

TP2 - Nuages de Points et Modélisation 3D

Chloé Habasque - Colin Coërchon

22 Janvier 2025

1 CloudCompare ICP

1.1 Question 1

Dans ce TP, nous retrouvons ainsi notre petit lapin en nuages de points 3D. Nous allons nous essayer à l'implémentation de l'algorithme ICP (Iterative Closest Point).

Partie 1 :

L'objectif ici est de tester l'algorithme ICP déjà implémenté dans le logiciel "CloudCompare".

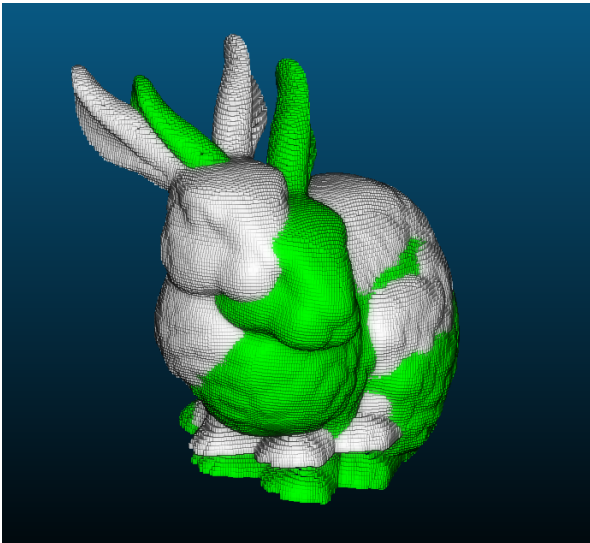


FIGURE 1 – Le lapin original (en blanc) et le lapin perturbé (en vert)

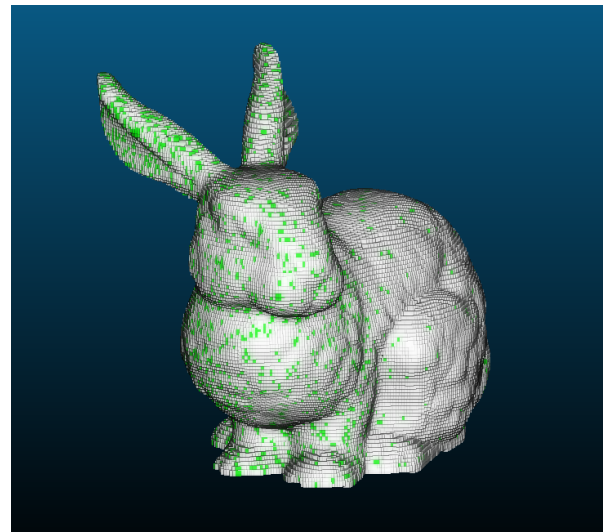


FIGURE 2 – Le lapin original (en blanc) et le lapin perturbé aligné par ICP (en vert)

Dans ce premier exemple, l'ICP marche parfaitement, et nous obtenons un très bon résultat présenté dans la figure 2. Le logiciel donne comme erreur finale : **Final RMS = 1.1644e-08**.

Partie 2 :

Nous testons maintenant cet algorithme ICP sur un lapin retourné.

