Rapport de projet 150h

Guillaume BOEHM
Master 2 I3D
Université de Strasbourg

Projet encadré par Pierre Kraemer

January 5, 2020

Contents

Introduction

Dans le cadre de mon projet 150 de master, j'ai été amené à travailler avec Pierre Kraemer sur le projet CGoGN du laboratoire ICube. CGoGN est une librairie C++ de modélisation géométrique utilisant des cartes combinatoires, un puissant modèles de représentation topologique. Cette librairie possède dors et déjà de nombreux outils de parcours et modification des maillages tels que des parcours sur la totalité du maillage par type de cellules, ou des algorithmes de simplification de maillage. CGoGN utilise de la métaprogrammation avec des patrons afin d'autoriser l'utilisation de modèles différents.

Dans ce contexte j'ai du prendre en main la librairie et implémenter l'algorithme de simplification de maillage topstoc[1] en l'adaptant au modèle des cartes combinatoires. L'algorithme topstoc utilise une sélection stochastique des sommets du maillage à conserver basée sur leur courbure locale ainsi qu'un partitionnement topologique pour reconstituer le maillage.

Par la suite je vais développer mon implémentation de l'algorithme dans Section 1, présenter les résultats obtenus dans Section 2 et finir avec mon ressentit du projet dans Section 3.

1 Implémentation

La majorité des fonctions utilitaires de CGoGN sont des fonctions globales rangées dans des espaces de noms, j'ai donc implémenté l'algorithme topstoc dans une fonction `topstoc` sous l'espace de noms `cgogn::modeling`. Le module `SurfaceModeling`, par le biais duquel il est possible d'effectuer différentes opérations de modification de maillage, permet ensuite d'appliquer la fonction `topstoc` à un maillage.

L'algorithme implémenté peut être séparé en trois parties détaillées ci-dessous : la sélection des sommets, le partitionnement topologique puis la reconstruction du maillage.

1.1 Sélection des sommets

Ma première implémentation de la sélection des sommets me servait de bouchon et ne faisait que sélectionner les k premiers sommets stockés dans le maillage, avec k le nombre de sommets voulus. Par la suite j'ai implémenté une variante de la méthode amenée dans [1].
À chaque sommet v du maillage est associé une valeur caractéristique x_v pour guider la sélection, en l'occurrence les information de courbure autour du sommet :

$$x_v = \frac{\sum_{u \in N_v} \sigma(\mathbf{n}_v^{\mathrm{T}} \mathbf{n}_u)}{|N_v|}$$

Avec N_v les sommets du 1-ring de v, $\mathbf{n}_v$ le vecteur normal de v et σ une fonction décroissante monotone sur $[-1,1] \to [0,1]$, pour laquelle j'ai utilisé la fonction conseillée dans l'article :

$$\sigma(d) = (1-d)/2$$

Ces valeurs caractéristiques sont ensuite utilisées dans une fonction de distribution $\mathscr{P}$ afin de conserver le sommet v si $r < \mathscr{P}(x_v)$, avec r une variable aléatoire continue sur $[0,1]$.
Afin d'éviter de laisser les sommets de courbure nulle à une probabilité de 0 et se retrouver avec un sous-échantillonnage des zones plates, la formule suivante est utilisée pour la fonction de distribution :

$$\mathscr{P}(x) = \frac{\left(1 + \alpha(\frac{x}{\{\bar{x_v}\}} - 1)\right)}{X_v}$$

Avec α un coefficient d'adaptabilité, pour mon projet j'ai utilisé $\alpha = 2/3$, $\{\bar{x_v}\}$ la moyenne des x_v de V, et X_v la plus grande valeur de x_v sur V.
Cette formule s'éloigne légèrement de la formule proposée dans [1] :

$$\mathscr{P}(x) = k\left(1 + \alpha(\frac{x}{\{\bar{x_v}\}} - 1)\right)$$

Qui ne résulte pas en une fonction $\mathscr{P} : [0,1] \rightarrow [0,1]$.

On se retrouve donc avec $\tilde{k} \approx k$ sommets à conserver pour le nouveau maillage, la prochaine étape est de décider comment relier les sommets sélectionnés.

1.2 Partitionnement topologique

Une fois les sommets sélectionnés il faut trouver les triangles qui permettront de conserver la topologie de l'objet, pour cela

1.3 Reconstruction du maillage

2 Résultats

3 Conclusions

References

[1] T. Boubekeur and M. Alexa. Mesh simplification by stochastic sampling and topological clustering. *Computer and Graphics – Special Issue on IEEE Shape Modeling International 2009*, 33:241–249, 2009.