N° d'ordre : D.U. 1966
EDSPIC : 452

Université Blaise Pascal – Clermont-Ferrand II

École Doctorale
Sciences pour l'Ingénieur de Clermont-Ferrand

THÈSE

présentée par

Jean-Marie Favreau

pour obtenir le grade de

Docteur d'Université

Spécialité : Informatique

Outils pour le pavage de surfaces

Soutenue publiquement le 22 octobre 2009 devant le jury :

M. Jean-Marc Chassery	Rapporteur	Directeur de Recherche au CNRS, Grenoble
M. Bruno Lévy	Rapporteur	Directeur de Recherche à l'INRIA Nancy
M. Pierre Alliez	Examinateur	Chargé de Recherche à l'INRIA Sophia-Antipolis
M. Éric Colin de Verdière	Examinateur	Chargé de Recherche à l'ENS
M. Yan Gérard	Visiteur	Maître de Conférences à l'Université d'Auvergne
M. Philippe Mahey	Président	Professeur à l'Université Blaise Pascal
M. Vincent Barra	Directeur de thèse	Professeur à l'Université Blaise Pascal

À Maëlyn

Remerciements

Je tiens tout d'abord à remercier Vincent Barra, qui m'a accueilli au LIMOS et accompagné pendant ces trois années de thèse, pour sa disponibilité de tous les instants et la liberté d'exploration scientifique qu'il m'a offerte. J'ai particulièrement aimé découvrir à ses côtés combien il était enrichissant d'aiguiser sans cesse sa curiosité scientifique, sans pour autant faire l'erreur de l'éparpillement.

Le sujet de ma thèse n'aurait pu exister sans l'année de pré-doc que j'ai passée au sein de l'Équipe de Recherche en Imagerie Médicale (ERIM) de Clermont-Ferrand, aussi je souhaite remercier les membres avec qui j'ai échangé, tout d'abord Jean-Jacques Lemaire, auprès de qui j'ai découvert que les chirurgiens et le monde médical en général étaient avides d'avancées scientifiques et informatiques ; Simone Hemm, qui m'a aidé à comprendre la complexité de la recherche en imagerie médicale, aux interfaces entre disciplines riches de cultures différentes, et enfin Laurent Sarry, dont la gourmandise scientifique est à chaque fois aussi passionnante.

Ma gratitude va également à Jean-Marc Chassery et Bruno Lévy, qui m'ont fait l'honneur d'être les rapporteurs de cette thèse. Je tiens notamment à remercier Bruno Lévy pour la richesse des discussions scientifiques que nous avons eu lors de ma visite à Nancy. Merci également à Pierre Alliez, Éric Colin de Verdière et Yan Gérard qui ont accepté de prendre part à mon jury de thèse. J'ai particulièrement apprécié travailler avec Yan Gérard aux questions sur la quadrangulation et de découpage en cylindres, et les échanges réguliers que nous avons eu avec Éric Colin de Verdière au sujet des n-cets ont été très enrichissants.

Je voudrais également remercier les autres chercheurs avec qui j'ai eu l'occasion d'échanger pendant ces trois années : Émilie Kœnig et Pierre Chainais qui ont apporté aux réunions d'équipe un œil extérieur enrichissant sur les questions que j'explorais, Christophe Tilmant pour les discussions autour des méthodes d'imagerie médicale, et Marc Chevaldonné notamment pour les échanges autour de la conception logicielle. Ma gratitude va également à Laurent Fuchs, dont le tour d'horizon des différentes structures topologiques m'a été très utile. Merci également à l'équipe de Lourdes Agapito, au département de Science Informatique du Queen Mary College de l'Université de Londres, qui m'a accueillie pendant deux mois de l'été 2009. J'ai ainsi pu découvrir les questions de géométrie spécifiques à la capture de mouvements.

Je souhaite aussi remercier les différentes personnes qui m'ont ouvert les portes de l'infographie à Allegorithmic, me faisant ainsi découvrir les réalités et besoins multiples de la profession. En particulier, je remercie Emmanuel Laurent pour sa patience et sa pédagogie, ainsi que Thierry Frey et Sylvain Lefebvre pour les discussions scientifiques autour de ces problématiques.

Ma reconnaissance va à Samuel Blanchard, qui a été pendant ces trois années un interlocuteur privilégié quand il s'agissait d'explorer les mathématiques discrètes et leurs structures algébriques. Merci à lui d'avoir été patient, et de m'avoir offert ses connais-

sances, sa disponibilité et sa gentillesse. J'ai retrouvé avec ces discussions le même plaisir que nous avions à suivre l'excellent cours de topologie algébrique de François Laudenbach.

Ces quelques années à Clermont-Ferrand ont été l'occasion de tisser de nombreuses relations d'amitié. Ainsi, je tiens à remercier Romain et Romuald pour les longues soirées passées à *geeker*, Nicolas et Pauline pour les bons moments et bons repas partagés, Susanne pour les nombreuses promenades nocturnes à travers la ville, Vanessa pour nos discussions débridées. Un grand merci à Claire, qui m'a soutenu dans cette dernière ligne droite de rédaction.

Et puisque les liens d'amitié s'entretiennent aussi à distance, je veux remercier les amis avec qui j'ai continué à échanger : Claire et Raphaël, pour leur bonheur communicatif, les libristes de Gulliver Gwénaëlle, David, Frédéric et Arnaud, ainsi qu'Emmanuel, Géraldine et Josselin, les nantais expatriés. Ces échanges électroniques, visites et conversations téléphoniques ont été le moteur de mon quotidien.

Je ne pourrais finir ces remerciements sans une pensée personnelle aux membres de ma famille. Je remercie tout d'abord mes parents et grands-parents, qui m'ont apporté leur soutien quand il était nécessaire. Merci également du fond du cœur à mes sœurs Tifenn et Émeline, ainsi que ma fille Maëlyn et sa mère Elise, pour m'avoir offert leur soutien moral et partagé ces moments de bonheur qui permettent de continuer à avancer.

Sommaire

Introduction **1**

I Contexte **3**

1 Variétés topologiques **5**
1.1 Espaces topologiques 6
1.2 Classification 7
1.3 Schéma polygonal 11
1.4 n-cets 12

2 Variétés discrétisées **17**
2.1 Structures élémentaires 18
2.2 Invariants topologiques 22
2.3 Mesures géométriques sur les surfaces 25
2.4 Pavage de surfaces 29
2.5 Caractéristique d'Euler généralisée et conséquences 35
2.6 Conclusion 37

II Pavage sous contraintes **39**

3 Contraintes topologiques **41**
3.1 Découpages élémentaires 42
3.2 Découpage en disque 47
3.3 Découpage minimal 49
3.4 Conclusion 57

4 Contraintes géométriques **59**
4.1 Distance non euclidienne 60
4.2 Détection des extremums 63
4.3 Patch des surfaces 67
4.4 Conclusion 69

5 Contraintes sur les tuiles **71**
5.1 Considérations générales 72
5.2 Découpage en cylindres 73
5.3 Nombre de sommets d'une tuile unique 82
5.4 Découpage en quadrangles 84

5.5 Conclusion 90

III Applications **91**

6 Découpages pour l'infographie **93**
6.1 Distances non euclidiennes 94
6.2 UV mapping de formes organiques 96
6.3 Segmentation organique 102
6.4 Conclusions et perspectives 105

7 Application médicale **107**
7.1 Contexte général 107
7.2 Cartographie du cortex 108
7.3 Application à la stimulation corticale 112
7.4 Conclusion et perspectives 117

Conclusion **119**

Annexes **125**

A Taglut **125**
A.1 Langages et bibliothèques 126
A.2 Structures de données et implémentation 128
A.3 Échanges et interactions 132

B Variations autour du dépliage **137**
B.1 Circle packing : descriptif rapide 137
B.2 Contraintes aux bords 138
B.3 Parallélisation 140
B.4 Multi-résolution 140
B.5 Conclusion 141

C Géométrie des nuages de points **143**
C.1 Travaux similaires 144
C.2 Du nuage de points à l'estimation de distortion globale 144
C.3 Résultats et perspectives 148

D Algorithmique illustrée **153**

Bibliographie **161**

Index **169**

Liste des figures **171**

Table des matières **175**

Introduction

L'avènement de l'outil informatique a imposé l'ordinateur comme solution incontournable au traitement de données. En particulier, cela a rendu possible la manipulation d'informations complexes et volumineuses, telles que peuvent l'être par exemple les descriptions d'objets 3D. De nombreux domaines de l'ingénierie sont ainsi devenus créateurs et utilisateurs de données décrivant des surfaces, qu'elles soient issues d'acquisitions réelles ou modélisées virtuellement. Il existe ainsi un grand nombre de descriptions de telles données, depuis les approches mathématiques issues de l'analyse jusqu'à la modélisation informatique discrète.

Les outils de manipulation de surfaces sont naturellement devenus essentiels à la modélisation et à l'analyse des données disponibles. Parmi ces outils, le découpage de surfaces est une approche significative car elle répond à de nombreux besoins classiques, depuis la segmentation jusqu'à la paramétrisation de surfaces discrètes, en passant par la classification ou la mise en correspondance d'informations. Moins directement, le découpage de surfaces est utilisé dans l'animation d'objets virtuels, ou encore en imagerie médicale où il est l'une des briques élémentaires du traitement des données.

Comme nous le soulignons dans le deuxième chapitre, il existe une dualité entre découpage et pavage de surfaces. De nombreuses approches permettent de les décrire, aussi bien issues de domaines mathématiques que de l'informatique. On peut par exemple citer la géométrie différentielle, qui propose des outils de calcul puissants sur ces structures. De la même manière, un grand nombre d'outils développés pour le traitement d'images peut être envisagé lorsque l'on manipule des données surfaciques.

Nous présentons dans ce mémoire un cadre général permettant de décrire les propriétés topologiques et géométriques des pavages, sans pour autant être lié à une approche de construction. Nous utilisons par la suite des outils issus de la topologie algébrique pour classifier et décrire de tels pavages, puis nous proposons plusieurs approches permettant de produire des pavages répondant à des contraintes topologiques, géométriques, et de propriétés des tuiles.

Si les modèles formels et les algorithmes sont souvent sur le papier intéressants à explorer pour eux-mêmes, afin de cerner leur fonctionnement et d'analyser leurs capacités propres, la question de l'application à des exemples concrets arrive inévitablement à un moment ou un autre.

Cette problématique complémentaire nous a ainsi intéressé pendant ces trois années, aussi nous avons exploré certaines des possibilités offertes par le cadre général et les algorithmes que nous avons proposé. Outre les questionnements liés aux domaines applicatifs que nous avons abordés, la motivation commune de ces différents travaux a consisté à

explorer les réponses que pouvaient apporter l'introduction d'informations géométriques et topologiques au cœur de ces problèmes spécifiques.

Ces différentes applications de nos travaux – dont deux principales sont présentées dans ce mémoire – ont été l'occasion de rencontrer plusieurs spécialistes de ces différents domaines, et d'échanger avec eux autour de leurs problématiques spécifiques. Ces échanges inter-disciplinaires ont été un réel enrichissement, et ont amené des questions originales auxquelles nous avons tenté d'apporter une réponse.

Ce mémoire est structuré en trois parties.

La première partie pose les briques mathématiques essentielles à la suite de notre propos. Certains éléments sont classiques, et viennent ancrer le discours dans un contexte existant. Nous profitons ainsi du grand nombre d'outils mathématiques déjà développés pour décrire les propriétés des surfaces, et les manipuler. Quelques autres éléments présentés dans cette partie sont originaux, tout en s'appuyant sur des concepts classiques. Ainsi, la généralisation de la notion de lacet avec les n-cets permet de décrire des familles de pavages originales. Les structures de M-tuiles et M-pavages répondent quant à elles aux besoins spécifiques de description des pavages.

La deuxième partie s'appuie sur les concepts décrits dans la première partie pour proposer différentes méthodes de pavage de surface, répondant à des contraintes spécifiques. En particulier, nous nous sommes intéressés aux propriétés topologiques des surfaces, et avons proposé une approximation du découpage minimal en un disque topologique. Nous avons également intégré des propriétés géométriques, plus difficiles à mesurer, mais qui sont essentielles à de nombreuses applications. Enfin, nous avons exploité les propriétés combinatoires du formalisme de pavage proposé pour décrire quelques découpages en tuiles multiples.

La troisième partie propose deux exemples d'application de ces idées, en introduisant à chaque fois des contraintes applicatives aux pavages, afin d'ajuster les approches générales décrites précédemment. Dans un premier chapitre, nous proposons ainsi de montrer que les idées développées dans les deux premières parties peuvent aboutir à des outils pratiques à destination de l'infographiste. Puis dans un second chapitre nous présentons une application médicale à nos outils de manipulation de surface, en illustrant nos propos dans le contexte de la stimulation corticale.

Après avoir réalisé une synthèse des différents travaux réalisés pendant ces trois années, nous présentons en conclusion les perspectives qu'ils laissent entrevoir, aussi bien dans le contexte de la géométrie algorithmique que d'un point de vue applicatif.

Enfin, on trouvera dans une première annexe une description de l'outil logiciel développé pendant cette thèse. Il s'agit d'un prototype visant à illustrer la faisabilité des différentes idées présentées ici, et qui a permis notamment de mesurer l'efficacité pratique des différents algorithmes et méthodes. Dans une deuxième annexe, nous présentons quelques explorations réalisées autour de la question du dépliage, dans une optique d'amélioration de méthodes existantes. Dans une troisième et dernière annexe, on trouvera la description d'une méthode originale d'extraction d'informations géométriques depuis un nuage de points, avec une application à la capture de mouvements.

Première partie

Contexte

Chapitre 1
Variétés topologiques

Sommaire

1.1 Espaces topologiques 6
1.2 Classification 7
1.2.1 Chemins et lacets 8
1.2.2 Groupe fondamental 8
1.2.3 Genre et bords 10
1.3 Schéma polygonal 11
1.4 n-cets 12

Notre travail porte sur l'étude de propriétés topologiques et géométriques de surfaces manipulées à travers leur représentation informatique. Avant d'introduire au chapitre suivant les structures et algorithmes discrets qui nous permettent de manipuler concrètement une surface, nous présentons dans ce chapitre quelques notions mathématiques générales qui permettent de caractériser et décrire les propriétés topologiques des surfaces.

En particulier, nous présentons ici les définitions usuelles de *variété*, de *surface*, ainsi que les outils classiques permettant de classifier les surfaces suivant leurs propriétés topologiques : *chemin* et *lacet* permettant de définir le *groupe fondamental* d'une surface[1], ou encore les notions de genre et d'homéomorphisme. Le lecteur intéressé pourra consulter [15] qui présente une approche analytique de la topologie, ou encore [75, 50] pour les questions de topologie algébrique. On pourra également consulter [45] qui traite plus spécifiquement des surfaces. Ces ouvrages ont servi de référence à la rédaction des deux premières parties de ce chapitre.

Dans une dernière partie, nous proposons une structure originale, que nous appelons *n-cet*, permettant de capturer des informations topologiques, mais également des informations géométriques fines.

[1] Si nous avons choisi de présenter ici le groupe fondamental plutôt que les groupes d'homotopies, tel qu'il sera fait dans la section 2.2, c'est parce qu'il a d'une part l'avantage de nécessiter peu de constructions préalables, mais aussi parce qu'il est relativement transparent, comme le fait remarquer Allen Hatcher dans [50].

1.1 Espaces topologiques

Nous présentons ici une approche de la topologie qui consiste à redéfinir de nombreuses propriétés usuelles, comme la notion de limite ou la continuité, en terme d'ouverts. La définition d'espace topologique qui suit est ainsi la base de constructions complètes que nous n'aborderons pas ici, mais qui en découlent naturellement.

Définition 1.1 – *Un* espace topologique *E est un ensemble pour lequel on a choisi une famille de parties, dites* ouverts*, satisfaisant les propriétés suivantes :*
- *toute réunion (finie ou non) d'ouverts est un ouvert ;*
- *toute intersection finie d'ouverts est un ouvert ;*
- *l'ensemble E et l'ensemble vide $\varnothing$ sont des ouverts.*

Définition 1.2 – *On appelle* voisinage *d'un point x de E tout sous-ensemble V_x de E contenant un ouvert contenant x.*

La notion de voisinage ainsi définie permettra par la suite à définir une variété en caractérisant localement l'espace topologique :

Définition 1.3 – *On dit qu'un espace topologique E est* séparé *si pour tout $x, y \in E$ distincts, il existe V_x et V_y des voisinages disjoints de x et y.*

Exemple : $\mathbb{R}^n$ (avec $n > 0$) muni des ouverts définis comme réunions de pavés ouverts est séparé. En effet, étant donnés $x, y \in \mathbb{R}^n$ deux points distincts éloignés d'une distance d, il suffit de considérer comme voisinage les boules ouvertes centrée en chacun de ces points, et de rayon $\frac{d}{2}$.

Définition 1.4 – *Soient E et F deux espaces topologiques. On dit qu'une bijection $f : E \to F$ est un* homéomorphisme *si pour tout ouvert A de E, $f(A)$ est un ouvert de F, et si pour tout ouvert B de F, $f^{-1}(B)$ est un ouvert de E.*

Les deux espaces sont alors dit homéomorphes.

Ces définitions permettent d'introduire la notion de n-variété, centrale dans la suite de ce manuscrit :

Définition 1.5 – *Un espace topologique séparé E est appelé n-*variété *si tout point p de E possède un voisinage homéomorphe à une partie ouverte de $\mathbb{R}^n$.*

*De plus, E est une n-*variété à bord *si tout point p de E possède un voisinage homéomorphe à une partie ouverte de $\mathbb{H}^n = \{(x_1, \ldots, x_n) \in \mathbb{R}^n; x_n \geq 0\}$.*

Par la suite, nous nous intéressons aux bords de ces n-variétés :

Définition 1.6 – *Soit E une n-variété à bord. L'ensemble des points dont le voisinage est homéomorphe à un ouvert de $\mathbb{R}^n$ est appelé l'*intérieur *et est noté $\mathring{E}$. Le complémentaire dans E de cet ensemble est* le bord *de cette n-variété à bord, noté ∂E.*

De manière moins classique en topologie, on dira par la suite qu'un sous-ensemble du bord de E est un bord *s'il s'agit d'une composante connexe du bord.*

Exemple : Soit $A = \{x \in \mathbb{R}^3; |x| \leq 1\} \cup \{x \in \mathbb{R}^3; |x - 4| \leq 1\}$. Le bord de A est composé de deux bords $A_1^* = \{x \in \mathbb{R}^3; |x| = 1\}$ et $A_2^* = \{x \in \mathbb{R}^3; |x - 4| = 1\}$.

Propriété 1.1 – *Si E est une n-variété, alors ∂E est une $(n-1)$-variété sans bord, chacune de ses composantes connexes étant un bord de E.*

FIG. 1.1 – *Le ruban de Möbius, surface non orientable.*

Dans la suite de ce manuscrit, nous nous intéressons plus particulièrement aux surfaces, qui sont définies comme suit :

Définition 1.7 – *On appelle* surfaces *les* 2*-variétés, et* surfaces à bord *les* 2*-variétés à bord.*

Les bords des surfaces sont constitués de composantes connexes topologiquement équivalentes à des cercles.

L'orientation des n-variétés peut être décrite en s'appuyant sur la définition d'orientation de $\mathbb{R}^n$:

Définition 1.8 – *L'orientation de l'espace vectoriel $\mathbb{R}^n$ est définie par le choix d'une base B de $\mathbb{R}^n$. Étant données deux bases B_1 et B_2 de $\mathbb{R}^n$, on dit qu'elles définissent la même orientation si le déterminant de la matrice de passage de B_1 à B_2 est positif. Si le déterminant est négatif, on dit que l'isomorphisme associé à la matrice de passage inverse l'orientation de $\mathbb{R}^n$.*

Définition 1.9 – *Étant donnés une n-variété E et un voisinage V_x d'un point $x \in E \setminus \partial E$, la donnée d'un homéomorphisme de V_x à l'espace vectoriel orienté $\mathbb{R}^n$ définit une* orientation *de V_x. On dit qu'une n-variété E est* orientable *si on peut étendre de manière continue cette orientation à tout E.*

Dans le cas d'une surface, cette définition est équivalente à l'intuition d'orientation, qui consiste à choisir de manière continue pour chaque point de la surface un sens de rotation autour de ce dernier.

Exemple : Le ruban de Möbius (figure 1.1) est un exemple classique de surface non-orientable.

Par la suite, sauf précision contraire, on s'intéressera aux surfaces orientables à bord.

1.2 Classification

Dans ce mémoire, nous nous intéressons aux propriétés topologiques des surfaces, et notamment à un critère d'équivalence topologique, permettant de caractériser l'homéomorphisme de deux espaces topologiques E et F. Si ces deux espaces sont homéomorphes, il suffit pour le prouver d'exhiber un homéomorphisme. Si ce n'est pas le cas, alors il est nécessaire d'avoir un critère d'ordre topologique les différenciant.

Par exemple, si les surfaces sont à bord, alors l'un des critères que l'on peut utiliser est le nombre de bords de la surface. Si E et F n'ont pas le même nombre de bords, alors ces deux surfaces ne sont pas homéomorphes. Ainsi, la sphère de rayon unité, sans bord, n'est pas homéomorphe au disque de rayon unité, qui lui possède un bord. Intuitivement, on constate qu'il est nécessaire de déchirer la sphère pour obtenir le disque. À l'inverse, si deux surfaces ont le même nombre de bords, alors on ne peut rien dire quant à leur équivalence topologique. Il est donc nécessaire de mettre en place des outils plus fins de classification.

1.2.1 Chemins et lacets

L'un des outils classiques pour étudier la structure des espaces topologiques est la notion de chemin sur une surface.

Définition 1.10 – *Un* chemin *sur un espace topologique E est défini par la donnée d'une application continue $\gamma : [0,1] \to E$. On appelle* origine *(resp.* extrémité*) du chemin le point $\gamma(0)$ (resp. $\gamma(1)$).*

Remarque : Par facilité, on dira que γ est un chemin qui relie a et b si $\gamma(0) = a$ et $\gamma(1) = b$.

Définition 1.11 – *On dit qu'un chemin γ sur E est un* lacet *basé en $x \in E$ si $\gamma(0) = \gamma(1) = x$. On appelle x le* point-base *de γ, et $\mathcal{L}(E,x)$ l'ensemble des lacets sur E basés en x.*

Définition 1.12 – *On dit qu'un lacet γ sur E est* simple *si pour tout $a, b \in [0,1]$ on a $\gamma(a) = \gamma(b) \Rightarrow a = b$ ou $a = 0$ et $b = 1$ ou $a = 1$ et $b = 0$.*

Définition 1.13 – *Soient γ_0 et γ_1 deux chemins sur E tels que $\gamma_1(0) = \gamma_0(1)$. On définit le* chemin composé *$\gamma = \gamma_0 \cdot \gamma_1$ par :*

$$\gamma(t) = \begin{cases} \gamma_0(2t) & \text{si } 0 \leq t \leq \frac{1}{2}, \\ \gamma_1(2t-1) & \text{si } \frac{1}{2} \leq t \leq 1. \end{cases}$$

Définition 1.14 – *Un espace topologique E est dit* connexe par arcs *si quels que soient $x, y \in E$, il existe un chemin γ tel que $\gamma(0) = x$ et $\gamma(1) = y$.*

Ces notions de chemin et de lacet sont valables pour tout espace topologique, et donc en particulier pour les surfaces.

Définition 1.15 – *Un lacet γ sur une surface S connexe par arcs est dit* séparant *si le complémentaire dans S de γ n'est pas connexe par arcs.*

1.2.2 Groupe fondamental

L'ensemble des lacets sur un espace topologique E dispose de propriétés qui sont directement liées à la topologie de E. On commence par définir une relation d'équivalence sur les lacets, amenant à la définition d'une structure de groupe.

Définition 1.16 – *On dit que deux chemins γ_0 et γ_1 sur E sont topologiquement équivalents, ou* homotopes*, et on notera $\gamma_0 \sim \gamma_1$ s'il existe une application continue $\delta : [0,1] \times [0,1] \to E$ telle que :*
- *$\delta(t,0) = \gamma_0(t)$, $\delta(t,1) = \gamma_1(t)$, pour $0 \leq t \leq 1$;*
- *$\delta(0,s) = \gamma_0(0) = \gamma_1(0)$, pour $0 \leq s \leq 1$;*
- *$\delta(1,s) = \gamma_0(1) = \gamma_1(1)$, pour $0 \leq s \leq 1$.*

Intuitivement, cela revient à dire que l'on peut déformer de manière continue sur l'espace topologique γ_0 en γ_1, à extrémités fixées.

En utilisant cette notion d'homotopie, on peut construire sur l'ensemble des lacets d'une surface donnée une structure algébrique, en introduisant les notions suivantes :

Proposition 1.1 – *Soit ϵ_x le* chemin constant *d'image $x \in E$. Pour tout chemin γ d'origine y et d'extrémité x, on a :*

$$\epsilon_x \cdot \gamma \sim \gamma, \qquad \gamma \cdot \epsilon_y \sim \gamma.$$

FIG. 1.2 – *Deux générateurs du groupe fondamental de la surface tore (l_1 et l_2).*

Dans la structure algébrique construite par cette relation d'homotopie, on généralise le chemin constant :

Définition 1.17 – *On appelle* lacet trivial *un lacet homotope au chemin constant.*

Remarque : Sur une surface S, tout lacet trivial simple non constant est un lacet séparant.

Proposition-définition 1.1 – *Soit γ un chemin sur E tel que $\gamma(0) = y$, $\gamma(1) = x$. Notons γ^{-1} le chemin défini par $\gamma^{-1}(t) = \gamma(1-t)$. On a alors :*

$$\gamma \cdot \gamma^{-1} \sim \epsilon_x, \qquad \gamma^{-1} \cdot \gamma \sim \epsilon_y.$$

On dit alors que γ^{-1} est le chemin inverse *de γ.*

Définition-propriété 1.1 – *Soit E un espace topologique, et x un point de E. On appelle* groupe fondamental *(ou groupe d'homotopie) de E en x, et on note $\Pi(E, x)$ le quotient de $\mathcal{L}(E, x)$ par la relation d'équivalence $\sim$.*

Preuve : $\Pi(E, x)$ est bien un groupe, si l'on considère la loi de composition de la définition 1.13, l'élément neutre ϵ_x tel que présenté par la proposition 1.1 et l'inverse défini à la proposition-définition 1.1. □

Théorème 1.1 – *Si E est connexe par arcs, alors $\Pi(E, x) \simeq \Pi(E, y)$, $\forall x, y \in E$.*

Remarque : $\Pi(E, x)$ ne dépendant pas du choix du point base, on parle du groupe fondamental $\Pi(E)$ de E, en omettant x.

Exemple : Le groupe fondamental de la surface tore T est complètement défini par la donnée de deux lacets non-homotopes l_1 et l_2 (figure 1.2). En effet, tout lacet l de T basé en x est homotope à une composition de l_1 et l_2 : $l = \prod_k l_{f(k)}$, où f est une fonction définie de $\mathbb{N}$ dans $\{0, 1\}$ et $\prod_k$ la généralisation de l'opérateur « $\cdot$ » à k lacets. On dit que l_1 et l_2 sont des *générateurs* du groupe fondamental de la surface de T.

On étend alors la notion d'homotopie pour deux lacets quelconques, en s'affranchissant ainsi du point base :

Définition 1.18 – *Soit E un espace topologique, et γ_0 et γ_1 deux lacets sur E. γ_0 et γ_1 sont* homotopes *s'il existe une application continue $\delta : [0, 1] \times [0, 1] \to S$ telle que :*
- $\delta(0, s) = \gamma_0(0) = \gamma_1(0)$, *pour* $0 \leq s \leq 1$;
- $\delta(1, s) = \gamma_0(1) = \gamma_1(1)$, *pour* $0 \leq s \leq 1$.

Remarque : Si E est une surface, alors $\delta([0, 1] \times [0, 1]) \subset E$ est un cylindre de bords γ_0 et γ_1.

(a) Une sphère, surface de genre zéro.

(b) Un cube, surface de genre zéro.

(c) Surface de genre trois.

FIG. 1.3 – *Surfaces sans bord.*

Proposition 1.2 – *Si les groupes fondamentaux de deux espaces topologiques E_1 et E_2 ne sont pas isomorphes, alors les espaces ne sont pas homéomorphes.*

Preuve : Par contraposée : si E_1 et E_2 sont homéomorphes, alors il existe un homéomorphisme $f : S_1 \to S_2$. L'application $\pi_f : \Pi(E_1, x) \to \Pi(E_1, f(x))$ est un homomorphisme de groupes, ainsi que $\pi_{f^{-1}} : \Pi(E_1, x) \to \Pi(E_1, f^{-1}(x))$. π_f est donc bien un isomorphisme. □

Ces différentes définitions étant valides sur des espaces topologiques, elles le sont naturellement sur les surfaces. La proposition 1.2 offre donc un critère de classification supplémentaire sur les surfaces.

1.2.3 Genre et bords

Un autre critère de classification, plus intuitif et couramment utilisé peut être décrit de la manière suivante :

Définition 1.19 – *Le* genre *d'une surface S orientable sans bord est le nombre de lacets non homotopes que l'on peut placer sur S sans qu'ils ne s'intersectent.*

Intuitivement, on dit que c'est le nombre de *poignées* de la surface.

Exemple : Les figures 1.3(a) et 1.3(b) présentent deux surfaces sans bord ayant un genre de zéro. La figure 1.3(c) représente une surface de genre trois, que l'on appelle par facilité le tore à trois poignées.

Définition 1.20 – *Le* genre *d'une surface S à bord est le genre de la surface S' construite par recollement d'un disque en chacun des bords de la surface.*

Exemple : Le genre d'un cylindre ouvert est zéro, car en recollant chacun de ses bords à un disque (figure 1.4), on obtient une surface homéomorphe à une sphère.

Le genre et le nombre de bords d'une surface S sont aussi des critères de classification efficaces des surfaces orientées.

FIG. 1.4 – *Calcul du genre d'un cylindre ouvert C composé de deux bords ∂_1 et ∂_2 par recollements successifs de disques. Le genre du cylindre est donc le même que celui de la boule, c'est-à-dire de zéro.*

1.3 Schéma polygonal

Une représentation classique de la topologie d'une surface consiste à décrire à l'aide d'un polygone une construction par recollements. Intuitivement, il s'agit de reproduire pour toutes les surfaces la construction classique du ruban de Möbius ou du cylindre ouvert, tel qu'illustré par la figure 1.5.

(a) Schéma polygonal du ruban de Möbius. (b) Schéma polygonal du cylindre.

FIG. 1.5 – *Schémas polygonaux associés à deux surfaces classiques. On construit les surfaces en recollant les arêtes notées a, en respectant l'orientation indiquée.*

On généralise cette construction en s'appuyant sur la notion de schéma polygonal :

Définition 1.21 – *Soit S une surface sans bord. Étant donnés un polygone P, une orientation et une identification de ses arêtes, on dit que P est un* schéma polygonal *de S si l'espace quotient obtenu par l'identification sur P de ses arêtes orientées est homéomorphe à S.*

Remarque : Outre la représentation graphique des schémas polygonaux (figures 1.5, 1.6), on utilise également la notation par mot cyclique. Ainsi, les schémas polygonaux de la figure 1.6 s'écrivent $aba^{-1}b^{-1}$ et $abca^{-1}c^{-1}b^{-1}$.

Nous décrirons dans la section 2.4 un formalisme complet englobant notamment la notion de schéma polygonal d'une surface.

(a) Schéma polygonal du tore (quadrilatère). (b) Schéma polygonal du tore (hexagone).

FIG. 1.6 – *Deux schémas polygonaux distincts associés à la même surface.*

FIG. 1.7 – *Schéma polygonal de la sphère.*

Tel que le schéma polygonal a été décrit ici, il n'est pas unique. Ainsi, la figure 1.6 propose deux schémas polygonaux non équivalents du tore. Il existe cependant pour chaque surface un schéma polygonal non réductible, ou schéma polygonal canonique :

Propriété 1.2 – *Étant donnée une surface S de genre $g \neq 0$ sans bord, on peut construire un schéma polygonal décrit par un polygone à $4g$ côtés. Ce polygone est minimal en le nombre de côtés, et on l'appelle le* schéma polygonal réduit.

Preuve : On construit le plongement des bords de ce polygone dans S en choisissant un point p, puis en construisant une base de lacets du groupe fondamental $\Pi(S, p)$. Ces lacets sont au nombre de $2g$, et sont l'image de deux arêtes du polygone ainsi construit. On sait que ce nombre de lacets est minimal, sinon le complémentaire ne serait pas homéomorphe à un disque. □

Remarque : Il n'y a pas unicité du schéma polygonal réduit. En effet, par découpage et recollement, on peut faire varier le mot décrivant le bord du polygone, tout en le gardant réduit. Par convention, on appelle *schéma polygonal canonique* un polygone dont le bord a un mot cyclique de la forme $a_1 b_1 a_1^{-1} b_1^{-1} \cdots a_g b_g a_g^{-1} b_g^{-1}$. On peut montrer qu'un schéma polygonal canonique est minimal, et que tout schéma polygonal minimal peut être transformé par découpage et recollement en le schéma polygonal canonique de la surface.

Exemple : Dans le cas d'une surface de genre zéro, le schéma polygonal minimal est un polygone à deux côtés (figure 1.7).

1.4 n-cets

Au début de ce premier chapitre, nous avons utilisé comme éléments de base aidant à la classification des surfaces les chemins et les lacets. Nous introduisons ici un nouvel élément, que nous appellerons n-cet, et que l'on peut voir comme une généralisation des lacets. Si l'utilisation que nous en ferons par la suite est originale (voir section 5.2), cette notion a déjà été décrite notamment dans le contexte de la théorie de l'homologie de flots [11].

Définition 1.22 – *Soit S une surface. On appelle n-*cet *basé en (b_0, b_1) sur S un ensemble de n chemins $(\gamma_i)_{1 \leq i \leq n}$ non constants tels que $b_0 = \gamma_1(0) = \cdots = \gamma_n(0)$ et $b_1 = \gamma_1(1) = \cdots = \gamma_n(1)$. Un 2-cet est donc un lacet, et on appellera* tricet *un 3-cet.*

FIG. 1.8 – *Tricet valide sur une surface de genre zéro à trois bords.*

Étant donnés deux points b_0 et b_1, il est possible de construire un grand nombre de n-cets, assimilables à des (b_0, b_1)-flots. Dans le cas des flots, les classes d'homologie permettent de décrire la structure de l'ensemble des flots. On s'appuie ici sur la notion d'homotopie pour obtenir le même type de caractérisation, en s'affranchissant des points base :

Définition 1.23 – *On dit que deux n-cets $(\gamma_i)_{1\leq i\leq n}$ et $(\alpha_i)_{1\leq i\leq n}$, respectivement basés en (b_0^γ, b_1^γ) et (b_0^α, b_1^α) sont* homotopes *s'il existe deux chemins c_0 et c_1 ainsi qu'une permutation σ de $[\![1, n]\!]$ vérifiant :*
- $c_0(0) = b_0^\gamma$, *et* $c_0(1) = b_0^\alpha$;
- $c_1(0) = b_1^\gamma$, *et* $c_1(1) = b_1^\alpha$;
- γ_i *et* $c_0 \cdot \alpha_{\sigma(i)} \cdot {c_1}^{-1}$ *soient homotopes, pour tout* $i, j \in [\![1, n]\!]$ *avec* $i \neq j$.

De la même manière qu'est définie la notion de lacet simple (section 1.2.1), nous propose ici la notion de n-cet simple :

Définition 1.24 – *On dit qu'un n-cet $\mathcal{C} = (\gamma_i)_{1\leq i\leq n}$ est* simple *si l'intersection de toute paire de chemins qui le composent est réduite aux seuls points base.*

On caractérise également la validité d'un n-cet en s'intéressant à la classe d'homotopie des chemins qui le composent :

Définition 1.25 – *On dit qu'un n-cet $\mathcal{C} = (\gamma_i)_{1\leq i\leq n}$ est topologiquement* valide *si aucune paire de chemins qui le composent ne sont homotopes.*

Exemple : La figure 1.8 présente un exemple de tricet valide sur une surface de genre zéro à trois bords.

Enfin, cette définition nous amène à la caractérisation d'une surface suivant l'arité maximale d'un n-cet valide sur cette surface :

Théorème 1.2 – *Soit S une surface connexe de genre g sans bord. Alors tout n-cet valide sur S est composé d'au plus $n = 4g$ chemins entre ses deux points base.*

On s'appuie sur le lemme suivant pour montrer le théorème :

Lemme 1.1 – *Soit S une surface connexe de genre g sans bord. Alors il existe un n-cet valide sur S dont le complémentaire est homéomorphe à un disque, et il est composé de $2g + 1$ chemins entre les deux points base.*

(a) Résultat du découpage d'une surface de genre g par un n-cet valide basé en (b_0, b_1) et composé de $2g-1$ chemins, suivant la construction décrite dans la preuve du lemme 1.1.

(b) Existence des deux chemins complémentaires découpant la surface en disque.

FIG. 1.9 – *Illustration de la preuve du lemme 1.1*

Preuve : On utilise une preuve par récurrence pour montrer l'existence d'un tel n-cet, et le nombre de chemins qui le composent.

Soit S une surface homéomorphe à une sphère, et b_0, b_1 deux points distincts de S. Soit c un chemin simple reliant b_0 à b_1. Alors le complémentaire de c est homéomorphe à un disque, et on a vérifié la propriété pour $g = 0$.

Si la surface S a un genre $g > 0$, alors on peut la voir comme une surface S' de genre $g - 1$ auquel on aurait ôté un petit disque, et que l'on aurait remplacé par une poignée (surface de genre un à un bord). Dans ce cas, on utilise l'hypothèse de récurrence pour affirmer que S' peut être découpée à l'aide d'un n-cet composé de $2g - 1$ chemins reliant b_0 à b_1. En considérant sur S le même n-cet, on obtient donc par découpage une surface de genre un à un bord. Le bord de ce découpage est constitué d'une alternance de b_0 et de b_1, comme illustré sur la figure 1.9(a). On voit alors qu'il est possible de construire deux chemins sur S, dans le complémentaire de ce n-cet, ayant la propriété d'être non homotopes, ne se croisant pas, et reliant chacun b_0 à b_1 (figure 1.9(b)). Le complémentaire de ce découpage est homéomorphe à un disque, et on constate qu'il est nécessaire d'avoir exactement deux chemins ayant ces propriétés pour obtenir une telle topologie.

On a donc la propriété voulue par récurrence pour tout g. La figure 1.10(a) illustre la propriété sur une surface de genre un. □

Preuve du théorème 1.2 : Montrons maintenant que le nombre maximal de chemins composant un n-cet valide sur une surface de genre g est de $4g$. En utilisant le lemme 1.1, on sait que le découpage suivant un n-cet valide dont le complémentaire est homéomorphe à un disque impose $2g + 1$ chemins. Le pavage correspondant est le plongement d'un polygone à $4g + 2$ côtés dans S.

Cherchons maintenant à introduire de nouveaux chemins complétant le n-cet déjà construit à ce stade en respectant le critère de validité. Si l'on regarde dans le polygone du pavage, dont le bord est une alternance de points b_0 et b_1, alors les nouveaux chemins doivent relier deux sommets du polygone non consécutifs et dont l'image est différente dans S. Si l'on construit un ensemble de chemins ayant cette propriété et ne s'intersectant pas entre eux, alors on peut vérifier qu'ils constituent un découpage du polygone en quadrilatères. Le polygone étant composé de $4g+2$ côtés, on peut construire des ensembles à $\frac{(4g+2)-4}{2} = 2g-1$ chemins. Au delà, chacune des composantes connexes du complémentaire étant des quadrilatères, il n'existe plus de chemin pour compléter le n-cet.

Ainsi, on peut construire un n-cet valide à $(2g + 1) + (2g - 1) = 4g$ chemins sur une surface de genre g. □

La figure 1.10(b) illustre l'existence d'un 4-cet valide sur une figure de genre un sans bord.

(a) 3-cet valide sur une surface de genre un, le complémentaire étant un disque.

(b) 4-cet valide sur une surface de genre un.

FIG. 1.10 – *n-cets valides sur un tore.*

Chapitre 2
Variétés discrétisées

Sommaire

2.1 Structures élémentaires 18
2.1.1 CW-complexes et complexes simpliciaux 18
2.1.2 Chaînes 20
2.1.3 Cycles et bords 21
2.2 Invariants topologiques 22
2.2.1 Groupes d'homologie 22
2.2.2 Nombres de Betti 23
2.2.3 Caractéristique d'Euler et calcul d'invariants 23
2.3 Mesures géométriques sur les surfaces 25
2.3.1 Approximation de distances géodésiques 25
2.3.2 Distances non euclidiennes 28
2.4 Pavage de surfaces 29
2.4.1 Notion de m-complexe cellulaire 30
2.4.2 Tuiles et pavages 31
2.4.3 Dualité pavage / découpage 33
2.4.4 Produit d'un découpage 35
2.5 Caractéristique d'Euler généralisée et conséquences 35
2.5.1 Généralisation de la caractéristique d'Euler 36
2.5.2 Conséquences immédiates 36
2.6 Conclusion 37

Au chapitre précédent, nous avons discuté de la question de la définition générale des surfaces, ainsi que de leur classification suivant des propriétés topologiques. Cependant, les surfaces que nous souhaitons manipuler sont également géométriquement variées, notamment lorsqu'elles sont issues d'acquisitions réelles.

Dans ce chapitre, nous présentons une modélisation de surfaces discrètes à l'aide de complexes simpliciaux, qui vient enrichir les définitions déjà données au chapitre précédent. Nous prolongeons ensuite les outils de caractérisation déjà présentés par des méthodes efficaces rendues possibles par les propriétés discrètes de ces structures. Tout comme indiqué au chapitre précédent, le lecteur intéressé pourra consulter [81, 50] qui détaillent plus avant les différentes notions présentées ici.

Nous nous intéressons également aux notions de distance qui peuvent être définies sur ces structures discrètes, afin d'introduire des notions de géométrie au cœur des méthodes jusqu'ici purement topologiques.

Enfin dans un dernier temps, nous introduisons un cadre théorique original permettant d'exprimer de manière très flexible la notion de pavage sur une surface, ou d'une manière duale la notion de découpage.

2.1 Structures élémentaires

S'il existe un grand nombre de manières de décrire analytiquement les surfaces qui offrent une grande concision de description (graphes de fonctions, B-splines, etc.), elles ne peuvent être générées directement depuis des données d'acquisition brutes. Leur caractère de description globale est bien sûr un atout, mais n'offre pas la flexibilité d'une représentation discrète.

Ainsi, l'une des représentations discrètes les plus couramment utilisées est la représentation par maillage triangulaire. L'idée est d'approximer une surface donnée par un ensemble de morceaux de plans. Cette représentation peut être facilement générée depuis des données issus d'acquisitions volumiques [72, 65], ou de nuages de points [23].

La formalisation de ces surfaces triangulées par les complexes simpliciaux permet de profiter ensuite des outils de manipulation et de classification issues de la topologie algébrique. C'est la raison pour laquelle nous avons choisi de la présenter ici.

2.1.1 CW-complexes et complexes simpliciaux

De manière abstraite, on définit une structure cellulaire de la manière suivante :

Définition 2.1 – *Soit E un espace topologique. On dit que E est une* cellule *de dimension n si elle est homéomorphe à la boule fermée de dimension n, notée $\mathbb{B}^n$. On dit qu'il s'agit d'une* cellule ouverte *quand elle est homéomorphe à l'intérieur $\mathring{\mathbb{B}}^n$ de $\mathbb{B}^n$.*

Définition 2.2 – *Un* CW-complexe *est un espace topologique séparé E et une collection*[1] *de cellules e disjointes tels que :*

- *Pour chaque cellule e de la collection, il existe $n \in \mathbb{N}$ et une application continue $f_e : \mathbb{B}^n \to E$ qui projette homéomorphiquement $\mathring{\mathbb{B}}^n$ sur e, et qui projette $\partial\mathbb{B}^n$ dans une union finie de cellules ouvertes de dimension inférieure à n.*
- *Un ensemble A est fermé dans E si pour toute cellule e, $A \cap \bar{e}$ est fermé dans $\bar{e}$.*

Il existe différents modèles plus concrets qui correspondent à cette définition. Parmi ceux-ci, nous proposons plus loin (section 2.4.1) les 2-m-cellules, qui serviront à définir la notion de pavage de surface.

D'une manière plus simple, on définit les complexes simpliciaux, qui vérifient les propriétés des CW-complexes, en commençant par définir la notion de n-simplexe :

Définition 2.3 – *Un n-*simplexe*, ou* simplexe de dimension *n est défini par l'enveloppe convexe d'un ensemble $\{v_0, ..., v_n\} \subset \mathbb{R}^N$ de points linéairement indépendants*[2]*. Par commodité, on appelle* sommet *un 0-simplexe,* arête *un 1-simplexe et* triangle *un 2-simplexe. On notera également $[v_0, ..., v_n]$ le simplexe décrit par les $n+1$ points indépendants $v_0, ..., v_n$.*

[1] Une *collection* est une famille d'objets qui peut contenir un ou plusieurs exemplaires d'un même objet.

[2] Ce qui implique $N \geq n$

(a) Complexe simplicial composé de deux 2-simplexes, de sept 1-simplexes et de huit 0-simplexes.

(b) Ensemble de simplexes ne constituant pas un complexe simplicial, car l'intersection de t_0 et t_1 n'est pas un simplexe. De plus, l'une des faces de a ne fait pas partie de la collection.

(c) Complexe simplicial de $\mathbb{R}^3$ qui n'est pas une 2-variété.

FIG. 2.1 – *Collections de simplexes.*

Remarque : Il est facile de montrer qu'un n-simplexe est homéomorphe à une boule de dimension n, on peut donc voir les simplexes comme des cellules.

On définit alors les faces, le bord et l'intérieur d'un n-simplexe :

Définition 2.4 – *Étant donné un n-simplexe s décrit par l'ensemble $\{v_0, ..., v_n\} \subset \mathbb{R}^N$, on dit qu'un sous-ensemble e de $\{v_0, ..., v_n\}$ définit une* face *de s de dimension n. On appelle* sommets *de s toutes les faces de s de dimension* 0.

Les faces d'un n-simplexe s de dimension strictement inférieure à n forment le bord, *noté ∂s.*

*L'*intérieur *d'un n-simplexe s est défini par la relation $\mathring{s} = s - \partial s$.*

La définition de complexe simplicial est alors immédiate :

Définition 2.5 – *Un* complexe simplicial *K dans $\mathbb{R}^N$ est une collection de simplexes de $\mathbb{R}^N$ telle que :*
- *chaque face d'un simplexe de K est dans K ;*
- *l'intersection de deux simplexes de K est une face de chacun d'eux.*

Un complexe simplicial possède donc une structure cohérente, et la suppression d'un simplexe le composant n'est pas toujours possible sans casser cette cohérence. On peut cependant définir la notion de sous-ensemble d'un complexe simplicial, qui décrit la cohérence d'une sous-partie d'un complexe simplicial :

Définition 2.6 – *Soit L un sous-ensemble d'un complexe simplicial K, qui contient toutes les faces de ses éléments. Alors L est un complexe simplicial, et est appelé* sous-complexe *de K.*

Exemple : La figure 2.1(a) illustre la notion de complexe simplicial, alors que la figure 2.1(b) présente une collection de simplexes qui ne vérifie pas les propriétés de complexe simplicial. On remarque, en observant la figure 2.1(c), qu'un complexe simplicial peut ne pas être une variété. Or, on s'intéresse ici aux surfaces, il est donc nécessaire de proposer une définition plus restrictive pour modéliser correctement les surfaces de manière discrète.

Définition 2.7 – *Un n-*maillage *$\mathcal{M}$ est un complexe simplicial ayant la propriété d'être une n-variété. Par la suite, on appellera également* surface discrète *les 2-maillages.*

Remarque : On déduit immédiatement de la définition ci-dessus que les simplexes composant un n-maillage sont nécessairement de dimension au plus n. Un 2-maillage est ainsi composé de sommets, arêtes et triangles, de la même manière que les définitions usuelles sans cadre simplicial.

2.1.2 Chaînes

Puisqu'une grande famille de complexes simpliciaux peuvent être vus comme des variétés, on se propose de spécialiser ici aux structures discrètes que sont les complexes simpliciaux les définitions de chemins et lacets présentées dans la section 1.2.1.

Pour cela, il est nécessaire d'introduire la notion d'orientation sur un simplexe, qui servira de brique élémentaire à l'équivalent de chemin orienté précédemment défini. Cette définition utilise la notion d'*ordre* sur un ensemble, qui correspond à la notion intuitive de classement des éléments d'un ensemble.

Définition 2.8 – *Soit s un simplexe. On dit que deux ordres de ses sommets sont équivalents s'ils ne diffèrent que d'une permutation paire. Si la dimension de s est supérieure à zéro, alors l'ensemble des ordres sur ses sommets peut être séparé en deux classes d'équivalence. Chacune de ces classes est appelée* orientation *de s.*

Cette notion d'orientation d'un simplexe permet de définir la notion de chaînes orientées :

Définition 2.9 – *Soit K un complexe simplicial. Une n-*chaîne *de K est une fonction c définie depuis l'ensemble des n-simplexes orientés de K sur $\mathbb{R}$, telle que $c(s) = -c(s')$ si s et s' sont deux orientations opposées d'un même simplexe.*

On définit naturellement une opération d'addition sur les n-chaînes en additionnant leurs valeurs. Le groupe résultant est noté $C_n(K)$, et est appelé groupe des n-chaînes *de K.*

On appelle également chaîne élémentaire *c associée à un simplexe s la fonction définie par $c(s) = -c(s') = -1$ (s' étant l'orientation inverse de s) et $c(\tau) = 0$ pour tout τ différent de s et s'. Par abus de notation, on écrira s pour désigner à la fois le simplexe orienté et sa chaîne élémentaire associée.*

Remarque : En observant que toute n-chaîne est composée d'un ensemble fini de chaînes élémentaires, on peut facilement faire le lien entre une 1-chaîne et un chemin. En effet, on peut voir toute 1-chaîne comme un ensemble fini de chemins. Ce nombre de chemins peut être inférieur au nombre de chaînes élémentaires composant la 1-chaîne, si l'on procède par concaténation des chemins associés à chaque 1-chaîne, si celle-ci est possible. Par la suite, on appellera *chaîne simple* une 1-chaîne qui peut être couverte par un chemin unique.

