

Question 1: ICP

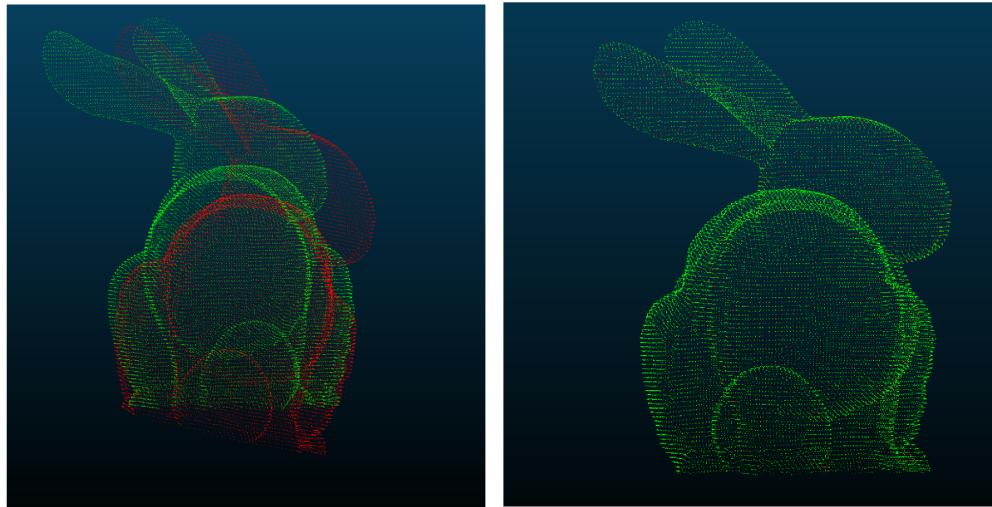


Figure 1: Left: reference cloud in green (template to match), misaligned candidate cloud in red. Right: After ICP alignment in Cloud Compare, RMSE = $7.2 \cdot 10^{-8}$

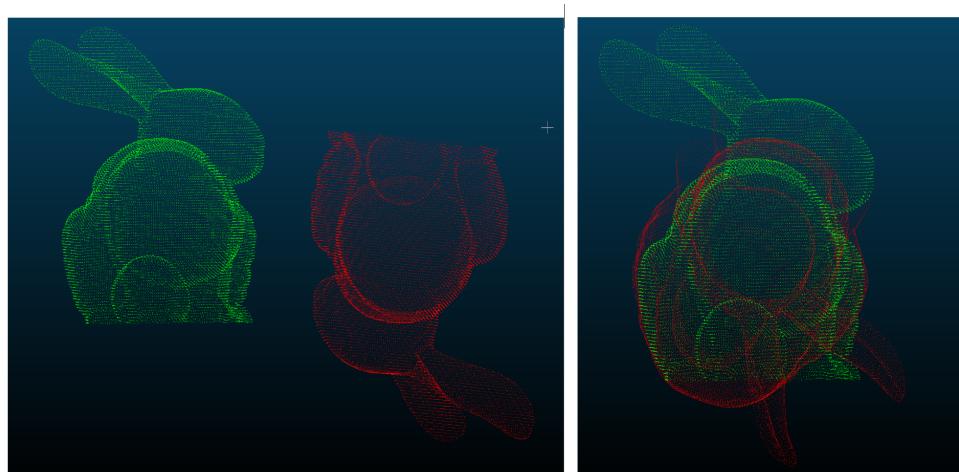


Figure 2: Left: reference cloud in green (template to match), misaligned candidate cloud in red (upside down initialization). Right: After ICP alignment in Cloud Compare, root mean squared error RMSE = 0.013

The ICP algorithm is very sensitive to initialization. For the bunny example (same geometry without noise), two different initializations provide very different results. Figure 1 shows a nearly perfect alignment, while Figure 2 shows a bad alignment probably due getting stuck in a local minimum.

