



TP Recalage de données 3D avec ICP.

CAIPA PRIETO Julian Andres

École Nationale Supérieure des Arts et Métiers

Computer Vision

Enseignant : Fakhreddine Ababsa

PA10

Campus de Paris, France

2024

Table des matières

Table des figures	3
1 Introduction	5
2 Objectifs	6
3 ICP simple	7
4 ICP avec bruit	9
5 Autres variantes de l'algorithme ICP : Avec Open3D	12
5.1 ICP depuis Scratch vs ICP <i>Open3D</i>	12
5.2 ICP Point-à-Point vs ICP généralisé	15
6 Conclusions	18
Bibliographie	19

Table des figures

Figure. 1	Nuages de points originaux.	8
Figure. 2	Résultats de l'algorithme ICP.	8
Figure. 3	Courbe de l'erreur de recalage.	9
Figure. 4	Nuages avec bruit gaussien.	10
Figure. 5	Résultats de l'algorithme ICP nuages avec bruit.	10
Figure. 6	Comparaison de la matrice de rotation entre les nuages normaux et les nuages bruités.	10
Figure. 7	Comparaison du vecteur de translation entre les nuages normaux et les nuages bruités.	11
Figure. 8	Comparaison du temps d'exécution entre les nuages normaux et les nuages bruités.	11
Figure. 9	Comparaison de l'erreur entre les nuages normaux et les nuages bruités.	11
Figure. 10	Erreur ICP simple et avec bruit.	11
Figure. 11	Résultats de l'algorithme ICP <i>Open3D</i>	14
Figure. 12	Comparaison de la matrice de rotation entre l'ICP Scratch et ICP en <i>Open3D</i>	14
Figure. 13	Comparaison du vecteur de translation entre l'ICP Scratch et ICP en <i>Open3D</i>	14
Figure. 14	Comparaison du temps d'exécution entre l'ICP Scratch et ICP en <i>Open3D</i>	14
Figure. 15	Comparaison de l'erreur entre l'ICP Scratch et ICP en <i>Open3D</i>	15
Figure. 16	Résultats de l'algorithme ICP généralisé <i>Open3D</i>	16
Figure. 17	Comparaison de la matrice de rotation entre l'ICP point-à-point et ICP généralisé.	16
Figure. 18	Comparaison du vecteur de translation entre l'ICP point-à-point et ICP généralisé.	16
Figure. 19	Comparaison du temps d'exécution entre l'ICP point-à-point et ICP généralisé.	16
Figure. 20	Comparaison de l'erreur entre l'ICP point-à-point et ICP généralisé.	17

Table des codes

1	Ajout de bruit.	9
2	Nuages dans <i>Open3D</i>	12
3	ICP en <i>Open3D</i>	13
4	Normaux et covariances.	15

1 Introduction

Dans ce travail pratique, nous abordons le recalage de nuages de points 3D, une problématique clé en vision par ordinateur, particulièrement utile dans les domaines de la modélisation 3D, de la robotique et de la navigation. Ce projet met en œuvre différentes approches pour aligner deux ensembles de points 3D, en utilisant l'algorithme Iterative Closest Point (ICP) ainsi que ses variantes avancées.

Le processus suit une méthodologie incrémentale : génération des données, implémentation de l'algorithme ICP à partir de zéro, exploration des impacts du bruit, et évaluation des variantes de l'algorithme à travers des outils graphiques et statistiques. Chaque étape vise à renforcer la compréhension des concepts fondamentaux liés au recalage 3D et à analyser la performance des algorithmes.

L'objectif final est de fournir une compréhension approfondie du fonctionnement de l'ICP, tout en explorant ses limitations et ses avantages lorsqu'il est appliqué à des données complexes, notamment dans des contextes bruités.

2 Objectifs

Le but de ce TP est d'implémenter une méthode de recalage de données 3D vue en cours, en l'occurrence l'algorithme ICP et ses variantes. Nous verrons notamment comment résoudre ce problème populaire d'estimation en utilisant Python, d'une part, et la librairie Open3D, d'autres parts (1).

