

多属性递进网格快速生成算法研究

郝齐辉, 谭同德, 李润知, 赵红领, 王 静

(郑州大学 信息工程学院, 河南 郑州 450001)

摘 要: 递进网格表示法能高效地存储网格信息,并能生成连续的细节层次模型,支持模型的多分辨率表示、递进传输、网格压缩和有选择精化。文中提出并实现了一种基于二次误差测度的多属性递进网格生成算法。同 Hoppe 提出的基于能量优化的算法和以往大多支持仅具有几何属性的算法相比,该算法具有生成速度快、质量高、可处理包含色彩、纹理等多属性递进网格模型的特点。

关键词: 二次误差测度;递进网格;边折叠;细节层次

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2006)11-0044-03

Research on High-Speed Algorithm to Generate Progressive Mesh with Multi-Attributes

HAO Qi-hui, TAN Tong-de, LI Run-zhi, ZHAO Hong-ling, WANG Jing

(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: Progressive meshes can store mesh information efficiently and generate continuous level of detail models. They support multiresolution representation, progressive transmission and selective refinement. Proposes and implements an algorithm to generate progressive mesh with multi-attributes based on quadric error metric. Compared with the algorithm proposed by Hoppe which is based on energy optimization, and those algorithms that only supports geometry attribute, this algorithm is characterised by high generation rate, high quality and the ability of processing progressive mesh with multi-attributes.

Key words: quadric error metric; progressive mesh; edge collapse; level of detail

0 引言

在计算机图形学中,经常采用多边形网格(通常是三角形网格)描述物体。由表面重构技术生成的多边形网格中的多边形数目非常多。在很多情况下,并不需要对物体的细节刻画得很详细的模型。例如,当观察者离物体较远或物体在运动时,只看到物体的大概轮廓,用三角形数目较少的粗糙网格表示即可;只有当观察者离物体较近或物体静止时,才需要一个比较精细的网格模型。

传统的网格的表示方法有几个缺点:首先,数据量庞大,不易于存储、传输和绘制,尤其在交互式可视化和虚拟现实,要求图形生成具有“实时性”,过于庞大的网格显然是不适合的;其次,这种表示法不支持递进传输和有选择精化;再次,用这种表示法不易于建立光滑的多细节层次的近似模型。

为了解决上述问题,H. Hoppe 提出了基于能量优化的PM生成方法^[1]。该算法的基本思想是:以边折叠作为网

格简化的基本操作,最初采用全局能量最小化方法来计算折叠所生成的新点坐标,并记录下该折叠操作,最后形成一组网格简化信息的序列。然后通过一系列顶点分裂操作就可以动态地恢复成原始网格。这种方法的主要缺点是:由于要建立和求解复杂的全局能量优化方程,因而计算复杂性高,很难达到实时。此外,目前国内外许多科研人员对递进网格模型进行了研究,提出了一些相关算法^[2~5]。但这些算法大多只支持具有几何信息的网格模型,而对于包含颜色、纹理等其它信息的网格模型,尚没有较好的算法。

基于上述讨论,文中笔者在深入研究了前人的工作以后,提出并实现了一种递进网格模型的生成算法。该算法仍使用基于边折叠的网格简化方法,但在选择折叠操作所生成的新点坐标时,不是用能量优化方法,而是使用扩展二次误差测度方法^[6]。同大多数以往算法相比,文中的算法不仅更快速,而且可处理包含色彩、纹理等多属性的递进网格模型。

1 算法描述

1.1 递进网格的生成

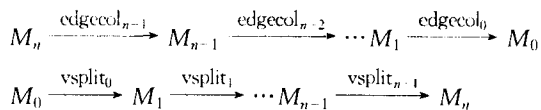
假设原网格为 M_n ,每次从 M_n 中找出一条边进行折

收稿日期:2006-03-06

基金项目:河南省科技攻关项目(0324210045)

