

基于四边形折叠的三角网格简化算法*

陈 华 鸿

(中山大学信息科学与技术学院, 广东 广州 510275)

摘 要: 通过定义三角网格模型中的两个以公共边相连的三角形构成一个空间四边形, 提出了一种新的基于这种空间四边形折叠的网格简化算法。该算法以四边形折叠为基本操作, 利用 Garland 的二次误差度量 (QEM) 做误差控制, 每次折叠操作可以减少 3 个顶点及 6 个面片, 从而实现比 Garland 的 QEM 算法、周昆等的三角形折叠算法更高的简化效率, 文中给出多个试验结果说明了该算法的有效性。

关键词: 网格简化; 三角形网络; 二次误差; 四边形折叠

中图分类号: TP391 **文献标识码:** A **文章编号:** 0529-6579 (2008) 04-0019-06

计算机图形学、虚拟现实、计算机辅助设计与医学图形等领域越来越多地应用到三维网格图形。复杂的三维网格模型对计算机的存储、绘制与传输等都提出很高要求, 然而在很多情况下高分辨率模型并不总是必要的, 这就需要对原始模型进行简化。几何元素删除法是一种重要的图形简化方法, 其原理是在保持一定的几何误差的前提下, 删除对模型几何特征影响较小的几何“图元”(点、边、面)^[1], 删除操作一直进行到模型不能再简化或达到指定近似误差为止。

国外对网格模型的几何元素删除简化的研究已有一系列成果: Schroeder 等^[2]提出了基于顶点删除的三角形网格模型简化方法; Hoppe 等^[3]建立了比较复杂的全局能量优化方程, 提出一种整体网格优化算法; Garland^[4]提出了一种基于二次误差度量 (Quadric Error Metric, 简称 QEM) 的简化算法, 得到了高质量的近似模型; Hamann^[5]根据所在顶点的主要曲率排序进行三角形消除提出了三角形折叠算法; Gieng^[6]提出通过面积加权进行三角形折叠。国内在这些方面也有一定进展: 周昆等^[7]提出了将三角形折叠与 QEM 算法相结合构造简化算法, 具有速度更快而效果较好的优点; 李现民等^[8]采用改进的蝶形细分算法来计算新顶点位置, 效果较好, 但是时间复杂度较高; 另外, 堂杰等^[9]基于边折叠操作采用计算检测点到三角面片的距离的方法控制简化过程中的 L_∞ 误差也取得了不错的简化效果。

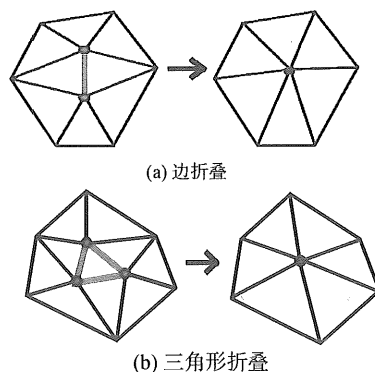


图1 边折叠操作和三角形折叠操作
Fig. 1 Edge collapse and triangle collapse

这些算法当中, Garland^[4]的 QEM 算法生成的模型质量仅次于 Hoppe^[3]的全局能量优化方法, 是一种简单高效的算法。简化的基本操作是边折叠 (实际上是点对折叠)。该算法的思想是为原始网格中的每个顶点计算一个误差矩阵, 根据误差矩阵计算网格中边折叠的代价和生成新点的位置, 然后按折叠代价从小到大进行折叠操作, 折叠生成的新点的误差矩阵为折叠边的两个顶点的误差矩阵之和。周昆等^[7]将 QEM 的误差控制方法运用到三角形折叠中, 即以三角形三个顶点的误差矩阵分别计算新顶点到相关平面的距离, 其最大值作为该三角形折叠误差, 以此进行折叠操作, 同样得到十分不错的简化效果。图 1 (a) 和 (b) 分别是 Garland 及周昆等的算法的基本思想。

本文提出了一种新的基于空间四边形 (以下

* 收稿日期: 2007-10-09

基金项目: 国家自然科学基金资助项目 (60533030)

作者简介: 陈华鸿 (1977 年生), 男, 工程师; E-mail: isschh@mail.sysu.edu.cn.

