

Recalage de deux nuages de points 3D au rapport d'échelle non uniforme

Flora Quilichini, Thomas Fisichella, Frédéric Payan, Marc Antonini
Laboratoire I3S

Résumé : Ce travail rentre dans le contexte d'un projet de recherche en collaboration avec le CHU de Nice dont l'objectif est le recalage rigide de deux nuages de points 3D représentant des organes du corps humain pour une application en chirurgie augmentée. Plus précisément, l'objectif est d'estimer la transformée $T(s, \Omega, t)$ (i.e. échelle, rotation, translation) entre un objet complet (source) et un objet partiel (cible) dont l'échantillonnage et le facteur d'échelle diffèrent. La première étape du recalage consiste au calcul de descripteurs locaux sur chacun des deux objets qui seront nécessaires pour la mise en correspondance des points. Pour cela, nous avons fait le choix des *Growing Least Squares* [1] car ce sont des descripteurs robustes au facteur d'échelle. En deuxième étape, nous utilisons un algorithme d'optimisation sur les paires de correspondance afin d'estimer la transformée $T(s, \Omega, t)$. Notre contribution est une extension de l'algorithme *Fast Global Registration* [3] au recalage de deux objets présentant un rapport d'échelle non uniforme selon les axes X , Y et Z . Nous avons testé notre méthode sur plusieurs modèles synthétiques possédant les propriétés de surface des organes réels (caractère lisse, présence de symétries, et redondance de la géométrie) et nous obtenons des premiers résultats concluants.

Mots-clés : Recalage, nuage de points 3D, estimation d'échelle

1 Introduction

Le recalage de nuages de points 3D est un sujet largement étudié en robotique et en vision par ordinateur. Les nombreuses contraintes posées par les données (présence de bruit, occlusion, désordre, chevauchement partiel, modalités d'acquisition différentes, déformations) en font un problème vaste et difficile, notamment lorsqu'il s'agit de données médicales acquises à partir de modalités différentes. Les données réelles que l'on doit recalculer proviennent d'un scanner à rayons X et d'un système multi-vues de caméras RGBD et présentent les contraintes suivantes : chevauchement partiel, différence d'échantillonnage, facteur d'échelle différent et distorsion.

Les problèmes de recalage sont classiquement résolus en trois étapes. La première étape consiste en une segmentation où les objets d'intérêt sont extraits de la scène. La deuxième étape consiste à mettre en correspondance un sous-ensemble de points localisés sur les objets d'intérêts par appariement. Cet appariement se fait généralement par la recherche de similarité entre des descripteurs locaux décrivant la surface au voisinage des points d'intérêt. La troisième et dernière étape consiste à calculer, à l'aide d'un algorithme d'optimisation exécuté sur les paires de corres-

pondances, la matrice de transformation permettant de recalculer les objets d'intérêt. Nous avons limité notre étude au recalage de deux modèles 3D déjà pré-segmentés (source et cible) et nous nous sommes focalisés uniquement sur les deux dernières étapes. Nous avons repris l'algorithme d'optimisation *Fast Global Registration* (FGR) [3] car il présentait deux avantages majeurs :

- *Robustesse aux paires aberrantes.* Sa fonction objectif possède un terme de pénalisation des mauvaises paires inversement proportionnel au terme favorisant les bonnes paires.
- *Rapidité & Précision.* L'algorithme atteint une précision de recalage comparable à celle d'un algorithme d'ajustement local correctement initialisé, sans nécessiter une telle initialisation (très coûteuse en temps de calcul).

Nous utilisons comme descripteur les *Growing Least Squares* (GLS) [1] car ce dernier est robuste à la quasi-totalité des contraintes mentionnées plus haut. Les auteurs de [1] se servent également des GLS pour estimer le rapport d'échelle, dans leur cas identique selon les axes X , Y et Z , entre la source et la cible. Cela leur permet de recalculer des modèles acquis à des échelles différentes.

2 Travail proposé

Ici, nous allons plus loin en proposant une méthode qui recalcule deux nuages de points 3D dont le rapport d'échelle est différent suivant les axes. Pour cela, nous avons modifié FGR de manière à lui faire estimer la matrice échelle S conjointement à la traditionnelle matrice rotation-translation Ωt . Nous utilisons les GLS mais tout autre descripteur robuste à l'échantillonnage et aux variations d'échelle peut lui être substitué. Pour construire notre extension, nous avons repris la fonction objectif à minimiser définie dans FGR [3] et nous y avons intégré le terme d'échelle (en gras) :

$$E(\Omega t, \mathbf{S}, \mathbb{L}) = \sum_{(p,q) \in \text{pairs}} l_{p,q} * ||p - \mathbf{S} * \Omega t * q||^2 + \Psi(l_{p,q}) \quad (1)$$

Dans [3], l'objectif E est minimisé par méthode itérative alternée. Les mises à jour du processus de ligne \mathbb{L} sont obtenues en dérivant E par rapport à $l_{p,q}$ et celles de Ωt en utilisant un algorithme de type Gauss-Newton. Intégrer les paramètres d'échelle dans une transformée globale T (échelle * rotation + translation) est délicat.