3 ICP simple

Dans la première partie de ce rapport, nous étudions la méthode ICP en la réalisant de la manière la plus simple possible. Une fonction appelée ICP, écrite depuis zéro, permet de mettre en œuvre l'algorithme en utilisant une approche itérative point à point. Le fichier des fonctions (*icp_functions.py*) contient cette fonction dûment commentée et expliquée. La fonction reçoit deux nuages de points : l'un fixe (de référence) et l'autre mobile (qui se déplace). Elle retourne la matrice optimale de rotation, le vecteur optimal de translation, le vecteur contenant les erreurs à chaque itération, ainsi que le vecteur des temps d'exécution pour chaque itération. Une boucle *while* exécute l'algorithme jusqu'à ce qu'un seuil d'erreur ou un nombre maximal d'itérations soit atteint. Cette boucle contient, de manière générale, les étapes clés de la fonction qui sont les suivantes :

- Une boucle *for* qui calcule les points les plus proches dans le nuage fixe Q pour chaque point du nuage mobile P , en utilisant la distance euclidienne. Ces points correspondants dans Q sont stockés dans un "nuage mappé" (ou "carte de points"), utilisé pour effectuer les calculs nécessaires à l'alignement optimal des deux nuages, c'est-à-dire pour trouver la meilleure transformation possible.
- Le calcul des centroïdes pour P et la carte de points. Ces centroïdes servent à recentrer les nuages autour d'eux, afin d'éliminer toute translation préalable et de faciliter le calcul de la rotation pure en utilisant le théorème de décomposition en valeurs singulières (SVD).
- Le calcul de la matrice de covariance, qui exprime la relation entre P centré et le nuage de référence.
- L'application du SVD sur la matrice de covariance, permettant d'obtenir la matrice de rotation qui minimise l'erreur quadratique moyenne. Cela repose sur les propriétés des bases orthonormées : V , T décrit l'orientation des données dans P et U celle des données dans Q . Leur produit représente ainsi la rotation qui minimise l'erreur.
- Le calcul du vecteur de translation, puis la mise à jour de la nouvelle position du nuage P en appliquant la minimisation de la fonction de coût fondamentale de l'ICP.
- Le calcul de l'erreur quadratique moyenne et du temps écoulé pour chaque itération.

Enfin, à la sortie de la boucle *while*, la matrice de rotation R et le vecteur de translation T optimaux sont mis à jour, et les variables souhaitées sont retournées.

Cette fonction ICP a été modifiée pour retourner R , T , le temps d'exécution et les erreurs comme demandé.

Une fonction appelée *ShowData* a été créée, permettant de visualiser dynamiquement les nuages de points à l'aide de la bibliothèque *Plotly*, en particulier du module *graph_objects*. Cette fonction rend possible l'interaction avec les nuages de points de manière interactive : le graphique 3D peut être manipulé, déplacé, et les deux nuages de points peuvent être observés sous différents angles. La fonction reçoit les deux nuages de points en entrée et les affiche sur un plan tridimensionnel. En utilisant cette fonction pour visualiser les nuages générés par *DataGeneration*, on obtient :

3D Point Cloud Visualization

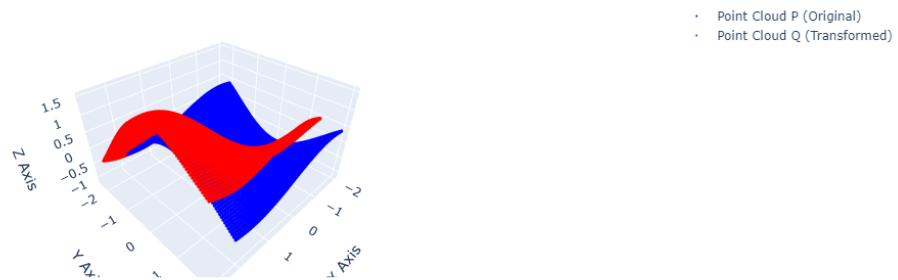


FIGURE 1 – Nuages de points originaux.

Suite à cette visualisation, l'algorithme ICP est appliqué à l'aide de la fonction créée. Celui-ci fournit les résultats de la matrice de rotation optimale, le vecteur de translation optimal, ainsi que le nombre d'itérations et le temps pris par celles-ci.

```
ICP completed in 27 iterations and 6.83 seconds
Optimal Rotation Matrix (R_opt):
[[ 9.99999995e-01  3.60776090e-06 -1.01512557e-04]
 [-3.60862229e-06  1.00000000e+00 -8.48537363e-06]
 [ 1.01512527e-04  8.48573991e-06  9.99999995e-01]]

Optimal Translation Vector (T_opt):
[ 0.0071674 -0.00016605  0.00175143]
```

FIGURE 2 – Résultats de l'algorithme ICP.

