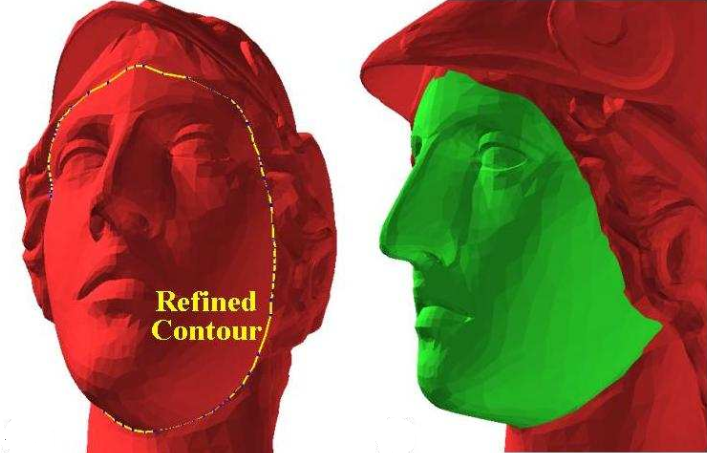


RAPPORT DE PROJET DE FIN D'ÉTUDE – M2 GIG 2019-2020

Segmentation 3D par la méthode du Livewire

****

*Projet réalisé par*

Pierre MATTIOLI

Johan MABILY

Erwan LERIA

*Projet encadré par*

Romain RAFFIN

Table des matières

[1) Introduction 2](#_Toc35630087)

[2) Analyse du problème posé 2](#_Toc35630088)

[3) Étude de la méthode du Livewire 3](#_Toc35630089)

[3.3) Critère d’orientation de la normale 3](#_Toc35630090)

[4) Étapes + signification + images d’exemples d’algorithmes ? 4](#_Toc35630091)

[5) Résultats obtenus 4](#_Toc35630092)

[6) Description et images résultats + difficultés rencontrées ? 4](#_Toc35630093)

[7) Conclusion 4](#_Toc35630094)

[8) Notre point de vue sur la méthode et l'algorithme + conclusion générale + perspectives futures... 4](#_Toc35630095)

[9) Bibliographie 5](#_Toc35630096)

# Introduction

Dans le cadre de notre seconde année de master informatique option géométrie et informatique graphique, il nous est proposé un projet de fin d'étude nous permettant de mettre en pratique nos connaissances et nos compétences professionnelles au travers d’un cahier des charges ayant pour finalité le développement et l'analyse d'algorithmes en accords avec nos intérêts professionnels et la spécialité de notre master.

# Analyse du problème posé

La méthode du Livewire, référencée également comme “Intelligent Scissors” est largement utilisée en 2D (sur des images) pour déterminer des zones, dont les contours sont adaptés au fur et à mesure que l’utilisateur trace le contour à la souris. On souhaiterait l’adapter à la sélection de zones sur un maillage 3D en fonction de critères plus larges (“qualité” des triangles, valence, ratio de forme…). Cela nécessite d’implémenter l’algorithme initial, d’en comprendre la construction et de proposer une structure générique de contraintes qui puissent être gérées par la segmentation.

On étudiera également l’aspect incrémental de la méthode (où l’utilisateur pourrait rajouter des nœuds de segmentation après coup), afin d’envisager l’algorithme sur une représentation multi-échelles (départ avec un objet décimé, suivi de la segmentation sur un maillage plus volumineux, voire trop).

# Identification de la partie à segmenter

## 3.1) Identification.

Le but de ce projet est de faire de la segmentation 3D de maillage, et pour cela, il faudrait dans un premier temps détecter les parties que l’on souhaite segmenter. On précise que les parties que l’on veut reconnaître sont des composantes topologiques telles que des bras, des jambes ou autres attributs morphologiques de ce genre.

Il se trouve qu’il existe déjà un algorithme, utilisant de l’intelligence artificielle, qui permet de détecter ces parties sur un nuage de points. L’inconvénient de cette méthode est qu’elle ne peut pas s’appliquer sur un maillage. Mais avec quelques étapes supplémentaires, nous pensons que nous pouvons retrouver la composante détectée sur un nuage de point, sur un maillage issu du même nuage.