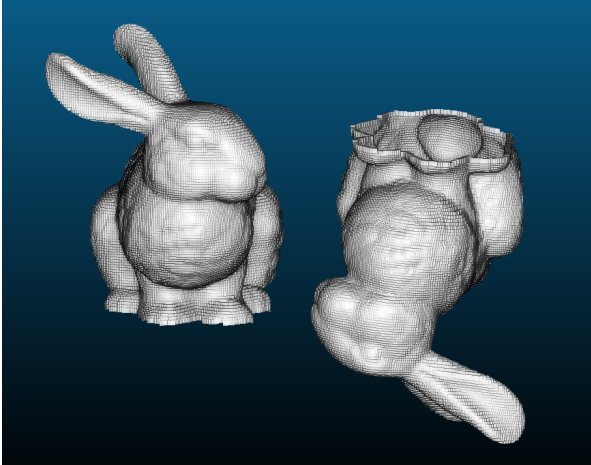


FIGURE 3 – Le lapin original et le lapin retourné

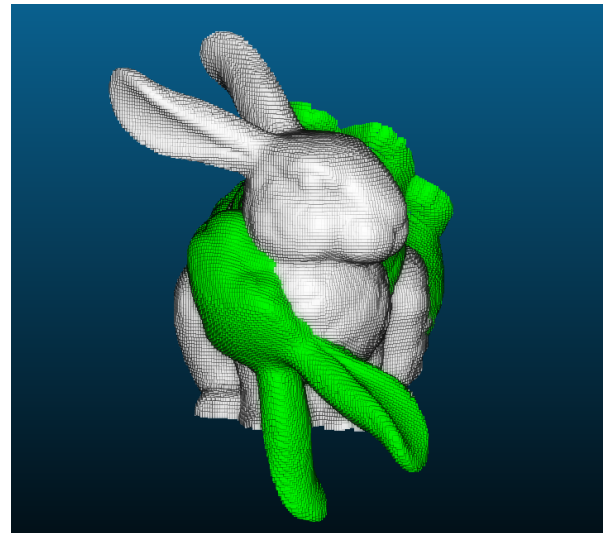


FIGURE 4 – Le lapin original (en blanc) et le lapin retourné aligné par ICP (en vert)

Dans cet exemple-ci, l'algorithme ICP ne fonctionne pas du tout comme le montre la figure 4. Le lapin aligné par ICP ne se superpose pas avec le lapin original. Le logiciel donne comme erreur finale : `Final RMS = 0.0133466`.

Partie 3 :

L'idée ici est maintenant de faire tourner cet algorithme ICP sur un nuage de points plus important ("Notre Dame Des Champs").

En premier lieu, nous prenons `Notre_Dame_Des_Champs_1` en référence. Le nuage à aligné est donc `Notre_Dame_Des_Champs_2`. Le résultat est donné figure 5.

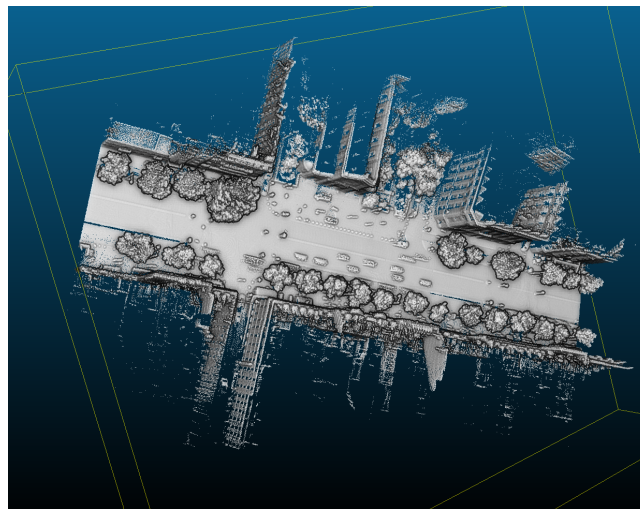


FIGURE 5 – Referent : `Notre_Dame_Des_Champs_1`, Aligned : `Notre_Dame_Des_Champs_2`

Le logiciel donne comme erreur finale : `Final RMS = 1.32`.

En second lieu, on inverse les deux nuages de points.

