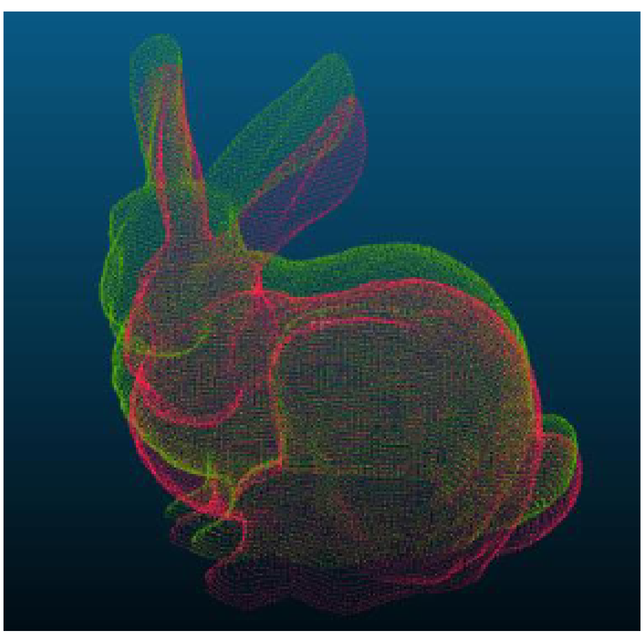
NPM3D - TP 2: Iterative Closest Points algorithm for point cloud registration

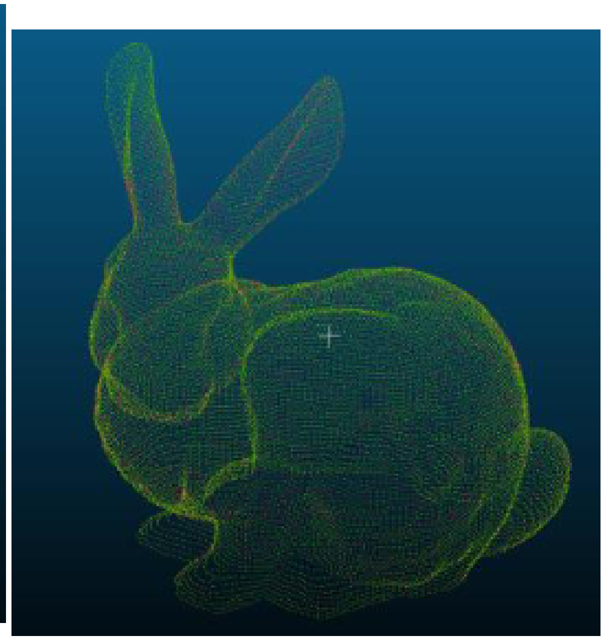
Adrien Golebiewski

IASD 2022-23

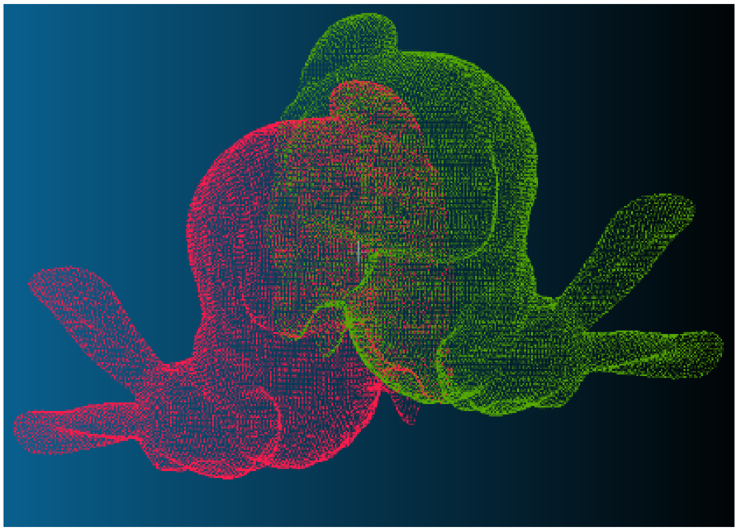
1 Question 1

The ICP gives good results for the perturbed rabbit cloud (see Figure 1) but the registration does not work when a rotation is applied to the cloud (see Figure 2). not work when a rotation is applied to the cloud (see figure 2).