简称四边形)折叠的网格简化算法(如图2所示),并利用Garland的QEM有效地控制简化网格与原始网格间的误差。与Garland算法的每次边折叠减少1个顶点与2个三角形面片及周昆等算法的每次三角形折叠减少2个顶点和4个三角形面片相比,本文算法一次四边形折叠能减少3个顶点及6个三角形面片,相当于3次边折叠和1.5次三角形折叠的效果,因此达到同样的简化目标只需要较少的折叠次数,具有更高的折叠效率。

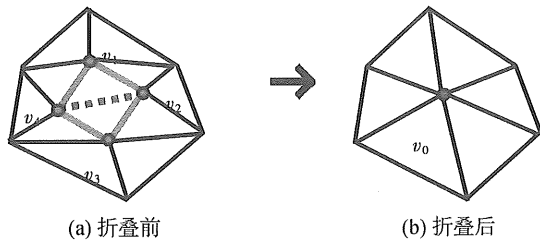


图2 四边形折叠操作,点 (v_1, v_2, v_3, v_4) 收缩成点 v_0

Fig. 2 Operation of quadrangle collapse, four vertices (v_1, v_2, v_3, v_4) are contracted into one vertex v_0

1 基于四边形折叠的网格简化算法

由于算法中涉及四边形、公共边和边界边等多个相关概念,为清楚起见,首先引入一些基本概念。

1.1 基本概念

定义1 空间一组三角形,沿公共边及顶点处相邻接,把这样的一组三角形定义为三角形网格 TM , TM 可由顶点集 $V = (v_1, v_2, \dots, v_n)$ 和三角形结合 $T = (T_1, T_2, \dots, T_m)$ 所组成的二元组 (V, T) 来表示。

定义2 对 TM 中任一条边,如果该边只为一个三角形所享有,则称该边为边界边,边界边上的顶点称为边界顶点,边界边所在的三角形被称为边界三角形,含一个或以上的边界三角形的四边形称边界四边形。

定义3 对 TM 中任一条边,如果该边为两个三角形所享有,则共享该边的两个相连的三角形构成一个四边形 $QUAD_i$ (如图2(a)所示),该边称为这个四边形 $QUAD_i$ 的公共边或内部边,而构成此四边形的其余边称为这个四边形的外部边。

定义4 对 TM 中任一顶点 v_i ,所有以 v_i 为一个顶点的三角形 T_{ik} 构成的集合,称为与顶点 v_i 相关的三角形集合 P_i 。

定义5 对 TM 中任一四边形 $QUAD_i$,与 $QUAD_i$ 的四个顶点相关的三角形集合的并集,称为与四边形 $QUAD_i$ 相关的三角形集合 C_{Qi} 。

定义6 如果存在三个三角形 T_{i0}, T_{i1}, T_{i2} 属于 C_i ,满足条件: T_{i1} 和 $T_{i(i+1) \bmod 3}$ 是相邻的,其中 $i = 0, 1, 2$,那么称 C_i 是周期^[5,7]的,否则称为非周期的。周期情形中三个三角形相交的公共点称为周期点。

