CSCI598B - Robot Mapping (SLAM)
Project Report

Group 6:
Farid Boulos, Keenan Buckley, Randi Higashi,
Bill Huynh, Priestly Barigala, Aarushi Doctor

April 30, 2024

Grades Table

Member Name	Grade
Barigala, Priestly	10
Boulos, Farid	10
Buckley, Keenan	10
Doctor, Aarushi	10
Higashi, Randi	10
Huynh, Bill	10

I - Homework 04

1 - Exercise 1

a) Step 1

Downsampling is done by using the Open3D VoxelDownSample() function on each point cloud (Voxel Grid downsampling).

For RGBD data: Figure 2 shows downsampling effects compared to Figure 13, with voxel size parameter 0.05.

For LiDAR data: Figure 4 shows downsampling effects compared to Figure 3, with voxel size parameter 0.4.

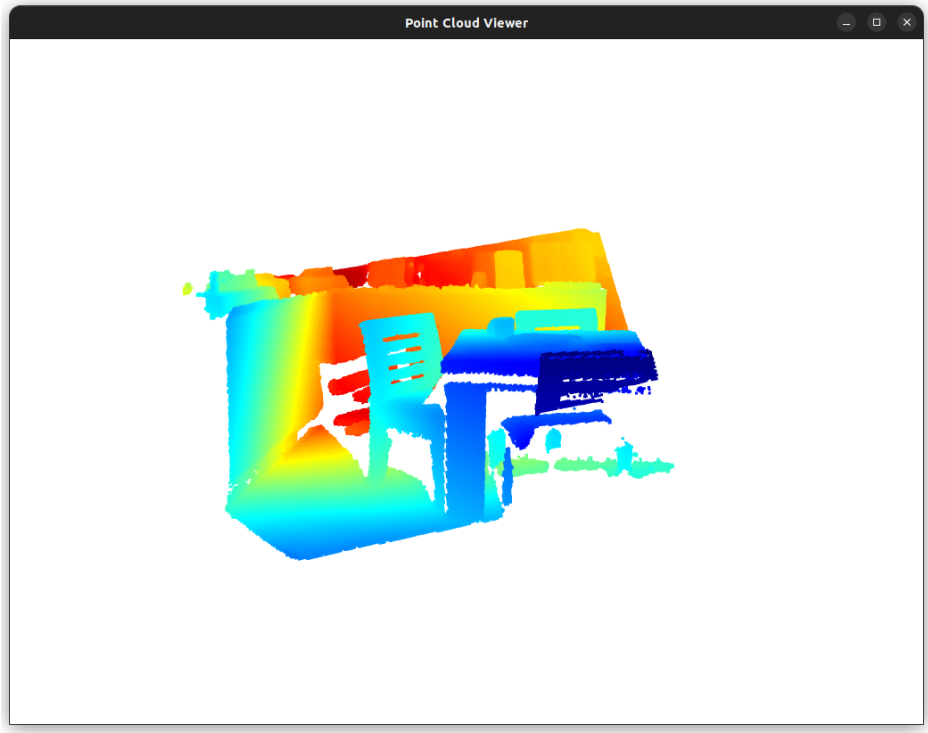

Figure 1: Snapshot of RGBD point cloud index 0 before downsampling.

