

# TER

## Recalage de nuages de points multi-vues issus d'un scanner à main



**MARSEILLE Luminy**  
**2017/2018**



# Introduction

Notre travail a consisté en une étude visant à aider à réaliser la thèse soutenue par la doctorante Manon Jubert. Thèse dont l'objectif est de créer le premier système d'usinage intelligent et qui est appuyée par IMC, une équipe de recherche travaillant au CEA (centre de Cadarache).

Par usinage intelligent, on entend une machine capable d'inspecter des pièces en temps réel au cours de leur usinage, à l'aide d'un bras robot articulé muni d'un scanner, qui peut alors enregistrer différentes vues de l'objet. On peut ensuite reconstituer informatiquement l'objet complet et ainsi surveiller l'état courant de la pièce pendant sa fabrication. Ce procédé devrait permettre de corriger les erreurs qui pourraient survenir entre deux scans ou de mieux comprendre l'apparition de ces erreurs.

Les vues de l'objet sont représentées par des nuages de points dans un espace de dimension 3. Pour des raisons techniques le scanners ne peut pas prendre une vue complète de l'objet en une fois, il est donc nécessaire de traiter les données obtenus pour reconstituer un scan complet. Cette méthode s'appelle le recalage de nuages de points. C'est dans cette partie du projet que nous sommes intervenus.

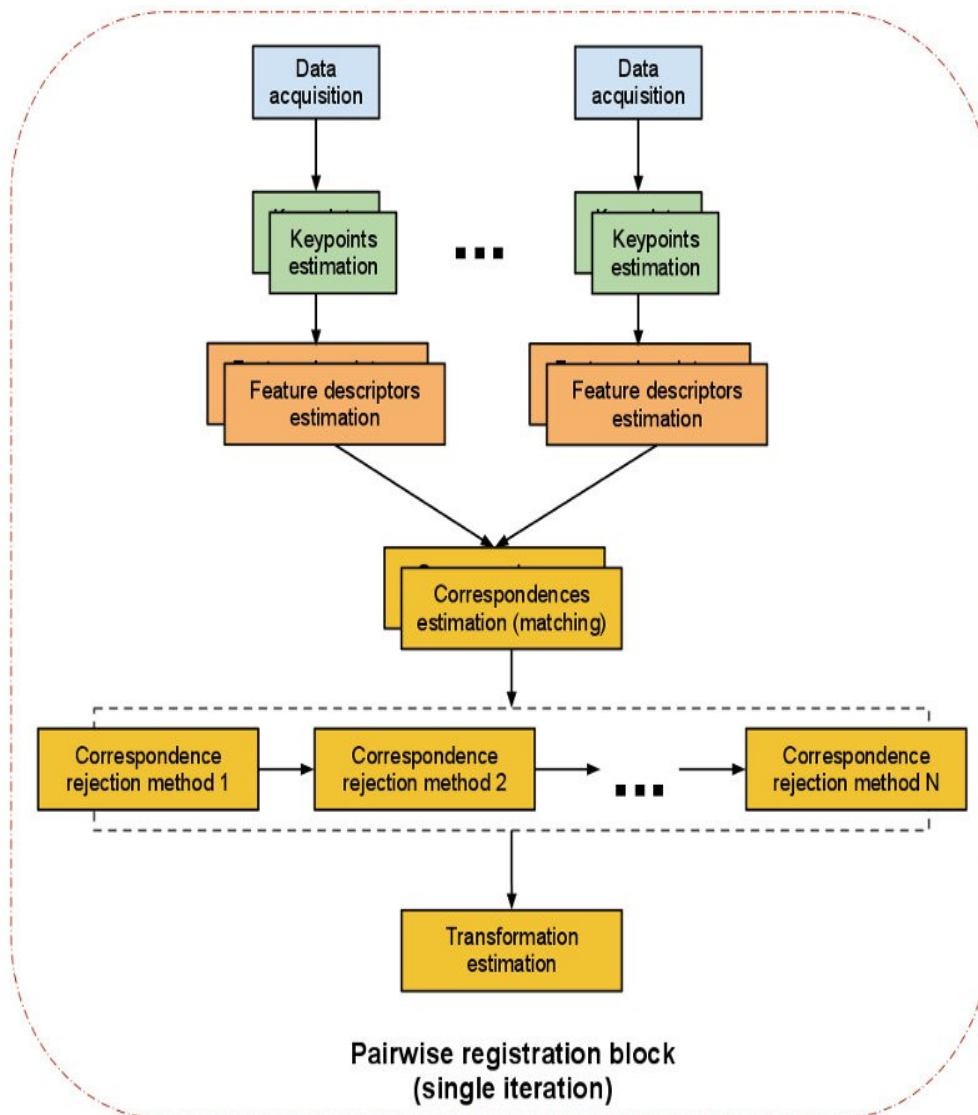
Il nous a été demandé de trouver la meilleure façon possible pour effectuer le recalage des données fournis par le scanners, en utilisant des méthodes de la librairie Point Cloud Library (PCL), qui est une bibliothèque open source pour le traitement d'image 2D / 3D et de nuages de points.

