

# TER

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



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



# Introduction

Dans l'optique de notre TER, il nous a été confié la tâche d'opérer des recalages de nuages de points issus d'un scanner à main. Ce projet prend part à la thèse soutenu par la doctorante Manon Jubert dans le cadre d'un plus grand projet produit par IMC et consistant à traiter des pièces d'usinage de façon intelligente.

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

# **1) Scanner, nuages de points, et recalage**

## **1.1) Histoire du projet**

Notre travail consiste à faire une étude visant à aider à réaliser la thèse soutenue par la doctorante Manon Jubert.

Thèse dont l'objectif est de produire du travail d'usinage intelligent et qui est appuyée par IMC, une équipe de recherche travaillant au CEA (centre de Cadarache).

Par usinage intelligent on entend un programme capable d'inspecter des pièces dans des chaînes de fabrication en temps réel, à l'aide d'un bras robot articulé muni d'un scanner. Ce scanner enregistre différentes vues de l'objet observé et peut alors surveiller directement son état.

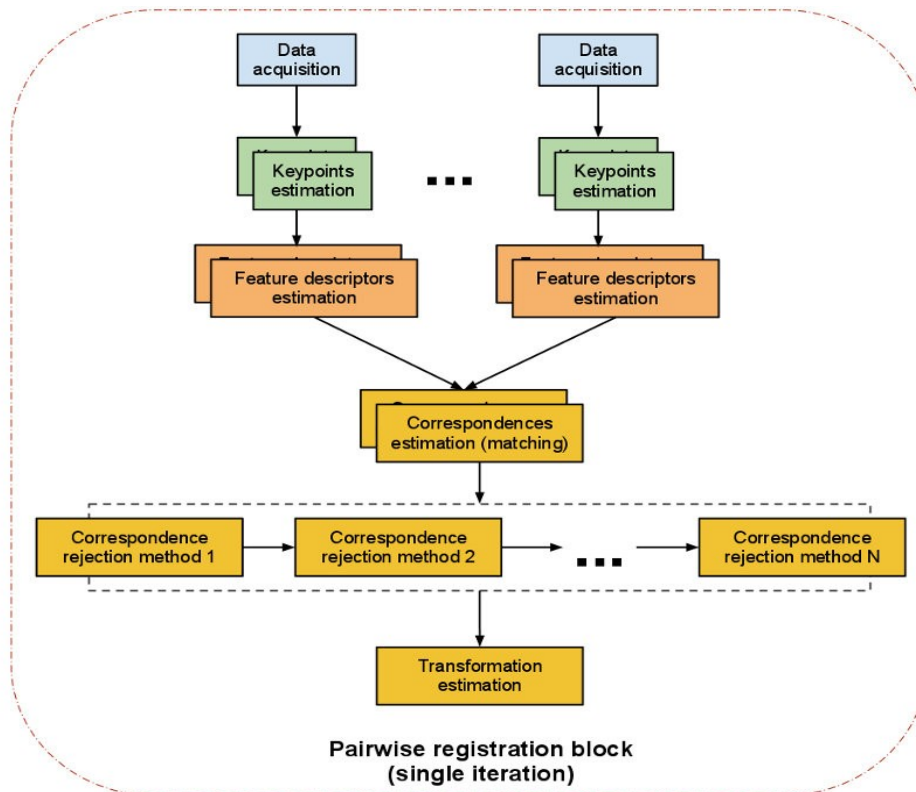
Les vues de l'objet sont représentées par des nuages de points et sont alors comparées, de sorte à pouvoir détecter des erreurs de fabrication et même à pouvoir les corriger directement ou mieux prévoir de prochaines erreurs. C'est ce que l'on appelle du recalage de points.

Notre participation réside donc à trouver la meilleure façon possible pour effectuer le recalage, 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.

## 1.2) Recalage de nuages de points

Procéder à l'alignement de nuages de points peut être fait grâce à 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 ci-dessous tiré de la librairie PCL 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 bruyantes, 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 de méthodes**

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.

## 2.2) Tests

Pour savoir quelles méthodes étaient les meilleures, nous avons dû réaliser des tests. Nous avons commencé par suivre des tutoriels disponibles sur le site de PCL, pour ensuite commencer à implémenter des petits programmes pouvant faire du recalage sur les pièces données pour le projet.

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

Pour les critères de sélection, 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 d'ajouter un peu plus de précision aux résultats. Le temps d'exécution est aussi un facteur important car on doit avoir des programmes qui s'exécutent en seulement quelques secondes.

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.

# **Conclusion**

Fait des recalages de 2 nuages de points entre eux.

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



# **Annexes**