作者简介:郝齐辉(1980-),女,河南人,硕士研究生,研究方向为虚拟现实;谭同德,教授,博士,研究方向为计算机辅助设计、图形图像处理、虚拟现实。

叠(edge collapse),经过 n 次折叠后,得到一个相对粗糙的简化网格 M_0 ,其中 m_0 为网格 M_0 的顶点数目。边折叠的逆过程是点分裂(vertex split,如图 1 所示),在 M_0 的基础上逐步地把点分裂的信息添加上去,就能恢复原网格 M_n 。



由于边折叠操作是一个可逆过程,因此可以把它的对偶操作——顶点分裂表示成五元组 $\text{vsplit}(s, l, r, cv_s, cv_{m_{0+i}})$,其中 cv_s 和 $cv_{m_{0+i}}$ 分别是 M_i 的后续网格 M_{i+1} 的 s 点和新顶点 m_{0+i} 的信息。这样, $(M_0, \{\text{vsplit}_0, \text{vsplit}_1, \dots, \text{vsplit}_{n-1}\})$ 就构成了递进网格。

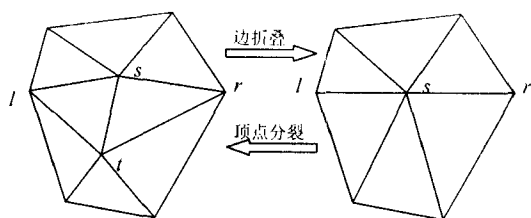


图 1 边折叠与顶点分裂操作

1.2 边折叠操作

边折叠操作需要解决两个关键问题:一是如何选择网格中的一条边进行折叠;二是如何确定边折叠后新顶点的位置。解决上述两个问题的原则是:折叠的边及其折叠后新顶点位置的选取应使简化后的网格与原网格的误差尽可能小。

笔者采用扩展二次误差测度作为文中算法的误差度量标准,来解决以上问题。扩展二次误差测度建立在原始二次误差测度的基础之上,也是采用点到一组平面的距离平方和作为误差测度。它继承了原始二次误差测度具有较高的计算速度、较小的内存消耗、得到的简化网格具有较高质量的优点。同时,它又把二次误差测度从三维欧氏空间推广到多维欧氏空间,可以使任意多维属性的处理在一个统一的框架下进行。

1.2.1 扩展二次误差测度的公式推导

扩展二次误差测度把顶点信息的表示由一个三维的几何向量,扩展为包含几何和其它属性信息的多维向量;把基于点到一组平面距离平方和的二次误差测度从三维欧氏空间推广到多维欧氏空间。下面以能够处理具有色彩、纹理属性的扩展二次误差测度为例,加以说明公式推导过程:

由于顶点的空间位置坐标值的范围和色彩、纹理分量值的范围不同,为了能够使这 3 种属性对简化误差的评估起到等价的作用,在网格数据初始化时,首先要将网格模型的几何空间坐标和颜色、纹理分量值归一化到 $(0,1)$ 之间。另外,算法中处理的是顶点的颜色,而实际图形文件中的数据多是三角面片的颜色。实现时对多颜色的顶点,将该顶点在各个面的颜色取平均值后,再赋给该顶点,实

践证明这种方法在处理模型颜色均匀变化的部分时效果尤其不错。

现在所用的顶点 $v = [xyzrgbtxty]^T$ (xyz 为几何坐标, rgb 为色彩值, $txty$ 为纹理坐标) 是一个八元组,是多元空间中的一个点,对于多元空间中的点到其平面的距离,做如下考虑:在多元空间 R^n (在色彩模型的情况下, $n=8$) 中有一个三角形 $T = (p, q, r)$, 设 $p = [p_x p_y p_z p_r p_g p_b p_{tx} p_{ty}]^T$, $q = [q_x q_y q_z q_r q_g q_b q_{tx} q_{ty}]^T$, $r = [r_x r_y r_z r_r r_g r_b r_{tx} r_{ty}]^T$ 。这样,三角形 $T = (p, q, r)$ 所在的平面,可以用两个相互垂直的向量和一个点来表示,如图 2 所示。

