

# 体积平方度量下的特征保持网格简化方法

周元峰<sup>1)</sup> 张彩明<sup>1),2)</sup> 贺 平<sup>1)</sup>

<sup>1)</sup>(山东大学计算机科学与技术学院 济南 250101)

<sup>2)</sup>(山东经济学院计算机科学与技术学院 济南 250014)

**摘 要** 提出了一种基于体积平方度量的三角形折叠网格简化新方法. 新方法通过极小化误差目标函数简化三角形网格. 简化误差定义为三角形简化后产生的网格模型平方体积变化, 并以三角形几何形状因子和法向因子作为约束. 简化误差的表示形式为一个二次目标函数, 因此, 每次简化后三角形网格的新顶点是一个线性问题的解. 与目前简化效率最好的 QEM 方法相比, 新方法不增加算法复杂度. 如果被简化的三角形是强特征三角形, 则用其高斯曲率最大的顶点作为新顶点, 以保持原始模型的细节特征; 对于非强特征三角形, 新顶点用极小化折叠误差确定. 对于边界三角形, 新顶点的位置由不同于内部三角形的方法进行计算, 保持了网格的边界特征. 最后用实例说明新方法的有效性.

**关键词** 网格简化; 三角形折叠; SVE; 特征因子; 目标函数; 高斯曲率

中图法分类号 TP391

DOI号: 10.3724/SP.J.1016.2009.00203

## Feature Preserving Mesh Simplification Algorithm Based on Square Volume Measure

ZHOU Yuan-Feng<sup>1)</sup> ZHANG Cai-Ming<sup>1),2)</sup> HE Ping<sup>1)</sup>

<sup>1)</sup>(School of Computer Science and Technology, Shandong University, Jinan 250101)

<sup>2)</sup>(School of Computer Science and Technology, Shandong Economic University, Jinan 250014)

**Abstract** A new algorithm for mesh simplification with triangle collapse based on square volume measure is presented in this paper. Models can be simplified by minimizing error objective function. Square volume error, shape factor and normal constraint factor of triangles are combined together to define the simplification error, which can be described as a quadratic objective function. New vertices replaced collapsed triangles can be determined linearly. Comparing with the QEM method which is the most effective method so far, computation complexity will not be increased. Gaussian curvature factor is computed for each collapsed triangle and used to identify strong feature triangle. For non-strong feature triangle, new vertex position is determined by minimizing the error objective function. For strong feature triangle, new vertex is taken the strong feature vertex of the three triangle vertices for preserving the model feature. Each collapsed triangle is processed as inner triangle or boundary triangle to keep the boundary feature. Experiments for the efficiency of the new algorithm are included.

**Keywords** mesh simplification; triangle collapse; SVE; feature factor; objective function; Gaussian curvature

收稿日期: 2007-10-25; 最终修改稿收到日期: 2009-01-05. 本课题得到国家“九七三”重点基础研究发展规划项目基金(2006CB303102)和国家自然科学基金(60673003, 60573180)资助. 周元峰, 男, 1980年生, 博士研究生, 主要研究方向为计算机图形学、数字几何处理. E-mail: yfzhou@mail.sdu.edu.cn. 张彩明, 男, 1955年生, 博士, 教授, 博士生导师, 主要研究领域为计算机图形学、计算机辅助几何设计和医学图像处理. E-mail: czhang@sdu.edu.cn. 贺平, 女, 1966年生, 博士研究生, 副教授, 主要研究方向为计算机图形学、计算机信息可视化技术.

## 1 引言

在计算机图形学、虚拟现实、计算机辅助设计应用中,三维几何模型通常用多边形网格来表示. 网格模型一般采用三角形网格和四边形网格,三角形网格因为其数学表示简单,通用性、灵活性较好,应用尤其广泛. 随着三维激光扫描技术的发展,三角网格模型的精度也得到了很大程度的提高,但是随着网格模型复杂程度的提高,对于计算机的存储、计算和绘制能力的要求也越来越高,而高复杂度的网格模型并不是经常需要,因此网格模型的简化和多分辨率表示显得更为重要.

网格模型简化是指用更少的网格对初始模型进行表示,并且在简化过程中尽量保持模型形状变化较小. 对于三角形网格简化的研究已有很多,目前常用的三角网格简化方法可分为几何元素删除法,网格重新划分法和几何元素折叠法. 几何元素删除法包括顶点删除和三角形删除<sup>[1-2]</sup>,这两种方法需要对被删除顶点或三角形所在区域进行重新三角化,不用产生新的网格顶点,但增加了简化操作的复杂性. 网格重新划分方法<sup>[3-5]</sup>需要将原有的网格顶点删除,通过优化方法重新计算新的网格顶点并作三角划分,得到具有更少顶点的逼近网格. 几何元素折叠法<sup>[6-16]</sup>包括边折叠与三角形折叠,会产生一定数量的新顶点,但不需要对折叠区域重新三角化,具有更好的稳定性且简化速度快. Garland 于 1997 年提出了一种基于二次误差的边折叠网格简化方法<sup>[6]</sup>,该方法简化速度快,简化生成的网格质量好,是目前普遍认同的一种简化方法,但是该方法误差定义为顶点到平面的距离平方和,并没有考虑网格自身的其他几何特性,如三角形网格自身形状、网格模型的内部体积变化等. 文献[7]中提出一种边折叠的网格简化算法,算法将网格简化和网格优化结合起来得到了形状较好的网格简化模型,但该方法需要将三维网格映射到二维平面后做参数化,并将参数化的结果再映射回三维网格达到网格优化的目的,操作繁琐. 文献[8]中根据网格顶点的法向确定简化误差,得到一种边折叠的网格简化方法. 文献[13]中提出了一种体积保持的多分辨率网格模型构造方法,文中采用了体积变化的绝对值作为误差标准,采用边折叠方法进行网格化简. 文献[14]简单地引入了顶点的尖锐特征度,在 Garland 算法的误差矩阵中加入一项成为新的误差矩阵,较好地取得了保留网格特征的效果,但是尖锐特征度的取法过于简单,而且

阈值和惩罚系数需要预先根据经验设定. 文献[15]中提出了一种局部体积误差的半边折叠网格简化方法,文中以半边体积变化的绝对值并结合网格曲率信息定义误差函数,但文中的曲率因子计算方法过于费时,并且边折叠后新顶点的位置并没有根据网格体积变化量来确定. 文献[16]用体积变化衡量网格简化质量,但也没有考虑三角网格形状以及网格细节特征.

