



**HAL**  
open science

## Reconstruction et complétion de maillages sous contraintes

Zhifan Jiang, Fabrice Jaillet, Florence Zara

► **To cite this version:**

Zhifan Jiang, Fabrice Jaillet, Florence Zara. Reconstruction et complétion de maillages sous contraintes. [Rapport de recherche] LIRIS UMR CNRS 5205. 2011. hal-01493740

**HAL Id: hal-01493740**

**<https://hal.science/hal-01493740>**

Submitted on 22 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Reconstruction et complétion de maillages sous contraintes

Zhifan Jiang, Fabrice Jaillet et Florence Zara

Laboratoire d'InfoRmatique en Image et Systèmes d'information  
LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1

**Résumé** Dans le cadre d'un projet visant la réalisation d'un simulateur d'accouchement, nous cherchons à créer les modèles géométriques des organes impliqués à partir d'images médicales. Mais ces images médicales ne contiennent généralement qu'une partie significative de l'organe. Nous nous focalisons ainsi sur la reconstruction des maillages générés et la complétion des parties manquantes. Pour cela, nous avons étudié différentes méthodes de modification et de complétion des maillages. Par rapport aux méthodes existantes, notre solution ne concerne pas seulement la complétion de maillages mais elle tient aussi compte de différentes contraintes pré-définies. Notre solution inclut la triangulation, le raffinement et la déformation sous contraintes des maillages surfaciques.

**Abstract** In the context of the Childbirth Simulator Project, we need to create the geometric models of organs from medical imaging. In the general case, the medical images contain only a significant part of the organ, thus, our problem is how to reconstruct and complete the mesh by filling the missing parts. To achieve this goal, we have studied the main approaches for mesh editing and holes filling in order to offer our approach. Compared with the existing methods, our approach also takes into account the various pre-defined constraints particularly during the mesh completion. Our method includes triangulation, refinement and constrained deformation of surface meshes.

**Keywords :** Medical image, Mesh generation, Remeshing, Holes filling, Smoothing, Deformation

# Table des matières

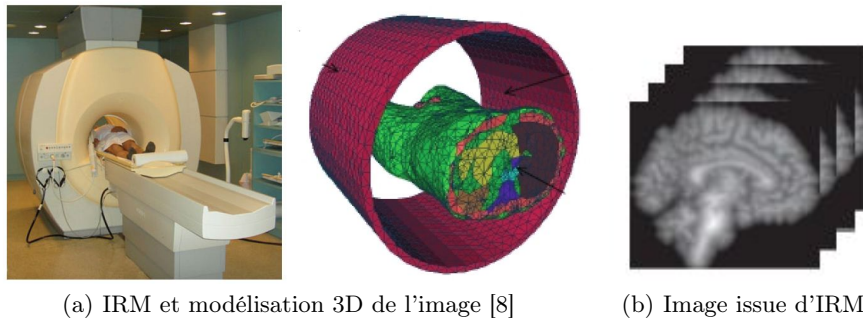
Reconstruction et complétion de maillages sous contraintes .....	1
<i>Zhifan Jiang, Fabrice Jaillet et Florence Zara</i>	
1 Introduction.....	3
1.1 Contexte .....	3
1.2 Objectif.....	4
2 État de l’art.....	4
2.1 Représentation de surfaces.....	5
2.2 Maillages .....	6
2.3 Structure de données .....	7
2.4 Génération de maillages .....	7
2.5 Lissage.....	10
2.6 Complétion de maillages .....	11
3 Contributions .....	12
4 Méthode de reconstruction .....	13
4.1 Maillage d’entrée .....	13
4.2 Contraintes .....	14
4.3 Approche proposée .....	15
4.4 Triangulation .....	15
4.5 Raffinement .....	17
4.6 Lissage.....	18
4.7 Déformation .....	21
4.8 Finition .....	24
5 Applications.....	24
5.1 Développement .....	24
5.2 Résultats .....	24
6 Conclusions et perspectives.....	26

## 1 Introduction

Grâce aux progrès de la technologie physique, l'imagerie médicale et ses applications jouent un rôle très important depuis ces dernières décennies dans le domaine médical, en particulier la technologie d'IRM qui a connu un rapide essor durant ces trente dernières années.

Divers systèmes d'acquisition (IRM, Scanner CT, etc.) permettent ainsi de collecter de larges volumes de données sur des organes humains (Figure 1). En fonction de la technologie utilisée, les données produites sont soit des séries de coupes parallèles, soit des ensembles de voxels, soit des nuages de points [11]. Pour différentes applications en informatique, nous souhaitons créer les modèles géométriques des organes du corps humain à partir des données obtenues par ce type d'acquisition. Mais les systèmes de numérisation ne produisent généralement qu'une partie significative de d'organes (ou du moins lors du traitement du patient, les images acquises ciblent uniquement un organe précis). Ainsi, les maillages construits à partir de ces acquisitions sont souvent incomplets (il manque une partie des organes). Ceci n'est pas favorable pour les traitements ultérieurs (visualisation, simulation, etc.).

Nous nous intéressons donc à concevoir une méthode permettant la reconstruction des maillages générés grâce à l'extrapolation des parties manquantes et ceci de manière automatique.



**FIGURE 1.** Acquisition de données

### 1.1 Contexte

Mon sujet de stage se situe dans le cadre d'un projet visant la réalisation d'un simulateur d'accouchement pour lequel nous cherchons à créer les modèles géométriques des organes. Le sujet concerne plus particulièrement une étape précise du traitement : reconstruction et complétion de maillages surfaciques générés. Dans la figure 2, nous avons mis en évidence le positionnement du sujet dans la chaîne de traitement complète.

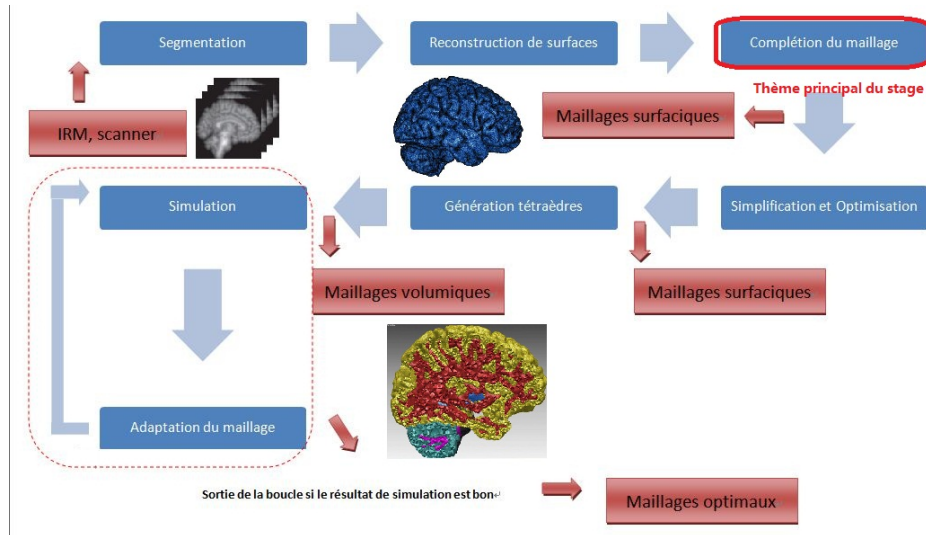


FIGURE 2. Chaîne de traitement

## 1.2 Objectif

**Objectif scientifique.** Notre but est de chercher un algorithme permettant de reconstruire et compléter des maillages d'une façon automatique. De plus, nous avons envie de tenir compte de différentes contraintes introduites lors de la complétion de maillages. Ces contraintes seront définies par la suite (Section 4.2).

**Objectif applicatif.** Nous voulons d'abord implémenter la méthode proposée dans une plateforme conçue au sein du laboratoire. Le deuxième but est d'intégrer la méthode dans la chaîne de traitements afin d'automatiser toute la chaîne de traitements.

## 2 État de l'art

Après cette présentation du contexte, nous allons maintenant rentrer dans le vif du sujet en clarifiant les concepts de base et en analysant les algorithmes et solutions existants, applicables à notre projet.

Étant donné que la complétion et la reconstruction de maillages nécessitent plusieurs étapes de traitements, nous avons dans un premier temps étudié les méthodes de représentation de surfaces, les concepts existants en informatique pour modéliser un objet. Cela nous permet de mieux comprendre les objets que

nous allons traiter et de mieux choisir les algorithmes à adapter. Puis, nous avons découpé l'étude de l'existant selon les différents types de traitement :

- Génération de maillages ;
- Complétion de maillages ;
- Lissage.

## 2.1 Représentation de surfaces

Dans les applications médicales, l'obtention des données à partir d'IRM est indispensable pour avoir les modèles volumiques des objets géométriques. Néanmoins mon sujet de stage se situe au milieu de la chaîne de traitement, donc se limite à traiter les modèles surfaciques des objets 3D. Par conséquent, la partie de l'état de l'art de la représentation des surfaces sera vue de façon plus précise. Cette partie se base principalement sur le contenu de l'ouvrage de Mario Botsch et al. [3]. D'un point de vue général, il existe deux grandes catégories de représentation de surfaces : **paramétrique** et **implicite**.

**Surface paramétrée** Une surface paramétrée est définie par une fonction de paramétrisation :  $\mathbf{f} : \Omega \rightarrow \mathcal{S}$ , où  $\Omega \subset \mathbb{R}^2$  est un domaine 2D et  $\mathcal{S} \subset \mathbb{R}^3$ . Ainsi, chaque point sur  $\mathcal{S}$  est relié par un couple  $(u, v) \in \Omega \subset \mathbb{R}^2$ . Comme illustration considérons une sphère ayant pour rayon  $r$  et pour centre l'origine. Cette sphère peut être définie par une fonction :

$$\mathbf{f} : [0, 2\pi] \times [-\pi/2, \pi/2] \rightarrow \mathbb{R}^3, (\theta, \phi) \rightarrow \begin{pmatrix} x(\theta, \phi) = r \cos \theta \cos \phi \\ y(\theta, \phi) = r \sin \theta \cos \phi \\ z(\theta, \phi) = r \sin \phi \end{pmatrix}.$$

L'avantage de cette représentation est que nous pouvons réduire de nombreux problèmes 3D sur la surface  $\mathcal{S} \subset \mathbb{R}^3$  aux problèmes 2D sur le domaine  $\Omega \subset \mathbb{R}^2$ . Par exemple, les voisins sur la surface  $\mathcal{S}$  sont faciles à trouver en prenant les voisins sur le domaine  $\Omega$ . Une simple composition de  $\mathbf{f}$  et une fonction de déformation  $\mathbb{R}^3 \rightarrow \mathbb{R}^3$  aboutit à des modifications efficaces de la géométrie de surface. D'autre part, la paramétrisation de surface peut devenir très compliquée, d'autant plus que la topologie de la surface peut être changée. Nous rappelons ici certaines surfaces particulières : les surfaces de révolution (comme les cylindres), les surfaces réglées (si et seulement si par tout point il passe au moins une droite entièrement contenue dans la surface). A part ces types de surfaces paramétrées, il faut mentionner aussi les surfaces se prêtant à une définition par produit tensoriel et par les fonctions B-Splines (en anglais NURBS), qui sont les plus utilisées en pratique, notamment dans les systèmes de CAO. Pour une lecture approfondie, se référer à l'ouvrage de Pascal Jean Frey [12] et à [4].

