

# 计算机图形学网格简化作业报告

计算机科学与技术系 顾煜贤 2017011421

June 30, 2019

# Contents

1	编译运行	3
2	代码整体框架	3
3	问题描述	4
4	简化算法	4
4.1	二次误差计算 . . . . .	4
4.2	有连边点的合并 . . . . .	4
4.3	无连边点的合并 . . . . .	4
4.4	最小堆维护 . . . . .	5
4.5	防止面片反转、重叠 . . . . .	5
5	主要结果	6
5.1	Dragon . . . . .	6
5.2	Two Sphere . . . . .	6

## 1 编译运行

```
cd hw3/src
make clean
make
```

```
./main ../objs/your_input.obj ../results/your_output.obj 0.01 0.1
```

（命令行参数意义依次为：输入模型文件，输出模型文件，面的简化比， $t$ （后文会定义））

## 2 代码整体框架

本次作业主要部分由mesh.cpp构成，其中定义了网格化模型并实现了网格简化算法接口。输入输出支持obj文件格式，但无法包含纹理文件，总计1201行代码。

各文件功能如下：

- main.cpp: 主程序入口
- mesh.h mesh.cpp: 定义了网格模型类，并实现了网格模型的加载，简化，保存函数。
- map.hpp heap.hpp kdtree.hpp: 分别实现了hash表，小顶堆，kd树数据结构。
- vec3.hpp vec4.hpp mat4.hpp: 分别实现了三维向量，四维向量，四阶矩阵类，并实现相应的方法。
- vertex.h vertex.cpp: 面片的定点类。
- face.hpp: 三角面片类。
- pair.hpp: 实现算法中待合并的两点构成的pair。
- utils.hpp: 定义了一些常量、全局函数。

各对象主要关系如图1。

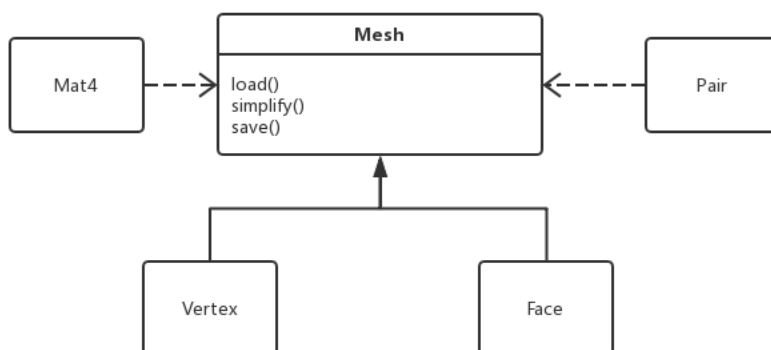


Figure 1: 各对象关系

### 3 问题描述

本次作业要求实现将三维三角面片模型进行网格简化的算法，要求网格质量以及误差较小，保持原图的拓扑结构，并且效率不能过低。

### 4 简化算法

本次作业主要算法为基于二次误差的网格简化算法[1]。

#### 4.1 二次误差计算

用一个点到以其为顶点的所有面的距离的平方和作为衡量网格变化的误差。设以点 $v$ 作为定点的面为 $p_1, p_2, \dots, p_k$ ，则点 $v$ 的误差：

$$\Delta(v) = \sum_{i=1}^N (v^T p_i)(p_i^T v) = \sum_{i=1}^N v^T (p_i p_i^T) v = v^T \left( \sum_{i=1}^N K_i \right) v = v^T Q v \quad (1)$$

可见，实际上仅用一个矩阵 $Q$ 便可刻画一个点与其周围所有面距离的平方和。若在简化过程中，两个点 $v_1, v_2$ 合成为一个点 $v$ ，则 $v$ 的二次误差矩阵 $Q = Q_1 + Q_2$ 。为使简化后网格变化最小，应使简化后点 $v$ 的二次误差 $\Delta(v) = v^T Q v$ 尽可能小。计算可得当 $\Delta(v)$ 极小时， $v$ 满足：