除了边折叠方法之外,基于三角形折叠的网格简化方法也是构造多分辨率网格模型的重要方法之一. 基于三角形折叠的网格简化方法每次删除一个三角形相当于作两次边折叠操作,因此在简化率相同情况下,速度要比边折叠方法快. Hamann 于 1994 年提出了一种基于三角形折叠的网格简化方法<sup>[9]</sup>,方法根据三角形的曲率和等角度计算简化后的误差,首先删除网格模型中狭长的三角形. Gieng 也提出了类似的三角形折叠简化算法<sup>[10]</sup>,三角形折叠代价定义为三角形面积与曲率的乘积,但是这种算法的计算复杂度较高,并且对于面积与曲率属于两个不同的几何量,简单的相乘不能有效地反映面积与曲率两个衡量标准. 文献[11]提出了一种基于二次误差的三角形折叠简化算法,并给出了一种传递简化误差的方法,但文中以点到被折叠三角形邻接三角形的最大距离作为简化误差,对三角形做折叠处理时误差矩阵对应的逆矩阵不存在,这意味着,对三角形简化时新顶点位置的计算是不稳定的. 从网格简化的效率和速度出发,本文采用三角形折叠策略进行网格简化,将简化网格的体积变化、网格形状、细节特征等因素结合起来定义新的目标函数,使得新顶点的确定同网格的几何误差之间具有了明确的关系.

## 2 基于 QEM 的三角形网格简化

QSlim 方法<sup>[6]</sup>是 Garland 等于 1997 年提出的,该方法基于边折叠机制(如图 1 所示),首先,为每一个原始网格中的顶点  $v_i = [x_i, y_i, z_i, 1]^T$  分配一个对称的误差矩阵  $Q_i$ ,每条边的误差矩阵  $\bar{Q}$  由两个顶点的误差矩阵之和得到:  $\bar{Q} = Q_i + Q_j$ . 假设边  $e$  折叠后产生的新顶点为  $\bar{v} = [\bar{x}, \bar{y}, \bar{z}, 1]$ ,则该折叠操作产

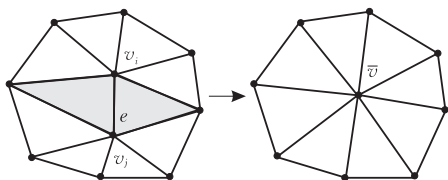


图 1 边折叠操作

生的误差可定义为  $\Delta(\bar{\mathbf{v}})$ ：

$$\Delta(\bar{\mathbf{v}}) = \bar{\mathbf{v}}^T \bar{\mathbf{Q}} \bar{\mathbf{v}} \tag{1}$$

新顶点的最佳位置可由极小化式(1)确定,即  $\partial\Delta(\bar{\mathbf{v}})/\partial\bar{x} = \partial\Delta(\bar{\mathbf{v}})/\partial\bar{y} = \partial\Delta(\bar{\mathbf{v}})/\partial\bar{z} = 0$ , 方程组整理为如下形式：

$$\begin{bmatrix} \bar{q}_{11} & \bar{q}_{12} & \bar{q}_{13} & \bar{q}_{14} \\ \bar{q}_{12} & \bar{q}_{22} & \bar{q}_{23} & \bar{q}_{24} \\ \bar{q}_{13} & \bar{q}_{23} & \bar{q}_{33} & \bar{q}_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{\mathbf{v}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

解得

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

在 QEM 方法中,式(1)中的未知系数由顶点到该顶点邻接三角形距离的和确定. 在图 1 中,假定顶点  $v_i, v_j$  的邻接三角形所在的平面方程为  $ax + by + cz + d = 0$ , 其中  $a^2 + b^2 + c^2 = 1$ . 令  $\mathbf{p} = (a, b, c, d)$ , 则点  $\bar{\mathbf{v}}$  到该三角形所在平面的距离平方为  $(\mathbf{p}^T \bar{\mathbf{v}})^2$ , 误差函数(1)可写为

$$\Delta(\bar{\mathbf{v}}) = \sum (\bar{\mathbf{v}}^T \mathbf{p}) (\mathbf{p}^T \bar{\mathbf{v}}) = \sum \bar{\mathbf{v}}^T \mathbf{p} \mathbf{p}^T \bar{\mathbf{v}} = \bar{\mathbf{v}}^T \sum \mathbf{p} \mathbf{p}^T \bar{\mathbf{v}},$$

其中,  $\mathbf{p}^T \mathbf{p} = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$ . 与式(1)比较可得

$$\bar{\mathbf{Q}} = \sum \mathbf{p} \mathbf{p}^T.$$

文献[11]将 QEM 方法引入基于三角形折叠的网格简化方法中,该方法为每个三角形建立一个对称的误差矩阵,简化步骤用三角形折叠代替 Garland 的边折叠方法(如图 2 所示). 文献[11]中的方法也采用点到三角形的距离作为误差标准,其网格简化误差定义为三角形折叠后产生的新顶点到该三角形的邻接三角形距离的最大值,新顶点的求解使得这个最大值最小化. 但是在确定折叠后产生的新顶点  $\bar{\mathbf{v}}$  时,其误差矩阵为奇异矩阵,因此矩阵的逆是不存在的. 文献中通过比较三角形的 3 个顶点、中心和三边的中点 7 个位置,取其中使得式(1)最小的点作为三

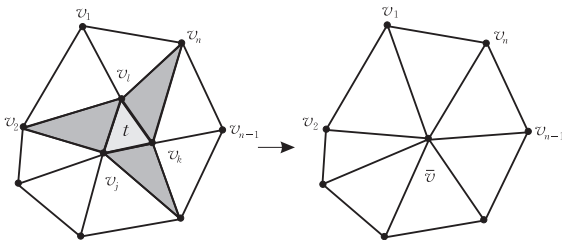


图 2 三角形折叠操作

角形折叠后的新顶点,但是这样做使得折叠误差的定义失去意义,因此其折叠误差的选取是不合理的. 并且,和 QEM 边折叠方法一样,新顶点的确定仅是在点到平面距离衡量标准下的最优解,没有考虑到三角网格自身的其他几何特性,如三角形网格的自身形状和网格顶点曲率等.

本文提出了一种新的基于三角形折叠网格简化方法,新方法采用网格模型平方体积误差最小策略,结合三角形特征因子,给出了类似于 QEM 的误差标准,并用该误差标准对网格模型做三角形折叠简化处理. 第 3 节给出了新方法的基本思想和详细步骤.

### 3 体积平方度量下的三角形折叠网格简化

进行三角形折叠网格简化包括两个重要的因素:一是确定被折叠的三角形,二是确定折叠三角形后新顶点的位置. 这两个因素决定了折叠简化后是否能得到优化的新网格. 网格折叠简化实质上可认为是一个优化问题,解决该问题的常用方法是定义目标函数,简化后的网格是目标函数的最优解,因此网格简化的关键是如何建立该目标函数. 显然,目标函数应是表达新旧两个网格间差别的函数,通过极小化目标函数得到新顶点的位置,并使新旧两个网格间的差别在某种意义上达到最小.

本文中的目标函数采用网格的平方体积变化作为衡量模型间差别的标准,为了尽量保持在简化过程中网格的特征不变,还在目标函数中增加了三角形的形状和法向特征因子约束条件,三角形曲率则用于保持网格中具有尖锐特征的网格顶点不变. 为了提高网格简化速度和确保简化过程的可解性,目标函数的极小化应归结为线性问题. 以下是对简化目标函数的详细描述.