En effet, la matrice d'échelle \mathbf{S} va introduire de la non-linéarité dans la méthode de Gauss-Newton (les coefficients de la matrice Jacobienne ne sont plus des constantes mais dépendent des paramètres d'échelle, de rotation et

de translation). A la place, nous avons décidé de scinder T en deux parties (Ωt et S). Ωt est mise à jour pareillement à [3], à S fixée. Puis nous appliquons la méthode de Gauss-Newton une deuxième fois pour mettre à jour S , à Ωt fixée. Le pseudo-code ci-dessous reprend l'algorithme initial présenté dans [3] avec notre contribution indiquée en bleu. Pour plus de détails concernant l'optimisation des *line process* ($l_{p,q}$) ou la méthode Gauss Newton, le lecteur est prié de se référer à l'article de référence [3]. Le descripteur utilisé dans [3] pour déterminer l'ensemble des points d'intérêt et leurs caractéristiques associées est le *Fast Point Feature Histograms* (FPFH) [2]; or, ce dernier n'étant pas robuste au changement d'échelle, nous lui avons substitué les *Growing Least Squares* [1]. Les paires de points correspondants sont ensuite obtenues en suivant une méthode basée sur la variation d'échelle [1].

Algorithm 1 Proposed FGR-based algorithm

```

input : A pair of surfaces ( $P, Q$ )
output : Transformation  $T$  that aligns  $Q$  to  $P$ 
Compute GLS features  $F(P)$  and  $F(Q)$ 
Build  $K$  by computing nearest neighbors between  $F(P)$ 
and  $F(Q)$  in log-scale space
 $T \leftarrow I, \mu \leftarrow D^2$ 
while  $iter < N_{max}$  or  $\mu > \delta^2$  do
   $\Omega t \leftarrow 0, \delta\Omega t \leftarrow 0, S \leftarrow 0, \delta S \leftarrow 0$ 
  Get which variable to update ( $\Omega t, S$  or both)
  ( $upd\Omega t, updS$ )  $\leftarrow$  updateVariable( $iter$ )
  for  $(p, q) \in K$  do
    Compute  $l_{p,q}$ 
    if is $\Omega t$  then
       $\delta\Omega t \leftarrow$  applyGaussNewton( $p - T * q * l_{p,q}$ )
       $\Omega t \leftarrow \Omega t + \delta\Omega t$ 
    end if
    if is $S$  then
       $\delta S \leftarrow$  applyGaussNewton( $p - T * q * l_{p,q}$ )
       $S \leftarrow S + \delta S$ 
    end if
  end for
  if is $\Omega t$  then
     $T \leftarrow$  updateT( $\Omega t, T$ )
  end if
  if is $S$  then
     $T \leftarrow$  updateT( $S, T$ )
  end if
  Every four iterations,  $\mu \leftarrow \mu/2$ 
   $iter \leftarrow iter + 1$ 
end while

```

3 Résultats

Les données cibles ont été générées à partir de données synthétiques sources auxquelles nous avons appliqué - dans l'ordre - les opérations suivantes :

- *Une transformation* Cette première étape nous permet de modéliser un changement de repère lors de l'acquisition de la donnée cible. Nous avons appliqué les mêmes transformations à nos données sources, à savoir : un changement d'échelle (uniforme ou non), suivi d'une rotation, puis d'une translation.

- *Un sous-échantillonnage par deux du nombre de points suivi d'un ré-échantillonnage des points restants.* Ici, l'idée est de modéliser deux acquisitions d'un même objet obtenues au moyen de deux dispositifs de résolutions différentes. Dans notre jeu de données, un maillage cible fixé contient deux fois moins de points que sa source associée, et ses points sont déplacés sur la surface de manière à ce que leurs positions ne coïncident pas avec celles des points de la source.
- *Une coupe (cropping) - facultatif* Cette dernière étape nous permet de modéliser une donnée cible "incomplète" par rapport à la source. Nous avons effectué des coupes de la source suivant différents axes. Afin de pouvoir mesurer la robustesse de notre algorithme à la surface de recouvrement (overlapping) entre la source et la cible, chaque donnée cible a été générée, avec une proportion différente du maillage source (fémur : 50%, hanche : 60% et veines : 40%)

Le tableau ci-dessous synthétise les résultats obtenus sur nos modèles. Nous obtenons des résultats de recalage très satisfaisants pour la plupart des tests. Seul le cas d'un fragment de veine - qui possède le pourcentage de recouvrement le plus faible, et une géométrie plutôt redondante dans la donnée complète - échoue. On remarque également que l'erreur de recalage diminue avec l'augmentation de la surface partagée entre la source et la cible.

