ME5413: Autonomous Mobile Robot

Homework 2: Localisation

Name: WANG Yexiang   Matric Number: A0304719A

Task 1: Writing the ICP Algorithm for Point Cloud Registration

The **Iterative Closest Point (ICP)** algorithm is used for point cloud registration by iteratively minimizing the distance between a source point cloud and a target point cloud through rigid transformations.

In this assignment, we have two point clouds stored in ***bunny1.ply*** and ***bunny2.ply***. The task requires using the ICP algorithm to align the two point clouds. The ICP algorithm is mainly divided into the following steps:

1. **Correspondence Matching**: Find the closest point in the target cloud for each point in the source cloud.
2. **Transformation Estimation**: Compute the optimal rigid transformation (rotation R and translation t) using Singular Value Decomposition (SVD).
3. **Apply Transformation**: Update the source cloud by applying (R,t).
4. **Iteration**: Repeat until convergence, based on small transformation updates or minimal error reduction.

Implementation of the ICP algorithm

In the ***task1*** folder, there are three Python files:
task1_icp.py: Traditional implementation of the ICP algorithm.
task1_pca.py: Uses PCA for initial alignment before applying the ICP algorithm.
task1_visualization.py: Visualize the initial state of the two point clouds and the effect after PCA initial alignment.

In ***task1_visualization.py***, the two point clouds can be visualized. The red represents the reference cloud, while the blue represents the original cloud. The result is shown in the figure below.

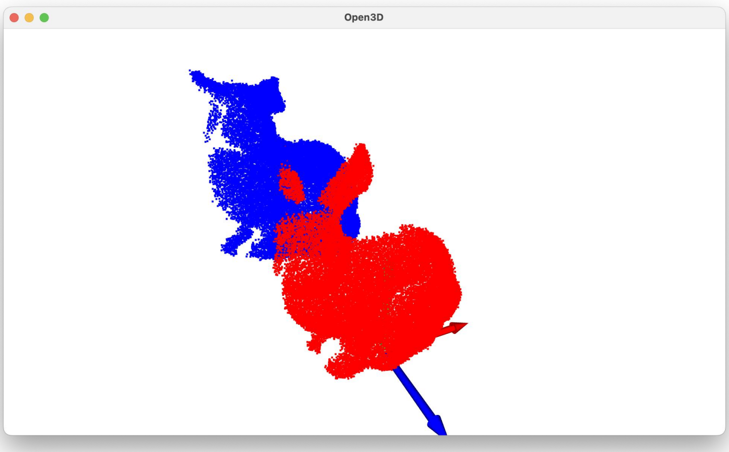


Figure 1 The initial state of the point clouds.

Traditional ICP method

In ***task1_icp.py***, the basic ICP algorithm is implemented. After 30 iterations, the result is shown in **Figure 2**. As can be seen, the error has been optimized to a relatively small value, indicating that the ICP algorithm has successfully aligned the point clouds. However, the process of finding matching points requires traversing all the points in the point cloud, which takes a **long time**. Specifically, in this task, as shown in **Figure 2**, each iteration takes nearly 20 seconds, and the total time for all thirty iterations is approximately 550 seconds.

Additionally, the convergence during the iterations is highly **unstable**, and it may take a large number of iterations to achieve a good result.As can be seen from the figure, the computed transformation T, specifically the values of T1_2_cur for each iteration, are very large. This indicates that the pose of the point cloud changes significantly in each iteration, and the convergence is highly unstable.

```
------------------------------------------
iteration = 30
time cost = 19.310661s
total time cost = 549.7647s
T1_2_cur =
[[-0.54663409 -0.81915385  0.17371857  3.04401938]
 [ 0.67731012 -0.31054     0.66694521 12.96741608]
 [-0.49238417  0.48223633  0.72457294 -5.14520333]
 [ 0.          0.          0.          1.        ]]
accumulated T =
[[ 0.81488831 -0.45231996 -0.36244129  8.26265743]
 [ 0.34282787  0.88033236 -0.32784751 -3.01419973]
 [ 0.46736077  0.14290413  0.87244044  8.03213842]
 [ 0.          0.          0.          1.        ]]
mean_error= 1.2477426307018569
```

Figure 2 Basic ICP Algorithm Results

The final result is shown in **Figure 3**. As can be seen, there is still some error, and the point clouds have not been completely aligned.The expected result is that the red and blue point clouds almost completely overlap.