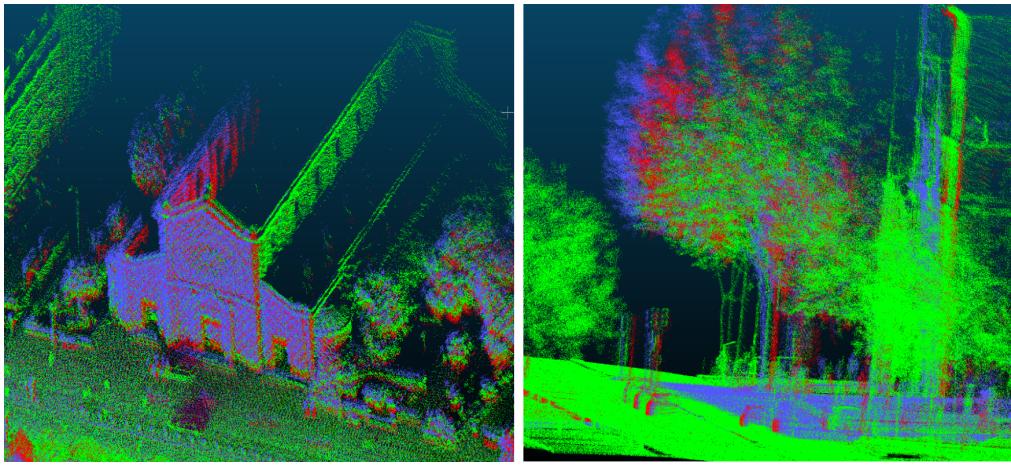


Figure 3: Reference cloud in green - the largest cloud, misaligned candidate cloud in blue and ICP registration result in red RMSE = 1.39. We're off by more than a meter in average after alignment, we observe that objects have moved between these scenes so without outlier removal, we can't naturally expect a perfect alignment no matter the quality of the registration algorithm.

The estimated transformation matrix is quite close to the identity matrix since the misaligned was not pretty tiny. Seems like most of the misalignment source came from the ground not being perfectly aligned (the blue ground is not horizontal in figure 4)

```
Transformation matrix - Scale: fixed (1.0)
0.999 0.008 0.032 6.017
-0.010 0.998 0.060 -8.688
-0.031 -0.060 0.998 -67.759
0.000 0.000 0.000 1.000
```

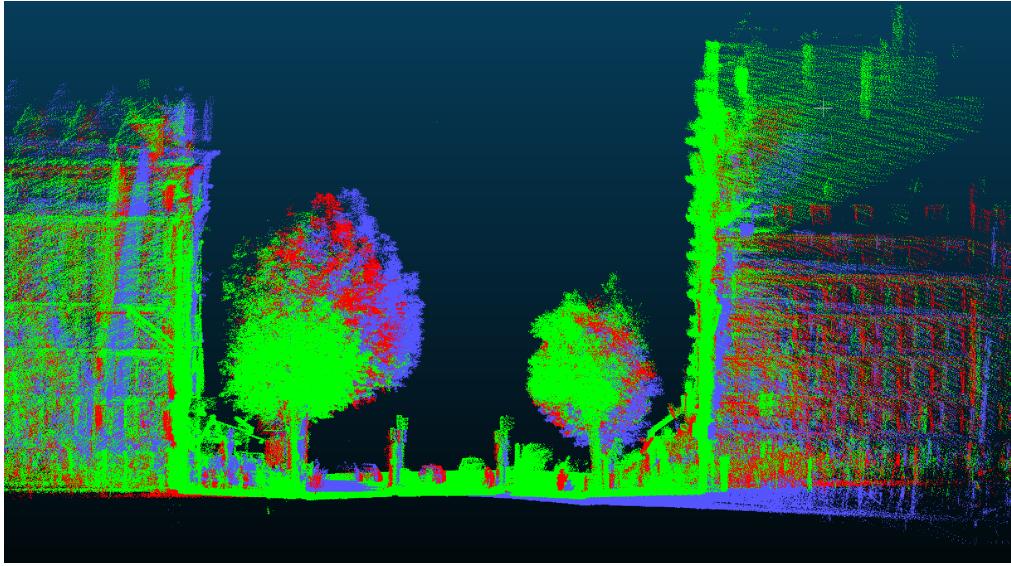


Figure 4: Most of the misalignment seem to have been on the Z axis - ground is not horizontal in the blue misaligned scene.

We use the largest cloud as reference cloud (Notre dame des champs 1). It seems more natural as the small candidate cloud can potentially match all its points to the bigger reference cloud (this does not work the other way around).

