

Université de Montpellier
Faculté des Sciences

MASTER 2 INFORMATIQUE
SPECIALITE Images, Games and
Intelligent Agents

RAPPORT DE STAGE

effectué à

Institut des Sciences de l'Évolution de
Montpellier

du **01 mars** au **31 août 2015**

par

PELADAN Cécile

M.Gérard SUBSOL, M.Renaud LEBRUN
et M.Rodolphe GIROUDEAU

Remerciements

Je tiens à remercier, dans un premier temps, toute l'équipe pédagogique de l'université Montpellier et les intervenants professionnels responsables du MASTER Informatique IMAGINA, pour avoir assuré la partie théorique de celui-ci.

Je remercie M.Rodolphe GIROUDEAU mon tuteur universitaire pour ses conseils et son aide tout au long de ce stage ainsi que M.Gérard SUBSOL pour l'aide et les conseils, qu'il m'a apportés lors des différents suivis.

Je tiens à remercier M.Renaud LEBRUN, mon maître de stage à l'ISEM pour son accueil et son encadrement tout au long de ce stage.

Je remercie aussi toute l'équipe de l'ISEM pour m'avoir fait découvrir le milieu de la paléontologie et de l'anatomie comparative ainsi que l'équipe du LIRMM pour leurs conseils et leurs critiques qui ont guidé mes réflexions.

Table des matières

Table des matières	2
Table des figures	3
Liste des tableaux	5
1 Introduction	6
1.1 Contexte du stage	6
1.2 Objectif du stage	8
1.3 Les données utilisées	9
1.4 Planning prévisionnel	10
2 Problème et Méthodologie	11
2.1 Les problèmes	11
2.1.1 Comment optimiser l’affichage des maillages ?	11
2.1.2 Comment comparer deux surfaces tridimensionnelles ?	11
2.2 Méthodologies adoptées et outils utilisés	12
2.2.1 Stratégie adoptée pour l’affichage	12
2.2.2 Stratégie adoptée pour la comparaison	12
3 Solutions apportées	17
3.1 L’amélioration de la fluidité d’affichage	17
3.2 L’utilisation de la méthode ICP pour le recalage	19
3.2.1 Interface-utilisateur	19
3.2.2 ICP non-déformable	23
3.2.3 ICP déformable	23
3.2.4 Fonctions de visualisation des résultats	24
4 Conclusion	27
4.1 Les Résultats obtenus	27
4.1.1 L’affichage	27
4.1.2 Le Recalage	28
4.2 Difficultés rencontrées	41
4.3 Apports	41
5 ANNEXE	42
5.1 Annexe A : Les éléments des maillages	42
5.2 Annexe B : Interface de la fonction de recalage	43
5.2.1 Présentation de l’interface	43
5.2.2 Gérer les modifications de l’interface	46
5.3 Annexe C : KD-Tree	47
5.4 Annexe D : Calcul de la Transformation	49
5.5 Annexe E : Structure de MeshTools	50

Table des figures

1.1	Microtomographe Rayon X in-vivo Skyscan 1076	7
1.2	Principe d'acquisition d'un fossile.	8
1.3	Ise-MeshTools avec un squelette étiqueté du <i>lemur catta</i> (Maki).	8
1.4	Crâne de <i>Mus musculus</i> (souris).	8
1.5	Cavité endocranienne d' <i>Eulemur fulvus</i> ayant 86.410 points et 172.864 facettes avec ses 16 landmarks	9
1.6	<i>Talus</i> du <i>Lemur Catta</i> ayant 258.663 points et 517.346 facettes avec 9 landmarks.	9
1.7	Molaire M1 supérieure de <i>Phaner</i> ayant 41.431 points et 82.858 facettes avec 12 landmarks.	9
1.8	Gantt prévisionnel	10
2.1	Schémas des principes de <i>Vertex Array</i> (VA)(gauche) et de <i>Vertex Buffer Object</i> (VBO) (droite)	12
2.2	Exemple de transformations rigide, similitude, et affine sur un cube	13
3.1	Crâne de <i>Leptadapis</i> avec ses oreilles internes, peu visible.	17
3.2	Crâne de <i>Leptadapis</i> avec ses oreilles internes bien visibles grâce à un changement d'ordre d'affichage des triangles selon l'axe de visualisation de la caméra.	17
3.3	<i>Talus</i> de <i>Lemur catta</i> en mode normal (droite). <i>Talus</i> de <i>Lemur catta</i> en mode "Tags" (milieu). <i>Talus</i> de <i>Lemur catta</i> en mode "Scalar" montrant sa courbure moyenne (gauche).	18
3.4	Crâne de loup domestiqué (ancien chien domestique) avec sa couleur d'origine.	18
3.5	Interface du Recalage.	19
3.6	Schéma montrant la relation des modes de transformation et leurs dépendances.	20
3.7	Deux maillages de cavités endocrâniennes de primate. En bleu le maillage Source : <i>Arctocebus calabarensis</i> , et en gris le maillage Cible : <i>Eulemur fulvus</i>	21
3.8	Exemple de résultats d'affichage	24
3.9	Exemple d'affichage des correspondances des landmarks et des points	24
4.1	Maillage de cavité endocranienne d' <i>Eulemur fulvus</i> avec ses 16 landmarks contenant 86.410 points.	28
4.2	Maillage de cavité endocranienne d' <i>Arctocebus</i> avec ses 16 landmarks contenant 285.665 points.	28
4.3	Mise en place des deux maillages et de leurs landmarks. Les landmarks jaunes sont les landmarks cibles. Les landmarks rouges sont les landmarks sources.	28
4.4	Mode Rigide des deux cavités endocrâniennes présentées : en bleu le maillage source <i>Arctocebus</i> et en jaune le maillage cible <i>Eulemur fulvus</i>	29
4.5	Résultats obtenus en mode Rigide	29
4.6	Mode Similitude des deux cavités endocrâniennes présentées : en bleu le maillage source <i>Arctocebus</i> et en jaune le maillage cible <i>Eulemur fulvus</i>	30
4.7	Résultats obtenus en mode Similitude	30
4.8	Mode Affine des deux cavités endocrâniennes présentées : en bleu le maillage source <i>Arctocebus</i> et en jaune le maillage cible <i>Eulemur fulvus</i>	31

4.9	Résultats obtenus en mode Affine	31
4.10	Mode Localement Affine des deux cavités endocrâniennes présentées à 5% du diamètre Source : en bleu le maillage source <i>Arctocebus</i> et en jaune le maillage cible <i>Eulemur fulvus</i>	32
4.11	Résultats obtenus en mode Localement Affine à 5%.	32
4.12	Mode Localement Affine des deux cavités endocrâniennes présentées à 2% du diamètre Source : en bleu le maillage source <i>Arctocebus</i> et en jaune le maillage cible <i>Eulemur fulvus</i>	33
4.13	Résultats obtenus en mode Localement Affine à 5% puis à 2%	33
4.14	Résultat du mode Localement Affine à 5% affichant en vert les points sources proche de son point cible le plus proche de 0.12mm de distance.	34
4.15	Résultat du mode Localement Affine à 5% puis à 2% affichant en vert les points sources proche de son point cible le plus proche de 0.12mm de distance.	34
4.16	Comparaison entre les formes obtenues (de gauche à droite) : maillage Cible d' <i>Eulemur</i> (marron), Maillage Source d' <i>Arctocebus</i> déformé en mode Localement Affine à 2% (jaune), Source déformé en mode Localement Affine à 5%(jaune), Source déformé en mode Affine (jaune), Source déformé au mode Similitude (jaune), maillage Source initial (marron).	35
4.17	Les deux maillages en haut sont le maillage Source (gauche) puis le maillage cible (droite). Au milieu, on a le recalage en mode Localement Affine à 5% (gauche) et la distance entre les deux maillages à un seuil de 0.05mm (droite). En bas, on a le recalage en mode Localement Affine (gauche) à 5% puis à 2% et la distance entre les deux maillages à un seuil de 0.05mm (droite).	36
4.18	Résultats obtenus en mode Localement Affine à 5% puis à 2% pour les maillages <i>tali</i>	36
4.19	Maillage de Molaire supérieur de <i>Planer</i> et de <i>Leptadapis</i> avec leurs 12 landmarks. Le maillage source est le premier maillage avec ses 41.431 points. Le maillage cible est le deuxième maillage avec ses 102.404 points.	37
4.20	Résultat obtenu des maillages de molaires avec quelques correspondances, soit 205 correspondances, au mode Localement Affine à 5%.	37
4.21	Résultat obtenu des maillages de molaires avec quelques correspondances, soit 205 correspondances, au mode Localement Affine à 5% en utilisant le paramètre qui enlève les valeurs extrêmes mis à 40%.	37
4.22	logiciel MorphoTools avec ses trois principales fenêtres : les espèces à comparer (gauche), l'analyse en composantes principales (milieu), la maillage déformable en fonction des modifications des paramètres (droite).	39
4.23	ACP portant sur le protocole impliquant 16 landmarks (gauche). La légende de couleur et de forme définissant les espèces et leur appartenance. ACP relative au protocole impliquant 572 semi-landmarks. (droite).	39
4.24	Crâne de gorille (primate anthropoïde) à gauche, et d' <i>Eulemur mongoz</i> (lému-rien), à droite.	40
5.1	Découpage de l'interface de la fonction de recalage	43

5.2	Diagramme de classes de ISE-MeshTools : les classes en blanc sont du logiciel d'origine et en vert sont celles que j'ai rajoutées	50
-----	---	----

Liste des tableaux

2.1	L'algorithme Iterative Closest Point.	13
3.1	Le changement de couleur selon le mode et l'interaction.	18
3.2	Table des transformations pour chaque mode du recalage.	20
3.3	Distance euclidienne du KD-Tree.	22
3.4	Algorithme ICP non-déformable du recalage.	25
3.5	L'algorithme de Itérative Closest Point Déformable de l'article [4].	26
4.1	Résultat de l'affichage en frame per second (FPS)	27
5.1	Algorithme du point le plus proche de complexité $O(n \cdot \log(n))$ avec D constant. 47	
5.2	Algorithme cherchant les points dans une sphère de center p et de rayon R de complexité $O(n \cdot \log(n))$	48

Chapitre 1

Introduction

1.1 Contexte du stage

Présentation de l'entreprise

Mon stage se déroule à l'Institut des Sciences de l'Évolution de Montpellier (ISEM¹) sur le campus de l'Université de Montpellier. Ce laboratoire héberge 10 plateformes techniques, et ses effectifs sont répartis en 5 départements et 17 équipes de recherche, travaillant sur des thèmes différents.

Les recherches portent sur l'origine et la dynamique de la biodiversité, sur les modalités et les mécanismes de l'évolution. Elles concernent la biodiversité d'hier et d'aujourd'hui sur un large éventail d'organismes et de milieux. Leurs approches pour ces recherches sont autant sur le terrain, qu'expérimentales et théoriques.

Les plateformes sont mises à la disposition de toutes personnes, interne ou extérieure à l'ISEM, désirant les utiliser.

Le département Forme

Ce département s'attache à une meilleure compréhension des mécanismes permettant l'émergence de la biodiversité. Il associe l'étude des mécanismes fins à l'origine de l'émergence et de la variabilité des formes. La vision paléontologique de cette évolution, c'est-à-dire la paléobiodiversité, fait partie de ce département.

Il est composé de deux équipes de recherche : Biologie du développement et évolution et Paléontologie.

L'équipe de Paléontologie

Dans cette équipe, les chercheurs étudient l'Évolution dans l'espace et dans le temps par l'intermédiaire des fossiles, et observent des "patterns" macro-évolutifs. Leur étude des fossiles permet également de constater la dynamique de la biodiversité au cours du temps dans son contexte paléogéographique, notamment par rapport aux changements environnementaux. En effet, la biodiversité actuelle est le résultat d'une longue histoire évolutive qui a été marquée et conditionnée par des phénomènes répétés et brefs de diversification (radiations adaptatives) et d'extinction (crises).

Ainsi, les recherches de l'équipe Paléontologie au sein du département Forme se focalisent sur :

- Phylogénie/ Paléobiogéographie (Origine et radiation des mammifères actuels ; Relations paléobiogéographiques Afrique/Asie/Europe ; Lignées évolutives, etc.)

1. <http://www.isem.univ-montp2.fr/>

- Paléoécologie et Paléobiologie (Identification et reconstruction des interactions écologiques entre les organismes passés afin de comprendre les mécanismes d'adaptations qui caractérisent les grandes radiations, etc.)
- Extinctions / Crises (Développement des approches pluridisciplinaires associées aux recherches en relation avec la compréhension des perturbations environnementales à l'origine des crises biotiques, etc.)

Je suis dans l'équipe de Paléontologie auprès de M.Renaud LEBRUN qui s'occupe du plateau "Microtomographie" de la plateforme MRI.

La plateforme MRI²



L'ISE-M héberge le plateau technique "microtomographie RX"; rattaché à la plateforme régionale MRI. Ce plateau dispose d'un microtomographie in-vivo SkyScan 1076 ainsi que de stations de travail pour l'analyse des images obtenues.



FIGURE 1.1 – Microtomographe Rayon X in-vivo Skyscan 1076

Le microtomographe permet d'acquérir des images tridimensionnelles de plantes et d'animaux vivants (souris, rat) à des résolutions comprises entre $9\mu\text{m}$ et $36\mu\text{m}$. Cette machine permet également d'imager des échantillons non vivants tels que des fossiles ou des objets non biologique.

Le microtomographe est utilisé dans le cadre de nombreux projets de recherche :

- Étude des structures internes (oreille interne, cerveau),
- Développement osseux,
- Reconstruction de fossiles,
- Vascularisation embryonnaire et chez les plantes,
- Qualité du maïs,
- Etc.

Principe d'acquisition :

Le couple détecteur / source de rayons X tourne autour de l'échantillon et génère des radios prises sous différentes orientations. A partir de l'ensemble de ces radiographies, on reconstruit une image 3D composée d'une pile de coupes. Cette image 3D peut alors être segmentée à l'aide de logiciels spécifiques (Avizo à l'ISEM) pour générer des maillages 3D des structures d'intérêt.

2. Montpellier RIO Imaging – <http://www.mri.cnrs.fr>

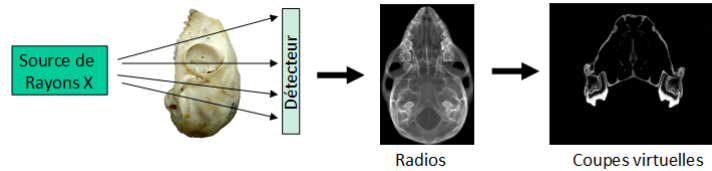


FIGURE 1.2 – Principe d’acquisition d’un fossile.

Pour poursuivre leurs recherches à partir des maillages 3D, les paléontologues, les paléoanthropologues et les biologistes vont devoir utiliser d’autres logiciels comme ISE-MeshTools.

ISE-MeshTools

ISE-MeshTools est un logiciel d’édition graphique de maillages 3D triangulaires (3D meshes), développé à l’ISE-M.

Ce logiciel, écrit en C++ par Renaud LEBRUN avec les bibliothèques VTK³ et FLTK⁴, permet notamment de manipuler des maillages 3D, de localiser des points de repère (landmarks et semi-landmarks), de construire des atlas anatomiques en étiquetant et en colorant des structures d’intérêt sur des maillages 3D et de rétro-déformer des spécimens endommagés. ISE-MeshTools a été mis à la disposition de la communauté sur le site MorphoMuseum⁵.

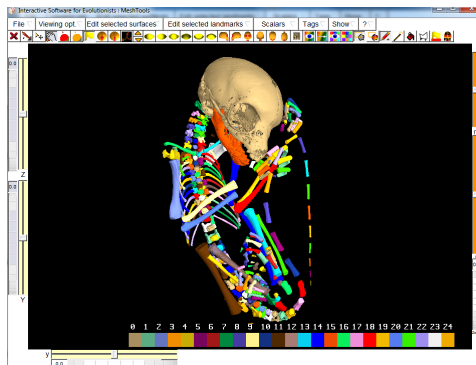


FIGURE 1.3 – Ise-MeshTools avec un squelette étiqueté du *lemur catta* (Maki).

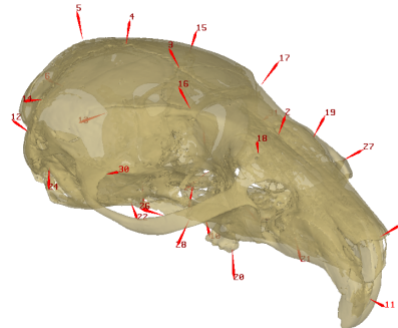


FIGURE 1.4 – Crâne de *Mus musculus* (souris).

1.2 Objectif du stage

Durant ce stage, j’ai implémenté des fonctionnalités visant à permettre la comparaison de la géométrie de spécimens biologiques représentés par des maillages 3D. Ces fonctionnalités impliquent d’utiliser des modèles déformables 3D de référence qui en s’ajustant aux maillages 3D, permettent de générer automatiquement des (semi-)landmarks 3D. Ces améliorations permettent d’augmenter les possibilités de comparaison des régions anatomiques où les points de repère sont difficiles à identifier.