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} v = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2)$$

这里解出的 $v$ 即为简化后的点的位置，而 $Q$ 为简化后点的二次误差矩阵。

#### 4.2 有连边点的合并

若两个点 $v_1 v_2$ 之间有相连的边，则它们组成一个pair，这个pair可以简化为一个点 $v$ ，简化后的二次误差为此简化的 $cost = v^T Q v$ 。为保证最大程度上保留原模型形状，每次应选择 $cost$ 最小的pair进行简化。

#### 4.3 无连边点的合并

对于无连边的点，当两点之间的距离小于一个给定参数 $t$ 时，也将它们视为一个可以简化的pair，用上述同样的方法计算简化后点的位置、二次误差矩阵，以及简化 $cost$ ，这样的pair也参与到所有pair按照 $cost$ 的排序中。

为了查询与每个点距离小于 $t$ 的点，朴素算法是直接遍历求解。但是这样的时间复杂度为 $O(n^2)$ ，对于一个拥有10w量级顶点的模型来说，时间成本是不可接受的。因此采用kdtree实现 $t$ 范围查询，实现这一步算法的加速。需要注意的是，obj文件坐标大小在0~1之间，因此当 $t$ 接近1时，算法退化为遍历求解的情况。不过 $t$ 的设置本身是为了合并那些本身距离很近，但是没有连边的点，因此一般情况下， $t$ 很小，算法加速有效。

## 4.4 最小堆维护

为实现每次选择 $cost$ 最小的pair进行合并，将所有pair按 $cost$ 组织成一个小顶堆，支持添加pair (add)，删除堆顶pair(del)，访问堆顶pair(query)，以及更新部分pair(update)。前三种操作均为堆结构的常规操作。需要实现update操作的原因是，当合并一个pair时，原来两个点周围的点构成的pair的 $cost$ 值都将发生变化，有些将改变其在堆中的位置。

采用lazy维护的方式实现小顶堆的update操作。具体来讲，将update拆分成remove和add两个操作。add为常规添加操作，remove为在堆任意位置删除操作。remove堆节点A时，将其打上lazy标记，代表其实际已经被删除，同时在删除堆顶元素的时候，如果堆顶元素被lazy标记，则继续删除，直到没有lazy标记的节点到达堆顶。可以证明，这样堆中没有lazy标记的节点仍然满足堆序性。

## 4.5 防止面片反转、重叠

obj文件读入时，面片三个点之间有顺序，这个顺序确定了面片的法向量方向。而在简化过程中，一个面三个点的顺序可能会改变，造成面片的反转。观察算法过程可以发现，在网格简化过程中只有面的减少，没有面的增加，并且对于保留下来的面，其三个节点仅仅是位置发生了变化，节点编号的引用并未发生变化（但是引用顺序可能发生变化）。因此可以使用词典数据结构，通过简化后每个面三个顶点的编号找到初始模型中的对应面，根据初始模型对应面的节点编号顺序，确定简化后面节点的顺序。

具体地，定义散列函数

$$hash((i_1, i_2, i_3)) = ((i_1 + i_2 + i_3)P^2 + (t_1 * t_2 + t_2 * t_3 + t_1 * t_3)P + t_1 t_2 t_3) \pmod{M} \quad (3)$$

其中， $i_1, i_2, i_3$ 为每个面定点的编号，P取131313。上述函数将一个无序三元组映射到了一个 $[0, M)$ 的整数上。其中利用了三个数轮换对称的三种组合。

