

网格简化总结报告

2012011372 赖国堃

2014/07

目录

1	程序描述	1
2	程序类介绍	1
2.1	Class Vec3f	1
2.2	Class simpleObject	1
2.3	Class modelsimplify	1
3	程序算法描述	1
4	程序亮点	2
5	程序运行结果	2
6	总结与收获	4
6.1	遇到的困难	4
6.2	收获	4

1 程序描述

本程序实现了基于边坍塌的网格简化算法，使用了堆优化，对于输入一个obj以及简化比例，即可得到一个新的简化过得obj文件。

2 程序类介绍

2.1 Class Vec3f

基础部件类，其内容为一个3维向量，重载了有关向量的基本运算。

2.2 Class simpleObject

提供读入obj操作，以及写入obj操作。

2.3 Class modelsimplify

主要功能即为网格简化，提供模型以及简化比例，返回新的模型。

3 程序算法描述

主要采取边坍塌算法，计算出每个点的Q矩阵之后，利用高斯消元方法，解出导数方程，求出新点以及耗费值。每次选取最小耗费的一条边删除。

之后加入了堆优化加速，我具体实现思路是，堆中存储每个点所对应最优的边及其耗费。当一个新点被添加之后他周围一圈点的Q值将会发生变化，那么他周围两圈内的耗费都将会发生变化。接下来我就将其两圈内的点从堆中删除，然后重新计算，这两圈点对应的最优值，重新添加入堆。这样保证了每次的修改操作复杂度为 $O(\log n)$ 。

4 程序亮点

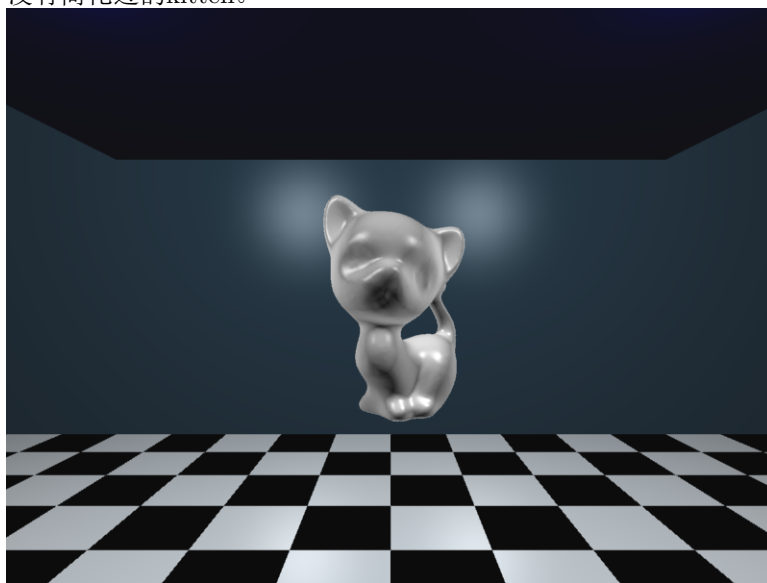
采用了高斯消元解法求新点，并没有简单的取中点，使得每次简化花费的耗散值减小，最后模型更加接近初始模型。

采用堆优化速度后，程序速度很快，5W个面片的kitten，简化比例0.5，耗时4025ms。20W个面片的中国龙，简化比例0.1时，只需要耗时29072ms，具体效果在程序结果中演示。