1.2 四边形的检索

算法中的四边形并不是三角形网格模型的基本图元,而是由两个相连的三角形组合而成,在执行算法的时候,四边形可以通过网格模型的内部边得到,每条内部边都对应两个相连的三角形,即都唯一对应一个由其组合成的四边形,该边则为这个四边形的公共边;由于边界边只对应一个三角形,即不存在以之为公共边的四边形,因此,在通过边检索四边形的时候,需要排除边界边。由图2可见,四边形 $QUAD\{v_1, v_2, v_3, v_4\}$ 是由两个三角形 $T\{v_1, v_2, v_4\}$ 及 $T\{v_2, v_3, v_4\}$ 通过公共边 $E\{v_2, v_4\}$ 连接而成,边 $E\{v_1, v_2\}$ 、 $E\{v_2, v_3\}$ 、 $E\{v_3, v_4\}$ 、 $E\{v_1, v_4\}$ 为四边形 $QUAD\{v_1, v_2, v_3, v_4\}$ 的外部边。

1.3 四边形折叠操作

简化的基本操作是四边形折叠,在图2(a)中,对四边形 $QUAD\{v_1, v_2, v_3, v_4\}$,找到与之相关的三角形集合 C_{Quad} ,对 $QUAD\{v_1, v_2, v_3, v_4\}$ 执行折叠操作,则其四个顶点 v_1, v_2, v_3, v_4 收缩为图2(b)中的 v_0 ,执行该次折叠操作,共减少3个顶点和6个三角面片。

1.4 误差的计算

简化的误差即四边形折叠的代价,本文引入Garland^[4]算法中的二次误差度量,把新顶点到旧顶点相连平面的距离的平方之和作为顶点移动后引起的误差。点 v 到某平面的距离的平方计算如下:

$$\Delta(v) = \Delta([v_x, v_y, v_z, 1]^T) = \sum_{pplanes(v)} (p^T v)^2 = \sum_{pplanes(v)} v^T (pp^T) v = v^T \left(\sum_{pplanes(v)} (K_p) \right) v = v^T Q(v_a) v \quad (1)$$

其中, K_p 为 4×4 对称矩阵:

$$K_p = pp^T = \begin{pmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{pmatrix} \quad (2)$$

由于四边形 $QUAD_i$ 折叠需要同时考虑4个顶点移动后造成的误差,以进行折叠操作的四边形4个

顶点 v_a 、 v_b 、 v_c 、 v_d 的误差之和作为四边形折叠到新顶点 v_0 的误差, 即

$$\varepsilon(QUAD_i) = \sum \Delta(v) = v^T Q_0 v = q_{11}x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x + q_{22}y^2 + 2q_{23}yz + 2q_{24}y + q_{33}z^2 + 2q_{34}z + q_{44} \quad (3)$$

1.5 新顶点位置

进行四边形折叠操作后, 新顶点的位置可以有如下方法选择:

1) 子集位置

即四边形四个顶点的其中一个顶点作为新顶点位置, 子集位置由于不产生新顶点, 需要存储及处理开销较小, 适合像移动终端设备这样的对简化效果要求不高, 但对处理开销相对严格的场合。

2) 最佳位置 (依据某种评价指标合成的新顶点)

由 (3) 式可知误差函数 $\varepsilon(QUAD_i)$ 是二次函数, 因此其最小值是个线性问题。由极值的性质, 我们对 (3) 式中的 x , y , z 求偏导数, 可得方程组: $\partial\Delta(v)/\partial x = \partial\Delta(v)/\partial y = \partial\Delta(v)/\partial z = 0$, 写成矩阵的形式, 即

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} v = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (4)$$

如果 (4) 式有解, 则新顶点 v 的位置由下式唯一给出:

$$v = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (5)$$

否则, 可以在四边形 $QUAD_i$ 的4个顶点、公共边中点和外部边的中点这9个位置中选取一个作为新顶点的位置。

1.6 算法归纳

基于四边形折叠的网格简化算法是采用新的四边形折叠作为基本简化操作, 以 QEM 作为误差度量, 具体的算法如下描述:

算法 基于四边形折叠的网格简化算法

步骤1 对原始网格中的每条非边界边构成的四边形 $QUAD_i$, 计算其误差矩阵 Q ;

