

文章编号:1001-9081(2010)09-2431-03

保持外形特征的网格简化算法

薛 峰,袁成凤

(合肥工业大学 计算机与信息学院,合肥 230009)

(anyunum2@163.com)

摘要:目前许多网格简化算法在大幅度减少算法简化时间的同时,忽略了模型简化后的仿真质量。为此提出一种基于外形特征保持的网格简化算法:依据顶点坍塌值由小到大的次序将顶点排入网格简化优先队列中;每次选择队首顶点向其折叠点合并,并且更新队列中受影响的顶点信息,再对队列重新排序。此算法数据结构存储简单,大规模简化后的模型仍然较好保持了原模型的视觉特征。

关键词:仿真质量;网格简化;优先队列;坍塌值;视觉特征

中图分类号: TP391.41 **文献标志码:**A

Mesh simplification algorithm based on keeping shape feature

XUE Feng, YUAN Cheng-feng

(School of Computer and Information, Hefei University of Technology, Hefei Anhui 230009, China)

Abstract: Currently, in order to reduce the consumption time in simplifying model, lots of mesh simplification algorithms ignore the simulation quality of simplified model. A mesh simplification algorithm based on keeping profile feature was proposed. All the vertices in the model were stored in a priority queue, which sorted vertices ascending according to their collapse values. The top vertex in the queue was to be folded first and updated the vertices which were affected, then the queue was sorted again. Compared to other algorithms, the data structure of this algorithm was simple to store and the simplified model also maintained the visual characteristics of the original model better.

Key words: simulation quality; mesh simplification; priority queue; collapse value; visual feature

0 引言

在计算机图形学的实体仿真技术中,三角形网格是实体描述的基本单位,任意复杂的对象实体都可用三角形面片表示、构造。随着虚拟现实和动漫产业的迅速发展,3D模型变得越来越复杂,尽管越精细的模型越能反映实体的细节特征,但同时会增加模型的存储、计算以及绘制成本,因此需要对原始模型进行简化,使模型的精细度和操作时间达到一个折中。好的简化网格模型既能很好地表现实体模型,又能够较好地保持原模型的拓扑结构和重要外形特征。

在网格简化算法方面,国内外学者做出了许多突出成就,这些简化算法大致可以分为三类。

1) 三角形折叠法。网格中某个三角形折叠成一个新顶点,同时删除与此三角形共边的相邻三角形,如文献[1-2]的算法,此类算法简化模型的速度非常快,但将三点聚集成一点易在模型简化的过程中积累误差。

2) 顶点删除法。如文献[3-4],算法反复选择候选点删除,并删除与该点相关的三角面,再对剩余空洞模型三角形化,该过程会增加算法的时间复杂度。

3) 边折叠算法。如文献[5-7],文献[6-7]中的算法均依据各自算法中定义的顶点重要度来决定边折叠的方向,将边的一端点向另一端点折叠。

基于误差度量法属于边折叠算法的一种,如文献[8-10]。Garland 的基于二次误差的网格简化(Quadric Error Metric, QEM)算法^[8]是一种经典网格简化算法,算法简化速度快,仿真效果好,被后续研究者大量引用;文献[9]在 QEM

算法的基础上,为顶点加入尖特征值,克服了 QEM 算法简化模型中网格过于均匀的不足,并且保持了模型的尖锐特征;文献[10]为了在二次误差中体现局部表面变化,将三角形重要度作为权值嵌入其中。但是基于二次误差的网格简化算法在低分辨率的状态下不能有效显示模型的一些细节特征,存储矩阵数据结构也较复杂。

在上述文献的基础上,本文提出一种新的网格简化算法。算法根据三角形面的法矢量(影响网格面视觉效果的主要因素)以及点的坍塌值(保持模型特征)对模型数据进行抽取和简化,能够较好地实现高度逼真与高度简化比的统一。

1 算法概述

1.1 算法关键问题

本文算法属于边折叠算法,算法的基本原理是:确定模型中坍塌值最小的点(冗余点),并根据一定法则选取与该冗余点相关的“待折叠边”,将该冗余点向待折叠边的另一端点折叠,实现网格的简化。图 1 展示了本文算法简化的基本过程,可见本文算法的关键步骤:1)冗余点 P_1 的选取;2)待折叠边 P_1P_2 的选取(图 1 中,由 P_1 向 P_2 折叠)。

图 1 所示的边折叠算法简化过程中,确定了 P_1 为模型的冗余点, P_2 称为 P_1 的“折叠点”。实际上,本文认为图 1 所示的边折叠过程是将冗余点 P_1 向其对应的“折叠点” P_2 折叠、合并过程。因此,边折叠网格简化算法的关键问题可转化为:

