

Remaillage géodésique par propagation de fronts

Geodesic Remeshing Using Front Propagation

Gabriel Peyré¹

Laurent Cohen²

¹ CMAP, Ecole Polytechnique, 91128 Palaiseau cedex, UMR CNRS 7641

² CEREMADE, Université Paris Dauphine, UMR CNRS 7534

peyre@cmapx.polytechnique.fr, cohen@ceremade.dauphine.fr

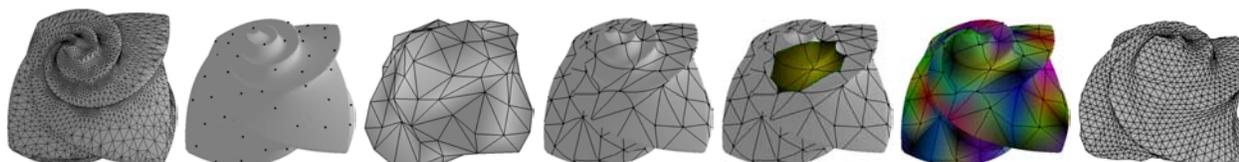


FIG. 1 – Les différentes étapes de la paramétrisation d'un maillage.

Résumé

Dans cet article, nous présentons une méthode pour remailler une variété triangulée. Notre travail utilise l'algorithme de Fast Marching de Sethian [16], qui a été étendu à des maillages arbitraires par Sethian et Kimmel dans [17]. Tout d'abord, un ensemble de points répartis de façon uniforme sur la surface est automatiquement choisi. Une triangulation de Delaunay géodésique de cet ensemble de points est ensuite créée, en utilisant la construction d'un diagramme de Voronoi calculé par Fast Marching. Nous utilisons les informations de distance pour trouver une paramétrisation simple de la variété. L'algorithme de Fast Marching permet d'obtenir un algorithme rapide, et donne de très bons résultats. Des exemples sont présentés à la fois pour des surfaces réelles et synthétiques. Pour conclure cet exposé, nous proposons une application de notre algorithme à l'aplatissement de surfaces. La méthode proposée inclue un schéma d'interpolation géodésique original.

Mots Clef

Remaillage, géodésiques sur des surfaces, algorithme de Fast Marching, paramétrisation de surfaces, reconstruction de surface, modèles déformables.

Abstract

In this paper, we present a method for remeshing triangulated manifolds by using geodesic path calculations and distance maps. Our work builds on Sethian's Fast Marching algorithm [16], which has been extended to arbitrary meshes by Sethian and Kimmel in [17]. First, a set

of points that are evenly spaced across the surface is automatically found. A geodesic Delaunay triangulation of the set of points is then created, using a Voronoi diagram construction based on Fast Marching. We use the distance information to find a simple parameterization of the manifold. Fast Marching algorithm makes this method computationally inexpensive, and gives very good results. Examples are shown for synthetic and real surfaces. To conclude this paper, we present an application of our algorithm to surface flattening. Our method includes a novel geodesic interpolation scheme.

Keywords

Remeshing, surface geodesics, Fast Marching algorithm, surface parameterization, surface reconstruction, deformable models.

1 Introduction

Les applications de l'échantillonnage de surface 3D sont très courantes de nos jours. Elles vont des calculs par éléments finis jusqu'à la génération d'images de synthèse, ce qui inclut une variété de questions relatives aux techniques de reconstruction de surfaces, comme les modèles déformables [6] utilisés en segmentation d'images 3D. La représentation la plus couramment employée pour les objets 3D est le maillage triangulaire, et la nécessité d'une triangulation de bonne qualité est évidente. Ainsi il est souvent nécessaire d'améliorer une structure 3D donnée (par exemple obtenue par imagerie médicale 3D ou par le modelage d'un artiste), en utilisant un algorithme de remaillage.

Le problème de la paramétrisation d'une variété est im-

portant puisqu’il est à la base d’une «bonne» représentation du maillage de départ. Par exemple, l’obtention d’une paramétrisation minimisant certains critères de distorsion est la première étape en vue d’effectuer du plaquage de texture ou du remaillage semi-régulier. De plus, une fois que l’on a obtenu une représentation semi-régulière de la surface, il est facile de calculer une transformée en ondelettes pour réaliser une compression de données [12].

1.1 Aperçu

Dans cet article, nous présentons au paragraphe 3 un algorithme de remaillage isotrope, qui calcule automatiquement un ensemble de points espacés sur la surface soit de façon uniforme, soit de façon adaptative en suivant une densité prescrite. Ces points sont définis de manière itérative, chaque nouveau point étant ajouté suivant une carte de distance (géodésique) à l’ensemble des points déjà calculés. Un algorithme rapide est présenté. A chaque étape, il met à jour la carte de distances en utilisant un Fast Marching. Le fonctionnement de l’algorithme de Fast Marching, ainsi que son utilisation pour les calculs de géodésiques sont rappelés au paragraphe 2. Enfin, au paragraphe 4, nous proposons un nouvel algorithme de paramétrisation qui utilise les informations géodésiques.

Pour parvenir à paramétrer une variété triangulée, nous commençons par calculer un maillage grossier en définissant un diagramme de Voronoi sur la surface. La paramétrisation est ensuite interpolée à l’intérieur de chaque triangle géodésique, en utilisant des données de distance appropriées. La figure 1 montre les différentes étapes qui composent le procédé de paramétrisation :

- Maillage original.
- Détermination automatique des points de base.
- Détermination de leur triangulation de Delaunay.
- Calcul des triangles géodésiques associés.
- Calcul des données de distance.
- Interpolation de la paramétrisation.
- Remaillage semi-régulier.

1.2 Travaux antérieurs

Le calcul de lignes géodésiques sur des variétés est un sujet classique, et de nombreux algorithmes ont été proposés. Parmi les plus intéressants, on peut citer :

- L’algorithme de plus court chemin de *Chen et Han* [4] est d’ordre quadratique, et reste difficile à implémenter.
- L’algorithme de *Fast Marching* de *Sethian et Kimmel* [17] permet de résoudre le problème avec des conditions aux limites, (points de départ et d’arrivée), qui n’a pas nécessairement de solution unique. L’algorithme est rapide, d’ordre $O(n \log(n))$, où n désigne le nombre de sommets de la triangulation.

Des applications des calculs de géodésiques ont été proposées, comme par exemple [13], qui applique l’algorithme de Fast Marching pour déterminer les *diagrammes de Voronoi* et des *offsets* de courbes sur des variétés. Cette idée

est utilisée dans notre algorithme pour déterminer la *triangulation de Delaunay* d’un ensemble de points.

