Nuages de Points et Modélisation 3D (NPM3D) TP1

Zhe HUANG, from IASD

15/01/2024

1 Question1

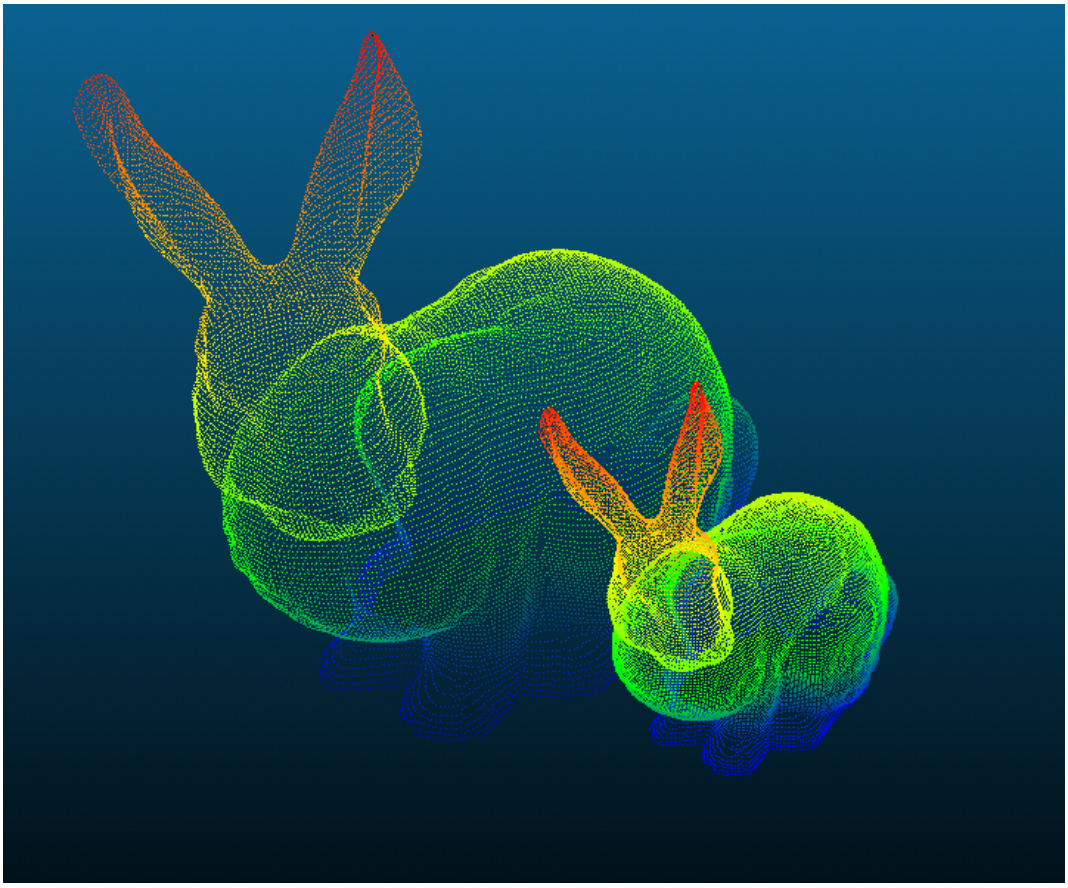

Figure 1: the original bunny(left) and the transformed bunny(right)