- 1) 如何选取网格中的冗余点 P_1 ;
- 2) 如何确定冗余点对应的折叠点 P_2 (即冗余点 P_1 的合并方向)。

收稿日期:2010-03-08;修回日期:2010-04-23。

作者简介:薛峰(1978-),男,安徽合肥人,副教授,主要研究方向:计算机图形图像学; 袁成凤(1985-),女,安徽合肥人,硕士研究生,主要研究方向:计算机图形学。

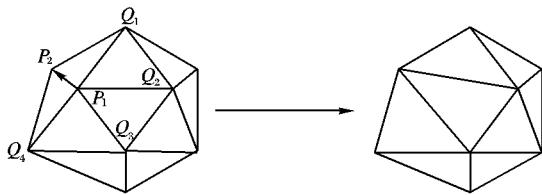


图1 点对折叠

1.2 算法基本流程

本文提出一种能够有效保持网格外形特征的网格简化算法,通过定义一种“顶点坍塌值”和“边坍塌值”度量方法,分别实现图1中冗余点 P_1 的选取和待折叠边 P_1P_2 的确定,最终实现网格的简化,具体简化流程如图2。

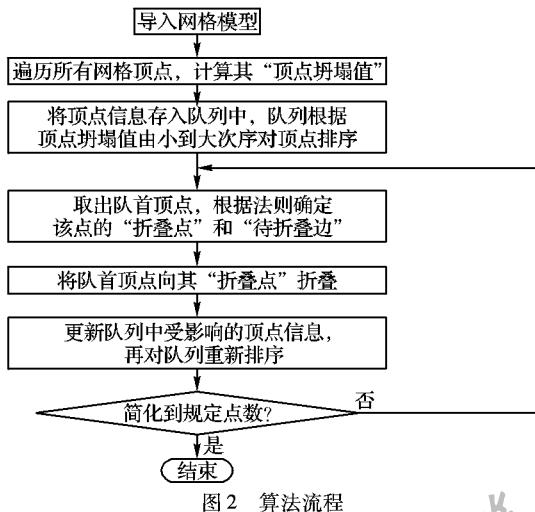


图2 算法流程

1.3 算法时间复杂度

本文计算模型中所有顶点的“坍塌值”,根据顶点坍塌值将顶点由冒泡排序算法排入队列中(时间复杂度约为 $O(n^2)$)。若需要从模型中删除 k 个顶点,每次取出队首顶点向其“折叠点”折叠,同时更新该点的邻接点信息,再由冒泡算法对队列重新排序,因此本文算法的时间复杂度约为 $O(kn^2)$ 。

2 算法核心步骤

由图2可知,本文算法的核心是:1)如何选取网格模型中的冗余点;2)如何确定冗余点对应的“折叠点”,即“折叠边”的确定。

2.1 兀余点的选取

根据什么样的准则选取冗余点,合并折叠边,并保证简化网格能够较大程度保持原始网格的外形特征是所有网格简化算法的目标。根据对相关网格简化算法的调研分析,本文提出一种能够有效保持网格外形特征的,基于“顶点坍塌值”的冗余点选取方法,坍塌值最小的顶点即为简化操作最先折叠的点。其中,“顶点坍塌值”的计算公式如式(1):

$$Cost(U) = Cur(U) \times \min\{Cost(U, U_i)\} \quad (1)$$

“顶点坍塌值”由顶点近似曲率值 $Cur(U)$ 与顶点最小邻接边坍塌值 $\min\{Cost(U, U_i)\}$ 的乘积确定。

2.1.1 顶点曲率值的计算

式(1)中 $Cost(U)$ 表示顶点 U 的近似曲率值,它反映模型在点 U 处的变化强度。

图3中,顶点 P 有5个邻接点 $P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), P_3(x_3, y_3, z_3), P_4(x_4, y_4, z_4), P_5(x_5, y_5, z_5)$,假设有一点 O ,该点坐标为 (x, y, z) ,点 O 到点 P 各个邻接点的距离之和 d 为:

$$d = \sum_{i=1}^n \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \quad (2)$$

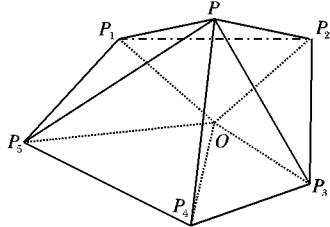


图3 点曲率定义

要求得 d 的最小值,即令 $d = 0$,再对 x, y, z 分别求偏导得出 O 点最佳坐标。连接点 O 与顶点 P 的 n 个邻接点构成 n 个平面,再求出顶点 P 到这 n 个平面的平均距离 $h(P)$,经分析可知, $h(P)$ 表示顶点 P 到其各个邻接点构成的平均平面的距离, $h(P)$ 越小, P 点与其邻接点构成的区域越平坦。

