TP2 - ICP Algorithm for Point Cloud Registration

NPM3D - M2 MVA

Tong ZHAO (tong.zhao@eleves.enpc.fr)

1 CloudCompare ICP

Q1. First of all, we perform ICP algorithm on Stanford Bunny. The file *bunny_original.ply* is used as the reference point cloud and *bunny_pertubed.ply* is used as the aligned point cloud. The result is shown below. We can observe the the two point clouds are perfectly aligned. The final RMS is nearly 0.

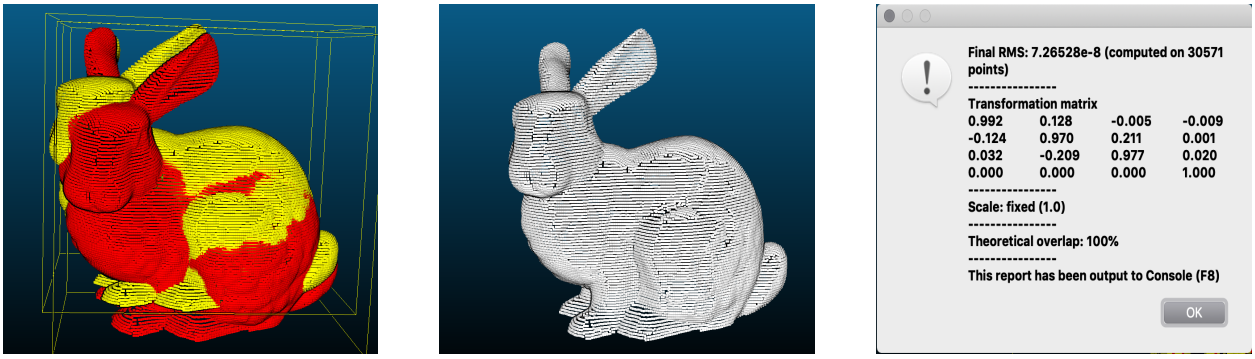

Figure 1: (a) Two original point clouds (b) aligned bunny (c) algorithm conclusion

Then we apply the algorithm on the NDDC example. The file *Notre_Dame_Des_Champs_1.ply* is used as the reference point cloud and the file *Notre_Dame_Des_Champs_2.ply* is used as the aligned point cloud. The reason of this setting is that the second file is a part of the first file, meaning that many points in the first file don't have matching points in the second file. If we calculate the RMS error by aligning the first to the second, the missing points will produce a large error which perturbs the algorithm. We visualize the result as following.

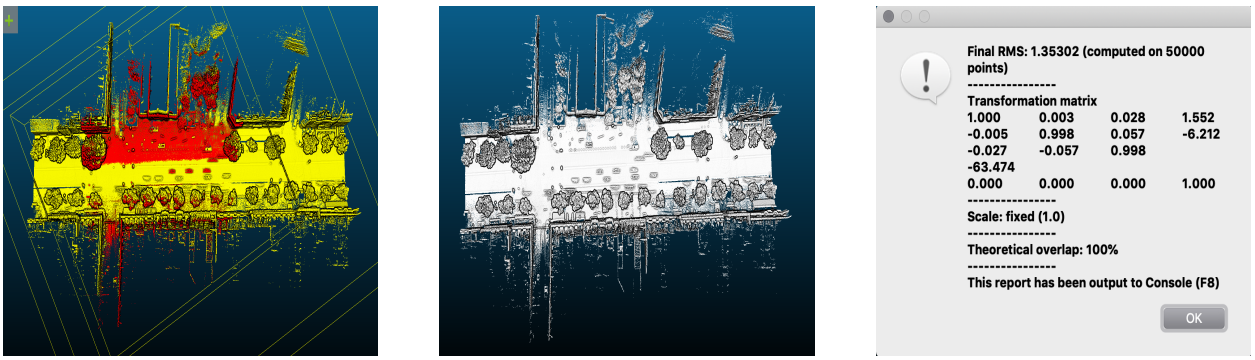

Figure 2: (a) Two original point clouds (b) aligned point cloud (c) algorithm conclusion

It is much worse than the one of bunny because that the two point clouds are not perfectly matched. Thus the default parameters does not work well with this case.