Dans un premier temps, nous expliciterons donc les tenants et aboutissements de ce projet, pour ensuite détailler le travail que nous avons fournis.

# 1) Recalage de nuages de points

Le recalage de nuages de points est un problème complexe en informatique. On parle de superposer des données qui n'ont en commun qu'une part de leur structure générale. De nombreuses méthodes de résolutions de cette question ont été proposées au fil du temps. Aujourd'hui on peut trouver un nombre important d'articles scientifiques consacrés au sujet.

Ici on procédera à l'alignement de nuages de points grâce à la librairie PCL, contenant des méthodes de recalage dont l'idée clé est d'identifier les points correspondants entre les nuages et de trouver une transformation qui minimise la distance (erreur d'alignement) entre les points correspondants (les zones d'intersection entre ces points doivent se chevaucher). Comme sur l'exemple de la feuille suivante tiré de la librairie avec une paire de jeux de données.



En effet on peut y voir les différentes étapes de recalage :

- A partir d'un ensemble de points, identifier les points-clés qui représentent au mieux les deux nuages.
- A chaque point clé, calculer un descripteur d'entité (feature descriptor).
- A partir de l'ensemble des descripteurs d'entités et de leurs positions XYZ dans les deux ensembles de données, estimer un ensemble de correspondances, basé sur les similarités entre les caractéristiques et les positions.
- Comme les données sont supposées être bruitées, toutes les correspondances ne sont pas valides, il faut donc rejeter ces mauvaises correspondances qui ont une mauvaise incidence sur le recalage.
- Avec l'ensemble restant des bonnes correspondances, estimer une transformation de mouvement.

Ce processus est répété, car la recherche de correspondance est affectée par la position et l'orientation relatives des nuages. Une fois que les erreurs d'alignement sont inférieures à un seuil donné, l'enregistrement est dit complet.

## 2) Le travail réalisé

En vu d'accomplir le travail demandé, nous avons été supervisé par M Arnaud Polette, mais nous avons également eu par moments l'occasion de consulter Mme Manon Jubert. De ce fait nous avons travaillé de façon à nous donner des objectifs à accomplir et à faire un compte-rendu à M Polette environ une fois par semaine. Nous avons eu quelques difficultés à tenir les délais les premières semaines, mais avons pu rattraper notre retard par la suite, et avons alors tout de même pu travailler de façon efficace.

### 2.1) Recherche

Notre travail consistait à trouver dans PCL, des méthodes ou algorithmes susceptibles d'opérer du recalage de nuages de points de façon optimale, par rapport aux vues des pièces définies dans le problème.

Nous nous sommes donc réparti la recherche entre nous pour cibler les différentes méthodes pouvant réunir les bons critères. Au total, nous en avons trouvé douze.

Pour garder la trace de cette recherche, nous avons ajouté des descriptions de ces méthodes dans un tableau, contenant une brève description de la méthode, le lien de la librairie, les avantages et/ou paramètres, ainsi que des articles universitaires correspondants.

Un des problèmes qui s'est posé assez rapidement dans l'avancement du projet était celui de comparer différents alignements, mais il fallait pour cela savoir quelles méthodes choisir.

