

# ELEC3210 Project 1 report

CHEUNG Ho Ching, 20831821

LEUNG Tsz Kit, 20863513

TAM Siu Ho, 20899419

October 11, 2023

## 1 Odometry

For the odometry, we have implicated it with the following steps:

1. Check if the point cloud queue is empty. If it is, the function sleeps and continues to the next iteration of the loop.
2. Lock the point cloud queue and retrieve the next point cloud data from the queue.
3. If this is the first frame, initialize the transformation matrix  $\mathbf{T}_{wb}$  to the identity matrix, set `firstFrame` to false, and set the reference cloud to the current point cloud.
4. Downsample the input point cloud with a voxel grid filter.
5. Downsample the reference cloud with a voxel grid filter.
6. Perform ICP registration with the current point cloud and the reference cloud and the guess ( $\mathbf{T}_{wb}$ ).
7. Update the reference cloud if the current frame is determined to be a key frame based on distance and rotation angle from the previous key frame, or if the time elapsed since the last key frame is greater than 5 seconds.
8. Publish the odometry estimation result.
9. Sleep for a short period of time to maintain a constant loop rate.

## 2 ICP registration

For the ICP algorithm, we have implemented the function with the following steps:

1. Initialize a KdTree with the target point cloud.
2. Initialize the transformation and previous transformation matrices to identity.
3. Transform the source point cloud using the initial guess transformation.
4. While the number of iterations is less than the maximum number of iterations:
  - (a) Find the nearest neighbor correspondences between the transformed source point cloud and the target point cloud using the KdTree.
  - (b) Compute the mean-centered source and target point clouds.
  - (c) Compute the  $3 \times 3$  matrix  $H$  from the mean-centered source and target point clouds.
  - (d) Compute the Singular Value Decomposition (SVD) of  $H$  using the `JacobiSVD` function.
  - (e) Compute the rotation and translation components of the transformation from the SVD factors.
  - (f) Update the transformation matrix with the computed rotation and translation components.
  - (g) Check if the difference between the current and previous transformations is less than the maximum distance threshold. If so, break out of the loop.
  - (h) Set the previous transformation to the current transformation.
5. Return the final transformation matrix.

### 3 Result

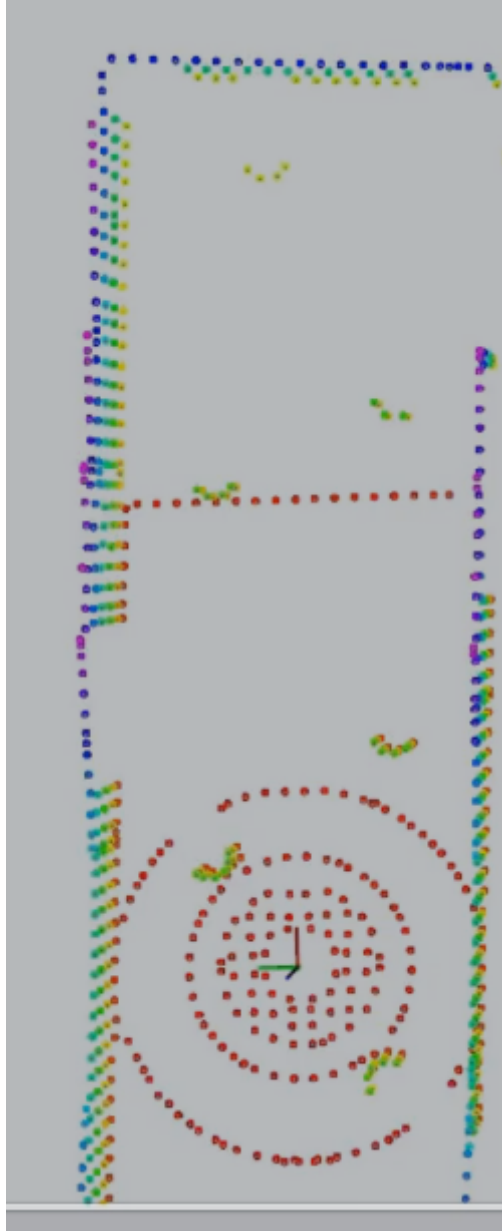


Figure 1: Result of odometry during simulation

### 4 Contribution to the project

Name	Contribution
CCHEUNG Ho Ching	Odometry
LEUNG Tsz Kit	ICP
TAM Siu Ho	Downsampling, report