

## 基于控制点误差控制的网格简化算法

朱经纬<sup>1,2</sup>, 王 乘<sup>1</sup>, 蒙培生<sup>1</sup>

(1. 华中科技大学 力学系, 湖北 武汉 430074;

2. 解放军理工大学 工程兵工程学院, 江苏 南京 210007)

(zhu\_jingwei@163.com)

**摘 要:**提出了一种基于控制点误差控制的网格简化算法,以初始网格三角形的中心点作为第一类控制点,以特征边的顶点作为第二类控制点,控制点与受控三角形之间的距离作为简化误差。根据设定的三角形权重,按照顺序进行三角形折叠操作,简化操作后必须满足控制点到受控三角形的距离小于阈值。

**关键词:**控制点;三角形折叠;误差控制

**中图分类号:** TP391.41 **文献标识码:** A

## Mesh simplification algorithm based on control point error control

ZHU Jing-wei<sup>1,2</sup>, WANG Cheng<sup>1</sup>, MENG Pei-sheng<sup>1</sup>

(1. Department of Mechanics, Huazhong University of Science & Technology, Wuhan Hubei 430074, China;

2. Engineering Institute of Engineering Corps, PLA University of Science & Technology, Nanjing Jiangsu 210007, China)

**Abstract:** In this paper, a mesh simplification algorithm based on control point error control was introduced. The first control points are the center points of the initial mesh triangles. The second control points are the vertexes of the characteristic lines. The distance between the control point to the under control triangle is the error between the initial mesh and the simplified mesh. The triangles are collapsed in order according to the weights of the triangles. The triangle collapse operation should meet the requirement that the distances between the control points and the under control triangles are less than the threshold value. Finally the validity of this algorithm is verified by the experiment.

**Key words:** control point; triangle collapse; error control

## 0 引言

三角形网格简化算法是在尽量保持模型的几何形状和拓扑结构的基础上,通过简化网格三角形和顶点,用简化模型替代初始模型。根据不同简化原理有多种简化算法:区域合并算法<sup>[1]</sup>,小波分解算法<sup>[2]</sup>,基于能量优化的边折叠算法<sup>[3]</sup>,基于二次测量误差的简化算法<sup>[4]</sup>,引入尖特征度的边折叠算法<sup>[5]</sup>,基于二次曲面拟合的三角形折叠算法<sup>[6]</sup>等。

本文提出一种以三角形折叠为基本操作的网格简化算法,以初始网格的三角形的中心点作为第一类控制点,以特征边的顶点作为第二类控制点,在网格内每个控制点有唯一对应的受控制三角形,每个三角形拥有第一类控制点集合与第二类控制点集合。以控制点与受控三角形之间的距离作为简化误差,其中第二类控制点与受控三角形之间的距离为系数距离。根据设定的三角形权重,按照顺序进行三角形折叠操作,通过投射使新顶点保持在原始网格上。简化操作后必须满足控制点到受控三角形的距离小于阈值。

## 1 基本概念

**定义 1** 三角形网格  $M$  由顶点集合  $V = \{v_1, v_2, \dots, v_n\}$  和三角形集合  $T = \{t_1, t_2, \dots, t_m\}$  所组成的二元组  $(V, T)$  来表示,三角形沿公共边及在顶点处相连接,一条边最多只能被两个三角形共有。初始网格记作  $M_0$ ,初始顶点集合  $V_0 =$

$\{v_{0,1}, v_{0,2}, \dots, v_{0,m}\}$ ,初始三角形集合  $T_0 = \{t_{0,1}, t_{0,2}, \dots, t_{0,m}\}$ 。

**定义 2** 三角形网格内以  $v_i$  为顶点的三角形构成的集合称为顶点  $v_i$  的相关三角形集合  $C(v_i)$ ,以  $v_i$  为顶点的三角形的数目称为  $v_i$  的阶,记为  $r_i$ 。

**定义 3** 三角形网格内与  $t_i$  的三个顶点相关的三角形的集合,称为三角形  $t_i$  的相关三角形集合  $C(t_i)$ 。

**定义 4** 三角形  $t_i$  的相关三角形集合  $C(t_i)$  有  $g$  个三角形,其中三角形  $t_r$  的法向量为  $N_r$ ,中心为  $X_r$ ,面积为  $A_r$ ,经过中心点  $X$  并且法向量为  $N$  的平面  $P(N, X)$  定义为三角形集合  $C(t_i)$  的平均平面