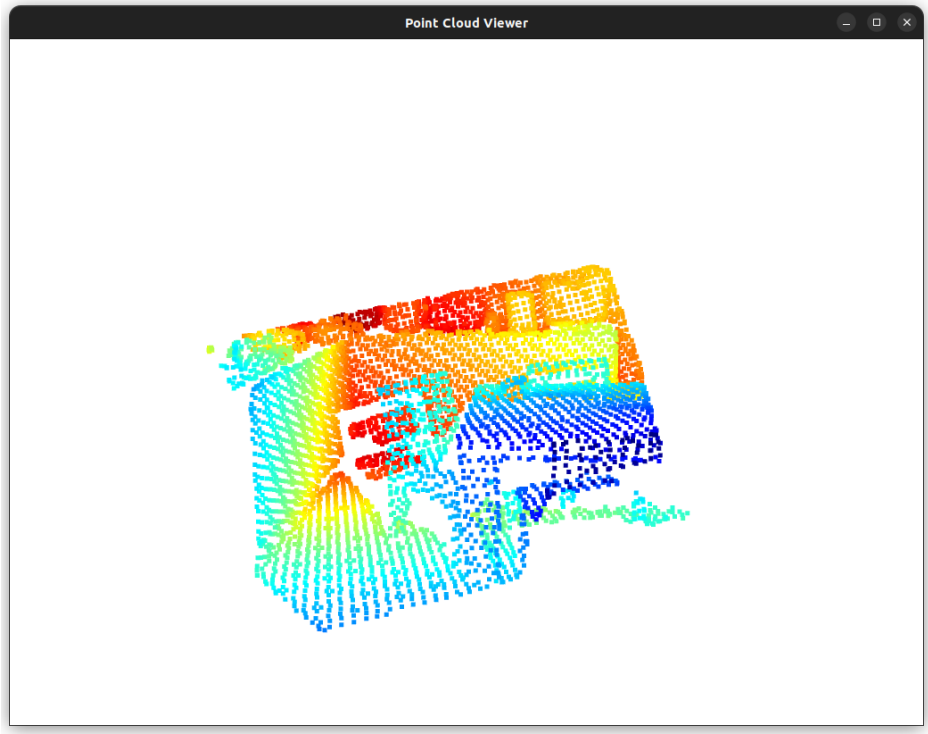

Figure 2: Snapshot of RGBD point cloud index 0 after downsampling.

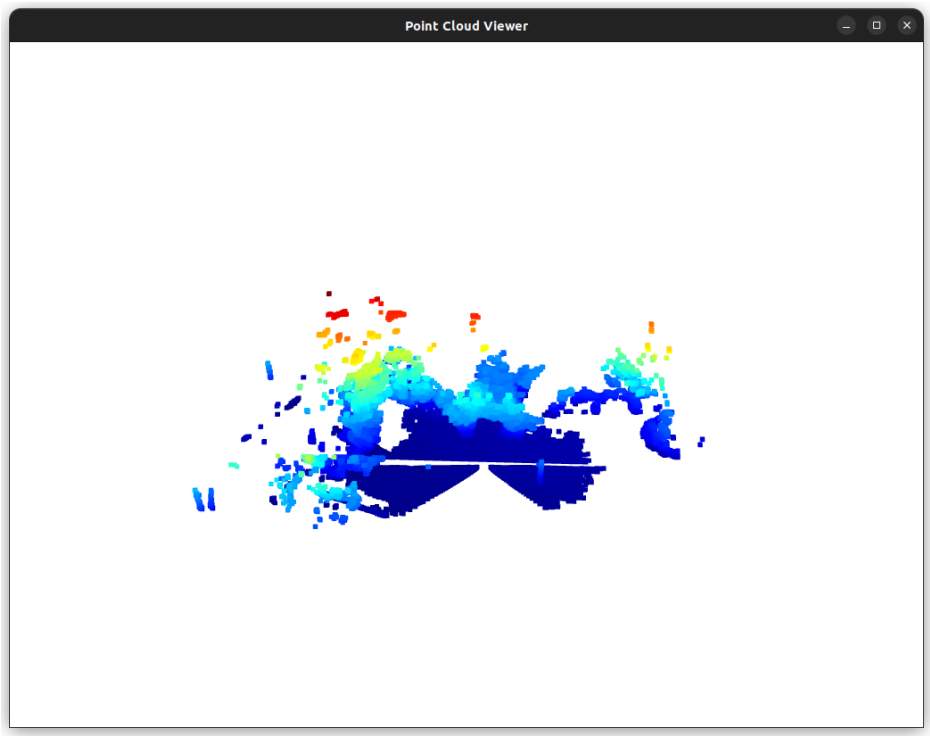

Figure 3: Snapshot of LiDAR point cloud index 0 before downsampling.