#### 3.1 基于平方体积误差(SVE)的目标函数定义

概念约定:给定一个三角形网格  $M$ ,其点、边和三角形分别记为  $v, e$  和  $t$ . 以点  $v$  为顶点的三角形称为  $v$  的邻接三角形,三角形  $t$  的 3 个顶点的邻接三角形集合称为该三角形的邻接三角形.

在进行三角形折叠网格简化时,三角形折叠后产生的新网格和初始的三角网格相比会产生体积变化. 好的网格简化方法应能够在简化过程中尽可能地保持网格体积变化小,使简化模型更好地逼近原始模型<sup>[13,15]</sup>. 三角形折叠简化操作可看做每次删除一个三角形,把删掉的三角形的 3 个顶点用一个新顶点代替,新顶点与删掉三角形相邻接的区域做重

新三角化. 新的三角形网格和原网格体积的变化定义为一些四面体体积之和. 这些四面体的底面为被删除三角形和与其邻接的三角形, 我们把被删除的三角形及其邻接三角形所在的区域称为被折叠区域. 如图 3 所示, 被折叠的三角形为  $\Delta v_0 v_1 v_2$ , 其邻接三角形分别为  $\Delta v_0 v_3 v_4$ 、 $\Delta v_0 v_4 v_5$ 、 $\Delta v_0 v_5 v_1$ 、 $\Delta v_1 v_5 v_6$ 、 $\Delta v_1 v_6 v_7$ 、 $\Delta v_1 v_7 v_2$ 、 $\Delta v_2 v_7 v_8$ 、 $\Delta v_2 v_8 v_0$ 、 $\Delta v_0 v_8 v_3$ , 生成的新顶点为  $\bar{v}$ . 因此, 在图 3 中折叠  $\Delta v_0 v_1 v_2$  后引起的体积变化为以  $\bar{v}$  为顶点、以这 10 个三角形为底面的四面体体积之和. 下面讨论折叠一个三角形后, 如何计算三角形网格的体积变化.

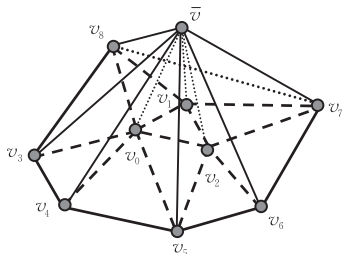


图 3 三角形折叠导致的网格体积变化

记简化过程中被折叠区域中的一个三角形  $t$  的面积为  $S$ . 设点  $\bar{v}$  的坐标为  $(\bar{x}, \bar{y}, \bar{z}, 1)$ . 由三角形  $t$  和点  $\bar{v}$  构成的三棱锥的体积记为  $V$ . 如图 3 中, 设  $t$  所在的平面方程为  $ax + by + cz + d = 0$ , 其中  $a^2 + b^2 + c^2 = 1$ . 令  $\mathbf{p} = (a, b, c, d)$ , 则  $V = \frac{1}{3} (\bar{\mathbf{v}} \cdot \mathbf{p}^T) \cdot S$ .

这是一个带符号的体积计算公式. 如果在简化后, 网格模型的一部分的符号体积为正, 一部分的符号体积为负, 虽然网格模型有变化, 但网格的符号体积变化总和却可能为零, 无法对网格的体积变化进行度量. 在以往方法中, 文献[13, 15]均采用符号体积误差的绝对值形式作为衡量误差, 这样做的困难在于需要判断体积变化的符号. 因此, 本文采用平方体积累加变化作为误差度量标准, 将这个变化定义为平方体积误差, 记为 SVE (Square Volume Error). 删除一个三角形引起的总的平方体积误差可表示为

$$\sum_{i=1}^m V_i^2 = \sum_{i=1}^m \frac{1}{9} (\bar{\mathbf{v}} \cdot \mathbf{p}_i^T \mathbf{p}_i \cdot \bar{\mathbf{v}}^T) \cdot S_i^2 \quad (2)$$

其中  $m$  为被折叠区域中的三角形个数.

为了得到体积平方变化较小的简化网格, 可把式(2)作为误差函数, 新顶点  $\bar{v}$  应使得式(2)的值尽可能小, 既通过极小化该误差函数确定新顶点  $\bar{v}$ . 本文中为网格中的每个三角形定义临时误差矩阵  $\mathbf{Q}_i$ :

$$\mathbf{Q}_i = \frac{1}{9} S_i^2 \cdot \mathbf{p}^T \mathbf{p} = \frac{1}{9} S_i^2 \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (3)$$

三角形被折叠后产生的网格平方体积误差由式(2)和式(3)结合可得, 每个三角形的折叠误差矩阵定义为三角形本身及其邻接三角形临时误差矩阵

的和:  $\bar{\mathbf{Q}} = \sum_{i=1}^m \mathbf{Q}_i$ , 其中  $i$  包括该三角形自身及其邻接

三角形. 由式(2)定义的误差函数只是对于网格模型平方体积变化进行约束. 对于网格简化来说, 简化后的网格质量也是需要考虑的重要因素, 如三角形形状较好和保持网格尖锐特征不变等. 下面我们对于式(2)施加新的约束, 将约束因子与网格的平方体积变化相结合, 得到更为合理的简化目标函数.

### 3.2 定义特征约束因子

在平方体积误差的基础上, 我们考虑删除对网格体积变化影响小的三角形的同时, 应将形状不好的三角形优先删除. 对于一般三角形网格, 等边三角形被认为形状最佳, 若有一个内角接近零度, 则认为该三角形形状差. 另外, 从网格简化角度来看, 两个等角度的三角形, 面积大的三角形应认为比面积小的三角形“形状”要好, 因此, 网格模型中一个三角形的形状可由下面的几何形状因子衡量:

$$W = 2 \cdot (1 - \cos(\min(\alpha))) \cdot S,$$

其中  $\min(\alpha)$  为三角形最小内角. 因为  $0 < \min(\alpha) \leq \pi/3$ , 可得  $0 < 2 \cdot (1 - \cos(\min(\alpha))) \leq 1$ . 这样对网格模型的每一个三角形都定义了一个形状因子. 因为对于每一次三角形折叠的化简操作, 是对被折叠三角形及其邻接三角形区域进行重组, 这样化简过程不仅仅要考虑被折叠三角形自身的形状因子大小, 还要考虑被折叠三角形的邻接三角形. 我们定义简化目标函数考虑到了被折叠区域的每一个三角形, 因此可以将形状因子和三角形折叠后产生的网格平方体积误差很好地结合起来.

