

TP1 - 3D Point Cloud and Modeling

Chloé Habasque - Colin Coërchon

January 15, 2025

1 Point clouds manipulations in Cloud Compare

2 Point clouds transformations in Python

Question 1 :

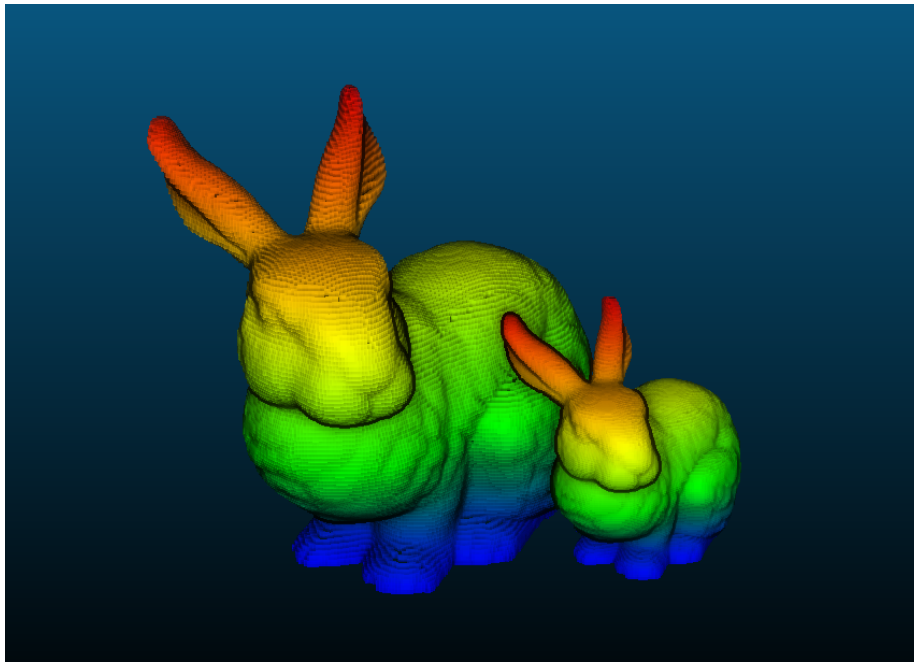


Figure 1: Screenshot of the original bunny and the transformed bunny together

3 Subsampling methods

Question 2 :

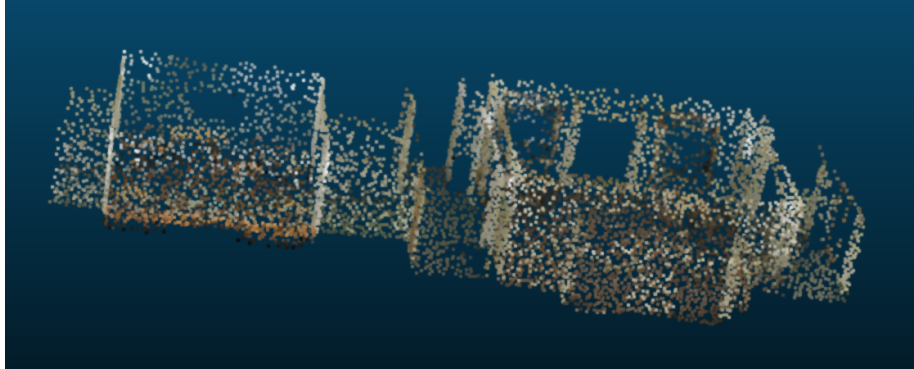


Figure 2: Screenshot of the decimated *indoor_scan* point cloud

4 Structures and neighborhoods

4.1 The concept of neighborhoods

Question 3 :

Here are the results obtained with our two brute force algorithms: spherical neighborhood and the k-nearest neighbors (KNN) :

10 spherical neighborhoods computed in 0.295 seconds.

10 KNN computed in 1.753 seconds.

Computing spherical neighborhoods on whole cloud : 25 hours.