但是, $h(P)$ 值不足以表现顶点 P 处的曲率变化,在图3中,顶点 P 的单位法向量记为 \vec{q} (顶点单位法向量根据该点邻接三角形单位法向量与三角形面积的加权平均求得),邻接三角形单位法向量集合记为 $U = \{\vec{u}_1, \vec{u}_2, \vec{u}_3, \vec{u}_4, \vec{u}_5\}$,若 U 中向量与 \vec{q} 之间夹角的最大值越大,则点 P 邻接三角形的向量方向越发散,模型位于点 P 处越弯曲,定义顶点 P 的弯曲度 $w(P)$ 为(若点 P 有 n 个邻接三角形):

$$w(P) = \max\{a \cos(\vec{q}, \vec{u}_i)\}; \quad 1 \leq i \leq n \quad (3)$$

用 P 点的 h 值和 w 值共同反映顶点 P 的曲率变化,因此顶点 P 的近似曲率值为:

$$Cur(P) = w(P) \times h(P) \quad (4)$$

顶点曲率值越小,该点对模型的影响越小。

2.1.2 边坍塌值的计算

顶点曲率值反应模型在该点周围的尖锐程度,依据该特征值能决定顶点是否删除,但若结合本文中定义的边坍塌值,则更好反映边折叠操作对周围区域造成的影响,在本文算法中,边坍塌值由边长和边的重要度共同确定。

如图4所示,边 (U, V) 折叠后,模型中三角形 $(U, P_2, P_3), (U, P_1, P_2), (U, P_1, P_4)$ 更新为 $(V, P_2, P_3), (V, P_1, P_2), (V, P_1, P_4)$ 。点 U 和点 V 的公共三角形 $(U, V, P_3), (U, V, P_4)$ 将从模型中删除,点对 (U, V) 折叠前后所有三角形(该边公共三角形除外)的单位法向量之间夹角总和定义为边 (U, V) 的重要度 $I(U, V)$:

$$I(U, V) = \frac{\sum_{i=1}^{num-2} a \cos(\vec{u}_i, \vec{v}_i)}{(num - 2)} \quad (5)$$

其中: num 为点 U 的邻接三角形数, \vec{u}_i 为边 (U, V) 折叠前 U 的邻接三角形单位法向量, \vec{v}_i 为边折叠后顶点 V 的新增邻接三角形单位法向量, \vec{v}_i 与 \vec{u}_i 一一对应。边长和边的重要度组成边的坍塌值 $Cost(U, V)$:

$$Cost(U, V) = \|U - V\| \times I(U - V) \quad (6)$$

2.2 折叠边的确定

选取了模型的冗余点后,算法的关键是确定边折叠方向。边坍塌值反应边折叠前后三角形的变化程度,边坍塌值越小,该边折叠对周围模型造成的影响越小,因此规定最小坍塌值的边为模型的冗余边。

由式(6)计算冗余点所有邻接边坍塌值,最小坍塌值的邻接边对应的邻接顶点称为该冗余点的“折叠点”,且该邻接边称为“折叠边”,将冗余点向其“折叠点”折叠、合并。

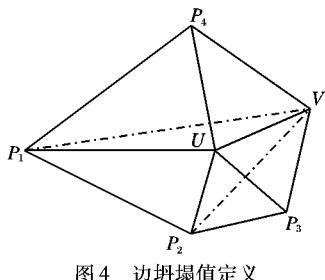


图4 边坍塌值定义

3 实验结果及分析

本文在微机系统(CPU为P4 2.4 GHz; RAM为512 MB)下,结合VC6.0++和OpenGL的编程环境实现了本文算法,实验选取的经典模型是具有代表性的牛模型、马模型以及骨骼模型。