3. bibliothèque comportant plusieurs méthodes : <http://www.vtk.org/>
 4. bibliothèque servant à construire une interface : <http://www.ftk.org/>
 5. <http://morphomuseum.com>

De plus, Le second objectif principal est d'optimiser le code existant, en particulier pour accélérer l'affichage 3D, en utilisant les fonctionnalités GPU les plus avancées d'OpenGL.

1.3 Les données utilisées

Après avoir scanné le spécimen original (crâne entier, os du pied ou dent isolée) la structure d'intérêt (moulage endocrânien, surface osseuse, surface externe de la dent) est segmentée afin d'obtenir un maillage de taille raisonnable, d'un ordre de grandeur d'environ 10^5 points.

Durant ce stage, j'ai utilisé ces maillages 3D de plusieurs centaines de milliers de points et facettes correspondant à diverses structures anatomiques de primates :

- Cavité endocrânienne qui est l'empreinte du cerveau à l'intérieur du crâne ;
- *Talus* (appelé également "astragale"), qui est un os situé au niveau de la cheville ;
- Première molaire supérieur (M1 sup.).

Un expert a associé à chacun de ces maillages un ensemble de points de repères 3D (également nommés landmarks), à des positions correspondant à des identités de structure, que l'on suppose être homologues. Pour une structure donnée (endocrâne, *talus* ou M1 sup.), le nombre de landmarks doit être identique, et les landmarks sont digitalisés dans le même ordre. Pour certains d'entre eux, différentes régions (zones du cerveau pour les endocrânes, facettes d'articulation pour les *tali*) que l'on suppose homologues ont été coloriées.

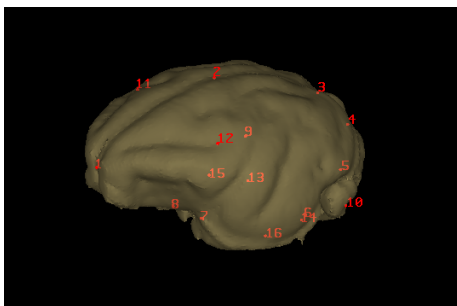


FIGURE 1.5 – Cavité endocrânienne d'*Eulemur fulvus* ayant 86.410 points et 172.864 facettes avec ses 16 landmarks

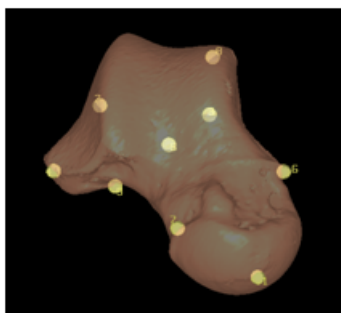


FIGURE 1.6 – *Talus* du *Lemur Catta* ayant 258.663 points et 517.346 facettes avec 9 landmarks.

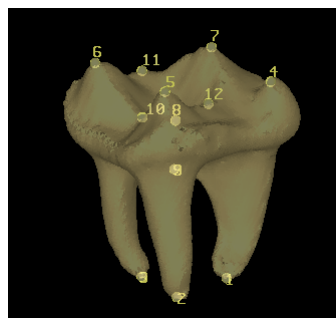


FIGURE 1.7 – Molaire M1 supérieure de *Phaner* ayant 41.431 points et 82.858 facettes avec 12 landmarks.

1.4 Planning prévisionnel

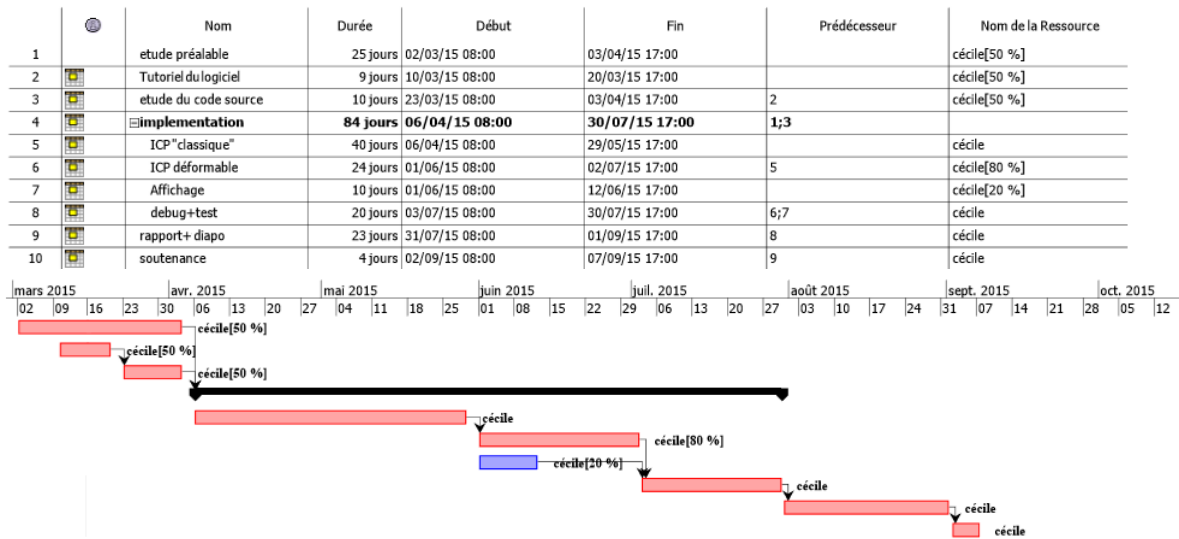


FIGURE 1.8 – Gantt prévisionnel

Pour comparer la morphologie des spécimens, une recherche bibliographique sur ce sujet doit être faite. De plus, on fait quelques approfondissements sur l'amélioration de l'affichage en utilisant les fonctionnalités GPU d'OpenGL.

A partir de ces articles trouvés, la comparaison et la déformation d'un maillage à un autre se fait en deux parties de manière globale (l'ICP "classique") puis de manière locale (ICP déformable).

Afin de comprendre mieux le logiciel, quelques tutoriels sont mis à la disposition. De plus, pour faciliter l'implémentation, une étude du code de la structure et des bibliothèques VTK et FLTK doit être faite.

On fait l'implémentation des deux comparaisons ainsi que l'affichage dans le logiciel avec des tests et des debugs.

Enfin, on fait la rédaction du rapport et des diapositives pour la soutenance.

Chapitre 2

Problème et Méthodologie

2.1 Les problèmes

2.1.1 Comment optimiser l'affichage des maillages ?

Le Logiciel ISE-MeshTools, lui, utilise les fonctions "immédiates" d'affichage d'OpenGL pour faire le rendu des maillages. Ces fonctions *glBegin* et *glEnd* vont encadrer la description des propriétés des points c'est-à-dire leurs couleurs et leurs coordonnées, leurs normales (*glColor3f*, *glVertex3f*, *glNormal3f*).

Pour calculer le nombre de valeurs envoyées au processeur graphique, on a

$$valeurs_{envoyées} = m \cdot np \cdot nPp \cdot nCr$$

où m est le nombre de facettes ; np est le nombre de points par facette ; nPp est le nombre de propriétés pour chaque point ; nCr est le nombre de coordonnées de chaque propriété.

Dans notre cas, on a $np = nPp = nCr = 3$. Pour un gros maillage d'environ 1 million de points et 2 millions de facettes, on a ainsi $2 \cdot 10^6 \cdot 27$ valeurs envoyées et donc environ entre 1 et 4 FPS.

On obtient alors un affichage saccadé dû au recalcul à chaque modification de ces maillages, par exemple lors d'un déplacement ou d'un changement de couleur ainsi qu'au nombre d'appels identiques à chaque frame et au transfert de ses données envoyées au GPU par chaque appel.

La technique de dessin "Immédiat" est donc très contraignante du point de vue de la performance pour une application "temps-réel".

On veut donc diminuer le nombre de valeurs envoyées au processeur graphique pour avoir plus d'images par seconde (FPS), et ainsi afficher rapidement ces maillages lorsque le point de vue est modifié.

Comment améliorer la fluidité de l'affichage durant une manipulation de ces maillages ?

2.1.2 Comment comparer deux surfaces tridimensionnelles ?

Afin de comparer les différences de morphologie entre deux échantillons de spécimens, les experts posent des points de repères sur chaque échantillon pour établir des relations dites d'homologie c'est-à-dire des points se correspondant pour une même signification anatomique ou biologique.

Cependant, ce travail est fastidieux et difficile pour un expert d'identifier précisément ces correspondances point à point entre deux objets. Certaines de ses structures peuvent comprendre des étendues relativement "planes" comme la facette d'articulation, les zones sans foramens ou encore les structures osseuses pauvres en sutures ; soit des étendues relativement pauvres en structures homologues bien établies, comme par exemple, un moulage endocrânien.

On cherche ainsi à définir un outil qui utiliserait l'ensemble des points des deux surfaces à comparer pour établir des correspondances de tous les points des surfaces.

Ceci fait référence au problème du recalage déformable de surfaces 3D.^{1 2 3}

Autrement dit, on veut avoir les correspondances des points entre les deux maillages sur toutes leurs surfaces.

2.2 Méthodologies adoptées et outils utilisés

2.2.1 Stratégie adoptée pour l'affichage

En OpenGL, il existe deux méthodes pour optimiser l'affichage : les vertex array (VA) et les vertex buffer object (VBO).

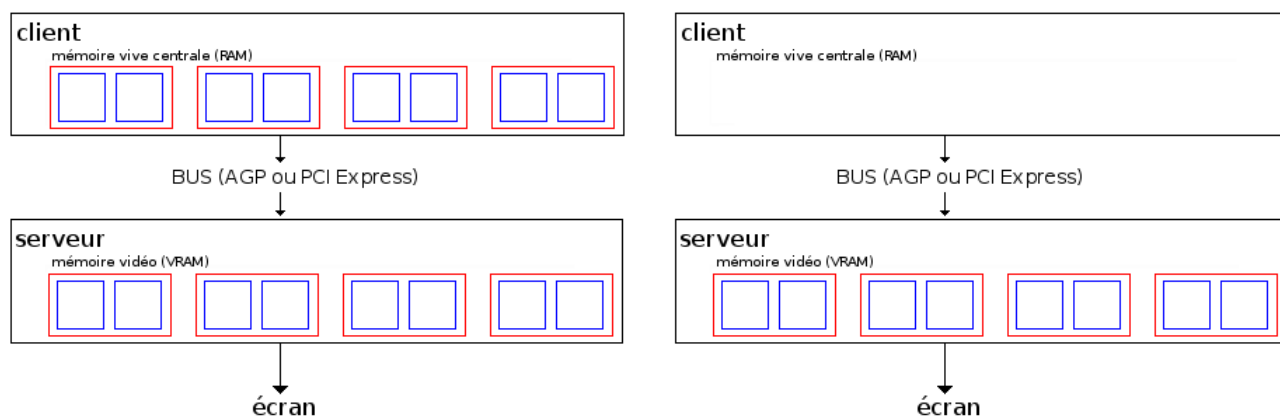


FIGURE 2.1 – Schémas des principes de *Vertex Array* (VA)(gauche) et de *Vertex Buffer Object* (VBO) (droite)

Comme le montre la figure ci-dessus⁴, les VAs permettent de dupliquer les données en mémoire vidéo puis de les traiter afin d'être affichées à l'écran. Mais ceci oblige à envoyer toutes ces données à la carte graphique à chaque fois qu'elles doivent être affichées. Les VBOs, eux, permettent d'envoyer les données directement vers la carte graphique. Ainsi, ces données ne sont pas dans la mémoire système mais dans la mémoire de la carte graphique ce qui permet un rendu plus rapide.

Pour notre affichage, les VBOs sont plus efficaces que les VAs puisqu'on manipule et modifie continuellement de gros maillages.

Chaque objet dispose de ses propres buffers pour gérer son affichage.

2.2.2 Stratégie adoptée pour la comparaison

Etude bibliographique

Dans de nombreux articles traitant sur le recalage, j'ai pu constater que l'algorithme Itérative Closest Point est énormément utilisé.

L'ICP a été créé par Paul J. Besl et Neil D. McKay (1992) [2] ainsi que Zhang (1992)[7].

Cependant, dans ces articles, j'ai pu constater qu'il existe de nombreuses variantes de l'ICP, en particulier sur l'espace de transformation.

1. <http://www.ncbi.nlm.nih.gov/pubmed/23661012>

2. <http://onlinelibrary.wiley.com/doi/10.1002/ar.23084/abstract>

3. <http://www.pnas.org/content/108/45/18221.abstract>

4. <http://sdz.tdct.org/sdz/rendus-optimises-dl-va-vbo.html#LesVBOdesdonneesetserveur>

ALGORITHME ICP : :
 ENTRÉE : surface Cible et surface Source
 SORTIE : surface Source modifiée

repète : À chaque itération i ,
étape 1 : On cherche les correspondances $M_i = (point_S, point_C) / point_S \in Source, point_C$
 est le point le plus proche de $point_{S'}$.
étape 2 : On calcule la transformation qui minimise la somme des distances au carré entre
 les
 points correspondants.
jusqu'à atteindre un critère d'arrêt.

TABLE 2.1 – L'algorithme Iterative Closest Point.

Definition 1. Les transformations suivantes vont servir à modifier successivement le maillage de différentes manières (voir TABLE 3.2) :

- La transformation Rigide permet de modifier l'orientation du maillage Source et de le déplacer (inclut la translation).
- La transformation Similitude est composée d'une transformation rigide et d'une transformation uniforme de 'mise à l'échelle'.
- La transformation Affine est une transformation d'un point par une matrice affine.

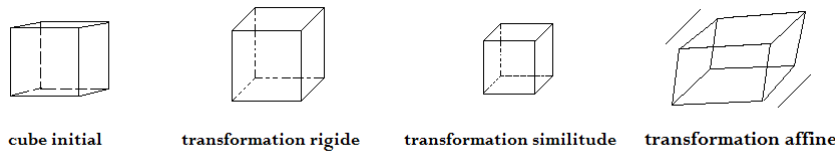


FIGURE 2.2 – Exemple de transformations rigide, similitude, et affine sur un cube

Amberg et al. (2007) [1] ne regardent que l'ICP déformable.

Chaque point de la source (M_i) va se déformer selon une transformation affine ayant en plus la translation X_i (matrice 4×3) qui lui est propre. Chacune de ces transformations va être stockée dans un ensemble $X = \{X_1, \dots, X_n\}$. Cet algorithme itératif a besoin des correspondants à l'itération précédente X_{j-1} pour trouver les nouvelles à l'itération suivante X_j .

Pour calculer cet ensemble, on utilise une fonction d'erreur $E(X)$ composée d'un terme de raideur E_s , d'un terme de distance E_d entre un sommet M_i et son point le plus proche en utilisant un Kd-tree et enfin d'un terme de landmarks E_l :

$$E(X) = E_d(X) + \alpha * E_s(X) + \beta * E_l(X)$$

(α et β sont des poids). A partir de cette équation, les auteurs isolent X .

Remarque :

- Les landmarks aident au repositionnement des deux modèles.
- On peut facilement ajouter des fonctionnalités pour trouver le point le plus proche, par exemple utilisation de la normale, ou la courbure.

Castellani et Bartoli (2012) [3] cherchent un recalage qui peut se faire en temps réel, en utilisant un algorithme basé sur ICP appelé LM-ICP. Cette technique mélange la méthode de Gauss-Newton et la descente de gradient. Tout d’abord ils font une déformation rigide puis la déformation non-rigide de cette méthode :

- Pour LM-ICP, ils utilisent la matrice Jacobienne qui va calculer les dérivées partielles de la fonction d’erreur E qui est le moindre carré non linéaire. Cette fonction E est composée de la transformation de distance cherchant la plus petite distance entre un point de la source et la cible. LM-ICP est 30 fois plus rapide à chaque itération contrairement à l’ICP basique. Sur un maillage d’environ 40.000 points, 20 itérations au total pour LM-ICP ont été appliquées.
- Pour le déformable (DLM-ICP), la fonction d’erreur E est une somme de termes de données et de pénalités : l’attraction globale, le traitement des frontières, le lissage spatial, temporel et enfin les contraintes de surface pour éviter la non-extensibilité de la surface. Puis comme précédemment ils transforment E en matrice jacobienne.

Castellani et Bartoli teste cet ICP déformable sur une surface 3D avec un capteur prenant les images à 25 FPS : une image 2D et information 3D. En résumé, cette méthode non rigide extrait les contours 2D et les projections 2D du modèle estimé puis fait la création du nuage de points sur la grille et des contours à l’initialisation. Ensuite, chaque itération utilise le résultat précédant comme une condition initiale. Ceci permet d’avoir une déformation en temps réel.

Fedmar et Ayache (1996)[4] vont appliquer l’algorithme ICP de manière rigide puis affine et enfin affine localement.

