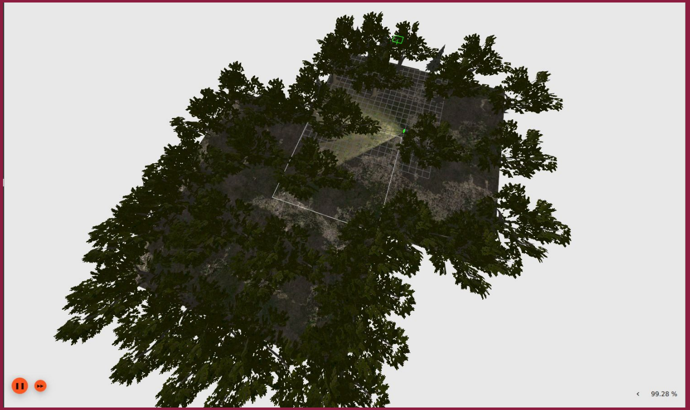
Date: 05.08.2025

RAS 598 – Space Robotics and AI

Project Presentation

Forest Ground Biomass Detection Using ALS Point Cloud



- Ryan Fernandes
- Shyam Kamlesh Ganatra
- Rhutvik Prashant Pachghare
- Vamshikrishna Gadde

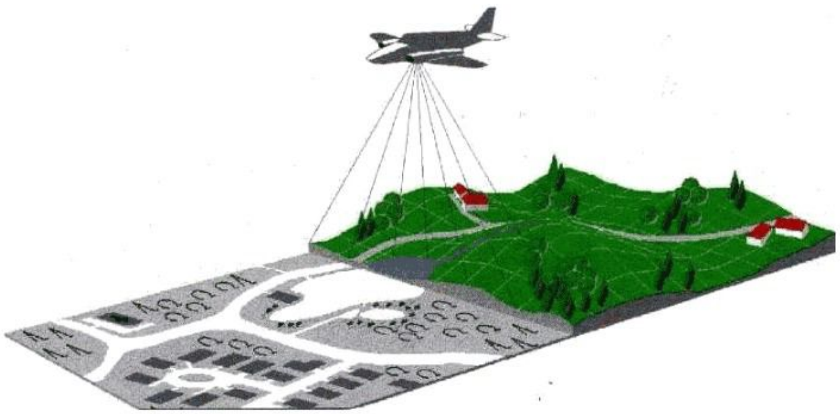
Motivation & Goal

- **Climate context:** Forest Aboveground Biomass (AGB) drives ~30 % of global land carbon, carbon markets need fine-scale maps.
- **Challenge:** Field plots cannot resolve tree-level variability at landscape scale.
- **Opportunity:** High-density ALS + robust allometry → per-tree AGB without felling.
- **Project goal:** Segment every tree across **16.75 km^2** and deliver a GIS layer with predicted AGB.

Project Overview

Estimating Forest Biomass Using ALS

- **Data**: High-resolution 3D forest structure
- **Enables**:
  - Accurate tree height measurement
  - Detailed canopy structure analysis
  - Precise biomass distribution estimation
  - Enhanced data processing
- **Workflow**: From data acquisition to estimating biomass
- **Total Points :** 3.2M