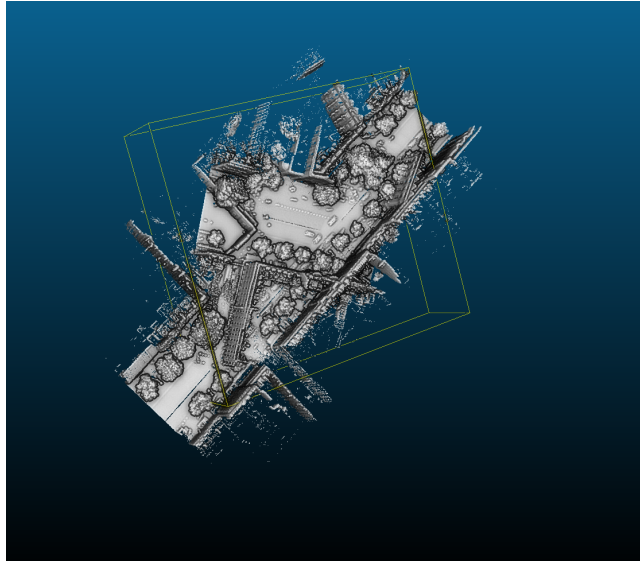


FIGURE 6 – Referent : Notre_Dame_Des_Champs_2, Aligned : Notre_Dame_Des_Champs_1

Le logiciel donne comme erreur finale : `Final RMS = 12.66`.

Le nuage de points référent doit donc être `Notre_Dame_Des_Champs_1.ply` qui contient 4 482 219 points, contre 1 907 902 points pour `Notre_Dame_Des_Champs_2.ply`. Sinon on obtient un **mauvais alignement** comme le présente la figure 6.

Réponse à la question 1 :

L'algorithme ICP fonctionne bien dans certains cas, mais échoue dans d'autres :

- Dans le premier exemple (lapin perturbé), l'alignement est parfait (`Final RMS = 1.1644e-08`).
- Dans le deuxième exemple (lapin retourné), l'alignement échoue (`Final RMS = 0.0133466`) car l'orientation du lapin retourné est peut-être trop différente du lapin original pour l'algorithme.

La différence entre le **nuage aligné** et le **nuage de référence** est que le nuage de référence reste fixe, tandis que le nuage aligné subit des transformations pour minimiser l'erreur RMS.

Ainsi, dans le cas de Notre Dame, **il est préférable de choisir le nuage contenant le plus grand nombre de points comme référence**, car l'algorithme associe chaque point du nuage "aligned" à un point du nuage de référence. Ainsi, le calcul est simplifié et plus efficace, car il y a moins de points à parcourir dans le nuage aligné. Cela permet un meilleur alignement, comme le montre la figure 5.

2 Rigid transformation between matched set of points

2.1 Question 2 :

Cette question portait sur l'implémentation de la fonction `best_rigid_transform` en python. Le pseudo-code de cette fonction est donné dans le polycopié et est rappelé dans le TP. Voici son implémentation :

```
def best_rigid_transform(data, ref):
    R = np.eye(data.shape[0])
    T = np.zeros((data.shape[0], 1))
```

```

P, P1 = ref, data
Pm = np.mean(P, axis=1).reshape(-1, 1)
Pm1 = np.mean(P1, axis=1).reshape(-1, 1)

Q = P - Pm
Q1 = P1 - Pm1

H = Q1 @ Q.T
U, S, Vt = np.linalg.svd(H)    # On teste avec H et H.T

R = Vt.T @ U.T
T = Pm - R @ Pm1

if np.linalg.det(R) < 0:
    print("Reflection detected")
    Vt[-1] *= -1
    R = Vt.T @ U.T

return R, T

```

Dans cette implémentation, la matrice $Q1$ joue le rôle de la matrice Q' du cours.

