

ACADÉMIE DE MONTPELLIER
UNIVERSITÉ MONTPELLIER II
— SCIENCES ET TECHNIQUES DU LANGUEDOC —

MÉMOIRE DE STAGE DE MASTER

SPÉCIALITÉ : **Recherche en Informatique**
Mention : **Informatique, Mathématiques, Statistiques**

effectué au LIRMM

—
sous la direction de GÉRARD SUBSOL

Modélisation de déformations 3D
Application à la paléo-anthropologie

par

Sébastien Escudier

Soutenu en Juillet 2007

Résumé

Dans ce document nous étudions et implémentons une méthode de déformation en 3D entre deux formes complexes avec une application en paléo-anthropologie et reconstruction faciale. Nous proposons ensuite une amélioration de cette méthode en utilisant des distances géodésiques à la place des distances euclidiennes, ainsi qu'une autre utilisation de ces distances géodésiques dans la sélection de points de repère nécessaires à la définition des déformations.

Table des matières

1	Introduction	3
1.1	Modélisation 3D pour la paléo-anthropologie	3
1.2	Les déformations 3D	5
2	État de l'art	7
2.1	Déformations élémentaires	7
2.2	Déformations de formes libres	8
2.3	Interpolation à base de point de repère	8
2.4	Choix de la méthode	9
3	Méthode retenue	10
3.1	Interpolation de Hardy	10
3.2	Implémentation	11
3.3	Résultats	13
3.3.1	Déformation d'une sphère	13
3.3.2	Déformation de crânes	14
4	Améliorations	17
4.1	Distances géodésiques	17
4.2	Calcul des distances géodésiques	17
4.2.1	Calcul sur le maillage	18
4.2.2	Algorithme de Dijkstra	19
4.2.3	Résultat de l'algorithme pour le calcul des distances	20
4.3	Raffinement du maillage	21
4.4	Résultat du calcul des distances géodésiques	21
4.5	Résultats de déformations en géodésique	22
4.6	Augmentation du nombre de points de repère	24
5	Conclusion et perspectives	25

Table des figures

1.1	Acquisition d'un crâne [7].	3
1.2	Images 3D d'un crâne de chimpanzé et d'un crâne d'homme moderne.	4
1.3	Évolution d'un crâne.	4
1.4	Schéma de reconstruction faciale [8].	5
1.5	Exemple d'appariement de points entre deux crânes.	6
2.1	Déformations élémentaires définies par Barr [5].	7
2.2	Déformation de forme libre [6].	8
3.1	Interpolation de Hardy.	11
3.2	Temps pour résoudre le système.	12
3.3	Déformation d'une sphère, avec 10 points de repère déplacés.	13
3.4	Déformation d'une sphère, avec transformation affine et translation.	13
3.5	Déformation d'un crâne humain vers un crâne de chimpanzé.	14
3.6	Déformation d'un crâne humain vers un autre crâne humain, vue 3/4.	15
3.7	Déformation d'un crâne humain vers un autre crâne humain, vue de face.	15
3.8	Déformation d'un visage de référence.	16
4.1	Déplacement de deux points de repère sur une courbe.	18
4.2	Maillage d'une petite partie d'un crâne.	18
4.3	Taille de la liste en fonction de l'itération de l'algorithme.	20
4.4	Ajout de points dit de Steiner sur un triangle.	21
4.5	Cercles de points équidistants suivant la méthode.	22
4.6	Cercles de points équidistants suivant la méthode.	22
4.7	Déformation de sphère en distance géodésique.	23
4.8	Déformation euclidienne et géodésique sur un crâne.	23
4.9	Chemin le plus court entre points de repère sur une orbite d'un crâne.	24
4.10	Déformation avec ajout de points sur une orbite.	24

Chapitre 1

Introduction

1.1 Modélisation 3D pour la paléo-anthropologie

La modélisation ainsi que les déformations en trois dimensions ont beaucoup d'applications, comme créer de nouvelles formes ou analyser et simuler des déformations de surfaces. Celle qui va nous intéresser plus particulièrement est la déformation entre deux formes complexes avec une application en paléo-anthropologie.

Lorsqu'un fossile ossement est retrouvé, il est possible de le passer dans un scanner médical à rayon X pour en obtenir une image numérique en trois dimensions. La figure 1.1 montre le processus : le crâne (A) est placé dans un scanner médical (B). Celui-ci va produire une série d'images qui correspondent à différentes coupes du crâne (C), que l'on pourra ensuite assembler pour produire une image en 3D (D).

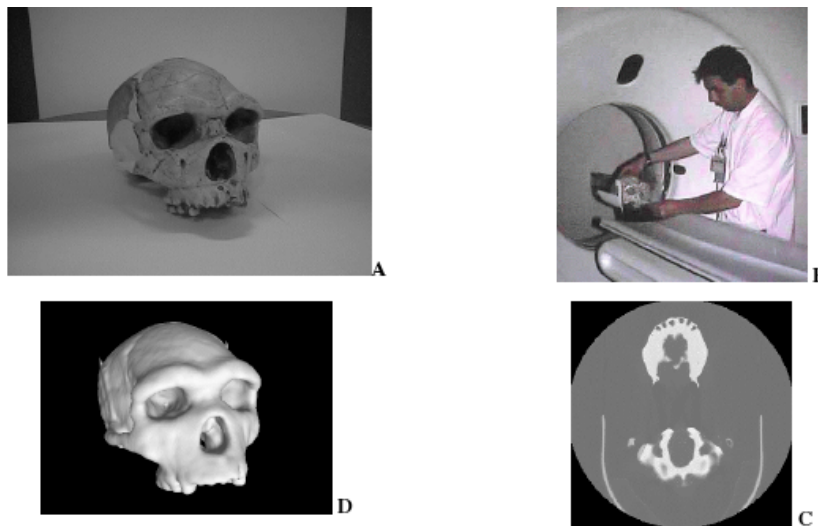


FIG. 1.1 – Acquisition d'un crâne [7].

À partir de cette image 3D, on peut envisager plusieurs applications.

La première est d'effectuer une comparaison entre deux crânes, par exemple entre le crâne d'un chimpanzé et celui d'une homme (voir figure 1.2). On veut alors calculer une déformation en trois dimensions entre ces deux crânes, pour pouvoir la visualiser et analyser les différences morphométriques afin d'étudier le processus évolutif.

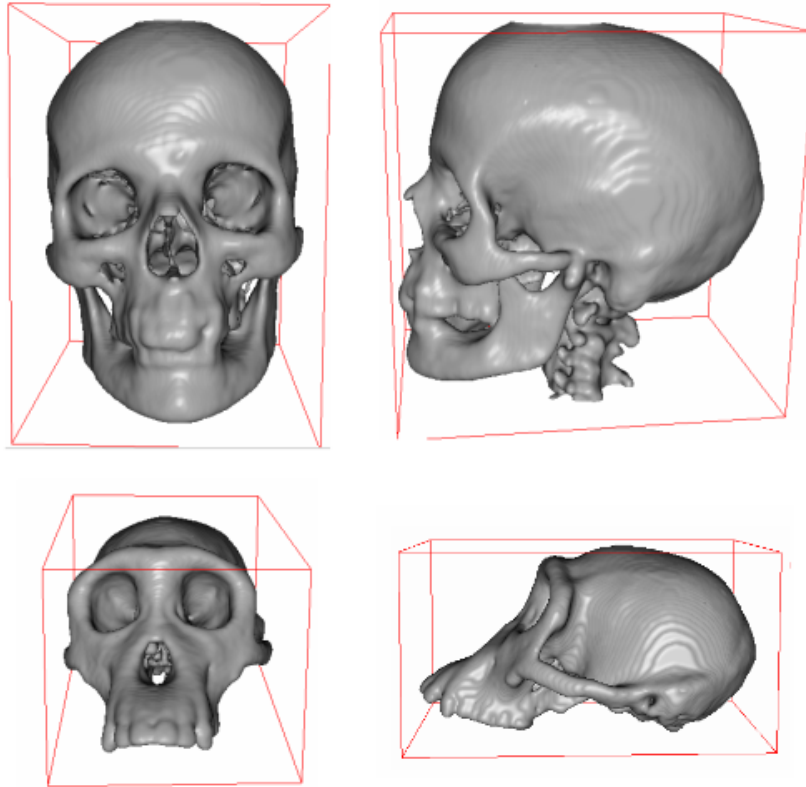


FIG. 1.2 – Images 3D d'un crâne de chimpanzé et d'un crâne d'homme moderne.

On peut aussi essayer d'interpoler une évolution progressive d'un crâne, comme sur la figure 1.3.



