ARAMS Project Documentation

The robotic Curator - Exploring an Art Gallery

Team members:

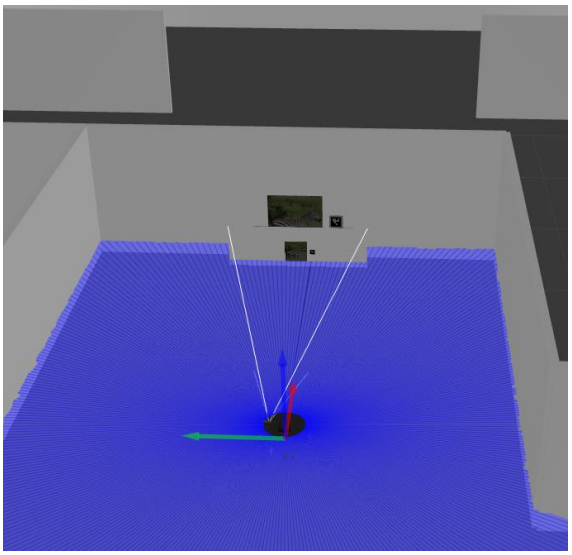
Seyedehmona Basiri – 3197979

Amitojsingh Arora - 3238617

SS 2024

1 Objectives of the project

- Implementation of autonomous navigation and exploration of an art gallery environment with a TurtleBot 3 Waffle.
- Detect and identify all AprilTags inside the gallery spawn at random locations.
- Implement a neural network based object detection algorithm to recognise and classify the paintings located next to AprilTags.
- After the detection of three AprilTags and paintings, return to the dock.

