

Rapport Structure d'Octree

I - Création d'octree

Un octree est un arbre composé de nœuds et de feuilles. Chaque nœud aura 8 fils qui seront des nœuds ou des feuilles.

Pour créer l'octree il faut commencer par créer la boîte englobante. A partir de cette dernière, nous pouvons la diviser en 8 cubes. Récursivement, nous pouvons traiter chaque cube.

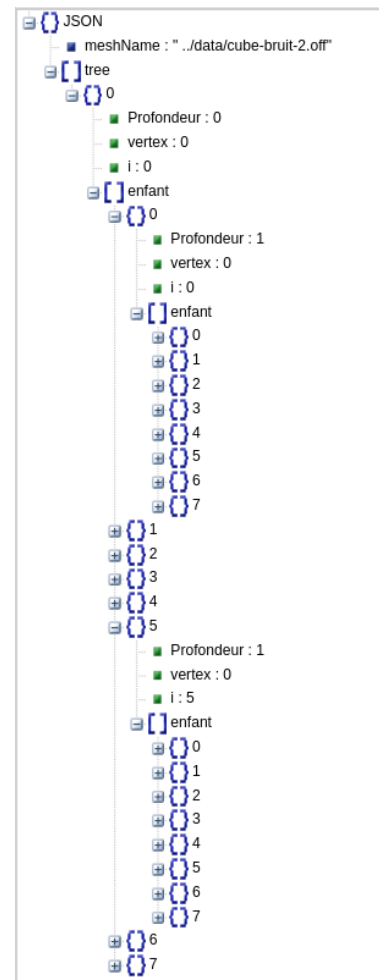
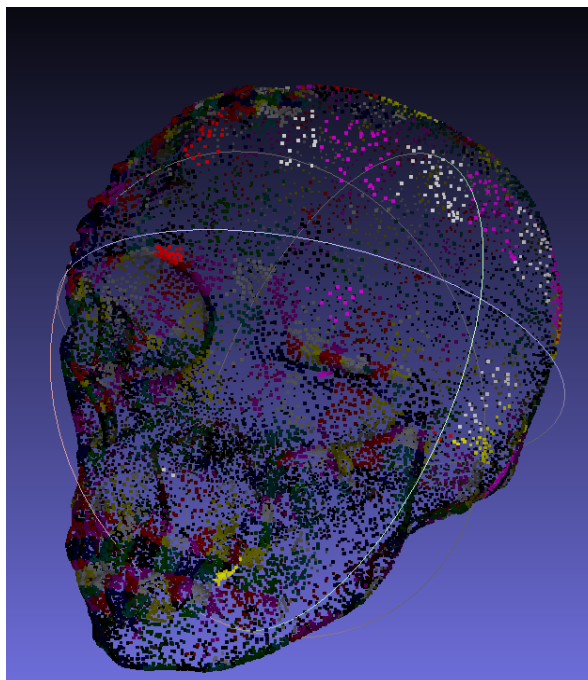
Pour définir si le nœud a des fils ou pas, on définit deux règles : un fils est une feuille si le cube englobe moins d'un certain nombre de points. L'autre règle est définie par la profondeur de l'octree. En effet, on parcourt l'arbre sur un certain nombre d'étages. Une fois la profondeur maximale atteinte, tous les nœuds deviennent des feuilles et le parcours récursif s'arrête.

II - Visualisation d'octree

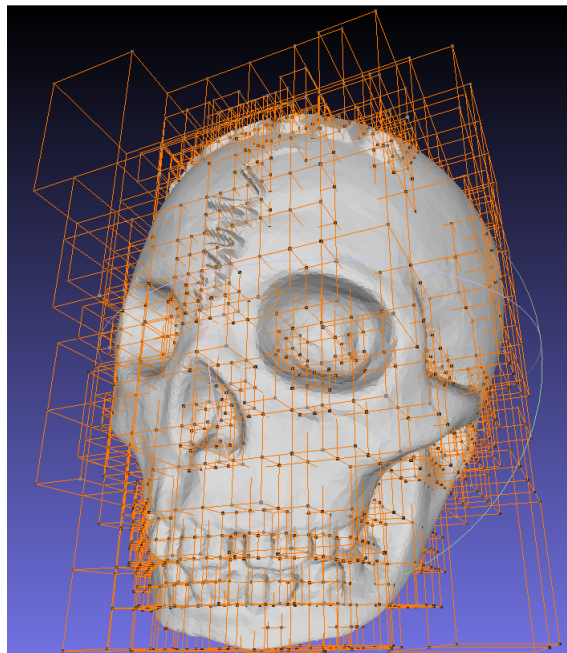
Jusqu'à maintenant, l'octree est quelque chose d'abstrait. Le but est de pouvoir visualiser l'arbre.

