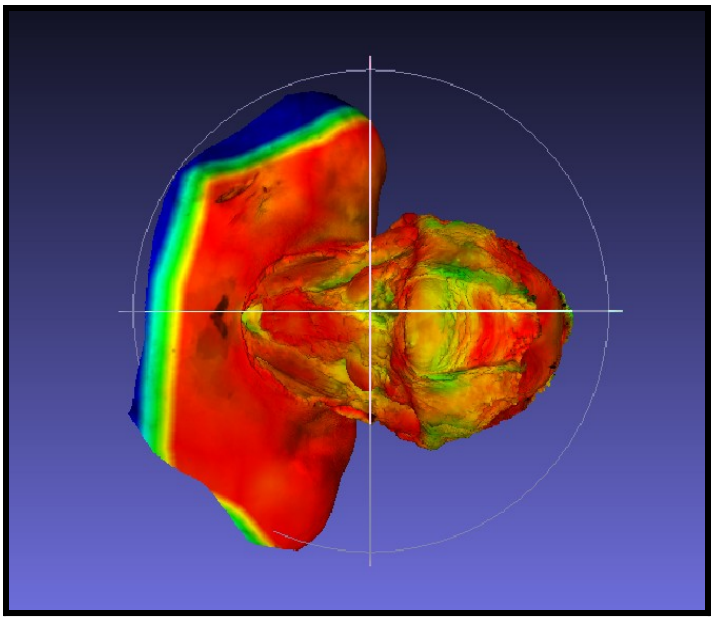
Rapport de stage

« Traitement de maillages 3D pour la simulation de dissection chirurgicale »



Rédigé par **Thibault ODORICO**

Tuteurs de stage : **Gérard Subsol** – **Noura Faraj**
Tuteurs universitaire : **Pascal Giorgi**

Pour l'obtention du Master IMAGINA

2020-2021

Remerciements

Tout d’abord j’adresse mes remerciements à Monsieur Gérard SUBSOL qui à proposé le sujet ainsi qu’à Madame Noura FARAJ. Ils ont tout deux acceptés de m’encadrer, fait profiter de leur savoir et de leurs grandes expériences lors de la réalisation de mon stage.

Monsieur Guillaume CAPTIER à réalisé les dissections permettant l’acquisition surfacique nécessaire à la réalisation de mon stage, je le remercie donc pour son travail ainsi que pour son intérêt et sa disponibilité lors des entretiens réalisés avec mes encadrants de stages.

Ce travail sera examiné et évalué par mes professeurs, je leur adresse donc mes remerciements pour avoir accepté de faire partie du Jury. J’exprime ma considération et mes vifs remerciements à tous mes enseignants du Département des sciences de l’informatique pour les enseignements qu’ils m’ont prodigués tout au long de mes années de formation en Licence informatique et en Master IMAGINA à la faculté des sciences de Montpellier.

Ce stage a été réalisé au sein de l’équipe ICAR du LIRMM pendant une durée d’environs 5 mois et demis, Je tiens donc à témoigner toute ma reconnaissance à l’ensemble de l’équipe ICAR ainsi qu’a tout le personnel administratif et technique de cette institution pour l’assistance et l’aide qu’ils m’ont offerts tout au long de ce stage.

Table des matières

1. Recherches.....6
Appréhension du problème.....6
Fonctionnalités nécessaires.....9
Recherches d'outils adéquates.....9
Sélection des outils optimaux.....9
2. Développement et Expérimentations.....9
Mise en place des outils.....9
Implémentation.....9
3. Résultats.....10
Algorithme final de Matching.....10
Présentation du visualiseur.....10

Présentation et Contexte

Le présent travail s'inscrit dans le cadre d'un stage de fin d'étude permettant l'obtention du Master IMAGINA spécialisé dans le traitement d'image, la modélisation 3D, la création de moteurs de jeu ainsi que la création d'agents intelligents. Il est encadré par deux tuteurs d'entreprise et un tuteur universitaire, respectivement Mr. Subsol Gérard, Mme. Faraj Noura et Mr. Pascal Giorgi. s'est effectué au sein de l'équipe ICAR.

