



Université de Montréal

# Images géométriques de genre arbitraire dans le domaine sphérique

par

Mathieu Gauthier

Département d'informatique et de recherche opérationnelle  
Faculté des arts et des sciences

Mémoire présenté à la faculté des études supérieures  
en vue de l'obtention du grade de  
Maître ès sciences (M.Sc.)  
en informatique

Octobre 2008

© Mathieu Gauthier, 2008

Université de Montréal  
Faculté des études supérieures

Ce mémoire de maîtrise intitulé

Images géométriques de genre arbitraire dans le domaine sphérique

présenté par  
Mathieu Gauthier

a été évalué par un jury composé des personnes suivantes :

Neil F. Stewart  
président-rapporteur

Pierre Poulin  
directeur de recherche

Pierre McKenzie  
membre du jury

# Sommaire

Alors que les algorithmes de paramétrisations sphériques se limitent aux modèles géométriques de genre 0, nous estimons qu'un large éventail de modèles de genre arbitraire peuvent bénéficier d'une représentation en domaine sphérique. Nous présentons un pipeline complet et robuste capable de générer des images géométriques sphériques à partir de surfaces de genre arbitraire où les trous sont représentés de façon explicite. Le modèle géométrique, représenté par un maillage de triangles, est modifié pour devenir topologiquement équivalent à une sphère en coupant le long des générateurs de chaque trou. Le modèle de genre 0 résultant est paramétré sur la sphère en  $\mathbb{R}^3$  pour être ensuite rééchantillonné de façon à préserver la connectivité entre les trous et réduire l'impact visuel de la reconstruction aux points de coupures. Contrairement aux images géométriques traditionnelles et multi-cartes, cette représentation limite la longueur du réseau de coupes, ce qui réduit la distortion dans les images géométriques générées. Nous illustrons notre approche en présentant des exemples de remaillages, de rendus en niveaux de détail, de placage de textures et d'édicions topologiques.

## **Mots clefs :**

Images géométriques, paramétrisation sphérique, topologie informatique, remaillage

# Abstract

While existing spherical parameterization algorithms are limited to genus-0 geometrical models, we believe a wide class of models of arbitrary genus can also benefit from use of the spherical domain. We present a complete and robust pipeline that can generate spherical geometry images from arbitrary genus surfaces where we have explicitly represented the holes. The geometrical model, represented as a triangle mesh, is first made topologically equivalent to a sphere by cutting along the generators on each hole, thus performing genus reduction. The resulting genus-0 model is then parameterized on the sphere, where it is resampled in a way that preserves connectivity between holes while at the same time reducing the visual impact of seams due to these holes. Unlike traditional and multi-chart geometry images, this representation limits the length of cut paths, thus reducing the distortion in the resulting geometry images. We illustrate our approach on remeshing, level-of-detail rendering, texture mapping, and topology editing.

## **Keywords :**

Geometry images, spherical parameterization, computational topology, remeshing

# Table des matières

<b>Remerciements</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Topologie informatique</b>	<b>4</b>
2.1 Définition . . . . .	4
2.2 Surface . . . . .	5
2.3 Théorie de Morse . . . . .	8
2.4 Graphe de Reeb . . . . .	11
<b>3 Paramétrisation de surface</b>	<b>14</b>
3.1 Introduction . . . . .	14
3.2 Origine . . . . .	15
3.3 Imagerie . . . . .	18
3.4 Applications . . . . .	20
3.5 Paramétrisation plane . . . . .	21
3.5.1 Paramétrisation conforme . . . . .	21
3.5.2 Paramétrisation équivalente . . . . .	23
3.5.3 Compromis . . . . .	24
3.6 Paramétrisation sphérique . . . . .	26
<b>4 Images géométriques</b>	<b>29</b>
4.1 Représentation de maillages . . . . .	29
4.2 Images géométriques traditionnelles . . . . .	32
4.2.1 Survol de l'algorithme . . . . .	32

4.2.2	Coupe . . . . .	33
4.2.3	Paramétrisation et échantillonnage . . . . .	35
4.3	Extensions . . . . .	36
4.4	Applications . . . . .	39
<b>5</b>	<b>Paramétrisation sphérique</b>	<b>42</b>
5.1	Survol de l'algorithme . . . . .	42
5.2	Initialisation . . . . .	45
5.3	Réduction du genre . . . . .	46
5.3.1	Propagation de la fonction de Morse . . . . .	47
5.3.2	Construction du graphe de Reeb . . . . .	49
5.3.3	Traçage des générateurs . . . . .	52
5.3.4	Classification des générateurs . . . . .	55
5.3.5	Optimisation des générateurs . . . . .	59
5.4	Paramétrisation sphérique . . . . .	60
<b>6</b>	<b>Échantillonnage</b>	<b>63</b>
6.1	Création du domaine d'échantillonnage . . . . .	63
6.2	Gestion des composantes de bord . . . . .	65
6.2.1	Identification des échantillons de bords . . . . .	68
6.2.2	Distribution des échantillons autour de la composante de bord . . . . .	70
6.2.3	Tressage . . . . .	71
6.3	Échantillonnage . . . . .	73
6.4	Reconstruction . . . . .	74
<b>7</b>	<b>Résultats et applications</b>	<b>76</b>
7.1	Implémentation . . . . .	76
7.2	Résultats et applications . . . . .	78
7.2.1	Remaillages et images géométriques . . . . .	79
7.2.2	Comparaison avec images géométriques traditionnelles . . . . .	83
7.2.3	Edition topologique . . . . .	83
<b>8</b>	<b>Conclusion et travaux futurs</b>	<b>86</b>



# Table des figures

1.1	Image géométrique du buddha et sa reconstruction. Pour fin de visualisation, les normales plutôt que les positions des sommets sont affichées. Image tirée de [GGH02]. . . . .	2
1.2	Images géométriques sphériques du buddha sur le domaine HEALPix générées avec notre méthode. Pour fins de visualisation, les images des normales sont affichées. . . . .	3
2.1	Une tasse est homéomorphe à un tore puisqu'il existe une déformation continue pouvant passer de l'un à l'autre. . . . .	5
2.2	Une sphère est une 2-variété alors qu'un cercle est une 1-variété. . . . .	6
2.3	Illustration du vocabulaire topologique utilisé dans ce document. Nous dirons que les lignes rouges sont des méridiens tandis que les bleues sont des parallèles. . . . .	6
2.4	Deux exemples de surfaces non orientables. . . . .	7
2.5	Objets de genres 0, 1 et 3. . . . .	7
2.6	Un tore est le produit de deux cercles générateurs. La ligne bleue est le générateur méridien tandis que la ligne rouge est le générateur parallèle. . . . .	8
2.7	Les trois types de points critiques sont identifiés par le nombre de directions négatives (en bleu) dans l'hessien. . . . .	9
2.8	Les courbes de niveau de la fonction d'altitude sur un tore et les quatre points critiques. . . . .	10
2.9	Balayage du tore avec la fonction d'altitude. . . . .	10
2.10	Exemple de graphe de Reeb pour un 2-tore. . . . .	12
3.1	Paramétrisation entre une surface et un domaine. . . . .	15

3.2	Reconstruction de la carte du monde de Ptolémée tirée du livre « <i>The Outline of History</i> » de H.G. Wells. . . . .	16
3.3	Projections de la Terre. . . . .	17
3.4	Placage de textures. Image courtoisie de Daren Horley. . . . .	18
3.5	Quelques applications de la paramétrisation. . . . .	20
3.6	Les polygones ont la même aire dans les deux paramétrisations. . . . .	23
3.7	Exemples de paramétrisations planaires. Les images sont tirées de [SPR06].	25
3.8	Séquence d’optimisation de paramétrisation sur la sphère. Image tirée de [FSD05]. . . . .	26
3.9	Exemple de paramétrisation sphérique de Saba <i>et al.</i> [SYGS05]. Les deux moitiés du modèle géométrique sont paramétrées dans le plan, projetées sur la sphère et ensuite optimisées sur celle-ci. . . . .	27
3.10	Exemples de paramétrisations sphériques de Friedel <i>et al.</i> . Images tirées de [FSD05]. . . . .	28
4.1	Exemple de maillage de triangles et représentation interne. . . . .	30
4.2	Exemple d’une image géométrique. A gauche, la couleur (RGB) encode la position (XYZ) du maillage associé au modèle géométrique de droite. Images tirées de [GGH02]. . . . .	32
4.3	Survol de l’algorithme original des images géométriques. Images tirées de [FSD05]. . . . .	33
4.4	La projection de Mollweide et la projection sinusoidale interrompue utilisent deux arbres de coupe différents pour produire une carte planaire. .	34
4.5	Couper le long des deux générateurs d’un tore permet de l’ouvrir en un plan. . . . .	34
4.6	En haut : Le graphe de coupe d’un 3-tore peut être trouvé en faisant croître une surface à partir d’un triangle initial. En bas : La coupe est nettoyée et révèle les 6 générateurs. Images tirées de la présentation de [GGH02]. . . . .	35
4.7	La procédure d’optimisation de la coupe tend à aller chercher les extrémités où l’échantillonnage est sous-optimal. Images tirées de la présentation de [GGH02]. . . . .	36

4.8	Chaque côté du segment de coupe $a$ ( $a$ et $a'$ ) reçoit la même longueur sur la bordure de la paramétrisation. De plus, les noeuds de l'arbre de coupe doivent coïncider avec les points de la grille d'échantillonnage. Images tirées de la présentation de [GGH02]. . . . .	37
4.9	Quelques extensions des images géométriques. . . . .	38
4.10	Quelques applications des images géométriques. . . . .	39
5.1	Vue d'ensemble du processus. . . . .	44
5.2	Fermeture centrée. . . . .	45
5.3	Couper un tore le long d'un de ses deux générateurs permet de l'ouvrir en un cylindre. . . . .	46
5.4	Le choix du générateur utilisé pour la coupe influence la qualité de la paramétrisation sphérique et donc de la reconstruction. . . . .	47
5.5	La fonction de Morse de Ni <i>et al.</i> évolue entre un point minimum et un point maximum ; elle ne crée pas de points critiques artificiels. . . . .	48
5.6	Les noeuds du graphe pointent vers l'endroit où l'événement topologique correspondant (face et sommet) s'est produit. Les arcs contiennent la liste des faces balayées entre les noeuds (de même couleur) ainsi que le contour. . . . .	50
5.7	Séparation et fusion de contours lors du balayage de la surface. . . . .	51
5.8	La procédure de Steiner et Fischer [SF03] crée $G_1$ en sélectionnant le plus court des contours parents, et $G_2$ en cherchant un chemin de $v_f$ vers lui-même à travers la région visitée n'intersectant pas un autre générateur précédemment identifié. . . . .	53
5.9	Subdivision barycentrique. . . . .	54
5.10	La subdivision barycentrique permet de créer de nouveaux chemins. Les lignes rouges représentent des obstacles et la ligne pointillée le nouveau chemin tracé. . . . .	55
5.11	L'algorithme complet et robuste de traçage des générateurs. . . . .	56
5.12	Valeur associée à chaque croisement. . . . .	57
5.13	Deux types de noeuds possibles entre deux générateurs. . . . .	57

5.14	Nous déplaçons un générateur dans la direction inverse de la normale. Si les deux générateurs sont enlacés, il s'agit d'un parallèle, sinon il s'agit d'un méridien. . . . .	58
5.15	À gauche, nous projetons les deux générateurs sur un plan $P$ perpendiculaire à la normale $\vec{N}$ . En haut à droite, les deux générateurs projetés apparaissent dans l'ordre rouge-bleu-rouge-bleu, ce qui est valide. En bas à droite, exemple de configuration invalide. . . . .	58
5.16	Le point du triangle le plus près du centre de la sphère est au centre du cercle circonscrit pour un triangle obtus (gauche) et au milieu de l'arête la plus longue pour un triangle aigu (droite). . . . .	61
5.17	Séquence de paramétrisation hiérarchique pour un modèle de tricératops.	62
6.1	Chaque point d'échantillonnage pointe vers $1..n$ pixels d'image(s). . . .	64
6.2	L'optimisation d'un domaine d'échantillonnage cubique uniformise mieux l'aire des faces, produisant une meilleure distribution des sommets. . . .	65
6.3	Le trou découpé et colmaté entre le pouce et l'index a créé deux composantes de bord dans la paramétrisation. La troisième composante de bord située à la base de la main a simplement été colmatée au chargement.	66
6.4	Une fissure (en blanc) est bien visible dans la reconstruction à l'endroit où le générateur a été coupé. . . . .	67
6.5	Identification des échantillons situés autour de la composante de bord et annulation de ceux situés à l'intérieur. . . . .	69
6.6	Nous laissons consciemment un échantillon nul au bout d'une petite crête.	70
6.7	Les aires normalisées des sous-triangles sphériques sont utilisées comme poids d'interpolation linéaire $\lambda_1, \lambda_2, \lambda_3$ . . . . .	71
6.8	Distribution des échantillons sur une composante de bord. Le domaine d'échantillonnage (grille régulière) est superposé sur la paramétrisation sphérique (triangulation). . . . .	72
6.9	Les trois configurations possibles pour $ E^-  = 3$ et $ E^+  = 4$ . . . . .	72
6.10	Exemple de remaillage récursif. Les lignes pointillées représentent deux points les plus près dans l'espace 3D. Au niveau 3 la récursion arrête car $ E^-  =  E^+ $ et au niveau 4, elle s'arrête car $ E^-  = 1$ . . . . .	73

6.11	Images géométriques pour le modèle de la main avec un domaine d'échantillonnage cubique de résolution $6 \times 16 \times 16$ . Pour fin de visualisation, la norme euclidienne de chaque position est colorée avec une palette arc-en-ciel allant du violet au rouge. Les pixels nuls (trous) sont en rouge foncé. . . . .	74
6.12	Lors de la marche autour de la composante de bord, nous évitons d'avancer sur un sommet adjacent à un sommet déjà visité. . . . .	75
6.13	À gauche, modèle original de la main avec 3000 sommets. À droite, remaillage de 1213 sommets à partir des images géométriques. . . . .	75
7.1	Cinq domaines d'échantillonnage. . . . .	78
7.2	Exemple de tressage d'une coupure sur la main. . . . .	79
7.3	L'échantillonnage des normales à une résolution supérieure (dans ce cas, 4 fois plus grande) permet de simuler de fins détails à la surface de l'amphore comparativement à l'image de gauche. . . . .	81
7.4	Nous complétons les images de normales pour permettre un meilleur filtrage au niveau des jointures. . . . .	81
7.5	En haut : Modèle du buddha avec ses 104 générateurs parallèles, sa paramétrisation sphérique et l'échantillonnage sur domaine <i>HEALPix</i> (normales). En bas : remaillages à différentes résolutions, dans l'ordre : $129^2$ , $65^2$ , $33^2$ , $17^2$ , $9^2$ , $5^2$ et $3^2$ . Le genre de chaque remaillage est respectivement : 11, 9, 9, 7, 4, 2 et 0. . . . .	82
7.6	En haut : Le modèle original du dragon, sa paramétrisation sphérique et l'échantillonnage avec un domaine <i>HEALPix</i> $65 \times 65 \times 12$ . En bas : Reconstruction sans normales, avec normales ( $513 \times 513 \times 12$ ) et avec fil de fer. . . . .	82
7.7	De haut en bas : remaillage de la main avec carte cubique $33 \times 33 \times 6$ , colmatage de trou entre le pouce et l'index, et creusage d'un nouveau trou. . . . .	85

# Liste des tableaux

7.1	Résultats obtenus avec notre méthode, de haut en bas : les enfants, l'amphore, la main et le culbuteur. . . . .	80
7.2	Comparaison entre les images géométriques traditionnelles et notre méthode sphérique de genre arbitraire. . . . .	84

# Remerciements

Bien que ce mémoire soit l'aboutissement de plusieurs mois de labeur de ma part, il serait égoïste de m'en attribuer tout le mérite. Je désire prendre quelques instants pour remercier les gens et organismes de mon entourage qui ont rendu possible la rédaction de ce document et de façon plus générale, mon cheminement académique des dernières années.

Je tiens à d'abord remercier cordialement Pierre Poulin qui a dirigé mes travaux des deux dernières années, m'a appuyé financièrement et a participé à la rédaction et la correction de ce mémoire. Merci à François Meunier qui m'a donnée l'occasion de travailler sur divers projets très motivants pendant et après mon Baccalauréat. Merci à Jacques Bergeron, Martin Paquette et Carl L'Archevêque du département de Psychologie de l'Université de Montréal pour leur support financier. Merci à Joël Robidoux, Yvan Chaîné et Serge Lachapelle de JCMB Technology pour leur support envers ma démarche et les horaires de travail flexibles.

Merci à Howard Stern et Jean-François Fillion pour leur divertissement radiophonique qui a accompagné chacune de mes heures de travail. Un remerciement spécial à ma famille pour leur encouragement moral et financier et à Claudia et Carmelle pour leur compréhension.

Finalement, merci au CRSNG, à la FQRNT et à l'Université de Montréal pour leur appui financier.

# Chapitre 1

## Introduction

*The answer appears to us before the question... Practical application is found by not looking for it, and one can say that the whole progress of civilization rests on that principle... Practical questions are most often solved by means of existing theories... It seldom happens that important mathematical researches are directly undertaken in view of a given practical use : they are inspired by the desire which is the common motive of every scientific work, the desire to know and understand.*

—Jacques Hadamard, *The Psychology of Invention in the Mathematical Field*

La représentation qu'on se fait d'un concept ou d'une idée a parfois un impact important sur la façon dont nous traitons un problème associé à celui-ci. Une représentation alternative peut souvent ouvrir la voie à une autre approche ou à de nouvelles possibilités. Par exemple, considérer un son, une image ou tout signal dans sa représentation de Fourier ouvre un monde de possibilités ; des opérations qui autrefois semblaient complexes, voire impossibles à exécuter dans un temps raisonnable, sont maintenant accomplies par le biais de simples additions et multiplications.

En infographie, la représentation d'objets tridimensionnels est largement dominée par la représentation surfacique par maillages polygonaux. Plus simplement, on exprime un objet en considérant uniquement sa surface et en modélisant celle-ci par une série de polygones planaires approximant la forme de ce dernier. Pour ce faire, on positionne

des sommets dans l'espace 3D (géométrie) que l'on assemble en groupes pour définir des faces (connectivité). Une telle structure est complexe à parcourir et requiert bon nombre d'accès aléatoires à la mémoire ce qui en réduit sa performance.

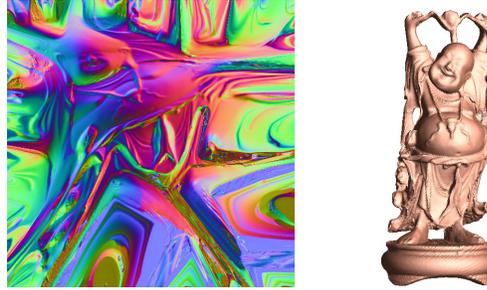


FIGURE 1.1 – Image géométrique du buddha et sa reconstruction. Pour fin de visualisation, les normales plutôt que les positions des sommets sont affichées. Image tirée de [GGH02].

Les *images géométriques* [GGH02] offrent une représentation alternative pour les maillages de triangles, qui unifie la géométrie et la connectivité d'un objet dans une structure similaire à une image. Dans cette représentation, chaque pixel (RGB) encode la position (XYZ) d'un sommet et chaque groupe de quatre pixels adjacents forme un quadrilatère. Pour encoder un objet dans une telle structure, un remaillage complètement régulier est nécessaire, ce qui implique de rééchantillonner la surface du modèle en passant par sa paramétrisation. Pour pouvoir paramétrer un objet dans le plan, on doit potentiellement lui appliquer une série de coupes pour le rendre topologiquement équivalent à un disque. Cette simple structure de données compacte encode encore implicitement la connectivité, réduit de beaucoup les accès aléatoires nécessaires au parcours des structures de données traditionnelles, et est bien adaptée au matériel graphique moderne.

L'une des difficultés avec les images géométriques vient de l'étirement et de la compression imposés par la paramétrisation. Les coupes appliquées à la surface ou la séparation de celle-ci en plusieurs cartes aide à réduire cette déformation, mais introduisent des discontinuités entre les différentes pièces. Par discontinuités, nous entendons des pixels ou l'accès à un pixel voisin requiert un saut vers une autre région de l'image.

Une autre solution consiste à utiliser une paramétrisation plus appropriée à la surface. **MMM** Comme noté par Praun et Hoppe [PH03] et par Sheffer *et al.* [SGD04],

la paramétrisation sphérique est souvent considérée plus naturelle pour les surfaces ne possédant pas de trous (genre 0). Nous croyons que ce type de paramétrisation peut aussi bénéficier à des surfaces de genre plus élevé. Ce mémoire montre comment amener les surfaces de genre plus élevé aux images géométriques sphériques.

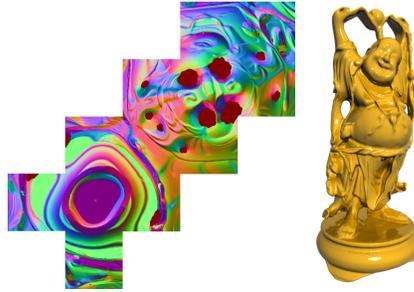


FIGURE 1.2 – Images géométriques sphériques du buddha sur le domaine HEALPix générées avec notre méthode. Pour fins de visualisation, les images des normales sont affichées.

Nous présentons un pipeline complet et robuste pour transformer les surfaces de genre arbitraire en images géométriques sphériques où les trous de la surface sont exprimés de façon explicite. Notre algorithme s’articule en trois étapes : (1) nous réduisons le genre de la surface à 0 pour la rendre homéomorphe à une sphère ; (2) nous paramétrisons cette surface sur la sphère ; et (3) nous rééchantillonons la paramétrisation pour créer les images géométriques.

Le reste du document est structuré comme suit. Les chapitres 2, 3 et 4 feront un rapide survol des trois principales branches de l’infographie qui servent de base à notre méthode, nommément, la topologie informatique, la paramétrisation de surface et les images géométriques. Du même jet, nous établirons également les travaux antérieurs de chacune de ces disciplines. Le chapitre 5 introduira notre méthode de réduction de genre et de paramétrisation sphérique hiérarchique, tandis que le chapitre 6 discutera des spécificités qu’implique l’échantillonnage d’une surface contenant des composantes du bord et de la reconstruction de la surface à partir des images géométriques. Le chapitre 7 sera consacré à la présentation et l’analyse de certains de nos résultats. Finalement, le chapitre 8 revisitera nos contributions et présentera certaines directions de recherche future.

## Chapitre 2

# Topologie informatique

*The rabbit-hole went straight on like a tunnel for some way, and  
then dipped suddenly down, so suddenly that Alice had not a  
moment to think about stopping herself before she found herself  
falling down a very deep well.*

—Lewis Carroll, *Alice in Wonderland*

Ce chapitre examine quelques notions élémentaires de topologie utilisées en infographie. Toutes ces notions seront abordées de façon intuitive et moins formelle, le but principal étant d'établir un vocabulaire pour le reste du document. La structure sera similaire à la présentation sur la topologie informatique du cours d'imagerie de Daniel Cohen-Or [CO04].

### 2.1 Définition

La topologie (du grec *topos*, place et *logos*, étude) est la branche des mathématiques qui étudie les propriétés d'un espace qui sont préservées lors d'une déformation continue [Wik08]. Bien que la topologie soit issue de la géométrie, elle ne s'intéresse pas aux propriétés métriques, telles la distance entre les points. Elle prône plutôt l'étude des propriétés qui décrivent comment un espace est assemblé, telles la connectivité et l'orientation. Un espace topologique est une structure mathématique permettant la définition formelle de concepts tels la convergence, la connectivité et la continuité.

Plus intuitivement, contrairement à la géométrie euclidienne qui dicte que deux objets sont équivalents si on peut passer de l'un à l'autre par une série de transformations rigides (rotations, translations, etc.), deux objets sont dits topologiquement *équivalents* s'ils ont simplement les mêmes propriétés (nombre de trous, de bords, etc.). Il est permis d'étirer et de tordre les objets, mais sans les rompre. Ce passage d'un espace à l'autre avec ces transformations se nomme *homéomorphisme*. Lorsqu'il est possible de passer d'un objet à un autre en appliquant ce type de transformations bi-continues, on dira que les deux objets sont *homéomorphes*. Par exemple, la tasse de la figure 2.1 est homéomorphe à un tore.

En imagerie, la topologie peut fournir de précieuses informations sur un modèle géométrique, comme d'en déterminer le nombre de trous et de bords. De façon générale, la topologie sera un outil précieux pour étudier la connectivité d'un objet.

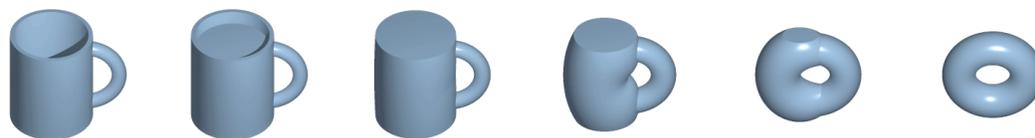


FIGURE 2.1 – Une tasse est homéomorphe à un tore puisqu'il existe une déformation continue pouvant passer de l'un à l'autre.

## 2.2 Surface

Avant de pouvoir étudier un objet dans son ensemble, on débutera par s'intéresser à ce qui se produit localement à sa *surface*. Dans l'espace euclidien tridimensionnel, on dira qu'une surface est quelque chose qui ressemble localement à un plan. De façon plus générale, on nommera *n-variété* sans frontière tout espace topologique qui ressemble localement à l'espace euclidien  $\mathbb{R}^n$ . Par exemple, une sphère est une 2-variété puisque pour une fourmi qui se balade sur sa surface, le voisinage ressemble à un plan. Par le même raisonnement, un cercle est une 1-variété puisque toute courbe observée de suffisamment près ressemble à une droite.

Puisque nous ne nous intéressons pas à la forme géométrique d'une surface, seulement trois informations suffisent à déterminer son type homéomorphique, soient le nombre de

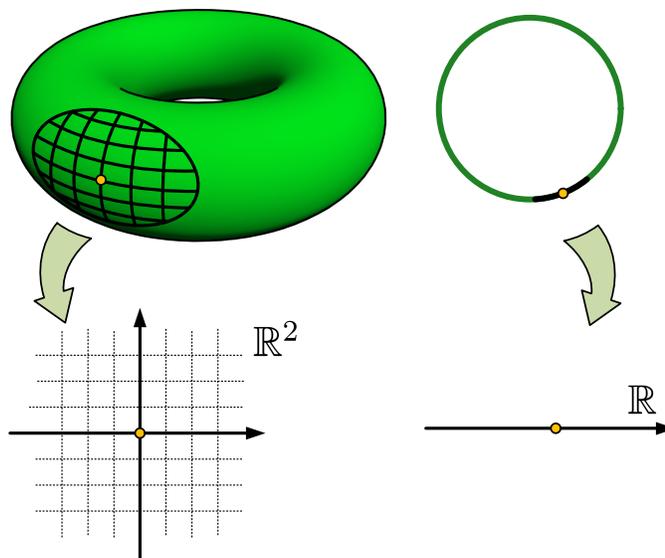


FIGURE 2.2 – Une sphère est une 2-variété alors qu'un cercle est une 1-variété.

composantes de bord qu'elle possède, son orientabilité, et sa caractéristique d'Euler.

Pour une 2-variété, tous les points dont le voisinage ressemble à un plan font partie de l'intérieur de la surface tandis que les points dont le voisinage ressemble à un demi-plan font partie du *bord*. Une *composante de bord* est un ensemble de points connectés faisant partie du bord. On dira d'une surface qui possède au moins un point sur le bord qu'elle est *ouverte*, et *fermée* si elle ne possède aucun point de bord. Dans ce document, nous utiliserons parfois l'abréviation CB pour désigner une composante de bord.

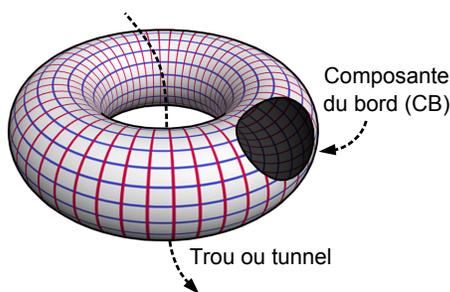
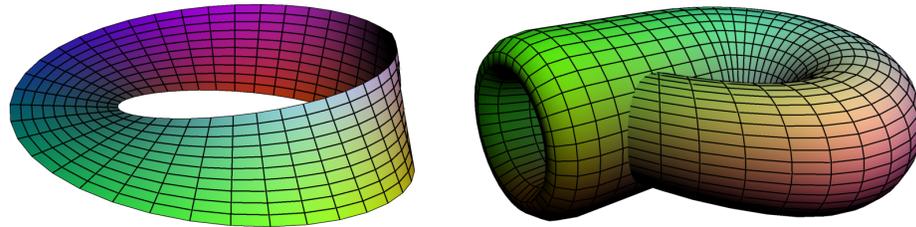


FIGURE 2.3 – Illustration du vocabulaire topologique utilisé dans ce document. Nous dirons que les lignes rouges sont des méridiens tandis que les bleues sont des parallèles.

Une surface est dite *orientable* si le concept de rotation horaire (ou anti-horaire) est maintenu de manière cohérente (*consistant*) sur toute la surface. De façon plus intuitive, une surface est orientable s'il est impossible pour un objet qu'on y glisse de revenir à la

même position, mais inversé (réflexion). La figure 2.4 montre deux exemples de surfaces non orientables. Pour un maillage de triangles, l'orientabilité globale peut être assurée en ordonnant dans la même direction tous les sommets à l'intérieur des faces (horaire ou anti-horaire). Il sera impossible pour une surface non orientable de faire une distinction claire entre l'intérieur et l'extérieur de celle-ci.

