Creating 2D Occupancy Grid Map using overhead infrastructure cameras

SIDDHANT SANTOSH SAWANT

VIJVAL NARAYANA

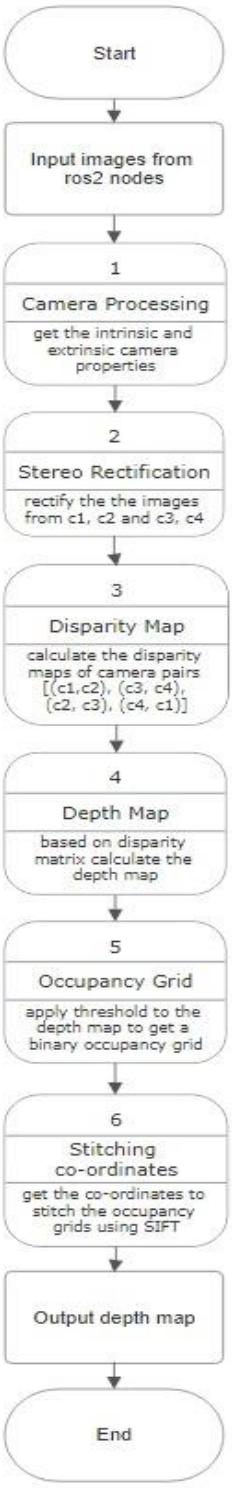
VIVAAN SINGH

MANIPAL INSTITUTE OF TECHNOLOGY

Abstract

Creating a 2D Occupancy Grid Map using overhead infrastructure cameras leverages ROS and Gazebo to capture images of a virtual environment, addressing key challenges in real-time environmental mapping for autonomous systems. Utilizing the Robot Operating System (ROS) and Gazebo simulation, virtual environments are generated to mimic real-world scenarios. The process involves capturing overhead images from virtual cameras, which are then processed to construct a 2D occupancy grid map. This map represents spatial occupancy information, with each cell indicating the probability of being occupied by an object. The approach tackles issues such as varying lighting conditions and occlusions within a controlled simulation environment, ensuring robust data collection and analysis. Advanced computer vision techniques and efficient algorithms for object detection and tracking are employed to create accurate and reliable occupancy maps. This methodology enhances situational awareness and navigation capabilities for autonomous systems , contributing to safer and more efficient operation in dynamic environments.

Approach


Start
Input images from ros2 nodes
1
Camera Processing
get the intrinsic and extrinsic camera properties
2
Stereo Rectification
rectify the the images from c1, c2 and c3, c4
3
Disparity Map
calculate the disparity maps of camera pairs [(c1,c2), (c3, c4), (c2, c3), (c4, c1)]
4
Depth Map
based on disparity matrix calculate the depth map
5
Occupancy Grid
apply threshold to the depth map to get a binary occupancy grid
6
Stitching co-ordinates
get the co-ordinates to stitch the occupancy grids using SIFT
Output depth map
End

Novelty

This project showcases a novel approach to grid mapping through the use of overhead cameras combined with depth mapping via stereo rectification and advanced image stitching techniques using OpenCV. The distinctiveness of this project can be summarized in the following key points:

1. Utilization of Overhead Cameras:
Employing overhead cameras provides a unique and comprehensive perspective for grid mapping. This vantage point is particularly beneficial for monitoring large areas unobstructed, ideal for applications in surveillance, robotics, and autonomous navigation.

2. Stereo Rectification for Depth Mapping:
By leveraging stereo rectification, the project achieves accurate depth mapping. This technique enhances the spatial understanding of the environment, providing critical 3D information necessary for precise grid mapping.

3. Robust Image Stitching with OpenCV:
The integration of OpenCV's powerful image stitching capabilities ensures seamless merging of multiple camera feeds. This results in a unified and continuous map, essential for real-time applications and comprehensive environmental monitoring.

4. Real-time Depth Information:
The system is designed to generate real-time depth information, facilitating dynamic and responsive mapping. This real-time capability is crucial for applications that require immediate updates and quick decision-making.

5. Scalability and Adaptability:
The methodology is both scalable and adaptable, allowing it to be applied to various environments and scenarios. Its flexibility ensures that the system can manage different sizes and complexities of mapping areas, making it suitable for a wide range of applications.