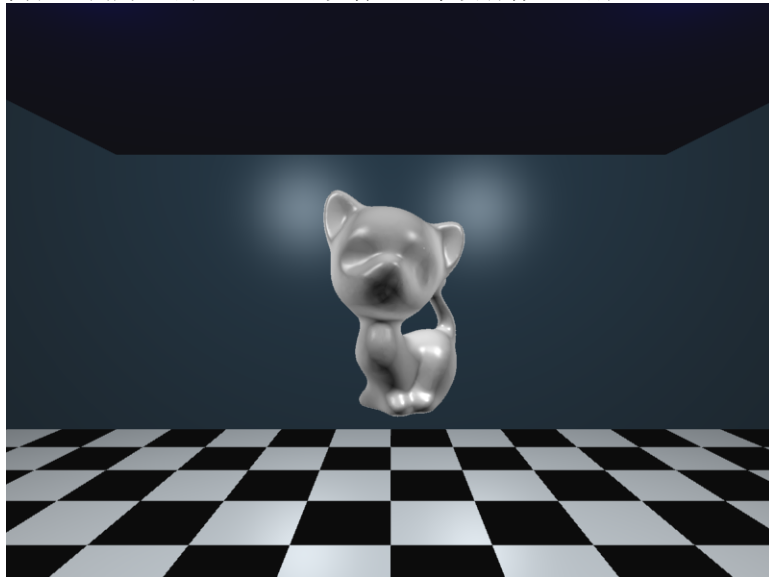
对于模型的显示是使用自己编写的光线追踪程序。

5 程序运行结果

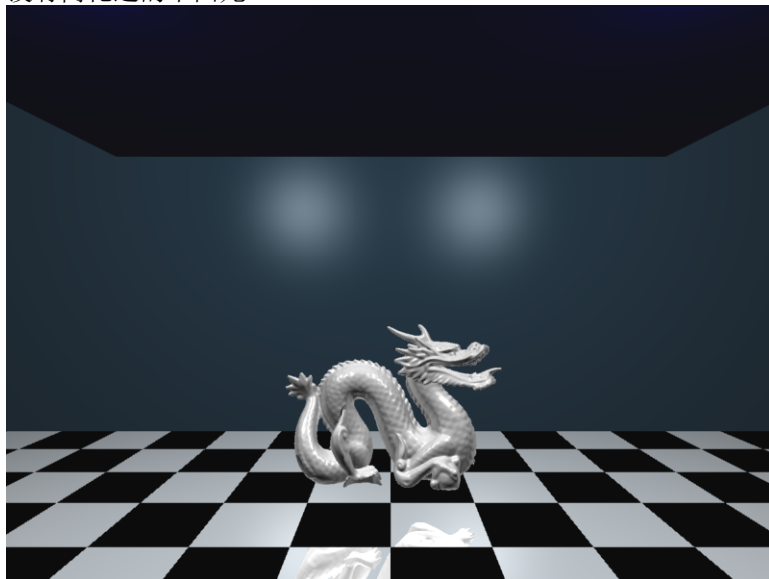
没有简化过的kitten。



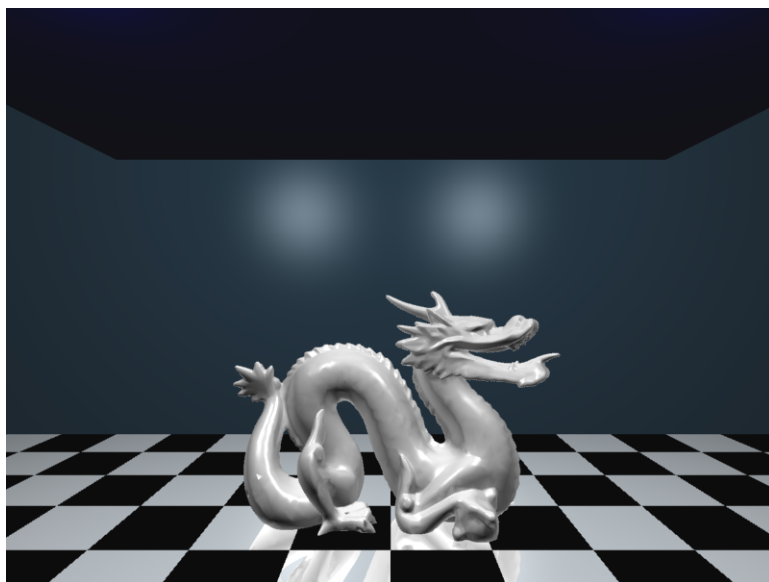
简化比例为0.5的kitten，可以看出基本没有什么差别。



没有简化过的中国龙。



简化比例为0.1的中国龙，明显身上的鳞片被简化，但是其他没有什么变化。



6 总结与收获

6.1 遇到的困难

调试过程中遇到了很多错误，主要是关于删除一个点之后，如何维护整个obj的信息，花了很长时间考虑了很多情况才解决。

6.2 收获

了解了网格简化的基本方法，锻炼了代码能力。