# Géométrie Algorithmique : Segmentation de maillages

## Objet du TP

L’objectif de ce travail était d’implémenter en C++, au moyen de la bibliothèque CGAL, différents algorithmes de traitement sur les maillages afin de les découper en plusieurs régions, en se basant sur la similarité des éléments qui les composent. Le calcul de propriétés locales permet de savoir à quel point deux faces sont similaires. Afin de séparer les régions, on travaille par seuillage sur les propriétés considérées, cette étape est nommée segmentation. Une fois le maillage segmenté en régions, l’objectif était d’appliquer une seconde segmentation afin de distinguer les régions qui ont des propriétés locales similaires mais qui ne sont pas connexes.

Dans ce rapport, nous détaillerons les différentes fonctions implémentées, ainsi que les résultats qu’elles ont fournis avant de discuter de l’efficacité des différentes stratégies.

## Fonctions implémentées

### Calcul de propriétés locales

La première étape du travail consistait donc à calculer des propriétés locales sur chaque face afin de pouvoir ensuite les comparer. Nous avons donc implémenté le calcul des propriétés suivantes :

* **Périmètre de la face**
* **Minimum des angles composant la face**
* **Angle entre la normale à la face et un vecteur donné**
* **Aire de la face**
* **Angle moyen entre la normale à la face et les normales des faces voisines**

### Segmentation à partir des propriétés locales

Une fois les fonctions ci-dessus implémentées, nous avons pu calculer les propriétés locales de chacune des faces d’un maillage. Il fallait ensuite regrouper ces faces en régions en les séparant à l’aide d’un ou plusieurs seuils. Pour cela, nous avons implémenté les fonctions de seuillage suivantes :

* **Seuillage simple par la moyenne :** on calcule la moyenne d’une des propriétés pour l’ensemble des faces, puis on utilise cette moyenne comme seuil pour séparer les faces en deux classes.
* **Seuillage multiple pour obtenir *n* classes :** On ordonne les valeurs de la propriété considérée pour toutes les faces, puis on subdivise cet ensemble en n groupes à l’aide des n-1 quantiles de ce jeu de données.
* **Seuillage par la méthode d’Otsu :** on utilise l’algorithme disponible sur <https://fr.wikipedia.org/wiki/Méthode_d’Otsu> afin de trouver le seuil qui maximise la variance inter-classes. Pour cela on commence par calculer l’histogramme de la propriété locale considérée pour toutes les faces, puis on calcule la variance inter-classes pour toutes les valeurs de l’histogramme, soit tous les seuils possibles. On conserve ensuite celui qui a donné la variance maximale.

