

Graphe dual d'un maillage

1 Modalités

Le projet est à rendre au plus tard le vendredi 22 décembre 2023, à 23h55 via la plateforme Moodle. Il est à réaliser exclusivement en binôme, durant votre temps libre.

1.1 Travail à rendre

Il vous est demandé un document synthétique (un rapport) tenant sur au maximum trois feuilles recto-verso, soit six pages, où vous ferez figurer les réponses aux questions ainsi que tout élément que vous jugeriez bon de préciser. Vous déposerez sur Moodle, dans une archive au format `.tar.gz` :

- le document synthétique au format électronique `.pdf` ;
- les sources dûment commentées de votre réalisation en langage C, ce qui inclut notamment un fichier `Makefile`, mais pas les fichiers objets `.o`. Un code qui ne compile pas ne sera pas accepté.
- les jeux de tests (les programmes de tests et les graphiques correspondant aux résultats de vos tests).

L'ensemble des programmes ayant servi à l'élaboration du projet doit être dûment commenté. Essayez d'attacher une grande importance à la lisibilité du code : il est indispensable de respecter les conventions d'écriture (style de codage) du langage C et de bien présenter le code (indentation, saut de ligne, etc.) ; les identificateurs (noms de types, variables, fonctions, etc.) doivent avoir des noms cohérents documentant leur rôle ; indiquez les pré-conditions, post-conditions, paramètres, la sortie et un descriptif de chaque fonction. Un commentaire se doit d'être sobre, concis et aussi précis que possible. Pensez à utiliser des outils comme `gdb` ou `valgrind` pour faciliter votre développement et optimiser votre programme.

1.2 Barème indicatif

- Code (programmation et algorithmique) : 13 points
- Code (documentation) : 2 points
- Tests (évaluation de performance, graphiques) : 2 points
- Rapport : 3 points

Pour chaque partie du sujet, le nombre de points y est indiqué. Il correspond à l'ensemble des points qui concerne cette partie (code + rapport + éventuellement tests si demandés)

2 Problème

L'objectif de ce projet est de générer le graphe dual d'un maillage. Un maillage (mesh en anglais) est la représentation d'un objet 3D définie par sa géométrie. Une façon commune de créer un maillage consiste à définir d'une part l'ensemble des positions des sommets dans un repère cartésien. D'autre part, la surface du maillage se définit en reliant ces sommets trois par trois pour former des faces (qui sont des triangles). Un des formats de données utilisés pour stocker ces informations est le format `.obj`. En voici un exemple représentant un tétraèdre :

```
v -0.5 0 0.5
v 0 0 -0.5
v 0.5 0 0.5
v 0 1 0
```

```
f 1 2 3
f 1 4 3
f 1 2 4
f 2 3 4
```

- Les lignes du fichier commençant par **v** (pour **vertex**, se traduisant par sommet) définissent la position 3D d'un sommet. Les trois valeurs suivantes désignent les coordonnées x y et z de ce point, on les interprétera comme des valeurs de type **float**.
- Les lignes du fichier commençant par **f** (pour **face**) suivie par 3 valeurs entières supérieures ou égales à 1 représentent une face. Chaque valeur désigne l'indice de sommet utilisé pour former la face. (l'indice numéro 1 correspond au premier sommet, l'indice numéro 2 au deuxième, etc...)

Le dual d'un maillage est un graphe non orienté reliant les centroïdes des faces adjacentes entre eux. Pour ce projet le centroïde correspond au barycentre du triangle. (Aucune pondération n'est effective sur les sommets : Il suffit de calculer la moyenne des trois sommets pour l'obtenir). Chaque face du maillage est composée de 3 côtés que l'on appelle des arêtes. Si deux faces différentes partagent une arête du maillage, alors les deux faces sont adjacentes et il faudra créer une arête duale reliant les deux centroïdes de ces faces. Une arête se définit par deux indices de sommets (par exemple 3-5). Les arêtes sont dites non-orientées, cela signifie que l'arête 3-5 qui appartiendrait à une face et l'arête 5-3 qui appartiendrait à une autre face sont considérées comme identiques (ou équivalentes). Il est possible de représenter un graphe dual dans un fichier **.obj**. Voici le graphe dual correspondant à l'exemple de maillage précédent :