FIG. 1.3 – Morphing entre deux crâne pour interpoler l'évolution au cours du temps [10].

La deuxième application possible est la reconstruction faciale. Il s'agit d'extrapoler le visage à partir de la forme du crâne. Les applications sont soit muséographiques (retrouver le visage d'un homme préhistorique), soit en médecine légale (identification d'ossements). La figure 1.4 montre un schéma possible :

1. On calcule la déformation entre un crâne de référence, dont le visage est connu, et le crâne dont on veut retrouver le visage.
2. On peut ensuite appliquer cette déformation au visage du crâne de référence pour extrapoler le visage inconnu.

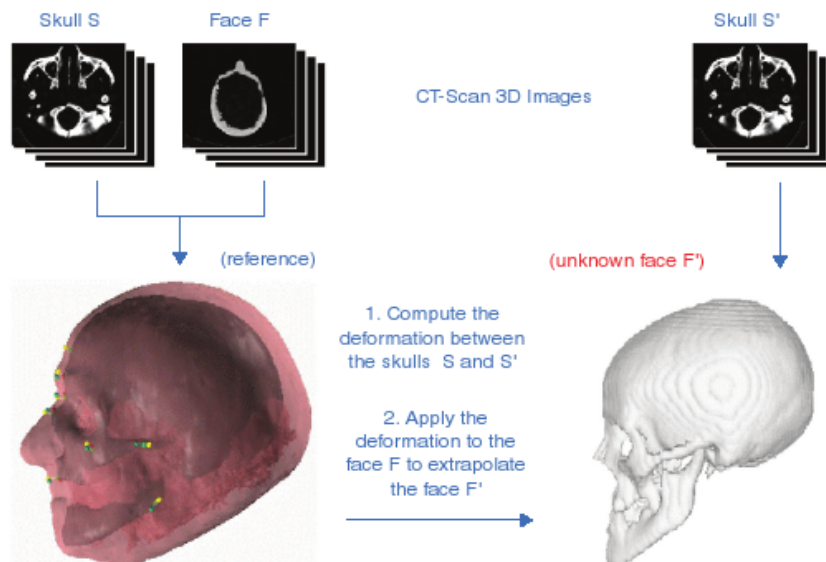


FIG. 1.4 – Schéma de reconstruction faciale [8].

1.2 Les déformations 3D

Pour résoudre ces deux applications, il nous faut donc une modélisation de déformations 3D. Le contexte de déformation dans lequel nous allons travailler est celui des déformations à base de points de repère (aussi appelés landmarks). Nous supposons connu en ensemble de points sur la structure d'origine, ainsi que leurs correspondances sur la structure cible. À partir de cet appariement, nous voulons calculer une déformation. La figure 1.5 montre un tel ensemble.

Pour la suite, nous noterons les points de repère sur la surface d'origine L_i^0 , et ceux sur la surface d'arrivée L_i^1 .

Ces besoins de déformation entre deux formes complexes vont donc nous amener à étudier plus généralement les déformations 3D. Celles-ci ont été largement étudiées, et il en existe de plusieurs types. Une première partie du problème va être d'étudier celles qui existent déjà pour les classer, puis de sélectionner la meilleure méthode pour notre application.

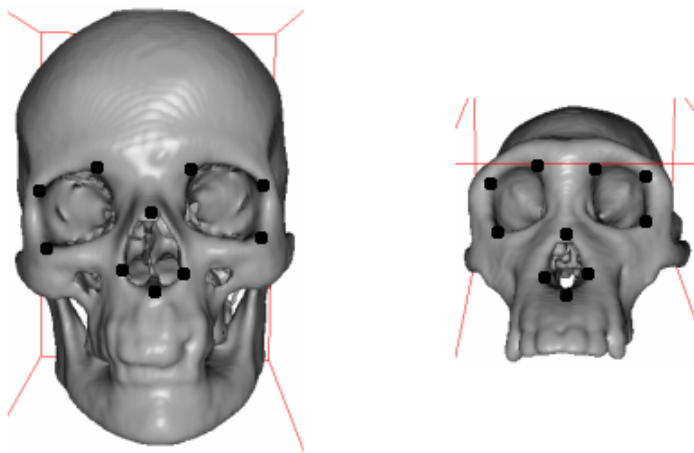


FIG. 1.5 – Exemple d'appariement de points entre deux crânes.

Chapitre 2

État de l'art

Il existe de nombreuses méthodes de déformation 3D, chacune ayant des propriétés spécifiques, et donc des avantages et inconvénients selon l'application visée. Un rapport de recherche récent [5] fait le point sur ces déformations, que l'on peut classer en plusieurs catégories.

Nous en avons sélectionnées quelques unes qui peuvent correspondre à notre application et nous les avons étudiés dans le but final d'en choisir une.

2.1 Déformations élémentaires

La méthode présentée par Barr [1] exprime des déformations locales à l'aide de matrices de transformation.

Une déformation locale modifie l'espace tangent ¹ du solide, alors qu'une déformation globale est une fonction qui modifie directement les coordonnées des points dans l'espace. Cette dernière est mathématiquement représentée par l'équation $X = F(x)$, où x représente les points du solide d'origine et X les coordonnées dans le solide déformé.

Avec une transformation locale, il est possible de transformer les vecteurs tangents et normaux du solide d'origine vers ceux du solide déformé seulement avec une multiplication entre ce vecteur et la matrice Jacobienne de la fonction F qui se calcule en prenant les dérivées partielles de F . On peut donc obtenir les coordonnées transformées de n'importe quel point grâce à F , ainsi que son nouveau vecteur normal et tangent, qui servent notamment à construire la géométrie locale ou obtenir l'orientation de la surface et des informations sur l'éclairage.

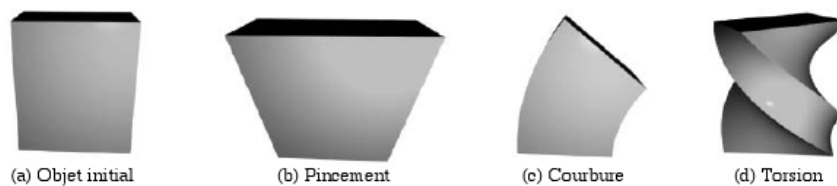


FIG. 2.1 – Déformations élémentaires définies par Barr [5].

¹Pour un point x appartenant au solide, l'espace tangent est défini comme l'ensemble des vecteurs tangents au solide en ce point.

À partir d'une combinaison de déformations élémentaires (voir figure 2.1) sur des solides simples, on peut obtenir des formes plus complexes.

2.2 Déformations de formes libres

Le principe des déformations de formes libres (Free Form Deformation ou FFD en anglais) est de calculer la déformation de n'importe quel point à partir de ses coordonnées dans un repère local prédéfini. C'est la déformation de ce repère qui va entraîner la déformation de l'objet.

Une bonne analogie, faites par Sederberg [6], est d'imaginer un cube de plastique dans lequel est incrusté l'objet que l'on souhaite déformer. Si on déforme ce cube, qui correspond à notre repère local, alors l'objet suit cette déformation (voir figure 2.2).

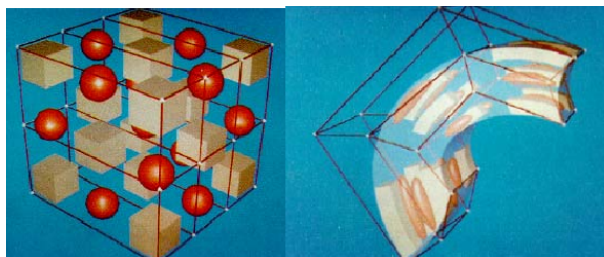


FIG. 2.2 – Déformation de forme libre [6].

Dans le modèle présenté par Sederberg [6], une fois le repère local choisi, il faut choisir des points de contrôle. La déformation sera ensuite définie en déplaçant ces points de contrôle, ce qui va donner une fonction qui transforme tous les points ayant des coordonnées dans ce repère local.

Le problème de ces points de contrôle est qu'ils ne sont pas intuitifs pour celui qui voudrait faire de la modélisation de déformation d'objets car on ne travaille pas sur les points de l'objet eux-mêmes. Hsu a alors proposé [3] une méthode permettant de manipuler directement l'objet sans passer par des points de contrôle. Le principe est de sélectionner n'importe quel point de l'objet et de le déplacer à une autre position. Les points de contrôle sont alors calculés automatiquement.