Et voici les résultats que nous obtenons sur l'alignement du lapin retourné en prenant comme référence le lapin original :

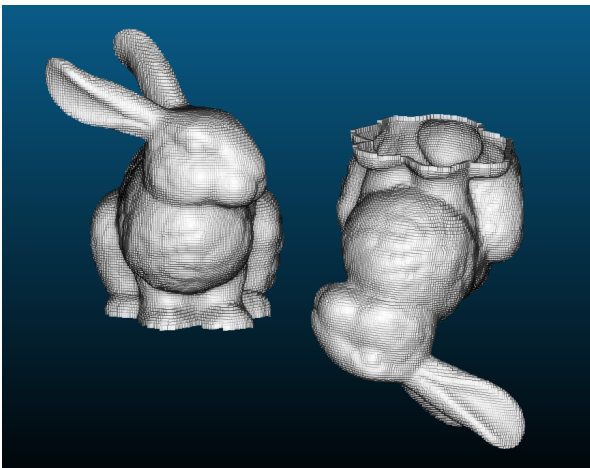


FIGURE 7 – Le lapin original et le lapin retourné

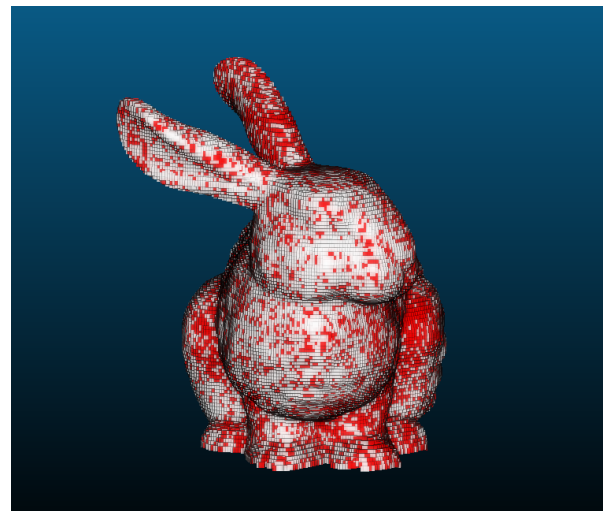


FIGURE 8 – Le lapin original (en blanc) et le lapin retourné aligné par ICP (en rouge)

L'algorithme donne comme erreur RMS moyenne :

- RMS Before = 0.16083363
- RMS After = 0.00000002

Le résultat est donc **parfait** puisque on obtient une précision de l'ordre de 10^{-8} dans la RMS.

Réponse à la question 2 :

L'alignement a fonctionné avec la fonction `best_rigid_transform` alors que CloudCompare ICP échouait, car cette fonction calcule directement la transformation rigide optimale entre deux ensembles de points correspondants, **sans dépendre d'une initialisation ou d'une convergence itérative comme l'ICP**. Cela élimine les problèmes liés à des mauvaises correspondances initiales, avec une orientation bien différente comme dans le cas du lapin retourné par exemple.

Cependant, pour les fichiers `Notre_Dame_Des_Champs_1` et `Notre_Dame_Des_Champs_2`, contenant plusieurs millions de points, la fonction `best_rigid_transform` est inefficace en raison de la taille des matrices nécessaires, cela entraînerait des limitations de mémoire et de temps de calcul.