4 Conclusion et perspectives

Nous avons développé une extension de FGR permettant de recalculer deux nuages de points dont le rapport d'échelle est différent selon les axes X, Y et Z. Nous obtenons de bons résultats sur des données synthétiques. Le principal facteur limitant de notre solution est le descripteur GLS, qui n'est théoriquement cohérent que lorsque la matrice de mise à l'échelle entre la source et la cible est uniforme. Afin de présenter une méthode complète, nous avons calculé les GLS en présence de disparités suffisamment faibles des coefficients d'échelle pour que le descripteur reste pertinent. Notons que notre algorithme parvient à estimer de plus grandes disparités, à condition qu'on lui donne un nombre suffisant de bonnes paires de correspondances. Le développement d'un descripteur robuste aux déformations d'échelle est la prochaine étape vers une solution plus générale au problème du recalage avec transformation non uniforme, voire déformations.

Références

- [1] Nicolas Mellado, Matteo Dellepiane, and Roberto Scopigno. Relative scale estimation and 3d registration of multi-modal geometry using growing least squares. *IEEE Transactions on Visualization and Computer Graphics*, 22 :1–1, 12 2015.
- [2] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.
- [3] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *ECCV*, 2016.

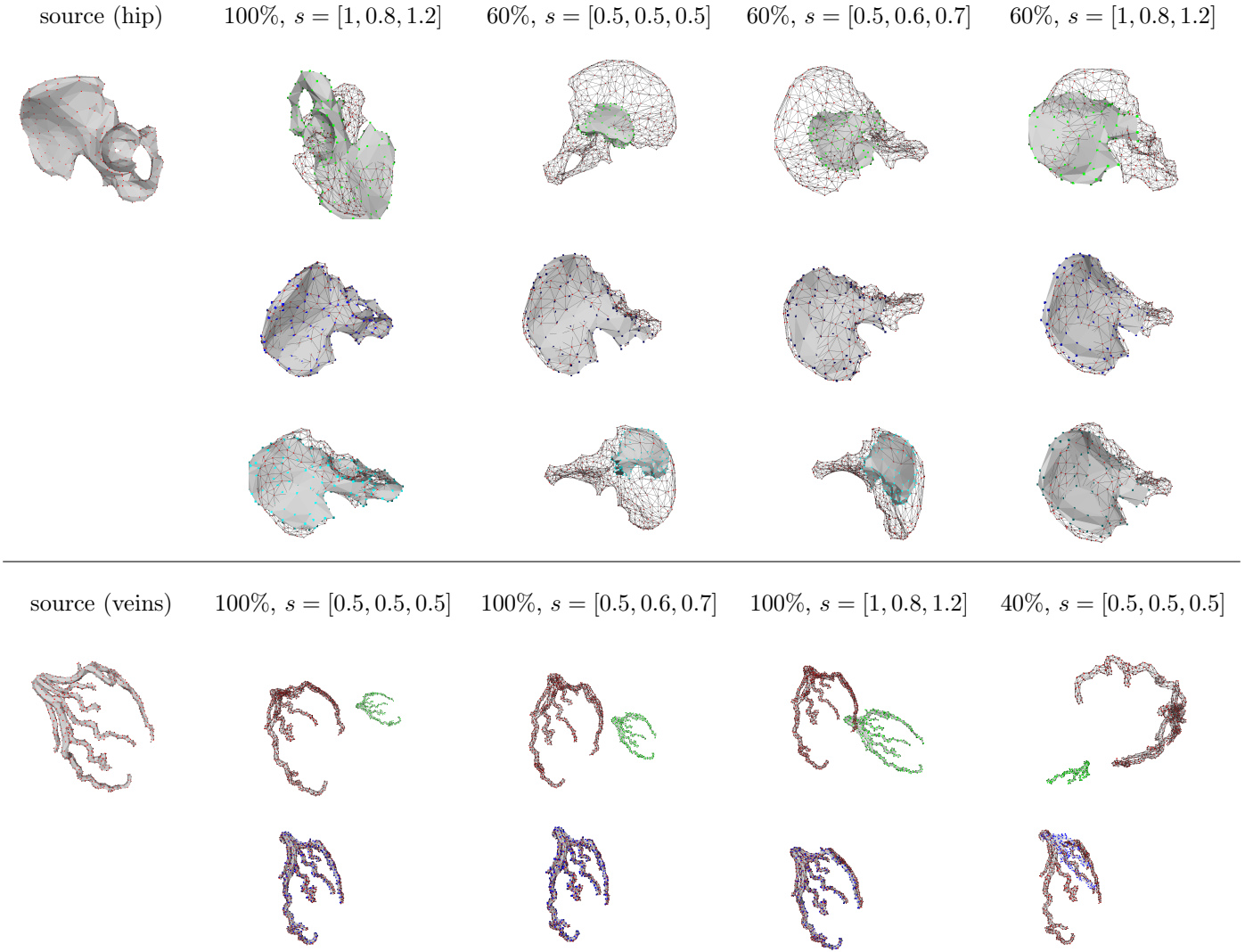


TABLE 1 – Nos données et nos résultats de recalage. Ligne 1 : donnée source (en rouge, à gauche) et données cibles (en vert) avec leur paramètres associés (portion de la surface source conservée et coefficients d'échelle). Ligne 2 : Résultats de recalage (en bleu) avec notre algorithme. Ligne 3 : Résultats de recalage (en cyan) avec FGR.

Source	Cible	MSE FGR+	MSE FGR	Subj eval FGR+	Subj eval FGR
FEMUR (500 pts)	full surface, scale = $[0.5 \ 0.5 \ 0.5]$	$5.35 * 10^{-4}$	$4.42 * 10^{-2}$	très bonne	mauvaise
