Point Cloud Learning Assignment (Days 19 - 21)

Q1. Normalize and Downsample for Deep Learning

Objective: Prepare a point cloud with standard size and scale for deep learning.

Instructions:

- Load a .pcd file using Open3D
- Center the point cloud (mean at origin)
- Scale it to fit inside a unit sphere (normalize)
- Randomly downsample it to exactly **1024 points**

Submit:

- Screenshot of the normalized 1024-point cloud
- Print: number of points
- Save and submit as: normalized_cloud.npy

Q2. Format Conversion: .pcd → .npy

Objective: Convert a point cloud into NumPy format used in DL pipelines.

Instructions:

- Take the normalized 1024-point cloud
- Save it using np.save("cloud.npy", points)
- Reload it with np.load() and verify the shape

Submit:

- Screenshot showing saved and loaded array shape
- Print first 3 points from the .npy file for confirmation

Q3. End-to-End Processing Script

Objective: Automate filtering, normalization, and saving.

Instructions:

- Write a Python script that:
    - Loads a .pcd file
    - Removes NaNs / non-finite points
    - Applies radius outlier filtering
    - Centers and normalizes the cloud
    - Performs voxel downsampling (voxel size = 0.05)
    - Saves the result to processed_output.pcd

Submit:

- Script code
- Screenshot of terminal output (before/after point counts)
- Screenshot of final processed point cloud

Q4. Apply Plane Segmentation on Processed Cloud

Objective: Detect ground or flat surface from automated output.

Instructions:

- Load the point cloud from Q3 (processed_output.pcd)
- Use RANSAC (segment_plane()) to detect a plane
- Extract and print the plane equation (A, B, C, D)
- Compute and print number of inliers

Submit:

- Printed plane equation and number of inliers
- Optional: screenshot showing plane in green and others in red