给定的网格模型一般在曲率变化较大的区域网格比较密集, 其局部平方体积误差也是很小的, 为了避免在简化过程中首先化简局部区域不平坦的密集三角形网格, 我们在目标函数(2)中引入一个特征约束因子. 与三角网格的曲率估计相比, 三角形的法向更容易获得, 为了不增加目标函数的计算复杂度, 本文采用法向定义特征约束因子. 在已有的方法中, 一般用点或者三角形的邻接三角形法向量的加权平均作为约束<sup>[5]</sup>, 但是这样容易产生较大误差. 如图 4 所示,  $t_0$  表示一个三角形网格的剖面示意图, 其两个相邻三角形分别为  $t_1, t_2$ . 图中  $t_1, t_2$  的法向  $\mathbf{n}_1$  和  $\mathbf{n}_2$  夹角接近  $180^\circ$ , 其平均值会接近于  $0^\circ$ , 这样容易导致网格尖锐区域被误判断为平坦区域. 因此, 本文提出一种新的法向约束因子.  $t_0$  的法向约束因子应该与



三角形  $t_0$  的法向和其邻接三角形法向的夹角成正比, 即  $t_0$  的法向和其邻接三角形法向的夹角越大,  $t_0$  的法向约束因子就应该越大. 令三角形  $t_0$  的邻接三角形为  $t_1 \cdots t_k$ , 相应的法向分别为  $\mathbf{n}_0 \mathbf{n}_1 \cdots \mathbf{n}_k$ . 由于所有三角形法向均为单位向量, 因此三角形  $t_0$  的法向和  $t_i$  的法向间的夹角对法向约束因子的贡献可由  $1 - \mathbf{n}_0 \cdot \mathbf{n}_i$  衡量, 所以  $t_0$  的法向约束因子可定义为

$$N = \frac{\sum_{i=1}^k (1 - \mathbf{n}_0 \cdot \mathbf{n}_i)}{k}.$$

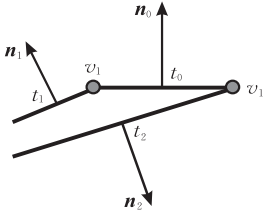


图 4 网格三角形剖面示意图及法向

由三角形法向约束因子定义可知:  $0 \leq N_0 \leq 1$ , 当三角形与其邻接三角形处于一个平面上时  $N_0 = 0$ , 当两个邻接三角形法向夹角越接近  $180^\circ$  时,  $N$  的值越接近于 1.

将每个三角形的形状因子和法向约束因子引入目标函数中, 由式(3)知, 网格中的每个三角形  $t_i$  的误差矩阵可定义如下:

$$\mathbf{Q}_i = \frac{1}{9} W_{\text{shape-factor}}^i \cdot N_{\text{normal-factor}}^i \cdot \mathbf{S}_i^2 \cdot \mathbf{p}_i^T \mathbf{p}_i \quad (4)$$

设  $\bar{\mathbf{v}} = [\bar{x}, \bar{y}, \bar{z}, 1]$  为新顶点位置, 基于三角形的误差矩阵式(4), 折叠一个三角形所产生的误差可由下面的目标函数定义:

$$E = \bar{\mathbf{v}} \left( \sum_{j=1}^m \mathbf{Q}_j \right) \bar{\mathbf{v}}^T = \bar{\mathbf{v}} \cdot \bar{\mathbf{Q}} \cdot \bar{\mathbf{v}}^T \quad (5)$$

其中

$$\bar{\mathbf{Q}} = \begin{bmatrix} \bar{q}_{11} & \bar{q}_{12} & \bar{q}_{13} & \bar{q}_{14} \\ \bar{q}_{12} & \bar{q}_{22} & \bar{q}_{23} & \bar{q}_{24} \\ \bar{q}_{13} & \bar{q}_{23} & \bar{q}_{33} & \bar{q}_{34} \\ \bar{q}_{14} & \bar{q}_{24} & \bar{q}_{34} & \bar{q}_{44} \end{bmatrix},$$

新顶点的最佳位置可由极小化目标函数(5)确定, 式(5)展开与式(1)具有相同的表达形式, 极小化求解过程为一个线性问题.

式(5)说明, 每个三角形的误差矩阵引入了该三角形的形状因子和法向约束因子, 因此, 每个三角形的折叠误差不仅考虑了被折叠三角形自身的形状和法向约束, 而且考虑了被折叠区域中所有三角形的形状和法向约束因子, 目标函数所作用的区域更大, 从而使目标函数更加合理.

### 3.3 网格三角形高斯曲率约束

除了要删掉形状不好的三角形外, 在简化过程中也需要保持网格尖锐部分的特征不变. 为了保持三角形网格中具有的细节特征, 文献[15]中采用了网格顶点两个主曲率的绝对值之和作为该顶点的特征因子, 但是主曲率的计算需要根据网格顶点估计的平均曲率和高斯曲率计算得到, 计算过程增加了网格化简的复杂度. 高斯曲率是曲面论中重要的内蕴几何量, 用于刻画一个曲面顶点的几何特征, 设曲面在  $v$  点处的两个主曲率为  $k_1, k_2$ , 它们的乘积  $k = k_1 \cdot k_2$  称为曲面于该点的总曲率或高斯曲率, 它反映了曲面的弯曲程度. 在以往的文献中, 只有针对三角形网格顶点估计其曲率的方法, 其中效果最好的为文献[17]中的方法, 定义如下:

$$K_v = \frac{2\pi - \sum_n \alpha_n}{S_\Sigma(v)},$$

其中  $\alpha_n$  为顶点  $v$  的邻接角, 如图 5 所示.  $S_\Sigma(v)$  为网格顶点  $v$  的邻接三角形面积之和. 曲面上任意一点的高斯曲率绝对值有明显的几何意义, 为包含曲面上一点  $v$  的一小块区域  $\sum \sigma$  同该区域经高斯映射所对应的区域  $\sum \sigma^*$  的面积之比在该区域折叠到  $v$  点时的极限值, 可用于刻画曲面的尖锐特征. 本文的简化方法基于三角形折叠, 我们通过估计网格顶点的高斯曲率引入三角形的曲率特征因子, 采用的方法是将一个三角形 3 个顶点的高斯曲率加权平均定义该三角形的曲率特征因子.

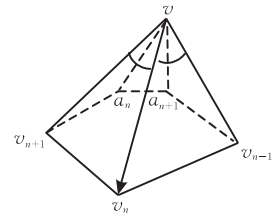


图 5 网格顶点及其邻接角位置

设三角形  $t$  的 3 个顶点为  $v_1, v_2, v_3$ , 定义  $t$  的曲率如下:

$$K = \left| \frac{2\pi - \sum_n \alpha_n}{S_\Sigma(v_1)} \right| \cdot \omega_{v_1} + \left| \frac{2\pi - \sum_n \alpha_n}{S_\Sigma(v_2)} \right| \cdot \omega_{v_2} + \left| \frac{2\pi - \sum_n \alpha_n}{S_\Sigma(v_3)} \right| \cdot \omega_{v_3} \quad (6)$$

其中  $\omega_{v_j}$  为 3 个顶点高斯曲率的权值, 定义为

$$\omega_j = \frac{S_\Sigma(v_j)}{S_\Sigma(v_1) + S_\Sigma(v_2) + S_\Sigma(v_3)}, \quad j = 1, 2, 3.$$

整理式(6)得到

$$K = \frac{\left| 2\pi - \sum_n \alpha_{1,n} \right| + \left| 2\pi - \sum_n \alpha_{2,n} \right| + \left| 2\pi - \sum_n \alpha_{3,n} \right|}{S_{\Sigma}(v_1) + S_{\Sigma}(v_2) + S_{\Sigma}(v_3)}.$$

我们将三角形的曲率因子进行单独处理,大于一定阈值的网格三角形定义为强特征三角形,三角形顶点中曲率绝对值最大的定义为强特征点.在三角形折叠过程中,首先保持强特征三角形中的强特征点不变,从而保证了原始网格中的尖锐特征顶点不会被删除掉.