2 Rigid Transformation Between Matched Set of Points

Q2. The implemented rigid transform function works well on the bunny example. The final RMS is 0. A comparaison between our function and CloudCompare ICP is shown as below.

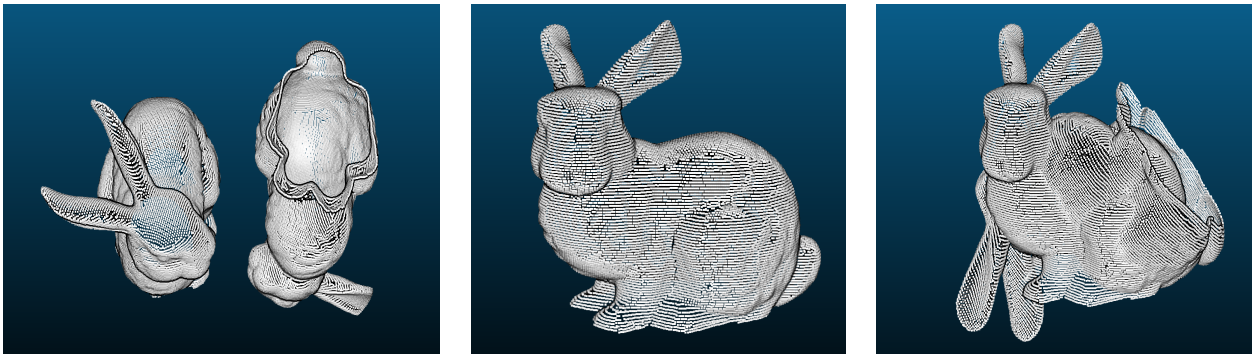

Figure 3: (a) Two original point clouds (b) aligned point cloud by function (c) aligned point cloud by CloudCompare

The failure of CloudCompare ICP is due to the main assumption of the function that the two point clouds should be roughly registered. We should also realize that the function can not align the real 3D scans of *Notre Dame des Champs*. The two point clouds have different point numbers and they are not matched.

3 Point to Point ICP

Q3. We use the implemented ICP function to align at first the 2D data.

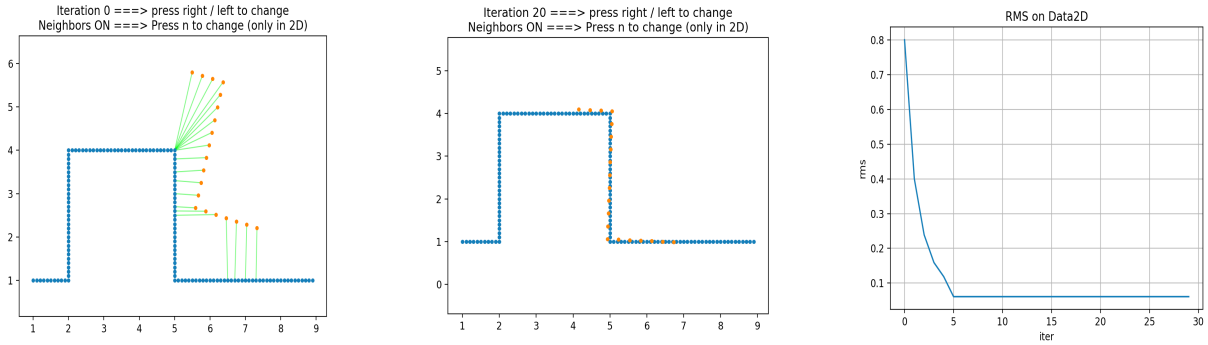

Figure 4: (a) Two original point clouds (b) aligned point cloud (c) RMS convergence

Then we apply this function to align the bunny example.

Figure 5: (a) Two original point clouds (b) aligned point cloud (c) RMS convergence

Q4. By using the NDDC example, we apply the implemented ICP function with different sampling limits. The following figures represent respectively the aligned results with $n = \{1000, 10000, 50000\}$.