*Maillage triangulé* Le maillage triangulé (notre modèle principal) peut aussi être considéré comme une paramétrisation de surface, qui n'est pas seulement une collection de triangles. Cela peut générer une paramétrisation de surfaces par morceaux. Chaque point  $\mathbf{p}$  à l'intérieur de triangle ( $\mathbf{a}, \mathbf{b}, \mathbf{c}$ ) peut alors être écrit

par une combinaison des trois sommets :  $\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$ , avec  $\alpha + \beta + \gamma = 1$  et  $\alpha, \beta, \gamma \geq 0$ . A partir de cette paramétrisation locale, nous pouvons concevoir une paramétrisation de surface en 3D basée sur le maillage triangulé en 2D [12].

**Surface implicite** Une surface implicite est présentée par une isosurface. Soit  $\mathbf{F}$  une fonction en valeur scalaire  $\mathbf{F} : \mathbb{R}^3 \rightarrow \mathbb{R}$ , alors  $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 | \mathbf{F}(\mathbf{x}) = 0\}$ . Reprenons l'exemple de la sphère comme illustration : une sphère ayant pour rayon  $r$  et pour centre l'origine, peut être définie par une isosurface associée à la fonction :

$$\mathbf{F} : \mathbb{R}^3 \rightarrow \mathbb{R}, (x, y, z) \rightarrow x^2 + y^2 + z^2 - r^2.$$

Généralement, la valeur négative de  $\mathbf{F}$  désigne les points à l'intérieur de l'objet, celle positive désigne les points à l'extérieur de l'objet. De plus, la surface évite les trous et les intersections. Par conséquent, la représentation implicite se prête à la géométrie solide constructive (CSG) où les objets complexes sont construits par les opérations Booléennes appliquées aux primitives géométriques. En revanche, elle n'est pas favorable pour les algorithmes comme trouver les voisins ou gérer les textures de surfaces. Il existe un algorithme très connu dit « Marching Cube » [10] qui permet de convertir la représentation implicite ou volumétrique en maillage triangulaire, notamment pour les modélisations CSG et les applications médicales.

## 2.2 Maillages

Dans cette partie, nous allons étudier le maillage, notamment le maillage surfacique traité dans notre application. Cette partie de mon rapport se base fortement sur l'état de l'art proposé par Julien Dardenne dans son manuscrit de thèse [8].

Un maillage est un ensemble de sommets reliés entre eux par des arêtes. Plusieurs caractéristiques les distinguent :

- Si les sommets sont tous situés dans un même plan, le maillage sera 2D. Dans le cas contraire, ce sera un maillage 3D. Dans notre cas, un maillage surfacique est plutôt un maillage 3D selon cette définition mais c'est un maillage 2D manifold (voir Section 4.1) intégré dans  $\mathbb{R}^3$ , qui représente une surface au sens géométrique.
- Les éléments du maillage peuvent être de différents degrés. Si tous les éléments sont des triangles, on parle alors de maillage triangulaire. Si tous les éléments sont des tétraèdres, le maillage est dit tétraédrique. L'intérieur de tout élément du maillage est non vide. L'intersection de l'intérieur de deux éléments est vide.
- La connectivité d'un maillage définit le type de connexions entre ses éléments. Un maillage est **structuré** si sa connectivité est fixe et identique pour chaque élément du maillage, par exemple, une grille. Un maillage est **non-structuré** si sa connectivité est quelconque, par exemple, la plupart des maillages triangulaires.

La représentation d'un maillage contient deux types d'information :

- des informations géométriques : les coordonnées des sommets
- des informations topologiques concernant la connectivité du maillage c'est à dire la description des éléments du maillage, de laquelle peuvent être déduites toutes les relations de voisinage entre les sommets.

### 2.3 Structure de données

En vue de représenter les informations géométriques et topologiques d'un maillage, différentes structures de données ont été proposées durant ces dernières années [3]. En parlant de la structure de données, outre les informations stockées, nous devons aussi considérer l'efficacité des algorithmes.

**Structures basées sur des faces** La plus simple idée est d'enregistrer toutes les faces avec les positions de leurs sommets. L'inconvénient est que cette structure de données ne contient pas d'informations concernant la connectivité. Mais elle est simple et efficace à stocker, donc c'est pourquoi elle est souvent utilisée dans les fichiers comme .OFF, .OBJ, etc. Il existe aussi une structure dérivée : pour chaque face, ses sommets et ses voisins (faces) sont enregistrés, et pour chaque sommet, une de ses faces associées est enregistrée (Figure 3).

**Structures basées sur des arêtes** Pour faciliter la recherche des informations de connectivités, des structures de données basées sur des arêtes ont été proposées. Cela est logique car les arêtes sont les intersections de faces, d'où la connectivité. La plus connue des structures est appelée « Winged-edge » [14]. Pour chaque arête, on enregistre ses deux extrémités (sommets), ses deux faces adjacentes, ainsi que ses arêtes précédente et suivante dans ces faces. Finalement, le parcours des sommets et des arêtes est plus rapide grâce aux informations de connectivités (Figure 4(a)).

**Structure basées sur demi-arêtes** C'est une version améliorée de la structure "Winged edge" qui évite de vérifier si un sommet est le premier ou le second d'une arête [3] [19]. La méthode divise chaque arête non-orientée en deux demi-arêtes orientées de façon opposée. Elle peut représenter les maillages orientables et 2-manifold ( voir Section 4.1). Dans cette structure, les demi-arêtes associées à une même face sont orientées dans le sens inverse des aiguilles d'une montre. Pour chaque demi-arête, on enregistre le sommet vers lequel elle pointe, la face qui lui est associée, sa demi-arête opposée, ainsi que la demi-arête précédente et suivante (Figure 4(b)).

### 2.4 Génération de maillages

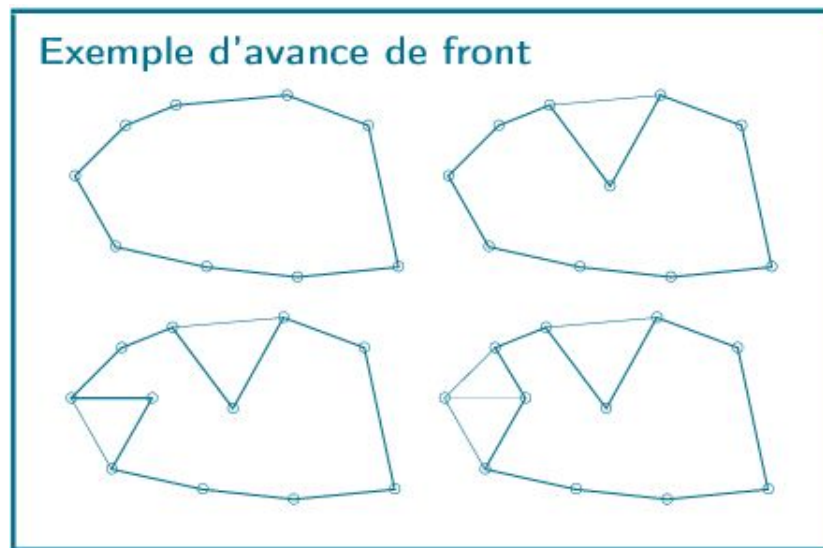
Dans cette partie, nous donnerons deux méthodes existantes connues de génération de maillages non structurés. Elles consistent à utiliser des maillages



composés de simplexes et sont entièrement automatiques. La description de ces deux méthodes se base fortement sur le contenu de l'ouvrage de Pascal Jean Frey [12].

**Méthodes frontales** L'idée de base de cette méthode consiste à construire le maillage élément par élément, à partir d'un front initial (frontière de l'objet à mailler, données d'entrée du problème). Cette technique procède par création et insertion de points (ou en s'appuyant sur des points déjà créés) et en connectant ces points à des entités du front pour former des éléments, ainsi le front est mis à jour progressivement.

*Schéma général* (extrait de [12], Figure 5). Cette méthode s'effectue en plusieurs étapes :



**FIGURE 5.** Progression du maillage par la méthode frontale : front initial (à gauche en haut), un point adéquat inséré (à droite en haut), un point existant est le point souhaité (à droite en bas). (Image extraite de [6])

1. Initialisation du front avec les entités composant la frontière du domaine, triées éventuellement selon un certain critère.
2. Définition de la fonction de distribution de tailles.
3. Sélection d'une entité du front. Ceci revient à :
  - créer un point adéquat  $P$  à partir de cette entité,
  - déterminer si un sommet  $V$  du maillage courant pourrait être utilisé au lieu de  $P$ . Si un tel point existe, considérer  $V$  comme le point  $P$  adéquat,

- valider ce choix au regard de la géométrie (détecter les intersections) et de la taille souhaitée,
  - et lorsqu'un point adéquat est identifié, mettre à jour le maillage et le front ;
4. Tant que le front n'est pas vide, un retour en 3.
  5. Optimisation éventuelle du maillage.

Avec cette méthode, nous pouvons plus facilement garantir la taille ou la forme de l'élément créé. Par contre, la détection des intersections n'est pas toujours très facile. De plus, cette méthode est délicate à implémenter en 3D et entraîne des problèmes de convergence.

**Méthodes de type Delaunay** La triangulation de Delaunay a été étudiée par le mathématicien Boris Delaunay. Afin de mieux comprendre les propriétés géométriques de cette triangulation, nous commençons par rappeler certaines définitions.

**Definition 1 (Cellules de Voronoï).** *Donnons un ensemble ouvert  $\Omega$  de  $\mathbb{E}^d$ , et  $n$  points  $\{P_0, P_1, \dots, P_{n-1}\}$ . Les  $n$  cellules de Voronoï  $\{V_0, V_1, \dots, V_{n-1}\}$  dont chacune est attachée à un point peuvent être définies, de telle sorte que tout point de  $V_i$  soit plus proche de  $P_i$  que tout autre point de l'ensemble :*

$$V_i = \{P \in \Omega \mid \|P - P_i\|_d \leq \|P - P_j\|_d, \forall j \neq i\}.$$

*Les cellules de Voronoï sont des polyèdres convexes.*

**Definition 2 (Triangulation de Delaunay).** *La triangulation de Delaunay est le dual du diagramme de Voronoï, qui est obtenu en reliant  $P_i P_j$  si et seulement si  $V_i$  et  $V_j$  sont adjacentes dans le diagramme de Voronoï. Ceci peut être étendue en 3D (avec tétraèdres).*

**Definition 3 (Critère de Delaunay).** *Dans une triangulation de Delaunay, la boucle ouverte circonscrite de chaque élément ne contient aucun autre point (sommet de triangle).*

La partie suivante est extraite de l'ouvrage de Pascal Jean Frey [12].