Le stage s'est effectué au LIRMM, Laboratoire d'Informatique, de Robotique et Microélectronique de Montpellier dépendant de l'Université de Montpellier. L'institut est divisé en trois départements de recherches, eux-même divisés en équipes de recherche. L'équipe ICAR où ce stage a été réalisé travaille sur l'interaction et le traitement des données visuelles telles que les images, les vidéos et les objets 3D. La partie de Modélisation et visualisation consiste à modéliser des ensembles de données complexes afin de pouvoir visualiser intuitivement ou manipuler des données pour en extraire les connaissances.

Introduction

Depuis plusieurs décennies, la visualisation 3D est devenu un moyen très pratique pour apprendre l'anatomie humaine. Il existe trois types de visualisations principales dans le cadre l'apprentissage de la médecine. Tout d'abord, la visualisation voxelique utilisée notamment dans l'imagerie médicale (IRM), introduit d'abord par le projet Visible Human Project qui avait permis la première visualisation 3D complète d'un corps humain en 1998. Ensuite, la visualisation polygonale schématisée à plusieurs niveaux (ossements, chairs, veines, …) réalisée par les logiciels du type Primal's 3D view, Anatomography et Zygote Body. Enfin, la visualisation surfacique texturée en haute définition permet d'avoir une représentation réaliste. C'est pourquoi cette visualisation sera utilisée dans ce projet pour simuler une dissection réaliste induisant une meilleure appréhension de l'anatomie humaine (affichage de superpositions d'organes, texture de la peau, assemblage des muscles, etc.)

Dans l'optique d'obtenir cette visualisation réaliste, des séries de maillages 3D surfaciques texturés à très haute définition ont été acquises au laboratoire d'anatomie de la Faculté de Médecine de Montpellier à l'aide d'un scanner surfacique (Artec Spider) . On appelle maillage un modèle en trois dimensions composé de plusieurs faces généralement triangulaires. Un maillage peut être séparé en sous maillages. C'est donc un vrai cadavre qui a été disséqué et scanné en 3D. Ces maillages correspondent aux différentes étapes de la dissection du cou. Les modèles obtenus sont presque similaires à l'exception des nouvelles parties disséquées à chaque étape. C'est ici qu'intervient le sujet de ce stage qui est le suivant « Traitement de maillages 3D pour la simulation d'une dissection chirurgicale ». En effet, ces maillages nécessitent donc divers traitements géométriques. Suite a plusieurs observations, trois objectifs ont été déterminés :

Tout d'abord, on observe que la majorité de l'information d'une série de maillage correspondant à une dissection est redondante. Afin d'optimiser la mémoire de stockage, on cherche à créer une super structure qui stockerait le minimum d'informations nous permettant de reconstruire chaque étape de dissection. Ainsi il serait possible de jongler d'une étape de dissection à une autre avec le minimum de redondance.

Par la suite, dans le cas où la représentation en trois dimensions est la plus réaliste possible, des évolutions peuvent survenir. En effet, le corps continu à se dégrader ou décomposer (couleur et texture de la peau) mais également, il peut y avoir des modifications apportées par le médecin chargé de la dissection (incisions, déplacements, prélèvements, …). C'est pourquoi chaque changement devra être analysé dans l'optique de le représenter ou non selon son importance.

Pour finir, dans le cas où les deux précédents objectifs seraient parfaitement réalisés, il faudrait adapter l'application à une interface tactile dans le but d'utiliser une tablette pour manier plus facilement le logiciel (utilisation d'un stylet avec pression pour inciser).

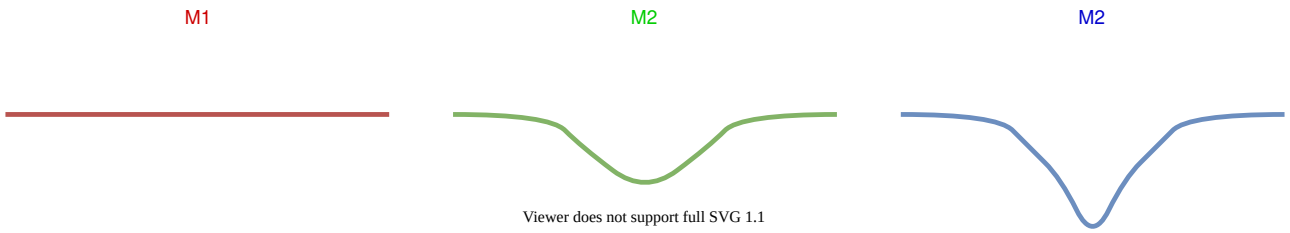
Tout au long de ce rapport nous tenterons donc de répondre à la problématique « Comment implémenter une structure de "super maillages" permettant de reconstruire facilement les étapes de

la chirurgie tout en optimisant l'espace mémoire utilisé et en réduisant le bruit d'acquisition ? ». En effet, dans une première partie, nous détaillerons la partie de recherche qui a été nécessaire pour commencer le travail. Puis dans un second temps, nous détaillerons tout le processus de développement de l'application qui a été fait pour amener à une partie exposant les résultats obtenus ainsi que leurs interprétations.

