Department of Computer Science and Technology, Tsinghua University

计算机图形学大作业一：网格分割

刘泓尊　2022210866　计算机系

2023 年 1 月 13 日

复现 Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts 的网格分割算法，实现了 **层次化 K 路分解**。

1 层次化 K 路分解：原理说明

首先读入 Mesh 文件，根据公式

$$W_{ij} = \delta \frac{Geod(f_i, f_j)}{avg(Geod)} + (1-\delta)\frac{Ang\_Dist(\alpha_{ij})}{avg(Ang\_Dist)}$$

生成对偶图及面片之间的距离 (权重), 其中 $Ang\_Dist(\alpha_{ij}) = \eta(1-\cos\alpha_{ij})$; 之后在对偶图上通过 `dijkstra` 算法获得任意两点间的最短距离矩阵 $\{D_{ij}\}$。

对于层次化 K 路分解的每个递归节点，进行 K 路分解算法:

1. 选择初始代表面片

   1.1 选择第一个代表面片：它到所有点的最短路径之和最小。

   1.2 依次添加其他代表面片，每个面片到之前所有代表面片的最短路径是最大的。形式化地:

   $$REP_k = \arg\max_{REP_k} \min_{i<k} Dist(REP_i, REP_k)$$

   1.3 计算 $G(k) = \min_{i<k} Dist(REP_i, REP_k)$, 选择是的 $G(k) - G(k+1)$ 最大的 k 作为该步分解的数量。

2. fuzzy-kmeans 算法迭代优化 k 个代表面片，优化方程

   $$F = \sum_p \sum_f P(f \in patch(p)) Dist(f, p)$$

   其中

   $$P(f \in patch(p_j)) = \frac{\frac{1}{Dist(f, REP(p_j)}}{\sum_l \frac{1}{Dist(f, REP(p_l))}}$$

具体地，对于每个 patch p_i，选择 $\sum_f P(f \in patch(p_i))Dist(f, p_i)$ 最小的 f 作为该 patch 的新代表面片。迭代直到 F 的值不再降低或者代表面片集合不再变化。

3. 将每个面片划分到具体的 patch 或者模糊部分。

   3.1 如果面片 f_i 属于 S_j 的概率显著大于其他面片 (占比 50%) 以上，就认为 f_i 属于 S_j.

   3.2 否则，认为 f_i 属于某两个 patch 的模糊部分。这两个 patch 满足: (1) 连通性条件：和该模糊面片的最短路上不包含已经属于第三个 patch 的点；(2)f_i 属于这两个 patch 的概率是较大的两个。

4. 对于上述步骤建立的每一个模糊区域，基于公式

$$Cap_{ij} = \frac{1}{1 + \frac{Ang\_Dist(\alpha_{ij})}{avg(Ang\_Dist)}} \quad if\{i, j \neq S, T\} \quad else \quad \infty$$

   建立网络流图，通过 Ford Fulkerson 算法找到最小割，使得每个模糊区域的面片都有最终归属。

K 路分解执行完后，对于每个子区域对应的子图，进行递归分解，直到终止。

终止条件为: 代表面片之间的平均最短距离小于阈值 threshold.

2 代码结构说明

main.py 该文件作为顶层可执行文件,负责参数解析 (argparse) 和模块调用 (HierarchicalKwayDecomposer, KWayDecomposer) 的功能。程序运行参数如下:

```
$ python main.py -h
usage: main.py [-h] --file FILE [--eta ETA] [--delta DELTA] [--eps EPS]
[--maxiter MAXITER] [--display] [--outdir OUTDIR]
[--threshold THRESHOLD] [--hi]

Hierarchical Mesh Decomposition

optional arguments:
  -h, --help        show this help message and exit
  --file FILE       .ply file to be processed.
  --eta ETA         parameter `eta` for Hierarchical Mesh Decomposition.
  --delta DELTA     parameter `delta` for Hierarchical Mesh
                    Decomposition.
  --eps EPS         parameter `eps` for Hierarchical Mesh Decomposition.
  --maxiter MAXITER max iteration for K-means in Hierarchical Mesh
                    Decomposition.
  --display         whether to display the intermediate results.
  --outdir OUTDIR   output directory to save results.
```

```
  --threshold THRESHOLD
stop condition: Distance between representatives < threshold.
  --hi              use hierarchical kway decomposition.
```