Il est également possible de visualiser le comportement de l'erreur au cours des itérations. Cela permet de comprendre comment l'approximation est progressivement corrigée à chaque itération jusqu'à atteindre le seuil d'erreur défini.

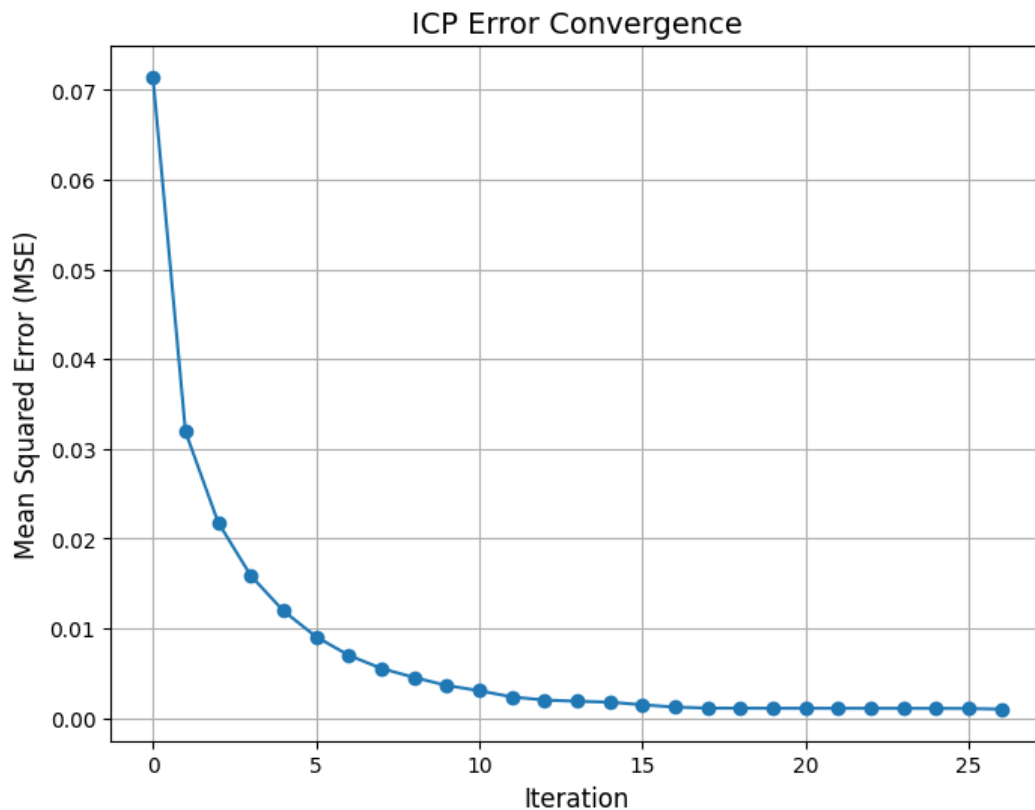


FIGURE 3 – Courbe de l'erreur de recalage.

De cette courbe, on peut constater que l'algorithme remplit adéquatement sa fonction. De plus, il s'avère être efficace en termes de calcul : l'erreur diminue rapidement à des valeurs très faibles, suivant une décroissance exponentielle, avant d'atteindre le seuil fixé. Ce comportement est attendu compte tenu de la précision requise.

4 ICP avec bruit

L'idée de cette deuxième partie est d'ajouter du bruit aux nuages de points et d'étudier le comportement de l'algorithme ICP. Pour ce faire, un bruit gaussien est ajouté à l'aide de la fonction `np.random.normal` de la bibliothèque `NumPy`, centré et avec un écart-type de 0.01.

```

1 noise_std = 0.01
2 P_noisy = P + np.random.normal(0, noise_std, P.shape)
3 Q_noisy = Q + np.random.normal(0, noise_std, Q.shape)

```

Code 1 – Ajout de bruit.

En ajoutant ce bruit, les nuages de points apparaissent comme le montre la figure suivante.