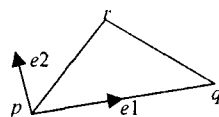


图 2 三角形 T 的表现形式

$$e_1 = (q - p) / \|q - p\|$$

$$e_2 = ((r - p) - e_1 \cdot (r - p)e_1) / \|(r - p) - e_1 \cdot (r - p)e_1\|$$

理论上,在多元空间 R^n 中,可以分别求出 $e_1, e_2, e_3, \dots, e_n$, 它们是 n 元空间中相互垂直的单位矢量。

现在求 R^n 空间中任意一点 v 到平面 T 的距离。设 $u = p - v$, 则有:

$$\|u\|^2 = u^T u = (u^T e_1)^2 + (u^T e_2)^2 + (u^T e_3)^2 + \dots + (u^T e_n)^2$$

上式所表示的实际上是勾股定理的扩展形式。

把上式重写成如下形式:

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

可见式子右边便是 v 到平面 T 的距离平方。于是有:

$$D^2 = \|u\|^2 - (u^T e_1)^2 - (u^T e_2)^2 = u^T u - (u^T e_1)(e_1^T u) - (u^T e_2)(e_2^T u)$$

可以定义三元组 $Q = (a, b, c)$ 来表示 D^2

$$D^2 = Q(v) = v^T a v + 2b^T v + c$$

求出对应的 a, b, c , 以便求“边折叠”后新顶点的位置矢量, 则 a, b, c 的值如下:

$$a = I - e_1 e_1^T - e_2 e_2^T$$

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

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

在这里, a 是一个 $n \times n$ 的矩阵, b 是一个 n 维向量, c 是一常数。

Q 称作二次误差测度或二次误差矩阵, $Q(v)$ 称作二次误差值。

将各个顶点的相邻面的误差矩阵相加, 就得到该顶点的误差矩阵。

1.2.2 折叠边及折叠后新顶点位置的选取

网格中每条边的误差矩阵为边的两个端点的误差矩

阵之和: $Q = Q_1 + Q_2$, 每次应选择当前误差值 $Q(v)$ 最小的边进行折叠。

要使折叠后的误差值最小, 则须对误差代价 $Q(v) = v^T Q v$ 求偏导, 令偏导值为 0, 解出最佳折叠后新顶点位置向量 $v = -a^{-1}b$, 相应的最小值为 $Q(v) = -b^T a^{-1}b + c$ 。有时 a 无法求逆矩阵, 也可在两个端点中选择一个代价较小的作为新顶点。这样的好处是程序更简单、速度更快、内存占用更小, 但简化的效果上要差一点。

1.3 算法梗概

根据上面的讨论, 文中提出的递进网格的生成方法可用下面的算法描述:

Step1 读入初始网格 $M = M_n$ 。

Step2 对 M 中的所有边, 计算折叠误差, 并根据误差的大小进行排序。

Step3 取折叠误差最小的边进行折叠操作, 记录折叠信息, 更新与之相关的所有信息。

Step4 重复 Step3, 直至当前的简化网格满足要求。

Step5 输出简化网格 M_0 和边折叠记录序列(即递进网格)。

2 实验结果

文中所介绍的算法已经在 P4 2.4G, 256M 内存, NVIDIA Geforce2 MX 400 64M 显卡的联想品牌机上用 VC++ 6.0 和 DirectX8.0 实现, 系统为 Microsoft Windows XP。

