Autonomous Navigation in Art Gallery

Submitted by
Anusha Rani Bheemishetty (3638633)
Shunnoy Sarkar (3660409)
15th September 2024

OVERVIEW

A robot should explore its environment with some a-priori knowledge e.g. represented as a map of the room. The art gallery consists of several "rooms' like a maze. The robot should explore these rooms and identify several targets (April-Tags) and the related painting close to the tag. The algorithm should control the robot without any human intervention and identify the class of objects in the painting and the AprilTag ids.

GOALS

1. Autonomous exploration of a randomly generated maze with 5x5 or 6x6 cells,
2. Avoid collisions,
3. Don't get stuck with your robot
4. Identify the three randomly spawned april-tags,
5. Detect and classify the objects in the three paintings,
6. Publish the id of the detected april-tags,
7. Publish the detected object in the painting and
8. Get back to your starting pose.

MILESTONES

Autonomous exploration of maze

The autonomous exploration of the maze has been done using the Nav2 stack and the robot is made to navigate the entire maze through predefined goal points. The goal points have been chosen as 25 synchronously distributed points in the generated maze and follow a certain order so that the entire maze is explored.

The waypoint_follower node that is responsible for navigating the robot through these goal points also makes the robot rotate 360 degrees after reaching every individual goal point. The intention of implementing this is to be able to detect the images and april tags that might be on any of the proximate walls.

The waypoint_follower also includes an algorithm that commands the goal point to be set to the origin as soon as three distinct images are detected (with a confidence level of 0.9 or more),that is implemented in the yolo_to_ros node, however the code fails to run as intended.

Avoid collisions

The Nav2 stack takes care of object collision. Initially the parameters in the nav2_params.yaml were altered so as to increase the linear speed of the robot, however at an increased speed the robot tends to crash with the walls, thus a maximum linear speed of 0.4m/s was chosen.

Identification of AprilTags

The detection of AprilTags is done by launching the v4l2_16h5.launch.yml file which is in the apriltag_ros package. It was observed however that the detection of april tags was not accurate due to some false positives also being published. Attempts to improve it by increasing the detection margin so as to filter out the lower-confidence detections were made but were not successful.

Detection and Classification of Objects

The detection and classification of the objects is done by the yolo_to_ros node. As mentioned before the yolo_to_ros node sends a boolean “True” value to the waypoint_follower node when three distinct objects have been detected and identified with a confidence score of 0.9 or greater. However the robot stops navigating as soon as the third image is detected. The problem could not be identified.

Other attempted alternatives

1. One of the fundamental principles of ROS is to not reinvent the wheel, so the usage of previously created packages like nav2_simple_commander in the navigation repository of ROS2 (https://github.com/ros-navigation/navigation2/tree/humble/nav2_simple_commander/nav2_simple_commander). was tried out. It however did not come to fruition and a new node had to be made which doesn’t utilise the nav2_simple_commander package.