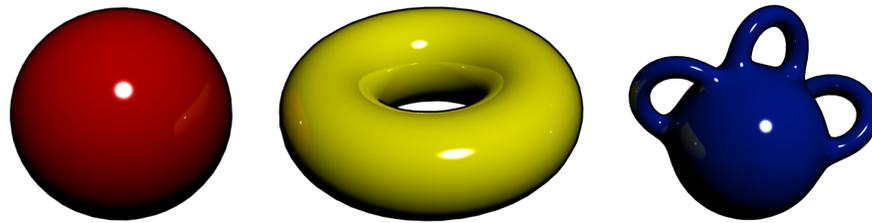


(a) Le ruban de Möbius

(b) La bouteille de Klein

FIGURE 2.4 – Deux exemples de surfaces non orientables.

De façon formelle, le *genre* (*genus*) est le nombre de coupes, le long de courbes fermées simples sans points communs, pouvant être appliquées à une surface sans la déconnecter. Concrètement, il s'agit du nombre de trous ou d'anses que possède un objet. Toute surface de genre  $n$  peut être vue comme une sphère sur laquelle sont greffés  $n$  anses ou tores (figure 2.5).



(a) Genre-0

(b) Genre-1

(c) Genre-3

FIGURE 2.5 – Objets de genres 0, 1 et 3.

Pour les maillages polygonaux, la *caractéristique d'Euler* est intimement liée au genre d'un objet. Originellement développée pour l'étude des polyèdres, elle s'avère être un outil puissant pour caractériser la structure des espaces topologiques. Elle est dénotée par la lettre grecque  $\chi$  et peut être évaluée par la simple formule

$$\chi = S - A + F \quad (2.1)$$

où  $S$ ,  $A$  et  $F$  sont respectivement le nombre de sommets, d'arêtes et de faces. Pour une surface orientée, elle peut également être calculée à partir du genre  $g$  et du nombre de composantes de bord  $b$ , tel que

$$\chi = 2 - 2g - b. \quad (2.2)$$

Donc, pour un maillage donné, il est possible d'en déterminer le genre à partir de sa caractéristique d'Euler :

$$g = \frac{(2 - b - \chi)}{2}. \quad (2.3)$$

Un tore peut être vu comme le produit de deux cercles et nous dirons de chacun de ces cercles qu'ils sont les générateurs du tore (figure 2.6). Nous utiliserons un vocabulaire analogue aux coordonnées géographiques et nommerons *méridiens* les lignes situées sur les plans de la surface de révolution et *parallèles* les lignes perpendiculaires à ces dernières. Sur la figure 2.3, les méridiens sont en bleu et les parallèles sont en rouge.

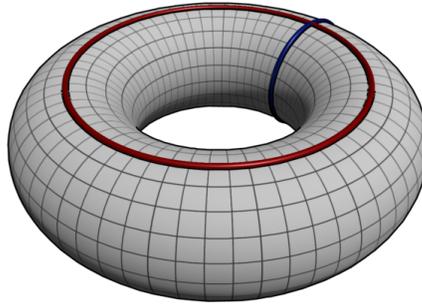


FIGURE 2.6 – Un tore est le produit de deux cercles générateurs. La ligne bleue est le générateur méridien tandis que la ligne rouge est le générateur parallèle.

## 2.3 Théorie de Morse

Bien que le nombre de composantes de bord et la caractéristique d'Euler permettent de déterminer le genre d'une surface, toutes ces informations ne disent rien sur la position ou la forme des trous et des anses. Ceci est normal, considérant que la topologie ne s'intéresse que peu à la géométrie de ces surfaces. Par contre, dans d'autres domaines, notamment en imagerie, il est tout à fait naturel de vouloir en connaître plus sur le comportement d'une surface. Par exemple, où est-ce que la surface se sépare et se rejoint pour former un trou ?

La théorie de Morse tente de répondre à ces questions en appliquant des notions de géométrie différentielle aux surfaces topologiques. Développée par Marston Morse au milieu du XXe siècle, cette technique étudie la topologie d'une surface en analysant le comportement des lignes de niveau d'une fonction définie sur celle-ci. Autrement dit, la théorie de Morse établit une relation entre les points critiques d'une fonction réelle sur la surface et la topologie de cette dernière. Plusieurs fonctions, dites *fonctions de Morse*, peuvent être utilisées pour effectuer cette tâche. Un point critique est un point où le gradient de la fonction est nul. Le *type* de point critique peut être déterminé en regardant le nombre de directions indépendantes où la fonction diminue. Plus formellement, le type de point critique est donné par le nombre de valeurs singulières négatives dans l'hessien de la fonction. Zéro, une ou deux valeurs négatives représentent respectivement un minimum, une selle (*saddle*) et un maximum.

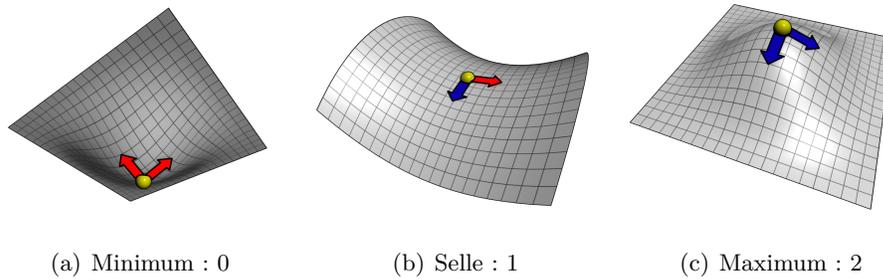


FIGURE 2.7 – Les trois types de points critiques sont identifiés par le nombre de directions négatives (en bleu) dans l'hessien.

Puisque cette technique s'illustre plus simplement par un exemple, regardons ce qui se produit lorsque nous l'appliquons à un tore posé verticalement. Nous utiliserons l'altitude de chaque point en guise de fonction de Morse. Avec cette fonction, quatre points critiques sont identifiés, soient un minimum, deux points de selle (*saddle point*) et un point maximum (figure 2.8).

Regardons les changements topologiques qui se produisent lorsque nous balayons la surface en suivant les courbes de niveau. La figure 2.9 illustre chaque étape du balayage.

1. Immédiatement au-dessus du point minimum, la surface a la forme d'une calotte.
2. En atteignant le premier point de selle, la surface se referme sur elle-même et prend la forme d'un cylindre.
3. La topologie demeure inchangée, les deux contours suivent les courbes de niveau.

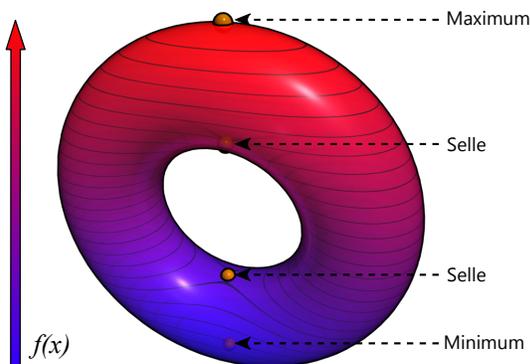


FIGURE 2.8 – Les courbes de niveau de la fonction d’altitude sur un tore et les quatre points critiques.

4. Les deux contours se touchent au second point de selle et le cylindre devient un tore perforé.
5. Cette topologie demeure inchangée.
6. Le point maximum est atteint et le tore se referme.

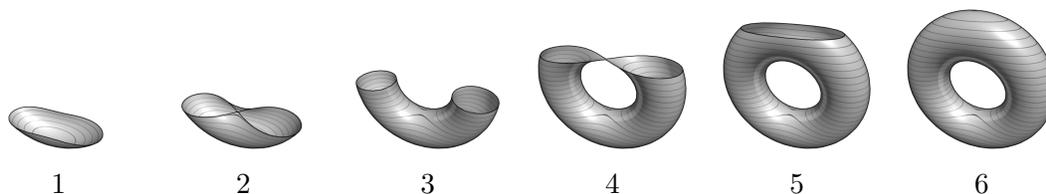


FIGURE 2.9 – Balayage du tore avec la fonction d’altitude.

De ce simple exemple, plusieurs idées viennent à l’esprit. D’abord, il est intéressant de noter que la topologie de la surface n’est pas modifiée entre deux points critiques. Une autre observation est que l’efficacité de cette procédure dépend fortement de la fonction de Morse choisie. Par exemple, positionner le tore à l’horizontal ne permettrait pas de correctement identifier les points critiques. Aussi, en utilisant une fonction simple comme l’altitude, il est possible d’avoir plus d’un point critique pour une même valeur de fonction donnée. Cette fonction doit donc être choisie sagement en fonction du type de surface à traiter.

Une *bonne* fonction de Morse doit être lisse ( $C^2$ ), éviter de créer des points critiques simultanés et tous les points critiques doivent être non-dégénérés. Par non-dégénérés, nous entendons que l’hessien de la fonction aux points critiques doit être non nul. Puis-

qu'en imagerie, nous travaillons avec des surfaces discrétisées en maillages polygonaux, il suffit d'associer une valeur de fonction de Morse à chaque sommet ou face et d'étudier les voisinages polygonaux pour trouver les points critiques.

Comme alternative à la fonction altitude de l'exemple précédent, Hilaga *et al.* [HSKK01] utilisent la distance géodésique sur la surface à partir d'un point de départ. Cette fonction possède l'avantage d'être moins sensible aux rotations que la fonction altitude, mais, tout comme cette dernière, risque de créer plusieurs points maximums lorsque la surface comporte plusieurs protubérances, ce qui peut gêner l'analyse. Ni *et al.* [NGH04] proposent une fonction de Morse lisse qui évolue entre des valeurs assignées sur des sommets contraints par l'utilisateur. En solutionnant une version relaxée de l'équation de Laplace, ils forcent la valeur de chaque autre sommet à être égale à la valeur moyenne de ses voisins. Ils proposent trois façons de calculer cette fonction et expérimentent avec différents poids. C'est cette fonction que nous préconiserons dans nos travaux, en nous en reparlerons à la section 5.3.1.

## 2.4 Graphe de Reeb

Quoique la théorie de Morse en elle-même est un outil puissant pour analyser la topologie d'une surface, elle échoue à en donner une bonne vue d'ensemble. En effet, les relations entre les différents points critiques peuvent être difficiles à établir. Pour pallier à ce problème, un autre outil est utilisé en combinaison avec la théorie de Morse, soit le *graphe de Reeb*.

Un graphe de Reeb est un graphe représentant la topologie d'une surface telle que décrite par une fonction de Morse. Dans ce type de graphe, les sommets représentent les points critiques de la surface et les arcs sont les composantes connectées entre les différents points critiques.

En regardant l'exemple de la figure 2.10, il apparaît évident que ce graphe peut fournir plus d'information que la théorie de Morse seule. Les points critiques de selle peuvent maintenant être classés en points de séparation et en points de fusion. Le nombre de boucles dans le graphe de Reeb est aussi égal au genre de la surface.

Pour un maillage de triangles, le graphe de Reeb peut être construit en balayant la surface et en ajoutant un sommet ou une face à la fois. Avec cette méthode, un ou

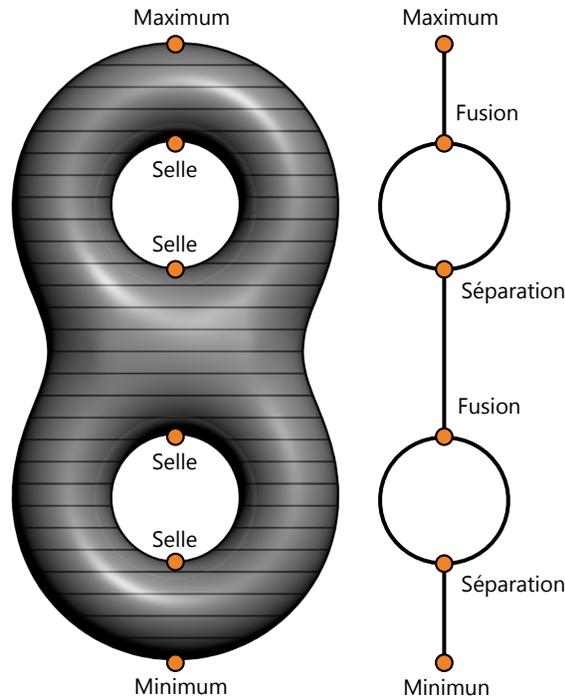


FIGURE 2.10 – Exemple de graphe de Reeb pour un 2-tore.

plusieurs contours évoluent sur la surface, se divisant aux points de séparation et se réunissant aux points de fusion. Un nouveau noeud est créé à chaque point critique et des arcs sont tracés vers les noeuds précédents. Cette stratégie a été employée par Zhang *et al.* [ZMT05] pour effectuer de la paramétrisation de surface sensible aux particularités de la géométrie. Nous utiliserons cette stratégie dans nos travaux (chapitre 5) puisqu'elle est simple et permet d'aisément incorporer (*embed*) le graphe à la surface du modèle.

Plus récemment une autre méthode pour calculer le graphe de Reeb a fait son apparition. Pascucci *et al.* [PSBM07] construisent le graphe à partir d'un flux de sommets et de triangles, et mettent à jour leur structure de données pour refléter la nouvelle topologie. Bien que cette méthode semble bien fonctionner, elle ne permet pas de positionner clairement les noeuds et les arcs à la surface du modèle géométrique. Leur méthode d'incorporation discrétise la fonction de Morse et ne fournit pas la précision dont nous avons besoin.

Les applications du graphe de Reeb sont multiples. Hilaga *et al.* [HSKK01] construisent une version hiérarchique du graphe pour étudier la similarité entre deux modèles. Le graphe d'un modèle géométrique fourni en entrée est utilisé pour effectuer une recherche

dans une base de données de modèles. Wood *et al.* [WHDS04] l'utilisent pour éliminer la topologie excessive d'un modèle, c'est-à-dire les petits trous indésirables qui sont souvent produits lors de la numérisation de modèles. Après avoir bâti le graphe de Reeb en utilisant un balayage aligné sur les axes, les minuscules trous sont localisés sur la surface en utilisant une approche volumétrique. Ils montrent des résultats de simplification topologique intéressants pour des modèles populaires et expérimentent aussi avec les images géométriques.

## Chapitre 3

# Paramétrisation de surface

*In our own time, algebra has become the most rarefied and demanding of all mental disciplines, whose objects are abstractions of abstractions of abstractions, yet whose results have a power and beauty that are all too little known outside the world of professional mathematicians. Most amazing, most mysterious of all, these ethereal mental objects seem to contain, within their nested abstractions, the deepest, most fundamental secrets of the physical world.*

—John Derbyshire, *Unknown Quantity*

Dans ce chapitre, nous explorerons les rudiments de la paramétrisation de surface. Nous ferons un rapide survol des techniques existantes pour obtenir et raffiner une paramétrisation en portant une attention particulière sur les deux domaines planaire et sphérique. La structure de cette section s’inspire librement des études de Floater et Hormann [FH05] et Sheffer *et al.* [SPR06].

### 3.1 Introduction

Une paramétrisation peut être vue comme une mise en correspondance une-à-une entre deux surfaces de topologies similaires. En imagerie, on s’intéresse habituellement à faire la correspondance entre la surface d’un modèle tridimensionnel et une autre surface quelconque que l’on nomme domaine de paramétrisation. Alors que traditionnellement

les modèles sont représentés par des maillages de triangles dans un espace 3D, le domaine est généralement planaire 2D. Nous verrons que celui-ci peut également être une surface quelconque. Lorsqu'une des surfaces est un maillage de polygones, l'opération est appelée paramétrisation de maillage (*mesh parameterization*). Cette correspondance permet d'utiliser le domaine comme un repère sur la surface du modèle pour effectuer une multitude d'opérations en imagerie et en traitement de maillage.

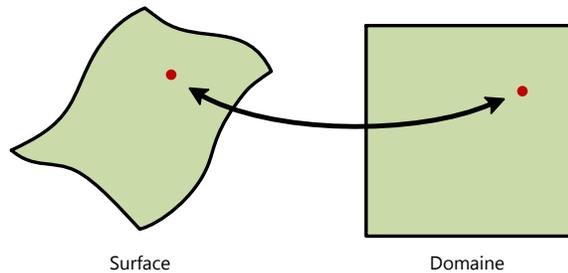


FIGURE 3.1 – Paramétrisation entre une surface et un domaine.

## 3.2 Origine

Le problème de la paramétrisation prend ses origines dans la cartographie. Bien que l'on retrace les origines de la cartographie aux hommes des cavernes, la cartographie moderne ne commence qu'un siècle av. J.-C. Armé de la notion récente que la Terre est une sphère et d'une estimation de sa circonférence, Claude Ptolémée (90-168 av. J.-C.) utilise des notions d'astronomie et de mathématiques pour cartographier le monde connu par ses contemporains. Ptolémée organise un système de méridiens et longitudes pour ensuite aplanir la surface sphérique de la terre en utilisant une projection en perspective. Il publie ses cartes dans un recueil de huit livres intitulé *Geographia* (figure 3.2).

Les premiers cartographes grecs et romains se sont rapidement butés au problème qu'est la représentation de la surface tridimensionnelle courbe de la Terre sur une carte planaire. De la même façon qu'il est impossible d'éplucher une orange en apposant aisément les pelures sur la table, il est impossible de cartographier même la plus petite région de la Terre sans introduire une distorsion. Cette distorsion peut affecter autant l'aire, la forme, la direction, la distance, que l'échelle des éléments sur la carte.

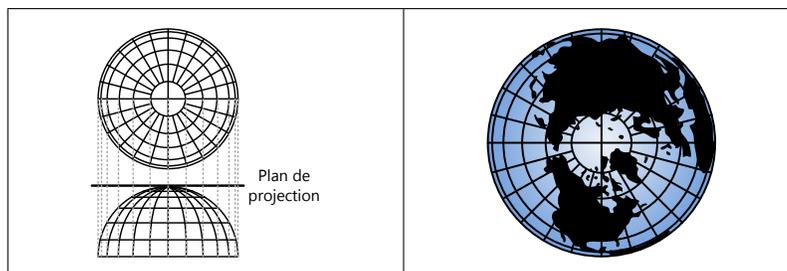
Pour pallier à ce problème, différentes projections ont été développées, chacune tentant de minimiser un certain type de distorsion ou un autre. La projection orthogra-



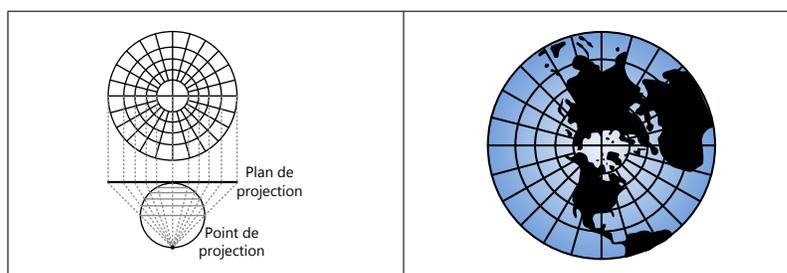
FIGURE 3.2 – Reconstruction de la carte du monde de Ptolémée tirée du livre «*The Outline of History*» de H.G. Wells.

phique (figure 3.3(a)), par exemple, est simplement une projection sur un plan tangent à la sphère. Elle déforme les aires plus on s'éloigne du centre de projection, mais préserve les distances aux parallèles. La projection stéréographique (figure 3.3(b)) est obtenue en sélectionnant un point de projection et en projetant l'hémisphère opposé sur son plan tangent. On dira de cette projection qu'elle constitue une *transformation conforme* puisqu'elle préserve les angles. La projection de Mercator (figure 3.3(c)) est obtenue en projetant d'abord la sphère sur un cylindre, qui lui peut être aisément développé sur un plan. Bien que cette projection ait tendance à élargir les formes plus on se rapproche des pôles, elle possède l'avantage de préserver les directions. Cette propriété en a fait la projection de choix pour les premiers navigateurs. La projection de Lambert (figure 3.3(d)) s'inspire du même principe, mais utilise un cône comme surface de projection. Il s'agit d'une projection *conforme* puisqu'elle préserve les angles.

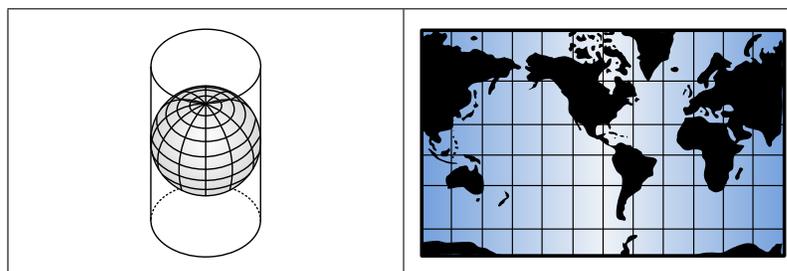
Bien que ces projections peuvent sembler arbitraires, elles font toutes appel au même principe fondamental, soit l'idée de projeter la terre sur une *surface développable*. Une surface développable est une surface qui peut être ouverte sur un plan sans aucune distorsion. Plus intuitivement, il s'agit d'une forme dont on peut faire rouler toute la surface sur un plan sans la faire «glisser». Les trois surfaces développables les plus communes sont le plan, le cylindre et le cône, et il n'est pas surprenant qu'elles soient les trois surfaces utilisées dans les quatre projections décrites ci-haut.



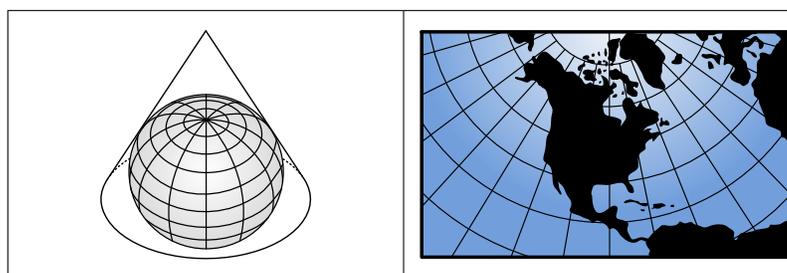
(a) Projection orthographique



(b) Projection stéréographique



(c) Projection de Mercator



(d) Projection de Lambert

FIGURE 3.3 – Projections de la Terre.

### 3.3 Imagerie

En imagerie, les premiers développements sur la paramétrisation de surface sont surtout motivés par les avancées faites sur le placage de textures (*texture mapping*). Pour ajouter du réalisme à une scène, on applique des images 2D à la surface des objets pour simuler les détails fins, ce qui serait impossible de représenter sans l'ajout massif de polygones à la scène. Pour ce faire, chaque point  $(x, y, z)$  de la surface du modèle 3D représenté par un maillage de polygones doit être associé à un point  $(u, v)$  de l'image 2D (domaine) à appliquer. Autrement dit, une paramétrisation entre la surface du modèle et l'image à appliquer doit être disponible. Certaines formes développables ou certains modèles paramétriques tels les surfaces bi-cubiques peuvent parfois jouir d'une paramétrisation naturelle, mais ceci n'est pas le cas pour un maillage de polygones quelconque.

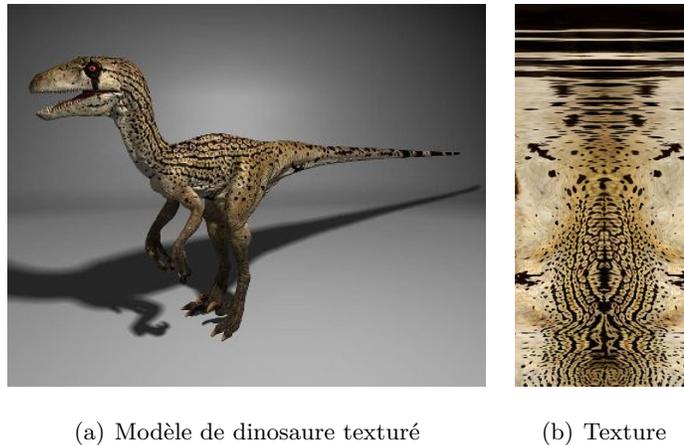


FIGURE 3.4 – Placage de textures. Image courtoisie de Daren Horley.

Le problème qui nous intéresse est donc de définir une paramétrisation valide pour un maillage triangulaire donné. Pour ce faire, il est d'abord nécessaire de bien définir la terminologie utilisée.

Plus formellement, une paramétrisation définit une correspondance  $F : D \leftrightarrow S$  entre les régions du domaine  $D \subset \mathbb{R}^2$  et de la surface  $S \subseteq \mathbb{R}^3$  du modèle tridimensionnel. Ceci est possible grâce au fait que les modèles tridimensionnels utilisés en infographie sont généralement définis par leur surface 2D (2-variété) représentée dans un espace 3D. Dans le cas d'un maillage de triangles, cette correspondance peut être simplement définie en associant une position  $(u, v)$  du domaine  $D$  à chaque sommet. Puisqu'un maillage de

triangles est essentiellement une fonction linéaire par morceaux, la correspondance entre un point sur la surface  $S$  et le domaine  $D$  peut se faire par une simple transformation affine. Cette correspondance permet d'effectuer diverses opérations comme si la surface était plane.

L'une des propriétés importantes d'une paramétrisation est sa *bijektivité*, c'est-à-dire qu'un point sur la surface  $S$  n'est associé qu'à un seul point du domaine  $D$ , et vice versa. Une façon simple de vérifier de la bijectivité d'une paramétrisation est de s'assurer qu'elle ne contient aucun croisement d'arêtes ou aucune face renversée dans  $D$ . Comme nous le verrons, certaines classes d'algorithmes relaxent cette condition de bijectivité.

Comme mentionné précédemment, le domaine est typiquement plane, ce qui implique que seuls les modèles *homéomorphes* à un disque peuvent être paramétrés sur ce domaine. Autrement dit, seuls les modèles possédant une seule composante de bord sont compatibles avec un domaine plane. Ces notions topologiques seront approfondies au chapitre 4. Ceci étant dit, pour la discussion, nous assumerons que tout modèle géométrique peut être converti en un disque en appliquant une série de coupes. Par exemple, la texture de la figure 3.4(b) ne peut être appliquée au dinosaure que si on *coupe* ici, le long de son ventre. Un autre domaine de paramétrisation fréquemment utilisé en imagerie est la sphère. Ce domaine est plus naturel pour paramétrer des modèles sans trous ni bords.

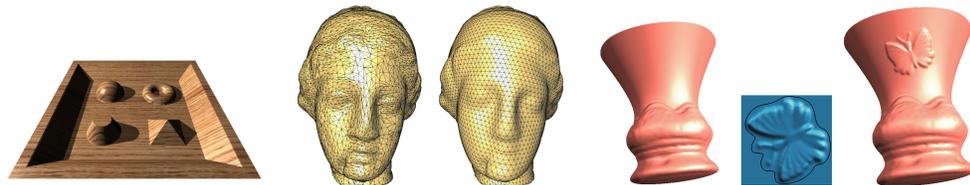
Tout comme les projections cartographiques altèrent la forme des continents, la forme des triangles dans la paramétrisation risque de différer de ceux sur le modèle, et cette distorsion affecte l'aire et les angles. On dit d'une paramétrisation qu'elle est *isométrique* si elle préserve les distances, c'est-à-dire deux points sont séparés par la même distance sur la surface que sur le domaine. Quoique cette propriété soit hautement désirable, elle n'est vraie que pour les surfaces développables telles les cylindres. Une paramétrisation est dite *conforme* si elle préserve les angles entre tous les segments, et *équivalente* si toutes les aires du domaine et de la surface sont proportionnelles. Une paramétrisation isométrique est à la fois conforme et équivalente.

Sachant qu'il est impossible de créer une paramétrisation parfaite pour un maillage de triangles arbitraire, l'objectif est d'en créer une qui minimise l'une ou l'autre de ces distorsions, ou une combinaison des deux. Le but est de créer un algorithme qui, prenant

un modèle, crée une paramétrisation qui minimise une fonction d'énergie basée sur une certaine métrique. Il existe plusieurs algorithmes capables d'effectuer cette tâche autant dans le domaine planaire que sphérique et nous verrons les plus importants aux sections 3.5 et 3.6.

### 3.4 Applications

Cela dit, les applications de la paramétrisation vont bien au-delà du placage de textures. Bien que le placage de textures permette de simuler l'illusion de détail en préservant le nombre original de polygones, l'illusion est ruinée lorsqu'on observe le modèle de trop près, puisque la texture ne produit aucun effet de relief. Le placage de normales (*normal mapping*) stocke la normale dans une texture, permettant ainsi à la texture d'interagir avec l'éclairage de la scène. Plus récemment, des techniques telles le *parallax mapping* [KTI<sup>+</sup>01] et le *relief mapping* [OBM00] utilisent de sophistiqués *pixel shaders* pour localement effectuer un lancer de rayons dans une texture contenant les détails de surface, permettant ainsi de simuler un effet de profondeur sur la surface du modèle. En plus d'appliquer une texture, certains algorithmes utilisent le repère fourni par la paramétrisation pour synthétiser directement la texture sur le modèle.



(a) *Relief mapping* (b) Remaillage [ACSD<sup>+</sup>03] (c) Transfer de détails [BMBZ02] [OBM00]

FIGURE 3.5 – Quelques applications de la paramétrisation.

Une autre application est le remaillage. Cette technique utilise la paramétrisation pour rééchantillonner la surface d'un modèle afin d'obtenir un maillage plus régulier ou exhibant certaines propriétés. Cette technique est à la base des images géométriques (*geometry images*) qui seront couvertes au prochain chapitre et dans la suite du mémoire. Ce remaillage peut ensuite être compressé aisément.

D'autres applications incluent le transfert de détails, l'achèvement de modèles (*mesh*

*completion*), l'édition de modèles ainsi que la visualisation médicale. Pour une liste plus exhaustive, le lecteur est référé à l'article de Sheffer *et al.* [SPR06].

### 3.5 Paramétrisation planaire

Dans cette section nous présenterons différents algorithmes pour créer une paramétrisation planaire. Tout comme Sheffer *et al.* [SPR06], nous classifions les algorithmes par le type de distorsion qu'ils tentent de minimiser. La figure 3.7 contient des exemples de paramétrisations pour la majorité des algorithmes présentés.