Pour se faire, nous avons commencé par écrire un fichier JSON. Pour chaque nœud, on donne la profondeur à laquelle il se trouve, le nombre de vertex qu'il englobe si jamais c'est une feuille sinon il est mis à 0, puis les enfants qu'il a ainsi que son rang dans les enfants qu'il compose. A droite est visible un exemple de JSON qu'on a pu obtenir avec le cube bruité.

Ceci étant fait, nous pouvons voir l'arbre mais ce n'est pas facile de faire la correspondance avec l'objet en lui-même. Il est donc préférable de donner des couleurs aux points en fonction du nœud auquel ils appartiennent comme nous avons pu le faire dans le TP précédent. La partie haute du Skull est plus visible sur le screen mais en 3D cela est plus évident à comprendre.



Il est bien plus lisible avec des couleurs mais nous ne pouvons pas distinguer les cubes. Ce serait beaucoup plus visuel d'avoir directement les cubes englobants.

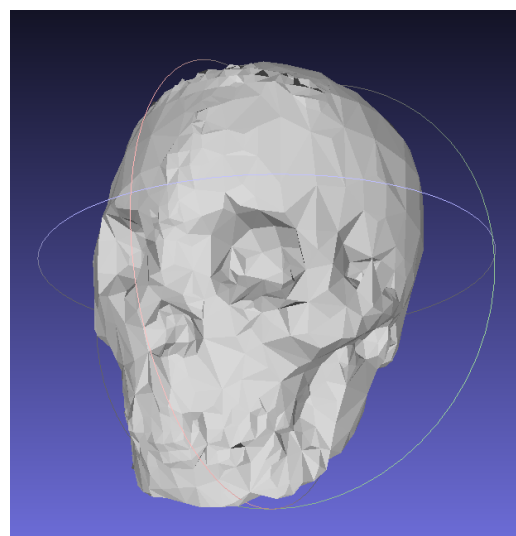
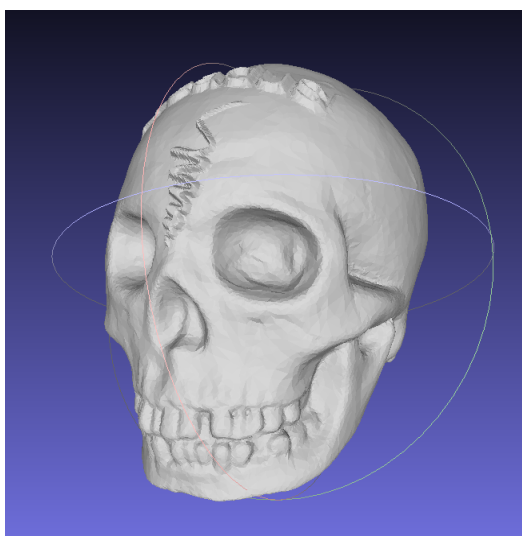


On peut s'apercevoir qu'il y a différentes tailles de cubes ce qui correspond aux différentes profondeurs pour les nodes qui sont des feuilles.

III - Simplification de maillage

Certains maillages ont beaucoup de points, on essaye donc de simplifier le maillage en diminuant le nombre de points. Le but est de fusionner les points qui appartiennent à la même feuille de l'octree. Ainsi nous ferons de nouvelles faces grâce aux nouveaux points. Si les faces se retrouvent avec un ou deux points, alors nous supprimerons ces "faces".

Dans un premier temps, on utilise une fonction afin de créer le point milieu de chaque feuille. Ensuite, on regarde pour chaque point s'il est lié avec un point dans une autre node. Si c'est le cas alors on lie les points milieux des nodes de ces deux points. Après avoir fait cela pour tous les points, on choisit de faire des faces de 3 points ou 4 points. Si certains points ne font pas partie d'une face, on supprime ces derniers afin de garder que les points utiles.



IV - Analyse de la simplification

La simplification est bien visible sur le skull, cependant cela n'est pas optimal. On peut apercevoir des trous dans le skull. Le problème est potentiellement lié au moment où l'on cherche à faire les faces.

On peut comparer les simplifications faites sur le skull avec différents paramètres de capacité et de profondeur.