6. Enhanced Environmental Perception:
Combining overhead perspectives with depth mapping provides a richer understanding of the environment. This multi-dimensional approach improves context awareness, aiding in better navigation and interaction with the mapped area.

This project represents a significant advancement in grid mapping technology by integrating stereo rectification and OpenCV-based image stitching. The innovative use of overhead cameras and real-time depth mapping opens new possibilities for accurate and comprehensive environmental mapping and monitoring.

Methodology

The methodology for generating and stitching depth maps from multiple cameras involves a structured approach that ensures accurate depth perception and seamless integration of multiple depth maps. Here's the step-by-step methodology:

1. Image Acquisition

Objective: Capture high-resolution images from multiple cameras positioned at different viewpoints.

Steps:

- Use OpenCV's cv2.imread() to read images from specified file paths.
- Ensure that all images are captured under similar lighting conditions and are in focus.

2. Camera Calibration and Parameter Definition

Objective: Define intrinsic and extrinsic parameters for each camera to facilitate accurate depth computation and image rectification.

Steps:

- **Intrinsic Parameters:** Calculate or provide camera matrix (focal length, principal point) and assume zero distortion coefficients for simplicity.
- **Extrinsic Parameters:** Define rotation matrices and translation vectors to represent the orientation and position of each camera in the 3D space.

3. Stereo Rectification

Objective: Align image pairs such that corresponding points are on the same horizontal lines (epipolar lines), simplifying the depth computation process.

Steps:

- Use cv2.stereoRectify() to compute rectification transformations for each camera pair.
- Apply these transformations to the images using cv2.initUndistortRectifyMap() and cv2.remap() to obtain rectified image pairs.

4. Depth Map Computation

Objective: Compute disparity maps for rectified image pairs and convert them to depth maps.

Steps:

- Use cv2.StereoSGBM_create() to create a StereoSGBM matcher for disparity computation.
- Compute disparity maps from the rectified image pairs.
- Convert disparity maps to depth maps using cv2.reprojectImageTo3D().

5. Depth Map Stitching

Objective: Combine depth maps from different camera pairs to form a single, continuous depth representation.

Steps:

- Define overlapping regions between the depth maps to ensure seamless stitching.

- Combine depth maps using pixel-wise minimum values in the overlapping regions to avoid abrupt transitions.
- Adjust depth values to account for alignment and ensure continuity.

6. Occupancy Grid Creation
Objective: Create a binary occupancy grid to identify and visualize obstacles within a specified distance from the cameras.
Steps:

- Threshold the combined depth map to mark regions within a certain distance (e.g., obstacles) as black (0) and free space as white (255).
- Use this binary occupancy grid to visualize obstacles and free space in the observed area.

Detailed Workflow:

1. Capture Images: Images are acquired from four cameras (C1, C2, C3, C4) using OpenCV's cv2.imread().

2. Define Camera Parameters:

- **Intrinsic parameters:** Camera matrix (focal length, principal point).
- **Extrinsic parameters:** Rotation matrices and translation vectors for each camera.

3. Stereo Rectification:

- Compute rectification transformations using cv2.stereoRectify().
- Apply transformations to obtain rectified images using cv2.initUndistortRectifyMap() and cv2.remap().

4. Compute Depth Maps:

- Use StereoSGBM to compute disparity maps.
- Convert disparity maps to depth maps using cv2.reprojectImageTo3D().

5. Stitch Depth Maps:

- Define overlapping regions and combine depth maps using pixel-wise minimum values.
- Adjust depth values to ensure smooth transitions between combined maps.

6. Create Occupancy Grid:

- Threshold the combined depth map to create a binary occupancy grid.
- Visualize obstacles (black) and free space (white).

This methodology ensures accurate depth perception and seamless integration of depth maps from multiple camera inputs, making it suitable for various applications such as robotics, autonomous navigation, and 3D scene reconstruction.

Pros & Cons of Stereo Rectification

Pros

- **Alignment of Images**:

  Stereo rectification aligns the left and right images into a common plane, simplifying the search for corresponding points. This alignment makes it easier to perform depth estimation and disparity calculations.

- **Reduced Computational Complexity**:

  By rectifying the images, the search for matching points is limited to horizontal lines (epipolar lines), which significantly reduces the computational complexity compared to searching across the entire image.