$$N = \frac{\sum_{r=1}^g (N_r \cdot A_r)}{\sum_{r=1}^g A_r}, N = \frac{N}{|N|} \quad (1)$$

$$X = \frac{\sum_{r=1}^g (X_r \cdot A_r)}{\sum_{r=1}^g A_r} \quad (2)$$

## 2 控制点的选取

如图 1 所示,以初始网格三角形的中点作为第一类控制点,初始三角形  $t_{0,i}$  的中心点为控制点  $vc_i$ ,  $vc_i$  在初始网格中

对应三角形即为  $t_{0,i}$ , 第一类控制点集合记为  $VC = \{vc_1, vc_2, \dots, vc_{m_0}\}$ 。在简化网格内每个控制点有唯一对应的三角形, 称为该控制点的受控三角形。每个三角形都有其第一类控制点集合, 三角形  $t_i$  的第一类控制点集合记为,  $VC(t_i) = \{vc_{c(1)}, vc_{c(2)}, \dots, vc_{c(a)}\}$ , 其中  $vc_{c(r)}$  的受控三角形即为  $t_i$ , 初始三角形  $t_{0,i}$  的第一类控制点集合  $VC(t_{0,i}) = \{vc_i\}$ 。以集合  $DC = \{dc_1, dc_2, \dots, dc_{m_0}\}$  记录第一类控制点到其受控三角形的距离, 其中  $dc_i$  为  $vc_i$  到受控三角形  $t_r$  的距离。

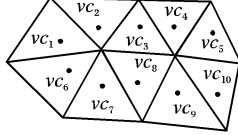


图1 第一类控制点

在网格简化过程中应尽量保持网格特征, 以初始网格特征边的顶点为第二类控制点, 第二类控制点集合记为  $VE = \{ve_1, ve_2, \dots, ve_{m_2}\}$ 。如图2所示, 初始网格三角形  $t_{0,i}$  与  $t_{0,r}$  相邻接, 如果  $t_{0,i}$  与  $t_{0,j}$  之间的夹角小于阈值  $\theta$ , 边  $l(v_{0,h}, v_{0,k})$  为特征边, 顶点  $v_{0,h}$  与  $v_{0,k}$  为第二类控制点。将  $v_{0,h}$  与  $v_{0,k}$  加入  $t_{0,i}$  或是  $t_{0,r}$  的第二类特征点集合, 顶点只能加入一个三角形的第二类特征点集合。

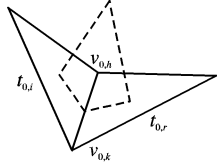


图2 第二类控制点

每个三角形有第二类控制点集合,  $t_i$  的第二类控制点集合记为  $VE(t_i) = \{ve_{e(1)}, ve_{e(2)}, \dots, ve_{e(b)}\}$ , 集合内的  $ve_{e(r)}$  的受控三角形即为  $t_i$ 。以集合  $DE = \{de_1, de_2, \dots, de_{m_2}\}$  记录第二类控制点到受控三角形的系数距离, 其中  $de_i$  为  $ve_i$  到其受控三角形  $t_r$  的系数距离。如果  $ve_i$  到其受控三角形  $t_r$  的欧氏距离为  $de'_i$ , 系数距离  $de_i = de'_i \cdot c$ , 其中  $c$  是特征系数, 有  $c \geq 1$ 。

以控制点到受控三角形的距离作为初始网格与简化网格之间的误差, 设定误差阈值  $d_{hold}$ , 在网格简化过程中必须有  $dc_i \leq d_{hold}$ 、 $de_i \leq d_{hold}$ 。在网格简化的过程中, 控制点与受控三角形之间的对应关系, 将随着网格的简化不断调整。

### 3 三角形折叠操作

如图3所示, 以  $t_i$  为目标三角形进行三角形折叠操作, 将  $t_i$  的顶点合并为  $v_{new}$ 。

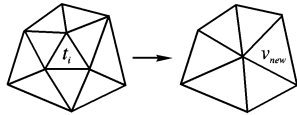


图3 三角形折叠操作

三角形的折叠顺序是影响网格简化效果的关键之一, 对网格内三角形设定一个权重, 将三角形按权重进行排序得到折叠顺序队列  $L_{collapse}$ , 权重小的三角形将首先被折叠。三角形  $t_i$  的权重系数为  $W_i$

$$W_i = A_i \cdot (w_\alpha \cdot \alpha_i + w_v \cdot rc_i + w_d \cdot d_{i,max}) \quad (3)$$