### Segmentation par composantes connexes

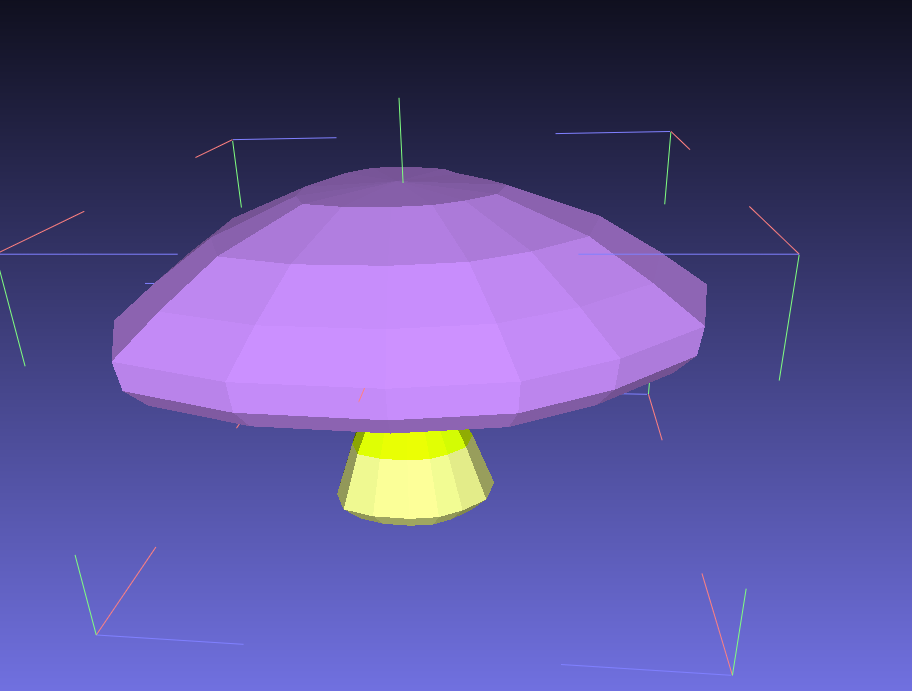
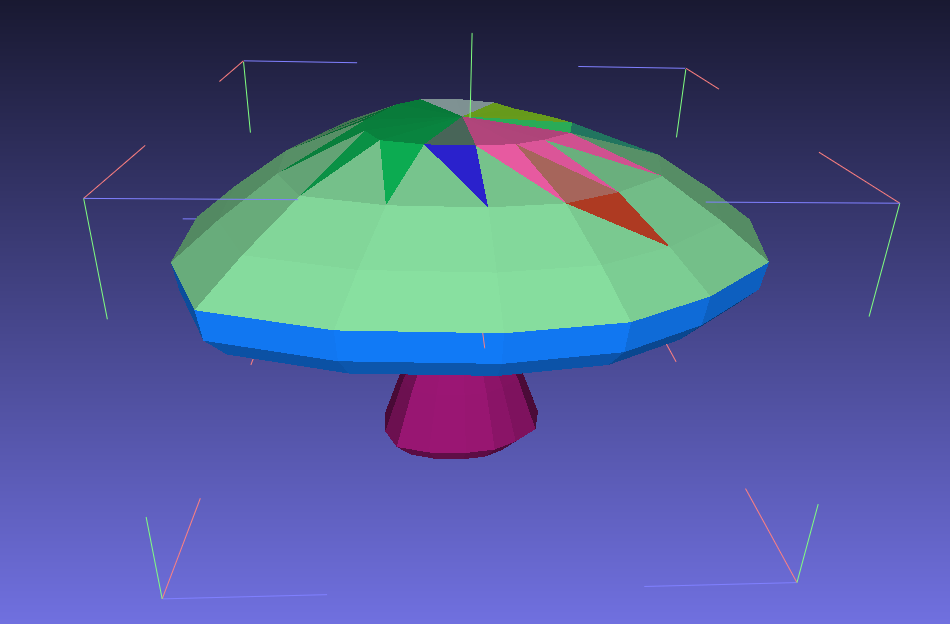
Après avoir obtenu une première segmentation par les propriétés locales, il a ensuite fallu séparer les classes qui ne sont pas connexes, c’est-à-dire qui n’ont pas au minimum une arête commune. Pour cela, on applique un parcours (en profondeur) à partir d’une des faces du maillage afin de visiter toutes les faces voisines ayant la même classe que la nôtre, toutes les faces ainsi parcourues sont alors ressemblées dans une nouvelle classe. On recommence un nouveau parcours depuis une autre face (encore non traitée) tant que toutes les faces n’ont pas été intégrées à une nouvelle classe.

## Analyse des résultats

### Tests de fonctionnement

Chacune des fonctions a été testée afin de vérifier son bon fonctionnement, que ce soit pour le calcul des propriétés, pour les différents seuillages ou pour la segmentation par composantes connexes. On se propose d’étudier quelques cas de tests permettant de mettre en évidence le bon fonctionnement des fonctions et d’avoir un regard critique sur les différentes alternatives.

