

# Modélisation géométrique

Université de Bordeaux

January 19, 2016

Gaboulaud Tony  
Mounsamy Yanis

## 1 Introduction

- Objectifs
- Outils utilisés

## 2 D'un nuage de point au maillage 3D

- Visualisation
- Décimation
- Reconstruction surfacique

## 3 Analyse

- Calcul de la valence
- Détection des trous
- Détection de coplanarité

## 4 Réparation

- Comblage de trou naïf avec le centre de gravité
- Comblage de trou avec la technique du ear clipping

## 5 Résultats

## Données en entrées

- Nuage de point
  - Positions
  - Normales

## Données en sorties

- Object prêt pour impression
  - Validité

Langage utilisé:

- C++

Framework utilisés:

- GLM
- GLFW
- CGAL: Surface mesh

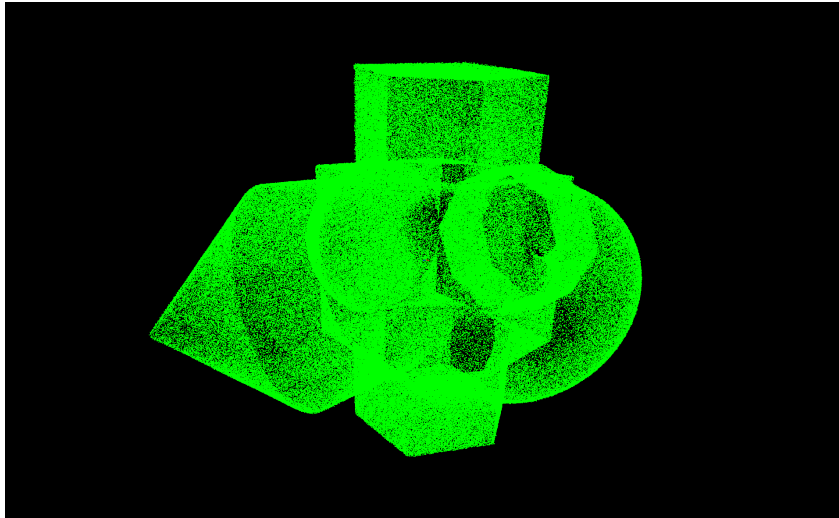


Figure 1: Visualisation du nuage de point

Méthode Octree:

- Récursif
- Condition d'arrêt
  - Nombre minimal de sommets par cellule
  - Profondeur de la récursion

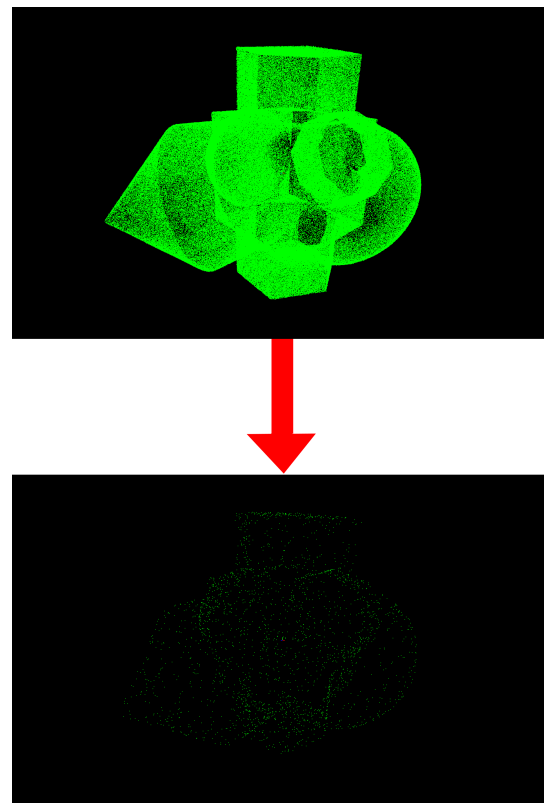


Figure 2: Résultat décimation

## Ball Pivoting Algorithm (BPA) [1]

- Utilisation d'un Octree pour avoir les voisins
- Trouver un premier triangle.
- Expansion de la triangulation en pivotant une sphère autour des arêtes frontières.
- Mise à jour des arêtes frontières.

