

几何与建模方向实验报告

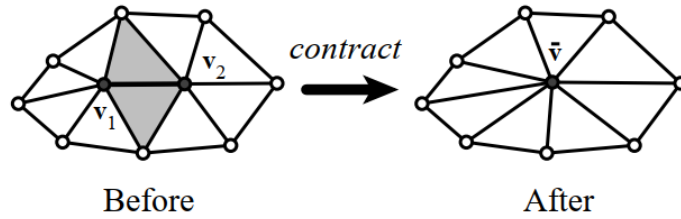
计试 001 苏悦馨 2204120515

1 算法流程简述

曲面简化问题是在尽量保持网格原有形态的情况下，减少网格中的顶点、边和面。本文采用基于二次误差度量 (QEM) 和边折叠操作的曲面简化算法^[1]。下面来分别阐述边折叠算法、QEM 算法以及最后的曲面简化流程。

1.1 边折叠算法

边折叠算法对应 *meshedit.cpp* 中的 *collapse_edge()* 函数。其所做操作为坍塌掉曲面中的一条边，删除多余的元素并重新连接曲面。如 [1] 中所示，将 v_1, v_2 之间的边 e 坍塌成点 \tilde{v} ，并将 v_1, v_2 的邻接点和点 \tilde{v} 相连，重新组成流形曲面。



1.2 QEM 算法

QEM 算法旨在计算折叠边后得到的点 \tilde{v} 的最优位置，并衡量出相应的代价。QEM 在求最优位置时，是从 \tilde{v} 离局部曲面位置尽可能近的角度来衡量的，具体算法如下：

- 1) 计算各面对应的矩阵^[2] $K = vv^T$ ，其中 $v = (\vec{n}, -\vec{n} \cdot p)$ ， \vec{n} 为平面法向量， p 为平面上一点。易证得点 (x, y, z) 到该平面的距离为 $(x, y, z, 1)^T K (x, y, z, 1)$
- 2) 定义各顶点 v_i 对应的矩阵^[2] 为 $K_i = \sum_{\text{all faces } ijk \text{ touching } v_i} K_{ijk}$
- 3) 对于待折叠边 e ，两个端点分别为 a, b ，折叠后得到的新顶点为 \tilde{v} 。则定义边折叠后得到新顶点的代价为 $(\tilde{v}, 1)^T (K_a + K_b) (\tilde{v}, 1)$ ，而 $\tilde{v} = \arg \min_v (v, 1)^T (K_a + K_b) (v, 1)$ 。新顶点对应的矩阵为 $K_a + K_b$ 。

1.3 曲面简化算法^[2]

基于以上的论述，下面来总体说明基于二次误差度量 (QEM) 和边折叠操作的曲面简化算法。

1. 由 QEM 算法得到各边折叠后对应的代价，放入最小堆中；
2. 取出最小堆堆顶的边 e ，判断其是否可以折叠，若不可，弹出该边直到取到最小堆堆顶可以折叠的边。设其两端点分别为 a, b ，从最小堆中删除 a, b 邻接的所有边；
3. 折叠边 e ，得到的新顶点 \tilde{v} ，重新计算 \tilde{v} 所有邻接边折叠对应的代价，放入最小堆中，回 2。直到流形网格的面数达到规定数目，算法停止。

2 程序效率的分析

下面推导该算法的计算复杂度，设某三角流形网格有 n 个面，设网格中点最多有 C 个邻接点

1. 为每个面计算矩阵 K 的复杂度为 $\theta(n)$
2. 为每个顶点计算矩阵 K 的复杂度为 $\theta(n)$
3. 为每一个边建立 *Edge_Record* 的总复杂度为 $\theta(n)$ ，将所有的 *Edge_Record* 结构逐个加入最小堆的复杂度为 $O(n \log n)$
4. 取出最小堆堆顶的边，判断其是否能删除需要 $O(C)$ 时间；从队列 *edge_queue* 和 *edge_records* 中删除边分别需要 $O(C \log n)$ 和 $O(C)$ 时间；将新生成顶点的邻接边加入队列 *edge_queue* 和 *edge_records*，分别需要 $O(C \log n)$ 和 $O(C)$ 时间，该步骤重复 $\frac{3n}{4}$ 次，故总共时间复杂度为 $O(Cn \log n)$

故该算法的时间复杂度为 $O(Cn \log n)$

3 测试结果

测试数据来源于 Scotty3d 内包含的 dae 文件和 Scotty3d 内置模型，随 Scotty3d 一同从思源学堂下载，具体包括如下：bunny.dae, cow.dae, Scotty3d 内置的 Torus，网格规模为 10^3 数量级。