Les méthodes de remaillage peuvent être classées grossièrement en deux catégories :

- *Remaillage isotrope* : une densité surfacique de points est définie, et l’algorithme essaie de positionner les nouveaux sommets de façon à se conformer à cette densité. L’algorithme de *Alliez et Al.* [2] utilise une diffusion d’erreur pour placer les sommets. L’algorithme de *Terzopoulos et Vasilescu* [20] utilise des modèles dynamiques pour calculer le remaillage.
- *Remaillage anisotrope* : l’algorithme essaie de prendre en compte les directions principales de courbure pour aligner localement les triangles et/ou les rectangles nouvellement créés. L’algorithme proposé dans [1] utilise les lignes de courbure pour construire un maillage à dominance rectangulaire. Les méthodes d’éléments finis utilisent énormément d’algorithmes de remaillage de cette sorte [14].

Notre méthode repose sur une distribution de points isotrope de façon à se ramener à un problème de type *Eikonal*.

Le dernier point abordé dans ce papier est la paramétrisation de variétés triangulées. Parmi les méthodes les plus efficaces on peut citer :

- *Les approches multirésolution* : l’algorithme MAPS [15] utilise une décomposition multirésolution pour construire la paramétrisation. Cette méthode marche bien pour choisir les points de bases sur la variété. Par contre, il est difficile de contrôler les frontières des régions servant à définir la paramétrisation.
- *Les approches utilisant une minimisation d’énergie* : introduites en premier par *Floater* [11], ces méthodes consistent en la résolution d’un système linéaire pour minimiser une certaine mesure de distorsion. Les chemins géodésiques sont souvent choisis comme frontières pour subdiviser la variété. Des choix canoniques pour l’énergie sont proposés dans [9].
- *Aplatissement géodésique de surface* : dans [21] *Zigelman et al* utilisent des informations géodésiques. Ils emploient le *multidimensional scaling* pour aplatir la variété tout en essayant de maintenir le plus possible les informations de distance.

2 Calculs géodésiques

2.1 Fast Marching sur une grille orthogonale

L’algorithme de Fast Marching classique est présenté dans [16]. Il permet de résoudre l’équation *Eikonale* qui est non linéaire :

$$\|\nabla U(x)\| = P(x), \quad (1)$$

où U représente la fonction de distance pondérée à un ensemble de points du plan, et $F = 1/P \geq 0$ est la vitesse de propagation du front. Il est employé notamment pour résoudre le problème des contours actifs [7]. Dans

cette section, nous considérons une triangulation orthogonale comme celle représentée sur la gauche de la figure 2. L'algorithme de Fast Marching utilise un schéma aux dif-

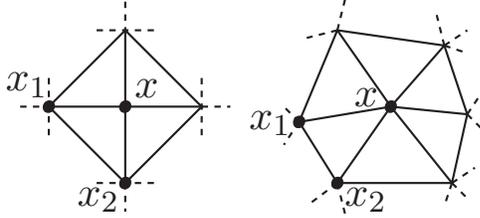


FIG. 2 – *Triangulation orthogonale et quelconque.*

férences finis décentré en amont pour calculer la valeur de u en un point $x_{i,j}$ de la grille :

$$\max(u - U(x_{i-1,j}), u - U(x_{i+1,j}), 0)^2 + \max(u - U(x_{i,j-1}), u - U(x_{i,j+1}), 0)^2 = h^2 P(x_{i,j})^2.$$

Un ordonnancement optimal des points de la grille permet de faire l'ensemble des calculs en un temps de l'ordre de $O(N \log(N))$, où N est le nombre de points.

2.2 Etude sur une triangulation plus générale

Nous suivons [17], dans le but de généraliser la construction précédente à une triangulation quelconque, comme celle représentée sur le côté droit de la figure 2. De façon plus précise, nous essayons de calculer la valeur de $U(x)$ à l'intérieur du triangle $\langle x, x_1, x_2 \rangle$, si ceci est possible. Comme plusieurs triangles autour de x peuvent fournir une valeur acceptable, nous ne prendrons en compte que la plus petite valeur. La difficulté est alors d'approcher la valeur de U à l'intérieur du triangle $\langle x, x_1, x_2 \rangle$, puisqu'il n'y a pas de système de coordonnées «naturel» (contrairement au cas d'une grille orthogonale). Dans [17], un tel système est choisi, et les calculs sont effectués grâce à des identités trigonométriques.

D'une façon plus générale, [18] propose une méthode de calcul du gradient en dimension arbitraire. Cette méthode revient à considérer la matrice M de taille 2×2 dont les lignes sont formées des vecteurs \vec{xx}_1 et \vec{xx}_2 . Nous pouvons alors calculer les dérivées de U en x selon les directions des vecteurs \vec{xx}_1 et \vec{xx}_2 , en utilisant une formule d'ordre 1 :

$$\nu := \begin{pmatrix} U(x) - U(x_1) \\ U(x) - U(x_2) \end{pmatrix} = U(x)a + b,$$

où $a = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ et $b = -\begin{pmatrix} U(x_1) \\ U(x_2) \end{pmatrix}$. Or, la dérivée selon la direction \vec{xx}_k est par définition égale à $\langle \nabla U, \vec{xx}_k \rangle$, d'où la formule $\nabla U \simeq M^{-1}\nu$. En introduisant ces relations dans l'équation Eikonale (1), et en notant $Q := (MM^T)^{-1}$, on obtient :

$$(a^T Q a)U(x)^2 - (2a^T Q b)U(x) + b^T Q b = P(x)^2.$$

Ceci est une équation du second degré, dont le discriminant est positif par la formule de *Cauchy-Schwartz*. Le critère du schéma en amont nous impose, comme dans le cas d'une grille orthogonale, que $-\nabla U(x)$ pointe vers l'intérieur du triangle. On peut alors montrer que si l'angle en x est aigu, alors le schéma est monotone, ce qui signifie que l'on peut trouver $U(x) \geq U(x_2) \geq U(x_1)$ pour tout $U(x_i)$.

La figure 3 montre la propagation d'un front et le calcul des chemins géodésiques (voir 2.3).

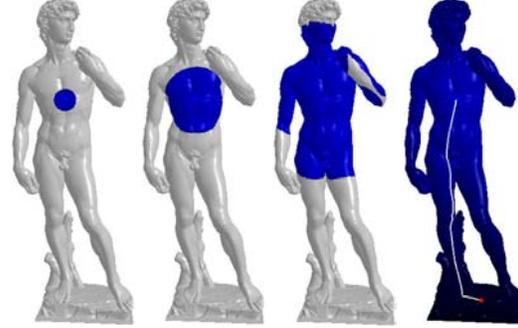


FIG. 3 – *Propagation de front et chemin géodésiques (sur la droite).*

Par contre, si la triangulation contient des angles obtus, alors le schéma numérique n'est plus monotone, ce qui peut conduire à des instabilités. Pour résoudre ce problème, nous suivons *Sethian* qui propose de «déplier» les triangles dans une région où l'on est sûr que l'étape de mise à jour marchera [17].

2.3 Extraction de géodésiques

Une fois que l'on a calculé la fonction U , qui représente la distance géodésique à un sommet v , nous avons besoin de déterminer un chemin géodésique joignant un point de départ v_0 à v . Pour y parvenir, nous cherchons la courbe paramétrée $C(t)$ vérifiant l'équation de rétro-propagation [7] :

$$\begin{cases} \frac{dC}{dt} = -\overline{\nabla U} \\ C(0) = v_0 \end{cases}.$$

Nous résolvons cette équation par une méthode numérique comme celle de *Runge Kutta* d'ordre 4 [10]. Dans ce but, nous devons calculer la valeur de $\overline{\nabla U}$ à n'importe quel endroit de la surface.