#### 3.5.1 Paramétrisation conforme

Le premier algorithme de paramétrisation nous vient de la théorie des graphes. Bien que la méthode de Tutte [Tut63] ait d'abord été introduite pour dessiner des graphes planaires, elle servira de base à bien des algorithmes de paramétrisation à venir. Cette méthode stipule que tout graphe planaire peut être représenté dans le plan en distribuant les noeuds de la frontière du graphe sur un polygone strictement convexe et en positionnant tous les autres noeuds au barycentre de leurs voisins. Pour ce faire, Tutte résout un système d'équations de la forme

$$\mathbf{L}\mathbf{u} = \mathbf{0}, \mathbf{L}\mathbf{v} = \mathbf{0} \quad (3.1)$$

où

$$L_{ij} = \begin{cases} -\sum_{k \neq i} w_{ik} & i = j \\ w_{ij} & (i, j) \in E \\ 0 & \text{sinon.} \end{cases} \quad (3.2)$$

L'ensemble  $E$  représente les arêtes du graphe. Le système d'équations est résolu indépendamment pour les coordonnées  $u$  et  $v$ . Dans la méthode originale, tous les poids  $w_{ij}$  sont mis à 1, positionnant ainsi chaque noeud interne à la position moyenne de ses voisins. Bien que cette méthode garantisse de générer une paramétrisation valide lorsque la condition de convexité de la frontière est respectée, la paramétrisation résultante laisse souvent à désirer. En effet, en minimisant la longueur des arêtes du graphe, cette procédure ne minimise ni la distorsion d'aire, ni celle d'angle.

Inspiré de ce résultat, Floater [Flo97] généralise la méthode et relaxe la condition de stricte convexité de la bordure. Il réalise qu'il est possible d'obtenir une paramétrisation

bijective en utilisant n'importe quels poids  $w_{ij}$  pour les arêtes, tant que ceux-ci sont positifs et que la matrice  $\mathbf{L}$  est symétrique ( $w_{ij} = w_{ji}$ ). Il démontre aussi que la procédure fonctionne tout aussi bien lorsque la bordure contient des sommets colinéaires. En expérimentant avec différents poids, il propose des poids qui préservent la forme des triangles (*shape-preserving parametrization*). Quelques années plus tard, il raffine son travail en introduisant les coordonnées en valeur moyenne (*mean-value coordinates* ou *MVC*) [Flo03], qui produisent des résultats similaires, mais qui sont beaucoup plus simples. Cette idée d'utiliser des poids associés aux angles des triangles avait déjà été exploitée quelques années auparavant par Eck *et al.* [EDD<sup>+</sup>95]. Par contre, leur méthode pouvait parfois générer des poids négatifs pour certaines configurations, ne garantissant plus la bijectivité de la paramétrisation.

Toutes les méthodes décrites ci-haut sont appelées *méthodes à combinaison convexe*, puisque la position de chaque sommet correspond à une somme pondérée de ses voisins. Ces méthodes requièrent toutes une frontière convexe et fixe, ce qui peut parfois limiter la qualité de la paramétrisation. La méthode *MIPS* (*Most Isometric ParametrizationS*) de Hormann et Greiner [HG00] lève cette contrainte en optimisant chaque sommet individuellement. Ils proposent une fonction d'énergie qui évalue localement la conformalité d'un triangle. L'algorithme démarre avec une paramétrisation de base, telle celle de Floater [Flo97], et optimise localement chaque sommet. Les sommets situés sur la frontière sont optimisés de la même façon que les autres, permettant à la frontière d'évoluer pendant la procédure de minimisation. Ceci garantit une paramétrisation bijective qui minimise l'énergie conforme mais à un coût computationnel beaucoup plus élevé que d'autres méthodes.

L'un des problèmes qui se produit lorsqu'on tente d'aplanir une surface triangulée est que, contrairement à une surface continue, la somme des angles autour d'un sommet n'est pas égale à  $2\pi$ . Par exemple, le sommet d'une pyramide aura une somme d'angles supérieure à  $2\pi$ . La méthode d'aplatissement basée sur les angles (*angle-based flattening* ou *ABF*) de Sheffer et De Sturler [SS00] s'attaque directement à ce problème en tentant de trouver un facteur d'échelle pour les angles de chaque sommet. Ils formulent un problème d'optimisation avec contraintes qu'ils résolvent pour finalement déduire les positions  $(u, v)$  finales de chaque sommet. Tout comme la méthode précédente, cet algorithme peut fonctionner avec une frontière libre, mais peut contenir des chevauche-

ments de triangles à grande échelle. Une version plus robuste (*ABF++*) et hiérarchique de l'algorithme est présentée par Sheffer *et al.* [SLMB05].

Finalement, deux méthodes semblables permettant de formuler linéairement une paramétrisation avec une frontière libre ont été introduites récemment. Les méthodes de Desbrun *et al.* [DMA02] et de Levy *et al.* [LPRM02] tentent de minimiser l'énergie conforme au sens des moindres carrés. Pour forcer le système à ne posséder qu'une seule solution, deux sommets sont maintenus fixes pendant la résolution du système. Bien que ces méthodes soient radicalement plus rapides que les autres méthodes à frontière libre, la solution trouvée peut contenir des faces renversées.

### 3.5.2 Paramétrisation équivalente

Très peu d'efforts semblent avoir été consacrés au développement d'algorithmes capables de fidèlement reproduire les aires des triangles dans l'espace de paramétrisation. La raison est que la minimisation de la distortion d'aire impose moins de contraintes et le nombre de solutions possibles pour un même problème est très élevé. Par exemple, plusieurs paramétrisations équivalentes peuvent être obtenues en faisant tourner sur eux-mêmes, les cercles de la figure 3.6.

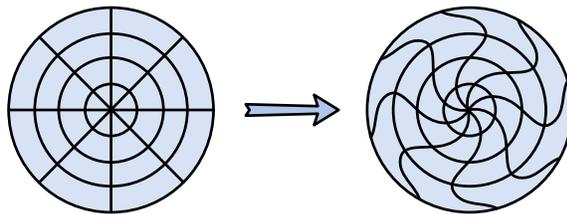


FIGURE 3.6 – Les polygones ont la même aire dans les deux paramétrisations.

L'algorithme développé par Degener *et al.* [DMK03] est un des seuls qui tentent de résoudre le problème. L'algorithme peut être vu comme une version hiérarchique de la méthode *MIPS*, mais utilisant une métrique mesurant la distorsion d'aire au lieu de la distorsion d'angle. Une séquence de maillages progressifs est créée et une certaine proportion de sommets est ajoutée à chaque itération. De plus, à chaque itération, chaque sommet est optimisé un-à-un en recherchant le minimum local de la fonction d'énergie.

### 3.5.3 Compromis

Une dernière classe d'algorithmes favorise un compromis entre la distorsion d'aire et d'angle. Ceci place un plus grand nombre de contraintes sur la solution et facilite la résolution. Les deux métriques  $L_2$  et  $L_\infty$  introduites par Sander *et al.* [SGSH02] sont un exemple de compromis et sont maintenant fréquemment utilisées comme mesures de distorsion. Ils nomment cette distorsion l'*étirement géométrique*. Utilisant une méthode d'optimisation hiérarchique similaire à *MIPS* [HG00], les temps d'exécution sont comparables aux autres méthodes non linéaires. Récemment, une implémentation rapide de plusieurs ordres de magnitude a été proposée par Yoshizawa *et al.* [YBS04]. Leur méthode débute avec une paramétrisation de Floater [Flo97], et mesure l'étirement géométrique autour de chaque sommet, modifient graduellement les poids du système d'équations et répètent le processus tant que l'énergie diminue.

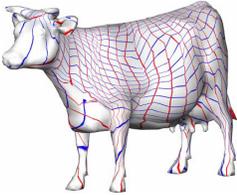
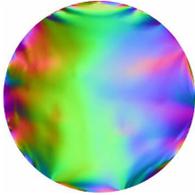
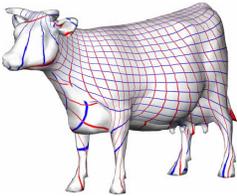
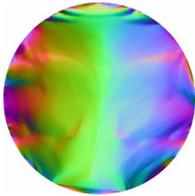
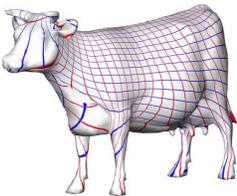
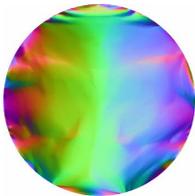
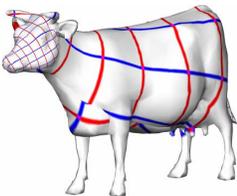
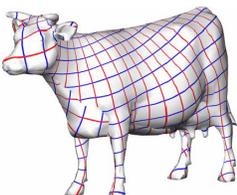
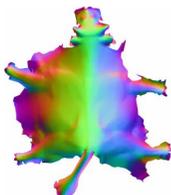
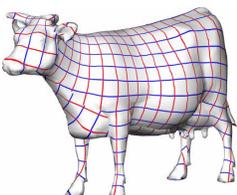
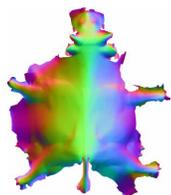
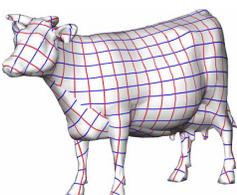
Tutte [Tut63]		
Eck <i>et al.</i> [EDD <sup>+</sup> 95]		
<i>Mean Value Coordinates</i> [Flo03]		
<i>Least Squares Conformal Maps</i> [LPRM02]		
<i>MIPS</i> [HG00]		
<i>ABF++</i> [SLMB05]		
<i>Geometric Stretch</i> [SSGH01]		

FIGURE 3.7 – Exemples de paramétrisations planaires. Les images sont tirées de [SPR06].

### 3.6 Paramétrisation sphérique

Le domaine sphérique n'a reçu passablement plus d'attention qu'au cours des dernières années. Les quelques méthodes qui ont vu le jour jusqu'à maintenant, à l'exception d'une, ne sont pas aussi matures que leur équivalent planaire. Le problème provient de la non-linéarité inhérente à la paramétrisation sphérique et du manque de conditions de bord. Contrairement aux méthodes planaires où la frontière est bien définie et même parfois fixe, les points sur la sphère sont libres de dériver, décuplant ainsi le nombre de solutions possibles. Pire encore, ce manque de points d'ancrage permet à la paramétrisation de dégénérer vers la solution triviale où tous les points se réunissent au même endroit. La figure 3.8 montre un exemple d'optimisation où l'un des triangles traverse l'équateur pour ensuite permettre à la configuration de s'effondrer, minimisant effectivement l'énergie à zéro.

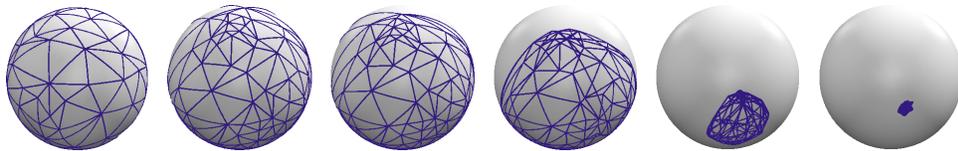


FIGURE 3.8 – Séquence d'optimisation de paramétrisation sur la sphère. Image tirée de [FSD05].

L'un des premiers algorithmes proposés est celui de Alexa [Ale99]. Après avoir trouvé un centre de projection valide, les sommets du modèle à paramétrer sont projetés sur la sphère unitaire et relaxés pour minimiser l'énergie. Pour tenter d'éviter une configuration dégénérée et prévenir les inversions de faces, certains sommets sont maintenus fixes pendant la relaxation. Même avec ces ancrages, la séquence d'optimisation peut quand même diverger. Dans ce cas l'algorithme doit revenir en arrière et essayer avec de nouveaux ancrages.

Une grande classe d'algorithmes essaient de résoudre d'abord le problème en 2D pour ensuite projeter cette solution sur la sphère unitaire. Haker *et al.* [HAT<sup>+</sup>00] génèrent d'abord une paramétrisation planaire, qu'ils projettent sur la sphère en utilisant une projection stéréographique. Puisque la projection stéréographique n'est pas bijective pour un maillage de triangles, cette procédure, en plus de créer des faces renversées, hérite des distorsions liées à la projection. Isenburg *et al.* [IGG01] coupent le modèle

en deux, distribuent la frontière sur le cercle unitaire et optimisent les deux disques résultants. Récemment, Saba *et al.* [SYGS05] reprennent la même idée et tentent d’optimiser la paramétrisation résultante en déplaçant les sommets un-à-un sur la sphère (figure 3.9). Zayer *et al.* [ZRS06] utilisent deux pôles spécifiés par l’usager et coupent le modèle entre ces deux pôles. Ils résolvent le problème sur un rectangle en coordonnées curvilignes et relaxent ensuite la paramétrisation sur la sphère.

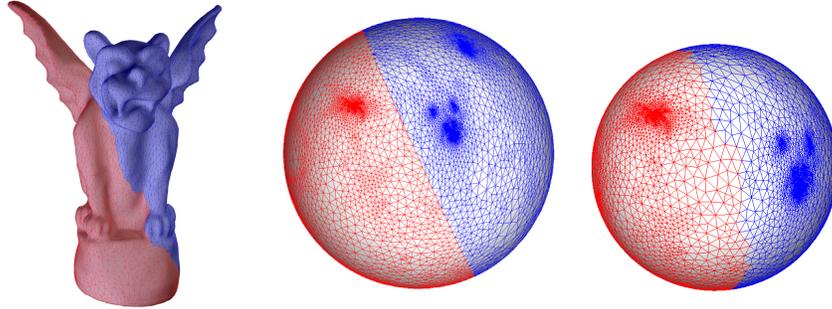


FIGURE 3.9 – Exemple de paramétrisation sphérique de Saba *et al.* [SYGS05]. Les deux moitiés du modèle géométrique sont paramétrées dans le plan, projetées sur la sphère et ensuite optimisées sur celle-ci.

Certains algorithmes qui fonctionnaient bien dans le domaine planaire ont été adaptés au domaine sphérique avec plus ou moins de succès. Praun et Hoppe [PH03] adaptent leur métrique  $L_2$  aux triangles sphériques. Ceux-ci ne possédant plus les propriétés des triangles plats, la métrique doit donc être intégrée numériquement, ce qui nécessite une subdivision très fine de chaque triangle lors de l’évaluation. Tout comme la technique originale, une séquence de maillages progressifs est créée. La version la plus simple du modèle étant un tétraèdre, celui-ci est aisément projeté sur la sphère. Les sommets sont insérés au centroïde de leurs voisins, puis optimisés un-à-un. Cette méthode crée une paramétrisation bien adaptée pour le remaillage, mais avec un temps de calcul plutôt élevé. Sheffer *et al.* [SGD04] adaptent aussi leur méthode *ABF* au domaine sphérique avec un succès mitigé.

Finalement, Friedel *et al.* [FSD05] étudient les problèmes de divergences observés lors de la paramétrisation sphérique et proposent un facteur de correction  $d_{min}^{-2}$  qui permet d’appliquer les énergies planaires classiques au domaine sphérique. Ce facteur de correction pénalise les grands triangles, faisant tendre leur énergie vers l’infini plus

ils s'approchent du centre de la sphère, empêchant ainsi un de ceux-ci de traverser l'équateur. Il dérivent les formes analytiques de quatre énergies typiquement utilisées dans le domaine planaire, celles de soit Tutte, Dirichlet (angle), la différence d'aire et l'étirement géométrique (*geometric stretch*). Ils expérimentent avec une énergie qui combine l'aire et les angles. Ces énergies sont ensuite dérivées et fournies à un puissant résolveur non linéaire qui réussit à produire une paramétrisation de grande qualité ne contenant aucune inversion de faces. Cette méthode sera privilégiée pour nos travaux pour sa rapidité et sa capacité à fournir une paramétrisation qui combine aire et angles.

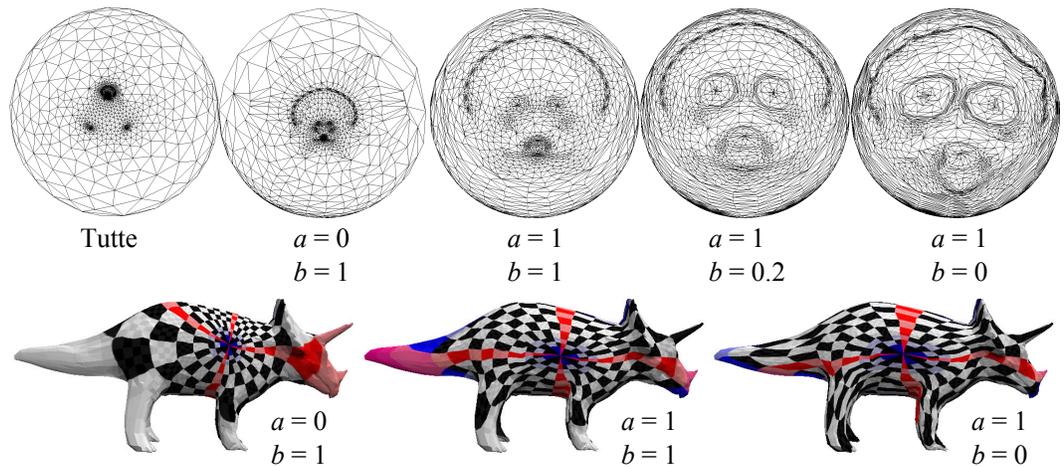


FIGURE 3.10 – Exemples de paramétrisations sphériques de Friedel *et al.*. Images tirées de [FSD05].

## Chapitre 4

# Images géométriques

*Throughout these tales it becomes clear that Man was running a race, if not with himself, then with some imagined follower who pressed close upon his heels, breathing on his back. Man was engaged in a mad scramble for power and knowledge, but nowhere is there any hint of what he meant to do with it once he had attained it.*

—Clifford D. Simak, *City*

Ce chapitre introduit le concept d'images géométriques (*geometry images*) et leurs différentes applications. Immédiatement après son introduction en 2002 par Gu *et al.* [GGH02], cette technique a reçu passablement d'attention et a déjà été appliquée à plusieurs branches de l'imagerie. Ce mémoire repose sur ce concept et en augmente la flexibilité. Ainsi, en plus de revoir rapidement la méthode originale, nous ferons état des avancés récentes et énumérerons les nombreuses applications qu'elles permettent.

### 4.1 Représentation de maillages

La vaste majorité des modèles géométriques utilisés en infographie emploient une représentation surfacique, c'est-à-dire qu'ils sont définis uniquement par leur surface. Les maillages de polygones, dont surtout les triangles, sont probablement la représentation surfacique la plus populaire, offrant un bon compromis entre la rapidité de rendu sur les cartes graphiques et la facilité d'édition grâce à de nombreux outils développés au fil des

années. Bien qu'il existe plusieurs structures de données très efficaces pour contenir un maillage de triangles, au plus simple, on peut le représenter par une série de sommets positionnés dans l'espace 3D, réunis en groupes de trois pour former des faces où chaque sommet et chaque face sont indexés dans un tableau tel qu'illustré à la figure 4.1. Une telle structure permet à chaque sommet d'avoir un nombre variable de voisins, on la dira donc *irrégulière*.

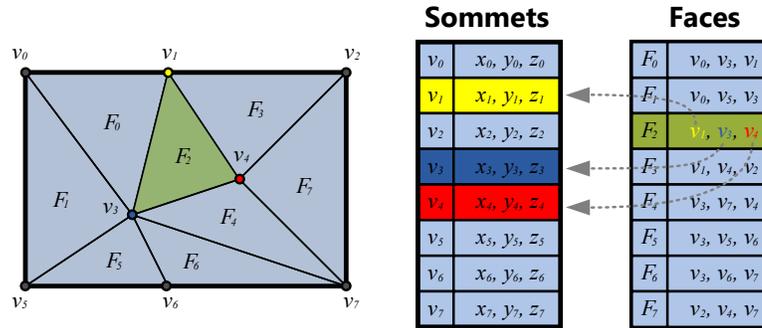


FIGURE 4.1 – Exemple de maillage de triangles et représentation interne.

Bien que cette structure soit extrêmement simple, en effectuer le rendu peut être assez coûteux. En effet, le rendu de chaque face requiert trois accès aléatoires (*random access*) à la mémoire où sont stockées les positions des sommets. Le problème se complique encore plus si une texture est appliquée au modèle. Les coordonnées de texture de chaque sommet pointent vers une région arbitraire de la texture, augmentant ainsi le nombre d'accès aléatoires. Les processeurs graphiques modernes réussissent à mitiger ce coût en conservant les sommets et les pixels des textures récemment utilisés en mémoire cache (*vertex caching* et *texture caching*). Pour maximiser l'efficacité de cette mise en cache, il existe différents algorithmes pour réorganiser et regrouper les triangles en éventails (*triangle fans*) et en chaînes (*triangle strips*). Ces algorithmes réduisent effectivement la mémoire requise et exploitent la localité des sommets, augmentant l'efficacité de la mise en cache.

Pour réduire encore plus ces accès aléatoires, on peut stocker notre maillage dans une structure beaucoup plus régulière. En 2002, Gu *et al.* [GGH02] s'inspirent de l'organisation des images elles-mêmes et proposent d'encoder un maillage de triangles dans un tableau 2D où chaque *pixel* représente non pas une couleur (RGB), mais la position d'un sommet (XYZ), et où chaque groupe de quatre pixels définit une face (figure 4.2). Ils

nomment ce type d'image une *image géométrique*. Puisque chaque face possède quatre sommets, il s'agit d'un quadrilatère qui peut aisément être traité comme deux triangles pour éviter des faces non planaires. Dans cette représentation, chaque sommet est uniquement connecté à ses quatre voisins immédiats, soient les *pixels* situés en haut, en bas, à gauche et à droite. Un maillage où chaque sommet possède un nombre fixe de voisins est dit *régulier*.

Une telle représentation possède plusieurs avantages dont celui de simplifier énormément le pipeline de rendu en éliminant plusieurs indirections qui étaient nécessaires pour un maillage irrégulier. De plus, chaque face étant formée de quatre pixels situés très près en mémoire les uns des autres, l'efficacité de la mise en cache dans un *rendering pipeline* traditionnel est assurée. La connectivité des arêtes et des faces n'a plus à être définie explicitement, ce qui en simplifie son parcours et réduit d'autant plus la mémoire nécessaire. Autrement dit, la position des sommets et la connectivité des faces sont ainsi unifiées dans une seule structure simple et efficace. Le fait que les sommets soient stockés dans une grille rectangulaire implique une paramétrisation naturelle dans le domaine planaire. Chaque pixel possède implicitement une coordonnée  $(u, v)$  unique, qui peut être utilisée pour accéder directement aux données stockées dans une texture (image). Finalement, cette représentation hérite de plusieurs des propriétés des images traditionnelles, permettant ainsi d'utiliser beaucoup d'algorithmes de traitement d'images existants, dont la compression, le lissage, le rehaussement, etc. Elle est aussi une candidate idéale pour être accédée efficacement par matériel graphique moderne dans un *shader* de pixels ou sommets.

Modifier la connectivité d'un maillage pour en obtenir un régulier peut être fait en effectuant un *remaillage*. Un remaillage est essentiellement un rééchantillonnage de la surface d'un modèle géométrique 3D pour tenter d'améliorer certaines propriétés du maillage. Par exemple, certains algorithmes de remaillage tentent de réduire le nombre de polygones tandis que d'autres cherchent à obtenir une meilleure distribution des sommets sur la surface. Dans le cas qui nous intéresse, on désire effectuer un remaillage qui fera en sorte que chaque sommet du nouveau modèle aura exactement quatre voisins et que ceux-ci s'inséreront aisément dans un tableau 2D (image). Nous verrons à la section 4.2 comment l'algorithme original de Gu *et al.* [GGH02] résout ce problème et nous verrons les améliorations proposées à ce jour à la section 4.3.

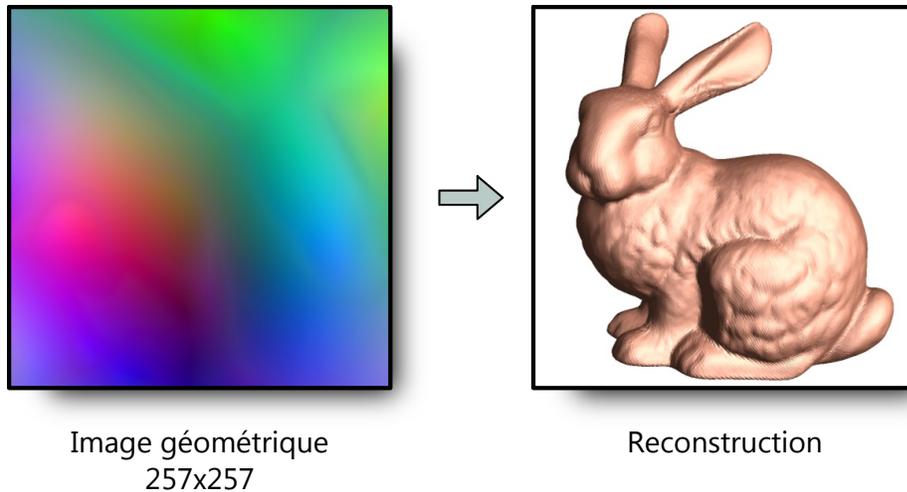


FIGURE 4.2 – Exemple d’une image géométrique. A gauche, la couleur (RGB) encode la position (XYZ) du maillage associé au modèle géométrique de droite. Images tirées de [GGH02].

## 4.2 Images géométriques traditionnelles

Dans cette section, nous verrons plus en profondeur la méthode originale des images géométriques telle que présentée pour la première fois [GGH02]. Cette démarche nous permettra de bien identifier les concepts, les défis et les problèmes associés à la génération des images géométriques.

### 4.2.1 Survol de l’algorithme

La figure 4.3 donne un aperçu des étapes à suivre pour encoder une image géométrique. L’algorithme reçoit un modèle géométrique en entrée et génère une image géométrique en sortie.

1. Le modèle est découpé pour lui donner une topologie de disque.
2. La bordure du modèle découpé est distribuée sur le carré unitaire et l’intérieur est paramétré dans le domaine planaire.
3. La paramétrisation résultante est échantillonnée avec une grille régulière et chaque échantillon est stocké dans l’image géométrique.
4. Le modèle géométrique rééchantillonné peut maintenant être reconstruit à partir de l’image géométrique.

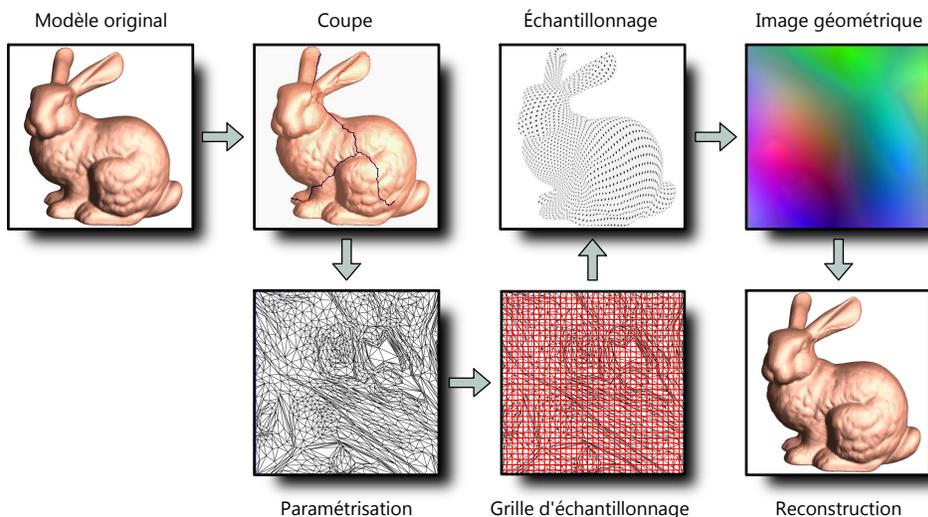


FIGURE 4.3 – Survol de l’algorithme original des images géométriques. Images tirées de [FSD05].

#### 4.2.2 Coupe

Le modèle en entrée est une 2-variété ( $2$ -*manifold*) orientable pouvant contenir un bord. Autrement dit, le modèle initial peut contenir des trous tel un tore ou des composantes de bord comme le haut et le bas d’un cylindre ouvert. Afin de pouvoir aplanir le modèle dans le domaine de paramétrisation, il est nécessaire de lui donner une topologie de disque. Ceci implique de couper judicieusement le modèle pour éliminer tous les tunnels et n’obtenir qu’une seule composante de bord continue. Cette composante de bord pourra ensuite être répartie sur le contour du carré unitaire pour démarrer la paramétrisation.

La figure 4.4 montre comment deux projections cartographiques coupent le globe terrestre pour obtenir une seule composante de bord continue et pouvoir ensuite l’aplanir. Dans le premier cas, une simple coupe du pôle Nord au pôle Sud est utilisée, tandis qu’un réseau plus sophistiqué est utilisé pour le deuxième. De façon générale, n’importe quelle arborescence de coupe permettra d’ouvrir un modèle sans trous ni bord. Bien que toutes ces coupes ne créent qu’une seule bordure, elles ne résulteront pas en la même qualité de paramétrisation.

Pour les modèles possédant des tunnels, la coupe doit inclure les générateurs définissant tous les tunnels. De façon générale, un modèle possédant  $g$  tunnels a  $2g$  générateurs.