3D Point Cloud Visualization

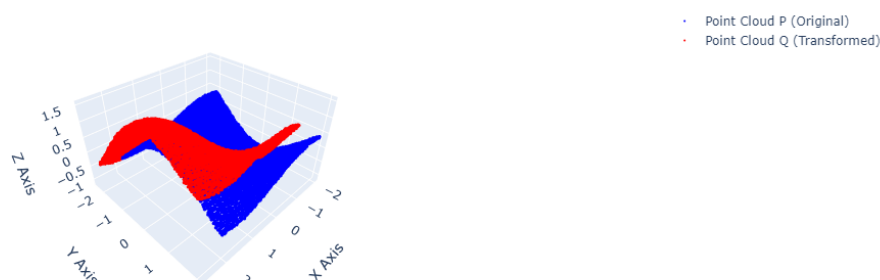


FIGURE 4 – Nuages avec bruit gaussien.

Et on obtient les résultats de l'ICP.

```
ICP completed in 22 iterations and 17.50 seconds
Optimal Rotation Matrix (R_opt) with noise:
[[ 9.99997743e-01  1.96552245e-03  8.07224162e-04]
 [-1.96476484e-03  9.99997630e-01 -9.38253383e-04]
 [-8.09066407e-04  9.36665259e-04  9.99999234e-01]]

Optimal Translation Vector (T_opt) with noise:
[0.00608297 0.00089997 0.00305952]
```

FIGURE 5 – Résultats de l'algorithme ICP nuages avec bruit.

Une fonction appelée *MatrixComparer* a été réalisée pour visualiser la comparaison entre les matrices et les statistiques de chaque méthode. Cette fonction présente tous les résultats dans des tableaux avec une notation scientifique pour faciliter leur interprétation. En comparant les résultats de l'ICP avec les nuages de points normaux et les nuages de points bruités, les observations suivantes ont été faites :

- Pour la matrice de rotation, la différence élément par élément était très faible, ce qui indique que les matrices sont relativement similaires, sans être exactement identiques. Cela s'explique par le fait que, bien que les valeurs soient faibles, elles sont dans la même échelle que celles de la matrice de rotation.

Element	Noiseless	Noisy	Difference Abs
R[0,0]	1.e+00	1.e+00	2.25e-06
R[0,1]	3.61e-06	1.97e-03	1.96e-03
R[0,2]	-1.02e-04	8.07e-04	9.09e-04
R[1,0]	-3.61e-06	-1.96e-03	1.96e-03
R[1,1]	1.e+00	1.e+00	2.37e-06
R[1,2]	-8.49e-06	-9.38e-04	9.3e-04
R[2,0]	1.02e-04	-8.09e-04	9.11e-04
R[2,1]	8.49e-06	9.37e-04	9.28e-04
R[2,2]	1.e+00	1.e+00	7.61e-07

FIGURE 6 – Comparaison de la matrice de rotation entre les nuages normaux et les nuages bruités.

- Pour le vecteur de translation, les différences étaient globalement faibles, mais légèrement plus élevées que celles observées pour la matrice de rotation. Cela montre que le bruit a davantage impacté la translation.

Component	Noiseless	Noisy	Difference Abs
Tx	7.17e-03	6.08e-03	1.08e-03
Ty	-1.66e-04	9.e-04	1.07e-03
Tz	1.75e-03	3.06e-03	1.31e-03

FIGURE 7 – Comparaison du vecteur de translation entre les nuages normaux et les nuages bruités.

- Le temps d'exécution a été sérieusement affecté, augmentant de plus de 10 secondes. Cela indique que la rapidité du recalage a diminué, ce qui est normal compte tenu de l'ajout de bruit.

Metric	Noiseless	Noisy	Difference Abs (s)
Time	6.78e+00	1.75e+01	1.07e+01

FIGURE 8 – Comparaison du temps d'exécution entre les nuages normaux et les nuages bruités.

- L'erreur à la dernière itération est pratiquement identique, ce qui est logique puisqu'elle dépend d'un seuil prédéfini.

Metric	Noiseless	Noisy	Difference Abs
Error	9.87e-04	9.39e-04	4.78e-05

FIGURE 9 – Comparaison de l'erreur entre les nuages normaux et les nuages bruités.

Ensuite, il est possible d'analyser l'évolution de l'erreur au fil des itérations pour comparer les deux méthodes.

