

# Recalage et fusion de modèles numérisés tridimensionnels de grande taille

MEMO-F-403 - Préparation au mémoire

Tim Lenertz

11 août 2014

# 1 Introduction

Pour des projets de documentation 3D, des objets ou environnements sont souvent numérisés sous forme de *nuages de points*. Un nuage de points consiste en un ensemble de points situés sur les surfaces des objets, définis par des coordonnées dans un repère orthonormal donné, et attribués par des valeurs scalaires ou vectorielles comme par exemple une couleur RGB.

Ces jeux de données sont usuellement capturés par des scanners 3D à télémètre laser, ou par photogrammétrie. Ces données brutes ne représentent qu'une partie de l'objet, et sont limitées par le champ de vision de la caméra, les surfaces cachées (sur le côté opposé au scanner de l'objet), les limites de résolution pour des parties éloignées ou qui ont une texture complexe. Afin d'en synthétiser un nuage de points représentant le modèle complet, plusieurs traitements doivent être effectués, notamment le recalage : Pour toutes les nuages de points partielles, une transformation affine est appliquée aux points afin de les mettre dans un repère commun.

Plusieurs techniques (semi)-automatisées ont été développées qui permettent de trouver une approximation des positions et orientations relatives des scans. Ils peuvent faire intervenir des photos, ou des marqueurs visuels ajoutés sur la scène. La matrice de transformation précise est ensuite déterminée algorithmiquement, afin de bien aligner les surfaces dans les différents ensembles de points. Typiquement, une forme de l'algorithme ICP<sup>1</sup> est utilisé, un algorithme qui ajuste la transformation en itérativement minimisant la distance entre les deux ensembles de points.

Ce mémoire se concentre sur le recalage de numérisations à distance d'un objet à grande dimensions, avec des numérisations à courtes distances de détails de l'objet. Donc les ensembles de points à recaler pourront avoir des densités de points très différentes, et auront un nombre de points très élevé. Le but est d'établir un workflow et de développer des algorithmes qui permettent d'effectuer ce type de recalage. Le travail est basé sur un projet de documentation 3D actuel du LISA.

## 1.1 Terminologie

## 1.2 Nuage de points

---

1. Iterative Closest Point

## 2 Etat de l'art

### 2.1 Documentation 3D

Des nuages de points créés via des processus de numérisation 3D sont utilisés dans plusieurs domaines pour représenter des objets réels. Les modèles 3D peuvent passer de représentations détaillées d'objets petits (p.ex. en dent), à des bâtiments ou même des sites entiers.

Pour collectionner les données brutes de l'objet pour lequel on veut générer un modèle 3D, deux techniques sont généralement utilisées : La *photogrammétrie* consiste à prendre des photos (possiblement stéréoscopiques) de l'objet, à partir desquelles on peut extraire algorithmiquement de l'information sur la profondeur des pixels. D'autre part, on utilise des *scanners tridimensionnels*, des appareils qui effectuent un balayage par laser de l'objet et produisent directement un nuage de points. Souvent les deux techniques sont combinées. Cela permet par exemple de compléter un scan laser très détaillé avec de l'information sur les couleurs des points.

Pour passer des données brutes à un modèle entier, toujours sous forme de nuage de points, un nombre important de post-traitements sont requis.

#### 2.1.1 Scanner tridimensionnels

Plusieurs types de scanners tridimensionnels existent.

### 2.2 Recalage

Afin de créer un modèle complet, on doit fusionner plusieurs scans qui sont prises de différents points de vues et qui sont tous limités par le champ de vision du scanner. Ces différents nuages de points ont tous des repères différents, relatifs à la position et à l'orientation du scanner, et possiblement ayant des échelles différents.

Puisqu'il s'agit toujours de repères cartésiens orthonormaux, on peut mettre un nuage de points dans un autre repère en appliquant une matrice de transformation rigide à tous les points du nuage. Une telle transformation consiste en une translation, rotation, et dans certains cas un facteur de redimensionnement. Dès que les nuages de points sont tous dans le même repère, on peut les fusionner par simple concaténation des points. En pratique, des post- et pré-traitements devront être effectués pour ajuster les densités et la distribution des points, corriger des erreurs dans les données brutes comme des points *outliers*, et autres.