牛原模型共有2904个点,5804个三角形,简化结果如图5所示。

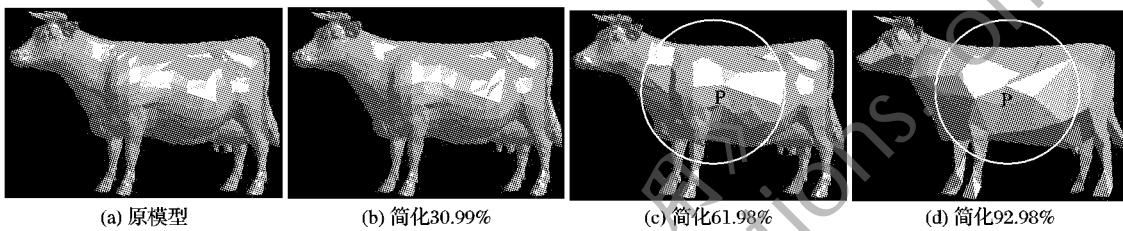


图5 牛模型简化过程

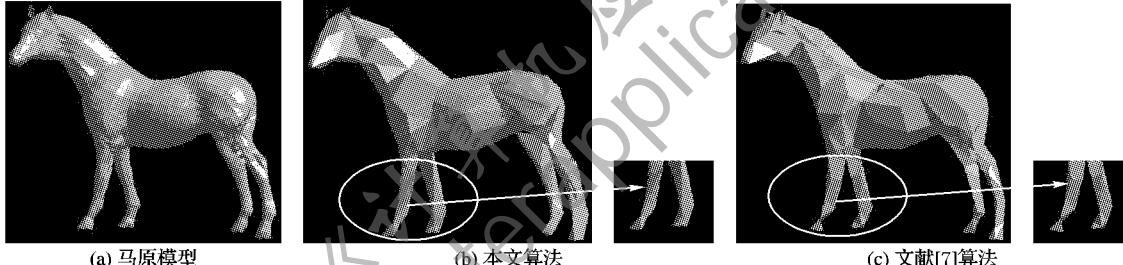


图6 马模型简化对比

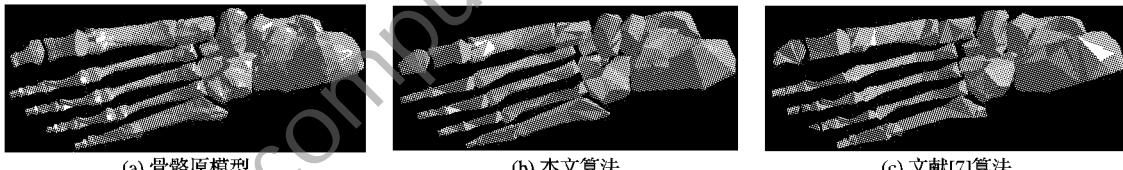


图7 骨骼模型简化对比

4 结语

本文算法是在了解国内外多个模型简化算法的基础上所做的研究,算法能有效地实现表面曲率变化不大的网格模型的简化,并保留原模型的形状特征和拓扑结构。实验表明,该算法在简单性、简化效率和简化质量方面均取得较好效果,在现实应用中,三维物体碰撞检测,游戏场景制作以及计算机动画方面都可以利用该算法对模型进行前期简化操作。

但是本文算法也存在一些不足之处,如在简化过程中未能有效处理边界点和边界三角形,算法不能简化带有空洞的模型。下一步的工作主要针对本文算法的不足之处进行改进,提出更好的网格简化算法。

参考文献:

- [1] HAMANN B. A data reduction scheme for triangulated surface [J]. Computer Aided Geometric Design, 1994, 11(2): 197–214.
- [2] PAN ZHIGENG, ZHOU KUN, SHI JIAOYING. A new mesh simplification algorithm based on triangle collapses [J]. Journal of Computer Science and Technology, 2001, 16(1): 57–63.
- [3] SCHROEDER W J, ZARGE J A, LORENSEN W E. Decimation of triangle meshes [J]. Computer Graphics, 1992, 26(2): 65–70.
- [4] 高山, 卢汉清, 周万宁. 基于细节的自适应网格简化[J]. 计算机辅助设计与图形学学报, 2003, 15(9): 1122–1127.
- [5] HOPPE H. Progressive meshes [C]// Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1996: 99–108.
- [6] 曾芬芳, 周琴, 姚煜. 基于边折叠的网格简化算法及其应用[J]. 计算机应用, 2002, 22(1): 7–8.
- [7] 王健, 何明一. 基于边顶点重要度的网格简化算法及应用[J]. 计算机应用研究, 2004, 21(4): 243–245.
- [8] GARLAND M, HECKBERT P S. Surface simplification using quadric error metrics [C]// Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1997: 209–216.
- [9] 刘晓利, 刘则毅, 高鹏东, 等. 基于尖特征值的边折叠简化算法[J]. 软件学报, 2005, 16(5): 669–675.
- [10] 杜晓辉, 尹宝才, 孔德慧. 基于加权二次误差测度的边折叠简化算法[J]. 北京工业大学学报, 2007, 33(7): 731–736.