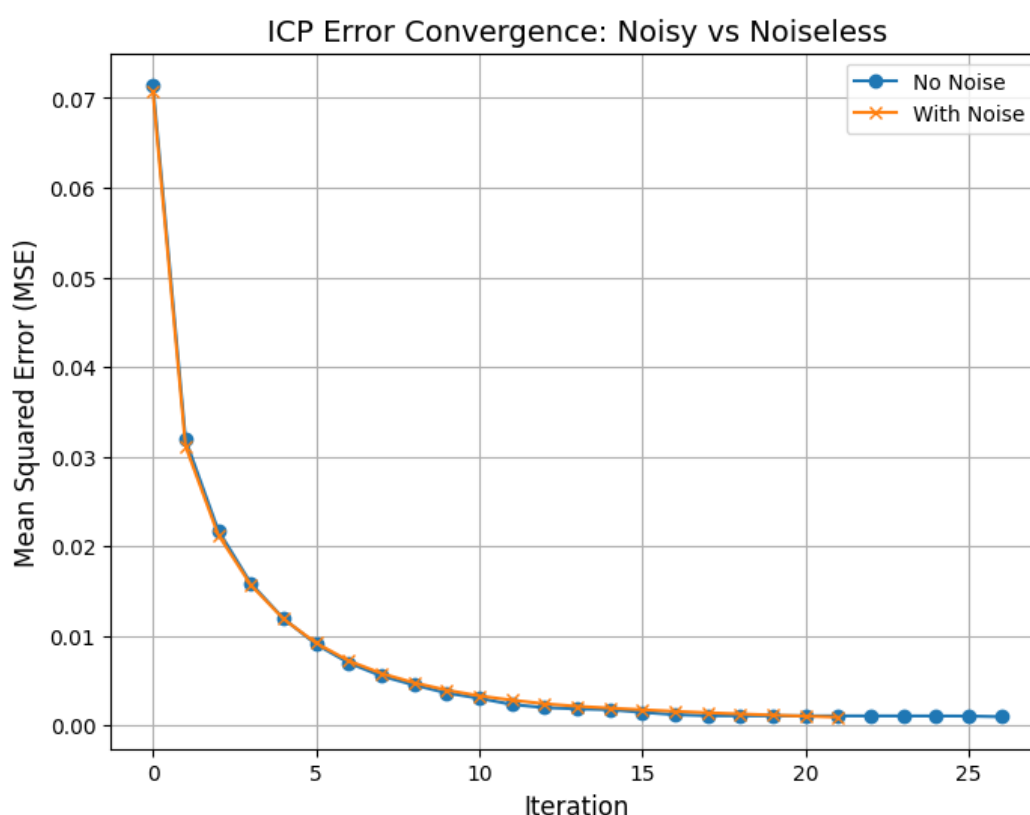


FIGURE 10 – Erreur ICP simple et avec bruit.

D'après le graphique de l'erreur, on peut conclure que, bien que l'ICP sans bruit nécessite plus d'itérations, l'ICP avec bruit présente généralement des erreurs plus élevées tout au long des itérations. Cela indique que l'algorithme nécessite plus de calculs pour ajuster les nuages de points, ce qui est évident dans le temps de calcul plus long.

De manière générale, on peut conclure que l'ICP appliqué aux nuages bruités demande plus de calculs, mais que la précision de la méthode est conservée. Cela est dû en grande partie au fait que le bruit ajouté est relativement faible. Si le bruit était plus important, les conditions changeraient et la précision de l'algorithme pourrait être compromise. Cela s'explique par l'augmentation des calculs et le risque accru de dépasser les seuils d'itérations, ce qui pourrait entraîner un recalage incorrect.

5 Autres variantes de l'algorithme ICP : Avec Open3D

Dans la partie finale du travail, l'objectif est d'étudier différentes variantes de l'algorithme ICP en utilisant la bibliothèque Open3D, qui contient une implémentation optimisée de l'algorithme, afin de comparer les performances de l'algorithme réalisé à partir de zéro, puis d'analyser la variante généralisée de l'algorithme ICP et de la comparer à sa version classique.

5.1 ICP depuis Scratch vs ICP Open3D

Pour commencer, la bibliothèque *Open3D* propose la méthode *registration_icp* pour appliquer l'algorithme de recalage. Il est d'abord nécessaire de convertir les nuages de points au format requis par *Open3D*. Une fonction simple est ainsi créée pour transformer les tableaux *NumPy* en nuages exploitables dans *Open3D*.

```
1 numpy_to_open3d(np_points):  
2     pcd = o3d.geometry.PointCloud()  
3     pcd.points = o3d.utility.Vector3dVector(np_points)  
4     return pcd  
5  
6 P_pcd = numpy_to_open3d(P)  
7 Q_pcd = numpy_to_open3d(Q)
```

Code 2 – Nuages dans *Open3D*.

