TP 2: Iterative Closest Points algorithm for point cloud registration

Martin Jolif

January 22, 2025

1 Question 1

The ICP performs well on the first example between *bunny_original.ply* and *bunny_perturbed.ply* as the to bunnies are close to each other.

However, it doesn't works in the second case between *bunny_original.ply* and *bunny_returned.ply*. Indeed, these two bunnies configurations are too different, ICP performs well only for close configurations.

In the last case, the ICP doesn't work as well when the reference point cloud is *Notre_Dame_Des_Champs_2.ply*. However, when the reference point cloud is *Notre_Dame_Des_Champs_1.ply*, ICP works well.

The main difference between the aligned cloud and the reference cloud is the transformation that ICP applies (rotation and translation) to align the source cloud with the reference. After alignment, the aligned cloud should match the reference cloud as closely as possible.

This is why *Notre_Dame_Des_Champs_1.ply* should be the reference point cloud. Indeed *Notre_Dame_Des_Champs_1.ply* has much more points in his point cloud than the *Notre_Dame_Des_Champs_2.ply* point cloud. Therefore, the *Notre_Dame_Des_Champs_2.ply* can match closely the reference point cloud. On the contrary, the *Notre_Dame_Des_Champs_1.ply* point cloud cannot match closely the *Notre_Dame_Des_Champs_2.ply* point cloud as it has a lot more points and therefore cannot be matched correctly.

2 Question 2

As we can see in the figure 1, the best rigid transformation function performed well between *bunny_original.ply* and *bunny_returned.ply*. The RMS error before the best rigid transformation function was 0.16083363 and is 0.00000001 after the best rigid transformation function.

The best rigid transformation function alignement worked well in our python code case because we have a prior knowledge of the corresponding points in the two point clouds (given

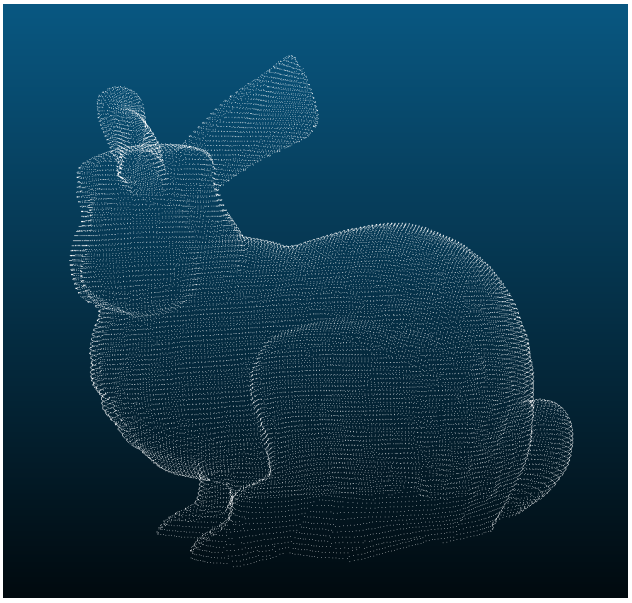
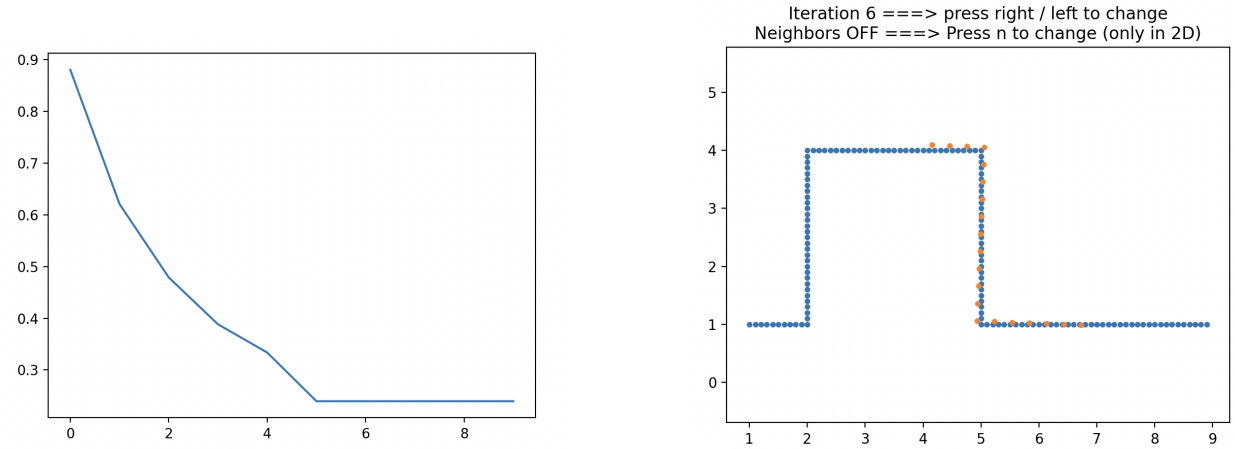


Figure 1: Best rigid transformation function on a bunny

by the ordering). It doesn't work well in the case of CloudCompare ICP because CloudCompare doesn't have this prior knowledge and should compute the corresponding points.