图 3 是一组实例, 执行结果验证了算法的有效性。从图中可以看出, 不仅网格模型的几何信息得到了较好的保留, 其色彩信息也得到了较好的保留。

3 结束语

文中提出并实现了一种新的递进网格简化处理方式。新的方法采用扩展二次误差测度表示简化网格与初始网

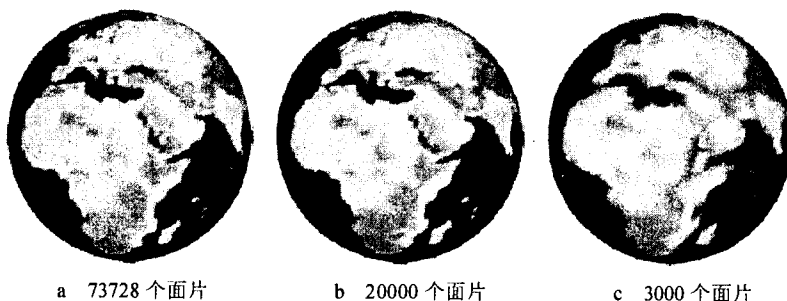


图 3 多属性递进网格模型简化实例

格的几何特征和其它属性信息的匹配程度, 利用二次误差值最小原理, 指导网格简化操作的进行。同 Hoppe 提出的基于能量优化的算法和以往大多只能完成几何特征简化的算法相比, 该算法不仅效率高, 而且可以保证简化模型同初始模型在几何特征和色彩、纹理等属性信息上尽可能相似。

在以后的研究中, 将考虑将该算法与视点和绘制算法结合起来, 形成与视点相关的递进网格, 从而支持实时图形绘制。

参考文献:

- [1] Hoppe H. Progressive meshes[J]. Computer Graphics, 1996, 30(1):99-108.
- [2] Taubin G, Gueziec A, Horn W, et al. Progressive forest split compression[C]//SIGGRAPH'98 Proceedings. [s. l.]: [s. n.], 1998:123-132.
- [3] 陶志良, 潘志庚, 石教英. 支持快速恢复的可逆递进网格及其生成方法[J]. 软件学报, 1999, 10(5):503-507.
- [4] 卢章平, 赵 泉. 基于“边折叠”的累进网格生成算法的研究[J]. 工程图学学报, 2004, 24(1):37-41.
- [5] 刘焕敏, 杨克俭, 王玉华. 一种面积加权的半边折叠网格简化算法及其递进网格构造[J]. 武汉理工大学学报, 2005, 29(2):76-78.
- [6] Garland M. Simplifying Surfaces with color and Texture using Quadric Error Metrics[C]//Proceedings of IEEE Visualization'98. [s. l.]: [s. n.], 1998:263-269.

(上接第 43 页)

Agent 系统的复杂性及其交互的动态性, 需要一种形式化的方法对交互协议进行建模分析和验证。文中提出一种基于着色 Petri 网的 MAS 交互进行建模研究方法, 通过 FIPA 交互协议建模过程, 可以看出 Agent 可利用这种方法来分析是否可理解接收到的协议、是否交互可被 Agent 的现有状态接收、是否与 Agent 自利的目标冲突等。

参考文献:

- [1] Nowostawski M, Purvis M, Cranefield S. A Layered Approach for Modelling Agent Conversations[C]//In: Proceedings of the 2nd International Workshop on Infrastructure for Agents,

MAS, and Scalable MAS. Montreal, Canada; [s. n.], 2001:163-170.

- [2] Cost R. Modelling Agent Conversations with Coloured Petri Nets[C]//Working Notes of the Workshop on Specifying and Implementing Conversation Policies. Seattle, Washington; [s. n.], 1999:59-66.
- [3] Poutakidis D, Padgham L, Winikoff M. Debugging Multi-Agent Systems Using Design Artefacts: The Case of Interaction Protocols[C]//In: Proceedings of the 1st International Joint-Conference on Autonomous Agents and Multi Agent Systems. Bologna, Italy; [s. n.], 2002:960-967.
- [4] 袁崇义. Petri 网原理[M]. 北京: 电子工业出版社, 2005.
- [5] 张维明. 信息系统建模[M]. 北京: 电子工业出版社, 2002.