MeshDecomposition/Mesh.py 包含 Mesh 模块，负责读入网格文件 (基于 trimesh 库), 然后根据权重公式建立对偶图。

MeshDecomposition/utils.py 包含随机生成颜色的函数 get_color，和导出.ply 文件的工具函数。

MeshDecomposition/Decomposition.py 包含 KWayDecomposer 模块，实现基本的 K 路分解算法。

dijkstra 函数用于生成对偶图任意两点间的距离矩阵；

init_representatives 函数将利用论文中的方法串行地选择 K 个初始代表面片；

fuzzy_kmeans 函数将基于初始代表面片进行迭代，最小化 F 的值，最终找到 K 个最终代表面片;

determine_fuzzy_patchs 函数负责基于前面找到的代表面片和相应的概率值，确定每个面片是属于哪个区域 (K_i)，或者属于哪两个区域之间的 fuzzy 区域；

build_flowgraph 函数将针对任意两个 patch 之间的 fuzzy 区域，建立网络流图，流量使用论文中给出的公式；

mincut 函数在上述生成的若干流图中执行 Ford Fulkerson 算法，找到最小割，确定每个面片的归属;

save_decomposition 函数对分好的区域进行随机染色，然后导出.ply 文件；

decompose 函数是上述流程的依次调用，以完成整个算法流水线。

MeshDecomposition/HierachicalDecomposition.py 包含 HierarchicalKwayDecomposer 模块，实现**层次化 K 路分解算法**。

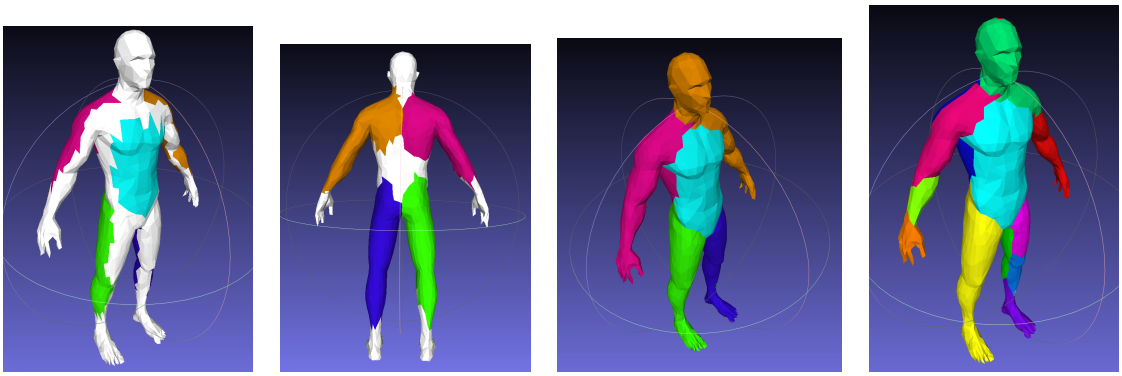
decompose_submesh 函数是递归函数,将针对传入的子图,依次执行 dijkstra、init_representatives、fuzzy_kmeans、determine_fuzzy_patchs、build_flowgraph、mincut 过程，然后根据归属矩阵将子图分解为 K_n 个子图 (n 代表递归树的某个节点)，并递归调用 decompose_submesh；递归返回后，将分割结果汇总并返回。

decompose 函数完成算法调用、随机染色和.ply 文件的导出。

3 运行结果分析

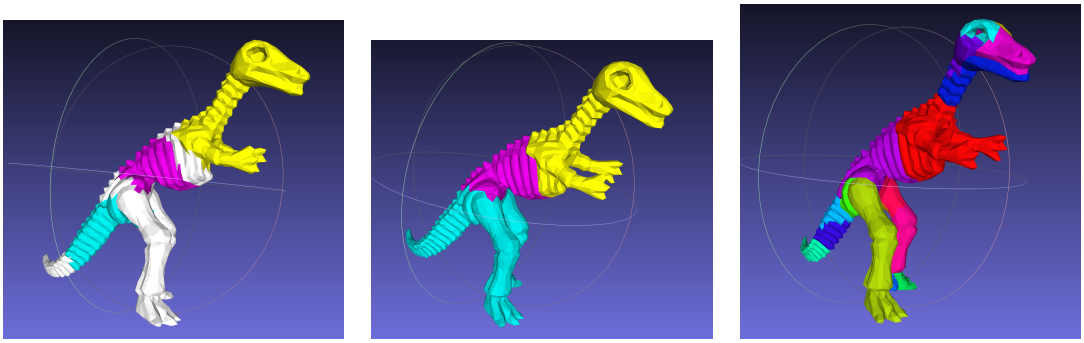
选择了 5 个样例输入：male.ply，pig,ply，horse.ply，dinosaur.ply，chair.ply。