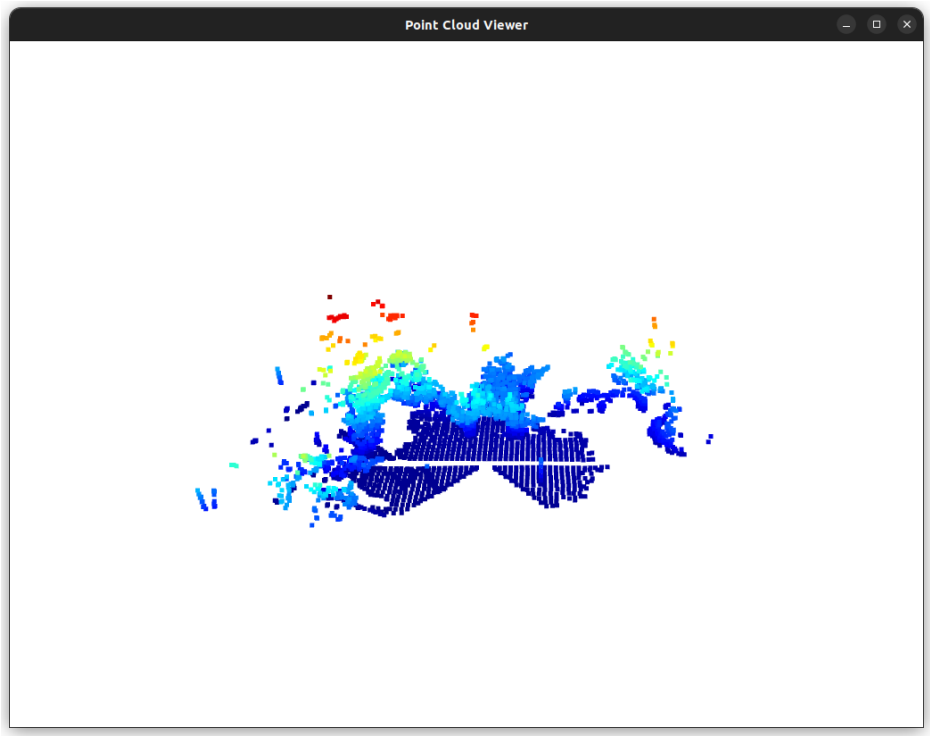

Figure 4: Snapshot of LiDAR point cloud index 0 after downsampling.

b) Step 2-3

ICP on RGBD data:

- Registration result returned by ICP: a transformation matrix (shown here is to transform from point cloud index 0 to point cloud index 1):

$$\begin{bmatrix} 0.997235 & -0.0639747 & 0.0378039 & 0.114175 \\ 0.0647506 & 0.997708 & -0.0196679 & 0.057978 \\ -0.0364589 & 0.0220614 & 0.999092 & -0.126685 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Result fusion of all point clouds:

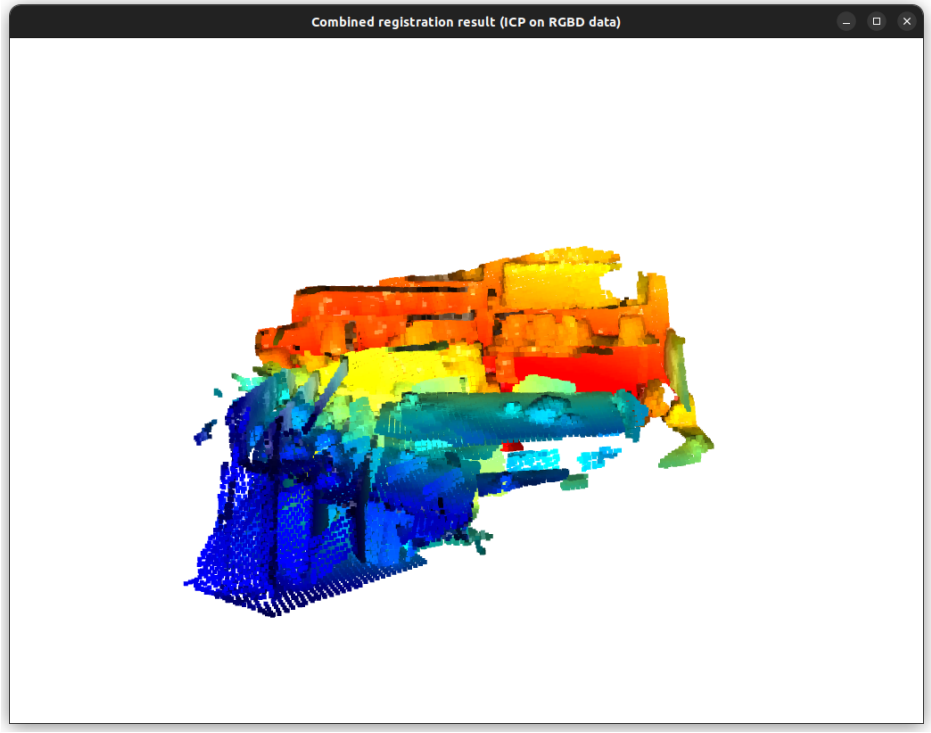

Figure 5: Result fused point clouds, using ICP on RGBD data.

- Judging from gradually registering the point clouds, ICP performs reasonably with the first 3 point clouds (indices 0 to 2). It starts showing inaccurate registration results from point clouds of index 3 on. **Overall, ICP failed to show an adequate registration result for all RGBD point clouds.**

RANSAC on RGBD data:

- Registration result returned by RANSAC: a transformation matrix (shown here is to transform from point cloud index 0 to point cloud index 1):

$$\begin{bmatrix} 0.99583 & -0.0709411 & 0.057351 & 0.0591225 \\ 0.0714476 & 0.997421 & -0.00682798 & 0.0265614 \\ -0.0567187 & 0.0108971 & 0.998331 & -0.121719 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- FPFH features are obtained using the ComputeFPFHFeature() function in Open3D. We need FPFH features for both the source and the target point clouds. To compute FPFH features, the normals of each point cloud and KD-tree search parameters are also necessary. We tried to search for ways to visualize feature points but could not find one.

- Result fusion of all point clouds:

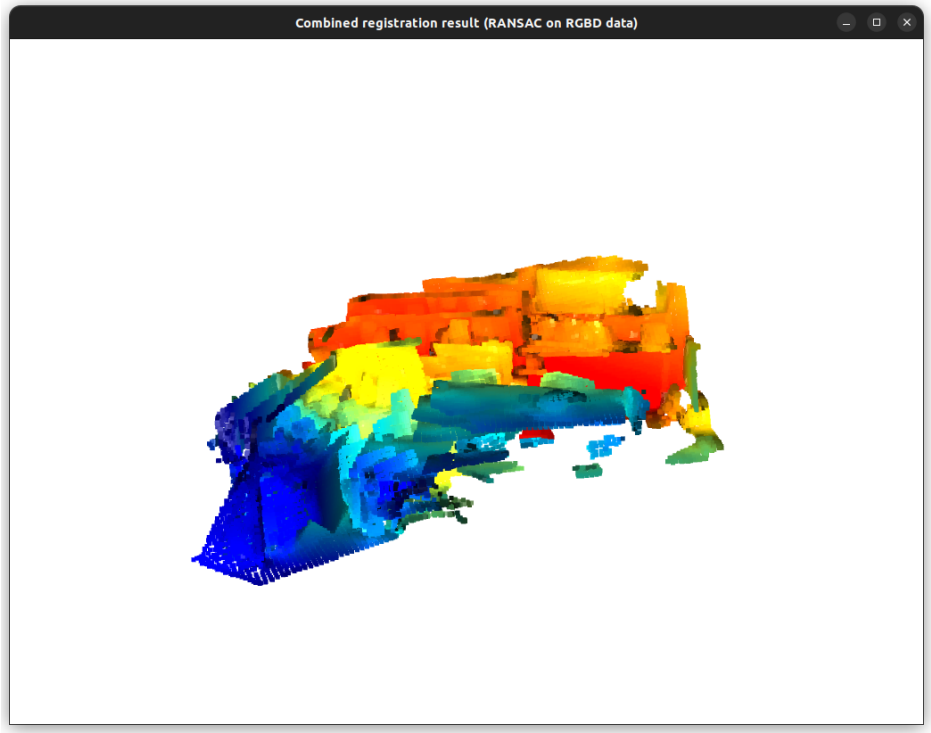

Figure 6: Result fused point clouds, using RANSAC on RGBD data.

