

一种改进的多属性模型压缩算法

冯 钦,倪桂强,罗健欣

(解放军理工大学 指挥自动化学院,江苏 南京 210007)

摘 要:传统的压缩算法在实际模型上应用时经常出现“空洞”和面重叠等问题,通过在 x 、 obj 、 $3ds$ 、 max 等模型实例上实验发现,出现上述问题的主要原因在于几何相似点,这些点在压缩过程中出现了偏差。文中提出了一种改进的压缩算法,该算法首先利用八叉树切分空间确定几何相似点,然后针对几何相似点采用一系列策略,使它们在压缩过程中仍然保持良好的几何连续性。对比实验表明,该算法不仅能克服传统算法出现的问题,还能更好地保持模型的整体性与连贯性。

关键词:模型简化;二次方拟合;多属性;渐进网格

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2012)04-0037-04

An Improved Surface Simplification Algorithm with Multi-Attributes

FENG Qin, NI Gui-qiang, LUO Jian-xin

(Institute of Command Automation, PLA University of Science & Technology, Nanjing 210007, China)

Abstract: Applying traditional algorithm in real models often leads to some problems such as faces overlaps and “holes”. Found this was because of those geometrically similar points which often resulted in wrong positions during simplification. Then propose a new algorithm. This algorithm searches geometrically similar points by the way of partitioning space with octonary tree, then put an emphases on those geometrically similar points and maintain their geometric continuity during simplification from some strategies. Experiments showed that the new algorithm could overcome those shortcomings and besides produced better integrality and consistency comparing with traditional algorithms.

Key words: surface simplification; quadric error metrics; multi-attributes; progressive mesh

1 背景介绍

随着各种高级模型设计工具的不断涌现和各种网络娱乐技术的日益成熟,使得人们对三维模型的精度和细节都提出了更高的要求,这也使得三维模型的复杂程度急剧增长。庞大的几何数据量对硬件设备的处理能力和速度提出了巨大挑战,同时也驱使相关研究人员对模型的压缩简化、存储显示过程提出了许多措施。Hoppe 在 1996 年首先提出递进网格(Progressive Mesh)^[1]的思想,从而将任意拓扑网格表示为一种无损、连续、可逆的编码方式。这种编码方式具有便于储存、显示和传输等优点。之后国内外很多学者对递进网格进行了研究^[2-12]。其中 Garland 利用二次方拟合的方法来控制网格简化的过程,其效果很好,并且效率和速度都很高^[4,8]。文中通过试验二次方拟合的方法

在实际模型简化中的效果,发现其在应用中存在一定的不足;主要体现在拟合的新生成点的几何坐标可能存在较大的视觉误差,这个问题将在第三节进行详细介绍。接下来一节将介绍原算法的基本思想。

2 利用二次方拟合的多属性模型压缩算法

2.1 多维空间中的二次方误差矩阵

压缩算法的基本思想就是边的“合并”操作。如图 1 中,左边是一个三角形网格,其中的边 $\langle v_i, v_j \rangle$ 通过“合并”之后就变成一个点 v_k 。具体来说就是,将一条边的两个顶点用一个新位置的顶点代替,新的节点将前两个顶点的所有邻居作为自己的邻居,建立新的拓扑;在此过程中面 (v_i, v_1, v_2) 和面 (v_j, v_1, v_2) 消失,从而实现了简化。为了从模型所有的边中挑选合适的边进行“合并”操作,需要一种优选机制。二次方拟合的方法就是通过边的二次方误差进行优选。下面介绍二次方误差矩阵的由来。

首先将点的其他属性与几何坐标一起组成一个 n 维向量 $v \in R^n$ 。设 R^n 中基为 e_1, e_2, \dots, e_n , 任取 R^n 中一

收稿日期:2011-08-19;修回日期:2011-11-25

基金项目:国家自然科学基金资助项目(60303023)

作者简介:冯 钦(1987-),男,硕士研究生,主要研究方向为模型压缩;倪桂强,博士,教授,主要研究方向为分布式计算、网络管理、软件工程等。

个三角形如图 2 所示。其中 p, q, r 为面 E_1 的三个顶点。假设确定 R^n 空间中的基时,以 p 点为原点, e_1, e_2 为 E_1 内的两个基,其他的基 e_3, \dots, e_n 与这两个基正交,并且分布在 R^n 的其他维中。接下来,讨论一下任意点 v 到 E_1 距离的计算方法。因为以 p 点为原点,第一步计算 $u = v - p$,即点 v 相对 p 点的向量。

