Group C 10

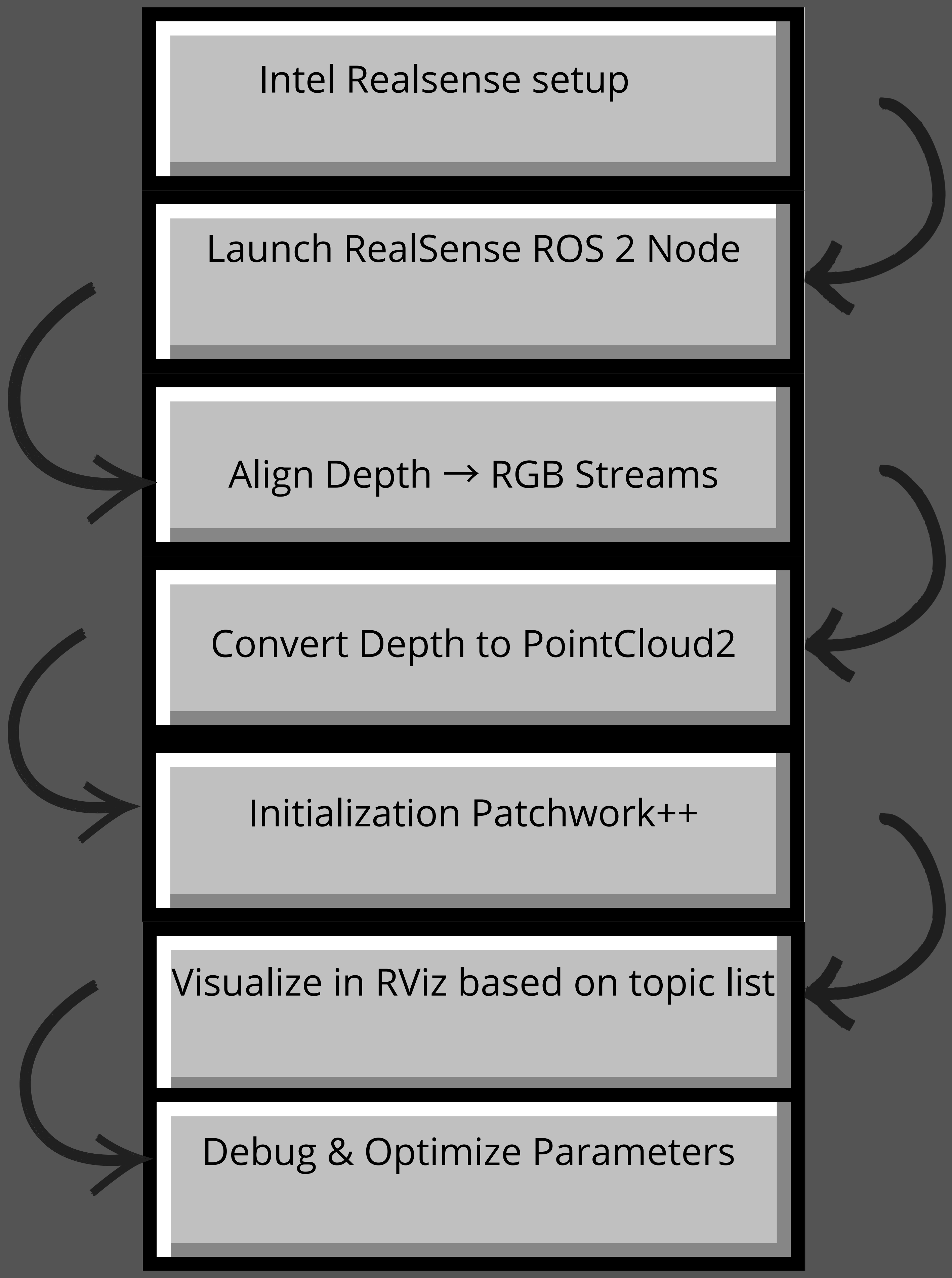
Automated Road Level Checker

Aishwarya Manoj - CB.SC.U4AIE23211
Aswathi Ranjith - CB.SC.U4AIE23215
Sanmita GJ - CB.SC.U4AIE23252
Lakshmi Ridhanya - CB.SC.U4AIE23270

Introduction

- Ground segmentation and point cloud processing are essential for enabling autonomous mobile robots to navigate complex environments.
- The process involves distinguishing ground surfaces from obstacles using 3D spatial data collected by depth sensors.
- Accurate segmentation enhances the robot's ability to perform tasks such as path planning, obstacle avoidance, and terrain analysis.
- Efficient interpretation of point cloud data supports safer and more reliable autonomous navigation.