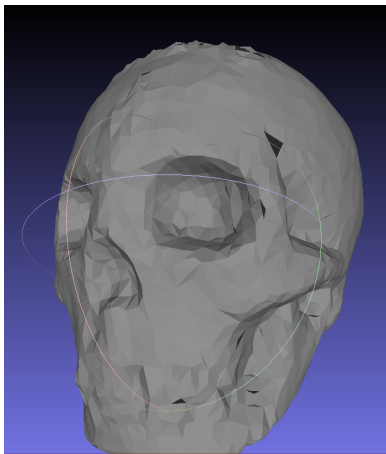


Image n°1
Capacité : 10
Profondeur : 5

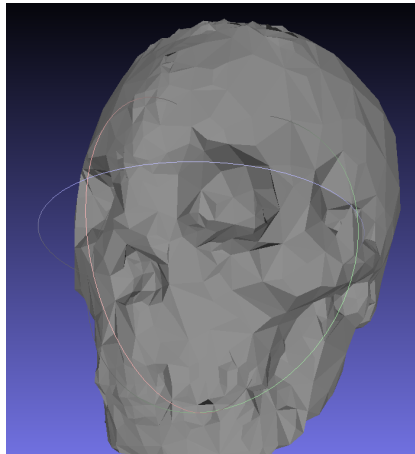


Image n°2
Capacité : 50
Profondeur : 5

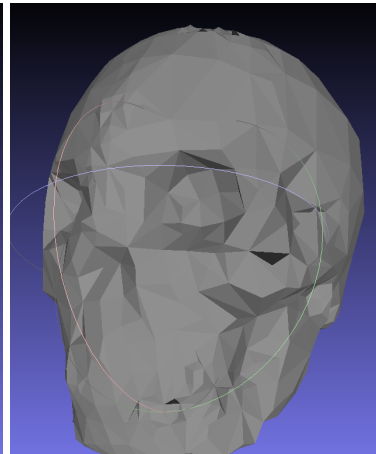


Image n°3
Capacité : 100
Profondeur : 5

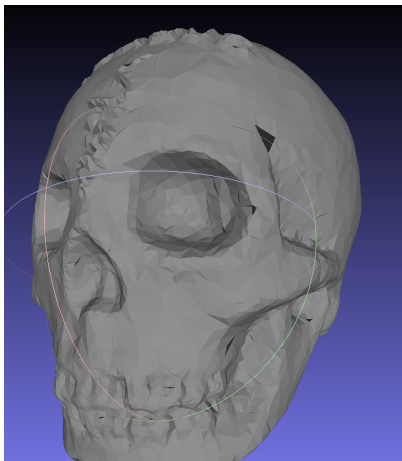


Image n°4
Capacité : 100
Profondeur : 10

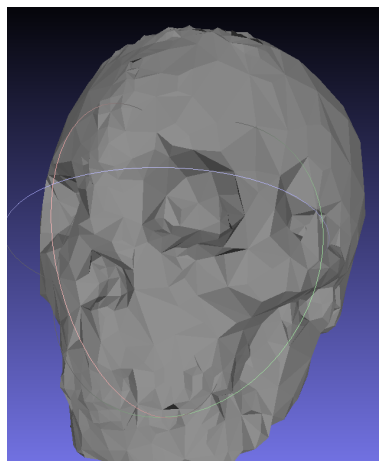


Image n°5
Capacité : 10
Profondeur : 25

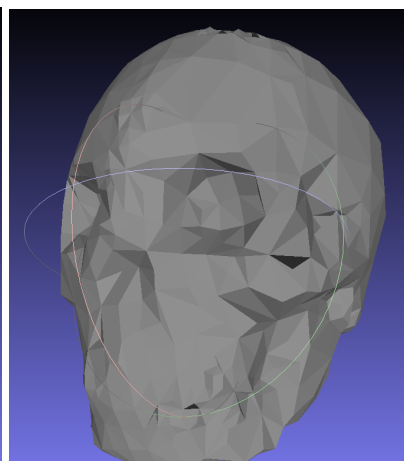
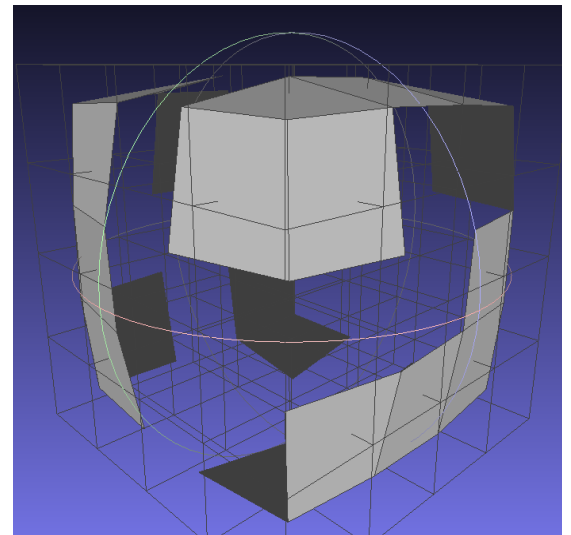
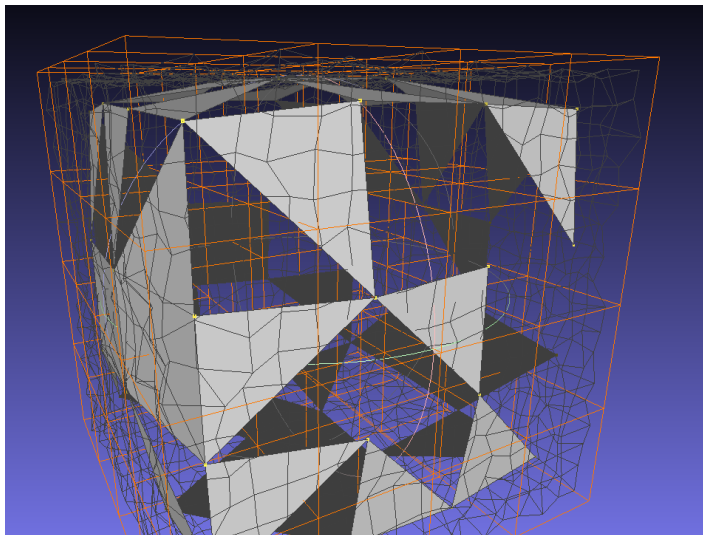