Pour classer les alignements proposés par les différentes méthodes, nous avons besoin d'un système de notation. Nous avons pu utiliser une méthode de pcl retournant un score en fonction de l'alignement (du recalage) effectué. Nous avons également utilisé la distance de Hausdorff qui permet de calculer la distance entre 2 nuages de points afin de comparer ces deux méthodes de notation.

### 2.3) Tests

Pour déterminer quelles méthodes étaient les meilleures, nous avons du réaliser des tests. Nous avons commencé par suivre des tutoriels disponibles sur le site de PCL pour prendre en mains la librairie. Dans un second temps nous avons implémenté des programmes permettant de faire du recalage sur les données qui nous ont été fournies. Ainsi nous avons les outils pour tester chaque méthodes.

Nous nous sommes donc à nouveau réparti le travail pour être plus efficace, et une fois les tests réalisés, nous avons pu les comparer et les tester sur une seule machine pour bien accorder nos résultats.

Ces derniers ont été rangés dans un tableau pour garder une trace et bien identifier quelles seraient les meilleures méthodes.

Il convient par ailleurs de préciser que la plupart de ces méthodes prennent deux nuages de point en argument quand certaines peuvent en prendre plusieurs et ceci peut, selon les résultats, être un critère de sélection en plus.

Le temps d'exécution est aussi un facteur important car on doit avoir des programmes qui s'exécutent en seulement quelques secondes.

# 3) Résultats

## 3.1) Méthodes étudiées

Voici les résultats de notre première phase de travail, la recherche des méthodes de recalage dans PCL.

### Méthodes de recalage de 2 nuages

#### **ICP**

L'algorithme « Iterative Closest Point (ICP) permet de de recaler deux vues partielles d'un même objet en minimisant itérativement la distance entre leurs deux nuages de points correspondants. Étapes:

- Associer les points par critères du plus proche voisin
- Estimation paramètres transformation avec fonction de coût quadratique moyenne
- Transformation des points
- Itération (ré-association des points)

On peut utiliser cette méthode en faisant varier des paramètres, tels que le seuil de correspondances maximales entre 2 nuages, le nombre maximum d'itérations successives ou redéfinir un seuil pour un certain nombre d'erreurs quadratiques.

#### **Paramètres important :**

- Le *seuil de correspondances des distances maximales* modifiera beaucoup les résultats car c'est ce paramètre qui permet de sélectionner un échantillon de points. Si le nuage est trop grand ou trop étiré et que le seuil est trop bas, les résultats seront très médiocres.
- Le paramètre *d'itérations maximales* est important car il nous permet de rechercher en combien d'itérations le recalage est achevé.

#### **GeneralizedICP**

L'algorithme «Generalized Iterative Closest Point (GeneralizedICP) est une méthode combinant l'ICP et le point-to-plane ICP en utilisant des matrices de covariances. Cette méthode permet de recaler deux nuages de points en un seul. Contrairement à l'ICP les deux nuages de points sont utilisés pour construire la matrice de transformation qui permet ensuite le recalage. Ainsi les deux nuages peuvent être considérés comme source du modèle. Plus sélectif dans le recalage (peut ne pas accepter des alignement que ICP aurait accepté).

Le seuil de correspondance maximal propre aux techniques ICP est plus robuste avec cette méthode (Il devient plus facile de trouver une bonne valeur pour ce paramètre).

#### **Paramètres important :** (cf ICP)

## ICPWithNormals

L'algorithme «Iterative Closest Point With Normals » (ICPWithNormals) est un cas spécial de ICP qui utilise une transformation par défaut basé sur la méthode Point-to-plane. La méthode a un paramètre supplémentaire qui est la profondeur de recherche dans un arbre de type KdTree (c.f. Librairie) pour estimer une normale sur le nuage cible.

Paramètres important :

- la *profondeur* de l'arbre est très importante, au plus elle est grande, et au plus les résultats seront précis, mais au plus l'exécution sera lente.

## ICPNonLinear

Une méthode dérivée de ICP utilisant l'algorithme de Levenberg-Marquardt pour optimiser le choix des paramètres qui sont réglés automatiquement mais cela ne marche pas pour toutes les données.