Methodology


Intel Realsense setup
Launch RealSense ROS 2 Node
Align Depth → RGB Streams
Convert Depth to PointCloud2
Initialization Patchwork++
Visualize in RViz based on topic list
Debug & Optimize Parameters

Working Principle of Intel RealSense D435i

- **Stereo Vision**: The camera has two infrared sensors (left and right) that capture images simultaneously. The camera can calculate the depth of objects in the scene by comparing the differences (disparity) between these two images.
- I**R Projector**: The IR projector emits a pattern of infrared dots onto the scene. This pattern helps the camera to detect depth in low-texture environments where stereo matching might be difficult.
- **RGB Sensor**: Besides depth sensing, the D435i has an RGB sensor that captures color images. The depth and RGB data are aligned to provide a combined RGB-D image.
- **IMU (Inertial Measurement Unit)**: The D435i also includes an IMU, which provides accelerometer and gyroscope data. This is useful for applications that require motion tracking or stabilization.

Camera Model and Mathematics

The camera model describes how 3D points in the world are projected onto the 2D image plane. The pinhole camera model is commonly used to describe this process.

- The intrinsic parameters of the camera define how the 3D world is mapped to the 2D image plane. These parameters are represented by the intrinsic matrix K:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where,

- fx,fy represents the focal lengths in pixels (related to the camera's field of view).
- cx,cy represents the principal point (the optical center of the image in pixels).

Depth Calculation

The depth Z of a point in the scene is calculated using the disparity between the left and right infrared images. The disparity d is the difference in the position of a point in the two images. The depth Z is given by:

$$Z = \frac{f_x \cdot B}{d}$$

where,

- fx: Focal length in pixels (from the intrinsic matrix).
- B: Baseline (distance between the left and right infrared sensors).
- d: Disparity (difference in pixel location of the same point in the left and right images).

Depth and RGB Alignment

- The D435i aligns the depth data with the RGB image using the extrinsic transformation between the depth sensor and the RGB sensor.

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

- This transformation is represented by a matrix Tdepthtocolor, which maps points from the depth sensor's coordinate system to the RGB sensor's coordinate system.
- The transformation is applied as follows:

$$P_{color} = P_{depthtocolor} \cdot P_{depth}$$

- Pdepth: 3D point in the depth sensor's coordinate system.
- Pcolor: Corresponding 3D point in the RGB sensor's coordinate system.