- RANSAC performs reasonably with the first 4 point clouds (indices 0 to 3). It starts showing inaccurate registration results from point clouds of index 4 on. **Overall, RANSAC also failed to show an adequate registration result for all RGBD point clouds. However, the results are better compared to ICP on RGBD data.**

ICP on LiDAR data:

- Registration result returned by ICP: a transformation matrix (shown here is to transform from point cloud index 0 to point cloud index 1):

$$\begin{bmatrix} 0.999425 & 0.0321274 & 0.0108199 & -0.756738 \\ -0.0321099 & 0.999483 & -0.00178754 & -0.0413416 \\ -0.0108717 & 0.00143908 & 0.99994 & -0.00956993 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Result fusion of all point clouds:

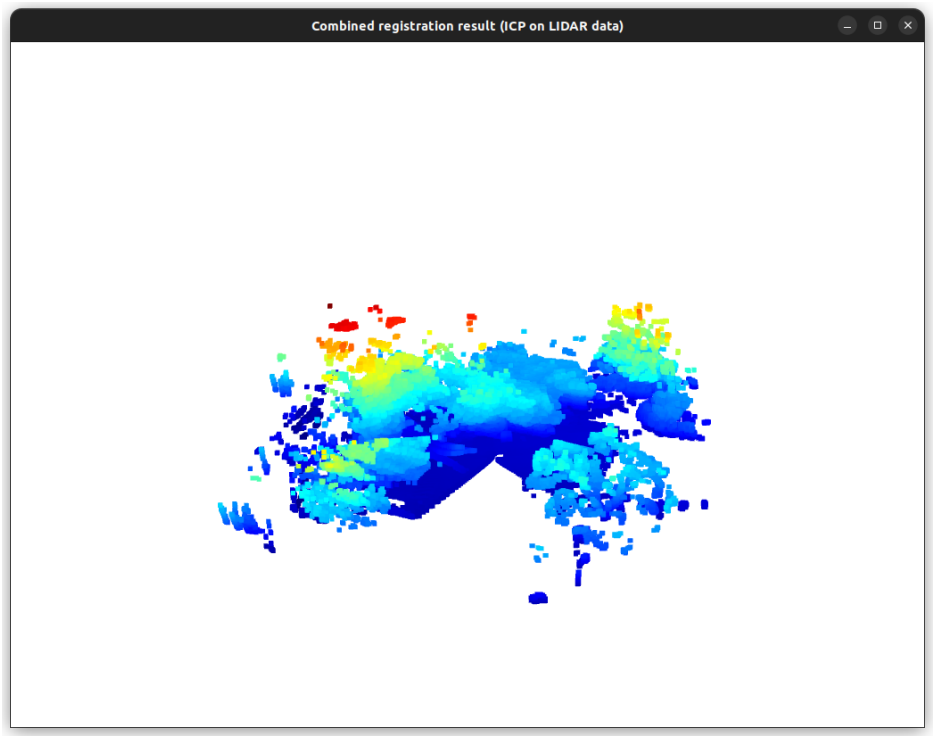

Figure 7: Result fused point clouds, using ICP on LiDAR data.