对于这 5 个输入，其参数选择、K 路分解和层次化 K 路分解的结果为:



(a) Fuzzy (White Area) (b) Fuzzy (back) (c) K 路分解 (第一层的结果) (d) 层次化 K 路分解

图 1: Male 的算法输出 (参数 $\delta = 0.4$, 递归终止的最小平均距离 =8.0)



(a) Fuzzy (White Area) (b) K 路分解 (第一层的结果) (c) 层次化 K 路分解

图 2: Dinosaur 的算法输出 (参数 $\delta = 0.4$, 递归终止的最小平均距离 =6.0)

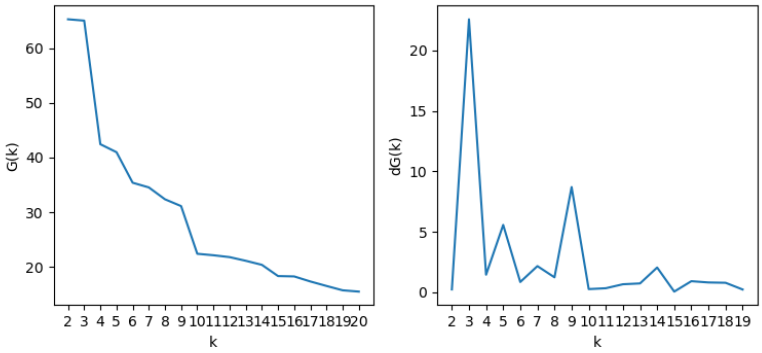
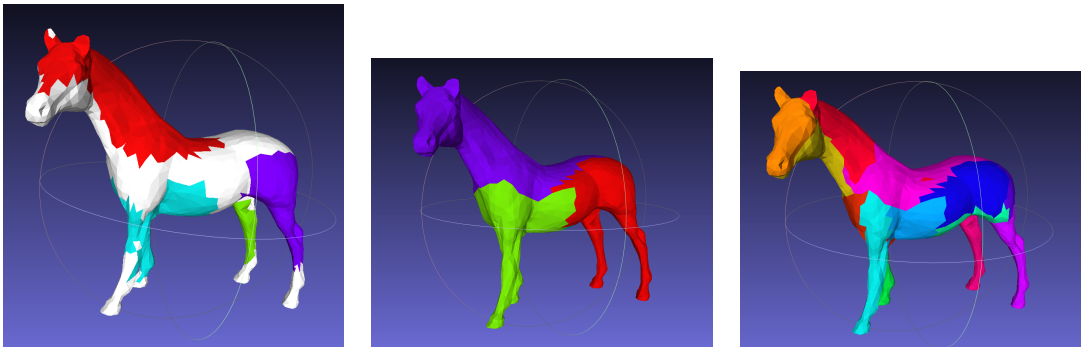
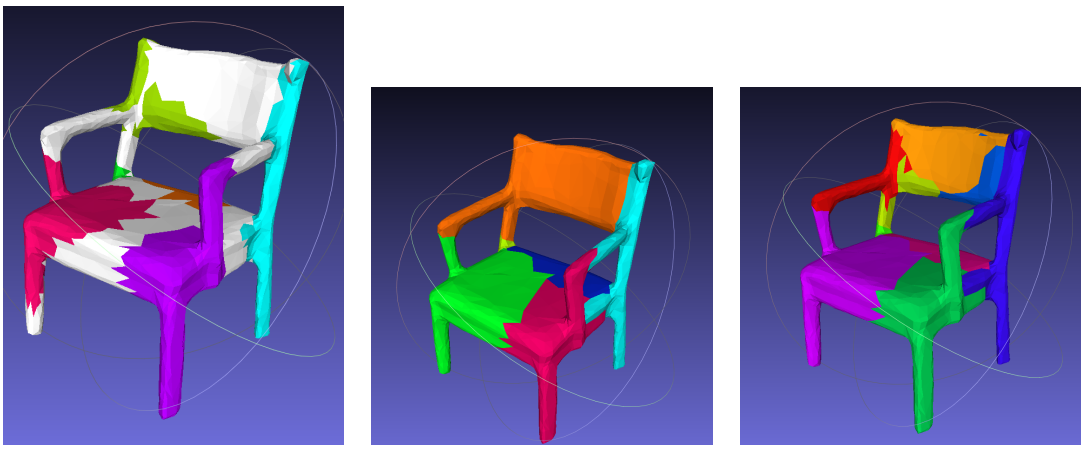