*Schéma général* Cette méthode s'effectue en plusieurs étapes :

1. Initialisation :
  - discrétisation de la frontière,
  - construction d'une triangulation initiale  $\mathcal{T}_B$  de la boîte englobante du domaine  $\Omega$ .
2. Insertion des points frontières dans  $\mathcal{T}_B$ .
3. Construction d'un maillage vide  $\mathcal{T}_e$  à partir de  $\mathcal{T}_B$  :
  - recouvrement des entités de frontière manquantes,
  - identification des composants connexes du domaine.

4. Création et insertion des points internes.
5. Optimisation de  $\mathcal{T}$ .

**Definition 4 (Noyau de Delaunay).** *L'insertion de  $P = P_{i+1}$  dans la triangulation de Delaunay  $\mathcal{T}_i$  avec les noeuds  $\{P_0, \dots, P_i\}$  peut s'écrire de la façon suivante :*

$$\mathcal{T}_{i+1} = \mathcal{T}_i - \mathcal{C}_P + \mathcal{B}_P,$$

où  $\mathcal{B}_P$  représente l'ensemble des éléments formés en joignant le point  $P$  aux arêtes externes de l'ensemble  $\mathcal{C}_P$  des simplexes dont les boucles circonscrites contiennent  $P$ .

La triangulation de Delaunay est favorable pour certains algorithmes genre localisation des points dans un domaine. En effet, les idées sur la triangulation de Delaunay permettent de développer des méthodes de construction de maillages rapides et efficaces. En revanche, en trois dimensions le critère de Delaunay est insuffisant pour garantir un maillage de qualité.

## 2.5 Lissage

Avant d'aborder la procédure de lissage, on a souvent besoin de raffiner le maillage grossier. Il existe plusieurs catégories de méthodes de remaillage et de raffinement : méthodes glouton, variationnelle et incrémentale [3] [15]. Nous nous intéressons ici à un algorithme incrémental en effectuant des modifications locales sur le maillage. Cela veut dire qu'à chaque itération, on sélectionne des éléments candidats, ensuite on effectue le raffinement par élément (par exemple, subdivision par sa plus grande arête [21] [22]), puis on propage la subdivision à certains de ses voisins.

Les méthodes de lissage consistent à prendre en compte les propriétés géométriques de surface. Nous commençons par introduire les propriétés de surfaces dans le cadre de la géométrie différentielle. Soit

$$\mathbf{x}(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix}, (u, v) \in \Omega \subset \mathbb{R}^2$$

une surface paramétrée par la fonction  $\mathbf{x}$ . Alors la métrique de surface est déterminée par les dérivées de  $\mathbf{x}$ . Nous utiliserons les notations suivantes :

$$\mathbf{x}_u = \frac{\partial \mathbf{x}}{\partial u}, \mathbf{x}_{uu} = \frac{\partial^2 \mathbf{x}}{\partial u^2}, \mathbf{x}_{uv} = \frac{\partial^2 \mathbf{x}}{\partial u \partial v}.$$

La normale à un point de la surface devient :  $\mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|}$ . On appelle l'opérateur suivant la première forme fondamentale ou le tenseur métrique :

$$\mathbf{I} = \begin{bmatrix} \mathbf{x}_u^T \mathbf{x}_u & \mathbf{x}_u^T \mathbf{x}_v \\ \mathbf{x}_u^T \mathbf{x}_v & \mathbf{x}_v^T \mathbf{x}_v \end{bmatrix}.$$

Cet opérateur permet de mesurer les valeurs comme la longueur de courbe, l'aire de surface, etc. On appelle l'opérateur suivant la seconde forme fondamentale qui est lié plutôt à la courbure à un point :

$$\mathbf{II} = \begin{bmatrix} \mathbf{x}_{uu}^T \mathbf{n} & \mathbf{x}_{uv}^T \mathbf{n} \\ \mathbf{x}_{uv}^T \mathbf{n} & \mathbf{x}_{vv}^T \mathbf{n} \end{bmatrix}.$$

Le problème de lissage consiste à minimiser une fonction décrivant la propriété géométrique de la surface, nommée fonction d'énergie. Le lissage par l'énergie de surface a été proposé par Kobbelt *et al.* [16] [17] [25]. Les deux principales énergies employées sont l'énergie de membrane et l'énergie de plaque définies par :

$$E_M(\mathbf{f}) = \int \mathbf{f}_u^2 + \mathbf{f}_v^2,$$

$$E_P(\mathbf{f}) = \int \mathbf{f}_{uu}^2 + 2\mathbf{f}_{uv}^2 + \mathbf{f}_{vv}^2,$$

où  $\mathbf{f}$  est une paramétrisation de surface (dans le cas du maillage triangulaire, elle se limite à une paramétrisation locale autour de sommets). Cette paramétrisation est construite par les polynômes d'interpolations en utilisant les sommets voisins. Donc du point de vue discrète, les valeurs  $\mathbf{f}_u, \mathbf{f}_v, \mathbf{f}_{uu}, \mathbf{f}_{uv}, \mathbf{f}_{vv}$  à chaque point de surface devient une combinaison linéaire des sommets. Nous pouvons ensuite convertir la minimisation de l'énergie en la résolution de l'équation aux dérivées partielles :  $\Delta \mathbf{f} = \mathbf{f}_{uu} + \mathbf{v}\mathbf{v} = 0$  et  $\Delta^2 \mathbf{f} = \mathbf{f}_{uuuu} + 2\mathbf{f}_{uuvv} + \mathbf{f}_{vvvv} = 0$ . Après discrétisation, ceci se transforme en un système linéaire à résoudre.

## 2.6 Complétion de maillages

La reconstruction de surfaces, quelque soit les sources de données qu'on prenne, ne peut pas éviter les déficiences dans le maillage construit. Nous allons dans cette partie donner un panorama sur les méthodes existantes de complétion de maillages surfaciques (en anglais, *hole filling*). Le trou peut être de petites lacunes irrégulières ou de grandes parties manquantes, et peut provenir de l'algorithme ou des données originales. Dans notre cas, les trous du maillage correspondent aux informations manquantes dans les données médicales initiales.

Nous ne considérons ici que les approches basées sur la surface, et non celles basées sur les volumes. Selon l'idée de Liepa [18], nous distinguons alors deux catégories de méthodes : approches non-géométrique et géométrique.

**Approche non-géométrique** Carr *et al.* [7] ont proposé une méthode en utilisant les fonctions radiales (Radial basis functions - RBF) pour interpoler la partie manquante dans le maillage. Zhao *et al.* [26] ont utilisé l'équation de Poisson pour calculer les positions de points interpolés. Ils ont dans un premier pas utilisé la méthode d'avancée de fronts (voir Section 2.4) afin de compléter le maillage initial avec un patch adéquat. Puis les normales désirées ont été calculées avec une fonction harmonique. Finalement, ils ont résolu une équation de

poisson pour remailler la surface. Ces méthodes non-géométriques s'avèrent très efficaces pour traiter les trous complexes et irréguliers. Mais la définition des fonctions et l'implémentation de l'algorithme ne sont pas très évidents.

**Approche géométrique** L'approche géométrique la plus représentative est celle de Liepa [18] qui a divisé le problème en trois étapes : triangulation, raffinement et lissage final. Dans chaque étape, il manipule les arêtes et les faces d'une façon géométrique intuitive. Il vise à compléter les trous arbitraires dans un maillage quelconque. Notons que l'algorithme n'est pas très efficace en terme de complexité, avec par exemple une complexité en  $\mathcal{O}(n^3)$  pour la procédure de triangulation.

**Complétion basée sur exemples** Plusieurs recherches ont été faites sur le problème de complétion par l'exemples, c'est-à-dire la complétion du maillage initial en tenant compte pas seulement du maillage lui-même mais aussi d'autres conditions. Pauly *et al.* [24] ont proposé dans ce sens une méthode dite « Complétion basée sur exemples ». Les maillages à compléter ont été construits à partir de 3D Scanner et les exemples correspondent à des modèles d'objets (maillages) présents dans une base de données. Cette méthode consiste ainsi à compléter le maillage en utilisant les objets connus. Par exemple, considérons le maillage d'un pot abîmé. En le comparant aux modèles existants, on peut savoir que ce n'est pas un verre et qu'il a une poignée, ainsi on a des informations précises pour compléter le maillage. Pour cela, des objets candidats sont sélectionnés dans la base de données (notons que pour cette étape, l'estimation d'erreur entre le modèle et le maillage a été définie). Puis un recalage non-rigide [2] est appliqué aux modèles existants pour qu'ils puissent mieux coller au maillage à traiter. Les parties utiles des modèles déformés sont ensuite segmentées. La dernière chose étape consiste alors à intégrer ces parties au maillage à compléter.

### 3 Contributions

Dans le cadre de ce travail, nous cherchons un algorithme de complétion de maillages applicable à notre cas des images médicales. Par ailleurs, il doit être générique pour que nous puissions intégrer nos contraintes spécifiques. Par rapport au cahier de charge, nous avons décidé d'utiliser l'approche proposée par Liepa [18] comme algorithme de base. Les justifications sont les suivantes :

1. L'approche de Liepa est une approche géométrique, intuitive et simple, mais elle correspond bien à notre besoin par rapport aux données d'entrée et le résultat qu'on veut.
2. Cette méthode ne change pas la structure initiale, cela veut dire qu'il n'y aura pas de changement sur la topologie de maillage.
3. De nombreuses méthodes non-géométriques sont conçues pour traiter des trous complexes et irréguliers, alors que dans notre cas, le trou correspond

à une partie d'information manquante dans l'image initiale. L'approche de Liepa est alors suffisante.

4. L'étape de complétion de maillages constitue une étape de pré-traitement dans le cadre de notre application (génération des données géométriques avant une simulation par exemple). Du coup la complexité ou l'efficacité de l'algorithme ne constitue pas un point critique.