- Although there are still inaccuracies in the registration, the result fused point clouds show a scene with several small trees. **We can say ICP did not fail to register the LiDAR point clouds.**

RANSAC on LiDAR data:

- Registration result returned by RANSAC: a transformation matrix (shown here is to transform from point cloud index 0 to point cloud index 1):

$$\begin{bmatrix} 0.998775 & 0.0375367 & 0.0322403 & -0.902379 \\ -0.0376827 & 0.999282 & 0.0039351 & -0.105385 \\ -0.0320694 & -0.00514518 & 0.999472 & -0.0202583 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- FPFH features are obtained using the ComputeFPFHFeature() function in Open3D. We need FPFH features for both the source and the target point clouds. To compute FPFH features, the normals of each point cloud and KD-tree search parameters are also necessary. We tried to search for ways to visualize feature points but could not find one.
- Result fusion of all point clouds:

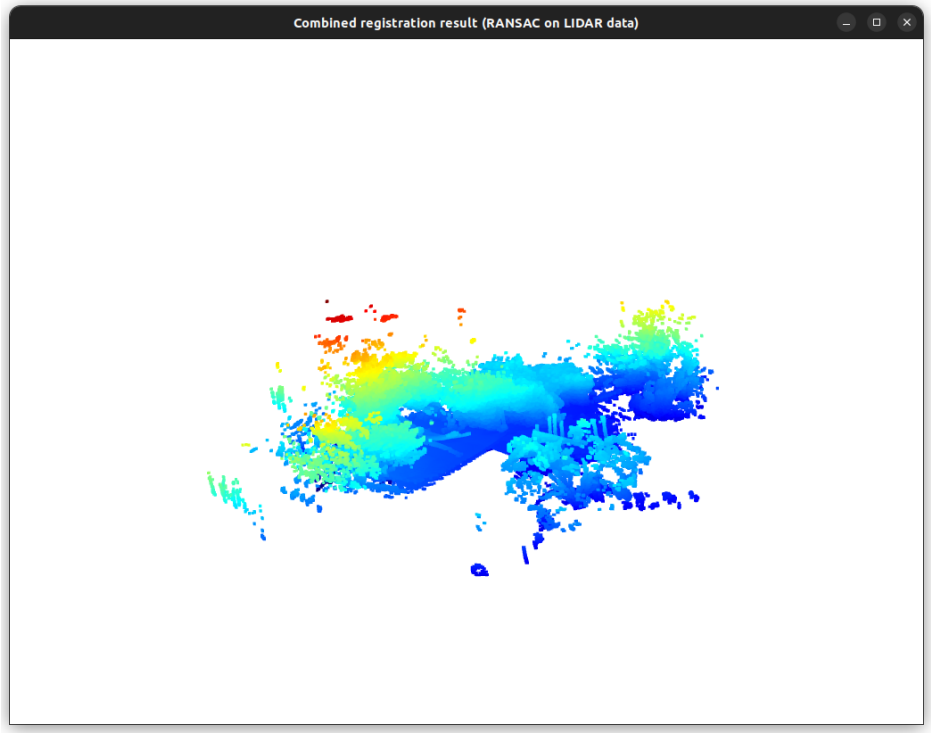

Figure 8: Result fused point clouds, using RANSAC on LiDAR data.

- Similarly to RANSAC results on RGBD data, we can make out a scene with several trees in the result fused point clouds here for LiDAR data. **Therefore, RANSAC performed adequately in registering LiDAR point clouds.**