图 3: Dinosaur 的 $G(k)$ 和 $G'(k)$ (参数 $\delta = 0.4$, 层次化 K 路分解第一层)



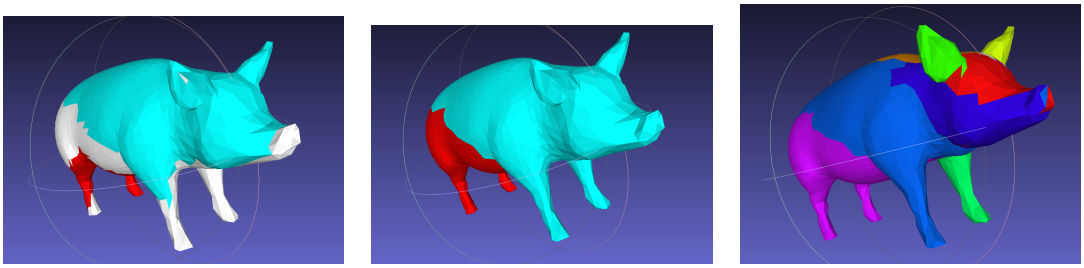
(a) Fuzzy (White Area)　(b) K 路分解 (第一层的结果)　(c) 层次化 K 路分解

图 4: Horse 的算法输出 (参数 $\delta = 0.5$, 递归终止的最小平均距离 =9.0)



(a) Fuzzy (White Area)　(b) K 路分解 (第一层的结果)　(c) 层次化 K 路分解

图 5: Chair 的算法输出 (参数 $\delta = 0.4$, 递归终止的最小平均距离 =7.0)



(a) Fuzzy (White Area)　(b) K 路分解 (第一层的结果)　(c) 层次化 K 路分解

图 6: Pig 的算法输出 (参数 $\delta = 0.5$, 递归终止的最小平均距离 =11.0)

从图6可以看到，层次化 K 路分解能够将区域进一步细分：基础的 K 路分解将模型分为 2 部分，之后层次化方法将 Pig 的每一条腿都分成了单独的区域，头和后背也分成了两个区域；

从图1、图2、图4可以看到，算法能很好的适配对称模型，并且能够将人和动物的四肢明确分开。对于 Dinosaur 模型，层次化分解算法进一步将上肢和头部分开了；图1显示，对于对

称模型，算法也能对称地划分区域。

从图5可以看到，算法能很好适配 Chair 这种高度复杂且不平滑的模型，能明确地将 Chair 的腿和扶手分成多个区域。

4 复杂度分析

对于有 V 个面片的网格模型, K 路分解复杂度为:

1. dijstra 算法: $O(V^2 \log V)$
2. 选择 K 个初始面片: $O(V^2 + KV)$, 其中 K 为划分数
3. fuzzy-keans 算法优化代表面片: $O(IV^2)$，其中 I 为迭代次数
4. 面片从属划分: $O(V + CV)$, 其中 C 是模糊区域面片数
5. Ford Fulkerson 最小割算法: $O(C^2 \log C)$

即 K 路分解复杂度为

$$O(V^2 \log V + V^2 + KV + IV^2 + V + CV + C^2 \log C) \sim O(V^2 \log V + IV^2)$$

对于层次化 K 路分解，如果有 N 个递归树节点，则整体复杂度为 $O\left(N(V^2 \log V + IV^2)\right)$