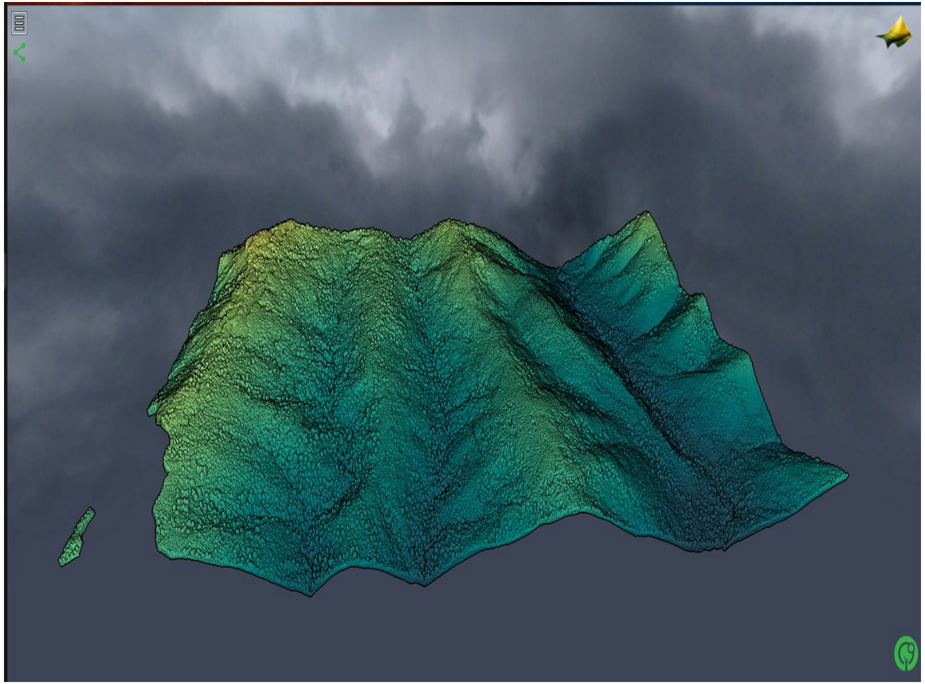


Airborne Laser Scanning (ALS) Data acquisition

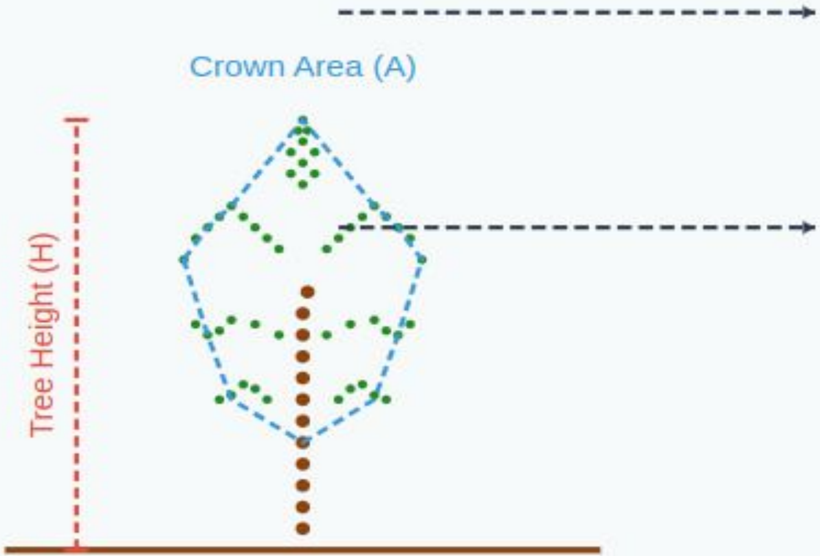
How does the ALS data look ?

📊 ALS Point Cloud Summary:

- X range: 688783.84 to 694028.95
- Y range: 4391504.49 to 4394698.56
- Z range (elevation): 1360.92 to 2181.45
- Area covered: 16.75 km²
- Total points: 18,000,000
- Density: 1.92 points/m²



Tree Parameter Extraction from Point Cloud



Allometric Biomass Formula

Biomass (kg) = a · H^b · A^c

Where:

H = Tree height (m)

A = Crown area (m²)

a = 0.05, b = 2.5, c = 1.0

```
height = z.max() - z.min() # Tree height
crown_points = [(x[i], y[i]) for i in range(len(x))] # XY projection
crown_area = convex_hull(crown_points).area # Crown area
biomass = 0.05 * (height ** 2.5) * (crown_area ** 1.0) # Biomass calculation
```

Tree-Level Biomass Estimation

- Used PDAL filters.cluster for tree segmentation.
- Extracted height, crown area per tree.
- Applied biomass formula for each tree.

The Code Behind the Madness !