Par recalage, on entend le procédé de trouver ces transformations rigides, de manière automatisé ou semi-automatisé. D'une part on a les algorithmes de recalage approximatif, donc le but est d'analyser le contenu des nuages de points et d'en identifier des régions qui représentent la même partie du modèle afin de trouver un recalage approximatif. Ensuite, ce recalage initial peut être raffiné par un algorithme de recalage précis pour bien aligner les surfaces des nuages de points. Généralement ICP est utilisé pour cela. Sachant que les nuages de points sont déjà alignés de manière à ce que les points qui représentent les mêmes parties du modèle sont proches l'un de l'autre, ICP procède en minimisant les distances entre ces points.

On peut distinguer entre les algorithmes locaux (séquentiels) qui opèrent sur deux nuages de points à la fois, et les algorithmes globaux (simultanés) qui effectuent un recalage de plusieurs nuages de points. Cette dernière méthode est plus complexe, mais a l'avantage qu'il n'y a pas d'erreur de recalage qui s'accumule lors des recalages deux-à-deux. Ces algorithmes tentent au lieu de distribuer l'erreur uniformément parmi les nuages.

La méthode préférée pour déterminer un recalage approximatif dépend surtout du type et de la forme de l'objet numérisé, et des données brutes dont on dispose. Certains algorithmes procèdent en identifiant

des droites ou des plans dans le modèle, et sont donc plus appropriés pour par exemple des façades de bâtiments. Si les scans ont des zones de chevauchement assez grandes, il peut être possible d'en déduire le recalage de façon automatisée. D'autres méthodes peuvent être utilisées avec des données collectionnées additionnelles, comme des photos prises par le même point de vue que le scanner ou des markers placés sur l'objet avant la numérisation. Par des algorithmes de détection de zones d'intérêt sur des images ou directement sur les nuages de points, on peut alors identifier des points communs dans les différents scans. Aussi les informations attribuées aux points, comme une couleur, une mesure de température, ou autres, peuvent servir en tant que dimension en plus des coordonnées 3D pour déterminer leur distance. Une première estimation de la pose des scans peut être issue de sensors odométriques ou GPS du scanner.

Certains algorithmes seront décrits en détail dans les pages suivantes, à savoir ICP [3] et ses variantes [2], ...

### 2.2.1 Iterative Closest Point

L'algorithme ICP ("itératif point le plus proche") est considéré comme la méthode standard pour l'alignement précis de deux nuages de points, après qu'un recalage approximatif initial a déjà été fait. Parmi les deux nuages de points pris comme entrée, l'un (*référence*) reste fixé, tandis que l'autre (*source*) est transformé par translation et rotation. L'algorithme procède en itérativement raffinant la transformation à chaque étape, de manière à ce que la distance entre les deux nuages de points soit minimisée.

Plus précisément, pour tout point  $s_i$  de l'ensemble *source*, on choisit un point  $c_i$  de l'ensemble *cible* pour lequel on estime qu'il représente plus au moins la même position dans le nuage de points. Comme les deux nuages de points sont déjà recalés approximativement, une méthode courante est de choisir le point le plus proche, avec la transformation actuelle. L'algorithme nécessite donc que les densités de points des deux ensembles soient similaires. Ensuite une transformation est appliquée aux points  $s_i$  qui minimise une métrique donnée de l'erreur de recalage.

La *cible* ne doit pas forcément être donnée sous forme de nuage de points, mais peut par exemple être une surface mathématique continue donnée sous forme paramétrique  $(x(\vec{t}), y(\vec{t}), z(\vec{t}))$  ou implicite  $g(x, y, z) = 0$ . On peut alors calculer numériquement les coordonnées d'un point  $c_i$  correspondant à chaque point  $s_i$ . [3]

Plusieurs variantes de l'algorithme ont été développées qui se distinguent par le choix des couples de points, des poids associés aux couples, le rejet de certains couples, et la façon comment la métrique d'erreur est définie et minimisée. [2]

Pour la minimisation, on doit calculer une transformation rigide (généralement translation et rotation, donc 6 degrés de liberté) transforme les points  $s_i$ . Il existe plusieurs méthodes déterministes qui nécessitent pas de prendre plusieurs échantillons dans l'espace des transformations rigides. Pour le calcul de la rotation on peut par exemple utiliser la décomposition en valeurs singulières, par une représentation sous forme de matrice orthonormale ou sous forme de quaternion [1].