Pour obtenir une variation continue de $\overline{\nabla U}$ sur la surface, nous introduisons un nouveau schéma d'interpolation qui diffère de celui employé dans [17]. Tout d'abord nous calculons la valeur du gradient de U en chaque sommet v de la triangulation. Ensuite, nous interpolons linéairement les valeurs de $\overline{\nabla U}$ sur chaque face. Pour estimer la valeur de $\overline{\nabla U}$ en un sommet v , nous considérons le 1-voisinage représenté sur la figure 4 (gauche), où les v_i sont les voisins de v . En remplaçant, de manière proportionnelle, chaque angle α_i par $\tilde{\alpha}_i$ de façon à obtenir une somme de 2π , nous aplatissons le voisinage de v sur un plan (figure 4, sur la droite). Nous pouvons alors approcher la valeur du gradient

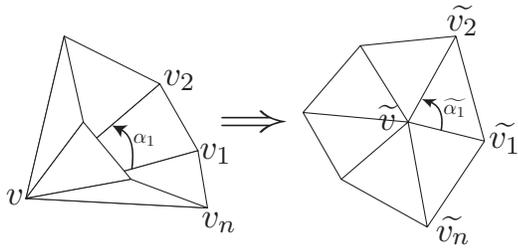


FIG. 4 – Aplatissement d'un voisinage.

sur chaque arête (v, v_{i+1}) :

$$\nabla U^{(i)} = \frac{U(v_i) - U(v)}{|v_i - v|}.$$

Le gradient en v_i satisfait donc $\langle \nabla U, e_i \rangle = \nabla U^{(i)}$, où $e_i := \frac{\tilde{v} - \tilde{v}_i}{\|\tilde{v} - \tilde{v}_i\|}$. Nous résolvons ce système linéaire sur-déterminé au sens des moindres carrés, ce qui revient à résoudre ce système 2×2 :

$$\begin{pmatrix} \sum e_{i0}e_{i0} & \sum e_{i0}e_{i1} \\ \sum e_{i0}e_{i1} & \sum e_{i1}e_{i1} \end{pmatrix} \begin{pmatrix} (\nabla U)_0 \\ (\nabla U)_1 \end{pmatrix} = \begin{pmatrix} \sum e_{i0}\nabla U^{(i)} \\ \sum e_{i1}\nabla U^{(i)} \end{pmatrix},$$

où (e_{i0}, e_{i1}) représente les coordonnées de e_i dans un système de coordonnées locales du plan d'aplatissement.

La figure 5 montre le calcul des chemins géodésiques pour différents modèles. Il est aussi possible de démarrer



FIG. 5 – Chemins géodésiques.

plusieurs fronts pour les faire évoluer ensemble, comme ceci est montré à la figure 6. Les différentes régions colorées forment un diagramme de *Voronoi* des points de départ, et les intensités représentent les distances géodésiques au point le plus proche.

3 Remaillage isotrope d'une triangulation

Notre approche ajoute des nouveaux sommets en utilisant les distances géodésiques sur la surface. Le résultat de

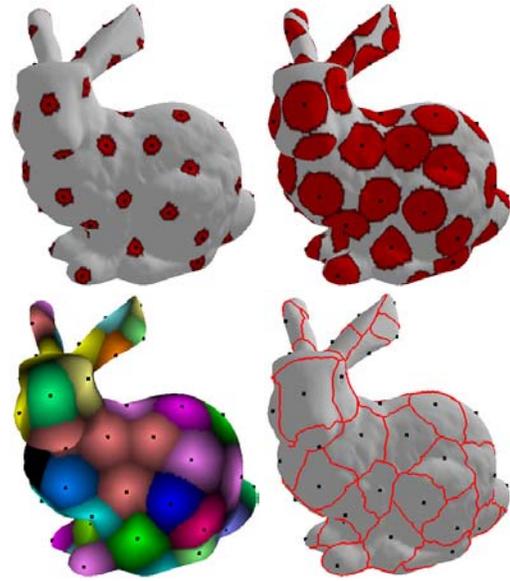


FIG. 6 – Calcul du diagramme de Voronoi (voir la figure en couleurs dans la version électronique).

l'algorithme est un ensemble de sommets distribué de façon uniforme sur la surface suivant la distance géodésique. La prise en compte d'une densité locale de sommets est abordée aux paragraphes 3.3 et 3.4.

Ceci constitue une extension de l'algorithme de [8] qui met à jour une propagation de front depuis un ensemble de points dans le but de trouver, de manière itérative, des paires de points à relier entre eux. On pourra aussi trouver des idées similaires dans [5]. Dans notre contexte, il y a deux principales différences. La première est que nous trouvons des points sur une surface triangulée à la place d'une image. La deuxième est que nous calculons une approximation précise du point où la distance géodésique maximale est atteinte.

3.1 Choix itératif des points de base

Nous allons maintenant décrire comment construire automatiquement un ensemble de points répartis de manière uniforme sur la surface triangulée. Un premier point x_1 est choisi au hasard sur la triangulation. Un choix plus élaboré consiste à remplacer ce point aléatoire par le point qui lui est le plus éloigné.

Supposons maintenant que l'on ait déjà calculé un ensemble de points $S = \{x_1, \dots, x_n\}$, en même temps que U_n , la carte de distance géodésique à S . Pour ajouter un nouveau point x_{n+1} , nous sélectionnons simplement un point de la variété qui est le plus éloigné de S , c'est à dire qui atteint le maximum de U_n . On note $U_{x_{n+1}}$ la carte de distance au point x_{n+1} . Pour calculer la nouvelle carte de distance U_{n+1} , nous utilisons le fait que U_{n+1} est égal à $\min(U_n, U_{x_{n+1}})$. Ainsi, nous n'avons besoin que de mettre à jour U_n en démarrant un front depuis x_{n+1} (en utilisant l'algorithme de Fast Marching exposé à la section 2), en prenant soin de le confiner à $\{x / U_{x_{n+1}}(x) \leq U_n(x)\}$.

Ceci assure que le processus de distribution des points prend grossièrement moins de $O(N \log(N)^2)$ opérations (ce qui serait le cas si l'on désire distribuer autant de points de base qu'il y a de sommets dans la triangulation originale, i.e. N).

A chaque itération, le nouveau point x_{n+1} n'est pas nécessairement un sommet du maillage original. Il peut être positionné de façon précise en interpolant la carte de distances. Pour calculer les points d'intersection de deux fronts, on autorise les fronts à s'interpénétrer sur une profondeur de 1 sommet. Pour placer x_{n+1} , on commence par déterminer le triangle où la distance minimale est atteinte. Dans la plus part des cas, 3 fronts se rencontrent sur ce triangle. On détermine les 3 points d'intersection selon les 3 arêtes. On place x_{n+1} à l'isobarycentre de ces 3 points. Il faut bien noter que v est aussitôt ajouté à l'ensemble des sommets du maillage original.