Exemple : la figure 2.2 présente un exemple de 1-chaîne composée de deux chaînes simples.

FIG. 2.2 – *1-chaîne sur un complexe simplicial.* $\{c_0, c_1, c_2\}$ *est une* 1*-chaîne composée de 3 chaînes élémentaires, ou encore de deux chaînes simples (*$\{c_0\}$ *et* $\{c_1, c_2\}$*).*

2.1.3 Cycles et bords

Une manière usuelle d'étendre la notion de bord d'un espace topologique à toute n-chaîne est donné par la définition suivante :

Définition 2.10 – *Étant donnés un complexe simplicial* K *et un entier* $n > 0$*, on appelle* opérateur de bord *et on note* ∂_n *l'homéomorphisme*

$$\partial_n : C_n(K) \to C_{n-1}(K)$$

défini pour un simplexe orienté $s = [v_0, ..., v_n]$ *par*

$$\partial_n s = \partial_n [v_0, ..., v_n] = \sum_{i=0}^{p} (-1)^i [v_0, ..., v_{i-1}, v_{i+1}, ..., v_n] .$$

Remarque : En fixant $C_n(K)$ comme le groupe trivial pour $n < 0$, il est immédiat que ∂_n est l'homéomorphisme trivial pour $n \leq 0$.

Exemple : La figure 2.3 illustre sur différentes chaînes l'application d'un opérateur de bord. On remarque notamment que si deux triangles adjacents sont correctement orientés, alors le bord de leur réunion ne contient pas l'arête qu'ils ont en commun.

FIG. 2.3 – *Application de l'opérateur de bord à différentes chaînes.*

Définition 2.11 – *Le noyau* Ker ∂_n *est appelé le groupe des* n-cycles *de* K *et noté* $Z_n(K)$*. L'image* Im ∂_{n+1} *est appelée le groupe des* n-bords *de* K *et est notée* $B_n(K)$*.*

Remarque : Ces deux groupes, définis formellement, correspondent bien à l'intuition que l'on a des cycles et des bords sur une telle structure (et sont le pendant de la notion de lacet et de bord sur les espaces topologiques). La définition de la notion de bord est immédiate. Pour le noyau de ∂_n, il suffit de constater qu'il est constitué de l'ensemble des n-chaînes de $C_n(K)$ qui ont un bord vide. Or c'est précisément la définition d'un cycle que d'être une p-chaîne sans bord.

On peut également remarquer qu'un bord est un cycle, c'est-à-dire que $B_n(K) \subset Z_n(K)$, ou encore $\partial_{n-1}\partial_n = 0$.

(a) c_0 et c_1 sont deux représentants de la même classe dans $H_1(K)$ car il existe une 2-chaîne s qui a pour bord $-c_0$ et c_1.

(b) c_0 et c_1 sont les représentants de la deux classes distinctes de $H_1(K)$ car il n'existe pas de 2-chaîne qui ait pour bord $-c_0$ et c_1.

FIG. 2.4 – *1-cycles sur une surface K (complexe simplicial) et illustration de $H_1(K)$.*

2.2 Invariants topologiques

La question de la classification des variétés que nous avons exploré dans le chapitre précédent peut être également retranscrite dans le contexte de la modélisation discrète par complexes simpliciaux.

Ainsi, les bords d'un complexe simplicial – dans le sens proposé au chapitre précédent – sont facilement déterminés par les outils que nous avons introduit ici. Il suffit pour cela de considérer le bord de l'ensemble des simplexes de taille maximale. Il est constitué d'un certain nombre de composantes connexes, correspondant aux bords de la variété.

De la même manière, la structure des complexes simpliciaux est le support d'outils – les groupes d'homologie – qui permettent une classification fine, à la manière du groupe fondamental.

2.2.1 Groupes d'homologie

On définit formellement les groupes d'homologie sur un complexe simplicial :

Définition 2.12 – *Étant donné un complexe simplicial K dans $\mathbb{R}^N$, on définit pour $0 \leq n \leq N$ le n-ième* groupe d'homologie *de K par :*

$$H_n(K) = Z_n(K)/B_n(K). \tag{2.1}$$

Nous nous intéressons dans ce travail aux surfaces. Dans ce contexte, les $H_n(K)$ avec $n > 2$ ne présentent pas d'information intéressante, car les complexes simpliciaux de type surface ne contiennent pas de simplexes de dimension 2 ou plus. Par contre, il est possible de donner un sens assez simple sur les surfaces aux trois premiers groupes d'homologie, qui s'étend assez bien pour les dimensions supérieures.

Ainsi, $H_0(K)$ est le groupe des ensembles des sommets de K, quotientés par le groupe des ensembles de bords de chaînes. Dit autrement, deux sommets de K ont une image équivalente dans $H_0(K)$ s'il existe une chaîne élémentaire qui les relie. $H_0(K)$ contient donc un élément par composante connexe dans K.

De la même manière, $H_1(K)$ est le groupe des ensembles de 1-cycles quotienté par le groupe des ensembles des bords de 2-chaînes. Sur les surfaces, il s'agit de l'ensemble des chemins par l'ensemble des bords des sous-parties de la surface. La figure 2.4(a) montre comment deux 1-cycles peuvent être équivalents, car cohérents comme bords d'une 2-chaîne. À l'inverse, les deux 1-cycles de la figure 2.4(b) ne sont pas équivalents, car aucune 2-chaîne correspondante n'existe.

Il est assez immédiat de constater que sur une telle surface, deux cycles ont le même représentant dans $H_1(K)$ s'ils sont homotopes. On a donc une correspondance assez fine entre $H_1(K)$ et le groupe fondamental $\Pi(S)$ de la surface S associée à K. On peut d'ailleurs montrer de manière générale la proposition suivante :

Proposition 2.1 – *Soit K un complexe simplicial, et E l'espace topologique qui lui est associé. Alors si l'on identifie les lacets et les 1-cycles, $H_1(E)$ est l'abélianisé*[3] *de $\Pi(E)$.*

On trouvera dans [50, théorème 2A.1 page 166] une preuve de cette proposition.

$H_1(K)$ offre de bonnes possibilités de calcul, notamment grâce à la représentation discrète de complexe simplicial. Ainsi, Peltier *et al.* [87] ont proposé et implémenté un algorithme efficace de calcul de groupes d'homologie, en utilisant une normalisation de la matrice d'incidence.

Enfin, $H_2(K)$ décrit le nombre de cavités qui sont créées par la surface. Ainsi, si la surface est composée d'une seule composante connexe, il existe deux situations distinctes :
- la surface est sans bord : dans ce cas, on peut définir un intérieur et un extérieur. La surface est donc le bord de la cavité qu'elle renferme, et $H_2(K)$ contient une classe d'équivalence correspondant à ce bord de volume.
- la surface a au moins un bord : dans ce cas, on ne peut pas définir d'intérieur. $H_2(K)$ est donc vide.

2.2.2 Nombres de Betti

Si l'on cherche à classifier les surfaces, c'est-à-dire à les différencier selon leur topologie, les trois premiers groupes d'homologie offrent un outil puissant et pratique. Dans le cas des surfaces orientables, on peut même réduire cette information de classification au rang[4] des groupes d'homologie. Historiquement, ce sont d'ailleurs ces valeurs appelées nombres de Betti qui ont été utilisées pour classifier les surfaces :

Définition 2.13 – *Étant donné un complexe simplicial K, le n-ième nombre de Betti $b_n(K)$ de K est défini comme le rang du n-ième groupe d'homologie $H_n(K)$ de K.*

Pour les surfaces, le nombre de Betti le plus intéressant est b_1, qui peut être interprété de manière intuitive comme *le nombre maximum de découpages que l'on peut faire dans la surface sans la séparer en deux composantes connexes.* La définition précise de découpage est donnée à la section 2.4.3. On peut voir ici un découpage comme un cycle sur la surface de l'objet, ou un chemin reliant deux bords. Par la suite, sauf indication inverse, on appellera b_1 *le nombre de Betti* d'un complexe K.

Dans le cas de variétés non orientables, les nombres de Betti ne décrivent pas pleinement la topologie. Il est alors nécessaire d'introduire également la notion de nombre de torsions ([81]). Ainsi, la bouteille de Klein (figure 2.5(a)) et le tore (figure 2.5(b)) ont un nombre de Betti égal à deux bien qu'ils ne soient pas homéomorphes. On peut également faire la même remarque au sujet du ruban de Möbius (figure 1.1) et du cylindre ouvert (figure 1.4) qui ont 1 pour nombre de Betti, le découpage correspondant traversant la surface à partir d'un bord.

2.2.3 Caractéristique d'Euler et calcul d'invariants

Si les nombres de Betti sont dans la pratique calculables [87], cette approche est dans les faits assez coûteuse. Historiquement, une autre manière de classifier les surfaces, fondée sur la caractéristique d'Euler, a précédé les nombres de Betti :

[3] L'abélianisé d'un groupe G est le plus grand quotient abélien de G.

[4] Le rang d'un groupe est le cardinal du plus grand ensemble d'éléments linéairement indépendants du groupe.

(a) La bouteille de Klein (à gauche plongée dans $\mathbb{R}^3$), dont le schéma polygonal (voir section 1.3) est $aba^{-1}b$, a un nombre de Betti égal à deux (voir découpages à droite).

(b) Le tore a un nombre de Betti égal à deux.

FIG. 2.5 – *Deux surfaces de même nombre de Betti mais non homéomorphes.*

Définition 2.14 – *Étant donné un 2-maillage $\mathcal{M}$, la* caractéristique d'Euler, *notée χ, est définie par :*

$$\chi(\mathcal{M}) = S(\mathcal{M}) - A(\mathcal{M}) + F(\mathcal{M}), \tag{2.2}$$

où $S(\mathcal{M})$ est le nombre de sommets de $\mathcal{M}$, $A(\mathcal{M})$ est le nombre d'arêtes de $\mathcal{M}$ et $F(\mathcal{M})$ est le nombre de faces de $\mathcal{M}$.

Cette définition initiale peut être élargie, et le lien avec les nombres de Betti vient assez rapidement :

Théorème 2.1 – *Étant donné un complexe simplicial K, la formule (2.2) se généralise en $\chi(K) = \sum(-1)^i S_i(K)$, où $S_i(K)$ est le nombre de i-simplexes de K. On montre alors que la caractéristique d'Euler est égale à la somme alternée des nombres de Betti du complexe simplicial :*

$$\chi(K) = \sum(-1)^i b_i(K). \tag{2.3}$$

On trouvera dans [50, théorème 2.44 page 146] une preuve algébrique de ce théorème.

L'avantage principal de la caractéristique d'Euler est la grande facilité avec laquelle elle peut être calculée sur une surface décrite par un complexe simplicial, en utilisant (2.2). Cela entraîne plusieurs résultats calculables avec une excellente efficacité :

Proposition 2.2 – *Soit $\mathcal{M}$ un 2-maillage à une seule composante connexe à b bords. Alors :*

$$b_1(\mathcal{M}) = 1 - \chi + \mathbb{1}_{b=0}. \tag{2.4}$$

Preuve : Immédiate en s'appuyant sur l'équation 2.3. □

D'un point de vue implémentation, suivant les choix qui ont été faits, la complexité de ce calcul peut même être constante. On trouvera décrits dans la section A.2.1 les choix d'implémentations de nos travaux, qui impliquent un temps linéaire en le nombre de sommets du maillage (le nombre d'arêtes et le nombre de points bord n'étant pas décrits par la structure).

Le genre est lui aussi rapidement calculable à partir de la caractéristique d'Euler :

Proposition 2.3 – *Soit $\mathcal{M}$ un 2-maillage à une seule composante connexe. Alors son genre $g(\mathcal{M})$ est donné par :*

$$g(\mathcal{M}) = \frac{2 - \chi - b}{2}, \tag{2.5}$$

où b est le nombre de bords de la surface décrite par $\mathcal{M}$.

Preuve : Si la surface décrite par $\mathcal{M}$ est sans bord, alors le nombre de Betti est égal à deux fois le genre de $\mathcal{M}$. On a donc immédiatement (2.5) en utilisant (2.3).

Si la surface $\mathcal{M}$ comporte b bords ($b > 0$), alors on construit K' en fermant chacun des bords de $\mathcal{M}$ par l'ajout au complexe simplicial d'un sommet, d'autant d'arêtes que nécessaire pour relier ce nouveau point aux points du bord, et d'autant de triangles que nécessaire pour fermer la surface. On a donc ajouté b sommets, et autant de triangles que d'arêtes. La caractéristique d'Euler $\chi(\mathcal{M}')$ de $\mathcal{M}'$ est donc égale à $\chi(\mathcal{M}) + b$. Or d'après la définition 1.20, le genre de $\mathcal{M}$ est égal au genre de $\mathcal{M}'$. On en déduit (2.5) en utilisant le premier point de la preuve. □

2.3 Mesures géométriques sur les surfaces

Dans les sections suivantes, nous présentons la notion de pavage (section 2.4.1) ainsi que la dualité entre pavage et découpage (section 2.4.3). Dans ces deux cas, un outil essentiel pour décrire ces notions est le chemin, tel qu'il a été donné par la définition 1.10. Dès lors que l'on veut comparer les chemins entre eux, la notion de longueur d'un chemin est indispensable.

Dans le cas où la surface (variété de dimension 2) est plongée dans $\mathbb{R}^3$, une manière classique de définir la longueur d'un chemin est de s'appuyer sur la distance euclidienne de $\mathbb{R}^3$:

Définition 2.15 – *Étant donné un chemin $\gamma : [0,1] \to S$ sur une sous-variété S de dimension deux de $\mathbb{R}^3$, on définit la* longueur *de γ par :*

$$L(\gamma) = \int_0^1 \|\gamma'(t)\|_3 \, \mathrm{d}t, \tag{2.6}$$

où $\|\cdot\|_3$ est la norme euclidienne de $\mathbb{R}^3$.

On peut naturellement généraliser cette définition en utilisant une norme différente.

Ainsi, si l'on dispose d'une mesure de longueur des chemins sur une surface (peu importe la manière dont elle a été définie), la notion de géodésique est immédiate :

Définition 2.16 – *Étant données une surface S et une distance L sur les chemins de S, on appelle* géodésique *entre $a \in S$ et $b \in S$ tout chemin reliant a à b de longueur minimale. On appellera* distance géodésique *la longueur de ces chemins.*

Intuitivement, la distance géodésique correspond donc à la plus courte distance entre a et b en suivant la surface, le chemin correspondant à cette plus courte distance n'étant pas forcément unique.

Exemple : La surface C représentée sur la figure 2.6 est un cylindre ouvert (de bords ∂_1 et ∂_2). Les points a et b, appartenant à C, sont placés de manière à être diamétralement opposés. Alors il existe deux géodésiques γ_1 et γ_2 entre a et b, et la distance géodésique entre a et b, correspondant à la longueur de ces deux géodésiques, est de $r\pi$ si r est le rayon du cylindre.

2.3.1 Approximation de distances géodésiques

Si la définition de distance géodésique décrite précédemment est exacte d'un point de vue mathématique, elle n'est pas calculable efficacement. Cependant, si l'on travaille avec des surfaces discrètes (ou 2-maillages), une manière simple de réaliser l'approximation des géodésiques est de s'appuyer sur la structure maillée que portent ces surfaces :

FIG. 2.6 – *Les deux géodésiques reliant a et b positionnés de chaque côté d'un cylindre ouvert.*

Définition 2.17 – *Étant donnés un 2-maillage $\mathcal{M}$, et une fonction longueur $l : A(\mathcal{M}) \to \mathbb{R}^+$ définie sur l'ensemble $A(\mathcal{M})$ des arêtes de $\mathcal{M}$, on appelle* géodésique discrète *entre deux sommets a et b de $\mathcal{M}$ toute chaîne élémentaire ayant pour bord $a - b$. On appelle* distance géodésique discrète *la longueur de ces chaînes.*

Remarque : Comme pour la définition de distance géodésique précédente, la chaîne élémentaire correspondant à la distance géodésique discrète n'est pas forcément unique.

L'un des algorithmes les plus classiques pour déterminer concrètement et efficacement les distances géodésiques discrètes est l'algorithme de Dijkstra [29] (Algorithme 2.1). Cet algorithme calcule à partir d'un ensemble de sommets source X la distance géodésique discrète à ces sommets pour chacun des autres sommets du maillage, et construit également une géodésique par sommet vers le plus proche sommet source[5].

[5]Les géodésiques peuvent alors être reconstruites en utilisant le tableau `pred[]` construit par l'algorithme 2.1 pour remonter depuis un sommet jusqu'à un sommet source.

Algorithme 2.1 : Algorithme de Dijkstra

Données : $\mathcal{M}$ un 2-maillage, $l : A(\mathcal{M}) \to \mathbb{R}^+$ une longueur sur les arêtes, et X un ensemble de sommets de $\mathcal{M}$

Résultat :
- `distance[ ]` contenant pour chaque sommet de $\mathcal{M}$ la distance à X
- `pred[ ]` contenant pour chaque sommet de $\mathcal{M}$ son prédécesseur vers X

```
début Initialisation
    pour chaque s ∈ M faire
        pred[s] ← indéfini;
        si s ∈ X alors
            distance[s] ← 0;
        sinon
            distance[s] ← +∞;
        fin
    fin
fin
O ← S(M);
tant que O ≠ ∅ faire
    Soit o le sommet de O le plus prêt de la source;
    O ← O − {o};
    pour chaque v voisin de o faire
        si distance[v] > distance[o] + l(v − o) alors
            distance[v] ← distance[o] + l(v − o);
            pred[v] ← o;
        fin
    fin
fin
```

Différentes variations sont envisageables si l'on traite un problème spécifique. Ainsi, si l'on souhaite juste connaître la distance entre deux sommets a et b d'un 2-maillage, il suffit d'arrêter l'exécution de la boucle principale de l'algorithme dès que l'on a retiré le point cible recherché de l'ensemble $\mathcal{O}$ des ouverts. De la même manière, si l'on cherche à connaître tous les sommets à une distance au plus d d'un sommet a, on pourra arrêter l'exécution de la boucle dès que l'on a retiré de $\mathcal{O}$ un sommet à une distance supérieure à d de a.

On montre facilement que la complexité de l'algorithme 2.1, dans sa version sans optimisation, est en $o\left(S(\mathcal{M}) \log S(\mathcal{M})\right)$. S'il existe des méthodes moins coûteuses pour calculer les distances minimales sur des graphes planaires, par exemple utilisant des méthodes de flots [62], il n'y a pas à notre connaissance de résultat améliorant cette complexité pour des graphes surfaciques. On peut cependant citer [88] qui propose une méthode heuristique de recherche de plus court chemin. Cependant, elle nécessite l'utilisation de points repères ou de distances pré-calculées, ce qui limite son utilisation lorsque l'on n'a pas de connaissances *a priori* sur la surface traitée.

Dans la suite de ce travail, nous avons essentiellement utilisé l'algorithme de Dijkstra, dans les trois variantes présentées ci-dessus. Cependant, il existe une autre famille de méthodes, qui répond à une question différente, et peut avoir son intérêt dans un contexte de surfaces discrétisées. Il s'agit de méthodes de calcul de distances exactes sur maillages, telles que celle proposée par [77] étendue par [98] à une version approchée rapide à erreur

(a) Une surface de genre 0 à trois bord, et trois chemins de découpage l_1, l_2 et l'_2.

(b) Une surface de genre 0 à un bord (issu du découpage d'un tore suivant un chemin non droit), et trois chemins de découpage l_1, l_2 et l'_2.

FIG. 2.7 – *Surfaces portant des chemins de découpages.*

bornée. Les auteurs y construisent des intervalles d'arêtes pour propager les distances, à la manière d'un algorithme de Dijkstra continu. L'article initial annonce une complexité en $o\left(S(\mathcal{M})^2 \log S(\mathcal{M})\right)$, que le second démontre être une estimation pessimiste de l'efficacité de l'algorithme.

2.3.2 Distances non euclidiennes

L'avantage principal de l'algorithme 2.1 est que la distance géodésique calculée ne dépend que du choix de la fonction $l : A(\mathcal{M}) \to \mathbb{R}^+$ de longueur des arêtes. Bien sûr, l'un des choix les plus immédiats est de choisir pour cette distance la distance euclidienne $\|\cdot\|_3$ de $\mathbb{R}^3$, puisque le graphe est plongé dans un espace à trois dimensions. Cependant, on peut facilement faire varier cette mesure pour y intégrer d'autres informations.

Ainsi, nous présentons plus loin une modification de cette distance permettant de prendre en compte la géométrie locale des surfaces (section 4.1.1), ou encore une modification guidée par l'utilisateur dans le contexte de l'infographie (section 6.1.3). Cette modification de la distance est également utile pour répondre efficacement à des questions algorithmiques, comme le calcul du plus court découpage, ou la préservation de propriétés des formes issues du découpage, comme nous le décrivons ici.

Chemins pour découpages

Si nous présentons plus loin (section 2.4.3) la notion de découpage d'un 2-maillage suivant une chaîne, sa compréhension intuitive est immédiate. Ainsi, la figure 2.7(a) présente une surface de genre 0 à trois bords, et portant trois chemins reliant chacun des bords différents. D'un point de vue intuitif, on aurait envie de regrouper l_2 et l'_2 en une même chaîne, et présenter l_1 et $l_2 + l'_2$ comme deux découpages possibles reliant ∂_1 et ∂_2. Dans ce cadre, on aura d'ailleurs envie de privilégier $l_2 + l'_2$ qui nécessite de « donner moins de coups de ciseaux », puisque leur longueur cumulée est plus petite que celle de l_1. Or, il existe un moyen simple de modifier la longueur des arêtes $l : A(\mathcal{M}) \to \mathbb{R}^+$ pour que l'algorithme 2.1 ait ce comportement. En effet, si l'on attribue à chaque arête qui appartient au bord de la surface une longueur de 0, et que l'on garde pour les autres

arêtes leur longueur euclidienne, alors le plus court chemin entre ∂_1 et ∂_2 sera un chemin empruntant l_2, puis la moitié de ∂_3, et enfin l'_2.

Le choix d'une telle mesure se généralise bien à toute surface à bord, et permet de produire des chemins de découpage efficaces, tenant compte des propriétés de bords des surfaces.

Cohérence du complémentaire

À l'inverse, on peut vouloir interdire le passage d'un plus court chemin par un bord. Ainsi, la figure 2.7(b) présente une surface de genre 0 à un bord, issue du découpage d'un cylindre ouvert suivant un chemin non droit. Elle porte trois chemins de découpage. Si l'on souhaite construire un chemin qui relie les points bords du haut aux points bords du bas, il existe au moins deux familles de découpages : la première est représentée par l_1, et sépare la surface en deux nouvelles surfaces. L'autre est $l_2 + l'_2$, qui divise la surface en trois nouvelles surfaces.

Nous verrons au chapitre 5 que les propriétés des surfaces produites par un découpage sont intéressantes à étudier. Dans ce contexte, on voudra favoriser le découpage suivant le chemin l_1, pour éviter de créer de petites surfaces aux propriétés de bord peu intéressantes, comme celles produites par $l_2+l'_2$. Pour s'assurer que l'algorithme 2.1 ait ce comportement, on ajuste alors la longueur des arêtes $l : A(\mathcal{M}) \to \mathbb{R}^+$ en fixant la longueur des arêtes du bord à une longueur supérieure à tous les plus courts chemins de la surface (on pourra dire $+\infty$). Alors les chaînes produites par ce découpage emprunteront un chemin du type de l_1, en évitant d'emprunter les bords.

2.4 Pavage de surfaces

Le cœur des travaux présentés dans ce mémoire porte sur la réalisation de pavages sur des surfaces. Intuitivement, on peut voir le pavage d'une surface comme le recouvrement de celle-ci par un ensemble de surfaces élémentaires, appelées *tuiles*, dont l'intersection est vide.

Dans cette section, nous proposons un modèle original formalisant ce concept. Il permet de décrire le pavage de toute surface par des tuiles de topologie quelconque. Ce formalisme permet de décrire non seulement des pavages classiques à l'aide de tuiles de genre zéro, mais également des pavages par cylindres (section 5.2), ou encore toutes les étapes intermédiaires des algorithmes que nous présenterons dans la partie II. Ce formalisme permet également de décrire les propriétés de forme des tuiles, telles qu'explorées dans la section 5.4.

Il existe dans la littérature différentes approches visant à décrire la structuration d'une variété d'un point de vue topologique, voire géométrique. La structure la plus classique est celle de complexe simplicial (définition 2.3). Si elle permet de décrire toute surface à l'aide de boules ouvertes élémentaires, et offre quelques possibilité de calcul (proposition 2.2), son manque de structure ne permet pas d'envisager de choses très fines sans ajouter de descriptions supplémentaires.

Les *cartes combinatoires*, initialement proposées par Jacques [56] et généralisées en g-cartes par Lienhardt [71] sont un modèle plus riche de description de variétés. Elles s'appuient sur des composants élémentaires appelés *brins* (ou demi-arêtes) et sur un ensemble d'opérateurs (une permutation et des involutions) sur les brins, et permettent la description d'objets géométriques complexes, orientables ou non, de dimension quelconque. Outre les possibilités descriptives qu'offre ce formalisme, il permet de calculer

FIG. 2.8 – 2-*m-cellules.*

efficacement différentes propriétés des objets géométriques, tels que le schéma polygonal canonique [24] ou des générateurs des groupes d'homotopie [25]. Cependant, les g-cartes ne permettent pas la flexibilité descriptive et structurante qu'offrent le formalisme que nous présentons ici : une face de g-carte est nécessairement homéomorphe à un disque.

2.4.1 Notion de m-complexe cellulaire

Afin de formaliser à la fois la structure de chaque cellule, et la structure globale de l'ensemble des cellules, on définit le concept de m-cellules et de m-complexes cellulaires[6], qui vérifient les propriétés de CW-complexes.

Définition 2.18 – *Un* m-complexe cellulaire C *(de dimension inférieure ou égale à 2) est un ensemble de m-cellules respectant les propriétés suivantes :*

- *Les* 2*-m-cellules sont des* 2*-variétés compactes connexes à bord, les* 1*-m-cellules sont des courbes simples, fermées ou ouvertes, et les* 0*-m-cellules sont des points.*
- *Le bord de chaque* i*-m-cellule est couvert par un* $(i-1)$*-sous-complexe, avec* $i \in \{1, 2\}$*.*
- *l'intersection de deux m-cellules est un sous-complexe de* C*.*

Chacun des bords d'une 2-variété étant homéomorphe à un cercle, une conséquence directe de la définition précédente est que chaque bord d'une 2-m-cellule est composé d'une séquence alternée de 1-m-cellules et de 0-m-cellules.

Exemple : La figure 2.8 présente quelques exemples de 2-m-cellules. Ainsi, les figures 2.8(a) et 2.8(b) représentent deux 2-m-cellules de genre zéro, à un bord. Cependant, la structure de leur bord est différente. La première est constituée d'une unique 1-m-celule, alors que la seconde est décrite par deux 1-m-cellules et deux 0-m-cellules. La figure 2.8(c) présente quant à elle une 2-m-cellule de genre zéro, dont deux bords sont composés chacun d'une unique 1-m-cellule, et d'un dernier bord ayant la structure du bord de la 2-m-cellule de la figure 2.8(b). Enfin, la figure 2.8(d) présente une 2-m-cellule de genre un, et comportant un unique bord, là aussi structuré par deux 1-m-cellules et deux 0-m-cellules.

De la même manière que nous avons défini la notion de faces et d'intérieur pour les n-simplexes (définition 2.4), on donne les définitions suivantes sur les m-cellules :

Définition 2.19 – *On appelle respectivement* arêtes *et* sommets *d'une* 2*-m-cellules chacune des* 1*-m-cellules et* 0*-m-cellules composant son bord. De manière générale, on appelle les* faces *d'une* i*-m-cellule les* $(i-1)$*-m-cellules constituant son bord, avec* $i \in \{1, 2\}$*.*

*On appelle l'*intérieur *d'une* i*-m-cellule cette* i*-m-cellule diminuée de ses* $(i-1)$*-m-cellules.*

[6]Nous avons utilisé ici le préfixe « m- » pour mettre en avant les propriétés de variétés des cellules, en utilisant l'initiale du mot anglais *manifold*.

FIG. 2.9 – *M-tuile sur une surface $\mathcal{M}$ (de genre deux sans bord) décrite par une 2-m-cellule de genre zéro à trois bords et par un plongement F.*

2.4.2 Tuiles et pavages

La structure de m-complexe cellulaire dispose des propriétés nécessaires à la description que nous voulons donner d'un pavage, à la fois d'un point de vue topologique, mais aussi d'un point de vue structurel.

On commence donc par définir la notion de tuile, qui s'appuie sur les m-complexes cellulaires, et qui est la composante élémentaire des M-pavages.

Définition 2.20 – *Soit $\mathcal{F}$ l'ensemble des plongements $F : C \to \mathcal{M}$ d'une 2-m-cellule C dans un 2-maillage $\mathcal{M}$ et vérifiant les propriétés suivantes :*
- *F est continue.*
- *Soit s un sommet de C. Alors $F(s)$ est un 0-simplexe de $\mathcal{M}$.*
- *Soit a une arête de C. Alors $F(a)$ est un ensemble de 1-simplexes de $\mathcal{M}$.*
- *Soit $\mathring{C}$ l'intérieur de C. Alors $F(\mathring{C})$ est un ensemble de 2-simplexes de $\mathcal{M}$.*
- *F restreint à $\mathring{C}$ est injectif.*
- *Soient a_0 et a_1 deux arêtes de C telles que $F(\mathring{a}_0) \cap F(\mathring{a}_1) \neq \emptyset$. Alors $F(a_0) = F(a_1)$.*

$F, F' \in \mathcal{F}$ sont dits équivalents si et seulement si pour chaque m-cellule de C, $F(X) = F(X')$. On appelle les classes d'équivalence définies par cette relation des M-tuiles. *Pour des raisons de simplicité, une M-tuile sera notée par un de ses représentants $(C, \mathcal{M}, F)$.*

Remarque : La propriété d'injectivité n'est imposée que sur la restriction de F à $\mathring{C}$. Cela implique que deux arêtes de C peuvent avoir une même image sur $\mathcal{M}$. Cependant, la dernière propriété implique que l'on ne peut avoir une superposition partielle par F de deux arêtes de C.

Une M-tuile sur un 2-maillage $\mathcal{M}$ est complètement définie par la donnée de C et de l'image de chacune des m-cellules composant C.

Exemple : La figure 2.9 est une représentation graphique d'une tuile sur une surface $\mathcal{M}$ de genre deux sans bord. La 2-m-cellule C_0 constituant la tuile est de genre zéro à trois bords, et le plongement F projette deux des bords de C_0 sur le même ensemble d'arêtes de $\mathcal{M}$.

La notion de M-tuile entraîne naturellement celle de M-pavage :

Définition 2.21 – *Un* M-pavage *d'un 2-maillage $\mathcal{M}$ par un m-complexe cellulaire C est une classe d'équivalence des plongements $F : C \to \mathcal{M}$ tels que :*
- *Pour toute 2-m-cellule c de C, $F_{|c}$ est une M-tuile.*
- *F est surjective.*
- *La restriction de F à l'union des m-cellules qui ne sont pas des bords est injective.*
- *Soient a_0 et a_1 deux arêtes de C telles que $F(\mathring{a}_0) \cap F(\mathring{a}_1) \neq \emptyset$. Alors $F(a_0) = F(a_1)$.*

FIG. 2.10 – *M-pavage sur une surface $\mathcal{M}$ (de genre deux sans bord) décrite par un m-complexe cellulaire composé de 2-m-cellules, et par un plongement F.*

FIG. 2.11 – *Représentation d'un schéma polygonal par M-pavage.*

Deux plongements F et F' sont dits équivalents si et seulement si pour chaque m-cellule X de C, on a $F(X) = F(X')$.

Remarque : Comme dans le cas des M-tuiles, un M-pavage sur un 2-maillage $\mathcal{M}$ est complètement défini par la donnée d'un m-complexe cellulaire C et par l'image de chacune des m-cellules de C.

Exemple : La figure 2.10 est la représentation graphique d'un M-pavage sur une surface $\mathcal{M}$. Le m-complexe cellulaire $C = C_0 \cup C_1$ est composé de deux 2-m-cellules, l'une de genre un, l'autre de genre zéro.

La notion de schéma polygonal peut être réinterprétée à l'aide du formalisme de M-pavage. La M-tuile unique constituant le M-pavage est décrite par le polygone, l'identification des arêtes étant assurée par la correspondance des images lors du plongement, et l'orientation des arêtes par la propriété de continuité du plongement. Ainsi, la figure 2.11 décrit le M-pavage d'un tore par un quadrilatère, équivalent au schéma polygonal décrit par la figure 1.6(a).

Comme nous l'avons illustré à la figure 1.6, le schéma polygonal sur une surface donnée n'est pas unique. De la même manière, on peut définir parmi les M-pavages des familles d'équivalence :

Définition 2.22 – *On dit que deux M-pavages $(C, \mathcal{M}, F)$ et $(C', \mathcal{M}, F')$ d'un 2-maillage S sont* topologiquement équivalents *si l'image sur S de l'intérieur des 2-m-cellules de C et C' sont identiques, et si l'ensemble des 1-m-cellules de C est homéomorphe à l'ensemble des 1-m-cellules de C'.*

Intuitivement, cela revient à dire que deux M-pavages $(C, \mathcal{M}, F)$ et $(C', \mathcal{M}, F')$ sont identiques si l'on peut transformer C en C' en supprimant ou en ajoutant des sommets sur les arêtes, en adaptant le plongement à cette nouvelle structure sans modifier l'image par ce plongement de l'intérieur des 2-m-cellules. La figure 2.12 présente un M-pavage topologiquement équivalent à celui présenté par la figure 2.10.

Proposition 2.4 – *Étant donné un M-pavage $(C, \mathcal{M}, F)$ d'un 2-maillage S, il existe un unique M-pavage $(C_0, \mathcal{M}, F_0)$ tel que $(C_0, \mathcal{M}, F_0)$ soit topologiquement équivalent à $(C, \mathcal{M}, F)$ et dont le nombre de 0-m-cellules soit minimal. On appelle un tel M-pavage le* M-pavage élémentaire *associé à $(C, \mathcal{M}, F)$.*

FIG. 2.12 – *M-pavage sur une surface $\mathcal{M}$ topologiquement équivalent à celui décrit par la figure 2.10.*

FIG. 2.13 – *Deux ensembles de 1-m-cellules non homéomorphes.*

Preuve : Étant donné un M-pavage $(C, \mathcal{M}, F)$ d'un 2-maillage $\mathcal{M}$, les 0-m-cellules que l'on peut retirer de C sont celles qui participent exactement à deux 1-m-cellules. En effet, si une 0-m-cellule participe à une seule 1-m-cellule, alors la suppression du sommet imposera la suppression de l'arête (figure 2.13). D'autre part, la suppression d'un sommet participant à plus de deux arêtes imposera un changement de topologie. Retirer un sommet participant exactement à deux arêtes entraîne la fusion de ces deux arêtes. La construction itérative par suppression de sommet étant unique, et puisque l'ordre de suppression des sommets importe peu, alors il existe un unique M-pavage au nombre de 0-m-cellules minimum, ces dernières correspondant à tous les sommets participant à un nombre d'arêtes différent de deux. □

Définition 2.23 – *On dit qu'un M-pavage est* élémentaire *s'il est lui-même son M-pavage élémentaire associé.*

2.4.3 Dualité pavage / découpage

Dans la suite, nous parlerons indifféremment de M-pavage et de découpage. En effet, définir un découpage sur une surface est équivalent à définir un pavage élémentaire sur cette surface. Pour cela, commençons par définir ce qu'est un découpage.

Définition 2.24 – *Soit $\mathcal{M}$ un 2-maillage. On appelle* découpage *$\mathcal{C}$ sur $\mathcal{M}$ un ensemble de chaînes simples et de cycles sur $\mathcal{M}$.*

La définition de découpage donnée ici n'impose pas que deux chaînes simples soient disjointes. Elles peuvent donc partager deux arêtes. Pour décrire plus précisément les découpages, on introduit la notion de découpage élémentaire :

Définition 2.25 – *Un découpage $\mathcal{C}$ sur un 2-maillage $\mathcal{M}$ est dit* élémentaire *si l'intersection des chaînes simples qui le composent est constitué uniquement des bords des chaînes, et si chacun des cycles n'intersecte ni une chaîne de $\mathcal{C}$, ni un autre cycle de $\mathcal{C}$.*

On appelle découpage élémentaire associé *à un découpage $\mathcal{C}$ sur un 2-maillage $\mathcal{M}$ tout découpage élémentaire $\mathcal{C}'$ tel que l'ensemble des arêtes couvertes par $\mathcal{C}'$ soit égal à l'ensemble des arêtes couvertes par $\mathcal{C}$.*

La définition de découpage élémentaire associé peut être complétée par une la proposition d'unicité suivante :

Proposition 2.5 – *Étant donné un découpage $\mathcal{C}$ sur un 2-maillage $\mathcal{M}$, il existe un unique découpage élémentaire associé à $\mathcal{C}$.*

Preuve : Soit $\mathcal{C}$ un découpage sur un 2-maillage $\mathcal{M}$. On montre facilement qu'il existe un découpage élémentaire associé à $\mathcal{C}$ en le construisant de la manière suivante : soit $A(\mathcal{C})$ (respectivement $S(\mathcal{C})$) l'ensemble des arêtes (respectivement sommets) couvertes par $\mathcal{C}$. Si $A(\mathcal{C})$ est vide, alors $S(\mathcal{C})$ constitue un ensemble de chemins constants élémentaires, et on a construit le découpage élémentaire associé. Sinon, soit $S_2(\mathcal{C}) \subset S(\mathcal{C})$ l'ensemble des sommets couverts par $\mathcal{C}$ qui participent exactement à deux arêtes de $A(\mathcal{C})$. Si $S(\mathcal{C})/S_2(\mathcal{C})$ est vide, alors $\mathcal{C}$ peut être vu comme un ensemble de cycles disjoints, et on construit facilement le découpage élémentaire associé. Sinon, on construit un ensemble de chaînes élémentaires reliant chacun des éléments de $S(\mathcal{C})/S_2(\mathcal{C})$, empruntant des arêtes de $A(\mathcal{C})$, et ne traversant que des sommets contenus dans $S_2(\mathcal{C})$. Les arêtes non couvertes restantes peuvent être structurées en cycles, et on a ainsi construit un découpage élémentaire associé à $\mathcal{C}$.

Montrons maintenant l'unicité de ce découpage élémentaire associé à $\mathcal{C}$. Supposons qu'il existe deux découpages élémentaires différents $\mathcal{C}_0$ et $\mathcal{C}_1$ associés à $\mathcal{C}$. Alors $S(\mathcal{C}_0)/S_2(\mathcal{C}_0)$ et $S(\mathcal{C}_1)/S_2(\mathcal{C}_1)$ sont différents, car sinon $\mathcal{C}_0$ et $\mathcal{C}_1$ seraient égaux. Mais $S(\mathcal{C})/S_2(\mathcal{C}) = S(\mathcal{C}_0)/S_2(\mathcal{C}_0)$ et $S(\mathcal{C})/S_2(\mathcal{C}) = S(\mathcal{C}_1)/S_2(\mathcal{C}_1)$, car ce sont des découpages élémentaires associés à $\mathcal{C}$. On a donc une contradiction, ce qui entraîne l'unicité. □

On définit ensuite la notion d'équivalence entre un découpage et un M-pavage :

Définition 2.26 – *Étant donné un 2-maillage $\mathcal{M}$, un découpage $\mathcal{C}$ sur $\mathcal{M}$ et un M-pavage $(C, \mathcal{M}, F)$ sont dits équivalents si le complémentaire dans $\mathcal{M}$ de $\mathcal{C}$ est égal à l'image par F de l'intérieur des 2-m-cellules de C.*

Énonçons maintenant le théorème d'unicité de l'équivalence entre découpage et M-pavage :

Théorème 2.2 – *Étant donnés un 2-maillage $\mathcal{M}$ et un découpage $\mathcal{C}$ sur $\mathcal{M}$, il existe un unique M-pavage élémentaire $(C, \mathcal{M}, F)$ équivalent à $\mathcal{C}$.*

Réciproquement, étant donnés un 2-maillage $\mathcal{M}$ et un M-pavage $(C, \mathcal{M}, F)$, il existe un unique découpage élémentaire $\mathcal{C}$ sur $\mathcal{M}$ équivalent à $(C, \mathcal{M}, F)$.

La preuve de ce théorème est immédiate en s'appuyant sur le lemme suivant :

Lemme 2.1 – *La donnée d'un découpage élémentaire $\mathcal{C}$ sur $\mathcal{M}$ est équivalente à la donnée d'un M-pavage élémentaire $(C, \mathcal{M}, F)$.*

Preuve : La synthèse des deux structures lorsque l'on choisit la représentation élémentaire est clairement équivalente. En effet, les chaînes simples et cycles d'un découpage élémentaire $\mathcal{C}$ sont les images par F des 1-m-cellules de C du M-pavage élémentaire $(C, \mathcal{M}, F)$ correspondant, les bords des chaînes simples étant les images par F des 0-m-cellules de C. Enfin, la topologie des composantes connexes du complémentaire des chaînes simples et cycles de $\mathcal{C}$ est celle des 2-m-cellules de C. □

Dans la suite de ces travaux, nous profitons de cette dualité pour utiliser indifféremment la notion de pavage ou de découpage, l'une pouvant être traduite en l'autre.

2.4.4 Produit d'un découpage

On s'intéressera également dans la suite de ce travail au produit d'un découpage $\mathcal{C}$ sur une surface discrète S. Pour des raisons de cohérence, par exemple parce que l'on veut pouvoir appliquer successivement plusieurs découpages, le produit du découpage d'une surface doit être une surface. Par la suite, on notera $\between(S, \mathcal{C})$ la surface résultant du découpage $\mathcal{C}$ sur la surface S.

Si $\mathcal{C}$ est composé de n lacets et chemins, on définit alors $\between(S, \mathcal{C})$ comme le résultat des n découpages élémentaires correspondants. Intuitivement, un découpage élémentaire suivant un chemin c (qui peut être un lacet) est produit en dupliquant chacune des arêtes par lesquelles passent les chemins de $\mathcal{C}$, et en dupliquant autant de fois que nécessaire les sommets qui sont contenus dans c, puis en corrigeant les bords des triangles de façon à préserver la structure de complexe simplicial.

Il est important de noter ici que si une arête composant c appartient au bord de S, alors elle ne sera pas dupliquée. En effet, puisque le produit d'un découpage est une surface discrète, chaque voisinage d'un sommet doit être homéomorphique à un disque ou un demi-disque. Un découpage ne peut donc générer d'arêtes non adjacentes à un triangle. En introduisant cette contrainte, il devient immédiat que la surface résultant d'un découpage $\mathcal{C}$ est égal au résultat du découpage élémentaire associé à $\mathcal{C}$.

L'une des manières de quantifier la qualité d'un découpage est de considérer la longueur de ce dernier. Plusieurs choix sont possibles pour définir la longueur d'un découpage. On aurait pu définir la longueur d'un découpage comme la somme des longueurs des lacets et chemins qui le composent. Cependant, ce choix impliquerait que deux découpages $\mathcal{C}$ et $\mathcal{C}'$ de longueur différente puissent produire des surfaces identiques. On choisit donc de définir la longueur d'un découpage uniquement en considérant les arêtes dupliquées pendant l'étape de découpage :

Définition 2.27 – *Étant donnés une surface S et un découpage $\mathcal{C}$, on définit la* longueur *de $\mathcal{C}$ comme la somme des longueurs des arêtes non bord qui sont couvertes par un chemin ou un lacet de $\mathcal{C}$.*

D'un point de vue pratique, on pourra donc calculer les chemins et lacets les plus courts en modifiant la longueur des arêtes bord par une affectation à 0, les algorithmes de recherche de plus court chemin travaillant directement avec des longueurs satisfaisant la définition de longueur d'un découpage donnée ci-dessus.

2.5 Caractéristique d'Euler généralisée et conséquences

La définition 2.14 s'appuie sur la structure de complexe simplicial des 2-maillages pour décrire une relation (la caractéristique d'Euler) entre le nombre de i-faces du complexe et le genre de la surface. L'équation 2.5 généralise cette formule aux surfaces à bord, toujours décrites par des complexes simpliciaux. La définition des 2-simplexes qui structurent ces complexes impose que ces faces de dimension 2 soient des triangles. En prenant une définition plus large, il est classique de généraliser la caractéristique d'Euler aux surfaces décrites par des faces polygonales, quel que soit leur nombre de côté. Nous présentons ici une généralisation de cette formule aux surfaces décrites par des m-complexes cellulaires.

2.5.1 Généralisation de la caractéristique d'Euler

Théorème 2.3 – *Soit C un m-complexe cellulaire décrivant une surface connexe de genre g à b bords, et composé de S 0-m-cellules, de A 1-m-cellules et de F 2-m-cellules. On note A_b le nombre de 1-m-cellules avec bord, et F_b le nombre de 2-m-cellules avec bord. Alors, si pour chaque 2-m-cellule f_i de C, on note g_i son genre et b_i son nombre de bords, on a la* caractéristique d'Euler généralisée *:*

$$S - A_b + 2F_b = 2 - 2\left(g - \sum_{f_i} g_i\right) - \left(b - \sum_{f_i} b_i\right). \tag{2.7}$$

Preuve : Commençons par montrer l'équation pour $A = A_b$ et $F = F_b$, c'est-à-dire dans le cas où toutes les 1-m-cellules de C sont non cycliques, et lorsque les 2-m-cellules ont toutes au moins un bord.

Pour cela, on montre d'abord que pour transformer une 2-m-cellule f_i de genre g_i à $b_i > 0$ bords en une 2-m-cellule de genre zéro à un bord, il est nécessaire de lui ajouter $2g_i + b_i - 1$ 1-m-cellules. Le nombre de bords de f_i étant non nul, il existe au moins une 0-m-cellule sur le bord de f_i. Nous avons déjà vu (propriété 1.3) que sur une surface de genre g_i, le découpage suivant $2g_i$ lacets basés en un point réduisait le genre à zéro. Cela revient donc à introduire $2g_i$ 1-m-cellules sur f_i (sans introduire aucune 0-m-cellule). Chacun des b_i bords qui constitue la 2-m-cellule résultante contient au moins une 0-m-cellule. Il suffit d'ajouter $b_i - 1$ 0-m-cellules pour réduire le nombre de bords de cette 2-m-cellule à un.

Si on procède de cette manière pour chacune des 2-m-cellules de C, alors on a introduit $\left(\sum_{f_i} 2g_i + b_i\right) - F$ 1-m-cellules sur C sans changer le nombre de 0-m-cellules ni le nombre de 1-m-cellules. De plus, chacune des 2-m-cellules résultantes sont des polygones, aussi la caractéristique d'Euler classique est vérifiée :

$$S - \left(A + \left(\sum_{f_i} 2g_i + b_i\right) - F\right) + F = 2 - 2g - b.$$

On en déduit immédiatement l'équation 2.7, avec $A = A_b$ et $F = F_b$.

Si maintenant $A \neq A_b$, alors on constate qu'il est nécessaire d'ajouter à chacune des $A - A_b$ 1-m-cellules non cycliques une 0-m-cellule pour appliquer ensuite la construction précédemment décrite. On en déduit là aussi immédiatement l'équation 2.7, avec $F = F_b$.

Enfin, si $F \neq F_b$, il existe des 2-m-cellules sans bord, c'est-à-dire qui décrivent des composantes connexes séparées de C. Il est facile de voir que si l'on retire une telle 2-m-cellule f_i de C, alors on réduit g de g_i, sans modifier le nombre de bords du complexe simplicial, sans non plus modifier le nombre de 1-m-cellules et de 0-m-cellules. On a donc l'équation 2.7 dans sa forme générale. □

2.5.2 Conséquences immédiates

De 2.7, on peut déduire quelques conséquences simples sur les pavages de surfaces par des tuiles non triviales.

Ainsi, si l'on décrit une surface avec des 2-m-cellules ayant toutes le même genre g_f et le même nombre b_f de bords, on a :

$$S - A_b + 2F = 2 - 2\left(g - Fg_f\right) - \left(b - Fb_f\right). \tag{2.8}$$

On remarque ainsi que si les 2-m-cellules sont des cylindres, la formule devient $S - A_b = 2 - 2g - b$: le nombre de cellules n'est pas contraint par la topologie de la surface à paver. Dit autrement, on peut ajouter autant de cylindres (par subdivision d'un cylindre en deux) sans changer la topologie de la surface complète. À l'inverse, pour toutes les topologies plus complexes ($b_f > 2$ ou $g_f > 0$), F n'est pas éliminé de la caractéristique d'Euler généralisée : le nombre de tuiles est contraint par la topologie de la surface.