3 Point to point ICP

3.1 Question 3 :

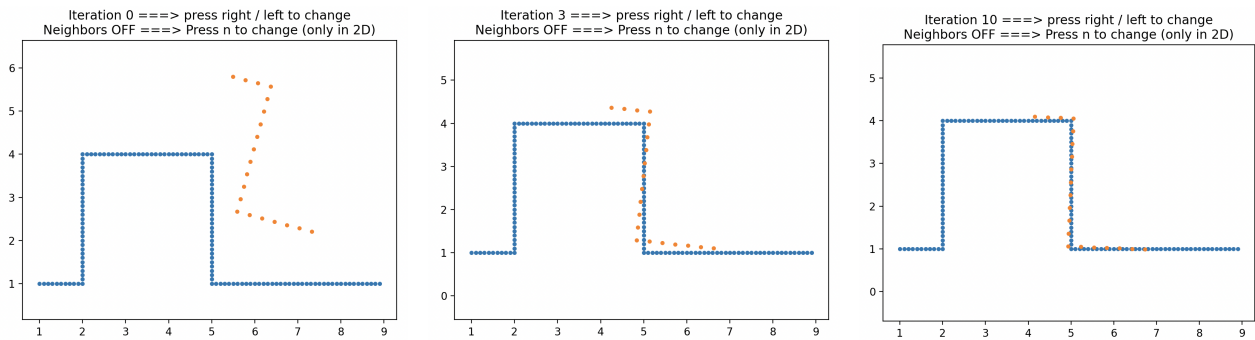


FIGURE 9 – Etapes de l'alignement pour l'image 2D

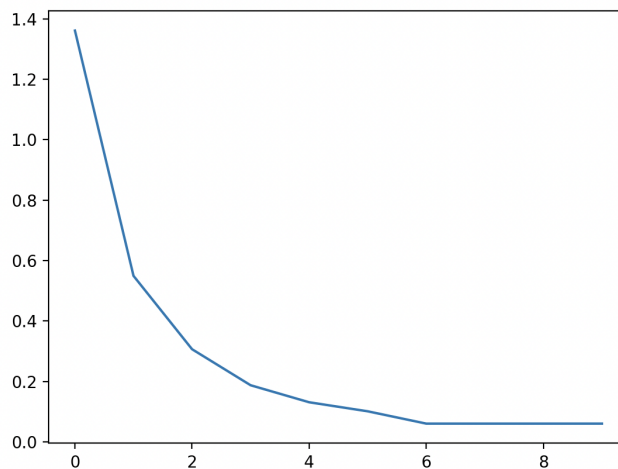
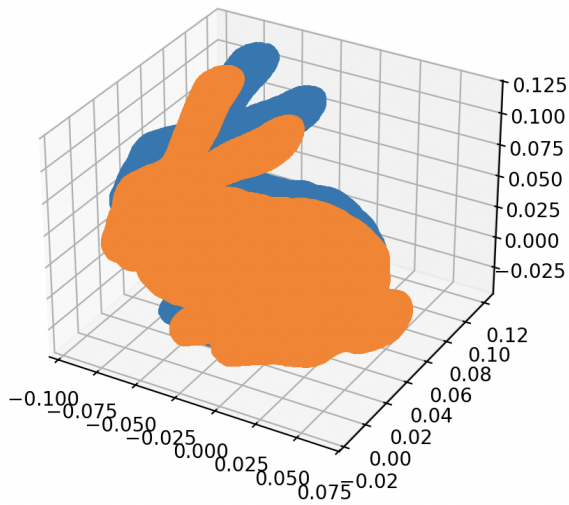
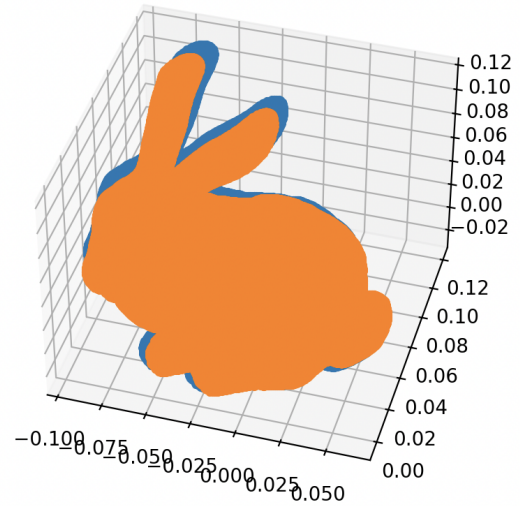


FIGURE 10 – Évolution de l'erreur RMS au fil des itérations de l'ICP pour l'image 2D

