

Projet de Modélisation Géométrique : Modified Butterfly Subdivision

Jed Moutahir, Brenda Tonleu, Steven Zheng

May 2023

1 Introduction

Dans le cadre de ce projet, nous avons implémenté l'algorithme "Modified Butterfly Subdivision" pour la représentation de surfaces de subdivision. Ce modèle diffère des méthodes vues précédemment lors des travaux pratiques (TP) tels que l'interpolation polynomiale, les courbes de Bézier, les splines et les B-splines. L'objectif de ce rapport est de présenter succinctement le modèle Modified Butterfly Subdivision, d'en discuter les avantages et les limitations, d'expliquer notre algorithme d'implémentation, de comparer ce modèle aux autres méthodes de subdivision étudiées dans le contexte du sujet, et de conclure en proposant des perspectives pour de futures améliorations.

2 Motivations

Le modèle de surface de subdivision que nous avons choisi d'implémenter dans ce projet est le Modified Butterfly Subdivision. Cette méthode présente des caractéristiques distinctes par rapport aux modèles de subdivision vus lors des travaux pratiques précédents, tels que la subdivision de Loop, Doo-Sabin et Catmull-Clark.

L'algorithme Modified Butterfly Subdivision se base sur la subdivision itérative d'un maillage initial pour obtenir des surfaces plus lisses et mieux définies. Contrairement à la subdivision de Loop qui se concentre sur la régularité du maillage et à Doo-Sabin et Catmull-Clark qui utilisent des produits tensoriels, Modified Butterfly Subdivision se distingue par sa capacité à traiter des mailles irréguliers et à gérer des coins ou des arêtes saillantes.

L'intérêt principal de Modified Butterfly Subdivision réside dans sa capacité à préserver les détails et les caractéristiques du maillage initial tout en améliorant sa régularité. Cela en fait un modèle particulièrement adapté pour la représentation de surfaces complexes avec des zones de déformation localisées.

3 Explications

L'algorithme commence par parcourir chaque face de la surface. Pour chaque arête de la face, le milieu de l'arête est calculé comme un barycentre des points autour de lui. Les coefficients du barycentre sont déterminés en utilisant les poids spécifiques du modèle Butterfly, avec un coefficient optimal prédéfini de $w = 1/16$.

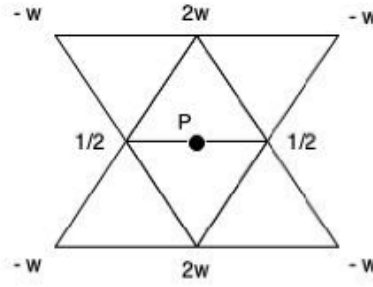


Figure 1: Cas général en forme de papillon

Pour les arêtes situées sur le bord de la surface, les coefficients du barycentre dépendent de la position des points voisins. Les nouveaux points situés sur ces arêtes deviennent également des barycentres, mais les coefficients sont déterminés en fonction de la configuration spécifique des points environnants.

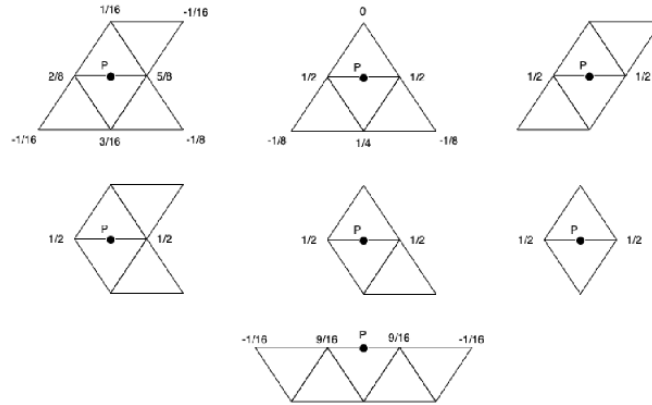


Figure 2: Cas général en forme de papillon

L'algorithme itère ensuite sur l'ensemble des faces de la surface, en calculant les nouveaux points pour chaque face. Les nouveaux éléments, tels que les arêtes

et les faces, sont intégrés à la structure de données de la surface. L'algorithme affiche la nouvelle figure obtenue une fois que toutes les faces ont été parcourues.

Pour optimiser le parcours des éléments de la surface, des listes d'indices sont utilisées pour représenter les points, les arêtes et les faces. Les coordonnées des points sont utilisées lors du calcul des nouveaux points, en utilisant la liste ordonnée des points environnants et les poids associés.

Il est important de noter que le modèle de subdivision Butterfly ne convient pas à tous les types de volumes. Bien qu'il puisse créer des figures arrondies, il ne permet pas d'obtenir une surface lisse comme d'autres méthodes telles que Doo-Sabin, Catmull-Clark ou Loop.