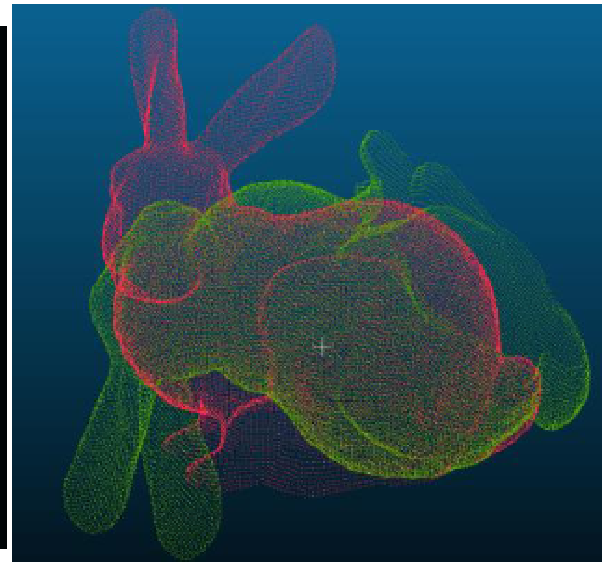




(a) ICP between Bunny original and Bunny perturbed



(a) ICP between Bunny original and Bunny returned



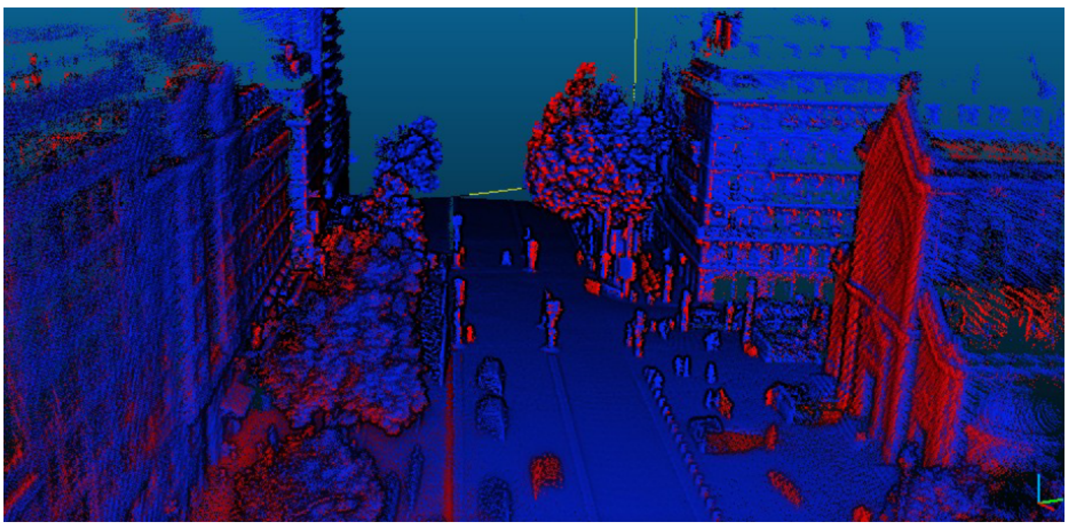


Figure 3: ICP for Notre Dame des Champs 1 and 2

ICP performs perfectly when the two clouds are loosely alligned, but not when the clouds are far from being aligned. When the two point clouds are from slightly different point of view and time, like with the Notre Dame des Champs, some artefacts and slight misalignment can be seen. In this example, **it is the biggest cloud point which should be the reference** because its lignes and planes can be used as external added reference for checking validity of ICP. Finally, we cannot appariate every point of Notre Dame Des Champs 1 to Notre Dame Des Champs 2 because Notre Dame Des Champs 1 represent some zones that are not present in Notre Dame Des Champs 2, causing the RMS to increase a lot if we take Notre Dame Des Champs 2 as the reference.