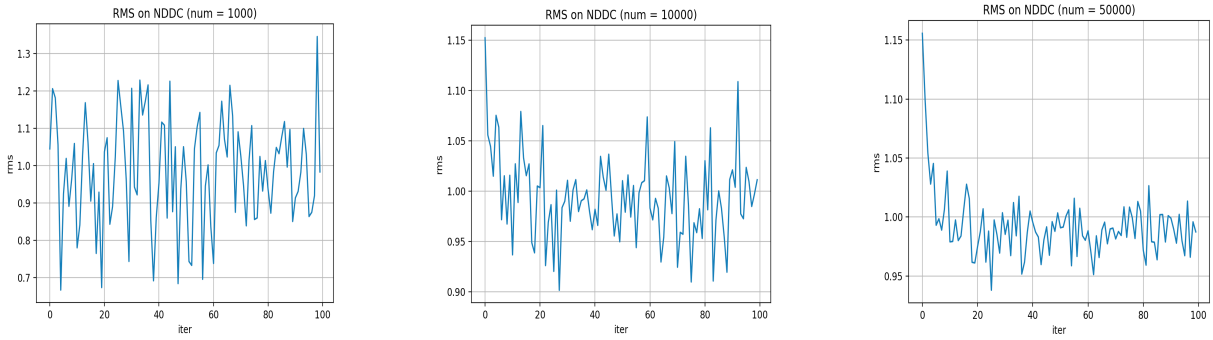

Figure 6: (a) $n = 1000$ (b) $n = 10000$ (c) $n = 50000$

We can see clearly from the results that a larger sampling limit leads to a better convergence. It is logical since that the convergence is guaranteed if we match all points., however the time-consuming is expensive.

4 Going Further

Q5. We compute the cloud to cloud distance of the NDDC example and the result is shown as below.

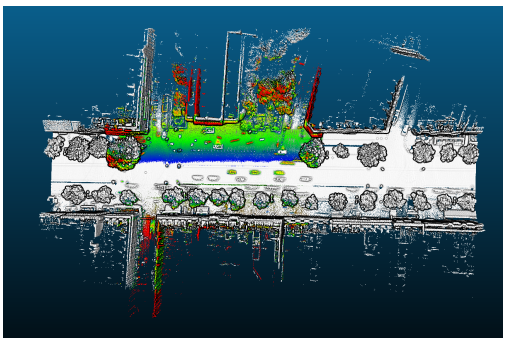

Figure 7: Cloud to cloud distance on NDDC example

Clearly the red points don't have a corresponding point in the reference point cloud. However during the algorithm, each time when we choose such points, they are matched with some points in the reference point cloud. Thus they perturbs the ICP convergence. As a consequence, if we calculate RMS on the whole data cloud, the errors caused by these matchings also gives us a bad estimation. So it is better to consider the RMS on all points having a good match point in the reference cloud.

Bonus 1. We compare the results using different final overlap parameters. The candidates are $\{20\%, 50\%, 70\%, 90\%\}$. We provide a script to calculate the 50% RMS, 70% RMS, 90% RMS and the complete RMS to evaluate the alignments. The table is shown as below.

RMS	CC Best	90% Overlap	70% Overlap	50% Overlap	20% Overlap
Complete RMS	1.007242	1.007519	1.007867	1.007576	1.008434
90% RMS	0.112029	0.114533	0.112429	0.112539	0.116532
70% RMS	0.068567	0.073628	0.070219	0.069142	0.069611
50% RMS	0.050900	0.055819	0.052496	0.051487	0.051220

Bonus 2. We add a parameter *final_ overlap* in the function *icp_ point_ to_ point_ stochastic* which removes the farthest points in each iteration. The best result is achieved by using *final_ overlap* = 0.7 and *sampling_ limit* = 100000. The aligned points clouds produced by our implementation and the one of CloudCompare are shown as below:

Figure 8: (a) CloudCompare's best align (b) My best align

The statistical results are shown as below:

RMS	Complete RMS	90% RMS	70% RMS	50% RMS
CC Best	1.007242	0.112029	0.068567	0.050900
My Best	1.006823	0.111940	0.068666	0.050996

We can observe that the our best result is slightly better than the one of CloudCompare.