Par exemple, si l'on souhaite décrire une surface de genre g à b bords à l'aide d'un m-complexe cellulaire composé de pantalons (surfaces de genre zéro à trois bords, voir [49]), et que l'on impose à toutes les 1-m-cellules d'être non circulaires, alors l'équation 2.8 s'écrit :

$$S - A - F = 2 - 2g - b$$

Si de plus on impose que les jonctions entre pantalons se fassent le long de 1-complexes cellulaires homéomorphes à un cercle, alors $S = A$ et on obtient que le nombre de pantalons qui décrivent une telle surface est de $2g + b - 2$. On retrouve le résultat proposé par Hatcher [49] lorsqu'il avait introduit la notion de pantalon.

2.6 Conclusion

Nous avons présenté dans ce chapitre les structures élémentaires classiques de description de surfaces discrètes, ainsi que les outils permettant de les manipuler. Nous avons ensuite présenté les méthodes de classification topologique classiques permises par ces structures. Nous avons ensuite montré qu'il était possible d'introduire au cœur de ces méthodes topologiques des informations géométriques, notamment en exploitant les possibilités offertes par les mesures de distances sur les surfaces discrètes.

Nous avons ensuite introduit le formalisme des M-pavages, permettant de décrire de manière fine les notions de pavage explorées dans ce mémoire, en soulignant la dualité de cette notion avec l'approche plus classique de découpage. Enfin, en nous appuyant sur les possibilités de description en tuiles non triviales de ce formalisme, nous avons montré qu'il était possible de généraliser la caractéristique d'Euler à tous les pavages.

Deuxième partie

Pavage sous contraintes

Chapitre 3
Contraintes topologiques

Sommaire

3.1 Découpages élémentaires 42
3.1.1 Découpage suivant un 1-cycle 42
3.1.2 Découpage suivant chemins et n-cets 45
3.2 Découpage en disque 47
3.2.1 Description du problème et motivations 47
3.2.2 Algorithme linéaire sans contrainte 48
3.2.3 Combinatoire et longueur de découpage 49
3.3 Découpage minimal 49
3.3.1 Définition et NP-difficulté 50
3.3.2 Découpage par lacets non séparants 50
3.3.3 Optimisations de l'algorithme 52
3.3.4 Algorithme approché 55
3.3.5 Discussion autour des variations 55
3.4 Conclusion 57

Nous avons défini au chapitre précédent le concept de M-pavage de surfaces, qui permet de décrire de manière précise et complète le produit d'un découpage sur une surface donnée. Dans ce chapitre, nous utilisons ce cadre général pour décrire différents algorithmes de découpage répondant à des contraintes topologiques sur les surfaces produites par le découpage.

Dans ce contexte, l'intérêt de l'utilisation de la structure de M-pavage est immédiat. En effet, cette structure permet de capturer les informations topologiques sur les surfaces produites par le découpage, en envisageant chaque M-tuile du découpage comme une abstraction des propriétés de ces surfaces (le genre et le nombre de bords par exemple).

Nous commençons d'abord par nous intéresser aux conséquences topologiques du découpage d'une surface suivant des structures élémentaires – lacets, chemins, n-cets – puis nous posons le problème du découpage en une surface homéomorphe à un disque. Nous proposons ensuite une variante de algorithme d'Erickson et Har-Peled [34] calculant une approximation du découpage minimal, et nous mettons en exergue les améliorations apportées, qui d'un point de vue pratique augmentent nettement les temps d'exécution de l'algorithme initial.

3.1 Découpages élémentaires

Intéressons-nous tout d'abord aux surfaces issues du découpage d'une surface suivant les structures élémentaires de 1-cycles, de chemins et de n-cets. Si dans le cas général on ne peut rien dire sur la géométrie, on peut décrire topologiquement les conséquences de ces découpages.

3.1.1 Découpage suivant un 1-cycle

Théorème 3.1 – *Soit S une surface discrète (à une composante connexe) de genre $g \neq 0$ et à n bords, et γ un 1-cycle non séparant sur S. Alors $S' = ✂(S, \gamma)$ est composée d'une seule composante connexe, son genre est compris entre 0 et $g-1$, et le nombre de bords de S' est compris entre 1 et $n+g$.*

Preuve : Soit γ un 1-cycle non séparant sur S. Par définition, le genre de $S' = ✂(S, \gamma)$ est au plus de $g-1$, car sinon γ ne serait pas un 1-cycle non séparant. Montrons d'abord que les bornes inférieures du genre et du nombre de bords de la proposition peuvent être atteintes. On utilise pour cela le lemme suivant :

Lemme 3.1 – *Soit γ un lacet non séparant sur une surface S de genre g à au moins un bord. Alors il existe un lacet simple dont le complémentaire dans S est une surface de genre 0 à 1 bord.*

Preuve : Montrons d'abord ce résultat sur une surface S de genre g à un bord B. Pour cela, on commence par identifier tous les points du bord de S en un point unique noté b. La surface $\check{S}$ associée à cette identification est donc une surface sans bord de genre g. On peut construire à partir de b un ensemble de lacets simples formant une base du groupe fondamental $\Pi(S, b)$. Ces lacets sont au nombre de $2g$, et tous basés en b. On sait que cet ensemble de lacets forme une réalisation du découpage de la surface suivant un schéma polygonal réduit. Le complémentaire de ces lacets est une surface de genre zéro à un bord. Si l'on reporte maintenant ces lacets sur la surface originale S, on obtient un ensemble de $2g$ chemins reliant deux points de B (que l'on pourra distribuer régulièrement). Le complémentaire de ces chemins est toujours de genre zéro à un bord.

Montrons maintenant que l'on peut construire un lacet simple couvrant tout ces chemins et une partie de B, si l'on a pris soin de bien construire l'ensemble de lacets. En effet, si l'on voit la surface $\check{S}$ comme composée d'un ensemble de g surfaces élémentaires de genre un à un bord, alors il est immédiat de voir que l'ensemble des lacets peut être structuré en g groupes de 2 lacets (figure 3.1(a)) correctement ordonnés autour de b (figure 3.1(b)). Sur S, les chemins sont donc également regroupés deux par deux le long de B (figure 3.1(c)). On peut alors, pour chaque paire de ces chemins, construire une section du lacet final (figure 3.1(d)) qui ne se croisera pas. On a ainsi construit sur une surface de genre g à un bord un lacet non séparant dont le complémentaire a un bord et est de genre zéro. Dans le cas où le nombre de bords est strictement plus grand, il suffit de choisir un bord particulier, et de faire la même construction que décrite ci-dessus, en imposant que tous les autres bords soient traversés par exactement un lacet basé en b. □

Montrons maintenant que le 1-cycle ne peut introduire plus de g bords :

Lemme 3.2 – *Soit γ un lacet non séparant sur une surface S de genre g à $n > 0$ bords. Alors le complémentaire de γ dans S a au plus $n+g$ bords.*

(a) Extrait de la surface et de deux lacets représentant une partie du schéma polygonal canonique.

(b) Exemple du regroupement des lacets deux par deux sur une surface de genre 3.

(c) Illustration des chemins correspondants sur la surface avec bord.

(d) Partie du lacet final correspondant aux deux chemins connectés au bord.

FIG. 3.1 – *Illustration de la preuve du lemme 3.1.*

Preuve : Montrons que le complémentaire d'un lacet γ non séparant sur une surface de genre g à $n > 0$ bords ne peut avoir plus de $n + g$ bords. Nous l'avons vu dans la démonstration du lemme précédent, autoriser un lacet à passer par les bords d'une surface revient à considérer le découpage par un ensemble de lacets sur une surface $\check{S}$, ces derniers étant ensuite recollés au niveau d'un ensemble de points correspondant aux bords de la surface initiale. On sait que le nombre de lacets non séparants et disjoints sur une surface donnée de genre g est au plus égal à g. Ces lacets introduisent donc $2g$ bords dans la surface. Si on les recolle ensuite pour qu'ils ne forment plus qu'une seule composante connexe, chaque recollement élémentaire de deux lacets en un point va réduire le nombre de bords de 1. Pour que les g lacets soient recollés, on doit procéder à $g-1$ rapprochements élémentaires, ce qui réduit le nombre de bords associés du complémentaire dans $\check{S}$ à $g+1$. Or, on procède à un tel recollement sur au moins un point de $\check{S}$ correspondant à un bord de la surface initiale. Le découpage de la surface par γ introduit donc au plus g bords sur S. □

Les deux lemmes prouvés ici décrivent le complémentaire d'une surface S par un lacet non séparant. Il est facile de transposer ce résultat au découpage d'une surface discrète par un 1-cycle non séparant, au besoin en raffinant la surface discrète pour interdire le croisement du 1-cycle. □

En utilisant la construction décrite dans le dernier lemme, on peut facilement décrire des découpages diminuant le genre de la surface

Proposition 3.1 – *Soit γ un lacet non séparant sur une surface S de genre g à au moins un bord, et $g' \leq g$ un entier. Alors il existe un lacet simple qui découpe S en une surface de genre $g - g'$ sans modifier le nombre de bords.*

Preuve : Il suffit pour cela de calculer, comme dans la preuve du lemme 3.1, un ensemble de $2g$ chemins reliant différents points d'un bord b de S, mais ne passant pas par les autres bords, puis de n'en retenir que $2g'$ en les sélectionnant par paires cohérentes suivant les sous-mots du schéma polygonal canonique associé. Le découpage n'augmente pas le nombre de bords, car il ne fait qu'agrandir le bord b. Il est évident qu'il réduit le genre de g'. □

Dans la suite, nous nous sommes intéressé au découpage d'une surface le long du plus court suivant un lacet non séparant. Dans ce cas, un lacet ne peut passer qu'une seule fois par chaque bord, car sinon on pourrait construire un lacet non séparant produisant un découpage plus court. On regarde donc les découpages respectant cette contrainte :

Théorème 3.2 – *Soit S une surface discrète (à une composante connexe) de genre $g \neq 0$ et à n bords, et γ un 1-cycle non séparant sur S passant au plus une fois par chacun des bords de S. Alors $S' = ✂(S, \gamma)$ est composée d'une seule composante connexe, son genre est égal à $g - 1$, et le nombre de bords de S' est compris entre 2 et $n + 2$.*

Preuve : Par définition de γ, $S' = ✂(S, \gamma)$ est constitué d'une seule composante connexe. Étudions maintenant la topologie de S'. Le lacet équivalent à γ dans $\check{S}$ (la surface produite par identification de tous les points d'un même bord de S) est un lacet simple qui passe au plus une fois par chacun des points issus des bords. Son complémentaire est donc une surface de genre $g - 1$ à deux bords. On en déduit naturellement la topologie de S' : cette surface est aussi de genre $g - 1$, et son nombre de bords est de $2 + m$, où m est le nombre de bords de S par lesquels γ ne passe pas. Puisque $0 \leq m \leq n$, on a le résultat.

□

Dans le cas continu, on peut toujours trouver sur une surface S à n bords de genre $g > 0$ un lacet séparant tel que le nombre de bords de S' soit égal à $n + 2$. Cependant, ce n'est pas le cas pour les 1-cycles non séparants sur des surfaces discrètes :

Proposition 3.2 – *Étant donnée une surface discrète S (à une composante connexe) de genre $g > 0$ et à n bords, il n'existe pas assurément un 1-cycle non séparant γ passant au plus une fois par chacun des bords de S tel que le découpage de S suivant γ produise une surface à $n + 2$ bords.*

Preuve : Il suffit pour cela de construire une surface qui n'a pas cette propriété. Étant donnée une surface S de genre g, on construit une nouvelle surface S' de la manière suivante. L'ensemble des sommets de S' est composé des sommets de S, auxquels on ajoute, pour chaque triangle de S, trois sommets de S' (par exemple en réalisant une homothétie centrée sur l'isobarycentre du triangle, et de rapport $\frac{1}{2}$). On a donc ajouté autour de chaque arête non bord de S quatre points (deux de chaque côté de l'arête), que l'on peut faire correspondre deux à deux suivant leur position le long de l'arête. On construit alors les triangles de S' en ajoutant deux triangles pour former un quadrilatère pour chacun de ces ensembles de quatre points, et en ajoutant au niveau de chaque extrémité des arêtes de S un triangle reliant le sommet original aux deux nouveaux points les plus proches. La figure 3.2 illustre une telle construction.

FIG. 3.2 – *Le maillage de gauche décrit une surface S homéomorphe à un disque. La surface de droite S' a été construite à partir de S de façon à ce que tout chemin de plus de deux arêtes croise le bord.*

La surface ainsi construite est de genre g, et a la particularité que tout chemin composé de plus de deux arêtes croise le bord. Il n'existe donc pas sur S' (de genre g) de 1-cycle non séparant qui ne croise pas le bord. Chaque 1-cycle non séparant de S' passe au moins par un bord, ce qui provoque une réduction (stricte ou non) du nombre de bords de la surface produite par le découpage suivant ce 1-cycle. □

3.1.2 Découpage suivant chemins et n-cets

Nous avons décrit assez finement à la section précédente les conséquences topologiques que pouvait produire le découpage d'une surface suivant un 1-cycle. Intéressons-nous maintenant aux conséquences topologiques d'un découpage par chemin simple et n-cet.

Découpage suivant un chemin simple

D'un point de vue topologique, découper une surface suivant un chemin simple revient à découper la surface suivant un lacet simple, puis à recoller la surface au niveau d'une des arêtes ayant servi au découpage. En faisant cette constatation, on peut donc déduire les propriétés suivantes en s'appuyant sur les résultats de la section précédente :

Théorème 3.3 – *Soit S une surface discrète (à une composante connexe) de genre $g \neq 0$ et à n bords, et c un chemin simple non séparant sur S. Alors $S' = ✂(S, c)$ est composée d'une seule composante connexe, son genre est compris entre 0 et g, et le nombre de bords de S' est compris entre 1 et $n + g - 1$.*

Preuve : En utilisant le théorème 3.1, et en suivant la remarque soulignant la similitude entre découpage par chemin et découpage par lacet puis recollement suivant une arête, on voit facilement que le genre de S' est au plus égal à g, en remarquant que le recollement le long d'une arête qui a servi de découpage augmentera le genre de un. On montre de plus facilement que la borne inférieure pour le genre peut être atteinte sur une surface ayant au moins un bord B, en utilisant le lacet γ construit dans la preuve du lemme 3.1, et en retirant de ce lacet toutes les arêtes du bord B.

Le nombre de bords résultant de ce découpage est lui au moins égal à 1, ce qui correspond au découpage suivant un chemin simple n'intersectant aucun bord. On montre ensuite que le nombre de bords introduits par ce découpage est au plus égal à $g - 1$ en

utilisant le lemme 3.2, et en remarquant que le recollement d'une arête permettant de transformer un découpage suivant un lacet diminue le nombre de bords de un. □

Si on ajoute comme contrainte supplémentaire que le chemin ne peut passer au plus qu'une fois par chacun des bords de la surface, alors on obtient le résultat suivant :

Théorème 3.4 – *Soit S une surface discrète (à une composante connexe) de genre $g \neq 0$ et à n bords, et c un chemin simple sur S passant au plus une fois par chacun des bords de S. Alors $S' = ✂(S, c)$ est composée d'une seule composante connexe, son genre est égal à g, et le nombre de bords de S' est compris entre 1 et $n + 1$.*

Preuve : On voit immédiatement que le genre de la surface n'est pas modifié par un tel découpage, par exemple en regardant le lacet correspondant dans la surface $\check{S}$ (la surface produite par identification de tous les points d'un même bord de S). D'autre part, un tel chemin introduit au plus un bord, car s'il en introduisait deux, ce ne serait pas un chemin mais un lacet. Enfin, si le chemin passe par chacun des bords de S, alors la surface résultante n'aura plus qu'un seul bord, d'où la borne minimale. □

En particulier, nous utiliserons dans la suite des chemins de découpage reliant un bord à un autre. Pour ces chemins, on a la propriété suivante, à la preuve immédiate :

Proposition 3.3 – *Soit S une surface discrète de genre $g \neq 0$ et à n bords, et c un chemin simple sur S reliant deux bords distincts de S. Alors le genre de $S' = ✂(S, c)$ est g, et le nombre de bords n' de S' est strictement inférieur à n. Si c ne croise pas d'autre bord de S, alors on a $n' = n - 1$.*

Découpage suivant un m-cet

Nous avons vu avec le lemme 1.1 que sur une surface S sans bord de genre g, un m-cet dont le complémentaire dans S était homéomorphe à un disque comportait exactement $2g + 1$ chemins reliant ses deux points base.

De manière plus générale, on peut remarquer que découper suivant un m-cet composé de $n \geq 2$ chemins $(\gamma_i)_{1 \leq i \leq m}$ revient à découper la surface suivant un lacet $\gamma_i \cdot \gamma_j^{-1}$, puis de découper suivant les $m - 2$ autres chemins, qui eux sont des chemins reliant deux bords.

En conjuguant le théorème 3.1 et le théorème 3.3, on obtient le résultat suivant :

Théorème 3.5 – *Soit S une surface discrète (à une composante connexe) de genre $g \neq 0$ et à n bords, et $(\gamma_i)_{1 \leq i \leq m}$ un m-cet non séparant sur S. Alors $S' = ✂(S, (\gamma_i)_{1 \leq i \leq m})$ est composée d'une seule composante connexe, son genre est compris entre 0 et $g - m + 1$, et le nombre de bords de S' est compris entre 1 et $n + g$.*

Preuve : Les bornes minimales du genre et du nombre de bord sont une conséquence directe de l'application successive du découpage par un lacet puis par un chemin entre bords. Le genre maximal est lui obtenu en remarquant qu'après avoir découpé suivant un premier lacet non séparant qui réduira au moins le genre de 1, chaque nouveau chemin réduira lui aussi le genre de 1, sinon il serait séparant.

Enfin, le nombre maximum de bords est borné par $n + g$, car chaque découpage introduisant un nouveau bord diminue le genre de la surface. Cette borne est bien atteinte dans le cas décrit par le théorème 3.1. □

Si l'on impose de plus que le m-cet ne passe qu'au plus une seule fois par chaque bord, on obtient les bornes topologiques suivantes :

Théorème 3.6 – *Soit S une surface discrète (à une composante connexe) de genre $g \neq 0$ et à n bords, et $(\gamma_i)_{1\leq i\leq m}$ un m-cet non séparant sur S passant au plus une fois par chacun des bords de S. Alors $S' = ✂(S, (\gamma_i)_{1\leq i\leq m})$ est composée d'une seule composante connexe, son genre est égal à $g - m + 1$, et le nombre de bords de S' est compris entre $1 + \mathbb{1}_{m\text{pair}}$ et $n + m$.*

Preuve : Le genre de la surface résultante est nécessairement de $g - m + 1$, car comme dans la preuve du théorème précédent, le premier lacet réduit le genre de un, et tous les chemins suivants du m-cet font de même.

Le nombre minimal de bords est atteint si tous les bords de S sont traversés par le m-cet, et si $m > 2$. Dans ce cas, le premier lacet ajoute deux bords. On remarque alors que si un découpage produit deux bords, alors l'ajout d'un chemin reliant ces deux bords pour compléter le m-cet réduira à un le nombre de bords. À l'inverse, si un découpage ne produit qu'un seul bord, tout nouveau chemin venant compléter le m-cet ajoutera un nouveau bord.

Enfin, le nombre maximal de bords est atteint dans le cas où le m-cet ne traverse aucun bord déjà existant de S, et en construisant le m-cet à partir d'un premier lacet (ajoutant deux bords), puis en ajoutant de nouveaux chemins qui relient à chaque fois deux sommets appartenant au même bord. □

3.2 Découpage en une surface homéomorphe à un disque

Les découpages suivant des lacets, chemins et n-cets tel que décrits à la section précédente sont les ingrédients élémentaires que l'on se propose d'utiliser dans les deux prochaines sections pour résoudre le problème du découpage en une surface homéomorphe à un disque.

Nous commençons ici par poser précisément le problème, puis nous décrivons rapidement les motivations d'une telle opération, et enfin nous proposons un algorithme élémentaire mais efficace pour répondre à ce problème s'il ne lui est ajouté aucune contrainte.

3.2.1 Description du problème et motivations

On profite de la dualité entre découpage et pavage pour décrire les propriétés des découpages qui vont nous intéresser ici :

Définition 3.1 – *On dit qu'un découpage $\mathcal{C}$ d'une surface S est un* découpage en disque *de cette surface s'il est équivalent à un M-pavage de cette surface par une M-tuile unique de genre zéro à un bord.*

Étant donnée une surface, il existe un grand nombre de découpages vérifiant la propriété définie ici. Il est possible, comme nous le présenterons dans la section suivante, de définir des contraintes pour raffiner le problème. Cependant, on peut remarquer plusieurs motivations indépendantes de ce raffinement qui rendent la question intéressante.

Tout d'abord, on peut noter qu'un tel découpage existe toujours, quel que soit la surface S considérée. Cette existence ne serait pas assurée si l'on s'intéressait au pavage suivant une tuile de genre ou de nombre de bords différent. L'existence d'un découpage en

disque est immédiate si l'on considère le résultat sur la topologie d'une série de découpages suivant un lacet ou un chemin, tel que décrits à la section précédente.

Une motivation essentielle de ce découpage est d'obtenir une surface qui pourra être ensuite dépliée sur un sous-ensemble de $\mathbb{R}^2$. En effet, les méthodes que nous exploiterons dans les différentes applications proposées dans la partie III nécessitent que les surfaces manipulées soient homéomorphes à un disque. Enfin, nous exploiterons également cette propriété topologique pour produire un pavage de surfaces en quadrangles (voir section 5.4).

3.2.2 Algorithme linéaire sans contrainte

Si l'on considère la question de trouver un découpage en disque d'un 2-maillage $\mathcal{M}$ donnée, sans imposer plus de contrainte, alors on peut résoudre le problème un un temps linéaire en le nombre de triangles de la surface :

Propriété 3.1 – *Étant donné un 2-maillage $\mathcal{M}$, on peut produire un découpage en disque de cette surface en $O(|T(\mathcal{M})|)$, où $T(\mathcal{M})$ est le nombre de triangles de $\mathcal{M}$.*

Preuve : L'algorithme que nous présentons ici s'appuie sur un parcours en largeur ou en profondeur du graphe G des triangles de S, où les sommets de G sont les triangles de $\mathcal{M}$, et les arêtes de G la relation d'adjacence des triangles dans $\mathcal{M}$.

Algorithme 3.1 : Découpage en disque sans contrainte

```
Données : M un 2-maillage
Résultat : C découpage sur M
début Construction de M'
    t ← triangle de M ;
    Ajouter t à la liste O ;
    Marquer t comme vu ;
    tant que O ≠ ∅ faire
        Soit t' le premier triangle de O ;
        Retirer t' de O;
        Ajouter t' à M' en préservant la propriété topologique de M';
        pour chaque t'' voisin de t' dans M faire
            si t'' n'est pas encore vu alors
                Ajouter t'' à la liste O ;
                Marquer t'' comme vu ;
            fin
        fin
    fin
fin
début Construction du découpage
    Ajouter à C toutes les arêtes de M correspondant aux bords de M';
fin
```

La procédure d'ajout d'un nouveau triangle t' à $\mathcal{M}'$ se fait en recollant t' à ses voisins déjà présents dans $\mathcal{M}'$ uniquement si le recollement ne change pas la topologie de la surface en cours de construction, qui doit être homéomorphe à un disque. Le test de modification de topologie peut facilement être opéré en étudiant la caractéristique d'Euler avant et après le possible recollement.

On peut remarquer ici que le découpage produit $\mathcal{C}$ n'est pas décrit sous sa forme la plus synthétique, car il est composé d'un ensemble de chemins ne comportant chacun qu'une seule arête. On pourrait facilement raffiner cette description en construisant des chemins couvrants cet ensemble d'arêtes, par exemple en reliant chaque sommet de $\mathcal{M}$ qui a au moins trois arêtes dans notre découpage proposé.

Le parcours décrit par l'algorithme 3.1 est linéaire en le nombre de sommets et d'arêtes de G. Or, le nombre d'arêtes de G est au plus égal à $3\,|T(\mathcal{M})|$. La complexité de cet algorithme est donc de $O(|T(\mathcal{M})|)$. □

Remarque : Si dans l'algorithme 3.1 on prend soin de trier l'ensemble O des triangles suivant la distance au premier élément ajouté, alors l'ensemble des sommets du bord construit par l'algorithme correspond à un sous-ensemble non arborescent des *lieux de découpage* (« cut locus » en anglais) de la surface. En quelques mots, le cut locus d'un sommet p sur une variété est l'ensemble des points qui possèdent plusieurs géodésiques les connectant à p. On appellera donc par la suite *cut locus élagué* le sous-ensemble non arborescent du cut locus.

3.2.3 Combinatoire et longueur de découpage

La section précédente décrit une manière de construire un découpage vérifiant la propriété donnée par la définition 3.1. Cependant, si l'on ajoute des contraintes à cette définition, alors la complexité d'un algorithme de construction augmente nettement.

L'un des critères les plus intuitifs que l'on peut chercher à optimiser est la longueur du découpage. Comme nous le verrons à la section suivante, Erickson et Har-Peled [34] ont montré que dans le cas général, ce problème était NP-difficile. Cependant, il est possible de contraindre un peu plus le problème, en imposant par exemple l'ensemble des points de croisement des chemins et lacets de découpage. Par exemple, en imposant un unique point de croisement, Erickson et Whittlesey [35] ont proposé un algorithme polynomial qui calcule le système de lacets le plus court basé en un point. Colin de Verdière [18] a récemment généralisé cette approche à un ensemble quelconque de points de croisement. Ce problème du découpage optimal peut également être abordé en cherchant le découpage le plus court *homotope* à un découpage donné. Colin de Verdière et Lazarus [19] ont ainsi proposé une méthode en temps polynomial répondant à cette question.

Cependant, ces différentes méthodes n'assurent pas que le schéma polygonal issu du découpage soit sous sa forme canonique. S'il existe des approches [68] permettant de calculer un schéma polygonal sous forme canonique, à notre connaissance il n'existe pas de résultat proposant une approximation du plus court découpage en schéma polygonal canonique.

3.3 Découpage minimal

Dans cette section, nous nous intéressons au découpage minimal d'une surface orientée quelconque en une surface homéomorphe à un disque. Plusieurs travaux antérieurs ont déjà exploré cette question. Après avoir posé le problème et remarqué avec Erickson et Har-Peled [34] que c'est un problème NP-difficile, nous proposons une approximation de ce découpage basée sur les lacets non séparants. Nous illustrons ensuite par quelques résultats pratiques l'efficacité applicative des optimisations que nous avons proposées [39].

3.3.1 Définition et NP-difficulté

On définit naturellement un découpage minimal en s'appuyant sur la notion de longueur d'un découpage décrite par la définition 2.27 :

Définition 3.2 – *Un découpage $\mathcal{C}$ d'une surface S de genre g à n bords en un disque est dit* minimal *s'il n'existe pas de découpage plus court en un disque de S.*

On montre que construire un tel découpage est NP-difficile, en considérant le théorème suivant :

Théorème 3.7 – *Calculer la longueur du découpage minimal en un disque d'un 2-maillage $\mathcal{M}$ est NP-difficile.*

La preuve de ce théorème consiste à ramener le problème de l'arbre de Steiner rectiligne, qui est un problème NP-difficile [44], à un problème de découpage minimal en un disque d'un 2-maillage.

L'algorithme 3.2 que nous proposons reprend ainsi en dernière étape un algorithme d'approximation de calcul d'arbre de Steiner minimum, à la manière de [34].

3.3.2 Découpage par lacets non séparants

Nous proposons ici un algorithme calculant une approximation d'un découpage vérifiant la propriété donnée par la définition 3.2, en s'appuyant sur la recherche de lacets non séparants les plus courts.

Algorithme global

L'algorithme 3.2 est assez similaire à celui présenté dans [34], à cela près qu'il utilise des lacets non séparants plutôt que des lacets non triviaux (qui eux peuvent être séparants). Ce choix permet de s'affranchir d'une étape finale de recollement des différentes composantes connexes.

De plus, Thomassen [102] a montré que les lacets non séparants vérifient la propriété des trois chemins :

Définition 3.3 – *Soit F un ensemble de cycles sur un graphe G, on dit que F vérifie la* propriété des trois chemins *si pour tout p_0, p_1 et p_2 chemins reliant deux sommets x et y sur G tels que leurs intérieurs ont une intersection disjointe, et tels que deux des cycles de $p_0 \cup p_1 \cup p_2$ ne soient pas dans F, alors le troisième cycle de $p_0 \cup p_1 \cup p_2$ n'est pas dans F.*

Or, un ensemble disposant d'une telle propriété certifie la validité de leur construction par propagation de front, toujours d'après [102], ce qui justifie la pertinence du choix de ce critère.

Nous avons également fait le choix de rechercher dans notre algorithme le lacet le plus court, et non uniquement une approximation du lacet le plus court, comme pro-

posé dans [34]. Les améliorations de l'algorithme que nous décrirons dans la section 3.3.3 permettent d'affiner la complexité effective de cette approche exacte.

Algorithme 3.2 : Découpage en disque par lacets non séparants

Données : Un 2-maillage $\mathcal{M}$ non homéomorphe à une sphère
Résultat : Un pavage $T = (C, \mathcal{M}, F)$
$T = (C, \mathcal{M}, F)$ M-tuile initialisée à $\mathcal{M}$;
tant que *le genre de C est non nul* **faire**
  Trouver le plus court lacet non séparant c de T;
  Découper T suivant c;
fin
tant que *Le nombre de bords de C est différent de* 1 **faire**
  Trouver le plus court chemin c de T reliant deux bords;
  Découper T suivant c;
fin

Cet algorithme constitue la trame générale du calcul d'une approximation du découpage minimal en un disque, la méthode de recherche des plus courts lacets et chemins qu'il utilise étant détaillée ci-dessous.

Plus court lacet non séparant depuis un point

Commençons d'abord par étudier le problème de la recherche du plus court lacet non séparant basé en un sommet s d'un 2-maillage $\mathcal{M}$.

Théorème 3.8 – *Soit $\mathcal{M}$ un 2-maillage 1-connexe non homéomorphe à un disque, composé de n sommets, et s un sommet de $\mathcal{M}$. On peut calculer le plus court 1-cycle sur $\mathcal{M}$ non séparant et passant par s en $O(n \log n)$.*

Preuve : On utilise pour cela une surface croissante S_s à partir de s sur $\mathcal{M}$, en respectant l'ordre des sommets défini par l'algorithme de Dijkstra sur $\mathcal{M}$ calculé depuis s. Lorsque l'on observe une jonction entre deux parties du bord de S_s, alors on a détecté un lacet qui passe par s et le point de jonction. On étudie alors le nombre de composantes connexes du complémentaire de ce lacet de $\mathcal{M}$. Si l'on observe plus d'une composante connexe, alors le lacet est séparant, et on continue le processus de propagation. À l'inverse, le lacet n'est pas séparant, et on a trouvé un lacet non séparant passant par s. Les lacets non séparants vérifiant la propriété des trois chemins [102], on a bien construit ici le plus court lacet non séparant.

Cet algorithme dépend donc uniquement de la complexité $O(n \log n)$ de l'algorithme 2.1, car le calcul de composantes connexes est linéaire en n. □

Le résultat présenté dans ce théorème est la brique élémentaire de l'algorithme général, et est utilisée pour calculer le plus court lacet non séparant sur la totalité de la surface.

Plus court lacet non séparant

Afin de calculer le plus court lacet non séparant sur la surface complète, on s'appuie sur un ensemble de points base potentiels qui assurent la validité du lacet obtenu :

Théorème 3.9 – *Soient $\mathcal{M}$ un 2-maillage 1-connexe non homéomorphe à un disque, et B l'ensemble des sommets d'un découpage de $\mathcal{M}$ en disque. On peut calculer le plus court 1-cycle sur $\mathcal{M}$ non séparant en $O(|B|\, n \log n)$.*

Preuve : Il suffit de constater que chaque 1-cycle non séparant passe nécessairement par un sommet d'un découpage en disque C, en remarquant que si un 1-cycle n'a pas de sommet dans C, alors il est uniquement dans le complémentaire de C, qui est homéomorphe à un disque. Or, sur une telle surface, tout lacet est trivial, donc séparant.

Ainsi, en calculant pour chacun des $|S(C)| = |B|$ sommets s de C le plus court 1-cycle non séparant basé en s (théorème 3.8), et en retenant le plus court d'entre tous, on calcule bien le plus court 1-cycle du 2-maillage avec la complexité annoncée. □

En utilisant l'algorithme 3.1, on peut construire un découpage en disque C en temps linéaire. On a donc un algorithme de calcul de plus court lacet non séparant en $O(|B|\, n \log n)$, avec en pratique $|B| << n$.

Plus courts chemins entre bords

Le calcul des plus courts chemins entre bords, c'est-à-dire la dernière étape de l'algorithme 3.2, correspond à une approximation du calcul de l'arbre de Steiner de poids minimum reliant chacun des bords.

Définition 3.4 – *Étant donnés un graphe G et un ensemble de sommets V de G, un* arbre de Steiner *est un arbre ayant pour support G et couvrant tous les sommets de V.*

La dernière étape de l'algorithme consistant, dans l'absolu, à construire un découpage ayant pour support les arêtes de la surface discrétisée, et passant par chacun des bords de la surface (générés ou non par les découpages suivant les lacets), on cherche donc bien ici à appliquer une variante du calcul de l'arbre de Steiner de poids minimum. Or, ce problème fait partie des 21 problèmes identifiés par Karp [60] comme étant NP-difficiles.

On construit donc une approximation d'un tel arbre, à la manière de [34], en calculant itérativement le plus court chemin entre deux bords. Pour cela, on procède en propageant simultanément depuis chacun des bords un front d'onde, en s'appuyant sur les ordres définis par l'algorithme de Dijkstra. Lorsque deux fronts d'onde se joignent, alors on découpe la surface suivant un plus court chemin joignant les deux bords et passant par le point de jonction. On fusionne les deux fronts d'onde correspondants, puis on continue la propagation des fronts restants.

Complexité

La complexité de la première partie de l'algorithme 3.2 dépend directement de la complexité de l'algorithme permettant de calculer le plus court lacet non séparant. Puisque chaque lacet non séparant va réduire le genre de la surface, on va procéder à g recherches de lacets non séparants. La complexité de la première partie de l'algorithme est donc de $O(g\,|B|\, n \log n)$, où B est l'ensemble des sommets d'un découpage en disque de la surface. La complexité de l'approximation d'arbre de Steiner est celle de l'algorithme de Dijkstra, c'est-à-dire $O(n \log n)$.

On en conclut donc la complexité globale de l'algorithme 3.2 en $O(g\,|B|\, n \log n)$, avec en pratique $|B| << n$.

3.3.3 Optimisations de l'algorithme

Nous décrivons ici quelques raffinements de l'algorithme qui ne sont pas significatifs d'un point de vue théorique, mais assurent une nette amélioration en pratique des temps de calcul sur 2-maillages, comme nous l'illustrerons à la section 3.3.3.

FIG. 3.3 – *Voisinage topologique V_{l_s} d'un sommet s appartenant à l'ensemble des points base B sur la surface S.*

Utiliser d'un Dijkstra borné

La première amélioration que nous introduisons ici concerne le calcul du plus court lacet non séparant. L'objectif est de profiter des calculs de plus courts lacets déjà effectués pour interrompre les recherches de lacets quand on sait qu'elles deviennent inutiles.

Ainsi, l'algorithme décrit dans la section 3.3.2 calcule pour chaque point base b le 1-cycle non séparant le plus court correspondant, et compare la longueur de ce dernier avec la longueur l du le plus court des lacets calculés précédemment. Ainsi, les lacets plus long que l seront ignorés. Or, le calcul du plus court lacet non séparant basé en b s'appuie essentiellement sur l'algorithme de Dijkstra. On propose d'utiliser la distance l du plus court lacet déjà calculé comme information supplémentaire, en modifiant l'algorithme pour qu'il calcule *le plus court lacet non séparant basé en b et plus court que l*. Cela entraîne naturellement l'arrêt du processus de croissance de région si la longueur est dépassée.

Afin d'améliorer la complexité de cette version modifiée, on introduit la notion de voisinage topologique (figure 3.3) :

Définition 3.5 – *Étant donnés un 2-maillage $\mathcal{M}$ et un sommet s de $\mathcal{M}$, on appelle* voisinage topologique*, de s, noté V_{l_s}, le sous complexe cellulaire de $\mathcal{M}$ couvert par la zone de croissance au moment de la détection du plus court lacet non séparant l_s basé en s.*

En considérant l'amélioration décrite dans cette section, on constate que la taille du voisinage topologique du premier point base visité décrit la complexité dans les pires des cas. Si l'on note m la taille du plus grand voisinage topologique, et B l'ensemble des points appartenant au découpage décrit par l'algorithme 3.1, on peut alors affirmer que $O(|B|\, m \log m)$ est la complexité au pire de la recherche du plus court lacet.

En pratique, $m << n$ et $|B| << n$, et on dispose ainsi d'un algorithme efficace de découpage topologique s'appuyant sur le calcul des lacets non séparants les plus courts, comme les résultats proposés plus loin l'illustrent.

Ordonner les points base

L'amélioration décrite ci-dessus entraîne une dépendance forte de l'efficacité de l'algorithme à l'ordre des points base traités : il devient intéressant de traiter d'abord les points correspondant aux lacets les plus courts pour tronquer plus tôt les autres calculs.

On propose ici d'introduire une étape de prétraitement, de type heuristique, permettant de visiter dès le début des lacets de longueur proche des plus courtes longueurs. Cette première partie de notre algorithme est semblable à la méthode proposée dans [34] pour calculer une approximation du découpage optimal. Elle s'appuie sur la structure particulière de l'ensemble des points base potentiels B. Si nous profitons ici de cet ensemble déjà calculé, elle n'est pas la seule envisageable. Ainsi, on trouve une approche similaire [57,

(a) Structure de l'ensemble des points base B calculés à partir d'un triangle t.

(b) Plus court lacet associé à un chemin composant B.

FIG. 3.4 – *Calcul de lacets approchés*

chapitre 6] où l'auteur utilise cette fois-ci un graphe de Reeb pour construire des lacets approchés semblables.

Dans ces deux cas, l'ensemble de points B est composé de chemins reliant des sommets d'arité au moins 3. La figure 3.4(a) illustre cette structure sur l'ensemble des points base d'un double tore. Pour chacun de ces chemins C, on construit une approximation du lacet le plus court en calculant dans le complémentaire de B le plus court chemin entre les deux bords issus de C (figure 3.4(b)).

Le nombre de ces lacets est en $O(g)$, et le plus court d'entre eux est une bonne approximation du lacet non séparant le plus court. Erickson et Har-Peled [34] ont montré que ce lacet non séparant avait au plus deux fois la longueur du lacet le plus court traversant B.

Cette première étape permet donc de stopper plus rapidement les propagations de front de l'algorithme initial.

Utiliser les découpages précédents

De la même manière que nous avons décrit comment tronquer la propagation de front lorsque la distance devient trop grande, on peut s'intéresser aux distances trop courtes. Ici, la complexité théorique de la méthode n'est en rien améliorée, mais cette approche permet d'éviter quelques calculs de nombre de composantes connexes, qui sont linéaires en le nombre de sommets du maillage.

On se place ici au cœur de l'algorithme 3.2, dans le cas où l'on a déjà trouvé un lacet de découpage l_1 et réduit le genre. Si le genre de la surface résultante est encore non nul, alors l'algorithme cherche de nouveau un lacet non séparant à partir d'un nouvel ensemble de points base potentiels B'. Le lacet l_1 était le plus court de tous les lacets non séparants de la surface, aussi on sait donc que les lacets non séparants que l'on cherche maintenant sont au moins aussi long que l_1.

L'optimisation consiste, lors de la détection d'une jonction de front dans la propagation d'onde depuis un point base de B', à ne calculer la topologie correspondant au lacet associé uniquement si sa longueur est supérieure ou égale à celle de l_1. En effet, il est immédiat de voir que si un lacet plus court est trouvé, il sera nécessairement séparant, sinon on l'aurait retenu à l'étape précédente au lieu de l_1.

Résultats pratiques

L'annexe A décrit l'implémentation qui a été effectuée des différents algorithmes décrits dans ce mémoire. Nous avons ainsi pu étudier la pertinence pratique des améliorations de l'algorithme 3.2.

Les résultats expérimentaux que nous présentons dans le tableau 3.1 ont été réalisés sur un ordinateur portable Intel® Core™ 2 Duo cadencé à 2.60 GHz. Ils illustrent l'efficacité pratique des optimisations proposées.

Surface	Sans optimisation	Avec optimisations
4-Ts	6,69 s	0,76 s
4-T	331 s	33,2 s
4-Th	962 s	49 s

TAB. 3.1 – *Temps de calcul de la première partie de l'algorithme 3.2 sans optimisation et avec optimisation sur des modèles synthétiques. 4-Ts, 4-T et 4-Th sont des surfaces de genre 4 composées respectivement de 2 202, 13 242 et 35 322 sommets.*

3.3.4 Algorithme approché

Si les résultats que nous avons décrits à la section précédente mettent en évidence la pertinence de nos optimisations, on constate que les temps de calcul sont tout de même élevés, comparés aux méthodes habituellement exploitées en infographie. Ces temps de calcul sont principalement dûs à la complexité de la méthode, elle-même conséquence du problème que l'on résout ici : le calcul exact des plus courts lacets sur la surface.

Dans un cadre applicatif en infographie, il peut être utile d'utiliser un calcul approché des plus courts lacets, afin d'accroître la vitesse d'exécution sans trop détériorer les découpages obtenus.

Une manière simple d'obtenir une telle approximation consiste à appliquer une méthode proche de celle servant à ordonner les points base (section 3.3.3). Elle consiste à calculer pour chacun des chemins structurant l'ensemble des points base une approximation du plus court lacet non séparant traversant le chemin, en propageant un front d'onde depuis un côté vers l'autre. [34] décrit cette méthode d'approximation, et montre que le plus court lacet trouvé par cette approche est au pire deux fois plus long que le plus court. On montre aussi facilement que la complexité d'une telle recherche de lacets est de $O(gn \log n)$, en constatant que le nombre de chemins structurant B est en $O(g)$, et que l'on applique une propagation de front pour chaque chemin.

Nous avons donc également implémenté cette version pour l'utiliser dans un cadre applicatif où les temps de calcul sont cruciaux. Le tableau 3.2 compare les temps de calculs et longueurs des lacets obtenus avec notre algorithme exact optimisé et l'algorithme approché décrit ici, sur des surfaces synthétiques de genre un à quatre. Les temps de calcul présentés dans ce tableau correspondent au temps nécessaire à la recherche de l'ensemble des g lacets non séparants les plus courts sur une surface de genre g. Les longueurs quant à elles correspondent à la longueur des bords de la surface après découpage suivant les lacets, soit le double de la longueur de ces derniers.

On constate tout d'abord la rapidité d'exécution de la méthode approximative comparée à l'approche initiale. Le coût de l'exactitude des solutions assurées par l'algorithme 3.2 avait été mesurée d'un point de vue complexité, on observe ici qu'elle est également élevée dans la pratique. Pour illustrer l'efficacité pratique du découpage approximatif, nous l'avons appliqué en 48 minutes sur une surface de genre 104, sans bord et composée de 543 652 triangles. La figure 3.5 présente le résultat de ce découpage.

On constate également que dans les exemples traités, les différences de longueur de chemin ne sont pas très élevées, et le sont d'autant moins pour une surface donnée que le nombre de sommets augmente sur le maillage.

3.3.5 Discussion autour des variations

L'algorithme 3.2 tel qu'il est proposé plus haut est séparé en deux étapes successives : le genre est tout d'abord ramené à zéro, puis le nombre de bords est réduit à un.

Surface		Algorithme exact		Algorithme approché	
Genre	Nb sommets	Temps (s)	Longueur	Temps (s)	Longueur
1	768	0,08	3,139 33	0,01	3,296 95
1	3 072	0,51	3,139 33	0,04	3,245 53
1	12 288	5,09	3,139 33	0,19	3,180 83
2	830	0,13	7,070 2	0,02	7,587 28
2	3 326	1,05	7,070 2	0,08	7,676 53
2	13 310	9,72	7,070 2	0,42	7,392 49
3	1 724	0,52	34,360 9	0,07	56,182 1
3	6 908	3,45	34,360 9	0,34	55,308 7
3	27 644	32,49	34,360 9	1,84	37,619
4	2 202	0,88	41,770 3	0,13	86,941 2
4	8 826	5,48	41,770 3	0,68	83,485 4
4	35 322	58,37	41,770 3	3,69	66,257 5

Tab. 3.2 – *Comparaison des méthodes exactes et approximatives de découpage selon les plus courts lacets non séparants.*

Fig. 3.5 – *Découpage d'une surface de genre* 104*, sans bord et composée de* 543 652 *triangles à l'aide de l'algorithme de recherche approximative du plus court lacet non séparant. Maillage 3D original fourni par le dépôt de formes laboratoire Computer Graphics de l'Université de Stanford.*

Une variation simple consiste à découper une surface donnée, initialement de genre g à n bords en une surface de genre g' et à n' bords, avec $g' \leq g$ et $n' \leq n + 2(g - g')$. Il suffit pour cela de s'arrêter plus tôt dans la recherche des lacets non séparants, lorsque l'on atteint un genre g'. On peut facilement voir que l'étape de réduction du nombre de bords (qu'elle réalise l'arbre de Steiner exact ou qu'elle soit approchée par la méthode décrite plus haut) préserve le genre de la surface. En effet, si elle ne le préservait pas, il existerait un cycle sur les arêtes de l'arbre de découpage complétées par les arêtes bord de la surface. Pour atteindre un nombre de bord souhaité, on interrompt là aussi le processus de découpage.

Cependant, cette approche ne permet pas de contrôler le nombre de 0-m-cellules du M-pavage correspondant. Nous décrirons dans la section 5.3 deux approches permettant de réduire le nombre de sommet du polygone de pavage.

3.4 Conclusion

Nous avons présenté au début de ce chapitre les conséquences topologiques d'un découpage sur une surface donnée, en étudiant les différentes propriétés combinatoires impliquées par le découpage suivant des structures classiques.

Nous avons ensuite étudié la question du découpage topologique d'une surface quelconque en une surface homéomorphe à un disque, d'abord d'un point de vue combinatoire, puis en étudiant la question sous la contrainte du plus court découpage. Nous avons proposé des améliorations et optimisations aux algorithmes de découpage déjà existants, en étudiant à la fois la qualité des découpages obtenus, et la complexité effective de ces approches.

La question de la topologie est la colonne vertébrale des outils de découpage et de pavage que nous présentons dans ce manuscrit. Les concepts présentés dans ce chapitre constituent donc le point de départ des explorations proposées au fil des prochains chapitres.

Chapitre 4
Contraintes géométriques

Sommaire

4.1 Distance non euclidienne 60
4.1.1 Courbure locale 60
4.1.2 Distorsion d'une paramétrisation conforme 61
4.2 Détection des extremums 63
4.2.1 Travaux antérieurs 63
4.2.2 Détection par transformation conforme 64
4.2.3 De la géométrie à la topologie 66
4.3 Patch des surfaces 67
4.3.1 Travaux similaires 67
4.3.2 α-jonctions 67
4.3.3 Simplification par ajustement de la topologie 68
4.4 Conclusion 69

Nous avons décrit au chapitre précédent un ensemble d'outils de pavage s'appuyant sur les propriétés topologiques des surfaces. Cependant, deux surfaces topologiquement identiques peuvent avoir des géométries radicalement différente, et la prise en compte des longueurs des chemins ne suffit pas à en saisir toute la complexité.

Bien que cette notion soit difficile à capturer de manière formelle, nous nous proposons dans ce chapitre de répondre en partie à cette question en décrivant des méthodes de manipulation topologique de surface n'ignorant pas la géométrie.

Dans un premier temps, nous décrivons comment des modifications de la distance euclidienne permettent de capturer une part de la géométrie locale des surfaces lors des étapes de découpage déjà évoquées jusqu'ici. Nous montrons ensuite comment une telle méthode répond à la question de la détection globale de sites géométriques structurant la surface. Enfin, nous nous intéressons aux surfaces issues d'acquisitions basse résolution, pour lesquelles la géométrie permet de détecter et de corriger d'éventuelles erreurs topologiques résultant du mode d'acquisition.

(a) Exemple de plus courts chemins sur un cube, avec longueur modifiée ou non par la courbure locale.

(b) Calcul de la courbure locale portée par l'arête e, et décrite par l'angle $\alpha(e)$.

FIG. 4.1 – *Calcul et utilisation de la courbure locale dans la notion de longueur d'arête.*

4.1 Utilisation de distances non euclidiennes

4.1.1 Courbure locale

Les différents algorithmes de découpage que nous utilisons dans ces travaux reposent sur le calcul de plus courts chemin, via l'algorithme de Dijkstra. Ce dernier manipule une notion géométrique élémentaire : la distance géodésique. Nous avons déjà décrit dans la section 2.3.2 différents choix de distances non euclidiennes pour modifier le comportement global de ces recherches de plus court chemin. Nous présentons ici une modification simple de cette distance pour appréhender des notions géométriques plus complètes.

Une manière simple de procéder sans pour autant changer les algorithmes qui seront décrits est de modifier là encore la longueur des arêtes, cette fois-ci en y introduisant une information géométrique, comme la courbure locale par exemple. On souhaite en effet favoriser le passage des chemins le long d'arêtes qui sont à forte courbure. La figure 4.1(a) illustre cette idée. Le cube de gauche présente un plus court chemin entre a et b. Le chemin présent sur le cube de droite passe quant à lui le long des arêtes qui ont une courbure locale importante. Ce choix évite de faire traverser de grandes zones plates aux chemins.