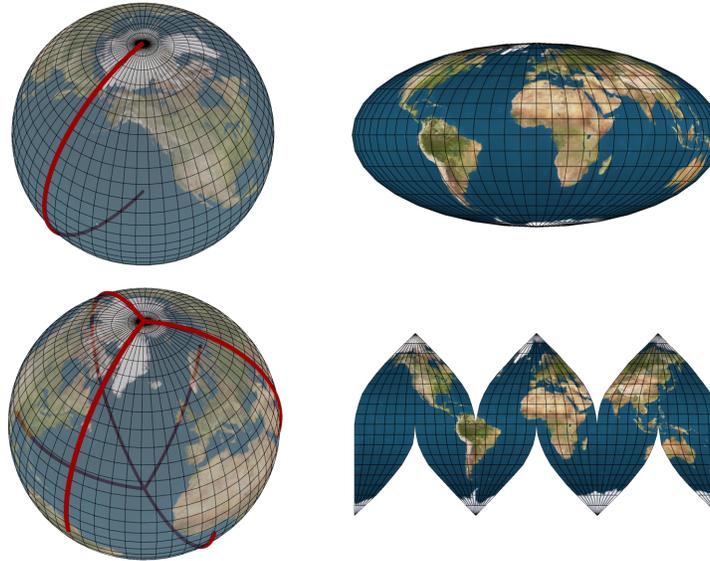


FIGURE 4.4 – La projection de Mollweide et la projection sinusoidale interrompue utilisent deux arbres de coupe différents pour produire une carte planaire.

La figure 4.5 montre un exemple de coupe valide pour un tore qui possède évidemment un seul tunnel.

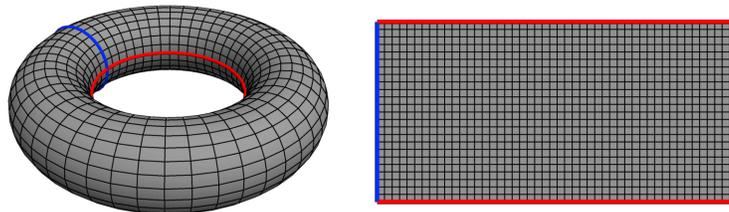


FIGURE 4.5 – Couper le long des deux générateurs d'un tore permet de l'ouvrir en un plan.

L'algorithme utilisé par Gu *et al.* s'inspire des travaux de Dey [Dey94] et effectue l'équivalent d'un remplissage par diffusion (*flood-fill*) à la surface du modèle, où les faces et arêtes sont éliminées une à une. En démarrant d'un triangle initial, tous les triangles et arêtes adjacents à un seul triangle sont éliminés itérativement (figure 4.6). Intuitivement, une surface croît à partir du triangle initial et s'arrête lorsqu'elle entre en collision avec elle-même, permettant ainsi d'encercler les paires de générateurs de tunnels. Cette procédure aura pour effet de produire un réseau de coupe valides, mais

qui pourra être excessivement complexe. La coupe est ensuite nettoyée en enlevant itérativement toutes les arêtes connectées adjacentes à une autre arête, et en cherchant de plus courts chemins entre les noeuds du graphe de coupe. Dans le cas d'un modèle ne possédant aucun trou ni bord, une minuscule coupe de deux arêtes est effectuée.

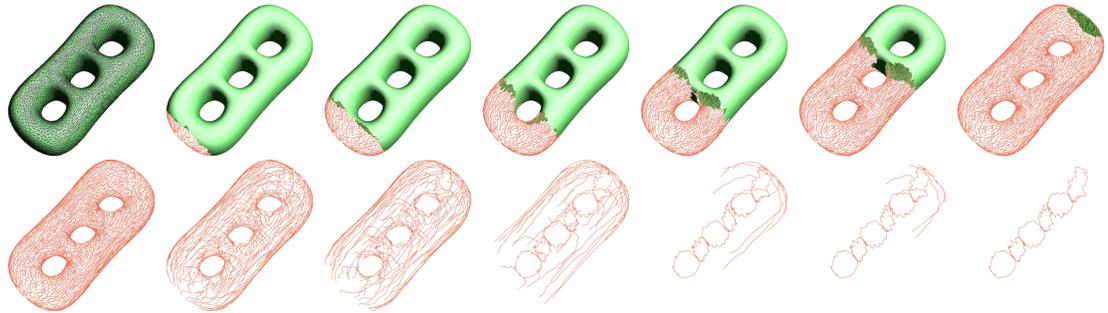


FIGURE 4.6 – En haut : Le graphe de coupe d'un 3-tore peut être trouvé en faisant croître une surface à partir d'un triangle initial. En bas : La coupe est nettoyée et révèle les 6 générateurs. Images tirées de la présentation de [GGH02].

Cette coupe initiale est souvent loin d'être optimale pour la paramétrisation, ce qui risque de créer un sous-échantillonnage de certaines régions. Pour assurer un échantillonnage raisonnable, il est important que la coupe passe par les différents points saillants du modèle. La composante de bord est d'abord distribuée sur le cercle unitaire et paramétrée en utilisant la méthode de Floater [Flo97], qui est à la fois simple et rapide. Le triangle offrant le plus pauvre échantillonnage selon la métrique  $L_2$  présentée au chapitre précédent, est ensuite identifié et le chemin le plus court jusqu'à la bordure est identifié et ajouté à la coupe. Le modèle est ensuite re-paramétré et la procédure continue jusqu'à ce que la qualité de la paramétrisation ne s'améliore plus (figure 4.7).

### 4.2.3 Paramétrisation et échantillonnage

À la fin de l'étape précédente, un réseau de coupe traversant les régions de fortes déformations et permettant de transformer le modèle en une topologie de disque a été identifié. La composante de bord unique qui demeure peut aisément être répartie sur le carré unitaire et l'intérieur peut être paramétré en utilisant la méthode de Sanders *et al.* [SSGH01] décrite au chapitre précédent. Cet algorithme permet de créer une paramétrisation qui préserve les distances, ce qui est particulièrement bien adapté pour

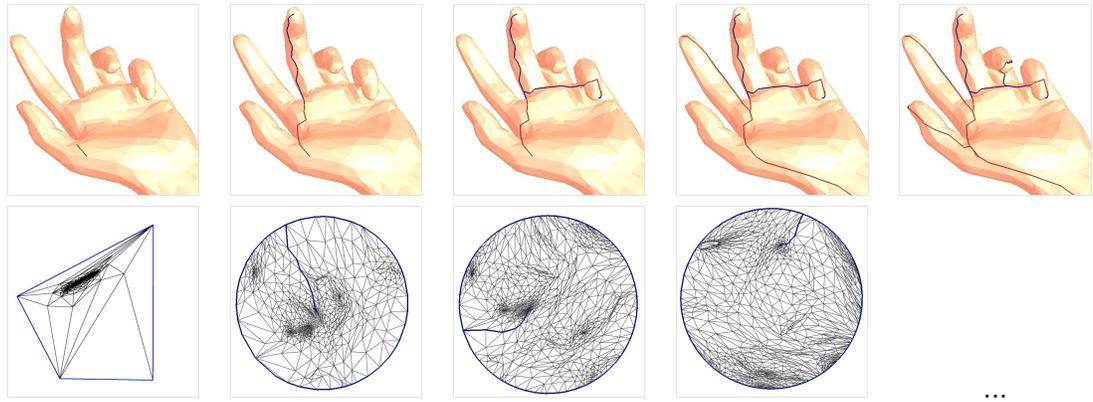


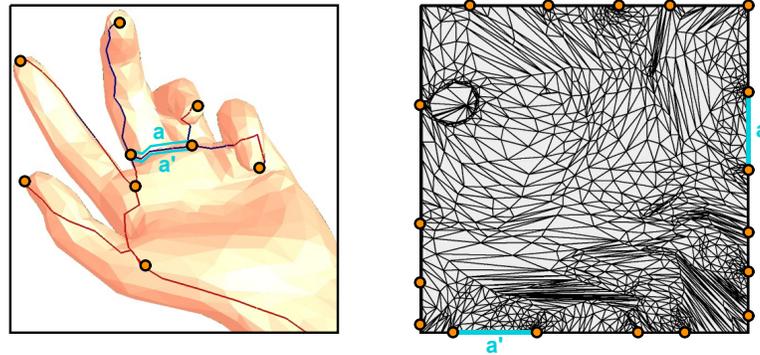
FIGURE 4.7 – La procédure d’optimisation de la coupe tend à aller chercher les extrémités où l’échantillonnage est sous-optimal. Images tirées de la présentation de [GGH02].

le rééchantillonnage. La paramétrisation est ensuite échantillonnée en utilisant une grille régulière et chaque position est stockée dans l’image géométrique. La résolution de la grille d’échantillonnage et de l’image est la même, et aura un impact direct sur la qualité de la reconstruction qu’il sera possible de faire. Bien sûr, d’autres propriétés associées à la surface telles les normales, tangentes et coordonnées  $(u, v)$  peuvent être échantillonnées, à différentes résolutions, par la même procédure.

Ceci étant dit, pour une reconstruction de qualité, la tâche de distribuer la composante de bord du modèle doit être effectuée méticuleusement. On doit faire attention que les deux côtés de tous les segments d’une coupe reçoivent le même nombre d’échantillons et que ces échantillons soient pris au même endroit, sans quoi des fissures apparaîtront à la surface du modèle. Pour imposer ces contraintes, on alloue la même distance pour les deux côtés de chaque segment de coupe et on force les points aux extrémités à coïncider avec des points d’échantillonnage (figure 4.8).

### 4.3 Extensions

Depuis 2002, l’année d’introduction des images géométriques, entre cinq et dix articles par année sont publiés sur le sujet. Les deux années qui ont suivi (2003 et 2004) sont surtout marquées par une série d’extensions à la méthode, tandis que les articles plus récents proposent plutôt des applications. Par extension, nous entendons tout changement à la philosophie de base et toute amélioration de la paramétrisation ou de l’échantillonnage des images. Par application, nous entendons tout ce qu’il est possible de faire à



(a) Modèle géométrique

(b) Paramétrisation

FIGURE 4.8 – Chaque côté du segment de coupe  $a$  ( $a$  et  $a'$ ) reçoit la même longueur sur la bordure de la paramétrisation. De plus, les noeuds de l'arbre de coupe doivent coïncider avec les points de la grille d'échantillonnage. Images tirées de la présentation de [GGH02].

partir d'une telle représentation. La majorité de nos contributions dans ce mémoire se situent dans la première catégorie. Cette section couvrira les principales extensions proposées, tandis que la prochaine dressera une courte liste des applications possibles.

Paramétrer un maillage sur le domaine planaire peut créer de fortes distorsions, surtout lorsque le modèle géométrique présente de longues protubérances. Puisqu'il est difficile de trouver une paramétrisation qui fonctionne bien pour l'ensemble du modèle, Sander *et al.* [SWG<sup>+</sup>03] appliquent le principe des atlas de textures aux images géométriques dans ce qu'ils nomment images géométriques multi-cartes (*multi-chart geometry images*). Faisant croître et optimisant un certain nombre de cartes sur la surface du modèle qu'ils paramétrisent individuellement, ils les assemblent ensuite soigneusement dans une seule texture en tentant de minimiser l'espace gaspillé, avant de les échantillonner en s'assurant de maintenir la connectivité entre les différentes cartes. Leur technique démontre une meilleure distribution des échantillons au prix d'une complexité un peu plus élevée et de quelques pixels gaspillés dans les images. Carr *et al.* [CHCH06b] améliorent cette technique en créant des cartes de forme carrée. Leur technique optimise et réaligne itérativement les cartes à la surface du modèle de façon à obtenir une forme plus ou moins carrée. Leur technique réussit à augmenter l'efficacité d'assemblage dans la texture et à augmenter la qualité de la reconstruction.

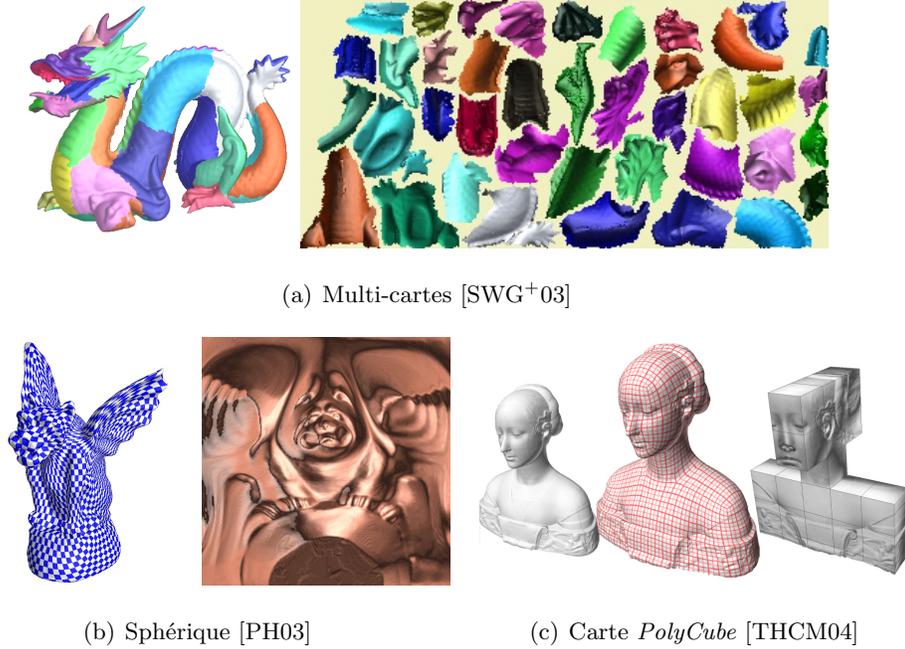


FIGURE 4.9 – Quelques extensions des images géométriques.

Pour les modèles sans trous ni bord, la sphère est un domaine de paramétrisation beaucoup plus naturel que le plan. Praun et Hoppe [PH03] suggèrent donc de paramétrer et d'échantillonner un modèle directement sur la sphère. Ils expérimentent avec plusieurs domaines se développant aisément sur le plan tels le cube, le tétraèdre et l'octaèdre. Pour ce type de modèle, ils augmentent le rapport signal sur bruit des reconstructions par rapport à la méthode originale de Gu *et al.*

Lorsque l'objet possède des trous ou n'a pas une topologie similaire à la sphère, il peut être avantageux d'utiliser plusieurs primitives pour bien couvrir l'ensemble du modèle. Cette idée se reflète dans les cartes *PolyCubes* (*PolyCube maps*) de Tarini *et al.* [THCM04]. S'inspirant de la carte cubique (*cube map*), ils proposent de projeter un modèle non pas sur un cube, mais sur un ensemble de cubes approximant sa forme. Ils expliquent comment projeter le modèle à la surface des cubes pour chacune des 63 configurations différentes qui peuvent survenir entre ceux-ci. Toutes les faces externes des cubes produisent une série d'images géométriques qui peuvent être assemblées dans une texture sans perte d'espace.

Alors que les méthodes énumérées ci-haut améliorent la paramétrisation ou proposent un nouveau domaine, Losasso *et al.* [LHSW03] proposent d'utiliser les images géométriques pour non pas stocker la position des sommets, mais la position de points

de contrôle d'une surface bicubique. Après avoir paramétré et échantillonné un modèle en utilisant la méthode des images géométriques sphériques sur un domaine en forme d'octaèdre, ils ajustent une surface bicubique en s'assurant que celle-ci se refermera sans fissure. Ils utilisent ensuite le matériel graphique pour subdiviser automatiquement la surface et en faire le rendu.

## 4.4 Applications

Cette section propose un rapide tour d'horizon des applications faites avec des images géométriques jusqu'à ce jour. Puisque peu de nos contributions se situent dans la catégorie des applications et que celles-ci sont plutôt nombreuses, la liste des applications sera volontairement brève et peu de détails seront donnés sur les différents algorithmes utilisés. Nous référons le lecteur à l'étude de Sheffer *et al.* [SPR06] pour plus de détails.

Briceño *et al.* [BSM<sup>+</sup>03] étendent la méthode pour encoder des animations. Dans ce qu'ils nomment les vidéos géométriques (*geometry videos*), ils créent une image géométrique à intervalles fixes dans l'animation et appliquent des techniques existantes de compression vidéo au résultat. Puisque le modèle se déforme durant l'animation, ils proposent d'utiliser une seule paramétrisation fixe pour toute la durée de l'animation. Cette paramétrisation étant un compromis pour l'animation entière, des déformations risquent de se produire.

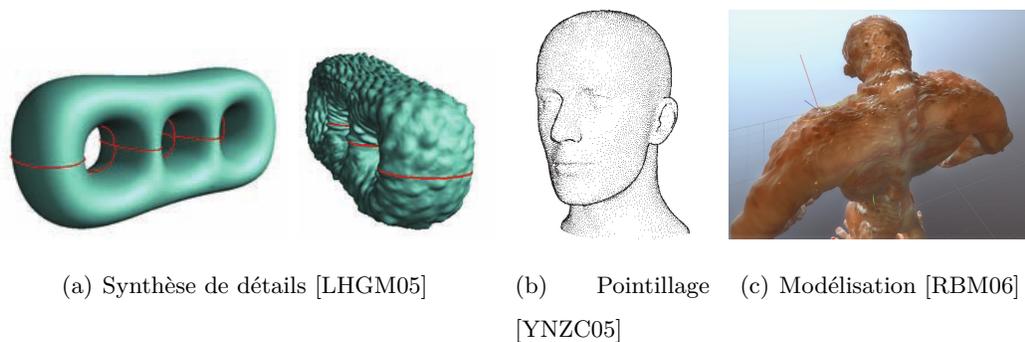


FIGURE 4.10 – Quelques applications des images géométriques.

Dachsbacher et Stamminger [DS04] utilisent une image géométrique en coordonnées écran pour effectuer le rendu de terrains en temps réel. Pour générer une image, une carte d'importance de la vue courante est construite et utilisée pour générer un maillage

stocké dans une image géométrique. Du bruit est ensuite ajouté à l'image pour simuler un fin relief sur le terrain.

Lai *et al.* [LHGM05] effectuent de la synthèse de détails dans l'espace des images géométriques. Un modèle en entrée est converti en image géométrique et un champ vectoriel est calculé à sa surface pour donner une orientation à suivre par la texture. Un échantillon de détail à appliquer est extrait des hautes fréquences d'un modèle géométrique existant et est converti en *texture géométrique*. Les détails sont finalement appliqués sur le modèle en faisant attention d'obtenir une transition douce sur les coupes.

Yuan *et al.* [YNZC05] font du rendu non-photoréaliste en étudiant la géométrie d'un objet dans l'espace des images géométriques. Ils proposent un algorithme de détection de silhouettes et de pointillages (*stippling*) fonctionnant entièrement dans l'espace des images géométriques.

Beneš et Villanueva [BV05] réalisent de la détection de collisions en surimposant une structure hiérarchique de sphères englobantes à l'image géométrique d'un modèle, qui elle aussi est stockée dans une image. Pour détecter les collisions entre deux modèles, les deux hiérarchies sont traversées pour identifier les points de contact précis. Au plus bas niveau, les intersections entre les triangles des deux images géométriques sont testées. Carr *et al.* [CHCH06a] construisent une structure de données similaire, mais l'appliquent au lancer de rayons. Ils démontrent une implémentation rapide d'un lanceur de rayons capable de faire le rendu d'un modèle stocké dans une image géométrique et ce, utilisant exclusivement le matériel graphique. Au lieu de calculer les intersections pour chaque rayon, Li *et al.* [LFY<sup>+</sup>06] adaptent la méthode des champs de lumière (*lightfields*) aux images géométriques qu'ils nomment *champs géométriques (geometry fields)*. Tous les points d'intersection possibles pour un rayon entrant et sortant de la boîte englobante d'un modèle sont calculés. Au lieu de stocker le point d'intersection directement, on conserve plutôt les coordonnées  $(u, v)$  dans l'image géométrique. Ceci permet de trouver non seulement le point d'intersection, mais la normale et d'autres propriétés de la surface du modèle. Ils démontrent l'efficacité de leur méthode avec une implémentation interactive d'un algorithme de réflexion entre objets.

Bien que l'idée de compresser les images géométriques avait déjà été explorée dans la méthode originale de Gu *et al.* [GGH02], Hoppe et Praun [HP05] reprennent le flambeau et proposent de compresser les images géométriques sphériques en utilisant une base

d'ondelettes.

Laga *et al.* [LTN06] utilisent la représentation des images géométriques pour estimer la similarité de modèles. Après avoir converti un modèle en image géométrique sphérique, ils le projettent sur une base d'harmoniques sphériques qu'ils utilisent ensuite pour rechercher dans une base de données de modèles.

Hernandez et Rudomin [HR05] développent une application capable de modéliser des expressions faciales. Ils interpolent entre différentes expressions faciales partageant la même paramétrisation en utilisant des zones d'influences de muscles définis par l'utilisateur. Ritschel *et al.* [RBM06] vont plus loin et développent une application d'édition de maillages similaire à la fonction de sculpture de *Z-Brush* [Pix00] basée entièrement sur les images géométriques. Au lieu d'utiliser l'algorithme complexe de coupe et de paramétrisation de Gu *et al.*, ils effectuent une itération de subdivision Catmull-Clark, ce qui crée un maillage composé uniquement de quadrilatères. Plusieurs autres itérations plus tard, ces mêmes quadrilatères deviennent des grilles de même résolution. Ces grilles créent un atlas d'images géométriques qui sera modifié par l'utilisateur pendant l'édition, l'application s'assurant automatiquement que les cartes voisines soient correctement mises à jour.

Finalement, Hernandez et Rudomin [HR06] présentent un algorithme capable de sélectionner automatiquement le niveau de détail approprié et de faire le rendu d'une image géométrique. En se basant sur la position de la caméra, ils sélectionnent le niveau de *mipmap* de l'image géométrique à utiliser, et font l'élimination des faces arrières et le rendu.

## Chapitre 5

# Paramétrisation sphérique

*Much of the work we do as scientists involves filling in the details about matters that are basically understood already, or applying standard techniques to new specific cases. But occasionally there is a question that offers an opportunity for a really major discovery.*

—Walter Alvarez, *T. Rex and the Crater of Doom*

Alors que les trois chapitres précédents ont passé en revue les trois sujets qui forment les fondations de nos travaux, soient la topologie informatique, la paramétrisation et les images géométriques, ce chapitre et le suivant contiennent une description détaillée de notre méthode pour paramétrer sur la sphère les surfaces de genre supérieur à 0. Nous nous limiterons ici à la description plus théorique des algorithmes utilisés et laisserons les détails d'implémentation au chapitre 7.

### 5.1 Survol de l'algorithme

Avant de passer aux détails de chacun des algorithmes utilisés, il est important d'avoir une vue d'ensemble du système dans sa totalité. Cette section tracera les grandes lignes du processus tandis que les sections à venir détailleront chacune des étapes.

Nous désirons paramétrer un maillage de triangles sur un domaine sphérique, et ce, même si son genre est supérieur à 0. Autrement dit, nous voulons rendre le domaine sphérique accessible aux modèles contenant des trous. Du même jet, nous inclurons aussi les modèles contenant des composantes de bord (CB).

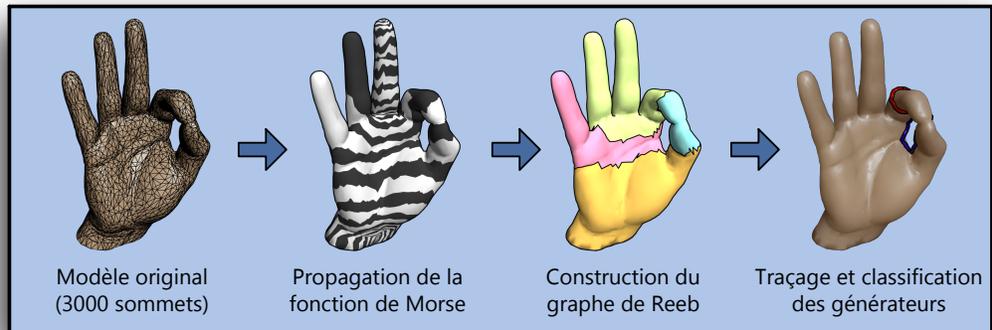
Le modèle en entrée est une 2-variété orientable pouvant contenir un bord, et sa surface est décrite par un maillage de triangles. Le résultat final du processus est une paramétrisation sphérique du modèle. Le processus entier se divise grossièrement en trois étapes et est illustré à la figure 5.1.

En initialisation, le modèle est d'abord chargé en mémoire et analysé. S'il contient des composantes de bord, celles-ci sont bouchées en ajoutant de nouvelles faces. Le genre de la surface peut ensuite être calculé. Si celui-ci est supérieur à zéro, nous débutons à l'étape 1, sans quoi nous passons immédiatement à l'étape 2.

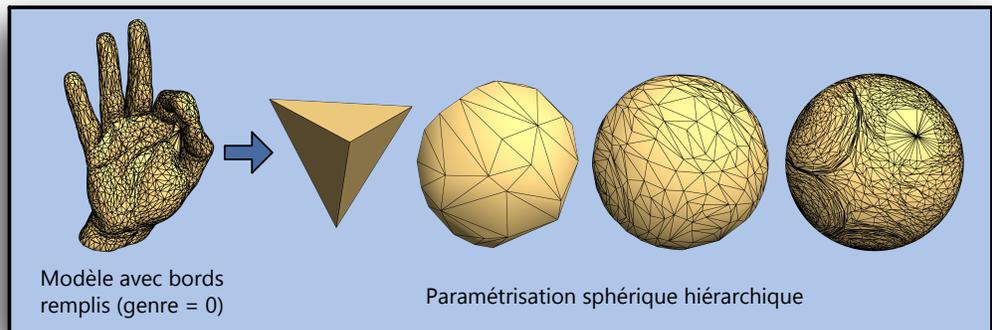
1. **Réduction du genre** : Nous désirons couper le modèle pour lui donner une topologie de sphère perforée. Pour ce faire, nous allons d'abord identifier les deux générateurs de chaque tunnel à la surface du modèle et en sélectionner un pour couper. Ceci correspond à réaliser les trois étapes suivantes.
  - (a) Une fonction de Morse est propagée sur toute la surface du modèle.
  - (b) Le graphe de Reeb est construit à partir de la fonction de Morse.
  - (c) Les générateurs sont extraits et le type de chacun est déterminé. Nous sélectionnons un seul générateur pour chaque paire de générateurs et nous coupons le modèle. Les composantes de bord sont ensuite bouchées pour obtenir une surface fermée de genre 0.
2. **Paramétrisation sphérique** : Nous paramétrons la surface sur le domaine sphérique en utilisant une approche hiérarchique basée sur les maillages progressifs [PH03].
3. **Échantillonnage et reconstruction**
  - (a) Le domaine d'échantillonnage est créé à la résolution voulue.
  - (b) Un ajustement est fait pour les échantillons situés sur le pourtour des composantes de bord et celles qui sont trop petites pour être échantillonnées sont éliminées.
  - (c) La surface est rééchantillonnée et reconstruite si désiré.

Les étapes 1 et 2 seront le sujet de ce chapitre tandis que la troisième sera traitée au prochain chapitre.

### 1) Réduction du genre



### 2) Paramétrisation sphérique



### 3) Échantillonnage et reconstruction

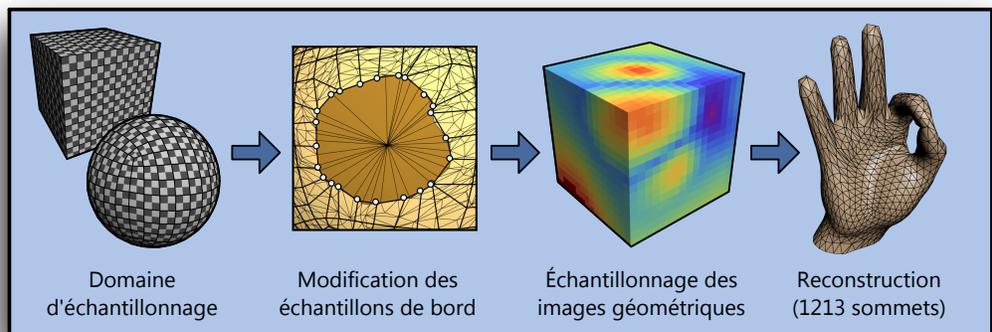


FIGURE 5.1 – Vue d'ensemble du processus.

## 5.2 Initialisation

Pour pouvoir paramétrer une surface sur le domaine sphérique, celle-ci doit être homéomorphe à une sphère et ne contenir aucun trou ni bord. Les composantes de bord existantes peuvent être éliminées aisément à l'ouverture du modèle en parcourant chacune de celles-ci et en ajoutant de nouvelles faces pour les boucher.

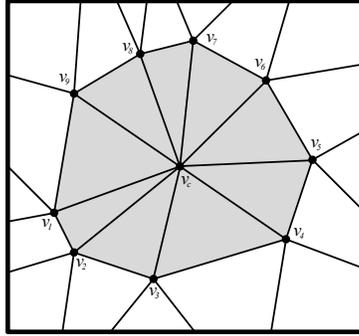


FIGURE 5.2 – Fermeture centrée.

Soit une composante de bord définie par les sommets  $v_1$  à  $v_n$ . Nous effectuons une fermeture centrée en ajoutant un nouveau sommet  $v_c$  situé au barycentre des sommets du bord et en le reliant en une série de  $n$  faces  $(v_{i+1}, v_i, v_c)$ ,  $i = 1..n$ . Pour une composante de bord ne possédant pas une forme convexe, il est possible que le barycentre ou les nouvelles faces créées intersectent la surface modèle ou les faces créées elles-mêmes. Ceci n'est pas un problème puisque ces triangles sont simplement ajoutés de façon temporaire uniquement pour rendre la surface homéomorphe à une sphère et permettre la paramétrisation sphérique. Ces faces seront ignorées lors du rééchantillonnage de la surface.

À la fin de ce traitement, le bord de la surface a été effectivement éliminé et le modèle est prêt à être traité. Nous calculons le genre à partir de la caractéristique d'Euler (équation 8.1) et déterminons si le modèle peut être paramétré immédiatement. Si le genre est égal à zéro, nous passons directement à la paramétrisation (section 5.4) sans quoi nous passons à la réduction du genre (section 5.3).

