A purple rectangular sign with white text

Description automatically generated

*A Report on*

***Creating 2D Occupancy Grid Map using overhead infrastructure cameras***

***Submitted for the Intel Unnati Industrial Training Program 2024***

***By***

**P NIKHIL (1NT21EC097)**

Under the Guidance of

**Dr. Viswanatha V**

Professor

Dept. of Electronics and Communication Engineering

Blue text on a white background

Description automatically generated

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**YELAHANKA, BENGALURU- 560064**

**Contents**

**Chapter 1 Introduction………………………………………………………………2**

**Chapter 2 Literature Survey………………………………………………………...2**

**Chapter 3 Motivation and Objectives……………………………………………....3**

**Chapter 4 Architecture of the Proposed System…………………………………...4**

**Chapter 5 Hardware/Software Used………………………………………………..4**

**Chapter 6 Design Specification/Dataset Description……………………………....4**

**Chapter 7 Expected Outcomes……………………………………………………....6**

**Chapter 1**

**Problem Definition**

The problem you are trying to solve is creating a 2D occupancy grid map using overhead infrastructure cameras in a Gazebo and ROS2 simulation environment. The goal is to generate a comprehensive representation of the environment that can be used for autonomous navigation of mobile robots.

Specifically, the problem defines the following key objectives:

1. Generating a detailed 2D occupancy grid map by stitching together images from multiple overhead cameras, providing a top-down view of the environment.
2. Supporting dynamic environments by updating the occupancy grid map in real-time as objects are moved around.
3. Integrating the occupancy grid map with the ROS2 navigation stack, allowing autonomous mobile robots to use the map for path planning and obstacle avoidance.

Adding semantic labeling to the occupancy grid map, such as identifying specific objects like tables, chairs, or other AMRs, to provide additional context for the navigation system.

By solving this problem, you aim to create a robust and adaptable environment representation that can be effectively utilized by autonomous mobile robots operating in complex, dynamic environments like warehouses or factories.

The proposed solution involves setting up the Gazebo simulation environment with overhead cameras, developing the occupancy grid mapping pipeline in ROS2, and integrating the resulting map with the ROS2 navigation stack. This approach leverages the capabilities of both Gazebo and ROS2 to create a realistic simulation and test environment for validating the occupancy grid mapping and navigation algorithms.

In this project, we aim to enhance the navigation and localization capabilities of autonomous robots by developing a real-time 2D occupancy grid map using overhead infrastructure cameras, leveraging Gazebo and ROS. Overhead cameras will capture comprehensive views of the environment, providing essential data to identify obstacles and free spaces. Through advanced image processing techniques, we will convert raw camera data into a usable format, enabling the generation of a 2D occupancy grid map where each cell represents the occupancy status of specific areas. Utilizing ROS, we will ensure real-time data handling and map updating, maintaining an accurate and current representation of the environment. The mapping and navigation algorithms will be rigorously tested in a simulated environment created in Gazebo before real-world deployment. Finally, the occupancy grid map will be integrated with ROS-based localization and navigation systems, allowing autonomous robots to accurately determine their positions and safely navigate through their environment.

**Chapter 2**

**Solution Approach**

The detailed solution approach for creating a 2D occupancy grid map using overhead infrastructure cameras in a Gazebo and ROS2 environment:

**1. Set up the Gazebo Simulation Environment**

1. Create a simulated room or warehouse environment in Gazebo, including static objects like tables, chairs, and boxes.
   * Use Gazebo's built-in models or create custom models for the objects in the environment.
   * Ensure the environment is large enough to accommodate the desired testing and validation scenarios.
2. Place 4 overhead RGB cameras in a 2x2 pattern to cover the room.
   * Position the cameras at an appropriate height and orientation to capture a comprehensive top-down view of the environment.
   * Ensure the camera fields of view overlap to provide complete coverage of the floor area.
   * Configure the camera parameters, such as resolution, frame rate, and field of view, to optimize the quality and coverage of the captured images.
   * Publish the camera data as ROS2 topics for use by the occupancy grid mapping pipeline.

**2. Develop the Occupancy Grid Mapping Pipeline**

1. ROS2 Package for Image Stitching and Occupancy Grid Mapping:
   * Create a new ROS2 package to handle the image stitching and occupancy grid mapping.
   * Implement ROS2 node(s) that subscribe to the topics of the 4 overhead cameras.
   * Use OpenCV or other image processing libraries to stitch the camera images together, creating a single, unified top-down view of the environment.
   * Segment the floor area in the stitched image into occupied and free pixels, generating the 2D occupancy grid map.
   * Experiment with different segmentation algorithms, such as thresholding, edge detection, or machine learning-based methods, to accurately identify occupied and free spaces.
   * Publish the occupancy grid map as a ROS2 topic for use by other components, such as the navigation stack.
   * Ensure the occupancy grid map is published in a format compatible with the ROS2 navigation stack (e.g., nav\_msgs/OccupancyGrid)
2. Real-time Occupancy Grid Map Updates:
   * Continuously monitor the camera feeds and update the occupancy grid map in real-time as objects are moved around.
   * Implement efficient algorithms and data structures to ensure the occupancy grid map can be updated quickly, even in dynamic environments.
   * Consider using techniques like incremental updates or region-of-interest updates to minimize the computational overhead.
   * Publish the updated occupancy grid map to the ROS2 topic, allowing other components to receive the latest environment representation.