步骤2 对原始网格中的每个四边形 $QUAD_i$, 通过其误差矩阵 Q 计算其折叠后生成的最优收缩目标点 v , 并通过 vQv^T 计算其折叠误差并排序入栈, 栈顶为最小误差值;

步骤3 选取误差最小的四边形进行折叠;

步骤4 更新与被折叠四边形相关的所有相关信息;

步骤5 如果四边形序列为空或已经达到简化目标, 转到步骤6, 否则转步骤3;

步骤6 结束。

2 相关问题

2.1 周期处理

周期是三角网格中的特殊结构^[5], 某些时候被认为是模型的属性所在, 如果直接删除将可能引起某些面片的重叠。因此, 必须在折叠前进行周期处理, 周期处理的情况有以下两种:

1) 四边形包含周期点 (如图3(a)所示)。这种情形下如果进行折叠只影响移去的边和顶点的数量, 因此此种情况可以继续四边形折叠。

2) 在四边形的 C_i 集中包含周期点 (如图3(b)所示)。 $T_1\{v_2, v_6, v_5\}$ 、 $T_2\{v_3, v_6, v_5\}$ 、 $T_3\{v_2, v_3, v_6\}$ 组成以 v_6 为周期点的周期, 当四边形 $QUAD\{v_1, v_2, v_3, v_4\}$ 进行折叠时, v_2, v_4 收缩成点 v_0 , 因此原来周期中的三个三角形 T_1, T_2, T_3 退化为一个边界三角形 $T\{v_0, v_5, v_6\}$ ($T\{v_0, v_5, v_6\}$ 与 $T\{v_0, v_5, v_7\}$ 不共面), 并且形成新的边界边 $E\{v_0, v_6\}$, $E\{v_5, v_6\}$ 以及与三面同时相连的特殊边 $E\{v_0, v_5\}$, 这种情况下如果进行折叠, 将会引起面片重叠, 因此必须进行进一步的处理。

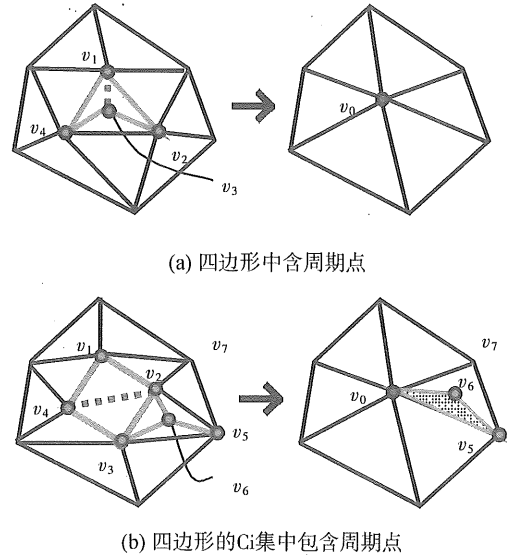


图3 周期处理的两种情况

Fig. 3 Two situations of the cycle process

防止第2种情况产生面片重叠, 几种适用于三角形折叠的周期处理方法同样也适用于本文算法: Gieng 等^[6]提出了一种交换边方法解决周期问题, 即把周期中的三角形与周期外的三角形的公共边进

行交换以除去周期; $W_u^{[10]}$ 则认为周期是网格模型的特殊属性所在, 因此在某个折叠涉及周期时, 则判断该操作为非法, 而取消该次操作; 此外, 也可以采用顶点删除的方法来解决, 即直接去除周期中价为 3 的顶点。另外, 如果碰到 C_{Q_i} 中存在价为 4 的点, 则认为 C_{Q_i} 中隐含周期^[6], 亦应该取消该次折叠, 以避免折叠后产生真正的周期。

2.2 边界处理

对于非闭合的网格模型, 存在含一条或多条边的四边形, 与其相关的三角形集合不能构成一个三角形环, 如忽略边界的情况, 则在简化过程中会造成边界退化。为避免这种退化, 我们在该三角形集合的每条边界边上作一个通过该边界边并与该边界边所在的边界三角形垂直的平面。