- **Improved Accuracy**:

  Rectification reduces geometric distortions, improving the accuracy of disparity estimation and subsequent depth mapping.

- **Simplified 3D Reconstruction**:

  With rectified images, the disparity map can be directly converted to a depth map, making the 3D reconstruction process more straightforward.

Cons

Pre-processing Requirement:

- **Computational Overhead:** The process of rectification itself can be computationally intensive, especially for high-resolution images.

- **Pre-processing Time:** Adds an extra step in the image processing pipeline, potentially increasing the overall processing time.

Dependency on Camera Calibration:

- **Calibration Accuracy:** Requires accurate camera calibration parameters to achieve good rectification. Errors in calibration can lead to poor rectification results.

- **Complex Setup:** Calibration of stereo camera systems can be complex and time-consuming.

Limited by Baseline and Field of View:

- **Baseline Distance:** Works best when the cameras have an appropriate baseline distance; too wide or too narrow baselines can affect performance.

- **Field of View Alignment:** Requires careful alignment of the cameras' fields of view, which can be challenging in some setups.

Handling Distortions:

- **Distortion Correction:** Involves correcting for lens distortions, which can be difficult for lenses with significant distortion or non-linearities.

- **Image Quality:** Rectification can sometimes introduce artifacts or degrade the image quality, affecting subsequent processing steps.

Pros & Cons of Depth Mapping

Pros

Enhanced Perception:

- **3D Information:** Provides valuable 3D spatial information that aids in understanding the environment's structure.

- **Object Detection:** Improves the accuracy of object detection and segmentation by incorporating depth data.

Improved Navigation:

- **Obstacle Avoidance:** Essential for autonomous vehicles and robots to detect and avoid obstacles.

- **Path Planning:** Aids in efficient path planning by providing information about free and occupied spaces.

Augmented Reality:

- **Realistic Interaction:** Enhances augmented reality applications by enabling realistic interactions between virtual and real objects.
- **Scene Understanding:** Facilitates better scene understanding and alignment of virtual objects in the physical world.

3D Reconstruction:

- **Detailed Models:** Helps in creating detailed 3D models of objects and environments.
- **Accuracy:** Provides more accurate and realistic reconstructions compared to using 2D images alone.

Cons

Computational Complexity:

- **Processing Power:** Generating and processing depth maps can be computationally intensive, requiring significant processing power.
- **Real-Time Challenges:** Achieving real-time performance can be challenging, especially for high-resolution depth maps.

Sensor Limitations:

- **Environmental Sensitivity:** Depth sensors can be affected by environmental conditions such as lighting, reflections, and occlusions.
- **Range Limitations:** Limited effective range and accuracy, especially for distant objects or in outdoor environments.

Data Quality:

- **Noise:** Depth maps can be noisy, with errors and artifacts that need to be filtered out.
- **Resolution:** Often have lower resolution compared to RGB images, which can affect the detail and accuracy of the depth information.

Integration Challenges:

- **Multi-Sensor Fusion:** Combining depth maps with other sensor data (e.g., RGB images) can be complex and require sophisticated algorithms.

Comparative Overview with respect to SLAM

Real-time Performance: Stereo depth mapping can provide real-time depth information more efficiently than SLAM, which may require more computational resources for real-time processing.

Complexity: SLAM is generally more complex due to the need for simultaneous localization and environment mapping, whereas stereo rectification focuses on depth perception alone.

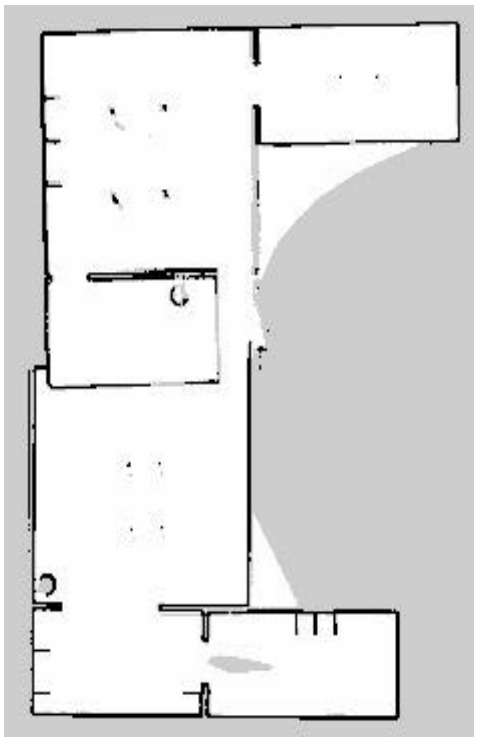
Use Cases: Stereo depth mapping is ideal for applications needing immediate and accurate depth information without extensive environmental mapping. In contrast, SLAM is essential for applications requiring comprehensive environment understanding and navigation.