En revanche, l'algorithme original de Liepa ne satisfait pas complètement à notre besoin, puisqu'il n'intègre pas la possibilité d'ajouter des contraintes pré-définies. Nous proposons ainsi de diviser la procédure de complétion en plusieurs étapes dont les trois premières sont celles de Liepa et les dernières correspondent à une déformation et des finitions. Ceci nous permet d'avoir dans un premier temps une surface complétée qui sera ensuite déformée pour qu'elle satisfasse aux contraintes pré-définies. Les autres contributions concernent l'algorithme :

1. Généralisation de l'opérateur *Umbrella* pour le lissage en calculant la formule avec une pondération quelconque.
2. Vérification de la convergence des méthodes de solution de systèmes linéaires dans notre cas.
3. Définition de l'algorithme de sélection de la zone à déformer.

## 4 Méthode de reconstruction

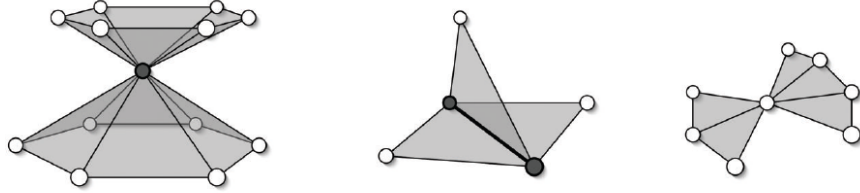
### 4.1 Maillage d'entrée

Comme l'indique le pipeline de traitement dans la section 1.1, notre méthode de reconstruction se limite à faire des traitements sur le maillage construit afin de pouvoir générer un maillage correct qui sera utilisé pour les étapes suivantes de notre application. Avant de présenter notre algorithme, il est nécessaire d'imposer quelques conditions au maillage d'entrée qui contient des parties manquantes (des trous entourés par les frontières du maillage). Ces critères sont vérifiés lors du pré-traitement.

**2-manifold** Un maillage triangulé surfacique est 2-manifold s'il ne contient ni arêtes non-manifolds, ni sommets non-manifolds, ni intersections. Et évidemment il n'y a pas de faces ouvertes ni d'éléments libres comme des arêtes ou des sommets [3].

**Arête non-manifold** Arête qui est commune à plus de deux éléments (triangles)(pour les arêtes frontières d'une surface ouverte, on ne trouve qu'un seul élément) (voir Figure 6).

**Sommet non-manifold** Sommet qui est commun à plus de deux arêtes frontières. Cela veut dire qu'il est commun à plus d'une « étoile » de triangles (voir Figure 6).



**FIGURE 6.** Deux surfaces se rencontrent au sommet non-manifold (gauche). Une arête non-manifold est commune à trois triangles (milieu). Un sommet non-manifold est commun à quatre arêtes frontières (droite) (images issues de [3]).

**Orienté** Un maillage est orienté si chaque élément est orienté de la même façon (voir Figure 7).

**Connecté** Un maillage est connecté s'il n'y a pas de parties isolées. Cela veut dire qu'il est vide à l'intérieur d'un trou entouré par des arêtes frontières (voir Figure 7).

## 4.2 Contraintes

Dans le cadre de ce travail, nous cherchons un algorithme qui complète le maillage brut en mettant l'accent sur des contraintes. Dans cette partie, nous allons clarifier ces contraintes imposées au maillage complété et reconstruit. Nous distinguons deux types de contraintes imposées au maillage : les contraintes locales et les contraintes extérieures.

**Contraintes locales** Les contraintes locales désignent les critères liés au maillage d'entrée, surtout les conditions aux frontières de celui-ci. Il est évident que le maillage reconstruit doit aussi respecter les critères de base appliqués au maillage d'entrée (voir Section 4.1). Puis nous désirons que la partie extrapolée soit similaire au maillage d'entrée, avec la même densité d'éléments et des éléments de forme similaire. De plus, la surface générée doit être aussi lisse et uniforme que possible. Cela implique une continuité de forme à la zone de transition aux frontières du maillage d'entrée (le contour de la partie manquante). Afin de respecter cette contrainte, les courbures des éléments autour de la frontière sont pris en compte. Dans les parties suivantes, nous allons interpréter et décrire ces contraintes mathématiquement.

**Contraintes extérieures** En plus des contraintes locales, nous pouvons également définir des contraintes extérieures. Contrairement aux contraintes locales, les contraintes extérieures n'ont souvent pas de relation avec le maillage d'entrée. Ces contraintes peuvent varier selon les utilisateurs et l'utilisation des modèles géométriques générés, c'est pour ça que nous ne pouvons pas les présenter

d'une manière globale. Dans le cadre de notre application, ces contraintes doivent permettre de répondre à différents besoins. Par exemple, nous désirons que le maillage complété ressemble à l'organe de référence issu d'un atlas anatomique. Pour cela, il existe des techniques qui maximisent la similarité entre un maillage et un exemple sous transformation rigide (dans le cas simple). Après il faut juste copier coller la surface correcte au maillage d'entrée. En outre, nous désirons aussi gérer des contacts avec d'autres organes. Pour cela, nous devons définir une zone de contact et des points correspondants. D'un point de vue géométrie algorithmique, l'essentiel est de définir des points de passage quelque soit les contraintes qu'on prenne. Ainsi dans le cadre de ce travail, nous nous concentrons essentiellement sur ce type de contrainte extérieure, c'est-à-dire un ensemble de points de passage par lesquels doit passer la surface reconstruite.

### 4.3 Approche proposée

En vue de compléter le maillage avec les contraintes citées, notre approche procède en deux étapes. La première partie de notre approche vise à « boucher le trou » du maillage d'entrée. Cela veut dire qu'on génère un maillage fermé en complétant le maillage d'entrée de telle sorte qu'il satisfasse d'abord les contraintes locales. Puis la deuxième partie de l'approche consiste à modifier ce maillage généré afin qu'il puisse satisfaire les contraintes extérieures. Les traitements sont toujours appliqués aux parties extrapolées (les nouveaux éléments générés), ainsi le maillage initial (maillage d'entrée) ne change pas. Ainsi, cela revient à coller au maillage initial des nouveaux morceaux adaptés aux contraintes.

Pour la première partie, nous avons décidé d'utiliser une méthode basée sur celle de Liepa [18]. Cette méthode de réparation de maillages est une approche géométrique orientée surface. L'algorithme se compose de trois sous-étapes : triangulation, raffinement et lissage. Nous allons voir ces étapes dans les parties suivantes en donnant la solution adaptative à notre cas.

Pour la deuxième partie, nous avons besoin d'appliquer une déformation de surface à la partie ajoutée au maillage initial. Il existe pour cela des techniques pour déformer un maillage surfacique comme celle de Botsch [20]. Nous allons voir et adapter ces techniques selon notre spécification. Enfin, après la déformation effectuée, une procédure de remaillage est nécessaire afin d'obtenir un maillage final correct.

Le pipeline complet de notre méthode est illustré par la figure 8.

### 4.4 Triangulation

**Identification des trous** La première étape avant tout est d'identifier les trous dans le maillage d'entrée, trous correspondant aux parties manquantes de l'image médicale. Nous enregistrons ensuite pour chaque trou les arêtes de son contour



(qui sont aussi les arêtes frontières du maillage d'entrée) et les sommets associés. L'idée est simple : parcourir toutes les arêtes pour trouver une arête frontière, l'enregistrer, trouver l'arête qui la suit jusqu'à ce que l'on ait une boucle fermée. Ainsi, on peut identifier un trou. Il faut juste répéter cette procédure jusqu'à il n'y a plus d'arêtes frontières à identifier.

**Triangulation du trou** Dans cette partie, nous présentons l'algorithme pour trianguler un trou [18]. Soit  $V = \{v_0, v_1, \dots, v_{n-1}\}$ ,  $v_i \in \mathbb{R}^3$  la liste des sommets appartenant au contour du trou. Soit  $P = (v_0, v_1, \dots, v_{n-1})$  le polygone associé qui présente le trou. Il faut remarquer que le polygone est un polygone 3D au sens topologique. Cela veut dire que ses sommets ne sont pas forcément sur le même plan. L'idée est de concevoir une façon de mesurer la qualité de chaque triangle (triple de sommets), ainsi nous pouvons mesurer la qualité totale après la triangulation. En optimisant cette qualité, nous obtenons la triangulation souhaitée. Soit  $L$  l'ensemble des valeurs qui présentent la qualité.  $L$  est muni d'un élément neutre  $0_L$  et de l'opération d'addition. On définit maintenant une fonction de qualité  $\Omega : V^3 \rightarrow L$ , qui associe à chaque triangle dont les sommets sont dans  $V$  une valeur de qualité. Soit maintenant  $Q_{i,j}$ ,  $0 \leq i \leq j < n$  la qualité totale optimale du polygone  $P_{i,j} = (v_i, \dots, v_j)$  après triangulation car il existe une seule triangulation du polygone qui optimise la qualité totale. Nous calculons la qualité optimale de  $P$  par la méthode incrémentale. (Voir Algorithme 1)

---

**Algorithme 1:** Triangulation du trou

---

```

Data :  $P_{0,n-1} = (v_0, v_1, \dots, v_{n-1})$ 
begin
  for  $i \leftarrow 0$  to  $n - 2$  do
     $Q_{i,i+1} \leftarrow 0_L$ 
  for  $i \leftarrow 0$  to  $n - 3$  do
     $Q_{i,i+2} \leftarrow \Omega(v_i, v_{i+1}, v_{i+2})$ 
  for  $j \leftarrow 3$  to  $n - 1$  do
    for  $i \leftarrow 0$  to  $n - j - 1$  do
       $k \leftarrow i + j$ 
       $Q_{i,k} \leftarrow \min_{i < m < k} \{Q_{i,m} + Q_{m,k} + \Omega(v_i, v_m, v_k)\}$ 
       $\lambda_{i,k} \leftarrow m$  (l'indice  $m$  avec lequel  $Q_{i,k}$  atteint la valeur
      minimum)
    Triangler( $P_{0,n-1}$ )
    // Fonction récursive Triangler permet de trianguler le
    // polygone en utilisant les indices  $\lambda_{i,k}$ . Voici la
    // définition:
    Triangler( $P_{i,j}$ ) {
      CreerTriangle( $i, \lambda_{i,j}, j$ )
      Triangler( $P_{i,\lambda_{i,j}}$ )
      Triangler( $P_{\lambda_{i,j},j}$ )
    }
  
```

---

Il ne reste qu'à définir l'ensemble  $L$  et la fonction de qualité  $\Omega$ . Selon Liepa [18], nous proposons aussi l'aire de surface comme mesure.

$$L = [0, \infty), 0_L = 0$$

$$\Omega(v_i, v_m, v_k) = \text{Aire du Triangle}(v_i, v_m, v_k)$$

Ainsi après l'étape de triangulation, nous optimisons la qualité totale en minimisant l'aire de la surface générée. On note ultérieurement  $\mathcal{T}_c$  cette première triangulation de la partie manquante,  $\mathcal{M}_c$  le maillage de trou complété et  $\mathcal{S}_c$  la surface générée.