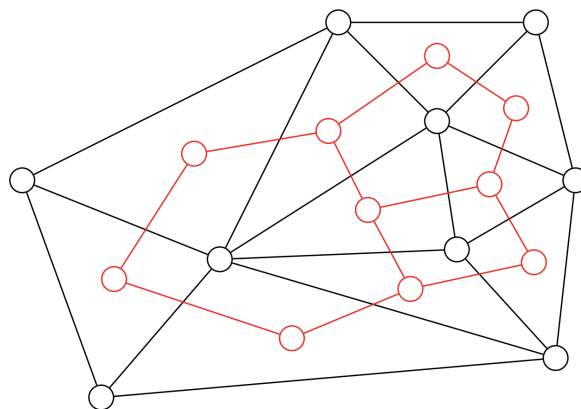
```
v 0.000000 0.000000 0.166667
v 0.000000 0.333333 0.333333
v -0.166667 0.333333 0.000000
v 0.166667 0.333333 0.000000
1 2 1
1 3 1
1 3 2
1 4 1
1 4 2
1 4 3
```

Dans l'écriture du graphe dual dans un **.obj**, vous devez définir les centroïdes sous forme de sommets. Pour représenter les arêtes duales reliant les centroïdes, utilisez la lettre 1 suivie des deux indices de sommets (1 correspond au premier sommet, 2 au deuxième, etc...)

On vous demande de concevoir un parseur de fichier **.obj** pour lire les informations du maillage, d'appliquer divers algorithmes permettant de générer son graphe dual, et d'écrire le résultat dans un fichier **.obj** de sortie. Une partie bonus consiste à colorier les centroïdes du graphe dual.

Voici un exemple graphique d'un maillage et de son graphe dual, comme représenté dans cet article de recherche :

https://www.researchgate.net/publication/338004794_Breaking_Permutation-Based_Mesh_Steganography_and_Security_Improvement.



Dans cette illustration :

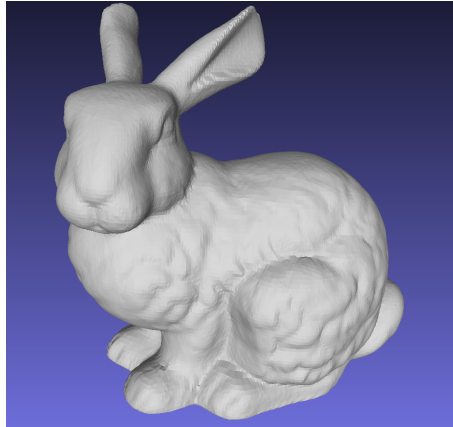
- Les cercles noirs représentent les sommets du maillage.
- Les lignes noires représentent les arêtes du maillage.
- Les cercles rouges représentent les centroïdes du graphe dual.

— Les lignes rouges représentent les arêtes duales du graphe dual.

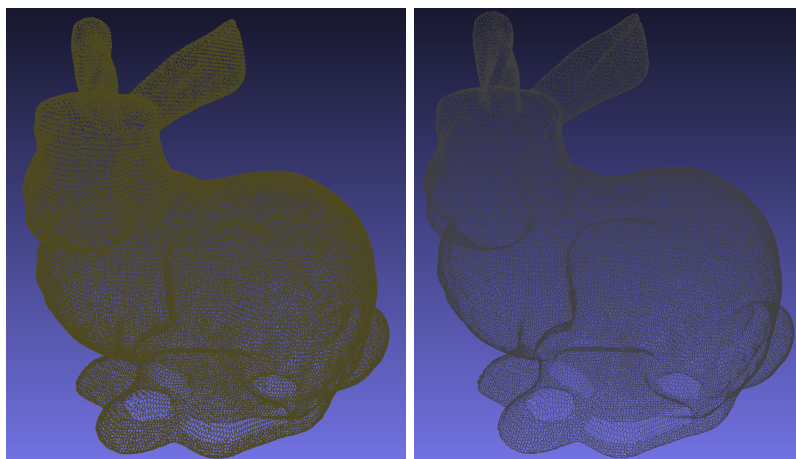
Pour visualiser les **.obj**, vous avez à disposition sur Moodle plusieurs obj simplifiés (avec uniquement des sommets et des faces) que vous pouvez utiliser pour vos tests. Vous pouvez les visualiser à l'aide de plusieurs logiciels gratuits :