2.3 Interpolation à base de point de repère

Le but de cette méthode est de faire correspondre un certain nombre de points dans les solides à déformer, en respectant certaines contraintes (notamment de régularité) : la déformation est guidée par un ensemble de points de repère dans le volume d'origine (L^0) ainsi que leur correspondance dans le volume d'arrivée (L^1). La déformation est alors définie par une fonction $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ qui satisfait :

$$F(\vec{L}_i^0) = \vec{L}_i^1 \quad (i = 1 \dots n)$$

Il existe de nombreuses méthodes pour calculer cette fonction, dont deux sont présentés dans un article de Fang [2]. La première, appelée méthode de Shepard, construit l'interpolation en utilisant une distance moyenne pondérée des points de repère. Le problème est que le lissage de la déformation n'est pas parfait, il apparaît souvent des pics ou des creux autour des points de repère.

La deuxième méthode fondée sur des Fonctions de Base Radiale permet de résoudre ce problème. Une fonction de base radiale est simplement défini comme une fonction $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ dont les valeurs dépendent seulement de la distance au point. L'interpolation est alors une combinaison linéaire de ces fonctions.

Une méthode efficace très utilisée est celle de Hardy, que nous étudierons en détail par la suite.

2.4 Choix de la méthode

Les déformations élémentaires sont trop simpliste pour notre application. Les déformations de forme libre sont complexes à mettre en œuvre et pas intuitives car basées sur des points de contrôle difficiles à mettre en place. De plus, elles sont plus destinées à la modélisation d'objets.

La méthode qui nous paraît le mieux correspondre à notre application est donc la méthode d'interpolation et nous avons choisi celle de Hardy qui à l'avantage d'être assez simple algorithmiquement et efficace.

Chapitre 3

Méthode retenue

3.1 Interpolation de Hardy

La méthode de Hardy est basée sur les Fonctions de Base Radiale (ou Radial Basis Function, notée $B()$), qui sont fonction d'un scalaire qui est la distance au point considéré. La fonction d'interpolation F est alors définie comme une combinaison linéaire vectorielle de celles-ci.

$$F(\vec{P}) = \sum_{i=1}^n \vec{h}_i B_i(\vec{P}) \quad (3.1)$$

\vec{h}_i est un coefficient que nous définirons par la suite, et la somme de 1 à n représente la combinaison linéaire sur les n points de repère. La fonction de base radiale, nommée multiquadrique de Hardy, par R.L Hardy (1971), est définie par :

$$B_i(\vec{P}) = (d_i^2(\vec{P}) + r_i^2)^\alpha \quad (3.2)$$

d_i représente la fonction de distance aux points de repère d'origine (notés \vec{L}^0), r_i est un paramètre de contrôle de raideur de la déformation autour des points de repère et α est un exposant généralement égal à 0.5 ou -1. La figure 3.1 fait le point sur les différentes notations. Les flèches représentent les déplacements des points de repère (\vec{L}_i), les pointillés représentent la distance entre le point P et ces points. D'après les équations 3.1 et 3.2, la fonction d'interpolation de Hardy s'écrit :

$$F(\vec{P}) = \sum_{i=1}^n \vec{h}_i (d_i^2(\vec{P}) + r_i^2)^\alpha \quad (3.3)$$

On ajoute en fait une transformation affine et une translation, \vec{t} , pour des considérations mathématiques (il y aura toujours une solution), et cela permet aussi de recaler les deux formes dans l'espace de tel sorte que les fonctions de base radiale n'auront plus qu'à effectuer des petites transformations qui représenteront la déformation réelle entre les deux formes. Les h_i sont appelés les coefficients de Hardy et les paramètres r_i , pour des résultats assez lisses sont généralement définis par : $r_i = \min_{j \neq i} d_i(L_j^1)$.

Le but étant de trouver une interpolation qui fasse coïncider les points de repère d'origine (\vec{L}_i^0) avec ceux d'arrivée (\vec{L}_i^1), on a les conditions suivantes :

$$F(\vec{L}_i^0) = \vec{L}_i^1 \quad (i = 1..n) \quad (3.4)$$

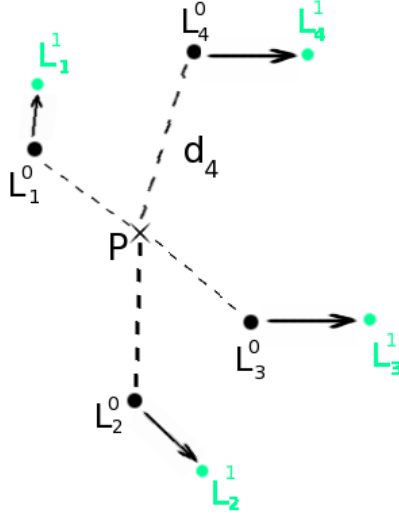


FIG. 3.1 – Interpolation de Hardy.

Cette condition 3.4 et la formule 3.3 nous donnent le système linéaire suivant :

$$\sum_{i=1}^n \vec{h}_i (d_i^2(\vec{L}_i^0) + r_i^2)^\alpha + \vec{t} = \vec{L}_i^1 \quad (3.5)$$

Ce système linéaire donne, en 3 dimensions, 3 systèmes d'équations. Pour x , nous avons :

$$\sum_{i=1}^n h_{i_x} (d_i^2(\vec{L}_i^0) + r_i^2)^\alpha + aL_{i_x}^0 + bL_{i_y}^0 + cL_{i_z}^0 + d = L_{i_x}^1 \quad (3.6)$$

Il y a donc n équations à $n+4$ inconnues. On rajoute les contraintes suivantes qui permettent d'avoir 4 équations supplémentaires :

$$\sum_{i=1}^n h_{i_x} L_{i_x}^0 = \sum_{i=1}^n h_{i_y} L_{i_y}^0 = \sum_{i=1}^n h_{i_z} L_{i_z}^0 = \sum_{i=1}^n h_{i_x} = 0 \quad (3.7)$$

Nous avons donc 3 systèmes (pour x , y , et z) de $n+4$ équations à $n+4$ inconnues qui permettent de calculer les coefficients de Hardy. On peut alors calculer le déplacement de n'importe quel point en fonction de sa distance aux différents points de repère grâce à la fonction d'interpolation F (formule 3.3).

3.2 Implémentation

Une fois cette méthode choisie, il faut maintenant l'implémenter. Le langage choisi est le C++. Pour appliquer la méthode de Hardy, la première chose à faire est de créer le système de $n+4$ équations à $n+4$ inconnues (qui sont les coefficients de Hardy).

Pour ce calcul de coefficients, il faudra connaître :

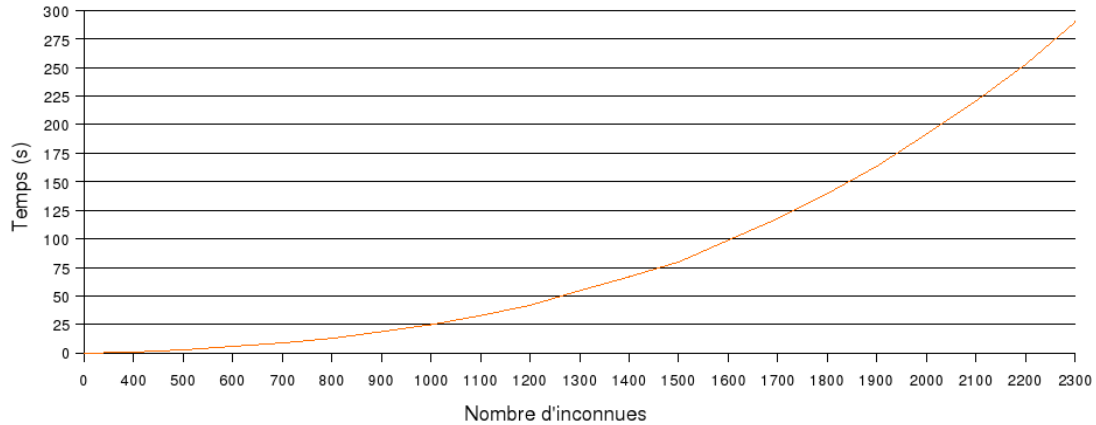


FIG. 3.2 – Temps pour résoudre le système grâce à la méthode de Gauss en fonction du nombre d'inconnues.

- Les distances entre chaque point de repère.
- Les r_i qui représentent la distance entre le point de repère i et le point de repère j le plus proche.

Les distances entre chaque point de repère sont calculées en temps constant. Pour les r_i , on effectuera un pré-calcul pour ne pas avoir à les recalculer à chaque équation. Ce précalcul est effectué en $O(n^2)$, la complexité totale du calcul des coefficients est donc en $O(n^2)$.

Pour résoudre le système linéaire nous avons choisi l'algorithme de Gauss, dont la complexité, est $O(n^3)$. L'algorithme 1 du calcul des coefficients de Hardy a donc une complexité totale en $O(n^3)$.