- transformation rigide : les auteurs cherchent une première estimation (R_0, t_0) en prenant au hasard un point M dans la surface Source S et en cherchant l’ensemble des points dans la Cible T ayant la courbure la plus proche de celle de ce point M . A partir de l’appariement estimé, ils calculent ensuite (R, t) et vérifient si cela fonctionne. Ils l’appliquent sur un sous ensemble S' de S . Si $(P \in S')$, $R * P + t$ est le point le plus proche de $Q \in T$ et que leur différence est plus petit d’un certain seuil alors (P, Q) est une bonne paire. Si la norme de chaque bonne paire est inférieure à un seuil, on recommence en choisissant un autre point M sinon on applique (R, t) trouvé à tous les points de S .
- transformation affine : pour l’initialisation, ils prennent $(A_0, b_0) = (R_{final}, t_{final})$. Pour calculer le point le plus proche, les points 3D sont transformés en vecteurs 8D en ajoutant la normale du point et ses deux principales courbures. On va appliquer l’ICP en utilisant une distance euclidienne en dimension 8. On utilise le kd-tree pour trouver le point le plus proche. Une fois convergé, on passe à la transformation affine locale.
- transformation affine locale : Pour chaque point M_k de S , on a une transformation affine (A_{ki}, b_{ki}) . On prend un voisinage. On fait une étape ICP en rigide. Puis après, on trouve une transformation affine (A_{ki}, b_{ki}) en lissant les (R_{ki}, t_{ki}) . On applique et on itère.

Ces algorithmes s’arrêtent si : soit la distance des correspondances des deux surfaces est inférieure à β soit la variation de la distance des correspondances des deux surfaces à deux itérations successives est inférieure à α soit on atteint son nombre d’itérations maximal.

Sur un maillage Source d’environ 30.000 points et de diamètre u , les auteurs obtiennent un résultat après 8 itérations en 8 minutes (sur des stations de travail datant des années 1990) pour la transformation affine. L’erreur, calculée par la distance moyenne entre un point transformé et son point le plus proche, passe de $0.0120u$ pour le rigide à $0.0098u$ pour la transformation affine. Pour la transformation localement affine sur le maillage,

après 3 itérations en 4 minutes, les auteurs obtiennent une erreur de 0.003u. Ils ont une superposition quasiment parfaite.

Papazov et Burschka (2011)[5] font un recalage déformable en utilisant une équation différentielle ordinaire (ODE). Le système ODE va permettre la déformation des points de la source vers des points correspondants sur la forme cible. Leur algorithme s'arrête lorsque le paramètre de rigidité est en dessous d'un seuil.

Les auteurs construisent un graphe déformable basé sur la forme source pour pouvoir améliorer la complexité. Ce graphe sous forme d'octree représente la maillage source avec une résolution plus petite. Il va être utilisé pour le recalage au lieu de la forme source S initiale. Une fois l'ICP appliqué, on va calculer l'interpolation TPS (thin-plate Spline) entre le graphe initial et obtenu. Enfin on l'applique à chaque point de S.

Ce recalage est testé sur plusieurs modèles différents complets ou incomplets avec environ entre 30.000 et 500.000 points. Les auteurs utilisent des landmarks, mis manuellement, pour calculer le déplacement rigide. Cet algorithme est comparé avec six autres algorithmes de recalage. Il a un meilleur temps de calcul et un meilleur taux d'erreur entre S déformé et T.

Le fait de passer par le graphe déformable permet de réduire considérablement le nombre de points pour le calcul du recalage. Par exemple pour un maillage S d'environ 40.000 points, on a environ 1.500 points pour le graphe déformable.

Rusinkiewicz et Levoy (2001) [6] comparent des différentes variations de l'ICP sur :

- la sélection des points : on prend tous les points, ou un sous-ensemble ou de manière aléatoire, etc.
- les correspondances des points : utilisation du Kdtree, faire une projection des points S sur le maillage Cible, utilisation des données supplémentaires (couleur, normale), etc.
- les poids pour les correspondances trouvées : un poids constant, basé sur la distance des points de la correspondances ou basé sur la compatibilité de leur normale, etc.
- les paires à rejeter : la distance de ces points est supérieur à un seuillage des distances entre points correspondants ; on enlève les 10% de paires en se basant sur la distance ; les paires qui sont sur les frontières du maillage ; etc.
- les erreurs métriques et la minimisation : utilisation de la somme des distances carrées entre les correspondances, utilisation de la distance carrée de chaque point source à un plan, trouver un nouvelle transformation se basant sur les correspondances pour minimiser les erreurs métriques, etc.

An final, Rusinkiewicz et Levoy construisent un algorithme d'ICP exécuté en temps réel en combinant certaines des variantes décrites ci-dessus. Ils proposent d'utiliser un algorithme basé sur la projection pour générer des points des correspondances ainsi qu'une erreur métrique point-à-plan et l'itération standard de l'ICP. Pour les autres étapes du processus de ICP, ils choisissent les plus simples, un échantillonnage aléatoire, une pondération constante, et un seuil de distance pour rejeter paires puisque ces étapes n'ont que peu d'incidence sur la vitesse de convergence.

Méthodologie adoptée

J'ai donc choisi de développer un recalage contenant une méthode d'ICP non-déformable (ou "classique") et une méthode d'ICP déformable.

La première méthode mettra en place une ressemblance globale entre les deux maillages à comparer. En plus des points des maillages, on pourra utiliser les landmarks placés sur

les objets par l'expert pour aider à calculer les trois transformations qu'elle contient : Rigide, Similitude et Affine.

La deuxième méthode permettra d'accentuer leurs ressemblances locales. Chaque point se déformera indépendamment des autres. J'ai appliqué la méthode de l'article de Feldmar et Ayache (1996) [4]. Cette méthode permettra de faire une déformation locale d'un point tout en considérant les points qui l'entourent dans un certain rayon R . (voir section 3.2.3 TABLE 3.5)

De plus, on veut prendre en compte les différentes données des points :

- les coordonnées du point (x, y, z) ,
- les coordonnées de la normale (n_x, n_y, n_z) ,
- les courbures principales (k_{min}, k_{max}) ,
- éventuellement la couleur (Tag) (ind_c).

Les normales permettront de distinguer les surfaces intérieures des surfaces extérieures des maillages. Les courbures serviront à préciser le point le plus proche dans le KDtree. Les tags sur les maillages définissant les mêmes zones aideront à filtrer les correspondances trouvés.

Le KDtree permettra de trouver le point le plus proche rapidement. Pour la distance euclidienne dans la recherche du point le plus proche, je me suis inspirée de la méthode employée dans l'article de Feldmar et Ayache(1996)[4]. Leur distance euclidienne permet d'utiliser des points à 8 dimensions.

De plus, pour calculer les différentes transformations, j'ai choisi de reprendre la transformation du moindre carré dans la bibliothèque VTK. Je l'ai adaptée pour obtenir une transformation du moindre carré pondérée qui servira à donner plus d'importance à certaines correspondances trouvées.

Après chaque ICP effectué, on affiche la source déformée et ses correspondances avec la cible si l'utilisateur le souhaite (visualisation de lignes entre les points de la Source et de la Cible).

Chapitre 3

Solutions apportées

3.1 L'amélioration de la fluidité d'affichage

Pour augmenter la fluidité du rendu visuel, j'ai mis en place le Vertex Buffer Object pour chaque surface à afficher.

Chaque maillage va contenir ses propres buffers :

- un buffer contenant toutes les informations de chaque point c'est-à-dire les coordonnées du point, les coordonnées de sa normale et sa couleur RGB.
- un deuxième buffer contenant la structure du maillage c'est-à-dire les indices des points construisant chaque triangle de la surface.

Si une des fonctionnalités suivantes qui joue sur la visualisation du maillage est activée, on modifie soit le rendu soit dans les buffers les éléments du maillage qui ont changé par cette activation.

Les différentes fonctionnalités de visualisation du logiciel

- On a 4 modes d'affichage que l'on peut changer qui sont dans la barre de menu : l'ombrage de Gouraud (Gouraud shading), le fil de fer (wireFrame), l'ombrage "flat triangle" et enfin le fil de fer avec l'ombrage "flat triangle".
- L'utilisateur a la possibilité de voir à travers un maillage les maillages situés à l'intérieur de ce dernier. Pour cela, on change l'ordre d'affichage des triangles selon l'axe de visualisation de la caméra. Les triangles les plus près de la caméra sont dessinés puis les plus loin de la caméra.

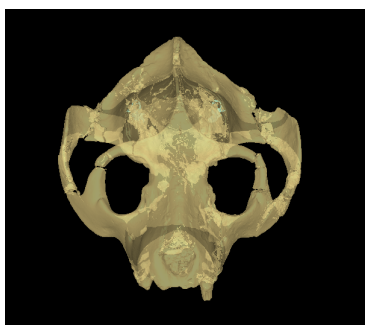


FIGURE 3.1 – Crâne de *Leptadapis* avec ses oreilles internes, peu visible.

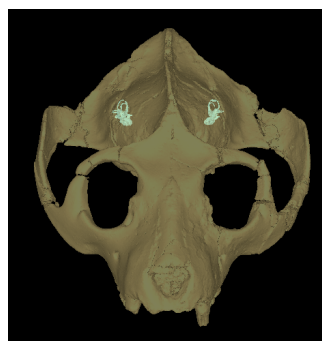


FIGURE 3.2 – Crâne de *Leptadapis* avec ses oreilles internes bien visibles grâce à un changement d'ordre d'affichage des triangles selon l'axe de visualisation de la caméra.

- Suivant les actions de l'utilisateur sur l'objet, la couleur du maillage peut être amenée à changer.

Le mode "Tags" permet à l'utilisateur de visualiser ou de colorier des régions d'intérêt sur le maillage.

Le mode "Scalar" permet d'afficher d'autres informations sur les caractéristiques du maillage comme la courbure.

Mode	Maillage sélectionné	Maillage non-sélectionné
Mode Normal	gris	couleur globale de départ
Mode "Tags"	couleur "Tags" si elle existe sinon couleur globale de départ	gris
Mode "Scalar"	gris	couleur "Scalar" si elle existe sinon couleur globale de départ

TABLE 3.1 – Le changement de couleur selon le mode et l'interaction.

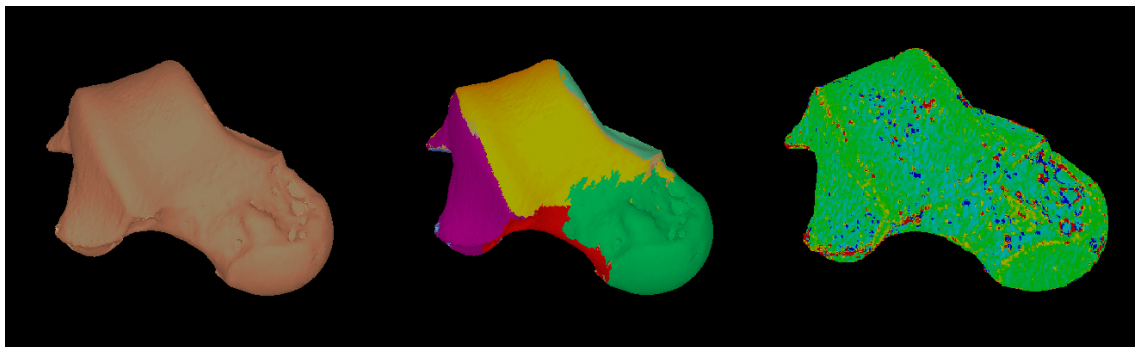


FIGURE 3.3 – Talus de *Lemur catta* en mode normal (droite). Talus de *Lemur catta* en mode "Tags" (milieu). Talus de *Lemur catta* en mode "Scalar" montrant sa courbure moyenne (gauche).

De plus, j'ai ajouté une fonctionnalité permettant à l'utilisateur d'afficher la couleur d'origine d'une surface pour un fichier .ply ou .vtk.



FIGURE 3.4 – Crâne de loup domestiqué (ancien chien domestique) avec sa couleur d'origine.

- L'utilisateur peut déplacer le maillage indépendamment de la caméra. Ceci permet par exemple à l'utilisateur de positionner différents maillages les uns par rapport aux autres, ou de centrer un maillage au niveau de l'origine du système de coordonnées, ou encore d'orienter un maillage de manière à le visualiser selon les orientations anatomiques standard. Dans ce cas-là, les points du maillage ne sont pas directement modifiés.

En effet, chaque surface contient deux matrices dites "de positions" qui vont être modifiées à chaque déplacement du maillage. On peut aussi les modifier directement dans l'interface du logiciel. L'une des deux permet de modifier l'orientation du maillage (rotation, translation) et l'autre de modifier son aspect, par exemple si l'objet se retrouve écrasé dans une des 3 directions (x,y,z), on augmente dans cette matrice le poids du paramètre associé.

Malgré toutes les modifications que l'on a pu faire sur une surface, en gardant les positions initiales des points et en ne modifiant que ces matrices, on peut réutiliser et remodeler le maillage dans son logiciel de création (AVIZO, par exemple). De plus, ceci permet aussi d'éviter de créer d'autre fichier du maillage dans une position différente.

Remarque : Les points et les normales contenus dans le VBO sont les points d'origine modifiés par les deux matrices.

3.2 L'utilisation de la méthode ICP pour le recalage

3.2.1 Interface-utilisateur

La structure englobante est l'interface du recalage dans le logiciel. Elle permet de définir les ICP et les données à utiliser comme la dimension des points, les poids utilisés dans le KDtree et le mode de transformation. (Voir ANNEXE B)

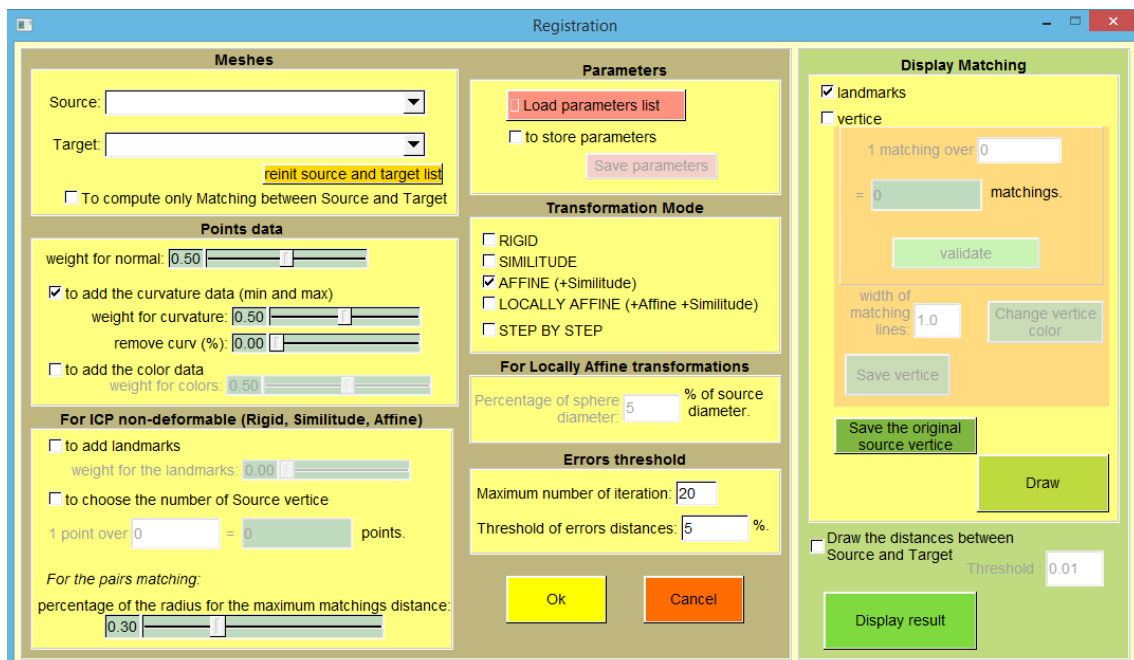


FIGURE 3.5 – Interface du Recalage.

La fonction du recalage est composée de deux ICP : ICP "classique" et ICP déformable.

L'ICP classique contient trois transformations : Rigide, Similitude et Affine.

L'ICP déformable, lui, n'a que la transformation Localement Affine.

Les Modes de transformation

Les transformations Rigide, Similitude et Affine vont changer le maillage Source de manière globale. Cependant, la transformation Localement Affine va agir de manière indépendante sur chaque point source.

Mode	Transformations				
	Rotation	Translation	Echelle uniforme	Echelle non-uniforme	Cisaillement
Rigide	x	x			
Similitude	x	x	x		
Affine	x	x	x	x	x
Localement Affine (pour chaque point)	x	x	x	x	x
Step-by-Step	dépend du mode dans lequel on se trouve.				

TABLE 3.2 – Table des transformations pour chaque mode du recalage.

Remarque :

- Le mode Step-by-Step permet de passer d'un mode à l'autre par la touche "Espace" ou le "ok" de l'interface (voir FIGURE 3.7). Ce mode commencera toujours par le mode Rigide.
- On peut aussi changer de mode par la touche "Espace" tout en gardant les mêmes données pour les autres modes.

Le mode sélectionné correspond au mode final à atteindre. Mais la fonction du recalage peut faire un ou deux modes de transformation avant de l'atteindre (voir ci-dessous FIGURE 3.8).