2 Question 2

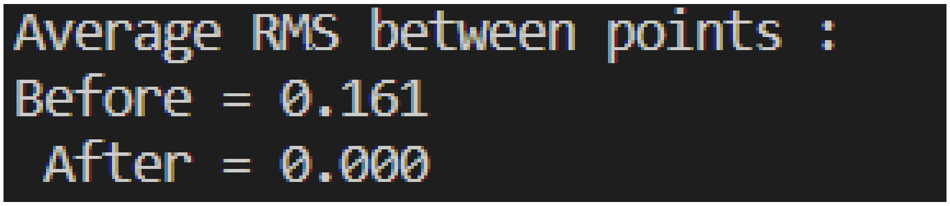


Figure 4: RMS errors

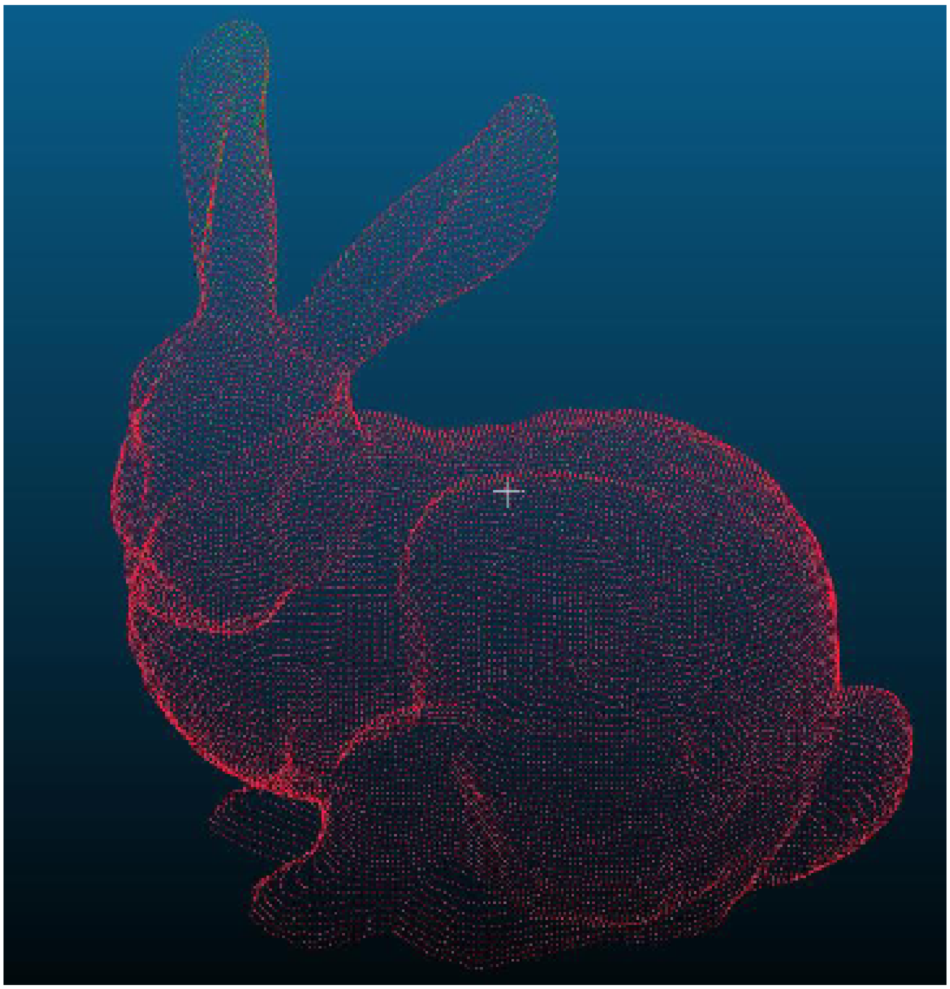


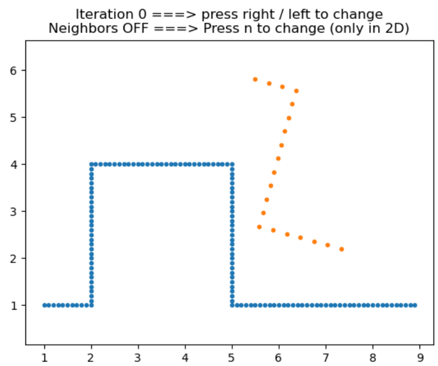
Figure 5: Result of alignment of point clouds with the rigid$_{t}ransformfunction$.

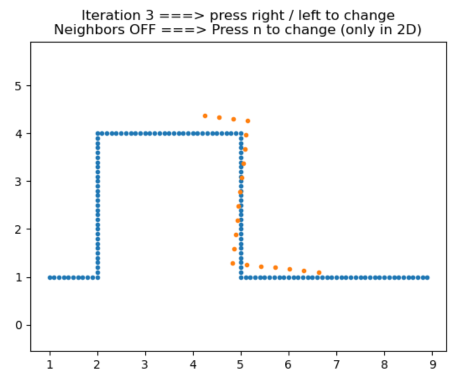
Fig.5 shows the result of the rigid transform function on the original point cloud of the rabbit and the returned rabbit. The RMS between the returned cloud and the reference cloud was **0.16** (fig.4).