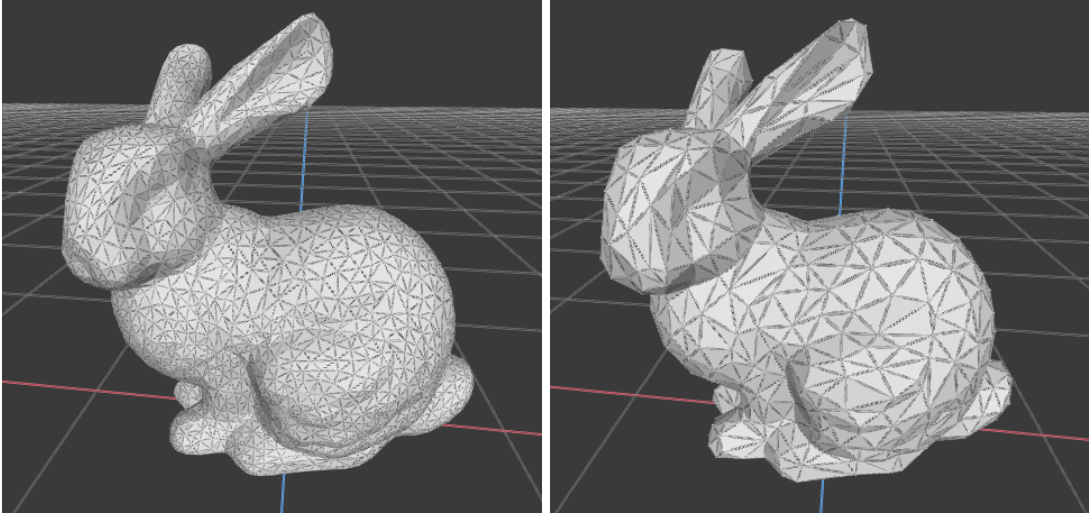


图 1: Scotty3D:bunny

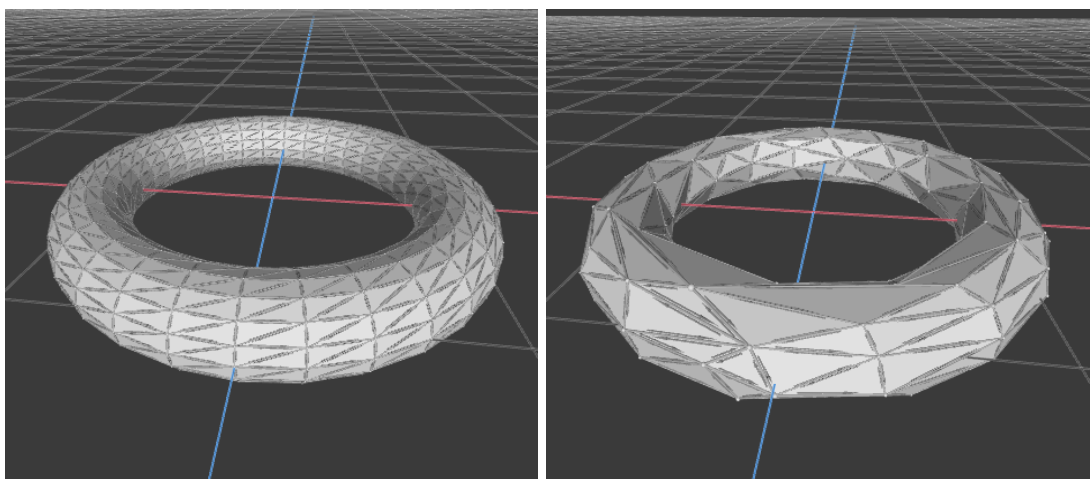


图 2: Scotty3D:Torus

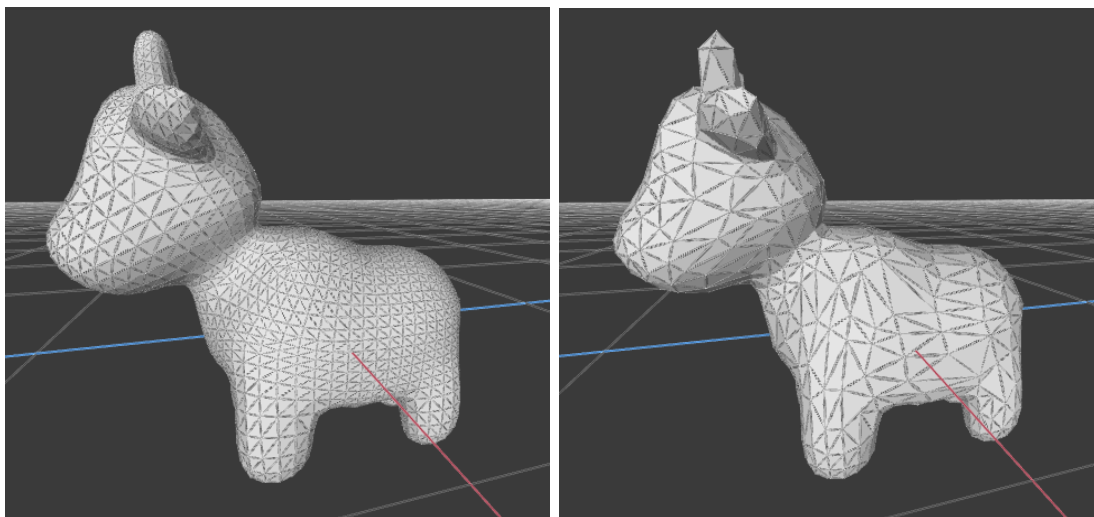


图 3: Scotty3D:Cow

4 Reference

- [1] Garland M , Heckbert P S . Surface simplification using quadric error metrics[J]. ACM SIG-GRAPH Computer Graphics, 1997, 1997:209-216.
- [2] Scotty3D Developer Manual: Simplification
- [3] https://en.cppreference.com/w/cpp/container/unordered_map
- [4] https://en.cppreference.com/w/cpp/container/unordered_set