En général, un algorithme ICP procède selon les étapes suivantes :

1. Sélectionner des sous-ensembles de points  $s_i$  et  $c_i$  à utiliser (possiblement tous les points).
2. Former les couples  $(s_i, c_i)$ .
3. Associer des poids aux couples. (optionnel)
4. Rejeter certains couples. (optionnel)
5. Calculer une métrique d'erreur à partir des couples. Par exemple  $\sum_i \|s_i - c_i\|^2$ .
6. Appliquer une transformation rigide aux points  $s_i$  qui minimise l'erreur.
7. Si l'erreur est inférieure à une valeur de seuil donnée, arrêter. Sinon, continuer avec 1 ou 2.

### Calcul de la transformation par quaternions

Une manière de calculer la translation et rotation qui est appliquée à chaque itération à l'ensemble de points *source* est décrite dans [3] et [1]. La procédure prend uniquement les  $n$  couples de points  $(\vec{s}_i, \vec{c}_i)$  comme entrée, où  $\vec{c}_i$  est un point de l'ensemble *cible*  $C$ , et  $\vec{s}_i$  un point de l'ensemble *source*  $S$ . On suppose que les transformations données par les itérations précédentes ont déjà été appliquées aux points  $s_i$ . La méthode reste valable pour tout choix de points, et  $C$  ne doit pas forcément être donné sous forme

d'ensemble de points, mais ses points  $\vec{c}_i$  peuvent aussi par exemple être calculés à partir d'une surface paramétrique.

Le but de trouver  $\vec{q}_T$  et  $\dot{q}_R$  qui minimisent le carré des distances des couples  $f(\vec{q})$ .

$$f(\vec{q}) = \frac{1}{n} \sum_{i=1}^n \|\vec{c}_i - \mathbf{R}\vec{s}_i - \vec{q}_T\|^2 \quad (2.1)$$

$\vec{q}_T = (q_4 q_5 q_6)^T$  donne la translation, et  $\mathbf{R}$  la matrice de rotation 3x3.

On cherche une transformation de la forme  $\vec{t}' = \mathbf{R}(\vec{s}) + \vec{q}_T$ , donc une rotation, suivie par une translation à appliquer aux points  $s_i$ , qui minimise les distances des points transformés  $t_i$  aux points  $c_i$  correspondants dans le nuage de points *cible*. Car on ne peut pas assumer que les deux nuages  $S$  et  $C$  auront exactement la même distribution de points, les erreurs  $e_i = \vec{c}_i - \vec{t}_i$  ne peuvent en général pas devenir 0. On minimise la somme des carrés des erreurs  $\sum \|e_i\|^2$ . Le développement décrit en [1] inclut aussi un facteur de redimensionnement, mais pour ICP on suppose généralement que les deux nuages de points ont déjà la même échelle. Un redimensionnement durant les itérations introduirait des problèmes avec le choix des couples de points.

Pour le développement suivant, on prend les coordonnées des points  $\vec{s}$  et  $\vec{c}$ , relatif au centres de masse des ensembles. On définit

$$\vec{\mu}_s = \frac{1}{n} \sum_i \vec{s}_i \quad \text{et} \quad \vec{\mu}_c = \frac{1}{n} \sum_i \vec{c}_i \quad (2.2)$$

Et pour tout  $i$ , on pose  $\vec{s}'_i = \vec{s}_i - \vec{\mu}_s$  et  $\vec{c}'_i = \vec{c}_i - \vec{\mu}_c$ . Donc  $\sum \vec{s}'_i = 0$  et  $\sum \vec{c}'_i = 0$ . L'erreur  $e_i$  peut être réécrit comme  $e_i = \vec{c}'_i - \vec{t}'_i = \vec{c}'_i - \mathbf{R}(\vec{s}'_i) - \vec{q}'_T$ . On peut montrer que la somme des carrés des erreur sera minimale quand  $\vec{q}'_T = 0$ , et donc la translation à appliquer correspondra à  $\vec{q}_T = \vec{q}_T - \mathbf{R}(\vec{\mu}_s)$  :