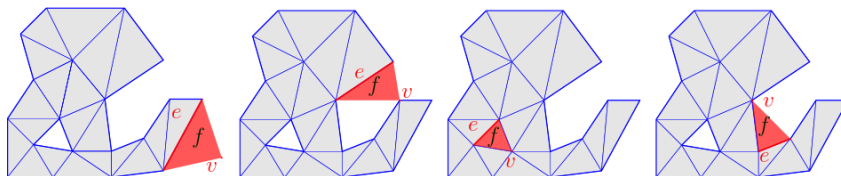


Figure 3: Différents cas d'expansion

- Parcours de tous les vertices
  - Dénombrement des arêtes autour de la vertice
- Attribution d'une couleur pour chaque valeur de valence



# Détection des trous

Utilisation d'une structure de half-edge fourni par SurfaceMesh

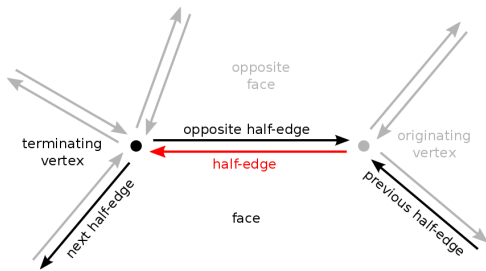


Figure 4:  
Représentation d'une  
structure de half-edge

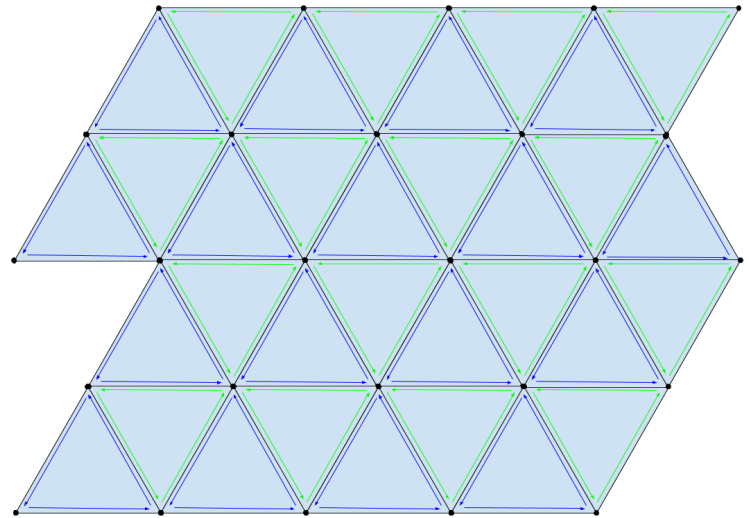


Figure 5: Représentation des half-edges  
sur un maillage triangulaire

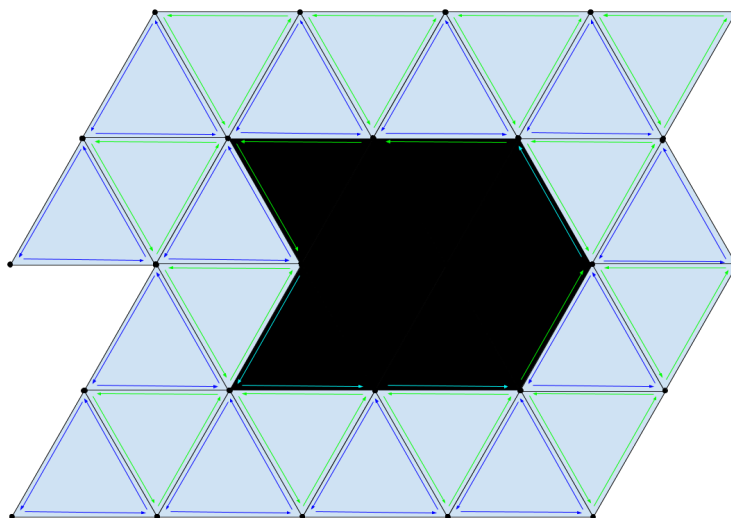


Figure 6: Trou sur un maillage triangulaire

- Chercher un half-edge ne possédant pas de face incidente (bord)
- Visiter son half-edge suivant
- Si c'est un bord, recommencer l'opération jusqu'à retomber sur le vertex de départ

## Comparaison avec MeshLab

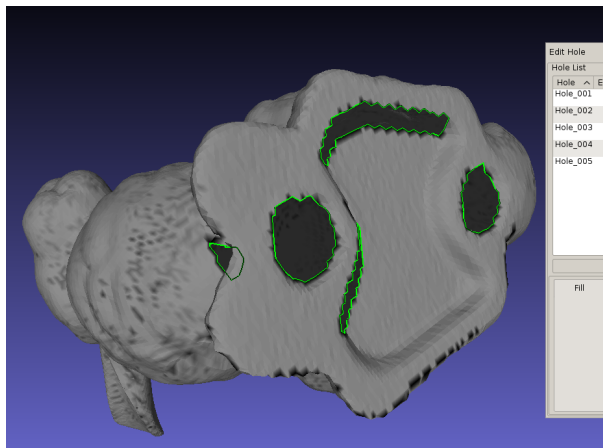