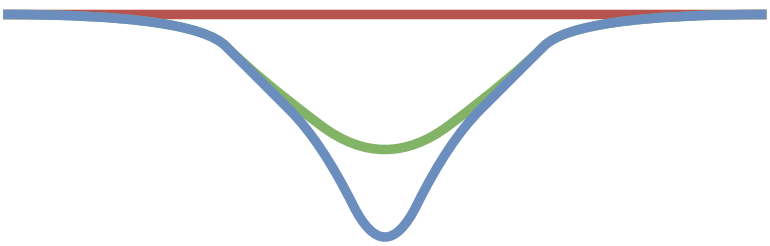
1. Recherches

Appréhension du problème

Pour avoir une bonne appréhension du problème est des objectifs nous allons partir d'un cas schématique simple. Soit 3 maillages, M1, M2, M3 représentés ci-dessous :



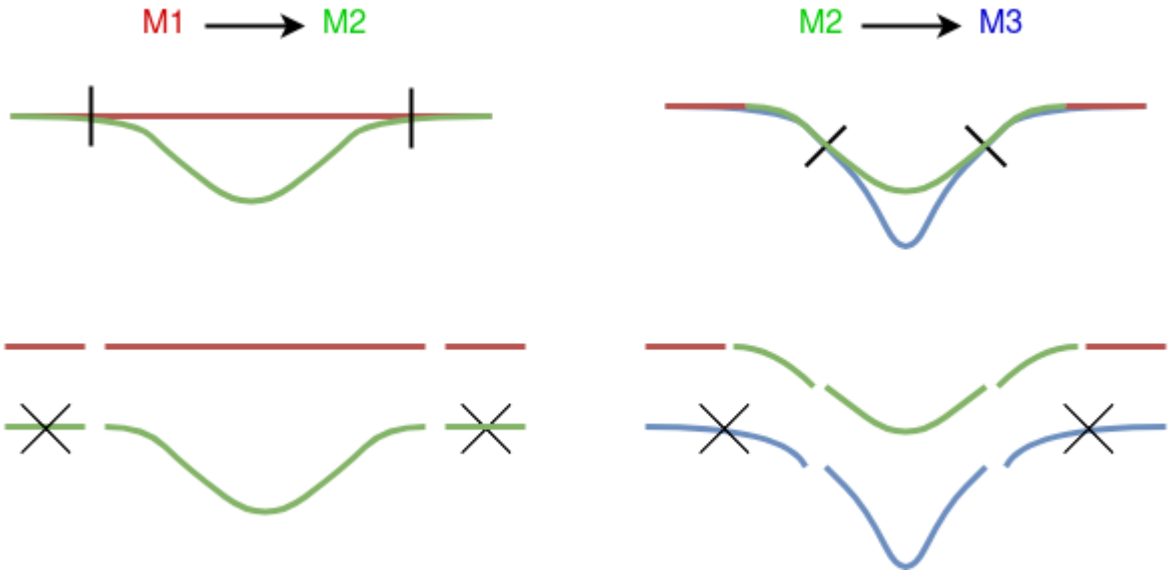
En superposant les maillages on peux se rendre compte que des parties sont partagées par la série de maillages il y a donc de la redondance dans l'information ce qui un n'est pas nécessaire.