Données : $\{L_i^0\}$: points de repère sur la surface d'origine, $\{L_j^0\}$: points de repère sur la surface d'arrivée

Résultat : Les coefficients de Hardy

```

1 pour tous les  $L_i^0$  faire
2   | Calculer  $r_i = \min_{j \neq i} d_i(L_j^1)$ ;
3   | pour tous les  $L_j^1$  faire
4   |   | remplir la matrice de coefficients du système avec la distance entre  $L_i^0$  et  $L_j^1$ 
4   |   | ainsi que  $r_i$ ;
5   | fin
6 fin
7 Exécuter l'algorithme de Gauss ;

```

Algorithme 1 : Calcul des coefficients de Hardy.

Nous avons effectué des essais pour connaître le temps nécessaire pour résoudre ces systèmes en fonction du nombre d'inconnues. Pour cela nous avons généré des systèmes d'équation avec des coefficients de type double tirés aléatoirement. On peut voir sur la figure 3.2 que le temps nécessaire à la résolution du système augmente bien en $O(n^3)$ et que pour 2000 inconnues il faut moins de 3 minutes, ce qui nous satisfait largement pour l'instant car nous allons travailler sur n compris entre 10 et 100. Par la suite, si nous avons un méthode permettant de générer un nombre plus important de points de repère, il faudra envisager un algorithme plus efficace.

Une fois le système résolu, il suffit d'appliquer la transformation à chaque point de l'objet ce

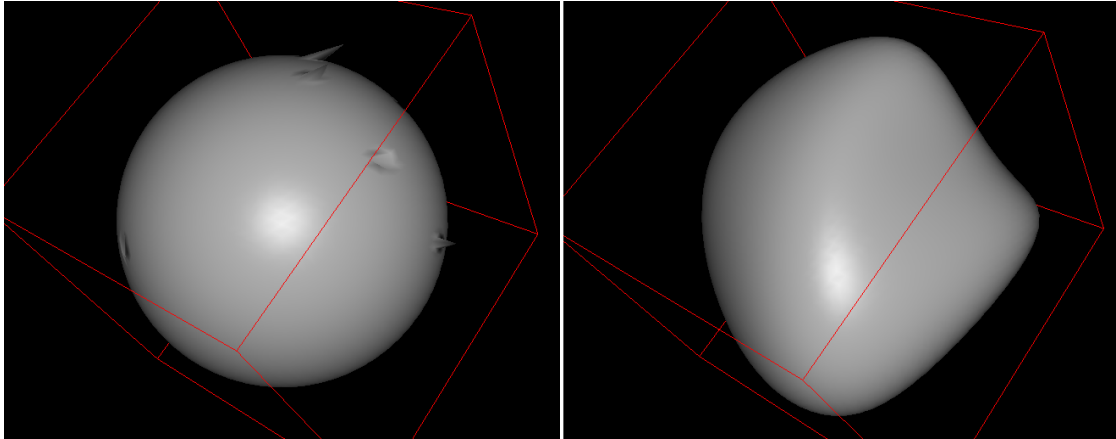


FIG. 3.3 – Déformation d’une sphère, avec 10 points de repère déplacés.

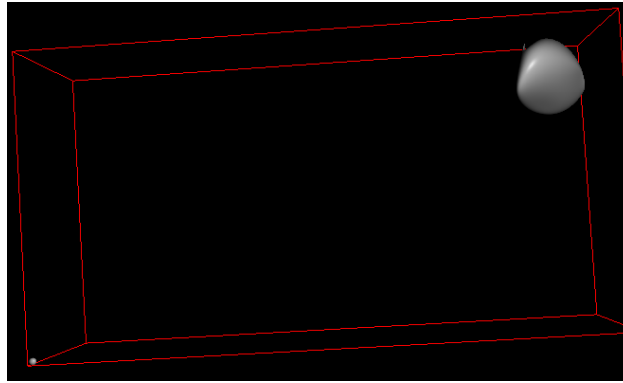


FIG. 3.4 – Déformation d’une sphère, avec transformation affine et translation.

qui se fait en temps constant ($O(p)$, p étant le nombre de points du solide).

3.3 Résultats

3.3.1 Déformation d’une sphère

La première expérience que nous avons faite pour tester la méthode a été de déformer une sphère. Nous avons donc pris une sphère discrète (composé d’environ 100 000 points) où nous avons déplacé un certain nombre de points aléatoirement (une dizaine), qui correspondront à nos points de repère. Les résultats sont assez bons car la déformation suit les points de repère, et est bien lisse. Sur la partie gauche de la figure 3.3 on peut voir la sphère d’origine avec quelques points de repère déplacés, et sur la partie droite la sphère déformée.

Nous avons ensuite effectué une transformation affine et une translation sur les points de repère pour voir si la méthode est robuste même dans des cas avec de forts décalages. On peut voir sur la figure 3.4 en bas à gauche la sphère de départ et en haut à droite la sphère déformée. Donc même avec une forte translation et une transformation affine, la déformation reste cohérente.

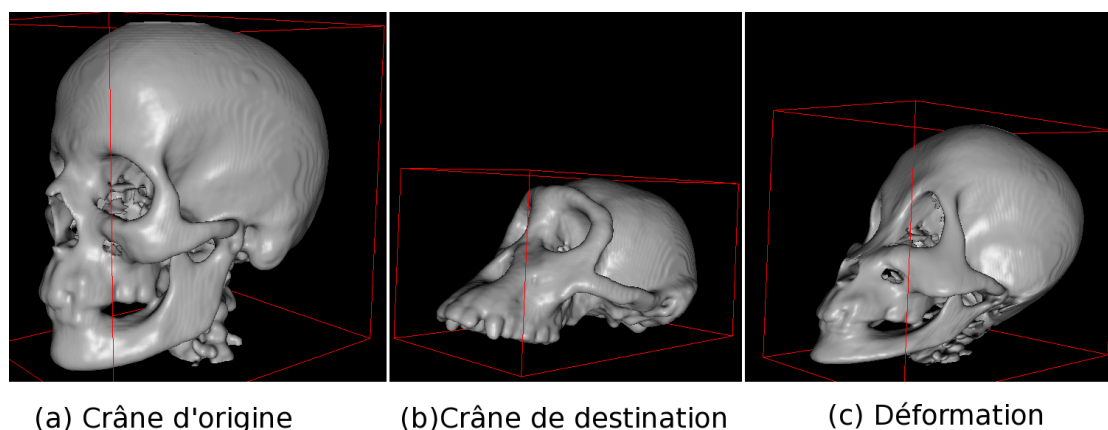


FIG. 3.5 – Déformation d'un crâne humain vers un crâne de chimpanzé.

3.3.2 Déformation de crânes

La déformation de la sphère étant correcte, nous voulons maintenant faire un essai sur un cas réel.

De l'homme au chimpanzé

Nous avons choisi un crâne de chimpanzé et un humain, sur lesquels nous avons sélectionné une vingtaine de points de repère manuellement. Ces points de repère correspondent à des spécificités anatomiques communes aux deux crânes (orbite de l'œil, nez...) que nous pouvons facilement faire correspondre. La figure 3.5 montre en (a) et (b) un crâne humain et un crâne de chimpanzé, et en (c) la déformation du crâne d'humain vers le crâne de chimpanzé. Ce premier résultat est très correct, la déformation est bien régulière, les points de repère d'origine déformés correspondent bien à ceux d'arrivée et il y a bien une ressemblance entre les crânes (b) et (c). Il faut noter que nous ne disposons pas de la mâchoire inférieure du crâne (b), et qu'il n'y a aucun point de repère sur le dessus des crânes car ils seraient difficile à placer précisément.

De l'homme à l'homme

Notre deuxième application étant la reconstruction faciale, nous nous plaçons maintenant dans le cas où nous avons un crâne dont le visage est inconnu. Nous disposons d'un crâne de référence dont nous connaissons le visage, et nous allons tout d'abord calculer une déformation entre ce crâne de référence et le crâne au visage inconnu. La figure 3.6 montre cette déformation en vue 3/4 et nous pouvons notamment observer un nez relevé et une mâchoire plus fine, donc plus proche du crâne de destination.

La figure 3.7 montre cette même déformation, vue de face. On peut voir un nez bien rétréci et la mâchoire élargie.