	full surface, scale = $[0.5 \ 0.6 \ 0.7]$	$9.54 * 10^{-4}$	$2.84 * 10^{-2}$	bonne	mauvaise
	full surface, scale = $[1 \ 0.8 \ 1.2]$	$5.92 * 10^{-4}$	$5.92 * 10^{-4}$	bonne	bonne
	fragment (50%), scale = $[0.5 \ 0.5 \ 0.5]$	$1.31 * 10^{-3}$	$6.97 * 10^{-3}$	très bonne	moyenne
	fragment (50%), scale = $[0.5 \ 0.6 \ 0.7]$	$5.21 * 10^{-3}$	$1.84 * 10^{-3}$	bonne	moyenne
	fragment (50%), scale = $[1 \ 0.8 \ 1.2]$	$3.79 * 10^{-3}$	$4.12 * 10^{-3}$	bonne	bonne
HIP (500 pts)	full surface, scale = $[0.5 \ 0.5 \ 0.5]$	$6.03 * 10^{-5}$	$5.44 * 10^{-2}$	très bonne	mauvaise
	full surface, scale = $[0.5 \ 0.6 \ 0.7]$	$1.00 * 10^{-4}$	$2.20 * 10^{-2}$	très bonne	mauvaise
	full surface, scale = $[1 \ 0.8 \ 1.2]$	$9.85 * 10^{-5}$	$2.54 * 10^{-3}$	très bonne	bonne
	fragment (60%), scale = $[0.5 \ 0.5 \ 0.5]$	$2.35 * 10^{-3}$	$6.87 * 10^{-2}$	très bonne	mauvaise
	fragment (60%), scale = $[0.5 \ 0.6 \ 0.7]$	$1.31 * 10^{-3}$	$3.05 * 10^{-2}$	très bonne	mauvaise
	fragment (60%), scale = $[1 \ 0.8 \ 1.2]$	$1.71 * 10^{-3}$	$3.84 * 10^{-3}$	très bonne	bonne
VEINS (1000 pts)	full surface, scale = $[0.5 \ 0.5 \ 0.5]$	$1.52 * 10^{-4}$	$4.80 * 10^{-2}$	très bonne	mauvaise
	full surface, scale = $[0.5 \ 0.6 \ 0.7]$	$1.70 * 10^{-4}$	$2.09 * 10^{-2}$	très bonne	mauvaise
	full surface, scale = $[1 \ 0.8 \ 1.2]$	$2.36 * 10^{-4}$	$5.63 * 10^{-3}$	très bonne	bonne
	fragment (40%), scale = $[0.5 \ 0.5 \ 0.5]$	$5.24 * 10^{-2}$	$5.62 * 10^{-1}$	mauvaise	mauvaise
	fragment (40%), scale = $[0.5 \ 0.6 \ 0.7]$	$1.03 * 10^{-1}$	$5.19 * 10^{-2}$	moyenne	mauvaise
	fragment (40%), scale = $[1 \ 0.8 \ 1.2]$	$1.77 * 10^{-3}$	$7.69 * 10^{-1}$	mauvaise	mauvaise

TABLE 2 – Comparaison de la qualité du recalage entre notre algorithme (FGR+) et FGR. Nous avons évalué le recalage de manière visuelle et subjective, puis nous avons calculé une métrique géométrique, la MSE point à point symétrique (moyenne de la MSE calculée de la source vers la cible et de la cible vers la source).