La prise en compte de cette courbure locale dans la longueur des arêtes peut être réalisée de manière multiplicative. Étant donnée une arête e qui est l'intersection de deux triangles, on note $\alpha(e) \in [0, \pi]$ l'angle formé par les normales de ces deux triangles. Ainsi, $\alpha(e) = 0$ si la surface décrite par les deux triangles est plane, et $\alpha(e) = \pi$ dans le cas d'une courbure extrême.

Si l'on souhaite que les découpages soient favorisés le long des arêtes portant des courbures locales fortes, il est donc nécessaire que le coefficient multiplicatif soit égal à 1 quand $\alpha(e) = 0$, et qu'il soit positif et décroissant en $\alpha(e)$. Afin de répondre à ce critère, on construit la famille de fonctions $f_c : e \mapsto 1 - c\frac{\alpha(e)}{\pi}$, avec $c \in [0, 1]$. Ces fonctions linéaires en $\alpha(e)$ permettent de calculer pour un coefficient c fixé suivant le besoin une nouvelle distance $l_c : A(\mathcal{M}) \to \mathbb{R}^+$ définie par $l_c(e) = \|e\|_3 \, f_c(e)$, où $\|e\|_3$ est la longueur euclidienne de e.

D'un point de vue applicatif, le paramètre $c \in [0, 1]$ devient un paramètre d'ajustement de l'importance de la prise en compte de la courbure locale dans le choix des découpages.

La figure 4.2 présente deux découpages topologiques d'une même surface, utilisant à chaque fois l'algorithme 3.2, mais en faisant varier la distance utilisée. L'utilisation de la courbure locale comme paramètre de distance modifie comme attendu le résultat de l'algorithme d'un point de vue géométrique, en ajustant les découpages suivant les besoins spécifiques, sans pour autant en changer le comportement topologique.

(a) Découpage topologique par l'algorithme 3.2 en utilisant la distance euclidienne.

(b) Découpage topologique par l'algorithme 3.2 en utilisant une distance tenant compte de la courbure ($c = 0,4$).

FIG. 4.2 – *Variation géométrique du découpage suivant la distance utilisée.*

4.1.2 Distorsion d'une paramétrisation conforme

Dans la section précédente, nous avons décrit une approche prenant en compte les informations de géométrie locales dans la mesure de distance sur les maillages. Ainsi, la longueur modifiée des arêtes est seulement déterminée par la longueur euclidienne, ainsi que par les deux triangles qui lui sont adjacents. Une approche moins spécifique pourrait consister à utiliser un petit voisinage autour de chacune des arêtes, dans lequel on calculerait la courbure locale par exemple. Cependant, cette approche est toujours locale.

Nous nous proposons ici de décrire une modification de la distance sur la surface prenant en compte une propriété géométrique globale. On s'appuie pour cela sur une paramétrisation de la surface.

Méthodes de paramétrisation

Paramétriser une surface consiste à définir une correspondance entre la surface et un espace donné, appelé *espace des paramètres*. Ce type de procédé est utilisé dans de multiples cadres, depuis le plaquage de textures à l'extrapolation de surfaces, en passant par le remaillage ou le morphing géométrique.

Lorsque la topologie et la géométrie des surfaces sont élémentaires (cylindres, sphères, etc.), il est possible de proposer des méthodes de projections adaptées. Ces approches permettent de répondre à des problèmes divers, par exemple en imagerie médicale [96] ou en vision [89]. Dans le cas où ces méthodes de projection ne sont pas applicables, une solution plus générique doit être envisagée.

La plupart des approches classiques de paramétrisation génériques nécessitent que la surface à manipuler soit là encore d'une topologie simple. Ainsi, beaucoup de ces méthodes [43] utilisent des surfaces homéomorphes à un morceau du plan. Cependant, une approche émergente de paramétrisation alternative consiste à travailler sur la surface sans se préoccuper de sa topologie, c'est-à-dire sans appliquer de méthodes de découpage, afin de créer une *paramétrisation globale* [97, 90]. La principale difficulté rencontrée par ces approches est de contrôler et limiter la complexité des points singuliers nécessairement créés lors de la paramétrisation.

Nous allons ici nous intéresser à un type de paramétrisation plus classique, le *dépliage*, où l'on construit une relation entre la surface et un morceau du plan. Dans ce type d'approche, la surface est homéomorphe à un disque, et on construit une fonction bijective entre les deux ensembles. Deux grandes familles de paramétrisation peuvent être rencon-

trées : les méthodes à bord fixe, où l'on choisit pour chacun des points du bord de la surface la position de son représentant dans le plan, et les méthodes à bord libre, où aucune contrainte n'est imposée sur la forme du bord. La limitation majeure des méthodes à bords libres est qu'elle n'interdit pas les auto-croisements : on n'assure pas la bijection du dépliage. Cependant, les méthodes à bords fixes ne constituent pas non plus une réponse satisfaisante, car elles nécessitent une forte connaissance *a priori* sur la surface, ce qui n'est pas toujours le cas.

Dans un cadre applicatif comme celui envisagé ici, la non bijectivité du dépliage peut être contournée si elle devient trop contraignante, par exemple en ajustant le découpage. Nous avons donc choisi d'explorer les méthodes de dépliage à bord libre.

Une approche classique consiste alors à préserver ou minimiser la distorsion des angles de la surface. Il s'agit de paramétrisation *conforme*. Différents algorithmes proposent une solution à la paramétrisation conforme. Nous avons utilisé pour nos travaux une méthode exacte basée sur la résolution d'un système d'équations, et nommée Angle Based Flattening (ABF) [94]. Cette approche a récemment été revue pour être traitée de manière linéaire [112]. Nous avons également utilisé pendant ces travaux une approche quasi-conforme, nommée circle packing [20], où seule la moyenne des angles est préservée (une présentation plus précise de l'algorithme peut être lue à la section B.1). Cette approche, bien que plus lente en général, est très robuste et est une alternative intéressante aux algorithmes exacts. Elle offre également l'avantage de pouvoir être parallélisée en mémoire partagée, ce que nous avons choisi de faire à l'aide d'`OpenMP`[1] pour profiter des machines multicœur qui tendent à se démocratiser et ainsi accroître la vitesse d'exécution. La section B.3 décrit l'approche de parallélisation que nous avons explorée.

Mesure de la distorsion

Disposer d'une surface homéomorphe à un disque et plongée dans $\mathbb{R}^3$ est l'unique paramètre nécessaire aux méthodes de dépliage conforme. On peut donc produire à partir d'une telle surface une bijection avec un morceau du plan, qui préserve les angles de la variété initiale.

En observant le résultat d'une telle projection (figure 4.5(b)), on remarque que si les angles sont préservés par cette approche, il n'en est pas de même pour les distances ou les aires. Une approche classique [59] pour décrire ce phénomène consiste à calculer pour chaque triangle le ratio entre son aire dans $\mathbb{R}^2$ et celle dans $\mathbb{R}^3$. Plus ce ratio sera uniforme sur l'ensemble de la surface, moins la surface aura été déformée lors du dépliage. Ce ratio est appelé le *facteur d'étirement*[2].

Nous avons choisi d'explorer dans ce travail une mesure différente de la distorsion, basée sur la longueur des arêtes. On mesure ici la *distorsion* de chacune des arêtes en calculant le ratio de ses longueurs dans $\mathbb{R}^3$ et $\mathbb{R}^2$:

Définition 4.1 – *Soit S une surface discrète, et v_1, v_2 deux sommets de S. On note*

$$D(v_1, v_2) = \|v_1 - v_2\|_3 \left(\|v_1 - v_2\|_2\right)^{-1}$$

la distorsion *de l'arête* (v_1, v_2).

L'avantage principal de cette approche est que ce ratio peut être utilisé comme poids pour les arêtes, comme nous le verrons à la section 4.2.2.

[1] OpenMP est une librairie C/C++ et Fortran permettant l'exécution de programmes parallèles avec mémoire partagée, `http://openmp.org`

[2] *stretch factor* en anglais

(a) Extremums sur une surface de genre zéro sans bord.

(b) Extremums à une échelle plus fine sur la même surface.

FIG. 4.3 – *Extremums sur une surface de genre zéro.*

Ainsi, on dispose d'une méthode pour mesurer localement une propriété qui elle est globale. En effet, cette distorsion dépend du dépliage de l'ensemble de la surface, et non uniquement de ce comportement dans un petit voisinage.

4.2 Détection des extremums

Lorsque l'on cherche à caractériser une surface complexe, les premières informations que l'on extrait sont en général liées à la topologie. Cependant, elles n'indiquent rien sur la géométrie : un tore et une tasse ont la même topologie, alors que leur géométrie est radicalement différente, et devra être prise en compte dans de nombreux contextes. La géométrie est une information plus difficile à abstraire que la topologie. Certaines approches travaillent sur une segmentation sémantique [80], d'autres encore exploitent des informations issues de l'analyse harmonique [106]. Nous avons choisi de nous intéresser ici à la notion d'extremum, car nous l'utiliserons dans un second temps pour produire un pavage par tuile unique, tout en prenant en compte la géométrie.

Intuitivement, un extremum peut être vu comme l'extrémité d'un « bras » d'une surface. La figure 4.3(a) présente une surface de genre zéro où cinq extremums ont été mis en évidence par une méthode manuelle. Il est assez immédiat de voir que la notion d'échelle est essentielle pour caractériser les parties d'une surface qui seront considérées comme « bras » ou non. La figure 4.3(b) représente la même surface que la figure 4.3(a), en s'intéressant à des détails d'une autre échelle.

Dans cette section, nous cherchons à établir une méthode de détection de ces extremums ne dépendant pas d'un paramètre d'échelle, tout en travaillant sur la surface globale.

4.2.1 Travaux antérieurs

Différents travaux ont déjà exploré la question de la détection des extremums. Sheffer *et al.* [93] ont ainsi décrit un cadre global utilisant une étape de découpage topologique et une détection des extremums locaux. Ces points sont sélectionnés en utilisant une estimation de la courbure dans un petit voisinage. L'étape topologique ne détecte pas précisément les plus courts chemins, ni les propriétés de géométrie globale de la surface. Tierny *et al.* [103] détectent des points extremums en construisant un graphe de Reeb. Après avoir sélectionné deux points correspondant à une bonne approximation du diamètre

du graphe, une distance géodésique depuis chacun de ces points est calculée. L'intersection des extremums locaux pour chacune de ces distances constitue alors les maximums locaux de la surface globale. Cette méthode semble limitée d'une part par le bruit éventuel sur les surfaces, ainsi que par l'étape d'intersection qui est approximative et pilotée par une variable à la discrétion de l'utilisateur.

Gu *et al.* [47] utilisent quant à eux une approche globale pour détecter les extremums, en s'appuyant sur une paramétrisation à bord fixe de la surface, et en sélectionnant le triangle ayant la déformation géométrique maximale comme l'extrémité d'un chemin jusqu'au bord existant. La paramétrisation et l'étape de découpage sont alors appliquées jusqu'à stabilisation de la déformation de la paramétrisation. Au delà des multiples étapes de paramétrisation, la limitation principale de cette approche est qu'elle utilise une paramétrisation à bord fixe. Le choix de la position des points bord dans le plan n'est ni simple ni unique. De plus, la déformation des triangles est largement dépendante de la forme du bord et de la distribution des points le long de ce dernier.

D'une manière plus générale, les travaux s'intéressant à l'analyse multirésolution de surfaces [73, 7] répondent en partie à la détection de ces extremums. On peut par exemple citer Roudet *et al.* [91] qui utilisent les ondelettes comme extracteur d'information sur les propriétés locales et globales de la surface, en localisant la rugosité de la surface.

L'approche que nous décrivons dans la suite [38] apporte une réponse globale à la détection de maximums, sans pour autant ignorer la diversité de l'échelle des détails. Elle offre l'avantage de proposer un cadre général flexible où pourront être intégrées différentes propriétés liées à l'application, notamment par la modulation de la distance, ce que ne permettent pas la plupart des méthodes que nous avons évoqué ci-dessus.

4.2.2 Détection par transformation conforme

Comme nous l'avons décrit à la section 4.1.2, la mesure de distorsion des arêtes est une information portée localement qui décrit une propriété globale sur la surface. On peut donc, pour une surface homéomorphe à un disque, mesurer après dépliage la distorsion de chacune des arêtes. Cette information est uniquement géométrique, et comme nous allons le montrer dans la suite, permet de détecter les extremums de la surface.

Découpage topologique initial

Puisqu'une telle méthode nécessite que la surface soit homéomorphe à un disque, il est nécessaire de procéder dans un premier temps à un découpage topologique. Dans le cas où la surface n'est pas homéomorphe à une sphère, l'algorithme 3.2 permet de réaliser une telle transformation, en cherchant à minimiser la longueur du découpage.

Dans le cas où la surface est homéomorphe à une sphère, il est nécessaire d'envisager une autre approche. D'un point de vue topologique, il suffit de choisir deux arêtes connectées sur la surface, et découper le long de ces dernières. On obtient une surface de genre zéro à un bord. Cependant, cette méthode n'est pas satisfaisante d'un point de vue géométrique. En effet, lors d'un dépliage, la surface se retrouverait extrêmement distordue (figure 4.4(a)). On doit donc envisager une autre approche pour répondre à cette problématique.

Notre choix s'est porté sur l'utilisation d'une analyse en composantes principales (ACP), qui consiste à déterminer l'axe principal du nuage de points associé au maillage par recherche du premier vecteur propre de la matrice de covariance des données. Une fois extrait cet axe principal, deux cas sont possibles : soit l'axe intersecte la surface, et

(a) Découpage topologique naïf non satisfaisant d'un point de vue géométrique lors du dépliage.

(b) Découpage topologique par ACP intégrant une information géométrique.

FIG. 4.4 – *Découpage topologique d'une surface de genre zéro sans bord.*

on choisira alors le premier et le dernier point de la surface le long de l'axe principal (figure 4.4(b), gauche), soit ce n'est pas le cas, et fera une recherche similaire dans un petit voisinage autour de l'axe.

Une fois ces deux points sélectionnés, il ne reste plus qu'à calculer une géodésique entre ces deux points pour disposer d'un découpage topologique respectant la géométrie de la surface (figure 4.4(b), droite).

Calcul des extremums

En appliquant un tel découpage topologique, on peut donc calculer pour n'importe quelle surface un ratio de distorsion pour chacune de ces arêtes.

On formalise donc la notion d'extremum déjà évoquée à la section 4.2 de la manière suivante :

Définition 4.2 – *Soit S une surface homéomorphe à un disque, et B l'ensemble de ses points bord. On définit une fonction distance $L_b(\cdot)$ qui à un sommet de S associe la longueur (au sens de $D(\cdot,\cdot)$) du plus court chemin jusqu'à B.*

On appelle sommet extremum *sur S chaque sommet maximum local de L_b.*

D'un point de vue algorithmique, l'avantage principal de cette approche est la complexité : les maximums locaux peuvent être calculés par l'algorithme de Dijkstra en $O(n \log n)$, en ayant auparavant appliqué une étape de dépliage. Nous verrons à la section 6.2.1 qu'il est possible dans un cadre applicatif de définir une méthode de sélection de ces extremums pour ne garder que les plus significatifs, suivant le contexte.

(a) Découpage topologique initial. La surface est teinté en fonction de la distorsion.

(b) Carte de distorsion de la surface après le découpage topologique initial.

(c) Surface après découpage géométrique.

(d) Carte de distorsion de la surface après le découpage géométrique.

FIG. 4.5 – *Découpage géométrique sur une surface de genre trois sans bord.*

4.2.3 De la géométrie à la topologie

L'ensemble des algorithmes que nous avons détaillé jusqu'à présent manipulaient des notions topologiques, en travaillant par exemple avec des lacets non séparants, ou des chemins entre bords. Or, les extremums que nous avons extraits sont des informations de type géométrique. Cependant, il est ici facile de traduire ces extremums en informations purement topologiques : il suffit pour cela d'enlever virtuellement un disque au niveau de chacun des extremums. Concrètement, on décidera que ces points sont des points bord, tout comme ceux issus du premier découpage.

Il nous suffit alors d'appliquer l'algorithme 3.2 qui calcule une approximation de l'arbre de Steiner minimal reliant les bords de la surface pour obtenir un découpage de la surface respectant la géométrie.

Exemple : La figure 4.5(a) représente une surface de genre trois sans bord sur laquelle on a appliqué un premier découpage topologique. Une fois dépliée de manière conforme à l'aide de l'algorithme ABF, on obtient la carte présentée par la figure 4.5(b). On a utilisé ici pour représenter la distorsion un code de couleur : bleu pour les valeurs inférieures à 1, rouge pour les valeurs supérieures. Dans cet exemple, la valeur minimum de distorsion pour une arête est de $0,06$, et la valeur maximale est de $2,77$.

On a ensuite calculé une approximation de l'arbre de Steiner des extremums détectés et du bord, puis découpé le long des chemins détectés (figure 4.5(c)). La carte dépliée de cette nouvelle surface est nettement moins distordue, comme l'illustre la figure 4.5(d). Les valeurs extrêmes de distorsion sont sur cet exemple devenues égales à $0,28$ et $2,52$.

Les idées présentées dans cette section sont plus longuement exploitées dans le chapitre 6.

4.3 Patch des surfaces

La méthode de découpage topologique et géométrique que nous avons décrit à la section précédente est une approche générale, qui s'applique à toute surface, quelle que soit sa topologie et sa géométrie. Cependant, il arrive que l'on dispose d'informations supplémentaires sur la surface, qui peuvent être exploitées pendant cette étape de découpage.

Ainsi, lorsque les surfaces sont issues d'acquisitions automatiques, comme les isosurfaces d'images volumiques dans un contexte médical, ou les surfaces construites à partir de nuages de points scannés, la topologie des données finales n'est pas nécessairement valide comparée à la topologie de l'objet physique initial. Des artefacts, erreurs d'approximation ou limitations dus à la résolution de l'acquisition peuvent entraîner des jonctions locales des surfaces. Nous décrivons ici une méthode permettant de corriger ces variations, en utilisant une approche topologique guidée par la géométrie.

4.3.1 Travaux similaires

Hong *et al.* [52], proposent une méthode assez semblable à la nôtre [39], appliquée au dépliage virtuel de surfaces de colon. L'approche explorée par Hong utilise les travaux de Colin de Verdière et Lazarus [19] pour modifier la topologie, en construisant pour chaque classe d'homotopie de la surface le plus court lacet représentant. Cependant leur approche comporte un défaut important, que nous résolvons par la solution présentée ici. En effet, au lieu de rechercher les plus courts lacets tels qu'ils le laissent supposer dans leur analyse, leur approche consiste à calculer les plus courts lacets dans une classe d'homotopie donnée, à point base fixé. Les illustrations qu'ils donnent ne laissent pas supposer de cette erreur d'appréciation, mais dans les faits, le risque peut être d'avoir de très longs lacets si le point base est éloigné de la petite erreur topologique que l'on cherche à corriger.

D'autres travaux utilisant des méthodes différentes ont déjà été présentées. L'une des approches couramment utilisées consiste à manipuler des données volumiques plutôt que surfaciques. On peut par exemple citer [113], qui utilise une squelettisation de l'objet à corriger, ou encore [58] qui travaille sur une propagation de front sur les voxels. Cependant ces méthodes ne peuvent manipuler que des objets fermés, et ne peuvent être influencées par des informations géométriques comme la courbure locale de la surface.

Il existe également des méthodes surfaciques alternatives, comme [111] qui cherche et supprime les petits tunnels, mais ne peut exploiter les informations géométriques des surfaces, comme les distances géodésiques ou la courbure locale.

4.3.2 α-jonctions

Les surfaces dont l'acquisition a été réalisée avec des imprécisions comportent des erreurs topologiques. Puisque ces variations dépendent du mode d'acquisition et de l'application envisagée, et puisque l'on ignore en général la topologie des surfaces réelles, nous avons choisi de décrire ces jonctions à l'aide d'un paramètre $\alpha \in \mathbb{R}^+$ que nous laisserons dans le cas applicatif à la discrétion de l'utilisateur.

On définit donc les jonctions incorrectes de la manière suivante :

(a) Site d'une erreur topologique. (b) Détection de l'α-jonction. (c) Découpage du lacet. (d) Fermeture des bords.

FIG. 4.6 – *Schéma d'une α-jonction et de sa correction topologique*

Définition 4.3 – *Soit S une surface discrète, et $\alpha \in \mathbb{R}^+$. On dit qu'un lacet γ est une α-jonction s'il s'agit d'un lacet non séparant de longueur inférieure à α, et qu'il est le plus court de sa classe d'homotopie.*

On remarque ici que d'un point de vue formel, une α-jonction peut ne pas être unique dans sa classe d'homotopie, si deux lacets ont la longueur minimale. Dans ce cas, il n'existe pas de manière de déterminer la localisation exacte de la jonction. Cependant, en pratique ces situations sont rares, et on choisira l'un des α-jonctions d'une même classe d'homotopie sans s'arrêter sur cette non unicité.

La figure 4.6(b) illustre par un schéma l'existence d'une α-jonction sur une surface synthétique.

4.3.3 Simplification par ajustement de la topologie

En partant du principe que ces jonctions soient incohérentes d'un point de vue topologique, nous présentons ici une approche automatique permettant de modifier localement la surface afin de corriger ces variations. Le principe général est simple, il consiste à ajuster la topologie au niveau des α-jonctions pour réduire le genre de la surface. Pour cela, on envisage d'appliquer l'algorithme suivant pour chaque α-jonction :

Algorithme 4.1 : Correction topologique d'une α-jonction

Données : un 2-maillage $\mathcal{M}$, un lacet γ non séparant
Résultat : Un 2-maillage $\mathcal{M}$ ajusté en γ
Découper $\mathcal{M}$ selon γ;
pour chaque *β bord de $\mathcal{M}$ créé par le découpage* **faire**
  Ajouter à $\mathcal{M}$ un nouveau sommet s isobarycentre des sommets de β;
  pour chaque *arête (s_1, s_2) de β* **faire**
    Ajouter un triangle (s_1, s_2, s) et les arêtes manquantes à $\mathcal{M}$;
  fin
fin

Comme l'illustre la figure 4.6, le principe de cet algorithme est de découper la surface suivant le lacet donné, puis d'ajouter deux éléments de surface (ou petits disques) pour combler les bords ainsi créés.

En conjuguant cette correction topologique à la recherche du plus court lacet non séparant décrit à la section 3.3.2, ou à ses variations optimisée (section 3.3.3) et approchée (section 3.3.4), on dispose d'un outil complet permettant de détecter successivement chacune des α-jonctions d'une surface donnée, en appliquant après chaque détection la correction topologique correspondante.

(a) Surface de genre un sans bord.

(b) Correction topologique en utilisant une distance euclidienne pour la recherche du plus court lacet non séparant.

(c) Correction topologique en utilisant une distance intégrant la courbure locale comme information géométrique (section 4.1.1).

FIG. 4.7 – *Corrections topologiques utilisant l'algorithme 4.1. Les surfaces ont été lissées après correction pour permettre un meilleur aperçu des corrections apportées.*

À noter qu'ici on peut encore optimiser l'outil de recherche de plus court lacet en tronquant la recherche pour ignorer les lacets plus long que α. D'un point de vue pratique, on remarque qu'à chaque correction topologique, on diminue fortement la taille de l'ensemble des points base servant à calculer le plus court lacet non séparant. Ainsi, la complexité effective de cet algorithme est meilleure que celle de l'algorithme 3.2, le nombre de plus courts lacets à explorer se réduisant à chaque correction topologique.

La figure 4.7 présente le résultat de la correction topologique appliquée à une surface de genre un sans bord, en utilisant d'abord une distance euclidienne (figure 4.7(b)), puis en utilisant comme information géométrique supplémentaire la courbure locale (figure 4.7(c)), comme nous l'avons décrit à la section 4.1.1.

4.4 Conclusion

Après avoir introduit au chapitre précédent les premiers éléments structurants des outils de pavage explorés dans ce manuscrit, nous avons présenté dans le chapitre courant quelques adaptations et ajustements envisageables dès lors que l'on intègre des notions de géométrie au cœur des méthodes.

Nous avons ainsi remarqué que l'adaptation de la mesure de distance sur les surfaces était un moyen local d'ajustement du comportement global des algorithmes. Puis nous avons proposé une méthode permettant d'introduire de nouveaux découpages, cette fois guidés par la géométrie, en profitant des possibilités offertes par une détection globale de propriétés géométriques multiéchelle. Enfin, nous avons décrit un processus de correction d'erreurs topologiques, en corrigeant la surface par patches afin d'en réduire le genre au besoin.

Chapitre 5
Contraintes sur les tuiles

Sommaire

5.1 Considérations générales 72
5.2 Découpage en cylindres 73
5.2.1 Étude combinatoire 73
5.2.2 Plus courts n-cets 75
5.2.3 Pavage par n-cets 76
5.3 Nombre de sommets d'une tuile unique 82
5.3.1 Contrôle du nombre de 0-m-cellules 82
5.3.2 Découpage en polygone minimal 83
5.4 Découpage en quadrangles 84
5.4.1 Travaux existants 84
5.4.2 Simplification de tuile unique 85
5.4.3 Simplification de pavage en cylindres 88
5.5 Conclusion 90

Dans les deux chapitres précédents, nous avons décrit comment la topologie et la géométrie d'une surface déterminent les choix de découpage. Ainsi, dans le chapitre 3, nous avons exploré le découpage d'une surface donnée en une surface homéomorphe à un disque. Puis dans le chapitre 4 nous avons décrit l'influence que pouvait avoir la géométrie dans un tel découpage, en introduisant des contraintes liées à la forme des surfaces manipulées.

Cependant, dans les deux cas, le pavage produit est composé d'une seule tuile à la topologie triviale, et on ne contrôle pas le nombre d'arêtes bordant la m-cellule associée. Dans le présent chapitre, nous nous intéressons à des problèmes moins classiques, où l'on impose des propriétés sur les tuiles du pavage.

Lorsqu'il s'agit de contrôler le nombre d'arêtes d'un pavage en tuile unique polygonale, plusieurs travaux ont répondu à cette question en construisant un système de lacets basé en un unique point, soit en imposant une contrainte d'homotopie [19], soit en relâchant cette contrainte [35].

D'autres travaux ajoutent une contrainte sur la forme des tuiles, en imposant à ce que les bords de ces dernières soient composés de quatre arêtes, et homéomorphes à un disque. Il s'agit ici de construire une quadrangulation de la surface. De nombreuses approches ont été envisagées, menant pour la plupart à des quadrangles en nombre important dont

(a) Le plus court chemin entre a et e emprunte le bord de la surface. Le polygone résultant $ebcd$ est pincé en e.

(b) En interdisant les arêtes bords, on produit des tuiles topologiquement correctes, car 2-variétés.

FIG. 5.1 – *Chemin produit après découpage des surfaces pincées.*

la taille n'est pas maîtrisée. On peut par exemple citer [30] qui propose une approche spectrale, ou [1] qui utilise un opérateur de Laplace discret pour contrôler les points de singularité. Nous détaillerons dans la section 5.4 les méthodes de quadrangulation.

Les travaux qui s'intéressent à des tuiles non disques sont assez rares. On peut cependant citer [49] qui propose des résultats combinatoires sur le pavage de surface par pantalons.

Dans ce chapitre, nous commençons par évoquer quelques considérations générales, nécessaires lorsque l'on s'intéresse à la topologie du complémentaire d'un découpage, puis nous montrons comment l'utilisation des n-cets est une réponse intéressante au pavage de surfaces en cylindres. Nous présentons ensuite une méthode permettant de contrôler partiellement le nombre de 0-m-cellules composant le pavage par un polygone, puis nous proposons deux approches de quadrangulation de surfaces s'appuyant sur des méthodes de découpage.

5.1 Considérations générales

La définition de découpage tel que nous l'avons présentée précédemment ne permet pas de contrôler précisément la topologie des tuiles, notamment lorsque les chemins se croisent ou empruntent une même arête. En effet, dans ce cas il arrive que les composantes connexes du complémentaire soient pincées (figure 5.1(a)). Plusieurs solutions sont possibles pour résoudre cette limitation. Colin de Verdière a décrit assez finement [17] l'une des approches envisageables. Elle consiste à modéliser un chemin en décrivant non seulement les arêtes qu'il emprunte, mais aussi la distribution des chemins qui empruntent une même arête.

Nous avons pris ici le parti de choisir une approche alternative, efficace dans un contexte applicatif. Cette approche consiste à choisir, tant que la chose est possible, les chemins ne passant pas par des points bord. Nous avons ainsi décrit dans la section 2.3.2 une modification simple de la longueur des arêtes pour imposer un tel comportement. La figure 5.1(b) illustre la conséquence de l'utilisation d'une telle distance.

Cependant, il arrive que l'on souhaite tout de même relier deux sommets, et que la seule solution soit d'emprunter une arête bord, ou une arête déjà empruntée par un autre chemin. Dans ce cas, nous avons opté pour le raffinement du maillage au voisinage de ces sites particuliers. On commence donc par calculer un chemin entre les deux sommets à relier sans interdire de passer par les bords. On prend soin de donner à chaque arête bord une longueur supérieure à la longueur de la plus grande arête du maillage, afin de favoriser

(a) Chemin initial contenant un point bord b non évitable.

(b) Étapes de raffinement des arêtes autour du point bord décollant le chemin du bord.

FIG. 5.2 – *Processus de raffinement de maillage pour le décollage des bords.*

les chemins internes. Si malgré cette contrainte le chemin calculé passe par un bord, alors on raffine itérativement le maillage en subdivisant les arêtes autour d'un sommet bord b emprunté. La figure 5.2 illustre ce raffinement local. On recalcule alors un chemin en interdisant le point b. On remarque ici que d'une part, il n'est pas nécessaire de recalculer un plus court chemin sur l'ensemble du maillage, il suffit de recalculer uniquement la fin du chemin. D'autre part, cette procédure n'introduit pas de nouveaux points bord dans le chemin, ce qui implique la convergence.

5.2 Découpage en cylindres

Si la tuile la plus classiquement rencontrée dans les travaux sur les pavages est la tuile homéomorphe à un disque, le cylindre est une autre topologie naturelle. En effet, ces tuiles capturent les sections « tubulaires » des surfaces, éléments courants dans de nombreux cas. Dans cette section, nous nous intéressons donc au découpage en cylindres de surfaces quelconques, en utilisant comme outils de découpages les n-cets, car ils offrent à nos yeux à la fois une approche flexible et bien formalisée. Nous verrons ainsi à la section 5.4.3 que ces pavages disposent de bonnes propriétés pour produire un pavage en quadrangles satisfaisant, notamment en terme de régularité géométrique et combinatoire.

Dans un premier temps, nous proposons ici une analyse combinatoire des pavages en cylindres, puis nous décrivons une méthode applicative permettant de produire un pavage en cylindres par 3-cets de faible longueur.

5.2.1 Étude combinatoire

Théorème 5.1 – *Soit une surface S connexe de genre g à n bords. On peut décomposer S en $n+g$ cylindres en découpant S suivant un c-cet valide, avec $c = n + 2g$.*

Preuve : Commençons par montrer que $c = n+2g$. Pour cela, on utilise la caractéristique d'Euler généralisée (équation 2.7). Chacune des tuiles étant un cylindre, leur genre g_i est nul et leur nombre de bord b_i est égal à deux. De plus, ce pavage ne comporte que deux

sommets (les points base du c-cet), aussi on a $S = 2$. On arrive donc naturellement à l'égalité $A_b = 2g + b$, où A_b est le nombre d'arêtes à bord du m-complexe cellulaire. On a donc bien $c = n + 2g$ car toutes les arêtes à bord du pavage sont des chemins du c-cet.

Intéressons-nous maintenant au nombre de cylindres composant ce pavage. Il y a deux types de cylindres : les cylindres qui possèdent une arête cyclique correspondant à un bord de S, et les cylindres possédant quatre arêtes correspondants à des chemins du c-cet. Les cylindres bord sont évidemment au nombre de n. Déterminons le nombre de cylindres F_{nb} non bord de ce pavage.

Commençons par remarquer qu'en refermant chacun des bords de S à l'aide d'un disque, on obtient une surface de genre g sans bord, que l'on peut paver à l'aide d'un $2g$-cet, par F_{nb} cylindres. Chacun de ces cylindres est bordé de quatre arêtes, et chaque arête participe à deux cylindres. On a donc immédiatement $F_{nb} = g$.

Le nombre de cylindres composant un pavage tel que décrit dans le théorème est donc $n + F_{nb} = n + g$. □

Exemple : La figure 5.3(a) présente un découpage en trois cylindres par un 6-cet d'une surface de genre 3.

Si d'un point de vue combinatoire, un tel découpage semble pertinent, il l'est nettement moins d'un point de vue pratique : n'utiliser que deux sommets entraînera des tuiles très distordues. On s'intéresse alors naturellement au découpage de surface par 3-cets multiples.

Théorème 5.2 – *Soit S une surface de genre g à n bords, avec $g > 1$ ou $g = 1$ et $n > 1$. On peut décomposer S en $2n + 3(g - 1)$ cylindres en découpant S suivant $2(g - 1) + n$ 3-cets valides d'intersection vide.*

Pour montrer ce résultat, on utilise le lemme suivant :

Lemme 5.1 – *Soient S une surface de genre g à n bords et $\mathcal{C}$ un c-cet valide sur S, avec $c > 3$. Supposons de plus que $\mathcal{C}$ décompose S en cylindres. Alors il existe un $(c - 1)$-cet $\mathcal{C}_1$ et un 3-cet $\mathcal{C}_2$ valides d'intersection vide tels que le découpage de S suivant $\mathcal{C}_1 \cup \mathcal{C}_2$ soit homéomorphe à $✂(S, \mathcal{C}) \cup S_c$, où S_c est un cylindre.*

Preuve : Le lemme est immédiat si l'on procède de la manière suivante : après avoir choisi un chemin γ_0 composant le c-cet original, on appelle γ_{-1} et γ_1 les deux chemins qui constituent chacun avec γ_0 la description d'un bord d'un cylindre. La décomposition décrite dans le lemme consiste à dupliquer les deux chemins γ_{-1} et γ_1 pour insérer un nouveau cylindre sur le pavage. □

La figure 5.3(b) présente un exemple de cette décomposition.

Preuve du théorème 5.2 : On utilise pour cela le théorème 5.1 qui indique qu'un découpage en cylindres d'une surface de genre g à n bords à l'aide d'un c-cet implique que $c = n + 2g$, et que le nombre de cylindres est égal à $n + g$. On utilise ensuite le résultat du lemme 5.1, en décomposant le c-cet $c - 3$ fois. On obtient donc $c - 2 = 2(g - 1) + n$ 3-cets, tout en ajoutant $c - 3$ cylindres aux $n + g$ initiaux, soit un total de $2n + 3(g - 1)$ cylindres. □

La figure 5.3(c) illustre une telle décomposition.

(a) Surface de genre 2 pavée par des cylindres aux bords décrits par un 6-cet.

(b) La même surface après une étape de décomposition du 6-cet et ajout d'un cylindre.

(c) La même surface après trois étapes de décomposition du 6-cet. La surface de genre trois est pavée en cylindres grâce à quatre 3-cets.

FIG. 5.3 – *Décomposition d'un* 6*-cet sur une surface de genre* 3 *avec ajout de cylindres.*

5.2.2 Plus courts n-cets

Les 3-cets constituent donc un outil de découpage intéressant pour produire un pavage en cylindres. Cette propriété peut d'ailleurs être généralisée pour des m-cets, avec $m > 2$.

Nous proposons dans cette section un algorithme issu d'une collaboration avec Éric Colin de Verdière, et permettant de calculer un m-cet valide sur une surface, étant donnés les deux points base. Si la méthode que nous proposons ici permet bien de calculer le plus court 2-cet valide, pour $m > 2$, elle calcule un m-cet valide, mais qui n'est pas nécessairement le plus court. Cependant, lorsque m est petit et que les surfaces manipulées ont des géométries nettement non planaires, on calcule bien le plus court m-cet. Les exemples à la fin de cette section permettront de le vérifier.

Théorème 5.3 – *Étant donnés un* 2*-maillage* $\mathcal{M}$ *à* n *sommets, et deux sommets* b_0 *et* b_1 *de* $\mathcal{M}$*, on peut construire un* m*-cet valide sur* $\mathcal{M}$ *basé en* (b_0, b_1) *en* $O(n \log n)$*, où* $m-1$ *est inférieur ou égal au nombre de chemins composant le cut locus élagué de* b_0 *et reliant des sommets d'arité au moins 3.*

Preuve : Montrons que l'algorithme 5.1 calcule bien un m-cet valide en $O(n \log n)$.

Tout d'abord, d'un point de vue complexité, il est immédiat de voir qu'il suffit de deux calculs de plus courts chemins, le premier depuis b_0 pour calculer le cut locus C, et le second depuis b_1 dans le complémentaire de C pour déterminer la seconde partie de chacun des chemins qui seront calculés.

Ensuite, il suffit de remarquer qu'aucune paire de chemins ainsi calculés ne sera homotope, de par la construction du cut locus. □

Si pour m quelconque on ne peut rien dire en général, pour $m = 2$ on a la propriété suivante :

Propriété 5.1 – *Soit* $\mathcal{M}$ *un* 2*-maillage, et deux sommets* b_0 *et* b_1 *de* $\mathcal{M}$*. Alors le* 2*-cet valide calculé par le théorème 5.3 est le plus court* 2*-cet valide basé en* (b_0, b_1)*.*

La preuve de ce théorème est adaptée d'une preuve proposée par Éric Colin de Verdière lors d'une discussion autour de la construction de plus courts 2-cets.

Algorithme 5.1 : Calcul de m-cet valide

Données : $\mathcal{M}$ un 2-maillage, b_0 et b_1 deux sommets distincts de $\mathcal{M}$, $m \in \mathbb{N}$
Résultat : (γ_i) un m-cet sur $\mathcal{M}$
$C \leftarrow$ cut locus associé à b_0;
Simplifier C en supprimant toutes les parties arborescentes;
$p \leftarrow$ plus court chemin de b_0 à b_1 qui ne croise pas C ;
$(\gamma_i) \leftarrow (\gamma_i) \cup \{p\}$;
pour chaque *Chemin c de C entre deux sommets d'arité au moins* 3 **faire**
  $p \leftarrow$ plus court chemin entre b_0 et b_1 croisant C en c;
  $(\gamma_i) \leftarrow (\gamma_i) \cup \{p\}$;
fin
$n_{(\gamma_i)} \leftarrow$ nombre de chemins de (γ_i);
si $n_{(\gamma_i)} \geq m$ **alors**
  Supprimer les $n_{(\gamma_i)} - m$ plus longs chemins de $n_{(\gamma_i)}$;
sinon
  Erreur;
fin

Preuve : On commence par remarquer que le premier chemin α_0 construit par l'algorithme et reliant b_0 à b_1 est le plus court chemin reliant ces deux points car il ne croise pas C.

Le second chemin α_1 construit est bien non homotope à α_0, par construction du cut locus. Montrons maintenant que α_1 est le plus court chemin reliant b_0 à b_1 et non homotope à α_0. Il suffit de remarquer pour cela que tout chemin non homotope à α_0 croise au moins une fois le cut locus. Si un chemin α_i reliant b_0 à b_1 le croise plusieurs fois, alors on peut construire un nouveau chemin $\alpha_{i'}$ plus court, en choisissant le point de croisement de C et α_i le plus proche de b_0, et en construisant le chemin le plus court reliant b_0 à b_1, et croisant C en cet unique point. Enfin, puisque c'est le plus court chemin de ce type qui a été retenu, alors on a bien le chemin non homotope à α_0 le plus court.

□

Si cette méthode de calcul de n-cets valides n'assure pas la construction du plus court n-cet dans le cas général, les exemples suivants permettent de constater que si les deux points base sont placés de chaque côté d'un pantalon, alors c'est bien le plus court n-cet qui est calculé. Nous proposons dans la section 5.2.3 une méthode d'ajustement par optimisation permet de converger naturellement vers des points base ayant de telles propriétés.

Ainsi, la figure 5.4 illustre le calcul d'un 3-cet sur un pantalon à partir de deux points base intuitivement satisfaisants. Cette configuration en pantalon constitue l'une des formes les plus répandues dans les modèles 3D que nous allons traiter par la suite.

5.2.3 Pavage par n-cets

Si l'on souhaite réaliser un pavage de la surface, il est nécessaire de mettre en place une méthode pour déterminer la position des points base des différents n-cets de découpage.

En nous inspirant de méthodes issues de l'ingénierie, nous proposons ici une approche

(a) Cut locus calculé à partir de b_0, premier point base du futur 3-cet.

(b) Construction des trois plus courts chemins entre b_0 et b_1 suivant l'algorithme 5.1.

FIG. 5.4 – *Calcul d'un 3-cet valide sur un pantalon.*

globale permettant d'approximer un découpage en 3-cets en explorant l'ensemble des 3-cets possibles, et en minimisant un critère modélisant la qualité d'un 3-cet.

Remarque : Comme nous l'avons vu avec le théorème 5.2, paver une surface à l'aide de 3-cets impose que la surface soit au moins de genre deux, ou de genre un avec au moins un bord. Dans la suite de cette section, nous nous intéresserons donc uniquement à ces configurations.

Critère de qualité d'un n-cet

Nous avons exploré différentes pistes pour décrire la qualité d'un n-cet. Cependant, si d'un point de vue intuitif cette description qualitative est évidente, la formalisation de cette intuition le semble moins. Nous avons donc choisi de retenir comme mesure la longueur des chemins de découpage, tel que déjà utilisé dans la section 3.3.

Si en pratique nous n'avons travaillé qu'avec des 3-cets, la méthode générale décrite dans les sections suivantes peut être appliquée à tout type de n-cet. Pour affiner cet outil lors de la comparaison de n-cets d'arités différentes, on pourra également utiliser la moyenne des longueurs des n chemins qui composent le n-cet, afin d'avoir une grandeur comparable. Dans le cas de deux n-cets de même arité, on aura ainsi un ordre partiel identique à la mesure s'appuyant uniquement sur les longueurs.

Recuit simulé

Le recuit simulé est une méthode d'optimisation originellement décrite en 1983 par Kirkpatrick, Gelatt et Vecchi [61]. Elle s'inspire d'un processus de fabrication métallurgique, et a pour principal objectif de faire converger le système modélisé vers le minimum d'une fonction énergie donnée, en évitant la convergence vers les minimums non globaux. L'élément original de ce processus est d'introduire une variable de température, qui simulera le refroidissement du système vers une solution satisfaisante.

Ainsi, à chaque pas du processus itératif, on s'appuie sur la dernière solution s_i trouvée, en choisissant aléatoirement parmi les états voisins une nouvelle réponse possible s_{i+1}. Si s_{i+1} est meilleure que s_i, alors on la conserve. Dans le cas contraire, une méthode aléatoire[1] inspirée de la thermodynamique est utilisée pour retenir ou non cette solution moins pertinente. C'est précisément ce dernier point qui permet au système de ne pas converger uniquement vers des minimums locaux. Pour peu que le problème modélisé

[1]Dans les faits, on utilisera $\text{rand}_u()$, un tirage uniforme pseudo-aléatoire dans $[0, 1]$ et une règle de Métropolis.

dispose de bonnes propriétés, on montre la convergence de cette méthode [78, 64], en étudiant la distribution d'une chaîne de Markov associée.

L'algorithme 5.2 décrit plus précisément le fonctionnement du recuit simulé tel que nous l'utilisons dans ces travaux. L'ensemble $\mathcal{E}$ des états possibles de notre problème est constitué des 3-cets sur une surface donnée qui auront été calculés à partir de leurs points base par l'algorithme 5.1. L'énergie à minimiser est définie par le critère de qualité décrit précédemment. Le système de voisinage structurant l'ensemble des 3-cets d'une surface est quant à lui précisé plus bas.

Algorithme 5.2 : Optimisation par recuit simulé

Données : Température initiale T_{init}, température minimum T_{min}, nombre d'itérations par température N_{step}, coefficient multiplicateur T_{mul}, valeur d'acceptation A_{min}, ensemble des solutions possibles S, fonction énergie $E : S \to S$, solution initiale s_0

Résultat : Solution minimale s

```
s ← s_0;
e = E(s);
T ← T_0;
répéter
    A ← 0;
    pour n ∈ [[0, N_step]] faire
        s_n ← voisin(s);
        e_n = E(s_n);
        if e_n < e ou rand_u() < e^{-(e_n - e)/T} then
            s ← s_n;
            e ← e_n;
            A ← A + 1;
        end
    fin
    A ← A/N_step;
    T ← T · T_mul;
jusqu'à T ≤ T_min ou A < A_min ;
```

Les différents paramètres de la méthode constituent des éléments qui peuvent être ajustés afin d'en améliorer la convergence. Ainsi, on laisse à l'utilisateur le soin de choisir la température initiale, le nombre d'itérations par valeur de température, ou encore le coefficient multiplicateur décrivant la transition thermique. Les conditions d'arrêt sont également laissées à la discrétion de l'utilisateur, comme la température minimale ou le pourcentage d'acceptation à atteindre avant l'arrêt.

Le théorème de Hajek [48] assure la convergence théorique de la méthode, en décrivant le comportement à l'infini de la loi de probabilité des minimums globaux lorsque le système suit une température décroissante.

Plusieurs définitions sont nécessaires à son énoncé :

Définition 5.1 – *On dit deux états e_1 et e_2 de l'ensemble des états possibles $\mathcal{E}$ sont* communiquants *à hauteur de $\hbar$ si $e_1 = e_2$ avec $E(e_1) < \hbar$ ou s'il existe une séquence d'états $e_1 = e_1^1, e_1^2, ..., e_1^n = e_2$ telle que $E(e_1^i) < \hbar$ et telle que e_1^{i+1} appartienne au voisinage de e_1^i.*

On caractérise alors le système en terme de réversibilité :

Définition 5.2 – *On dit qu'un système vérifie la propriété de* faible réversibilité *si pour tout $\hbar > 0$, et toute paire d'états $(e_1, e_2) \in \mathcal{E} \times \mathcal{E}$, e_1 est atteignable à hauteur de $\hbar$ depuis e_2, si et seulement si e_2 est atteignable à hauteur de $\hbar$ depuis e_2.*

Enfin, la notion d'états communiquants permet de définir celle de profondeur :

Définition 5.3 – *On définit la profondeur d'un état e comme la plus petite valeur $\hbar > 0$ telle qu'il existe un état e_2 avec $E(e_2) < E(e)$ qui communique avec e à hauteur de $V(e) + \hbar$.*

On énonce alors le théorème de Hajek de la manière suivante, la preuve étant donnée dans [48] :

Théorème 5.4 – *Soit $(T_n)_{n\in\mathbb{N}}$ une suite de réels positifs décroissants tels que $\lim_{n\to\infty} T_n = 0$, et supposons que le système de voisinage défini sur les états vérifie la propriété de faible réversibilité. Soit $(s_i)_{i\leq 0}$ la suite de solutions choisies. Soit $\mathcal{S}^*$ l'ensemble des minimums globaux. Alors :*

$$\lim_{k\to\infty} P\left[s_k \in \mathcal{S}^*\right] = 1$$

si et seulement si :

$$\sum_{k=1}^{\infty} e^{\frac{-d^*}{s_k}},$$

où d^* est la profondeur maximale des minimums locaux non globaux.

Si la convergence théorique de la méthode est assurée par ce théorème, la complexité de l'algorithme ne peut être mesurée aisément. On peut cependant noter que la complexité de calcul d'un tricet est en $O(n \log n)$, où n est le nombre de sommets du maillage, comme nous l'avions déjà remarqué dans la preuve du théorème 5.3. Quelques temps de calcul effectifs sont rapportés aux sections 5.2.3 et 6.3.2.

Choix des voisins

La qualité de la convergence du recuit simulé dépend de plusieurs facteurs, et notamment de la méthode de choix des n-cets voisins d'un n-cet donné.

Le système de voisinage que nous proposons s'énonce comme suit : deux n-cets sont voisins si leurs points base sont deux à deux voisins sur la surface discrète. On profite ainsi de l'ensemble des possibilités offertes par la structure de maillage triangulaire, comme les cliques par exemple. De plus, l'algorithme de calcul de n-cet valide basé en deux points (section 5.2.2) permet de construire facilement un voisin d'un n-cet donné.

Afin d'affiner le fonctionnement de notre algorithme, nous proposons de plus quelques heuristiques permettant de rapidement converger vers des n-cets que nous savons satisfaisants.

Dans un premier temps, nous avons remarqué que les n-cets équilibrés, c'est-à-dire dont les chemins avaient des longueurs proches, semblaient plus satisfaisants pour notre application. Nous avons donc choisit d'introduire une heuristique initiale dans l'algorithme général, lorsqu'un chemin composant un n-cet est nettement plus court ou plus long que la moyenne des longueurs. Dans ce cas, on choisi non plus au hasard le prochain n-cet dans l'algorithme, mais on déplace les points base suivant le chemin atypique, en le

(a) Choix des nouveaux points base en utilisant un chemin trop long.

(b) Choix des nouveaux points base en utilisant un chemin trop court.

FIG. 5.5 – *Recherche de nouveaux points base à partir d'un 3-cet en utilisant la longueur des chemins existants.*

raccourcissant lorsque l'on constate sa grande longueur, ou à l'inverse en le rallongeant si l'on constate sa faible longueur. La figure 5.5 illustre une telle construction.

Une autre heuristique, plus globale, et entraînant de grands sauts dans les voisinages des n-cets, consiste à calculer les nouveaux points base en exploitant plus globalement le n-cet courant. Le principe permet là aussi d'homogénéiser la taille des chemins. Pour un n-cet γ donné, on utilise les milieux géodésiques $(m_i)_{1\leq i\leq n}$ des chemins de γ. On calcule alors pour chaque point p du maillage la fonction $f_{(m_i)_{1\leq i\leq n}}$ correspondant à la somme des distances géodésiques entre p et chacun des m_i. On retient alors les minimums locaux de $f_{(m_i)_{1\leq i\leq n}}$ les plus proches des deux points base initiaux, et on dispose ainsi de deux nouveaux points base qui permettront la construction d'un nouveau n-cet.