Cette déformation étant assez convaincante, nous pouvons l'appliquer au visage du crâne d'origine pour extrapoler celui du crâne de destination. Nous disposons d'un moulage des deux visages, ce qui nous permet de comparer notre extrapolation à la réalité. La figure 3.8 montre en (c) le visage de référence (a) déformé. Si nous le comparons au visage que nous souhaitons retrouver (b) nous pouvons noter que le nez et la mâchoire supérieure sont bien relevés. Le menton est également un peu plus large. La difficulté de cette application, et que la ressemblance

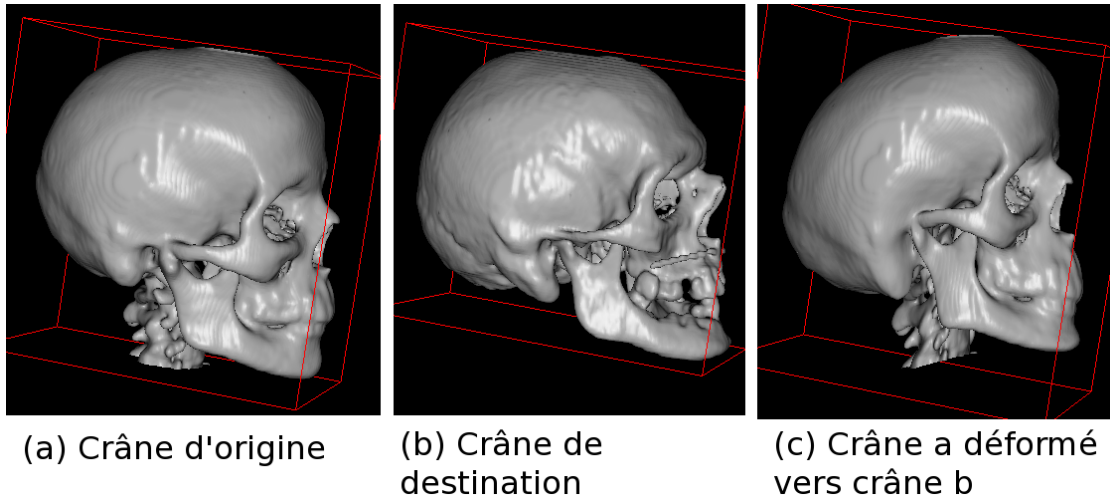


FIG. 3.6 – Déformation d'un crâne humain vers un autre crâne humain, vue 3/4.

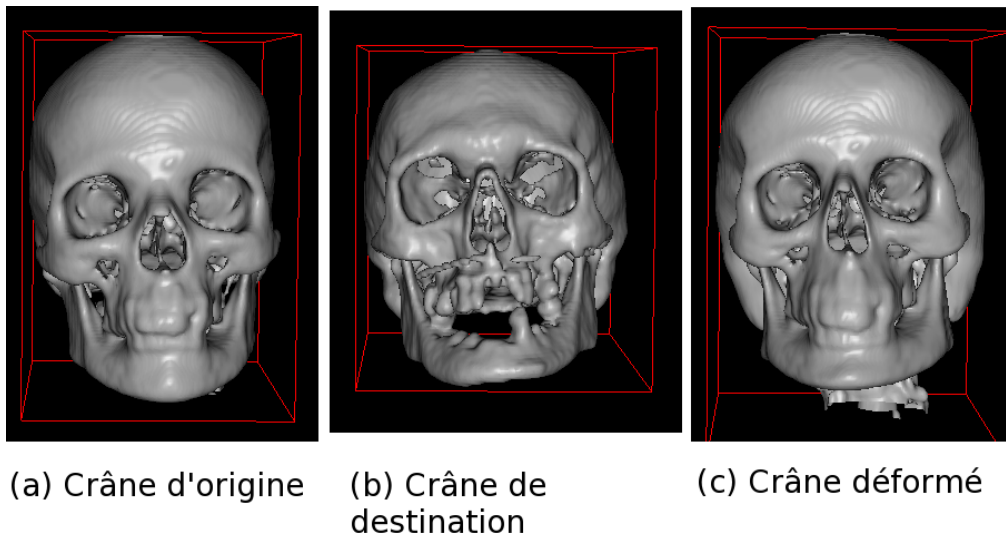


FIG. 3.7 – Déformation d'un crâne humain vers un autre crâne humain, vue de face.

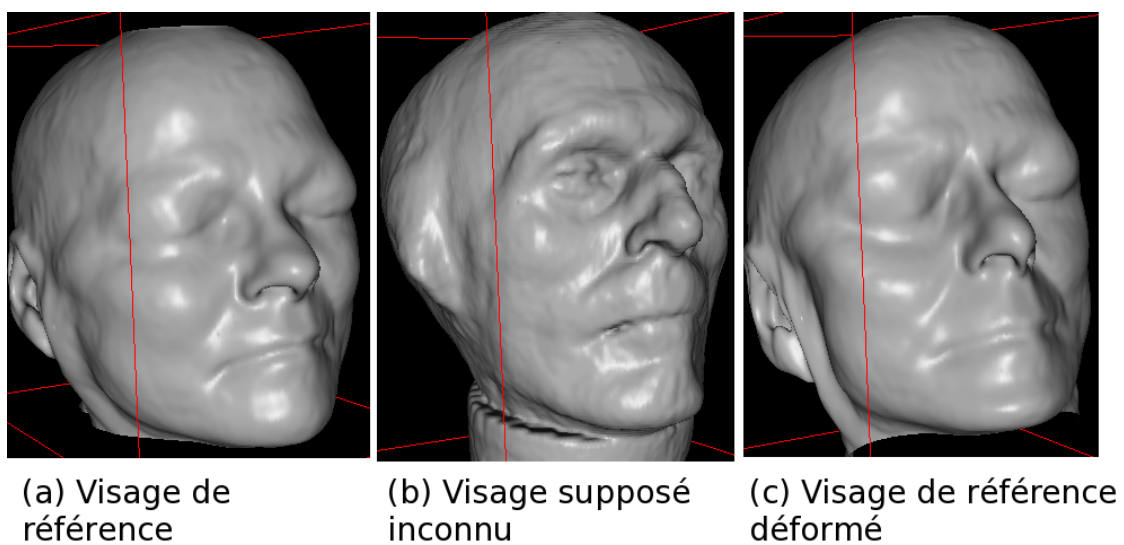


FIG. 3.8 – Déformation d'un visage de référence.

entre deux visages est subjective et dépend de plusieurs paramètres. En effet, l'orientation du visage va faire qu'on peut reconnaître ou pas une personne, et la texture de la peau, la couleur des yeux, des cheveux, vont également jouer un rôle. La connaissance de l'âge, du sexe, de l'ethnie et de la corpulence du sujet peuvent aussi être déterminant.

Chapitre 4

Améliorations

Nous avons vu que la méthode de Hardy fait intervenir une fonction distance. En effet l'interpolation est calculée en fonction de la distance de chaque point de repère aux autres points de repère ($d_i^2(\vec{L}_i^0)$ dans la formule 3.5). Puis la déformation de chaque point prend en compte sa distance à chaque point de repère ($d_i^2(\vec{P})$ formule 3.3). La fonction distance que nous avons utilisé était tout simplement la distance euclidienne entre les points.

Très simple à calculer car nous avons les coordonnées de chaque point, est-ce qu'elle représente le mieux l'influence que chaque point de repère doit apporter aux points du solide à déformer ?

Nous avons donc pensé à utiliser une autre distance, qui est la distance géodésique.

4.1 Distances géodésiques

La distance euclidienne est tout simplement la distance en «ligne droite» entre deux points. La distance géodésique est la distance qu'il faut parcourir entre deux points en passant par la surface de l'objet.

Quels peuvent être les avantages ?

Le problème de la distance euclidienne, est que deux points assez proche dans l'espace, peuvent être beaucoup plus éloignés si on doit suivre la surface pour les relier. On peut voir sur la figure 4.1 que le point au centre de la courbe est équidistant des deux points aux extrémités. Donc si les points aux extrémités se déplacent en sens opposé, le point central ne se déplacerait pas. Par contre, on voit bien que la distance géodésique (en suivant la courbe) n'est pas la même et que le point du milieu est bien plus proche du point de droite, et qu'il devrait être plus influencé par celui-ci.

À notre connaissance, cela n'a jamais été utilisé, même si une référence non publié (rapport de DEA) l'évoque pour une méthode autre que Hardy.

4.2 Calcul des distances géodésiques

L'inconvénient de cette distance géodésique est que son calcul n'est pas aussi immédiat qu'en distance euclidienne, où les coordonnées de deux points suffisaient à connaître immédiatement leur distance.

Nous avons cherché une méthode simple à mettre en œuvre pour ce nouveau calcul de distance.