Question 2: Rigid transformation estimation

- We are dealing with 2 point clouds where the points are matched and there is no noise between them.
- We obtain a perfect alignment RMS = 0.. Works perfectly because the input points are perfectly matched (no noise) and we use a **closed form solution** for the rigid transformation estimation.
- ICP struggles because of a bad initialization and gets stuck in a local minimum (translation was not too bad though, see figure 2).

For the real raw scene "Notre Dame des champs", we do not have a matching between points (and we don't have the same amount of points.). One would have to first find matches, which is the purpose of ICP studied in the next section.

Questions 3 and 4: ICP convergence

2D toy example

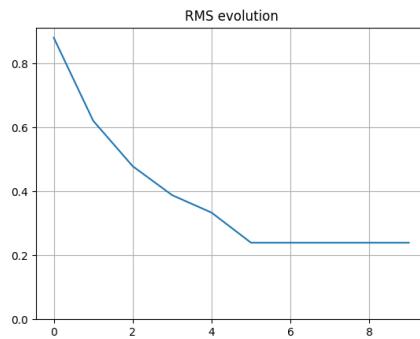


Figure 5: ICP 2D toy example: RMS error as a function of the number of iterations.

After the fifth iteration, the algorithm finished converging $\text{RMSE} = 0.2395$ as seen in 5. It cannot reach zero because it's simply not doable, see the zoom in figure 7.

In the 2D toy example, we're matching a large reference point cloud (in blue) and a candidate smaller point cloud (in orange). ICP will find the best rotation and 2D translations (3 degrees of freedom) and find the matches between the two point clouds (the whole candidate cloud with a subset of the reference cloud). Matches are shown in green in figure 6.

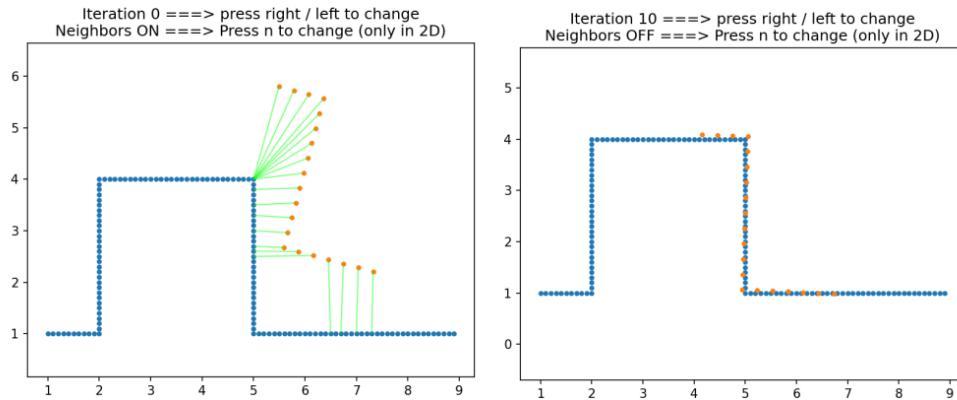


Figure 6: ICP 2D toy example: nearest neighbor search pairs (matching in green). Initialization on the left, end of convergence on the right side.

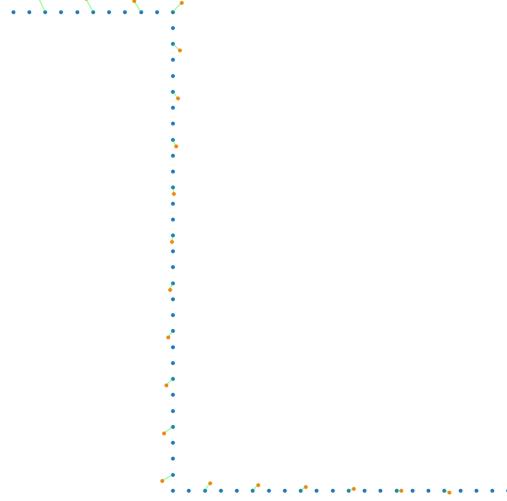


Figure 7: Zoom on the registered 2D point clouds after the fifth iteration. The algorithm has converged and due to the nature of the 2D point clouds, it does not seem possible to reach a smaller RMS error although the nature of our brain would prefer the vertical and horizontal lines to be aligned even if the distance was higher.

