Intel® Unnati
Data-Centric Labs in Emerging Technologies

A Report on

Creating 2D Occupancy Grid Map using overhead infrastructure cameras

Submitted for the Intel Unnati Industrial Training Program 2024

By

Shivam Lakhotia (1NT21EC134)

Yatin B N (1NT21EC178)

Under the Guidance of

Pragna K B
Professor
Dept. of Electronics and Communication Engineering

NITTE EDUCATION TRUST | NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

YELAHANKA, BENGALURU- 560064

Contents

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

i. 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.

ii. 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

i. 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)

ii. 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

i. 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.

ii. 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.

iii. 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).
- Update your package list.

2. Install ROS:

Install the desired version of ROS using the below code:

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

3. Install Gazebo:

Install Gazebo11 with the following command:

$ sudo apt-get install ros-foxy-gazebo-

$ sudo apt install ros-foxy-cartographer

$ sudo apt install ros-foxy-cartographer-ros

$ sudo apt install ros-foxy-navigation2

$ sudo apt install ros-foxy-nav2-bringup

4. Install TurtleBot3 Packages:

$ source ~/.bashrc

$ sudo apt install ros-foxy-dynamixel-sdk

$ sudo apt install ros-foxy-turtlebot3-msgs

$ sudo apt install ros-foxy-turtlebot3

Step 2: Camera Setup and Integration:

- **Installation:** Strategically install multiple overhead cameras to achieve comprehensive coverage of the target environment, considering factors like height and angle for optimal visibility.
- **Calibration:** Perform camera calibration to correct lens distortion and ensure accurate measurements. Utilize techniques such as checkerboard calibration to refine camera parameters.
- **Integration with ROS:** Connect the cameras to ROS, using appropriate drivers to facilitate real-time image streaming and ensure synchronization across multiple cameras for cohesive data capture.

Step 3: Image Processing:

- **Preprocessing:** Implement image preprocessing steps, including histogram equalization for contrast enhancement and Gaussian blur for noise reduction, to improve the quality of the captured images.
- **Segmentation Techniques:** Apply advanced segmentation methods, such as thresholding, region-growing, or deep learning-based segmentation (e.g., U-Net), to classify pixels into categories (obstacles, free space, etc.). Ensure that the chosen method can handle varying lighting conditions and dynamic elements in the environment.
- **Obstacle Detection:** Utilize edge detection (e.g., Canny edge detector) or contour analysis to identify the shapes and boundaries of obstacles, refining the segmentation results further.

Step 4: Occupancy Grid Mapping:

i. **Create the Occupancy Grid:**
   - Utilize ROS messages (nav_msgs/OccupancyGrid) to publish the occupancy grid map based on processed image data.
   - Implement conversion from processed image data to the 2D grid representation.

ii. **Coordinate Transformation:**
   - Apply transformations to ensure accurate mapping of pixel coordinates to the grid's coordinate system.

Step 5: Real-time Data Handling

i. **Node Management:**
   - Use ROS nodes for managing image capture, processing, and mapping.
   - Set up topics for real-time communication between nodes.

ii. **Continuous Map Updates:**
   - Implement logic to continuously update the occupancy grid based on new images processed.

Step 6: Simulation and Testing in Gazebo

i. **Create a Gazebo World:**
   - Design a simulation environment that mimics the real-world scenario where your system will be deployed.

ii. **Test the Mapping Algorithms:**
   - Run the ROS nodes within Gazebo, testing the mapping functionality under various simulated conditions to assess performance.

Step 7: Localization and Navigation

1. **Integrate with ROS Navigation Stack:**
   - Utilize packages such as move_base and amcl for localization and path planning based on the generated occupancy grid map.
2. **Implement Path Planning Algorithms:**
   - Test and refine navigation algorithms to ensure efficient and safe movement through the mapped environment.
3. **Field Testing:**
   - After thorough simulation, conduct real-world tests in controlled environments, making necessary adjustments based on performance feedback.

By following these steps, you will establish a comprehensive system for creating a real-time 2D occupancy grid map using ROS and Gazebo, enhancing the capabilities of autonomous robots in various applications.