2 - Exercise 2

a) Step 4

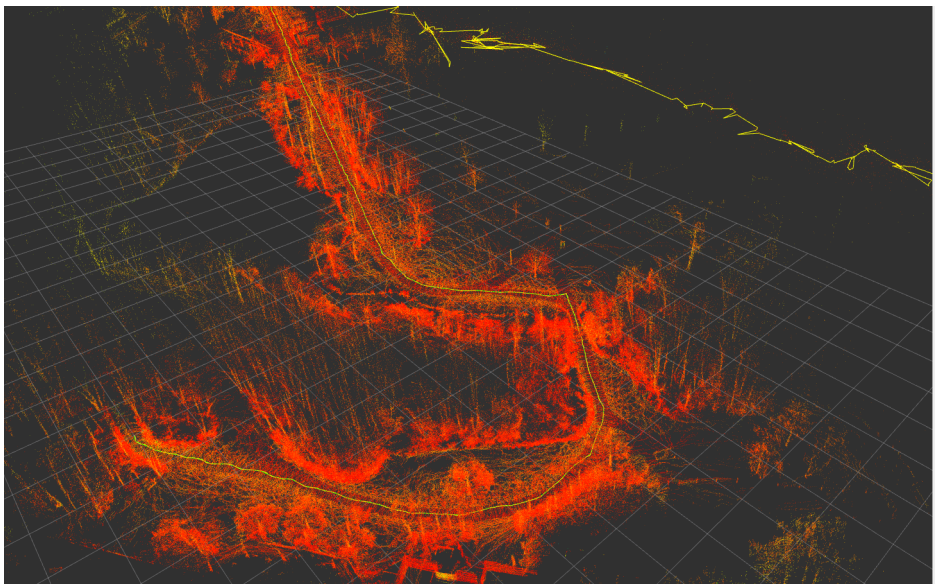

Figure 9: Result (slope area) trajectory/map of LIO-SAM algorithm on park_dataset ROS bag.

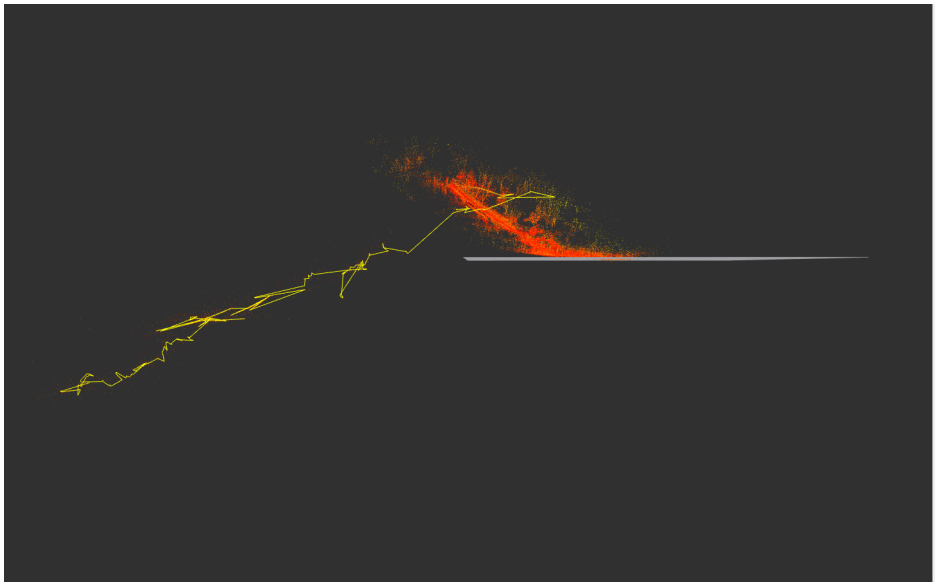

Figure 10: Result (horizontal view) trajectory/map of LIO-SAM algorithm on park_dataset ROS bag.