**3. Integrate the Occupancy Grid Map with the ROS2 Navigation Stack**

1. ROS2 Costmap Plugin:
   * Develop a ROS2 costmap plugin that can ingest the overhead camera-based occupancy grid map and make it available to the navigation stack.
   * Ensure the plugin can handle updates to the occupancy grid map in real-time, allowing the navigation stack to react to changes in the environment.
   * Integrate the costmap plugin with the ROS2 navigation stack, such as the nav2\_costmap\_2d package, to enable the use of the overhead camera-based occupancy grid map for path planning and obstacle avoidance.
2. Expansion of the SLAM Toolbox Grid Map:
   * Alternatively, create a ROS2 package that subscribes to the SLAM toolbox's grid map topic.
   * Implement algorithms to fuse the SLAM toolbox's grid map with the overhead camera-based occupancy grid map, providing a more comprehensive representation of the environment.
   * Consider techniques like map merging or map expansion to combine the two data sources.
   * Publish the expanded grid map for use by the navigation stack, allowing the AMR to leverage the additional environment information.
3. Validation and Testing:
   * Simulate an autonomous mobile robot (AMR) using the ROS2 navigation stack, configured to use the overhead camera-based occupancy grid map.
   * Observe the AMR's navigation performance, including its ability to plan paths, avoid obstacles, and reach desired goals.
   * Compare the AMR's map to the ground truth occupancy grid map generated from the overhead cameras to assess the accuracy and reliability of the integrated system.
   * Refine the integration and mapping algorithms as needed to ensure accurate and robust navigation performance.
   * Consider testing the system in various scenarios, such as dynamic environments with moving objects, to validate its adaptability and responsiveness.

By following this detailed solution approach, you can create a comprehensive and effective 2D occupancy grid mapping system using overhead infrastructure cameras in a Gazebo and ROS2 environment, and seamlessly integrate it with the ROS2 navigation stack for autonomous mobile robot applications.

**Chapter 3**

**Novelty of the approach**

The comparison of the proposed solution to prior art solutions, highlighting the novel aspects:

1. **Use of overhead cameras for occupancy grid mapping**:
   * The proposed solution leverages overhead infrastructure cameras to capture a more comprehensive top-down view of the environment.
   * Prior art solutions have used monocular vision, stereo vision, or laser range finders for occupancy grid mapping, but not overhead cameras.
2. **Real-time updates for dynamic environments**:
   * The ability to update the occupancy grid map in real-time as objects are moved around is a key feature of the proposed solution.
   * Some prior art solutions support mapping in dynamic environments, but the details of real-time updates are not always specified.
3. **Semantic labeling of the occupancy grid map**:
   * The optional semantic labeling of the occupancy grid map using object detection is a novel aspect of the proposed solution.
   * Prior art solutions have not explicitly mentioned this feature, although some have used segmented occupancy maps for robot sensing and navigation.
4. **Integration with the ROS2 navigation stack**:
   * The proposed solution focuses on seamlessly integrating the occupancy grid map with the ROS2 navigation stack, either through a custom costmap plugin or by expanding the SLAM toolbox's grid map.
   * While prior art solutions have used occupancy grid maps for navigation, the specific integration with the ROS2 navigation stack is not discussed in detail.

The key novel aspects of the proposed solution are:

1. **The use of overhead infrastructure cameras** for occupancy grid mapping, providing a more comprehensive top-down view of the environment.
2. **The ability to update the occupancy grid map in real-time** for dynamic environments.
3. **The optional semantic labeling of the occupancy grid map** using object detection, providing additional context for the navigation system.

By combining these novel features with the integration of the occupancy grid map into the ROS2 navigation stack, the proposed solution aims to provide a robust and adaptable environment representation for autonomous mobile robot navigation in complex, dynamic environments.

**Chapter 4**

**Methodology**

**Step 1: Install ROS and Gazebo**

1. **Set Up Your System:**
   * Ensure your operating system is compatible (Ubuntu is recommended, preferably 20.04 or later).

wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis\_tools/master/install\_ros2\_foxy.sh

sudo chmod 755 ./install\_ros2\_foxy.sh

bash ./install\_ros2\_foxy.sh