Image n°6
Capacité : 50
Profondeur : 25

On peut s'apercevoir que si on met peu de profondeur ou peu de capacité, le skull va être beaucoup moins bien formé. Il aura moins de points, il sera donc moins lisse. C'est exactement le but de cette partie. Les différents skulls présents ci-dessus montrent bien à quel point le programme simplifie le mesh.

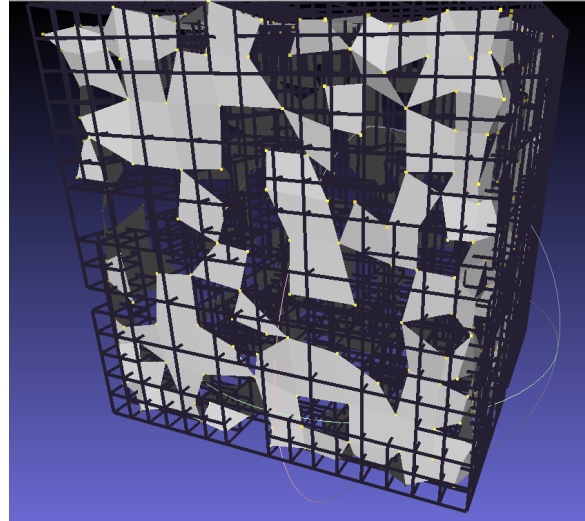
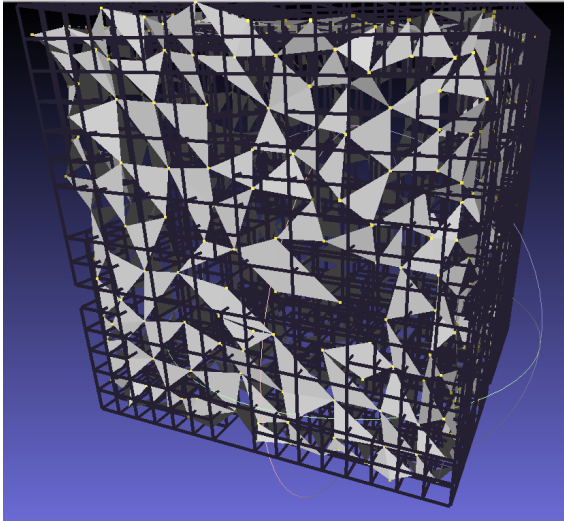
Nous pouvons voir que les changements sont peu visibles entre l'image 2 et 3 car la capacité est trop grande par rapport au nombre de points présents.

Voir que le projet est plutôt fonctionnel est une bonne chose, cependant il faut voir les problèmes qu'on a eu avec la reconstruction du mesh. Il est possible de voir sur les images suivantes le problème de création des faces. On obtient qu'une face sur deux que ce soit en carré ou en triangle.



Il semblerait que les faces soient générées avec les points dans le bon sens afin d'avoir aucune face mal générée. Dans ce cas, l'erreur serait présente au moment de la définition des faces. En effet, si les points n'ont pas de liens directs à travers deux nodes, il se peut que le lien entre les points milieux n'existe pas et donc logiquement la face ne se crée pas. Pour résoudre ce problème, nous n'avons pas eu le temps nécessaire.

Ainsi, nous en avons conclu que le problème venait sûrement des faces qui sont trop petites et qui sont donc incluses dans le node. Ceci entraînerait pas de lien avec le node voisin et donc empêcherait la création de certaines faces pendant la simplification.



Ce problème est tout aussi présent avec plus ou moins de capacité par feuille même si on garde le même paramètre de profondeur.

V - Conclusion

Dans l'ensemble, le TP a été bien avancé mais nous n'avons pas réussi à faire un projet complet et complètement opérationnel. Le problème est au moment de la génération des faces ou de la liaison entre les points milieux. Peut-être qu'il nous manque un bout de code afin de mieux faire les liaisons entre tous les points milieux dont leurs feuilles ont des points en liés.