如图 4 示, $QUAD\{v_a, v_b, v_c, v_d\}$ 是以 v_a, v_c 为公共边的边界四边形, v_c, v_d 是网格模型的边界边, 作平面 a 垂直三角形 $T\{v_a, v_c, v_d\}$, 并与 $T\{v_a, v_c, v_d\}$ 相交于边 v_c, v_d , 因此计算 v_d 及 v_c 两顶点的误差矩阵 Q 时, 就需要考虑新顶点到平面 a 的距离, 由于计算机结果与这个垂直平面的选取无关, 这样做就将点到边界边的距离加入到误差测度之中, 从而保证新顶点不会偏离边界, 保持了模型的边界。另外, 在一定次数限制下, 对边界四边形进行惩罚加权可以起延迟折叠的作用, 也能进一步保持边界特征。

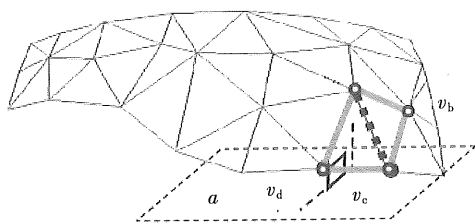


图 4 作平面 a 通过边界边 $E\{v_c, v_d\}$ 并垂直于边界三角形 $T\{v_a, v_c, v_d\}$

Fig. 4 Generation of a perpendicular plane on running through the boundary edge $v_d v_c$

2.3 一致性问题

在进行四边形折叠操作的时候, 有可能会出现三角形相互重叠或者三角面方向翻转的现象。用以下方法解决该问题: 当对某四边形进行折叠的时候, 比较与此四边形相关联三角面的法向在折叠前后的变化情况, 如变化过大, 则不对该四边形进行折叠。

3 实验结果

3.1 数据结构

由于算法涉及点、边、面和四边形等多种数据, 因此, 需要设计清晰合理的数据结构, 提高算法的效率。本算法在实际编程中使用的数据结构有: 顶点表、三角形表、 P_i 表和误差矩阵表, 每次折叠操作得到的新顶点坐标直接放入原顶点表代替原数据。

3.2 实验结果

本算法已经用 C++ 语言实现, 图 5 (a) - (d) 为马模型的原始模型及系列简化模型, 从图中可以看出采用了 Garland 的二次误差的作为误差控制的新算法较好地完成了对初始网格的向下采样, 网格的细节得到保留。从图 5 (b), (c) 分别可以看出: 模型简化至 80% 时, 简化模型仍比较接近初始模型; 简化至 98% 时, 模型的也能保持一定的轮廓特征。

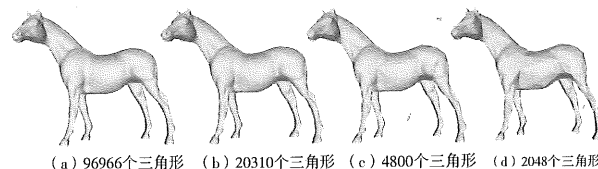


图 5 四边形折叠简化效果, 简化程度分别为 0%、80%、95%、98%

Fig. 5 Simplified model using quadrangle collapse, triangles' decimations are 0%, 80%, 95%, 98%

与边折叠和三角形折叠方法比较, 本文算法的特点是每次折叠可以减少更多的三角面片, 从而达到一定的简化效果只需要较少的简化次数; 但是这也可能同样意味着简化品质的降低。实验中就新算法与 Garland 算法产生的简化图形进行了比较: 对于多数模型, Garland 算法由于只执行点对折叠, 每步简化只作少量变动, 在简化品质上表现十分出色, 由图 6 (b), (c) 可以看出, Garland 算法具有稍好的简化品质; 由于新算法同样是采用了 Garland 的二次误差度量, 有效地对误差进行控制, 因此, 新算法的简化效果也接近于 Garland 算法, 实际测试中, 某些模型采用 Garland 算法或采用新算法会有不同的简化效果: 例如在苹果模型的简化中, Garland 算法得到的简化模型同样在果身部分有较好的近似性, 但是在苹果的枝干附近区域, 新算法得到的简化模型效果则更为接近原始模型的“均匀”特性。