### 3.4 确定新顶点位置

通过极小化目标函数式(5)确定新顶点  $\bar{v}$  的位置,即  $\frac{\partial E}{\partial x} = \frac{\partial E}{\partial y} = \frac{\partial E}{\partial z} = 0$ . 整理可得线性方程组:

$$\begin{bmatrix} \bar{q}_{11} & \bar{q}_{12} & \bar{q}_{13} & \bar{q}_{14} \\ \bar{q}_{12} & \bar{q}_{22} & \bar{q}_{23} & \bar{q}_{24} \\ \bar{q}_{13} & \bar{q}_{23} & \bar{q}_{33} & \bar{q}_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

线性方程组有解的条件是存在逆矩阵. 展开式(7)的行列式得:

$$\bar{q}_{13}(\bar{q}_{12}\bar{q}_{23} - \bar{q}_{22}\bar{q}_{13}) - \bar{q}_{23}(\bar{q}_{11}\bar{q}_{23} - \bar{q}_{12}\bar{q}_{13}) + \bar{q}_{33}(\bar{q}_{11}\bar{q}_{22} - \bar{q}_{12}\bar{q}_{12}) \quad (8)$$

式(8)可作为判断被删除三角形与其邻接三角形是否在同一平面上的依据. 若被删除三角形与邻接三角形不在同一平面上,则式(7)有解. 若被删除三角形与邻接三角形处于同一平面上,则式(8)的值为零,此时极小化式(5)得不到  $\bar{v}$ ,可采用下面的方法处理.

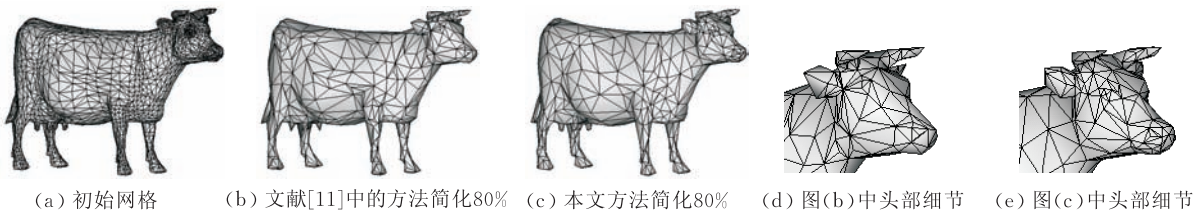


图 6 cow 模型简化 80%

### 3.5 边界三角形处理以及误差修正

若一个三角形只有一个顶点为边界顶点,则称该三角形为顶点边界三角形(如图 7 中的三角形 A). 若有两个顶点为边界顶点,则称该三角形为边界三角形(如图 7 中的三角形 B). 如果边界点在三角形网格中属于特征点,进行网格化简时应尽量保持边界特征不变. 因此,对于点边界三角形折叠后新顶点的位置取为该三角形边界点的位置;对于边界三角形折叠后新顶点的位置取为两个边界点  $v_i, v_j$  的加权平均. 这样处理可尽量保持原始网格的边界不变.

$$\bar{v} = \omega v_i + (1 - \omega) v_j, \quad 0 \leq \omega \leq 1$$

当被删除三角形与邻接三角形处于同一平面时,进行三角形折叠时应将新顶点的位置也置于该平面上,这样不会导致网格模型体积变化. 新顶点的位置由三角形 3 个顶点  $v_i, v_j, v_k$  位置的加权平均确定.

$$\bar{v} = L_i v_i + L_j v_j + L_k v_k,$$

其中  $L_i + L_j + L_k = 1$ . 此时,  $L_i, L_j, L_k$  可以看作  $\bar{v}$  在三角形  $v_i v_j v_k$  中的面积坐标值. 权值的选择应确保新顶点的位置在被删除三角形内部. 权值由下式定义:

$$L_i = \frac{A_{v_i}}{A_{v_i} + A_{v_j} + A_{v_k}}, \quad L_j = \frac{A_{v_j}}{A_{v_i} + A_{v_j} + A_{v_k}},$$

$$L_k = \frac{A_{v_k}}{A_{v_i} + A_{v_j} + A_{v_k}},$$

其中  $A_{v_i}, A_{v_j}, A_{v_k}$  分别为  $v_i, v_j, v_k$  的邻接三角形面积之和. 这样生成新顶点的位置更为接近邻接区域的重心位置,使得生成的网格形状较好.

当被折叠的三角形为强特征三角形时,说明该三角形具有高斯曲率因子大的顶点,我们称之为强特征顶点. 在网格简化过程中,强特征顶点应该作为网格的细节特征被保留下来,因此本文对被折叠三角形的 3 个顶点的高斯曲率因子进行比较,新顶点的位置不再通过极小化目标函数得到,而由被折叠三角形中高斯曲率因子最大的强特征网格顶点代替. 如图 6 所示,对于 cow 模型用文献[11]方法和本文方法分别进行简化 80% 得到的结果可以看出,本文方法在模型细节处(如牛的头部和腹部)可以保持更好的几何特征.

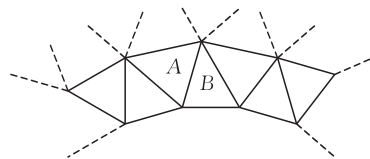


图 7 边界三角形

权因子  $\omega$  仍然根据网格平方体积变化最小确定. 将上式  $\bar{v}$  代入式(5),令  $\partial \Delta(\bar{v}) / \partial \omega = 0$ , 得到  $\omega$  的值,从而确定新顶点  $\bar{v}$ .