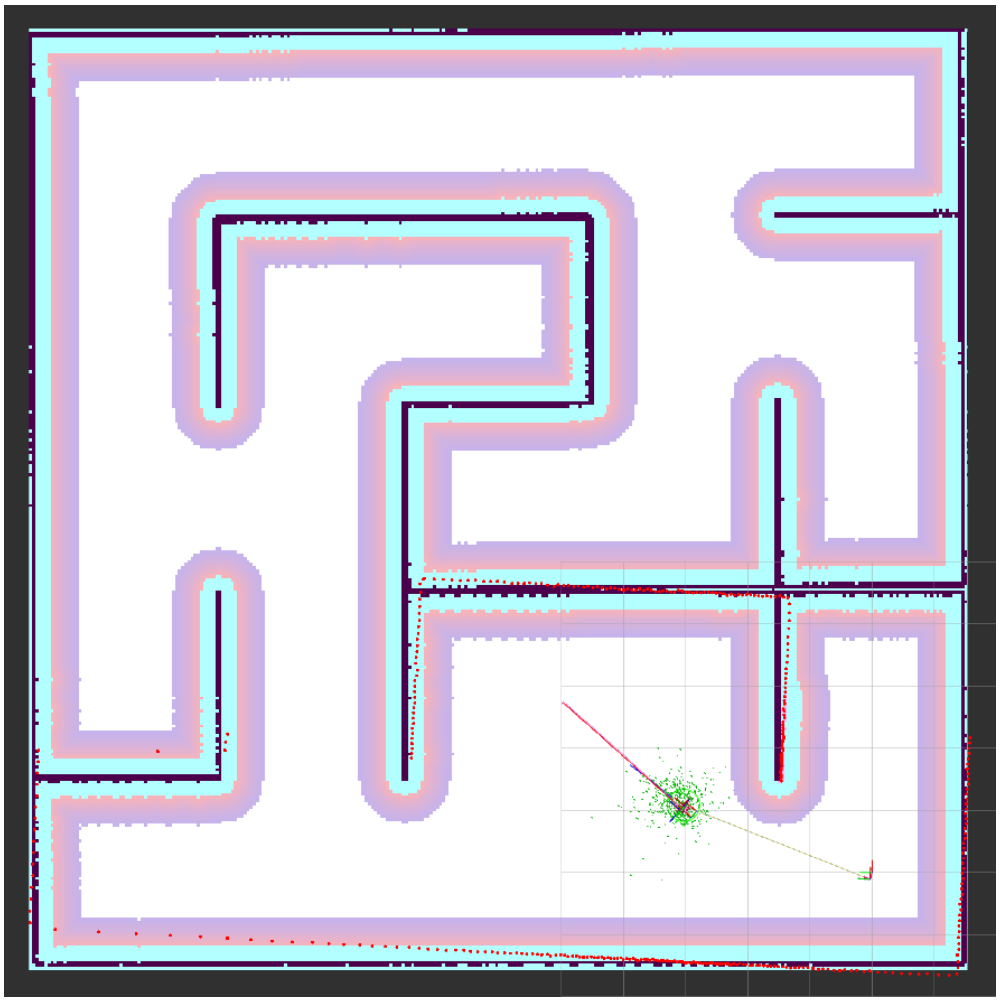


2 Prerequisites and Packages

The following ROS2 packages were used in the project:

- **SLAM Toolbox:** to generate a map of the environment and localize robot within the provided map.
- **Rqt Steering:** Manual control of the robot during initial testing and mapping.
- **AprilTag Detections:** The package "Apriltag_ros" by Christian Rauch https://github.com/christianrauch/apriltag_ros.git was used to detect AprilTags.
- **Auto mapper package:** A package made by Omar Salem https://github.com/Omar-Salem/auto_mapper.git which used frontier exploration to map the arena.

- **Nav2:** for path planning and autonomous navigation.
- **YOLOv8:** For object detection and classification of paintings. YOLOv8n was used as the neural network model to detect the paintings next to AprilTags.



3 Approach and Algorithm

3.1 Mapping the maze

A combination of Slam Toolbox and Nav2 was used in the auto mapper package to autonomously generate a map of the art maze. The map is then used for navigation and exploration.

3.2 Autonomous Navigation

For autonomous navigation, the approach involves adapting the “auto explorer” node source code. Instead of random exploration, 16 key coordinates across the map were strategically chosen to ensure complete coverage of the environment.

Robot is programmed to navigate to the predefined key points and perform a 360-degree rotation, allowing the sensors and camera to get a better view of the surroundings. This approach enhances the detection of AprilTags and objects. After Detection of 3 tags and images, the robot goes back to its initial pose.

3.3 AprilTag Detection

The “apriltag_ros” package was used for detecting AprilTags. The package processes images captured by the robot´s camera and identifies the ID of each Tag.

3.4 Image classification with YOLOv8

Once AprilTag is detected, next step is identifying the paintings. For this purpose, YOLOv8 was used. A custom dataset consisting of paintings in the maze was created for this purpose. The dataset was then labeled. After labeling, the YOLOv8 model was trained on this dataset to ensure it could accurately detect and classify the paintings during the robot's exploration.

4 Running the project

To let the robot explore the maze the after cloning and building the workspace, the following commands are required.

First the maze is to be generated from the art gallery workspace in gazebo:

ros2 launch tb3_gazebo arams.launch.py

then the nav2 server from my_robot_navigation and the localization with the SLAM toolbox from my_robot_slam package is initialized:

ros2 launch my_robot_navigation robot_nav.launch.py

ros2 launch my_robot_slam localization.launch.yaml

from launching the nav2 servers Rviz should also open and by launch localization the map should load. The initial position estimate of the is to be given in Rviz.

Now the files for Apriltag and Image detection are to be run and they are as follows:

ros2 launch apriltag_ros apriltag.launch.yml

ros2 run yolo_ros yolo_to_ros

After the initalisation of the setup now the auto explorer node is run to let the robot explore the arena:

ros2 run my_robot_navigation art_explorer

With running these launch files and nodes the robot shall start exploring the maze and identifying the Apriltags and images.

5. Some Challenges

Initially, the navigation approach involved random goal generation, which was inefficient and did not guarantee complete exploration of the environment.

To address this, a more structured approach was adopted by selecting approximately 15 key points across the map and navigating to these points.

The YOLOv8 model sometimes misclassified paintings when the robot approached them from certain angles. This issue affected the accuracy of painting recognition.