Géométrie numérique

Séance 4 – Lissage et simplification de maillages

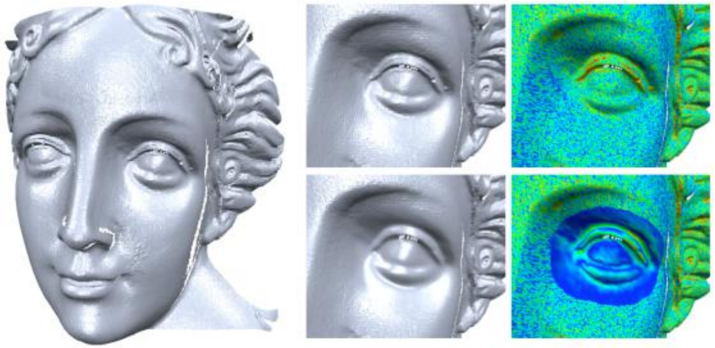
Franck Hétroy-Wheeler

M1 I3D et IIRVIJ – 2018-2019

UNIVERSITÉ DE STRASBOURG

Lissage : contexte

- Numérisation d'objet ⇒ **bruit** !



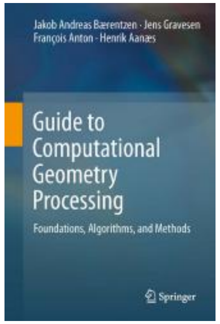
Courbure moyenne
Courtesy M. Botsch et al.

Théorie

- Bruit assimilable à **hautes fréquences** géométriques
- Débruitage : **filtre** passe-bas
- Nécessité d'une théorie du "traitement du signal géométrique"

Références bibliographiques

- Chapitre 4 du livre "Polygon Mesh Processing" : `http://www.pmp-book.org/`
- Chapitre 9 du livre "Guide to Computational Geometry Processing" : `https://books.google.fr/books?id=0lb4_pLIyP8C`



UNIVERSITÉ DE STRASBOURG

Transformée de Fourier

- Fonction univariée $f : \mathbb{R} \rightsquigarrow \mathbb{C}$
- **Transformée de Fourier** :

$$F(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-2i\pi\omega x} dx$$

- Transformée de Fourier inverse :

$$f(x) = \int_{-\infty}^{+\infty} F(\omega) e^{2i\pi\omega x} d\omega$$

- En discret : N échantillons du signal f

$$f(k) = \sum_{n=0}^{N-1} F(n) e^{2i\pi kn/N}$$

UNIVERSITÉ DE STRASBOURG

Filtrage

- **Filtre** = produit de la transformée de Fourier par une autre fonction L :
  - $F' = FL$
- Filtre passe-bas naïf : remplacer les hautes fréquences par 0
- En discret :
  - $L(n) = 0$ si n au-dessus d'un seuil
  - $L(n) = 1$ sinon
- Dans le domaine spatial : **convolution** $f * l$
  - $f'(x) = \int_{-\infty}^{+\infty} f(x-t)l(t)dt$ dans le cas continu
  - $f'(k) = f(k) * l(k) = \sum_{n} f(n)l(k-n)$ en discret

UNIVERSITÉ DE STRASBOURG

Interprétation géométrique

- Sous l'hypothèse que le filtre l est à support compact et symétrique :

$$f'(k) = \sum_{n=-w}^{w} f(n)l(k-n)$$

- **Moyenne pondérée** des valeurs de f au **voisinage** de k
- Filtre = fonction de pondération

Lissage laplacien

- Rappel : opérateur de **Laplace-Beltrami** appliqué à une fonction f pour un sommet v_i

$$\triangle f(v_i) = \frac{1}{\|N(v_i)\|} \sum_{v_j \in N(v_i)} (f(v_j) - f(v_i))$$