Pavage en cylindres

La figure 5.6(a) présente le résultat de l'algorithme 5.2 sur une surface de genre zéro à trois bords : on obtient bien un pavage en cylindres, généré par le découpage de la surface suivant un 3-cet court.

Dans le cas de surfaces à la topologie plus complexe, nous avons choisi d'appliquer itérativement cette méthode jusqu'au pavage final par cylindres. L'algorithme 5.3 synthétise ce processus, et les figures 5.6(b) et 5.6(c) présentent un résultat de pavage en cylindres sur des surfaces de genre deux et quatre sans bord.

Algorithme 5.3 : Pavage en cylindres par 3-cets

Données : $\mathcal{M}$ un 2-maillage de genre > 1
Résultat : $\mathcal{M}$ un 2-maillage découpé en cylindres
tant que $\mathcal{M}'$ *composante connexe de* $\mathcal{M}$ *non cylindrique* **faire**
  $c \leftarrow$ 3-cet calculé par recuit simulé sur $\mathcal{M}'$;
  Découper $\mathcal{M}$ suivant c;
fin

Une piste que nous n'avons pas exploré mais qui semble intéressante serait de faire varier le nombre de chemins composant le n-cet. Ainsi, en utilisant la mesure de qualité décrite à la section 5.2.3, l'algorithme 5.2 permettrait également d'optimiser le nombre de chemins suivant la surface manipulée. Par exemple, dans l'exemple de la figure 5.6(c), les deux 3-cets qui se situent au centre de la figure auraient pu être remplacés par un unique 4-cet, produisant ainsi un découpage plus satisfaisant.

Résultats et perspectives

La complexité théorique de l'algorithme de recuit simulé est difficile à évaluer.

(a) Résultat de l'algorithme 5.2 sur une surface de genre zéro à trois bords.

(b) Résultat de l'algorithme 5.3 sur une surface de genre deux sans bord.

(c) Résultat de l'algorithme 5.3 sur une surface de genre quatre sans bord.

FIG. 5.6 – *Pavages en cylindre de surfaces par* 3*-cets et recuit simulé.*

Cependant, à titre d'exemple, le temps de calcul nécessaire au calcul du découpage du simple pantalon composé de 1 679 points (figure 5.6(a)) a nécessité $1,8$ secondes de calcul sur notre machine de travail, le découpage du double tore à 1 474 points (figure 5.6(b)) a nécessité 4 secondes, quand le découpage du bretzel composé de 2 080 points (figure 5.6(c)) a pris onze secondes. Nous avons également repris la surface du bretzel, en densifiant la surface pour obtenir 8 562 points. Le découpage de cette nouvelle surface a alors nécessité 85 secondes.

Pour ces exemples, les différents paramètres de la méthode ont été ajustés de la manière suivante : la température initiale était réglée à 2, chaque étape nécessitait 20 itérations, et le coefficient multiplicateur de décroissance de la température était de $0,4$. En ajustant ces valeurs, il est possible de diminuer le temps d'exécution du traitement sans pour autant réduire la qualité du découpage. Ainsi, nous avons repris la surface du bretzel, cette fois-ci en densifiant le maillage pour obtenir 51 402 points, et nous avons légèrement réduit la température initiale à 1, ainsi que le coefficient multiplicateur à $0,2$. On obtient alors un découpage proche de celui présenté à la figure 5.6(c) en 11 minutes contre 18 minutes avec les réglages initiaux.

Dans la section 6.3, des résultats complémentaires sont donnés dans le cas où l'algorithme de pavage en cylindres est utilisé pour répondre à une question de segmentation de surfaces organiques. On y trouvera notamment une illustration de la stabilité au bruit de la méthode.

Il est clair que ces temps de calculs ne seront pas satisfaisants pour toutes les applications. Cependant, différentes pistes d'amélioration peuvent être envisagées. Tout d'abord, une approche multi-échelle semble être envisageable. En effet, appliquer l'algorithme de recuit simulé sur un maillage au faible nombre de sommets permettrait d'extraire rapidement une région où la recherche serait ensuite menée sur le maillage complet.

Une remarque plus globale pourrait également entraîner une révision complète de l'approche décrite dans cette section. Ainsi, on remarque une grande similitude dans la structure générale d'un découpage en cylindres par n-cets et celle d'un graphe de Reeb qui aurait été construit à partir de la même surface. En effet, si l'on construit un graphe où les sommets seraient les n-cets du pavage, et les arêtes les cylindres reliant deux n-cets, alors on obtient une description topologique complète de la surface. Est-il alors possible d'aider au positionnement des n-cets en s'appuyant sur la construction préalable d'un graphe de Reeb de la surface ? Cette question nous semble intéressante, et laisse entrevoir

une autre approche à la construction des n-cets, en travaillant sur les surfaces cylindriques complémentaires plutôt que directement sur les chemins.

5.3 Nombre de sommets d'une tuile unique

Lorsque nous avons décrit dans la section 3.3 le problème du découpage minimal, nous nous sommes principalement concentrés sur la longueur du découpage. La méthode que nous avons décrite pour répondre à cette question ne permet pas de contrôler le nombre de sommets de la tuile résultante. Or, comme nous le verrons par la suite (section 5.4), il peut être intéressant de maîtriser ce nombre, pour produire des pavages plus spécifiques.

Nous proposons donc dans cette section une approche permettant de maîtriser le nombre de sommets du pavage résultant, en contraignant le pavage sur une surface de genre g à n bords à être un polygone à $g + n$ sommets.

5.3.1 Contrôle du nombre de 0-m-cellules

On souhaite donc proposer ici une méthode permettant de découper une surface de topologie quelconque en une tuile unique, tout en maîtrisant à la fois la longueur du découpage et le nombre de sommets du polygone de pavage. Par la suite, nous appellerons *points multiples* les points du maillage correspondant aux sommets du polygone de pavage, c'est-à-dire les sommets du maillage participant à plus de deux arêtes du découpage ou du bord.

La modification que nous nous proposons d'apporter à l'algorithme 3.2 consiste à ne créer de points multiples que lorsque c'est nécessaire. Ainsi, l'étape de recherche puis de découpage le long des lacets non séparants les plus courts peut être pilotée sans qu'un point multiple ne soit créé : il suffit pour cela d'interdire le passage des lacets par les bords de la surface, et par les lacets déjà découpés, au besoin en ajustant la surface (*cf.* section 5.1).

Cependant, dès lors que l'on va relier les bords entre eux par une approximation de l'arbre couvrant minimal, on va devoir créer des points multiples. L'algorithme 5.4 proposé ici créé exactement un seul point multiple par lacet construit et bord initial de la surface, en maintenant à jour une liste B des bords non encore visités, augmentée de la liste des points multiples déjà créés.

La figure 5.7(b) présente le résultat de l'exécution de cet algorithme sur une surface de genre deux.

(a) Découpage topologique initial (algorithme 3.2).

(b) Découpage avec contrainte des points multiples (algorithme 5.4).

(c) Découpage avec unique point multiple.

FIG. 5.7 – *Variations des contraintes sur les points multiples lors du découpage topologique d'un double tore.*

Algorithme 5.4 : Découpage avec contrôle de points multiples

Données : $\mathcal{M}$ un 2-maillage non homéomorphe à une sphère
Résultat : Un pavage $T = (C, \mathcal{M}, F)$
$T = (C, \mathcal{M}, F)$ M-tuile initialisée à $\mathcal{M}$;
tant que *le genre de C est non nul* **faire**
  Trouver le plus court lacet non séparant c de T;
  Découper T suivant c;
fin
$B \leftarrow$ ensemble des bords (issus de $\mathcal{M}$ et des découpages);
tant que $B \neq \varnothing$ **faire**
  $c \leftarrow$ plus court chemin de T reliant deux éléments de B;
  $b_1, b_2 \leftarrow$ éléments de B reliés par c;
  si b_1 *et* b_2 *bords issus d'un même lacet* **alors**
    $e_1 \leftarrow$ extrémité de c dans b_1 ;
    $e_2 \leftarrow$ sommet correspondant dans b_2;
    $c \leftarrow$ plus court chemin reliant e_1 à e_2;
    Retirer b_1 et b_2 de B;
    Ajouter e_1 et e_2 dans B;
  sinon
    pour chaque b_i **faire**
      si b_i *bord* **alors**
        Retirer b_i de B;
        Ajouter l'extrémité de c correspondant à b_i dans B;
      fin
    fin
  fin
  Découper T suivant c ;
fin

5.3.2 Découpage en polygone minimal

Découper une surface de genre g sans bord suivant un schéma polygonal minimal implique l'existence d'un point multiple unique, base d'une famille de $2g$ lacets. La construction d'un tel ensemble de lacets est classique [35, 19], car elle correspond à la recherche des générateurs optimaux du groupe d'homotopie. Cependant, il est intéressant de constater qu'une simple déclinaison de l'algorithme 5.4 permet d'aller plus loin, en proposant une méthode de sélection du point base des lacets. Cette sélection est certes une heuristique, car déterminée par les deux premiers plus courts lacets calculés, mais permet à l'utilisateur de s'affranchir du choix d'un point base.

Pour cela, on commence par calculer le plus court lacet non séparant (section 3.3.2) sur le maillage $\mathcal{M}$ non homéomorphe à une sphère. On cherche alors le plus court chemin c non séparant reliant les deux bords issus du découpage par le premier lacet, et tel que $c(0) = c(1)$ dans $\mathcal{M}$. Ce point multiple $c(0) \in \mathcal{M}$ devient alors le point base des générateurs du groupe d'homotopie, que l'on calculera en utilisant le cut locus associé, puis en déterminant pour chaque classe d'homotopie le lacet le plus court associé.

La figure 5.7(c) présente le résultat d'un tel découpage sur une surface de genre deux.

5.4 Découpage en quadrangles

Nous présentons dans cette section deux méthodes permettant de construire un pavage par des tuiles à la topologie prédéfinie, appelées quadrangles :

Définition 5.4 – *On appelle* quadrangulation *sur une surface discrète $\mathcal{M}$ la donnée d'un M-pavage $(C, \mathcal{M}, F)$ tel que chaque 2-m-cellule de C soit un* quadrangle*, c'est-à-dire une surface de genre zéro à un bord composé de quatre arêtes.*

Cette structure particulière offre la possibilité de paramétriser tuile par tuile une surface à l'aide de fonctions mathématiques, comme les patches de Bézier [108, 33]. En particulier, le fait que les quadrangles soient décrits par un M-pavage assure qu'il n'y ait pas de jonctions en « T », c'est-à-dire pas de sommet d'un quadrangle dont l'image sur $\mathcal{M}$ se situe dans l'intérieur de l'image d'une arête d'un autre quadrangle. Bien qu'une publication récente de Bazilevs [5] présente une méthode d'ajustement de courbes mathématiques ne nécessitant pas cette contrainte, la plupart des approches imposent un tel découpage, et seront donc utilisables sur une quadrangulation telle que définie ici.

Il est important de noter ici que certains auteurs utilisent une définition différente de la notion de quadrangulation. Ainsi, Hetroy [51] interdit dans sa définition à deux quadrangles de s'intersecter en plus d'une arête. Notre définition de quadrangles revient alors à sa définition de pré-quadrangulation, qui permet de décrire un plus grand nombre de situations.

Remarque : La preuve du théorème 1.2 s'appuie sur une quadrangulation comme structure d'un découpage par n-cet valide.

Propriété 5.2 – *Soit $\mathcal{M}$ un 2-maillage de genre g, et soit $T = (C, \mathcal{M}, F)$ une quadrangulation sur $\mathcal{M}$, composée de s 0-m-cellules. Alors le nombre de tuiles de T est égal à $s + 2g - 2$.*

Preuve : La preuve est immédiate si l'on applique la caractéristique d'Euler au m-complexe cellulaire C. Supposons que ce dernier comporte s sommets, a arêtes et f faces (ou quadrangles). On a $s - a + f = 2 - 2g$. En remarquant que $2f = a$, car les faces sont des quadrangles, on a alors $f = s + 2g - 2$. □

5.4.1 Travaux existants

Il existe dans la littérature plusieurs grandes familles d'approches pour résoudre le problème de la quadrangulation. Si les objectifs sont divers, on peut par exemple citer la paramétrisation par patches de Bézier [33], la simulation par éléments finis [36], la construction de surfaces de subdivision [9], ou encore le placement de mosaïques [66].

Si la grande majorité des méthodes s'appuie sur un champ de vecteur pour reconstruire la quadrangulation, quelques approches n'exploitent pas cette propriété. Ainsi, Eck et Hopp [33] construisent par recollement de régions triangulaires les quadrangles de leur pavage. Hétroy et Attali [51] s'appuient quant à eux sur la construction d'un graphe de Reeb sur la surface, qu'ils enrichissent par la suite de chemins complémentaires. Cette dernière approche est à mettre en relation avec nos deux méthodes de découpage car, comme nous l'indiquerons par la suite, il s'agit de capturer le maximum d'informations topologiques en laissant une grande part de la géométrie à l'intérieur des tuiles.

Parmi les approches exploitant un champ de vecteur surfacique, on retrouve des méthodes s'appuyant comme la précédente sur la construction de chemins. Ainsi, Alliez *et*

al. [2] construisent un ensemble de lignes de courbure importante, qui serviront de support à la quadrangulation. La densité des lignes est notamment un des paramètres possibles de la méthode.

Une autre approche, proposée par Lay *et al.* [67] consiste à générer un maillage quandrangulaire, puis à procéder de manière incrémentale pour ajuster ce maillage en l'alignant sur le champ de vecteur. Les auteurs soulignent que le principal avantage de leur méthode est la simplicité des opérations élémentaires utilisées. Ainsi, Ray *et al.* [90] proposent une approche plus complexe, mais permettant de répondre au même problème d'alignement, cette fois-ci en s'appuyant sur une méthode de paramétrisation globale périodique. La fonction d'énergie à minimiser prend ici en compte à la fois l'invariance en translation et en rotation. Cette approche, aussi bien que la précédente, font appel à des techniques d'optimisation non linéaires.

Dong *et al.* [31] construisent quant à eux une forme harmonique qui assure de bonnes propriétés géométriques. Cependant, cette approche nécessite que l'utilisateur sélectionne un certain nombre de points extrêmes, et ne peut contrôler avec précision les points singuliers de la paramétrisation, et donc de la quadrangulation. Pour répondre à ce besoin de maîtrise des points singuliers de la paramétrisation, Alliez [1] décrit différents types de singularités que peut porter un sommet. Il introduit ensuite une extension de l'opérateur de Laplace qui supporte une partie de ces singularités.

Enfin, une approche radicalement différente, issue de l'analyse spectrale a été proposée puis développée depuis quelques années [30, 53]. Elle consiste à étudier les modes de vibrations de la surface, en étudiant les valeurs propres du spectre de cette dernière. L'aspect multirésolution de cette approche permet de générer des quadrangulations à différentes échelles pour une même surface. L'exploration des différentes échelles permet de sélectionner une bonne solution en terme d'orthogonalité.

Les méthodes présentées ici sont variées, et tentent toutes de proposer une solution au problème de la quadrangulation. Cependant, ces méthodes s'intéressent en majorité à des questions de régularité des quadrangles plutôt qu'à leur nombre ou taille.

Les deux approches que nous présentons ici ont la particularité de produire des quadrangles de grande taille, et donc peu nombreux. Il existe différentes raisons de privilégier les tuiles de grande taille à un pavage plus fin. D'une part, la longueur des bords des tuiles est nettement réduite, ce qui entraîne un nombre réduit de régions où la paramétrisation devra être ajustée avec soin. D'autre part, le fait de disposer de tuiles de grande taille permet d'envisager une manipulation multirésolution de la surface, par exemple en subdivisant chaque quadrangle en quatre quadrangles de taille plus petite. Enfin, les tuiles de grande taille permettent d'envisager une paramétrisation par patches relativement globale où seule la topologie intervient, ce qui peut faciliter le traitement et la manipulation de données complexes.

D'une manière générale, on peut dire que ces deux approches capturent majoritairement la topologie de la surface, en laissant au cœur des tuiles les propriétés géométriques de la surface. Cependant, puisqu'elles reposent essentiellement sur la recherche de plus courts chemins, il est possible là aussi d'intégrer quelques informations géométriques, sans coût supplémentaire, en modifiant la notion de distance, tel que présenté à la section 2.3.2.

5.4.2 Simplification de tuile unique

Nous avons décrit aux sections 3.3 puis 5.3 plusieurs approches pour réaliser un pavage d'une surface par une tuile polygonale unique. Ces méthodes approchent le découpage minimal, tout en permettant de contrôler plus ou moins le nombre de sommets du polygone.

(a) À gauche : un découpage ne préservant pas le $\mathcal{Q}$-critère. À droite : un découpage préservant le $\mathcal{Q}$-critère.

(b) Découpage d'un polygone à 12 côtés. a est le sommet sélectionné. Les v_i sont les potentielles extrémités du chemin de découpage pour vérifier le $\mathcal{Q}$-critère.

FIG. 5.8 – *Découpage d'une tuile polygonale paire en respectant le $\mathcal{Q}$-critère.*

La première méthode de quadrangulation que nous présentons ici s'appuie sur un découpage préliminaire en tuile unique, et réduit itérativement le nombre de sommets des polygones d'un pavage jusqu'à n'obtenir que des quadrangles. Pour assurer que l'on obtiendra bien une quadrangulation après ces réductions, on introduit la notion de $\mathcal{Q}$-critère :

Définition 5.5 – *On dit qu'une M-tuile T vérifie le* $\mathcal{Q}$-critère *si la 2-m-cellule associée à T est un n-polygone pair, avec $n \geq 4$. On dit qu'un M-pavage $(C, \mathcal{M}, F)$ vérifie le $\mathcal{Q}$-critère s'il est vérifié par chaque M-tuile composant $(C, \mathcal{M}, F)$.*

Puisque nous avons travaillé sur la maîtrise du nombre de points multiples lors du découpage en une tuile unique, les découpages que nous réalisons ensuite ne doivent pas en introduire de nouveau. Il s'agit donc de découper la tuile polygonale unique en plusieurs polygones par l'ajout de 1-m-cellules reliant des 0-m-cellules déjà existantes. Il est facile de voir que si une M-tuile ne vérifie pas le $\mathcal{Q}$-critère, alors on ne pourra pas produire de quadrangulation : au moins un polygone aura un nombre impair de sommets.

L'algorithme 5.5 s'appuie sur une méthode itérative qui introduit à chaque étape un découpage de longueur minimale préservant le $\mathcal{Q}$-critère, comme l'illustre la figure 5.8(a).

Algorithme 5.5 : Découpage en quadrangles via tuile unique

Données : $\mathcal{M}$ un 2-maillage non homéomorphe à une sphère
Résultat : Un pavage $T = (C, \mathcal{M}, F)$ en quadrangles
$T = (C, \mathcal{M}, F) \leftarrow$ découpage de $\mathcal{M}$ en tuile polygonale unique;
tant que *C contient un polygone P à plus de 4 côtés* **faire**
  $c \leftarrow$ plus court chemin sur $\mathcal{M}$ entre deux sommets de P préservant le $\mathcal{Q}$-critère;
  Découper T suivant c ;
fin

Étudions maintenant la complexité de cet algorithme, en commençant par le coût du plus court chemin :

Propriété 5.3 – *Étant donnée une M-tuile polygonale P à m côtés sur un 2-maillage $\mathcal{M}$ à n sommets, on peut calculer le plus court chemin vérifiant le $\mathcal{Q}$-critère entre deux sommets de P en $O(m^2 n \log n)$.*

Preuve : Le plus court chemin entre deux points donnés étant calculé à l'aide de l'algorithme de Dijkstra, la complexité de cette étape élémentaire est en $O(n \log n)$. On

(a) Quadrangulation après découpage topologique initial (algorithme 3.2).

(b) Quadrangulation après découpage avec contrainte des points multiples (algorithme 5.4).

(c) Quadrangulation après découpage avec unique point multiple.

FIG. 5.9 – *Résultat des variations de contraintes sur les points multiples lors de la quadrangulation d'un double tore.*

remarque ensuite qu'étant donné un sommet de P, il existe $\frac{m}{2} - 2$ sommets potentiels dont le découpage suivant le chemin correspondant vérifiera le $\mathcal{Q}$-critère. Le nombre de chemins à construire est donc de $\frac{m(m/2-2)}{2}$, d'où une complexité en $O(m^2 n \log n)$.

□

On en déduit la complexité de la quadrangulation :

Propriété 5.4 – *Étant donnée une M-tuile polygonale P à m côtés sur un 2-maillage $\mathcal{M}$ à n sommets, la quadrangulation itérative décrite par la seconde partie de l'algorithme 5.5 est réalisée en $O(m^3 n \log n)$.*

Preuve : Si le polygone initial comporte m sommets, une preuve par simple récurrence permet de vérifier qu'il faut découper suivant $\frac{m}{2} - 2$ chemins la tuile pour obtenir des quadrangles. La propriété 5.3 entraîne alors la complexité de la quadrangulation en $O(m^3 n \log n)$. □

Exemple : La figure 5.9 présente le résultat de la quadrangulation appliquée après les découpages en tuiles polygonales de la figure 5.7.

En utilisant la formule de la propriété 5.2, on remarque que le nombre minimal de quadrangles est atteint quand d'étape de découpage en tuile unique est réalisé avec un unique point multiple (section 5.3.2). Dans ce cas, on a $2g - 1$ quadrangles pour une surface de genre g.

La version intermédiaire de découpage que nous avons proposée (algorithme 5.4) produit quant à elle $3g + n - 2$ quadrangles sur une surface de genre g à n bords, car $g + n$ points multiples sont créés pendant l'étape de découpage en tuile unique.

Si d'un point de vue combinatoire, l'approche décrite ici semble intéressante. On s'appuie de plus sur les longueurs des chemins de découpage, afin d'en réduire la mesure. Les premiers exemples présentés à la figure 5.9 semblent donc confirmer l'intérêt de la méthode. Cependant, lorsque la surface commence à se complexifier, on constate que la qualité des quadrangles est nettement détériorée. Ainsi, la figure 5.10 reproduit le résultat de la quadrangulation d'une surface de genre quatre. Si l'on détaille les tuiles, on constate que plusieurs tuiles, telle que celle notée a, ont une géométrie distordue. En effet, deux

FIG. 5.10 – *Gauche : quadrangulation après découpage topologique initial (algorithme 3.2) d'une surface de genre quatre. Droite : détail sur la tuile a à la géométrie médiocre.*

(a) Pavage en cylindres par 3-cets d'une surface de genre zéro à quatre bords.

(b) Découpage du cylindre central en deux quadrangles

FIG. 5.11 – *Construction de deux quadrangles depuis un cylindre issu de découpage par* 3*-cets (section 5.2.3).*

des points opposés du quadrangle sont projetés sur le même sommet du maillage. La paramétrisation par patch de Bézier d'une telle surface est fortement compromise.

En explorant ces résultats moins bons, il est évident qu'une autre approche doit également être envisagée pour résoudre ces limitations.

5.4.3 Simplification de pavage en cylindres

L'approche que nous présentons dans cette section vise à corriger les limitations évoquées ci-dessus, tout en proposant une méthode de pavages en quadrangles s'appuyant toujours sur la construction de plus courts chemins.

En observant les résultats de moins bonne qualité (figure 5.10), le principal défaut que nous avons relevé est la non prise en compte de la structure majoritairement tubulaire de la surface. C'est ainsi que nous avons imaginé puis exploré les pistes possibles de découpage de la surface en cylindres (section 5.2). Nous avons notamment décrit une méthode de découpage à l'aide de 3-cets permettant de paver une surface à l'aide de cylindres. La figure 5.6(c) reprend ainsi la surface utilisée à la figure 5.10. On constate alors qu'avec une approche de ce type, les régions tubulaires sont réellement détectées.

L'avantage supplémentaire de ce découpage par 3-cets, et plus généralement de tout pavage en cylindres d'une surface à l'aide de n-cets est que chaque nouveau bord créé par ces structures dispose d'exactement deux points multiples.

Dans chacun de ces cylindres, on se propose donc d'utiliser ces points multiples pour raffiner le pavage en remplaçant chaque cylindre par deux quadrangles (figure 5.11).

L'algorithme 5.6 découpe un cylindre dont les bords sont ou non munis de deux points multiples en s'appuyant comme précédemment sur la recherche de plus courts chemins.

Si le premier découpage réalisé par cet algorithme utilise un plus court chemin classique, le second mérite un peu plus d'attention. Il s'agit de trouver le plus court chemin entre deux points, *à distance régulière* d'un autre chemin. Tout d'abord, la raison de ce choix est simple : on obtient par cette approche deux quadrangles dont la largeur le long

Algorithme 5.6 : Découpage d'un cylindre en deux quadrangles

Données : $\mathcal{M}$ un 2-maillage décrivant un cylindre, avec 0 ou 2 points multiples par bord
Résultat : Un pavage $T = (C, \mathcal{M}, F)$ en quadrangles
$T = (C, \mathcal{M}, F)$ M-tuile initialisée à $\mathcal{M}$;
$B_0, B_1 \leftarrow$ bords de $\mathcal{M}$;
si B_0 *et* B_1 *n'ont pas de point multiple* **alors**
  $c \leftarrow$ plus court chemin sur $\mathcal{M}$ entre B_0 et B_1;
  $c(0)$ et $c(1)$ deviennent des points multiples;
  Le point le plus éloigné de $c(0)$ sur B_0 devient un point multiple;
  Le point le plus éloigné de $c(1)$ sur B_1 devient un point multiple;
sinon si B_0 *ou* B_1 *n'a pas de point multiple* **alors**
  $c \leftarrow$ plus court chemin sur $\mathcal{M}$ entre un point multiple et l'autre bord;
  $c(1)$ devient un point multiple;
  Le point le plus éloigné de $c(1)$ sur son bord devient un point multiple;
fin
`/* On a maintenant 2 points multiples par bord */`
$c_1 \leftarrow$ plus court chemin sur $\mathcal{M}$ entre deux points multiples sur bords opposés;
Découper T suivant c_1;
$c_2 \leftarrow$ plus court chemin sur $\mathcal{M}$ entre les deux points multiples restants, à distance régulière de c;
Découper T suivant c_2;

de ces chemins ne varie qu'en fonction de la section du cylindre. On évite ainsi de pincer inutilement un quadrangle en gonflant son voisin.

La recherche d'un tel chemin s'effectue là encore en profitant des possibilités de modification de la distance locale (section 2.3.2). Soient b_1 et b_2 les deux points extrémité du chemin à calculer, et c le chemin déjà existant. On commence par calculer pour chacun des points p du cylindre sa distance $d_c(p)$ à c. Puis on estime ensuite la distance moyenne d_m que l'on va vouloir observer entre les deux chemins, en calculant la moyenne des distances des b_i à c : $d_m = \frac{d_c(b_1)+d_c(b_2)}{2}$.

Puisque l'on souhaite favoriser le passage par les sommets les plus loin de cet ensemble, on définit une nouvelle longueur d'arête $l_a((p_1, p_2)) = \|(p_1, p_2)\|_3 + |d_c(p_1) - d_m| + |d_c(p_2) - d_m|$. En quelques mots, on « rallonge » les arêtes dont les extrémités sont loin de l'isocourbe définie par d_m sur $d_c()$.

En utilisant cette longueur élémentaire pour les arêtes pour découper chaque cylindre déjà partiellement découpé, on produit donc un pavage en quadrangles, respectant la topologie, et exploitant les propriétés géométriques de la surface pour produire des quadrangles peu déformés. La figure 5.12 reprend la même surface que celle présentée pour le découpage précédent (figure 5.10).

Outre l'aspect de régularité des formes, on peut souligner un autre intérêt à cette approche. L'utilisation d'un découpage en tricets puis d'une séparation de chacune des tuiles par deux lacets implique que chaque point multiple issu du découpage participera à exactement six quadrangles s'il s'agit d'un point non bord, et à deux quadrangles s'il s'agit d'un point bord.

La programmation des processeurs graphiques (*Graphics Processing Unit*, ou GPU)

FIG. 5.12 – *Quadrangulation après découpage par cylindres (section 5.2.3).*

est l'un des domaines où la régularité des structures de données est essentiel. Ainsi, on pourrait profiter du découpage produit par notre quadrangulation pour stocker les textures d'un maillage 3D à représenter, la régularité de la structure des tuiles couvrant le maillage assurant une plus grande performance du rendu final.

5.5 Conclusion

Nous avions décrit au chapitre 2 le formalisme des M-pavages, avec lequel il est possible de décrire des tuiles à la topologie et à la combinatoire non triviales. En profitant de ces possibilités, nous nous sommes intéressés dans ce chapitre à la description de pavages avec contrôle de la topologie des tuiles.

Dans un premier temps, nous avons exploité les possibilités combinatoires d'un découpage par n-cets pour proposer une méthode de découpage en cylindres, qui s'appuie sur une technique d'optimisation pour ajuster les n-cets découpant la surface.

Puis nous avons proposé deux approches permettant de construire un pavage de surfaces en quadrangles, structure combinatoire aux intérêts applicatifs multiples.

Troisième partie

Applications

Chapitre 6
Découpages pour l'infographie

Sommaire

6.1 Distances non euclidiennes 94
6.1.1 Cartes de visibilité 94
6.1.2 Cartes d'orientation principale 95
6.1.3 Carte de contraintes utilisateur 95
6.2 UV mapping de formes organiques 96
6.2.1 Sélection des extremums 97
6.2.2 Découpage organique et paramétrisation 97
6.2.3 Résultats 98
6.3 Segmentation organique 102
6.3.1 Segmentation cylindrique 103
6.3.2 Résultats 104
6.4 Conclusions et perspectives 105

L'infographie a depuis vingt ans envahi notre quotidien, rendant possible par le traitement informatique la création de visuels jusqu'alors impossibles à produire. Ainsi, le monde du cinéma a radicalement évolué, intégrant à toutes les étapes de la production des réalisations artistiques numériques. La démocratisation des jeux vidéos a également été accompagnée d'une maîtrise croissante des possibilités de créativité infographique. Parmi ces possibilités numériques, la manipulation de données en trois dimensions, depuis la modélisation d'objets virtuels jusqu'au rendu photoréalistique, constitue un élément incontournable du paysage artistique contemporain [26].

D'un point de vue méthodes et logiciels, la multiplication de ces réalisations s'est accompagnée de plusieurs phénomènes. Tout d'abord, l'émergence de logiciels complets et toujours plus performants[1] répondants à des besoins pointus des professionnels de l'infographie a grandement facilité leur travail. On observe également depuis quelques années la démocratisation de l'accès à ces outils, avec des logiciels destinés au grand public, permettant de créer sans nécessairement être expert du domaine. Enfin, la masse des documents disponibles sur le réseau mondial comme sur les bases de données spécialisées permettant d'envisager des traitements automatisés, l'avènement d'outils de traitement automatiques semble inévitable.

[1] On peut par exemple citer 3D Studio Max, ZBrush, ou encore UVLayout

Ces différents domaines d'utilisation des logiciels de traitement de données 3D, s'ils répondent à des besoins différents, nécessitent dans tous les cas une assistance de l'humain par la machine, afin de faciliter, automatiser et accélérer les réalisations. De nombreux travaux vont dans ce sens, depuis la modélisation [21, 55] jusqu'à la paramétrisation pour texturage [94], ou encore la squelettisation [104] et la segmentation sémantique [80].

Nous décrivons dans ce chapitre, en nous appuyant sur les idées développées pendant les parties I et II, un ensemble d'outils de manipulation de surfaces à destination de l'infographie. Ces outils exploitent à la fois des notions topologiques et géométriques, tout en laissant une part de liberté à l'utilisateur, en lui permettant de piloter intuitivement ces algorithmes par des ajustements spécifiques. L'objectif est donc ici d'automatiser certains traitements, tout en laissant un contrôle artistique et applicatif à l'utilisateur.

6.1 Distances non euclidiennes

Les algorithmes de découpage que nous présentons dans ce mémoire s'appuient essentiellement sur des calculs de plus courte distance le long des arêtes, en utilisant l'algorithme de Dijkstra. Il est donc possible, suivant les besoins, d'ajuster localement la notion de distance pour modifier le comportement global de ces découpages. Ainsi, nous avons montré à la section 4.1.1 qu'il était possible de tenir compte de la géométrie locale en introduisant un coefficient multiplicatif linéairement dépendant de la courbure locale.

Nous présentons dans cette section différentes modifications complémentaires de la longueur des arêtes, permettant de répondre à des besoins spécifiques à une application en infographie. Ces différentes modifications étant toutes basées sur un principe de pondération multiplicative de la distance euclidienne, elles peuvent donc toutes être conjuguées pour répondre aux besoins de l'utilisateur. Si d'un point de vue logiciel, ces ajustements n'influent que sur la distance utilisée, l'interface utilisateur ne doit pas évoquer cette notion, car il est possible d'associer à chaque fois une sémantique spécifique, qui rend l'outil plus intuitif.

6.1.1 Cartes de visibilité

Lorsque l'on dispose d'informations sur l'utilisation future de l'objet que l'on est en train de découper, il est possible d'introduire des modifications de distances adaptées. Ainsi, Sheffer et Hart [93] proposent d'introduire une notion de visibilité associée à chaque arête de la surface, déterminée en observant l'objet depuis un ensemble de points de vue régulièrement répartis sur une sphère autour de l'objet.

Les auteurs proposent de calculer pour chacune des faces de la surface un pourcentage de visibilité, estimé en calculant depuis chacune des vues si la face est visible. Ils déterminent ensuite la visibilité $V(e)$ d'une arête e en calculant la moyenne des visibilités des triangles adjacents. Une nouvelle longueur d'arête tenant compte de la visibilité est alors calculée par $l_V(e) = \|e\|_3 V(e)$.

Dans le cadre d'une application spécifique où l'on connaît à l'avance la position des caméras, on peut facilement modifier la méthode décrite ici pour ne vérifier la visibilité des faces que depuis les positions utilisées, afin d'augmenter les zones de découpage possible.

FIG. 6.1 – *Découpage par la méthode PCA (section 4.2.2) en utilisant une distance euclidienne (gauche), et une distance définie par orientation (droite). Maillage 3D original fourni par le dépôt de formes AIM@SHAPE.*

6.1.2 Cartes d'orientation principale

De manière plus générale, les infographistes exploitent régulièrement les propriétés géométriques des objets en favorisant les découpages à travers les régions qu'ils savent potentiellement moins visibles. À titre d'exemple, les animaux à quatre pattes seront classiquement découpés en favorisant les arêtes situées sous l'animal.

Nous proposons donc d'introduire cet usage au cœur de nos méthodes de découpage en utilisant une carte multiplicative liée à une direction que l'on laissera à la discrétion de l'utilisateur. Dans les faits, cette modification permet d'ajuster naturellement les découpages en favorisant une région au choix de l'utilisateur (la panse de l'animal dans l'exemple cité précédemment).

Afin de construire une telle carte, on laisse à l'utilisateur le soin de choisir la direction et le sens du « bas », puis on détermine le sommet du maillage le plus loin du barycentre dans cette direction. Au besoin, si l'axe ainsi déterminé n'intersecte pas la surface, on utilisera un petit voisinage. On considère alors que ce point c est le centre de la partie cachée, et on calcule pour chaque sommet s du maillage la distance géodésique $d(s, c)$ à c. On calcule alors pour chaque arête $e = (a, b)$ une nouvelle distance $l_o(e) = \|e\|_3 (\alpha d(a, c)d(b, c) + \epsilon)$, où α et ϵ sont des coefficients que l'on laisse à la discrétion de l'utilisateur, le paramètre α décrivant l'importance relative donnée à la direction d'intérêt, le coefficient ϵ permettant d'éviter les arêtes trop courtes. Ainsi, en utilisant cette modification de la longueur des arêtes définie par le sommet c, on favorise les découpages de chemins dans la zone indiquée par l'utilisateur. La figure 6.1 illustre la pertinence d'une telle mesure. Les deux images présentées sont issues d'un découpage par la méthode PCA décrite à la section 4.2.2. Cependant, si l'image de droite utilise la distance euclidienne classique pour calculer les géodésiques, la figure de gauche intègre une information d'orientation donnée par l'utilisateur.

6.1.3 Carte de contraintes utilisateur

Enfin, si les deux sections précédentes décrivent la mise à disposition de l'utilisateur d'ajustements simples, peu de possibilités de réglage sont disponibles, ce qui pourrait gêner un utilisateur averti habitué à ajuster chacun de ses découpages, comme le pratiquent beaucoup les infographistes.

La méthode que nous décrivons ici permet à l'utilisateur de guider entièrement les

(a) (b) (c)

FIG. 6.2 – *Découpage d'une surface homéomorphe à un disque en utilisant un seul extremum. (a) : distance euclidienne uniquement. (b) : carte de poids dessinée par un utilisateur sur la surface. (c) : distance modifiée par la carte de l'utilisateur. Maillage 3D original fourni par le dépôt de formes laboratoire Computer Graphics de l'Université de Stanford.*

outils de découpages, en ajustant pour chaque région de la surface un degré de non-découpabilité. Il suffit pour cela de lui proposer une interface permettant de donner pour chaque arête e de la surface un poids $p(e)$ comprise entre 0 et 1. Comme précédemment, on calcule alors une nouvelle distance $l_a(e) = \|e\|_3 \, p(e)$ pour les arêtes, qui tiendra compte du choix de l'utilisateur.

L'acquisition de cette fonction de poids peut être diverse. L'approche permettant une grande liberté artistique consiste à proposer une interface complète de peinture sur surface, offrant à l'utilisateur la possibilité de choisir pour chaque partie du maillage le degré de solidité. Ce type de fonctionnalités est couramment implémenté dans les modeleurs 3D, qui utilisent par exemple ces valeurs pour faciliter les animations à l'aide de structures rigides.

La figure 6.2 illustre ainsi les possibilités offertes par cette approche. La surface représentée est homéomorphe à un disque, et un sommet extremum a été sélectionné (voir section 6.2.1). La figure 6.2(a) présente le découpage (voir section 4.2.3) calculé en utilisant la distance géodésique classique. La figure 6.2(c) le découpage issu de l'utilisation de la carte des poids utilisateur (figure 6.2(b)). On constate que les zones décrites comme interdites (en rouge) on été évitées, et que les zones conseillées par l'utilisateur (en bleu) ont été empruntées par le chemin de découpage.

6.2 UV mapping de formes organiques

La majeure partie des réalisations en infographie exploitent la possibilité de placer des textures à la surface des objets. Ces textures peuvent être utilisées lors du rendu, par exemple pour décrire la couleur des objets, mais aussi comme carte de déformations volumique (*bump mapping*), ou encore comme carte de réflectance. L'origine de ces données surfaciques est très variable. Elles peuvent être générées automatiquement, par exemple en utilisant des méthodes procédurales [27, 69], extraites d'une modélisation plus fine du maillage, ou encore réalisée par un infographiste.

Une grande partie de ces méthodes nécessite lors de la création de la texture une repré-

sentation planaire de la surface (appelée *UV mapping*), par exemple issue d'un dépliage de la surface (section 4.1.2). D'autre part, même si les textures ont été générées directement depuis les données 3D, il est nécessaire de les stocker, ou encore de les transmettre à la carte graphique pour assurer un rendu rapide lors d'animations temps réel par exemple. Dans ce contexte, la cohésion des textures associées à chaque triangle est essentielle, et là encore, un dépliage de la surface est une étape essentielle.

Comme nous l'avons évoqué dans les chapitres précédents, une paramétrisation de la surface nécessite un découpage topologique, éventuellement suivi d'un découpage géométrique. Si dans le cas de formes géométriques régulières ces découpages peuvent être guidés par les arêtes aiguës, dans le cas de formes organiques la géométrie globale devient essentielle. Nous décrivons ici une approche semi-automatique permettant de produire un dépliage de surfaces organiques, exploitant à la fois la topologie et la géométrie de la surface à texturer, mais aussi les contraintes spécifiques décrites par l'utilisateur.

6.2.1 Sélection des extremums

Si d'un point de vue algorithmique nous avons décrit à la section 4.2.2 une manière précise de calculer les extremums d'une surface à la géométrie variable, on remarque que dans un cadre applicatif, ils peuvent être trop nombreux. On se propose donc ici de définir une méthode semi-automatique afin de faciliter la sélection des extremums théoriques qui seront retenus par l'utilisateur.

En utilisant la notion de distorsion $(v_1, v_2) \mapsto D(v_1, v_2)$ des arêtes définie précédemment, on définit la distorsion associée à un sommet, en calculant la moyenne des distorsions des arêtes auquel il participe :

Définition 6.1 – *Soit S une surface discrète, et v un sommet de S. On définit $D(v)$ la* distorsion *en v par :*

$$D(v) = \frac{1}{|(v, v_i) \in E(S)|} \sum_{(v,v_i)\in E(S)} D(v, v_i).$$

On utilise alors cette mesure de la distorsion locale pour construire un ordre sur les extremums. Nous constaterons par la suite que plus cette distorsion est petite, plus l'extremum capture un détail moins important de la surface. Cette importance n'est pas ici liée à la taille du détail, mais à sa forme globale : les grands bras seront considérés plus importants que les régions légèrement bosselées (voir figure 6.4).

On profite alors de cet ordre pour proposer à l'utilisateur différents moyens de sélection semi-automatique des extremums. Parmi les choix possibles, on peut proposer de ne garder que les extremums dont la distorsion est supérieure à la moyenne des distorsions de tout les extremums. De la même manière, on peut utiliser la médiane. De manière plus flexible, on profite de l'ordre défini sur les extremums détecter pour proposer une interface où un curseur permet de sélectionner une valeur seuil de distorsion à partir de laquelle les extremums sont conservés.

6.2.2 Découpage organique et paramétrisation

La génération d'une carte plane *UV mapping* pour la manipulation des textures de la surface nécessite une étape de paramétrisation. Nous avons décrit à la section 4.1.2 différentes méthodes de paramétrisation conforme, qui produisent de tels dépliages. Comme nous l'avons évoqué alors, ces méthodes ne peuvent être appliquées que sur des surfaces

homéomorphes à un disque. Un découpage à la fois topologique et géométrique est donc nécessaire.

En reprenant l'approche déjà évoquée à la section 4.2, nous proposons ici un outil complet à destination de l'infographiste, dont le comportement peut être ajusté grâce aux différents réglages que nous avons décrit dans les deux parties différentes.

Ainsi, puisque cet outil s'appuie sur des méthodes de découpage, on peut proposer à l'utilisateur différents ajustements qui modifieront cette distance (section 6.1). Comme nous l'avons déjà décrit, ces ajustements permettent de tenir compte plus finement de la géométrie locale, en utilisant la courbure, de la visibilité moyenne des différentes régions, ou encore de propriétés spécifiques à l'application telles qu'une direction privilégiée ou une carte de découpabilité définie par l'utilisateur.

Une fois le découpage topologique réalisé, on applique un découpage géométrique en commençant par sélectionner les extremums, et en proposant à l'utilisateur d'en ajuster le nombre (section 6.2.1). Ce découpage géométrique peut là encore être adapté en utilisant les distances modifiées décrites plus haut.

Enfin, la dernière étape de dépliage produit la carte plane, qui deviendra le support des textures.

6.2.3 Résultats

La section A.3.4 décrit l'implémentation d'une interface graphique, intégrée à `blender`[2], qui propose à l'utilisateur de réaliser un tel dépliage à partir d'un maillage surfacique. Bien que rudimentaire et peu intuitive d'utilisation, ce prototype illustre la faisabilité des idées que nous avons développées dans cette partie.

Ainsi, nous présentons dans cette section quelques exemples de réalisation issues de l'exécution de nos outils de découpage à destination de l'infographie.

Capture de détails multi-échelle

L'approche de détection des extremums proposée est conçue pour ne pas être limitée par la présence de détails géométriques de taille différente. Ainsi, la figure 6.3 présente une surface homéomorphe à un disque, où des détails de plusieurs tailles sont présents.

Pour comparaison, nous avons appliqué la méthode de détection des extremums proposée par Tierny *et al.* [103], en faisant varier le paramètre de correspondance des extremums (figures 6.3(a) et 6.3(b)). Cependant, comme nous nous y attendions, cette approche ne permet pas à la fois de capturer les petits et grands détails.

À l'inverse, l'approche par transformation conforme que nous présentons ici capture les extremums sans contrainte d'échelle (figure 6.3(c)).

Variations du nombre d'extremums

Afin d'illustrer pleinement les possibilités de capture des extremums sans limitation d'échelle, et notamment la flexibilité de notre approche, nous avons synthétisé une surface « multi-échelle » en assemblant cinq lapins de Stanford[3] de tailles décroissantes (figure 6.4(a)). Les différents éléments géométriques – oreilles, queue, flancs – constituent les éventuels extremums, plus ou moins saillants, que notre méthode doit détecter. La

[2] *blender* est un logiciel libre de modélisation et de rendu 3D. `http://blender.org`

[3] `http://graphics.stanford.edu/data/3Dscanrep/`

(a) Détection des extremums suivant [103], avec un petit ϵ.

(b) Détection des extremums suivant [103], avec un grand ϵ.

(c) Détection par transformation conforme.

FIG. 6.3 – *Détection d'extremums à différentes échelles. Les figures (a) et (b) illustrent les limitations de la méthode de [103].*

extremums	Surface		Distorsion			
#	#sommets	long. bord	minimum	maximum	moyenne	écart-type
0	6 800	364, 1	$3.7 \times e^{-12}$	$2, 7 \times e^{6}$	6 683, 6	46 593, 1
19	7 061	1 696, 8	0, 055	18, 5	1, 23	1, 58
96	7 389	3 154, 3	0, 087	18, 2	1, 30	1, 80

TAB. 6.1 – *Étude quantitative de la distorsion des longueurs des arêtes en fonction du nombre d'extremums utilisés pour le découpage. La surface dépliée a été mise à l'échelle en utilisant la médiane des distorsions comme unité. La figure 6.4 présente le résultat graphique de ces découpages.*

surface ainsi créée comporte 6 800 sommets et 13 510 triangles, et est homéomorphe à un disque.

Sur cet objet, le nombre global d'extremums détectés est de 96. Nous avons également choisi un seuillage manuel des extremums ne prenant en compte que les détails les plus saillants (c'est-à-dire les oreilles). La figure 6.4 présente la surface initiale, et la surface découpée suivant les deux sélections d'extremums, le seuillage manuel n'étant qu'un exemple parmi ceux que nous avons testé. Les figures du haut permettent ainsi de visualiser les découpages (en rouge). Les figures du bas quant à elles ont été générées en dépliant la surface découpée à l'aide d'une transformation conforme (ABF), puis en dessinant dans l'espace 2D correspondant une texture de damier régulier noir et blanc. Cette représentation permet de visualiser intuitivement la régularité du dépliage, et donc la qualité du découpage.

Le premier résultat immédiat est le manque de régularité du dépliage lorsqu'il n'y a pas de découpage (figure 6.4(a)). Ce résultat était prévisible, et les découpages présentés par les deux autres figures entraînent des dépliages plus réguliers. Il n'est pas évident de distinguer à l'aide des damiers la qualité des deux découpages. En analysant les résultats quantitatifs portés par le tableau 6.1, on dispose de plus d'informations pour comparer ces deux découpages.

Tout d'abord, on constate que le passage de 19 à 96 extremums a entraîné une augmentation notable de la longueur des bords de la surface, et donc augmenté d'autant le nombre de régions où la texture ne sera pas continue. D'autre part, on constate à la lecture de la troisième partie du tableau, consacrée aux distorsions de la longueur des arêtes, que les gains en terme de distorsion ne sont pas très élevés.

Ces résultats nous confortent dans l'idée de laisser à l'utilisateur la possibilité d'ajuster le nombre d'extremums à utiliser. D'autre part, on remarque en regardant la partie supérieure de la figure 6.4 que les extremums sélectionnés après les 19 premiers correspondent

(a) Surface initiale homéomorphe à un disque.

(b) Découpage en utilisant 19 extremums (sélection manuelle).

(c) Découpage en utilisant le nombre maximum d'extremums détectés (96 extremums).

FIG. 6.4 – *Illustration de la distorsion du dépliage en fonction du nombre d'extremums utilisés pour le découpage. Le tableau 6.1 présente quelques résultats quantitatifs sur ces découpages. Maillage 3D original fourni par le dépôt de formes laboratoire Computer Graphics de l'Université de Stanford, modifié par nos soins.*

(a) Résultat du découpage topologique.

(b) Résultat du découpage géométrique avec 80 extremums sélectionnés.

FIG. 6.5 – *Découpage topologique et géométrique d'une surface de genre deux, sans bord, et à la géométrie non triviale.*

à des parties de la surface moins courbe (queue, flancs), ce qui confirme la pertinence de l'ordre que nous avons choisi sur les extremums.

Topologie non triviale

En suivant la méthode décrite dans la section précédente, nous avons réalisé une nouvelle surface, de genre deux et sans bord, mais disposant d'une géométrie non triviale. En observant la figure 6.5, on constate que le découpage topologique n'est clairement pas suffisant, même s'il est nécessaire pour produire un dépliage conforme. L'introduction de découpages géométriques, correspondant à la fois aux détails fins des lapins, et aux zones de courbure importante de la partie double tore assure une paramétrisation nettement plus régulière sur la globalité.

Utilisation de la courbure locale

Toutes les méthodes de découpage que nous avons décrites jusqu'à présent peuvent être ajustées par le choix d'une distance spécifique. Le découpage des surfaces en utilisant les extremums ne fait pas exception à la règle. Ainsi, la figure 6.6 illustre l'utilisation d'une distance tenant compte de la courbure locale lors du découpage suivant les extremums.