Figure 7: Détection des trous avec MeshLab

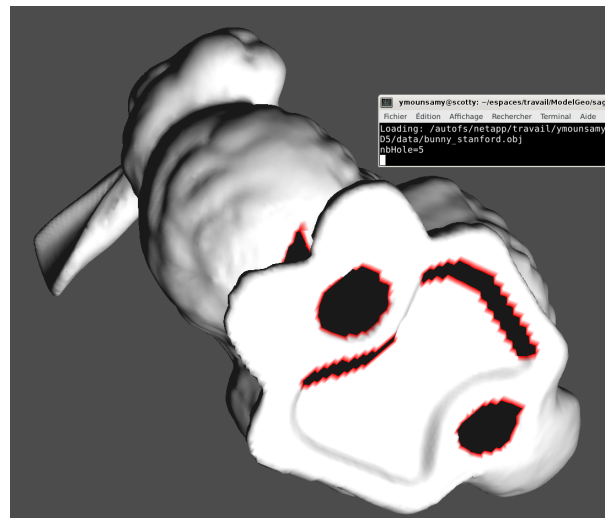


Figure 8: Détection des trous avec notre programme

# Détection de coplanarité [3]

- Prendre trois points consécutifs bordant le trou pour former un triangle
- Tester la coplanarité de chacun des points de part et d'autre du triangle
- Faire la même chose jusqu'à non coplanarité
- Former un trou à partir de l'ensemble de point
- Sélectionner un nouveau triangle
- Faire la même chose jusqu'à ce que le trou soit entièrement divisé en sous trou.

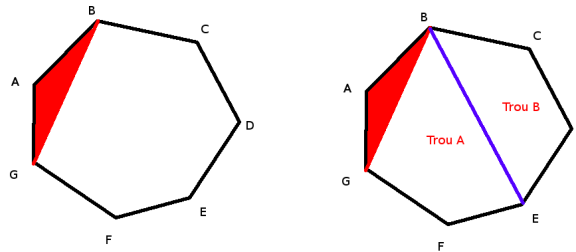


Figure 9: Test de coplanarité et création de "sous-trous"

# Comblage de trou naïf avec le centre de gravité

## Méthode

- Calcul du centre de gravité et sa normale
- Triangulation:
  - On crée une face avec deux sommets consécutifs et le centre de gravité