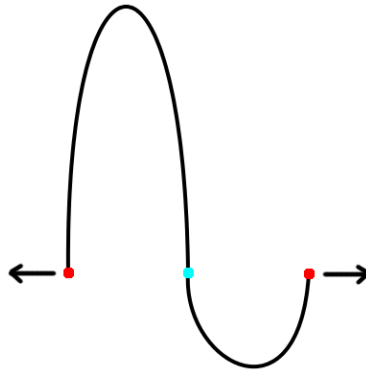


FIG. 4.1 – Déplacement de deux points de repère sur une courbe.

4.2.1 Calcul sur le maillage

La première idée que nous avons eu a été de calculer la distance géodésique entre deux points en suivant les arêtes du maillage des objets car ceux-ci sont modélisés grâce à un maillage triangulé (vois figure 4.2). Bien sûr, il faut que le maillage soit «correctement» discrétisé, et ce ne serait pas des distances géodésique exactes, mais cela peut fournir une première approximation.

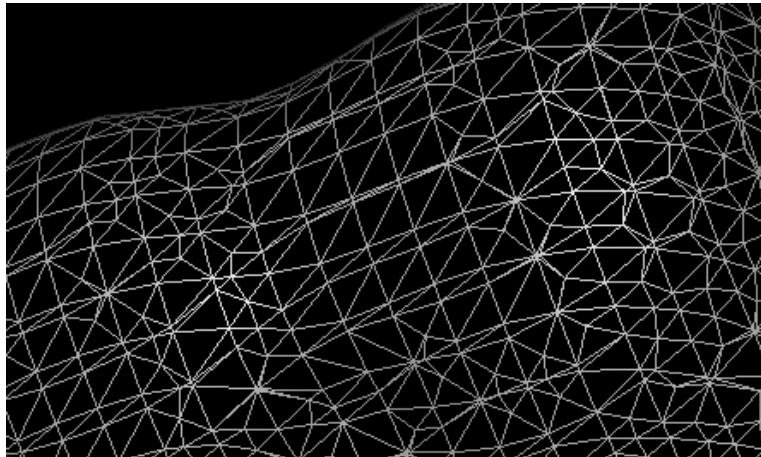


FIG. 4.2 – Maillage d'une petite partie d'un crâne.

Pour connaître la distance entre deux points, il faut donc trouver le plus court chemin entre ces deux points sur le maillage. Celui-ci peut être assimilé à un graphe, les sommets étant les points, les arêtes étant les droites des triangles, et trouver le plus court chemin dans un graphe est un problème bien connu.

L'algorithme de Dantzig-Ford et de Ford-Bellman permettent de le résoudre dans un graphe dont le poids des arêtes peuvent être négatifs. L'algorithme de Dijkstra ne marche que sur les graphes avec des poids positifs ou nuls, ce qui est notre cas puisque pour nous le poids d'une arête correspond à la distance entre deux points. Nous avons choisi cet algorithme pour sa meilleure complexité.

4.2.2 Algorithme de Dijkstra

Implémentation la plus simple

La complexité de l'algorithme de Dijkstra dépend de la structure de donnée utilisée. Tout d'abord voyons le principe de cet algorithme, qui calcule les plus courts chemins depuis un point source pour tous les autres points.

Au départ il y a un ensemble E qui contient tous les sommets sauf la source. Le but va être de parcourir n fois cet ensemble, pour à chaque étape extraire le sommet le plus proche de la source, comme nous pouvons le voir dans l'algorithme 2.

Données : Un graphe G , s : sommet source appartenant à G
Résultat : Un tableau avec les distances de tous les points à s

```
1 Initialisation : Tous les sommets reliés à la source ont leur distance connue, les autres
   ont une distance  $\infty$ ;  
2 pour  $i$  de 1 à  $n$  faire  
3   |  $min = \text{ExtraireDistanceMinimum}$  de  $E$ ;  
4   | pour Tous les sommets de  $E$  reliés à  $min$  faire  
5   |   | Mettre à jour les distances avec le poids des arêtes;  
6   |   fin  
7 fin
```

Algorithme 2 : Dijkstra.

La fonction *ExtraireDistanceMinimum* renvoie le sommet de E le plus proche de la source. L'implémentation la plus simple de cet algorithme est de parcourir la liste E en $O(n)$ pour en extraire le minimum. La mise à jour des distances s'effectue une fois par arête, le nombre d'arêtes étant en $O(n)$ la complexité totale est $O(n^2)$.

Dans notre cas, nous avons besoin de la distance des points du solide à déformer aux points de repère (voir formule 3.3, $d_i^2(\vec{P})$ étant la distance).

Les déformations de sphères et de crânes comportaient une vingtaine de points de repère, ce qui reviendra à appliquer l'algorithme de Dijkstra 20 fois par déformation. La sphère la plus dense que nous ayons fait 200 000 sommets, et le crâne humain 300 000. Nos premiers essais avec la version de Dijkstra en $O(p^2)$ ont montré des temps de calculs assez long (plusieurs dizaines de minutes pour un seul point de repère), ce qui serait pénalisant pour faire des essais sur un crâne et plusieurs dizaines de points de repère.

Amélioration de la complexité

Pour améliorer cette complexité, il faudrait implémenter une structure de donnée plus efficace qui permette de mettre à jour une liste de distance et d'effectuer une recherche dans cette liste. Des solutions sont connues avec une complexité en $O(\ln(n))$, mais leur implémentation aurait été trop complexe.

Nous avons alors pensé à une autre amélioration possible : au lieu de faire une recherche sur tous les sommets pour en extraire le minimum, on pourrait tenir à jour une liste des sommets ayant une distance connue (voir algorithme 3).

En effet, pour extraire le sommet le plus proche, seuls les sommets dont leurs distances ont été mises à jour (ligne 5 de l'algorithme 2) ou qui sont reliés à la source sont susceptibles d'être sélectionnés car au départ les distances de tous les sommets sont inconnues. Donc cela peut réduire considérablement le nombre de sommets à parcourir.

L'algorithme 3 montre cette modification.

```

Données : Un graphe  $G$ ,  $s$  : sommet source appartenant à  $G$ 
Résultat : Un tableau avec les distances de tous les points à  $s$ 
1 Tous les sommets reliés à la source ont leur distance connue, les autres ont une
  distance  $\infty$ ;
2  $Liste \leftarrow$  tous les sommets reliés à  $s$ ;
3 pour  $i$  de 1 à  $n$  faire
4    $min = \text{ExtraireDistanceMinimum}$  de  $Liste$ ;
5    $Liste \leftarrow Liste - min$  ;
6   pour Tous les sommets reliés à  $min$  faire
7     Mettre à jour les distances;
8     Rajouter dans  $Liste$  les sommets dont les distances ont été mises à jour ;
9   fin
10 fin

```

Algorithme 3 : Dijkstra modifié.

La complexité de cet algorithme de Dijkstra modifié dépend grandement de la taille de la liste et donc de la structure du graphe, ou du maillage dans notre cas. Dans le cas d'un graphe avec des sommets de degré important, l'avantage peut ne pas être si sensible car la taille de la liste à parcourir augmenterait très rapidement. Mais dans notre cas d'un maillage triangulé le gain est important.

En pratique, sur un crâne d'environ 300 000 sommets, au lieu de parcourir tous les sommets à chaque itération, on parcourt au maximum 2000 sommets (voir figure 4.3). Les gains de temps sont alors considérables, et l'algorithme de Dijkstra se finit en quelques dizaines de secondes, voir quelques minutes pour les graphes les plus gros (300 000 sommets).

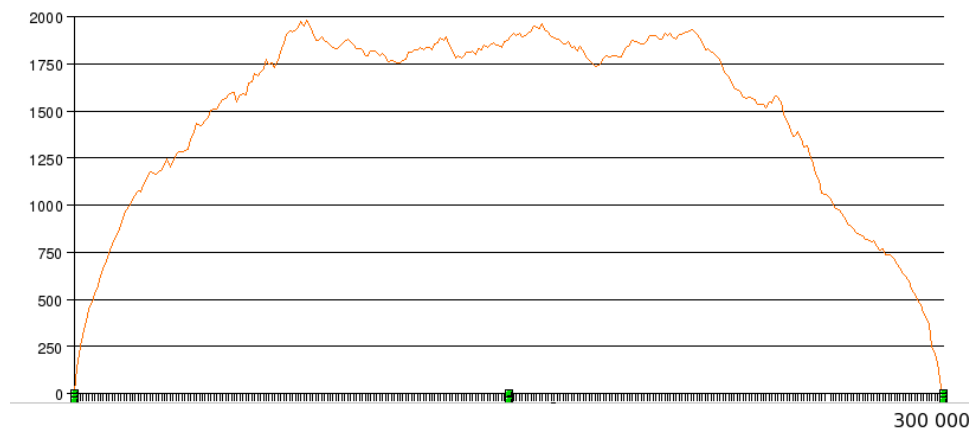


FIG. 4.3 – Taille de la liste en fonction de l'itération de l'algorithme.