Nous décidons d'arrêter l'algorithme soit quand le dernier point ajouté x_{n+1} vérifie $U_n(x_{n+1}) \leq \delta$ (où δ est un seuil fixé), soit quand un nombre de points donné a été distribué.

La figure 7 montre trois étapes du procédé d'insertion de points sur un carré (avec en plus la triangulation obtenue, qui est expliquée au paragraphe suivant). L'intensité reflète la distance géodésique au point de base le plus proche.

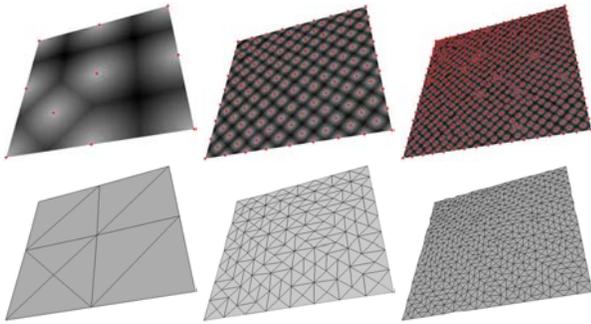


FIG. 7 – Insertion de points dans un carré.

3.2 Calcul des triangles géodésiques

Un fois que l'on dispose de l'ensemble complet de points de base S_{n_0} , il nous faut déterminer quels points relier entre eux pour obtenir notre triangulation de base \mathcal{T} qui est construite de manière incrémentale durant l'algorithme. Pour y parvenir, pendant la distribution des points de base, nous gardons trace des *points selles*, qui sont les sommets v qui satisfont les trois critères suivants :

- Quand la valeur de $U(v)$ est calculée par l'algorithme de Fast Marching, deux fronts provenant de deux points de bases différents v_i et v_j doivent se rencontrer pour la première fois en v .
- L'ajout de l'arête $[v_i, v_j]$ à la triangulation de base \mathcal{T} doit conserver la triangulation valide (par exemple l'arête doit avoir moins de deux faces adjacentes).

Il faut bien noter que quand la carte de distance U_{n+1} est mise à jour, un point selle v peut disparaître (si v est atteint par le front provenant de v_{n+1}), et bien sûr, de nouveaux points selles peuvent être créés.

L'ensemble des points selles nous indique quels points de base doivent être reliés entre eux pour obtenir une triangulation valide \mathcal{T} . Nous pouvons aussi tracer sur le maillage original les chemins géodésiques correspondant aux arêtes de \mathcal{T} . Si v est un point selle où deux fronts provenant de v_i et v_j se rencontrent, il suffit de réaliser une descente de gradient comme décrit à la section 2.3, depuis v , à la fois dans la direction de v_i et dans la direction de v_j .

La figure 8 montre la triangulation de Delaunay \mathcal{T} obtenue (sur la gauche), ainsi que les triangles géodésiques correspondants (sur la droite) obtenus pour une sphère et un tore.

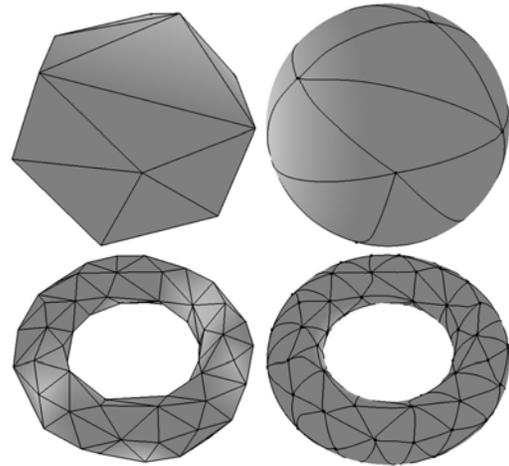


FIG. 8 – Triangulation de Delaunay de base (gauche) et géodésique (droite) pour deux exemples.

Enfin, il se peut que des trous apparaissent dans la triangulation obtenue (par exemple s'il y a des trous dans le maillage original). Si nous voulons éviter de tels trous (par exemple si nous voulons utiliser le nouveau maillage pour paramétrer la variété), il nous faut remplir ces trous. Ceci peut être effectué en «dépliant» le trou sur un plan, puis en calculant une triangulation de Delaunay contrainte de l'intérieur du trou aplati.

3.3 Remaillage adaptatif

L'algorithme présenté aux paragraphes 3.1 et 3.2 peut être étendu pour travailler avec une fonction de vitesse F arbitraire pour la propagation du front. Dans les régions où F est élevée, la triangulation résultante va être dense, alors que dans les régions où F est faible, la triangulation va être plus lâche. Cette fonction F peut refléter un besoin de l'utilisateur de raffiner certaines régions spécifiques en y plaçant plus de sommets.

La figure 9 montre une distribution uniforme de points (à gauche) et une distribution de points avec une séparation du maillage en deux régions, une avec un F élevé, et une autre avec un F faible (à droite).

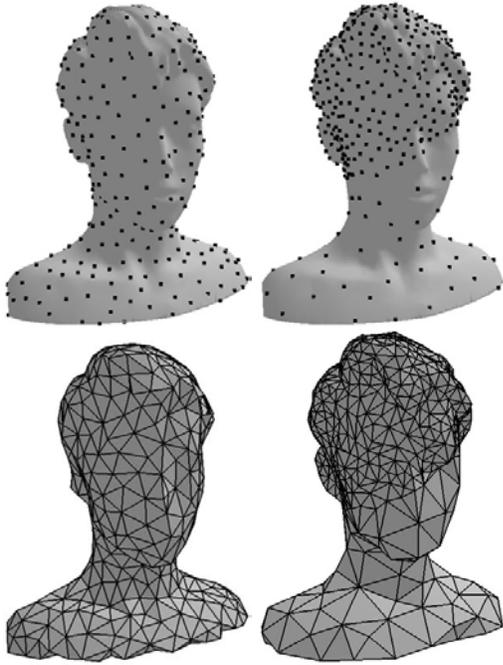


FIG. 9 – Distribution de points uniforme (à gauche) et prescrite par l'utilisateur (à droite). Triangulations correspondantes (en bas).

Quand un maillage est obtenu par acquisition laser tri-dimensionnelle, une image du modèle peut être plaquée sur l'objet 3D. En utilisant une fonction F qui est inversement proportionnelle à la norme du gradient de l'image, l'utilisateur peut raffiner certaines régions avec des fortes variations d'intensité. La figure 10 montre un tel modèle 3D, accompagné de la fonction de vitesse F utilisée, ainsi que la distribution de points résultante.