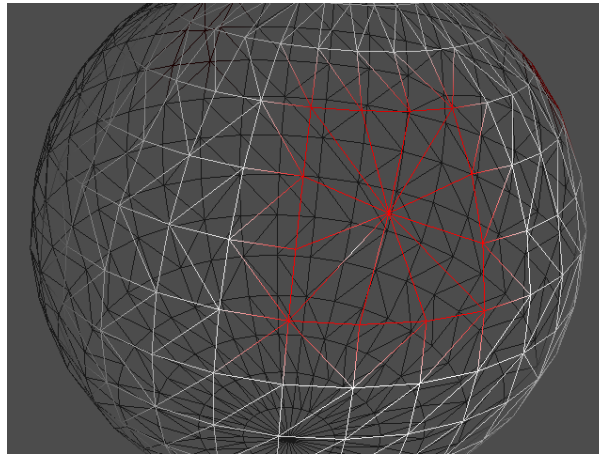


Figure 10: Resultat: Triangulation avec le centre de gravité

## Méthode

Une "oreille" de polygone est un triangle formé par 3 vertices consécutifs  $v_0, v_1, v_2$  tel que  $\widehat{v_0 v_1 v_2} < \pi$ .

- On part d'un vertex  $v_i$ , et on vérifie s'il forme une oreille avec les deux vertices suivants  $v_{i+1}$  et  $v_{i+2}$ 
  - Si oui:
    - On crée la face formée par les trois vertices  $(v_i, v_{i+1}, v_{i+2})$
    - On supprime les arêtes  $v_i v_{i+1}$  et  $v_{i+1} v_{i+2}$  du trou
    - On recommence l'opération en conservant le même sommet de départ
  - Si non:
    - On prend le sommet suivant comme sommet de départ et on répète l'opération.