2.3 Pipeline Execution (`run_pipeline`)

The main processing pipeline:

1. **Load point cloud data** from LAS file

```
las_path = os.path.expanduser('~/space_robotics_project/data/processed/downsampled_points.las')
las = laspy.read(las_path)
x, y, z = las.x, las.y, las.z
```

2. **Perform tree clustering**

```
clusters = self.fake_clustering(x, y, z, cell_size=3.0)
```

3. **Calculate biomass** for each tree cluster

```
height = np.max(points[:, 2]) - np.min(points[:, 2])
crown = MultiPoint(points[:, :2]).convex_hull
area = crown.area if crown.is_valid else 0.0
biomass = 0.05 * (height ** 2.5) * (area ** 1.0)
```

4. **Create visualization markers** for each tree

```
marker = Marker()
marker.header = Header()
marker.header.frame_id = "map"
marker.id = cluster_id
marker.type = Marker.CYLINDER
marker.action = Marker.ADD
# [marker configuration...]
```

5. **Publish markers** for RViz visualization

```
self.publisher.publish(marker_array)
```

2.4 Tree Clustering (`fake_clustering`)

The current implementation uses a simple grid-based clustering approach:

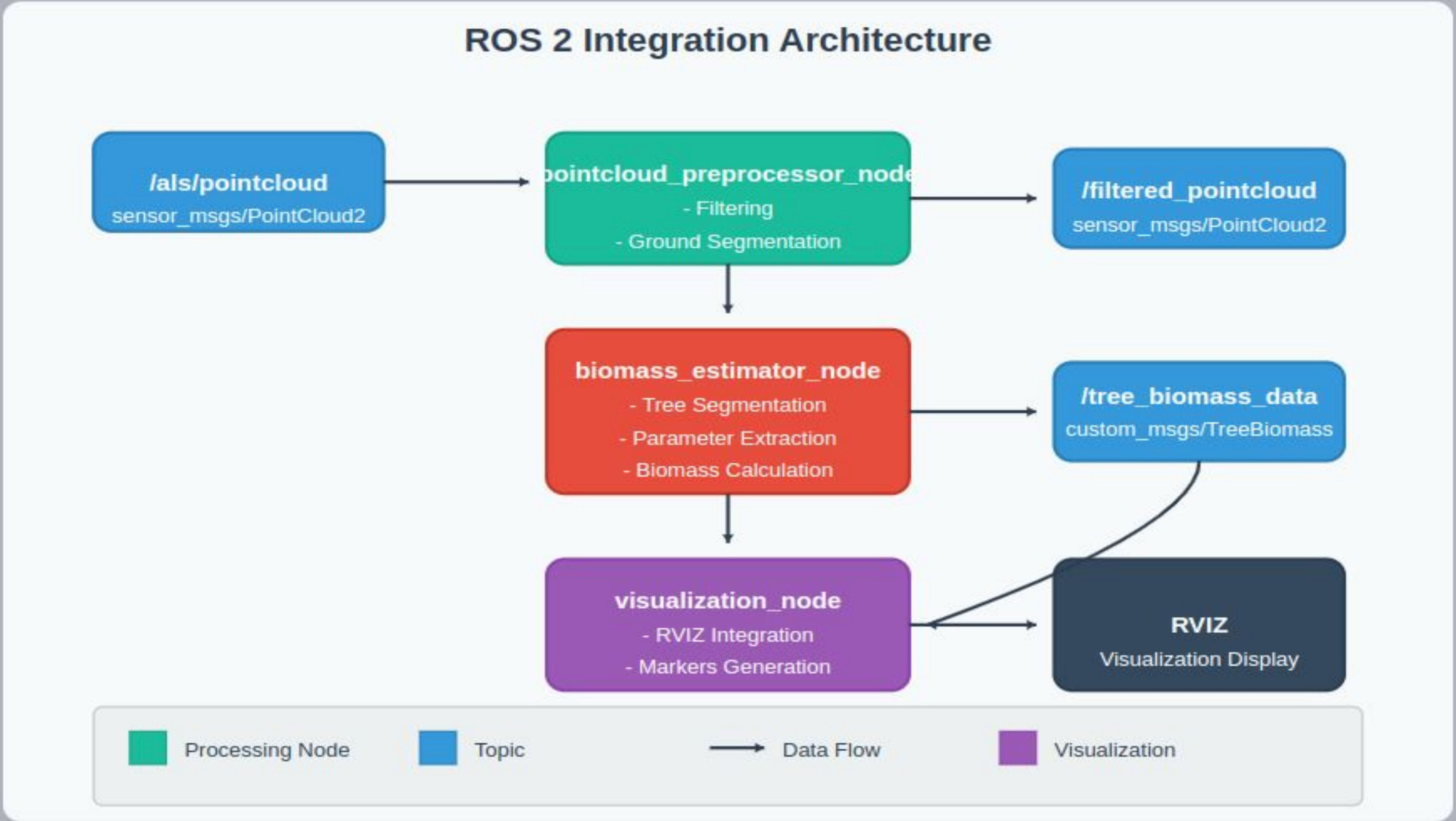
```
def fake_clustering(self, x, y, z, cell_size=5.0, min_points=50):
    from collections import defaultdict
    grid = defaultdict(list)

    # Assign points to grid cells
    for i in range(len(x)):
        key = (int(x[i] // cell_size), int(y[i] // cell_size))
        grid[key].append([x[i], y[i], z[i]])

    # Convert grid cells to clusters
    cluster_count = 0
    clustered = {}

    for i, pts in enumerate(grid.values()):
        if len(pts) >= min_points:
            clustered[i] = np.array(pts)
            cluster_count += 1

    return clustered
```