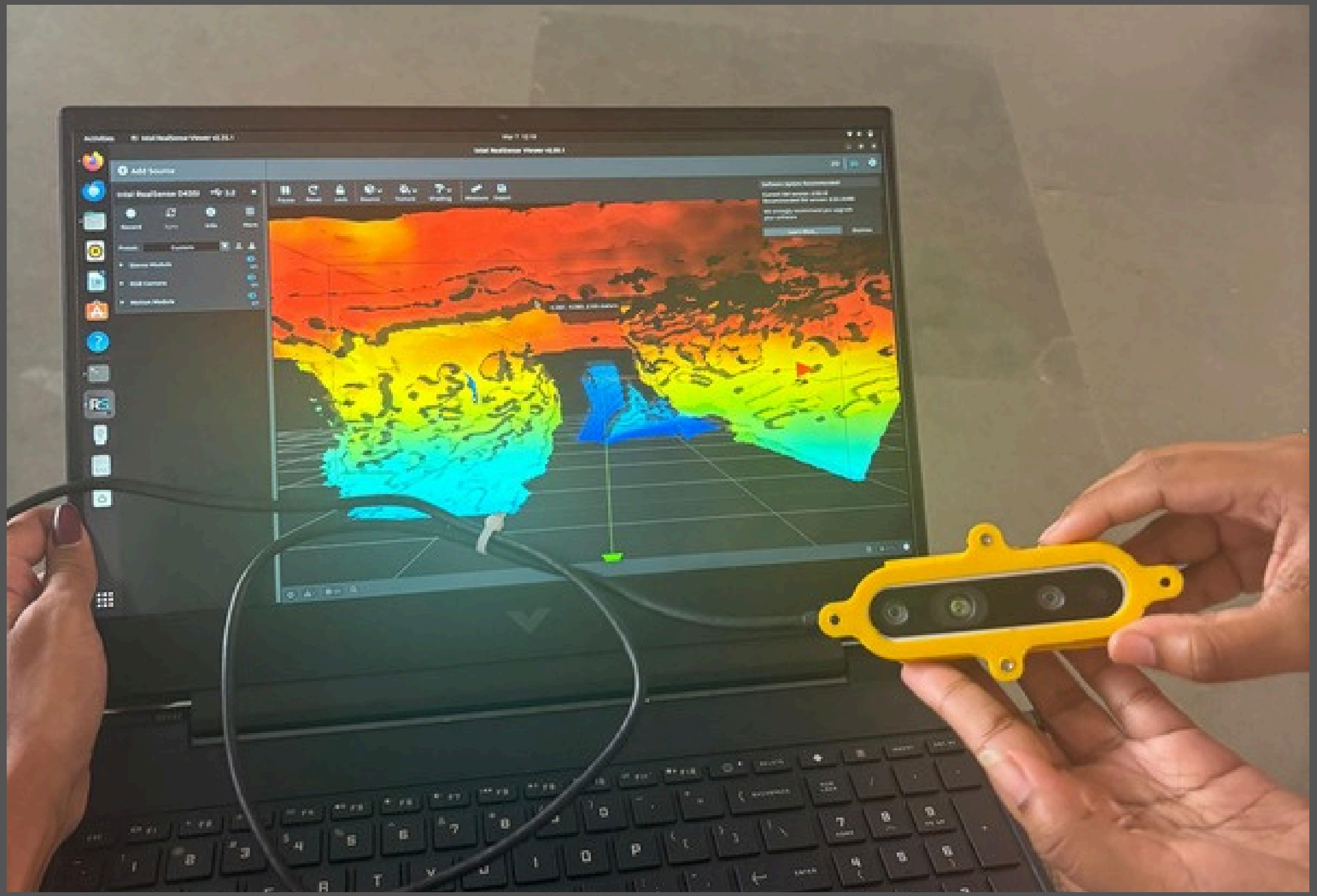
Point Cloud Generation

- The depth image can be converted into a 3D point cloud using the camera's intrinsic parameters.
- Each pixel in the depth image corresponds to a 3D point (X,Y,Z)in the camera's coordinate system. The 3D coordinates are calculated as:

$$X = \frac{(u - c_x) \cdot Z}{f_x}, \quad Y = \frac{(v - c_y) \cdot Z}{f_y}, \quad Z = \mathrm{Depth}(u, v)$$

where,

- (*u*,*v*): Pixel coordinates in the depth image.
- Depth(u,v): Depth value at pixel (*u*,*v*).



ROS 2 Integration & TF Transformations

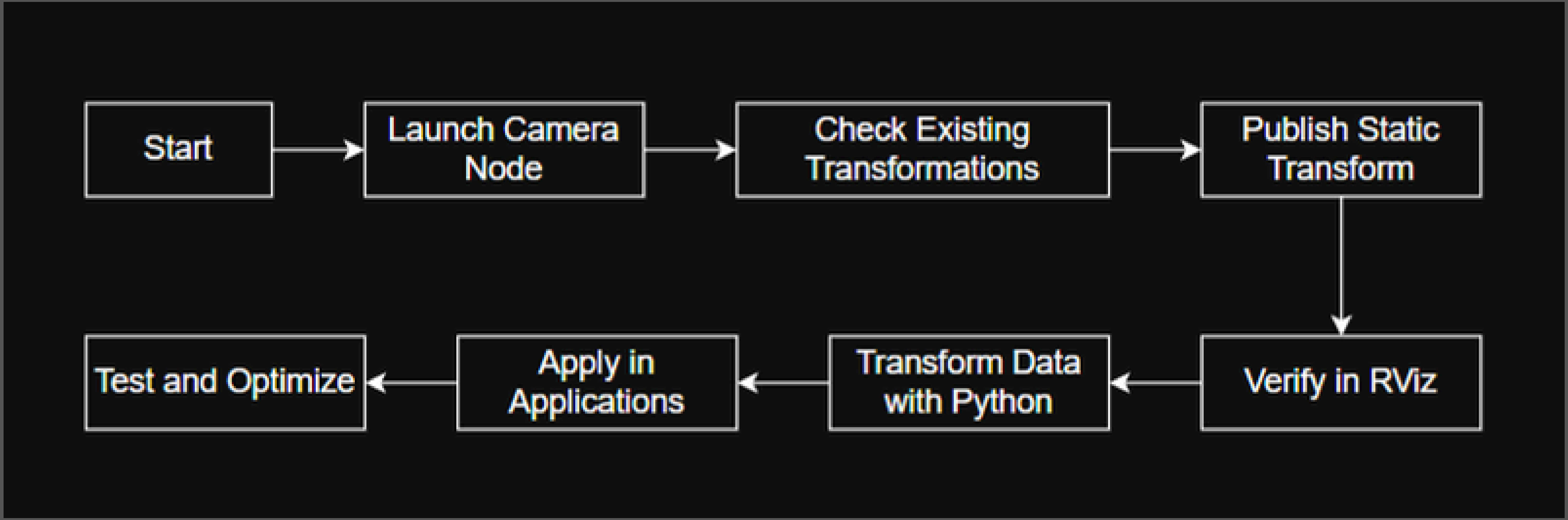
What is ROS 2?

- Robot Operating System 2 (ROS 2) is a middleware framework for real-time robotics applications.
- Provides sensor data collection, transformation, and visualization capabilities.
- Enables multi-node communication, making it ideal for integrating Intel RealSense with ROS 2.

Why Use ROS 2?

- Supports real-time data processing for robotics.
- Flexible transformations using tf2, making it easier to align sensor frames.
- Allows sensor fusion by integrating multiple data sources into a common coordinate system.