#### 4.5 Raffinement

Une fois la première étape de triangulation terminée, nous traitons le cas où la demande de raffinement (d'enrichissement) est formulée au niveau des éléments (nouveaux triangles générés par la triangulation). Nous utilisons ici un algorithme incrémental en effectuant des modifications locales sur  $\mathcal{M}_c$ . Cette méthode est proposée dans [18]. Elle remaille  $\mathcal{M}_c$  en tenant compte des contraintes de densité et de forme. Le critère de sélection des éléments à raffiner est basé sur la longueur moyenne des arêtes autour d'un sommet. Ces valeurs sont ensuite propagées à l'intérieur de  $\mathcal{M}_c$  en subdivisant les triangles. Par élément, nous subdivisons le triangle et effectuons le cas échéant une bascule d'arête afin d'assurer une triangulation de Delaunay (voir Algorithme 2).

**Definition 5.** Soit  $\|v_i - v_j\|$  la longueur de l'arête avec les extrémités  $v_i$  et  $v_j$ . Notons que  $\sigma(v_i)$  est la longueur moyenne des arêtes autour de  $v_i$  qui appartiennent au maillage initial (voir Figure 9). Soit  $\alpha$  le coefficient qui contrôle la densité.

**Definition 6 (Bascule d'arête).** Une arête est dite illégale au sens de Delaunay si le cercle circonscrit de l'un de ses triangles adjacents contient le troisième sommet de l'autre. Dans ce cas, il faut basculer l'arête pour qu'elle vérifie la condition de Delaunay. La Figure 10 illustre le cas en 2D. Notons qu'il est possible d'étendre la définition en 3D en utilisant la sphère circonscrite.

**Algorithme 2:** Raffinement de  $\mathcal{M}_c$ 


---

```

Data :  $\mathcal{T}_c$ 
Result :  $\mathcal{T}_c$  et  $\mathcal{M}_c$ 
begin
  for  $i \leftarrow 0$  to  $n - 1$  do
    | Calculer  $\sigma(v_i)$ 
  while Raffinement continue do
    | // Subdiviser les triangles
    repeat
      |  $\mathcal{T}_c.isChanged \leftarrow false$ 
      foreach  $triangle(v_i, v_j, v_k) \in \mathcal{T}_c$  do
        | // Calculer le barycentre  $v_c$ 
        |  $v_c \leftarrow (v_i + v_j + v_k)/3$ 
        |  $\sigma(v_c) \leftarrow (\sigma(v_i) + \sigma(v_j) + \sigma(v_k))/3$ 
        | // Vérifier s'il est nécessaire de raffiner
        | if  $\exists m \in \{i, j, k\} : (\alpha \|v_c - v_m\| >$ 
        |  $\sigma(v_c) \text{ and } \alpha \|v_c - v_m\| > \sigma(v_m))$  then
          | Subdiviser  $triangle(v_i, v_j, v_k)$  en trois :
          |  $triangle(v_c, v_j, v_k), (v_i, v_c, v_k), (v_i, v_j, v_c)$ 
          | if  $arête(v_i, v_j).isBorder = false$  then Basculer
          |  $arête(v_i, v_j)$ 
          | if  $arête(v_i, v_k).isBorder = false$  then Basculer
          |  $arête(v_i, v_k)$ 
          | if  $arête(v_j, v_k).isBorder = false$  then Basculer
          |  $arête(v_j, v_k)$ 
          |  $\mathcal{T}_c.isChanged \leftarrow true$ 
        |
      until  $\mathcal{T}_c.isChanged = false$ 
      | // Basculer les arêtes le cas échéant
      repeat
        |  $\mathcal{T}_c.isChanged \leftarrow false$ 
        | foreach  $arête(v_i, v_j) \in \mathcal{T}_c$  do
          | if  $arête(v_i, v_j).isBorder = false$  then Basculer  $arête(v_i, v_j)$ 
          |  $\mathcal{T}_c.isChanged \leftarrow true$ 
        |
      until  $\mathcal{T}_c.isChanged = false$ 

```

---

**4.6 Lissage**

Jusqu'à maintenant, un maillage bien fermé et raffiné est généré. Le traitement continue à modifier  $\mathcal{M}_c$  pour obtenir une surface  $\mathcal{S}_c$  lisse, et surtout arrondir le changement de forme au niveau du contour de  $\mathcal{M}_c$  et le maillage initial. Souvent la procédure de lissage consiste à minimiser une énergie fonctionnelle. Ce genre de méthodes implicites existe depuis plusieurs années et elles sont tirées des idées de [25] [17] [9]. Notre algorithme utilise les concepts de [25]

en généralisant ses opérateurs en vue d'améliorer le  $\mathcal{M}_c$  modifié dans la section 4.5.

Nous minimisons ainsi l'énergie de plaques définie par :

$$\mathcal{E}_{TP}(f) = \int f_{uu}^2 + 2f_{uv}^2 + f_{vv}^2$$

Pour cela il suffit de résoudre l'équation aux dérivées partielles (voir section 2) :

$$\Delta^2 f = f_{uuuu} + f_{uuvv} + f_{vvvv} = 0$$

Cela peut aussi être considérée comme une contrainte locale (section 4.2). Notons que  $f$  est une paramétrisation de la surface, en revanche le  $\mathcal{M}_c$  qu'on traite est un maillage, c'est-à-dire une représentation discrète de surface, donc il faut discrétiser cet opérateur Laplacien. Par ailleurs, dans ce cas, la valeur de  $f$  représente simplement la position du sommet, ainsi résoudre  $f$  signifie recalculer les positions des sommets.

*Opérateur « Umbrella »* Soit  $p$  un sommet quelconque à l'intérieur de  $\mathcal{M}_c$ .  $\mathcal{N}(p)$  est l'ensemble de ses voisins directs. La discrétisation de l'opérateur  $\Delta$  est définie par cet opérateur "Umbrella" :

$$\mathcal{U}(p) = \left(\frac{1}{\omega(p)}\right) \sum_{p_i \in \mathcal{N}(p)} \omega(p, p_i) p_i - p$$

Ainsi, l'opérateur  $\Delta^2$  devient :

$$\mathcal{U}^2(p) = \left(\frac{1}{\omega(p)}\right) \sum_{p_i \in \mathcal{N}(p)} \omega(p, p_i) \mathcal{U}(p_i) - \mathcal{U}(p)$$

où  $\omega(p) = \sum_{p_i \in \mathcal{N}(p)} \omega(p, p_i)$  et le coefficient de pondération  $\omega(p, p_i) = \|p - p_i\|$ . Ici nous utilisons l'opérateur « Umbrella » au second degré, ainsi on ne tient pas seulement compte de l'influence des voisins directs mais aussi de celle des voisins en second ordre. Par conséquent, la continuité de forme au contour est mieux assurée.

*Système linéaire* La contrainte  $\Delta^2 f = 0$  est interprétée comme suivant : pour tout  $v_i$  à l'intérieur de  $\mathcal{M}_c$  (pas sur le contour),

$$\mathcal{U}^2(v_i) = 0$$

Ainsi la résolution de l'équation aux dérivées partielles  $\Delta^2 f = 0$  est convertie en celle d'un système linéaire dont chaque ligne correspond à l'équation ci-dessus. On calcule le coefficient de  $v_i$  à la ligne  $i$  pour la méthode de résolution itérative.

$$\mathcal{U}^2(v_i) = \left(\frac{1}{\omega(v_i)}\right) \sum_{v_{ij} \in \mathcal{N}(v_i)} \omega(v_i, v_{ij}) \mathcal{U}(v_{ij}) - \mathcal{U}(v_i) = 0$$

$$\begin{aligned}
& \left( \frac{1}{\omega(v_i)} \sum_{v_{ij} \in \mathcal{N}(v_i)} \omega(v_i, v_{ij}) \underbrace{\left( \frac{1}{\omega(v_{ij})} \sum_{v_{ijk} \in \mathcal{N}(v_{ij})} \omega(v_{ij}, v_{ijk}) v_{ijk} \right)}_{\mathcal{U}(v_{ij})} - v_{ij} \right) \\
& - \underbrace{\left( \frac{1}{\omega(v_i)} \sum_{v_{ij} \in \mathcal{N}(v_i)} \omega(v_i, v_{ij}) v_{ij} \right)}_{\mathcal{U}(v_i)} = 0 \\
& \frac{1}{\omega(v_i)} \sum_{v_{ij} \in \mathcal{N}(v_i)} \left( \frac{\omega(v_i, v_{ij})}{\omega(v_{ij})} \sum_{v_{ijk} \in \mathcal{N}(v_{ij})} \omega(v_{ij}, v_{ijk}) v_{ijk} \right) \\
& - \left( \frac{2}{\omega(v_i)} \sum_{v_{ij} \in \mathcal{N}(v_i)} \omega(v_i, v_{ij}) v_{ij} \right) + v_i = 0 \\
& \frac{1}{\omega(v_i)} \sum_{v_{ij} \in \mathcal{N}(v_i)} \left( \frac{\omega(v_i, v_{ij})}{\omega(v_{ij})} \omega(v_i, v_{ij}) v_i + \frac{\omega(v_i, v_{ij})}{\omega(v_{ij})} \sum_{v_{ijk} \neq v_i} \omega(v_{ij}, v_{ijk}) v_{ijk} \right) \\
& - \left( \frac{2}{\omega(v_i)} \sum_{v_{ij} \in \mathcal{N}(v_i)} \omega(v_i, v_{ij}) v_{ij} \right) + v_i = 0 \\
& \left( 1 + \frac{1}{\omega(v_i)} \sum_{v_{ij} \in \mathcal{N}(v_i)} \frac{\omega(v_i, v_{ij})^2}{\omega(v_{ij})} \right) v_i + \underbrace{\dots\dots}_{\text{combinaison linéaire sans le terme } v_i} = 0
\end{aligned}$$

D'où le coefficient du terme  $v_i$  à la ligne  $i$  :

$$C_{ii} = 1 + \frac{1}{\omega(v_i)} \sum_{v_{ij} \in \mathcal{N}(v_i)} \frac{\omega(v_i, v_{ij})^2}{\omega(v_{ij})}$$

Maintenant nous résolvons le système linéaire par la méthode de Jacobi ou de Gauss-Seidel. On obtient à la  $n^{\text{ime}}$  itération :

$$\begin{aligned}
C_{ii} v_i^{(n+1)} + \mathcal{U}^2(v_i^{(n)}) - C_{ii} v_i^{(n)} &= 0, \quad 0 \leq i \leq n-1 \\
v_i^{(n+1)} &= v_i^{(n)} - \frac{1}{C_{ii}} \mathcal{U}^2(v_i^{(n)})
\end{aligned}$$

Cela permet de calculer explicitement les composantes  $v_i^{(n+1)}$  à partir des composantes  $v_i^{(n)}$ . Si on choisit une pondération uniforme, c'est-à-dire  $\omega(v_i, v_{ij}) = 1$ , on retrouve la même forme que [17].