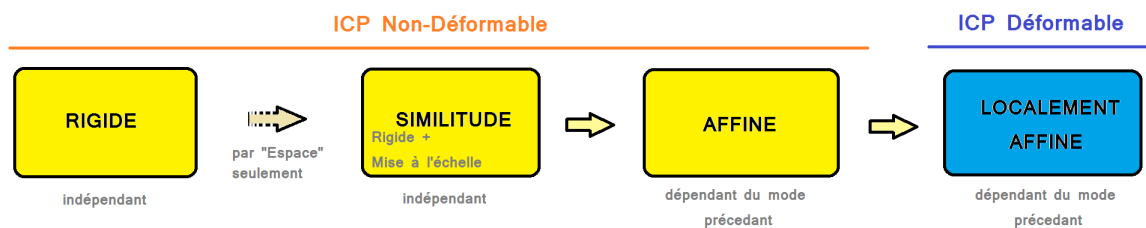


FIGURE 3.6 – Schéma montrant la relation des modes de transformation et leurs dépendances.

En effet, par exemple, pour le mode Affine, si les objets sont de position et d'échelle différentes, l'objet défini comme Source doit se repositionner et modifier sa taille pour s'approcher au mieux de l'objet Cible. On doit alors appliquer la transformation Similitude avant de passer à la transformation Affine.

Remarque :

- Le mode de Similitude contient le mode Rigide en plus de la "mise à l'échelle". Ce mode n'est pas obligé de faire le mode précédent (voir FIGURE 3.9).

- A la fin du recalage, on affiche le nom du mode de transformation final dans l'interface du logiciel pour permettre plus de lisibilité à l'utilisateur.

Utilisation des landmarks selon les Modes de transformation

Tous les modes ne se servent pas forcément de ces points de repère.

Les modes Rigide et Similitude utilisent les landmarks puis les points du maillage.

En effet, pour ces modes, ceci permet d'avoir une position et une taille approximative par rapport au maillage Cible. Puis on refait l'ICP une deuxième fois avec les sommets pour affiner. Ceci permet de gagner aussi en temps de calcul.

Cependant, pour le mode Affine, on n'utilise pas les landmarks pour approximer l'objet cible. En effet, les landmarks étant placés par l'expert, on n'a que quelques landmarks qui sont placés de manière approximative. La transformation affine peut être donc très approximative. On voit bien sur la figure ci-dessous que l'utilisation des landmarks éloigne le maillage Source de la Cible (voir figure ci-dessus). L'utilisation seulement des points du maillage Source permet une meilleure précision dû au grand nombre de points servi pour calculer la transformation affine.

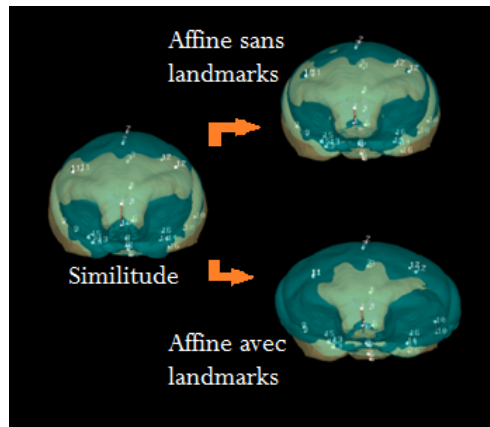


FIGURE 3.7 – Deux maillages de cavités endocrâniennes de primate. En bleu le maillage Source : *Arctocebus calabarensis*, et en gris le maillage Cible : *Eulemur fulvus*.

Enfin, pour le mode Localement Affine, on utilise tous les points du maillage Source puisque chaque point va avoir sa propre transformation affine indépendamment des autres.

Remarque :

Durant l'ICP non-déformable, les coordonnées des landmarks sont modifiées au fur et à mesure des étapes de transformation, et ce quel que soit le mode afin de voir à l'affichage le déplacement de ces derniers.

Cependant, on peut utiliser les landmarks en même temps que les sommets pour calculer les transformations de l'ICP non-déformable. Les landmarks auront un pourcentage d'importance défini par l'utilisateur. Ceci permet la possibilité à l'utilisateur d'ajuster comme il le souhaite la déformation de la source.

Initialisation du Recalage

Avant de faire les ICP, on va vérifier et/ou calculer certains paramètres :

- les objets Source et Cible doivent exister (ANNEXE A) ;
- le nombre de landmarks Source et Cible doit être identique ;

- si l'option a été cochée, la couleur doit être présente sur les maillages sources et cibles qui doivent comporter le même nombre de couleurs ;
- les courbures principales sont calculées. Les valeurs extrêmes de ces données sont modifiées selon un pourcentage choisi dans l'interface (voir ANNEXE B).

Une fois les vérifications faites et validées, on récupère les données des points que l'utilisateur a sélectionné dans l'interface (voir ANNEXE B).

Selon les données que l'utilisateur a sélectionnés dans l'interface, la dimension des points change. Au maximum, on peut avoir des points à 8 dimensions (voir Section 2.2.2 dans paragraphe : Méthodologie adoptée).

Après que les données sont récupérées, on construit le KDtree basé sur les points à N dimensions du maillage cible (voir ANNEXE C).

Remarque :

- Le maillage Source est colorié en bleu pour le différencier du maillage Cible.
- Si la donnée couleur est sélectionnée dans l'interface, elle ne servira qu'à filtrer, une fois les correspondances obtenues.
- Les points Source à N dimensions sont utilisés pour la recherche du point le plus proche.

KD-tree

Pour trouver le point le plus proche, j'ai utilisé une distance Euclidienne adaptée pour le recalage.

DISTANCE EUCLIDIENNE : :

ENTRÉE : Deux points $P(x, y, z, n_x, n_y, n_z, k_{min}, k_{max})$ et $P'(x', y', z', n'_x, n'_y, n'_z, k'_{min}, k'_{max})$ à 8 dimensions (au maximum), deux tableaux α et β .

SORTIE : Une distance au carrée entre ces deux points.

$$d^2(S, T) = (1 + \beta_{Normale} + \beta_{Courbure}) * (\alpha_1 * [(x - x')^2 + (y - y')^2 + (z - z')^2]) \\ + \beta_{Normale} * (\alpha_2 * [(n_x - n'_x)^2 + (n_y - n'_y)^2 + (n_z - n'_z)^2]) \\ + \beta_{Courbure} * (\alpha_3 * (k_{min} - k'_{min})^2 + \alpha_4 * (k_{max} - k'_{max})^2).$$

TABLE 3.3 – Distance euclidienne du KD-Tree.

- où $\beta_{Normale}$ et $\beta_{Courbure}$ sont les poids donnés par l'utilisateur dans l'interface (voir ANNEXE B).
- où $\alpha_1 = 1/Max$: où Max est le maximum des différences $|max - min|$ de chaque coordonnée des points cibles ; α_2 pour les normales. Pour les courbures principales, α_3 et α_4 sont égaux à $1/|max - min|$ où min est le minimum des valeurs pour une courbure principale des points cibles et de même pour max . Ils servent à normaliser la contribution de la normale, de la courbure par rapport aux coordonnées euclidiennes.

Avec cette formule, on donne toujours plus d'importance aux coordonnées des points qu'à leurs normales et à leurs courbures.

Calcul de la transformation aux moindres carrés pondérés

A partir des correspondances trouvées, la transformation aux moindres carrés permet de trouver la matrice (4x4) à appliquer sur le maillage Source pour modifier ces points afin de se rapprocher du maillage Cible. Cependant, si on a des appariements incorrects, la transformation va en tenir compte et sera faussée. Elle peut rétrécir le maillage Source. Par conséquent, ce maillage se déforme tout en augmentant leur dissemblance.

D'après l'article de Feldmar et Ayache (1996) [4], la transformation du moindre carré pondéré permet de donner un coefficient de fiabilité à chaque appariement et de prendre en compte ce paramètre dans le calcul de la transformation. Ainsi, les auteurs évitent le rétrécissement en donnant plus d'importance à certaine correspondance. Ils choisissent les valeurs de courbure comme poids pour les correspondances.

Dans l'ICP non-déformable, le poids que l'on affecte aux correspondances est en fonction de la distance entre les deux points qu'ils contiennent et de leurs couleurs (VOIR TABLE 3.4).

3.2.2 ICP non-déformable

Pour chaque mode de transformation de cet ICP, on va appliquer l'algorithme ci-dessous (voir TABLE 3.4). Seule la transformation du moindre carré pondéré va changer selon le mode dans lequel on se trouve.

Cependant, lorsque l'on utilise seulement les landmarks, l'étape 2 de l'algorithme varie. En effet, on ne filtre pas les correspondances des landmarks. De plus, les poids des landmarks sont identiques et égales à 1 puisqu'ils sont tous essentiels pour le calcul de la transformation aux moindres carrés pondérés.

3.2.3 ICP déformable

Pour le mode Localement Affine, j'ai utilisé l'algorithme suivant (voir TABLE 3.5).

Pour le prétraitement, on utilise un deuxième KD-tree basé sur les points 3D du maillage Source pour créer les ensembles de $Sphere(M_k)$. J'ai dû modifier la recherche du point le plus proche pour trouver les ensembles de points dans la surface Source définis par un rayon R (voir ANNEXE C).

Remarque :

Le voisinage n'est calculé qu'une seule fois en tout début puisque les maillages Source et Cible étant proche grâce à la transformation Similitude et Affine, on considère que les distances des points d'un voisinage ne varient pas assez pour le considérer hors du voisinage dans un certain rayon.

3.2.4 Fonctions de visualisation des résultats

À la fin du recalage, j'ai ajouté quelques fonctionnalités pour permettre à l'utilisateur de voir les différents résultats et de les sauvegarder :

- La distance moyenne X d'erreur de chaque itération du recalage : sous forme de tableau et sous forme de graphique. Ces résultats peuvent être enregistrés.
- Le nombre d'itérations et le temps de calcul pour chaque mode : sous forme de tableau pour donner une indication sur l'exécution du recalage.

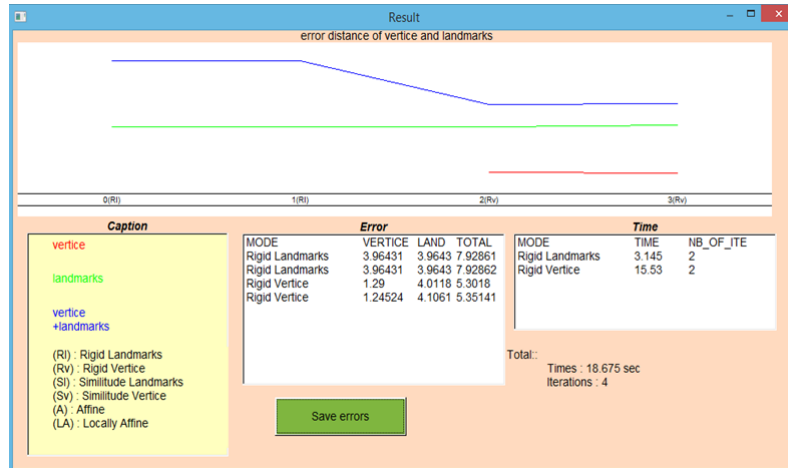


FIGURE 3.8 – Exemple de résultats d’affichage

- Les paramètres de chaque mode peuvent être enregistrés.
- Les correspondances des landmarks et des points peuvent être affichées. Par défaut, les paires de landmarks sont en rouge et celles des points en bleu.

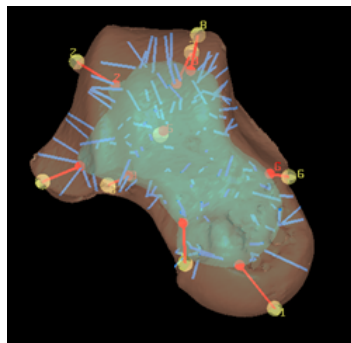


FIGURE 3.9 – Exemple d’affichage des correspondances des landmarks et des points

- Les paires des correspondances des points peuvent être aussi enregistrées si l'utilisateur le souhaite.

L'utilisateur peut aussi voir et enregistrer les paires de correspondance entre deux maillages sans pour autant faire le recalage. Par exemple, pour un maillage source déformé enregistré et son maillage cible, on regarde seulement les correspondances des $(P_{Source}, P_{Cible_le_plus_proche})$ hors recalage.

Remarque :

Pour l’affichage ou pour l’enregistrement, le nombre de paires de correspondances des points est défini par l'utilisateur dans l'interface.

ALGORITHME ICP NON-DÉFORMABLE : :

ENTRÉE : surface Cible et surface Source

SORTIE : surface Source modifiée

répète : À chaque itération,

étape 1 : Création des paires de correspondances (P_{Source}, P_{Cible}) .

Pour chaque point P_{Source} (N dimensions) :

- Calcul de son point le plus proche P_{Cible} (N dimensions) en utilisant le Kd-tree de la cible.

étape 2 : Filtrage en ajoutant des poids w_j aux correspondances trouvées.

Pour chaque paire de correspondance $(P_{Source}, P_{Cible})_j$:

- $w_j = 0$.
- Si P_{Source} et P_{Cible} ont le même indice de couleur alors $w_j += \beta_{Couleur}$ (où $\beta_{Couleur}$ est défini dans l'interface)
- Si $|P_{Source} - P_{Cible}| < \epsilon$ alors $w_j += 1$.
- Si $w_j = 0$, alors $(P_{Source}, P_{Cible})_j$ supprimé.

étape 3 : Déformation de la surface Source.

- Calcul de la transformation aux moindres carrés pondérés des correspondances filtrées.
- Application de la transformation calculée sur le maillage source et les points de repères.
- Mise à jour des éléments du maillage Source.(voir annexe)

étape 4 : Calcul la distance moyenne X d'erreur :
$$\frac{\sum_{j=0}^{Nb_{Correspondances}-1} |newP_{Source,j} - P_{Cible,j}|}{Nb_{Correspondances}}$$

jusqu'à : Si $X > \text{seuil}$ OU Si le nombre maximal d'itérations est atteint
alors arrêt.

TABLE 3.4 – Algorithme ICP non-déformable du recalage.

ALGORITHME ICP DÉFORMABLE : :

ENTRÉE : surface Cible et surface Source, rayon R

SORTIE : surface Source modifiée

PRÉTRAITEMENT : : Pour chaque $M_k \in$ Source :

- Calcule un ensemble de points dans la surface Source défini par une sphère de centre M_k et de rayon R en utilisant le KD-tree.

répète : À chaque itération i,

étape 1 : Pour chaque point M_k de la Source, on calcule sa transformation rigide $(R_{k,i}, t_{k,i})$:

- On construit un ensemble des correspondances (M_l, N_l) , où $M_l \in Sphere(M_k)$ et N_l le point le plus proche de M_l dans la surface Cible.
- $(R_{k,i}, t_{k,i})$ est la transformation rigide qui minimise la distance carrée entre les correspondances (M_l, N_l) .

étape 2 : On lisse les déplacements rigides $(R_{k,i}, t_{k,i})$ pour calculer la déformation affine au point M_k :

$$\sum_{M_l \in Sphere(M_k)} \beta(M_l, M_k) (R_{l,i} \cdot M_k + t_{l,i})$$

où

$$\beta(M, M_k) = \frac{\alpha(M, M_k)}{Somme}$$

$$\alpha(M, M_k) = 1 - \frac{\| \vec{M, M_k} \|_2}{Diamètre}$$

$$Somme = \sum_{M \in Sphere(M_k)} \alpha(M, M_k)$$

étape 3 : Calcule la distance moyenne X d'erreur : $\frac{\sum_{k=0}^{NbSource-1} |newM_k - N_k|}{NbSource}$

jusqu'à : Si X > seuil OU Si le nombre maximal d'itérations est atteint
alors arrêt.

TABLE 3.5 – L'algorithme de Itérative Closest Point Déformable de l'article [4].

Chapitre 4

Conclusion

4.1 Les Résultats obtenus

Les tests suivants ont été réalisés sur une machine de core i7, de 2.00GHz et 8 Go de mémoire vive avec une carte graphique NVIDIA GEFORCE 840M.

4.1.1 L'affichage

Les résultats

Pour tester les deux affichages, j'ai manipulé les maillages à la souris.

D'après le tableau ci-dessous, on peut voir que l'utilisation du VBO améliore nettement la fluidité de l'affichage.

Maillages	Avec fonction "immédiate" (en FPS)	Avec VBO (en FPS)
<i>Talus de Lemur catta</i> contenant 258.663 points	8 - 9	66
Crâne de <i>Leptadapis</i> avec ses oreilles internes contenant 1.123.735 points	1 - 4	66

TABLE 4.1 – Résultat de l'affichage en frame per second (FPS)

Les limites

La version d'OpenGL doit supérieure ou égale à 1.4 pour l'utilisation du *vbo* (apparition de cette fonctionnalité). De plus, si le driver d'OpenGL n'est pas mis à jour, l'affichage risque de ne pas fonctionner sur les gros maillages. Je laisse la possibilité à l'utilisateur dans l'interface soit d'activer le VBO ou d'activer l'ancienne méthode d'affichage.

4.1.2 Le Recalage

4.1.2.1 Résultats des modes présentés

Les tests suivants ont été effectués avec les maillages de cavité endocrânienne.