如图 8 所示,图(a)中阴影区域称为“环”区域,简化过程中删除“环”区域中的一个三角形时,会产生三角形重叠现象(如图 8(b)所示),因此为了避免产生错误的三角形网格,在每次简化操作后,判断该折叠

区域中是否产生这样的“环”区域,如果存在,则将该“环”区域合并为一个三角形。

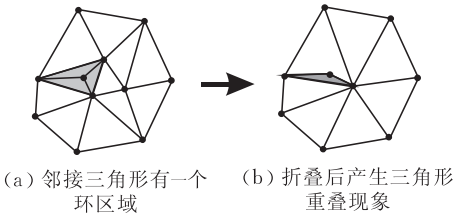


图 8 简化过程中出现的三角形折叠现象

当网格模型的大部分三角形位于同一平面上时,每个与其邻接三角形共面的三角形根据目标函数得到的折叠误差都为零,因此无法判断三角形化简的顺

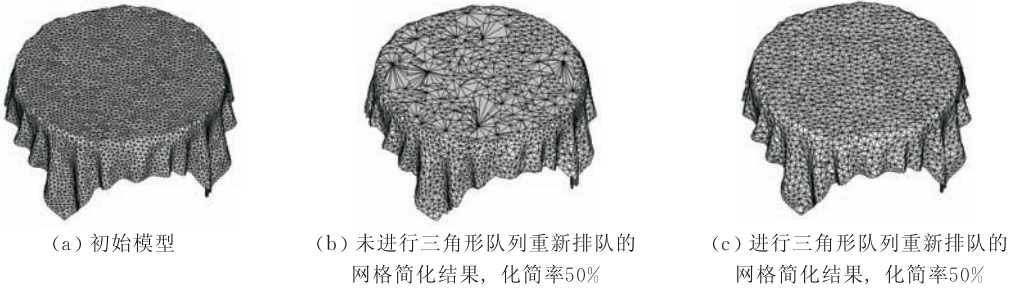


图 9 三角形共面的非封闭网格简化

3.6 算法描述

本文采用堆排序的方法对三角形序列按照折叠误差的大小进行排序,并存储三角形误差队列,每次化简提取堆中误差最小的三角形进行简化操作. 建立一个三角形折叠过程队列,用于记录被简化的三角形. 队列中要包含被简化三角形顶点的信息,以及邻接顶点信息. 这样每一步三角形折叠操作的结果记录之后可以建立一个多分辨率模型,由简化模型经过逆操作可以返回初始网格模型。

算法描述：

1. 对于初始网格中的每一个三角形,根据式(4)计算其误差矩阵  $Q_i$ ;
2. 如果三角形为内部三角形,若式(8)的值不为零,则极小化目标函数计算新顶点的位置;否则,根据式(9)确定新顶点的位置;

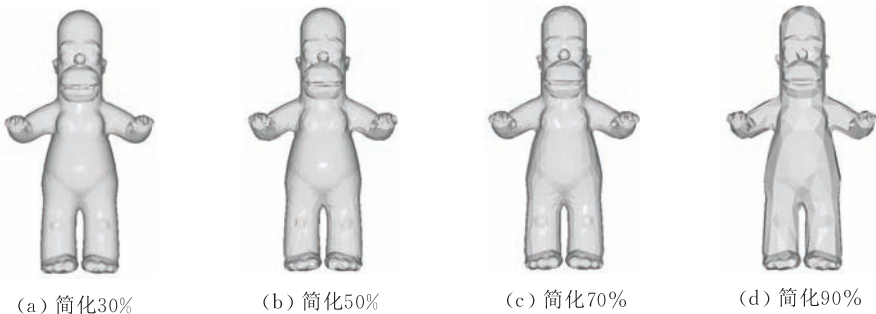


图 10 渐进网格构造

序. 本文给出一种排队方法,如果被折叠三角形与其邻接三角形处于同一平面上,则折叠后产生的新三角形也将共面,更新其折叠误差,如果折叠误差仍为零,则将新三角形插入到堆队列中折叠误差为零的三角形队列最后. 这样避免新产生的三角形在下次折叠操作中马上又被化简掉. 图 9 给出了对于一个大部分三角网格位于同一平面上的非封闭模型简化 50% 的结果,其中图(a)为初始网格,图(b)为本文方法没有对三角形进行重新排队的简化结果,图(c)为本文方法对三角形进行重新排队的简化结果. 明显可以看出,没有对误差为零的三角形进行重新排队得到的简化网格质量较差。

如果三角形为边界三角形,新顶点的位置由三角形的边界顶点或式(10)确定;

3. 计算每个三角形的折叠误差,并按照误差由小到大将三角形顺序加入到三角形误差堆队列中;

4. ①从三角形折叠队列中取出堆队列的堆顶三角形,计算其曲率因子大小,根据设定阈值判断是否为强特征三角形,如果是,三角形折叠后新顶点的位置由三角形中的强特征顶点代替;如果不是,则三角形折叠后新顶点为步 2 中计算的新顶点位置。

②更新新生成三角形的误差矩阵  $Q_i$ ,重新计算新生成三角形和被折叠区域的邻接三角形的折叠误差,按照误差大小将三角形顺序插入三角形队列中,若折叠误差为零,则插入到堆队列中误差为零的三角形队列最后;

5. 若达到所需简化率,算法终止,否则转步 4.

同样,本文方法也很容易实现对于渐进网格的构造,图 10 给出了一个构造渐进网格的实体模型实例。



4 实例比较

本节给出的实例是在 2.8GHz Pentium(R) 4 CPU、1GB 内存 PC 上用 Visual Studio 2005 实现. 本文提出的网格简化算法基于三角形网格折叠, 用文献[11]中基于 QEM 的三角形折叠简化方法同本文的方法进行比较. 比较的实例是 Venus 模型 (2838 个顶点, 5672 个三角片), 图 11 是基于 QEM 的三角形折叠简化方法和新方法对一个拓扑比较简

单的网格模型 Venus 模型简化的结果. 图(a)为原始网格模型, 图(b)与(d)为分别用文献[11]中的方法和新方法将网格模型三角片数量简化 80%后的网格, 图(c)与(e)为简化 90%后的网格. 从图中可以看出, 用新方法简化后的网格形状较为均匀, 很少出现形状较差的三角形网格. 为了比较两种方法在简化过程中网格模型的体积变化, 网格平方体积误差由式(2)定义, 每次进行三角形折叠操作后将体积误差进行累加记录. 表 1 给出了两种方法对 Venus 模型简化的网格平方体积误差比较.

