基于二次误差的网格简化

于雪璐　计科 00　2010011387

实验目的

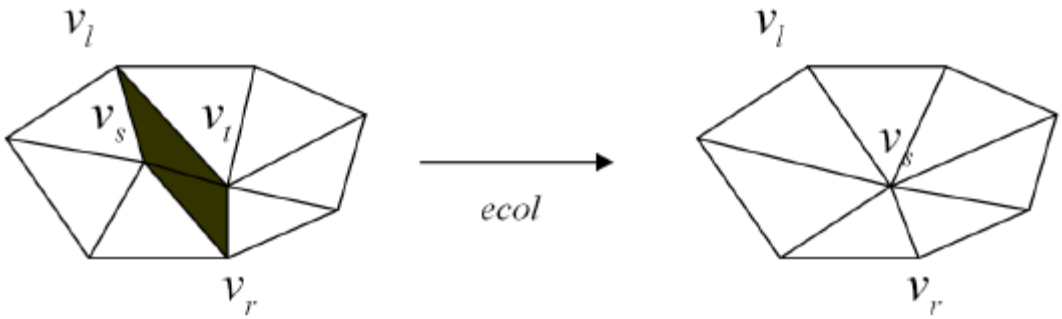
实现边坍塌（edge-collapse）的网格简化方法。

实验内容

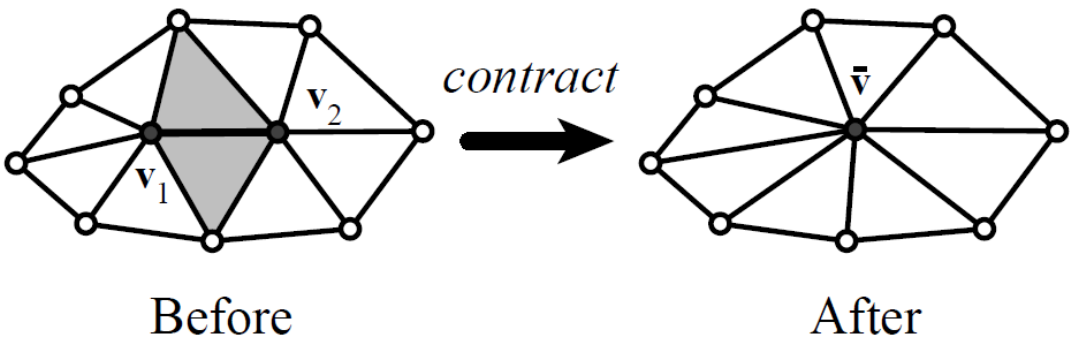
实现基于二次误差的边坍塌网格简化方法，采用基于 binary min-heap 和 ELF-hash 的优先队列来维护边收缩的误差队列。程序能指定输入输出的 obj 文件，以及面数的简化比（输出面数占输入面数的百分比），例如命令行程序可以支持如下参数
mesh_simp.exe　输入.obj　输出.obj　简化比（例如 0.3）。

实验原理

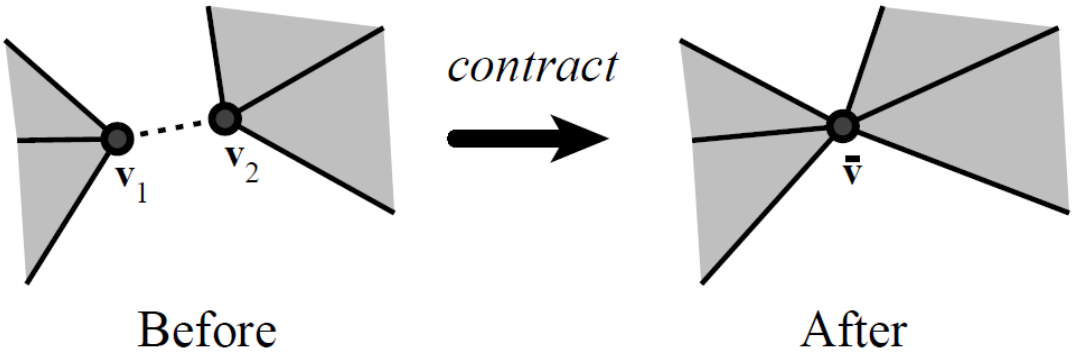
Hoppe 的边收缩(edge collapse)操作可推广为一般的顶点合并变换来描述(v_1,v_2)→v，其含义是将场景中的两个 v_1 顶点 v_2 移到一新的位置 v，将连向 v_1,v_2 的所有边都连向 v_1，并删除所有退化的边和面片。



点对(v_1,v_2)合并的原则
(1)v_1,v_2为某一表面上的相邻点，即v_1,v_2为一条边



(2) v_1-v_2<t， t为用户给定的阈值参数



Garland 和 Heckbert 引进了二次误差度量来刻划每一个顶点移动后引起的误差。对表面上的每一个顶点 v_a 均有许多三角面片与之相邻，记 plane(v_a) 为这些三角形所在的平面方程所构成的集合，即

$$\text{plane}(v_a)=\begin{cases}(a,b,c,d)|ax+by+cz+d=0,\\(a^2+b^2+c^2=1)\text{ is coefficients of the adjacent plane of }v_a\end{cases}$$

则我们采用如下的二次函数来度量 v_a 移动到 v 时产生的误差：

$$\Delta(v_a\rightarrow v)=\sum_{p\in plane(v_a)}(pv^T)^2$$

其中 v=(x， y， z， 1) 为齐次坐标。展开上式得到

$$\Delta(v_a\rightarrow v)=\sum_{p\in plane(v_a)}(pv^T)^2=v\left(\sum_{p\in plane(v_a)}K_p\right)v^T=vQ(v_a)v^T$$