- MeshLab <https://www.meshlab.net/>
- Blender <https://www.blender.org/>

Les maillages à votre disposition sur Moodle sont de différentes tailles et vous les utiliserez autant que possible pour vos tests et pour la génération des graphiques de performances. Vous pouvez également créer vos propres maillages (de préférences très simples), pour tester vos algorithmes. Voici un exemple de fichier **.obj** importé avec MeshLab :



Et voici le graphe dual associé sur MeshLab (à gauche arêtes + sommets, et à droite les arêtes uniquement)



Remarque importante : Pour ce projet, on ne s'intéressera qu'à des maillages **2-manifold**. Cela signifie qu'une arête ne peut être partagée que par deux faces au maximum.

Attention : Votre programme prendra 3, voir 4 paramètres en entrée :

- Le premier paramètre est le maillage d'entrée au format **.obj**.
- Le deuxième paramètre est le fichier de destination contenant le graphe dual associé, également au format **.obj**.
- Le troisième paramètre indique l'algorithme utilisé pour générer les arêtes duales. (Voir partir 4)
- Un quatrième paramètre optionnel pour indiquer si la coloration du graphe dual est appliquée. (Voir partie 5)

Voici un exemple d'utilisation : `./grapheDual bunny.obj bunnyDual.obj AVL`

3 Création d'un parseur

3.1 Lecture d'un graphe dual (2 points)

On vous demande en premier lieu d'écrire des fonctions permettant de lire un fichier **.obj** simplifié pour le stocker dans diverses structures de données. On dit "simplifié" car le format

.obj contient en effet diverses autres fonctionnalités qu'on ne vous demandera pas d'implémenter. Les fichiers **obj** de tests ne contiennent que des sommets (associant leurs positions) et des faces (associant 3 par 3 les indices de sommets pour former les faces).

Créez une structure **Maillage** contenant toutes les informations utiles propres au maillage décrit.

Remarque importante : L'indice 1 dans un **.obj** correspond au premier sommet, alors que l'indice du premier élément d'un tableau en C est 0. Il est conseillé de faire la transformation dès que possible dans votre programme. Lors de la création du fichier **.obj** de sortie représentant le graphe dual, n'oubliez pas d'appliquer la transformation réciproque si nécessaire.

3.2 Ecriture d'un graphe dual (2 points)

Écrivez une fonction permettant d'écrire dans un fichier **.obj** un graphe dual avec comme entrée une liste ou un tableau de centroïdes (qui sont représentés par des sommets) et les arêtes duales également sous la forme d'un tableau ou d'une liste liant les centroïdes via leurs indices. Étant donné qu'à ce stade vous n'avez pas encore calculé ces graphes, créer en un vous même pour tester la fonctionnalité.

Attention : Les centroïdes ne doivent en aucun cas être dupliqués !

4 Génération du graphe dual

4.1 Génération des centroïdes (2 points)

Créer une fonction qui à partir du maillage en entrée génère un tableau de centroïdes. Pour n faces présentes dans le maillage, vous devez renvoyer n centroïdes. Le centroïde d'indice i dans le tableau représente le centroïde de la face d'indice i dans le maillage. Pour n faces présentes dans le fichier **.obj** initial, vous devez avoir n centroïdes dans le fichier **.obj** de sortie représentant le graphe dual.

4.2 Génération des arêtes duales

4.2.1 Questions préliminaires (2 points)

- Créer une structure de donnée **Arete** représentant une arête. Vous pouvez y ajouter plusieurs informations que vous pourriez juger utiles pour la génération des arêtes duales du graphe dual, comme par exemple l'indice de la face associée.
- Écrire une fonction **sontEquivalentes** prenant en paramètre deux arêtes et qui renvoie 1 si les deux arêtes sont équivalentes, et 0 sinon.
- Pour comparer deux arêtes, écrivez une fonction **estSuperieureA** qui renvoie 1 si une arête est supérieure à une autre. (À vous de définir un ordre sur les arêtes). Cette fonction sera nécessaire pour trier des arêtes ou pour les insérer dans un arbre.

