

# 使用二次误差矩阵进行表面简化

小组成员：张宏官（006118）

胡小龙（006099）

CAD 中心

## 一、 背景

许多计算机图形应用要求复杂的高度细节化的模型来保持其逼真度。因此产生了大量高解析度模型。然而相当多的领域没有必要使用这么复杂的模型，或者想要控制算法的计算时间，他们需要一些简化的模型。表面简化算法就是要完成这样的工作。

目前的表面简化算法有很多种。我们的大作业中采用 Michael Garland 和 Paul S. Heckbert[1]的二次误差矩阵算法。此算法与其他算法相比具有以下三个优点：

- 效率高。它能迅速完成对一个复杂模型的简化；
- 质量好。简化后的模型能保留原模型的重要特征，与原模型非常相似；
- 普遍性。它具有聚合（即将不连通的区域连结起来）的特性。这个特性是其他算法所不具备的。

我们大作业的任务就是将此算法在计算机上加以实现。

## 二、 算法的基本思想

我们的简化算法是基于顶点对收缩的迭代过程。顶点对包括边对和非边对，如图 1 和图 2 所示。

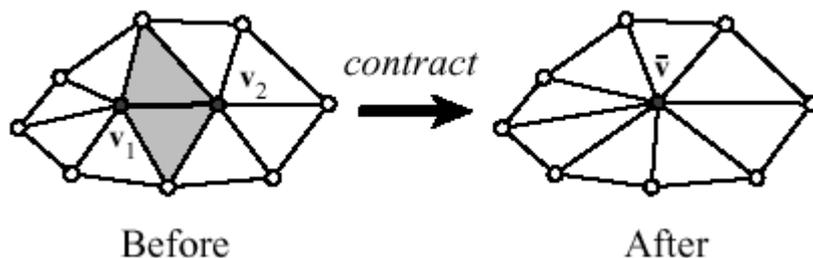


图 1 边对收缩

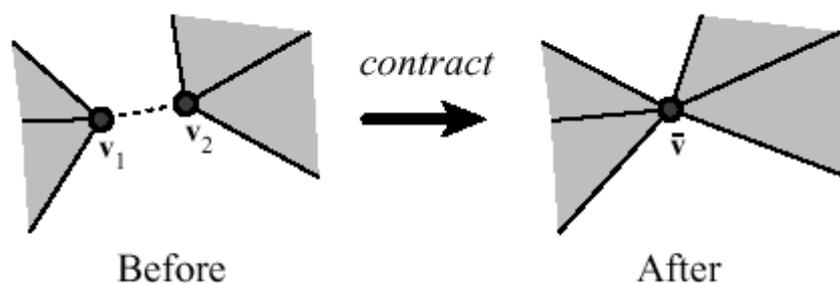


图2 非边对收缩

初始化模型为  $M_n$ ，从模型中挑选出所有用于迭代的顶点对（包括边对和非边对）。在每次迭代中，从顶点对中按照某种原则选出一个顶点对，将此两顶点合并为一个顶点，与原顶点相对应的边、面也相应的进行合并。经过若干次迭代，得到简化的模型  $M_g$ 。在迭代过程中，每迭代一次即产生一个中间模型，因此，该算法产生一系列模型： $M_n, M_{n-1}, \dots, M_g$ 。图3为一个简单的例子。

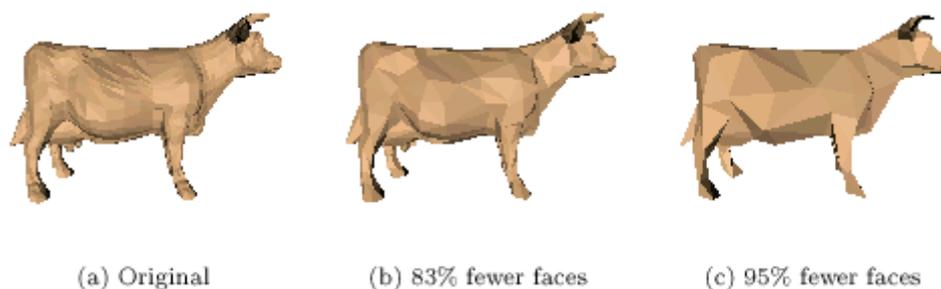


图3

### 三、算法的实现

#### (一) 算法步骤

1. Select a set of candidate vertex pairs;
2. Assign a cost of contraction to each candidate;
3. Place all candidates in a heap keyed on cost with the minimum cost pair at the top;
4. Repeat until the desired approximation is reached:
  - (a) Remove the pair  $(v_i, v_j)$  of least cost from the heap
  - (b) Contract this pair
  - (c) Update costs of all candidate pairs involving  $v_i$

## (二) 顶点对的选择

算法的第一步就是选择顶点对。最简单的做法就是选择初始模型  $M_0$  中所有的边对，这对简单的连通模型也许是最好的方法。但是对于那些由若干个部分组成的模型就不太合适了，此时考虑非边对的收缩可以产生更好的近似，图 4 就是一个简单的例子。