- Laplacien avec poids uniformes
- En pratique on l'applique à $f =$ fonction coordonnées, on note $x_i = f(v_i)$
- $\triangle x_i =$ moyenne des coordonnées des voisins de $v_i - x_i$
- $\Rightarrow \triangle x_i$ représente le **vecteur déplacement** du sommet v_i entre le maillage d'origine et le maillage filtré

Laplacien : représentation matricielle

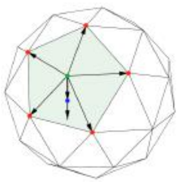
$$\triangle f(v_i) = \frac{1}{\|N(v_i)\|} \sum_{v_j \in N(v_i)} (f(v_j) - f(v_i))$$

- $L =$ **matrice laplacienne**, taille $n \times n$
- Diagonale : valeurs $-\frac{1}{\|N(v_i)\|}$
- Case (i,j) : valeur $\frac{1}{\|N(v_i)\|}$ si $v_j \in N(v_i)$, 0 sinon
- **Matrice creuse**
- Si f prend la valeur x_i en chaque v_i et qu'on note X le vecteur des x_i, le vecteur des $\triangle f(v_i)$ est le **produit matrice-vecteur** LX

Algorithme

- Paramètre λ
- Déplacer chaque sommet v_i dans la **direction du Laplacien** et d'une valeur **proportionnelle à λ** :

$$X \leftarrow X + \lambda LX$$



- λ contrôle le degré de lissage
- **Itérer** un nombre fixé de fois
- Voir **T.P.**

Exercice : lissage laplacien 2D

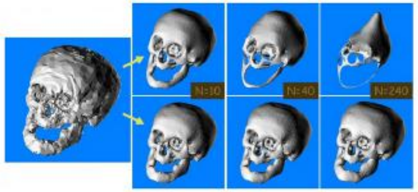
1. Dessiner un carré (suffisamment grand) comprenant 4 sommets sur chaque côté (dont les deux coins), soit 12 sommets au total.
2. Appliquer le lissage laplacien une première fois à tous les sommets, avec $\lambda = \frac{1}{2}$. Quels sont les sommets déplacés ?
3. Appliquer le lissage laplacien une deuxième fois. Quels sont les sommets déplacés ?
4. À votre avis, vers quelle forme converge le carré lissé ?

Lissage de Taubin

- Inconvénient : **perte de volume**
- Idée de Taubin (1995) : contre-balancer avec un **déplacement dans la direction opposée**

$$X \leftarrow X + \lambda L X$$
$$X \leftarrow X - \mu L X$$

- Attention à bien choisir λ et μ
- Voir T.P.

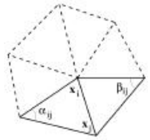


Courtesy Douglas Lanman

Alternative (rappel)

- Inconvénients **Laplacien à poids uniformes** :
  - Pas de prise en compte de la **géométrie locale**
  - Problème si triangles de tailles/angles très différents
- Laplacien avec **poids cotangents** [Desbrun et al. 1999] :

$$\triangle x_i = \frac{1}{2A_i} \sum_{v_j \in N(v_i)} (cotan\alpha_{ij} + cotan\beta_{ij})(x_j - x_i)$$

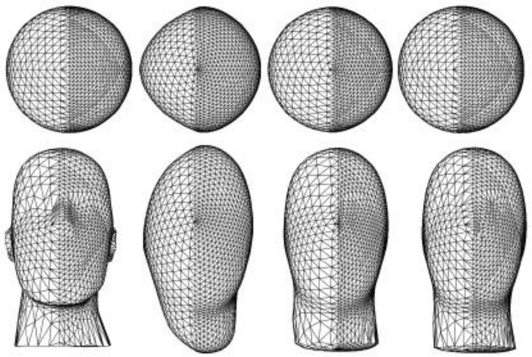


[Meyer et al. 2003]

- Dans ce cas on parle aussi de **flot de courbure moyenne** à cause de la relation

$$\triangle \mathbf{x} = -2H\mathbf{n}$$

Comparaison



[Desbrun et al. 1999]