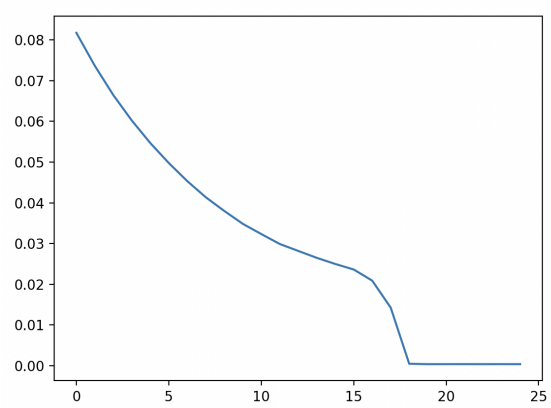
This function would not be able to align the 3D scans of "Notre Dame des Champs" because it is designed to process two point clouds with the same number of points which is not the case for "Notre Dame des Champs".

3 Question 3

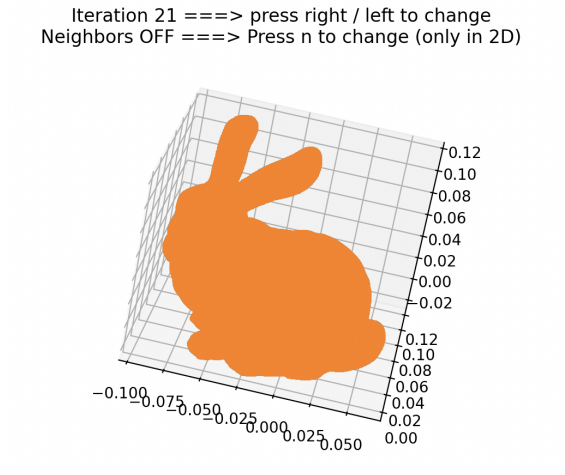


RMS evolution depending on the iteration number Convergence of the ICP

Figure 2: 2D example



RMS evolution depending on the iteration number



Convergence of the ICP

Figure 3: Bunny example

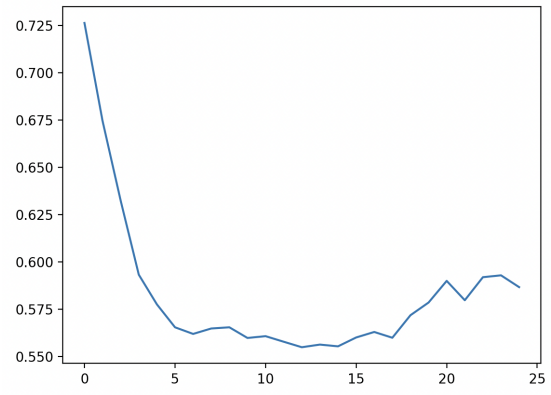
4 Question 4

For the 2D example, the RMS diminish to reach a minimum of 0.239 quit fast (just need 5 iteration) but don't converge to zero.

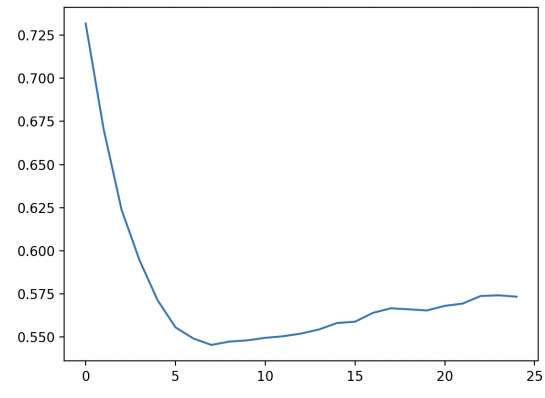
On the contrary, for the bunny example, the RMS diminish slower (need 18 iterations) and converge to zero this time.

We can lso observe that increasng the number of point in the point cloud may reduce the variance of the RMS during iterations.

5 Question 5



sampling_limit = 1000



sampling_limit = 10000

Figure 4: RMS evolution depending on the iteration number

These times, the RMS loss isn't more monotonic. It seems to reach a minimum and then to increase again.