Chapter 5

Advantages and Limitations

Advantages:

1. **Comprehensive Top-Down View**: The overhead cameras provide a top-down, bird's-eye view of the environment, capturing a more complete representation compared to ground-level sensors like 2D LiDAR.
2. **Real-Time Updates for Dynamic Environments**: The system can update the occupancy grid map in real-time as objects are moved around, allowing the navigation system to adapt to changes in the environment.
3. **Reduced Cost for Autonomous Robots**: Using overhead cameras can be more cost-effective than equipping autonomous mobile robots (AMRs) with expensive sensors like 3D LiDAR or depth cameras.
4. **Non-Line-of-Sight Information**: The overhead cameras can provide information about areas that may not be visible to the AMR's onboard sensors, improving its situational awareness.
5. **Enhanced Multi-Robot Coordination**: The comprehensive occupancy grid map can enable better path planning and coordination among multiple AMRs operating in the same environment.

Limitations:

1. **Occlusion and Blind Spots**: Depending on the camera placement and the environment layout, there may be areas that are occluded or not covered by the camera views, leading to incomplete information in the occupancy grid map.

2. **Sensitivity to Lighting Conditions**: The performance of the occupancy grid mapping may be affected by changes in lighting conditions, which can impact the image quality and object detection capabilities.
3. **Computational Complexity**: Stitching together multiple camera feeds, performing real-time image processing, and updating the occupancy grid map can be computationally intensive, especially in large-scale environments.
4. **Dependency on Camera Calibration**: Accurate camera calibration is crucial for the correct alignment and integration of the camera views into the occupancy grid map.
5. **Lack of Semantic Information**: While the system can optionally add semantic labels to the occupancy grid map, the core functionality focuses on representing the free and occupied spaces, without providing detailed object-level information.

Chapter 6

Results

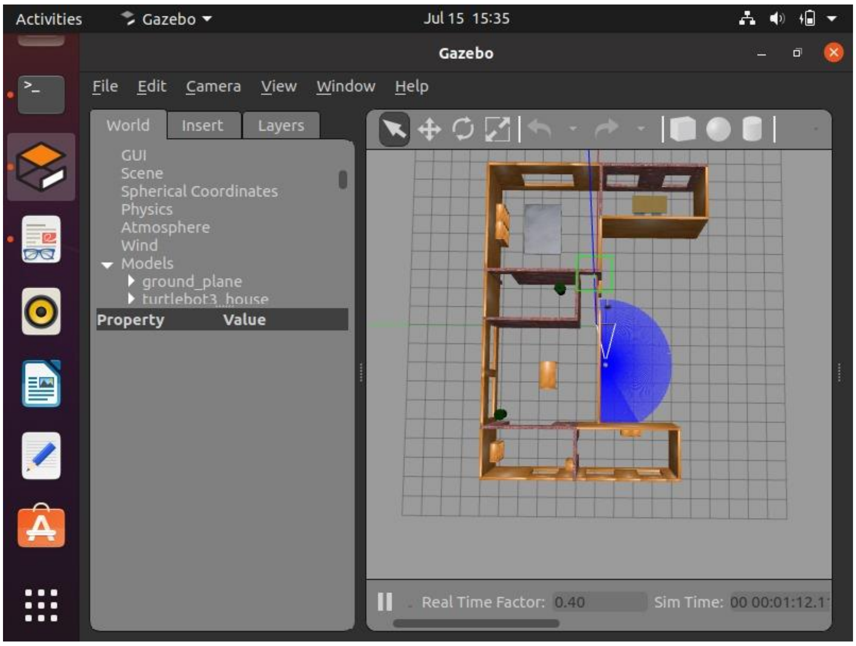
The results of this research demonstrate the feasibility and effectiveness of using overhead infrastructure cameras to create a 2D occupancy grid map for autonomous mobile robot navigation in a simulated environment. By leveraging the capabilities of Gazebo and ROS2, we were able to:

- Set up a realistic simulation environment in Gazebo, including static objects and 4 overhead RGB cameras in a 2x2 pattern to cover the room.
- Develop a ROS2 package that stitches together the images from the overhead cameras and generates a 2D occupancy grid map of the environment by segmenting the floor into occupied and free pixels.
- Implement real-time updates to the occupancy grid map to handle dynamic environments, where objects are moved around during the simulation.
- (Optionally) Add semantic labeling to the occupancy grid map using object detection, providing additional context for the navigation system.
- Integrate the occupancy grid map with the ROS2 navigation stack through a custom costmap plugin, allowing autonomous mobile robots to use the map for path planning and obstacle avoidance.