### 5.3 Réduction du genre

Comme nous avons vu au chapitre 2, topologiquement parlant, toute surface de genre  $n$  peut être vue comme une sphère sur laquelle sont fusionnés  $n$  tores ou anses. Puisqu'un tore est le produit de deux cercles, nous dirons de ces cercles qu'ils sont les *générateurs* du tore (figure 2.6). Individuellement, chaque trou peut être traité comme un tore et être ouvert en un cylindre en coupant l'un ou l'autre de ses générateurs. Ceci est illustré par une séquence de déformations à la figure 5.3. Toute surface de genre  $n$  peut donc être ouverte en une topologie de sphère perforée en appliquant  $n$  coupes, chaque coupe représentant l'un des deux générateurs de chaque tore. Par topologie de sphère perforée, nous entendons une surface de genre 0 avec bord.

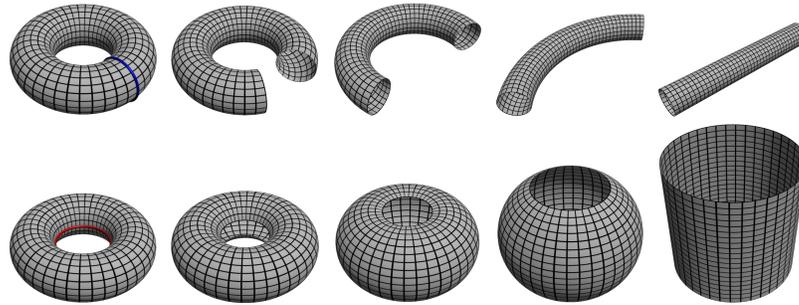
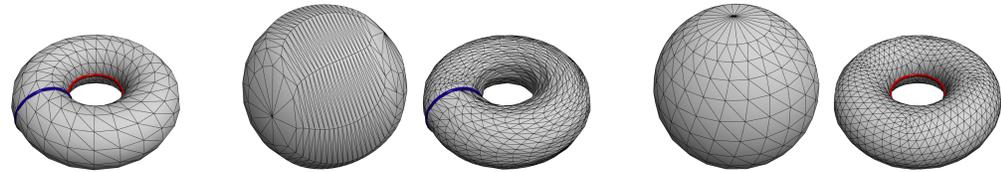


FIGURE 5.3 – Couper un tore le long d'un de ses deux générateurs permet de l'ouvrir en un cylindre.

Les deux générateurs permettent manifestement d'ouvrir le tore en un cylindre et de diminuer de 1 le genre de la surface. Nous pouvons donc choisir arbitrairement l'un des deux générateurs de chaque tore et réduire le genre de la surface à zéro. En revanche, les deux coupes ne permettront généralement pas la même qualité d'échantillonnage et de reconstruction. En effet, puisque la surface est ultimement destinée à être paramétrée sur la sphère, il est important de choisir le générateur qui mènera à la paramétrisation sphérique la plus *naturelle*. La figure 5.4 illustre cette situation. Dans le cas d'un simple tore, il semble plus naturel de couper le cercle rouge que le cercle bleu puisque la paramétrisation résultante est beaucoup moins distorsionnée et permet une reconstruction de plus grande qualité.

Bien que nous ne proposons pas de mécanisme pour sélectionner automatiquement le générateur le plus approprié, nous proposons de déterminer le type de chaque générateur



(a) Modèle original avec générateurs. (b) Coupe du générateur méridien, paramétrisation et reconstruction. (c) Coupe du générateur parallèle, paramétrisation et reconstruction.

FIGURE 5.4 – Le choix du générateur utilisé pour la coupe influence la qualité de la paramétrisation sphérique et donc de la reconstruction.

et de couper tous ceux de même type. La procédure d'identification et de découpage des générateurs nécessite la création du graphe de Reeb suivant une fonction de Morse. Les prochaines sous-sections aborderont ces étapes.

### 5.3.1 Propagation de la fonction de Morse

Comme mentionné au chapitre 2, une bonne fonction de Morse doit permettre de décrire fidèlement la topologie d'une surface sans créer de points critiques artificiels. La fonction *équitable* (*fair*) de Ni *et al.* [NGH04] progresse doucement entre différents points spécifiés comme contraintes par l'utilisateur. Dans le cas qui nous intéresse, nous désirons une fonction possédant un seul minimum et un seul maximum. Cet effet est bien illustré à la figure 5.5. Bien que le point maximum spécifié soit situé sur le majeur de la main, la fonction ne crée aucun point maximum sur les autres doigts, ce qui aurait été le cas pour une fonction comme l'altitude ou la distance géodésique.

Alors que la plupart des algorithmes qui étudient la topologie d'une surface, propagent la fonction de Morse aux sommets de la surface, nous proposons plutôt de la calculer pour chaque face du modèle. Cette modification mineure nous permettra de bâtir le graphe topologique de façon plus cohérente comme nous le verrons à la prochaine sous-section.

Ni *et al.* proposent trois façons d'évaluer cette fonction sur toute la surface du modèle. Bien qu'ils proposent une méthode hiérarchique plus rapide, nous optons pour une méthode plus simple à laquelle nous combinerons un puissant solveur, ce qui nous permettra d'obtenir des performances raisonnables. Nous utilisons donc la formulation

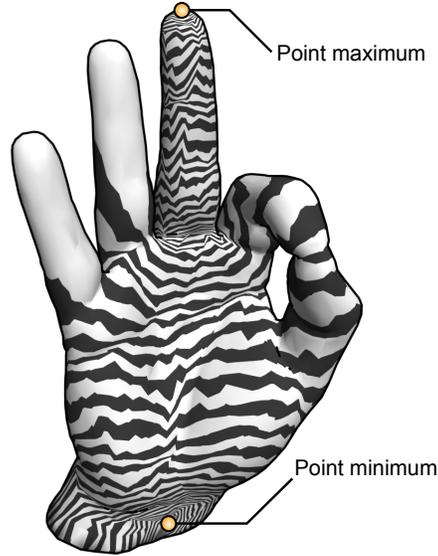


FIGURE 5.5 – La fonction de Morse de Ni *et al.* évolue entre un point minimum et un point maximum ; elle ne crée pas de points critiques artificiels.

linéaire du problème avec les poids combinatoires, ce qui donne un système d'équations linéaires très similaire à ceux que l'on trouve dans plusieurs algorithmes de paramétrisation tels ceux de Tutte [Tut63] et de Floater [Flo97]. Dans l'algorithme original de Ni *et al.*, les poids combinatoires associés à chaque arête sont simplement l'inverse de la valence de chaque sommet  $1/\text{deg}(v_i)$ . Puisque nous calculons maintenant la fonction sur les faces et que celles-ci sont toujours adjacentes par une arête à exactement trois autres faces, un poids uniforme de  $1/3$  est simplement utilisé. Nous assemblons toutes les faces du modèle dans le  $n$ -vecteur  $\mathbf{f}$  et solutionnons l'équation de Laplace sous forme matricielle

$$\mathbf{L}\mathbf{f} = \mathbf{0} \quad (5.1)$$

où

$$L_{ij} = \begin{cases} 1 & i = j \\ -1/3 & f_j \in \mathcal{N}(f_i) \\ 0 & \text{sinon.} \end{cases} \quad (5.2)$$

Nous imposons finalement les contraintes pour la face minimale  $f_{min}$  et maximale  $f_{max}$  en remplaçant les rangées de la matrice  $\mathbf{L}$  par la rangée correspondante de la

matrice identité.

$$\hat{L}_{ij} = \begin{cases} -1 & i = j, f_i = f_{min} \\ 1 & i = j, f_i = f_{max} \\ 0 & i \neq j, f_i = f_{min} \text{ ou } f_i = f_{max} \\ L_{ij} & \text{sinon.} \end{cases} \quad (5.3)$$

Nous solutions finalement le système

$$\hat{\mathbf{L}}\mathbf{f} = \mathbf{b} \quad (5.4)$$

où

$$b_i = \begin{cases} 1 & f_i = f_{max} \\ -1 & f_i = f_{min} \\ 0 & \text{sinon.} \end{cases} \quad (5.5)$$

Après avoir résolu ce système d'équations, nous obtenons une valeur numérique entre -1 et 1 pour chaque face du modèle. À l'exception des deux faces contraintes, la valeur de chaque face est la valeur moyenne des trois faces adjacentes.

### 5.3.2 Construction du graphe de Reeb

Nous construisons le graphe de Reeb de façon similaire à Hilaga *et al.* [HSKK01], en ce sens que la construction se fait en balayant la surface par un ou plusieurs contours, et ce, dans l'ordre défini par la fonction de Morse. Ce processus peut être vu comme une surface qui croît à partir d'un point de départ et qui s'intersecte elle-même pour révéler la structure topologique du modèle géométrique. À chaque itération, nous ajoutons un seul triangle à cette surface et déterminons à quel moment la surface s'intersecte. Ce faisant, nous créons et connectons les noeuds et arcs qui composent le graphe.

Bien qu'un graphe de Reeb n'est pas orienté, le fait qu'il soit créé dans l'ordre de la fonction de Morse et que celle-ci est strictement croissante permet de lui donner une certaine orientation. Nous référons donc aux arcs et noeuds s'éloignant du point minimum en tant qu'*enfants* et aux autres en tant que *parents*. En plus des relations parent/enfant des différents noeuds et arcs qui composent le graphe, nous conservons quelques informations complémentaires qui serviront dans les étapes subséquentes de l'algorithme.

Pour chaque noeud du graphe, nous associons un type qui peut être l'un des quatre suivants : point minimum, point maximum, point de séparation ou point de fusion. Un point minimum est un arc possédant un seul arc enfant tandis qu'un point maximum est un arc possédant un seul arc parent. Lorsque la fonction de Morse décrite à la section

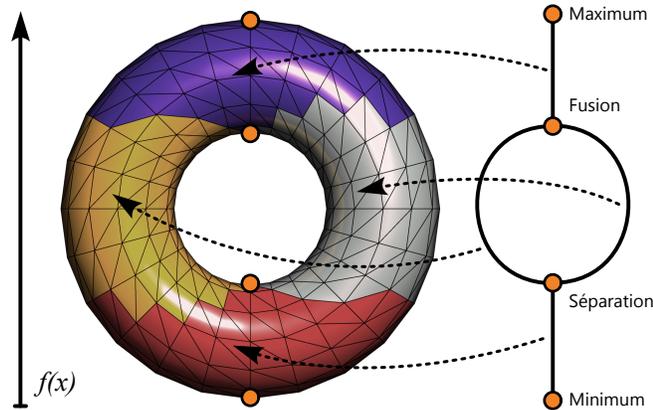


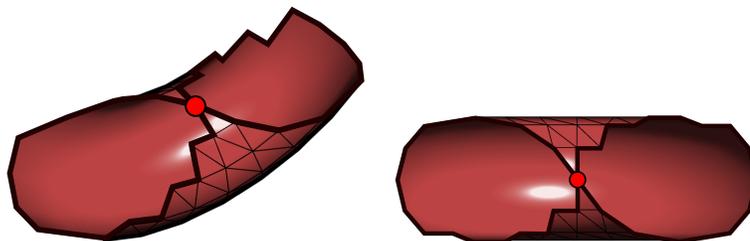
FIGURE 5.6 – Les noeuds du graphe pointent vers l’endroit où l’événement topologique correspondant (face et sommet) s’est produit. Les arcs contiennent la liste des faces balayées entre les noeuds (de même couleur) ainsi que le contour.

précédente est utilisée, il ne devrait y avoir qu’un seul noeud minimum et un seul noeud maximum. Un point de séparation est un noeud possédant un arc parent et deux arcs enfants. À l’inverse, un point de fusion est un noeud possédant deux arcs parents et un seul arc enfant. Le graphe devrait compter autant de noeuds de fusion que de noeuds de séparation, et ce nombre devrait être égal au genre du modèle géométrique. Finalement, nous conservons aussi l’index de la face et du sommet, sur le modèle géométrique, qui a mené à la création de ce noeud.

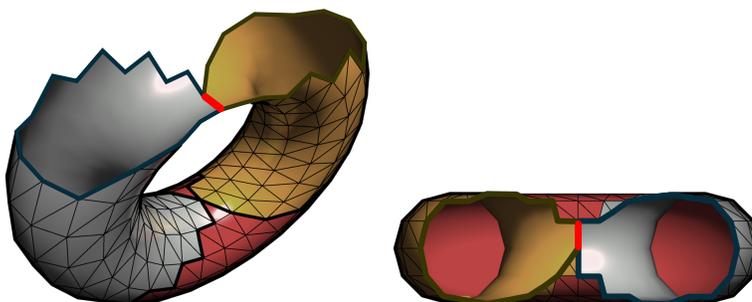
De façon similaire, pour chaque arc du graphe, nous conservons la liste des faces du modèle géométrique qui sont associées à cet arc ainsi que le contour ayant servi à balayer ces faces. Un contour est simplement une boucle connectée d’arêtes ne s’intersectant pas. Un arc ou contour sera dit *actif* s’il ne possède toujours pas de noeuds enfants, c’est-à-dire qu’il peut toujours avancer sur la surface. La figure 5.6 montre un exemple 2D des liens entre le graphe de Reeb et le modèle géométrique.

L’algorithme démarre en localisant le triangle possédant la plus petite valeur de fonction de Morse et crée un noeud de type *minimum* et un arc sortant de ce noeud. Puisque nous démarrons avec un seul triangle, il n’y a initialement qu’un seul contour actif et il est composé de trois arêtes.

À chaque itération, les contours actifs sont parcourus et la face adjacente possédant la plus petite valeur de Morse et n’ayant pas été visitée est sélectionnée. Les arêtes



(a) Séparation : un contour s'auto-intersecte puisqu'il traverse deux fois le même sommet.



(b) Fusion : deux contours fusionnent car ils partagent une arête.

FIGURE 5.7 – Séparation et fusion de contours lors du balayage de la surface.

de cette face sont fusionnées au contour et nous vérifions si le contour s'intersecte ou fusionne avec un autre contour.

Un contour est considéré comme s'auto-intersectant s'il passe deux fois par le même point. Dans cette situation, un nouveau noeud de type *séparation* est créé ainsi que deux nouveaux arcs enfants. Le contour est divisé en deux et chacun des contours est associé à un des deux nouveaux arcs. Les deux arcs enfants sont actifs alors que l'arc parent est désactivé.

Deux contours sont considérés comme fusionnés s'ils partagent une arête, de sorte que si cette arête est retirée, nous obtenons un seul contour ne s'intersectant pas. Lorsque ceci se produit, nous créons un noeud de *fusion* et un nouvel arc auquel nous associons le contour fusionné. Le nouvel arc est actif et prend la relève des deux arcs qui ont été fusionnés. Ces derniers sont désactivés. Ces deux situations sont illustrées à la figure 5.7. Il est intéressant de noter que ces deux configurations sont bel et bien des points de selle tels que décrits par la théorie de Morse. En effet, autour des points de fusion et de séparation, la fonction augmente dans une direction et diminue dans une autre.

Un contour disparaît lorsqu'un point critique maximum est atteint. Cette situation est triviale puisqu'elle implique simplement la création d'un noeud *maximum* et la désactivation de l'arc. Par construction, nous n'avons qu'un seul noeud maximum dans le graphe.

Le résultat de toute cette opération est un graphe topologique complet décrivant fidèlement la topologie du modèle géométrique. Les noeuds du graphe indiquent les endroits où les différents événements topologiques se sont produits et les arcs contiennent les différentes régions de la surface qui ont été balayées entre les différents noeuds. On dira de ce graphe qu'il est *semi-incorporé* sur la surface puisque nous connaissons la position des noeuds, mais n'avons pas tracé les arcs entre les noeuds. Si nous le désirons, nous pouvons utiliser un algorithme de plus court chemin pour tracer ceux-ci à travers les arêtes de la surface, mais ceci n'est pas nécessaire pour nos besoins.

### 5.3.3 Traçage des générateurs

Bien que le graphe de Reeb décrive grossièrement où se situent les trous à la surface du modèle géométrique, il ne spécifie pas la position ni la forme des générateurs de chaque trou. Conséquemment, avant de pouvoir déterminer le type de chacun de ceux-ci, ils doivent être tracés à la surface du modèle. Nous chercherons donc d'abord à déterminer la forme de chaque paire de générateurs. Un générateur sera simplement un chemin défini par une liste d'arêtes sur la surface, qui ne s'intersecte pas. Chaque paire de générateurs se croise à exactement un seul sommet.

Pour effectuer cette tâche, nous utilisons une procédure similaire à Steiner et Fischer [SF03]. Nous apportons ensuite bon nombre de modifications pour rendre la méthode plus robuste. Une fois les générateurs bien localisés sur la surface, nous tenterons d'identifier le type de chacun.

Soient  $G_1$  et  $G_2$ , les deux générateurs d'une paire décrits par une liste d'arêtes et  $v_f$ , le sommet où ceux-ci se croisent. Nous dirons de  $G_1$  qu'il est le *frère* de  $G_2$  et vice versa.

La procédure de Steiner et Fischer parcourt un à un tous les noeuds de fusion du graphe topologique dans l'ordre qu'ils ont été créés et trace chaque paire de générateurs. Le sommet où la fusion a eu lieu est choisi comme étant  $v_f$ , soit le point commun aux deux générateurs. Un noeud de fusion indique la présence d'un trou puisque deux

contours sont entrés en collision pour n'en former qu'un seul. Le plus court des deux contours est choisi comme étant la forme de  $G_1$ . Le second générateur,  $G_2$ , est ensuite trouvé en effectuant une recherche de plus court chemin de  $v_f$  vers lui-même à travers les faces qui ont été visitées avant le point de fusion tout en évitant de croiser un générateur précédemment identifié. Cette simple procédure est appliquée à tous les points de fusion, ce qui permet d'identifier toutes les paires de générateurs. La figure 5.8 illustre ce procédé.

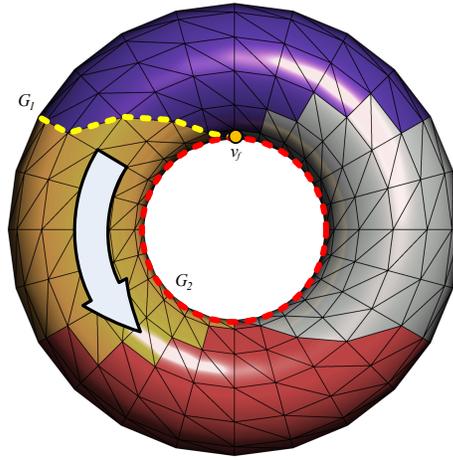


FIGURE 5.8 – La procédure de Steiner et Fischer [SF03] crée  $G_1$  en sélectionnant le plus court des contours parents, et  $G_2$  en cherchant un chemin de  $v_f$  vers lui-même à travers la région visitée n'intersectant pas un autre générateur précédemment identifié.

Néanmoins, lorsqu'appliqué à des surfaces plus complexes, cet algorithme présente quelques problèmes. Premièrement, en sélectionnant l'un des deux contours comme  $G_1$ , aucune vérification n'est faite à savoir si celui-ci intersecte un générateur existant. Deuxièmement, dans certaines régions pauvrement tessellées et présentant beaucoup de trous, il est possible que la recherche de plus court chemin échoue. Ceci se produit lorsque trop peu de chemins sont disponibles et que ceux-ci sont bloqués par d'autres générateurs précédemment identifiés.

Pour remédier à ces problèmes, nous proposons une autre séquence d'opérations plus robuste qui, entre autres, évite que les générateurs ne s'intersectent entre eux et permet l'ajout de nouveaux sommets lorsqu'aucun chemin n'est disponible.

Nous parcourons les noeuds de fusion de façon similaire à Steiner et Fischer. Nous

sélectionnons temporairement le plus court des contours parents pour  $G_1$ . Nous tentons ensuite de tracer  $G_2$  en recherchant un plus court chemin qui démarre et se termine à  $v_f$  à travers les faces visitées. Nous utilisons l'algorithme A\* [HNR68] et la distance euclidienne entre deux points comme heuristique. S'il est impossible de trouver un chemin dans les faces visitées, nous relaxons cette condition et essayons à nouveau en cherchant simplement un chemin de  $v_f$  vers  $v_f$  qui démarre d'un côté  $G_1$  et revient de l'autre.

Si, pendant la recherche du plus court chemin, le passage est bloqué par un ou plusieurs autres générateurs précédemment tracés, nous gardons la liste des collisions, dans l'ordre qu'elles sont survenues. Dans l'éventualité où la recherche de chemin échoue, nous effectuons une série de subdivisions de la surface pour permettre la création de nouveaux chemins entre les générateurs ayant généré des collisions.

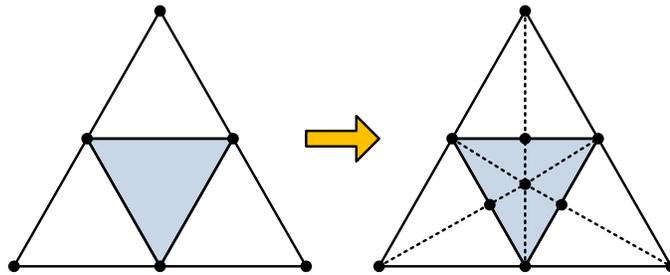


FIGURE 5.9 – Subdivision barycentrique.

La subdivision barycentrique (figure 5.9) fractionne chaque triangle en six en ajoutant un nouveau sommet au barycentre du triangle ainsi qu'aux points milieu des arêtes. Elle peut être vue comme la combinaison d'une subdivision *1-pour-3* à partir du centre du triangle et d'une subdivision *2-pour-4* au milieu de chaque arête. Ce type de subdivision possède l'avantage de créer une série de nouveaux chemins alternatifs dans les régions pauvrement tessellées en permettant à l'algorithme de passer par le nouveau sommet inséré au centre du triangle. La figure 5.10 montre un exemple 2D de ceci.

Lorsque la recherche du plus court chemin échoue, nous débutons par subdiviser toutes les faces adjacentes à  $G_1$  et essayons à nouveau de trouver un chemin qui démarre d'un côté  $G_1$  et revient de l'autre. Si cette tentative échoue toujours, nous tentons de subdiviser les faces adjacentes aux générateurs qui ont généré des collisions. Nous traitons ceux-ci un à un et reessayons de tracer un chemin autour du générateur frère entre chaque subdivision.

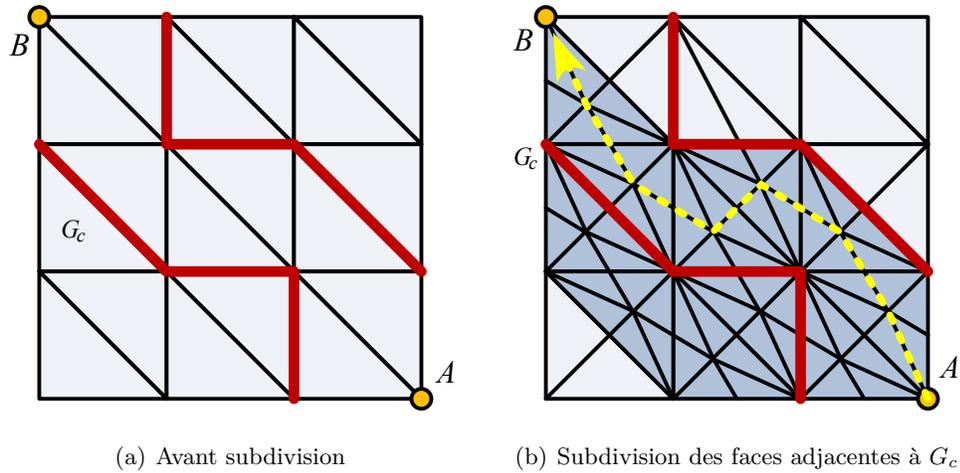


FIGURE 5.10 – La subdivision barycentrique permet de créer de nouveaux chemins. Les lignes rouges représentent des obstacles et la ligne pointillée le nouveau chemin tracé.

$G_2$  étant tracé, nous réoptimisons  $G_1$  en cherchant un chemin autour de  $G_2$  et en appliquant, si nécessaire, la même procédure de subdivision. Lorsque les deux générateurs sont tracés, nous marquons les arêtes de  $G_1$  et  $G_2$  comme étant *bloquées* pour éviter qu'elles soient traversées plus tard par un autre générateur.

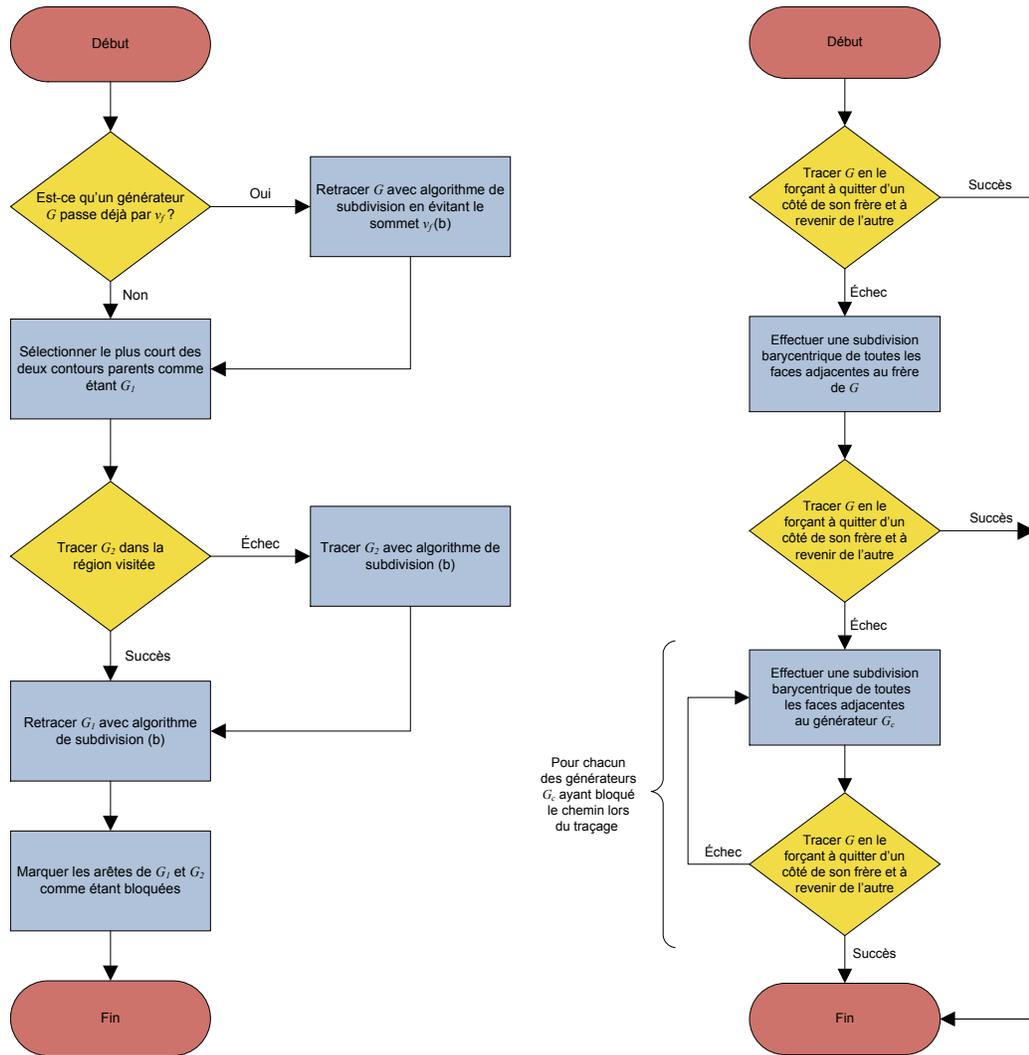
Finalement, dans l'éventualité où un générateur passe déjà par un point de fusion, nous retraçons d'abord celui-ci en le forçant à éviter ce sommet et continuons le traitement de façon régulière. Les organigrammes de la figure 5.11 résument la procédure.

### 5.3.4 Classification des générateurs

Comme nous l'avons vu en début de chapitre, il est possible d'ouvrir un tore en un cylindre en le coupant le long d'un des deux générateurs. Chaque paire de générateurs contient un générateur *méridien* et un générateur *parallèle*. À l'aide du graphe topologique, nous sommes en mesure de tracer chaque paire de de générateurs à la surface du modèle géométrique, mais n'avons pas encore déterminé le type de chacun.

Topologiquement parlant, il n'y a aucune différence entre un et l'autre. En effet, il a été démontré que le groupe fondamental d'un tore est commutatif (abélien) et qu'il existe une déformation continue permettant d'inverser l'un et l'autre. Si nous nous éloignons des notions purement topologiques et incorporons l'information géométrique que nous possédons sur la surface, il devient possible d'associer un type à chacun.

Steiner et Fischer [SF03] proposent une méthode inspirée de la théorie des noeuds



(a) Algorithme pour tracer tous les générateurs (b) Traçage du générateur  $G$  avec subdivision

FIGURE 5.11 – L'algorithme complet et robuste de traçage des générateurs.

capable d'effectuer cette tâche. Pour des fins de complétude, nous résumons les grandes lignes de leur méthode et pour ce faire, nous devons d'abord établir quelques concepts.

L'*enlacement* (*linking number*) est un nombre entier invariant décrivant comment deux courbes sont enlacées, l'une par rapport à l'autre. L'enlacement de deux courbes peut être calculé en projetant d'abord celles-ci sur un plan et en étudiant chacun des endroits où elles se croisent. En donnant une orientation arbitraire à chacune des courbes, on peut affecter à chaque croisement une valeur numérique de  $-1$  ou  $+1$ . La demi-somme de tous les croisements est la valeur d'enlacement.

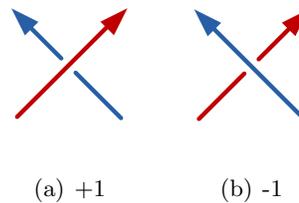


FIGURE 5.12 – Valeur associée à chaque croisement.

L'idée est donc de traiter chaque paire de générateurs comme une boucle et d'analyser comment ils sont interreliés lorsqu'un des deux est déplacé dans la direction inverse de la normale. Les deux boucles peuvent être soit séparées ou bien liées de façon à ce qu'il soit impossible de les séparer. Une valeur d'enlacement de  $+1$  ou  $-1$  signifie un entrelac de Hopf (*Hopf link*) tandis qu'une valeur de  $0$  signifie un entrelac trivial (*unlink*). Si le générateur déplacé dans la direction inverse de la normale au point de croisement génère un entrelac de Hopf, il s'agit d'un parallèle, sinon il s'agit d'un méridien (figures 5.13 et 5.14).