*Remarque* Vu que dans notre algorithme  $\omega(v_i, v_{ij})$  n'est pas une constante, puisqu'il varie à chaque itération selon les positions des sommets  $v_i$  (pour  $0 \leq i \leq n-1$ ), nous n'avons pas un système linéaire au sens strict. Néanmoins, nous pouvons le traiter comme un système linéaire dont les coefficients  $\omega$  n'ont pas de grandes influences sur la résolution. Par contre, pour que le résultat puisse converger, il faut appliquer la méthode de Gauss-Seidel car on tient compte, au fur-et-à-mesure, des valeurs  $v_i^{(n+1)}$  déjà calculées pour obtenir  $\omega$ .

## 4.7 Déformation

Nous appliquons la déformation de surface sur le  $\mathcal{M}_c$  modifié. Notons que c'est lors de cette étape de déformation que nous pouvons ajouter nos contraintes extérieures définies par un ensemble de points de passage.

**Modèle mathématique** D'abord, nous rappelons les modèles mathématiques de déformation [3].

**Definition 7 (Déformation de surface).** *La déformation de la surface initiale  $\mathcal{S}$  à la surface déformée  $\mathcal{S}'$  est décrite par une fonction de déplacement  $d : \mathcal{S} \rightarrow \mathbb{R}^3$ . On associe à chaque point  $p \in \mathcal{S}$  un vecteur de déplacement  $\overrightarrow{d(p)}$ . D'où  $\mathcal{S}' = \{p + \overrightarrow{d(p)} | p \in \mathcal{S}\}$ .*

Si on veut appliquer la déformation sur certaines zones (points), on peut pré-définir  $\overrightarrow{d(p)}$  pour un ensemble de points  $\{p | p \in \mathcal{H} \subset \mathcal{S}\}$ , donc  $\mathcal{H}$  est une zone de contraintes. De même façon, nous pouvons définir une partie fixe de la surface initiale  $\mathcal{F} \subset \mathcal{S}$ . Nous avons donc : (illustré par la Figure 11).

$$\begin{cases} \overrightarrow{d(p)} = \overrightarrow{d_p} \quad \forall p \in \mathcal{H} \\ \overrightarrow{d(p)} = \overrightarrow{0} \quad \forall p \in \mathcal{F} \end{cases}$$

Ensuite nous allons gérer la partie restante de  $\mathcal{S} : \mathcal{R} = \mathcal{S} \setminus (\mathcal{H} \cup \mathcal{F})$  afin qu'elle se déforme d'une manière intuitive et physique. Cela va donner une bonne transition entre la partie  $\mathcal{H}$  déformée et la partie fixe  $\mathcal{F}$ . Nous allons adapter pour cela la méthode de Botsch [20].

Supposons que  $\mathcal{S}$  et  $\mathcal{S}'$  sont des surfaces paramétriques (notre maillage triangulé 2-manifold peut bien correspondre à ce cas). Soit  $p$  la fonction de paramétrisation  $p : \Omega \subset \mathbb{R}^2 \rightarrow \mathcal{S} \subset \mathbb{R}^3$ . Ainsi, la fonction de déplacement est définie par :  $d : \Omega \rightarrow \mathbb{R}^3$ . On associe alors à chaque point  $p(u, v)$  un vecteur de déplacement  $\overrightarrow{d(u, v)}$ , donc  $p'(u, v) = p(u, v) + \overrightarrow{d(u, v)}$  et  $\mathcal{S}' = p'(\Omega)$ .

Nous utilisons la première et la seconde formes fondamentales  $\mathbf{I}(u, v)$  et  $\mathbf{II}(u, v)$  pour mesurer la propriété géométrique de surface, qui sont indépendantes de la paramétrisation (voir Section 2). Une énergie basée sur le changement de la première et la seconde formes fondamentales a été introduite pour mesurer l'étirement et la flexion de déformation. Soit  $\mathbf{I}'$  et  $\mathbf{II}'$  les formes correspondant à la surface déformée  $\mathcal{S}'$ . L'énergie est alors définie par :

$$E(\mathcal{S}') = \iint_{\Omega} k_s \left\| \mathbf{I}'(u, v) - \mathbf{I}(u, v) \right\|_F^2 + k_b \left\| \mathbf{II}'(u, v) - \mathbf{II}(u, v) \right\|_F^2 dudv.$$

Dans cette équation, les coefficients  $k_s$  et  $k_b$  sont utilisés pour contrôler l'étirement et la flexion. Nous nous intéressons ensuite à minimiser cette énergie en appliquant des conditions aux limites (les déplacements pré-définis à  $\mathcal{H}$  et  $\mathcal{F}$ ). Afin de faciliter la minimisation de cette énergie non-linéaire, les différences des

formes fondamentales sont remplacées par les dérivées partielles de la fonction de déplacement  $\mathbf{d}$ .

$$E(\mathbf{d}) = \iint_{\Omega} k_s(\|\mathbf{d}_u\|^2 + \|\mathbf{d}_v\|^2) + k_b(\|\mathbf{d}_{uu}\|^2 + 2\|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2) dudv,$$

avec  $\mathbf{d}_u = \frac{\partial}{\partial u}\mathbf{d}(u, v)$  et  $\mathbf{d}_{uv} = \frac{\partial^2}{\partial u \partial v}\mathbf{d}(u, v)$ . Notons que les deux termes ressemblent aux énergies à minimiser lors de la phase de lissage (voir Section 2), alors que maintenant on ne minimise plus l'aire ou la courbure de surface, mais nous minimisons l'influence de changement. Afin de minimiser cette énergie  $E(\mathbf{d})$ , nous avons besoins de résoudre l'équation aux dérivées partielles suivante :

$$-k_s \Delta \mathbf{d} + k_b \Delta^2 \mathbf{d} = 0,$$

$$\begin{aligned} \Delta \mathbf{d} &= \mathbf{d}_{uu} + \mathbf{d}_{vv}, \\ \Delta^2 \mathbf{d} &= \mathbf{d}_{uuuu} + 2\mathbf{d}_{uuvv} + \mathbf{d}_{vvvv}. \end{aligned}$$

Il faut tenir compte des contraintes lors de la résolution de cette équation.

**Application au maillage discret** Maintenant nous allons appliquer ces modèles mathématiques à notre cas discret. La surface qu'on traite  $\mathcal{M}_c$  est un maillage triangulé, donc une version discrète de la surface paramétrique  $\mathcal{S}$ . Ainsi, la fonction de déplacement  $\mathbf{d}$  est notée comme suite :  $\forall p_i(x, y, z) \in \mathcal{S}$ ,  $\overrightarrow{\mathbf{d}(p_i)} = \mathbf{d}_i(dx, dy, dz)$ . La déformation de  $\mathcal{M}_c$  est présentée par les équations suivantes :

$$\begin{cases} \mathbf{d}_i = \overrightarrow{0}, \forall p_i \in \mathcal{F}, \\ \mathbf{d}_i = \overrightarrow{\mathbf{d}_i}, \forall p_i \in \mathcal{H}, \\ -k_s \Delta \mathbf{d}_i + k_b \Delta^2 \mathbf{d}_i = \overrightarrow{0}, \forall p_i \in \mathcal{R}. \end{cases}$$

*Discrétisation de l'opérateur  $\Delta$*  L'idée est similaire à celle de l'opérateur « Umbrella » (Section 4.6). On le calcule en utilisant les valeurs de fonction à ses voisins. Soit  $v_i$  un sommet dans le maillage, alors

$$\Delta \mathbf{d}(v_i) = \omega_i \sum_{v_j \in \mathcal{N}(v_i)} \omega_{ij}(\mathbf{d}(v_j) - \mathbf{d}(v_i)),$$

les coefficients  $\omega_i$  et  $\omega_{ij}$  seront décrits après. Nous pouvons aussi écrire l'équation en utilisant notre notation :

$$\Delta \mathbf{d}_i = \omega_i \sum_{v_j \in \mathcal{N}(v_i)} \omega_{ij}(\mathbf{d}_j - \mathbf{d}_i).$$

De la même façon, on a :

$$\Delta^2 \mathbf{d}_i = \omega_i \sum_{v_j \in \mathcal{N}(v_i)} \omega_{ij}(\Delta \mathbf{d}_j - \Delta \mathbf{d}_i).$$

*Système linéaire* En vue de résoudre l'équation aux dérivées partielles, nous écrivons ces relations sous forme de matrices. Soit  $\{v_0, v_1, \dots, v_{n-1}\}$  les sommets dont les déplacements  $\{\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{n-1}\}$  sont à calculer.

$$\begin{pmatrix} \Delta \mathbf{d}_0 \\ \vdots \\ \Delta \mathbf{d}_{n-1} \end{pmatrix} = \underbrace{\mathbf{DM}}_{\mathbf{L}} \begin{pmatrix} \mathbf{d}_0 \\ \vdots \\ \mathbf{d}_{n-1} \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} \omega_0 & 0 & \dots & 0 \\ 0 & \omega_1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \omega_{n-1} \end{pmatrix} \quad \mathbf{M} : m_{ij} = \begin{cases} -\sum_{v_k \in \mathcal{N}(v_i)} \omega_{ik}, & i = j \\ \omega_{ij}, & v_j \in \mathcal{N}(v_i) \\ 0, & \text{sinon.} \end{cases}$$

Donc,  $\mathbf{D}$  est une matrice diagonale et  $\mathbf{M}$  est une matrice symétrique. De la même façon, on a :

$$\begin{pmatrix} \Delta^2 \mathbf{d}_0 \\ \vdots \\ \Delta^2 \mathbf{d}_{n-1} \end{pmatrix} = \mathbf{DM} \begin{pmatrix} \Delta \mathbf{d}_0 \\ \vdots \\ \Delta \mathbf{d}_{n-1} \end{pmatrix} = \mathbf{L}^2 \begin{pmatrix} \mathbf{d}_0 \\ \vdots \\ \mathbf{d}_{n-1} \end{pmatrix}.$$

Ainsi, l'équation aux dérivées partielles que nous allons résoudre se transforme en un système linéaire :

$$\underbrace{(-k_s \mathbf{L} + k_b \mathbf{L}^2)}_{n \times n} \underbrace{\begin{pmatrix} \mathbf{d}_0 \\ \vdots \\ \mathbf{d}_{n-1} \end{pmatrix}}_{n \times 3} = \underbrace{\begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}}_{n \times 3}.$$

*Remarque* Il existe différentes façons pour définir  $\omega_i$  et  $\omega_{ij}$ , bien qu'on utilise souvent la définition proposée par Desbrun *et al.* [23]. Dans le système linéaire, il faut aussi considérer les contraintes, cela veut dire qu'on enlève les  $\mathbf{d}_i$  pré-définis de l'équation. Ce sont les variables connus lors du calcul. Nous pouvons résoudre ce système linéaire par plusieurs méthodes, par exemple, celle du gradient conjugué. Cependant la méthode de Jacobi ou de Gauss-Seidel est souvent plus facile à implémenter, pour cela, il suffit de récrire l'équation composante par composante en trouvant les coefficients de chaque terme comme nous avons fait dans la section 4.6.