$A_i$  为  $t_i$  的面积;  $w_\alpha$ 、 $w_v$  和  $w_d$  是权重系数,  $0 \leq w_\alpha, w_v,$

$w_e \leq 1$ ; 等角度权重  $\alpha_i = 2 \cdot \left( \sum_{k=1}^3 \cos \beta_k - 1 \right)$ , 其中  $\beta_k$  是三角

形  $t_i$  的内角; 阶权重  $rc_i = \frac{|r_{new} - 6|}{12}$ ,  $r_{new}$  是三角形折叠后目标顶点的阶, 算法中认为顶点的阶的最佳值为 6; 误差权重  $d_{i,max}$  是  $t_i$  的第一类控制点到  $t_i$  的距离与第二类控制点到  $t_i$  的系数距离的最大值。

取折叠顺序队列  $L_{collapse}$  的头部三角形记为  $t_i$ ,  $t_i$  为目标三角形,  $t_i$  的相关三角形集合为  $C(t_i)$ ,  $C(t_i)$  的第一类控制点集合  $VC(C(t_i)) = \{vc_{c(1)}, vc_{c(2)}, \dots, vc_{c(A)}\}$ , 第二类控制点的集合  $VE(C(t_i)) = \{ve_{e(1)}, ve_{e(2)}, \dots, ve_{e(B)}\}$ ,  $C(t_i)$  的控制点集合是  $C(t_i)$  内各个三角形控制点集合的并集。  $VC(C(t_i))$  对应初始网格上的三角形集合为  $T(C(t_i)) = \{t_{0,c(1)}, t_{0,c(2)}, \dots, t_{0,c(A)}\}$ 。计算得到三角形集合  $C(t_i)$  的平均平面  $P(N, X)$ , 过点  $X$ , 沿法线  $N$  方向的直线为  $L(N, X)$ , 直线  $L(N, X)$  与三角形集合  $T(C(t_i))$  内的三角形的交点即为目标顶点  $v_{new}$ 。将  $t_i$  折叠到目标顶点  $v_{new}$  得到三角形集合  $C(v_{new})$ 。

计算  $VC(C(t_i))$  内各个控制点到  $C(v_{new})$  内各个三角形的距离, 得到集合  $DC(C(v_{new})) = \{(d_{c(1)}, t_{h(1)}), \dots, (d_{c(A)}, t_{h(A)})\}$ , 其中  $(d_{c(r)}, t_{h(r)})$  为记录控制点  $vc_{c(r)}$  到三角形集合  $C(v_{new})$  内三角形  $t_{h(r)}$  的距离最近, 距离为  $d_{c(r)}$ 。  $d_{c(1)}$ 、 $d_{c(2)}$ 、 $\dots$ 、 $d_{c(A)}$  中最大值记为  $d_{c,max}$ 。

计算  $VE(C(t_i))$  内各个控制点到  $C(v_{new})$  内各个三角形的系数距离, 得到集合  $DE(C(v_{new})) = \{(d_{e(1)}, t_{k(1)}), \dots, (d_{e(B)}, t_{k(B)})\}$ , 其中  $(d_{e(r)}, t_{k(r)})$  为记录控制点  $ve_{e(r)}$  到三角形集合  $C(v_{new})$  中三角形  $t_{k(r)}$  的系数距离最近, 系数距离为  $d_{e(j)}$ 。  $d_{e(1)}$ 、 $d_{e(2)}$ 、 $\dots$ 、 $d_{e(B)}$  中最大值记为  $d_{e,max}$ 。

如果  $d_{c,max} \leq d_{hold}$  并且  $d_{e,max} \leq d_{hold}$ , 表明折叠操作形成的误差在阈值范围内; 如果  $d_{c,max} > d_{hold}$  或者  $d_{e,max} > d_{hold}$ , 表明折叠操作形成的误差超出阈值范围, 三角形  $t_i$  折叠操作不合法, 目标三角形  $t_i$  的权重  $W_i = W_i \cdot p$ , 其中  $p$  为惩罚系数,  $p > 1$ , 将三角形  $t_i$  按照新权重插入折叠顺序队列  $L_{collapse}$ 。