还有一种可能产生面片重叠/反转的情况，即简化过程中发生了不合理的边塌缩（图2）。明显的是，如果发

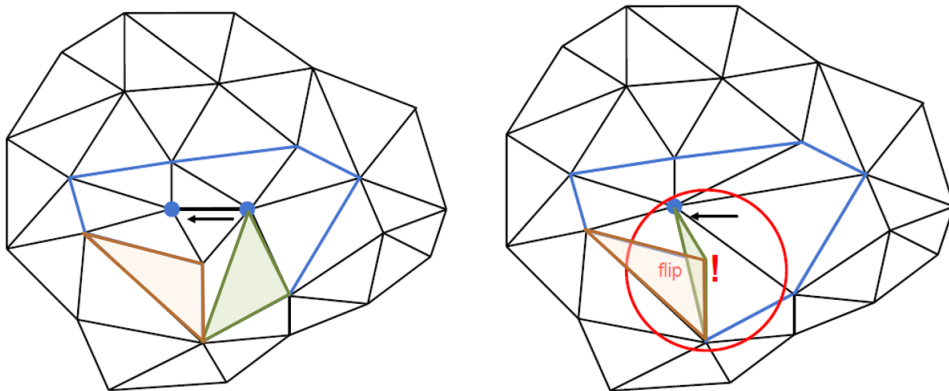


Figure 2: 简化过程中的面片重叠

生面片的重叠，则比会发生面片的反转（假设网格模型没有缺陷）。为了解决这种情况，每次进行边塌缩的时候分

别计算塌缩前后与发生变化的面的法向量，前面提到，简化过程中只会发生面片的减少，因此所有简化后的面都能对应到一个简化之前的面。通过比较它们之间的法向量之差，就可以限定在此步简化中，面能够多大程度被反转。

## 5 主要结果

这里展示两个代表性的结果，更多结果详见hw3/result

### 5.1 Dragon

简化一个龙的模型：fixed.perfect.dragon.100K.0.07.obj，各种参数的简化效率如表1。

简化比	t	用时(s)
0.5	0	4.42
0.1	0	5.73
0.01	0	5.99
0.01	0.005	6.92
0.01	0.01	9.63
0.02	0.02	29.67

Table 1: 简化效率

简化效果如图3

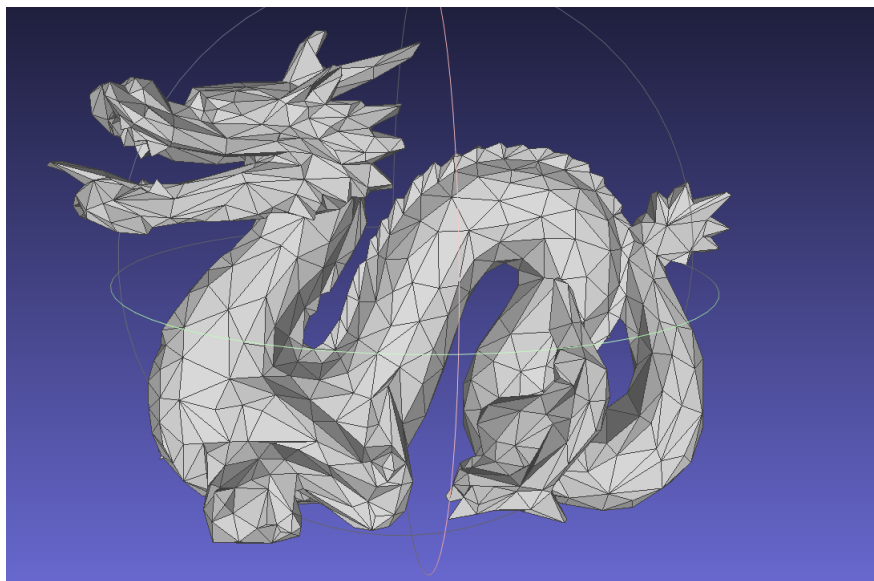


Figure 3: 简化效果

### 5.2 Two Sphere

$t$ 值的设置使简化过程中可以将两个原来没有没有连边的点连在一起，下图简化sphere-merge.obj模型，简化比0.05。原模型如图4，简化后如图5( $t = 0.5$ )与图6( $t = 0$ )。可见，加入 $t$ 之后简化可以将两个球连到一起。