Maillage initial — Poids uniformes — [Fujiwara 1995] — Poids cotangents

Démo

`https://geometrycollective.github.io/geometry-processing-js/projects/geometric-flow/index.html`

- Appuyer plusieurs fois sur **Integrate** pour voir le lissage
- Garder un pas de temps faible pour voir l'évolution
- Tester différents modèles

Approche spectrale

- Problème de l'approche précédente : coupe brutalement les hautes fréquences
  - Difficile de distinguer les **détails** géométriques du **bruit** : tout est filtré
- Observation : les fonctions de base de la **transformée de Fourier** sont des **fonctions propres du Laplacien**

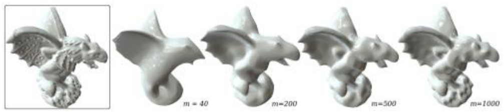
$$1D \ : \triangle(e^{2i\pi\omega x}) = \frac{d^2}{dx^2} e^{2i\pi\omega x} = -(2\pi\omega)^2 e^{2i\pi\omega x}$$

- $\Rightarrow$ les vecteurs propres du Laplacien discret correspondent aux **vibrations naturelles** du maillage et ses valeurs propres aux **fréquences naturelles**

[Vallet et Lévy 2008]

Lissage spectral

- Algorithme :
  - Calculer la **décomposition spectrale** du Laplacien
  - Ne garder que les vecteurs propres correspondant aux m **plus petites valeurs propres** (basses fréquences)
  - Revenir à l'espace Euclidien



[Vallet et Lévy 2008]

- Remarques :
  - Simples produits matrice-vecteur
  - Inconvénient : assez lent
  - Matrices creuses, utiliser des bibliothèques dédiées

Lissage anisotropique

- **Problème** des approches précédentes : lissage **uniforme** dans toutes les directions
- $\Rightarrow$ les caractéristiques géométriques saillantes (**arêtes vives**) sont arrondies

[Clarenz et al. 2000]

- **Solution** : **pénaliser** le lissage au niveau des arêtes vives
  - Même principe qu'en traitement d'images (arête = changement brusque d'intensité)
  - Définition arêtes vives : maxima locaux **courbures principales**

Filtrage bilatéral

Cas des images 2D [Tomasi and Manduchi 1998] :

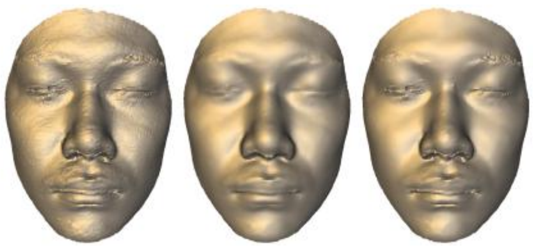
- $I(p_i) \leftarrow \frac{1}{\sum_{j\in N(i)} W_c(\|p_i-p_j\|)W_s(|I(p_i)-I(p_j)|)} \sum_{j\in N(i)} W_c(\|p_i-p_j\|)W_s(|I(p_i)-I(p_j)|)I(p_i)$

  avec $W_c(p)=e^{-p/(2\sigma_c^2)}$ filtre de lissage gaussien et $W_s(I)=e^{-I/(2\sigma_s^2)}$ fonction de similarité en intensité
- W_s pénalise les grandes variations d'intensité
- Avantage : **anisotropique**

Filtrage bilatéral

Extension aux maillages [Fleishman et al. 2003, Jones et al. 2003] :

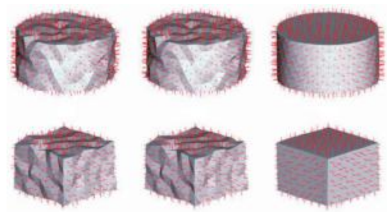
- Déplacement dans la direction de la normale n (connue) : $x_i \leftarrow x_i + dn$
- Approximation locale de la surface lisse en x_i : plan passant par x_i et de normale n
- $\Rightarrow$ Remplacer l'intensité I par le produit scalaire entre n et $x_i - x_j$