ROS 2 Integration Architecture
/als/pointcloud
sensor_msgs/PointCloud2
pointcloud_preprocessor_node
- Filtering
- Ground Segmentation
/filtered_pointcloud
sensor_msgs/PointCloud2
biomass_estimator_node
- Tree Segmentation
- Parameter Extraction
- Biomass Calculation
/tree_biomass_data
custom_msgs/TreeBiomass
visualization_node
- RVIZ Integration
- Markers Generation
RVIZ
Visualization Display
Processing Node
Topic
Data Flow
Visualization

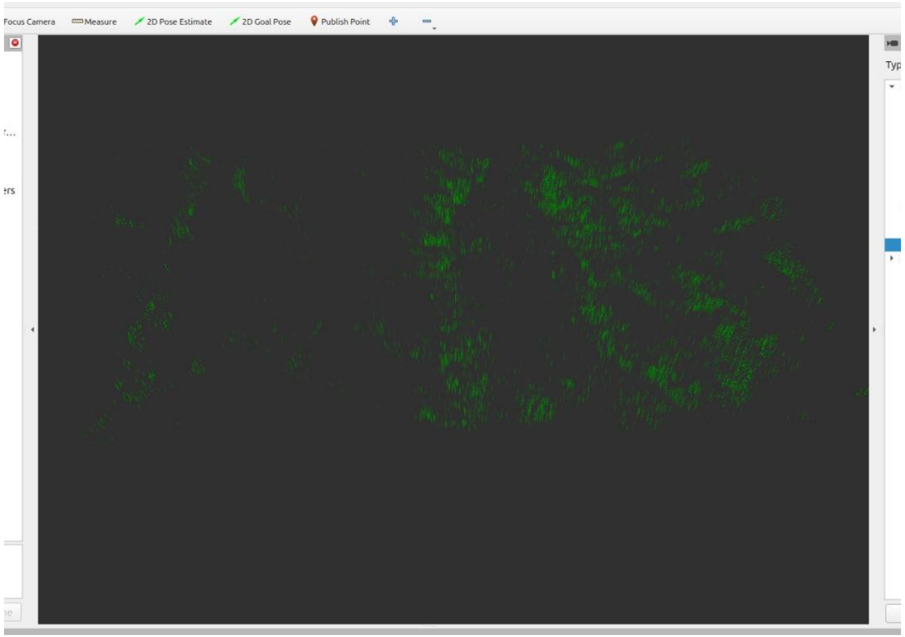
ROS Visualizations

BioMass of each tree

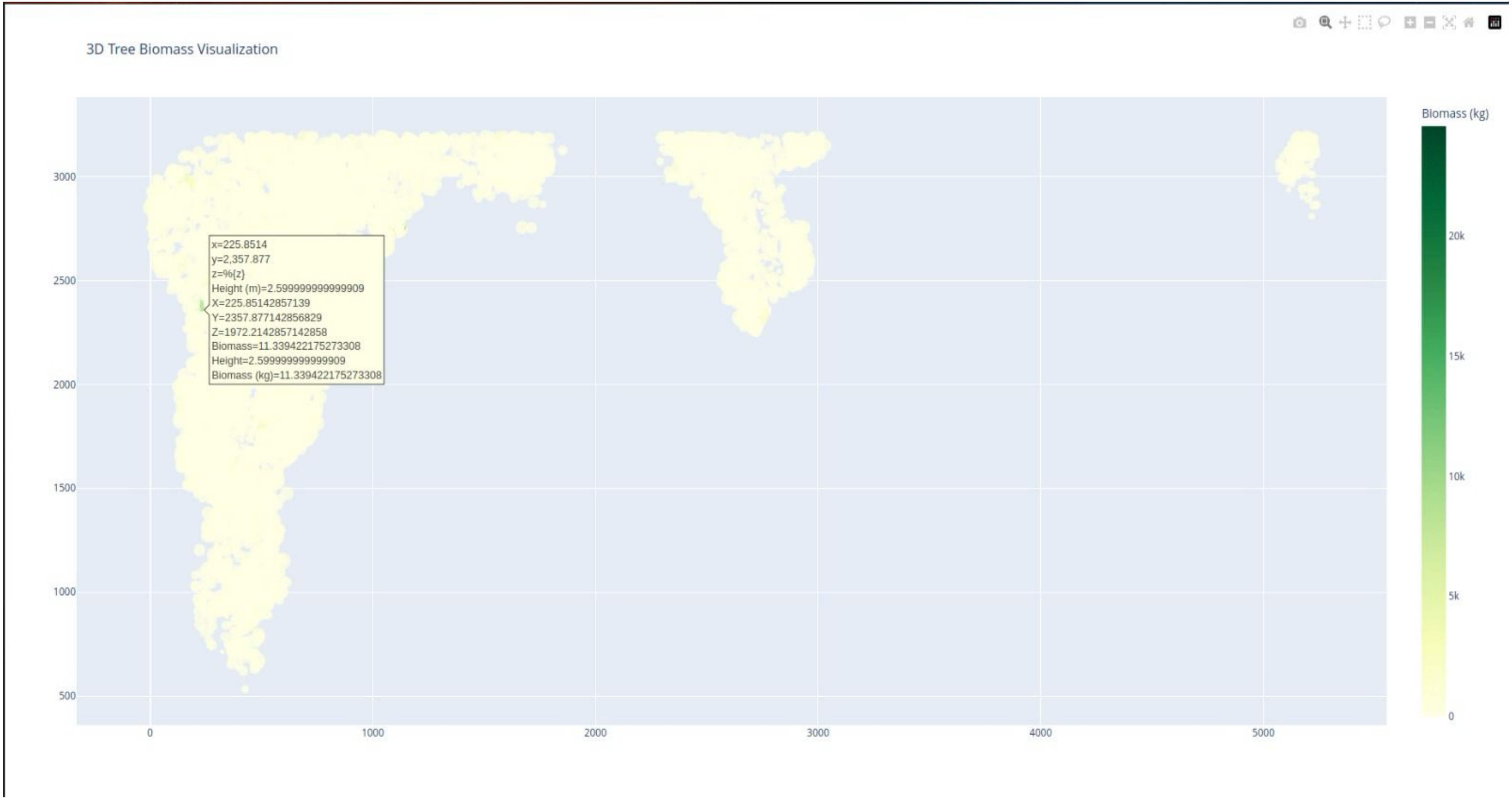
Results

Results: Biomass Estimation from ALS Data

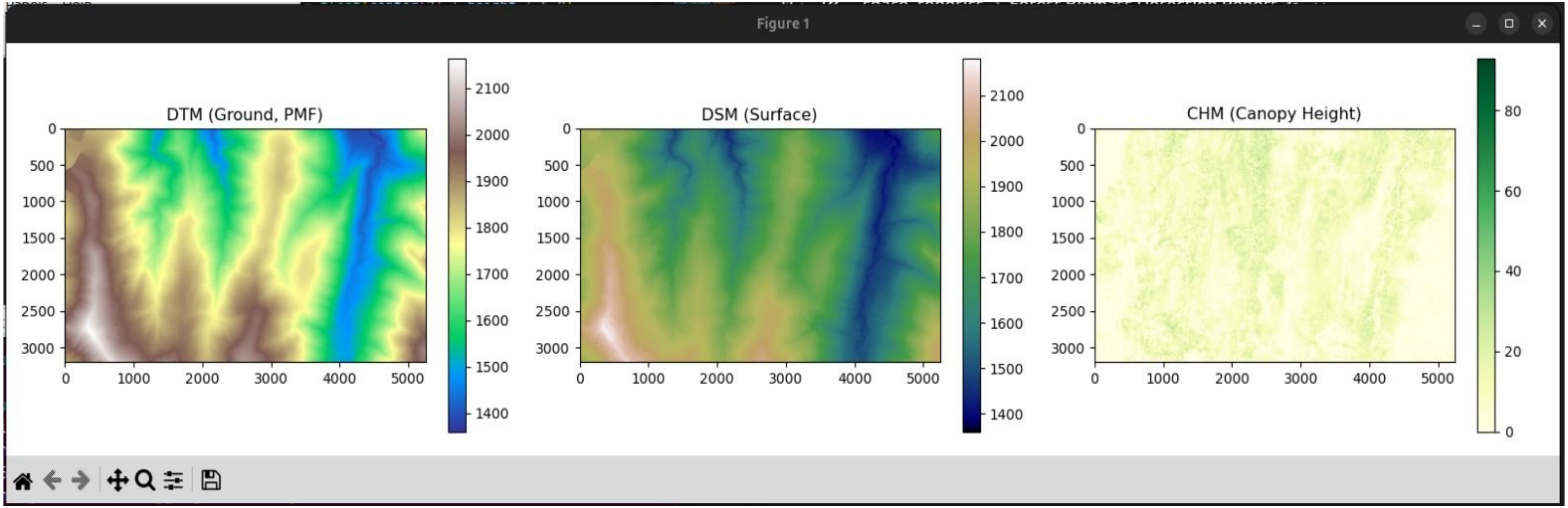
- **Visualization**: 3D point cloud of forest structure generated using Airborne Laser Scanning (ALS).
- **Data Insights**:
    - Distinct vertical and spatial patterns representing individual trees.