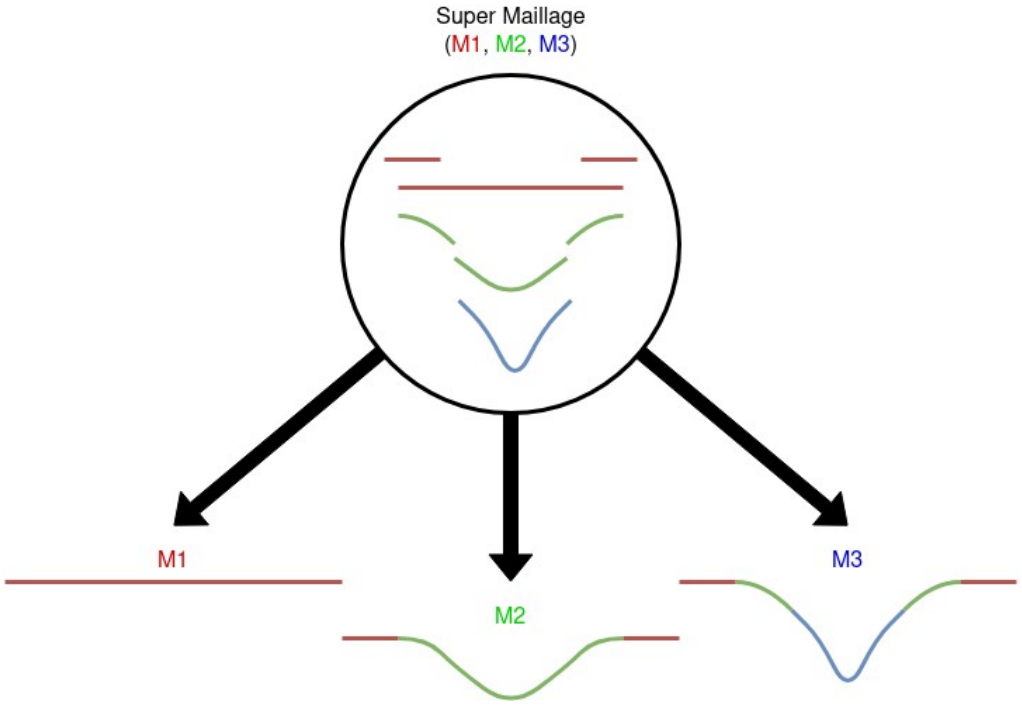
L'idée ici est parcourir la série de maillage à l'aide d'un algorithme qui fera une comparaison deux à deux des maillages successifs de la manière suivante :

Dans un premier temps on détectera les limites entres zones similaires et zones différentes ce qui peut être accompli en calculant les distances points à points.

Dans un second temps on décomposera les deux maillages en sous parties cela nous permettra de supprimer la redondance de l'information.



Dans un troisième temps, nous stockerons les parties utiles dans un super maillage en prenant soin de noter les associations pour pouvoir par la suite reconstruire notre série de maillage au complet mais cette fois-ci avec un gain mémoire conséquent.



Au premier abord le problème pourrait paraître simple. Cependant la tâche se complique avec le jeu de données fournis. En effets, la série de maillage chirurgical acquise est à très haute résolution, elle est donc non seulement très gourmande en mémoire mais également longue à traiter géométriquement. Il faut ainsi utiliser des traitements et outils performants pour pouvoir offrir un algorithme utilisable. De plus, les acquisitions ne pouvant être parfaites, les maillages possèdent de légers décalages sur des parties que l'on souhaiteraient être à la base identiques entre 2 maillages. Des seuils d'erreur paramétrables sur les distances et à plusieurs autre niveaux ainsi que des zones de transitions entre les parties similaires et différentes des maillages seront ainsi pris en compte pour obtenir un résultats visuel satisfaisant.

L'algorithme de découpage entre deux maillage M1 et M2 est donc le suivant :

1. Sélection des points distants entre M1 et M2 selon un seuil paramétrable

   → Points Distants de M2 appartenant à M1 : **PDM1**

   → Points Distants de M1 appartenant à M2 : **PDM2**

2. Utilisation des points distants comme graines d'un algorithme de « Region Growing » selon un paramètre de propagations des régions

   → Régions Distantes de M2 appartenant à M1 : **RDM1**

   → Régions Distantes de M1 appartenant à M2 : **RDM2**

3. Sélection des régions intéressantes selon un critère de taille (les régions trop petites seront ignorées)

   → Régions Distantes de M2 puis seuillés appartenant à M1 : **RDSM1**

   → Régions Distantes de M1 puis seuillés appartenant à M2 : **RDSM2**

4. Création d'un découpage de M1 à l'aide de **RDSM1** selon un paramètre de zone de transition à conserver (Cela permettra de greffer les régions distantes de M2 sur une base appartenant à M1)

   → Base de Greffe appartenant à M1 : **BGM1**

5. Projection des zones de transition de **BGM1** sur les régions de **RDSM2** correspondantes avec une structure MLS (Cela permettra d'avoir une transition fluide entre les deux)