### Exemple 1 : Comparaison de deux propriétés locales

**Périmètre + seuil d’Otsu :** **Aire de la face + seuil d’Otsu :**

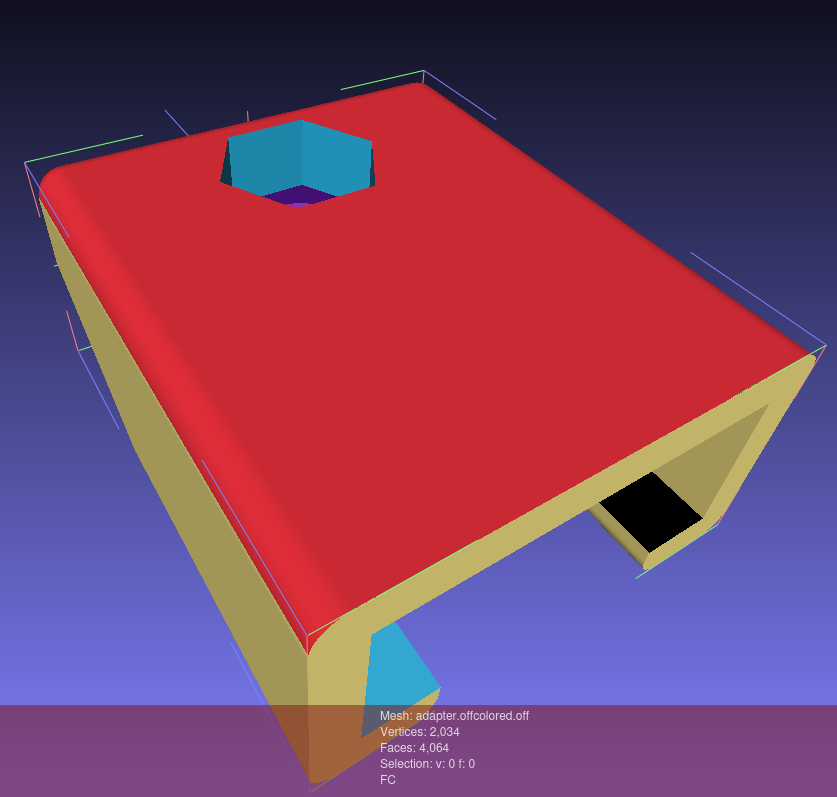
Cet exemple a été réalisé à partir du maillage mushroom.off, sur lequel on a mesuré le périmètre des faces (à gauche) et l’aire des faces (à droite). On a ensuite appliqué le seuillage d’Otsu puis une segmentation par composantes connexes. Dans les deux cas, le pied et le chapeau sont bien distingués, car ils ont à la fois un périmètre et une aire très différents. Dans le cas de l’utilisation de l’aire comme propriété locale, le haut du chapeau est classé différemment du reste du chapeau car les faces de cette zone ont une aire plus faible. Ces deux images comportent plus de deux classes différentes, ce qui montre que la segmentation par composantes connexes a bien fonctionné, car la méthode d’Otsu que nous avons implémenté ne découpe le maillage qu’en deux classes. On peut également observer sur l’image de droite que les composantes connexes sont bien identifiées à partir d’une arête en commun car un point ne suffit pas à considérer deux faces comme voisines. Par exemple, le triangle marron entouré de triangles roses et d’un triangle rouge ne prend pas la couleur vert clair, car il n’a aucune arête commune avec la région verte, bien qu’il ait deux points communs avec celle-ci.

### Exemple 2 : Angle minimum de la face + seuil d’Otsu :

### 

Sur cette image, on a utilisé l’angle minimum d’une face comme propriété locale discriminante. On peut voir que les parties allongées du maillage, qui ont des petits angles internes appartiennent toutes à une même classe, et que les deux extrémités qui ont de plus grands angles internes appartiennent à des classes différentes.

### Exemple 3 : Angle entre la normale de la face et un vecteur donné + seuil d’Otsu :



L’angle entre la normale à la face et un vecteur, ici (0, 1, 0), permet de distinguer l’orientation des faces. Par exemple, au niveau de la forme hexagonale, on a la face du dessus (horizontale) qui est rouge, les faces intérieures (verticales) qui sont bleues, et le fond (horizontal) qui est bleu foncé.