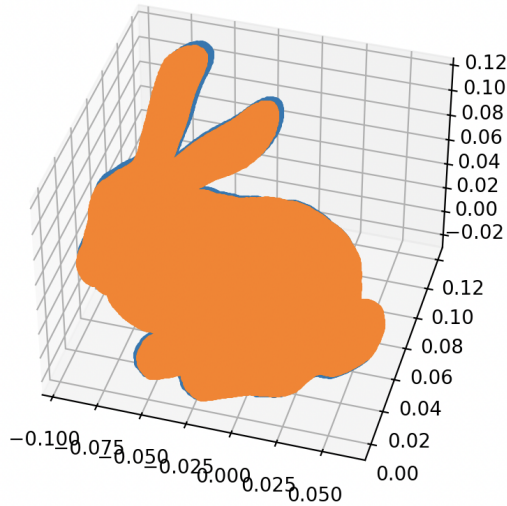
Iteration 0 ==> press right / left to change
Neighbors OFF ==> Press n to change (only in 2D)



Iteration 5 ==> press right / left to change
Neighbors OFF ==> Press n to change (only in 2D)



Iteration 10 ==> press right / left to change
Neighbors OFF ==> Press n to change (only in 2D)



Iteration 19 ==> press right / left to change
Neighbors OFF ==> Press n to change (only in 2D)

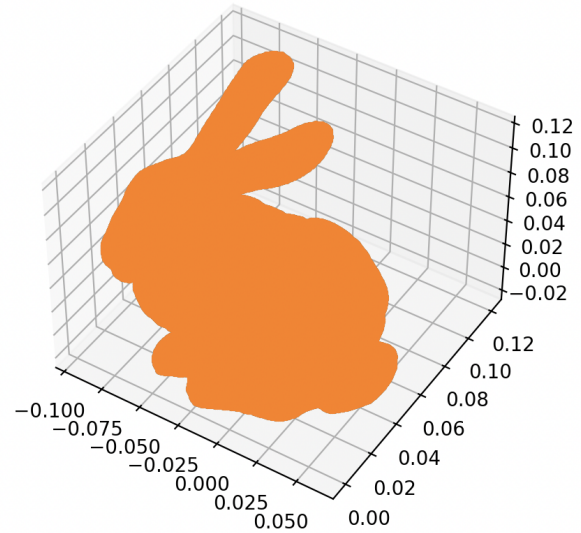


FIGURE 11 – Étapes de l'alignement du lapin perturbé (orange) sur le lapin original (bleu)

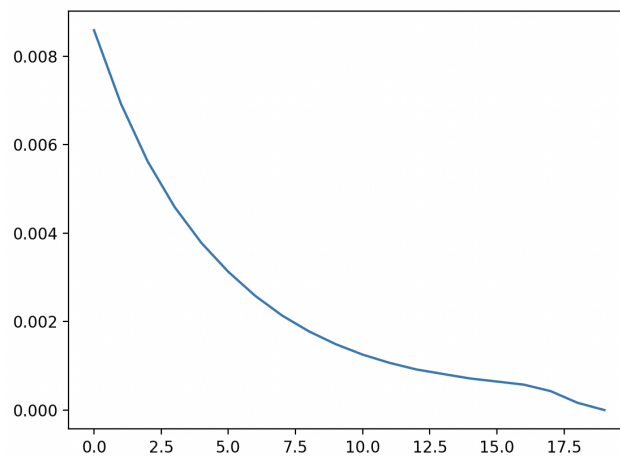


FIGURE 12 – Évolution de l'erreur RMS au fil des itérations de l'ICP pour les lapins

3.2 Question 4 :

Dans les deux cas (image 2D et le lapin perturbé), l'erreur RMS converge vers 0 au bout de respectivement 10 et 20 itérations de notre algorithme ICP.

Les résultats sont donc excellents et correspondent aux résultats de l'ICP du logiciel CloudCompare.