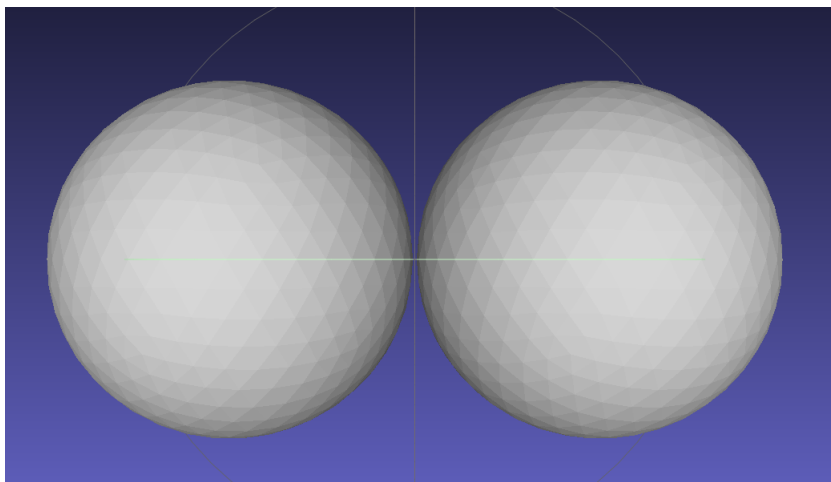


Figure 4: sphere-merge 原模型

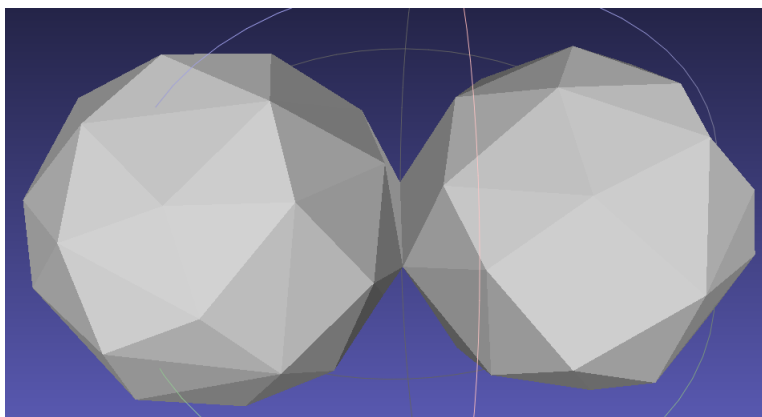


Figure 5: sphere-merge 简化后 ( $t=0.5$ )

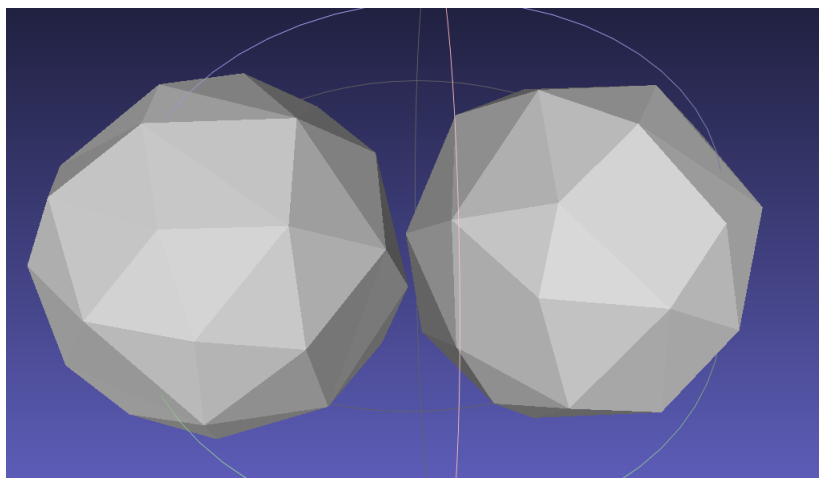


Figure 6: sphere-merge 简化后 ( $t=0$ )

## References

- [1] Garland, Michael, and P. S. Heckbert. "Simplifying Surfaces with Color and Texture Using Quadric Error Metrics." Visualization 1998.