Environment Dynamics: SLAM is better suited for dynamic environments where the layout can change, while stereo depth mapping is more suitable for static or controlled environments.

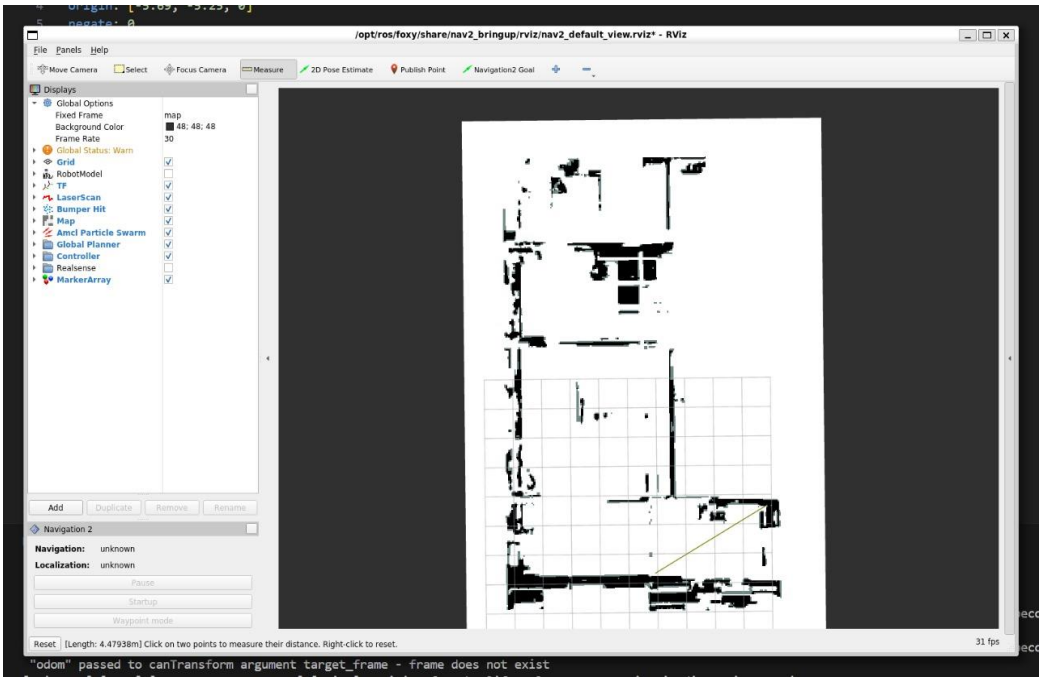
Both technologies have their strengths and are often used complementarily in advanced robotics and autonomous systems to leverage the benefits of both depth perception and comprehensive environment mapping.