Workflow : TF Transformations in ROS 2



Mathematics behind TF Transformations

Homogeneous Transformation Matrix

- In 3D space, transformations (rotations + translations) are represented using homogeneous transformation matrices:

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

where:

- R (3×3 matrix) → Defines rotation in 3D space.
- t (3×1 vector) → Defines translation (shift in position).
- 0 (1×3 vector) → Represents the homogeneous coordinate system.
- 1 (scalar) → Maintains the transformation in a 4×4 format.
- Using this matrix, any 3D point can be transformed from one frame to another.

ROS 2 TF Transformation Formula

To transform a 3D point from frame A to frame B, use:

$$P_B = T_{A \to B} \cdot P_A$$

where $T_{A \to B}$ is the transformation matrix from Frame A to Frame B.

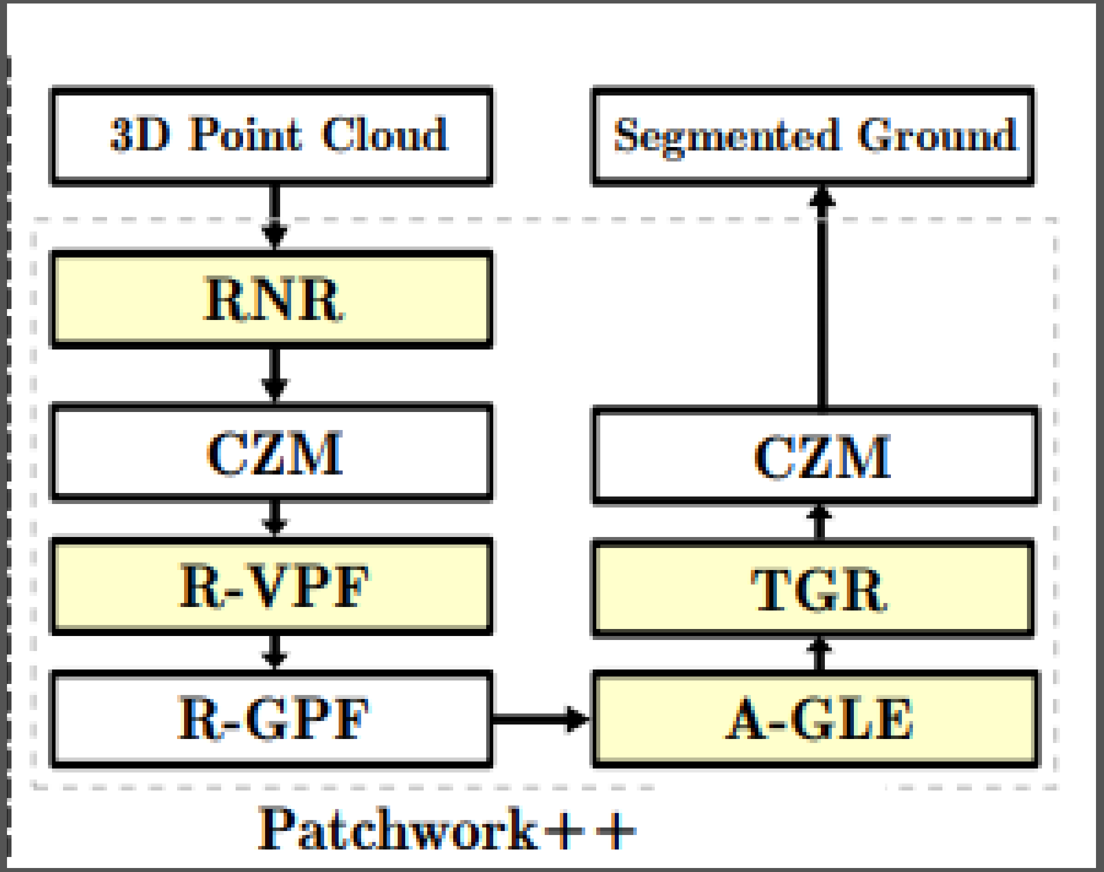
Applying Transformations in ROS 2

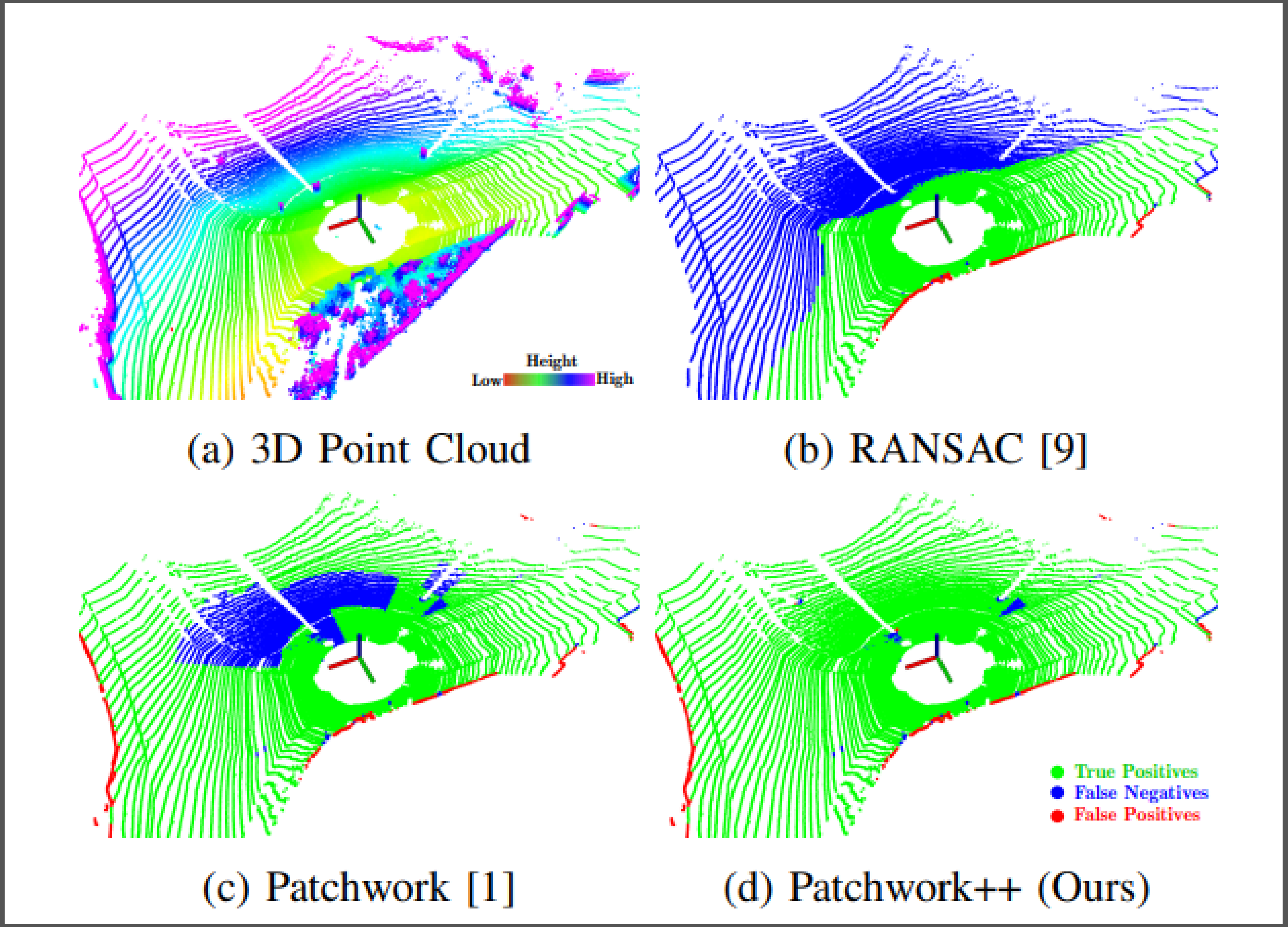
- In ROS 2, transformations between coordinate frames are essential for ensuring accurate sensor data interpretation.
- Transforming a 3D point from the Camera Frame to the Robot Base Frame:

$$P_{\text{robot}} = T \cdot P_{\text{camera}}$$

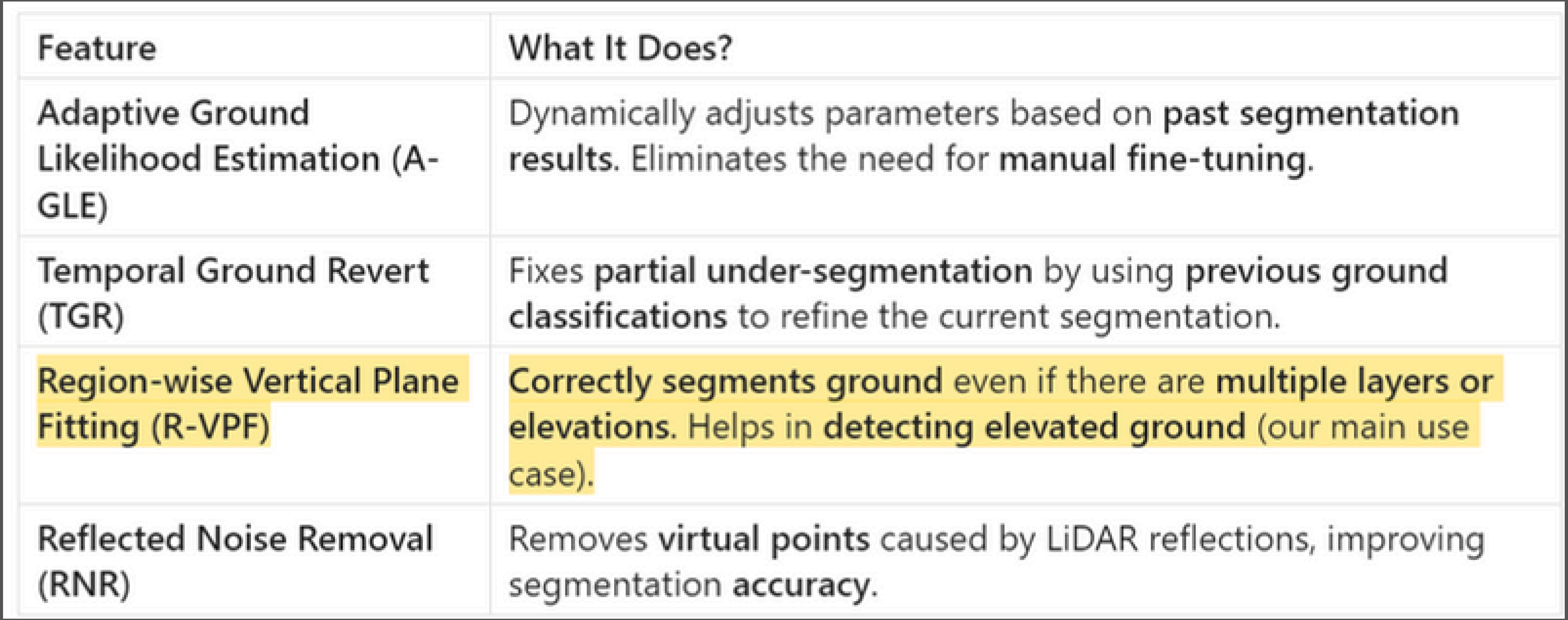
Patchwork++ for Ground Segmentation

Patchwork++ is a robust ground segmentation method designed for 3D LiDAR perception. It improves upon the original Patchwork method by addressing partial under-segmentation issues and handling non-flat ground surfaces more effectively.