[Fleishman et al. 2003]

Lissage des normales

- Idée : lisser le **champ de normales** puis en déduire les positions des sommets par approximation
- Exemple : [Shen and Barner 2004]
  - Itérer k fois $x_i \leftarrow x_i + \pi \sum_{j\in N_1(i)} \sum_{f\in F_{ij}} n_f(n_f^T(x_i - x_j))$ avec F_{ij} l'ensemble des deux faces incidentes à l'arête x_ix_j, n_f la normale lissée à la face f et n_f^T la transposée de n_f.



[Shen and Barner 2004]

Lissage : bilan

- Théorie du **traitement du signal géométrique**
- Bruit = hautes **fréquences** géométriques
- Filtrage : construction d'un **Laplacien**
- Extensions : préservation des **arêtes vives** et des **détails** géométriques

Toujours un sujet d'actualité

- [Wei et al. TVCG 2015]
- [Zhang. et al. TVCG 2015]
- [Lu et al. TVCG 2016]
- [Wang et al. SIGGRAPH Asia 2016]
- [Lu et al. CAGD 2017]
- [Centin et Signoroni TVCG 2017]
- ...

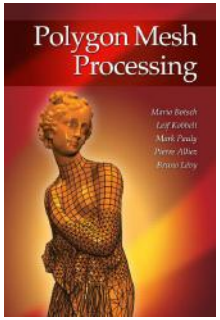
Simplification : contexte

- Maillages obtenus par numérisation souvent **gigantesques**
  - Des millions de sommets/faces
- Problèmes : **calculs** et **visualisation** lents
- Objectif : **réduire** le nombre de sommets et de faces

Statue "Bimba" (CNR, Italie)
3.75 millions triangles

Références bibliographiques

- Chapitre 7 du livre "Polygon Mesh Processing" : `http://www.pmp-book.org/`
- Chapitre 11 du livre "Guide to Computational Geometry Processing" : `https://books.google.fr/books?id=0lb4_pLIyP8C`



Simplification : trois grandes approches

1. **Rééchantillonnage** (resampling)
2. Regroupement (**clustering**) de sommets
   - Les sommets sont regroupés en **régions** (**clusters**)
   - $\sim$ Une face par région
3. Simplification **itérative**
   - Les sommets sont supprimés un à un
   - Faces voisines "fusionnées"

Rééchantillonnage

- **Distribuer** des points sur le maillage puis les **connecter** (nouveau maillage)
- Distribution : aléatoire, uniforme, ...
- Intérêt : contrôler la **connectivité**
  - Semi-régularité : **multirésolution**

(j) Base mesh (162 faces) (k) Original mesh (69,473 faces) (l) LOD using multiresolution approx.

[Eck et al. 1995]

- Inconvénient : aliasing le long des arêtes vives

Regroupement hiérarchique vs. incrémental

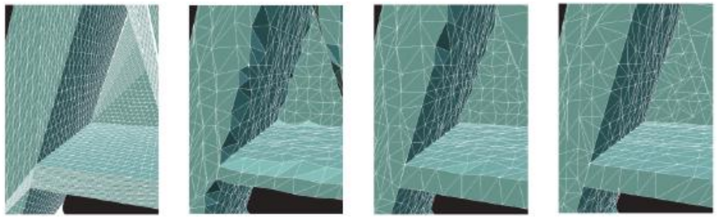
Vertex clustering :

- Les sommets sont regroupés en **régions** (**clusters**)
- Approche **top-down** :
  - **Diviser** récursivement le maillage en régions (ex. : octree)
  - Stop quand taille ou géométrie locale atteint un seuil min
- Approche **bottom-up** : **croissance de régions**
  - **Ajout** itératif de sommets voisins à une région
  - Stop quand taille ou géométrie locale atteint un seuil max
  - Recommencer pour une nouvelle région

UNIVERSITÉ DE STRASBOURG

Etapes suivantes

- Choix d'un **sommet représentatif** par région
  - Critères géométriques (exemple plus loin)



Courtesy M. Botsch et al.