Result

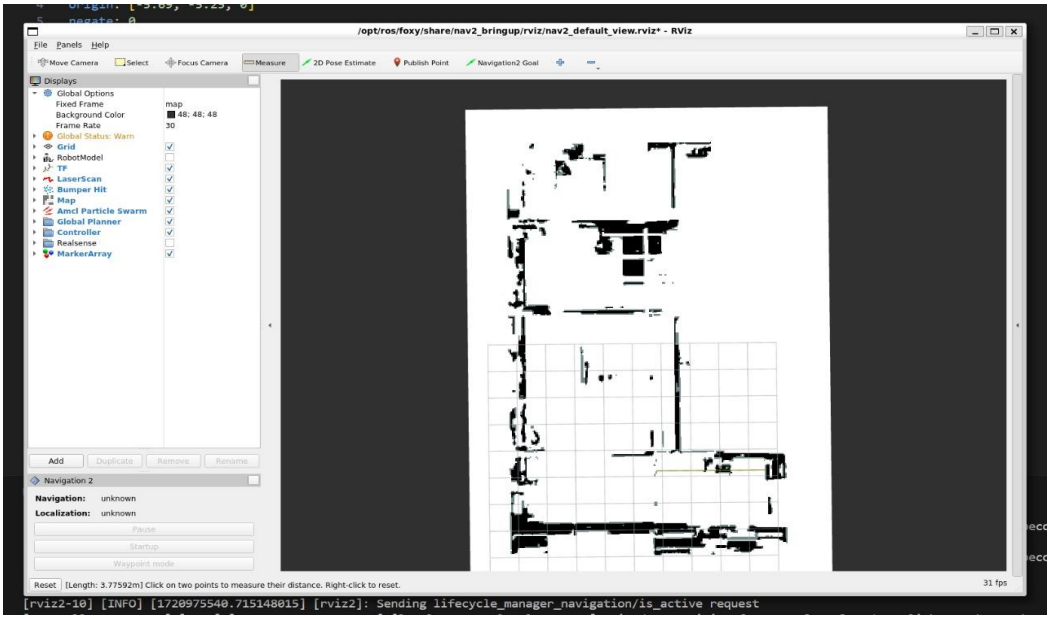
Fused map of the environment, detailed dimensions and Error Estimates



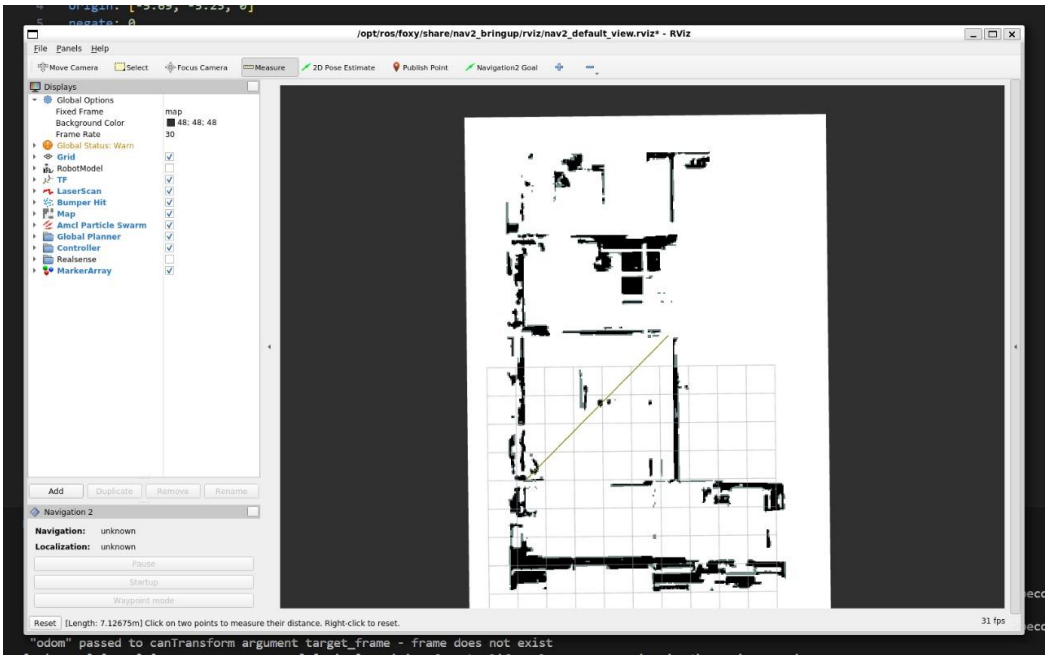
Length- 4.60234m , Ground Truth Length-4.21862 , Error- 8.33%