## FPCS

Basé sur la méthode de RANSAC qui sélectionne 4 points clefs dans chaque nuage et cherche à les aligner via la méthode 4PCS pour calculer le score de l'alignement. Cette opération est répétée sur plusieurs sélections de 4 points pour retenir le meilleur recalage. Une part d'aléatoire intervient dans cette méthode.

FPCS est présenté comme une méthode d'initialisation pour d'autres méthodes de recalage.

**Paramètres important :**

- *ApproxOverlap*: Estimation du chevauchement entre la source et la cible, ce paramètre est déterminant pour obtenir un résultats précis.
- *Delta* : ajuste la précision du calcul des points importants, plus ce paramètre est élevé plus le résultat est précis au détriment du temps de calcul.
- *NumberOfSamples*: nombre d'échantillons extraient du nuage complet (de deux au nombre de points du nuage complet).

## KFPCS

Basé sur la méthode FPCS, elle présente des caractéristiques et un fonctionnement similaire .

KFPCS sélectionne un sous ensemble de points importants en utilisant la méthodes DoG puis les aligne en utilisant la méthode 4PCS et applique la transformation obtenue sur les nuages d'origines. Un part d'aléatoire intervient dans cette méthode.

KFPCS est présenté comme une méthode d'initialisation pour d'autres méthodes de recalage.

**Paramètres important :**

- *ApproxOverlap*: Estimation du chevauchement entre la source et la cible, ce paramètre est déterminant pour obtenir un résultats précis.
- *Delta* : ajuste la précision du calcul des points importants, plus ce paramètre est élevé plus le résultat est précis au détriment du temps de calcul.
- *NumberOfSamples*: nombre d'échantillons extraient du nuage complet (de deux au nombre de points du nuage complet).



## **SAMPLE CONSENSUS**

La méthode est économe, elle ne teste pas toutes les combinaisons mais seulement les plus probables, et les classe ensuite par ordre décroissant de marge d'erreur. Elle peut aussi traiter plus de deux nuages à la fois.

### **Paramètres important :**

- *MinSampleDistance* : La distance minimale entre deux échantillons.
- *MaxCorrespondenceDistance* : Distance maximale entre un point source un point cible correspondant.
- *MaximumIterations* : Nombre maximum d'itérations de l'algorithme.

## **LUM**

Basé sur la méthode GraphSLAM, gérant les données de recalage dans un graphe. Les sommets représentant les contraintes et les données de correspondances entre les points et les arrêtes représentant les transformations et les données du nuage.

### **Paramètres important :**

- *MaxIterations* : Le nombre maximal d'itérations de la méthode compute() après quoi elle devra s'arrêter (elle pourra aussi s'arrêter si les conditions de convergence sont remplies)
- *SetConvergenceThreshold* : Distance moyenne maximale pour la convergence.

## **ELCH**

Basé sur la méthode LUM, elle est censé améliorer grandement sa vitesse en utilisant une autre méthode. Le résultat ne sera toutefois pas toujours parfaitement précis, notamment dans de très gros nuages de points.

### **Paramètres important :**

## **Méthodes de recalage de plus de 2 nuages**

## **INCREMENTAL REGISTRATION**

Cet outil permet d'utiliser une méthode de type ICP sur un flux de nuage de point.

### **Paramètres important :**

- *Registration* : Il est important de spécifier quelles méthodes doivent être utilisées pour recalcr le flux de nuages. Cette méthodes doivent être de types ICP.

## **META REGISTRATION**

Cet outil permet d'utiliser la méthode de son choix pour recalcr un flux de nuage où chaque nuage est aligné sur l'ensemble des précédents.

Attention, cette méthode accumule tous les points enregistrés et devient de plus en plus coûteuse avec chaque nuage de points enregistrés.

### **Paramètres important :**

- *Registration* : Il est important de spécifier quelles méthodes doivent être utilisées pour recalcr le flux de nuages.

## **JointICP**

L'algorithme « Joint Iterative Closest Point (JointICP) est une extension de ICP qui permet de tester plusieurs nuages à la fois. Le comportement reste le même que ICP si l'on ne lui donne que 2 nuages.

### **Paramètres important :**

# Conclusion

Fait des recalages de 2 nuages de points entre eux.

On parle de la meilleure méthode qu'on a trouvé.

Tim : C'est moi le meilleur. (et le plu bo)

# **Annexes**