4 Going further (BONUS)

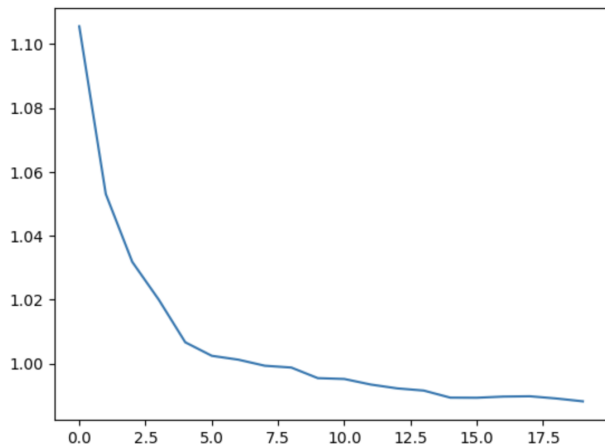


FIGURE 13 – RMS avec un alignement de 1000 points

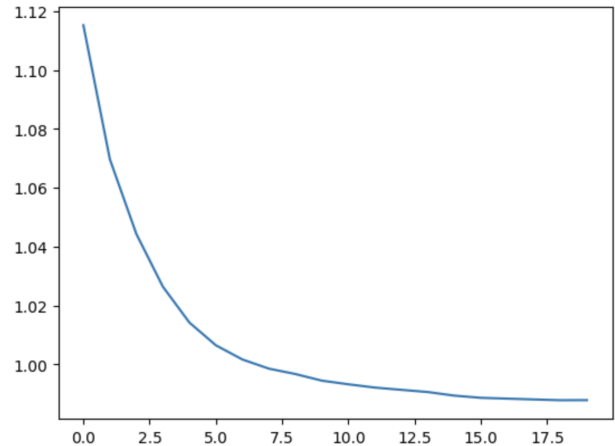


FIGURE 14 – RMS avec un alignement de 10000 points

On obtient une erreur RMS, après 20 itérations, de **0.98** que ce soit en alignant avec 1000 points ou 10 000 points pris aléatoirement, voir [figure 13](#) et [figure 14](#). L'erreur est légèrement meilleure que l'alignement avec le logiciel qui donnait une erreur RMS de **1.32**.

Iteration 20 ==> press right / left to change
Neighbors OFF ==> Press n to change (only in 2D)

Iteration 20 ==> press right / left to change
Neighbors OFF ==> Press n to change (only in 2D)

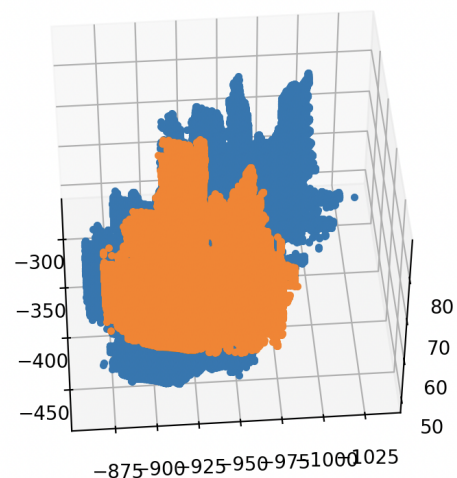
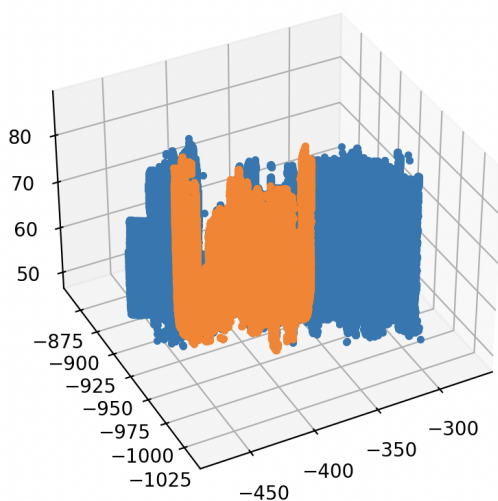


FIGURE 15 – Résultat de l'alignement à la 20ème itération