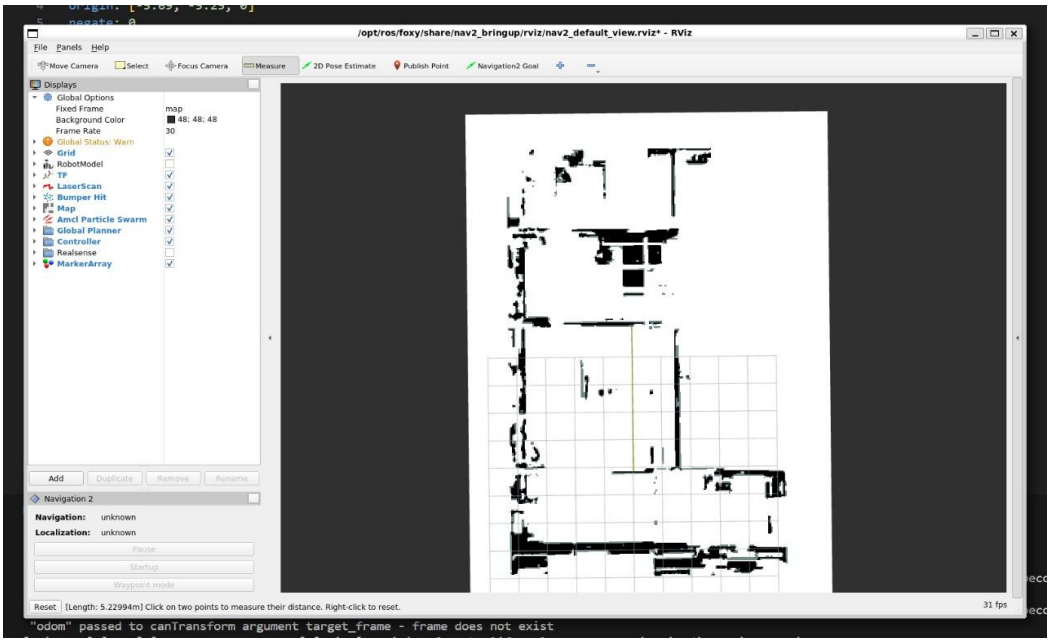
Length- 4.47938m , Ground Truth Length-5.09171m , Error- 12.03%



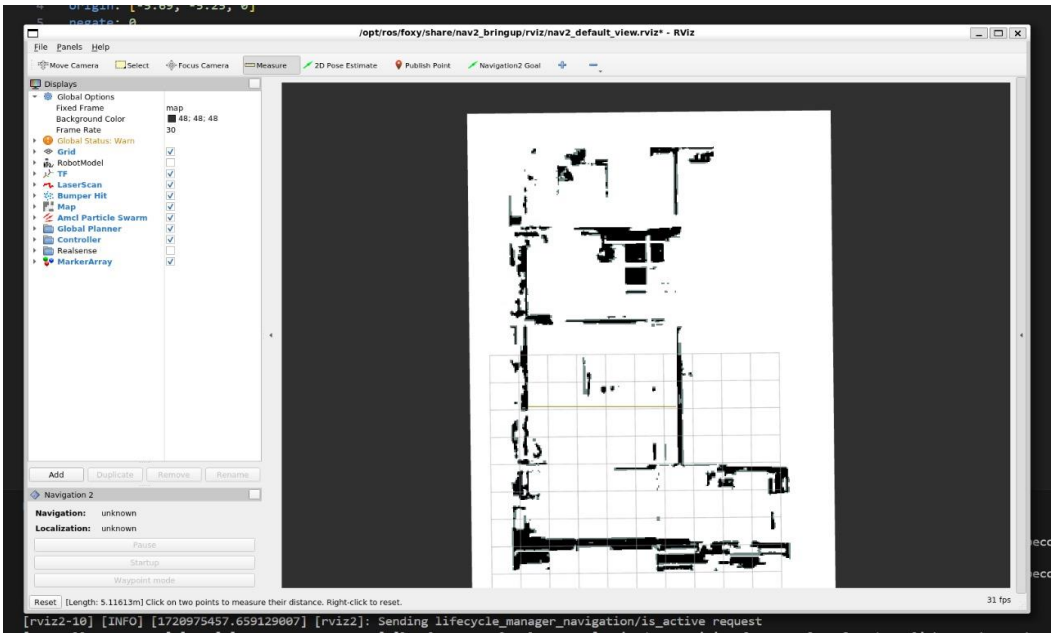
Length- 3.77592m , Ground Truth Length-4.22567m , Error- 10.65%