# Comblage de trou avec la technique du ear clipping

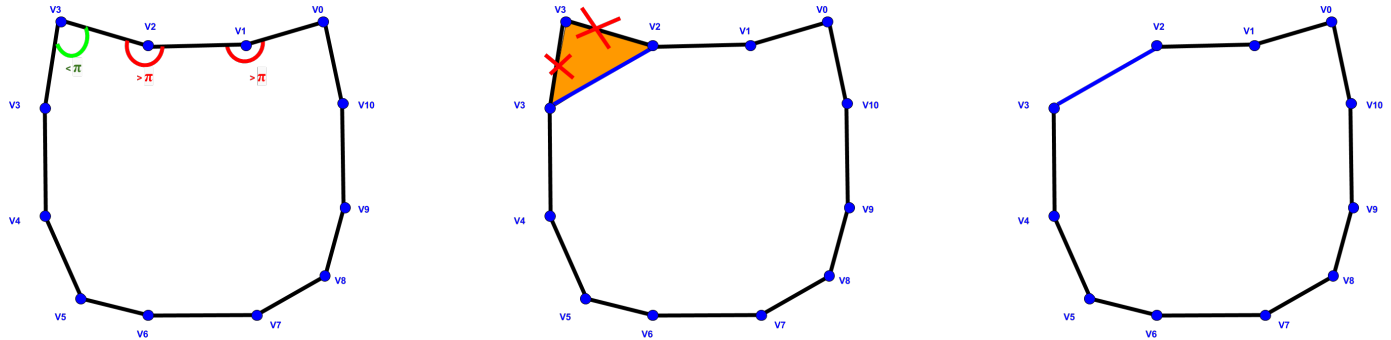


Figure 11: Illustration du ear clipping

# Comblage de trou avec la technique du ear clipping

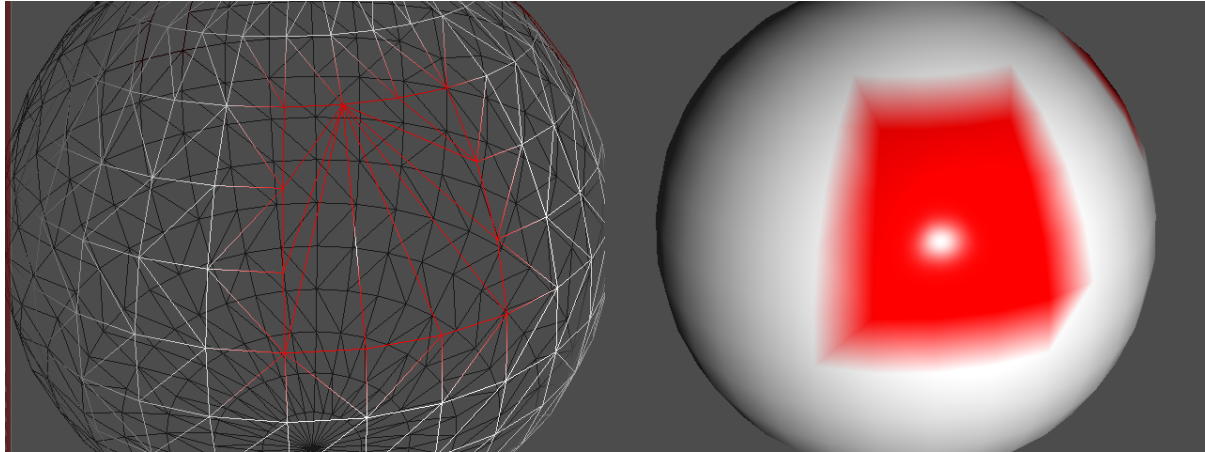


Figure 12: Resultat: Triangulation avec la méthode de ear clipping



## Limitations

- Les angles doivent être correctement orientés.
- Les trous doivent être planaires.

## Solutions possibles

- Déterminer le volume de la boîte englobante du trou, si celui-ci est inférieur à un certain seuil (par exemple 0.1), considérer le trou comme étant planaire
- Utiliser le test de coplanarité.