4.2.3 Résultat de l'algorithme pour le calcul des distances

Cette méthode de calcul de distances géodésiques n'étant pas exacte, nous avons voulu savoir quelle en était la précision.

Pour cela nous avons lancé cet algorithme sur une sphère sur laquelle il est facile de calculer la distance géodésique exacte pour la comparer avec la distance approchée. Il est alors apparu

des écarts pouvant aller jusqu'à 20%. De plus, le maillage n'étant pas uniforme, l'orientation des triangles ainsi que leur densité jouent un rôle dans la précision des distances. Ainsi, selon le placement du point source et des points que l'on calcule, les distances peuvent être plus ou moins précises, comme nous pourrions le visualiser par la suite.

Une solution aurait été d'avoir un algorithme permettant de calculer les distances géodésiques exacte. Plusieurs solutions sont disponibles, dont un article récent [9] que nous avons étudié qui propose des algorithmes. Mais L'implémentation de l'algorithme et de la structure de donnée nécessaire aurait été complexe, et nous avons cherché une méthode plus accessible.

4.3 Raffinement du maillage

La méthode que nous avons sélectionné, est présentée dans un article [4] présentant différentes approximations de plus courts chemins sur des surfaces. Le principe est de rajouter des points sur les arêtes du maillage de manière à raffiner ce dernier (voir figure 4.4), et ainsi rendre le calcul des distances plus précis. Nous pouvons ajouter un nombre plus ou moins important de points, mais avec un point supplémentaire par arête les erreurs de distance sont réduites à environ 1% sur la sphère.

Pour créer ces nouveaux points, il faut parcourir toutes les arêtes du graphe, ajouter un point au milieu de celles-ci, et relier ce dernier à tous les autres points situés sur les arêtes adjacentes. Les distances pour les nouveaux liens créés sont tout simplement les distances euclidiennes entre les sommets.

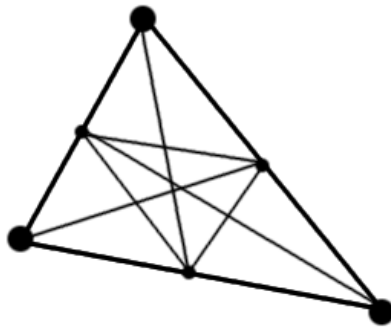


FIG. 4.4 – Ajout de points dit de Steiner sur un triangle [4], qui permettent de rajouter des arêtes.

4.4 Résultat du calcul des distances géodésiques

Pour comparer l'efficacité du calcul des distances, nous avons voulu les visualiser.

Nous avons donc pris une sphère sur laquelle nous avons calculé les distances de tous les points depuis un point source, puis nous avons colorié les points qui se trouvent à certaines distances de ce point source.

En distance exacte, ces points forment des cercles sur la sphère. Nous pouvons voir sur la figure 4.5 que les distances calculées avec les points de Steiner sont meilleures et s'approchent des distances exactes. Sur la figure 4.6, nous pouvons voir que dans certains cas, si le point source

est placé à un endroit du maillage qui n'est pas avantageux, les résultats sont plus approximatifs. Cette figure montre que plus on s'éloigne du point source, plus les résultats sont mauvais.

Les points de Steiner améliorent tout de même grandement l'efficacité du calcul, mais l'erreur par rapport aux distances exactes n'est pas négligeable, ce qui peut poser des problèmes lors du calcul des déformations car les erreurs peuvent être faibles si on se trouve proche des points de repère, et plus importantes quand on s'éloigne et qu'on se trouve à des endroits où le maillage ne favorise pas les chemins les plus courts.

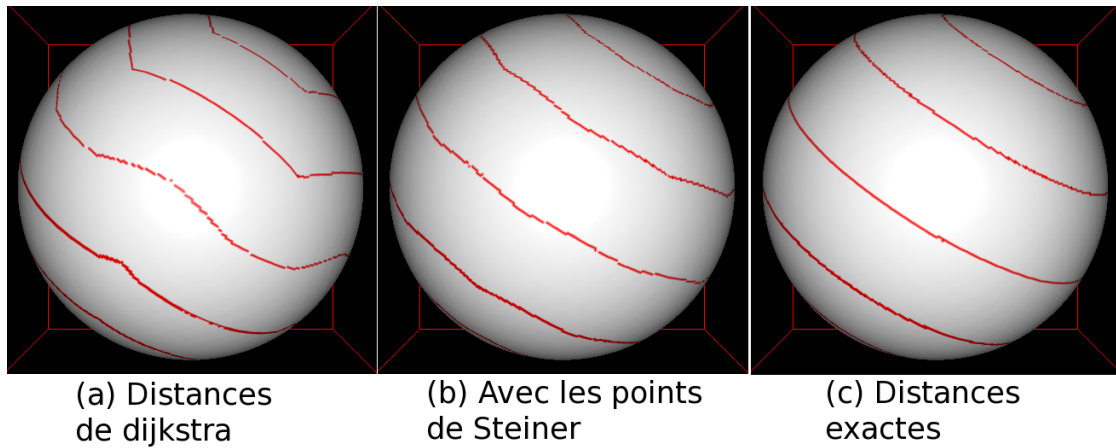


FIG. 4.5 – Cercles de points équidistants suivant la méthode.

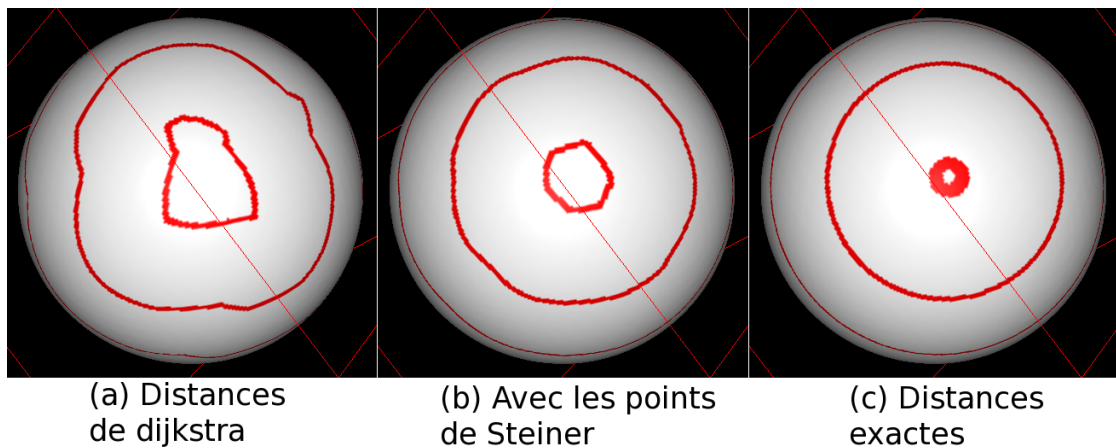


FIG. 4.6 – Cercles de points équidistants suivant la méthode.

4.5 Résultats de déformations en géodésique

Pour tester la déformation en distance géodésique, nous avons commencé par déformer une sphère. La figure 4.7 montre que la déformation en utilisant les distances avec le maillage raffiné

par les points de Steiner est plus précise, car les défauts occasionnés par les erreurs de calcul des distances sont en partie gommés.

Mais la comparaison avec une déformation euclidienne n'est pas significative car sur une sphère les différences entre le rapport des distances aux points de repère entre les deux modes de calcul sont minimales.

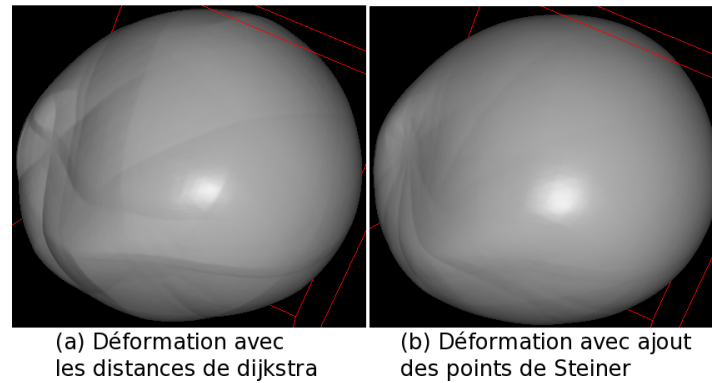


FIG. 4.7 – Déformation de sphère en distance géodésique.