On constate que le résultat issu de l'exploitation la courbure locale produit un découpage mieux adapté à la forme manipulée. L'illustration du haut permet de vérifier que le découpage avec prise en compte de la courbure locale ne traverse pas de grandes zones plates, et sépare mieux les deux faces du tube. Le détail de découpage représenté en bas de chacune des illustrations permet de vérifier la qualité du découpage au niveau du bouchon du tube, qui dans le second cas emprunte bien la partie la plus aiguë.

(a) Résultat du découpage géométrique avec distance euclidienne.

(b) Résultat du découpage géométrique avec distance prenant en compte la courbure locale.

FIG. 6.6 – *Découpage géométrique d'une surface de genre zéro, sans bord, en faisant varier la distance utilisée.*

Enfin, la projection après dépliage d'un damier permet de vérifier que la régularité de la paramétrisation est accrue en imposant la préservation des régions planes.

6.3 Segmentation organique

Avec l'augmentation croissante des possibilités de génération et de traitement des données 3D, il est devenu essentiel de disposer d'outils d'abstraction pour en faciliter la manipulation.

Parmi ces outils d'abstraction, plusieurs approches s'appuient sur un pavage de la surface, que ce soit pour faciliter l'animation à travers une étape de squelettisation [104], pour réaliser des morphings entre objets [95, 114], afin de détecter les collisions [70], ou encore à des fins de simplification de surfaces [16]. Ces étapes de pavages, que l'on rencontre plus couramment sous le nom de segmentation, peuvent être guidées par de nombreuses informations

Ainsi, on constate une grande variété d'approches proposant une solution à cette question [3]. Selon les auteurs de cette étude comparative, les méthodes de segmentation peuvent être séparées en deux catégories. Les premières s'appuient essentiellement sur des propriétés géométriques locales pour construire des patches vérifiant un certain critères de régularité géométrique. Les secondes, qualifiées d'approches sémantiques, capturent des propriétés de forme plus globales.

Parmi ces dernières méthodes, plusieurs d'entre elles exploitent à la fois des informations topologiques et géométriques, et peuvent en cela être rapprochée de la méthode que nous décrivons par la suite. Ainsi, Mortara *et al.* [79] procèdent à une extraction des parties tubulaires de la surface, en explorant l'intersection de cette dernière avec une sphère centrée en chacun des points de la surface. Cette méthode dépend d'un paramètre – le rayon de la sphère – correspondant à la taille des détails que l'on souhaite extraire.

Tierny *et al.* [104] construisent à l'aide d'un graphe de Reeb une décomposition hiérarchique de la surface, chaque patch étant identifié par une arête du graphe. La construction du graphe de Reeb est déterminée par la détection d'extremums, déjà présentée à la section 4.2.1. De manière générale, les méthodes s'appuyant sur la construction de squelettes courbes [22] intègrent également une notion de topologie.

Nous présentons dans cette section une approche de segmentation, dont l'originalité

réside dans l'utilisation d'informations géométriques traduites d'un point de vue topologique. La segmentation résultant de ce découpage présente également la particularité d'être composée uniquement de cylindres ou de disques (éventuellement sphères dans un cas limite), topologies propices à l'abstraction de la surface qui les porte, que ce soit pour la squelettisation ou pour en extraire des primitives de formes.

6.3.1 Segmentation cylindrique

Lorsque la surface dispose d'une topologie non triviale, nous avons proposé à la section 5.2 une méthode de pavage de la surface par cylindres. Cette segmentation s'appuie sur la construction successive de tricets, qui découpent topologiquement la surface, en intégrant des propriétés de géométrie simple comme la longueur de découpage.

Cependant, quand la surface à manipuler ne possède pas une telle topologie, cette méthode est inefficace. Nous nous proposons de décrire dans cette section une approche globale, en profitant des possibilités topologiques de cette première méthode, tout en en intégrant au cœur du processus des propriétés géométriques plus globales de la surface,

Comme nous l'avions déjà décrit à la section 4.2.3, il est possible assez simplement de transformer des informations géométriques en informations topologiques. Après avoir construit un ensemble de points géométriquement remarquables, on retire autour de chacun de ces points un petit disque, ajoutant ainsi un bord supplémentaire à la surface. L'algorithme 6.1 décrit la structure générale de la méthode que nous proposons, qui permet de produire une segmentation en cylindres, disques et sphères sur une surface quelconque.

Algorithme 6.1 : Segmentation organique

Données : $\mathcal{M}$ un 2-maillage
Résultat : Une segmentation organique $T = (C, \mathcal{M}, F)$
$E \leftarrow$ un ensemble d'extremums de $\mathcal{M}$;
Retirer un petit disque à $\mathcal{M}$ autour de chaque sommet de E;
$T \leftarrow$ pavage en cylindres de $\mathcal{M}$ (algorithme 5.3);
Rajouter à $\mathcal{M}$ les disques ôtés précédemment;

Tout d'abord, vérifions que chacune des tuiles résultant de ce découpage sont d'un point de vue topologique des disques, des cylindres ou des sphères. L'algorithme 5.3 utilisé à la fin du processus produit un découpage en cylindres d'une surface donnée. Ce découpage étant ensuite complété en comblant certains bords à l'aide de disques, les seules topologies possibles sont bien celles attendues.

Décrit de cette manière, l'algorithme ne précise pas la manière de détecter l'ensemble des extremums du maillage. En effet, toutes les approches peuvent être envisagées. L'une d'elles consiste naturellement à reprendre les idées présentées à la section 6.2.1, qui laisse à l'utilisateur la possibilité d'ajuster suivant ses besoins pratiques la quantité d'extremums souhaités. Cependant, cette approche nécessite un premier découpage topologique, car elle s'appuie sur un dépliage de la surface. Si cette première étape peut être vue dans certains cas comme un inconvénient, ce n'est pour autant pas un défaut majeur, puisque le découpage initial peut être recousu au besoin après détection des extremums, afin de redonner sa topologie originale à la surface.

Dans les résultats que nous détaillons ci-dessous, nous avons illustré la flexibilité de l'algorithme 6.1 en utilisant également une autre approche pour détecter les extremums.

(a) Surface après suppression des points extremums (section 6.2.1).

(b) Surface après découpage en cylindres (section 5.2).

FIG. 6.7 – *Segmentation en cylindres d'une main. Maillage 3D original fourni par le dépôt de formes AIM@SHAPE.*

6.3.2 Résultats

Nous avons appliqué cette méthode de segmentation à différentes surfaces, en faisant varier le mode de sélection des extremums.

Ainsi, la figure 6.7 présente la segmentation d'une main en cylindres et disques, après détection des extremums par la méthode de dépliage de à la section 6.2.1. Chacun des doigts de la main est ainsi décrit par une surface homéomorphe à un disque, et correspondent aux extremums qui ont été repérés. Ce pavage est complété de quatre cylindres, qui décrivent chacun une jonction entre deux des éléments déjà présents. Avec les mêmes paramètres pour le recuit simulé que ceux utilisés dans la section 5.2.3, le découpage de cette surface à 3 897 sommets s'est effectuée sur notre machine de travail en 20 secondes, dont deux secondes nécessaires à la détection des extremums.

La figure 6.8 reprend deux surfaces où la sélection des extremums a été réalisée par une approche n'utilisant que les distances sur la surface [103].

Nous avons ajusté pour la figure 6.8(a) le paramètre de proximité pour n'avoir que cinq extremums, afin de produire une segmentation générale de la surface. Chacun des membres géométriques est ainsi décrit dans la segmentation finale par un patch homéomorphe à un disque, que viennent compléter deux cylindres décrivant le buste du personnage. De nouveau avec les mêmes paramètres, le découpage de cette surface à 4 525 sommets s'est effectuée en 20 secondes.

La surface de la figure 6.8(b) quant à elle ne porte qu'un unique extremum, localisé sur la tête de la bouée. On profite dans ce cas des possibilités offertes par notre méthode de découpage en cylindres, qui peut s'adapter à toute topologie, puisque la surface initiale

(a) Segmentation en cylindres d'un personnage humanoïde. Maillage 3D original fourni par le dépôt de formes AIM@SHAPE.

(b) Segmenentation en cylindres d'une surface de genre un.

FIG. 6.8 – *Segmentation d'objets issus d'un modeleur 3D.*

est de genre un. Cette surface composée de 18 495 a été découpée en 82 secondes, toujours avec les mêmes réglages.

Enfin, pour compléter cette étude, nous avons appliqué notre méthode de découpage sur une surface bruitée (figure 6.9). Le résultat obtenu illustre la stabilité de l'approche face à une perturbation locale de l'objet.

6.4 Conclusions et perspectives

Nous avons esquissé dans ce chapitre différents outils destinés à l'infographiste, exploitant les propriétés topologiques et géométriques des surfaces, tout en offrant une liberté de modulation à l'utilisateur. Nous avons ainsi montré que les algorithmes présentés dans la partie II pouvaient être exploités dans un cadre applicatif, et qu'un grand nombre d'ajustements étaient possibles, en intégrant des informations liées à aux propriétés locales de la surface, mais aussi spécifiques à l'application.

Cependant, si les différentes méthodes présentées ici ont leur place dans un prototype pour illustrer leurs capacités, il semble que les temps de calcul obtenus soient loin d'être satisfaisants pour être proposés sans plus d'adaptation à un public d'infographistes. Certes, la diversité des données manipulées par ces derniers est suffisamment grande pour que l'on puisse envisager l'utilisation occasionnelle du découpage pour l'UV mapping ou la segmentation par cylindres. Cependant, la manipulation de surfaces à très grand nombre de sommets n'est pas rare, et la complexité des algorithmes présentés ici ne faciliterait pas leur traitement.

Si l'on souhaite proposer concrètement ces méthodes pour une application en infogra-

Fig. 6.9 – *Segmentation en présence de bruit.*

phie, il va donc être nécessaire de travailler à l'amélioration des méthodes. Plusieurs pistes sont envisageables.

D'une part, une approche générale pour améliorer ces découpages est de travailler non sur les plus courts lacets et chemins, mais sur des approximations de ces derniers, comme nous avons commencé à le présenter à la section 3.3.4. Nous avons montré que les temps de calcul étaient alors considérablement réduits.

Dans le même esprit, l'une des pistes à explorer est l'utilisation d'approximations de géodésiques à la complexité moins élevée que l'algorithme de Dijkstra. Nous avons ainsi déjà cité à la section 2.3.1 différents travaux qui vont dans ce sens, et qui pourraient être le point de départ d'une amélioration des performances de nos algorithmes, si la non exactitude des chemins calculés n'est pas trop pénalisante.

D'autre part, les outils – chemins, boucles, n-cets – que nous avons manipulés nous semblent intéressants, et il nous semblerait dommage d'abandonner l'exploration de leurs possibilités dans un cadre infographique sans chercher à les exploiter à l'aide d'autres outils. Ainsi, la segmentation en cylindres décrite dans ce chapitre s'appuie sur l'utilisation d'un processus de recuit simulé pour ajuster les tricets. Comme nous l'avons évoqué à la section 5.2.3, plusieurs améliorations de ce processus de positionnement sont déjà envisageables, et pourraient améliorer grandement l'efficacité du processus, offrant ainsi des outils pratiques aux performances proches de celles attendues.

Chapitre 7
Application médicale

Sommaire

7.1 Contexte général 107
7.2 Cartographie du cortex 108
7.2.1 Travaux antérieurs 109
7.2.2 Chaîne de traitement générale 109
7.3 Application à la stimulation corticale 112
7.3.1 Contexte médical 112
7.3.2 Construction des cartes 113
7.3.3 Résultats et discussion 114
7.4 Conclusion et perspectives 117

Nous avons décrit dans la section 4.3 une approche permettant de corriger topologiquement une surface en tenant compte de sa géométrie, en particulier lorsque les données sont issues d'une acquisition basse résolution.

L'un des principaux champs d'application pour ce type d'approche est l'imagerie médicale. L'acquisition puis la manipulation d'images médicales permet en effet la construction de surfaces complexes, sans pour autant que l'on maîtrise l'exactitude de sa topologie. Nous proposons donc dans ce chapitre une chaîne complète de traitement permettant d'obtenir à partir d'une image volumique la carte plane d'une zone d'intérêt sur la surface d'un cerveau. Nous montrons ensuite comment ces informations peuvent être utilisées dans un cadre applicatif concret, celui de la stimulation corticale, comme support à des informations anatomiques et applicatives. Enfin, nous montrons [37] que ces cartes permettent d'offrir aux praticiens une navigation interactive de visualisation des données.

7.1 Contexte général

Les avancées technologiques de ces dernières dizaines d'années sont à l'origine de la grande diversité des modalités d'imagerie qui permettent d'observer aussi bien l'anatomie que les caractéristiques fonctionnelles médicales. Parmi ces modalités, on remarque l'importance prise par les modalités tridimensionnelles, comme l'imagerie par résonance magnétique (IRM), le scanner, la tomographie par émission de positons (TEP) ou encore l'imagerie par résonance magnétique fonctionnelle (IRMf).

La caractéristique commune de ces différents modes d'acquisition est la structuration des données qu'ils apportent. En effet, ces acquisitions tridimensionnelles sont numérisées de manière discrète, en une grille régulière en chacune des trois dimensions. Chaque *voxel* – ou élément de volume – porte une valeur correspondant à la mesure d'une grandeur physique dans un petit voisinage de l'espace d'acquisition. On dispose donc d'une fonction image $I : [\![0, N_x]\!] \times [\![0, N_y]\!] \times [\![0, N_z]\!] \to A$ qui pour chaque voxel de la grille retourne la valeur correspondant à l'acquisition, souvent exprimée dans $A \subset \mathbb{R}$. Ces informations d'acquisition sont généralement complétées par la taille des voxels dans chacune des directions, qui peut être différente si l'échantillonnage n'est pas isotrope.

Dans la suite de ce chapitre, nous nous sommes intéressés à des données issues d'acquisitions IRM et scanner de cerveaux humains. Dans chacun de ces cas, les images volumiques portent en chaque voxel une valeur réelle, que l'on interprète classiquement comme un niveau de gris.

Ainsi, les niveaux de gris portés par une image IRM sont physiquement déterminés par la réponse des atomes à une combinaison d'ondes électromagnétiques, et permettent de mesurer les propriétés chimiques des tissus du patient. Interprétés par un expert, une image IRM (figure 7.2(a)) apporte des informations de type anatomique, permettant de différencier les différents tissus tels que la matière grise (MG), la matière blanche (MB), ou le liquide céphalo-rachidien (LCR). Les images produites par scanner sont quant à elles issues de la réponse des tissus soumis à un faisceau de rayons X, et reconstruites par tomographie. Elles permettent de mesurer la densité des tissus, et notamment mettent en évidence la localisation des os, ou encore des pièces métalliques implantées telles que les électrodes.

Ces différentes informations sont classiquement utilisées par les praticiens pour visualiser les différents éléments anatomiques et fonctionnels du patient à traiter. Un prétraitement informatique de ces images volumiques est souvent appliqué pour faciliter l'accès à ces informations, et permettre une interprétation sans entrave. Il est courant par exemple pour un praticien d'utiliser une visualisation 3D de la surface du cerveau pour localiser des zones d'intérêt en fonction du contexte clinique. Cependant, cette représentation est vite limitée, car difficile à manipuler, et ne permet pas une compréhension globale immédiate des phénomènes étudiés.

Dans la suite, nous nous intéressons au cas où la région d'intérêt étudiée est le cortex cérébral. Nous présentons alors une approche originale permettant de produire à partir d'un volume IRM anatomique une carte plane du cortex. Cette représentation offre à la fois une bonne visualisation globale (section 7.3.3) tout en respectant l'anatomie du patient et en permettant la projection d'informations spécifiques (section 7.3.2).

7.2 Cartographie du cortex

Il existe différentes approches permettant de produire une carte plane à partir d'une surface donnée. Nous avons décrit à la section 4.1.2 différentes méthodes de paramétrisation envisageables pour générer de tels dépliages. Différentes méthodes, parfois moins complexes, ont été utilisées pour produire une représentation des informations de la surface corticale. Après avoir décrit les travaux antérieurs répondant à cette problématique, nous décrivons l'approche complète que nous avons développée, en soulignant l'originalité de nos choix.

7.2.1 Travaux antérieurs

La localisation et la visualisation des différentes informations de la surface corticale sont des tâches complexes, que de nombreuses équipes de praticiens résolvent en reproduisant ces informations sur une représentation schématique de la région d'intérêt, la position des sillons étant représentée à partir d'une reconstruction superficielle d'une tomographie informatisée (TI) [84, 83]. Une autre approche usuelle consiste à projeter sur un atlas classique du cerveau comme l'atlas de Talairach [99] ou de normaliser les informations du patient dans l'espace de Talairach après une projection des données sur une coupe représentative d'un volume IRM. Ces deux approches sont limitées en précision, l'anatomie du patient étant remplacée par une référence externe plutôt que d'utiliser son anatomie propre, et les données nécessaires à cette opération étant souvent absentes.

Un outil pour l'analyse topographique du cortex est donc essentiel pour les neurochirurgiens, non seulement pour comprendre la structure topologique, mais aussi pour localiser avec précision les éléments liés aux procédés opératoires. Il est nécessaire de disposer d'un outil de localisation précis, spécifique à chaque cerveau, et prenant en compte les spécificités morphologiques de l'individu. De plus, ce mode de représentation doit être adapté pour compenser les irrégularités des convolutions du cortex. Plusieurs auteurs ont ainsi proposé des reconstructions 2D de la surface corticale pour répondre plus efficacement aux limitations des représentations 3D.

Van Essen *et al.* [107] ont construit une surface lissée, puis projeté la surface corticale sur un ellipsoïde. Cette représentation offre un bon aperçu de la surface corticale, mais avec quelques limitations, notamment liées aux topologies souvent non triviales des surfaces extraites des données volumiques originales.

Warnking *et al.* [109] ont proposé un algorithme pour aplatir le cortex visuel. Après avoir projeté les nœuds dans le plan en utilisant des nœuds représentatifs comme amers, leur méthode itérative minimise la distorsion sur la distance géodésique. La principale limitation de ces méthodes est la non bijectivité entre la surface originale et la carte résultante : deux points sur la surface du cortex originale pourront être représentés à la même position sur la surface dépliée. Un outil de connexion entre les données 3D et la carte résultante est donc impossible à mettre en place.

K. Hurdal *et al.* [54] ont quant à eux calculé un dépliage bijectif d'un cervelet en utilisant 27 acquisitions IRM d'un même sujet, afin de corriger les erreurs topologiques potentiellement portées par une unique acquisition. Or, dans un protocole médical standard, autant d'images ne sont pas disponibles, et cette méthode n'est donc pas applicable.

Enfin, nous avons décrit dans la section 4.3.1 plusieurs approches de correction topologique de surfaces qui répondent partiellement à la problématique que nous nous proposons de résoudre dans ce chapitre.

Contrairement aux approches citées ici, la méthode que nous proposons ici est complète, depuis la segmentation jusqu'au dépliage, respecte à la fois la topologie et la géométrie des surfaces, et est applicable à des données IRM issues de la routine clinique.

7.2.2 Chaîne de traitement générale

Étant donné une acquisition IRM et une région d'intérêt, nous décrivons ici un processus complet permettant de produire une carte plane de la région d'intérêt. Après une détection puis reconstruction de la surface corticale, une correction topologique est appliquée, suivie d'un dépliage dans le plan. La carte résultante est finalement utilisée pour afficher des informations comme les aires ou la position des électrodes. La figure 7.1 décrit la chaîne

FIG. 7.1 – *Schéma du fonctionnement complet de la chaîne de traitement dans le contexte de la stimulation corticale.*

complète de traitement. Chaque étape de segmentation, dépliage et calcul de la carte est détaillée dans cette section.

Segmentation

Le volume IRM dont nous disposons est composé de voxels dont les niveaux de gris sont déterminés par l'anatomie du cerveau et des éléments environnants, tel que l'os du crâne. Avant de pouvoir manipuler la surface corticale, il est donc nécessaire de calculer les différentes structures que nous allons utiliser.

Dans un premier temps, nous avons appliqué un processus de suppression du scalp et de l'os (Brain Surface Extraction, ou BSE [92]), qui a récemment été présentée comme l'algorithme le plus performant comparé aux algorithmes classiques [42]. Cette méthode utilise d'abord un filtre de diffusion anisotropique pour réduire du bruit sur les données IRM tout en respectant les frontières des structures. Une détection de contours est ensuite effectuée, en utilisant l'opérateur de Marr-Hildreth. La plus grande composante connexe est alors détectée sur le volume binaire résultant par un processus morphologique (figure 7.2(b)).

Étant donné ce masque du cerveau, la segmentation des tissus est réalisée en utilisant une méthode floue de classification non supervisée, précisément l'algorithme Possibilistic Clustering [63]. On suppose ici que toute l'information présente est déterminée par la matière blanche, la matière grise, et le liquide céphalo-rachidien. Trois classes sont alors extraites, décrites par des cartes floues (figure 7.2(c)), qui représentent chacune la distribution de possibilités d'un tissu cérébral dans l'image. Il a été prouvé que la méthode utilisée ici calculait bien l'appartenance réelle des voxels aux classes de tissu [4], et non le degré de partage entre de toutes les classes comme l'algorithme flou classique C-means [6].

(a) Coupe d'un volume IRM avant segmentation.

(b) Masque résultant de la méthode BSE [92] de suppression du scalp et de l'os.

(c) Résultat de la segmentation par classification floue des trois tissus cérébraux.

FIG. 7.2 – *Résultat de la segmentation des tissus cérébraux d'un volume IRM après application d'une suppression de scalp et d'os.*

Reconstruction de surface

Les deux cartes d'appartenance floues à la matière blanche ($u_{MB,\cdot}$) et à la matière grise ($u_{MG,\cdot}$) sont alors utilisées pour produire le masque du cerveau.

Étant donné un coefficient $\delta \in [0, 1]$, un statut d'appartenance au parenchyme cérébral est calculé en utilisant pour chaque voxels v le critère d'appartenance $u_{MG,x} + u_{MB,x} \geq \delta$. L'image résultante est donc un volume binaire qui décrit l'appartenance au parenchyme. Les quelques défauts résultants éventuels sont alors supprimés en utilisant une opération morphologique comme une ouverture. Cette dernière étape produit un volume binaire qui définit le masque du cerveau. En pratique, on utilise une valeur fixée de δ ($\delta = 0.70$). L'algorithme des marching cubes [72] est ensuite appliqué pour construire le maillage de la surface corticale, en utilisant l'isosurface définie par le bord du masque. Dans l'idéal, la surface résultante est alors un 2-maillage tel que nous l'avons défini précédemment (définition 2.7). Certaines ambiguïtés peuvent cependant persister [82], mais nous n'avons pas eu à en souffrir, d'une part parce que les surfaces extraites dans le cadre applicatif choisi ne sont pas trop bruitées, mais aussi parce que le traitement topologique décrit au paragraphe suivant résout les éventuelles perturbations restantes.

Corrections et découpage topologique

On dispose donc maintenant d'un 2-maillage décrivant la surface corticale. Nous ne proposons pas ici de construire une carte plane de l'ensemble du cortex cérébral, mais uniquement d'une région d'intérêt. Ce choix a notamment été guidé par l'application visée (section 7.3), mais aussi par le fait que nous souhaitions appliquer un dépliage conforme (section 7.2.2).

En pratique, la région d'intérêt est décrite par une coordonnée 3D (correspondant au centre) dans le système de coordonnées de l'IRM, et par un rayon décrivant la taille du voisinage à considérer. On calcule donc l'intersection du 2-maillage et de la boule décrite par la région d'intérêt, puis on conserve la plus grande composante connexe du 2-maillage résultant afin d'obtenir la surface incluse dans la région d'intérêt. On dispose alors d'une surface discrète de genre g à n bords, dont on sait qu'elle résulte d'une acquisition basse résolution qui a pu introduire des erreurs topologiques.

On choisit donc d'appliquer ici la méthode de correction topologique par patches décrite à la section 4.3, en laissant à l'utilisateur la possibilité d'ajuster la longueur maximale

des α-jonctions qui sont corrigées. Une fois appliquée cette correction, la surface discrète n'est pas nécessairement homéomorphe à un disque, et il est nécessaire de lui appliquer un découpage topologique (section 3.3) afin d'achever sa préparation en vue d'un dépliage.

Remarque : En ajustant légèrement l'étape de correction topologique, il est possible de produire une surface complète du cortex, avec une topologie valide, c'est-à-dire un genre et un nombre de bords nuls. Pour cela, on choisira de ne pas appliquer de sélection par région d'intérêt, et on appliquera la méthode de correction topologique par patches à l'ensemble de la surface, en traitant toutes les α-jonctions sans limite de longueur.

Dépliage

On dispose donc maintenant d'une surface homéomorphe à un disque correspondant à la région d'intérêt. Cependant, nous n'avons aucun contrôle sur la forme des bords de ce disque topologique. L'utilisation d'une méthode de dépliage à bord fixe est donc à proscrire, car il est impossible de placer sans plus d'information les points du bord dans le plan.

D'autre part, l'objectif étant de produire des cartes de la surface qui soient lisibles simplement par des chirurgiens, il est nécessaire d'obtenir des cartes les moins éloignées pour l'œil expert d'une vision 3D classique. Nous avons donc choisi d'utiliser une transformation conforme de la surface, afin de préserver les angles, et ainsi d'assurer une certaine cohérence sur la position des sillons les uns par rapport aux autres.

Comme nous l'avons indiqué à la section 4.1.2, notre choix s'est porté sur une première méthode de dépliage quasi-conforme, nommée circle packing [20], qui offre une bonne robustesse et a déjà été appliquée dans un cadre médical [54]. Nous avons également choisi d'utiliser une méthode de dépliage conforme – Angle Based Flattening [94] – qui produit un dépliage à la conformité non plus approchée, mais exacte.

7.3 Application à la stimulation corticale

La stimulation chronique du cortex moteur (SCM) par l'utilisation d'électrodes péridurales dans le contexte du contrôle de douleurs chroniques a été mise en évidence à de nombreuses reprises [84, 8, 85]. Cette pratique chirurgicale consiste à implanter des électrodes entre la dure-mère et l'os du crâne, au dessus du sillon central, et à installer un boîtier de stimulation qui délivrera des impulsions électriques régulièrement.

7.3.1 Contexte médical

Le positionnement des électrodes, et donc la localisation des sites d'implantations, est essentiel dans la qualité du contrôle des douleurs. Cependant, le choix de la position des électrodes est aujourd'hui en partie déterminé par une approche empirique. Une analyse comparative *a posteriori* de la qualité des soins suivant la position des électrodes est un élément essentiel pour affiner les choix futurs des praticiens, comme nous l'avions remarqué dans une publication préliminaire [40] de la méthode que nous présentons ici.

Or la localisation d'informations à la surface du cortex n'est pas facilitée par l'utilisation de représentations 3D. De plus, l'anatomie de chaque patient ne pouvant être identifiée à un volume de référence, il est nécessaire d'envisager une représentation adaptée au patient, et facile d'accès. La cartographie planaire de la région d'intérêt du cortex cérébral constitue donc une réponse pertinente à ces besoins.

7.3.2 Construction des cartes

Afin de répondre aux besoins de visualisation des neurochirurgiens qui pratiquent la stimulation corticale, nous avons mis en place différentes projections sur les cartes dépliées, en exploitant les différentes modalités disponibles. Pour cela, on calcule les informations nécessaires en chacun des points de la surface du cerveau, dans la zone d'intérêt, puis l'on s'appuie sur le résultat du dépliage (section 7.2.2) pour reporter les valeurs sur la carte plane.

Électrodes

La première information que nous avons choisi de projeter sur les cartes dépliées est la position des électrodes. Puisque cette information ne peut être ponctuelle (car la projection n'est pas forcément unique), nous avons choisi de construire une carte des distances pour chacun des contacts composant les électrodes (figure 7.3).

Ainsi, pour un contact c, on construit une carte des distances euclidiennes qui à chaque sommet s de la surface discrète associe la distance euclidienne $|c, s|_3$ de $\mathbb{R}^3$. Une fois projetée dans le plan par la fonction résultat du dépliage, on obtient une carte avec un « point chaud » correspondant à la partie de la surface la plus proche de c.

En construisant une telle carte pour chacun des contacts c de l'électrode, et en fusionnant ces cartes par addition, on obtient une carte globale permettant de lire la position de chacune des électrodes à la surface du cortex. Pour des questions de lisibilité, nous avons choisi d'inverser sur la carte finale les valeurs afin que les zones blanches correspondent aux points les plus proches des électrodes (figure 7.6(b)).

Profondeur à l'os

La deuxième carte que nous avons choisi de construire a pour objectif de faciliter la lecture, en introduisant des informations anatomiques connues pour le praticien. La forme des sillons et convolutions, si elle varie d'un individu à l'autre, constitue un élément structurant fort de la surface du cerveau. Cette deuxième carte a donc été construite de manière à permettre aux praticiens de retrouver ces informations.

Nous avons pour cela calculé en chaque sommet s du maillage décrivant la surface du cerveau la distance euclidienne entre s et l'os du crâne. Cette information, interprétée comme une profondeur, et calculée par une méthode approchée (type distance de chanfrein) est là aussi projetée sur la carte en utilisant le résultat du dépliage. On obtient alors une carte en niveaux de gris (figure 7.6(a)), que nous avons là aussi choisi d'inverser, afin que les sillons soient représentés dans des teintes sombres.

Étiquetage manuel

La dernière carte que nous avons construit utilise un troisième type de données éventuellement disponibles. Ainsi, lorsque l'on cherche à situer avec efficacité les électrodes, il peut être utile de disposer d'une classification automatique ou d'un détourage manuel étiquetant les différents voxels suivant leur appartenance à une région anatomique donnée. Cette classification étant généralement réalisée dans le volume 3D, la fonction de projection dans le plan permet de construire une carte plane portant ces informations d'étiquetage (figure 7.6(c)). Si nous avons ici exploité les informations de segmentation manuelles disponibles, la méthode peut être également utilisée de manière plus générale,

FIG. 7.3 – *Schéma d'une électrode utilisée lors de la stimulation corticale.*

FIG. 7.4 – *Coupe issue d'un volume 3D résultat d'un détourage manuel des aires anatomiques*

par exemple en exploitant les informations issues du recalage d'un atlas anatomique sur le volume du patient.

7.3.3 Résultats et discussion

Nous avons appliqué notre algorithme à douze volumes IRM de patients présentant des douleurs neuropathiques et traités par SCM. La localisation anatomique de la cible a été calculée en utilisant un système de navigation 3D (ISIS, Saint-Martin d'Hères, France). Pendant l'opération, deux électrodes, chacune avec quatre contacts (voir figure 7.3), ont été implantées sur la dure mère près du sillon central.

Protocole

Le protocole expérimental fournit une IRM anatomique préopératoire (1.5T, taille de voxel $0.49 \times 0.49 \times 1$ mm^3) et un scanner postopératoire (taille de voxel $0.53 \times 0.53 \times 1.25$ mm^3) indiquant la position finale des quatre contacts des deux électrodes. On dispose également d'un détourage manuel des contacts des électrodes et des aires anatomiques (figure 7.4) telles que les circonvolutions frontales supérieure (F1), moyenne (F2) et inférieure (F3) ainsi que les circonvolutions post-centrale (GPost) et pré-centrale (GPre). Ces segmentations sont exprimées dans le référentiel de l'IRM après fusion des données TI et IRM réalisée en utilisant l'algorithme d'information mutuelle [110].

Étant données les informations disponibles, nous avons fixé le centre de la carte au barycentre des huit contacts détourés. Ainsi, les seules étapes manuelles de notre méthode dans ce contexte sont le choix du rayon de la carte, et la longueur maximale des α-jonctions. La méthode de dépliage complète et automatisée, depuis la segmentation jusqu'au dépliage, a été appliquée sur les douze patients.

(a) Carte produite sans correction topologique des α-jonctions.

(b) Interface interactive de navigation 2D/3D. Un curseur commun est visible sur les deux vues.

FIG. 7.5 – *Cartes produites par le dépliage avec et sans correction topologique des α-jonctions.*

Pertinence de la correction topologique

Afin de valider la pertinence de l'utilisation de la correction topologique des α-jonctions, nous avons également appliqué la totalité de la chaîne en supprimant l'étape de correction. Les topologies complexes sont alors prises en compte par l'outil de découpage topologique. Cette étape introduit alors de grands découpages à travers la surface, pour relier les bords aux α-jonctions. La complexité de la forme du bord qui résulte de ce découpage entraîne un dépliage conforme qui n'est pas satisfaisant, comme l'illustre la figure 7.5(a) : la bijection entre surface originale et le morceau du plan n'est pas du tout assurée, la carte est « froissée ». La figure 7.5(b), produite en appliquant sur la même surface la chaîne complète permet de comparer le résultat obtenu avec ou sans utilisation de la correction des α-jonctions.

Interface de navigation interactive

Si d'un point de vue théorique le dépliage conforme à bord libre n'assure pas une bijection entre surface initiale et surface dépliée, en pratique dans cette application, les cartes ne comportent pas de replis, excepté parfois sur les bords de la carte. Cette qualité du dépliage nous a permis de développer une interface de navigation 2D/3D, présentant simultanément une carte plane et une vue en coupe de l'IRM initial (figure 7.5(b)). La navigation interactive utilise le résultat du dépliage, et la propriété de bijection résultante pour localiser un curseur commun aux deux vues. L'utilisateur peut à sa guise piloter les vues depuis la carte plane, ce qui entraînera une mise à jour de la vue 3D avec éventuel changement de coupe, ou utiliser la vue 3D pour afficher sur la carte plane le point correspondant le plus proche.

En présentant dans cette interface l'ensemble des cartes générées (section 7.3.2), les praticiens disposent d'une visualisation complète et fonctionnelle des différentes propriétés anatomiques, fonctionnelles et applicatives.

La jeunesse d'une telle représentation et son originalité rendent difficile son évaluation par des praticiens. Cependant, les deux neurochirurgiens avec qui nous avons travaillé autour de ce développement ont pu tester le prototype, et constater que la vue planaire, couplée à la navigation interactive, permettait une visualisation efficace des différentes informations utiles.

(a) Carte de la distance à l'os (dépliage par ABF).

(b) Carte de la distance aux contacts des électrodes (dépliage par ABF).

(c) Carte de la segmentation manuelle (dépliage par ABF. En vignette, coupe d'une vue 3D correspondante).

(d) Carte de la distance à l'os (dépliage par circle packing).

(e) Carte de la distance à l'os (dépliage par ABF, avec approximation des plus courts lacets non séparants).

FIG. 7.6 – *Visualisation d'informations en utilisant la projection issue d'une IRM anatomique.*

Temps d'exécution

Dans le cas d'un traitement postopératoire, voire même dans le cas d'une préparation d'intervention guidée par l'image, les temps de calculs que nous avons obtenus pour la chaîne de traitement complète sont largement satisfaisants. En effet, sur les données que nous avons exploitées, et pour une zone d'intérêt d'un rayon de 120 mm, la génération des cartes depuis l'IRM init ale nécessite moins d'une heure sur un ordinateur portable Intel® Core™ 2 Duo cadencé à 2.60 GHz.

Plus précisément, le temps d'exécution sur un exemple typique est le suivant : environ 10 secondes pour calculer le masque de suppression du scalp et de l'os, puis environ 4 minutes pour la segmentation et le calcul du masque du cerveau. 6 minutes sont ensuite nécessaires pour appliquer la correction des α-jonctions (on a ici choisit $\alpha = 15$ mm). Le découpage topologique final prend lui 18 secondes. Nous avons testé à la fois un dépliage par circle packing (figure 7.6(d)), qui prend environ 15 minutes, et un dépliage par ABF (figure 7.6(b)), dont le temps d'exécution est sur l'exemple cité de 25 secondes.

À noter ici que les courbures gaussiennes importantes dues aux correction des α-jonctions ne facilitent pas la convergence de ces algorithmes. Ainsi, la première version d'ABF que nous avons utilisée manipulait des flottants simple précision, et divergeait sur de nombreuses surfaces issues d'acquisitions IRM. La version d'ABF embarquée par

`Graphite`[1] manipule elle des flottants double précision, et a déplié avec succès les différentes surfaces corticales que nous avons traité.

Enfin, nous avons également modifié la chaîne d'exécution pour intégrer l'algorithme d'approximation du lacet non séparant le plus court (section 3.3.4), à la fois pour le découpage topologique, mais également lors de la correction des α-jonctions. Le premier point à noter, auquel nous nous attendions, est que l'on doit augmenter la longueur maximale des α-jonctions prises en compte par les corrections topologiques. Sans cela, très peu de jonctions sont prises en compte, car les lacets trouvés sont nécessairement plus longs que les lacets optimaux. D'un point de vue temps de calcul, sur l'exemple cité précédemment, si l'on choisi une longueur maximale de 5 cm, alors la correction des α-jonctions nécessite 12 secondes, et le découpage final prend 10 secondes. Nous avons constaté que les cartes générées (figure 7.6(e)) par cette méthode d'approximation n'ont pas la régularité des cartes produites par l'algorithme exact.

7.4 Conclusion et perspectives

Comme nous l'avons présenté au fil de ce chapitre, la complexité des données issues d'acquisitions médicales nécessite d'envisager des traitements élaborés, tant d'un point de vue traitement d'images que manipulation de surfaces. Ainsi, la méthode que nous avons présenté ici prend en compte à la fois des informations géométriques et topologiques pour répondre aux besoins spécifiques du domaine.

La chaîne de traitement que nous avons présentée ici est suffisamment générique pour être appliquée dans différents contextes médicaux. Nous avons détaillé comment les cartes générées permettaient de visualiser les informations utiles à une analyse postopératoire des résultats d'une stimulation corticale. Pendant notre collaboration avec les neurochirurgiens pratiquant ces stimulations, nous avons évoqué la possibilité de construire, toujours en utilisant le dépliage, d'autres cartes qui pourraient leur être utiles. Ainsi, une carte représentant l'épaisseur de la matière grise, utile dans l'étude de la maladie d'Alzheimer pourrait être construite. Une carte plane représentant le potentiel électrique généré par les électrodes pourrait également être calculée en utilisant les propriétés de continuité électrique des différents tissus. Ces pistes n'ont pas été explorées par manque de temps, mais aussi par manque de données pertinentes.

D'autre part, si nous avons décrit l'utilisation de ces cartes surfaciques dans une utilisation postopératoire, pour l'analyse des résultats, l'ensemble des méthodes présentées ici pourrait parfaitement être utilisée pendant la phase préopératoire pour visualiser les zones cérébrales à stimuler avant implantation des électrodes. L'ajustement virtuel des électrodes pourrait d'ailleurs être étudié, la projection des positions de ces dernières sur la carte étant une étape peu coûteuse en temps.

Enfin, la méthode de correction des α-jonctions est une approche pertinente pour réduire le genre de la surface corticale à zéro, afin d'obtenir une surface homéomorphe à une sphère. Les applications d'une telle transformation sont alors multiples. Le recalage des surfaces sur une surface de référence ou dans une approche inter-sujets pourrait permettre d'étudier les propriétés statistiques des configurations anatomiques, par exemple dans la détection de pathologies de type Alzheimer. Le recalage de surfaces issues de données acquises chez un même patient à plusieurs étapes de l'évolution de sa maladie permettrait également d'en étudier la progression. Ces différentes idées sont à l'origine d'un

[1] `Graphite` est une plateforme dédiée à la recherche en synthèse d'image, modélisation 3D et géométrie numérique. `http://alice.loria.fr/software/graphite/`

partenariat PHC FAST 2010 avec Olivier Salvado et Oscar Acosta du groupe d'imagerie biomédical CSIRO de l'*Australian e-Health Research Center* (AEHRC), où l'on envisage une approche de correction topologique globale pour étudier sur un grand ensemble de patients les variations anatomiques dans le cadre de la maladie d'Alzheimer [46].

Conclusion

Les travaux de cette thèse s'articulent autour de l'utilisation d'informations topologiques et géométriques pour le pavage de surfaces. Nous nous sommes d'une part intéressés à la formalisation d'un cadre descriptif de ces pavages, en nous appuyant sur des modèles combinatoires de la géométrie discrète. Puis nous avons exploité ce formalisme pour faciliter la description d'algorithmes de découpages, répondant successivement à des contraintes topologiques, géométriques, ou encore de forme des surfaces résultantes. Enfin, nous avons exploré les possibilités offertes par ces formalismes et algorithmes, en les déclinant dans un cadre applicatif où ils ont permis d'apporter des réponses à des besoins spécifiques.

Si la structure générale de ce manuscrit laisse apparaître un cloisonnement ferme entre les différentes parties de nos travaux, dans les faits elles sont intimement liées. D'une part, dans le sens direct, on a pu constater que chaque partie exploitait fortement les résultats présentés plus tôt, en les utilisant à la manière d'outils pour répondre à des besoins toujours plus spécifiques. D'autre part, dans le sens indirect, il est évident que les idées que nous avons présentées en amont sont guidées par les besoins des chapitres suivants.

C'est ce lien fort entre formalisme et application qui a été l'un des moteurs de nos travaux, dont nous résumons ici les principales contributions.

Nous nous sommes d'abord intéressés à la modélisation de pavages de surfaces, en cherchant à décrire à la fois les propriétés combinatoires, géométriques et topologiques de ces pavages. Nous avons ainsi introduit la notion de M-complexes cellulaires, qui permet de décrire de manière générale l'interaction entre éléments de différentes dimensions ne disposant pas nécessairement d'une topologie triviale. Nous avons ensuite proposé le formalisme des M-pavages, permettant de décrire à l'aide de M-complexes cellulaires un pavage quelconque de surface.

La raison d'être de ce formalisme était de pouvoir décrire l'ensemble des pavages possibles sur une surface, tout en enrichissant cette description d'informations combinatoires et topologiques. Ainsi, nous avons tout d'abord établit une formule d'Euler généralisée sur les M-complexes cellulaires, qui synthétise de manière concise la combinatoire et la topologie des pavages. Nous avons également au fil de ce mémoire profité de ce formalisme pour décrire des pavages aux propriétés particulières, que ce soit en terme de topologie avec les pavages en cylindres ou de combinatoire avec les pavages en quadrangles.

Nous avons également défini au chapitre 1 une généralisation de la notion de lacet en introduisant les n-cets, composés d'un ensemble fixe de chemins reliant deux points base.

Nous avons alors établi quelques résultats sur l'arité d'un n-cet suivant le genre de la surface le portant, et proposé un algorithme de construction de n-cet valide, en l'exploitant par la suite pour produire des pavages de surfaces. Ainsi, après avoir souligné l'intérêt majeur du découpage suivant des n-cets pour le pavage de surfaces en tuiles cylindriques,

nous avons décrit une méthode d'optimisation permettant de produire efficacement un tel pavage. Cette approche a ensuite été exploitée au chapitre 6 dans le cadre applicatif de l'infographie, afin de produire une segmentation de formes organiques.

Outre cet algorithme de découpage, nous avons proposé plusieurs algorithmes de découpage, répondant successivement à des contraintes d'ordre topologiques et géométriques.

Dans un premier temps, nous avons proposé une variante de algorithme d'Erickson et Har-Peled [34] de découpage minimal en une surface homéomorphe à un disque. En ajustant notamment la complexité de la méthode par des raffinements, on améliore ainsi nettement l'efficacité pratique de l'algorithme original.

Nous avons ensuite proposé deux algorithmes intégrant des notions géométriques. La première méthode exploite une étape de transformation conforme pour déterminer un ensemble de sommets extremums de la surface, et construit un découpage respectant ces sites particuliers. Dans le cadre applicatif de l'infographie, nous avons montré comment cette approche permettait de proposer un dépliage efficace de la surface d'un modèle 3D, en tenant compte des formes de la surface sans limitation d'échelle.

Le deuxième algorithme géométrique que nous avons proposé exploite les configurations locales spécifiques de la surface pour en corriger la topologie, par un découpage suivi de deux coutures. L'une des applications premières de cette méthode est la correction de données issues d'acquisitions basse résolution. Nous avons ainsi présenté dans le dernier chapitre de ce manuscrit un cadre applicatif complet en imagerie médicale, montrant comment cet algorithme permettait de construire une carte plane de la surface corticale en le couplant à un découpage topologique initial. Une chaîne complète de traitement de données a été proposée pour générer ces cartes porteuses d'information depuis les données volumiques brutes.

Enfin, nous avons proposé deux approches pour produire des pavages en quadrangles, en profitant de la description des M-pavages pour synthétiser ces approches. Le premier algorithme s'appuie sur un premier découpage en une surface homéomorphe à une disque, alors que le second profite des bonnes propriétés combinatoires des pavages en cylindres par n-cets pour proposer des quadrangles peu distordus.

Un certain nombre d'éléments proposés dans ce manuscrit nous semblent devoir être améliorés, ou encore prolongés.

Si les possibilités de description que permettent les M-pavages ont partiellement été explorées, en décrivant des pavages par tuiles non planaires ou quadrangulaires, il nous semble intéressant d'envisager des prolongements à l'utilisation de cet outil.

Ainsi, les M-pavages nous semblent être un bon outil pour abstraire les segmentations de pavages. Une utilisation envisageable serait de guider les segmentations d'une surface, à la manière des g-cartes qui permettent de décrire les segmentations volumiques [32]. De la même manière, les M-pavages pourraient être utilisés afin de comparer deux pavages de surfaces, en mesurant par exemple le nombre de divisions et recollements à effectuer sur les M-tuiles, tout en profitant des propriétés topologiques de ces dernières.

Parmi les idées présentées dans ce mémoire, les n-cets nous semblent être l'un des éléments les plus propices à la poursuite de ces travaux. En effet, de nombreuses questions algorithmiques restent ouvertes, et mériteraient une attention particulière.

Tout d'abord, nous avons proposé un algorithme permettant de calculer un n-cet valide à partir de deux points. Cependant, si $n > 2$, l'algorithme n'assure pas la construction du

plus court n-cet valide. Obtenir un tel algorithme nous paraît être un problème difficile, mais intéressant.

D'autre part, comme nous l'avons déjà évoqué lorsque nous avons décrit la méthode de pavage en cylindres par n-cets, cette approche n'est pas pleinement satisfaisante. Plusieurs autres approches sont ainsi envisageables, et mériteraient d'être explorées.

Ainsi, il semble exister un lien fort entre la structure d'un pavage en cylindres par n-cets et un graphe de Reeb associé à la surface. Cette relation est évidente dès lors que l'on considère le graphe dont les nœuds sont les n-cets, et dont les arêtes sont construites suivant les cylindres du pavage. Il devient alors naturel de se demander s'il est possible d'aider au positionnement des n-cets en s'appuyant sur la construction préalable d'un graphe de Reeb de la surface.

Une autre relation forte existe entre le pavage d'une surface en cylindres par tricets, et le pavage d'une surface en pantalons. En construisant le graphe associé au pavage en pantalons, où les nœuds sont les pantalons, et les adjacences entre pantalons formant les arêtes, on constate que le graphe est identique à celui décrit ci-dessus pour un pavage en cylindres par tricets. Là encore, cette dualité pourrait être exploitée pour améliorer l'un ou l'autre des pavages.

Enfin, nous avons décrit dans les deux derniers chapitres des applications au pavage de surfaces, d'une part pour l'infographie, et d'autre part dans le contexte de l'imagerie médicale. La dernière section de ces deux chapitres propose déjà les perspectives spécifiques à chacun des domaines, aussi nous ne les reprendrons pas ici.

Cependant, de manière plus générale, nous avons constaté en explorant ces deux champs d'applications que de nombreuses méthodes issues de la géométrie algorithmique y auraient leur place. Nous avons également pu constater que si les problématiques de ces deux domaines semblaient *a priori* relativement éloignées, les méthodes de pavages décrites dans ces travaux constituaient l'un des exemples de la grande proximité de leurs besoins. Prolonger et entretenir les relations entre ces différentes communautés constitue donc l'une des perspectives intéressantes de ces travaux.

Annexes

Annexe A
Taglut

Sommaire

A.1 Langages et bibliothèques 126
A.1.1 Solutions existantes 126
A.1.2 Réalisation d'un outil original 127
A.2 Structures de données et implémentation 128
A.2.1 Maillage 128
A.2.2 Calcul de distances 129
A.2.3 Pavage 130
A.2.4 Dépliage 130
A.2.5 Flexibilité de l'outil 131
A.3 Échanges et interactions 132
A.3.1 Formats d'échange 133
A.3.2 Exécutables spécifiques 134
A.3.3 Adaptateur python 134
A.3.4 Interfaçage avec blender 135

Nous décrivons dans ce chapitre les développements logiciels qui ont été réalisés pendant cette thèse. En effet, comme la partie applicative le laisse supposer, nous avons implémenté la majeure partie des algorithmes décrits dans ce manuscrit, afin d'étudier leur comportement, et de manipuler des données réelles, par exemple issues de l'imagerie médicale (chapitre 7).

Dans une première partie, nous décrivons les bibliothèques et logiciels existants qui auraient pu être la base des développements de cette thèse. Nous présentons ensuite les raisons d'un choix différent, ainsi que les briques logicielles qui ont finalement été utilisées pour le développement.

Dans une deuxième partie, nous présentons quelques choix d'implémentation qui ont été effectué au cœur de `Taglut`[1], en soulignant les liens avec les modèles et algorithmes correspondants. Nous montrons notamment que des choix ont été réalisés pour favoriser la performance des exécutables, ou encore pour permettre une souplesse dans les développements futurs.

[1] `Taglut` : Topological and Geometrical Library - a Useful Toolkit

Enfin, la dernière partie de ce chapitre est dédiée aux possibilités d'échanges et d'interactions qu'offrent l'outil logiciel présenté ici.

