

TP1 - Maillages

Alexandra Bac

Modélisation géométrique
Polytech Marseille - IRM 4A

Ce TP porte sur les maillages et on utilisera la bibliothèque C++ (très connue) CGAL. Le tuto CGAL vous donnera tout ce dont vous avez besoin pour l'aborder. Par ailleurs, vous téléchargerez le matériel de TP comportant un “starting code” qui vous aidera à démarrer.

Démarrage à partir du “starting code”

- Avant toute chose, éditez votre fichier `.bashrc` à la racine de votre répertoire et ajoutez :

```
export PATH=/amuhome/bruasse.a/Public/Libs/cmake-3.22.2/bin:${PATH}
export PATH=/amuhome/bruasse.a/Public:${PATH}
```

Fermez votre terminal puis rouvrez-en un (ou exécutez `source ~/.bashrc`).

- Téléchargez l'archive et décompressez-la.
- Allez dans le répertoire `build`
- `cmake ..`
- `make`
- `./poly_starter ../data/bunnyLowPoly.obj`

Vous pouvez maintenant visualiser le fichier généré (`test.off`) avec meshlab (comme vous avez inclus le chemin vers mon répertoire public, il suffit de taper `meshlab test.off` dans un terminal).

Organisez correctement votre code en créant vos fonctions liées aux maillages dans `mesh_functions.hpp` et `mesh_functions.cpp`.

Maintenant, à vous !

1 Parcourir le maillage : degrés, courbures et cartes associées

Exercice 1.

1. Commencez par écrire une fonction :

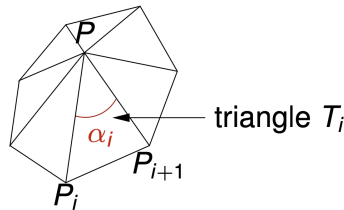
```
vector<Mesh::Vertex_index> vneigh (Mesh::Vertex_index v, Mesh &m)
```

renvoyant (en utilisant la structure de demi-arêtes et non les circulateurs) les indices des voisins du sommet v .

2. Définir une propriété entière `deg` attachée aux sommets et y calculer le degré de chaque sommet.
3. Créer une carte de couleurs par degrés (on associera une couleur à chaque degré)

Exercice 2 (Courbures discrètes : prototype simple).

Dans cet exercice, on souhaite estimer la courbure en chaque sommet du maillage. Pour cela, on utilisera un estimateur très simple de la courbure gaussienne au sommet P :



$$K_P = \frac{2\pi - \sum_i \alpha_i}{\sum_i \text{Aire}(T_i)}$$

$$\text{avec } \text{Aire}(T_i) = \frac{1}{2} \left\| \overrightarrow{PP_i} \wedge \overrightarrow{PP_{i+1}} \right\|$$

1. En partant des fonctions précédentes, écrire une fonction :

```
void gaussian_curv(Mesh &m, Mesh::Property_map<vertex_descriptor,double> &Kcourb) ;
```

prenant un maillage et une propriété permettant d'enregistrer les courbures gaussiennes et calculant et stockant la courbure en chaque sommet.

2. Ouvrez la classe `distrib` permettant de réaliser des statistiques de base sur un ensemble de valeurs. Comment la manipuler pour calculer les statistiques d'une distribution ?
3. En utilisant cette classe, calculer la moyenne ($kmean$), min ($kmin$), max ($kmax$), écart type ($kstdev$) de la courbure gaussienne sur l'ensemble des sommets.

4. On veut colorer les sommets selon leur courbure de telle sorte que :

$$\begin{aligned} K_P \in [kmin, kmean - kstdev] &\mapsto \text{bleu} \\ K_P \in [kmean - kstdev, kmean + kstdev] &\mapsto \text{couleur entre bleu et rouge} \\ K_P \in [kmean + kstdev, kmax] &\mapsto \text{rouge} \end{aligned}$$

Que fait le code ci-dessous :

```
CGAL::Color color_ramp(double vmin, double vmax, double mean, double stdev, double v, CGAL::Color col1, CGAL::Color col2)
{
    stdev = stdev/2 ;
    if (vmin < mean-stdev)
        vmin = mean-stdev ;
    if (vmax > mean+stdev)
        vmax = mean+stdev ;

    if (v < vmin)
        v = vmin ;
    else if (v > vmax)
        v = vmax ;
    double per ;
    CGAL::Color c ;
    CGAL::Color col_first, col_second ;
    if (v <= mean)
    {
        per = (v-vmin)/(mean-vmin) ;
        col_first = col1 ;
        col_second = CGAL::white() ;
    }
    else
    {
        per = (v-mean)/(vmax-mean) ;
        col_first = CGAL::white() ;
        col_second = col2 ;
    }

    c.red() = floor(col_first.red() + per*(col_second.red()-col_first.red())) ;
    c.green() = floor(col_first.green() + per*(col_second.green()-col_first.green())) ;
    c.blue() = floor(col_first.blue() + per*(col_second.blue()-col_first.blue())) ;
    return c ;
}
```

5. Construire la carte de couleur du maillage par la courbure gaussienne en chaque sommet.
6. Tester sur différents modèles et interpréter.