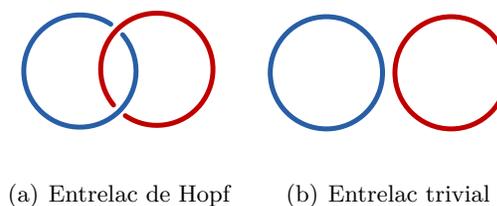


FIGURE 5.13 – Deux types de noeuds possibles entre deux générateurs.

Le calcul de l'enlacement requiert la projection des segments sur un plan. Cette projection doit éviter de créer des situations dégénérées qui prohiberaient le calcul de l'enlacement, tel un segment projeté en un point ou deux segments qui se superposent.

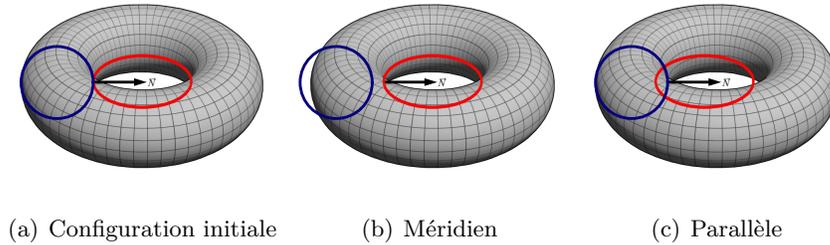


FIGURE 5.14 – Nous déplaçons un générateur dans la direction inverse de la normale. Si les deux générateurs sont enlacés, il s’agit d’un parallèle, sinon il s’agit d’un méridien.

Dans leur méthode originale, Steiner et Fischer [SF03] proposent un algorithme sophistiqué qui accomplit cette tâche. Nous utilisons une recherche simple par force brute en essayant différents plans de projection aléatoires jusqu’à ce nous trouvions une projection valide.

L’une des hypothèses faites dans cet algorithme est que la normale au plan donne une direction valide pour déplacer l’un des générateurs. Pour déterminer si la normale est une direction valide, nous projetons les segments des deux générateurs qui contiennent le sommet commun sur un plan perpendiculaire à la normale. Nous parcourons ensuite les segments en direction horaire (ou anti-horaire) et vérifions que deux segments provenant du même générateur ne sont pas consécutifs (figure 5.15). Lorsque la normale n’est pas une direction de déplacement valide, nous tentons de la perturber légèrement dans plusieurs directions aléatoires pour tenter de la corriger.

Dans des régions pauvrement tessellées, il est possible qu’aucune normale valide ne puisse être trouvée. Bien que cela soit plutôt rare, si cela s’avère être le cas, nous abandonnons la méthode de Steiner et Fischer [SF03] et nous nous tournons vers une méthode beaucoup moins certaine. Si on regarde les deux générateurs d’un tore, il apparaît évident que le parallèle encercle le *tunnel* du tore et son centre devrait être vide tandis que le méridien entoure l’anse du tore et son centre devrait être solide. Sachant cela, nous pouvons vérifier que le barycentre d’un des contours est situé à l’intérieur du modèle géométrique tandis que l’autre est à l’extérieur. Par contre, cette stratégie ne fonctionne que pour des générateurs de forme convexe et ne garantit donc pas de résultat valide.

Si cette seconde méthode échoue, nous assignons un type arbitraire aux générateurs. Certaines configurations sont simplement trop dégénérées pour nous permettre de fidè-

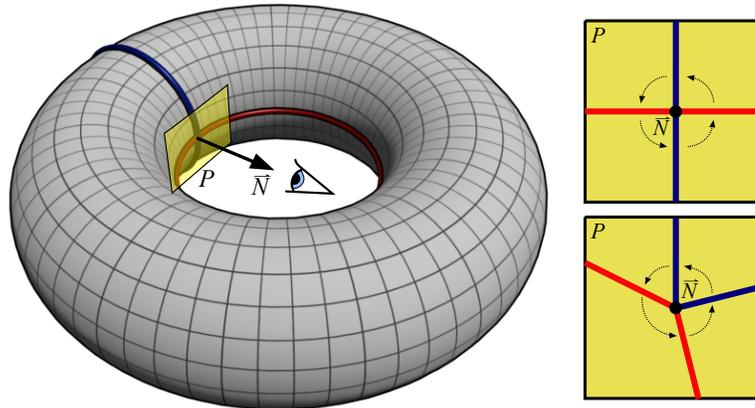


FIGURE 5.15 – À gauche, nous projetons les deux générateurs sur un plan  $P$  perpendiculaire à la normale  $\vec{N}$ . En haut à droite, les deux générateurs projetés apparaissent dans l'ordre rouge-bleu-rouge-bleu, ce qui est valide. En bas à droite, exemple de configuration invalide.

lement assigner un type. Par exemple, un tore qu'on écraserait de façon à ce que tous les sommets soient situés sur le même plan ne pourra être traité avec aucune de ces méthodes.

### 5.3.5 Optimisation des générateurs

Tracer les générateurs avec cet algorithme peut parfois produire des générateurs beaucoup trop longs. Puisque nous allons couper uniquement les générateurs d'un seul type et ignorer les autres, nous pouvons optimiser les générateurs du type qui nous intéresse. Notre stratégie est basée sur l'observation que lorsque tous les générateurs sont tracés, nous pouvons temporairement en enlever quelques-uns et les retracer dans un ordre différent pour obtenir un autre ensemble valide de générateurs.

Nous traitons tous les générateurs du type choisi dans l'ordre décroissant de leur longueur. Nous retraçons le générateur en utilisant un algorithme de plus court chemin et conservons la liste des générateurs du type inverse qui ont bloqué le chemin pendant la recherche. Nous éliminons temporairement ceux-ci un à la fois, et réessayons de trouver un plus court chemin ; nous arrêtons dès qu'un meilleur chemin a été trouvé. Nous retraçons ensuite tous les générateurs qui avaient été éliminés. Après quelques itérations de cette procédure, nous aurons réduit les générateurs du type choisi.

Comme dernière étape, pour chaque générateur du type choisi  $G_1$  et son générateur frère  $G_2$ , nous recherchons un plus court chemin à partir de tous les sommets de  $G_2$ . Nous gardons le plus court chemin trouvé et l'assignons comme étant la nouvelle forme de  $G_1$ .

## 5.4 Paramétrisation sphérique

À cette étape-ci, le modèle géométrique a un genre 0 et toutes les composantes de bord ont été colmatées. La surface a donc une topologie de sphère et est prête à la paramétrisation. Comme mentionné au chapitre 4, nous utilisons une version hiérarchique de la méthode de Friedel *et al.* [FSD05]. Dans cette méthode, le problème de paramétrisation sphérique est présenté comme un problème d'optimisation non linéaire sans contraintes. Autrement dit, nous cherchons à trouver les positions  $(\theta, \phi)$  de chaque sommet qui minimisent une certaine fonction d'énergie.

Nous utilisons la fonction d'énergie combinée qu'ils proposent. Il s'agit tout simplement d'une combinaison linéaire de l'énergie de Dirichlet, qui préserve les angles et la différence d'aire. Pour chaque triangle  $(A, B, C)$  d'angles  $(\alpha, \beta, \gamma)$ , elle est donnée par

$$E_{combinée} = a \cdot E_{aire} + b \cdot E_{dirichlet} \quad (5.6)$$

où

$$E_{aire} = d_{min}^{-2} \cdot \frac{Aire(A, B, C)^2}{AireSurface} \quad (5.7)$$

et

$$E_{dirichlet} = d_{min}^{-2} \cdot (\cot \alpha \cdot (B - C)^2 + \cot \beta \cdot (A - C)^2 + \cot \gamma \cdot (A - B)^2). \quad (5.8)$$

Le terme  $d_{min}^{-2}$  représente l'inverse de la distance minimale entre le triangle planaire et le centre de la sphère dans l'espace de la paramétrisation tandis que le terme  $AireSurface$  représente l'aire du triangle dans l'espace 3D. Pour un triangle aigu, le point le plus près du centre est au milieu de l'arête la plus longue et pour un triangle obtus, il est situé au centre du cercle circonscrit au triangle (figure 5.16). Ce terme est une correction appliquée aux énergies planaires. Elle stabilise la procédure d'optimisation en gardant sous contrôle les gros triangles qui essaieraient de traverser l'équateur, créant ainsi une paramétrisation dégénérée.

Les termes  $a$  et  $b$  de l'équation 5.6 servent à balancer la préservation d'aire et d'angle désirée pendant la paramétrisation. Il est important de noter que ces termes peuvent prendre une valeur positive quelconque et ne sont pas restreints entre 0 et 1 puisque les énergies de préservation d'aire et de Dirichlet ne partagent pas les mêmes unités de mesure. Ils servent à créer un équilibre entre deux valeurs d'énergies arbitraires.

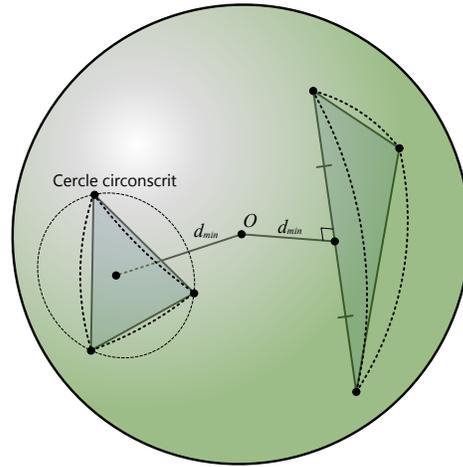


FIGURE 5.16 – Le point du triangle le plus près du centre de la sphère est au centre du cercle circonscrit pour un triangle obtus (gauche) et au milieu de l'arête la plus longue pour un triangle aigu (droite).

Cette énergie peut être minimisée en utilisant un solveur standard tel la méthode de région de confiance ou la recherche linéaire de Newton (*Newton trust region* et *Newton line search*). Puisqu'il s'agit essentiellement d'une descente de gradient, il est important de choisir une solution de départ suffisamment proche du minimum global pour éviter que le solveur reste coincé dans un minimum local qui mènerait à une paramétrisation de mauvaise qualité. Dans leur article original, Friedel *et al.* [FSD05] proposent de projeter simplement le modèle sur la sphère unitaire à partir du centre de masse du modèle géométrique. Ils montrent que même si la paramétrisation de départ possède des faces inversées, le solveur réussit quand même à créer une paramétrisation bijective. Bien que cette méthode fonctionne assez bien pour des modèles ayant globalement l'allure d'une sphère, elle amène le solveur à diverger pour un grand nombre de modèles. De plus, même si le solveur réussit à converger vers une solution valide, une paramétrisation initiale de mauvaise qualité risque d'avoir un impact

négalif sur le temps de calcul.

Pour pallier à ces problèmes, nous adoptons une stratégie hiérarchique similaire à Praun et Hoppe [PH03]. Nous débutons par décimer le modèle en effectuant une série de fusions de demi-arêtes (*half-edge collapse*) en utilisant une mesure d'erreur quadrique. Le résultat de cette procédure est une séquence de maillages progressifs. Pour une surface de genre 0, le niveau le plus brut possède toujours quatre sommets et peut aisément être paramétré sur une sphère en lui donnant la forme d'un tétraèdre. Cette configuration initiale nous permet de démarrer la procédure d'optimisation.

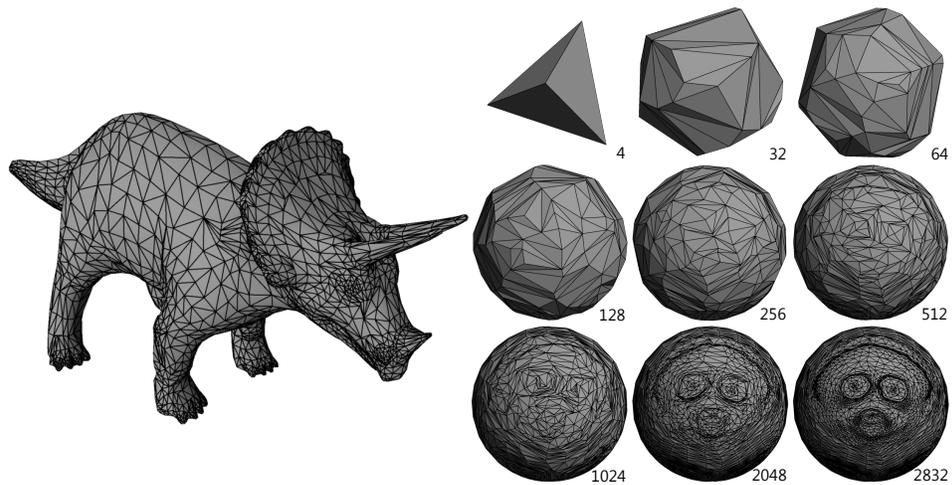


FIGURE 5.17 – Séquence de paramétrisation hiérarchique pour un modèle de tricératops.

À chaque itération, nous effectuons une série de divisions de demi-arêtes (*half-edge split*) jusqu'à ce que le nombre de sommets augmente d'un certain facteur (*e.g.* 2.0). À chaque division, nous positionnons le nouveau sommet au barycentre sphérique du noyau de ses voisins. Nous optimisons ensuite la paramétrisation en lançant le solveur sur la nouvelle configuration.

Cette stratégie possède l'avantage de toujours fournir une bonne solution initiale au solveur. En effet, contrairement à la méthode de projection de Friedel *et al.* [FSD05], la paramétrisation ne contient aucune inversion de face, et ce, pendant toute la durée du processus. De plus, le fait que les nouveaux sommets soient insérés dans le noyau de leurs voisins, qui eux, sont globalement bien positionnés, diminue la charge de travail du solveur dans la mesure où cela évite une réorganisation massive de la paramétrisation (figure 5.17).

Les composantes de bord qui ont été colmatées au chargement du modèle ou lors de la réduction du genre sont traitées de façon identique aux autres triangles. En revanche, nous ajoutons un facteur de correction d'aire  $\lambda$  pour celles-ci. Ce facteur d'échelle nous permet de réduire ou d'augmenter l'aire qu'elles occupent dans la paramétrisation en modifiant la valeur d'aire *AireSurface* (équation 5.7) fournie au solutionneur.

## Chapitre 6

# Échantillonnage

Alors que le chapitre précédent nous a décrit comment paramétrer une surface de genre supérieur à 0 sur le domaine sphérique, ce chapitre discute des spécificités liées à l'échantillonnage d'une telle paramétrisation. Nous présenterons également un algorithme capable d'effectuer à la fois l'échantillonnage de cette dernière et la reconstruction de la surface originale.

### 6.1 Création du domaine d'échantillonnage

Tout comme la méthode des images géométriques, nous cherchons à rééchantillonner une surface en passant par sa paramétrisation. Dans le cas d'une paramétrisation plane 2D de forme rectangulaire ou carrée, cette tâche peut être effectuée en utilisant une simple grille régulière. Lorsque la paramétrisation repose sur un domaine sphérique, la tâche est moins triviale puisque les échantillons doivent reposer sur la surface de la sphère et pouvoir mener à une reconstruction de qualité.

Nous nommons *domaine d'échantillonnage* tout ensemble de points pouvant servir à l'échantillonnage d'une paramétrisation et possiblement à la reconstruction de la surface. Plus spécifiquement, nous nommons *domaine d'échantillonnage sphérique*, lorsque ces points sont disposés à la surface de la sphère. Puisque nous nous attardons uniquement au domaine sphérique dans ce travail, nous omettrons le mot sphérique, sauf aux endroits où il pourrait y avoir ambiguïté.

Plus spécifiquement, les points d'échantillonnage doivent être organisés de façon à permettre la reconstruction de la surface. Ceux-ci doivent donc contenir une notion de

connectivité entre eux pour permettre la création de triangles ou de quadrilatères. Qui plus est, si les échantillons doivent être stockés dans des images géométriques, ceux-ci doivent être disposés comme dans une image. Chaque échantillon doit toujours avoir quatre voisins (sauf aux bords de l'image où ils en ont deux ou trois) et permettre la création de quadrilatères.

Un bon exemple de domaine d'échantillonnage répondant à ces critères est la carte cubique (*cubemap*). Celle-ci peut être aisément projetée sur la sphère unitaire et possède six faces carrées pouvant être traitées comme des images. D'autres solides de Platon comme le tétraèdre ou l'octaèdre peuvent être également projetés sur la sphère, mais ne semblent pas avoir la configuration requise pour pouvoir stocker les échantillons dans une image. En revanche, ceux-ci peuvent être dépliés pour obtenir un carré, ce qui les rend aussi admissibles comme domaine d'échantillonnage (figure 7.1).

Le domaine d'échantillonnage est créé explicitement sous forme d'un maillage de quadrilatères. Les sommets sont désignés comme étant les points d'échantillonnage du domaine. Si l'on désire stocker le rééchantillonnage dans des images géométriques, on associe à chaque échantillon (sommets) une ou plusieurs positions d'image (figure 6.1). Par exemple, la majorité des échantillons pris sur une carte cubique seront conservés dans une seule des six images. Les échantillons pris à la jonction de deux faces du cube seront stockés dans deux images tandis que les échantillons situés aux coins du cube seront stockés dans trois images.

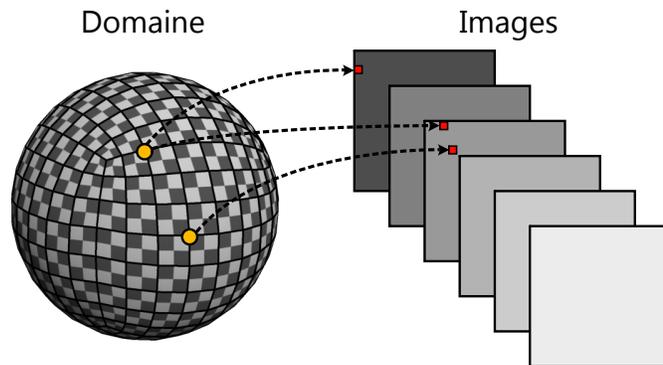


FIGURE 6.1 – Chaque point d'échantillonnage pointe vers 1.. $n$  pixels d'image(s).

Bien que stocker le même échantillon dans plusieurs images est redondant, ceci rend chaque image indépendante et complète en ce sens où chaque image contient l'information pour créer tous les quadrilatères composant le modèle ou une partie de celui-ci.

Sans cette redondance, l'algorithme de remaillage devrait parfois lire les positions dans les images adjacentes.

Lorsqu'un domaine n'est pas directement défini sur la sphère, celui-ci est projeté sur celle-ci. Cette projection crée une distribution inégale des échantillons. Pour optimiser cette distribution, nous appliquons l'algorithme de paramétrisation présenté au chapitre précédent en donnant un fort poids à la préservation d'aire et un plus faible à la préservation d'angle. Cette procédure, similaire à celle de Praun et Hoppe [PH03], améliore de beaucoup la distribution des échantillons comme on peut le voir à la figure 6.2.

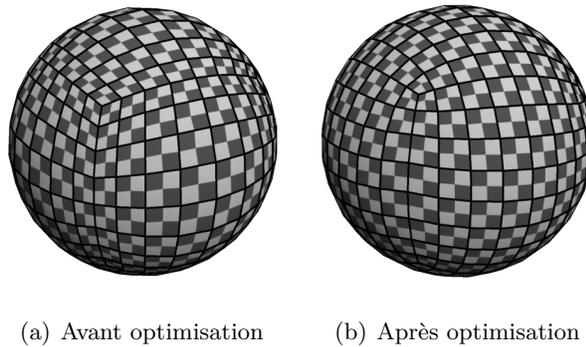


FIGURE 6.2 – L'optimisation d'un domaine d'échantillonnage cubique uniformise mieux l'aire des faces, produisant une meilleure distribution des sommets.

Le domaine d'échantillonnage est finalement prêt à être utilisé pour remailler la surface. Nous n'avons qu'à superposer celui-ci à la paramétrisation et effectuer un lancer de rayon pour chaque sommet du domaine. Le point d'intersection trouvé peut servir à interpoler n'importe quel attribut de la surface, notamment la position, la couleur ou la normale et, si désiré, stocker ces valeurs dans des images géométriques.

Toutefois, si le modèle géométrique contenait un bord initialement ou si des trous ont été coupés pendant la réduction de genre, une attention spéciale doit être portée à ceux-ci.

## 6.2 Gestion des composantes de bord

Bien que les trous et composantes de bord aient été colmatés pour donner au modèle une topologie de sphère, ceux-ci ne l'ont été que pour rendre le modèle admissible à la paramétrisation sphérique. Les triangles ajoutés durant ce processus ne sont que

temporaires et ne doivent pas être échantillonnés directement. Nous pouvons donc les éliminer et obtenir une topologie de sphère perforée. S'il y avait  $b$  composantes de bord dans le modèle géométrique original et que celui-ci était de genre  $g$ , nous avons maintenant  $2g + b$  composantes. La main de la figure 6.3 a un genre 1 et possède aussi une composante de bord à sa base, nous retrouvons donc un total de trois composantes.

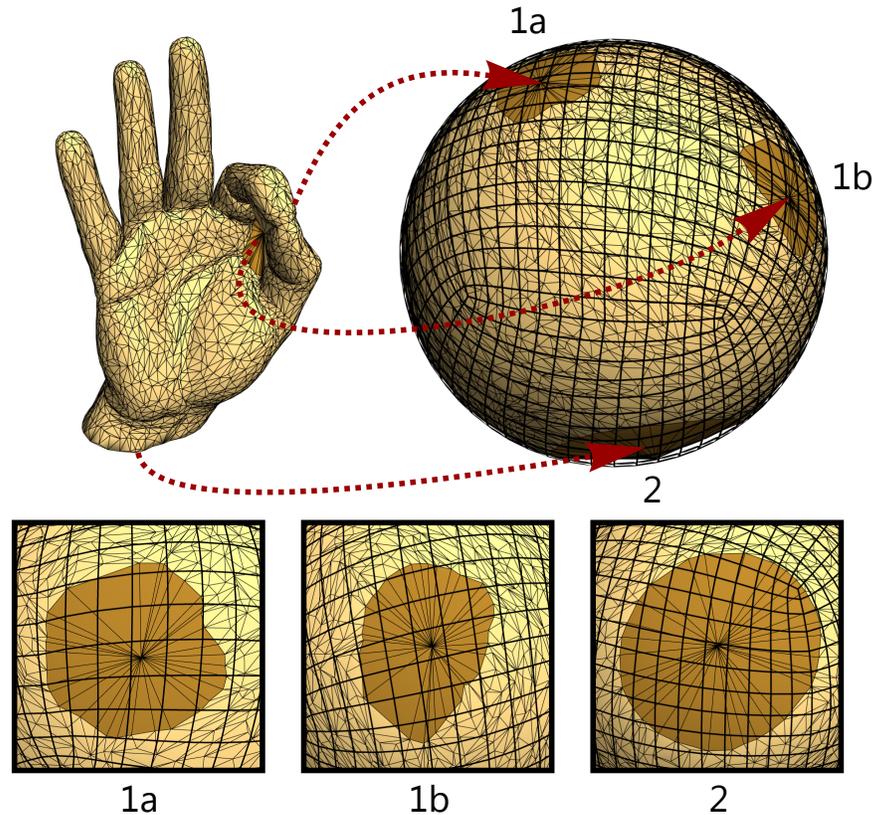


FIGURE 6.3 – Le trou découpé et colmaté entre le pouce et l'index a créé deux composantes de bord dans la paramétrisation. La troisième composante de bord située à la base de la main a simplement été colmatée au chargement.

Le problème est donc d'échantillonner une surface contenant des composantes de bord, et ce, en portant une attention spéciale aux paires issues du même trou. En effet, à chaque endroit où un générateur a été découpé pendant la réduction de genre, une paire de composantes a été créée. Pour permettre une reconstruction de qualité, il faut s'assurer de bien échantillonner chaque côté du trou original. Effectuer un échantillonnage et une reconstruction naïve sans considérer ce problème aurait pour effet de créer des fissures visibles à la surface aux endroits où les générateurs ont été coupés. La

figure 6.4 montre un exemple de ce phénomène.

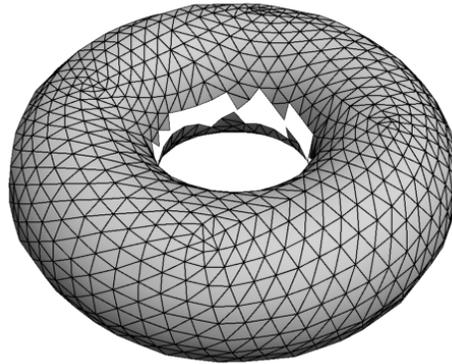


FIGURE 6.4 – Une fissure (en blanc) est bien visible dans la reconstruction à l’endroit où le générateur a été coupé.

La solution est d’échantillonner la surface de façon à ce que les deux côtés du trou aient les mêmes attributs de positions, normales, etc. Pour ce faire, deux stratégies peuvent être utilisées. La première serait de déplacer les sommets de la paramétrisation situés près des composantes de bord pour les faire coïncider avec les sommets du domaine d’échantillonnage, et ce, en restant cohérent de chaque côté du trou. À l’inverse, la deuxième méthode est de laisser la paramétrisation intacte, mais de déplacer légèrement les points échantillons pour les faire coïncider avec les composantes de bord. En pratique, la première méthode est assez difficile à appliquer puisqu’elle nécessite de modifier la paramétrisation ; nous opterons donc pour la seconde.

L’algorithme de modification des échantillons de bord s’articule en trois étapes :

1. Les sommets du domaine d’échantillonnage situés immédiatement aux abords d’une composante sont identifiés. Si celle-ci fait partie d’une paire, nous traitons l’autre en même temps.
2. On distribue les échantillons identifiés sur le contour de la composante de bord possédant le moins d’échantillons.
3. On tresse la fissure en faisant correspondre les échantillons situés aux abords de la seconde composante de bord à ceux de la première.

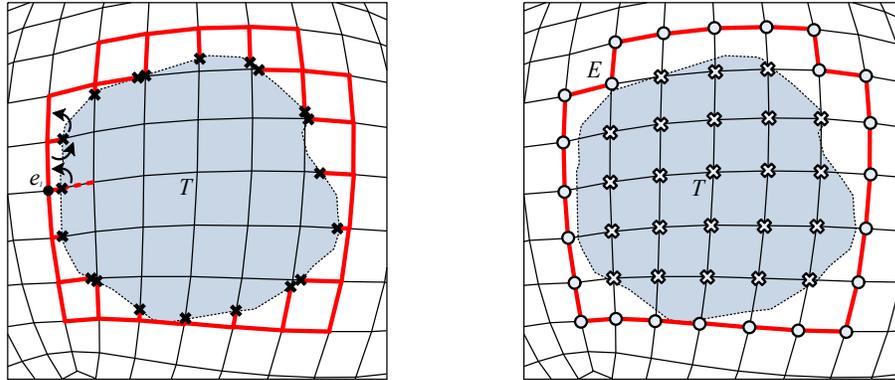
### 6.2.1 Identification des échantillons de bords

Nous débutons par identifier l'ensemble des sommets du domaine d'échantillonnage situés sur le pourtour de la composante de bord. Puisque les composantes de bord peuvent avoir des formes fortement concaves et que le domaine d'échantillonnage peut être quelconque, cette tâche ne peut être exécutée en effectuant un simple lancer de rayons. Tout comme un robot qui tenterait de contourner un obstacle en tournant toujours du même côté lorsqu'il entre en collision avec celui-ci, nous parcourons le contour de la composante dans la direction horaire (ou anti-horaire) en pivotant chaque fois que nous détectons une collision avec le contour de la composante de bord.

Soit  $T = \{t_1, t_2, \dots, t_n\}$ , l'ensemble des arêtes sur le modèle géométrique définissant la composante de bord. Nous désirons identifier l'ensemble des sommets  $E = \{e_1, e_2, \dots, e_m\}$ , sur le domaine d'échantillonnage, situés autour de  $T$ . Nous débutons par trouver une arête du domaine d'échantillonnage qui intersecte l'une des arêtes de  $T$  et sélectionnons le sommet du domaine qui est situé à l'extérieur de  $T$  comme étant le sommet de départ  $e_1$ . Il s'agit d'un test d'intersection entre deux grands arcs de cercle. Nous parcourons ensuite les voisins du sommet dans le sens anti-horaire en cherchant une arête ne générant pas d'intersection. Nous avançons ensuite au prochain sommet  $e_2$ , passons à la prochaine arête en sens anti-horaire suivant celle contenant  $e_1$  et testons à nouveau pour une intersection avec  $T$ . Nous itérons de cette façon jusqu'à ce que nous atteignons  $e_1$  à nouveau. Dans l'éventualité où il cela n'est pas possible, nous recommençons le processus avec une nouvelle arête de départ. La figure 6.5(a) illustre ce processus.

Le résultat de cette procédure est un chemin sur le domaine d'échantillonnage donnant la liste des sommets  $E$  situés immédiatement sur le bord de la composante. Les échantillons situés directement à l'intérieur de celle-ci sont identifiés en effectuant l'équivalent d'un remplissage par diffusion (*flood-fill*). Ces échantillons seront discartés pendant l'échantillonnage et leurs pixels associés, dans les images géométriques, recevront une valeur nulle (figure 6.5(b)).