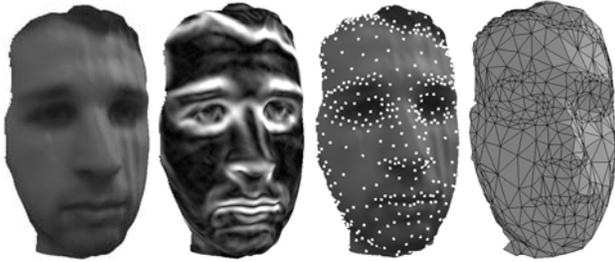


FIG. 10 – Modèle original, fonction F , distribution adaptative de points, et triangulation correspondante.

3.4 Remaillage utilisant les informations de courbure

La densité locale de sommets peut aussi refléter des propriétés géométriques de la surface originale. Le choix le plus naturel est d'adapter le nouveau maillage de façon à être plus fin dans les régions où la courbure locale est plus forte. Pour y parvenir, nous avons choisi une fonction de vitesse F inversement proportionnelle à $\max(|\lambda_1|, |\lambda_2|)$, où les λ_i sont les valeurs propres du tenseur de courbure local (deuxième forme fondamentale).

La figure 11 montre, pour deux modèles avec des bords saillants, une comparaison entre deux types de remaillages. Le premier ne prend pas en compte la courbure, et les bords ne sont pas respectés. Le deuxième utilise les informations de courbure, et les modèles sont presque parfaitement reconstruits.

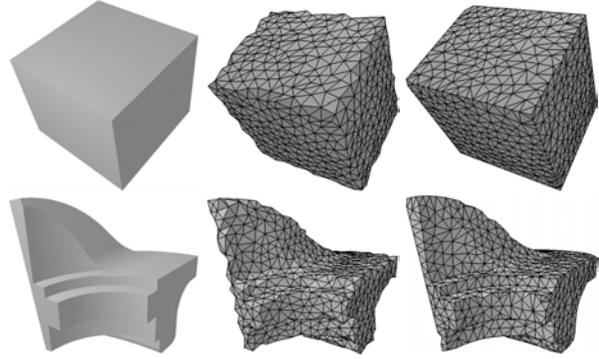


FIG. 11 – Modèle original (gauche), remaillage sans prise en compte de la courbure (milieu) et avec prise en compte de la courbure (droite).

4 Application à la paramétrisation de maillage

4.1 Position du problème

Une paramétrisation f d'une variété triangulée 3D \mathcal{V} correspond à un ensemble d'applications $f_i : \mathcal{U}_i \rightarrow \mathcal{V}$, où les $\mathcal{U}_i \subset \mathbb{R}^2$, et les f_i sont des homéomorphismes dont les images couvrent tout \mathcal{V} . Pour simplifier, nous allons considérer une variété triangulée \mathcal{V}_0 qui ne contient que très peu de triangles, mais qui a la même topologie que \mathcal{V} . Les applications f_i seront linéaires par morceaux, et les \mathcal{U}_i seront les triangles de \mathcal{V}_0 . La figure 12 montre une telle paramétrisation. Les fonctions f_i^{-1} sont affines sur chaque petit triangle de \mathcal{V} , et l'image de chaque triangle \mathcal{U}_i de \mathcal{V}_0 par f_i délimite une région de \mathcal{V} .

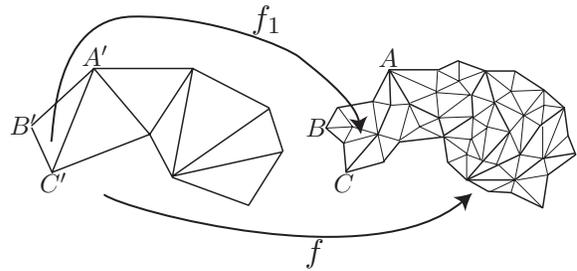


FIG. 12 – Paramétrisation linéaire par morceaux.

Comme domaine de base pour la paramétrisation f , nous choisissons le maillage de base construit avec notre algorithme présenté à la section 3. Ainsi les frontières de chaque région $f_i(\mathcal{U}_i)$ sont des chemins géodésiques et ces chemins doivent être ajoutés au maillage original. La figure 13 montre comment effectuer cette inclusion.

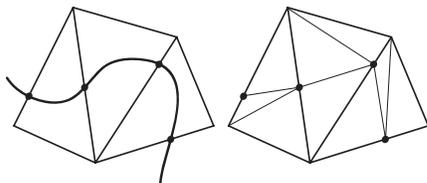


FIG. 13 – La triangulation avant l’inclusion d’un chemin (gauche) et après (droite).

Comme à l’intérieur de chaque petit triangle d’une région $f_i(\mathcal{U}_i)$ la fonction f_i^{-1} est affine, pour chaque sommet x de \mathcal{V} , nous avons uniquement besoin de connaître les coordonnées barycentriques de $f_i^{-1}(x)$ à l’intérieur du triangle \mathcal{U}_i . La question qui reste en suspens est de savoir comment calculer ces coordonnées barycentriques judicieusement.

4.2 Interpolation de la paramétrisation

La dernière étape est le calcul des coordonnées barycentriques (ν_1, ν_2, ν_3) de $f^{-1}(v)$ à l’intérieur d’un triangle donné $\langle x_1, x_2, x_3 \rangle$, pour chaque sommet v . Il faut bien noter que les points $v_1 = f(x_1)$, $v_2 = f(x_2)$, et $v_3 = f(x_3)$ sont des points de base déterminés au paragraphe 3.1. Pour y parvenir, nous allons utiliser les distances géodésiques (t_1, t_2, t_3) entre v et chacun des v_i . En fait, nous allons placer le point $f^{-1}(v)$ à l’intérieur du triangle $\langle x_1, x_2, x_3 \rangle$ en conservant les distances géodésiques.

Pour y parvenir, en connaissant les distances (t_1, t_2, t_3) d’un point x à trois sommets (x_1, x_2, x_3) d’un triangle, nous avons besoin de calculer les coordonnées barycentriques (ν_1, ν_2, ν_3) de x de façon analogue au cas d’un triangle plan. Nous notons (l_1, l_2, l_3) les longueurs des côtés opposés à chaque point (par exemple, l_1 est donné par le temps d’arrivée en v_3 du front issu de v_2). Maintenant, dans le plan, la formule de Héron affirme que l’aire A_3 du triangle $\langle x, x_1, x_2 \rangle$ est $A_3 = \sqrt{p(p-l_3)(p-t_1)(p-t_2)}$, avec $p := (t_1 + t_2 + l_3)/2$. La figure 14 montre le triangle considéré. D’une façon similaire, nous pouvons cal-

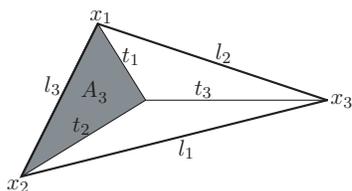


FIG. 14 – Détermination des coordonnées barycentriques d’un point x

culer A_1 et A_2 , ce qui nous permet d’obtenir les coordonnées barycentriques par $\nu_i = \frac{A_i}{A_1 + A_2 + A_3}$. Ces formules nous donnent la paramétrisation d’un point x , à partir du moment où aucun autre point n’a les mêmes valeurs pour (t_1, t_2, t_3) . Cette condition est satisfaite en pratique, à condition que \mathcal{V}_0 soit topologiquement équivalent à \mathcal{V} , et que le maillage de base soit suffisamment fin.