Computing KNN on whole cloud : 148 hours.

4.2 Hierarchical structures

Question 4.a :

At some point, it becomes much faster (and more logical) to perform a "brute force" search rather than continuing to subdivide the space using hyperplanes until a single point remains in each subdivision. This is why **setting leaf_size to 1 is far from optimal**. By default, in the Scikit-learn library, the default value of `leaf_size` is 40.

Empirically, for our dataset, we found that the optimal value for `leaf_size` is 15.

Question 4.b :

See [Figure 3](#).

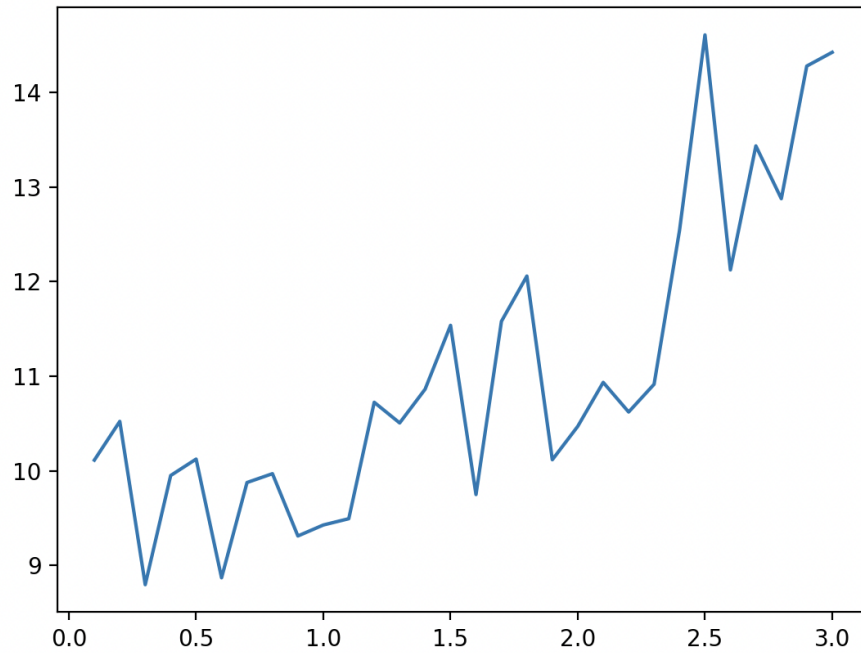


Figure 3: Timings obtained with KDTree as a function of radius

It takes around 1 hours to search 20cm neighborhoods for all points in the cloud.

5 Going further (BONUS)

See [Figure 4](#).

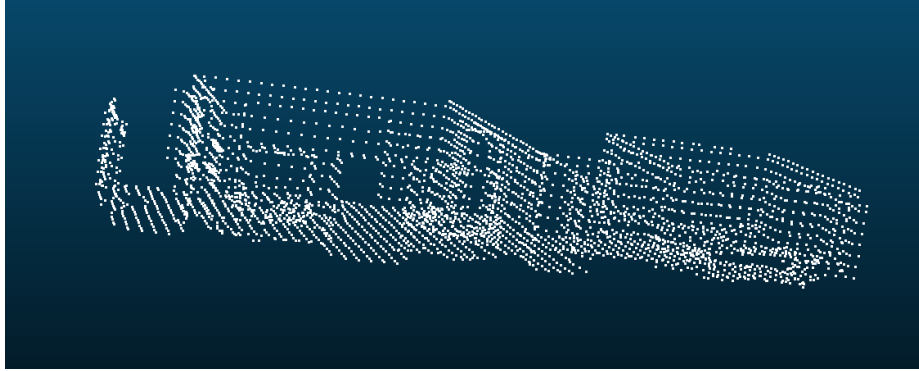


Figure 4: Screenshot of the decimated indoor_scan point cloud with grid sampling with $d1=0.3$