Lors de la marche autour de la composante, nous ne permettons pas au chemin d'avancer sur un sommet directement adjacent à un sommet précédemment visité, la seule exception à cette règle étant le point de départ. Cette simple règle vise à laisser



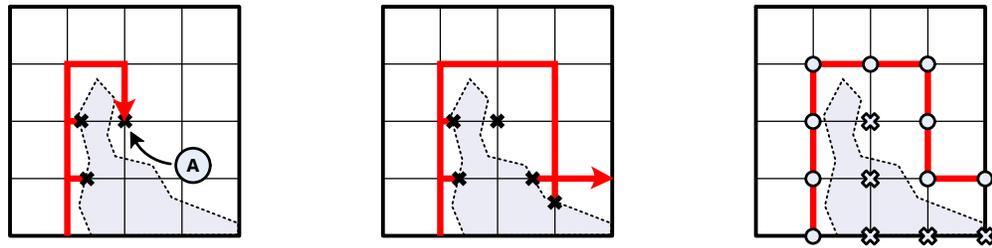
(a) Nous marchons le long de la bordure à partir du point de départ  $e_1$  en tournant lorsque nous détectons une intersection (croix noires).

(b) Un remplissage par diffusion nous permet de désactiver (X) les échantillons à l'intérieur de la composante de bord.

FIGURE 6.5 – Identification des échantillons situés autour de la composante de bord et annulation de ceux situés à l'intérieur.

délibérément un échantillon nul au milieu de petites composantes de bord ou de petites protubérances (figure 6.6). La raison pour laquelle nous ajoutons cette contrainte est pour simplifier la tâche du remaillage et elle sera éclaircie plus tard dans ce chapitre. Cette règle implique que la taille du plus petit ensemble  $E$  possible est 8. De façon similaire, si la composante provient d'une paire, nous ne permettons pas au chemin d'avancer vers un sommet adjacent à un sommet appartenant à la composante soeur (l'autre côté du trou bouché). Cette règle vise à éviter la création de quadrilatères dégénérés.

S'il n'y a pas assez d'échantillons disponibles pour encercler la composante de bord, nous essayons d'appliquer un facteur d'échelle à celle-ci, à partir du sommet central  $v_c$ , dans l'espace de la paramétrisation. Nous essayons quelques facteurs d'échelle prédéfinis (*e.g.* 0.5, 0.25, et 0.1) et essayons d'encercler la composante entre chaque transformation. Si après l'application du dernier facteur d'échelle, il est toujours impossible d'identifier un ensemble d'échantillons de bord valides, nous contractons la composante en son centre, ce qui aura pour effet de faire disparaître le trou complètement lors du remaillage. Cette fonctionnalité est particulièrement utile lorsqu'utilisée avec des domaines d'échantillonnage à basse résolution comme, par exemple, lors de la création d'une séquence de *mipmaps*. Nous en reparlerons au prochain chapitre.



(a) Le point A est adjacent à un point déjà traversé.

(b) L'algorithme évite donc ce sommet.

(c) L'échantillon est discarté par le remplissage par diffusion.

FIGURE 6.6 – Nous laissons consciemment un échantillon nul au bout d'une petite crête.

Si la composante de bord  $T$  provient d'une paire générée par le découpage d'un générateur, nous traitons l'autre membre de la paire immédiatement. La composante possédant le moins d'échantillons  $E$  est désignée par  $T^-$  tandis que l'autre est désignée par  $T^+$ . Les échantillons de chacune de ces composantes seront respectivement désignés par  $E^-$  et  $E^+$ . Si la composante ne fait pas partie d'une paire puisqu'elle était présente dans la surface originale,  $T^+$  et  $E^+$  seront vides. Pour garantir une reconstruction sans fissures, nous procédons en deux étapes. Dans un premier temps, les échantillons  $E^-$  sont distribués autour de  $T^-$  pour approximer la forme de celle-ci. Ensuite, les échantillons  $E^+$  sont déplacés pour les contraindre à échantillonner exactement les mêmes positions que  $E^-$ .

### 6.2.2 Distribution des échantillons autour de la composante de bord

Les échantillons de bord ayant été identifiés, nous devons maintenant les déplacer pour les aligner avec la silhouette de la composante. Le défi est de distribuer un nombre limité d'échantillons  $E^-$  de façon à approximer le mieux possible la forme originale de  $T^-$ . Plusieurs stratégies peuvent être utilisées pour accomplir ce travail. L'une d'elle serait de distribuer uniformément les échantillons  $E^-$  sur le bord de  $T^-$ . Par contre, puisque ceux-ci sont parfois distribués de façon inégale autour de la composante, nous croyons qu'il est préférable d'opter pour une stratégie tenant compte de la géométrie.

Pour chaque échantillon  $e_i^-$ , nous cherchons dans l'espace du monde le point le plus proche  $p_i^-$  sur les arêtes de  $T^-$ . Nous débutons par calculer la position des points  $e_i^-$

dans l'espace du monde. Ceci est effectué par une interpolation barycentrique sphérique dans le sens de Carfora [Car07]. Dans le domaine planaire, les coordonnées barycentriques sont proportionnelles aux aires des sous-triangles correspondants. De la même façon, sur le domaine sphérique, nous pouvons utiliser les ratios d'aires des sous-triangles sphériques comme poids pour effectuer une interpolation linéaire (figure 6.7). Ceci nous permet d'interpoler linéairement n'importe quel attribut présent aux sommets du modèle géométrique tel la position 3D, la normale ou des coordonnées de texture. De cette façon, nous pouvons aisément faire la correspondance entre la position de tout point dans l'espace du monde et dans l'espace de la paramétrisation.

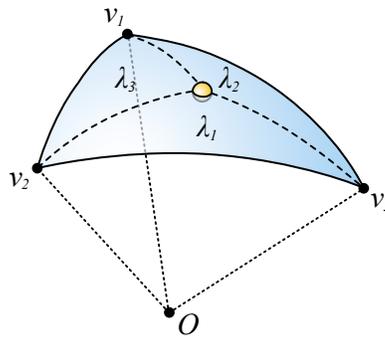
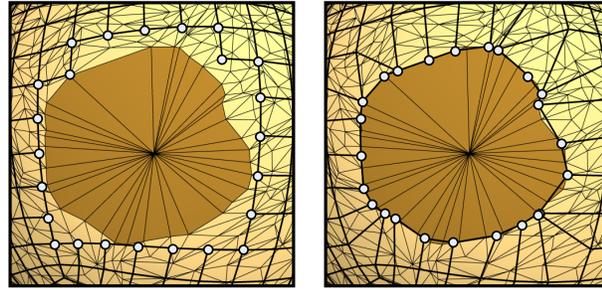


FIGURE 6.7 – Les aires normalisées des sous-triangles sphériques sont utilisées comme poids d'interpolation linéaire  $\lambda_1, \lambda_2, \lambda_3$ .

La séquence des points  $P^-$  trouvés étant parfois légèrement différente de la séquence des échantillons autour de la composante, nous retrions ceux-ci. Nous utilisons finalement ces nouvelles positions pour modifier la position des échantillons  $E^-$  dans l'espace de la paramétrisation. La figure 6.8 montre un exemple de l'application de cette procédure dans l'espace de la paramétrisation.

### 6.2.3 Tressage

La dernière étape vise à recoudre la fissure et ne s'applique qu'aux composantes de bord membres d'une paire. Nous désirons déplacer chaque échantillon de  $E^+$  de façon à fermer hermétiquement la fissure et éviter de créer des arêtes en «T» (*T-vertex*). Une des façons de garantir ceci est de forcer chaque échantillon de  $E^+$  à coïncider exactement avec l'un des échantillons de  $E^-$  et ce, en respectant la séquence des échantillons. Bien sûr, il doit y avoir au moins un  $e_i^+$  associé pour un  $e_i^-$  donné. Puisque  $|E^-| \leq |E^+|$ ,



(a) Avant

(b) Après

FIGURE 6.8 – Distribution des échantillons sur une composante de bord. Le domaine d'échantillonnage (grille régulière) est superposé sur la paramétrisation sphérique (triangulation).

il est possible que plusieurs  $e_i^+$  soient assignés au même  $e_j^-$ , ce qui en fait une mise en correspondance surjective (figure 6.9). La qualité du tressage obtenu peut être mesurée en additionnant l'aire des quadrilatères générés dans la reconstruction. Bien qu'il serait possible de trouver la configuration optimale en effectuant une recherche exhaustive, ceci n'est pas une approche pratique. En effet, il peut être démontré qu'il y a  $\binom{|E^+|-1}{|E^-|-1}$  configurations différentes, ce qui rend la recherche difficile lorsque la différence entre le nombre d'échantillons de chaque ensemble croît.

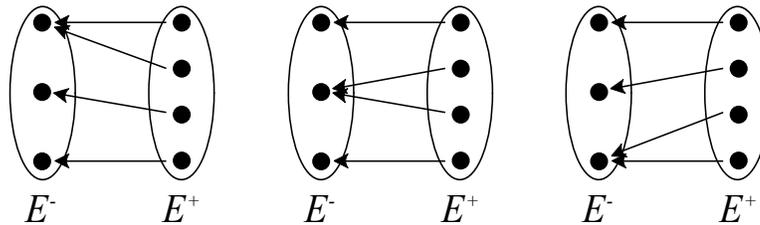


FIGURE 6.9 – Les trois configurations possibles pour  $|E^-| = 3$  et  $|E^+| = 4$ .

Pour obtenir un remaillage de qualité sans tester l'ensemble des possibilités, nous adoptons une stratégie de type *diviser pour régner*. Nous débutons par faire pivoter les éléments de  $E^+$  de façon à ce que  $e_1^+$  soit le point le plus près de  $e_1^-$  dans l'espace 3D. Nous divisons ensuite  $E^-$  en deux sous-ensembles  $E1^-$  et  $E2^-$  égaux. Nous divisons aussi  $E^+$  en deux sous-ensembles  $E1^+$  et  $E2^+$  de façon à ce que  $e2_1^+$  soit le point le plus près de  $e2_1^-$ . Ce faisant, nous imposons la contrainte que  $|E1^-| \leq |E1^+|$  et  $|E2^-| \leq |E2^+|$ . Nous répétons récursivement cette procédure en subdivisant à nouveau

chacun des ensembles par deux jusqu'à ce  $|E^-| = 1$  ou  $|E^-| = |E^+|$ . Dans le premier cas, nous affectons à tous les éléments de  $E^+$  la position de  $e_1^-$ , tandis que dans le second cas nous assignons  $e_i^+ = e_i^-$ . La figure 6.10 montre un exemple de ceci. Nous effectuons cette procédure en démarrant avec chacun des  $e_i^-$  et choisissons la configuration générant l'aire de remaillage la plus petite.

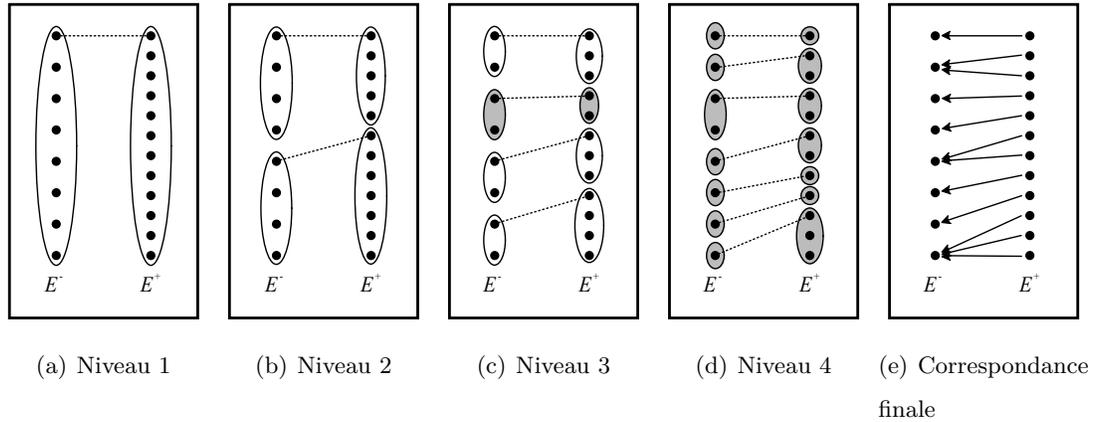


FIGURE 6.10 – Exemple de remaillage récursif. Les lignes pointillées représentent deux points les plus près dans l'espace 3D. Au niveau 3 la récursion arrête car  $|E^-| = |E^+|$  et au niveau 4, elle s'arrête car  $|E^-| = 1$ .

Il est important de noter que lorsqu'un trou est découpé, les deux composantes de bord ainsi générées ont leurs sommets ordonnés dans l'ordre opposé dans l'espace de la paramétrisation. Il faut donc inverser l'un des deux ensembles préalablement au traitement. Aussi, lorsque plusieurs échantillons de  $E^+$  sont associés à un seul de  $E^-$ , un quadrilatère dégénéré possédant deux ou trois sommets identiques est créé. À défaut de créer des intersections en «T», ceci est inévitable lorsque le nombre d'échantillons affecté à chaque composante est différent.

### 6.3 Échantillonnage

Lorsque les échantillons situés près des composantes ont été traités, la procédure d'échantillonnage est plutôt simple. Un rayon est lancé à partir de chaque sommet du domaine d'échantillonnage en direction de l'origine, et le point d'intersection avec la paramétrisation est trouvé. Évidemment, le lancer de rayons peut être accéléré avec une structure de données spatiale quelconque. La position 3D au point d'intersection peut

être calculée en utilisant l'interpolation barycentrique sphérique présentée à la section précédente. En effet, n'importe quel attribut présent dans le modèle géométrique original peut être interpolé linéairement de cette façon.

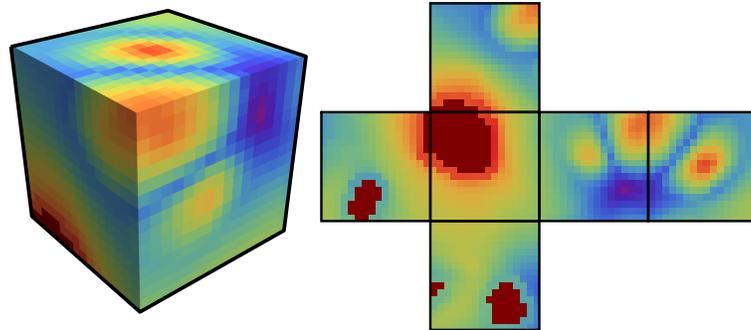


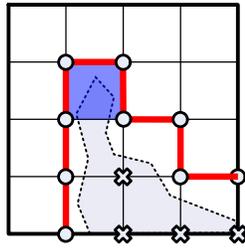
FIGURE 6.11 – Images géométriques pour le modèle de la main avec un domaine d'échantillonnage cubique de résolution  $6 \times 16 \times 16$ . Pour fin de visualisation, la norme euclidienne de chaque position est colorée avec une palette arc-en-ciel allant du violet au rouge. Les pixels nuls (trous) sont en rouge foncé.

Chaque valeur peut être utilisée pour effectuer un remaillage immédiatement ou peut être stockée dans des images géométriques en la copiant simplement dans le ou les pixels associés à chaque sommet du domaine (figure 6.11). Pour les échantillons situés à l'intérieur des composantes ayant été précédemment discartées, nous copions une valeur nulle dans les images.

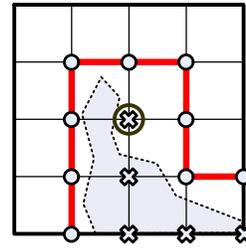
## 6.4 Reconstruction

Le modèle peut être reconstruit à partir du domaine ou des images. Dans les images, chaque carré de quatre pixels  $(x_i, y_j)$ ,  $(x_{i+1}, y_j)$ ,  $(x_{i+1}, y_{j+1})$  et  $(x_i, y_{j+1})$  représente un quadrilatère. Si un seul de ces pixels est nul, le quadrilatère est discarté. C'est d'ailleurs pour cette raison que nous laissons un pixel vide lorsque nous contournons une petite composante de bord ou une petite protubérance (figure 6.12). Si cette contrainte n'était pas appliquée, l'algorithme de remaillage ne pourrait pas déterminer quand créer ou ne pas créer de face. Pour éliminer l'ambiguïté, il faudrait ajouter de l'information supplémentaire ce qui alourdirait le procédé.

Pour générer un maillage de triangles, on peut diviser chaque quadrilatère en deux



(a) Un quadrilatère sera créé entre les quatre sommets en bleu.



(b) Éviter le sommet encerclé élimine l'ambiguïté.

FIGURE 6.12 – Lors de la marche autour de la composante de bord, nous évitons d'avancer sur un sommet adjacent à un sommet déjà visité.

triangles en ajoutant une arête de façon à générer la diagonale la plus courte. Nous ajoutons également une protection pour éviter de créer des triangles possédant un ratio d'aspect inférieur ou supérieur à des limites  $ratio_{min}$  et  $ratio_{max}$  spécifiés par l'utilisateur. La figure 6.13 montre un exemple de remaillage. Des exemples plus complexes seront présentés dans le prochain chapitre.

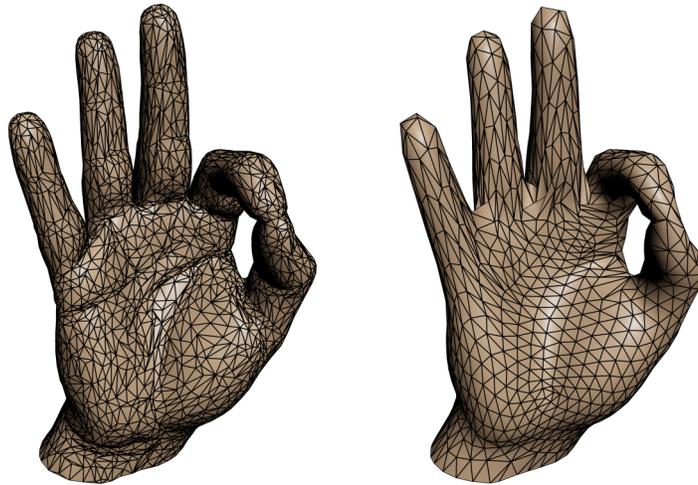


FIGURE 6.13 – À gauche, modèle original de la main avec 3000 sommets. À droite, remaillage de 1213 sommets à partir des images géométriques.

# Chapitre 7

## Résultats et applications

*And rather than hide that, I would rather put that out on the radio and let someone see the full range of emotions. If you're going to be strong on the radio, you got to let it all out, even the ugly stuff. And you can't apologize for it.*

—Howard Stern

### 7.1 Implémentation

Nous avons implémenté l'algorithme décrit en C++. Nous utilisons une représentation en demi-arête (*half-edge data structure*) pour toutes nos opérations sur des maillages.

L'algorithme de réduction de genre a été implémenté fidèlement à sa description. Pour simplifier nos expérimentations, lorsque l'utilisateur ne spécifie pas de triangles minimum et maximum pour la propagation de la fonction de Morse, nous calculons la boîte englobante de l'objet et prenons deux faces adjacentes aux points les plus éloignés sur l'axe le plus long. C'est d'ailleurs cette stratégie qui a été employée pour tous les exemples qui seront présentés dans ce chapitre.

L'algorithme de paramétrisation sphérique a été implémenté de façon similaire à Friedel *et al.* [FSD05]. Les fonctions d'énergies sphériques ont d'abord été codées avec *Maple* [Map]. Les premières et secondes dérivées de ces fonctions ont ensuite été générées et converties en C++ en utilisant les fonctionnalités de différentiation automatique. Le

code généré a ensuite été manuellement optimisé pour en augmenter la performance et la robustesse, ce qui a mené à des gains significatifs. Le code résultant a ensuite été combiné avec *TAO* [BMM<sup>+</sup>07], une puissante librairie implémentant divers algorithmes d’optimisations linéaires et non linéaires, avec et sans contraintes.

Pour la plupart des problèmes linéaires, nous utilisons leur implémentation du bi-gradient conjugué stabilisé (*Stabilized Bi-Conjugate Gradient* ou *BiCGStab*). Pour l’algorithme de paramétrisation, nous avons expérimenté avec les trois solveurs non linéaires sans contraintes disponibles, nommément *LMVM* (*Limited Memory, Variable Metric*), *NLS* (*Newton Line Search*) et *NTR* (*Newton Trust Region*). *LMVM* possède l’avantage de ne pas nécessiter de dérivée seconde puisque celle-ci est estimée numériquement à partir du gradient de la fonction et fonctionne très bien pour des modèles contenant peu de sommets. Les deux méthodes de Newton exploitent la dérivée seconde pour bâtir un estimé quadratique du voisinage de la fonction et décider les paramètres de la fonction à utiliser pour la prochaine itération. Bien que *NTR* devrait théoriquement être plus rapide que *NLS*, nous avons fait face à des problèmes de convergence avec *NTR* lorsque le nombre de variables devient important. Nous n’avons pas été en mesure de déterminer s’il s’agit d’un problème dans l’implémentation de *TAO* ou si cette méthode est plus sensible aux imprécisions numériques qui peuvent survenir lorsque les triangles deviennent très petits. Pour toutes ces raisons, nous avons favorisé la méthode *NLS* pour générer nos résultats.

Lorsque nous avançons dans la séquence de maillages progressifs et effectuons des subdivisions de demi-arêtes, il est possible que de très petits triangles soient créés temporairement. Pour cette raison, nous contraignons l’aire des triangles et les poids d’angles que nous fournissons au solveur à ne pas être inférieure à une valeur spécifiée par l’usager (*e.g.*  $10^{-8}$ ) pour maintenir la stabilité numérique.

Nous avons expérimenté avec cinq domaines d’échantillonnage soient la carte cubique, le tétraèdre, l’octaèdre, le *HEALPix* et le Isocube, tous illustrés à la figure 7.1. Les trois premiers sont des solides de Platon et doivent être projetés sur la sphère. Le cube se déplie en six images carrées, le tétraèdre en deux images carrées (ou une image rectangulaire) et l’octaèdre s’ouvre en une seule image.

Alors que ces trois domaines nécessitent d’être projetés sur la sphère, certains domaines sont définis directement sur celle-ci. Le *HEALPix* [GHB<sup>+</sup>05] (*Hierarchical Equal*

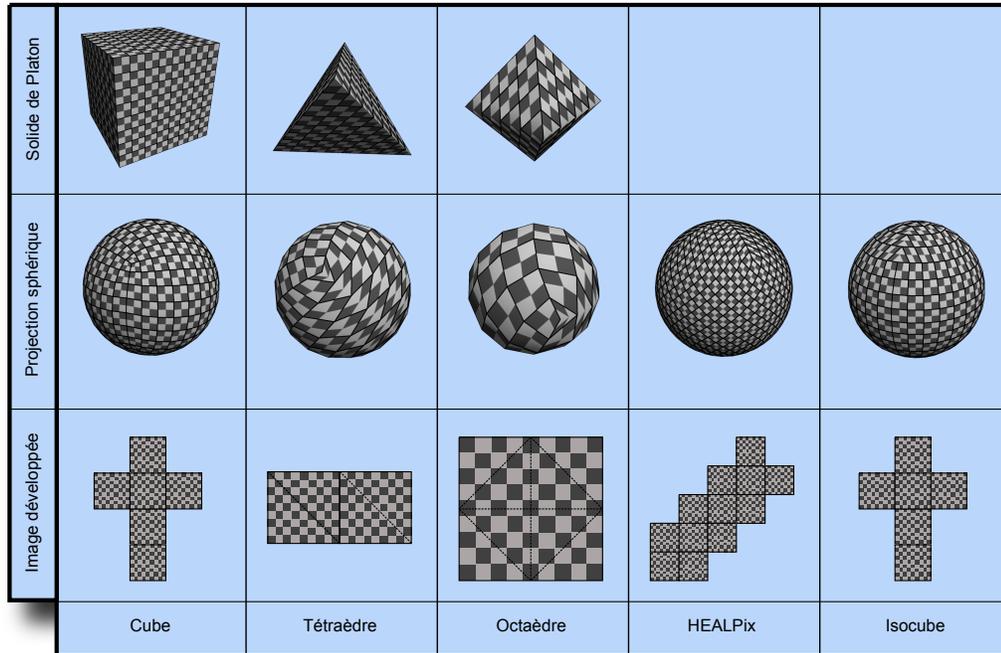


FIGURE 7.1 – Cinq domaines d'échantillonnage.

*Area isoLatitude PIXelisation*) est une partition hiérarchique de la sphère où chaque région couvre la même aire, ce qui, dans l'optique d'un rééchantillonnage, garantit une distribution plus uniforme des échantillons. Originellement développée pour l'étude du cosmos, cette pixelisation peut être vue comme l'assemblage de 12 images carrées. L'isocube [WWL07] est une variante de la carte cubique qui uniformise l'aire des pixels des images.

## 7.2 Résultats et applications

Cette section présente certains résultats que nous avons obtenus avec notre implémentation. Nous présenterons d'abord des exemples de remaillages et d'images géométriques générés avec notre méthode pour ensuite les comparer avec la méthode originale des images géométriques. Nous démontrerons finalement un exemple d'édition topologique de la surface rendue possible grâce à la représentation explicite des trous dans les images.

### 7.2.1 Remaillages et images géométriques

Le tableau 7.1 montre les résultats de notre algorithme pour quatre modèles de genres variant entre 1 et 8, échantillonnés avec quatre domaines différents. L'algorithme de traçage des générateurs a réussi à identifier correctement tous les générateurs parallèles malgré le fait que nous n'avons pas spécifié explicitement de triangles minimum et maximum pour la fonction de Morse. Puisqu'il y a relativement peu de générateurs par rapport au nombre de triangles, aucune subdivision barycentrique n'a été nécessaire.

Nous pouvons aisément constater que la paramétrisation sphérique a mené à une distribution assez uniforme des échantillons sur la surface, même lorsque le modèle présente des protubérances, ce qui peut être observé sur les remaillages en fil de fer.

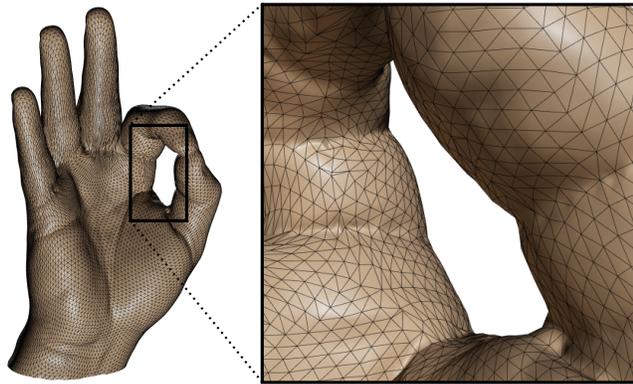


FIGURE 7.2 – Exemple de tressage d'une coupure sur la main.

L'algorithme de tressage des trous fonctionne bien et ne crée pas, tel qu'attendu, de jointures visibles ni de sommets en «T». La figure 7.2 montre un exemple de jointure pour le modèle de la main. Pour certains modèles, nous avons échantillonné les normales à une résolution supérieure et utilisé les textures résultantes pour faire du placage de normales (*normal mapping*) dans les rendus. Cette technique permet de créer l'illusion de détails fins qui interagissent avec l'illumination de la scène. L'exemple de l'amphore est particulièrement convaincant (figure 7.3).

Pour permettre un filtrage de qualité près des coupures, nous remplissons les pixels nuls dans les images de normales en solutionnant l'équation de Laplace. Ceci force chaque pixel inconnu à être égal à la moyenne de ses quatre voisins, les pixels situés sur la bordure servant de contraintes. La figure 7.4 montre un résultat de cette procédure.

En réduisant successivement la résolution du domaine d'échantillonnage, il est pos-

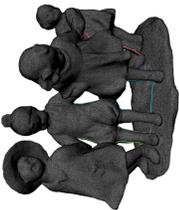
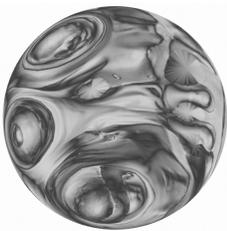
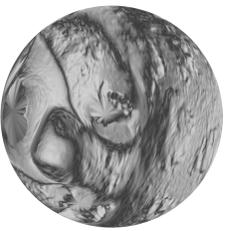
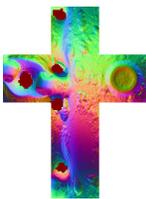
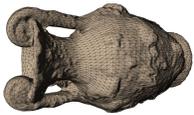
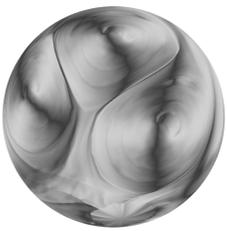
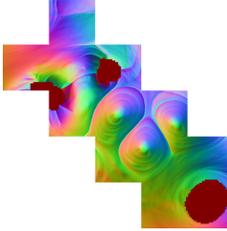
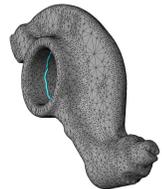
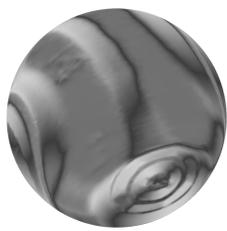
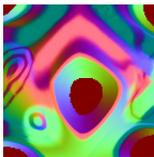
Modèle original	Paramétrisation	Image géométrique	Remaillage	Fil de fer
 100000 sommets (genre 8)		 Isocube $129 \times 129 \times 6$	 90609 sommets	
 280683 sommets (genre 2)		 Cube $33 \times 33 \times 6$ Normales $129 \times 129 \times 6$	 5880 sommets	
 50085 sommets (genre 1)		 HEALPix $33 \times 33 \times 12$ Normales $129 \times 129 \times 12$	 11130 sommets	
 10044 sommets (genre 1)		 Octaèdre $129 \times 129$ Normales $513 \times 513$	 14960 sommets	

TABLE 7.1 – Résultats obtenus avec notre méthode, de haut en bas : les enfants, l’amphore, la main et le culbuteur.

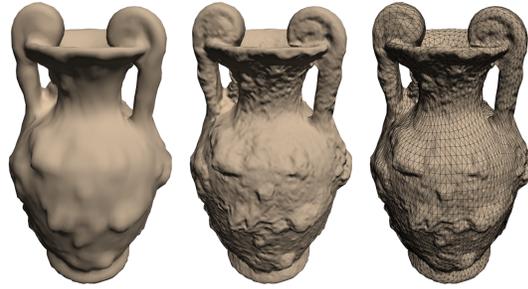


FIGURE 7.3 – L'échantillonnage des normales à une résolution supérieure (dans ce cas, 4 fois plus grande) permet de simuler de fins détails à la surface de l'amphore comparativement à l'image de gauche.