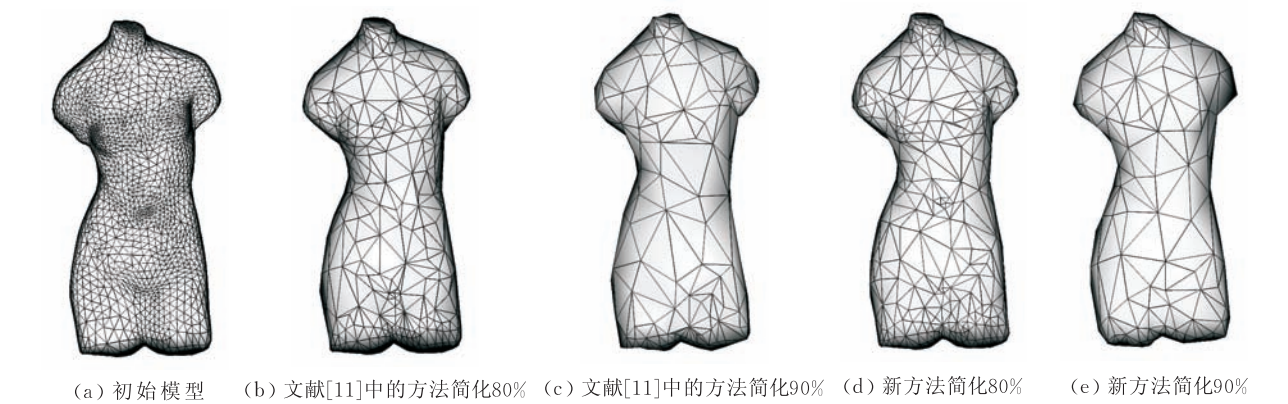


图 11 Venus 模型简化结果比较

除了平方体积误差比较之外, 本文还采用简化后网格模型和原始模型之间的 Hausdorff 距离作为误差标准进行了比较. Hausdorff 距离定义为

$$Dis_{Hausdorff}(A, B) = \max\{\min_{p_A \in A} \{d(p_A, p_B)\}\},$$

其中  $d(p_A, p_B)$  为点  $p_A$  与点  $p_B$  之间的距离.

表 1 Venus 模型网格简化误差比较

	平方体积误差	
	简化比 80%	简化比 90%
文献[11]中的方法	9.226871E-6	1.523151E-5
本文方法	6.127509E-6	1.108359E-5
	Hausdorff 距离	
	简化比 80%	简化比 90%
文献[11]中的方法	3.436112E-3	2.651125E-2
本文方法	3.002252E-3	3.553428E-2

从表 1 中可以看出, 对网格进行相同比例的简化后, 新方法产生的平方体积误差要小于基于 QEM 的三角形折叠简化方法产生的平方体积误差. 从 Hausdorff 距离的比较结果可以看出, 在简化比率不很大的情况下, 本文方法具有较好的误差效果.

图 12 给出了一个较复杂的实体模型简化结果, Triceratops 模型 (11322 个网格顶点, 22640 个三角面片) 分别用本文方法和 QSlim 方法<sup>[6]</sup>以及文献[11]中的方法将模型分别简化 70% (简化后网格数量为 6792) 和 95% (简化后网格数量为 1132), 其

中图(b)和(c)为 QSlim 方法的结果, 图(d)和(e)为文献[11]中方法的简化结果, 图(f)和(g)为新方法得到的简化结果. 从图中可以看出, 新方法得到的简化模型网格比 QSlim 方法和文献[11]中的方法得到的简化网格更加均匀, 而后两种方法部分网格过于狭长, 并且在某些细节部分 (如模型头部) 堆积了过多的三角形, 导致网格模型从视觉上看上去效果很差. 图 13 给出了三种方法在模型简化 70% 时的模型部分细节表示, 其中图(a)为 QSlim 方法, 图(b)为文献[11]中的方法, 图(c)为本文方法. 从图中可以明显看出, 前两种方法得到的三角形形状大小很不均匀, 且形状较差, 而本文方法得到的网格形状则较为理想. 表 2 给出了三种方法的平方体积误差和 Hausdorff 距离误差比较以及程序运行时间比较.

表 2 Triceratops 模型网格简化误差、运行时间比较

	平方体积误差		简化比 70% 的运行时间/ms
	简化比 70%	简化比 95%	
QSlim 方法	1.157243E-5	5.668615E-5	547
文献[11]中的方法	9.226872E-6	2.452733E-5	359
本文方法	6.127509E-6	1.825221E-5	375
	Hausdorff 距离		简化比 95% 的运行时间/ms
	简化比 70%	简化比 95%	
QSlim 方法	1.012878E-3	5.823386E-2	750
文献[11]中的方法	3.229115E-3	6.236649E-2	453
本文方法	2.556391E-3	8.732161E-2	469



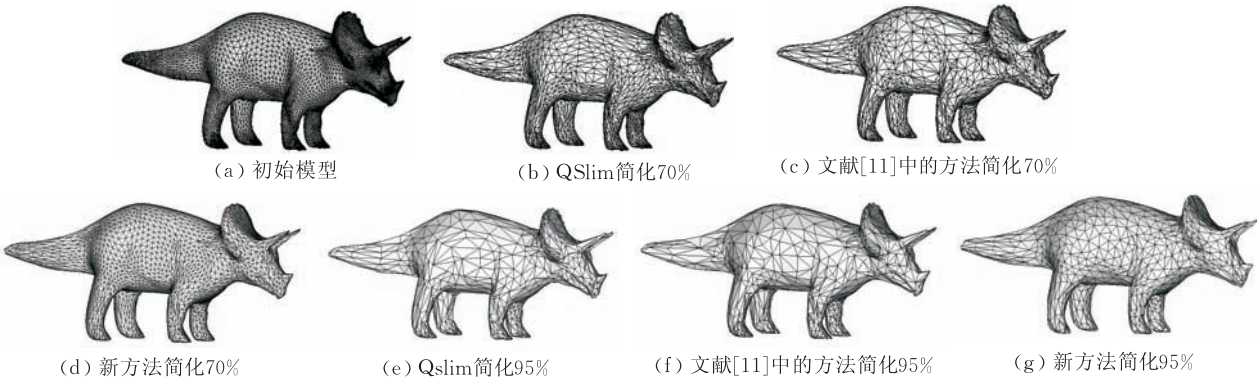


图 12 Triceratops 模型简化结果比较

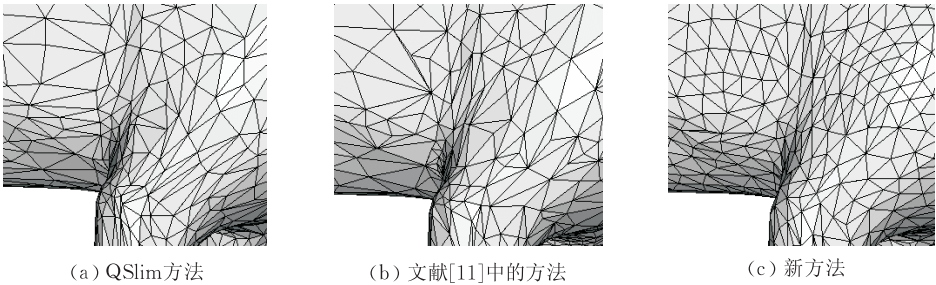


图 13 Triceratops 模型简化网格细节

从表 2 中可以看出,由于本文方法需要计算网格三角形曲率,简化时间要比基于 QEM 的三角形折叠简化方法稍长一些,但因为在网格简化中保持运算为线性复杂度,仍然具有较好的简化速度,并且要比 QSlim 方法快.从误差角度衡量,简化后的网格平方体积误差要小于其他两种方法,Hausdorff 距离误差也在网格简化比小的情况下要好于文献[11]中的方法.