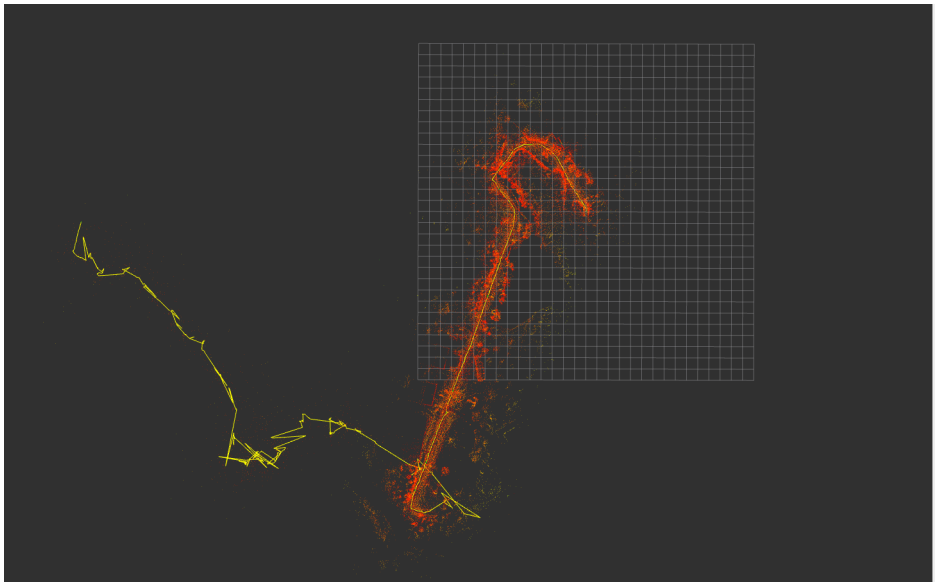

Figure 11: Result (full, top view) trajectory/map of LIO-SAM algorithm on park_dataset ROS bag.

The zigzag path at the end signals an error. This happens as soon as the scanmatcher python program reports an error with the IMU (often a "large velocity" warning and request for IMU preintegration reset). According to some issues posted on the LIO-SAM repo by TixiaoShan, this is caused by unsynced timestamp between lidar data and IMU data. However, all the data comes from the rosbag, and this is supposed to be a working demo without changing the default settings, so we are not sure.

II - Homework 05

2 - Exercise 2

a) Step 1

The camera used was a phone camera. The package used for calibrating the camera was opencv-python. The intrinsic parameters were found to be:

The camera intrinsic matrix: $\begin{bmatrix} 1453 & 0 & 539 \\ 0 & 1452 & 975 \\ 0 & 0 & 1 \end{bmatrix}$

The distortion coefficients:
k1 = 0.143
k2=-0.359
p1=0.00344
p2=0.00430
k3=0.169

b) Step 2

Snapshots of the video of the Guggenheim Hall taken are given below.
We did loop around the building twice for loop closures. ORB-SLAM succeeded in finding the loop closure 5 times. However, whenever there was a tree in the camera there was a discontinuity observed in the video. Because of these discontinuities we ended up with 3 maps instead of 1. The largest of which however, was able to cover the west, north and east sides of Guggenheim hall.

Figure 12: 5 Screenshots of the video shot of Guggenheim Hall

c) Step 3

Snapshot of the final ORB-SLAM map:

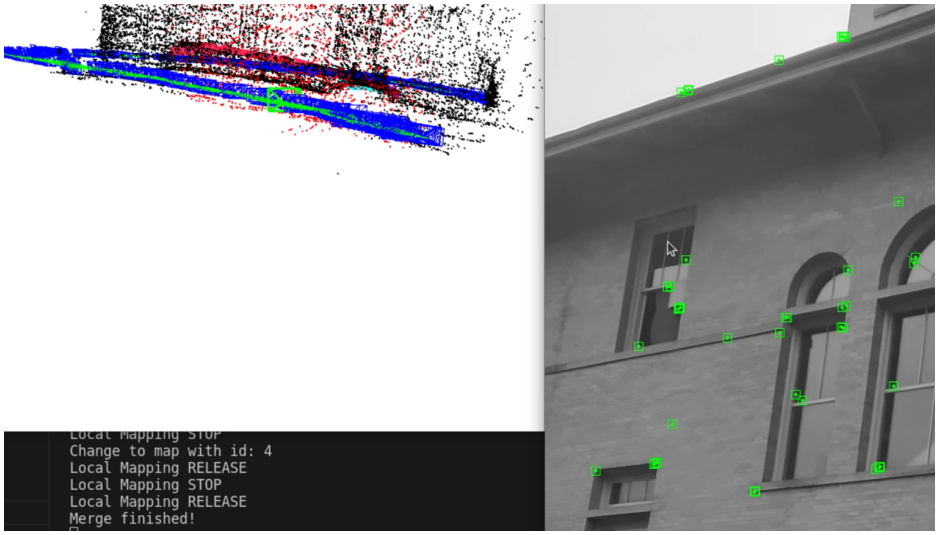

Figure 13: Snapshot of the final ORB-SLAM map

...