DGP 第一次作业：网格简化

SA22001009 陈泽豪

March 10, 2024

1 Introduction

四边形误差度量（Quadric Error Metrics，简称 QEM）是计算机图形学领域中用于网格简化的一种高级算法。该算法由 Michael Garland 和 Paul S. Heckbert 在 1997 年的论文《使用四边形误差度量的表面简化》中提出，自此，QEM 算法成为 3D 模型处理领域的基石之一。这一算法的主要目标是在尽可能保持原始外观的前提下降低 3D 网格的复杂度，这对于视频游戏和虚拟现实应用中的渲染效率和实时性能尤为重要。

QEM 背后的核心思想是将网格表面的近似误差表示为点到表面的平方距离（四边形误差）之和。这些误差被封装在称为四边形的数学结构中，它本质上是一个与网格中每个顶点关联的 4x4 对称矩阵。QEM 能够通过评估这种折叠引起的四边形误差，来计算折叠任何给定边的成本。这个成本代表了边折叠对整体网格形状的影响，允许算法优先考虑哪些边进行折叠，以实现最有效的简化，同时最小化细节损失。

在实际应用中，QEM 已被广泛用于从游戏和仿真中模型的渲染优化到工程中的有限元分析模型预处理等多种场景。其影响力不仅限于简化，还推动了网格修复、重网格和多分辨率建模技术的进步。

总之，四边形误差度量革新了网格简化的方法，提供了一种既稳健、高效又通用的减少 3D 模型复杂度的方法。其开发标志着计算机图形学中的一个重要里程碑，并继续激发网格处理技术的改进和创新。

2 整体的算法框架

整个代码的实现主体实际上可以分为如下的几步。

2.1 计算每个点上的 Q 矩阵

在四边形误差度量（QEM）算法中，Q 矩阵是核心概念之一，用于表示顶点周围局部几何误差的累积。Q 矩阵的计算主要基于顶点周围的面的平面方程。每个面的平面方程可以表示为 $ax+by+cz+d=0$ 的形式，其中 (a,b,c) 是面的单位法向量，d 是平面到原点的距离，而 (x,y,z) 是平面上任意一点的坐标。

对于给定顶点 v 的 Q 矩阵，其计算步骤如下：

- 对于顶点 v 的每个邻接面，首先计算该面的平面方程 $ax+by+cz+d=0$。
- 构建面的四维向量 $p=[a,b,c,d]$，并计算该向量的外积，即 $p*p^T$，得到一个 4x4 的矩阵。

- 累加：对于顶点 v 的每个邻接面，将步骤 2 中得到的 4x4 矩阵相加，累积的结果就是顶点 v 的 Q 矩阵。

2.2 计算每条边的 cost，并得到最优合并点

对于每条边 e，连接顶点 v_1, v_2，其代价和最优化点计算如下：

- 合并 Q 矩阵：计算边两端顶点的 Q 矩阵之和，即有 $Q_e = Q_{v_1} + Q_{v_2}$。
- 计算折叠代价：代价是折叠后顶点到所有相关平面的距离平方和的最小值。假设折叠后的顶点位置为 v，计算代价为 vQ_ev^T。
- 最优化点的求解：最优化点是使得代价最小的 v。通常，通过求解 $Q_ev = 0$ 来找到，这个最优化点。在某些情况下，Q_e 是不可逆的，此时可以选择边的中点或其他启发式方法确定 v。