After transformation, the RMS drops to around 10-8.

This method worked well and even **better than the ICP method** of CloudCompare because we ensured that R was a rotation, while the CloudCompare ICP, without an initial alignement, cannot work properly. Besides, it uses the fact that **the order of the points is the same in both models** (as if all the points of the cloud were matched targets). For example, **this would not work for the Notre Dame** : the two cloud points do not have the same number of points (dimension error)

3 Question 3





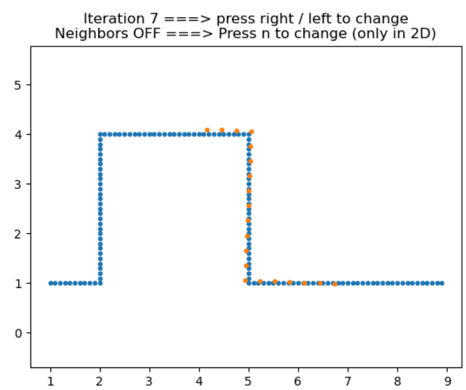


Figure 7: 2D points before and after ICP alignment

Plot the RMS during ICP convergence for those the 2D example :

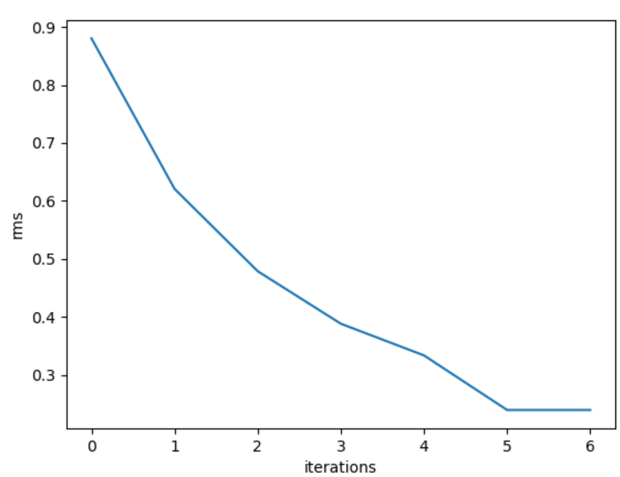
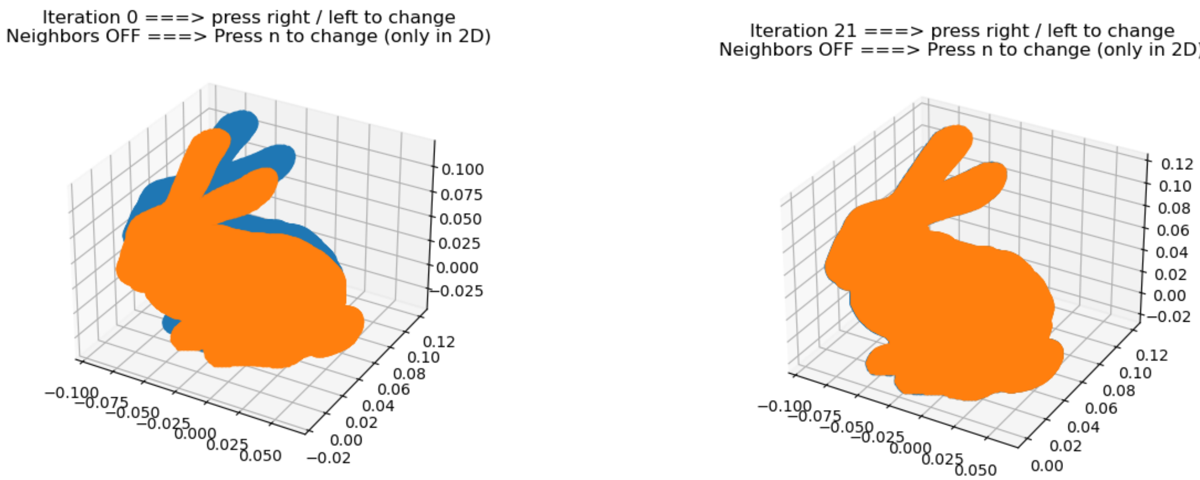


Figure 8: Graph of the RMS value duritng ICP iteration for the data2D image

And we have the following convergence results for the bunny with the 3D plot :



(a) 3D point cloud before (left) and after ICP alignment (right)

And the plot the RMS during ICP convergence for those the bunny example :