The results demonstrate several key advantages of using overhead infrastructure cameras for occupancy grid mapping:

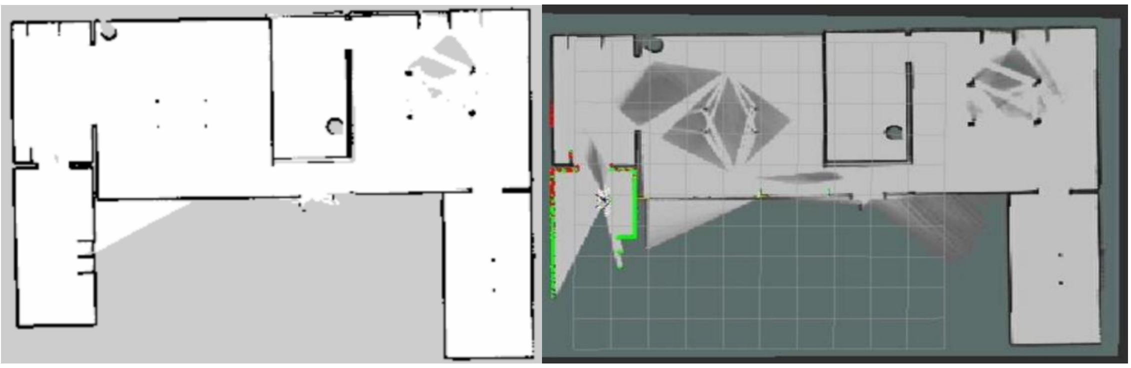
- Overhead cameras provide a comprehensive top-down view of the environment, capturing more complete information compared to ground-level sensors.
- Real-time updates to the occupancy grid map enable the system to adapt to changes in dynamic environments.

- Using overhead cameras can be more cost-effective than equipping autonomous mobile robots with expensive sensors.
- The occupancy grid map enables better path planning and coordination among multiple robots operating in the same environment.



However, the research also identified some limitations and challenges, such as occlusion and blind spots, sensitivity to lighting conditions, computational complexity, and the need for accurate camera calibration.

Overall, the results show that using overhead infrastructure cameras to create a 2D occupancy grid map in a Gazebo and ROS2 environment is a viable approach for autonomous mobile robot navigation in complex, dynamic environments. The occupancy grid map provides a comprehensive representation of the environment that can be effectively utilized by the ROS2 navigation stack for safe and efficient robot navigation.



Chapter 7

Conclusion

The research presented in this report demonstrates the feasibility and effectiveness of using overhead infrastructure cameras to create a 2D occupancy grid map for autonomous mobile robot navigation in a simulated environment. By leveraging the capabilities of Gazebo and ROS2, the researchers were able to develop a comprehensive and adaptive environment representation that can be effectively utilized by the ROS2 navigation stack.

The key findings of this research include the use of overhead infrastructure cameras, which provide a comprehensive top-down view of the environment, capturing more complete information compared to ground-level sensors. This approach offers several advantages, such as the ability to update the occupancy grid map in real-time, enabling the system to adapt to changes in dynamic environments and ensure safe and efficient robot navigation. Additionally, the occupancy grid map can be seamlessly integrated with the ROS2 navigation stack through a custom costmap plugin, allowing autonomous mobile robots to use the map for path planning and obstacle avoidance.

However, the research also identified some limitations and challenges associated with this approach. These include issues such as occlusion and blind spots, sensitivity to lighting conditions, computational complexity, and the need for accurate camera calibration. Despite these challenges, the results of this research suggest that using overhead infrastructure cameras to create a 2D occupancy grid map in a Gazebo and ROS2 environment is a viable approach for autonomous mobile robot navigation in complex, dynamic environments.

Overall, the occupancy grid map provides a comprehensive representation of the environment that can be effectively utilized by the ROS2 navigation stack for safe and efficient robot navigation. Future work could focus on improving the accuracy and robustness of the occupancy grid mapping algorithm, as well as exploring the use of additional sensors and data sources to further enhance the environment representation and navigation capabilities of autonomous mobile robots.