Les 4 sous parties suivantes ont pour but de générer les arêtes duales reliant les centroïdes. Pour chacune des parties, expliquez dans votre rapport votre algorithme ainsi que sa complexité algorithmique. De plus, évaluez les performances de l'algorithme en inscrivant les temps d'exécution dans un graphique que vous devez insérer au rapport ou en annexe. L'abscisse du graphique correspond au nombre d'arêtes du maillage d'entrée, et l'ordonnée correspond au temps d'exécution pour créer l'ensemble des arêtes duales. Utilisez des échelles logarithmiques pour chacun des deux axes.

Attention : Dans la mesure du temps d'exécution, ne mesurez pas l'écriture de vos arêtes duales dans le fichier d'obj de sortie, uniquement leur génération. Vous pouvez reprendre le code que vous avez écrit en TP.

4.2.2 Tri par sélection (2 points)

Utilisez l'algorithme du tri par sélection pour trier l'ensemble des arêtes du maillage et écrivez une fonction qui génère les arêtes duales à partir de ce tableau trié.

4.2.3 Tri par tas (3 points)

De la même manière que pour le tri par sélection, générez les arêtes duales mais en utilisant le tri par tas.

4.2.4 Arbre AVL (3 points)

- Créez une structure données **ArbreAVL** qui représente un arbre AVL d'arêtes avec les fonctions d'insertion et de recherche dédiées. Les deux fonctions pourraient être fusionnées en une seule pour de meilleures performances.
- Expliquez comment générer les arêtes duales à partir d'un AVL d'arêtes.
- Générez les arêtes duales en utilisant un AVL.

4.2.5 Table de hachage (2 points)

Pour la table de hachage, vous êtes invités à utiliser la bibliothèque liée à ce répo en ligne : <https://github.com/sheredom/hashmap.h>

Insérez le fichier .c et .h dans votre projet.

- Créez une fonction qui permet de créer une clé pour la table de hachage à partir d'une arête, et à qui cette arête sera associée.
- Comme pour l'AVL faites des insertions et des recherches pour créer les arêtes duales.

5 Coloration du graphe dual (2 points)

On propose de colorier les différents centroïdes du graphe dual. Pour cela, on choisit un centroïde (par exemple le premier) qui sera considéré comme le centroïde noir. Il est considéré comme étant à distance 0 de lui même. Les centroïdes voisins du centroïde noir sont considérés de distance 1, et leurs centroïdes voisins de distance 2, et successivement. Le(s) centroïde(s) le(s) plus éloigné(s) du centroïde noir est/sont le(s) centroïde(s) blanc(s).

- Expliquez quel algorithme de parcours vu durant cet enseignement permet de réaliser ce coloriage ?
- Quelle structure de données est à utiliser pour gérer l'adjacence et pourquoi ?
- Implémentez cet algorithme sur le graphe dual pour calculer les distances de chaque centroïde au centroïde de départ.
- Nous allons enfin ajouter une information complémentaire à chaque centroïde dans le fichier **.obj** de sortie :

```
v 3.454909 1.560489 8.599300 0.5 0.5 0.5
```

Comme précédemment, les trois premières valeurs suivantes le v correspondent aux coordonnées x y z du centroïde. Les trois valeurs suivantes correspondent aux composantes RGB (rouge, verte, bleu) de la couleur du sommet. Le centroïde noir aura pour valeurs 0.0 0.0 0.0. Le (ou les) centroïde(s) le(s) plus éloigné(s) a/ont pour valeurs 1.0 1.0 1.0. Les autres centroïdes auront des couleurs intermédiaires (des nuances de gris). Plus un centroïde est éloigné du centroïde noir de distance 0, plus il est clair. Dans notre exemple, le centroïde de couleur 0.5 0.5 0.5 est gris, il est donc à mi-distance entre le centroïde noir et blanc. Vous pouvez utiliser des couleurs différentes pour créer d'autres dégradés de couleurs.

Voici un résultat du graphe dual colorié en noir et blanc ainsi qu'en rouge et vert :