Avec les nuages au format *Open3D*, le recalage ICP est effectué en spécifiant les nuages à utiliser, la transformation initiale (dans ce cas, la matrice identité), le critère de convergence (100 itérations et une valeur RMS (Root Mean Square Error) équivalente au seuil de 0,001 utilisé dans l'algorithme manuel avec MSE), ainsi que deux paramètres importants qui méritent une explication plus détaillée, la distance maximale de correspondance et la méthode d'estimation.

```
1 max_correspondence_distance = 0.05
2 init_transformation = np.eye(4)
3
4 convergence_criteria = o3d.pipelines.registration.ICPConvergenceCriteria(
5     relative_fitness=1e-5,
6     relative_rmse=0.0316,
7     max_iteration=100
8 )
9
10 start_time = time.time()
11 result_icp = o3d.pipelines.registration.registration_icp(
12     source=P_pcd,
13     target=Q_pcd,
14     max_correspondence_distance=max_correspondence_distance,
15     init=init_transformation,
16     estimation_method=o3d.pipelines.registration.
17     TransformationEstimationPointToPoint(),
18     criteria=convergence_criteria
19 )
20 end_time = time.time()
21 elapsed_time = end_time - start_time
```

Code 3 – ICP en *Open3D*.

La distance maximale de correspondance est un paramètre absent de l'algorithme réalisé à la main et qui vise à l'optimiser. Ce paramètre agit comme un seuil lors de la comparaison des points du nuage mobile P avec ceux du nuage fixe Q . Dans l'algorithme manuel, la différence entre un point du nuage P et tous les points du nuage Q est calculée pour trouver les correspondances les plus proches dans Q pour chaque point de P . Cependant, il ne vérifie pas si la distance minimale est suffisamment petite pour être considérée comme une correspondance valide. Ainsi, chaque point est associé, quelle que soit la distance par rapport au nuage fixe. Le seuil de distance maximale de correspondance permet d'éviter cela : si la distance entre un point P et son point correspondant Q est supérieure à ce seuil, la correspondance est rejetée. Cela évite d'introduire des erreurs dans le calcul de la transformation ou de ralentir la convergence.

Quant à la méthode d'estimation, elle fait référence au type d'ICP utilisé dans l'algorithme. Les deux méthodes les plus courantes sont point-à-point, qui minimise les distances entre des paires de points correspondants, et point-à-plan, qui minimise la distance entre un point et un plan normal au nuage de référence.

En utilisant la méthode de *Open3D* on obtient :

```

Transformation Matrix:
[[ 9.98267919e-01  2.68056474e-02 -5.23700164e-02  2.23888810e-02]
 [-2.78819461e-02  9.99412517e-01 -1.99303516e-02 -2.30028653e-02]
 [ 5.18050040e-02  2.13560086e-02  9.98428847e-01 -6.65788858e-04]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]

Inlier RMSE: 0.03293642719375333
Fitness: 0.08015625
ICP completed in 0.07 seconds

```

FIGURE 11 – Résultats de l'algorithme ICP *Open3D*.

Avec l'aide de MatrixComparer, les résultats entre l'ICP réalisé à partir de zéro et l'ICP de *Open3D* avec la méthode point-à-point ont été analysés, car le point-à-point est la méthode utilisée dans la version manuelle. Il a été constaté que :

- Pour la matrice de rotation, la différence terme à terme était relativement faible, soulignant que dans les deux méthodes, les signes restent identiques. Cependant, dans la méthode manuelle, les valeurs varient considérablement entre elles, avec des valeurs dans la matrice très faibles par rapport à d'autres, par exemple l'entrée [0,0] par rapport à l'entrée [1,0], tandis que dans la méthode *Open3D*, les valeurs sont plus uniformes.

Element	Scratch	Open 3D	Difference Abs
R[0,0]	1.e+00	9.98e-01	1.73e-03
R[0,1]	3.61e-06	2.68e-02	2.68e-02
R[0,2]	-1.02e-04	-5.24e-02	5.23e-02
R[1,0]	-3.61e-06	-2.79e-02	2.79e-02
R[1,1]	1.e+00	9.99e-01	5.87e-04
R[1,2]	-8.49e-06	-1.99e-02	1.99e-02
R[2,0]	1.02e-04	5.18e-02	5.17e-02
R[2,1]	8.49e-06	2.14e-02	2.13e-02
R[2,2]	1.e+00	9.98e-01	1.57e-03

FIGURE 12 – Comparaison de la matrice de rotation entre l'ICP Scratch et ICP en *Open3D*.