3D ICP

Just like we had noticed in the rigid alignment, the two bunnies can be perfectly aligned ($RMSE = 0$). There is a **convergence acceleration** near the end, probably due to the fact although visually we don't see much differences in terms of transformation between steps 15 to 18. The algorithm is able to suddenly refine the matches. (*Visualizing the rotation and translation errors would probably show that the convergence was almost finished earlier.*)

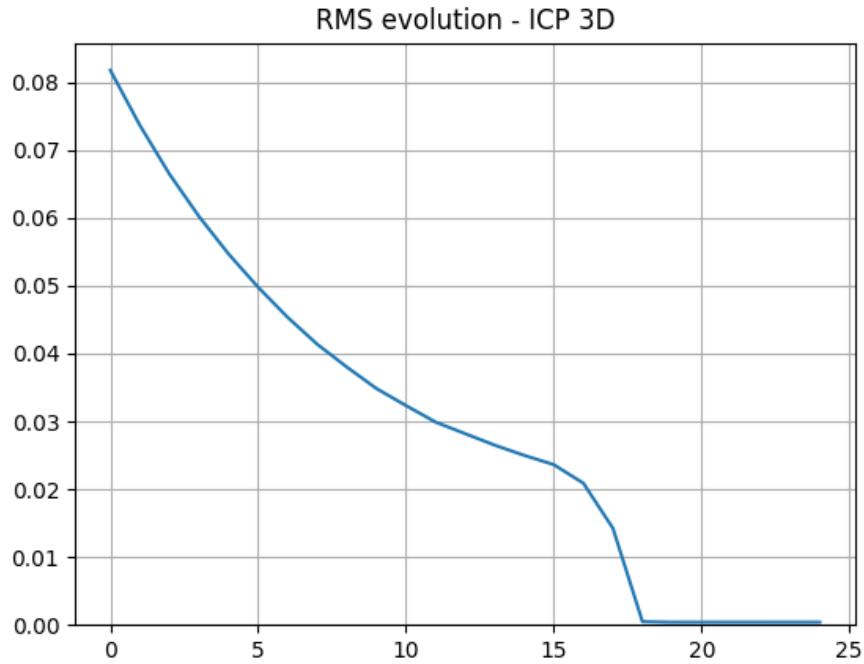


Figure 8: ICP 3D RMSE. Convergence

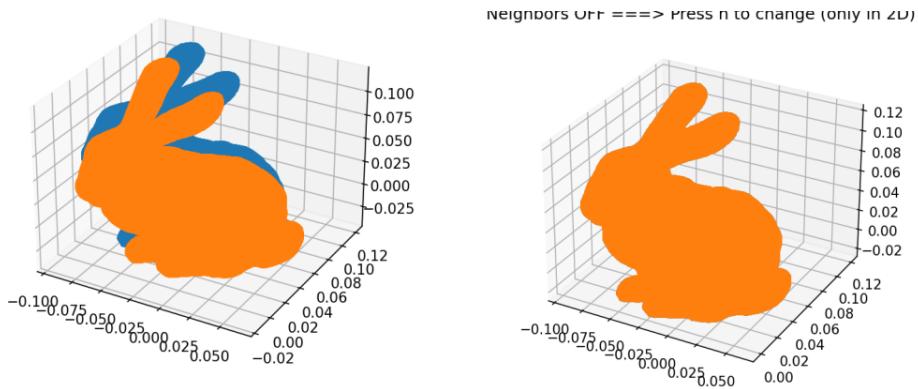


Figure 9: ICP 3D example: Initialization on the left, end of convergence on the right side. Registration in the end here.

Bonus question: ICP on large point clouds

We're able to align the 2 large point clouds in approximately 5 seconds with an error of roughly 50cm. Building the KD tree takes quite some time (4.6s). Error seems to slightly increase, one would have to keep an history of the transformations to keep the best one. But the issue is that to pick the best registration, you'd need to compute on a lot of points. Whether using 1k or 10k points, the minimum RMSE is not so different but the approximation of the RMSE on the current randomly selected points is quite noisy. Using 10k points provides less noisy RMSE estimates (see figure 11) and given the KD tree configuration picked, is actually even faster than using 1k points. (see figure 10).