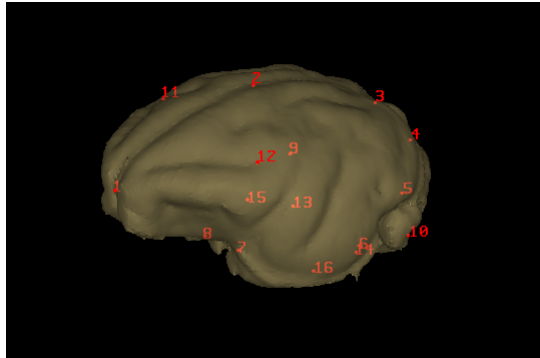


FIGURE 4.1 – Maillage de cavité endocrânienne d'*Eulemur fulvus* avec ses 16 landmarks contenant 86.410 points.

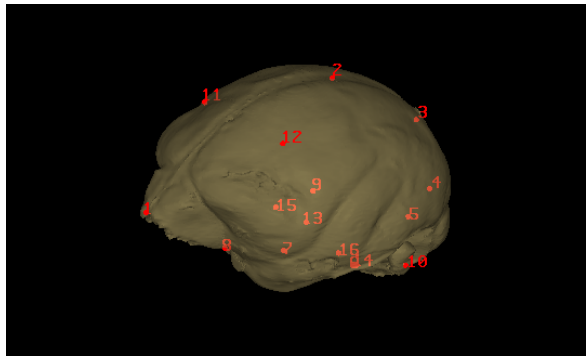


FIGURE 4.2 – Maillage de cavité endocrânienne d'*Arctocebus* avec ses 16 landmarks contenant 285.665 points.

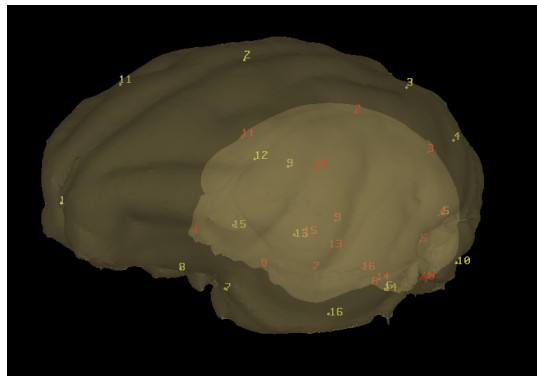


FIGURE 4.3 – Mise en place des deux maillages et de leurs landmarks. Les landmarks jaunes sont les landmarks cibles. Les landmarks rouges sont les landmarks sources.

Les paramètres de ces tests sont les suivants :

- 0.5 poids normale,
- 0.5 poids courbure,
- 2 est le diviseur du nombre de points source.
- 30% pour le filtre sur les distances des correspondances,
- 20 itérations maximum,
- 5% d'erreur entre l'erreur précédente et l'erreur actuelle,
- 5% du diamètre Source pour le mode Localement Affine.

Pour plus de détails sur les paramètres voir ANNEXE B : Interface.

Le mode Rigide :

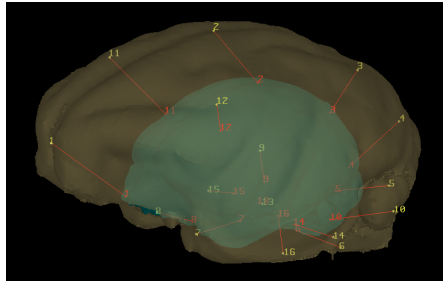


FIGURE 4.4 – Mode Rigide des deux cavités endocrâniennes présentées : en bleu le maillage source *Arctocebus* et en jaune le maillage cible *Eulemur fulvus*.

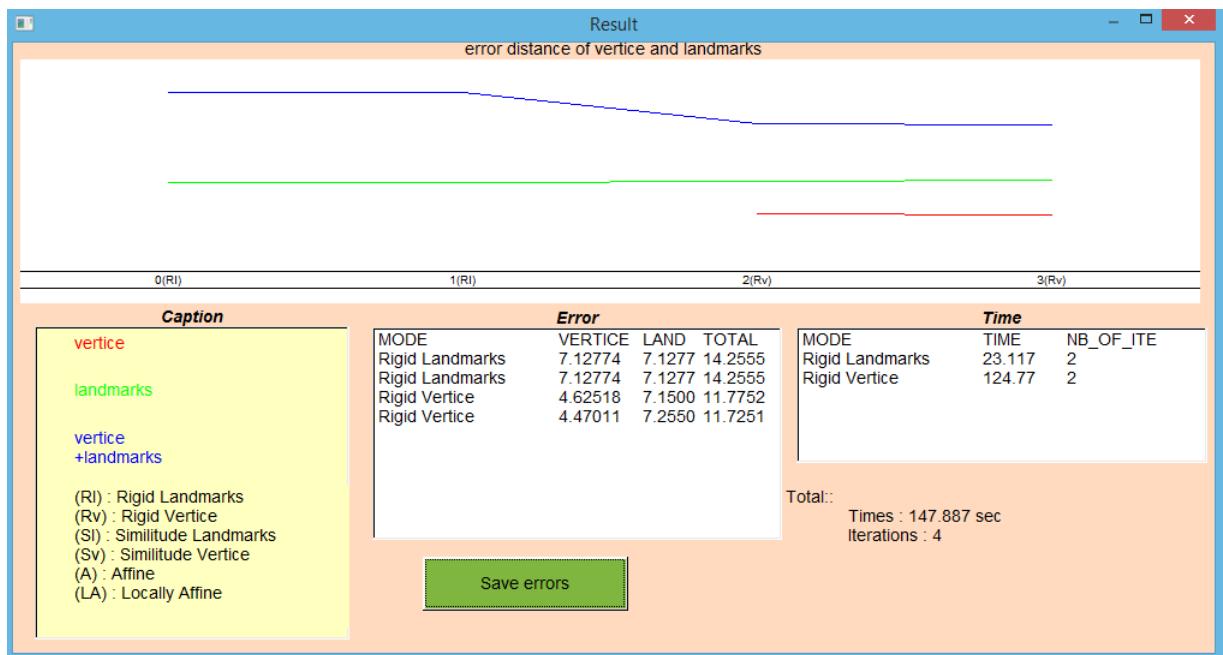


FIGURE 4.5 – Résultats obtenus en mode Rigide

Le mode Similitude :

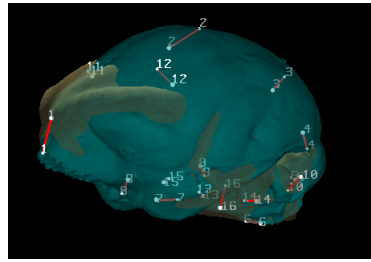


FIGURE 4.6 – Mode Similitude des deux cavités endocrâniennes présentées : en bleu le maillage source *Arctocebus* et en jaune le maillage cible *Eulemur fulvus*.

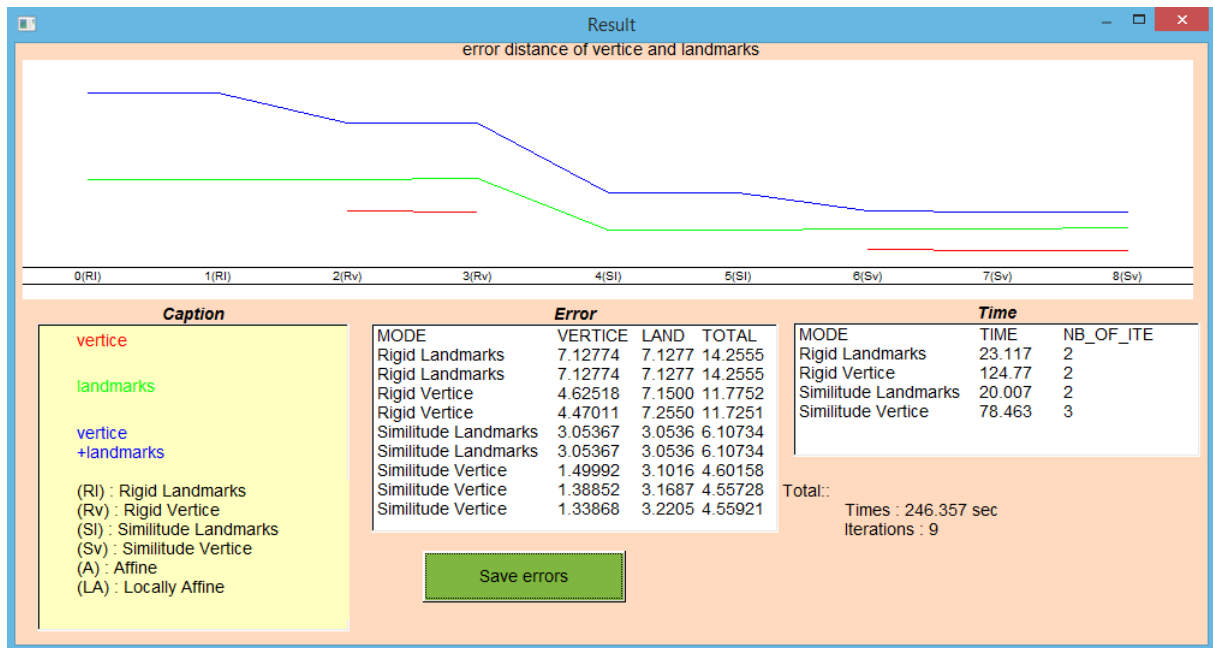


FIGURE 4.7 – Résultats obtenus en mode Similitude

Le mode Affine :

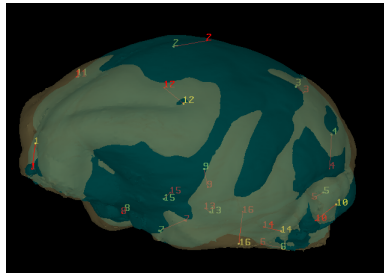


FIGURE 4.8 – Mode Affine des deux cavités endocrâniennes présentées : en bleu le maillage source *Arctocebus* et en jaune le maillage cible *Eulemur fulvus*.

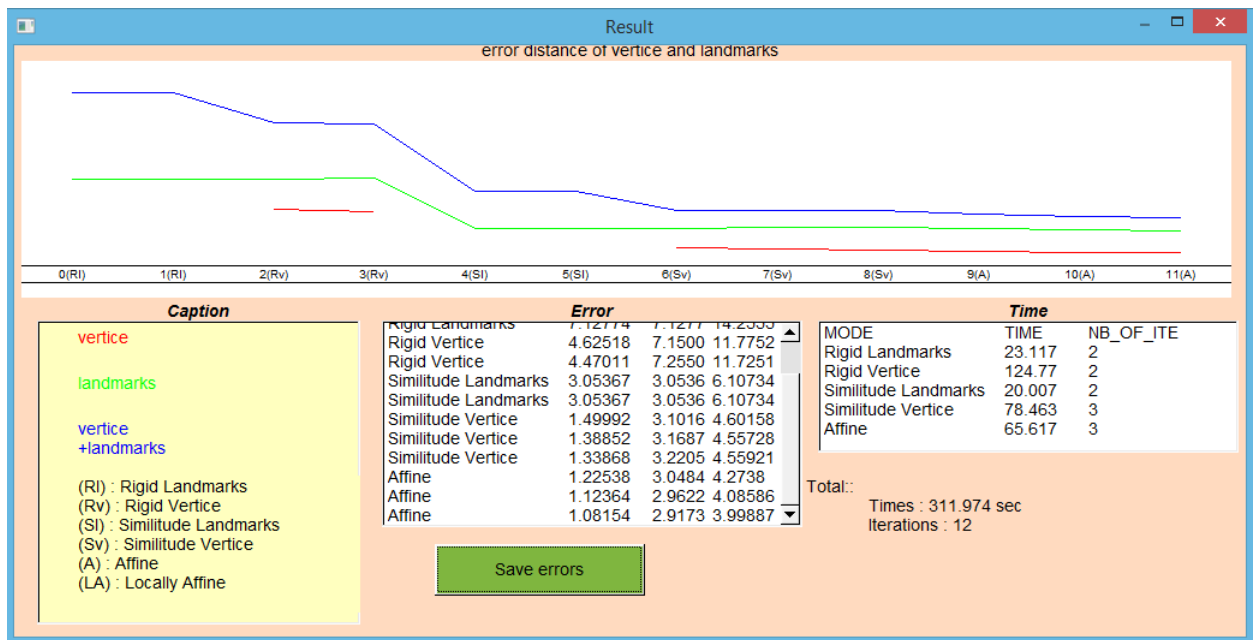


FIGURE 4.9 – Résultats obtenus en mode Affine

On constate que les distances des erreurs moyennes des points source diminuent de 7.12mm à 1.06mm. Les distances des erreurs moyennes des landmarks sont passées de 7.12mm à 3.22mm. Malgré cette diminution on constate que leurs distances augmentent légèrement lorsqu'ils ne sont pas utilisés.

Ceci est dû au fait que les landmarks sont posés manuellement sur les maillages.

Le mode Localement Affine avec un rayon de 5% de diamètre Source.

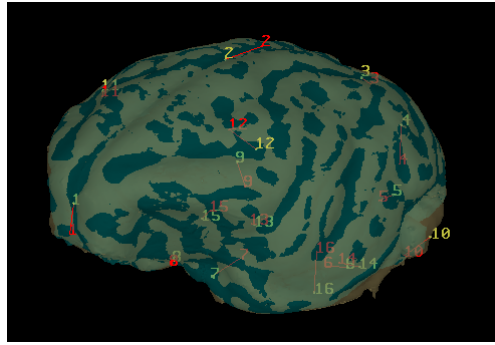


FIGURE 4.10 – Mode Localement Affine des deux cavités endocrâniennes présentées à 5% du diamètre Source : en bleu le maillage source *Arctocebus* et en jaune le maillage cible *Eulemur fulvus*.

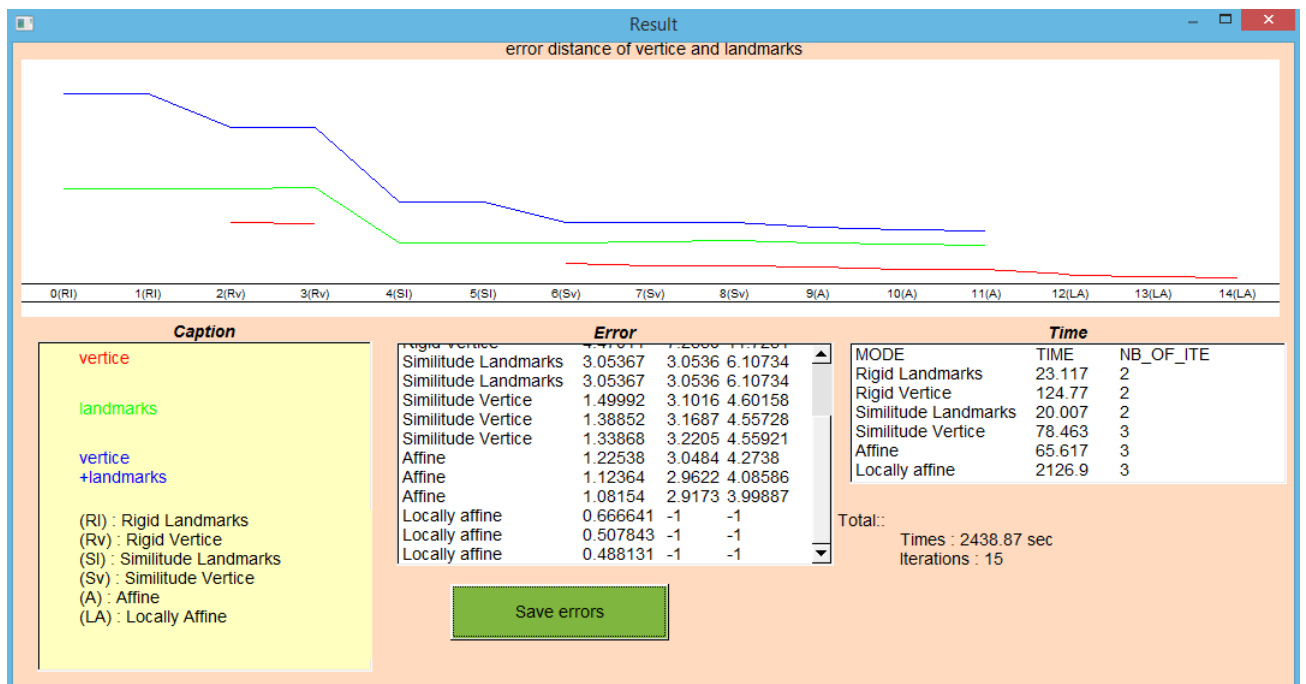


FIGURE 4.11 – Résultats obtenus en mode Localement Affine à 5%.

On constate bien un rapprochement du maillage source vers le maillage cible. Cependant, il n'est pas assez proche sur certaine zone. Pour apporter plus de précision, on a réitéré le mode Localement Affine en changeant le paramètre permettant de définir le rayon des sphères. Ce paramètre est mis à 2%.

Le mode Localement Affine avec un rayon de 2% de diamètre Source.