Résultat de notre implémentation avec kate\_leg.obj

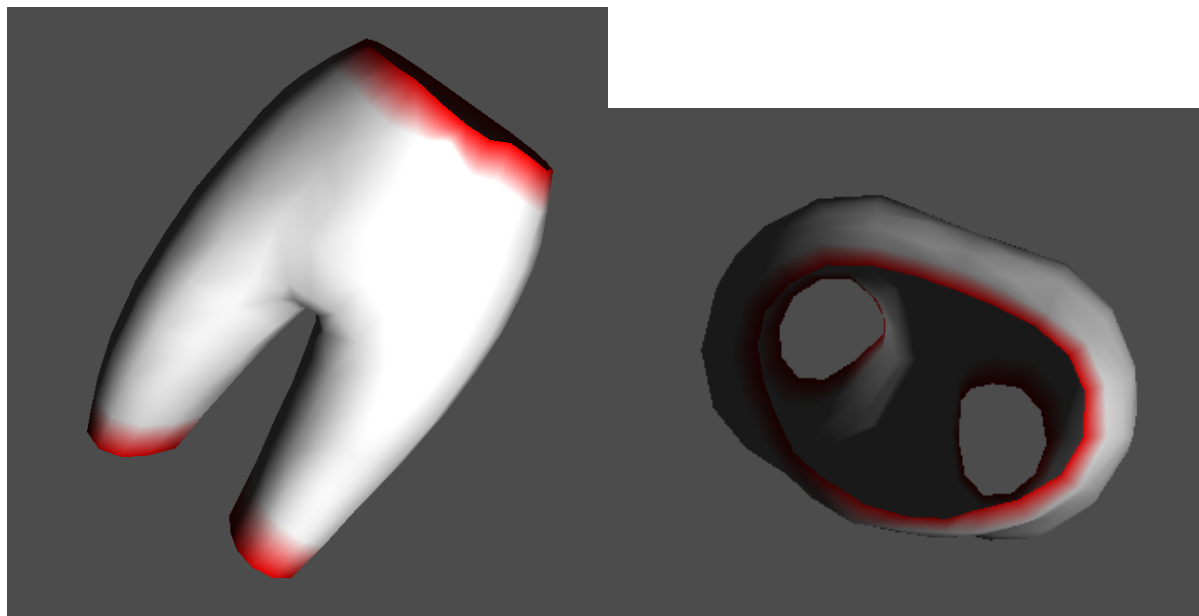


Figure 13: Resultat: Détection des trous : 3 trous détecté

Résultat de notre implémentation avec `kate_leg.obj`

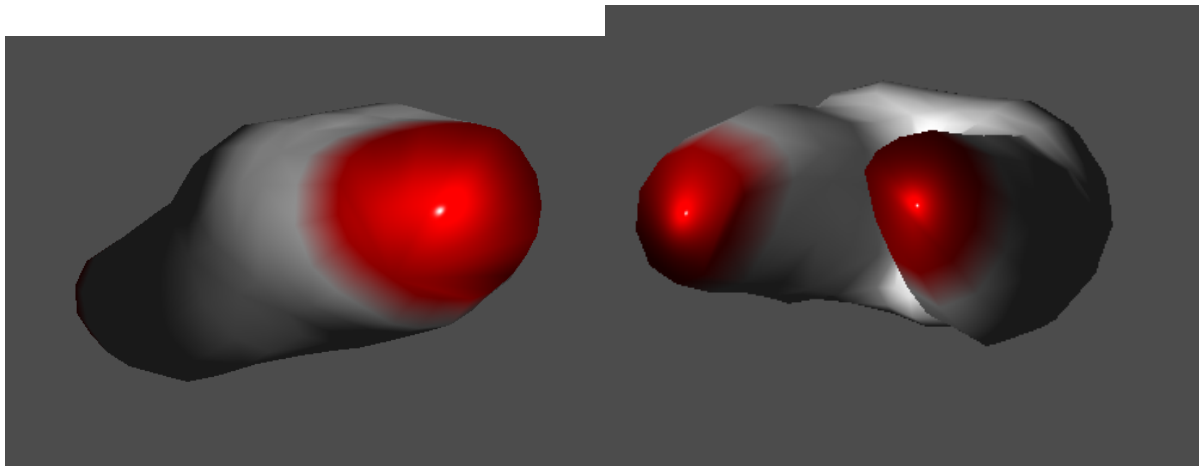


Figure 14: Resultat: Réparation des trous

Résultat de notre implémentation avec kate\_fur.obj

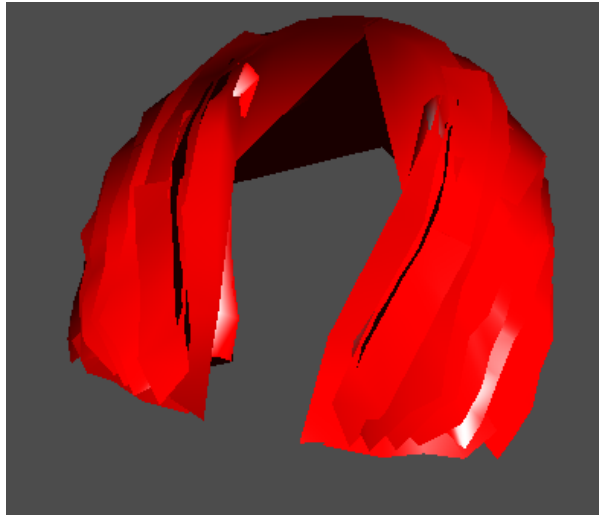


Figure 15: Résultat: 24 trous détectés, réparation incomplète...

# References



[1] Julie Digne (2014)

An Analysis and Implementation of a Parallel Ball Pivoting Algorithm  
*Image Processing On Line*, 149 – 168.



[2] D.Eberly (2002)

Triangulation by ear clipping



[3] R. Liu, D. Burschka, and G. Hirzinger (2007)

On the way to water-tight mesh

*International Society for Photogrammetry and Remote Sensing* , 1– 7.

Merci de votre attention