2 Question2

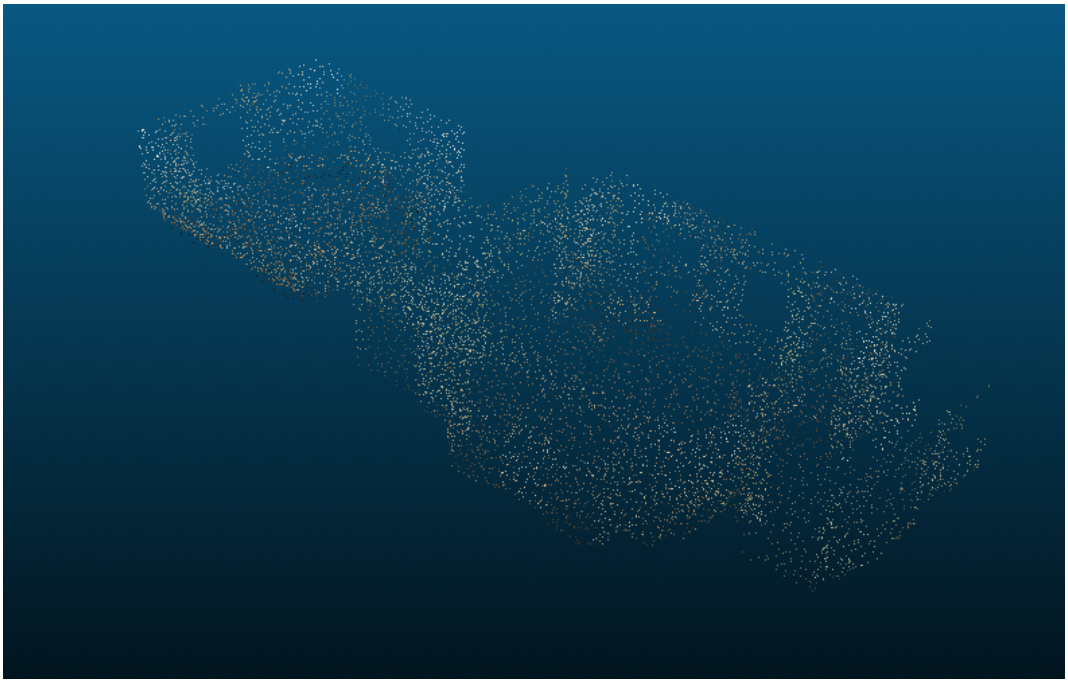

Figure 2: the decimated 'indoor scan' point cloud

3 Question3

Method	**Time for 10 Queries**	**Estimated Time for Entire Cloud**
Spherical Neighborhoods	0.166 seconds	14 hours
K-Nearest Neighbors (KNN)	2.336 seconds	197 hours

Table 1: Spherical Neighborhoods and K-Nearest Neighbors (KNN)

4 Question4

4.1 Question 4.a

Leaf Size	Experiment 1 (seconds)	Experiment 2 (seconds)	Experiment 3 (seconds)
30	1.726	1.706	1.709
40	1.727	1.715	1.719
50	1.556	1.560	1.557
60	1.552	1.559	1.550
70	1.555	1.575	1.552
80	1.554	1.578	1.553
90	1.569	1.584	1.569
100	1.469	1.460	1.440
110	1.448	1.441	1.434
120	1.457	1.461	1.435
130	1.471	1.477	1.435
140	1.481	1.479	1.443

Table 2: Timing results for spherical neighborhood searches with varying leaf sizes

After conducting a series of experiments of which the results are in Table 2, a leaf size of 110 consistently resulted in the fastest search times across all three experiments.

The optimal leaf size is not 1 because, with a leaf size of 1, the KDTree would have as many leaves as data points, resulting in a deep tree where each point is in its own leaf node. This extreme granularity would mean that many tree nodes must be visited and checked individually during the search process, which significantly increases the computational burden.

Thus, the optimal leaf size, as evidenced by the experimental results, supports a balance between the depth of the tree and the manageability of search within each leaf node, leading to more efficient neighborhood searches.