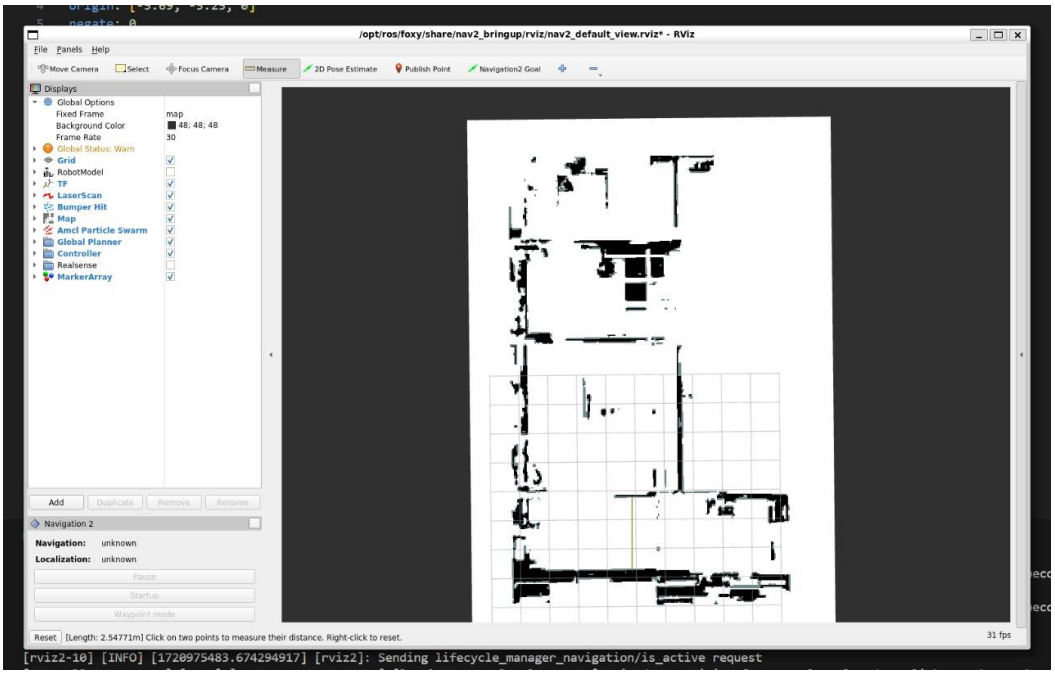
Length- 7.12675m , Ground Truth Length-7.20996m , Error- 1.16%



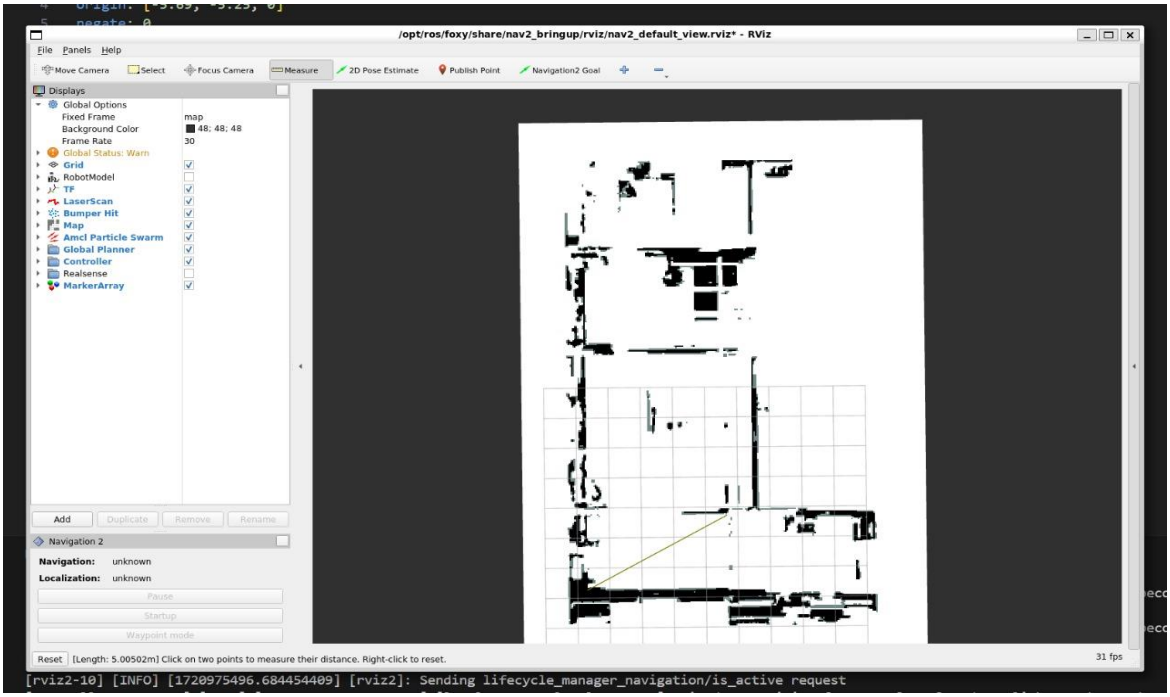
Length- 5.22994m , Ground Truth Length-4.94927m , Error- 5.37%