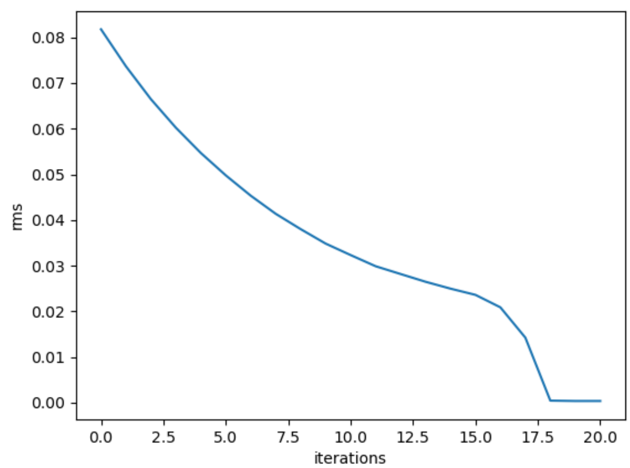


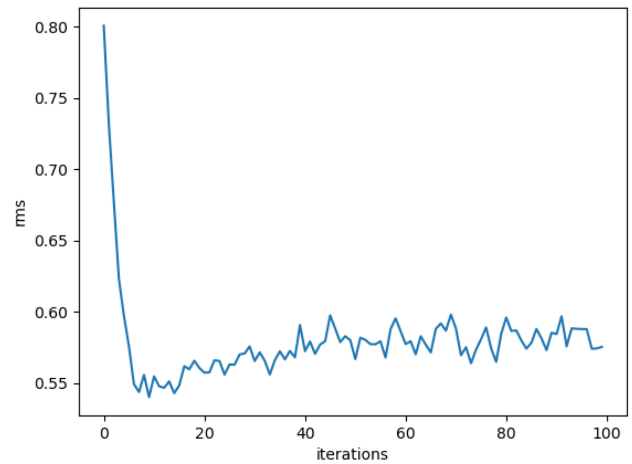
Figure 10: Graph of the RMS values during ICP iterations for the bunny volume

4 Question 4

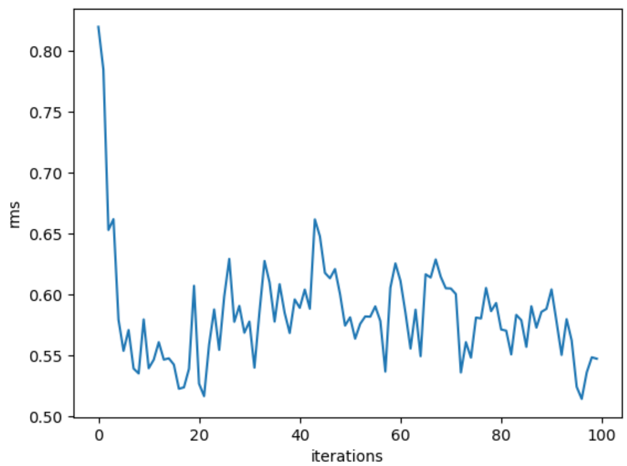
In both cases, the algorithm is able to converge to a very good solution. We can see that increasing the number of points also increases the number of iterations required.

5 Bonus

(b) RMS ICP convergence for 10000 points



(a) RMS ICP convergence for 1000 points



And the graph below with the two curves combined :

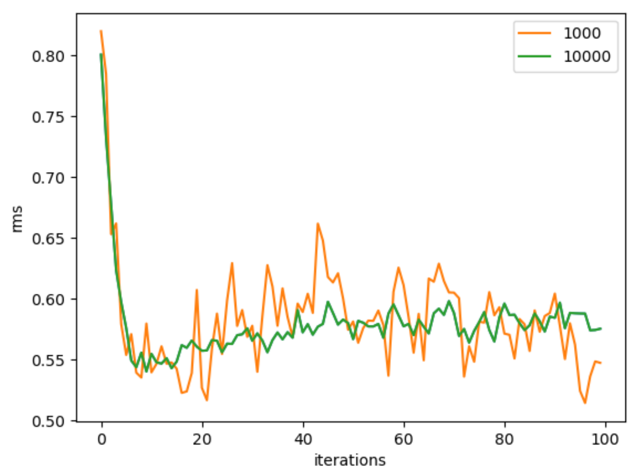


Figure 12: Graph of the RMS values during ICP iterations for the bunny volume

We can see that it does not really converge, with a RMS largely over zero. With more points (10 000), the variance (noise due to the choice of the points) is reduced and the curve becomes even smoother.