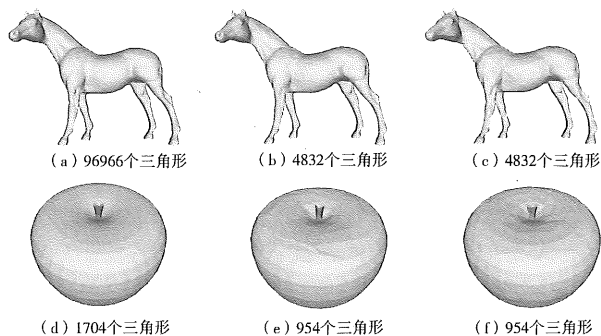


图6 Garland算法与本文算法简化效果比较

其中(a),(d)为原始网格模型,(b),(e)为Garland算法简化效果,(c),(f)为本文算法简化效果。马、苹果模型简化程度分别为95%、44%、24%

Fig. 6 Performances between Garland's and the proposed algorithm above.

(a), (d) are the original models. (b), (e) are simplified using Garland's algorithm whereas, (c), (f) using proposed algorithms. Horse and apple models are decimated to 95% and 44%, respectively

3.3 运行时间

与QEM算法一样,新算法分为初始化阶段(步骤1~步骤2)和简化阶段(步骤3~步骤6)两个部分。假设模型原有 n 个面片,简化目标为 m 个面片,一次迭代中移去面片数目限定为常数 k ,最大点度限定为常数 g 。在初始化阶段,构造所有的二次矩阵需要花费 $O(n)$ 时间,把所有三角形折叠误差排序入栈需要 $O(n \log n)$,因此总共初始化时间复杂度为 $O(n \log n)$ 。简化阶段单次迭代的代价为 $O(\log(n - k_i))$,迭代的次数为 $(n - m)$,所以简化阶段总的代价为: $\log n + \log(n - k) + \log(n - 2k) + \log(n - 3k) + \dots + \log m = \log n! - \log m! = O(n \log n - m \log m)$ 。因此算法总的时间复杂性为 $O(n \log n)$,与Garland的QEM算法和周昆等三角形折叠算法处于同等的数量级。

但是由于每次减少三角形面片数目不同,Garland算法需要进行 $(n - m)/2$ 次折叠,周昆等算法需要进行 $(n - m)/4$ 次折叠,本算法每次四边形折叠操作,可以减少6个三角形面片和3个顶点,只需进行 $(n - m)/6$ 次折叠,大大减少了排序次数及其时间损耗,因此实际简化速度比Garland算法和周昆等^[7]的算法更快。

由表1可以看出本文算法的执行时间快于周昆等的算法,由于周昆等的三角形折叠算法快于Garland算法,因此,新算法也同样快于Garland算法。

表1 周昆等的算法和新算法的执行时间比较

Tab. 1 Comparison of implementation time between Zhou et al.'s and the quadrangle collapse algorithm

原始模型 (三角形个数)	简化模型 (三角形个数)	周昆等算法 ^[7] s	新算法 s
Shark (734)	560	0.293	0.194
Cow (5804)	1448	11.399	9.151
Ball (5000)	508	10.609	8.906
Apple (1704)	956	2.115	1.754