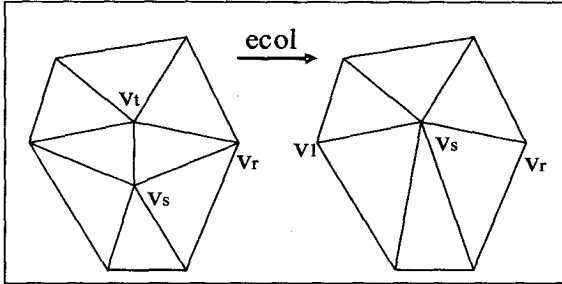


图 1 模型简化的基本操作

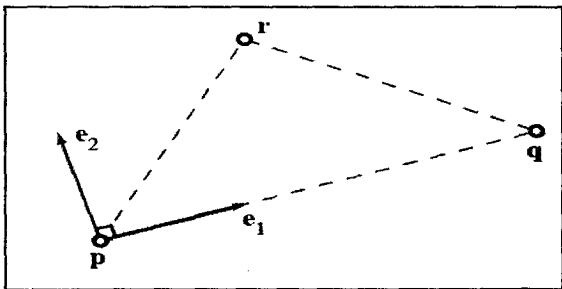


图 2 R^n 中面 E_1 的表示

计算 u 的模的平方:

$$\|u\|^2 = (u \cdot e_1)^2 + \dots + (u \cdot e_n)^2 \quad (1)$$

变换得:

$$\|u\|^2 - (u \cdot e_1)^2 - (u \cdot e_2)^2 = (u \cdot e_3)^2 + \dots + (u \cdot e_n)^2 \quad (2)$$

右边部分即为点 v 到 E_1 的距离。进一步化简:

$$\begin{aligned} D^2 &= \|u\|^2 - (u^T \cdot e_1)^2 - (u^T \cdot e_2)^2 \\ &= u^T \cdot u - (u^T \cdot e_1)(e_1^T \cdot u) - (u^T \cdot e_2)(e_2^T \cdot u) \\ &= v^T \cdot v - 2p^T \cdot v + p \cdot p - v^T \cdot (e_1 \cdot e_1^T) \cdot v \\ &\quad + 2(p \cdot e_1) \cdot e_1^T \cdot v - (p \cdot e_1)^2 - v^T \cdot (e_2 \cdot e_2^T) \cdot v \\ &\quad + 2(p \cdot e_2) \cdot e_2^T \cdot v - (p \cdot e_2)^2 \end{aligned} \quad (3)$$

抽象

$$D^2 = v^T A v + 2b^T v + c \quad (4)$$

其中:

$$A = I - e_1 \cdot e_1^T - e_2 \cdot e_2^T,$$

$$b = (p \cdot e_1) \cdot e_1 + (p \cdot e_2) \cdot e_2 - p,$$

$$c = p \cdot p - (p \cdot e_1)^2 - (p \cdot e_2)^2 \quad (5)$$

A 为 $n \times n$ 矩阵, b 为 n 维向量, c 为常数。假设 $v = (v_1, v_2, \dots, v_n)^T$, 那么定义 $V = (v_1, v_2, \dots, v_n, 1)^T$, 再定义

$$Q(V) = V^T \cdot Q \cdot V = D^2 \quad (6)$$

为二次方误差。其中 Q 即为多维空间中的二次方误差矩阵:

$$Q = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix} \quad (7)$$

每个面都有一个这样的“ Q 矩阵”,如果只包含几何信息,则 Q 为四维;如果包含几何和纹理信息,则 Q 为 6 维,以此类推。

2.2 基于二次方拟合的多属性模型压缩算法

利用二次方拟合的多属性模型压缩算法的基本思想如下:

1) 模型中每个点定义一个误差矩阵,这个误差矩阵等于其周围邻居面的“ Q 矩阵”之和。

2) 计算模型中所有边的“合并”误差以及“合并”的最优点。方法大致如下:

对于一条边 $e(v_1, v_2)$ 为 e 的两端点,假设 v_1, v_2 “合并”之后得到 v_3 。点 v_3 的误差矩阵定义为 $Q_3 = Q_1 + Q_2$ 。首先尝试通过最优化误差矩阵 Q_3 的方法计算出 v_3 的坐标。因为

$$Q_3(V_3) = V_3^T Q_3 V_3 \quad (8)$$

最小化 $Q_3(V_3)$, 即解方程 $\partial Q_3 / \partial v_1 = \partial Q_3 / \partial v_2 = \partial Q_3 / \partial v_3 = 0$ 如果有最优解,则使用最优解,否则选取 v_1, v_2 和 $(v_1, v_2)/2$ 三者中计算 $Q_3(V_3)$ 误差最小的点作为“合并”后的点,边“合并”的误差计算方法也是类似的。将整个模型所有边的误差排序。