En fait, on a que  $\vec{q}'_T = \vec{q}_T - \vec{\mu}_c + \mathbf{R}(\vec{\mu}_s)$ . La somme des carrés peut être développée en

$$\sum_{i=1}^n \|\vec{c}'_i - \mathbf{R}(\vec{s}'_i)\|^2 - 2\vec{q}'_T \sum_{i=1}^n (\vec{c}'_i - \mathbf{R}(\vec{s}'_i)) + n\|\vec{q}'_T\|^2 \quad (2.3)$$

Le second terme sera toujours 0 puisque les coordonnées sont pris relatif aux centres de masse. Le premier ne dépend pas de  $\vec{q}'_T$ . Donc l'expression est minimisée par  $\vec{q}'_T = 0 \iff \vec{q}_T = \vec{\mu}_c - \mathbf{R}(\vec{\mu}_s)$ .

Pour trouver la rotation  $\mathbf{R}$ , on maximise

$$\sum_{i=1}^n \vec{c}'_i \cdot \mathbf{R}(\vec{s}'_i) \quad (2.4)$$

En effet, le produit scalaire de deux vecteurs est maximal quand ils ont la même direction. On cherche la rotation sous forme d'un quaternion  $\dot{q}_R$ .<sup>1</sup>

Une rotation peut être encodée dans un quaternion unitaire, c'est à dire pour lequel  $q_0 \geq 0$  et  $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$ .  $\dot{q}$  est purement imaginaire si  $q_0 = 0$ . (Les trois autres composants peuvent être considérés comme termes imaginaires comme chez les nombres complexes.) On représente un vecteur sous forme de quaternion purement imaginaire, en mettant ses coordonnées dans  $q_1, q_2, q_3$ . On peut montrer que si  $\dot{s}$  est purement imaginaire et  $\dot{q}_R$  est unitaire, alors  $\dot{q}_R \dot{s} \dot{q}_R^*$  reste purement imaginaire, et peut donc représenter une opération sur un vecteur.  $\dot{q}_R^*$  est le conjugué de  $\dot{q}_R$ .

L'article [1] montre qu'en utilisant les propriétés des quaternions, on peut former à partir des ensembles de points  $\vec{s}_i$  et  $\vec{c}_i$  une matrice symétrique 4x4  $\mathbf{N}$  telle que

$$\sum_{i=1}^n \vec{c}'_i \cdot \mathbf{R}(\vec{s}'_i) = \sum_{i=1}^n \dot{c}'_i \cdot (\dot{q}_R \dot{s}'_i \dot{q}_R^*) = \dot{q}_R \mathbf{N} \dot{q}_R^* \quad (2.6)$$

---

1. Une matrice de rotation  $\mathbf{R}$  peut en être déduit à partir d'un quaternion  $\dot{q} = (q_0 q_1 q_2 q_3)^T$  par la formule

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix} \quad (2.5)$$

Il est aussi montré que le quaternion unitaire  $\hat{q}_R$  qui maximise l'expression est le vecteur propre correspondant à la valeur propre la plus grande de  $\mathbf{N}$ . Le calcul des valeurs propres nécessite la résolution d'une équation polynomiale du quatrième degré. Il existe des solutions de forme fermée pour ce problème.

Donc, cette méthode permet de déterminer la rotation et translation à appliquer à  $\mathbf{M}$  par une série de calculs, sans passer par des méthodes approximatives.

### **Variantes de l'algorithme**

#### **Version simultanée**

Une approche qui permet de faire un recalage ICP *simultané* de plusieurs nuages de points est décrite dans [?].

#### **2.2.2 Extended Gaussian Images**

On peut générer à partir d'un nuage de point son EGI (Extended Gaussian Image), qui est une forme d'encoder la forme d'un objet convexe. Ceci permet d'estimer la rotation à appliquer, sans partir d'un recalage initial.

# Bibliographie

- [1] Berthold K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. In *Journal of the Optical Society of America A*, volume 4, page 629. Department of Electrical Engineering, University of Hawaii at Manoa, April 1987.
- [2] Szymon Rusinkiewicz; Marc Levoy. Efficient variants of the icp algorithm. volume 3-D Digital Imaging and Modeling, pages 145–152. Stanford University, 2001.
- [3] Paul J Besl; Neil D. McKay. A method for registration of 3-d shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 14, pages 239–256. IEEE, February 1992.