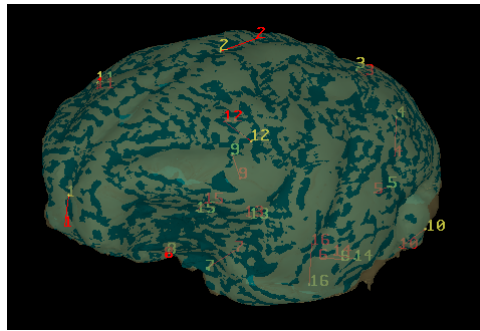


FIGURE 4.12 – Mode Localement Affine des deux cavités endocrâniennes présentées à 2% du diamètre Source : en bleu le maillage source *Arctocebus* et en jaune le maillage cible *Eulemur fulvus*.

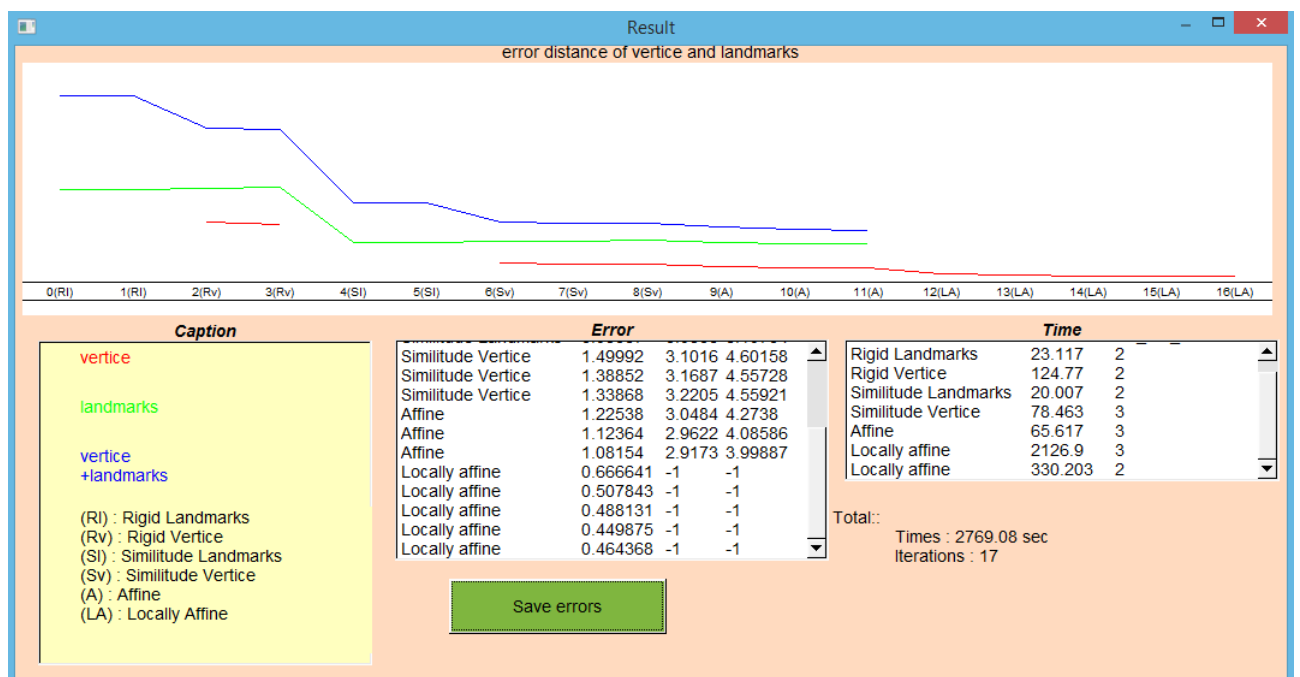


FIGURE 4.13 – Résultats obtenus en mode Localement Affine à 5% puis à 2%

Remarque :

Le temps de calcul pour le mode localement affine dépend surtout du nombre de points pris dans chaque ensemble ($Sphere(M_k)$) construit.

En effet, dans le premier cas à 5%, chaque point a un ensemble contenant entre 1.000 et 2.000 points. Ce calcul met au total environ 672 secondes, soit environ 11 minutes pour 285.665 points. Dans le deuxième cas à 2%, le calcul se fait en 110 secondes soit environ 1 minutes pour des ensembles contenant entre 150 et 300 points.

On voit un net rapprochement sur l'ensemble du maillage source déformé à 2%, contrairement au test à 5% pour un seuil de distance de 0.1mm (voir figure ci-dessous). De plus, on peut le constater graphiquement. On atteint finalement, une distance d'erreur de 0.46mm entre les deux maillages.

Ceci montre bien que le paramètre R dans l'algorithme ICP déformable (VOIR TABLE 3.5) est un paramètre majeur permettant d'accroître la précision de la déformation du

maillage Source et ainsi d'augmenter la ressemblance du maillage Source vers le maillage Cible.

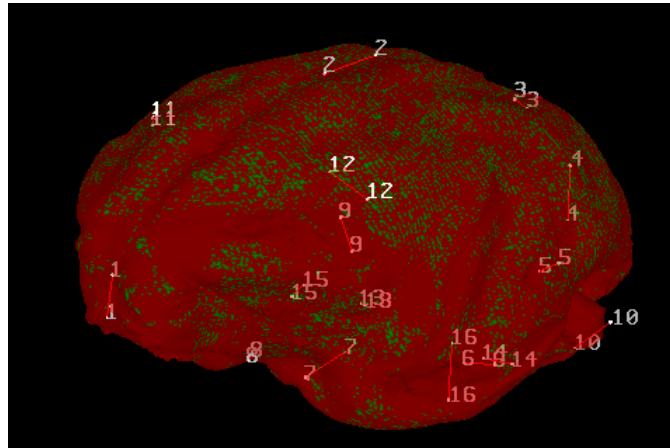


FIGURE 4.14 – Résultat du mode Localement Affine à 5% affichant en vert les points sources proche de son point cible le plus proche de 0.12mm de distance.

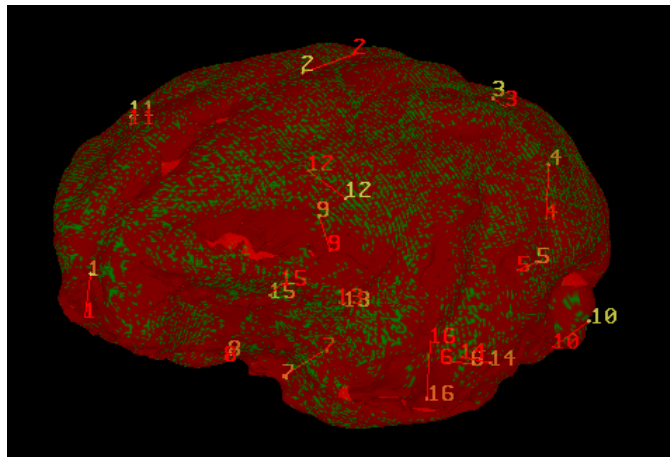


FIGURE 4.15 – Résultat du mode Localement Affine à 5% puis à 2% affichant en vert les points sources proche de son point cible le plus proche de 0.12mm de distance.

La figure ci-dessous montre bien la progression de la déformation du maillage Source vers le maillage Cible et sa ressemblance avec ce dernier.

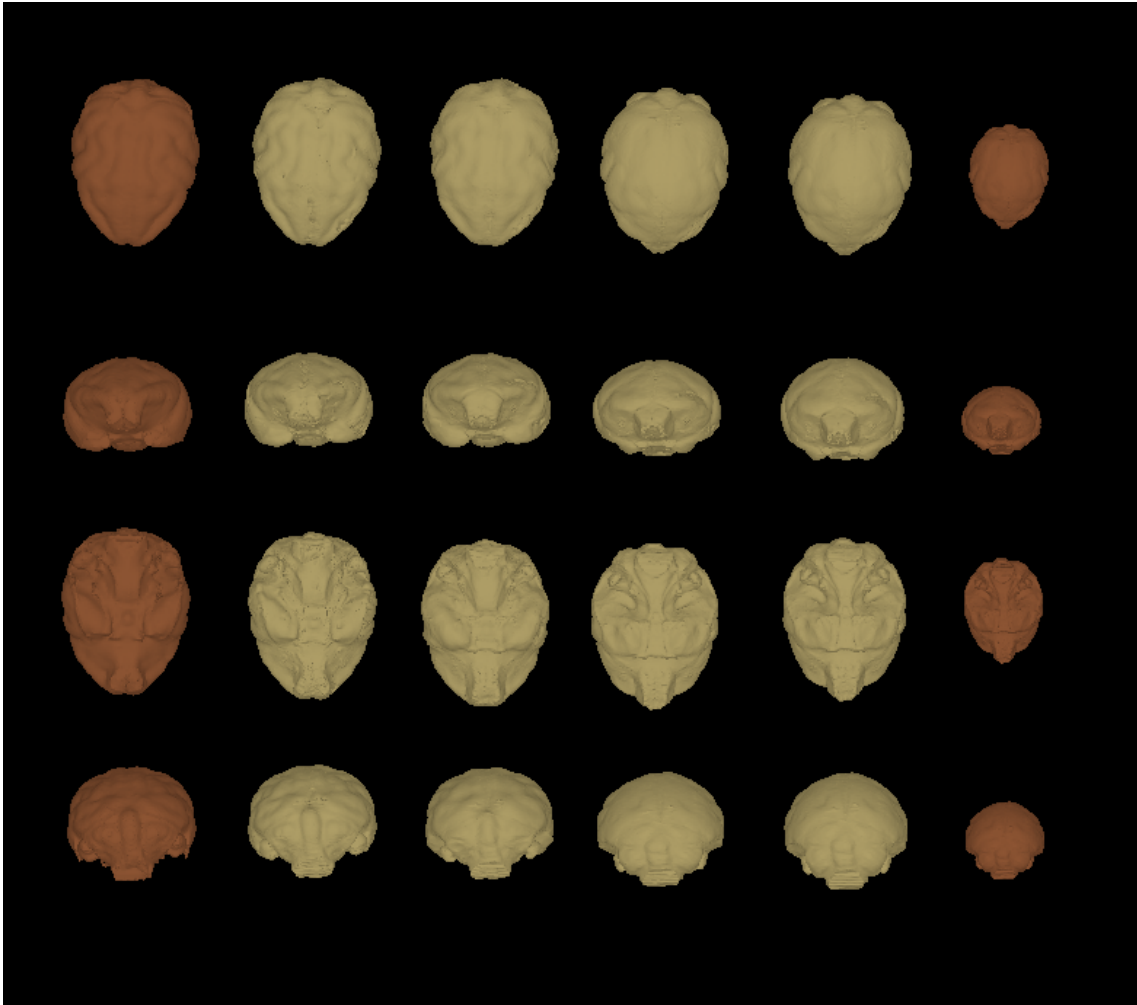


FIGURE 4.16 – Comparaison entre les formes obtenues (de gauche à droite) : maillage Cible d'*Eulemur* (marron), Maillage Source d'*Arctocebus* déformé en mode Localement Affine à 2% (jaune), Source déformé en mode Localement Affine à 5%(jaune), Source déformé en mode Affine (jaune), Source déformé au mode Similitude (jaune), maillage Source initial (marron).

Résultats avec les *Tali*

Avec les mêmes paramètres et procédés, on obtient de bon résultat sur les maillages de *tali* ayant 9 landmarks chacun. Le maillage source est l'*Adapis* avec 130.466 points et 260.924 facettes, et la cible le *Lemur catta* avec 258.663 points et 517.346 facettes.

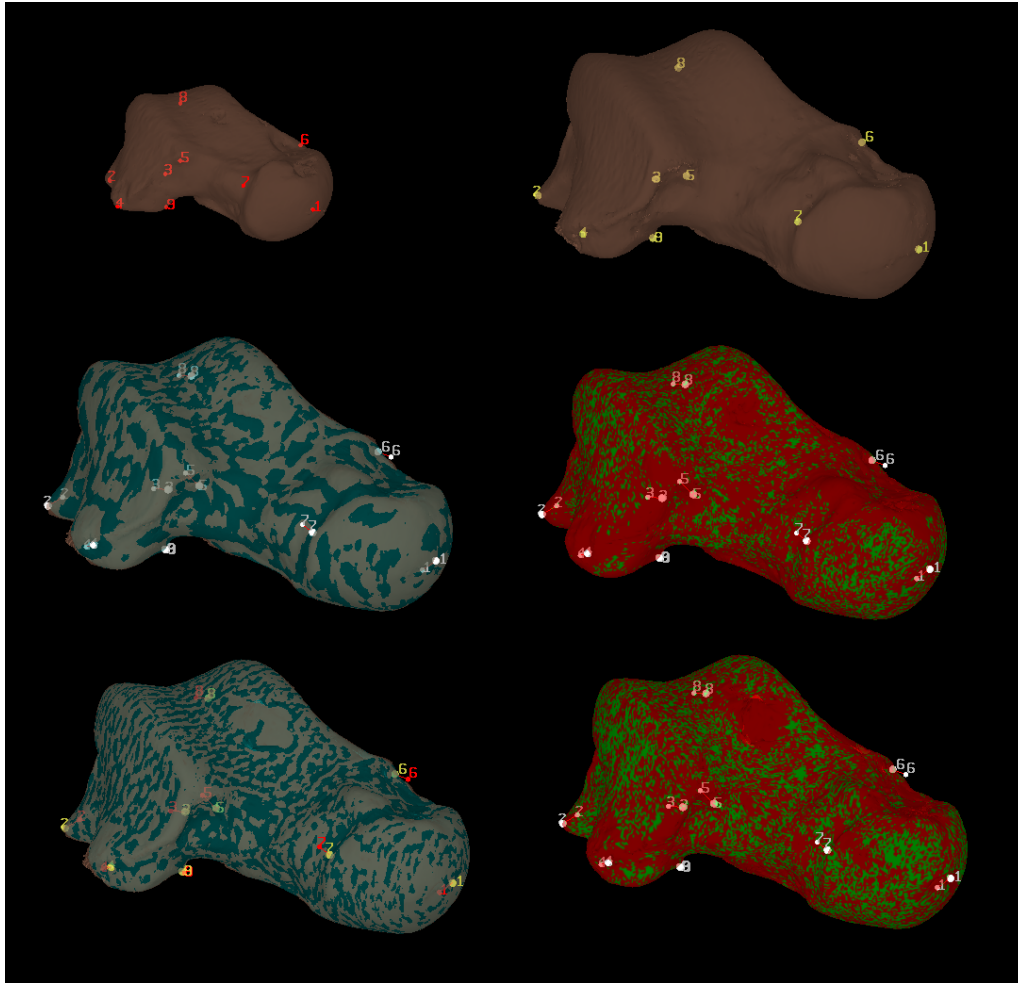


FIGURE 4.17 – Les deux maillages en haut sont le maillage Source (gauche) puis le maillage cible (droite). Au milieu, on a le recalage en mode Localement Affine à 5% (gauche) et la distance entre les deux maillages à un seuil de 0.05mm (droite). En bas, on a le recalage en mode Localement Affine (gauche) à 5% puis à 2% et la distance entre les deux maillages à un seuil de 0.05mm (droite).

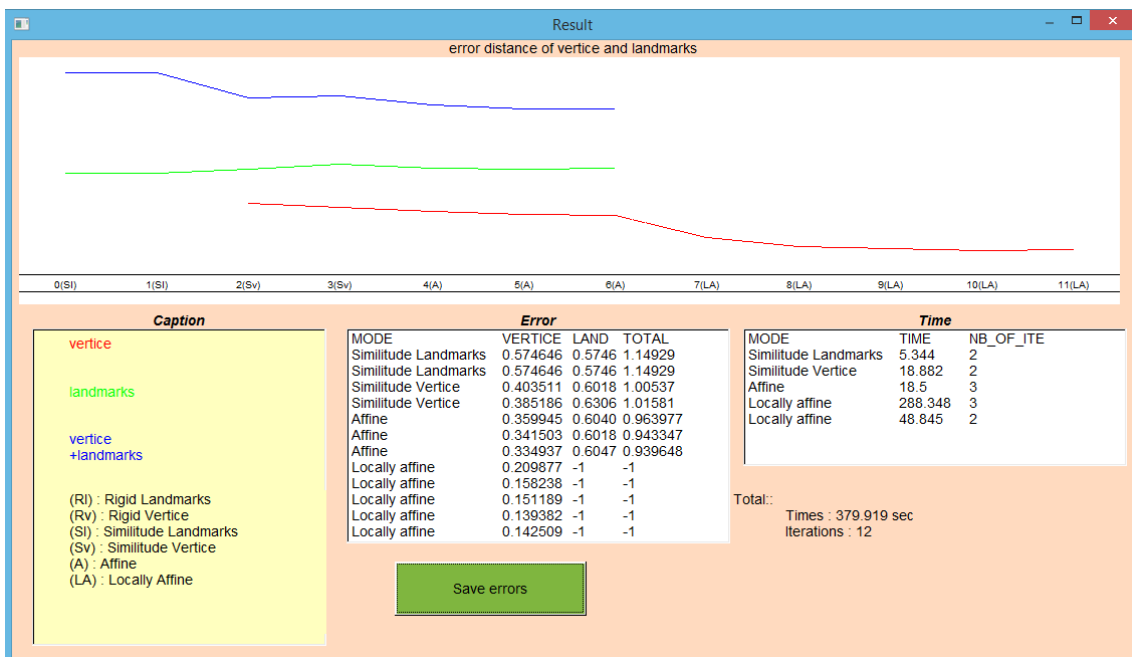


FIGURE 4.18 – Résultats obtenus en mode Localement Affine à 5% puis à 2% pour les maillages *tali*.