*Sélection de la zone* Il ne nous reste qu'à définir la zone de déformation ( $\mathcal{H}$ ). On a un ensemble de points de passage par lesquels doit passer la surface déformée. L'idée est de choisir les sommets dans le maillage  $\mathcal{M}_c$  qui sont les plus



proches de ces points de passage. Puis on les déplace vers les points de passage.

---

**Algorithme 3:** Sélection de zone  $\mathcal{H}$

---

**Data :** Contrainte : points de passage  $\{p_0, \dots, p_m\}, \{v_0, \dots, v_{n-1}\} \subset \mathcal{M}_c$

**Result :**  $p_i.candidat \quad \forall i \in \{0, \dots, m\}$

**begin**

**for**  $i \leftarrow 0$  **to**  $n - 1$  **do**

$v_i.candidat \leftarrow j$

    //  $j$  tel que  $\min_{0 \leq j \leq m} \|v_i - p_j\|$  est atteint

**for**  $j \leftarrow 0$  **to**  $m$  **do**

$p_j.candidat \leftarrow i$

    //  $i$  tel que  $\min_{v_i.candidat=j} \|v_i - p_j\|$  est atteint

---

## 4.8 Finition

C'est la dernière étape de complétion et de reconstruction du maillage initial. Normalement après l'étape de déformation, nous avons déjà un maillage reconstruit qui respecte les contraintes souhaitées. Cependant si on est dans le cas d'une déformation grosse faisant apparaître des triangles fins, nous avons besoin de remailler la surface déformée. Nous pouvons simplement appliquer la même procédure que dans la section 4.5 ou utiliser d'autres méthodes de raffinement [21] [22]. De toute façon, le remaillage ne change pas les positions de sommets calculées. D'où très peu d'influences sur la forme de surface.

## 5 Applications

### 5.1 Développement

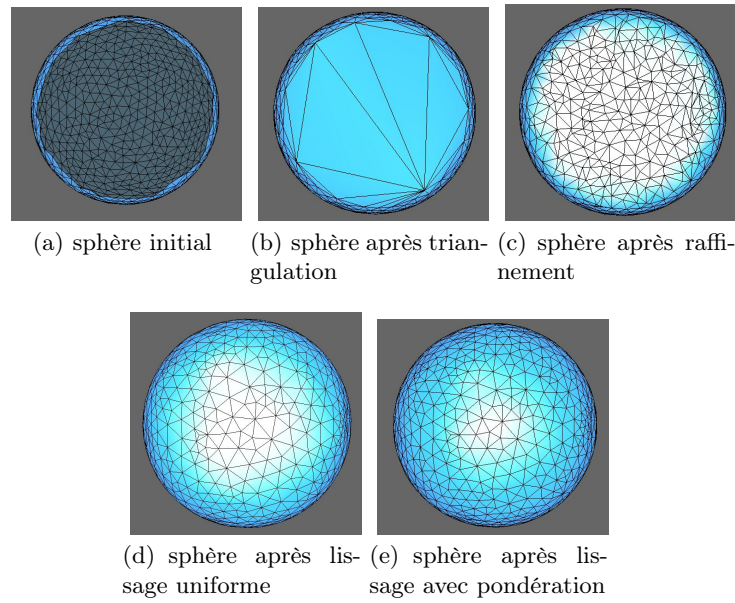
Le développement s'est fait dans la Plate-forme modulaire (MEPP) de traitement de maillages basée sur CGAL, Qt et libQGLViewer. Le développement a été réalisé en C++. Quelques avantages de la plate-forme MEPP :

- Utilisation de la librairie CGAL. C'est une librairie robuste et enrichissante utilisée dans nombreux établissements.
- L'interface graphique et la visualisation déjà intégrées qui nous permet de nous concentrer sur l'algorithme.
- Création simple de nouveaux composants, par exemple notre méthode de reconstruction.

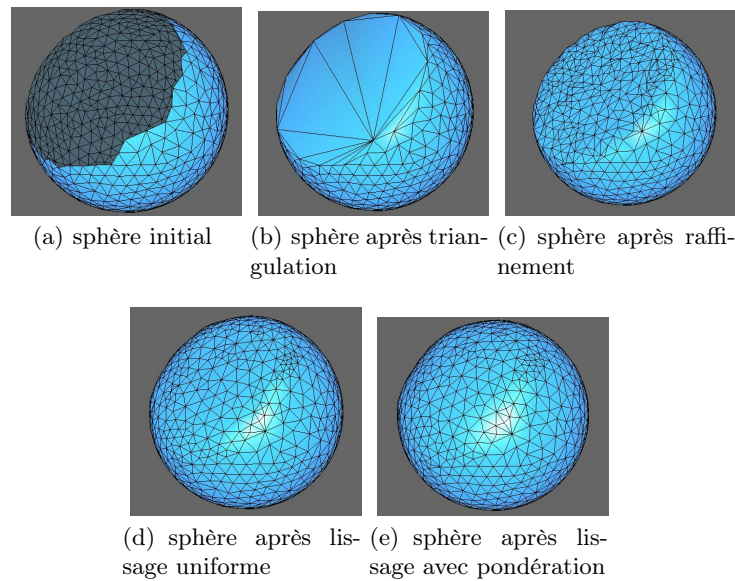
La Figure 12 présente l'architecture de cette plate-forme.

### 5.2 Résultats

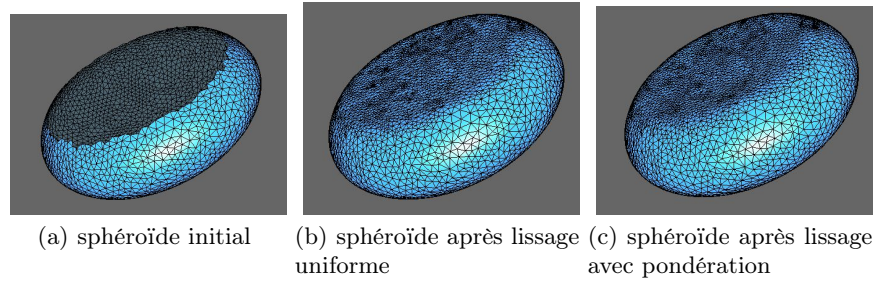
Nous montrons ici par quelques images les résultats de l'algorithme.



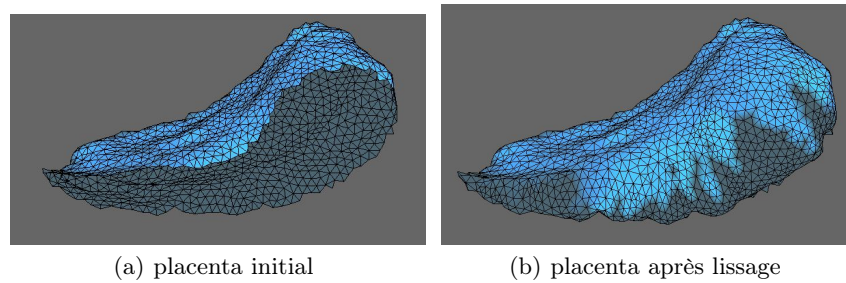
**FIGURE 13.** Sphère (700 sommets) -vue du dessus



**FIGURE 14.** Sphère (700 sommets) -vue isométrique



**FIGURE 15.** Sphéroïde (3000 sommets) -vue isométrique



**FIGURE 16.** Placenta (1200 sommets)

L'implémentation du pipeline complet n'est pas encore terminée, par contre un outil de génération de maillage à partir de fonctions implicites a été mis en place et la complétion sous contraintes locales a aussi été implémentée. Notons que nous avons choisi la sphère comme modèle de test parce que nous verrons clairement les propriétés géométriques avec une sphère grâce à sa courbure uniforme.

## 6 Conclusions et perspectives

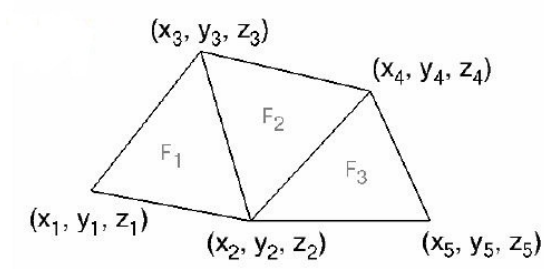
Dans ce travail, nous avons étudié des méthodes permettant de reconstruire et compléter des maillages automatiquement. Nous avons amélioré une approche géométrique existante afin de prendre en compte nos contraintes spécifiques. Les développements effectués ont été intégrés à la plate-forme MEPP : module de reconstruction et complétion de maillages. Les résultats obtenus correspondent principalement aux résultats souhaités. Cela montre que l'approche choisie est utilisable dans notre cas. A partir de ceci, nous pourrions donner des solutions plus applicables.

Comme la méthode a pour but de servir aux projets utilisant les images médicales pour faire des simulations bio-mécaniques, il reste encore plusieurs points à améliorer : la contrainte de « zone à éviter » pourra être traitée. Nous pourrions aussi envisager des cas où plusieurs contraintes doivent être respectées. Il existe aussi des cas où on veut compléter plusieurs maillages en même temps dont l'un dépend de l'autre.