其中

$$K_p=p^Tp=\begin{bmatrix}a^2&ab&ac&ad\\ab&b^2&bc&bd\\ac&bc&c^2&cd\\ad&bd&cd&d^2\end{bmatrix}$$

$$Q(v_a)=\sum_{p\in plane(v_a)}K_p$$

这样，对每一顶点v，在预处理时，我们均可按上述方法计算矩阵Q(v_a)，进而就可对其移动进行误差度量了。但由于每次合并时，需同时移动两点，故必须考虑同时移动多个顶点后形成的误差。

Garland和Heckbert简单地采用加法规则来刻划多点移动而形成的误差，对点对合并(v_1,v_2)→v，其误差为$\Delta v=\Delta(v_1\rightarrow v)+\Delta(v_2\rightarrow v)=v(Q(v_1)+Q(v_2))v^T=vQv^T$

其中 Q=Q(v_1)+Q(v_2)。因而，应选取 v 使误差达到最小。

由极值的性质知，v 满足系统方程：

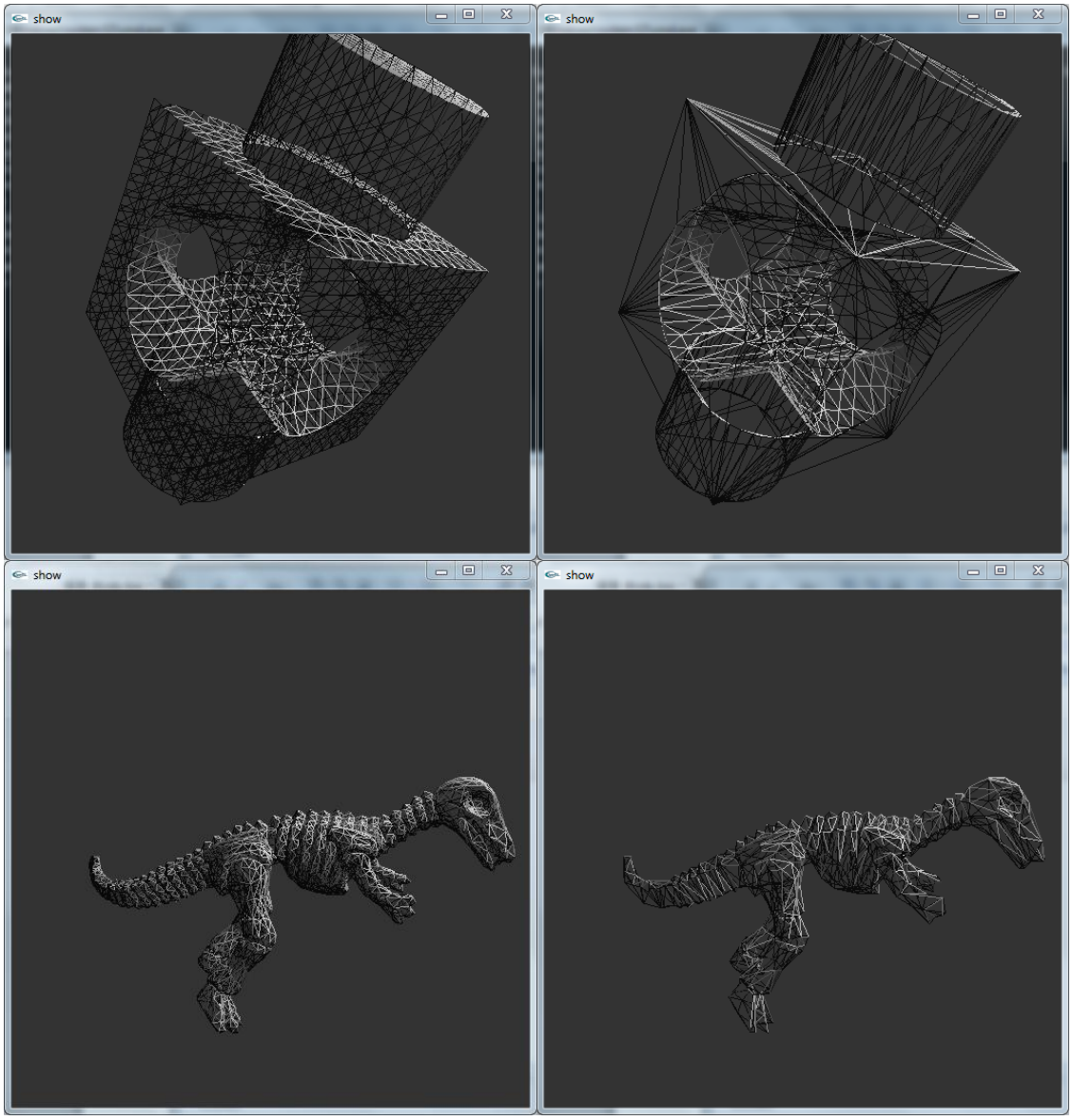
$$\frac{\partial\Delta(v)}{\partial x}=\frac{\partial\Delta(v)}{\partial y}=\frac{\partial\Delta(v)}{\partial z}=0$$

即

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

实验效果

以下均为简化比 0.3 的简化效果



实验总结

本次试验首先需要找到一个好的算法来进行简化，我经过多番查找确定使用基于二次误差测度的网格简化。在数据结构方面请教了同学，提高速度。最终编写之后出现很多 bug，调试的很痛苦，最终可以运行的较好。至今仍有一些偶尔简化过慢的情况难以解决。