- Pour le vecteur de translation, en général, les différences sont faibles, avec le même phénomène d'une plus grande uniformité dans la méthode *Open3D*.

Component	Scratch	Open 3D	Difference Abs
Tx	7.17e-03	2.24e-02	1.52e-02
Ty	-1.66e-04	-2.30e-02	2.28e-02
Tz	1.75e-03	-6.66e-04	2.42e-03

FIGURE 13 – Comparaison du vecteur de translation entre l'ICP Scratch et ICP en *Open3D*.

- Le temps a été sérieusement affecté, avec une différence de 6,71 secondes. Cela est dû au fait que la méthode *Open3D* est extrêmement optimisée et génère une convergence beaucoup plus rapide que celle obtenue avec la méthode manuelle, pour diverses raisons, notamment les critères de distance maximale de correspondance expliqués précédemment.

Metric	Scratch	Open 3D	Difference Abs (s)
Time	6.78e+00	7.00e-02	6.71e+00

FIGURE 14 – Comparaison du temps d'exécution entre l'ICP Scratch et ICP en *Open3D*.

- L'erreur à la dernière itération est pratiquement identique, ce qui est logique puisqu'il s'agit d'un seuil défini.

Metric	Scratch	Open 3D	Difference Abs
Error	3.14e-02	3.29e-02	1.52e-03

FIGURE 15 – Comparaison de l'erreur entre l'ICP Scratch et ICP en *Open3D*.

En général, il apparaît que, bien que l'algorithme manuel soit assez rapide, efficace et précis, l'algorithme de la bibliothèque *Open3D* est bien meilleur à tous égards, car il trouve une transformation rigide beaucoup plus conforme et équilibrée, et il est beaucoup plus rapide en termes de temps d'exécution. Même si on lui permet d'itérer au-delà du seuil d'erreur imposé, l'erreur diminue considérablement sans consommer beaucoup plus de temps, ce qui témoigne de son efficacité.

5.2 ICP Point-à-Point vs ICP généralisé

Ensuite, la version généralisée de l'ICP est implémentée pour la comparer à la version point-à-point. La version généralisée de l'ICP de *Open3D* est une combinaison du point-à-point et du point-à-plan, permettant de calculer dynamiquement les correspondances et le recalage des nuages en simultané avec des correspondances entre points et plans normaux aux nuages. Pour implémenter la méthode, deux modifications sont nécessaires.

Premièrement, il faut calculer les normales et les covariances. Pour cela, on utilise des fonctions de la bibliothèque, présentées comme suit :

```

1 P_pcd.estimate_normals(search_param=o3d.geometry.KDTreeSearchParamHybrid
2 (radius=0.1, max_nn=30))
3 Q_pcd.estimate_normals(search_param=o3d.geometry.KDTreeSearchParamHybrid
4 (radius=0.1, max_nn=30))
5
6 P_pcd.estimate_covariances(search_param=o3d.geometry.KDTreeSearchParamHybrid
7 (radius=0.1, max_nn=30))
8 Q_pcd.estimate_covariances(search_param=o3d.geometry.KDTreeSearchParamHybrid
9 (radius=0.1, max_nn=30))

```

Code 4 – Normaux et covariances.

Ensuite, il faut changer la méthode de la fonction *registration_icp* en *estimation_method=o3d.pipelines.registration.TransformationEstimationForGeneralizedICP()*.

En utilisant la méthode généralisée, les résultats suivants ont été obtenus.

```

Generalized ICP Results:
Transformation Matrix:
[[ 0.9788418 -0.0489831 -0.19866905 0.49999991]
 [-0.01089022 0.95707691 -0.2896294 -0.29999983]
 [ 0.20432851 0.28566491 0.93629345 0.19999999]
 [ 0. 0. 0. 1. ]]

Inlier RMSE: 5.688812559467573e-07
Fitness: 1.0
GICP completed in 0.04 seconds

```

FIGURE 16 – Résultats de l'algorithme ICP généralisé *Open3D*.