3) 每次都“合并”都挑选误差最小的边。

4) 更新相关拓扑结构;更新相关边的误差;记录“合并”过程信息。

5) 如果模型达到要求则结束,否则重复 3)。

3 基于二次方拟合的多属性模型压缩算法应用在实际模型上的问题

首先简单说明一下实际中模型的一些特殊性。实际中每个点基本上都有多个属性,如几何、纹理、颜色、法向、材质等;一般地,假设一个几何空间中的点 v_1 有两个的其他属性(如两个的纹理坐标,这种情况在实际中比较常见,作者在 $.x$, $.max$, $.3ds$ 等格式的模型中任意抽取实例测试均有类似情况,具体地见第 5 节实验结果),那么在实际编号中其会被标号成两个点 v_1' 和 v_1'' ,这两个点具有相同的几何坐标,但是不同的其他属性;而且原来的拓扑结构也会改变,假设原来面 e_1, e_2 包含 v_1 ,那么可能变成 e_1 包含 v_1' , e_1 包含 v_1'' 。下面举例原算法有可能出现问题的情景:

1) 假设原图 1 情景中的 v_1 和 v_s 两点在图 3a 情景中各具有两个不同的纹理坐标,那么分别形成两对新的点: v_{11} 和 v_{12} , v_{s1} 和 v_{s2} 。当前一种可能的拓扑结构为:

边连接关系可能为:点 v_4, v_1 与 v_{11} 相连,点 v_1, v_2 与

v_{s1} 相连,点 v_5, v_r 与 v_{t2} 相连,点 v_3, v_r 与 v_{t2} 相连,而边 $< v_{t1}, v_{s1} >$ 与边 $< v_{t2}, v_{s2} >$ 重合,其他拓扑信息不变。

面索引关系可能为:面 $e_1(v_1, v_4, v_{t1})$ 、面 $e_2(v_1, v_{s1}, v_{t1})$ 、面 $e_3(v_1, v_{s1}, v_{t2})$ 、面 $e_4(v_2, v_{s1}, v_{t3})$ 、面 $e_5(v_3, v_{t2}, v_r)$ 、面 $e_6(v_{t2}, v_{s2}, v_r)$ 、面 $e_7(v_5, v_{t2}, v_r)$ 、面 $e_8(v_5, v_{t2}, v_4)$,其他拓扑信息不变。

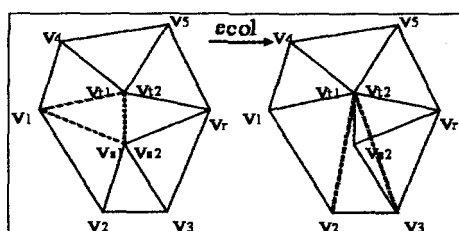
假设将要“合并”边 $< v_{t1}, v_{s1} >$ 到点 v_{t1} ,涉及到的面有 e_2, e_3, e_4 。那么就会形成图 3a 右边的情况,“合并”之后面之间会重叠。

2) 假设原图 1 情景中的 v_t, v_s 和 v_l 三点在图 3b 情景中各具有两个不同的纹理坐标,那么分别形成三对新点: v_{t1} 和 v_{t2}, v_{s1} 和 v_{s2}, v_{l1} 和 v_{l2} 。当前一种可能的拓扑结构为:

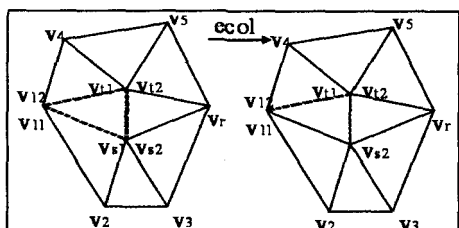
边连接关系可能为:点 v_4, v_{t2}, v_{l1} 与 v_{t1} 相连,点 v_{t1} 与 v_{s1} 相连,点 v_{t2}, v_{s2}, v_3, v_r 与 v_{t2} 相连,点 v_5, v_r 与 v_{t2} 相连,而边 $< v_{t1}, v_{s1} >$ 与边 $< v_{t2}, v_{s2} >$ 重合;

面索引关系可能为:面 $e_1(v_{t2}, v_4, v_{t1})$ 、面 $e_2(v_{t1}, v_{s1}, v_{t1})$ 、面 $e_3(v_{t2}, v_{s2}, v_{t2})$ 、面 $e_4(v_2, v_{s2}, v_{t3})$ 、面 $e_5(v_3, v_{t2}, v_r)$ 、面 $e_6(v_{t2}, v_{s2}, v_r)$ 、面 $e_7(v_5, v_{t2}, v_r)$ 、面 $e_8(v_5, v_{t2}, v_4)$,其他拓扑信息不变。