La déformation d'un crâne avec des distances géodésiques ne fait pas non plus apparaître de différences majeures. Ceci est certainement dû au faible nombre de points de repère. Nous avons essayé de générer un exemple de déformation sur un crâne avec des points de repère susceptibles de produire plus de différence entre les deux modes de déformation. La figure 4.8 (a) montre le crâne de départ avec deux vecteurs de déformation. Nous pouvons observer en (b) et (c) les deux différences suivantes :

- Les points situés en dessous du vecteur du haut suivent mieux la déformation de ce dernier dans le cas géodésique, car ils sont beaucoup plus influencés par celui-ci. Cette zone est donc mieux déformé vers le haut et garde sa forme initiale car ils ne sont pratiquement pas déformé vers le bas dans le même temps.
- L'os sur lequel nous avons effectué une déformation vers le bas (appelé arcade zygomatique) reste plus fine et donc plus proche de l'original en mode géodésique. Cela est dû au fait que le vecteur supérieur est beaucoup plus loin qu'en distance euclidienne, et ce dernier a donc moins d'influence dans la déformation de cette arcade.

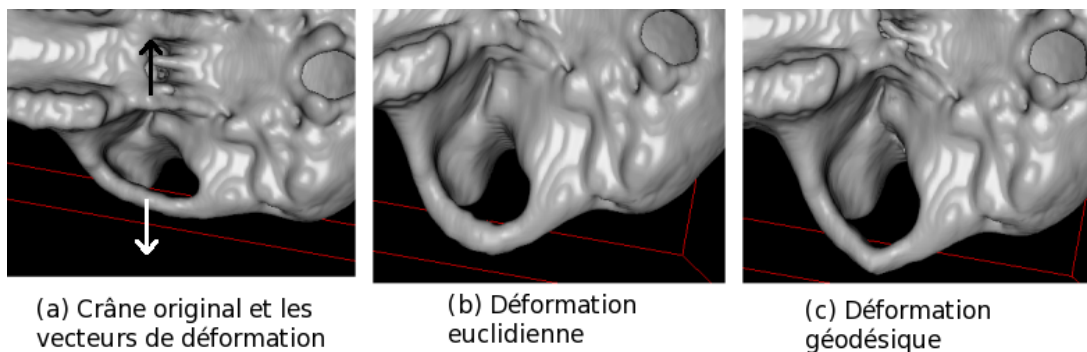


FIG. 4.8 – Déformation euclidienne et géodésique sur un crâne.

4.6 Augmentation du nombre de points de repère

Ce calcul des distances géodésiques peut également avoir une autre utilité.

Lors de la déformation d'un crâne de chimpanzé vers un crâne humain par exemple, les points de repère ont été choisis à des endroits spécifiques qui correspondent à des points anatomiques précis pour pouvoir faire la correspondance entre les points des deux crânes. Par exemple, nous avons choisi trois points sur l'orbite de l'œil qui sont facilement identifiables. Or, ces spécificités anatomiques peuvent nous servir à définir plus de points de repère.

Si nous visualisons le chemin emprunté sur le maillage par la distance géodésique entre les points de l'orbite des deux crânes de référence, on se rend compte qu'ils empruntent sensiblement le même chemin quel que soit le crâne, ce qui est logique car même si la forme de l'orbite n'est pas tout à fait la même, le chemin le plus court entre deux points sur l'orbite suivra l'intérieur de celle-ci, comme nous pouvons le voir sur la figure 4.9.

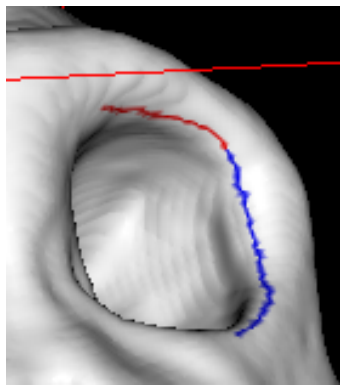


FIG. 4.9 – Chemin le plus court entre points de repère sur une orbite d'un crâne.

Notre idée a été de prendre un certain nombre de points sur ce chemin, répartis de la même manière sur les deux crânes pour en faire des nouveaux points de repère. Ces points supplémentaires vont permettre d'avoir une déformation plus précise.

La figure 4.10 montre, sur le crâne de droite, pour l'orbite gauche, que la déformation de la partie supérieure de l'orbite est plus proche du crâne original (à gauche), car plus aplatie que celle de l'orbite droite où nous n'avons pas ajouté de point supplémentaire.

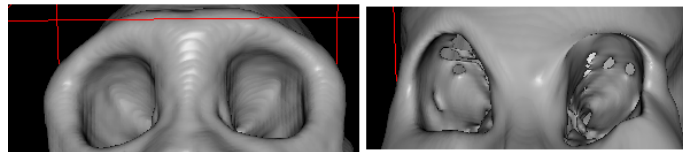


FIG. 4.10 – Déformation avec ajout de points sur une orbite.

Nous n'avons pas eu le temps de travailler plus sur cette méthode de sélection de point de repère, mais nous pensons que cela peut être une aide précieuse à la sélection manuelle qui est fastidieuse et dont la précision n'est pas assurée. On peut par exemple envisager, une fois la ligne de plus court chemin tracée, de pouvoir la modifier légèrement manuellement pour bien la mettre en correspondance sur les deux crâne grâce à des algorithmes de déformation de courbe.

Chapitre 5

Conclusion et perspectives

Dans ce stage nous avons sélectionné une méthode de modélisation de déformation 3D pouvant s'appliquer à la déformation de crâne en paléo-anthropologie et reconstruction faciale, et nous l'avons implémentée.

Nous avons fait plusieurs expériences de déformation et les résultats sont convaincants.

Nous avons ensuite proposé une amélioration consistant à utiliser les distances géodésiques à la place des distances euclidiennes.

Même si cette méthode n'a pas montré de différences notables dans la déformation d'un crâne tel que nous l'avons appliquée, nous pensons que cela est dû au faible nombre de points de repère utilisés et des expériences sur des points plus précis montrent que cela peut être intéressant.

Il faudrait donc effectuer des expériences supplémentaires et étudier plus de cas, d'autant plus que cette question de déformation en utilisant les distances géodésiques n'a à notre connaissance pas été étudiée précédemment.

Ce calcul des distances géodésiques peut également servir à définir des chemins le long desquels on peut augmenter le nombre de points de repère, et donc améliorer les déformation. Cette méthode pourrait également être approfondie, pour en faire une aide à la localisation de points de repère.

Bibliographie

- [1] Alan H. Barr. Global and Local Deformations of Solid Primitives. In *SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 21–30, New York, NY, USA, 1984. ACM Press.
- [2] Shiaofen Fang, Rajagopalan Srinivasan, Raghu Raghavan, and Joan T. Richtsmeier. Volume Morphing and Rendering an Integrated Approach. *Comput. Aided Geom. Des.*, 17(1) :59–81, 2000.
- [3] William M. Hsu, John F. Hughes, and Henry Kaufman. Direct Manipulation of Free-Form Deformations. In *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 177–184, New York, NY, USA, 1992. ACM Press.
- [4] Mark Lanthier, Anil Maheshwari, and Jörg-Rüdiger Sack. Approximating weighted shortest paths on polyhedral surfaces. In *6th Annual Video Review of Computational Geometry, Proc. 13th ACM Symp. Computational Geometry*, pages 485–486. ACM Press, 4–6 1997.
- [5] Romain Raffin, Gilles Gesquière, and Raphaël La Greca. Déformations de modèles géométriques. Technical Report LSIS.RR.2007.001, LSIS, Janvier 2007.
- [6] Thomas W. Sederberg and Scott R. Parry. Free-Form Deformation of Solid Geometric Models. In *SIGGRAPH '86 : Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, New York, NY, USA, 1986. ACM Press.
- [7] G. Subsol, B. Mafart, A. Silvestre, and M.A. de Lumley. 3D Image Processing for the Study of the Evolution of the Shape of the Human Skull : Presentation of the Tools and Preliminary Results. pages 37–45. XIV International Congress of Prehistoric and Protohistoric Science, BAR International Series, 2002.
- [8] G. Subsol and G. Quatrehomme. *Automatic 3D Facial Reconstruction by Feature-Based Registration of a Reference Head*, chapter 5. Computer-Graphic Facial Reconstruction. M. Marks, J. Clement (eds.), 2005.
- [9] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J. Gortler, and Hugues Hoppe. Fast Exact and Approximate Geodesics on Meshes. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 553–560, New York, NY, USA, 2005. ACM Press.
- [10] David F. Wiley, Nina Amenta, Dan A. Alcantara, Deboshmita Ghosh, Yong J. Kil, Eric Delson, Will Harcourt-Smith, Katherine St. John, F. James Rohlf, and Bernd Hamann. Evolutionary morphing. *vis*, page 55, 2005.