4.1.2.2 Cas particulier du maillage de la molaire

Chez les mammifères, parmi les restes fossiles, les dents sont les éléments les plus fréquemment préservés. Les dents sont ainsi probablement les restes fossiles les plus étudiés, et de nombreuses hypothèses scientifiques sont construites sur la base de comparaisons de morphologies dentaires. Elles donnent de nombreuses informations (âge, mode d'alimentation).

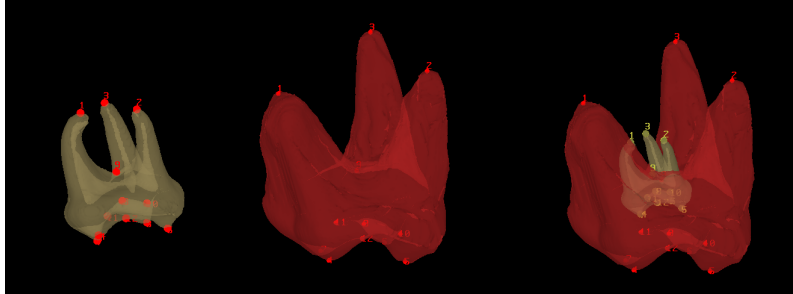


FIGURE 4.19 – Maillage de Molaire supérieur de *Planer* et de *Leptadapis* avec leurs 12 landmarks. Le maillage source est le premier maillage avec ses 41.431 points. Le maillage cible est le deuxième maillage avec ses 102.404 points.

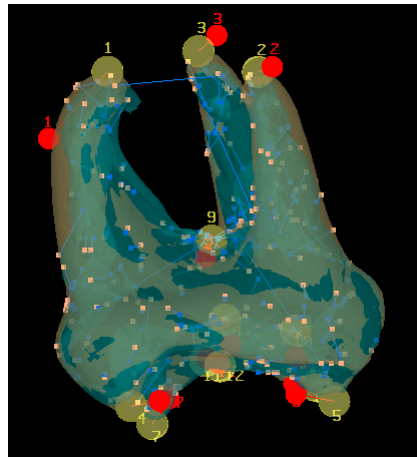


FIGURE 4.20 – Résultat obtenu des maillages de molaires avec quelques correspondances, soit 205 correspondances, au mode Localement Affine à 5%.



FIGURE 4.21 – Résultat obtenu des maillages de molaires avec quelques correspondances, soit 205 correspondances, au mode Localement Affine à 5% en utilisant le paramètre qui enlève les valeurs extrêmes mis à 40%.

Les deux maillages de molaires présentés ci-dessus contiennent 3 racines. L'une des deux a une surface à l'intérieur de son maillage.

Les paramètres évoqués précédemment sont les mêmes pour ce test.

Dans un premier temps, on fait le recalage avec un paramètre *Percentage of sphere diameter* de 5%. Le résultat obtenu n'est pas correct. En effet, on a des correspondances allant d'une racine Source à une autre racine opposée Cible (voir FIGURE 4.18).

Pour arranger le problème, on utilise le paramètre qui enlève les valeurs extrêmes des courbures principales (VOIR ANNEXE B). En effet, des petites valeurs de la courbure sur le maillage Cible sont isolées à certain endroit et faussent la recherche du point le plus proche.

On obtient des correspondances correctes puisque les points source sur la pointe de la racine vont bien vers les points cible de la pointe de la racine cible.

Cependant, le temps de calcul pour la recherche du point le plus proche augmente fortement. Il passe de 35 secondes pour le recalage sans le paramètre à 5 minutes pour le recalage avec le paramètre.

Variation de la dimension des points

En faisant varier la dimension des points, on remarque que les normales sont importantes. En effet, comme on a une surface interne dans le maillage cible, la normale évite de prendre les points de la surface interne dans les correspondances. Cependant, ce n'est pas le cas pour les points ayant 3 dimensions (x, y, z) et à 5 dimensions $(x, y, z, k_{max}, k_{min})$.

4.1.2.3 Comparaison des appariements automatiques et manuels.

On a voulu comparer les landmarks pris par l'expert et ceux créés par la méthode de recalage.

Un échantillon de 7 endocrânes de primates actuels (comprenant 3 lémuriens de Madagascar, 3 loriformes et 1 anthropoïde) a été constitué et comparé. Un expert a placé 16 landmarks homologues sur chacun de ces moulages endocrâniens. Les méthodes de recalage développées ont été utilisées de manière à extraire 572 semi-landmarks sur chacun des endocrânes des individus de cet échantillon.

Ces deux jeux de landmarks (16 vs. 572) ont été analysés à l'aide des techniques de géométrie morphométrique. Les configurations de landmarks ont été alignées par surposition Procrustes, et les résidus Procrustes ont été injectés dans une analyse en composantes principales, de manière à analyser les grandes tendances de variation de forme au sein de cet échantillon.

Les étapes de superposition Procrustes et l'ACP ont été calculées à l'aide du logiciel MorphoTools (FIGURE 4.22), développé par Mr Renaud Lebrun et Mr Mathias Specht.

Les résultats de l'ACP portant sur 16 landmarks (FIGURE 4.23) montrent une faible discrimination des lémuriformes, loriformes, et simiiformes. En revanche, le pouvoir discriminant de l'analyse portant sur les 572 semi-landmarks est bien meilleur. Les endocrânes apparentant à des lémuriens et ceux des loriformes se projettent en effet tous sur des valeurs positives de l'axe 1, tandis que l'endocrâne de macaque se projette à une valeur négative de l'axe 1. L'axe 2 discrimine quant à lui très nettement les lémuriens des loriformes.

Ce premier test souligne l'intérêt et confirme l'utilité des méthodes de recalage développées dans le cadre de ce stage pour conduire des analyses comparatives précises portant sur la morphologie de structures pauvres en landmarks.

On débute que la méthode de recalage permet de générer des nombreux landmarks et précises sur les maillages.

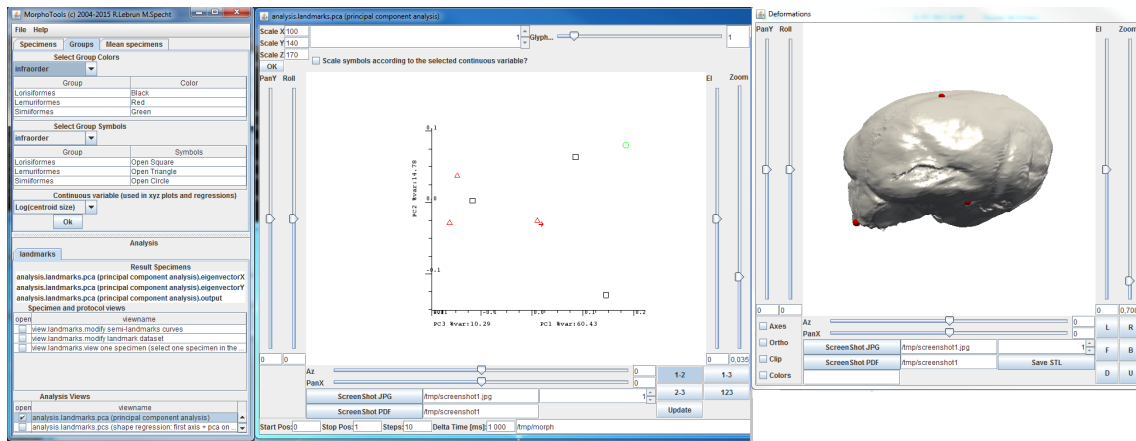


FIGURE 4.22 – logiciel MorphoTools avec ses trois principales fenêtres : les espèces à comparer (gauche), l’analyse en composantes principales (milieu), la maillages déformable en fonction des modifications des paramètres (droite).

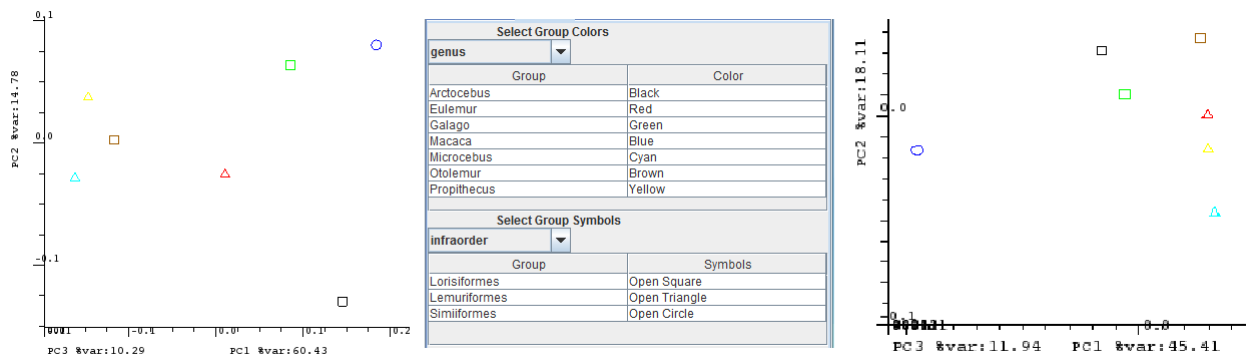


FIGURE 4.23 – ACP portant sur le protocole impliquant 16 landmarks (gauche). La légende de couleur et de forme définissant les espèces et leur appartenance. ACP relative au protocole impliquant 572 semi-landmarks. (droite).

4.1.2.4 Bilan

Les limites

L’emploi des méthodes de recalage présentées pour comparer des structures anatomiques n’est envisageable que dans la mesure où leur architecture ne diffère pas sensiblement. Par exemple, il ne serait pas possible de comparer directement les morphologies globales de crânes d’anthropoïdes et de lémuriniens. Ceux-ci diffèrent en effet de manière sensible dans la structure de leur orbite (présence d’une cloison post-orbitaire chez les anthropoïdes, et d’une simple barre post-orbitaire chez les lémuriniens) et dans leur formule dentaire (présence de 2 prémolaires chez la plupart des anthropoïdes, et de 3 prémolaires chez la majorité des lémuriniens actuels) (voir FIGURE 4.24).

Les améliorations

En plus d’utiliser la couleur des points d’une correspondance et leur distance, on pourrait augmenter les poids des correspondances en utilisant les courbures des points. Par exemple, la moyenne des courbures des points dans une correspondance ou en prenant le maximum entre leurs courbures principales.

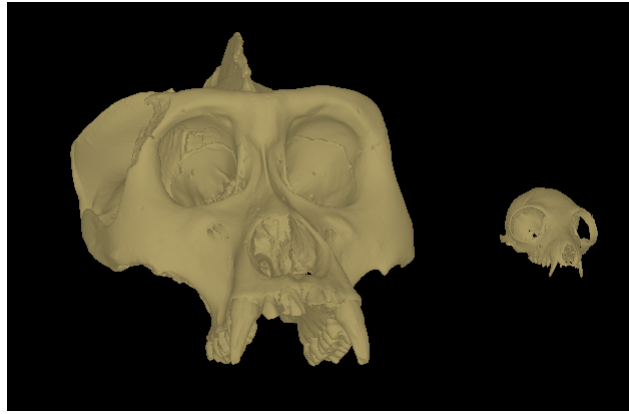


FIGURE 4.24 – Crâne de gorille (primate anthropoïde) à gauche, et d'*Eulemur mongoz* (lémurien), à droite.

Les perspectives

On pourrait calculer la normale et les courbures directement après la transformation au lieu d'utiliser la fonction de VTK qui refait le calcul sur le maillage (VOIR l'article [4]).

4.2 Difficultés rencontrées

J'ai eu quelques difficultés durant ce stage, notamment :

- la mise en place du VBO, plus exactement trouver toutes les informations qui peuvent modifier le maillage comme sa couleur, sa position, son apparence.
- me familiariser avec la bibliothèque vtk et la structure du logiciel.

4.3 Apports

Ce stage a permis d'ajouter un confort visuel pour le logiciel ISE-MeshTools en mettant en place le VBO.

La méthode de recalage va donner plus de possibilité de comparaison entre deux maillages. Elle peut aussi permettre de nouvelles études sur la morphologie entre les espèces.

Pour ma part, il m'a permis de mettre en pratique les connaissances acquises lors de ma formation comme implémenter l'algorithme de l'ICP ainsi que le VBO dans une application concrète.

J'ai pu aussi découvrir le milieu de la paléontologie et de l'anatomie comparative.

Chapitre 5

ANNEXE

5.1 Annexe A : Les éléments des maillages

Un maillage S contient :

- liste de facettes,
- liste de points 3D (x, y, z) ,
- liste des normales des points (n_x, n_y, n_z) ,
- centre,
- rayon,
- listes des 2 courbures des points,
- et éventuellement une liste des couleurs des points.

Le centre du maillage est le barycentre de ces points.

Le rayon est la distance la plus grande entre son centre et un de ces points :

$$\max_{P_i \in S} (|P_i - \text{centre}_S|).$$

La couleur d'un point est définie par un indice allant de 0 à 24. Ces couleurs correspondent aux "Tags" mis par l'utilisateur pour identifier des structures sur le maillage.

Le calcul de la normale et de la courbure se fait avec les fonctions de vtk : `vtkPolyDataNormals`, `vtkCurvatures`.

5.2 Annexe B : Interface de la fonction de recalage

5.2.1 Présentation de l'interface

L'interface se compose en deux parties :

- L'une contient les paramètres pour l'algorithme de recalage, utilisé avant de faire les ICP.
- Et l'autre contient les paramètres de visualisation et de sauvegarde des résultats, utilisé une fois que l'ICP a terminé.

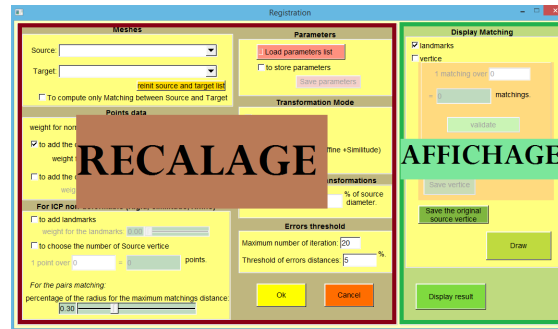


FIGURE 5.1 – Découpage de l'interface de la fonction de recalage

5.2.1.1 Les paramètres pour le recalage :

Après avoir chargé et sélectionné les deux maillages et leurs landmarks respectifs, on les place dans l'interface. Si la fenêtre est déjà ouverte, le bouton 'reinit' permet d'actualiser la liste des maillages chargés pour choisir la source et la cible dans MeshTools.

De plus, l'utilisateur peut faire soit le recalage soit seulement les appariements des points (Source,Cible). Les appariement des points permet d'avoir les correspondances (P_{Source}, P_{Cible}) sans faire la fonction de recalage.

Les données du maillage :

- le poids de la normale (sa valeur entre 0 et 1),
- le poids de la courbure (sa valeur entre 0 et 1),
- le poids de la couleur (sa valeur entre 0 et 1).

Les poids de la normale et de la courbure vont servir pour la recherche du point le plus proche dans le KDtree (voir ANNEXE C). Le poids de la couleur va servir pour filtrer les correspondances à chaque étape.

De plus, les données courbures ont un paramètre supplémentaire qui permet de seuiliser les valeurs extrêmes selon le pourcentage choisi.

En effet, on seuille les $x\%$ de valeurs extrêmes. Pour chaque courbure principale, on trie les valeurs de manière croissante. Puis on enlève les $x/2\%$ des valeurs minimums : x_{min} (resp. max). On prend comme valeur min le plus grand des x_{min} (resp. max le plus petit des x_{max}). Toutes ces valeurs enlevées ont pour valeur min (resp. max).

Remarques :

1. Si les poids sont nuls, on aura que les coordonnées des points pour le KDtree (voir ANNEXE C).

2. La couleur étant une donnée externe que l'utilisateur a ajouté sur le maillage, si elle est prise en compte, on vérifie qu'elle existe et que les deux maillages ont le même nombre de Tags à l'initialisation.
3. Les courbures sont mises à jour seulement si on a un poids non-nul contrairement à la normale qui sert pour visualiser le maillage une fois le recalage terminé.

Les données pour l'ICP non-déformable :

— *to add landmarks*

Ajouter les landmarks dans ces transformations en même temps que les sommets. La valeur récupérée est un pourcentage sur le nombre des sommets utilisés pour ces transformations. Ceci permet de l'influencer, de donner une certaine importance à la présence des landmarks sur le maillage source dans ces transformations.

— *to choose the number of Source vertice*

Choisir le nombre de sommets : l'utilisateur va paramétrer le nombre de sommets utilisés dans l'algorithme de recalage en saisissant le diviseur d sur le nombre de points de l'objet. On va donc prendre : "1 point sur d du nombre de sommets". Le diviseur d va servir à avoir une certaine homogénéité des points récupérés sur le maillage et à diminuer le nombre de points à diminuer le temps de calcul.