Pour calculer les trois distances (t_1, t_2, t_3) pour chaque

sommet v , nous démarrons simplement un front depuis v , en prenant soin de le confiner au voisinage géodésique de v , comme représenté sur la gauche de la figure 15. En faisant ceci, nous sommes sûrs que chaque sommet du maillage original est atteint par trois fronts et seulement trois. Sur la droite de la figure 15, on peut voir la paramétrisation géodésique obtenue. Nous avons simplement assigné une couleur aléatoire à chaque sommet de base, et nous avons interpolé la couleur en utilisant la paramétrisation.

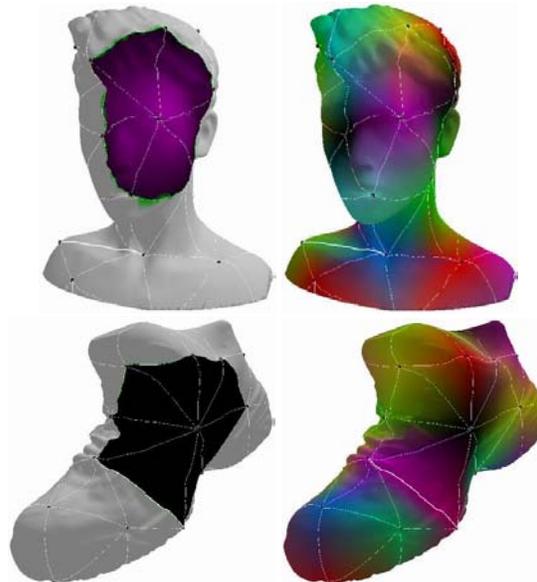


FIG. 15 – Exemples de paramétrisations (voir la figure en couleur dans la version électronique).

4.3 Application à la construction de maillages multirésolution

Il est possible d’échantillonner chaque triangle de base suivant un schéma de subdivision régulier, comme ceux représentés à la figure 16. La triangulation résultante est appelée un maillage *semi-régulier*. Cette classe de trian-

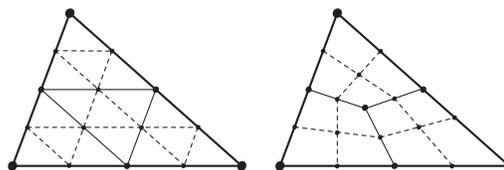


FIG. 16 – Schéma de subdivision à base triangulaire et rectangulaire.

gulation est de première importance, à la fois parce que la connectivité est connue à l’avance (ce qui est un grand avantage pour la compression), et parce que le procédé de subdivision aboutit à une représentation multirésolution naturelle du maillage (permettant de réaliser une compression par ondelettes). La figure 17 montre le processus de subdivision.

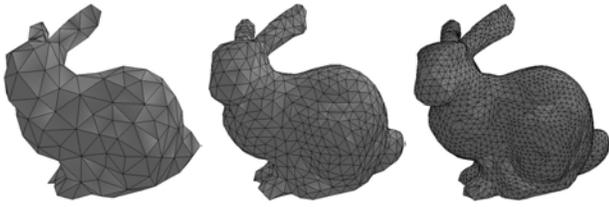


FIG. 17 – Trois étapes de subdivision semi-régulière.

5 Aplatissement de surfaces

Dans cette section, nous montrons une dernière application de notre algorithme de distribution de points introduit au paragraphe 3. Il s'agit d'aplatir une surface en commençant par aplatir l'ensemble des points fournis par notre algorithme.

5.1 Aplatissement de l'ensemble des points de base

Le problème de l'aplatissement de surface est un cas particulier du problème de paramétrisation présenté au paragraphe 4.1. Il s'agit, étant donnée une variété triangulée \mathcal{V} homéomorphe à un disque, de trouver une carte $f : \mathcal{U} \rightarrow \mathcal{V}$, où $\mathcal{U} \subset \mathbb{R}^2$ est un certain domaine plan.

Pour y parvenir, nous utilisons l'algorithme proposé par *Zigelman et Al.* dans [21], en l'appliquant à l'ensemble des points de base $\{x_1, \dots, x_n\}$ déterminés par notre algorithme. Nous l'améliorons d'une part en proposant une méthode automatique pour déterminer les points de bases, d'autre part en utilisant un schéma d'interpolation original utilisant des calculs de distances sur la surface.

Pour résoudre le problème de l'aplatissement, on souhaite trouver des points $f(x_i) = \tilde{x}_i \in \mathbb{R}^2$ qui vérifient $d(x_i, x_j) \approx \|\tilde{x}_i - \tilde{x}_j\|$, où d désigne la distance géodésique sur la surface \mathcal{V} , et $\|\cdot\|$ désigne la norme euclidienne sur \mathbb{R}^2 . Pour ce faire, *Zigelman* utilise la méthode de réduction de dimension appelée *multidimensional scaling*. Il s'agit de construire la matrice carrée de taille M telle que $M_{ij} = d(x_i, x_j)^2$. On note ensuite $J = Id - \frac{1}{n}\mathbf{1}$, où $\mathbf{1}$ désigne la matrice dont toutes les entrées sont égales à 1. Le centrage en moyenne de M est effectué par le calcul de $B = -\frac{1}{2}JMJ$. L'aplatissement est réalisé par le calcul des deux plus grandes valeurs propres λ_1 et λ_2 de B , ainsi que des vecteurs propres associés u_1 et u_2 . La position du point \tilde{x}_i est alors donnée par $\tilde{x}_i = (\sqrt{\lambda_1}u_1[i], \sqrt{\lambda_2}u_2[i])^T$.

5.2 Interpolation de l'aplatissement

On peut alors se demander pourquoi on n'applique pas cette technique d'aplatissement à tous les points du maillage \mathcal{V} d'origine. Ceci est dû au fait que le calcul de la matrice M , lorsque le nombre de points considérés est grand, prend beaucoup de temps. Il est donc nécessaire de se restreindre à un petit nombre de points représentatifs, et notre algorithme fournit justement des points bien répartis sur la surface.

Une fois les positions des \tilde{x}_i déterminées, il est donc nécessaire de les interpoler pour pouvoir calculer l'image

$f(v) \in \mathbb{R}^2$ d'un sommet v quelconque de notre maillage. Pour ce faire, on va utiliser une interpolation barycentrique, qui sera introduite au paragraphe suivant. A chaque sommet v elle associe, pour $i = 1 \dots n$, un ensemble de poids $\lambda_i(v)$ tels que

$$\lambda_i(v) \geq 0, \quad \sum_{i=1}^n \lambda_i(v) = 1 \quad \text{et} \quad \lambda_i(x_j) = \delta_i^j. \quad (2)$$

Il est alors simple de déterminer l'image de v , en posant

$$f(v) = \sum_{i=1}^n \lambda_i(v) \tilde{x}_i.$$

5.3 Interpolation par voisins naturels

La méthode d'interpolation par *voisins naturels* a été introduite par *Sibson* dans [19]. Il s'agit d'utiliser les informations fournies par les diagrammes de Voronoi pour construire une interpolation barycentrique régulière.