Length- 5.11613m , Ground Truth Length-5.24478m , Error- 2.46%



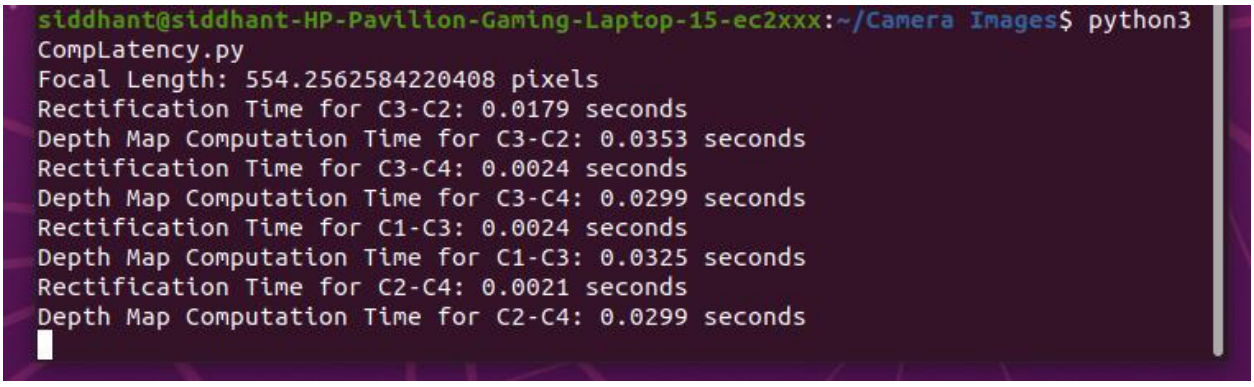
Length- 2.54771m , Ground Truth Length-2.12293m , Error- 16.68%



Length- 5.00502m , Ground Truth Length-4.62502m , Error- 7.6%

Computational Latency of the mapping algorithm

Total Rectification Time : 0.0248 seconds
Total Depth Map Computation Time: 0.1276 seconds
Total time : 0.1524 seconds



Source Code

[GitHub - SiddhantSawant24/Intel-Unnati: Creating 2D Occupancy Grid Map using overhead infrastructure cameras]

Learning

Camera Calibration:

- Understanding the importance of precise camera calibration to accurately map 3D world coordinates to 2D image coordinates.
- The challenges and techniques in calibrating multiple cameras to ensure a consistent and accurate occupancy grid.

Image Processing Techniques:

- The application of various image processing algorithms for detecting objects and distinguishing them from the background.
- Techniques such as background subtraction, edge detection, and morphological operations proved crucial for accurate object detection.

Object Detection and Tracking:

- Implementation of machine learning models for detecting and classifying objects in the camera's field of view.
- The importance of real-time object tracking algorithms to maintain accurate and continuous tracking of objects as they move through the scene.

Grid Mapping:

- Methods to translate object positions into a 2D grid map representation.
- Handling occlusions and ensuring that the grid map accurately reflects the current state of occupancy despite visual obstructions.

Data Fusion:

- Integrating data from multiple cameras to create a unified occupancy grid.
- Techniques for synchronizing and merging data from different viewpoints to ensure a comprehensive and accurate map.

Conclusion

The project on creating a 2D occupancy grid map using overhead infrastructure cameras demonstrated the potential of leveraging advanced image processing and machine learning techniques to achieve accurate and real-time occupancy mapping.

Despite challenges such as lighting variations, environmental factors, and the need for computational efficiency, the iterative development approach allowed the team to address these issues systematically. The project highlighted the importance of interdisciplinary collaboration, adaptive algorithms, and continuous testing in real-world scenarios.

The outcomes of this project pave the way for future enhancements and scalability, with promising directions including the integration of additional sensor types and advanced machine learning models