Et en comparant les résultats avec MatrixComparer entre l'ICP point-à-point et l'ICP généralisé, il a été constaté que :

- Pour la matrice de rotation, la différence terme à terme était relativement faible, soulignant que dans les deux méthodes, les signes restent identiques à l'exception d'un terme, et les valeurs dans chaque matrice ne varient pas beaucoup entre elles.

Element	ICP Point-Point	Generalized ICP (GICP)	Difference Abs
R[0,0]	9.98e-01	9.79e-01	1.94e-02
R[0,1]	2.68e-02	-4.9e-02	7.58e-02
R[0,2]	-5.24e-02	-1.99e-01	1.46e-01
R[1,0]	-2.79e-02	-1.09e-02	1.7e-02
R[1,1]	9.99e-01	9.57e-01	4.23e-02
R[1,2]	-1.99e-02	-2.9e-01	2.7e-01
R[2,0]	5.18e-02	2.04e-01	1.53e-01
R[2,1]	2.14e-02	2.86e-01	2.64e-01
R[2,2]	9.98e-01	9.36e-01	6.21e-02

FIGURE 17 – Comparaison de la matrice de rotation entre l'ICP point-à-point et ICP généralisé.

- Pour le vecteur de translation, en général, les différences sont faibles, avec le même phénomène d'une plus grande uniformité dans les deux vecteurs. Cependant, l'uniformité est beaucoup plus grande dans la méthode généralisée, et il y a un changement de signe dans la transformation en Z.

Component	ICP Point-Point	Generalized ICP (GICP)	Difference Abs
Tx	2.24e-02	5.e-01	4.78e-01
Ty	-2.30e-02	-3.e-01	2.77e-01
Tz	-6.66e-04	2.e-01	2.01e-01

FIGURE 18 – Comparaison du vecteur de translation entre l'ICP point-à-point et ICP généralisé.

- Le temps est inférieur dans la méthode généralisée, ce qui montre que l'utilisation simultanée du point-à-point et du point-à-plan améliore la vitesse de convergence de l'algorithme, avec une différence de 0,0318 secondes, importante compte tenu de l'échelle de temps des résultats d'exécution des deux méthodes.

Metric	ICP Point-Point	Generalized ICP (GICP)	Difference Abs (s)
Time	7.00e-02	3.82e-02	3.18e-02

FIGURE 19 – Comparaison du temps d'exécution entre l'ICP point-à-point et ICP généralisé.

- L'erreur est considérablement plus faible dans la méthode généralisée, ce qui indique une réduction beaucoup plus importante de l'erreur par itération, bien qu'ayant le même seuil.

Metric	ICP Point-Point	Generalized ICP (GICP)	Difference Abs
Error	3.29e-02	5.69e-07	3.29e-02

FIGURE 20 – Comparaison de l'erreur entre l'ICP point-à-point et ICP généralisé.

En général, la méthode généralisée est considérablement meilleure que la méthode point-à-point, et bien supérieure à l'algorithme manuel, grâce à l'utilisation d'améliorations qui la rendent beaucoup plus précise et rapide. L'optimisation de la méthode généralisée est notable dans les résultats du recalage.

6 Conclusions

Le projet a permis d'explorer l'algorithme Iterative Closest Point (ICP) ainsi que ses variantes, avec une mise en œuvre combinant une implémentation manuelle et l'utilisation de la bibliothèque Open3D. Ces approches ont permis d'évaluer les performances et les limites de cet algorithme de recalage sur des nuages de points 3D.

La version manuelle de l'ICP, bien qu'efficace en termes de précision, montre des limites importantes en termes de vitesse et de robustesse face au bruit, comparée aux versions optimisées de Open3D.

L'ajout de bruit gaussien a permis de tester la stabilité de l'algorithme. Les résultats montrent que, bien que la précision globale soit préservée, le temps de recalage augmente et la transformation obtenue est légèrement moins homogène.

La version généralisée de l'ICP a surpassé la méthode classique point-à-point en termes de rapidité et de précision. Elle a permis une meilleure convergence et une réduction plus importante de l'erreur tout en maintenant une transformation rigide cohérente.

En conclusion, ce projet a démontré la puissance des versions modernes et optimisées de l'ICP, tout en mettant en évidence les avantages d'une bonne configuration des paramètres pour un recalage efficace et précis.

Bibliographie

- [1] ABABSA, F. Recalage de données 3d algorithmme icp : Iterative closest point. *TP ICP* (2024).