假设将要“合并”边 $< v_{t1}, v_{s1} >$ 到点 v_{t1} ,涉及到的面只有 e_2 。那么就会形成图 3b 右边的情况,原本的面 e_2 变成一个“空洞”。



(a) 简化中出现面重叠的情况



(b) 简化中出现“空洞”的情况

图 3 原算法有可能出现问题的情景

上述问题出现的原因在于处理那些几何坐标相似但是其他属性不同的点(以下称为几何相似点(Geom-similarers))时所采取的措施;通过将它们重新编号,设置成几何坐标相同,但其他属性不同的“新点”,最后破坏了原始模型的拓扑结构。新设置的点都只具有“原始点”的部分拓扑信息,这使得原算法在构建点的误差矩阵时不准确,而使用这些不准确的误差矩阵拟合出来的点在几何和拓扑上都有可能偏离原算法的设想。

4 一种改进的利用二次方拟合的多属性模型压缩算法

改进措施主要针对几何相似点。这些点在几何空间中是重合的,但是在联合纹理等其他属性形成的高维空间中却是不同的点;如果只是简单地将它们融合,那么纹理、法向等信息会被破坏。经过试验,改进算法主要思想如下:

1) 预处理:使用八叉树结构对原模型进行几何空间细分;具体来说就是按照几何坐标,第一次先将模型中的点划分到 8 个子空间中,第二次分别将各个子空间继续划分为 8 个子空间,以此类推,最后模型中距离相近的点分散在各个细分空间中。根据设定的距离阈值(Dis_geom_sim,两个点之间的距离小于这个阈值就判定这两个点为几何相似点)进一步判定,并将几何相似点集中到一个个集合中。对每个集合中点的几何坐标取平均值,并将集合中的点的几何坐标都设置成这个平均值。

2) 初始化:在构建每个点的误差矩阵时,不单构建一个包含点各个属性的高维矩阵(Quadric_All),还构建一个只包含几何属性的四维误差矩阵(Quadric_Geom)。Quadric_All 等于这个点所有邻居面的 Quadric_All 之和。Quadric_Geom 等于这个点所有邻居面以及其几何相似点所有邻居面的 Quadric_Geom 之和。

3) 计算模型中所有边的“合并”误差,并按从小到大排序。

4) 简化:每次挑选最小误差的边。在边“合并”时,如果两个端点都是非几何相似点,完全采用 Quadric_All 拟合最终点;如果其中一个端点是几何相似点,那么先采用 Quadric_All 拟合,再采用 Quadric_Geom 拟合,最终点的几何坐标采用 Quadric_Geom 拟合的坐标,其他属性坐标采用 Quadric_All 拟合的坐标,并且将同一个集合中的其他几何相似点的几何坐标更新成最终拟合点的几何坐标。

5) 更新:更新相关拓扑结构;更新相关边的误差;更新同一个集合中的其他几何相似点的 Quadric_Geom 矩阵(用新生成点的 Quadric_Geom 矩阵替换这些点的矩阵),记录“合并”过程信息。

6) 如果模型达到要求则结束,否则重复 4)。

5 实验效果对比

实验采用 Intel(R)Core(TM)2 Duo CPU E6550、主频 2.33GHz、内存 1.96G、显卡 ATI Radeon HD 2400 的 DELL 型号主机,编程环境为 VC++6.0 和 OpenGL 图形接口,系统为 Windows XP。实验中的模型为 3ds 格式。对比实验结果如图 4 和图 5 所示。两个原始模型

如图 6 所示。通过实验结果可以看到改进算法:

1) 消除了“空洞”和面重叠情况,如图 4 所示。

2) 适当调整 Dis_geom_sim, 可以连接模型中的不同模块,保持良好的整体性,如图 5 所示。

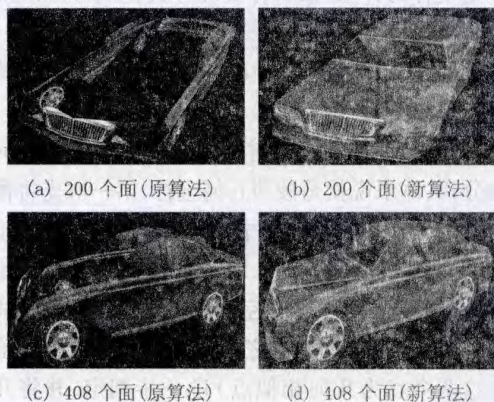


图 4 在汽车模型上对比实验

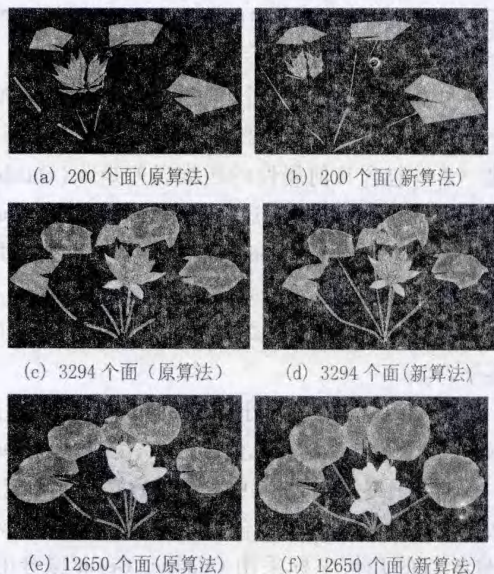


图 5 在荷花模型上对比实验

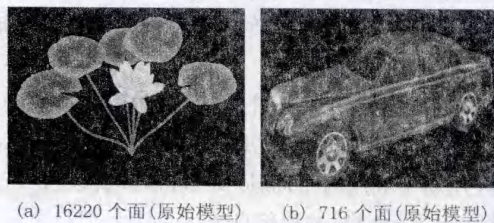


图 6 两个原始模型

值得一提的是在实际的情况中,往往如下参数的调整对简化效果起到一定的作用:

(1) 加大不同纹理之间边“合并”的误差作为惩罚;

(2) 加大不同模块之间边“合并”的误差作为惩罚;

(3) 在改进算法的第 3) 步中,使用两个矩阵的最优误差的加权和作为最终误差;

(4) 边“合并”时,加大端点的邻居面或者邻居边数超过一定阈值时的误差作为惩罚;

(5) 边“合并”时,加大端点的相似点数超过一定阈值时的误差作为惩罚。

6 结束语

文中根据实际模型的特点提出一种改进的基于二次方拟合的多属性模型压缩算法。通过实验看出,该算法解决了原算法在实际模型上出现的问题,实际简化效果有很大提高。

在此特感谢 Michael Garland, 实验代码中关于高维矩阵计算部分引自 Michael Garland 的程序;感谢模型数据的提供者 <http://www.3dcool.net>, 实验中模型数据均来自这个网站。

参考文献:

- [1] Hoppe H. Progressive Meshes [C]//SIGGRAPH 96 Proc. [s. l.]:[s. n.], 1996:99-108.
- [2] 谷冬冬,潘正运. 渐进网格简化模型的改进算法[J]. 计算机工程与设计, 2008, 29(18): 4648-4650.
- [3] 罗大庸,冯良波. 3D 多层次模型简化算法的研究[J]. 计算机技术与发展, 2010, 20(4): 91-100.
- [4] Garland M, Heckbert P S. Simplifying Surfaces with Color and Texture Using Quadric Error Metrics [C]//Proceedings of IEEE Visualization 98, [s. l.]:[s. n.], 1998:263-269.
- [5] Sander P, Snyder J, Gortler S, et al. Texture mapping progressive meshes [C]//ACM SIGGRAPH 2001 Proceedings. [s. l.]:[s. n.], 2001:409-416.
- [6] Prost R, Valette S. Wavelet-Based Progressive Compression Scheme for Triangle Meshes: Wavemesh [J]. IEEE Transactions on Visualization and Computer Graphics, 2004, 10(2): 123-129.
- [7] 陶志良,潘志庚,石教英. 基于能量评估的网格简化算法及其应用[J]. 软件学报, 1997, 8(12): 881-888.
- [8] Garland M, Heckbert P S. Surface Simplification Using Quadric Error Metrics [J]. Computer Graphics, 1997, 31(3): 209-216.
- [9] 周 昆,潘志庚,石教英. 基于三角形折叠的网格简化算法 [J]. 计算机学报, 1998, 21(6): 506-513.
- [10] 韩 猛,钟广军. 一种适用于网络环境的三维网格压缩算法[J]. 计算机技术与发展, 2007, 17(12): 8-11.
- [11] 顾耀林,朱晓燕. 基于渐进网格的复杂模型简化算法[J]. 计算机工程与应用, 2007, 43(2): 78-80.
- [12] 蔡 苏,赵沁平. 三维网格压缩方法综述[J]. 计算机科学, 2006, 33(5): 1-4.