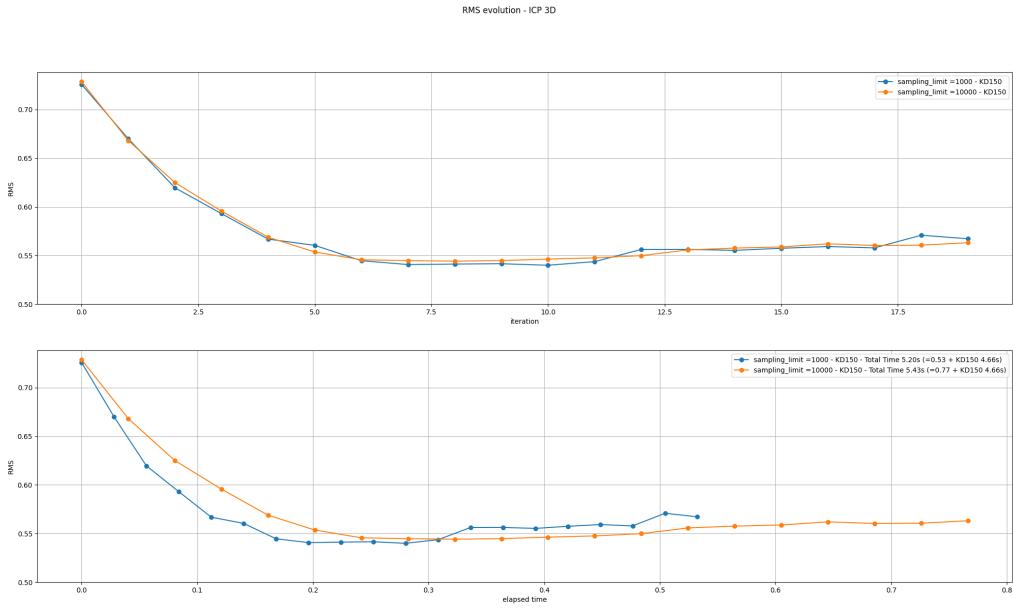


Figure 10: True RMSE (estimated over the whole point cloud) when doing ICP over a random selection of 1000 and 10.000 matched points.

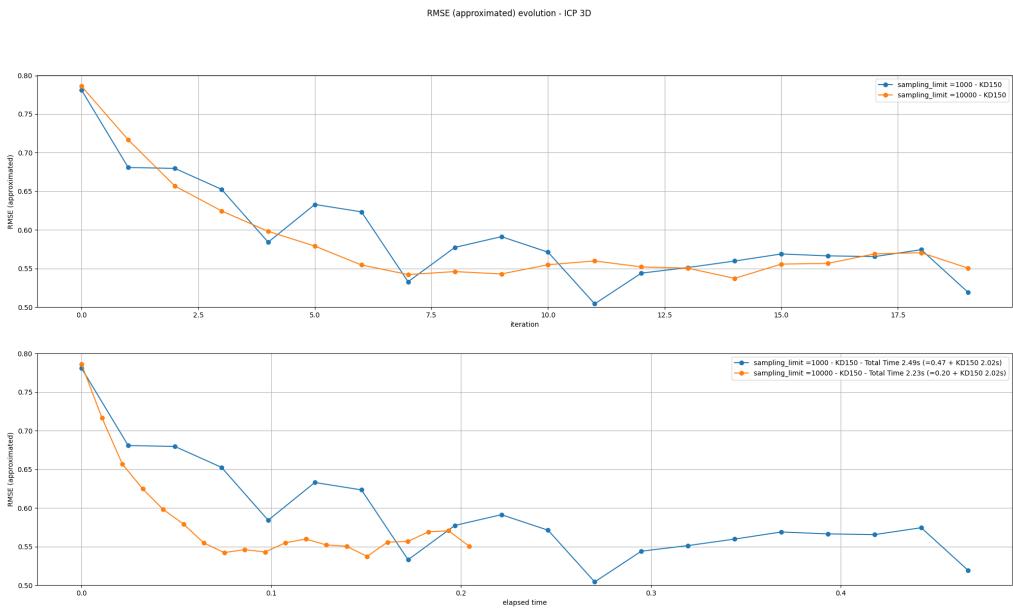


Figure 11: RMSE approximated over 1000 and 10.000 matched points. This would be the criterion to decide if the algorithm has converged. Bottom: Top: horizontal axis is the number of iterations. Bottom: horizontal axis is the time spent in the registration, KD tree construction is omitted but takes a few seconds.