A.1 Langages et bibliothèques

Au démarrage de cette thèse, l'ancrage applicatif était fort, porté en partie par les problèmes de dépliage de la surface du cerveau [40]. Le choix de solutions logicielles s'est alors rapidement posé. Plusieurs solutions existantes ont été envisagées, que nous détaillons dans une première section. Si nous avons finalement choisi de développer une solution alternative, les bibliothèques et outils logiciels qui la composent sont des éléments fiables et largement répandus.

A.1.1 Solutions existantes

Les prémices des idées explorées pendant ces trois années ont été réalisées à l'ERIM – Équipe de Recherche en Imagerie Médicale, une Équipe d'Accueil (EA 3295) basée à Clermont-Ferrand – et le développement logiciel avait été réalisé sous forme de modules à `ImageLib`[2], le logiciel développé au sein de ce laboratoire. Une solution aurait donc été de poursuivre sur cette plateforme, en profitant des développements déjà réalisés. Cependant, différentes limitations nous ont fait éliminer cette solution. Tout d'abord, le logiciel était uniquement prévu pour fonctionner avec Microsoft Windows. D'autre part, si la possibilité de manipuler des données discrètes volumiques était très complète, la manipulation de données surfaciques n'était pas au cœur de ses possibilités. Enfin, nous souhaitions pouvoir produire un logiciel sous licence libre, ce qui a définitivement éliminé `ImageLib`.

Les outils logiciels que nous allions développer étant essentiellement des prototypes, une solution aurait été de choisir `matlab` comme environnement de programmation. Cependant, nous ne l'avons pas retenu, d'une part parce qu'il n'offrait pas une grande souplesse en terme de manipulation d'objets abstraits, mais également en raison d'un manque de vitesse de traitement.

Nous nous sommes donc orienté vers des solutions compilées, et puisqu'il s'agissait de développer un outil logiciel spécifique et complet, il est rapidement devenu évident qu'il était préférable, non d'utiliser un logiciel complet, mais de s'appuyer sur une bibliothèque flexible et adaptable.

Parmi les bibliothèques de manipulation de données scientifiques, deux projets libres ont retenu notre attention, parce qu'ils correspondaient au mieux à nos critères. Ainsi, `VTK`[3] est un outil `C++` de visualisation et de manipulation de données, qui dispose de nombreuses méthodes issues notamment de l'imagerie médicale. De manière similaire, `CGAL`[4] est un projet `C++` regroupant des algorithmes de manipulation de données géométriques et topologiques.

Si ces deux bibliothèques auraient pu être des solutions envisageables, nous avons choisi de ne pas les retenir, en privilégiant une solution indépendante. D'une part, nous avons ainsi pu profiter d'une partie des algorithmes écrits pour les modules d'`ImageLib`, en reprenant une part de leur fonctionnement interne. D'autre part, nos travaux se plaçant à la frontière entre différents domaines, aucune de ces deux bibliothèques n'était totalement

[2] `http://www.u-clermont1.fr/erim/spip.php?article5`
[3] `http://www.vtk.org/`
[4] `http://www.cgal.org/`

adaptée à nos besoins. Enfin, la complexité de ces outils a été une des raisons de notre choix. Comme nous le verrons dans la section suivante, nous avons privilégié l'utilisation de bibliothèques dédiées et plus légères, afin de répondre à chaque fois à des besoins spécifiques.

A.1.2 Réalisation d'un outil original

Puisque nous avions choisi de développer une bibliothèque indépendante et efficace, nous avons opté pour le langage `C++` en raison de ses performances et du grand choix de bibliothèques existantes. Ainsi, nous avons au fil de ces trois années successivement ajouté des dépendances, pour répondre aux nouveaux besoins que nous rencontrions.

Le mode de développement a ainsi plutôt été guidé par les réflexions scientifiques que par une conception architecturale figée. En suivant les principes de conceptions des méthodes agiles, nous avons régulièrement travaillé à la refactorisation de l'architecture interne de l'outil, en l'ajustant à chaque étape aux nouveaux besoins, aussi bien en terme de généricité qu'en terme de nouvelles fonctionnalités. En suivant ces principes, nous avons donc favorisé la documentation développeur, grâce à l'outil `doxygen`[5], et en utilisant les possibilités offertes par `cmake`[6] pour automatiser les étapes de configuration et de compilation du projet.

L'une des premières bibliothèques que nous avons choisi d'utiliser a été `CImg`[7], un outil de manipulation d'images, à la fois léger et complet. L'importation, le traitement et la visualisation des données médicales volumiques ont ainsi été résolus simplement. Si nous avons utilisé cette librairie pour la visualisation de surfaces 3D, nous avons aussi rapidement utilisé `GLUT`[8] pour profiter des possibilités matérielles d'accélération graphique de la visualisation.

Un adaptateur python a été intégré pour permettre l'écriture de scripts de manipulation. Pour cela, nous avons fait le choix d'utiliser `SWIG`[9], un outil d'automatisation de ce type d'interfaçage. Cependant, l'évolution de l'outil logiciel n'a pas été linéaire, et une première interface scriptable avait été réalisée, cette fois-là en s'appuyant sur un dialecte XML spécifique.

Enfin, nous avons également intégré l'utilisation de différentes bibliothèques à usage scientifique. Tout d'abord, nous avons incorporé un outil de paramétrisation de surface, en profitant des possibilités offertes par `OpenNL`[10], un solveur de systèmes linéaires. Lorsque nous avons eu besoin d'intégrer un générateur pseudo-aléatoire fiable, notre choix s'est porté sur la librairie `GSL`[11], qui propose notamment une variante du générateur « Mersenne Twister » [76]. Enfin, nous avons implémenté une version parallèle avec mémoire partagée de l'algorithme de circle packing [20], en nous appuyant sur l'interface de programmation `OpenMP`[12].

La totalité des bibliothèques utilisées ici permet de distribuer sous une licence libre un outil les utilisant. Nous avons donc choisi de distribuer à la sortie de ces trois années notre code source sous licence GPL[13].

[5] `http://www.doxygen.org/`
[6] `http://www.cmake.org/`
[7] `http://cimg.sourceforge.net/`
[8] `http://www.opengl.org/resources/libraries/glut/`
[9] `http://www.swig.org/`
[10] `http://alice.loria.fr/index.php/software/4-library/23-opennl.html`
[11] `http://www.gnu.org/software/gsl/`
[12] `http://openmp.org`
[13] `http://www.gnu.org/licenses/gpl-2.0.txt`

Dans la section suivante, nous détaillons quelques choix d'implémentation qui ont été réalisés dans cet outil.

A.2 Structures de données et implémentation

La plupart des données que nous avons manipulées sont des 2-maillages (définition 2.7). Nous présentons ici la description logicielle que nous avons choisi d'adopter, ainsi que les différents outils que nous avons déployé autour de cette structure.

A.2.1 Maillage

Lorsqu'il s'agit de décrire en mémoire un maillage triangulaire, tel que ceux que nous avons choisi d'utiliser, différentes solutions sont envisageables. Ainsi, l'une des représentations les plus concises consiste à utiliser une liste de coordonnées, correspondant aux points du maillage, et une liste de triangles, ces derniers étant décrits par les identifiants des points associés. C'est ce type de description, très dense, qui est utilisé par la plupart des formats de fichier couramment utilisés (voir section A.3.1).

Cependant, la manipulation de données décrites en utilisant cette représentation n'est pas aisée. Tout d'abord, on remarque que pour connaître les sommets voisins d'un point donné, il est nécessaire de parcourir la liste entière des triangles. D'autre part, sorti du contexte de sa liste, impossible d'avoir un moyen d'identifier uniquement un sommet.

Afin de pallier ces limitations, la description de points de maillages peut être enrichie en y intégrant la liste des triangles auquel il participe, ainsi qu'une étiquette l'identifiant. Une manière immédiate de choisir cet identifiant est d'attribuer à chaque point son index dans la liste des points. Bien que cette approche soit coûteuse lors de l'insertion et suppression en milieu de la liste de points, elle offre un grand confort d'utilisation.

D'autre part, un grand nombre des algorithmes que nous avons décrits exploitent la liste des voisins d'un sommet donné. Il est même fréquent de vouloir parcourir cette liste de manière ordonnée suivant la topologie locale autour d'un point. Pour répondre à ce besoin spécifique, on peut par exemple enrichir chaque point de la liste de ses voisins.

Si ces informations sont redondantes, et donc augmentent l'espace mémoire utilisé par la description d'un maillage, elles offrent l'avantage d'une grande facilité d'utilisation, et améliorent d'autant l'efficacité des programmes en temps de calcul. C'est donc cette structure générale que nous avons décidé d'utiliser. Nous avons naturellement profité des possibilités de modélisation objet offertes par le langage `C++` pour décomposer les différents éléments (figure A.1), et les agrémenter d'opérateurs tels que translation, homothétie, calcul de normes, etc.

Remarque : Si nous avons choisi de décrire ici la structure par une « soupe de triangles », d'autres approches sont aussi envisageables, comme celle envisagée par l'outil `Moka`[14], un modeleur 3D à base de cartes combinatoires. Ce formalisme, tel que nous l'avons rapidement décrit à la section 2.4 s'appuie sur un élément de base, le brin (ou demi-arête). Si cet outil est un atout majeur au calcul dans les groupes d'homologies, il ne ferait qu'alourdir le fonctionnement interne de notre bibliothèque, où seule la longueur des arêtes est un outil essentiel.

[14] `http://liris.cnrs.fr/moka/`

FIG. A.1 – *Diagramme de classe simplifié décrivant la structure des points d'un maillage.*

FIG. A.2 – *Mécanisme de création de distances par* prototype *et* fabrique.

A.2.2 Calcul de distances

Comme nous l'avons présenté à de nombreuses reprises tout au long de ce manuscrit (sections 2.3.2, 4.1, 4.2.2, 5.4.3), de nombreuses informations globales, voire locales, peuvent être intégrées dans le processus de découpage en modifiant uniquement la notion de longueur des arêtes du maillage.

Il était donc indispensable que l'outil logiciel développé dispose d'une grande modularité sur ce point. Ainsi, les méthodes de recherche de chemin s'appuient toutes sur une implémentation de l'algorithme de Dijkstra, qui elle-même utilise un descripteur de distance interchangeable.

Pour faciliter le développement et l'ajout de futures distances, nous avons pris le parti d'écrire une interface décrivant le fonctionnement d'une distance. Ainsi, toutes les autres distances décrites dans la suite en héritent, qu'elles aient leur définition propre (distance euclidienne) ou qu'elles exploitent elles aussi une autre distance (modification par courbure moyenne).

Afin de faciliter l'utilisation de ces distances, nous avons utilisé les patrons de conception *prototype* et *fabrique* afin de faciliter la duplication d'objets distance. Ainsi, comme l'illustre la figure A.2, l'utilisateur de distances ajuste le comportement de la fabrique de distances `LengthFactory` en sélectionnant par descripteur le type de distance souhaité (`setLength(string)`), puis accède à une instance de distance souhaitée sur un maillage donné (`getLength(Mesh)`). Il pourra ensuite dupliquer cette instance à l'aide de la méthode `clone()`, sans se préoccuper de son type réel.

A.2.3 Pavage

L'une des originalités des travaux présentés dans ce mémoire est la notion de M-pavage (définition 2.21), qui permet de modéliser de manière abstraite une structure de surface. Cependant, la formalisation de ces idées étant arrivée assez tardivement dans l'évolution du logiciel, il n'a pas été intégré directement. Cependant, on retrouve implicitement ces idées dans le fonctionnement des outils de découpage.

En effet, l'ensemble des algorithmes de découpage fonctionnent de manière identique, en manipulant conjointement deux maillages. Le premier, non modifié, correspond au maillage d'entrée du découpage. Il correspond au 2-maillage S support des M-tuiles décrites à la section 2.4.2. Le second maillage quant à lui est mis à jour à chaque étape du processus de découpage, lorsqu'un chemin ou un lacet a été déterminé comme faisant partie du découpage final $\mathcal{C}$. Ainsi, ce maillage, qui correspondra à la fin du processus à $\mathscr{X}(S, \mathcal{C})$, porte les informations correspondant aux M-tuiles. En effet, on peut aisément calculer le nombre de bords de la surface, et déterminer à l'aide de la caractéristique d'Euler (section 2.2.3) le genre de chacune de ces tuiles, et ainsi retrouver les informations décrites par les M-pavages.

D'un point de vue pratique, cette modélisation du découpage par deux maillages permet de faciliter les calculs, puisque l'on peut notamment rechercher les plus courts chemins dans le second maillage, déjà partiellement découpé, sans avoir à se préoccuper par exemple des croisements de chemins.

A.2.4 Dépliage

L'algorithme de circle packing [20] a été le premier algorithme de paramétrisation implémenté dans `Taglut`. Comme nous l'avons décrit à la section 4.1.2, il existe un grand nombre d'autres approches pour répondre à la question du dépliage. Cependant, la méthode ABF [94] semblait la plus appropriée quand nous avons cherché à introduire une nouvelle méthode de dépliage conforme à bord libres. Nous avons de plus profité de la disponibilité d'une implémentation libre (intégrée à `blender`) pour faciliter son intégration dans notre outil.

Puisque nous souhaitions que l'outil soit extensible par la suite, nous avons tout d'abord opéré une refactorisation de l'architecture du logiciel, en utilisant d'une part une classe `Mapping2D3D` servant à décrire la correspondance entre surface initiale et surface dépliée, et d'autre part en créant une classe d'abstraction des outils de dépliage, nommée `UnfoldingMethod` (figure A.3). Cette classe décrit sans les implémenter différentes étapes des processus de dépliage : l'initialisation du processus, soit par le constructeur en utilisant les structures 3D, soit par une méthode alternative telle que nous la décrirons à la section B.4. Suit alors le dépliage proprement dit, avec la méthode `processUnfolding()`. Puis l'étape suivante, comme pour les méthodes de dépliage, consiste à calculer à partir des grandeurs optimisées, les coordonnées 2D de la surface dépliée (méthode `compute2DLocation()`). Enfin, un accesseur `getMapping()` permet à l'utilisateur de la classe de dépliage d'accéder à la description de la correspondance 2D/3D.

La raison première de cette intégration est le manque d'efficacité de circle packing. Parmi ses défauts, on peut retenir d'une part son manque de vélocité, très dépendant de l'initialisation, le manque de contrôle des points angles autour des points bords, et la quasi-conformalité, qui entraîne la création de triangles proches de triangles équilatéraux. Si l'introduction d'ABF a résolu ces limitations, nous avons tout de même cherché à améliorer le fonctionnement de circle packing, en explorant différentes pistes. L'annexe B

FIG. A.3 – *Diagramme de classe simplifié décrivant les méthodes de dépliage.*

décrit quelques pistes explorées, depuis le développement d'une variante originale de circle packing introduisant un meilleur contrôle des bords jusqu'à une approche multirésolution de dépliage. L'intégration de ces méthodes a été grandement facilitée par la refactorisation de l'architecture du logiciel évoquée plus haut.

A.2.5 Flexibilité de l'outil

Les différents éléments que nous avons décrit dans les sections précédentes constituent les élements de base que permettent de manipuler `Taglut`. Cependant, les possibilités offertes ne limitent pas l'utilisateur à des algorithmes de dépliage ou de découpage. Ainsi, nous avons eu l'occasion, lors de collaborations, de travailler sur des thématiques différentes de celles présentées dans ce mémoire. Le développement logiciel qui a accompagné ces réflexions a été possible en s'appuyant sur les fonctionnalités de `Taglut`.

Génération de textures procédurales

Nous avons travaillé avec Émilie Kœnig, Pierre Chainais, Marc Chevaldonné et Yan Gérard à la conception d'un algorithme de génération de textures procédurales sur maillages triangulaires, en transcrivant dans le monde du discret le principe des cascades de poissons composées [10].

Bien que ces travaux soient encore en cours de développement, la librairie `Taglut` a ainsi été la base d'un premier prototype servant à démontrer la faisabilité des idées explorées. La figure A.4 présente ainsi le résultat de la génération d'une texture procédurale sur la surface d'un maillage triangulaire de genre deux.

Estimation de la distorsion locale d'un nuage de points

Pendant l'été 2009, j'ai effectué un séjour département de sciences informatiques du Queen Mary College de l'Université de Londres, au sein de l'équipe de Lourdes Agapito. J'ai notamment travaillé à l'estimation de la distorsion locale d'un nuage de points décrivant une surface, en vue d'améliorer l'initialisation de méthodes d'optimisation de reconstruction de surfaces à déformations non rigides. La méthode développée est une variation de l'approche par offset proposée par Chazal *et al.* [13], et détaillée plus longuement dans l'annexe C. Nous avons là encore appuyé nos développements sur `Taglut`.

Fig. A.4 – *Texture procédurale générée sur une surface de genre deux.*

Fig. A.5 – *Normales d'une surface décrite par un nuage de points.*

La figure A.5 présente une estimation des normales locales d'un nuage de point calculée en utilisant `Taglut`.

A.3 Échanges et interactions

Plusieurs raisons ont motivé le développement des outils que nous présentons dans cette section. Tout d'abord, nous avons eu à manipuler au cours de ces trois années différentes données issues de différentes sources, depuis les données d'imagerie médicale jusqu'aux surfaces venant de l'infographie ou d'autres champs scientifiques tels que la géologie. D'autre part, nous souhaitions pouvoir profiter des possibilités offertes par les différents logiciels de manipulation de surfaces, comme par exemple `blender`, ou encore `vrweb`.

Ainsi, nous avons à la fois intégré à `Taglut` la possibilité d'importer et d'exporter plusieurs formats usuels, mais nous avons également développé un ensemble d'interfaces, à l'ergonomie variée, pour répondre aux différents besoins rencontrés, depuis une utilisation par scripts automatisés jusqu'à une interface graphique pour permettre des démonstrations aisées.

A.3.1 Formats d'échange

Le choix des formats supportés a essentiellement été guidé par les besoins qui sont arrivés au fil de ces trois années. Certains sont redondants, d'autres sont très spécifiques, mais l'ensemble constitue les bases d'une « caisse à outils » pour la manipulation de données discrètes.

Maillages

La majeure partie des données manipulées par l'outil sont des surfaces triangulées, ou « soupes de triangles ». Pour faciliter l'import de telles données, depuis les bibliothèques de maillage disponibles en ligne, mais aussi pour profiter des possibilités des modeleurs 3D, nous avons donc ajouté successivement plusieurs formats de fichiers décrivant des surfaces triangulées.

Le premier format supporté en entrée et sortie est le `VRML` (Virtual Reality Markup Language, dans sa version 1). Ce langage est une norme internationale ISO, dont l'objectif premier était à l'origine la description de mondes virtuels 3D et interactifs. Le format `PLY` (Polygon File Format) a également été ajouté comme format d'entrée et sortie. Il s'agit d'un format initialement prévu pour stocker des données acquises par scanner 3D, et proposé par l'Université de Stanford. Afin de faciliter l'interaction avec les graphistes d'Allegorithmic[15], nous avons également ajouté le format `OBJ` (Object 3D) en lecture et écriture, qui est l'un des formats classiques dans l'univers de l'infographie. Nous avons également intégré une possibilité d'export en `OFF` (Object File Format), qui est l'un des formats lus en entrée par les outils utilisant `CGAL`.

La particularité de tous ces formats est de disposer d'une syntaxe ASCII simple, ce qui facilite grandement les possibilités d'import et d'export.

Volumes 3D

Puisqu'une partie de nos activités a consisté à manipuler des données médicales (chapitre 7), et notamment des images volumiques issues d'acquisitions IRM, nous avons rapidement intégré la possibilité d'importer directement des images 3D, en proposant de détecter les isosurfaces à l'aide de l'algorithme marching cubes [72]. Nous avons pour cela profité des nombreuses possibilités offertes par la bibliothèque `CImg`, qui outre le format de fichiers bruts, dispose via `medcon`[16] du support du format `dicom`, très répandu notamment pour le stockage de données IRM.

Autres formats

Enfin, au fil de nos besoins, nous avons ajouté le support de différents formats, à commencer par les formats de stockage de données 2D/3D, permettant de décrire les dépliages de surfaces (ou UV mappings en infographie). Ainsi, nous avons étendu à ces données notre support des formats `PLY` et `OBJ`, et nous avons également utilisé pour nos besoins propres des formats basés sur un dialecte XML, afin par exemple de sauver les rayons obtenus lors du dépliage par l'algorithme de circle packing.

Pour nos besoins propres toujours, et pour faciliter l'échange des résultats de découpages en quadrangles, nous avons défini un format simple permettant de résumer les

[15] Allegorithmic est une société clermontoise développant des solutions logicielles de textures procédurales. `http://www.allegorithmic.com/`

[16] `http://xmedcon.sourceforge.net/`

principales informations d'un tel découpage. Considérant que les détails de ce format ne constituent pas un point essentiel à cette annexe, nous renvoyons le lecteur intéressé au code source et à la documentation développeur de `Taglut`.

Enfin, nous avons également intégré dans une moindre mesure la possibilité de charger des fichiers décrivant des nuages de points (formats `PLY` et `XYZ`), pour satisfaire à des besoins occasionnels.

A.3.2 Exécutables spécifiques

Si l'ensemble de `Taglut` peut être compilé comme une librairie indépendante, et être liée dynamiquement, nous l'avons entourée d'un certain nombre d'outils exécutables qui profitent des possibilités offertes par son interface de programmation (API).

Parmi ces exécutables, on peut citer des outils de visualisation de maillage et de dépliage (`meshViewer`, `mappingViewer`), ou encore un outil permettant de connaître les propriétés topologiques des surfaces, comme le nombre de composantes connexes, le genre ou le nombre de bords (`meshInfos`). Un outil de dépliage est également disponible (`meshUnfolder`), accompagné de nombreux outils de manipulation pour patcher (`meshPatcher`), simplifier (`meshPruner`), lisser (`meshSmoother`), découper (`meshCutter`), etc.

Le principe est ici de disposer d'un grand nombre d'outils, répondant chacun à une tâche simple, mais largement paramétrables, afin d'être intégrés par exemple à des scripts shell d'automatisation de traitement.

Nous avons également développé quelques exécutables spécifiques à une application donnée, comme `brainCorrection`, qui corrige topologiquement une surface extraite depuis une segmentation d'IRM de cerveau, ou `textureGenerator`, qui calcule globalement sur la surface une texture procédurale à l'aide d'une approche discrète.

Cependant, si ces outils indépendants répondaient à des tâches simples, nous avons rapidement eu envie d'automatiser certaines tâches, en s'affranchissant des contraintes imposées par l'utilisation de scripts shell (nombre limité d'opérations possibles, nécessité de sauver les maillages entre deux traitements, etc.). Dans un premier temps, l'exécutable `mScript` est venu compléter cet ensemble d'outils en proposant le chargement de scripts décrits dans un langage spécifique, développé comme dialecte XML. Nous renvoyons ici encore le lecteur intéressé au code source et à la documentation développeur et utilisateur, où l'on pourra consulter notamment le fichier de description du langage, au format `XSD`.

A.3.3 Adaptateur python

Enfin, pour faciliter les interactions avec d'autres programmes (et notamment `blender`, comme nous le verrons juste après), nous avons ajouté un adaptateur python à la librairie, afin de disposer de toutes ses possibilités directement dans un langage de script. Nous avons pour cela utilisé `SWIG`[17], un outil d'automatisation de ce type d'interfaçage. Nous avons ainsi pu à peu de frais proposer un module python reprenant chacune des classes et méthodes de `Taglut`, tout en préservant les performances de la librairie, puisque le cœur des exécutions est toujours réalisé par du code `C++` compilé.

L'existence de telles possibilités d'automatisation de traitements par scripts a entraîné le remplacement progressif par des scripts python aussi bien des exécutables que des scripts XML décrits précédemment. Les principaux atouts de python dont nous avons

[17] `http://www.swig.org/`

(a) (b)

(c) (d)

FIG. A.6 – *Captures d'écran de l'interface à* `Taglut` *dans* `blender`.

ainsi bénéficié sont sa facilité d'utilisation, le fait qu'il soit interprété – et ne nécessite donc pas de phase de compilation –, ainsi que la grande flexibilité du langage.

A.3.4 Interfaçage avec blender

`Blender` est un environnement libre complet permettant la modélisation, l'animation et le rendu de scènes 3D complexes. Développé par une communauté internationale très large, il est en constante évolution, et dispose de nombreuses fonctionnalités, notamment en terme d'automatisation. Ainsi, une grande partie du logiciel peut être pilotée par scripts, à travers une interface de programmation écrite en python.

Nous avons donc écrit un module pour `blender`, qui profite à la fois de l'API de ce logiciel, et de l'API de `Taglut`, pour proposer au cœur de l'interface graphique de `blender` les méthodes de notre librairie. La figure A.6 présente quelques captures d'écran de ce module. L'interface se décline en trois versions différentes : une version simple (figure A.6(a)) où seules les options principales des algorithmes sont proposées, une version avancée (figure A.6(c)) où une grande part des options sont accessibles, et enfin une version complète (figure A.6(d)) proposant de très nombreuses options d'ajustement des méthodes. Cette granularité permet d'offrir à différents publics l'accès à une librairie aux possibilités d'ajustement multiples.

Le panneau principal de cette interface est séparé en deux colonnes. La colonne de droite, portant pour titre *Cutting*, propose de sélectionner ou non chacune des étapes du processus qui sera exécuté. Ainsi, on peut ou non cocher *Patch borders*, *Patch*, *Topological cut*, *Geometrical cut*. Ces différentes étapes seront exécutées dans l'ordre haut vers bas quand l'utilisateur appuiera sur le bouton *Apply cut*. En face de chaque bouton étape coché, un sous-menu est déplié (figure A.6(b)) qui permet d'ajuster le comportement de la méthode correspondante.

D'autre part, la colonne de gauche regroupe dans sa partie supérieure un outil de dépliage (*UV mapping*), que l'on peut ajuster en choisissant par exemple l'un des algo-

rithmes de dépliage embarqué par `Taglut`. En dessous, plusieurs boutons permettent de construire un maillage correspondant au bord de la surface courante, ou encore d'afficher les informations topologiques du maillage sélectionné.

Les images non vectorielles présentes dans ce manuscrit ont ainsi été réalisées en utilisant *blender*, et en profitant au mieux du module `Taglut` décrit ici.

Annexe B
Variations autour du dépliage

Sommaire

B.1 Circle packing : descriptif rapide 137
B.2 Contraintes aux bords 138
B.3 Parallélisation 140
B.4 Multi-résolution 140
B.5 Conclusion 141

Nous avons précédemment décrit (section 4.1.2) les méthodes de paramétrisation de surfaces existantes. Les approches par dépliage ont ainsi retenu notre attention au cours de ces trois années. Parmi ces dernières, nous nous sommes particulièrement intéressés à la méthode ABF [94], produisant des dépliages conformes à bords libres. Cependant, nous avions initialement commencé à travailler avec l'algorithme de dépliage quasi-conforme circle packing [20].

Si la robustesse de circle packing est son principal avantage, son manque d'efficacité a été la raison première de l'intégration d'ABF. En effet, le manque de vélocité, très dépendant de l'initialisation, et le manque de contrôle des points angles autour des points bords ont motivécet abandon. Cependant, nous avons tout de même exploré quelques pistes d'amélioration de circle packing, que nous décrivons dans cette annexe.

B.1 Circle packing : descriptif rapide

Commençons d'abord par décrire rapidement le fonctionnement de l'algorithme circle packing.

Il s'agit d'une méthode itérative qui minimise la distorsion sur les angles formés par les triangles autour de chaque sommet. Pendant la phase d'initialisation, un cercle est associé à chaque sommet du maillage, la valeur de son rayon étant arbitrairement fixée.

Pendant le processus itératif, chaque cercle ne faisant pas partie du bord de la surface est traité. Son rayon est réajusté en n'utilisant que les rayons des cercles voisins. Pour réaliser ce réajustement, le rayon du cercle central est modifié de manière à faire coïncider la somme des angles formés par les cercles voisins avec 2π (figure B.1).

La reconstruction converge alors vers une solution unique, les ajustements se faisant de moins en moins importants. Lorsqu'ils atteignent un seuil défini par l'utilisateur, l'algorithme est stoppé et on peut reconstruire la surface.

FIG. B.1 – *Ajustement local des cercles dans l'algorithme de circle packing [20]. En haut : l'angle initial est inférieur à 2π, le rayon du cercle central est diminué. En bas : l'angle initial est supérieur à 2π, le rayon du cercle central est augmenté.*

Cette reconstruction se fait simplement, en posant dans le plan un segment, puis en ajustant successivement la position des autres sommets du maillage en fonction des angles déterminés algorithmiquement.

B.2 Contraintes aux bords

Dans un premier temps, nous nous sommes intéressés aux limitations de la méthode quant à la gestion des bords. En effet, dans la version proposée par les auteurs de circle packing, la somme des angles des triangles adjacents à un sommet du bord n'était pas pris en compte. Nous avons donc choisi d'implémenter une variante de circle packing, à bords dynamiques.

Nous sommes partis du principe que si une surface était déjà plongée dans le plan, son dépliage devait en garder le bord identique. Dans ce cas, la somme des angles autour de chacun des points bord devait rester certes inférieure à 2π, mais inchangée. Cependant, si cette contrainte est envisageable pour une surface plane, elle ne peut plus être imposée dès lors que la surface est courbe. En effet, si l'on veut pouvoir dessiner le bord dans le plan, il est nécessaire que la somme des angles α_i formés par les arêtes bords aient la propriété suivante :

$$\sum_i \pi - \alpha_i = 2\pi, \tag{B.1}$$

la figure B.2 illustrant cette propriété.

Or, cette relation n'est pas forcément vérifiée par les angles des bords d'une surface courbe. Nous avons donc choisi de garder la cohérence entre ces différents angles, en écrivant la contrainte sur les angles α'_i de $\mathbb{R}^2$ associés :

$$\alpha'_i = \alpha_i \cdot \frac{2\pi}{\sum_i \pi - \alpha_i}. \tag{B.2}$$

En réalisant cette modification, on conserve bien la forme initiale dans le cas où la surface est déjà plane, et on l'ajuste dans le cas où la surface est courbe.

FIG. B.2 – *La somme des angles des tangentes d'un polygone du plan est égale à* 2π

FIG. B.3 – *Application de l'algorithme circle packing (milieu) et circle packing avec contraintes aux bords (droite) sur deux maillages déjà plans (gauche).*

La figure B.3(a) présente le résultat des algorithmes de circle packing original et modifié, sur une surface initialement plane à la forme particulière. On constate dans un premier temps que la structure du maillage triangulaire, couplée à l'ajustement quasi-conforme de l'algorithme de circle packing produit une paramétrisation fortement perturbée (figure B.3(a) milieu). L'ajout de contraintes sur les bords semble solutionner cette limitation (figure B.3(a) milieu).

Cependant, en appliquant les deux mêmes algorithmes à une autre figure plane, constituée d'un nombre plus important de triangles internes (figure B.3(b)), on constate que si les bords ont la forme attendue avec notre version modifiée de l'algorithme, la régularité des triangles après dépliage imposée par la méthode ne permet pas de produire un dépliage de qualité. À titre de comparaison, la méthode ABF produit elle des maillages identiques à ceux présentés à gauche sur les figures.

Si la méthode originale était déjà particulièrement peu véloce, cette nouvelle version a encore vu une forte augmentation des temps de calculs, ce qui nous a fait abandonner cette tentative d'amélioration de circle packing.

B.3 Parallélisation

Circle packing est un algorithme itératif, qui à chaque étape j du processus calcule pour chacun des cercles c_i un nouveau rayon r_i^{j+1}, en n'utilisant que les rayons des cercles voisins à c_i, $C_v(c_i)$. Il est important de noter ici que les rayons des cercles voisins utilisés pendant le calcul, les rayons des cercles $C_v(c_i)$ peuvent avoir ou non été mis à jour. Cela dépend uniquement de l'ordre de parcours de la liste. La stabilité de l'algorithme n'est pas perturbée par cet aléas, et l'on observe bien la convergence.

Partant de ce constat, nous avons implémenté une version parallélisée de l'algorithme de circle packing, en utilisant les possibilités offertes par l'interface de programmation d'`OpenMP`. Il s'agit d'une parallélisation que l'on pourrait caractériser de légère, puisque les différents processus s'exécutant simultanément partagent leur mémoire. Ici, l'information partagée est toute la structure du maillage à déplier, ainsi que les rayons des cercles associés aux sommets.

Pour chacune des étapes de l'algorithme itératif, l'ensemble des points à traiter est séparé en N sous-ensembles de taille identique. On créé alors N processus distincts, qui vont traiter chacun un sous-ensemble des points, en mettant à jour uniquement la valeur des rayons associés. Par cette méthode, on assure qu'il n'y ait pas d'accès concurrent en écriture, et on divise par N le temps de calcul nécessaire à une étape.

Ce type de parallélisation dispose de plusieurs avantages. Tout d'abord, il n'est pas nécessaire d'opérer de copies de l'état de la mémoire avant le lancement des processus. C'est là la fois un gain en terme de vitesse d'exécution, mais également en terme d'empreinte mémoire. D'autre part, la démocratisation des processeurs multicœurs et des ordinateurs multiprocesseurs permet de tirer avantage de cette approche, qui est parfaitement adaptée à ce type d'architecture. `OpenMP` propose d'ailleurs de détecter automatiquement, lors du démarrage de l'application, le nombre de processeurs disponibles sur la machine, afin d'adapter le nombre de processus parallèles qui seront lancés.

B.4 Multi-résolution

Nous nous sommes ensuite intéressés à l'initialisation de l'algorithme circle packing. En effet, la vitesse de convergence de cette méthode d'optimisation est en grande partie déterminée par la qualité de l'initialisation. Nous avons donc cherché à explorer les possibilités d'une approche multirésolution.

L'algorithme B.1 décrit la structure générale de l'algorithme de dépliage avec initialisation basée sur un dépliage appliqué à un maillage plus simple. D'un point de vue implémentation, nous avons profité de la description générique des algorithmes de dépliage (figure A.3) pour autoriser l'exécution de la méthode multirésolution avec tous les algorithmes de dépliage. La méthode `clone()` permet ici de dupliquer l'algorithme de dépliage pour l'appliquer à une résolution différente sans pour autant connaître la classe utilisée. La méthode `initDataUsing2D()` quant à elle permet d'initialiser le dépliage d'un niveau n grâce au résultat du dépliage du niveau $n-1$.

L'outil de simplification de maillage que nous avons utilisé dans ce processus de multirésolution supprime du maillage au plus la moitié des sommets, en évitant de supprimer les sommets géométriquement trop importants, c'est-à-dire ayant une courbure locale importante.

En guise de conclusion, on peut constater que l'initialisation par interpolation d'un dépliage de surface simplifiée n'est pas efficace. En effet, la précision nécessaire sur les

Algorithme B.1 : Dépliage multirésolution

Données : M un 2-maillage, n la profondeur de résolution, D la méthode de dépliage à utiliser
Résultat : (M, M') dépliage de M, avec M' maillage dans $\mathbb{R}^2$
si $n = 0$ **alors**
  $(M, M') \leftarrow D(M)$;
fin
sinon
  $M_d \leftarrow$ simplification(M);
  $(M_d, M'_d) \leftarrow$ Dépliage multirésolution$(M_d,\ n-1)$;
  Initialiser D sur M avec (M_d, M'_d), en interpolant ;
  $(M, M') \leftarrow D(M)$;
fin

longueurs des rayons est telle que l'initialisation décrite ici ne permet pas de réduire de manière marquante le processus de convergence de circle packing.

B.5 Conclusion

Les différentes modifications présentées dans ce chapitre ont partiellement répondu aux limitations de circle packing, essentiellement sur la question de la vélocité. Cependant, ces améliorations ne sont pas significatives à nos yeux, et nous avons donc définitivement opté pour l'utilisation d'ABF comme méthode de dépliage par défaut dans la suite de nos travaux.

Ces différentes expériences ont tout de même permis d'une part de constater que la quasi-conformité était loin d'approcher la conformité, comme le laisse entrevoir les différents résultats présentés à la figure B.3, et d'autre part de proposer un cadre d'abstraction des méthodes de dépliage, lequel nous a été utile dans la mise en place de l'approche multirésolution.

Annexe C
Géométrie des nuages de points

Sommaire

C.1 Travaux similaires 144
C.2 Du nuage de points à l'estimation de distortion globale 144
C.2.1 Estimation des normales 144
C.2.2 Estimation des bords 147
C.2.3 Distortion locale 147
C.2.4 Distortion globale comme estimateur de forme 148
C.3 Résultats et perspectives 148
C.3.1 Résultats 148
C.3.2 Application à la capture de mouvements 150
C.3.3 Perspectives 150

Nous avons présenté au fil des chapitres précédent différents outils de manipulation de surfaces, qui exploitent les propriétés géométriques et topologiques de ces objets. Cependant, les données en trois dimension ne se résument pas aux surfaces. Outre les données discrètes volumiques, dont nous avons présenté au chapitre 7 un exemple d'utilisation, une autre forme de données en trois dimension est le nuage de points. Ce type de données est une des formes les plus répandues de données brutes issues d'acquisitions, grâce notamment aux possibilités offertes par les scanners 3D. C'est également le type de données que produisent les différents travaux autour de la capture de mouvements, qui reconstruisent ces nuages de points à partir d'une série d'images en deux dimensions.

Lors d'un séjour pendant l'été 2009 comme invité au département de sciences informatiques du Queen Mary College de l'Université de Londres, j'ai travaillé au sein de l'équipe de Lourdes Agapito autour des questions de reconstruction de formes issues de capture de mouvements. En particulier, nous nous sommes intéressés à l'initialisation d'une méthode de reconstruction non rigide [41], et plus précisément à la mise en place d'un estimateur de distortion permettant de sélectionner le nuage de points 3D le moins distordu parmi un ensemble de nuages de points donnés.

Dans cette annexe, nous décrivons une méthode d'extraction d'informations géométriques à partir d'un nuage de points décrivant une surface, ces calculs préliminaires permettant d'estimer une valeur de distortion globale de la surface. Nous présentons ensuite quelques résultats et perspectives de cet outil.

C.1 Travaux similaires

L'extraction de descripteurs de forme à partir de surfaces décrites par nuages de points ou maillages triangulaires est une question très largement abordée [101]. Cependant, la plupart de ces méthodes visent à extraire des informations complètes quoi que synthétiques, afin par exemple de permettre les recherches dans une base de données d'objets 3D par similarité. Il n'existe pas à notre connaissance d'articles décrivant une méthode d'estimation de la distortion de surfaces, telle que nous présentons dans ces travaux.

La principale information que nous exploitons pour estimer la distortion globale est la normale à la surface en chacun des points du nuage. L'estimation de ces normales est un problème classique. Comme le soulignent Dey, Li et Sun [28], il existe deux grandes approches permettant d'extraire une telle information géométrique d'un nuage de points. La première consiste à utiliser une méthode d'optimisation sur un système de tenseurs associés aux points [100, 74], en utilisant le système de voisinage pour propager et lisser les normales. Cette approche nécessite donc de connaître les voisins d'un point donné, ce que l'on ne connaît pas *a priori* dans un nuage de points. La seconde approche est plutôt combinatoire, et consiste à utiliser un diagramme de Voronoï ou une triangulation de Delaunay pour extraire les normales [86, 101].

L'approche que nous présentons est une alternative à ces deux premières. Il s'agit d'une approche discrète de la méthode par offset présentée par Chazal *et al.* [14, 12]. Outre la discrétisation de l'approche, rendant possible des calculs pratiques, nous proposons une contribution originale à ces travaux, en estimant localement de la taille de l'offset. D'autre part, nous avons exploité de manière originale ces premiers résultats afin de calculer une approximation de la normale du nuage de points, et de la propriété de bord en chacun des points.

C.2 Du nuage de points à l'estimation de distortion globale

Soit $\mathcal{P}$ un ensemble de points dans $\mathbb{R}^3$. On cherche à décrire la distortion globale de la surface que décrivent ces points. Pour cela, on commence par estimer localement les informations géométriques nécessaires à l'estimation d'une mesure de distortion locale.

C.2.1 Estimation des normales

On cherche d'abord à estimer en chacun des points de $\mathcal{P}$ la normale à la surface que décrivent ces points.

Normale par offset

Étant donné un rayon d'offset $r \in \mathbb{R}$, et un point $p \in \mathcal{P}$, on note K_q^r la boule fermée de centre q et de rayon r. Soit K^r l'union des boules K_q^r, $q \in \mathcal{P}$. On sait que le bord ∂K^r de K^r est un polyhèdre sphérique [14] : ses faces sont des sous-ensembles surfaciques de ∂K_q^r, ses arêtes sont des arcs de cercles de $\partial K_i^r \cup \partial K_j^r$ (avec $i \neq j$), et ses sommets sont des points situés à l'intersection de trois ou plus K_q^r.

On appelle *surface d'offset* de p, notée $S_r(p)$, le sous-ensemble de ∂K^r associé à $K_p \cup \partial K^r$. Il s'agit d'une union de faces du polyhèdre sphérique ∂K^r. En utilisant

FIG. C.1 – *Estimation de l'axe normal par la méthode de l'offset.*

(a) (b) (c) (d)

FIG. C.2 – *(b)–(d) : subdivisions successives de l'icosaèdre original (a).*

cette structure de maillage sur ∂K^r, on définit le *voisinage* de $p \in \mathcal{P}$ par $N_r(p) = \{q \in \mathcal{P}; q \neq p, S_r(p) \cap S_r(q)\}$.

La figure C.1 présente un schéma 2D d'une telle construction. La partie rouge et $S_r(p)$, et $N_r(p) = \{n_0, n_1\}$.

En supposant que $S_r(p) \neq \varnothing$, on définit l'axe normal associé à p comme la moyenne des droites (p, q), avec $q \in S_r(p)$.

Discrétisation du calcul de la normale

Si d'un point de vue continu, l'axe est correctement défini dans la description précédente, il est nécessaire d'introduire une étape de discrétisation du problème afin de rendre calculable efficacement la normale associée à p, avec un offset de rayon r.

Tout d'abord, on décrit ∂K_p^r par une sphère discrète. En partant d'un icosaèdre, on applique N divisions successives (figure C.2), en positionnant à chaque étape les nouveaux sommets sur la sphère. La précision de la normale obtenue est partiellement dépendante de N. D'un point de vue pratique, une valeur de $2 \leq N \leq 4$ a étée utilisé pour générer les résultats expérimentaux (section C.3.1). On note $\partial \hat{K}_p^r$ le maillage obtenu par raffinement de l'icosaèdre, et $V(\partial \hat{K}_p^r)$ ses sommets.

Le sous-ensemble $\hat{S}_r(p)$ de $V(\partial \hat{K}_p^r)$ qui décrit $S_r(p)$ est calculé en comparant avec r chacune des plus courtes distances entre $v \in V(\partial \hat{K}_p^r)$ et $p' \in \mathcal{P} \setminus \{p\}$. Ainsi, seuls les sommets hors des autres boules font partie de $\hat{S}_r(p)$.

L'axe normal associé à p est alors calculé en utilisant une Analyse par Composantes Principales (ACP) des droites décrites par les vecteurs $(p_s - p)$, avec $p_s \in \hat{S}_r(p)$.

Finalement, le voisinage $\hat{N}_r(p)$ de p est calculé en construisant le bord $\partial \hat{S}_r(p)$ de $\hat{S}_r(p)$, constitué de l'ensemble des sommets de $V(\partial \hat{K}_p^r) \setminus \hat{S}_r(p)$ connectés par une arête à

(a) La surface d'offset peut être vide pour des points très entourés.

(b) La surface d'offset peut être une sphère pour des points isolés.

FIG. C.3 – *Rayons homogènes sur un nuage de points à la distribution non homogène.*

un sommet de $\hat{S}_r(p)$. De ce bord, on déduit $\hat{N}_r(p) \subset \mathcal{P}$ en associant à chaque point de $\hat{\partial S}_r(p)$ le point de $\mathcal{P} \setminus \{p\}$ le plus proche.

Si l'on considère qu'aucune structure de données particulière n'a été utilisée pour structurer les points de $\mathcal{P}$ dans $\mathbb{R}^3$, alors chacun des points de $\mathcal{P}$ doit être visité pour chaque sommet de $V(\partial \hat{K}_p^r)$. La complexité des l'algorithme calculant $\hat{S}_r(p)$ et celui calculant le voisinage de p est alors de $o(|\mathcal{P}||V(\partial \hat{K}_p^r)|)$. Puisque l'on applique ce calcul à l'ensemble du nuage de points, on obtient une complexité globale pour notre algorithme en $o(|\mathcal{P}||^2 V(\partial \hat{K}_p^r)|)$.

Estimation du rayon local

Dans le calcul de la normale associée à un point, le rayon r de l'offset est un paramètre global. Chazal *et al.* [14] fixent manuellement ce rayon à une valeur inchangée pour la totalité du nuage de points. Au delà de la question de la sélection d'une bonne valeur pour un nuage de points donné, cette approche semble être valide pour un nuage de points à la distribution homogène, mais difficilement applicable pour un ensemble non homogène (figure C.3).

Nous décrivons ici une approche qui, si elle n'est pas complètement satisfaisante, permet d'ajuster la valeur du rayon à chaque point du nuage.

Étant donné un point $p \in \mathcal{P}$, un rapport de couverture $c \in [0, 1]$ de la sphère et un angle maximum acceptable $\alpha_{max} \in [0, \pi/2]$, on estime le rayon $r(p)$ de l'offset associé à p de la manière suivante :

- Pour chaque point $p_j \in \partial \hat{K}_p^1$ de la sphère unitée discrète, on note $\overrightarrow{v_j} = p_j - p$. Étant donné ce vecteur, on estime le rayon seuil r_j à partir duquel p_j quitte $\hat{\partial S}_r(p)$ de la manière suivante :
  - Pour chaque $n_i \in \mathcal{P} \setminus \{p\}$, on calcule la longueur r_i, j des deux côtés égaux du triangle isocèle défini par $\overrightarrow{v_j}$ et $n_i - p$ (figure C.4). Pour des raisons pratiques, on fixe $r_{i,j} = \infty$ si l'angle $\alpha_{i,j} = \alpha(\overrightarrow{v_j}, n_i - p) \geq \alpha_{max}$, pour éviter les trop grandes valeurs de rayons qui produisent des sphères déraisonnablement grandes.
  - r_j est alors défini comme la valeur minimale des $r_{i,j}$.
- Le rayon $r(p)$ de la sphère associée à p est alors calculé en utilisant le r_j tel que $|\{p_j; r_j \leq r(p)\}| < c|\{p_j; r_j \neq \infty\}|$.

Le rapport de couverture c et l'angle maximum acceptable α_{max} en paramètres de la méthode permettent donc de contrôler la quantité de surface qui servira à calculer l'axe

FIG. C.4 – *Étape intermédiaire de l'estimation du rayon de l'offset associé à p.*

FIG. C.5 – *Estimation de la propriété de bord par la méthode de l'offset. q est un point bord, alors que p n'en est pas un.*

normal et les voisinages. Ainsi, chaque point du nuage disposera d'un rayon spécifique prenant en compte la géométrie locale du nuage.

C.2.2 Estimation des bords

En utilisant la surface d'offset discrétisée $\hat{S}_r(p)$ associée à chaque point p de la surface, il est possible de déterminer si p est un point appartenant au bord de la surface décrite par $\mathcal{P}$. Pour cela, on utilise la plus grande composante connexe de $\hat{S}_r(p)$, notée $C_{max}(\hat{S}_r(p))$, et on calcule la moyenne des vecteurs $\tilde{n}(p) = (p, p')$, avec $p' \in C_{max}(\hat{S}_r(p))$.

En mesurant l'angle formé par $\hat{S}_r(p)$ et l'axe normal, on peut alors estimer l'appartenance au bord, comme l'illustre le schéma en deux dimensions de la figure C.5 : si cet angle est très petit, il s'agit d'un point intérieur, sinon il s'agit d'un point bord. En pratique, un seuil laissé à la discrétion de l'utilisateur permet d'ajuster la détection. Les exemples présentés à la section C.3.1 ont été réalisés en utilisant une valeur de seuil de $3\pi/8$.

C.2.3 Distortion locale

Dans le contexte applicatif que nous envisageons (section C.3.2), les nuages de points sont peu denses, et une grande partie des informations de courbure et de distortion peuvent être observés localement. Aussi, nous avons choisi d'exploiter cette information purement locale comme descripteur de distortion.

On estime localement la *distortion locale de surface*, en calculant pour chaque point $p \in \mathcal{P}$ du nuage une moyenne pondérée des angles formés par l'axe normal de p et de l'axe normale de chacun de ses voisins :

$$s(p) = \frac{1}{|\hat{N}_r(p)|} \sum_{n \in \hat{N}_r(p)} \frac{\tilde{d}}{\|n - p\|_3} \alpha(p, n), \tag{C.1}$$

où $\|\cdot,\cdot\|_3$ est la distance euclidienne dans $\mathbb{R}^3$, $\alpha(p,n) \in [0,\pi/2]$ est l'angle entre les deux axes normaux associés à p et n, et $\tilde{d}$ est la distance médiane observée entre deux points voisins de la totalité du nuage de points. Nous avons introduit ici une pondération dans le calcul de la moyenne pour réduire l'impact des voisins en fonction de leur éloignement à p, tout en tenant compte de l'éloignement global observé sur la totalité du nuage de points. Ainsi, les points très proches – relativement à la distance médiane observée sur tout le maillage – seront plus influents que les points plus éloignés.