- **Remaillage** par connexion de chaque sommet aux représentants voisins
  - Pour toute arête $v_i v_j$ du maillage initial, créer l'arête $rep(v_i)rep(v_j)$ si $rep(v_i) \neq rep(v_j)$

UNIVERSITÉ DE STRASBOURG

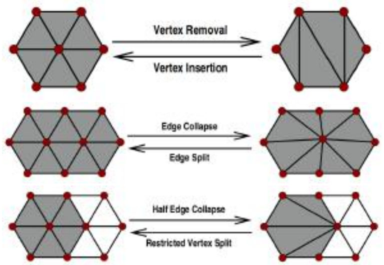
Avantages et inconvénients

- **Avantages** :
  - Très rapide (complexité souvent linéaire)
  - Robuste, peu de cas dégénérés
- **Inconvénient** :
  - Qualité géométrique du résultat (forme des polygones, erreur locale, ...) faible
- Pas de contrôle de la **topologie** (simplification)

UNIVERSITÉ DE STRASBOURG

Simplification itérative

- Les sommets sont **supprimés itérativement**
- Opérations topologiques possibles :
  - **Vertex removal** [Schröder et al. 1992]
  - **Edge collapse** [Hoppe 1996]
  - **Halfedge collapse** [Kobbelt et al. 1998]



Courtesy M. Botsch et al.

UNIVERSITÉ DE STRASBOURG

Questions géométriques

- **Ordre** dans lequel les sommets sont supprimés ?
  - **File de priorité**
- Définition du **coût** d'une opération de simplification et critère d'arrêt ?
- **Remaillage** local ?
  - Edge collapse : cf. représentant dans le cas du clustering

Exemple : métrique d'erreur quadrique

- M. Garland et P. Heckbert, "Surface Simplification using Quadric Error Metrics", SIGGRAPH 1997
- Surface approchée localement en un sommet v par l'**ensemble des plans** portés par les triangles incidents en v
- **Erreur quadrique** en v = somme des distances au carré de v à ces plans

$$e(v) = \sum_{F \in N(v)} d(v, P(F))^2$$

  - Représente la **déviation** de v par rapport à la surface
  - Initialement nulle

Exemple : métrique d'erreur quadrique

- **Forme quadratique** :

$$d(v, P(F))^2 = X_v^T Q_F X_v$$

  - $X_v = (x_v y_v z_v 1)^T$ (coordonnées homogènes), Q_F matrice symétrique 4×4
- Erreur en v représentée par $Q_v = \sum_{F \in N(v)} Q_F$
- Coût d'une fusion $v_1 v_2$ et position du sommet résultant : $\min(Q_{v_1} + Q_{v_2})$

Algorithme de Garland et Heckbert

- **Exercice** : écrire l'algorithme en pseudo-langage, de manière la plus détaillée possible
- Paramètres : maillage + nombre final de sommets ou erreur max d'un sommet

[Garland and Heckbert 1997]

- Vous le coderez en **T.P.**

Simplification itérative

- **Avantages** :
  - Bonne qualité géométrique du résultat
  - Prise en compte de critères variés
- **Inconvénient** :
  - Moins rapide mais raisonnable ($O(n^2)$ pire cas, $O(n \log n)$ moyenne)
- Contrôle de la **topologie** : condition du lien (edge collapse)

Simplification : bilan

- Trois approches principales
- Technique la plus répandue : **edge collapse**
- **Compromis** temps de calcul/qualité
- Extensions :
  - Critères **non géométriques** (texture, dépendance au point de vue)
  - Algorithmes **“out-of-core”** (streaming)

Merci