The reason for not achieving this effect could be that the number of iterations is too few, or there may be noise in the point cloud, making it difficult to fully align during the nearest point matching.

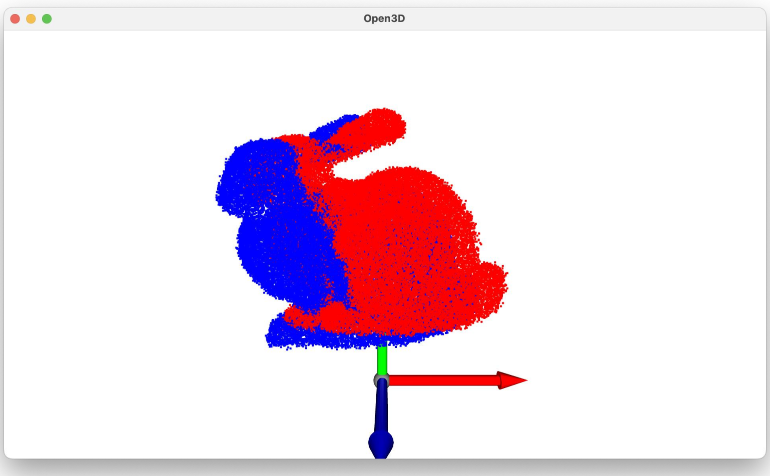


Figure 3 The Final Result of the ICP Algorithm after 30 Iterations

Use the PCA method for initial alignment followed by ICP

ICP method has the drawback of being prone to local minima and requiring a good initial alignment. Therefore, based on ICP, I first perform an initial **PCA pre-alignment**, followed by ICP, to improve the effectiveness of the ICP algorithm. PCA is commonly used in machine learning for data preprocessing, image compression, and pattern recognition. Therefore, PCA can be used to perform an initial alignment of point clouds based on their feature vectors.

The final results are shown in **Figure 4** and **Figure 5**. As can be seen, the effect after performing the pre-alignment is very good, and it reaches the set threshold. In fact, the PCA alignment itself works very well. From **Figure 5**, it can be observed that the red and blue point clouds have completely merged together.

```
pca_error= 0.015948888741845753
-----------------------------------------
iteration = 1
time cost = 18.251229s
total time cost = 18.251229s
T1_2_cur =
[[ 9.99999999e-01  2.27615907e-05 -2.34838500e-05 -1.38918703e-04]
 [-2.27616324e-05  1.00000000e+00 -1.77516475e-06 -5.23736564e-05]
 [ 2.34838095e-05  1.77569928e-06  1.00000000e+00  3.52335478e-05]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
accumulated T =
[[ 0.88812821 -0.30340986 -0.34521115  5.71097501]
 [ 0.19993141  0.93138003 -0.30423456 -5.33225798]
 [ 0.41383054  0.20118074  0.88784604  6.85930463]
 [ 0.          0.          0.          1.        ]]
mean_error= 0.01595092489340399
------- fully converged! -------
time cost: 18.251229 s
```

