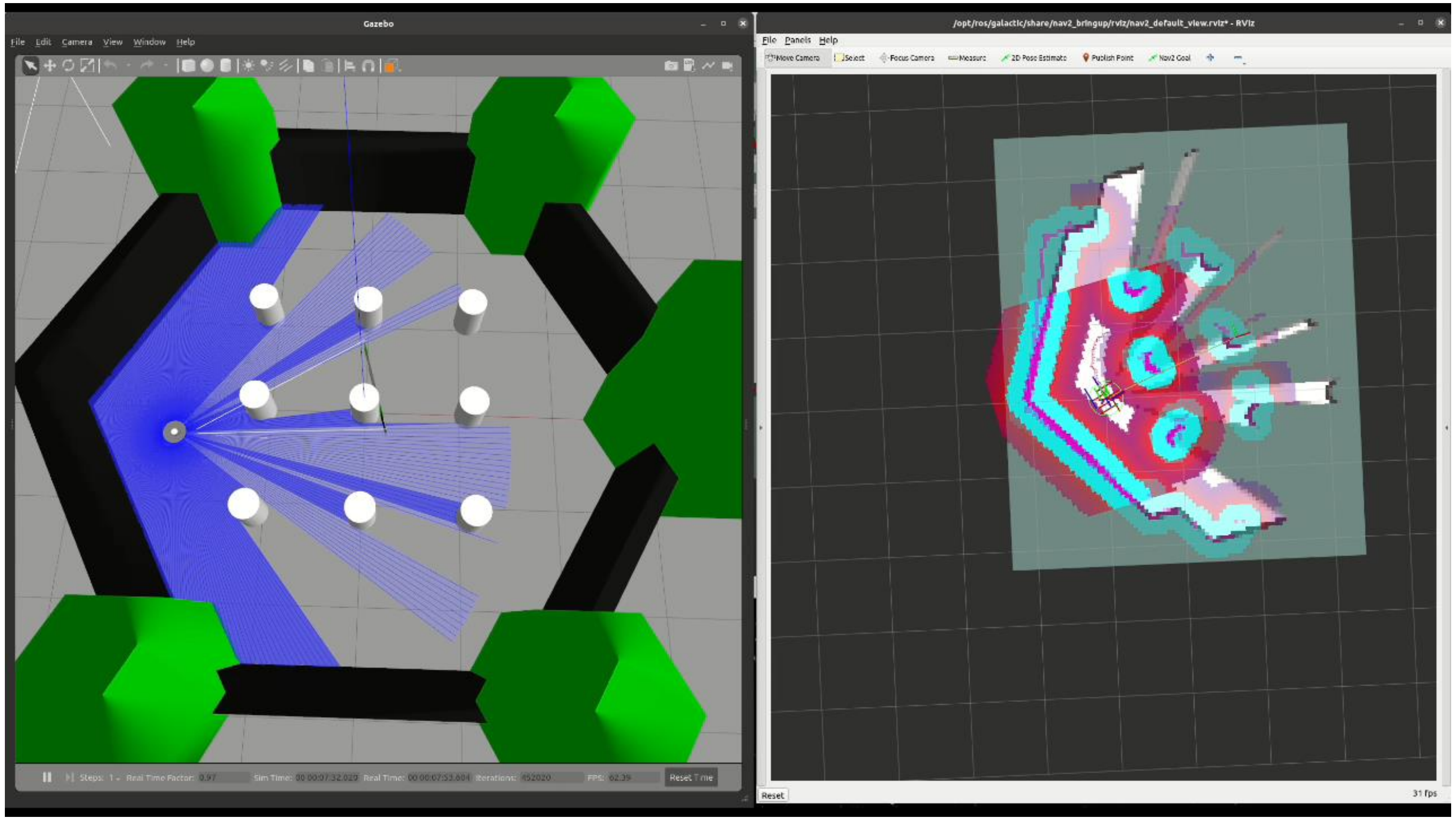
- The overall environment to be mapped must be enclosed for the mapping task to end autonomously

# Simulation Demo (Fast Mapping Configuration)



Note: Video has been sped up 2x

# Simulation Demo (Thorough Mapping Configuration)



Note: Video has been sped up 2x

Managed by

## Future Works

- We plan to add the option for users to request the next exploration goal via a ROS client-service model for more control instead of the mobile robot doing the task till completion
- Support for more ROS-based SLAM methods
- Increase it versatility in mapping unenclosed environments
- Add more customizable features

## When will this package be release?

Alpha Version – Looking at a release at the end of the year

Beta Version – To be confirmed

# References

[1] W. Selby, “Building maps using google cartographer and the OS1 Lidar Sensor,” *Ouster*. [Online]. Available: https://ouster.com/blog/building-maps-using-google-cartographer-and-the-os1-lidar-sensor/. [Accessed: 11-Jun-2022].

[2] C. M. Correia da Costa, “Carlosmccosta/dynamic_robot_localization: Point cloud registration pipeline for robot localization and 3D perceptionCarlos Miguel Correia da Costa,” *GitHub*. [Online]. Available: https://github.com/carlosmccosta/dynamic_robot_localization. [Accessed: 10-Jun-2022].

[3] “Robotis e Manual,” *Turtlebot3*. [Online]. Available: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/. [Accessed: 09-Jun-2022].

[4] J. Hörner, “HRNR/m-explore: Ros Packages for Multi Robot Exploration,” *GitHub*. [Online]. Available: https://github.com/hrnr/m-explore. [Accessed: 07-Jun-2022].

[5] Robo-Friends, “Robo-friends/M-explore-ros2: Explore_lite port to ROS2,” *GitHub*. [Online]. Available: https://github.com/robo-friends/m-explore-ros2. [Accessed: 07-Jun-2022].

[6] S. M. LaValle, “Rapidly-Exploring Random Trees,” *Rapidly-exploring random trees*, 29-Aug-2001. [Online]. Available: http://msl.cs.uiuc.edu/~lavalle/cs497_2001/book/iplan/node16.html. [Accessed: 07-Jun-2022].

[7] U. Hassan Abdul-Rahman and S. Mukhopadhyay, “Hasauino/rrt_exploration: A ROS package that implements a multi-robot RRT-based map exploration algorithm. it also has the image-based frontier detection that uses image processing to extract frontier points.,” *GitHub*. [Online]. Available: https://github.com/hasauino/rrt_exploration. [Accessed: 07-Jun-2022].

# Thank You!

Managed by