Nous précisons que les composantes des nuages de points étudiés, nous ont été fournies au préalable. En effet notre travail consiste surtout à transposer ce résultat sur un maillage. Pour cela, une première étape sera d’extraire les points composants la bordure du nuage segmenté, et de les projeter sur des sommets du maillage à des positions équivalentes. Une seconde étape sera de relier logiquement par des arêtes les sommets trouvés sur le maillage.

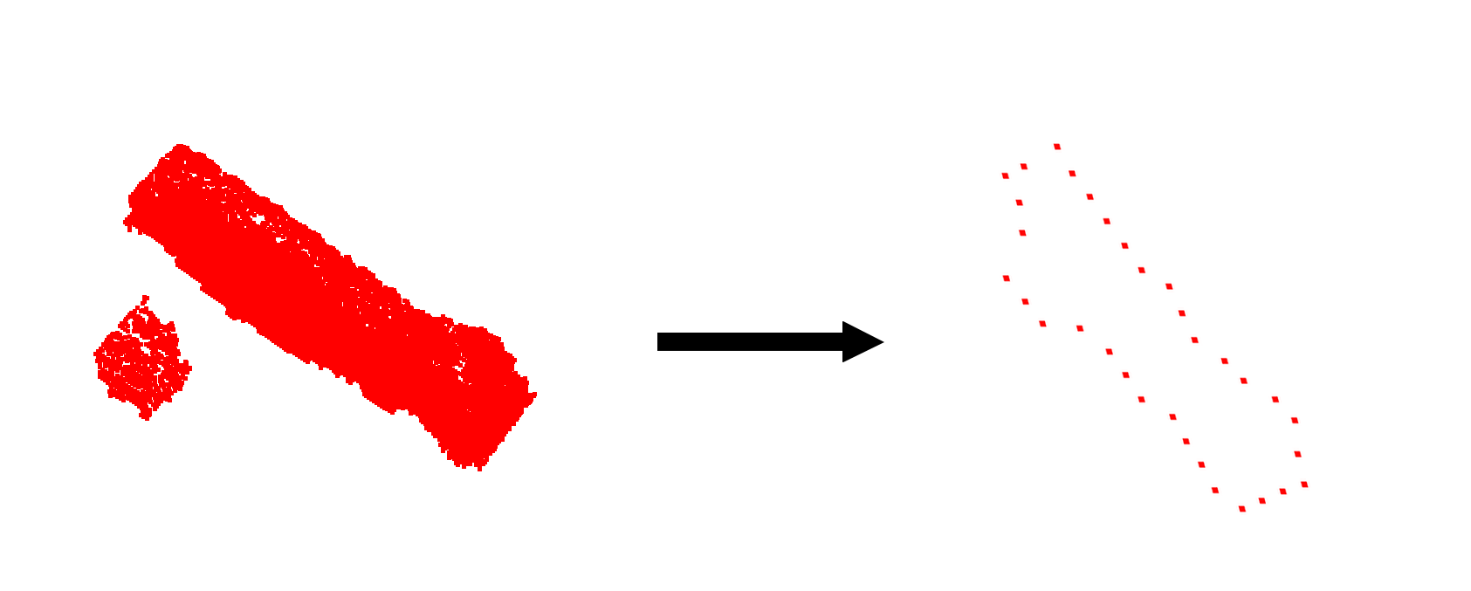
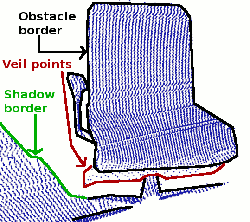
Dans les prochaines sections, les références au nuage de point correspondront au nuage de point représentant la partie segmentée du nuage d’origine.

## 3.2) Extraction de contour

Avant de pouvoir projeter, il faut déterminer quels sont les points qui constituent le contour du nuage. Cela peut se faire manuellement à l’aide d’un visualisateur grâce à des outils comme Meshlab ou Blender, mais nous favoriserons une méthode plus automatique.

Nous avons opté pour une extraction de bordure utilisant du Range Imaging. Le Range Imaging est en fait le nom d’une collection de techniques utilisées pour produire une image 2D à partir d’une scène 3D. L’image 2D contient alors des pixels qui indiquent les distances de tous les points de la scène 3D à un autre point spécifique.