Dans ce paragraphe, nous allons brièvement expliquer comment elle fonctionne, et montrer comment on peut la calculer sur une surface triangulée grâce à l'algorithme de Fast Marching. Pour chaque point de base x_i , on note

$$V_i := \{x \in \mathcal{V} \mid \forall j \neq i, \quad d(x, x_j) > d(x, x_i)\}$$

la cellule de Voronoi associée à x_i . Si on considère un sommet v quelconque de la triangulation, on peut aussi définir la cellule de Voronoi associée à v pour l'ensemble des points $\{v, x_1, \dots, x_n\}$:

$$V(v) := \{x \in \mathcal{V} \mid \forall j, \quad d(x, x_j) > d(x, v)\}.$$

La figure 18 montre les ensembles V_1 et $V(v)$ sur une surface plane. On définit alors les poids $\lambda_i(v)$ par

$$\lambda_i(v) := \frac{\text{Aire}(V(v) \cap V_i)}{\text{Aire}(V(v))}.$$

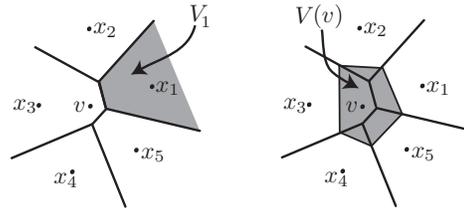


FIG. 18 – Cellules de Voronoi sur un domaine plan.

Il est facile de voir que les conditions (2) sont bien respectées, et que les fonctions λ_i ont un support très compact autour de v_i . De plus, dans le cas d'une surface plane, on peut montrer que les fonctions λ_i sont continues, et de classe C^∞ sauf aux points x_i . De plus, cette méthode possède une propriété de reproduction des coordonnées, puisque l'on a $\sum \lambda_i(x) x_i = x$. Cette propriété est très agréable pour une paramétrisation, puisqu'elle assure que

si la triangulation de départ est plane, et que $x_i \mapsto \tilde{x}_i$ est affine, alors la paramétrisation f est encore affine.

L'aplatissement de triangulation est très important pour réaliser du plaquage de texture. Ceci consiste à appliquer une image $I : [0, 1] \times [0, 1] \mapsto [0, 255]$ sur la surface \mathcal{V} avec le minimum de distorsion possible. Quitte à dilater notre paramétrisation, on peut supposer que l'on a $f(\mathcal{V}) \subset [0, 1] \times [0, 1]$. Le plaquage de texture est alors réalisé facilement, en assignant à chaque point $v \in \mathcal{V}$ la position $f(v)$ dans $[0, 1] \times [0, 1]$, donc en particulier la couleur $I(f(v))$.

5.4 Calcul de l'interpolation par Fast Marching

Les coefficients $\lambda_i(v)$ peuvent être calculés de façon très rapide grâce à l'algorithme de Fast Marching. On rappelle que grâce à notre algorithme présenté au paragraphe 3, on dispose de U_n , la carte de distance géodésique aux points $\{x_1, \dots, x_n\}$. En interpolant les frontières où les différents fronts se rencontrent (comme on l'a déjà fait à la section 3.1 pour placer x_{n+1}), on est ainsi en mesure de calculer les différentes cellules de Voronoi V_i . Pour chaque sommet v considéré, il nous faut calculer la cellule de Voronoi $V(v)$. Ceci est effectué très rapidement en faisant évoluer un front depuis v et en le confinant à la région $\{x \mid d(x, v) \leq U_n(x)\}$. Une fois de plus, les frontières de $V(v)$ sont interpolées linéairement.

Le seul point difficile est le calcul de l'aire de la région $V(v) \cap V_i$. Ceci est effectué progressivement en considérant, sur chaque triangle T de \mathcal{V} l'aire de l'intersection de $V(v) \cap T$ et $V_i \cap T$. A cause du schéma d'interpolation linéaire utilisé, on voit facilement que chacune de ces deux parties sont des polygones convexes à 3 ou 4 côtés. Les différents cas sont représentés à la figure 19. A chaque fois, des flèches indiquent les provenances des fronts atteignant chaque sommet du triangle T . Ainsi, le calcul final ne nécessite que l'écriture d'un algorithme déterminant l'intersection de deux polygones convexes à 4 côtés.

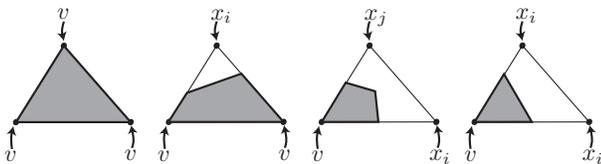


FIG. 19 – Les différentes configurations pour $V(v) \cap T$.

La figure 20 présente l'aplatissement du visage déjà rencontré, puis l'application d'une texture. On se référera à la figure 10 pour une vue du modèle non aplati. Les points de base sont marqués en foncé. Sur le visage aplati (gauche), les niveaux de gris ne sont qu'indicatifs et représentent l'ombrage du modèle d'origine (non aplati).

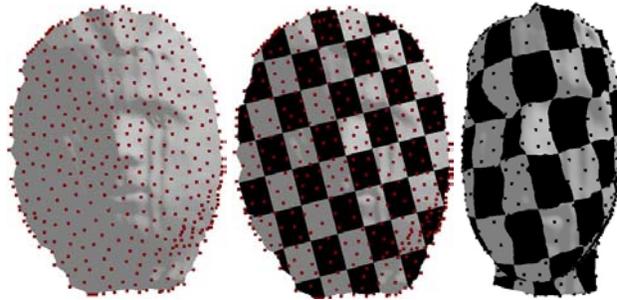


FIG. 20 – Surface aplatie, texture sur le domaine aplati, et sur le modèle d'origine.

6 Remarques finales et travaux à venir

Les temps d'exécution de notre algorithme sont très raisonnables, et il faut environ 10 secondes pour distribuer 500 points sur un modèle d'environ 5000 sommets. La distribution de plus de points est pratiquement immédiate, grâce à la structure incrémentale de notre méthode. La construction du maillage basse résolution et l'étape de paramétrisation prend approximativement le même temps. Ainsi notre implémentation actuelle nous permet de traiter un modèle complet en moins d'une minute.

L'algorithme de remaillage isotrope proposé dans cet article donne de très bons résultats, similaires à ceux présentés dans [20] pour un maillage homéomorphe à un disque, et dans [2] pour un maillage plus général. Même pour des maillages complexes de genre élevé (comme le David), notre algorithme capture la topologie de la surface très rapidement. La propriété de répartition uniforme des distances entre les points est moins forte qu'une propriété de répartition en densité des points, mais elle semble suffisante pour la plupart des applications, et permet des calculs quasiment en temps réel. De plus, elle permet, sans modifier l'algorithme, d'inclure de très nombreuses contraintes telles que la prise en compte de la courbure ou d'une texture.