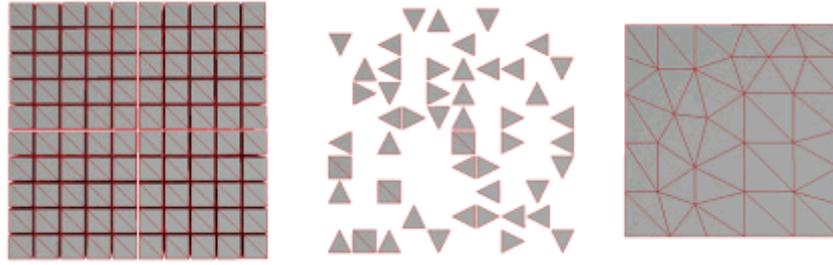


图 4

一个顶点对  $(v_i, v_j)$  只要满足下列条件之一，就是一个合法的顶点对：

1.  $(v_i, v_j)$  is an edge, or
2.  $(v_i, v_j)$  is not an edge and  $\|v_i - v_j\| < \tau$

$\tau$  为用户指定的一个的阈值。特殊情况，当  $\tau = 0$  时，即得到一个简单的边收缩算法。较高的阈值允许互不相连的顶点配对。很显然我们在选择  $\tau$  时要十分慎重，如果阈值  $\tau$  取得过高，会导致在大范围内模型分离的部分连接起来，而这并不是我们所想要的，最差的情况下会产生  $O(n^2)$  条收缩边。

## (三) 二次误差矩阵

### (1) 新收缩点的位置

算法的第二步就是选择用于迭代的收缩边，这也是该算法的关键所在。不同的顶点对收缩的结果对模型结构产生的影响是不一样的，有的变化大，有的变化很小，即收缩是要付出一定的代价 (cost) 的。算法为每一顶点对计算收缩成本 (contraction cost)，每次收缩成本最低的顶点对。为了计算收缩成本，在每一顶点处引入误差矩阵  $Q$  ( $4 \times 4$  阶)，定义顶点  $v = [v_x, v_y, v_z, 1]^T$  处的误差为二次型  $\Delta(v) = v^T Q v$ ，算法名称即由此而来。对于一个给定的收缩  $(v_i, v_j) \rightarrow \bar{v}$ ，我们必须得到一个新的误差矩阵  $\bar{Q}$  来近似顶点  $\bar{v}$  处的误差，在该算法中采用简单相加的原则，即  $\bar{Q} = Q_1 + Q_2$ 。顶点  $\bar{v}$  处的误差二次型为  $\Delta(\bar{v}) = \bar{v}^T (Q_1 + Q_2) \bar{v}$ 。

为了执行收缩  $(v_i, v_j) \rightarrow \bar{v}$ ，必须知道新顶点  $\bar{v}$  的位置，简单的做法就是选择  $v_1$ 、 $v_2$  或者  $(v_1 + v_2)/2$  作为新顶点的位置，具体选择哪一个取决于使  $\Delta(\bar{v})$  最小。

但是更一般的做法是对  $\Delta(\bar{v})$  求偏导得到最小值。由  $\frac{\partial \Delta}{\partial x} = \frac{\partial \Delta}{\partial y} = \frac{\partial \Delta}{\partial z} = 0$  得到：

$$\bar{v} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

(2) 误差矩阵 Q 的计算

$$Q = \sum_{P \in \text{Planes}(v)} K_p, \quad K_p = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

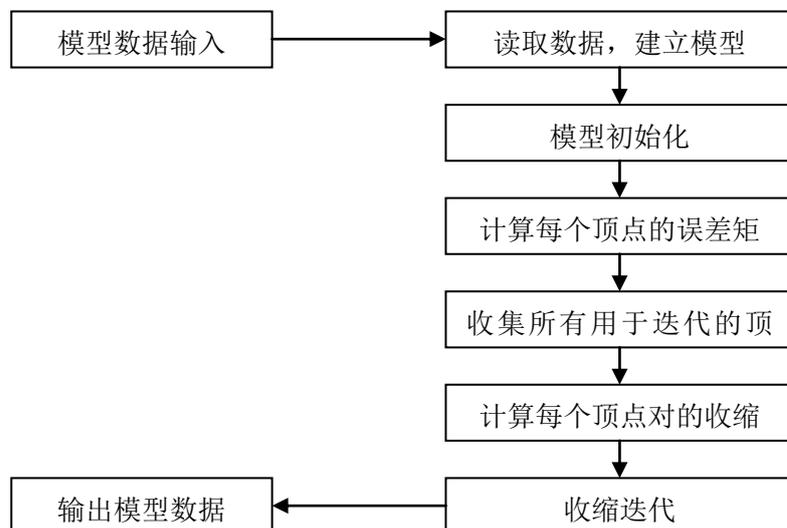
$\text{Planes}(v)$  为交于顶点  $v$  的所有平面的集合。平面  $P$  的矢量表示形式为  $n^T v + d = 0$ ，其中  $n = [a, b, c]^T$  并且  $a^2 + b^2 + c^2 + d^2 = 1$ 。对 Q 的详细解释见参考文献[1]。

#### (四) 程序实现

(1) 输入模型

算法的输入模型是已经经过三角剖分的表面模型。模型数据由顶点和三角面组成，顶点用  $x$ 、 $y$ 、 $z$  坐标表示，三角面由构成它的三个顶点的序号表示。

(2) 程序流程图



### (3) 应用举例

以一个牛的模型为例，该模型共有 2904 个顶点，5804 个面。在 CPU 为赛扬 450、256M 内存的机器上计算，计算时间如下表：

目标面	$\tau$	建模时间(s)	初始化时间(s)	简化时间(s)
1000	0	0.331	0.420	0.952
100	0	0.281	0.460	1.122
0	0	0.270	0.471	1.131

### 参考文献：

- [1] Surface Simplification Using Quadric Error Metrics, Michael Garland, Paul S. Heckbert.
- [2] Quadric-Based Polygonal Surface Simplification, Michael Garland.