La méthode que l’on utilise permet d’obtenir les « bords obstacles » à partir de d’une image obtenue par Range Imaging, ce qui veut dire les points de transitions entre le fond et le premier plan. Les « Veil Points » et « Shadow borders » que nous pouvons voir dans l’image ci-dessous ne sont pas importants pour ce que l’on cherche à faire.



## 3.3) Projection

Une fois que nous avons extrait le contour, il s’agit de pouvoir le retrouver sur le maillage. Sachant que le maillage a une position calquée sur le nuage de point, ceci va nous faciliter la tâche. Nous utiliserons donc un simple test de distance min, afin d’attribuer un point à un sommet du maillage.

Pour chaque point P du nuage de point

Pour chaque sommet S du maillage  
 dist = Calculer distance de P à S

Si dist < minDist Alors minDist=dist Et minS = S

Contour += Smin

Selon la taille et la topologie du maillage, plusieurs points peuvent être attribués au même sommet. Cela n’est pas gênant mais il faudra quand même éviter d’avoir des doublons inutiles.

Selon comment le contour a été construit, les sommets ne seront probablement pas tous ajoutés dans l’ordre. On utilise l’algorithme ci-dessous pour les replacer.

Pour chaque sommet S du Contour

// TODO

# Étude de la méthode du Livewire

Notre but est donc d’implémenter au mieux la méthode décrite dans l’article « Intelligent Scissoring for Interactive Segmentation of 3DMeshes » réalisé par William Kiefer. Toutefois, cette méthode est initialement optimisée pour une interaction avec un utilisateur, devant tracer à la main un contour sur un maillage. Nous ne nous concentrerons donc pas sur cette partie puisqu’elle ne sera pas prise en compte dans notre projet. Bien sûr nous apporterons quelques modifications et ajustements pour mieux correspondre à notre approche et aux résultats souhaités.

Une fois que nous avons choisis les sommets sur le maillage qui représentent des parties du contour, nous devons appliquer l’algorithme d’Intelligent Scissoring pour relier ces sommets par des arêtes. Afin de relier tous ces sommets, l’algorithme sera appelé de manière itérative et le nombre d’appels sera égal au nombre de sommets choisis. Par exemple si nous avons deux sommets, le nombre d’appel sera de deux : un pour relier les deux sommets, et un pour fermer le contour.

À la manière d’un Dijkstra, nous devons considérer toutes les arêtes du maillage pour calculer le meilleur chemin entre 2 sommets. Mais au lieu d’utiliser seulement la distance entre deux sommets, nous utiliserons une fonction de coût modulable selon des critères (décrite dans la section ..), pour donner un poids à chaque arête.

La forme de l’algorithme est calquée sur la méthode du livewire 2D, mais au lieu de considérer le N-voisinage, on considère les sommets adjacents à un sommet en utilisant les arêtes qui les relient. C’est d’ailleurs avec ces arêtes que nous allons appliquer des critères de coût pour choisir un chemin.

## 3.3) Critère d’orientation de la normale

Ce critère permet d’encourager le chemin de moindre coût à traverser l’arrière du maillage, c’est-à-dire de choisir un chemin qui passe par la partie non visible par l’utilisateur. De plus, sans ce paramètre, le chemin choisit pourrait avoir tendance à seulement suivre les bords visibles du maillage. Le principe repose sur le fait de donner un moindre coût aux arêtes, dont les normales de leurs faces adjacentes sont aux mieux alignées avec la direction du point de vue de l’utilisateur. Ce critère sera surtout utilisé pour fermer un contour et encourager le chemin à ne pas repasser sur la partie du contour déjà tracée.

# Bibliographie

Intelligent Scissoring for Interactive Segmentation of 3D Meshes – William Kiefer : <http://will.kiefer.io/papers/wkiefer_thesis.pdf>

Livewire Segmentation Technique

<https://en.wikipedia.org/wiki/Livewire_Segmentation_Technique>

3D

<http://www.cb.uu.se/~filip/Thesis/PaperI.pdf>

<http://leogrady.net/wp-content/uploads/2017/01/grady2006minimal.pdf>