Cette adaptabilité du remaillage est donc un réel apport de notre méthode. Les courbes de la figure 21 représentent les distances de Hausdorff entre la surface de départ (figure 11 en bas à droite) et la surface remaillée avec de plus en plus de points. Elles montrent que, dans le cas d'un maillage avec des bords saillants, notre remaillage adaptatif apporte une nette amélioration par rapport au remaillage uniforme.

Le schéma de paramétrisation donne un atlas de cartes régulières dans les régions où la variété est lisse. Il reste encore à faire des tests de distorsion pour comparer notre méthode à des schémas classiques, comme celui de Floater [11].

La méthode d'aplatissement de surface proposée est originale de par l'extension de l'interpolation par voisins naturels à des surfaces. Elle est très rapide, grâce à l'utilisation de l'algorithme de Fast Marching et à une mise à jour incrémentale des cellules de Voronoi. Enfin, elle donne de

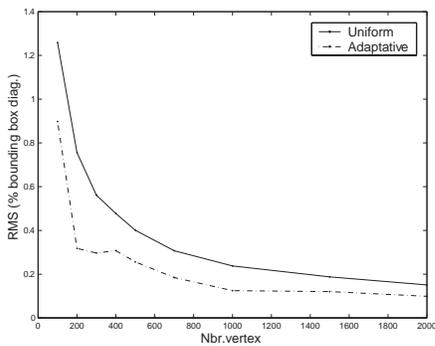


FIG. 21 – Erreur de remaillage mesurée par la distance de Hausdorf.

très bons résultats à cause des propriétés de régularité et de reproduction des coordonnées du schéma d'interpolation utilisé.

Les méthodes de calculs géodésiques prenant en compte des contraintes externes apparaissent donc comme capables de résoudre de nombreux problèmes géométriques avec élégance et simplicité. La rapidité des algorithmes présentés laisse espérer un champ d'applications bien plus large que celui du remaillage 3D. Par exemple, l'analyse de données géométriques (qui est au cœur des problèmes de reconnaissance) peut exploiter les résultats des calculs de distances et de paramétrisation comme autant de caractéristiques intrinsèques d'un objet donné.

Nos directions de recherche futures incluent :

- Prendre en compte des zones spéciales du maillage (comme par exemple les arêtes où il y a des irrégularités) pendant le remaillage et la paramétrisation. Il est possible de définir une densité linéaire de sommets le long de ces zones 1D.
- Trouver des bornes théoriques pour la distorsion de la paramétrisation dans le cas de surfaces lisses.

7 Conclusion

Nous avons décrit un algorithme itératif pour distribuer un ensemble de points sur une surface en utilisant une fonction de densité locale. Cet algorithme est rapide et efficace, et aboutit à une méthode simple pour trouver automatiquement une triangulation de Delaunay géodésique de la surface originale. Nous avons appliqué cette construction pour un domaine de base pour la paramétrisation d'une variété triangulée. Cette méthode permet d'obtenir un atlas de cartes lisses du maillage de base vers la triangulation originale, permettant des constructions multirésolution et de la compression en ondelettes. Nous avons également développé une méthode d'aplatissement de surface qui généralise l'interpolation par voisins naturels à des variétés triangulées.

Références

[1] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy, et M. Desbrun. Anisotropic Polygonal Remeshing. *ACM Transactions on Graphics*, 2003.

[2] P. Alliez, E. Colin de Verdière, O. Devillers, et M. Isenburg. Isotropic Surface Remeshing. *International Conference on Shape Modeling and applications*, 2003.

[3] M. Bertalmio, G. Sapiro, L-T. Cheng, et S. Osher. Variational Problems and PDE's on Implicit Surfaces. *Proc. IEEE VLSM 2001*, pages 186–193, 2001.

[4] J. Chen et Y. Hahn. Shortest Path on a Polyhedron. *Proc. 6th ACM Sympos. Comput Geom.*, pages 360–369, 1990.

[5] L. Cohen et T. Deschamps. Groupement de composantes connexes à l'aide de chemins minimaux. In *Actes de la conférence RFIA'02*, pages 723–732, Angers, Janvier 2002.

[6] Laurent D. Cohen et Isaac Cohen. Finite Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-15, 1993.

[7] Laurent D. Cohen et R. Kimmel. Global Minimum for Active Contour models: A Minimal Path Approach. *International Journal of Computer Vision*, 24(1):57–78, Aug. 1997.

[8] L.D. Cohen. Multiple Contour Finding and Perceptual Grouping Using Minimal Paths. *Journal of Mathematical Imaging and Vision*, 14(3), 2001. Presented at VLSM01.

[9] M. Desbrun, M. Meyer, et P. Alliez. Intrinsic Parameterizations of Surface Meshes. *Eurographics conference proceedings*, 21(2):209–218, 2002.

[10] W.H. Press et Al. *Numerical Recipes in C: The Art of Computer Programming*. Cambridge University Press, 1988.

[11] M. S. Floater, K. Hormann, et M. Reimers. Parameterization of Manifold Triangulations. *Approximation Theory X: Abstract and Classical Analysis*, pages 197–209, 2002.

[12] I. Guskov, K. Vidimce, W. Sweldens, et P. Schröder. Normal Meshes. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proc.*, Ann. Conf. Series, pages 95–102, 2000.

[13] R. Kimmel et J. A. Sethian. Fast Voronoi Diagrams on Triangulated Surfaces. In *Proc. of the 16th European Workshop on Comp. Geom. (EUROCG-00)*, pages 1–4, 2000.

[14] G. Kunert. Towards Anisotropic Mesh Construction and Error Estimation in the Finite Element Method. *Numerical Methods in PDE*, 18:625–648, 2002.

[15] A. W. F. Lee, P. Schröder, W. Sweldens, L. Cowsar, et D. Dobkin. MAPS: Multiresolution Adaptive Parameterization of Surfaces. *Computer Graphics*, 32(Ann. Conf. Series):95–104, 1998.

[16] J.A. Sethian. *Level Sets Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.

[17] J.A. Sethian et R. Kimmel. Computing Geodesic Paths on Manifolds. *Proc. Natl. Acad. Sci.*, 95(15):8431–8435, 1998.

[18] J.A. Sethian et A. Vladimirsky. Fast Methods for the Eikonal and Related Hamilton-Jacobi Equations on Unstructured Meshes. *Proc. Natl. Acad. Sci. USA*, 97(11):5699–5703, 2000.

[19] R. Sibson. A Vector Identity for the Dirichlet Tessellation. *Math. Proc. Cambridge Philos. Soc.*, 87:151–155, 1980.

[20] D. Terzopoulos et M. Vasilescu. Sampling and Reconstruction with Adaptive Meshes. In *IEEE CVPR '91*, pages 70–75, 1991.

[21] G. Zigelman, R. Kimmel, et N. Kiryati. Texture Mapping Using Surface Flattening via Multi-dimensional Scaling. *IEEE Trans. on Visualization and Computer Graphics*, 8(1):198–207, 2002.