C.2.4 Distortion globale comme estimateur de forme

La *distortion globale de la surface* est alors estimée en calculant la moyenne des distortions locales de surface :

$$s(\mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} s(p). \tag{C.2}$$

Comme nous l'illustrerons dans les résultats, la valeur calculée par notre estimateur de distortion globale permet de détecter les différentes distortions des objets manipulés, dès lors qu'elles sont répercutées sur les normales. Ainsi, un ensemble de points distribués sur un plan seront estimées comme non distordues. Si l'on gonfle progressivement cette surface, on observera localement une modification des angles formés par les normales, qui entraîneront naturellement une modification cohérente de $s(\mathcal{P})$. La forme qui sera mesurée comme la moins distordue par notre estimateur de distortion sera celle avec le moins de perturbations locales, avec la forme la plus lisse.

C.3 Résultats et perspectives

Nous avons appliqué notre méthode de calcul de normales à un grand nombre de nuages de points, générés d'une part de manière virtuelle à l'aide d'un modeleur 3D, et d'autre part issues d'acquisitions 2D puis de reconstructions rigides.

C.3.1 Résultats

La figure C.6 présente le résultat du calcul des normales appliqué à différents nuages de points, issus de modeleurs ou d'acquisitions réelles. Les sphères écrasées représentent les différents points, et sont teintées en fonction de la distortion locale, sur une échelle propre à chaque nuage de points : plus la sphère est rouge, plus la distortion locale est importante. Les sphères sont écrasées dans le sens de la normale, afin d'en visualiser l'orientation. Le tableau C.1 reproduit les valeurs de distortion calculées sur chacun de ces ensembles de points, les maximums correspondant aux sphères rouges de la figure C.6, et les minimums aux sphères blanches.

Afin d'observer la pertinence de notre mesure de distortion, nous avons calculé sa valeur pour un ensemble de nuages de points issus de la déformation d'un même morceau de plan. La figure C.7 présente quelques résultats issus de ces déformations. Bien que nous n'ayons pas reproduit les images correspondantes, nous avons également vérifié sur cet exemple que le changement d'échelle et l'étirement ne modifiaient pas la valeur de notre estimateur.

Ces résultats confirment donc la capacité de notre outil de mesure à désigner la surface la moins distordue.

FIG. C.6 – *Estimation des normales et des distortions locales sur plusieurs nuages de points.*

Objet $\mathcal{P}$	Minimum $\{s(p)\}_{p\in\mathcal{P}}$	Maximum $\{s(p)\}_{p\in\mathcal{P}}$	$s(\mathcal{P})$
Coussin	0,12	0,53	0,31
Humain	0,20	2,57	0,77
Visage	0,11	0,76	0,42
Bâton	0,75	1,18	0,91
Tore	0,17	0,43	0,26
Flamme	0,49	0,87	0,61

TAB. C.1 – *Distortion globale et locale sur plusieurs nuages de points*

(a) Surface originale. $s(\mathcal{P}_1) = 0$.

(b) Gonflement de la surface. $s(\mathcal{P}_2) = 0,022$, $s(\mathcal{P}_3) = 0,071$, $s(\mathcal{P}_4) = 0,114$.

(c) Vrillement de la surface. $s(\mathcal{P}_5) = 0,016$, $s(\mathcal{P}_6) = 0,032$, $s(\mathcal{P}_7) = 0,066$.

(d) Pliure de la surface. $s(\mathcal{P}_8) = 0,018$, $s(\mathcal{P}_9) = 0,050$.

FIG. C.7 – *Distortion globale et normales calculées sur un ensemble de nuages de points issus de distortions depuis un même ensemble.* $c = 0.3$

Nombre de points restants	Erreur angulaire moyenne sur les normales
86	0,026
81	0,039
72	0,057
64	0,082
54	0,104
45	0,120

TAB. C.2 – *Erreur angulaire moyenne des normales sur* 50 *élagages à partir d'un même nuage de points.*

Enfin, nous avons étudié la robustesse de la méthode de calcul de normales à la suppression de points. Il s'agit en effet d'une situation classique dans le contexte applicatif qui nous intéresse, où il est fréquent que tous les points n'aient pu être localisés dans une frame donnée. Pour cela, nous avons généré à partir d'un nuage de points (figure C.6 de 90 points, en haut à gauche) un ensemble de nuages de points élagués. Pour différents ratios d'élagage $r \in [0, 1]$, nous avons généré 50 nuages de points en supprimant aléatoirement le nombre de points correspondant. Nous avons alors appliqué à chacun de ces nuages de points notre estimation de normales, et les avons comparé avec les normales correspondantes du nuage de point original. Le tableau C.2 reproduit les erreurs moyennes observées pour quelques ratios d'élagage, et permet d'observer la bonne stabilité du calcul des normales à la suppression de points.

C.3.2 Application à la capture de mouvements

Dans leur approche de reconstruction 3D non rigide, Fayad *et al.* [41] utilisent une séquence d'images 2D, desquelles sont extraites les coordonnées 2D de points à la surface d'un objet à reconstruire. Ils utilisent ensuite une technique d'optimisation pour ajuster une forme 3D non rigide à cette séquence de points 2D.

L'initialisation de cette méthode d'optimisation est un point critique de la méthode, la vélocité et la qualité de la reconstruction en dépendant. Une manière classique de calculer une forme initiale est de procéder à une reconstruction rigide [105] à partir de quelques frames de la séquence 2D où l'objet est connu comme étant *a priori* peu déformé.

Nous nous proposons ici d'estimer, à partir de la séquence 2D complète, une bonne approximation de la forme 3D peu déformée. En supposant que l'objet ne subit pas de déformations brutales au fil du temps, on commence par découper la séquence en plages de N frames (typiquement $N = 10$), et on procède à une reconstruction rigide de cette surface [105]. On dispose ainsi d'une séquence de nuages de points 3D. En appliquant à chaque nuage de points notre estimateur de distortion globale de surface, on estime la surface la moins distordue, qui servira d'initialisation au processus de reconstruction 3D non rigide global.

Les résultats associés à cette application étant encore en cours de génération, nous ne pouvons les intégrer à ce document au moment de sa rédaction.

C.3.3 Perspectives

Bien que ces premiers résultats semblent intéressants, certains points peuvent facilement être affinés, pour améliorer la vélocité des calculs, ainsi que la qualité des normales obtenues et de l'estimateur de distortion locale.

Une approche évidente à l'amélioration de la complexité de l'algorithme d'estimation de normales est d'utiliser une gestion des points du nuage un peu plus fine, par exemple en subdivisant l'espace de manière hiérarchique, à l'aide d'un octree, et en utilisant cette structure lors de la recherche du rayon seuil pour chacun des points de la sphère.

D'autre part, on sait que l'exactitude des normales obtenues est en grande partie dépendante dans notre approche du raffinement choisi pour la description des sphères. Une solution envisageable pour affiner ces calculs serait d'ajouter le long des arcs de cercle arêtes du polyhèdre sphérique des points qui seraient aussi utilisés dans le calcul par ACP.

De manière plus générale, une distortion de l'ensemble des points dans une direction orthogonale à la normale sera invisible par l'estimateur de forme local et global que nous avons décrit ici. Pour répondre en partie à cette limitation, une solution serait d'étudier la distortion des angles formés par les arêtes bord de la surface. Cependant, cette propriété est fortement dépendante de ce qui serait considéré comme un angle de bord non distordu.

Enfin, nous avons ici travaillé dans un contexte où les nuages étaient peu denses, où notre approche est suffisante pour obtenir un descripteur de forme scalaire. Cependant, dans le cas de nuages de points plus denses, l'information locale peut devenir insuffisante pour estimer la distortion globale. Plusieurs approches sont alors envisageables, depuis l'utilisation de voisinages plus grands pour calculer la distortion locale, jusqu'à l'utilisation d'outils plus complets comme les ondelettes, en étudiant la distance du nuage de points à une forme issue des basses fréquences.

Annexe D
Algorithmique illustrée

Cette annexe regroupe quelques planches de bande dessinée illustrant chacune un concept ou algorithme abordé dans cette thèse.

DÉCOUPAGE

PAVAGE

DUALITÉ DÉCOUPAGE/PAVAGE

DÉPLIAGE

OPTIMISATION DE TRICETS

CUT LOCUS ÉLAGUÉ

Bibliographie

[1] P. Alliez : Quadrangle surface tiling through contouring. *In* R. R. Martin, M. A. Sabin et J. R. Winkler, éds : *IMA Conference on the Mathematics of Surfaces*, vol. 4647 de *Lecture Notes in Computer Science*, p. 29–41, 2007.

[2] P. Alliez, D. Cohen Steiner, O. Devillers, B. Lévy et M. Desbrun : Anisotropic Polygonal Remeshing. *ACM Transactions on Graphics*, 22:485–493, 2003.

[3] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, A. Tal et I. IMATI-CNR : Mesh segmentation - a comparative study. *In IEEE International Conference on Shape Modeling and Applications, 2006. SMI 2006*, p. 7–7, 2006.

[4] V. Barra et J.-Y. Boire : Tissue segmentation on MR images of the brain by possibilistic clustering on a 3D wavelet representation. *Journal of Magnetic Resonance Imaging.*, 11(3):267–78, 2000.

[5] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott et T. Sederberg : Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2009.

[6] J. C. Bezdek : *Pattern Recognition with Fuzzy Objective Function Algorithms (Advanced Applications in Pattern Recognition)*. Springer, juil. 1981.

[7] G.-P. Bonneau, G. Elber, S. Hahmann et B. Sauvage : Multiresolution analysis. *In* L. De Floriani et M. Spagnuolo, éds : *Shape Analysis and Structuring*, Mathematics+Visualization, chap. 3, p. 83–114. Springer, jan. 2008.

[8] D. Carroll, C. Joint, N. Maartens, D. Shlugman, J. Stein et T. Z. Aziz : Motor cortex stimulation for chronic neuropathic pain : a preliminary study of 10 cases. *Pain*, 84(2-3):431–7, 2000.

[9] E. Catmull et J. Clark : Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, (10):350–355, 1978.

[10] P. Chainais : Infinitely divisible cascades to model the statistics of natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2105–2119, 2007.

[11] E. W. Chambers, J. Erickson et A. Nayyeri : Homology flows, cohomology cuts. *In Proceedings of the 41st ACM Symposium on Theory of Computing (STOC 2009)*, 2009.

[12] F. Chazal, D. Cohen-Steiner et A. Lieutier : Normal cone approximation and offset shape isotopy. *Computational Geometry : Theory and Applications*, 42(6-7):566–581, 2009.

[13] F. Chazal, D. Cohen Steiner, A. Lieutier et B. Thibert : Stability of Curvature Measures. Rapport de recherche RR-6756, INRIA, 2008.

[14] F. Chazal, D. Cohen-Steiner, A. Lieutier et B. Thibert : Stability of curvature measures. *CoRR*, abs/0812.1390, 2008.

[15] G. Choquet : *Cours de topologie.* Masson, Paris, 2e édn, 1972.

[16] D. Cohen-Steiner, P. Alliez et M. Desbrun : Variational shape approximation. *ACM Transactions on Graphics. Special issue for SIGGRAPH conference*, p. 905–914, 2004.

[17] É. Colin de Verdière : *Raccourcissement de courbes et décomposition de surfaces.* Thèse de doctorat, Université Paris 7, déc. 2003.

[18] É. Colin de Verdière : Algorithms for graphs on surfaces, 2008. Cours 2-14-2 du Master Parisien de Recherche en Informatique (MPRI).

[19] É. Colin de Verdière et F. Lazarus : Optimal system of loops on an orientable surface. *Discrete & Computational Geometry*, 33(3):507–534, mars 2005.

[20] C. R. Collins et K. Stephenson : A circle packing algorithm. *Computational Geometry : Theory and Applications*, 25:233–256, 2003.

[21] F. Cordier et H. Seo : Free-form sketching of self-occluding objects. *IEEE Computer Graphics and Applications*, 27(1):50–59, 2007.

[22] N. Cornea, D. Silver et P. Min : Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13(3): 530–548, 2007.

[23] B. Curless et M. Levoy : A volumetric method for building complex models from range images. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96)*, p. 303–312, New York, NY, USA, 1996. ACM Press.

[24] G. Damiand et S. Alayrangues : Computing canonical polygonal schemata with generalized maps. *In Proceedings of the International Conference on Topological & Geometric Graph Theory*, vol. 31 de *Electronic Notes in Discrete Mathematics*, p. 287–292, Paris, France, août 2008.

[25] G. Damiand, S. Peltier et L. Fuchs : Computing Homology Generators for Volumes Using Minimal Generalized Maps. *Lecture Notes in Computer Science*, 4958:63, 2008.

[26] J. Dargie : Modeling techniques : movies vs. games. *SIGGRAPH Computer Graphics*, 41(2):2, 2007.

[27] S. Deguy : *Le Modèle TdMAP : Un Générateur de Milieux Complexes Partiellement Identifiable.* Thèse de doctorat, Laboratoire de Logique Algorithmique et Informatique - Clermont 1, 2001.

[28] T. Dey, G. Li et J. Sun : Normal estimation for point clouds : A comparison study for a Voronoi based method. *In Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, p. 39–46, 2005.

[29] E. W. Dijkstra : A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[30] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci et J. C. Hart : Spectral surface quadrangulation. *ACM Transactions on Graphics*, 25(3):1057–1066, 2006.

[31] S. Dong, S. Kircher et M. Garland : Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer aided geometric design*, 22(5):392–423, 2005.

[32] A. Dupas et G. Damiand : First Results for 3D Image Segmentation with Topological Map. *Lecture Notes in Computer Science*, 4992:507, 2008.

[33] M. Eck et H. Hoppe : Automatic reconstruction of B-spline surfaces of arbitrary topological type. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, p. 325–334. ACM New York, NY, USA, 1996.

[34] J. Erickson et S. Har-Peled : Optimally cutting a surface into a disk. *In Proceedings of the eighteenth annual symposium on Computational geometry (SCG '02)*, p. 244–253, New York, NY, USA, 2002.

[35] J. Erickson et K. Whittlesey : Greedy optimal homotopy and homology generators. *In Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '05)*, p. 1038–1046, Philadelphia, PA, USA, 2005.

[36] A. Ern et J. Guermond : *Theory and practice of finite elements.* Springer Verlag, 2004.

[37] J.-M. Favreau et V. Barra : Conformal flattening of the cortical surface using a topological and geometrical cutting. *In Proceedings of the 2nd South-East European Conference on Computational Mechanics '09 (SEECCM'09)*, Rhodes, Greece, juin 2009.

[38] J.-M. Favreau et V. Barra : Cutting an organic surface. *In Proceedings of the 25th European Workshop on Computational Geometry (EuroCG'09)*, Brussels, Belgium, mars 2009.

[39] J.-M. Favreau et V. Barra : Low-resolution surface mapping : a topological and geometrical approach. *In Proceedings of the 25th European Workshop on Computational Geometry (EuroCG'09)*, Brussels, Belgium, mars 2009.

[40] J.-M. Favreau, S. Hemm, C. Nuti, J. Coste, V. Barra et J.-J. Lemaire : A tool for topographic analysis of electrode contacts in human cortical stimulation. *In Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, 2007*, Rio de Janeiro, Brazil, oct. 2007. ICCV 2007.

[41] J. K. Fayad, A. Del Bue, L. Agapito et P. M. Q. Aguiar : Principal moments for efficient representation of 2d shape. *IEEE International Conference on Image Processing (ICIP'09)*, 2009.

[42] C. Fennema-Notestine, I. B. Ozyurt, C. P. Clark, S. Morris, B.-G. Amanda, M. W. Bondi, T. L. Jernigan, B. Fischl, F. Segonne, D. W. Shattuck, R. M. Leahy, D. E. Rex, A. W. Toga, K. H. Zou, Morphometry BIRN et G. G. Brown : Quantitative evaluation of automated skull-stripping methods applied to contemporary and legacy images : Effects of diagnosis, bias correction, and slice location. *Human Brain Mapping*, 27(2):99–113, fév. 2006.

[43] M. S. Floater et K. Hormann : Surface parameterization : a tutorial and survey. *In* N. A. Dodgson, M. S. Floater et M. A. Sabin, éds : *Advances in multiresolution for geometric modelling*, p. 157–186. Springer Verlag, 2005.

[44] M. R. Garey et D. S. Johnson : The rectilinear steiner tree problem is np-complete. *SIAM J. Appl. Math.*, 32:826–834, 1977.

[45] A. Gramain : *Topologie des surfaces.* Presses Universitaires de France, Paris, 1971.

[46] F. Gris, J.-M. Favreau, O. Acosta, V. Barra et O. Salvado : A combined voxel and surface based method for topology correction of brain surfaces. 2010. Submitted to SPIE Medical Imaging.

[47] X. Gu, S. J. Gortler et H. Hoppe : Geometry images. *ACM Transactions on Graphics*, 21(3):355–361, 2002.

[48] B. Hajek : Cooling schedules for optimal annealing. *Mathematics of Operations Research*, p. 311–329, 1988.

[49] A. Hatcher : Pants decompositions of surfaces, 2000.

[50] A. Hatcher : *Algebraic topology.* Cambridge University Press, Cambridge, 2002.

[51] F. Hétroy et D. Attali : Topological quadrangulations of closed triangulated surfaces using the Reeb graph. *Graphical Models*, 65(1-3):131–148, 2003.

[52] W. Hong, X. Gu, F. Qiu, M. Jin et A. E. Kaufman : Conformal virtual colon flattening. *In Symposium on Solid and Physical Modeling*, p. 85–93, 2006.

[53] J. Huang, M. Zhang, J. Ma, X. Liu, L. Kobbelt et H. Bao : Spectral quadrangulation with orientation and alignment control. *ACM Transactions on Graphics*, 27(5):1–9, 2008.

[54] M. K. Hurdal, P. L. Bowers, K. Stephenson, D. W. L. Sumners, K. Rehm, K. Schaper et D. A. Rottenberg : Quasi-conformally flat mapping the human cerebellum. *In Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI '99)*, p. 279–286, 1999.

[55] T. Igarashi et J. Hughes : A suggestive interface for 3D drawing. *In International Conference on Computer Graphics and Interactive Techniques.* ACM New York, NY, USA, 2007.

[56] A. Jacques : Constellations et graphes topologiques. *Combinatorial Theory and Applications*, 2:657–673, 1970.

[57] S. Jaume : *Topology Simplification Algorithm for the Segmentation of Medical Scans.* Thèse de doctorat, Université Catholique de Louvain, 2004.

[58] S. Jaume, P. Rondao et B. Macq : Open Topology : A Toolkit for Brain Isosurface Correction. *In MICCAI Open Source Workshop*, 2005.

[59] M. Jin, Y. Wang, S.-T. Yau et X. Gu : Optimal global conformal surface parameterization. *In Proceedings of the conference on Visualization '04 (VIS '04)*, p. 267–274, Washington, DC, USA, 2004.

[60] R. M. Karp : Reducibility among combinatorial problems. *In* R. E. Miller et J. W. Thatcher, éds : *Complexity of Computer Computations*, p. 85–103. Plenum Press, 1972.

[61] S. Kirkpatrick : Optimization by simulated annealing : Quantitative studies. *Journal of Statistical Physics*, 34(5):975–986, 1984.

[62] P. N. Klein : Multiple-source shortest paths in planar graphs. *In Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '05)*, p. 146–155, Philadelphia, PA, USA, 2005.

[63] R. Krishnapuram et J. M. Keller : A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110, 1993.

[64] P. Laarhoven et E. Aarts : *Simulated annealing : theory and applications.* Springer, 1987.

[65] J.-O. Lachaud : Topologically defined iso-surfaces. *In Proceedings of the 6th International Workshop on Discrete Geometry for Computer Imagery (DGCI '96)*, p. 245–256, London, UK, 1996.

[66] Y. Lai, S. Hu et R. Martin : Surface mosaics. *The Visual Computer*, 22(9):604–611, 2006.

[67] Y. Lai, L. Kobbelt et S. Hu : An incremental approach to feature aligned quad dominant remeshing. *In Proceedings of the 2008 ACM symposium on Solid and physical modeling*, p. 137–145. ACM New York, NY, USA, 2008.

[68] F. Lazarus, M. Pocchiola, G. Vegter et A. Verroust : Computing a canonical polygonal schema of an orientable triangulated surface. *In Proceedings of the 17th Annual ACM Symposium on Computational Geometry*, p. 80–89, 2001.

[69] S. Lefebvre et C. Dachsbacher : Tiletrees. *In Proceedings of the 2007 symposium on Interactive 3D graphics and games (I3D '07)*, p. 25–31, New York, NY, USA, 2007.

[70] X. Li, T. Woon, T. Tan et Z. Huang : Decomposing polygon meshes for interactive applications. *In Proceedings of the 2001 symposium on Interactive 3D graphics*, p. 35–42. ACM New York, NY, USA, 2001.

[71] P. Lienhardt : Subdivisions of n-dimensional spaces and n-dimensional generalized maps. *In Proceedings of the fifth annual symposium on Computational geometry (SCG '89)*, p. 228–236, New York, NY, USA, 1989.

[72] W. E. Lorensen et H. E. Cline : Marching cubes : A high resolution 3d surface construction algorithm. *In Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH '87)*, p. 163–169, New York, NY, USA, 1987.

[73] M. Lounsbery, T. D. DeRose et J. Warren : Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, 1997.

[74] D. Lu, H. Zhao, M. Jiang, S. Zhou et T. Zhou : A surface reconstruction method for highly noisy point clouds. *In* N. Paragios, O. D. Faugeras, T. Chan et C. Schnörr, éds : *VLSM*, vol. 3752 de *Lecture Notes in Computer Science*, p. 283–294. Springer, 2005.

[75] W. S. Massey : *A Basic Course in Algebraic Topology.* Num. 56 de Graduate Texts in Mathematics. Springer, 1991.

[76] M. Matsumoto et T. Nishimura : Mersenne twister : A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, jan. 1998.

[77] J. S. B. Mitchell, D. M. Mount et C. H. Papadimitriou : The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.

[78] D. Mitra, F. Romeo et A. Sangiovanni-Vincentelli : Convergence and finite-time behavior of simulated annealing. *In 24th IEEE Conference on Decision and Control*, vol. 24, 1985.

[79] M. Mortara, G. Patane, M. Spagnuolo, B. Falcidieno et J. Rossignac : Plumber : a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies. *In Proceedings of the ninth ACM symposium on Solid modeling and applications*, p. 339–344. Eurographics Association Aire-la-Ville, Switzerland, Switzerland, 2004.

[80] M. Mortara, G. Patanè et M. Spagnuolo : From geometric to semantic human body models. *Computers and Graphics*, 30(2):185–196, 2006.

[81] J. R. Munkres : *Elements of Algebraic Topology.* Perseus Books, Cambridge, 1984.

[82] T. S. Newman et H. Yi : A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854–879, oct. 2006.

[83] J. P. Nguyen, J. P. Lefaucher, C. Le Guerinel, J. F. Eizenbaum, N. Nakano, A. Carpentier, P. Brugieres, B. Pollin, S. Rostaing et Y. Keravel : Motor cortex stimulation in the treatment of central and neuropathic pain. *Arch Med Res*, 31(3):263–5, 2000.

[84] J. P. Nguyen, J. P. Lefaucheur, P. Decq, T. Uchiyama, A. Carpentier, D. Fontaine, P. Brugieres, B. Pollin, A. Feve, S. Rostaing, P. Cesaro et Y. Keravel : Chronic motor cortex stimulation in the treatment of central and neuropathic pain. Correlations between clinical, electrophysiological and anatomical data. *Pain*, 82(3):245–51, 1999.

[85] C. Nuti, R. Peyron, L. Garcia-Larrea, J. Brunon, B. Laurent, M. Sindou et P. Mertens : Motor cortex stimulation for refractory neuropathic pain : four year outcome and predictors of efficacy. *Pain*, 118(1-2):43–52, 2005.

[86] D. OuYang et H. Feng : On the normal vector estimation for point cloud data from smooth surfaces. *Computer-Aided Design*, 37(10):1071–1079, 2005.

[87] S. Peltier, S. Alayrangues, L. Fuchs et J.-O. Lachaud : Computation of homology groups and generators.. *Computer & Graphics*, 30(1):62–69, fév. 2006.

[88] G. Peyre et L. D. Cohen : Landmark-based geodesic computation for heuristically driven path planning. *In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, p. 2229–2236, Washington, DC, USA, 2006.

[89] W. Puech, A. Bors, I. Pitas et J.-M. Chassery : Projection distortion analysis for flattened image mosaicing from straight uniform generalized cylinders. *Pattern Recognition*, 34(8):1657–1670, 2001.

[90] N. Ray, W. Li, B. Lévy, A. Sheffer et P. Alliez : Periodic global parameterization. *ACM Transactions on Graphics (TOG)*, 25(4):1460–1485, 2006.

[91] C. Roudet, F. Dupont et A. Baskurt : Multiresolution mesh segmentation based on surface roughness and wavelet analysis. *In* J. L. Chang Wen Chen, Dan Schonfeld, éd. : *Proceedings of SPIE – Volume 6508 Visual Communications and Image Processing 2007 (VCIP'07)*, jan. 2007.

[92] D. W. Shattuck, S. R. Sandor-Leahy, K. A. Schaper, D. A. Rottenberg et R. M. Leahy : Magnetic resonance image tissue classification using a partial volume model. *NeuroImage*, 13(5):856–876, 2001.

[93] A. Sheffer et J. C. Hart : Seamster : inconspicuous low-distortion texture seam layout. *In Proceedings of the conference on Visualization '02 (VIS '02)*, p. 291–298, Washington, DC, USA, 2002.

[94] A. Sheffer, B. Lévy, M. Mogilnitsky et A. Bogomyakov : Abf++ : Fast and robust angle based flattening. *ACM Transactions on Graphics*, (2):311–330, avr. 2005.

[95] S. Shlafman, A. Tal et S. Katz : Metamorphosis of polyhedral surfaces using decomposition. *In Computer Graphics Forum*, vol. 21, p. 219–228. Citeseer, 2002.

[96] B. Smith : Image reconstruction from cone-beam projections : necessary and sufficient conditions and reconstruction methods. *IEEE transactions on medical imaging*, 4(1):14–25, 1985.

[97] D. Steiner et A. Fischer : Planar parameterization for closed 2-manifold genus-1 meshes. *In Proceedings of the ninth ACM symposium on Solid modeling and applications (SM '04)*, p. 83–91, Aire-la-Ville, Switzerland, Switzerland, 2004.

[98] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler et H. Hoppe : Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.*, 24(3):553–560, juil. 2005.

[99] J. Talairach : *Atlas d'anatomie stéréotaxique. Repérage radiologique indirect des noyaux gris centraux des régions mésencéphalo-sous-optiques et hypothalamiques de l'homme.* Paris : Masson & Cie, 1957.

[100] C. Tang et G. Medioni : Curvature-augmented tensor voting for shape inference from noisy 3d data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 858–864, 2002.

[101] J. Tangelder et R. Veltkamp : A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2008.

[102] C. Thomassen : Embeddings of graphs with no short noncontractible cycles. *Journal of Combinatorial Theory Series B*, 48(2):155–177, 1990.

[103] J. Tierny, J.-P. Vandeborre et M. Daoudi : Invariant High-Level Reeb Graphs of 3D Polygonal Meshes. *In Proceedings of the 3rd IEEE International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'06)*, p. 105–112, Chapel Hill, North Carolina, USA, 2006.

[104] J. Tierny, J.-P. Vandeborre et M. Daoudi : Enhancing 3d mesh topological skeletons with discrete contour constrictions. *The Visual Computer*, 24:155–172, 2008.

[105] C. Tomasi et T. Kanade : Shape and motion from image streams under orthography : a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[106] B. Vallet et B. Lévy : Spectral geometry processing with manifold harmonics. *Computer Graphics Forum (Proceedings Eurographics)*, 2008.

[107] D. Van Essen, H. A. Drury, S. Joshi et M. I. Miller : Functional and structural mapping of human cerebral cortex : solutions are in the surfaces. *Advances in neurology*, 95(3):788–795, 1998.

[108] J. Van Wijk : Bicubic patches for approximating non-rectangular control-point meshes. *Computer Aided Geometric Design*, 3(1):1–13, 1986.

[109] J. Warnking, M. Dojat, A. Guérin-Dugué, C. Delon-Martin, S. Olympieff, N. Richard, A. Chéhikian et C. Segebarth : fMRI retinotopic mapping – step by step. *NeuroImage*, 17(4):1665–1685, déc. 2002.

[110] W. Wells, P. Viola, H. Atsumi, S. Nakajima et R. Kikinis : Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51, 1996.

[111] Z. Wood et I. Guskov : Topological noise removal. *Proceedings of Graphics Interface*, p. 19–26, 2001.

[112] R. Zayer, B. Lévy et H.-P. Seidel : Linear angle based parameterization. *In Proceedings of the ACM/EG Symposium on Geometry Processing conference*, 2007.

[113] Q.-Y. Zhou, T. Ju et S.-M. Hu : Topology repair of solid models using skeletons. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):675–685, 2007.

[114] M. Zöckler, D. Stalling et H. Hege : Fast and intuitive generation of geometric shape transitions. *The Visual Computer*, 16(5):241–253, 2000.

Index

A

algorithme
- circle packing **137**
- de Dijkstra **27**

arête
- d'un simplexe **18**
- d'une m-cellule **30**

B

bord
- d'un complexe simplicial **21**
- d'un simplexe **19**
- d'une variété, (le) **6**
- d'une variété, (un) **6**, 7, 24
- opérateur de **21**

brin **29**

C

caractéristique d'Euler **24**
- généralisée **36**

carte combinatoire **29**

cellule **18**, 19
- ouverte **18**

chaîne **20**
- élémentaire **20**, 22, 26
- groupe des **20**

chaîne
- simple **20**

chemin **8**, 20, 25
- composé **8**
- constant **8**
- extrémité **8**
- homotope **8**
- inverse **9**
- origine **8**
- reliant **8**, 25

complexe simplicial **19**

cut locus **49**
- élagué **49**

CW-complexe **18**, 30

cycle **21**

D

découpage **33**, 51
- élémentaire **33**
- en disque **47**
- minimal **50**

dépliage **61**

dimension
- d'un simplexe **18**

distance géodésique **25**
- discrète **26**

distorsion
- d'un sommet **62**, **97**
- d'une arête **62**

E

espace des paramètres **61**

espace topologique **6**
- connexe par arcs **8**
- séparé **6**

états communiquants **78**

extremum **65**

F

face
- d'un simplexe **19**
- d'une m-cellule **30**

facteur d'étirement **62**

faible réversibilité **79**

G

genre **10**, 24

géodésique **25**
- discrète **26**

graphe de Reeb 63

groupe
- d'homologie **22**
- d'homotopie **9**

fondamental **9**
générateur **9**

H

homéomorphe **6**, 10
homéomorphisme **6**

I

intérieur
d'une n-variété **6**
intérieur
d'un simplexe **19**
d'une m-cellule **30**
IRM **107**
IRMf **107**

L

lacet **8**
homotope **9**, 10
non séparant 50
séparant **8**, 9
simple **8**
trivial **9**
lieu de découpage **49**
longueur
d'un chemin **25**
d'un découpage **35**, 49, 50

M

m-complexe cellulaire **30**
M-pavage **31**
élémentaire **33**
élémentaire associé **32**
équivalence topologique **32**
M-tuile **31**
maillage 31
maillage **20**

N

n-cet **12**
homotope **13**
simple **13**
valide **13**
nombre de Betti **23**

O

ordre **20**
orientation
d'un simplexe **20**
d'une n-variété **7**

P

paramétrisation
conforme **62**
globale **61**
poignée **10**
point multiple **82**
point-base 8
propriété des trois chemins **50**

Q

$\mathcal{Q}$-critère **86**
quadrangle **84**
quadrangulation **84**

S

schéma polygonal **11**
canonique **12**
réduit **12**
simplexe **18**
sommet
d'un complexe simplicial **18**
d'un simplexe **19**
d'une m-cellule **30**
sous-complexe simplicial 19
surface **7**
connexe par arcs **8**, 8
discrète **20**
à bord **7**

T

TEP **107**
triangle **18**
tricet **12**
tuile **29**

V

variété **6**
à bord **6**
orientable **7**, 23
voisinage **6**
voisinage topologique **53**
voxel **108**

Liste des figures

1.1 Le ruban de Möbius, surface non orientable. 7
1.2 Deux générateurs du groupe fondamental de la surface tore (l_1 et l_2). . . . 9
1.3 Surfaces sans bord. 10
1.4 Calcul du genre d'un cylindre ouvert C composé de deux bords ∂_1 et ∂_2 par recollements successifs de disques. Le genre du cylindre est donc le même que celui de la boule, c'est-à-dire de zéro. 11
1.5 Schémas polygonaux associés à deux surfaces classiques. On construit les surfaces en recollant les arêtes notées a, en respectant l'orientation indiquée. 11
1.6 Deux schémas polygonaux distincts associés à la même surface. 11
1.7 Schéma polygonal de la sphère. 12
1.8 Tricet valide sur une surface de genre zéro à trois bords. 13
1.9 Illustration de la preuve du lemme 1.1 14
1.10 n-cets valides sur un tore. 15

2.1 Collections de simplexes. 19
2.2 1-chaîne sur un complexe simplicial. $\{c_0, c_1, c_2\}$ est une 1-chaîne composée de 3 chaînes élémentaires, ou encore de deux chaînes simples ($\{c_0\}$ et $\{c_1, c_2\}$). 21
2.3 Application de l'opérateur de bord à différentes chaînes. 21
2.4 1-cycles sur une surface K (complexe simplicial) et illustration de $H_1(K)$. . 22
2.5 Deux surfaces de même nombre de Betti mais non homéomorphes. 24
2.6 Les deux géodésiques reliant a et b positionnés de chaque côté d'un cylindre ouvert. 26
2.7 Surfaces portant des chemins de découpages. 28
2.8 2-m-cellules. 30
2.9 M-tuile sur une surface $\mathcal{M}$ (de genre deux sans bord) décrite par une 2-m-cellule de genre zéro à trois bords et par un plongement F. 31
2.10 M-pavage sur une surface $\mathcal{M}$ (de genre deux sans bord) décrite par un m-complexe cellulaire composé de 2-m-cellules, et par un plongement F. . . 32
2.11 Représentation d'un schéma polygonal par M-pavage. 32
2.12 M-pavage sur une surface $\mathcal{M}$ topologiquement équivalent à celui décrit par la figure 2.10. 33
2.13 Deux ensembles de 1-m-cellules non homéomorphes. 33

3.1 Illustration de la preuve du lemme 3.1. 43
3.2 Le maillage de gauche décrit une surface S homéomorphe à un disque. La surface de droite S' a été construite à partir de S de façon à ce que tout chemin de plus de deux arêtes croise le bord. 45
3.3 Voisinage topologique V_{l_s} d'un sommet s appartenant à l'ensemble des points base B sur la surface S. 53

3.4 Calcul de lacets approchés 54
3.5 Découpage d'une surface de genre 104, sans bord et composée de 543 652 triangles à l'aide de l'algorithme de recherche approximative du plus court lacet non séparant. Maillage 3D original fourni par le dépôt de formes laboratoire Computer Graphics de l'Université de Stanford. 56

4.1 Calcul et utilisation de la courbure locale dans la notion de longueur d'arête. 60
4.2 Variation géométrique du découpage suivant la distance utilisée. 61
4.3 Extremums sur une surface de genre zéro. 63
4.4 Découpage topologique d'une surface de genre zéro sans bord. 65
4.5 Découpage géométrique sur une surface de genre trois sans bord. 66
4.6 Schéma d'une α-jonction et de sa correction topologique 68
4.7 Corrections topologiques utilisant l'algorithme 4.1. Les surfaces ont été lissées après correction pour permettre un meilleur aperçu des corrections apportées. 69

5.1 Chemin produit après découpage des surfaces pincées. 72
5.2 Processus de raffinement de maillage pour le décollage des bords. 73
5.3 Décomposition d'un 6-cet sur une surface de genre 3 avec ajout de cylindres. 75
5.4 Calcul d'un 3-cet valide sur un pantalon. 77
5.5 Recherche de nouveaux points base à partir d'un 3-cet en utilisant la longueur des chemins existants. 80
5.6 Pavages en cylindre de surfaces par 3-cets et recuit simulé. 81
5.7 Variations des contraintes sur les points multiples lors du découpage topologique d'un double tore. 82
5.8 Découpage d'une tuile polygonale paire en respectant le $\mathcal{Q}$-critère. 86
5.9 Résultat des variations de contraintes sur les points multiples lors de la quadrangulation d'un double tore. 87
5.10 Gauche : quadrangulation après découpage topologique initial (algorithme 3.2) d'une surface de genre quatre. Droite : détail sur la tuile a à la géométrie médiocre. 88
5.11 Construction de deux quadrangles depuis un cylindre issu de découpage par 3-cets (section 5.2.3). 88
5.12 Quadrangulation après découpage par cylindres (section 5.2.3). 90

6.1 Découpage par la méthode PCA (section 4.2.2) en utilisant une distance euclidienne (gauche), et une distance définie par orientation (droite). Maillage 3D original fourni par le dépôt de formes AIM@SHAPE. 95
6.2 Découpage d'une surface homéomorphe à un disque en utilisant un seul extremum. (a) : distance euclidienne uniquement. (b) : carte de poids dessinée par un utilisateur sur la surface. (c) : distance modifiée par la carte de l'utilisateur. Maillage 3D original fourni par le dépôt de formes laboratoire Computer Graphics de l'Université de Stanford. 96
6.3 Détection d'extremums à différentes échelles. Les figures (a) et (b) illustrent les limitations de la méthode de [103]. 99
6.4 Illustration de la distorsion du dépliage en fonction du nombre d'extremums utilisés pour le découpage. Le tableau 6.1 présente quelques résultats quantitatifs sur ces découpages. Maillage 3D original fourni par le dépôt de formes laboratoire Computer Graphics de l'Université de Stanford, modifié par nos soins. 100

6.5 Découpage topologique et géométrique d'une surface de genre deux, sans bord, et à la géométrie non triviale. 101
6.6 Découpage géométrique d'une surface de genre zéro, sans bord, en faisant varier la distance utilisée. 102
6.7 Segmentation en cylindres d'une main. Maillage 3D original fourni par le dépôt de formes AIM@SHAPE. 104
6.8 Segmentation d'objets issus d'un modeleur 3D. 105
6.9 Segmentation en présence de bruit. 106

7.1 Schéma du fonctionnement complet de la chaîne de traitement dans le contexte de la stimulation corticale. 110
7.2 Résultat de la segmentation des tissus cérébraux d'un volume IRM après application d'une suppression de scalp et d'os. 111
7.3 Schéma d'une électrode utilisée lors de la stimulation corticale. 114
7.4 Coupe issue d'un volume 3D résultat d'un détourage manuel des aires anatomiques 114
7.5 Cartes produites par le dépliage avec et sans correction topologique des α-jonctions. 115
7.6 Visualisation d'informations en utilisant la projection issue d'une IRM anatomique. 116

A.1 Diagramme de classe simplifié décrivant la structure des points d'un maillage. 129
A.2 Mécanisme de création de distances par *prototype* et *fabrique*. 129
A.3 Diagramme de classe simplifié décrivant les méthodes de dépliage. 131
A.4 Texture procédurale générée sur une surface de genre deux. 132
A.5 Normales d'une surface décrite par un nuage de points. 132
A.6 Captures d'écran de l'interface à `Taglut` dans `blender`. 135

B.1 Ajustement local des cercles dans l'algorithme de circle packing [20]. En haut : l'angle initial est inférieur à 2π, le rayon du cercle central est diminué. En bas : l'angle initial est supérieur à 2π, le rayon du cercle central est augmenté. 138
B.2 La somme des angles des tangentes d'un polygone du plan est égale à 2π . 139
B.3 Application de l'algorithme circle packing (milieu) et circle packing avec contraintes aux bords (droite) sur deux maillages déjà plans (gauche). . . . 139

C.1 Estimation de l'axe normal par la méthode de l'offset. 145
C.2 (b)–(d) : subdivisions successives de l'icosaèdre original (a). 145
C.3 Rayons homogènes sur un nuage de points à la distribution non homogène. 146
C.4 Étape intermédiaire de l'estimation du rayon de l'offset associé à p. 147
C.5 Estimation de la propriété de bord par la méthode de l'offset. q est un point bord, alors que p n'en est pas un. 147
C.6 Estimation des normales et des distortions locales sur plusieurs nuages de points. 149
C.7 Distortion globale et normales calculées sur un ensemble de nuages de points issus de distortions depuis un même ensemble. $c = 0.3$ 149

Table des matières

Introduction **1**

I Contexte **3**

1 Variétés topologiques **5**
1.1 Espaces topologiques 6
1.2 Classification 7
1.2.1 Chemins et lacets 8
1.2.2 Groupe fondamental 8
1.2.3 Genre et bords 10
1.3 Schéma polygonal 11
1.4 n-cets 12

2 Variétés discrétisées **17**
2.1 Structures élémentaires 18
2.1.1 CW-complexes et complexes simpliciaux 18
2.1.2 Chaînes 20
2.1.3 Cycles et bords 21
2.2 Invariants topologiques 22
2.2.1 Groupes d'homologie 22
2.2.2 Nombres de Betti 23
2.2.3 Caractéristique d'Euler et calcul d'invariants 23
2.3 Mesures géométriques sur les surfaces 25
2.3.1 Approximation de distances géodésiques 25
2.3.2 Distances non euclidiennes 28
2.4 Pavage de surfaces 29
2.4.1 Notion de m-complexe cellulaire 30
2.4.2 Tuiles et pavages 31
2.4.3 Dualité pavage / découpage 33
2.4.4 Produit d'un découpage 35
2.5 Caractéristique d'Euler généralisée et conséquences 35
2.5.1 Généralisation de la caractéristique d'Euler 36
2.5.2 Conséquences immédiates 36
2.6 Conclusion 37

II Pavage sous contraintes **39**

3 Contraintes topologiques **41**
3.1 Découpages élémentaires . . . 42
3.1.1 Découpage suivant un 1-cycle . . . 42
3.1.2 Découpage suivant chemins et n-cets . . . 45
3.2 Découpage en disque . . . 47
3.2.1 Description du problème et motivations . . . 47
3.2.2 Algorithme linéaire sans contrainte . . . 48
3.2.3 Combinatoire et longueur de découpage . . . 49
3.3 Découpage minimal . . . 49
3.3.1 Définition et NP-difficulté . . . 50
3.3.2 Découpage par lacets non séparants . . . 50
3.3.3 Optimisations de l'algorithme . . . 52
3.3.4 Algorithme approché . . . 55
3.3.5 Discussion autour des variations . . . 55
3.4 Conclusion . . . 57

4 Contraintes géométriques **59**
4.1 Distance non euclidienne . . . 60
4.1.1 Courbure locale . . . 60
4.1.2 Distorsion d'une paramétrisation conforme . . . 61
4.2 Détection des extremums . . . 63
4.2.1 Travaux antérieurs . . . 63
4.2.2 Détection par transformation conforme . . . 64
4.2.3 De la géométrie à la topologie . . . 66
4.3 Patch des surfaces . . . 67
4.3.1 Travaux similaires . . . 67
4.3.2 α-jonctions . . . 67
4.3.3 Simplification par ajustement de la topologie . . . 68
4.4 Conclusion . . . 69

5 Contraintes sur les tuiles **71**
5.1 Considérations générales . . . 72
5.2 Découpage en cylindres . . . 73
5.2.1 Étude combinatoire . . . 73
5.2.2 Plus courts n-cets . . . 75
5.2.3 Pavage par n-cets . . . 76
5.3 Nombre de sommets d'une tuile unique . . . 82
5.3.1 Contrôle du nombre de 0-m-cellules . . . 82
5.3.2 Découpage en polygone minimal . . . 83
5.4 Découpage en quadrangles . . . 84
5.4.1 Travaux existants . . . 84
5.4.2 Simplification de tuile unique . . . 85
5.4.3 Simplification de pavage en cylindres . . . 88
5.5 Conclusion . . . 90

III Applications 91

6 Découpages pour l'infographie 93
6.1 Distances non euclidiennes . . . 94
6.1.1 Cartes de visibilité . . . 94
6.1.2 Cartes d'orientation principale . . . 95
6.1.3 Carte de contraintes utilisateur . . . 95
6.2 UV mapping de formes organiques . . . 96
6.2.1 Sélection des extremums . . . 97
6.2.2 Découpage organique et paramétrisation . . . 97
6.2.3 Résultats . . . 98
6.3 Segmentation organique . . . 102
6.3.1 Segmentation cylindrique . . . 103
6.3.2 Résultats . . . 104
6.4 Conclusions et perspectives . . . 105

7 Application médicale 107
7.1 Contexte général . . . 107
7.2 Cartographie du cortex . . . 108
7.2.1 Travaux antérieurs . . . 109
7.2.2 Chaîne de traitement générale . . . 109
7.3 Application à la stimulation corticale . . . 112
7.3.1 Contexte médical . . . 112
7.3.2 Construction des cartes . . . 113
7.3.3 Résultats et discussion . . . 114
7.4 Conclusion et perspectives . . . 117

Conclusion 119

Annexes 125

A Taglut 125
A.1 Langages et bibliothèques . . . 126
A.1.1 Solutions existantes . . . 126
A.1.2 Réalisation d'un outil original . . . 127
A.2 Structures de données et implémentation . . . 128
A.2.1 Maillage . . . 128
A.2.2 Calcul de distances . . . 129
A.2.3 Pavage . . . 130
A.2.4 Dépliage . . . 130
A.2.5 Flexibilité de l'outil . . . 131
A.3 Échanges et interactions . . . 132
A.3.1 Formats d'échange . . . 133
A.3.2 Exécutables spécifiques . . . 134
A.3.3 Adaptateur python . . . 134
A.3.4 Interfaçage avec blender . . . 135

B Variations autour du dépliage **137**
B.1 Circle packing : descriptif rapide 137
B.2 Contraintes aux bords 138
B.3 Parallélisation 140
B.4 Multi-résolution 140
B.5 Conclusion 141

C Géométrie des nuages de points **143**
C.1 Travaux similaires 144
C.2 Du nuage de points à l'estimation de distortion globale 144
C.2.1 Estimation des normales 144
C.2.2 Estimation des bords 147
C.2.3 Distortion locale 147
C.2.4 Distortion globale comme estimateur de forme 148
C.3 Résultats et perspectives 148
C.3.1 Résultats 148
C.3.2 Application à la capture de mouvements 150
C.3.3 Perspectives 150

D Algorithmique illustrée **153**

Bibliographie **161**

Index **169**

Liste des figures **171**

Table des matières **175**

Résumé

Alors que l'on observe une disponibilité croissante de données décrivant des objets 3D, il semble essentiel de disposer de moyens de traitement efficaces de ces derniers. Ainsi, nous présentons dans ce mémoire un ensemble d'outils de manipulation de surfaces, qui exploitent à la fois leurs propriétés géométriques et topologiques.
Après avoir décrit différents résultats classiques de topologie, et les structures et résultats fondamentaux de la topologie algorithmique, nous présentons les concepts de M-tuiles et M-pavages, offrant notamment une grande souplesse combinatoire, et permettant de décrire finement le résultat d'algorithmes de découpage topologique. En s'appuyant sur les possibilités de description de ce formalisme, nous présentons différents algorithmes de découpage de surface, prenant en compte non seulement la topologie et la géométrie des surfaces, mais également les propriétés des M-tuiles issues de ces découpages. Nous présentons également dans ce mémoire une généralisation des lacets par les n-cets, permettant notamment de décrire une approche originale de pavage de surfaces en cylindres puis en quadrangles.
Enfin, deux applications de ces outils de découpage sont présentées. Dans un premier temps, nous déclinons ces algorithmes de découpage dans le contexte de l'infographie, en proposant un ensemble d'outils d'aide à la manipulation de surfaces. Puis nous présentons dans un second temps une chaîne complète de traitement de données issues de l'imagerie médicale, permettant la visualisation dynamique de données complexes sur des cartes planes de la surface du cerveau, en illustrant sa pertinence dans le contexte de la stimulation corticale.
En conclusion de ces travaux, nous présentons les perspectives que laissent entrevoir ces développements originaux, notamment en exploitant les possibilités offertes par les n-cets et les M-pavages, qui semblent multiples. Nous soulignons également la richesse qu'assure une exploration des domaines applicatifs par des outils issus de la géométrie algorithmique.
Mots-clés : surfaces, graphes, découpage, pavage, topologie algorithmique, géométrie

Abstract

The amount of data available to describe the geometry of a 3D object is ever increasing, and therefore ways to efficiently process it are essential. In this dissertation, we present a collection of tools for surface manipulation that exploit both their geometric and topological properties.
First we describe some classical results of topology, the framework and main algorithms for surface cutting. We then introduce the M-tiles and M-tiling concepts, which notably offer great combinatorial flexibility while allowing to describe the results from topological cutting algorithms with precision. Thanks to the descriptive power of this formalism we introduce various cutting algorithms, which take into account not only the topology and geometry of the surfaces but also the properties of the M-tiles resulting from the cuttings. We also introduce in this dissertation a generalization of loops we call n-loops, which allow the description of a new approach for tiling surfaces with cylindrical patches first and then quadrangles.
Two applications of these cutting tools are finally presented. First, in the context of computer graphics, we describe a set of tools based on the new cutting algorithms to aid surface manipulation. Then, we describe a complete framework for medical imaging, that produces a dynamic visualization interface to display complex data on planar brain maps. The suitability of this visualization is illustrated in the context of cortical stimulation.
Finally, we discuss the future perspectives offered by our novel developments, *e.g.* by exploiting the potential of n-loops and M-tilings, which we believe to be numerous. We also underline the richness provided by exploiting computational geometry tools in the context of the aforementioned application fields.
Keywords: surfaces, graphs, cutting, tiling, computational topology, geometry