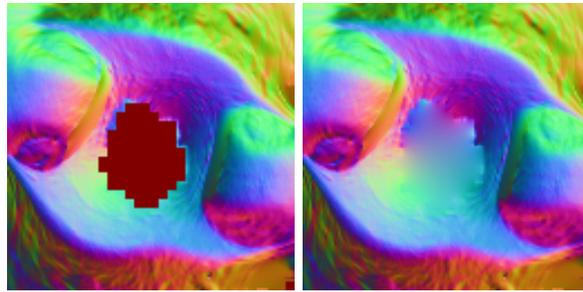


FIGURE 7.4 – Nous complétons les images de normales pour permettre un meilleur filtrage au niveau des jointures.

sible de créer des images géométriques représentant l'objet à différents niveaux de détail de façon similaire à un *mipmap*. Au fur et à mesure que la résolution diminue, la taille des trous pouvant être représentés diminue car trop peu d'échantillons sont disponibles pour bien les représenter. Lorsque ceci se produit, le genre du modèle s'en trouve réduit. La figure 7.5 montre un exemple de séquence *mipmap* avec le modèle du buddha.

Bien que l'algorithme ait fonctionné sur tous les modèles que nous avons essayés, certains modèles n'ont simplement pas une forme suffisamment sphérique pour bénéficier du plein potentiel de la méthode. Le dragon de la figure 7.6 est un exemple de modèle où la forme trop allongée crée des étirements visibles sur la surface du modèle reconstruit. Bien que les normales réussissent à masquer en partie le problème, un domaine de paramétrisation plus approprié que la sphère devrait être préférable.

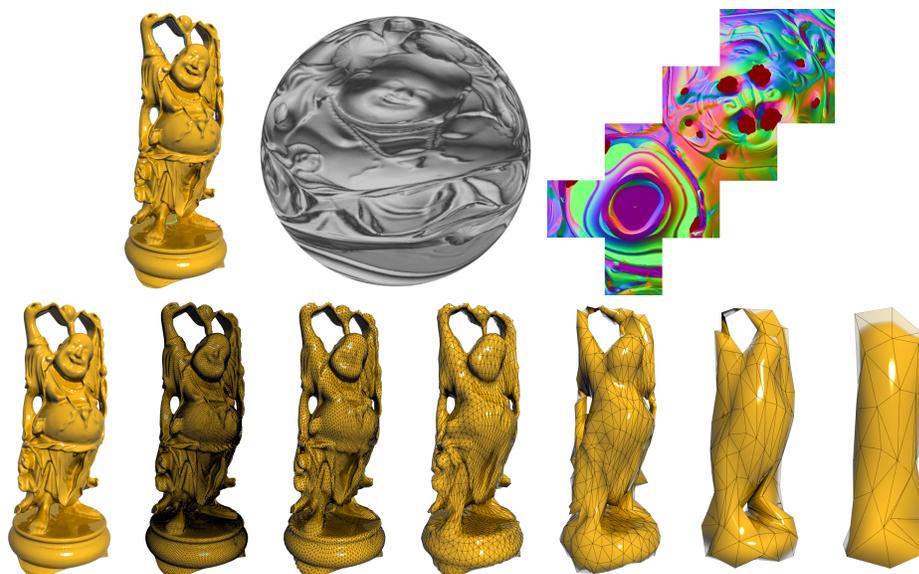


FIGURE 7.5 – En haut : Modèle du buddha avec ses 104 générateurs parallèles, sa paramétrisation sphérique et l'échantillonnage sur domaine *HEALPix* (normales). En bas : remaillages à différentes résolutions, dans l'ordre :  $129^2$ ,  $65^2$ ,  $33^2$ ,  $17^2$ ,  $9^2$ ,  $5^2$  et  $3^2$ . Le genre de chaque remaillage est respectivement : 11, 9, 9, 7, 4, 2 et 0.

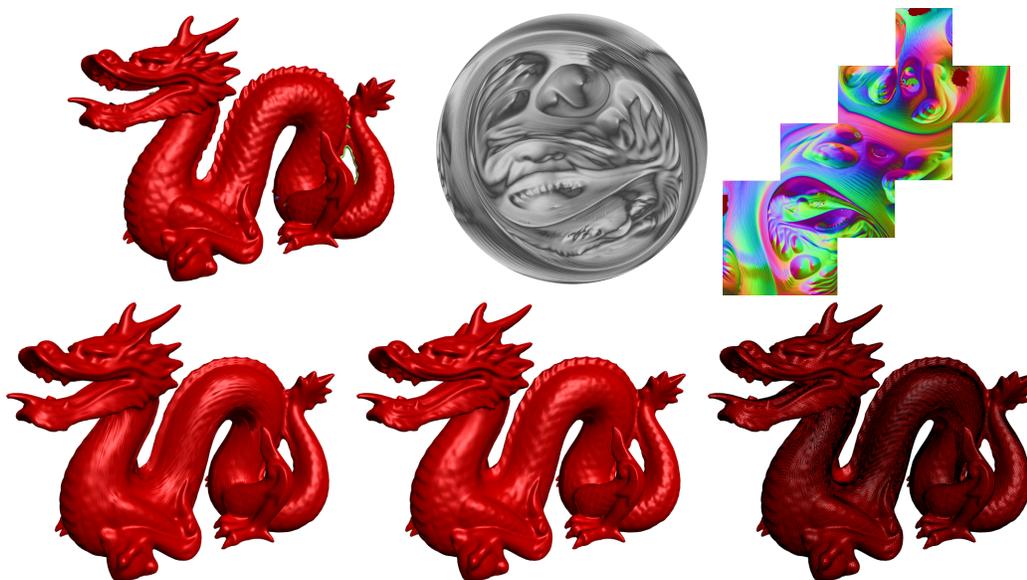


FIGURE 7.6 – En haut : Le modèle original du dragon, sa paramétrisation sphérique et l'échantillonnage avec un domaine *HEALPix*  $65 \times 65 \times 12$ . En bas : Reconstruction sans normales, avec normales ( $513 \times 513 \times 12$ ) et avec fil de fer.

### 7.2.2 Comparaison avec images géométriques traditionnelles

À titre de comparaison, nous avons implémenté la méthode originale des images géométriques de Gu *et al.* [GGH02] et mesuré le PSNR [CRS98] (*Peak Signal to Noise Ratio* ou *Rapport Signal sur Bruit de Crête*) des surfaces reconstruites à partir des images géométriques. Nous définissons le PSNR comme étant  $-20 \log_{10}(h)$  où  $h$  est la distance de Hausdorff par rapport à la diagonale de la boîte englobante.

Nous avons utilisé l’implémentation rapide de la paramétrisation par l’étirement géométrique [YBS04] (*fast geometric stretch*), qui est plus rapide de plusieurs ordres de magnitude que la méthode originale. Par contre, nous avons fait face à des problèmes de convergence lorsque le nombre de sommets devient trop important. Pour cette raison, nous avons limité la comparaison à des modèles de quelques milliers de sommets. Nous avons généré des images géométriques de modèles en utilisant différents domaines d’échantillonnage possédant approximativement le même nombre de pixels que la méthode originale.

Dans tous les cas, le PSNR de nos images géométriques sphériques est plus élevé et ce, malgré les quelques centaines de pixels gaspillés dans chaque image pour représenter les trous. Les remaillages et les paramétrisations résultantes sont aussi visuellement plus douces. Les résultats de cette comparaison sont résumés dans le tableau 7.2.

### 7.2.3 Edition topologique

Puisque nos images géométriques représentent explicitement les trous (*i.e.* chaque trou est représenté par deux composantes de bord) par des pixels nuls, il devient possible d’éditer la topologie de l’objet directement dans les images en utilisant des algorithmes de traitement d’images.

À titre d’expérience, nous avons solutionné l’équation de Laplace pour remplir les pixels nuls dans les composantes de bord. En forçant chaque pixel à être la moyenne de ses voisins et en utilisant les pixels connus comme conditions de bord, il devient possible de fermer de façon lisse un trou existant. Préalablement au remplissage de la composante, nous éliminons quelques rangées de pixels autour de celle-ci.

À l’inverse, il est possible d’ajouter un tunnel à un objet en annulant deux groupes de pixels dans les images et en les mettant en correspondance. Nous débutons par

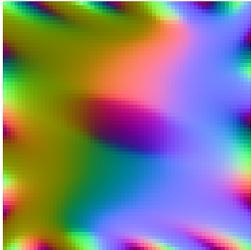
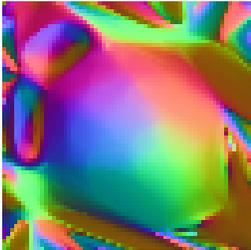
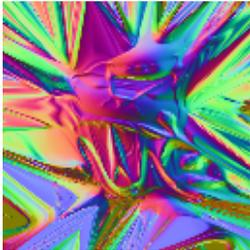
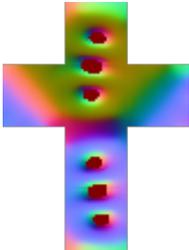
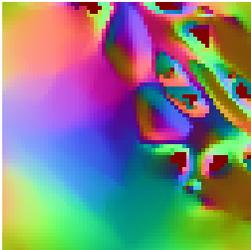
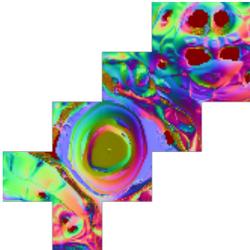
3-Tore	Crâne	Buddha
Modèles originaux		
 <p>4348 sommets genre-3</p>	 <p>3638 sommets genre-4</p>	 <p>20000 sommets genre-6</p>
Images géométriques traditionnelles		
 <p>65 × 65 4225 pixels PSNR 46.00</p>	 <p>65 × 65 4225 pixels PSNR 31.71</p>	 <p>129 × 129 16641 pixels PSNR 38.60</p>
Images géométriques sphériques de genre arbitraire		
 <p>Cube 27 × 27 × 6 4374 pixels (14% nuls) PSNR 46.38</p>	 <p>Octaèdre 65 × 65 4225 pixels (6% nuls) PSNR 34.91</p>	 <p>HEALPix 37 × 37 × 12 16428 pixels (11% nuls) PSNR 39.21</p>

TABLE 7.2 – Comparaison entre les images géométriques traditionnelles et notre méthode sphérique de genre arbitraire.

dessiner deux contours de périmètres égaux (*i.e.* de formes possiblement différentes mais contenant le même nombre de pixels) et annulons tous les pixels à l'intérieur des contours. Dans l'espace 3D, nous déplaçons les deux contours à la position moyenne de leurs barycentres et appliquons quelques itérations de relaxation pour rendre leur forme plus douce. Une étape de relaxation déplace chaque sommet du contour vers la position moyenne de ses deux voisins. Les sommets des deux contours sont ensuite fusionnés de façon à minimiser la distance totale de déplacement. Quelques rangées de pixels sont ensuite éliminées autour de chaque contour pour ensuite être remplies en utilisant encore l'équation de Laplace.

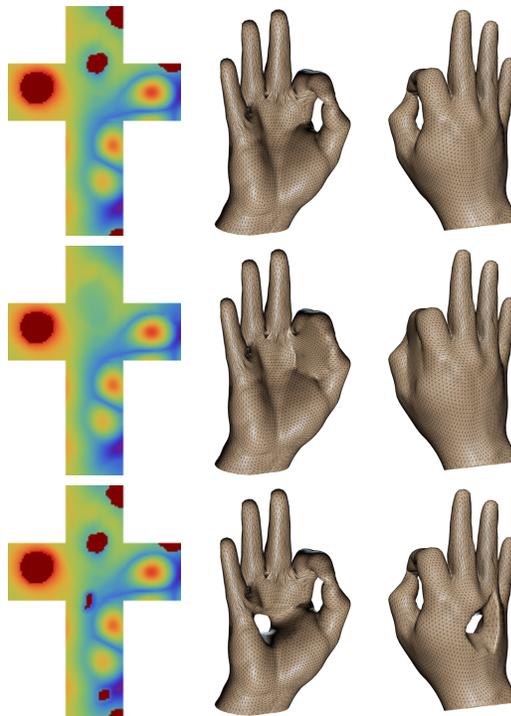


FIGURE 7.7 – De haut en bas : remaillage de la main avec carte cubique  $33 \times 33 \times 6$ , colmatage de trou entre le pouce et l'index, et creusage d'un nouveau trou.

La figure 7.7 montre les résultats de ces deux expériences. Dans un premier temps, le trou entre le pouce et l'index est colmaté. On peut voir que deux trous disparaissent dans les images de façon lisse. Dans un deuxième temps, on creuse un nouveau tunnel qui part de la paume de la main vers le dos de la main. On peut voir que dans l'espace des images, deux nouveaux groupes de pixels nuls de formes différentes sont créés et que leurs voisinages se confondent doucement avec le reste de l'image.

## Chapitre 8

# Conclusion et travaux futurs

*As I "galaxy-gaze" through time upon their diversity of colors, shapes, sizes, brightnesses, and structural detail, the boundary between knowledge and ignorance calls to me. When I reach for the edge of the universe, I do it knowing that along some paths of cosmic discovery, there are times when, at least for now, one must be content to love the questions themselves.*

—Neil deGrasse Tyson

La paramétrisation sphérique est très appropriée pour représenter en images géométriques les maillages de triangles de genre 0. Après avoir regardé les résultats de notre algorithme, il apparaît évident que la paramétrisation sphérique est également appropriée pour les surfaces de genres plus élevés ou même en présence de composantes de bord.

Pour produire ce type de résultats, nous avons présenté un pipeline complet, mais plus encore, nous avons développé une méthodologie robuste pour toutes les surfaces sur lesquelles nous l'avons essayée. Nous avons également amélioré l'efficacité d'un certain nombre d'étapes critiques dans le code. Nous avons été en mesure de paramétrer le buddha et le dragon de Stanford à partir de leurs représentations originales digitalisées, même si ces modèles présentent plusieurs anomalies topologiques.

Parmi nos contributions, nous avons amélioré le traçage et la classification des générateurs, hiérarchisé la méthode de paramétrisation sphérique sans contraintes et développé un algorithme de remaillage complet pour les paramétrisations sphériques conte-

nant des trous.

Bien que nous avons démontré que la réduction de genre fonctionne bien pour les surfaces avec des trous allant de petits à grands, nous croyons qu'il est possible de créer une paramétrisation naturelle aux surfaces de genre  $n$  sans avoir besoin de toujours réduire leur genre. Une telle paramétrisation pourrait être créée si nous sommes en mesure de trouver une paramétrisation naturelle au  $n$ -tore.

Couper la surface le long de tous les générateurs parallèles ou méridiens semble produire les meilleurs résultats. Nous devons développer une meilleure compréhension de cette observation et, si ceci ne s'avère pas être toujours le cas, nous devons être en mesure de sélectionner automatiquement le générateur qui mènera à la paramétrisation la plus naturelle pour chaque configuration de tunnels.

Puisque nous sommes en mesure de diminuer ou d'augmenter arbitrairement la taille des trous dans l'espace paramétrique, nous devons étudier quelle taille est la plus appropriée pour bien représenter les trous dans les images géométriques et par conséquent, sur la surface reconstruite.

L'échantillonnage régulier des images géométriques dans l'espace paramétrique ne permet pas la préservation des arêtes saillantes. Nous voulons tenter de déplacer les sommets du domaine d'échantillonnage pour mieux préserver ces arêtes ou développer une représentation alternative qui le permettrait.

$$\chi = V - E + F \tag{8.1}$$

$$g = \frac{(2 - b - \chi)}{2}. \tag{8.2}$$

# Bibliographie

- [ACSD<sup>+</sup>03] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy et Mathieu Desbrun. « Anisotropic polygonal remeshing ». Dans *SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers*, pages 485–493, New York, NY, USA, 2003. ACM.
- [Ale99] Marc Alexa. « Merging Polyhedral Shapes with Scattered Features ». Dans *SMI '99 : Proc. Intl Conf. on Shape Modeling and Applications*, pages 202–210, 1999.
- [BMBZ02] Henning Biermann, Ioana Martin, Fausto Bernardini et Denis Zorin. « Cut-and-paste editing of multiresolution surfaces ». *ACM Trans. Graph.*, volume 21, numéro 3, pages 312–321, 2002.
- [BMM<sup>+</sup>07] Steve Benson, Lois Curfman McInnes, Jorge Moré, Todd Munson et Jason Sarich. « TAO User Manual (Revision 1.9) ». Rapport technique ANL/MCS-TM-242, Mathematics and Computer Science Division, Argonne National Laboratory, 2007. <http://www.mcs.anl.gov/tao>.
- [BSM<sup>+</sup>03] Hector M. Briceño, Pedro V. Sander, Leonard McMillan, Steven Gortler et Hugues Hoppe. « Geometry videos : a new representation for 3D animations ». Dans *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 136–146. Eurographics Association, 2003.
- [BV05] Bedřich Beneš et Nestor Gómez Villanueva. « GI-COLLIDE : collision detection with geometry images ». Dans *SCCG '05 : Proceedings of the 21st Spring Conference on Computer Graphics*, pages 95–102. ACM Press, 2005.

- [Car07] Maria Francesca Carfora. « Interpolation on spherical geodesic grids : A comparative study ». *J. Comput. Appl. Math.*, volume 210, numéro 1-2, pages 99–105, 2007.
- [CHCH06a] Nathan A. Carr, Jared Hoberock, Keenan Crane et John C. Hart. « Fast GPU ray Tracing of dynamic meshes using geometry images ». Dans *GI '06 : Proceedings of the 2006 Conference on Graphics Interface*, pages 203–209. Canadian Information Processing Society, 2006.
- [CHCH06b] Nathan A. Carr, Jared Hoberock, Keenan Crane et John C. Hart. « Rectangular Multi-Chart Geometry Images ». Dans *SGP'06 : Eurographics Symposium on Geometry Processing*, pages 181–190, 2006.
- [CO04] Daniel Cohen-Or. « Computational Topology Powerpoint Presentation », 2004. <http://cg.cs.tu-berlin.de/~sorkine/courses/cg/cg2004/>.
- [CRS98] Paolo Cignoni, Claudio Rocchini et Roberto Scopigno. « Metro : measuring error on simplified surfaces ». *Computer Graphics Forum*, volume 17, numéro 2, pages 167–174, juin 1998.
- [Dey94] Tamal K. Dey. « A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection ». Dans *SCG '94 : Proceedings of the tenth annual symposium on Computational Geometry*, pages 277–284, New York, NY, USA, 1994. ACM.
- [DMA02] Mathieu Desbrun, Mark Meyer et Pierre Alliez. « Intrinsic Parameterizations of Surface Meshes ». *Computer Graphics Forum*, volume 21, numéro 3, pages 209–218, 2002. Eurographics conference proceedings.
- [DMK03] P. Degener, J. Meseth et R. Klein. « An adaptable surface parameterization method ». Dans *Proceedings of the 12th International Meshing Roundtable*, pages 201–213, 2003.
- [DS04] Carsten Dachsbacher et Marc Stamminger. « Rendering Procedural Terrain by Geometry Image Warping ». Dans *Rendering Techniques 2004 (Proceedings of Eurographics Symposium on Rendering)*, pages 103–110, 2004.
- [EDD<sup>+</sup>95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery et Werner Stuetzle. « Multiresolution analysis of arbitrary

- meshes ». Dans *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer Graphics and Interactive Techniques*, pages 173–182, New York, NY, USA, 1995. ACM.
- [FH05] Michael S. Floater et Kai Hormann. « Surface parameterization : a tutorial and survey ». Dans *Advances in Multiresolution for Geometric Modelling*, pages 157–186. Springer, 2005.
- [Flo97] Michael S. Floater. « Parametrization and smooth approximation of surface triangulations ». *Computer Aided Geometric Design*, volume 14, numéro 3, pages 231–250, 1997.
- [Flo03] Michael S. Floater. « Mean value coordinates ». *Computer Aided Geometric Design*, volume 20, numéro 1, page 2003, 2003.
- [FSD05] Ilja Friedel, Peter Schröder et Mathieu Desbrun. « Unconstrained spherical parameterization ». Dans *SIGGRAPH '05 : ACM SIGGRAPH 2005 Sketches*, page 134. ACM Press, 2005.
- [GGH02] Xianfeng Gu, Steven J. Gortler et Hugues Hoppe. « Geometry images ». Dans *Proceedings of the 29th annual Conference on Computer Graphics and Interactive Techniques*, pages 355–361. ACM Press, 2002.
- [GHB<sup>+</sup>05] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke et M. Bartelmann. « HEALPix : A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere ». *The Astrophysical Journal*, volume 622, pages 759–771, avril 2005.
- [HAT<sup>+</sup>00] Steven Haker, Sigurd Angenent, Allen Tannenbaum, Ron Kikinis, Guillermo Sapiro et Michael Halle. « Conformal Surface Parameterization for Texture Mapping ». *IEEE Transactions on Visualization and Computer Graphics*, volume 6, numéro 2, pages 181–189, 2000.
- [HG00] Kai Hormann et Günther Greiner. « MIPS : An Efficient Global Parameterization Method ». Dans P.-J. Laurent, P. Sablonnière et L. L. Schumaker, éditeurs. *Curve and Surface Design : Saint-Malo 1999*, Innovations in Applied Mathematics, pages 153–162. Vanderbilt University Press, Nashville, TN, 2000.

- [HNR68] Peter E. Hart, Nils J. Nilsson et Bertram Raphael. « A Formal Basis for the Heuristic Determination of Minimum Cost Paths ». *IEEE Transactions on Systems Science and Cybernetics*, volume 4, numéro 2, pages 100–107, 1968.
- [HP05] Hugues Hoppe et Emil Praun. « Shape Compression using Spherical Geometry Images ». Dans *Advances in Multiresolution for Geometric Modeling*, pages 27–46. Springer Berlin Heidelberg, 2005.
- [HR05] Benjamín Hernández et Isaac Rudomin. « FacePaint : Intuitive Modeling of Animatable Facial Expressions for High Density Models using Geometry Images and Shader Programs ». Dans *Proceedings International Conference in Computer Animation and Social Agents (CASA) 2005*, 2005.
- [HR06] Benjamín Hernández et Isaac Rudomin. « Simple dynamic LOD for geometry images ». Dans *GRAPHITE '06 : Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 157–163. ACM Press, 2006.
- [HSKK01] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura et Toshiyasu L. Kunii. « Topology matching for fully automatic similarity estimation of 3D shapes ». Dans *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer Graphics and Interactive Techniques*, pages 203–212, New York, NY, USA, 2001. ACM Press.
- [IGG01] Martin Isenburg, Stefan Gumhold et Craig Gotsman. « Connectivity shapes ». Dans *VIS '01 : Proceedings of the conference on Visualization '01*, pages 135–142, Washington, DC, USA, 2001. IEEE Computer Society.
- [KTI<sup>+</sup>01] Tomomichi Kaneko, Toshiyuki Takahei, Masahiko Inami, Naoki Kawakami, Yasuyuki Yanagida, Taro Maeda et Susumu Tachi. « Detailed shape representation with parallax mapping ». Dans *Proceedings of the International Conference on Artificial Reality and Telexistence 2001*, pages 205–208, 2001.
- [LFY<sup>+</sup>06] Shengying Li, Zhe Fan, Xiaotian Yin, Klaus Mueller, Arie E. Kaufman et Xianfeng Gu. « Real-time Reflection using Ray Tracing with Geometry

- Field ». Dans *Eurographics 2006 Proceedings (Short Papers)*, pages 29–32, 2006.
- [LHGM05] Y.-K. Lai, S.-M. Hu, D. X. Gu et R. R. Martin. « Geometric texture synthesis and transfer via geometry images ». Dans *SPM '05 : Proceedings of the 2005 ACM symposium on Solid and Physical Modeling*, pages 15–26. ACM Press, 2005.
- [LHSW03] Frank Losasso, Hugues Hoppe, Scott Schaefer et Joe D. Warren. « Smooth geometry images ». Dans *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry Processing*, pages 138–145. Eurographics Association, 2003.
- [LPRM02] Bruno Lévy, Sylvain Petitjean, Nicolas Ray et Jérôme Maillot. « Least squares conformal maps for automatic texture atlas generation ». *ACM Trans. Graph.*, volume 21, numéro 3, pages 362–371, 2002.
- [LTN06] Hamid Laga, Hiroki Takahashi et Masayuki Nakajima. « Spherical parameterization and geometry image-based 3D shape similarity estimation ». *Visual Computer (CGS 2004 special issue)*, volume 22, numéro 5, pages 324–331, 2006.
- [Map] Maple. « <http://www.maplesoft.com/> ».
- [NGH04] X. Ni, M. Garland et J. Hart. « Fair Morse functions for extracting the topological structure of a surface mesh ». *ACM Trans. Graph.*, volume 23, numéro 3, pages 613–622, 2004.
- [OBM00] Manuel M. Oliveira, Gary Bishop et David Mcallister. « Relief texture mapping ». Dans *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 359–368, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [PH03] Emil Praun et Hugues Hoppe. « Spherical parametrization and remeshing ». *ACM Trans. Graph.*, volume 22, numéro 3, pages 340–349, 2003.
- [Pix00] Pixologic. « Z-Brush », 2000. <http://pixologic.com>.
- [PSBM07] Valerio Pascucci, Giorgio Scorzelli, Peer-Timo Bremer et Ajith Mascarenhas. « Robust on-line computation of Reeb graphs : simplicity and speed ». *ACM Trans. Graph.*, volume 26, numéro 3, page 58, 2007.

- [RBM06] Tobias Ritschel, Mario Botsch et Stefan Müller. « Multiresolution GPU Mesh Painting ». Dans *Eurographics 2006 Short Papers*, pages 17–20, 2006.
- [SF03] Dvir Steiner et Anath Fischer. « Explicit Representation of Object Holes in Topological and Cut Graphs ». *The 4th Israel-Korea Bi-National Conference, Tel-Aviv, Israel*, février 2003.
- [SGD04] Alla Sheffer, Craig Gotsman et Nira Dyn. « Robust spherical parameterization of triangular meshes ». *Computing*, volume 72, numéro 1-2, pages 185–193, 2004.
- [SGSH02] Pedro V. Sander, Steven J. Gortler, John Snyder et Hugues Hoppe. « Signal-specialized parametrization ». Dans *EGRW '02 : Proceedings of the 13th Eurographics workshop on Rendering*, pages 87–98, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
- [SLMB05] Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky et Alexander Bogomyakov. « ABF++ : fast and robust angle based flattening ». *ACM Trans. Graph.*, volume 24, numéro 2, pages 311–330, 2005.
- [SPR06] Alla Sheffer, Emil Praun et Kenneth Rose. « Mesh parameterization methods and their applications ». *Found. Trends Comput. Graph. Vis.*, volume 2, numéro 2, pages 105–171, 2006.
- [SS00] Alla Sheffer et Eric De Sturler. « Surface parameterization for meshing by triangulation flattening ». Dans *Proc. 9th International Meshing Roundtable*, pages 161–172, 2000.
- [SSGH01] Pedro V. Sander, John Snyder, Steven J. Gortler et Hugues Hoppe. « Texture mapping progressive meshes ». Dans *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer Graphics and Interactive Techniques*, pages 409–416, New York, NY, USA, 2001. ACM.
- [SWG<sup>+</sup>03] Pedro V. Sander, Zoe Justine Wood, Steven J. Gortler, John Snyder et Hugues Hoppe. « Multi-chart geometry images ». Dans *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry Processing*, pages 146–155. Eurographics Association, 2003.
- [SYGS05] Shadi Saba, Irad Yavneh, Craig Gotsman et Alla Sheffer. « Practical Spherical Embedding of Manifold Triangle Meshes ». Dans *SMI '05 : Proceedings*

- of the International Conference on Shape Modeling and Applications 2005*, pages 258–267, Washington, DC, USA, 2005. IEEE Computer Society.
- [THCM04] Marco Tarini, Kai Hormann, Paolo Cignoni et Claudio Montani. « PolyCube-Maps ». *ACM Trans. Graph.*, volume 23, numéro 3, pages 853–860, 2004.
- [Tut63] W. T. Tutte. « How to draw a graph ». *Proc. London Math. Soc.*, volume 13, pages 743–767, 1963.
- [WHDS04] Zoe Justine Wood, Hugues Hoppe, Mathieu Desbrun et Peter Schroeder. « Removing excess topology from isosurfaces ». *ACM Trans. Graph.*, volume 23, numéro 2, pages 190–208, 2004.
- [Wik08] Wikipedia. « Topology », 2008. <http://en.wikipedia.org/wiki/Topology>.
- [WWL07] Liang Wan, Tien-Tsin Wong et Chi-Sing Leung. « Isocube : Exploiting the Cubemap Hardware ». *IEEE Transactions on Visualization and Computer Graphics*, volume 13, numéro 4, pages 720–731, 2007.
- [YBS04] Shin Yoshizawa, Alexander Belyaev et Hans-Peter Seidel. « A Fast and Simple Stretch-Minimizing Mesh Parameterization ». Dans *SMI '04 : Proceedings of the Shape Modeling International 2004*, pages 200–208, Washington, DC, USA, 2004. IEEE Computer Society.
- [YNZC05] Xiaoru Yuan, Minh X. Nguyen, Nan Zhang et Baoquan Chen. « Stippling and Silhouettes Rendering in Geometry-Image Space ». Dans *Proceedings of Eurographics Symposium on Rendering 2005 (EGSR'05)*, pages 193–200, 2005.
- [ZMT05] Eugene Zhang, Konstantin Mischaikow et Greg Turk. « Feature-based surface parameterization and texture mapping ». *ACM Trans. Graph.*, volume 24, numéro 1, pages 1–27, 2005.
- [ZRS06] Rhaleb Zayer, Christian Rössl et Hans-Peter Seidel. « Curvilinear Spherical Parameterization ». Dans *Shape Modeling International (SMI)*, pages 57–64, Matsushima, Japan, 2006. IEEE.