## Références

1. *CGAL User and Reference Manual*, October 2010. Release 3.7.
2. Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes : reconstruction and parameterization from range scans. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, 2003.
3. Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon Mesh Processing*, chapter 1,6,8. AK Peters / CRC Press, 2010.
4. Sylvain Brandel. Modélisation géométrique, 2010-2011. Cours Université Lyon 1.
5. Mathieu Brévilliers. Construction de la triangulation de delaunay de segments par un algorithme de flip, 2009.
6. Ricardo CAMARERO. Méthode d'avancé de front, Janvier 2007. Ecole Polytechnique Montréal.
7. J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 67–76, New York, NY, USA, 2001. ACM.
8. Julien Dardenne. *Maillage 3D de structures anatomiques pour la simulation électromagnétique et thermique*. PhD thesis, L'Institut National des Sciences Appliquées de Lyon, 2009.
9. Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
10. William E.Lorensen and Harvey E.Cline. Marching cubes : A high resolution 3d surface construction algorithm. In *ACM SIGGRAPH*, pages 163–170, 1987.
11. Pascal Jean Frey. Génération et adaptation de maillages de surfaces à partir de données anatomiques discrètes. Technical report, INRIA, 2003.
12. Pascal Jean Frey and Paul-Louis George. *Maillages, applications aux éléments finis*. Hermès Science Publications, 1999.
13. Thomas Funkhouser. *Polygonal meshes*, 2006. Princeton University.
14. Bruce G.Baumgart. Winged-edge polyhedron representation. Technical report, Stanford University, 1972.
15. Frank J.Bossen and Paul S.Heckbert. A pliant method for anisotropic mesh generation. In *Proc.5th Int.Meshing Roundtable*, 1996.
16. Leif Kobbelt. Discrete fairing. Technical report, University of Erlangen-Nürnberg, 1997.
17. Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 105–114, New York, NY, USA, 1998. ACM.
18. Peter Liepa. Filling holes in meshes. In L.Kobbelt, P.Schröder, and H.Hoppe, editors, *Eurographics Symposium on Geometry Processing*, pages 200–207, 2003.

19. L.Kettner. Using generic programming for designing a data structure for polyhedral surfaces. *Computational Geometry : Theory and Applications*, pages 65–90, 1999.
20. M.Botsch and O.Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1) :213–230, 2008.
21. M.C.Rivara. New mathematical tools and techniques for the refinement and/or improvement of unstructured triangulations. In *Proc.5th Int.Meshing Roundtable*, 1996.
22. M.C.Rivara. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. *International Journal for Numerical Methods in Engineering*, 40(18) :3313–3324, 1997.
23. Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics III*, pages 35–57, 2003.
24. Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus Gross, and Leonidas J. Guibas. Example-based 3d scan completion. In *Proceedings of the third Eurographics symposium on Geometry processing*. Eurographics Association, 2005.
25. Leif P.Kobbelt. Discrete fairing and variational subdivision for freeform surface design. *Visual Computer*, 16(3-4) :142–158, 2000.
26. Wei Zhao, Shuming Gao, and Hongwei Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 2007.



VERTEX TABLE			
v <sub>1</sub>	x <sub>1</sub>	y <sub>1</sub>	z <sub>1</sub>
v <sub>2</sub>	x <sub>2</sub>	y <sub>2</sub>	z <sub>2</sub>
v <sub>3</sub>	x <sub>3</sub>	y <sub>3</sub>	z <sub>3</sub>
v <sub>4</sub>	x <sub>4</sub>	y <sub>4</sub>	z <sub>4</sub>
v <sub>5</sub>	x <sub>5</sub>	y <sub>5</sub>	z <sub>5</sub>

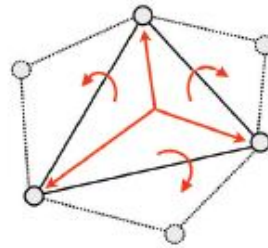
FACE TABLE			
F <sub>1</sub>	v <sub>1</sub>	v <sub>2</sub>	v <sub>3</sub>
F <sub>2</sub>	v <sub>2</sub>	v <sub>4</sub>	v <sub>3</sub>
F <sub>3</sub>	v <sub>2</sub>	v <sub>5</sub>	v <sub>4</sub>

(a) Sans connectivités

Vertex	
Point	position
FaceRef	face

Face	
VertexRef	vertex[3]
FaceRef	neighbor[3]



(b) Avec connectivités

FIGURE 3. Structures basées sur faces (images extraites de [13] et [3]).

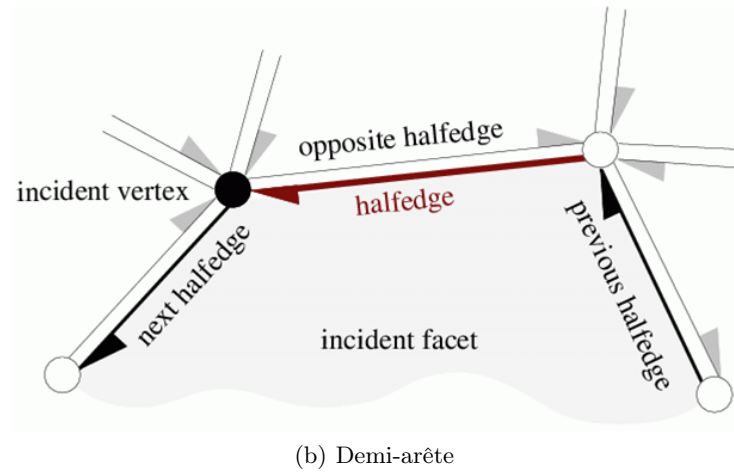
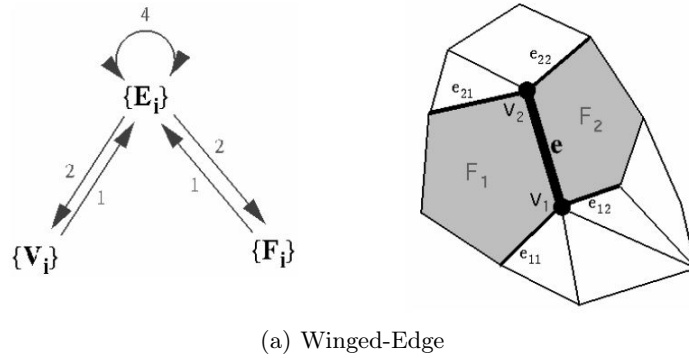


FIGURE 4. Structures basées sur arêtes (images extraites de [13] et [1]).

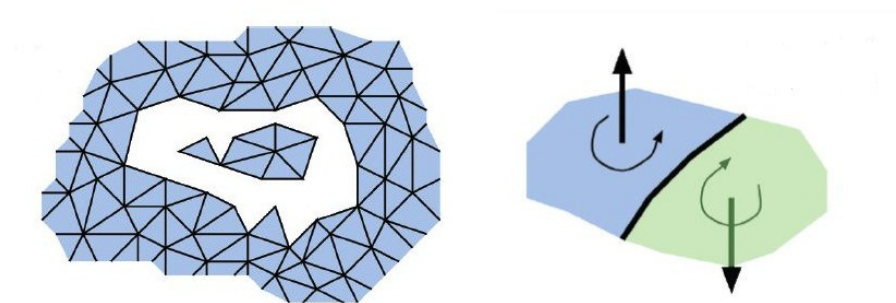
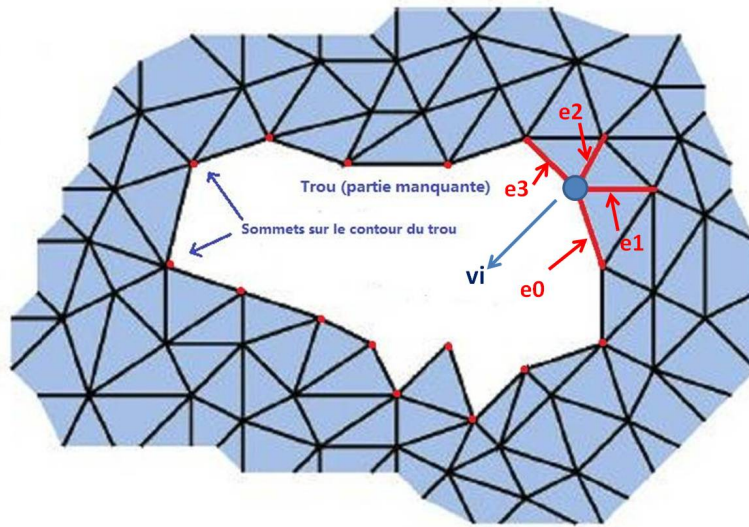


FIGURE 7. Partie isolée et élément libre (gauche). Deux éléments sont orientés différemment (droite) (image extraite de [3]).

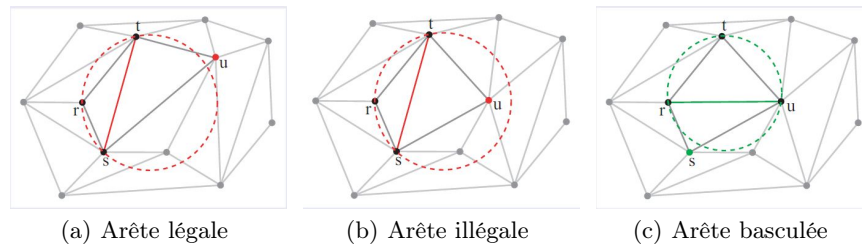




**FIGURE 8.** Pipeline de traitements de notre méthode



**FIGURE 9.**  $\sigma(v_i) = (\|e_0\| + \|e_1\| + \|e_2\| + \|e_3\|)/4$



**FIGURE 10.** Bascule d'arête (images issue de [5]).

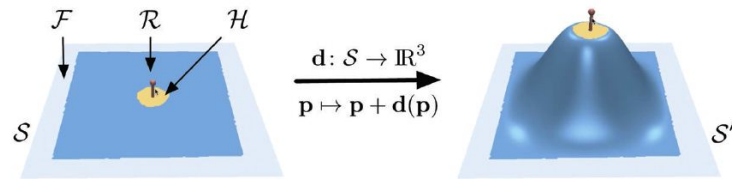


FIGURE 11. Déformation de surface

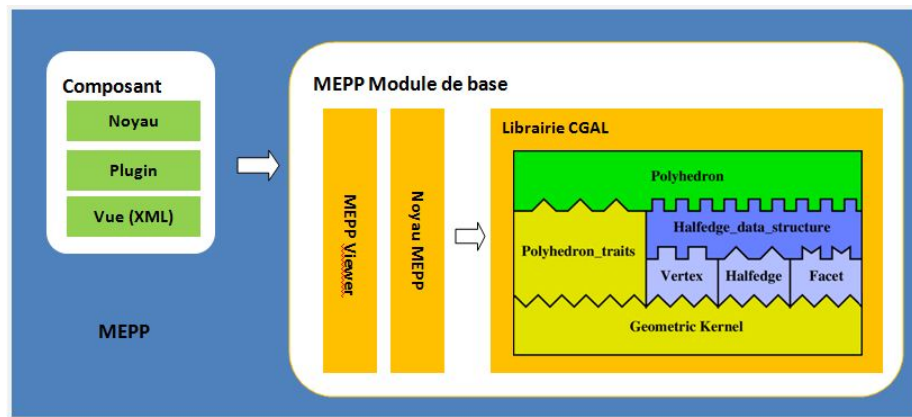


FIGURE 12. Architecture de MEPP