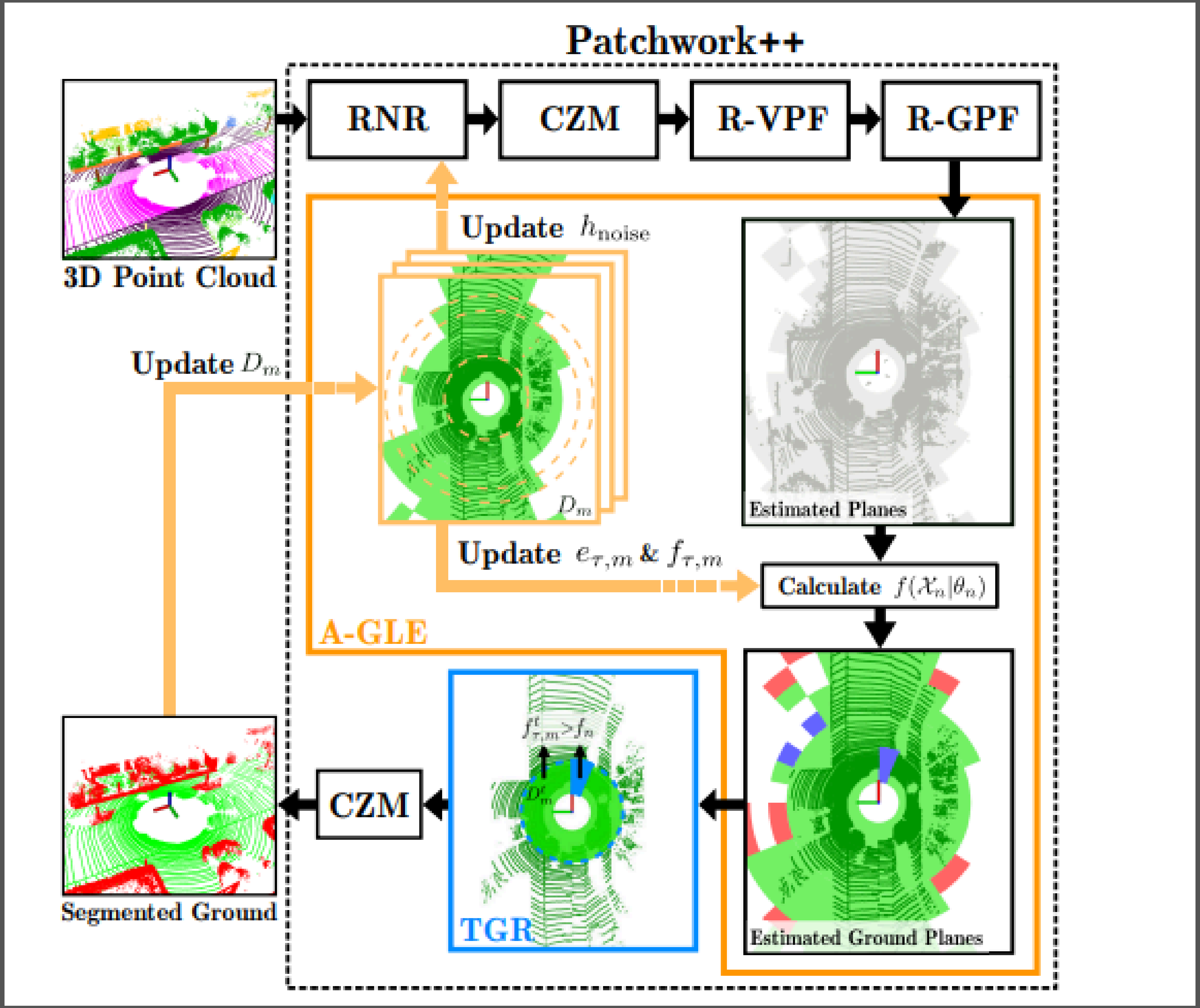


Features



Feature	What It Does?
Adaptive Ground Likelihood Estimation (A-GLE)	Dynamically adjusts parameters based on **past segmentation results**. Eliminates the need for **manual fine-tuning**.
Temporal Ground Revert (TGR)	Fixes **partial under-segmentation** by using **previous ground classifications** to refine the current segmentation.
Region-wise Vertical Plane Fitting (R-VPF)	**Correctly segments ground** even if there are **multiple layers or elevations**. Helps in **detecting elevated ground** (our main use case).
Reflected Noise Removal (RNR)	Removes **virtual points** caused by LiDAR reflections, improving segmentation **accuracy**.

Workflow



$$f_{\tau,m} \leftarrow \mathrm{mean}(F_m) + b_m \cdot \mathrm{stdev}(F_m)$$

This equation is a statistical thresholding method:

- It takes the mean of Fm as a baseline estimate.
- It adds a multiple (bm) of the standard deviation to account for variability.
- This ensures a more adaptive threshold based on the data distribution, rather than a fixed value.

$$e_{\tau,m} \leftarrow \mathrm{mean}(E_m) + a_m \cdot \mathrm{stdev}(E_m)$$

This equation defines a statistical threshold for error estimation:

- The mean of Em represents a baseline estimate of the error.
- The standard deviation accounts for variability, ensuring that the threshold adapts dynamically to different distributions.
- The scaling factor am determines how sensitive the threshold is to variations

Real-Time Visualization in RViz

Why Use RViz?

- Provides 3D visualization of point clouds and sensor data.
- Assists in interactive debugging and validation of sensor outputs.
- Helps in understanding spatial relationships within the robot’s environment.

Methodology

- Launch RViz from the terminal.
- Set the Fixed Frame to base_link for consistent reference.
- Add the PointCloud2 display.
  Select the relevant topic (e.g., /camera/depth/points).
- Adjust visualization properties (e.g., color transformer, size).

Mathematics Behind Point Cloud Visualization

- **Coordinate Transformation:**

Transforms sensor data from its local frame to a common world frame (e.g., base_link or map).

Homogeneous Transformation Matrix:

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

Where:

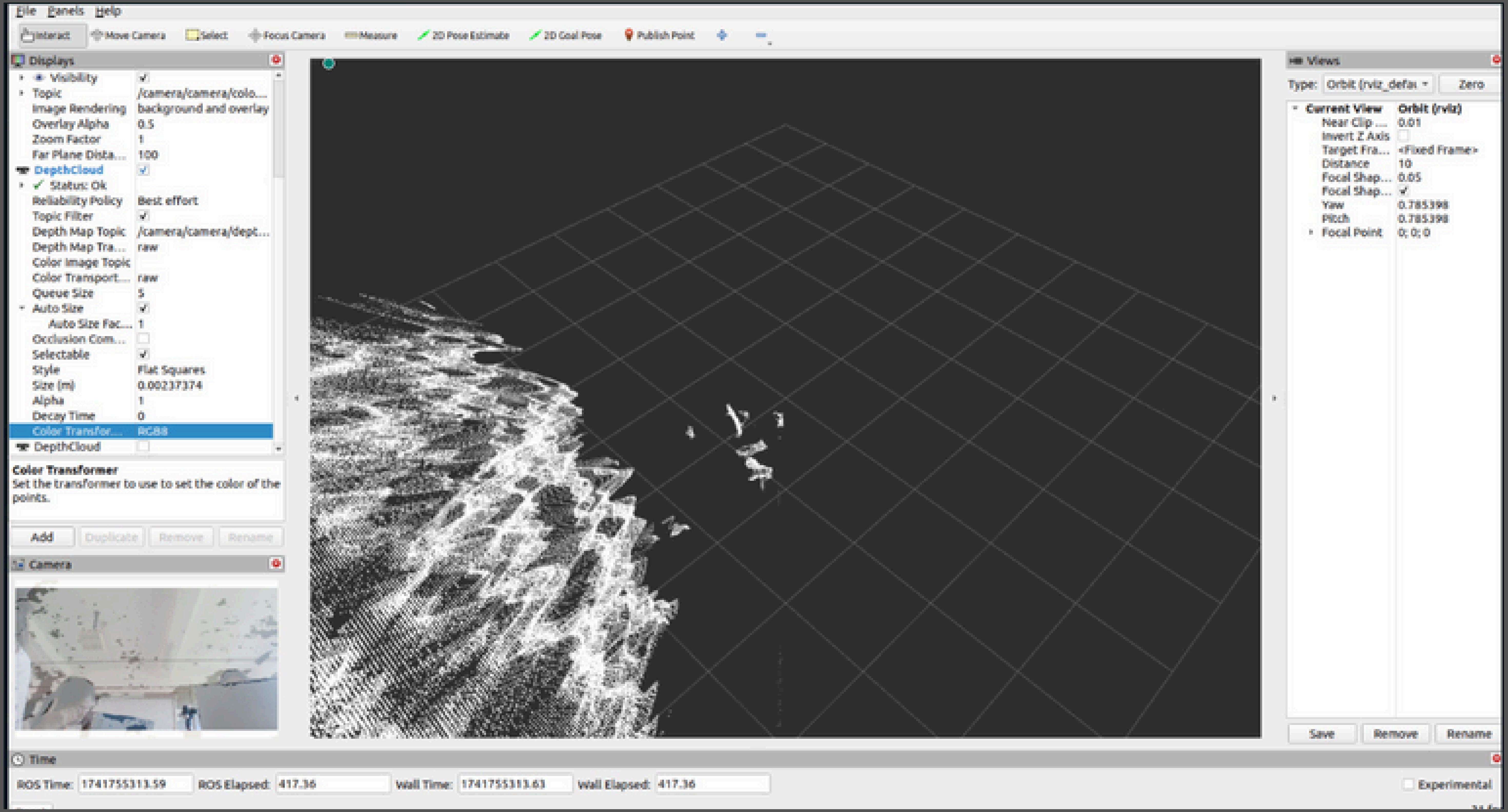
- R is the rotation matrix.
- t is the translation vector.

- Point Cloud Transformation Equation:

$$P_{world} = T \cdot P_{sensor}$$

Converts point coordinates from sensor space to world space for accurate visualization.

OUTPUT



Visibility	✓
Topic	/camera/camera/colo...
Image Rendering	background and overlay
Overlay Alpha	0.5
Zoom Factor	1
Far Plane Dista...	100
DepthCloud	✓
✓ Status: Ok	
Reliability Policy	Best effort
Topic Filter	✓
Depth Map Topic	/camera/camera/dept...
Depth Map Tra...	raw
Color Image Topic	
Color Transport...	raw
Queue Size	5
Auto Size	✓
Auto Size Fac...	1
Occlusion Com...	
Selectable	✓
Style	Flat Squares
Size (m)	0.00237374
Alpha	1
Decay Time	0
Color Transfor...	RGB8
DepthCloud	

Color Transformer
Set the transformer to use to set the color of the points.

Type: Orbit (rviz_defa

Current View	**Orbit (rviz)**
Near Clip ...	0.01
Invert Z Axis	
Target Fra...	<Fixed Frame>
Distance	10
Focal Shap...	0.05
Focal Shap...	✓
Yaw	0.785398
Pitch	0.785398
Focal Point	0; 0; 0

ROS Time: 1741755313.59 ROS Elapsed: 417.36 Wall Time: 1741755313.63 Wall Elapsed: 417.36

Debugging & Optimization

1. Identify the Problem
    - High latency, dropped frames, or high CPU usage during point cloud processing.
2. Debugging Steps
    - Check topic list: ros2 topic list.
    - Inspect topic info: ros2 topic info /camera/depth/points.
    - Monitor frequency: ros2 topic hz /camera/depth/points.
3. Optimize Data Size
    - Compress point cloud data using point_cloud_transport.
    - Publish compressed data to /camera/depth/points/compressed.
4. Adjust Camera Settings
    - Reduce frame rate: ros2 param set depth_frame_rate 15.
    - Lower resolution: ros2 param set depth_width 320.
5. Monitor System Performance
    - Check CPU, memory, and network usage.
    - Ensure stable frame rate and reduced latency.

Future Work & Expansion

Enhancements & Optimization

- Fine-tune Patchwork++ for improved ground segmentation. and object classification using threshold parameters.
- Enhance point cloud accuracy using IMU fusion.
- Optimize ROS 2 nodes for real-time processing.

AI Integration & Automation

- Extend mapping capabilities to larger environments.
- Improve detection of surface irregularities and road conditions.
- Enhance data visualization for better analysis.

Final Goal

- Develop a real-time 3D mapping system for surface detection, object classification, and deployment.