    - High point density supports fine-scale analysis of canopy height and volume.
- **Biomass Estimation**:
    - Tree-wise biomass derived from structural attributes (height, canopy density).
    - Spatial distribution of biomass clearly visualized across the forest area.
- **Tool Used**: Processed and visualized in **RViz** using ROS framework.
- **Significance**: Confirms potential of ALS for precise, scalable biomass estimation in forest monitoring.



Point Cloud of forest


3D Tree Biomass Visualization
x=225.8514
y=2,357.877
z=%{z}
Height (m)=2.599999999999909
X=225.85142857139
Y=2357.877142856829
Z=1972.2142857142858
Biomass=11.339422175273308
Height=2.599999999999909
Biomass (kg)=11.339422175273308
3000
2500
2000
1500
1000
500
0
1000
2000
3000
4000
5000
Biomass (kg)
20k
15k
10k
5k
0

DTM,DSM & CHM Analysis



Challenge & Future Work

Challenges Encountered Section:

- Memory management issues and solutions
- Point cloud handling
- ROS 2 environment compatibility issues
- Computational efficiency

Future Work Section:

- Short-term improvements (tree segmentation, ML refinement, ROS 2 enhancements)
- Long-term research directions (multi-sensor fusion, edge computing)
- Potential applications (carbon credits, climate science)

Conclusion

- Complete ALS-based biomass estimation pipeline.
- Estimation methods: allometric
- ROS 2 enables real-time forest analytics.

Contribution

- Shyam Kamlesh Ganatra - ROS2 integration, Gazebo simulation,Research on Database and visualization.Part documentation

- Ryan Fernandes - Worked on Biomass estimation using Allometric model ,data processing ,Research on Database.Part documentation

- Rhutvik Prashant Pachghare - Biomass estimation implementation and Dataset for ALS

- Vamshikrishna Gadde - General Research