— *percentage of the radius for the maximummatchings distance*

Choisir un seuil de filtre ϵ : ce seuil est un pourcentage sur un rayon défini par le plus petit rayon entre les maillages source et cible. Il permet de définir une distance maximale $= \epsilon\% \cdot \min(r_{source}, r_{cible})$. Lors de l'étape de mise en correspondance, tous les couples de points qui auront une distance supérieure seront exclus. Cela permet en particulier de recalculer un maillage complet avec un maillage partiel.

Remarque :

- Selon le pourcentage récupéré pour le paramètre "*to add landmarks*" peut faire varier les poids des sommets :
 1. Si ce pourcentage est égal à 1 alors dans le calcul des transformations, on n'utilisera que les landmarks.
 2. Si ce pourcentage est inférieur à 1 alors dans le calcul des transformations, on prendra en compte les landmarks et les sommets.
 3. Si ce pourcentage est égal à 0 alors on aura que les sommets dans le calcul des transformations.
- Pour le choix du nombre de sommets, par défaut :
si le nombre de points du maillage source est supérieure à 100.000 points alors le diviseur est :

$$dec = nbDePoints/100.000$$

sinon on prend tous les points du maillage source.

Ceci permet de garder un nombre raisonnable de points pour calculer les ICP non-déformables.

Les paramètres des modes :

Ils peuvent être sauvegardés ou être chargés depuis un fichier .txt. Si le fichier est chargé, alors la fonction de recalage ne se terminera que lorsque tous les modes du fichier auront été exécutés.

Dans ce fichier, à chaque mode, on a :

- un booléen pour indiquer si le mode a été traité (1) ou non (0).
- nom du mode,
- poids des normales,
- poids des courbures,
- poids des couleurs,
- poids de l'ajout des landmarks,
- diviseur du nombre des points source,
- pourcentage du rayon pour filtrer les correspondances,
- nombre d'itérations maximum,
- seuil d'erreur pour l'arrêt des itérations ICP,
- pourcentage du rayon de la sphere pour le Localement Affine,
- pourcentage pour modifier les valeurs extrêmes des courbures.

Mode de transformation :

L'utilisateur va pouvoir sélectionner le mode de transformation qu'il veut appliquer sur les deux modèles :

- Rigide,
- Similitude,
- Affine,
- Localement Affine,
- et enfin STEP BY STEP.

Remarque :

1. Le dernier mode permet de passer d'un mode à l'autre en utilisant la touche "Espace" ou le bouton "ok" de l'interface.
2. La touche "Espace" permet aussi de passer d'un mode à l'autre avec les mêmes données.

Pour l'ICP déformable :

L'utilisateur peut donner un pourcentage du rayon de la Source. Si pour un rayon donné, un ensemble ne contient pas de points, on arrête l'ICP déformable. Par défaut, ce paramètre est à 5% du rayon source.

Pour l'arrêt des itérations de l'ICP, on peut choisir :

- le nombre d'itération maximal. Par défaut il est à 20 itérations.
- le pourcentage des distances des erreurs ρ . Par défaut il est à 5%.

Ce pourcentage ρ sert à définir le seuil d'arrêt.

Chaque erreur de distance, soit des landmarks soit des sommets, sont calculées par la moyenne d'erreur :

$$X(i) = \frac{\sum_{j=0}^{N-1} d(S_j, T_j)}{N}$$

Où N est le nombre de correspondances. $d(S_j, T_j)$ est la distance d'une paire de points ou de landmarks correspondants.

Le critère d'arrêt est basé sur ces distances moyennes d'erreur X :

$$\frac{X(i-1) - X(i)}{X(i-1)} \geq \delta \text{ où } i \text{ est l'indice de l'itération de l'ICP pour un mode donnée.}$$

5.2.1.2 Les paramètres d'affichage

Une fois que l'ICP a terminé, on peut :

→ Afficher des correspondances :

On peut afficher les correspondances des landmarks et/ou des sommets, changer aussi la couleur de ces sommets, enregistrer les correspondances affichées et les sommets d'origine de la source. Par défaut, en bleu on a les correspondances des sommets et en rouge celles des landmarks.

→ Visualiser sur le maillage Source la distance entre les points Source et les points Cible des correspondances pour le mode Localement Affine. En vert, les distances inférieures au seuil choisi et en rouge celles supérieures au seuil choisi.

→ Voir aussi les autres résultats de l'ICP :

- la courbe et un tableau des distances des correspondances des landmarks et des sommets à chaque itération,
- enregistrer ces distances.
- les temps de calcul durant chaque mode et leur nombre d'itérations.

5.2.2 Gérer les modifications de l'interface

Si un des paramètres du recalage dans l'interface change mais pas son mode, on refait l'ICP de ce mode.

Cependant, si les données changées modifient la dimension des points, on refait le Kdtree basé sur la maillage Cible, avant de faire le recalage.

5.3 Annexe C : KD-Tree

Construction : ¹

Pour construire le KD-tree pour les points à D dimensions, je calcule le point médian en cherchant les coordonnées minimum et maximum pour chaque élément du point. Ceci me donne une complexité de $O(D \cdot n)$, n étant le nombre de points. Puis pour trier les points à N dimensions, j'ai utilisé le `qsort` de C++ de complexité $O(D \cdot n \cdot \log(n))$.

Finalement, la complexité de la construction du KD-tree est $O(D \cdot n \cdot \log(n))$.

Pour construire le KD-tree :

- On sélectionne l'axe basé sur la profondeur.
- On trie la liste de points avec `qsort`.
- On crée le noeud qui est le point médian calculé.
- On construit les sous-arbres gauche et droit.

Chaque feuille a au maximum un point.

Chaque point dans le Kdtree contient :

- une liste de ses coordonnées servant à construire l'arbre,
- et une liste contenant les informations du point comme son indice d'origine et éventuellement sa couleur.

Ses fonctions : ²

```
NODE :: RecherchePointLePlusProche (p : ref point ; d : profondeur ; dist : ref distance ;  
pcandidat : point)  
  
if(Estunefeuille)  
  Recherche dans la feuille :  $p_{candidat} = p_{feuille,i}$  tel que  $\min(Dist(p, p_{feuille,i}))$   
  //Où  $Dist()$  est la distance Euclidienne entre les points à  $D$  dimensions.  
else  
  if(  $p(axe_{noeud}) \leq center_{noeud}(axe_{noeud})$  )  
    if(  $p(axe_{noeud}) - dist \leq center_{noeud}(axe_{noeud})$  )  
      NoeudGauchenoeud.RecherchePointLePlusProche ( p, d+1, dist, pcandidat )  
    if(  $p(axe_{noeud}) + dist > center_{noeud}(axe_{noeud})$  )  
      NoeudDroitnoeud.RecherchePointLePlusProche ( p, d+1, dist, pcandidat )  
  else  
    if(  $p(axe_{noeud}) + dist > center_{noeud}(axe_{noeud})$  )  
      NoeudDroitnoeud.RecherchePointLePlusProche ( p, d+1, dist, pcandidat )  
    if(  $p(axe_{noeud}) - dist \leq center_{noeud}(axe_{noeud})$  )  
      NoeudGauchenoeud.RecherchePointLePlusProche ( p, d+1, dist, pcandidat )
```

TABLE 5.1 – Algorithme du point le plus proche de complexité $O(n \cdot \log(n))$ avec D constant.

1. référence : <http://gfx.cs.princeton.edu/proj/trimesh2/>

2. référence : http://andrewd.ces.clemson.edu/courses/cpsc805/references/nearest_search.pdf


```

NODE : : RechercheDesPointsDansUnRayon(p : ref point ; d : profondeur ; R : rayon ;
lp_candidat : liste des indices des points)

if( Estunefeuille )
    Recherche dans la feuille : Ajoute  $Indice\_p_{feuille,i}$  dans  $lp\_candidat$  tel que
 $Dist(p, p_{feuille,i}) \leq R$  //Où  $Dist()$  est la distance Euclidienne entre des points 3D.
else
    if(  $p(axe_{noeud}) \leq center_{noeud}(axe_{noeud})$  )
        if(  $p(axe_{noeud}) - R \leq center_{noeud}(axe_{noeud})$  )
            NoeudGauchenoeud.RechercheDesPointsDansUnRayon ( p, d+1, R, p_candidat )
        if(  $p(axe_{noeud}) + R > center_{noeud}(axe_{noeud})$  )
            NoeudDroitnoeud.RechercheDesPointsDansUnRayon ( p, d+1, R, p_candidat )
    else
        if(  $p(axe_{noeud}) + R > center_{noeud}(axe_{noeud})$  )
            NoeudDroitnoeud.RechercheDesPointsDansUnRayon ( p, d+1, R, p_candidat )
        if(  $p(axe_{noeud}) - R \leq center_{noeud}(axe_{noeud})$  )
            NoeudGauchenoeud.RechercheDesPointsDansUnRayon ( p, d+1, R, p_candidat )

```

TABLE 5.2 – Algorithme cherchant les points dans une sphère de center p et de rayon R de complexité $O(n \cdot \log(n))$.

5.4 Annexe D : Calcul de la Transformation

Pour faire la transformation de l'ICP, nous avons adapté la classe `vtkLandmarksTransform` afin de la modifier pour intégrer le poids dans le calcul. Cette transformation sert à calculer la matrice de la transformation du moindre carré pondéré pour les correspondances trouvées et pondérées.

Fonction du moindre carré pondéré :

$$S(A, t) = \sum w_i \cdot \|p_i - A \cdot q_i - t\|^2$$

Où p_i est un point cible et q_i un point source. A est une matrice de rotation et t un vecteur de translation. La norme $\|\cdot\|$ est la norme euclidienne.

Où w_i est un poids pour la correspondance associée.

Afin d'obtenir la transformation du moindre carré pondéré, les équations suivantes m'ont permis de placer correctement les poids w_i dans la transformation du moindre carré de `vtkLandmarkTransform`.

Soit $(P_{1,i}, P_{2,i})$ où $P_{1,i}(x_{1,i}, y_{1,i})$ et $P_{2,i}(x_{2,i} - b \cdot y_{2,i})$

$$S(a, b, c, d, t_x, t_y) = \sum w_i \cdot (x_{2,i} - a \cdot x_{1,i} - b \cdot y_{1,i} - t_x)^2 + (y_{2,i} - c \cdot x_{1,i} - d \cdot y_{1,i} - t_y)^2$$

On calcule les dérivées partielles par rapport aux paramètres a, b, c qui doivent être égales à 0 pour le minimum :

$$0 = \sum w_i \cdot x_{1,i} \cdot (x_{2,i} - a \cdot x_{1,i} - b \cdot y_{1,i} - t_x)$$

$$0 = \sum w_i \cdot y_{1,i} \cdot (x_{2,i} - a \cdot x_{1,i} - b \cdot y_{1,i} - t_x)$$

$$0 = \sum w_i \cdot x_{1,i} \cdot (y_{2,i} - c \cdot x_{1,i} - d \cdot y_{1,i} - t_y)$$

$$0 = \sum w_i \cdot y_{1,i} \cdot (y_{2,i} - c \cdot x_{1,i} - d \cdot y_{1,i} - t_y)$$

$$0 = \sum w_i \cdot (x_{2,i} - a \cdot x_{1,i} - b \cdot y_{1,i} - t_x)$$

$$0 = \sum w_i \cdot (y_{2,i} - c \cdot x_{1,i} - d \cdot y_{1,i} - t_y)$$

5.5 Annexe E : Structure de MeshTools

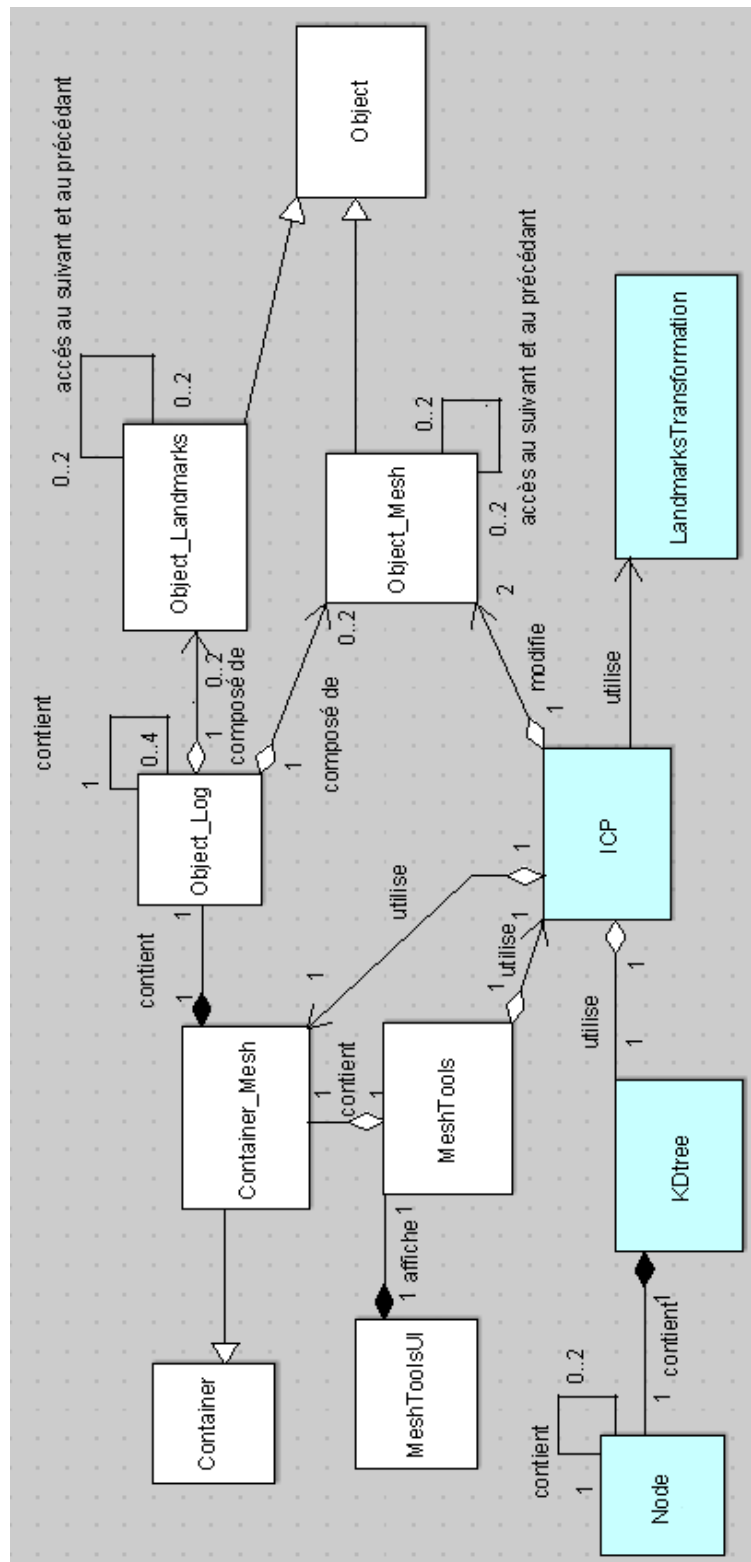


FIGURE 5.2 – Diagramme de classes de ISE-MeshTools : les classes en blanc sont du logiciel d'origine et en vert sont celles que j'ai rajoutées

Glossaire

landmarks sont des points que l'utilisateur a placé manuellement sur les deux modèles et qui correspondent à des identités de structures connues sur des objets biologiques; on les nomme également points « homologues », point de repère. 8, 9

paléoanthropologue spécialiste de l'étude des caractères anatomiques et biologiques de l'homme fossile. 8

paléobiodiversité étude des caractères divers des espèces animales et végétales fossilisées. 6

paléobiogéographie biogéographie des espèces fossiles. 6

paléobiologie étude biologique des organismes fossiles. 7

paléoécologie étude des organismes fossiles dans leur milieu. 7

phylogénie science étudiant la formation et le développement des espèces animales et végétales. 6

vascularisation densité du réseau des vaisseaux sanguins. 7

Bibliographie

- [1] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07), pages 1-8, 2007.
- [2] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. IEEE, Transactions on pattern analysis and machine intelligence, Vol 14, No 2, pages 239-256, February 1992.
- [3] U. Castellani and A. Bartoli. 3d shape registration. 3D Imaging, Analysis, and Applications, 2012.
- [4] J. Fedmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. Int'l. J. Computer Vision, vol.18, no. 2,pages 99-119, May 1996.
- [5] C. Papazov and D. Burschka. Deformable 3d shape registration based on local similarity transforms. Computer Graphics Forum, vol. 30, no. 5, Aug. 2011.
- [6] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In Third International Conference on 3D Digital Imaging and Modeling, pages 145-152, 2001.
- [7] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision, vol 13, no.2, pages 119-152, June 1992.

Source

- <http://morphomuseum.com/meshtools>
- <http://www.vtk.org/>
- <http://www.ftk.org/>
- http://andrewd.ces.clemson.edu/courses/cpsc805/references/nearest_search.pdf
- <http://gfx.cs.princeton.edu/proj/trimesh2/>
- <http://raptor.developpez.co/tutorial/opengl/vbo/>
- http://www.songho.ca/opengl/gl_vbo.html
- <http://web.cse.ohio-state.edu/~whmin/courses/cse5542-2013-spring/9-VBO.pdf>