### Exemple 4 : Comparaison entre la méthode d’Otsu et le seuillage simple par la moyenne

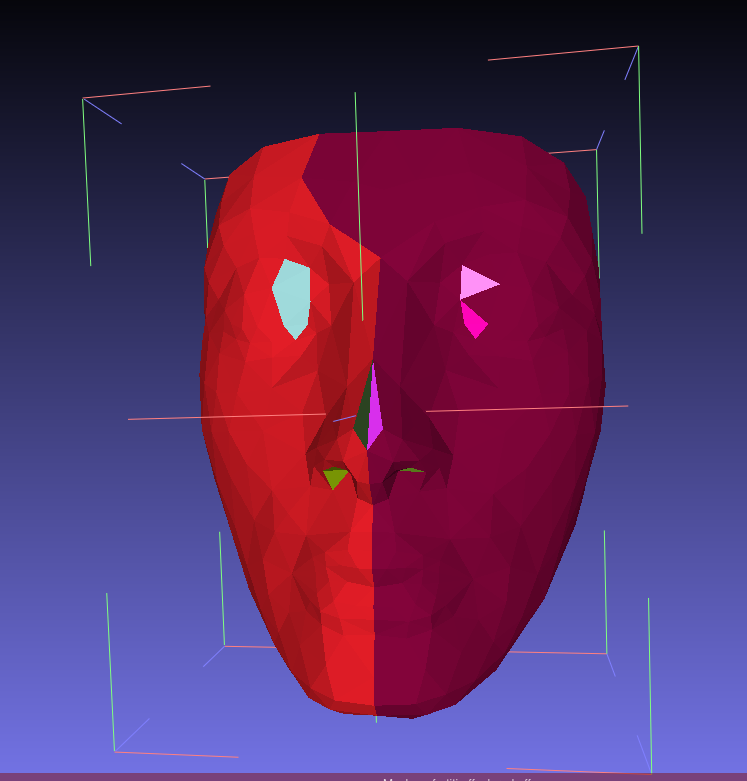
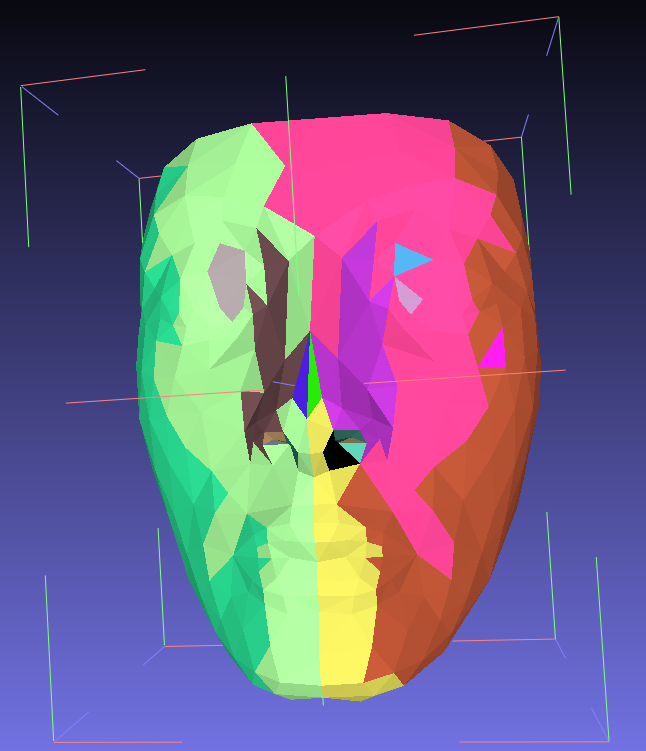
L’image ci-dessous à gauche a été obtenue en utilisant comme propriété locale l’angle moyen entre la normale de la face et les normales des faces voisines, puis une segmentation à l’aide d’un seuillage par la méthode d’Otsu.

L’image ci-dessous à droite a été obtenue en utilisant une segmentation par seuillage simple par la moyenne au lieu de la méthode d’Otsu.

On peut voir que les classes faces et arêtes sont bien délimités avec la méthode d’Otsu, alors qu’avec le seuillage simple par la moyenne une face trop courbée est découpée en plusieurs régions (voir partie droite de l’objet dans l’image).

### Exemple 5 : Seuillage multiple pour obtenir *n* classes

Nous avons enfin testé notre seuillage multiple (qui permet d’obtenir n classes après segmentation, image de droite) face à un seuillage simple (par méthode d’Otsu, image de gauche) et on peut bien voir que le nombre régions majoritaires passe de 2 à 4 entre l’image de gauche et l’image de droite :

La propriété locale ici utilisée était l’angle de la normale à la face par rapport au vecteur (1,0,0).

### Conclusion

Ces différents essais mettent en évidence qu’il n’existe pas une mesure locale polyvalente que l’on pourrait utiliser efficacement sur n’importe quel maillage, mais plutôt que chaque mesure locale couvre des cas différents. Par exemple, la segmentation en utilisant l’angle de la normale d’une face à un vecteur est plutôt bien adaptée au cas de l’exemple 3 si l’on souhaite extraire selon leur différence d’orientation, mais n’est pas très pertinente dans le cas de l’exemple 5 (image de gauche) car elle finit par couper le masque en deux. En définitive, tout dépend de ce que l’on souhaite extraire du maillage.