如图4所示, 如果目标三角形  $t_i$  邻接环结构  $loop(t_2, t_3, t_4)$ ,  $t_i$  折叠后会造成  $t_3$  与  $t_4$  重合, 这将破坏网格的拓扑结构。如果  $t_i$  邻接环结构,  $t_i$  权重  $W_i = W_i \cdot p$ , 将  $t_i$  按照新权重插入折叠顺序队列  $L_{collapse}$ 。

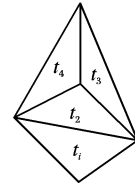


图4 三角形邻接环结构

如图5所示, 如果折叠三角形  $t_i$  到新顶点  $v_{new}$  会造成三角形  $t_2$  翻转, 这将破坏网格的拓扑结构, 这就需要在进行三角形折叠后, 计算  $v_{new}$  邻接的三角形的法向是否发生大的偏转, 如果偏转过大, 取消三角形折叠操作,  $t_i$  权重  $W_i = W_i \cdot p$ , 将  $t_i$  按照新权重插入折叠顺序队列  $L_{collapse}$ 。

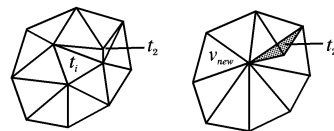


图5 折叠三角形后发生翻转

若三角形  $t_i$  折叠操作合法, 需要重新分配控制点。将第

一类控制点集合  $VC(C(t_i))$  内的各个控制点分配给三角形集合  $C(v_{new})$  内的三角形根据集合  $DC(C(v_{new}))$  内的记录  $(d_{c(r)}, t_{h(r)})$ , 将控制点  $vc_{c(r)}$  加入三角形  $t_{h(r)}$  的第一类控制点集, 在集合  $DC$  内  $dc_{c(r)} = d_{c(r)}$ 。将第二类控制点集合  $VE(C(t_i))$  内的各个控制点分配给三角形集合  $C(v_{new})$  内的三角形的第二类控制点集合, 根据集合  $DE(C(v_{new}))$  内的记录  $(d_{e(r)}, t_{k(r)})$ , 将控制点  $ve_{e(r)}$  加入三角形  $t_{k(r)}$ , 在集合  $DE$  内  $de_{e(r)} = d_{e(r)}$ 。计算  $C(v_{new})$  相关区域内三角形的权重, 相关区域包括  $C(v_{new})$  内三角形以及  $C(v_{new})$  内三角形相连接的三角形, 将相关区域内三角形按照权重插入折叠顺序队列  $L_{collapse}$ 。

基于控制点误差控制的网格简化算法步骤:

步骤 1: 取得初始网格的第一类控制点集合与第二类控制点集合, 并将控制点分配给三角形;

步骤 2: 计算初始网格内各个三角形的权重, 得到折叠顺序队列  $L_{collapse}$ ;

步骤 3: 由折叠顺序队列  $L_{collapse}$  头部取三角形作为目标三角形;

步骤 4: 检测目标三角形是否邻接环结构, 如果邻接环结构, 将目标三角形权重乘以惩罚系数后重新插入折叠顺序队列, 转步骤 3;

步骤 5: 计算得到目标顶点, 折叠目标三角形到目标顶点;

步骤 6: 进行翻转检测, 如果有翻转现象, 取消折叠操作, 将目标三角形权重乘以惩罚系数后重新插入折叠顺序队列, 转步骤 3;

步骤 7: 计算第一类控制点与第二类控制点到三角形距离, 如果最大距离小于阈值, 折叠操作成立; 否则, 取消折叠操作, 目标三角形权重乘以惩罚系数, 将三角形重新插入折叠顺序队列, 转步骤 3;

步骤 8: 重新分配第一类控制点与第二类控制点, 重新计算相关区域三角形权重, 将相关三角形插入队列, 判断是否满足结束条件, 如满足, 结束程序; 否则, 转步骤 3。

#### 4 误差的度量

Cignoni 在 1998 年提出的算法<sup>[7]</sup>对模型之间的误差进行度量, 两个模型表面  $S_1$  与  $S_2$ , 最大误差为:

$$E_{\max}(S_1, S_2) = \max(q(v, S_2)), v \in M_0$$

平均误差(mean error)为:

$$E_{\text{mean}}(S_1, S_2) = \frac{\int q(v, S_2) \cdot ds}{|S_1|}$$

其中  $q(v, S_2)$ ,  $v \in S_1$  为面  $S_1$  上点  $v$  到面  $S_2$  的距离,  $|S_1|$  为面  $S_1$  的面积。在本文中

$$E_{\max}(M_0, M) = \max(dc_i) \quad (4)$$

$$E_{\text{mean}}(M_0, M) = \frac{\sum_{i=1}^{m_0} (dc_i \cdot A_{0,i})}{\sum_{i=1}^{m_0} A_{0,i}} \quad (5)$$