   → Base de Greffe appartenant à M1 Projeté sur les régions de **RDSM2** : **BGPM1**

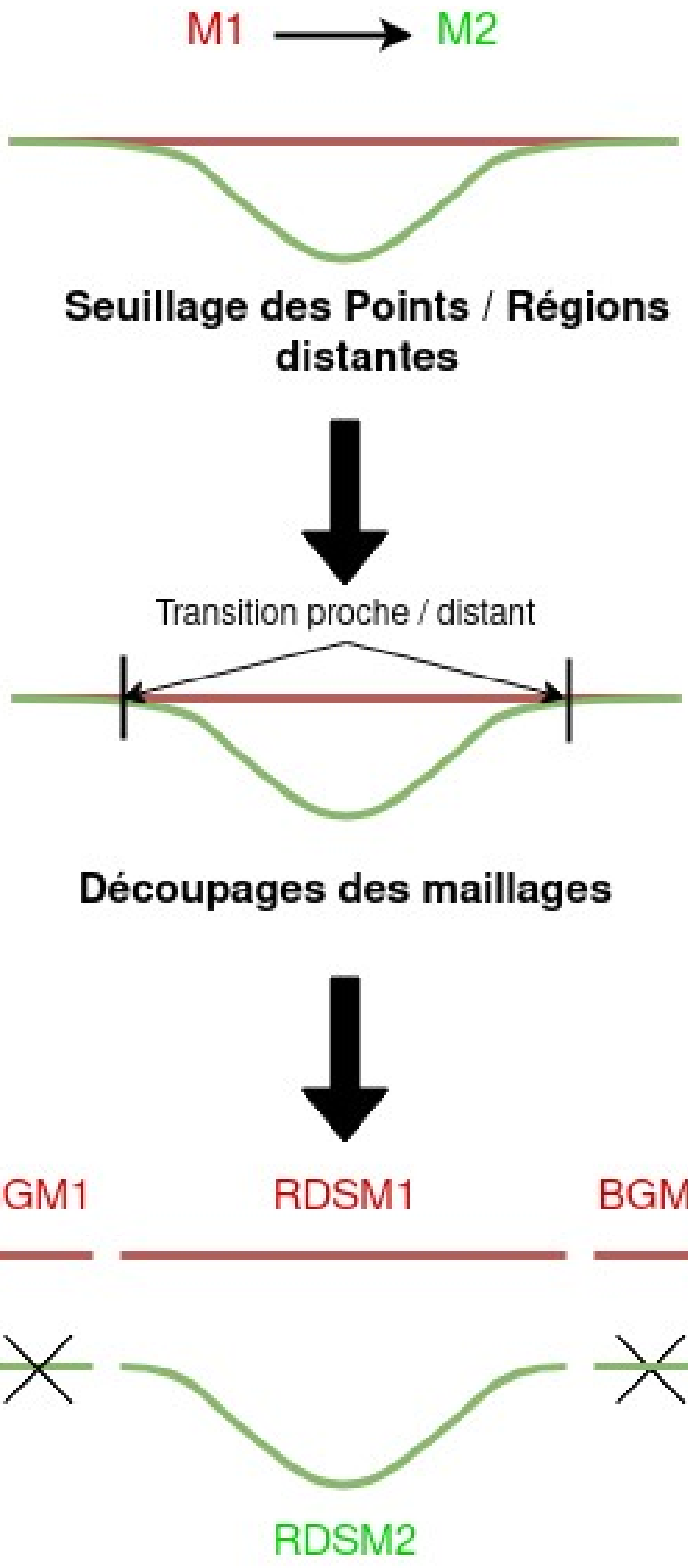
   → (Optionnel) Projeté **RDSM2** sur **BGPM1** (meilleur résultats visuel ?)