Figure 4 PCA-based ICP Algorithm Results

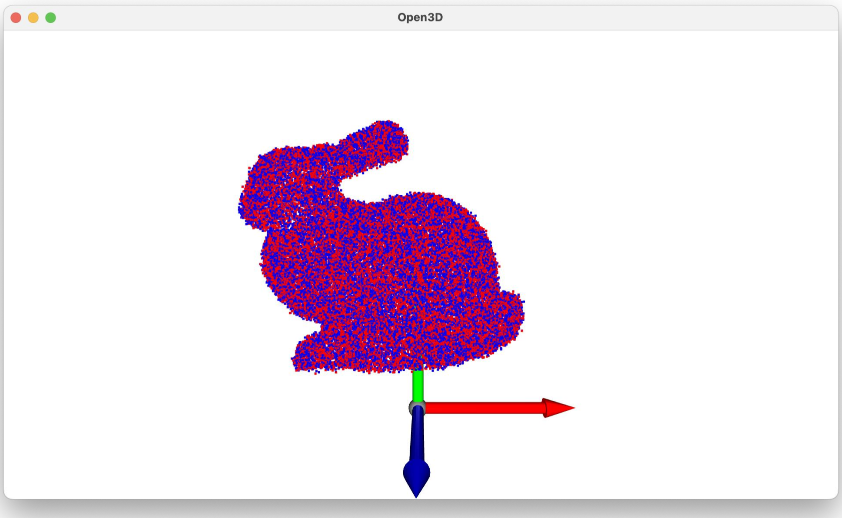


Figure 5 The Final Result of PCA-based ICP Algorithm

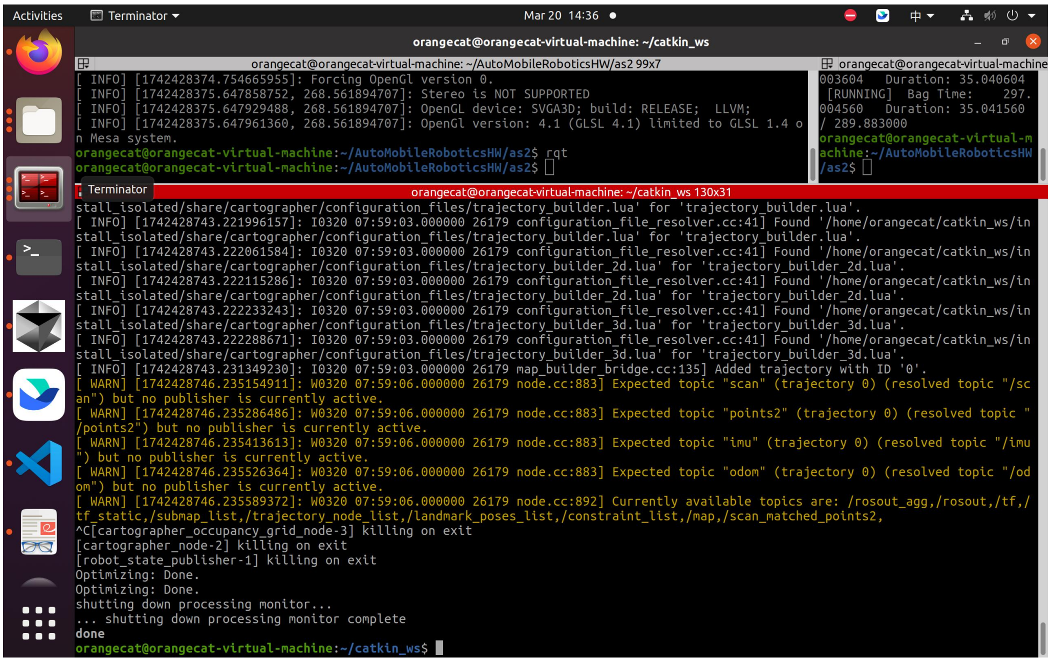
Improvements for the future

1. Use random sampling to reduce the number of matching points and speed up the matching process. Since point cloud data itself contains errors and noise, random sampling will not significantly affect the final results.
2. Use the KD-tree data structure for acceleration, enabling faster nearest point matching.

Task 2: Running a SLAM Algorithm in ROS

Unfortunately, I was not able to complete task 2. I encountered issues with using Cartographer for mapping that I could not resolve. Therefore, in this section, I can only briefly introduce the problems I faced.

After configuring the lua file and launch file, when running the launch file, the following error occurs. Additionally, when I use ***rostopic echo*** for the corresponding topic, I find that no data is being transmitted. I suspect the issue occurred during the configuration of the topic and remap settings, but unfortunately, I was not able to resolve it in time.



What confuses me the most is that when I use the ***cartographer_rosbag_validate*** command to run the rosbag, it works fine, and I can see the topic names for the data. However, when I configure it myself, the data fails to be transmitted.