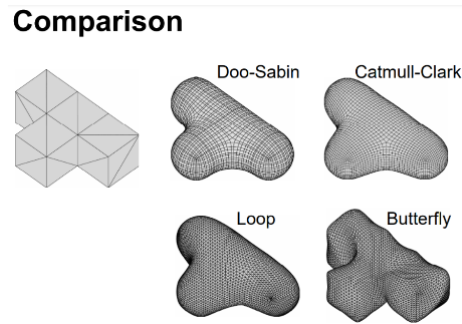


Figure 3: Cas général en forme de papillon

Cependant, le modèle Butterfly est particulièrement adapté à la création de volumes rugueux et irréguliers, tels que des paysages montagneux. Il permet d'obtenir rapidement un résultat satisfaisant avec un faible coût de stockage et une utilisation efficace de la bande passante.

En résumé, l'algorithme de subdivision Butterfly est un choix approprié pour générer des surfaces rugueuses et irrégulières, telles que des paysages, en économisant la mémoire et la bande passante. Cependant, il convient de noter que la performance de l'algorithme peut se dégrader au-delà d'un certain nombre d'itérations en raison de la complexité croissante des calculs et du stockage requis.

4 Exemples

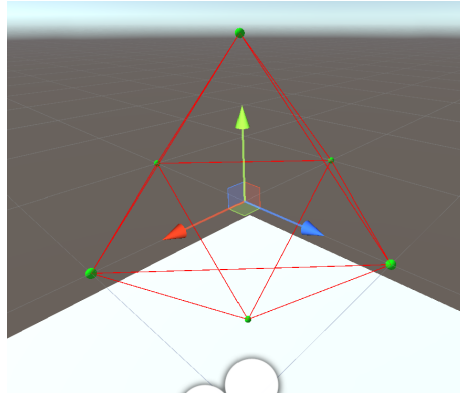


Figure 4: Figure initial

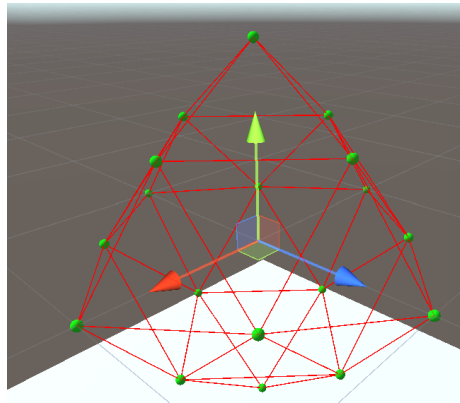


Figure 5: 1ère itération

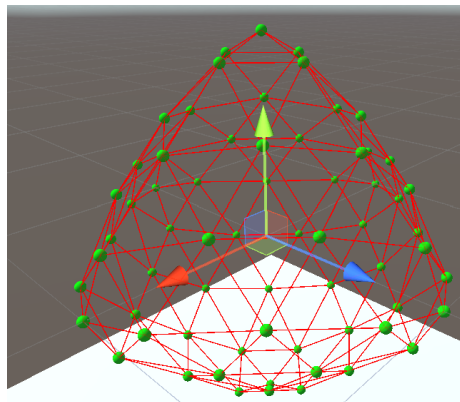


Figure 6: 2ème itération

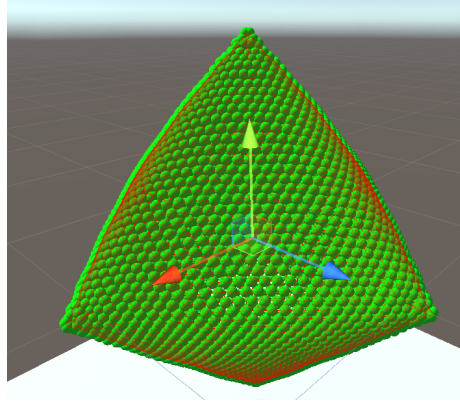


Figure 7: 5ème itération

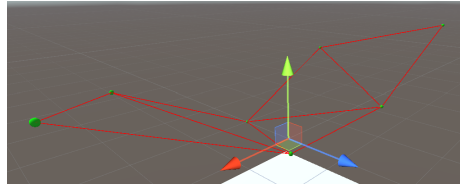


Figure 8: Figure initial

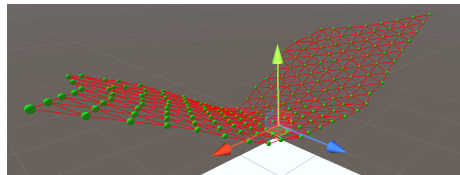


Figure 9: 4ème itération

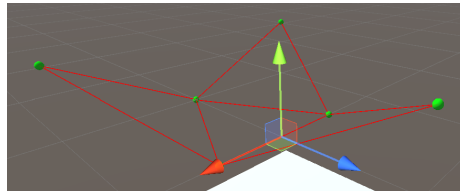


Figure 10: Figure initial

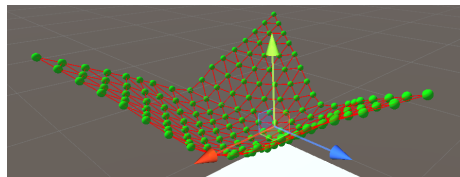


Figure 11: 4ème itération