6. Fusion de **BGPM1** avec **RDSM2** à l'aide d'un lissage laplacien (meilleur résultats visuel ?)

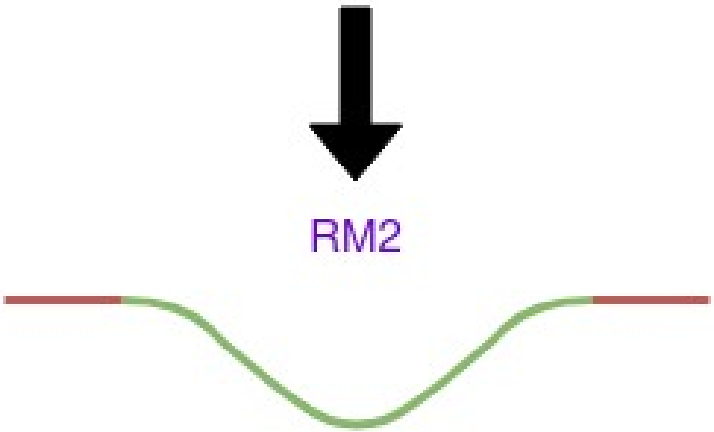
   → Reconstruction de M2 à partir des morceaux similaires de M1 : **RM2**

7. Indexation des composants nécessaire à la reconstruction de M1 et M2 en récoltant **RDSM1**, **RDSM2** ainsi que les éléments produits par la différence de **RM2** avec **RDSM2** et

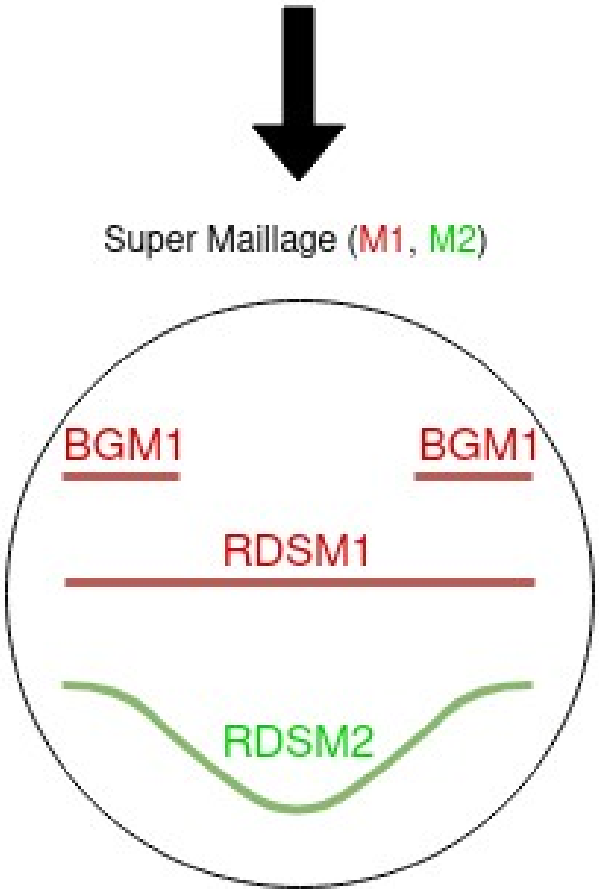
   → Composants de reconstruction de **M1** et **M2** : **CRM1M2**


M1
M2
Seuillage des Points / Régions distantes
Transition proche / distant
Découpages des maillages
BGM1
RDSM1
BGM1
RDSM2

Grèffe de M2 sur M1



Sauvegarde des composants de reconstruction



Fonctionnalités nécessaires

(Recensement des fonctionnalités et explication de leurs utilités)

Recherches d'outils adéquates

(Recherches bibliographiques)

Sélection des outils optimaux

(Pourquoi certaines technologies s'adaptaient plus au projet que d'autres)

2. Développement et Expérimentations

Mise en place des outils

(Utilisation de Cmake pour pouvoir compiler en multi-plateforme)

(Utilisation d'un répertoire Github avec travis pour partager et tester les build en multi-plateforme)

(Mise en place d'un outils de format de code source clang-format, cmake-format, ...)

(Optimisation de la durée de compilation)

→ Accélère le temps de développement, comparaison avant après

(Création d'un programme test (affiche un modèle 3D))

Implémentation

(Création d'un jeux de données simple)

→ Accélère le temps de développement, comparaison avant après

(Implémentation de l'algorithme de Matching avec le visualiseur CGAL)

→ Limites du visualiseur observés + ajout de fairing pour améliorer le rendu

→ Préciser le nouveaux protocole

(Implémentation d'un visualiseur avec Qt5 + LibQGLViewer)

3. Résultats

Algorithme final de Matching

(Comparaison des différentes versions)

(Afficher les distances de hausdorf)

(Cas limites)

(Points d’améliorations)

Présentation du visualiseur

(Fonctionnalités)

(Utilisation)

(Application sur de vrais jeux de données)

Conclusion

Références bibliographique