其中  $dc_i$  为第一类控制点  $vc_i$  到其对应受控三角形  $t_k$  的距离,  $A_{0,i}$  为初始网格三角形  $t_{0,i}$  的面积,  $\sum_{i=1}^{m_0} A_{0,i}$  为初始网格的面积。本文中最大误差即为阈值  $d_{hold}$ 。

#### 5 实验结果

采用 VC++ 6.0 实现了算法, 并进行了数据实验。原始数据为  $68 \times 512 \times 512$  的螺旋 CT 扫描数据, 数据的层距 3mm, 同层数据点间距 0.4 mm, 采用 MC 算法生成的原始等值面如图 6(a) 所示, 网格内有 439 852 个三角形以及 219 554 个顶点。取特征边的角度阈值  $\theta$  为 0.6, 特征系数为  $c = 3$ , 惩罚系数  $p = 5$ , 权重系数  $w_a, w_v, w_e = 1$ 。

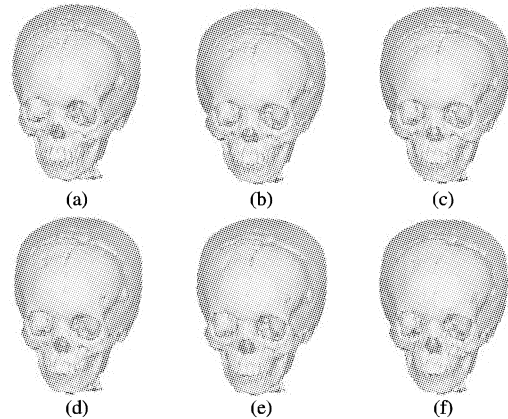


图 6 头部模型简化

表 1 模型的简化表

	$d_{hold}$	三角形\顶点数目	三角形简化率(%)	平均误差
a	0	439 852 \ 219 554	0	0
b	0.2	143 819 \ 71 560	67.3	0.087
c	0.4	70 003 \ 34 652	84.1	0.13
d	0.6	41 315 \ 20 308	90.4	0.19
e	0.8	28 142 \ 13 722	93.6	0.24
f	1.0	17 986 \ 8 644	95.9	0.33

本文提出一种基于控制点误差控制的网格简化算法, 在初始网格内提取第一类与第二类特征点, 以特征点与受控三角形之间的距离作为简化误差。根据设定的三角形折叠权重, 按照顺序进行三角形折叠操作。三角形折叠的目标顶点保持在原始网格上, 实验表明算法可以有效地进行网格简化, 并保持原始模型的拓扑结构。考虑到模型的多样性, 算法需要进一步改进和完善, 如自适应地选取阈值和特征系数等。

#### 参考文献:

- [1] KALVIN AD, TAYLOR RH. Surfaces: polygonal mesh simplification with bounded error[J]. IEEE Computer Graphics and Applications, 1996, 16(3): 64-77.
- [2] LOUNSBERRY M, DEROSE T. Multiresolution analysis for surfaces of arbitrary topological type[J]. ACM Trans. on Graphics, 1997, 16(1): 34-73.
- [3] HOPPE H, DEROSE T. Mesh optimization[A]. Proceeding of the Computer Graphics[C]. 1993, 19-26.
- [4] GARLAND M, HECKBERT PS. Surface simplification using quadric error metric[A]. Proceeding of the Computer Graphics[C]. 1997, 209-216.
- [5] 刘晓利, 刘则毅, 高鹏东, 等. 基于尖特征度的边折叠简化算法[J]. 软件学报. 2005, 16(5): 668-674.
- [6] GIENG TS, HAMANN B. Constructing Hierarchies for Triangle Meshes[J]. IEEE Transaction on Visualization and computer Graphics, 1998, 4(2): 145-161.
- [7] CIGNONI P, ROCCHINI C, SCOPIGNO R. Metro: Measuring error on simplified surfaces[J]. Computer Graphics Forum, 1998, 17(2): 167-174.