2.3 取出 cost 最小的边，进行边的 collapse

这一步进行完 collapse 之后还要进行当前最新点的 1 邻域的边上的 cost 重新计算。

3 最终结果展示

使用的为实验室框架，直接点击运行并放入图片，如下所示，点减少一半的效果如下：

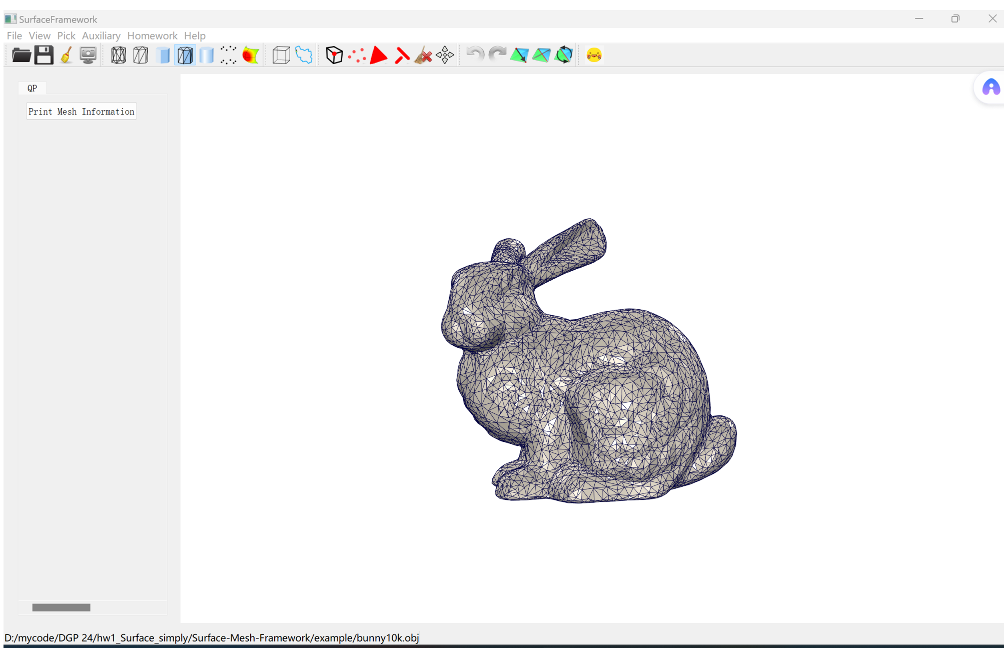


Figure 1: 打开示意图

然后通过设计的 QEM 算法 ui 进行实验，直接点击右上角的笑脸：

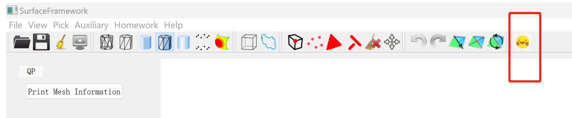
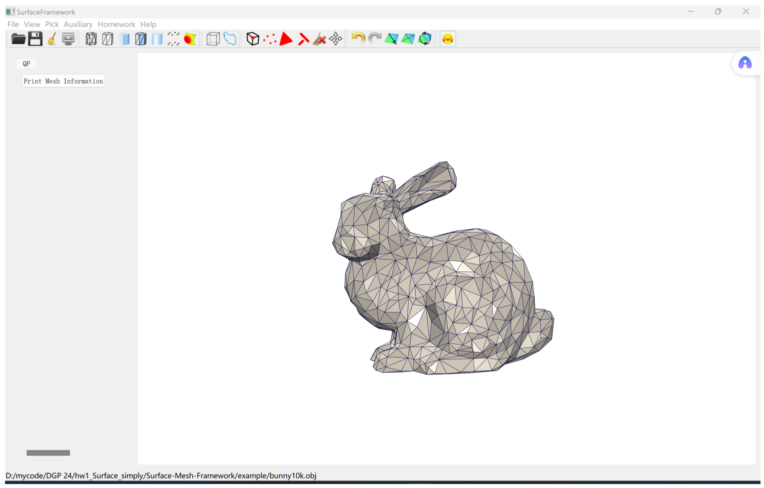
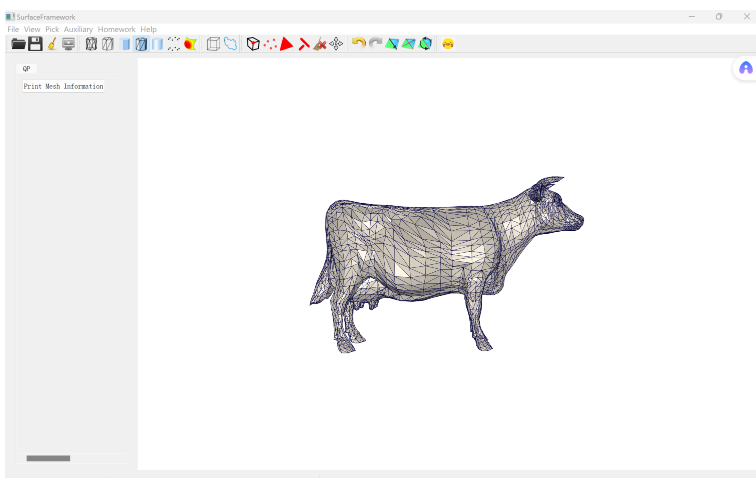


Figure 2: 具体按钮

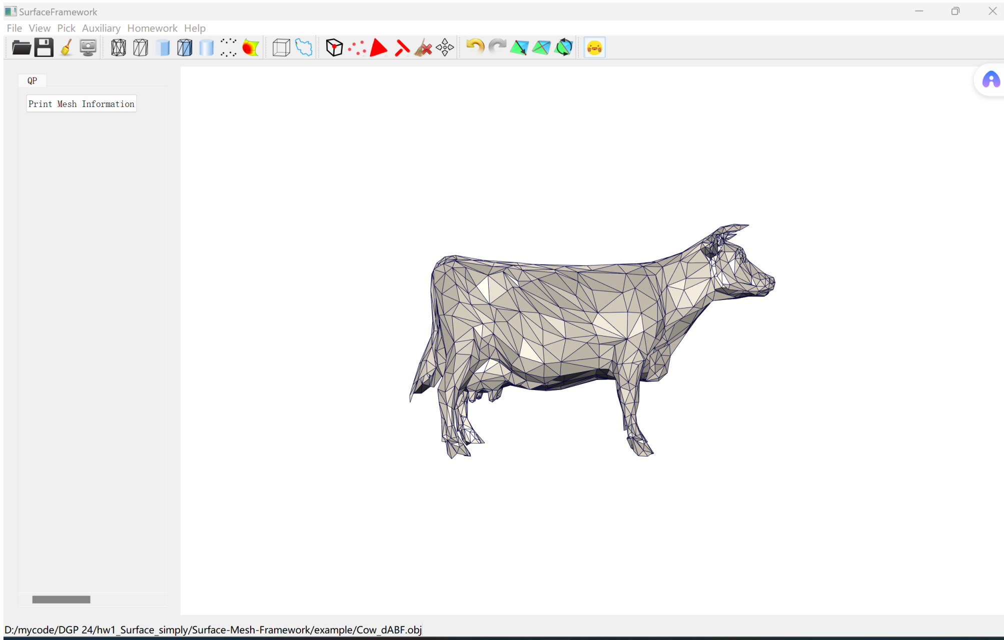
就可以得到结果如下：



我们在此基础上再给一个模型演示如下：



他得到的结果为：



4 总结

通过对比不同模型的简化结果，我们发现 QEM 算法在处理包含大量细节的复杂模型时特别有效，能够在不显著影响视觉质量的前提下实现高比例的简化。此外，算法的性能在不同类型的模型上表现出一致性，说明了其广泛的适用性和稳定性。