4 结论

本文在边折叠和三角形折叠这些基于几何元素删除的图形简化算法基础上,延伸出一种新的间接地基于四边形折叠的网格简化算法。该算法简单明了,由于每次折叠操作删除的几何要素比边折叠和三角形折叠更多,因此只需要经过较少的简化次数便可以达到简化目的,在简化速度上亦优于上述两者,适合大型复杂模型的简化或对简化速度要求高于品质要求的场合,例如移动终端的实时交互。另外,由于误差计算采用了Garland^[4]算法所提出的二次误差,因此简化模型的形状和拓扑得到了较好的保持,同时也很容易实现结合颜色和纹理属性的图形简化。

下一步的研究工作:一方面可以结合边折叠、三角形折叠及本文提出的四边形折叠各自速度及效果上的优点,构造四边形与两者相结合的简化质量及速度都更理想的算法。另一方面,通过分析已有多种折叠简化算法的共性,可以进一步归纳出基于多边形折叠的图形简化算法。

参考文献:

- [1] 何晖光,田捷,张晓鹏,等. 网格模型化简综述[J]. 软件学报,2002,13(12):2215-2224.
HE H G, TIAN J, ZHANG X P, et al. A survey on mesh simplification [J]. Journal of Software. 2002, 13(12): 2215-2224.
- [2] SCHROEDER W J, ZARGE J A, ROSENSEN W E. Decimation of triangle meshes [J]. Computer Graphics, 1992, 26(2): 65-70.
- [3] HOPPE H, DEROSE T, DUCHAMP T, et al. Mesh optimization [C]. Proceedings of the SIGGRAPH'93, 1993:19-26.
- [4] GARLAND M. Quadric-based polygonal surface simplification [D]. Carnegie Mellon University, 1999.
- [5] HAMANN B. A data reduction scheme for triangulated Surface [J]. Computer Aided Geometric Design, 1994, 11(2):197-214.
- [6] GIENG T, HAMANN B, JOY K I, et al. Smooth hierar-

- chical surface triangulations [C]. Proceedings of the IEEE Visualization 97. San Francisco: IEEE Computer Society Press, 1997: 379 - 386.
- [7] 周昆, 潘志庚, 石教英. 基于三角形折叠的网格算法 [J]. 计算机学报, 1998, 21 (6): 506 - 513.
- ZHOU K, PAN Z G, SHI J Y. Mesh simplification algorithm based on triangle collapse [J]. Chinese Journal of Computers, 1998, 21 (6): 506 - 513.
- [8] 李现民, 李桂清, 张小玲, 等. 基于子分规则的边折叠简化方法 [J]. 计算机辅助设计与图形学学报, 2002, 14(1): 8 - 13.
- LI X M, LI G Q, ZHANH X L, et al. Edge collapse simplification based on subdivision [J]. Journal of Computer-aided Design & Computer Graphics, 2002, 14(1): 8 - 13.
- [9] 堂杰, 张福炎. 一种基于误差控制的网格多分辨率模型生成算法 [J]. 计算机学报, 2005, 28(9): 1534 - 1540.
- TANG J, ZHANG F Y. An algorithm for generating multiresolution model of mesh based on error control [J]. Chinese Journal of Computers, 2005, 28(9): 1534 - 1540.
- [10] WU J H, HU S M, TAI C L, et al. An effective feature-preserving mesh simplification scheme based on face contraction [J]. Proceedings of Pacific Graphics, 2001: 12 - 21.

Mesh Simplification Algorithm Based on Quadrangle Collapse

CHEN Hua-hong

(School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275, China)

Abstract: This paper presents a method to define the quadrangle in the triangle mesh and puts forward a new mesh simplification algorithm based on quadrangle collapse. The algorithm uses iterative collapse of quadrangle to simplify models and maintains surface error approximations using quadric metrics. Three vertices and six faces are collapsed every simplification. The algorithm can rapidly produce high quality approximations of polygonal models by fewer times of collapse than Garland's and ZHOU's algorithms. Experiment results demonstrate the efficiency of the algorithm.

Key words: mesh simplification; triangle mesh; quadric error metrics (QEM); quadrangle collapse