4.2 Question 4.b

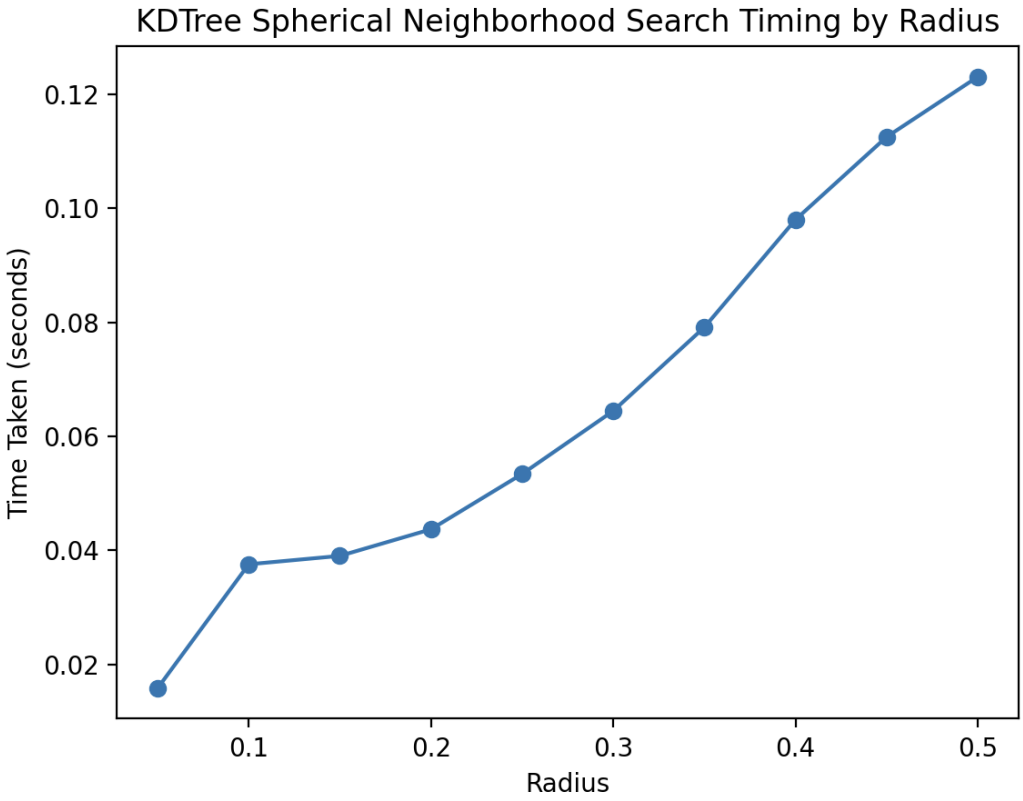

Figure 3: Timing results obtained with KDTree as a function of redius

The timing appears to increase with the radius. This trend is expected because as the radius increases, the volume of the spherical neighborhood being searched within the point cloud also increases.

Using the same estimation method, it takes 132.82 seconds to search 20cm neighborhoods for all points in the cloud.

5 Question Bonus

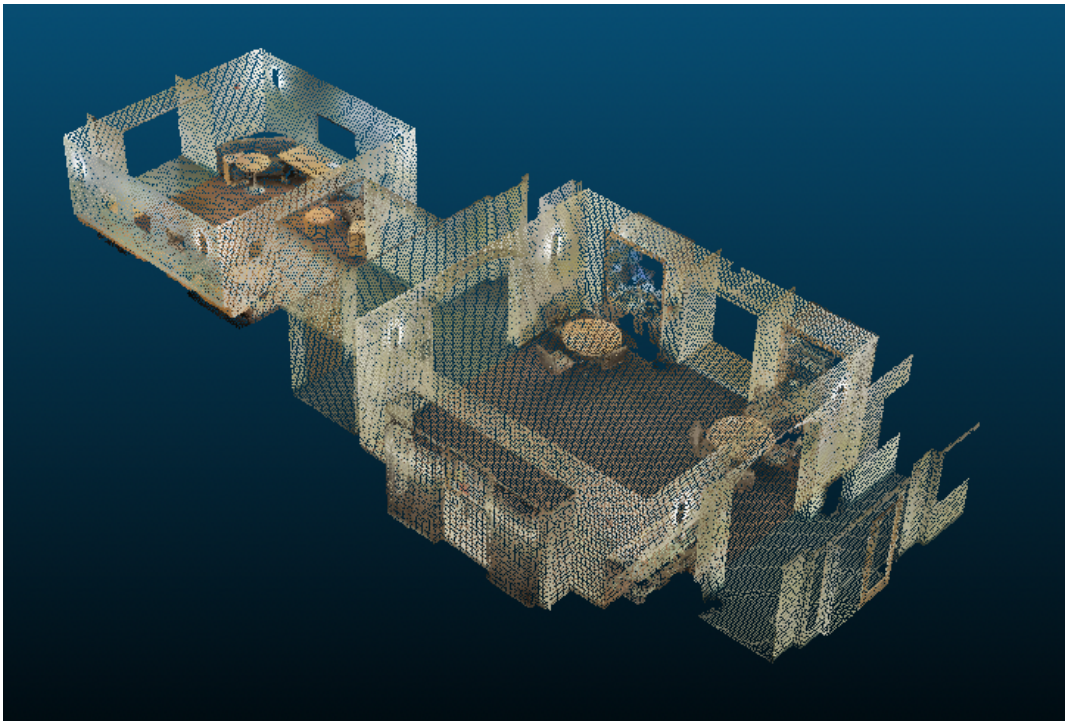

Figure 4: the grid subsampled indoor scan point cloud

Grid subsampling maintains spatial uniformity by dividing the space into a grid and keeping only one point per grid cell. This method contrasts with decimation from Part C, which merely skips points at regular intervals. As a result, grid subsampling achieves a more even distribution of points, better capturing the shape of the original cloud.