5 结 论

本文提出了一种新的基于三角形折叠的网格简化方法,新方法根据三角形折叠引起的网格平方体积误差结合网格三角形的两个特征因子:几何形状因子和法向约束因子定义了三角形折叠操作的目标函数.新方法首先折叠三角形及其邻接三角形区域较为平坦、形状较差且对网格模型体积变化影响最小的三角形网格.根据三角形的曲率特征因子定义强特征三角形,对于强特征三角形,折叠后的新顶点为三角形 3 个顶点的强特征点,对于非强特征三角形,新顶点由极小化误差函数得到.实例表明,新方法简单、稳定,对于网格体积能够较好地保持,且简化后的网格质量较好.同时,与以往方法相比,在简化比率相同的情况下,新方法得到的简化网格与原始网格在视觉上差别明显小.进一步的研究工作是

将网格优化与网格简化相结合,在网格简化同时能够得到全局优化的三角形网格模型.

参 考 文 献

[1] William J S, Jonathan A Z, William E L. Decimation of triangle meshes. Computer Graphics, 1992, 26(2): 65-70

[2] Kalvin A, Taylor R. Superfaces: Polygonal mesh simplification with bounded error. IEEE Computer Graphics & Application, 1996, 16(3): 64-77

[3] Turk G. Retiling polygonal surfaces. Computer Graphics, 1992, 26(2): 55-64

[4] Zhou Kun, Ma Xiao-Hu, Pan Zhi-Geng, Shi Jiao-Ying. An improved algorithm of triangle mesh based on retiling. Journal of Software, 1998, 9(6): 405-408(in Chinese)  
(周昆, 马小虎, 潘志庚, 石教英. 基于重新划分的三角形网格简化的一种改进算法. 软件学报, 1998, 9(6): 405-408)

[5] Hoppe H, DeRose T. Mesh optimization. Computer Graphics, 1993, 27(1): 19-26

[6] Garland M, Heckbert P S. Surface simplification using quadric error metrics. Computer Graphics, 1997, 31(3): 209-216

[7] Li Ji-Tuo, Chen Guo-Dong. Mesh simplification and optimization with edge collapse and Mass-Spring model. Journal of Computer Aided Design & Computer Graphic, 2006, 18(3): 426-432(in Chinese)  
(李基拓, 陆国栋. 基于边折叠和质点-弹簧模型的网格简化优化算法. 计算机辅助设计与图形学学报, 2006, 18(3): 426-432)

- [8] Jiang Sui-Ping, Zhou Ming-Tian, Dai Ying. Mesh simplification based on normals. *Chinese Journal of Computers*, 1999, 22(10): 1074-1079(in Chinese)  
(蒋遂平, 周明天, 戴颖. 基于法向的网格简化. *计算机学报*, 1999, 22(10): 1074-1079)
- [9] Hamann B. A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design*, 1994, 11(3): 197-214
- [10] Gieng T S, Hamann B, Joy K I. Smooth hierarchical surface triangulations//*Proceedings of the IEEE Visualization*. Phoenix, Arizona, USA, 1997: 379-386
- [11] Zhou Kun, Pan Zhi-Geng, Shi Jiao-Ying. Mesh simplification algorithm based on triangle collapse. *Chinese Journal of Computers*, 1998, 21(6): 506-513(in Chinese)  
(周昆, 潘志庚, 石教英. 基于三角形折叠的网格简化算法. *计算机学报*, 1998, 21(6): 506-513)
- [12] Chang Eun-Young, Ho Yo-Sung. Three-dimensional mesh simplification by subdivided edge classification//*Proceedings of the IEEE Region 10 International Conference on Electrical and Electronic Technology*. Singapore, 2001: 39-42
- [13] Liu Xin-Guo, Bao Hu-Jun, Wang Ping-An, Peng Qun-Sheng. Volume preserved multi-resolution mesh modeling. *Chinese Journal of Computers*, 2000, 23(9): 905-910 (in Chinese)  
(刘新国, 鲍虎军, 王平安, 彭群生. 体积保持的多分辨率多边形网格的光顺造型. *计算机学报*, 2000, 23(9): 905-910)
- [14] Liu Xiao-Li, Liu Ze-Yi, Gao Peng-Dong, Peng Xiang. Edge collapse simplification based on sharp degree. *Journal of Software*, 2005, 16(5): 669-675(in Chinese)  
(刘晓利, 刘则毅, 高鹏东, 彭翔. 基于尖特征度的边折叠简化算法. *软件学报*, 2005, 16(5): 669-675)
- [15] Ji Zhong-Ping, Liu Li-Gang, Wang Guo-Jin. Feature preserving mesh simplification based on corner cutting. *Journal of Computer Research and Development*, 2006, 43(12): 2144-2151(in Chinese)  
(计忠平, 刘利刚, 王国瑾. 基于割角的保特征的网格简化算法. *计算机研究与发展*, 2006, 43(12): 2144-2151)
- [16] Peter Lindstrom, Greg Turk. Fast and memory efficient polygonal simplification//*Proceedings of IEEE Visualization*. North Carolina, USA, 1998: 279-286
- [17] Desbrun M, Meyer M, Schroder P, Alan H B. Discrete differential geometry operators for triangulated 2-manifolds//*Proceedings of Visualization and Mathematics*. Berlin, Germany, 2002: 52-58



**ZHOU Yuan-Feng**, born in 1980, Ph.D. candidate. His research interests include computer graphics, geometric modeling and processing.

**ZHANG Cai-Ming**, born in 1955, professor, Ph.D. supervisor. His research interests include computer graphics, computer aided geometric design, and medical image processing.

**HE Ping**, born in 1966, Ph.D. candidate, associate professor. Her research interests include computer graphics, computer information visualization technology.

## Background

Polygonal meshes are widely used in computer graphics, visualization and CAD/CAM systems. Triangular mesh and quadrangular mesh are two common mesh presentation forms. Triangular mesh is used more widely for its simple expression and steady topology. For more complex models, the current graphics systems are hard to handle them. As models get larger, they are more difficult to store, transfer, render and modify. One of the best solution is to represent the complex model in multi-resolution models via mesh simplification. Mesh simplification should use less Polygons to express the original models, and keep less distortion comparing with original models. Most of existing methods can't give good performance both in simplification quality and simplification speed. In this paper, a new method is presented for mesh simplification based on triangle collapse. For preserving the given model as far as possible and improving the

quality of model together, square volume error (SVE) is used to define mesh variation after simplification. Combining with triangle shape factor and normal constraint factor to define the collapse error of a triangle which is a quadratic objective function. New vertex which replaces collapsed triangle is determined linearly by minimizing this objective function. Gaussian curvature filter factor is used to judge the strong feature triangles so that the strong feature triangles can be preserved. This method is computationally efficient and easy to implement. Experimental results show that the method can produce ideal mesh simplification results. This work was supported by the National Basic Research Program (973 Program) of China (No.2006CB303102) and the National Nature Science Foundation of China (Nos.60673003, 60573180).