

递进网格的一种快速生成算法

成迟蕙 潘志庚 石教英

(浙江大学 CAD&CG 国家重点实验室, 杭州 310027)

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。此文给出了一种递进网格的生成算法,该算法能有效地控制简化网格与原始网格的误差,速度较快并能很好地保持原模型的边界特征。

关键词 虚拟现实, 网格简化, 递进网格, 细节层次模型

1 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,我们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,我们只看到物体的大概轮廓,用一三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,我们才需要一个比较精细的网格模型^[1]。

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中,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,我们希望有一种网格的表示方法,当观察者离物体较远或物体在快速运动时,从原始数据中抽取关键信息,构成一粗糙网格表示该物体;当观察者离物体越来越远或物体运动越来越慢时,将该物体更细节的信息逐渐的、连续的添加到原粗糙网格上,直至恢复物体的全部信息。这就是递进网格(Progressive Mesh, 简称 PM)^[2]。在 PM 中,任一拓扑形状的网络 $M = M^n$, 由一个相对粗糙

的网格 M^0 和一细节记录的序列构成,将序列中记录的细节信息逐次添加到 M^0 上,得到 M^1, M^2, \dots, M^n , 最终能完全恢复原网格 $M = M^n$ 。

PM 表示最早由 Microsoft 的 H. Hoppe 提出,他还给出了基于能量优化的 PM 生成方法^[2,3]。该算法的基本思想是:以边折叠作为网格简化的基本操作,为控制递进网格的质量,从初始网格 M 表面采样点集 X 以记录初始网格的信息,根据 X 与简化网格之间的距离计算能量函数 E ,该能量函数表示了简化网格与初始网格的匹配程度,利用能量最小原理,指导网格简化操作的进行。这种方法的主要缺点是:算法实现复杂;计算量大,速度慢。

基于上述讨论,我们提出了一种快速生成递进网格的算法。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,而是使用二次误差度量方法。为了控制细节记录的生成顺序,引入了二次误差的度量队列,它能有效地控制边折叠的次序,避免局部区域的过度简化和另一些区域的简化不够,大大增强了递进网格的质量。

2 算法描述

2.1 基本概念

三角形网格是由三维空间中的三角形相互通过

• 本项研究工作受国家自然科学基金面上项目、重点项目(No. 69633010)以及浙江大学曹光彪科技基金共同资助

收稿日期:1997-12-02;收到修改稿日期:1998-03-11

边和顶点连接而成的分段线性曲面,其中每条边最多包含在 2 个三角形中。我们可以用网格中顶点的位置和相互关系来表示一个网格。定义网格 $M = (K, V)$, 其中 K 包含了顶点的连接关系; $V = \{v_1, v_2, \dots, v_m\}$, $v_i \in R^3$ 表示网格 M 的顶点在三维空间中的位置。

对 M 中的任意边 $\{i, j\} \in K$, 如果该边只为一个三角形 $\{i, j, k\} \in K$ 所有, 则称该边为边界边, 该边的两个顶点称为边界顶点, 该边所在的三角形 $\{i, j, k\}$ 称为边界三角形。

对网格 M 中任一顶点 $\{i\} \in K$, 所有以 $\{i\}$ 为一个顶点的三角形 $\{i, j, k\} \in K (\{j\}, \{k\} \in K)$ 构成的集合, 称为与顶点 $\{i\}$ 相关的三角形集合, 记为 $\text{star}\{\{i\}, K\}$ 。对网格 M 中任一边 $\{i, j\} \in K$, 与点 $\{i\}$ 相关的三角形集合与与点 $\{j\}$ 相关的三角形集合的并集, 称为与边 $\{i, j\}$ 相关的三角形集合, 记为 $\text{star}\{\{i, j\}, K\}$ 。

2.2 递进网格的生成

假设原网格为 $M = M^n$, 每次从 M 中找出一条边进行折叠 (edge collapse), 经过 n 次折叠后, 得到一简化网格, 它有 m_0 个顶点, 记为 $M^0 = M'$ 。边折叠的逆过程是点分裂 (vertex split) (如图 1 所示), 在 M^0 的基础上逐步的把点分裂的信息添加上去, 就能恢复原网格 M^n 。

$$M = M^n \xrightarrow{\text{ecol}_{n-1}} M^{n-1} \xrightarrow{\text{ecol}_{n-2}} \dots \xrightarrow{\text{ecol}_0} M^0 = M'$$

$$M^0 = M' \xrightarrow{\text{vsplit}_0} M^1 \xrightarrow{\text{vsplit}_1} \dots \xrightarrow{\text{vsplit}_{n-1}} M = M^n$$

第 i 次边折叠时, 记下折叠边 $\{s, t\}$, 相关的 2 个顶点 $\{l_i\}, \{r_i\}$, 及该次边折叠前 $\{s_i\}, \{t_i\}$ 点的坐标, 它与点分裂所必须的信息是一致的, 记为

$$\text{vsplit}(s_i, l_i, r_i, A_i = (V_{si}^{i+1}, V_{m_0+i+1}^{i+1}))$$

(其中的 t_i 即第 $m_0 + i + 1$ 个顶点)

这样, $(M^0, \text{vsplit}_0, \text{vsplit}_1, \dots, \text{vsplit}_{n-1})$ 就构成了递进网格。

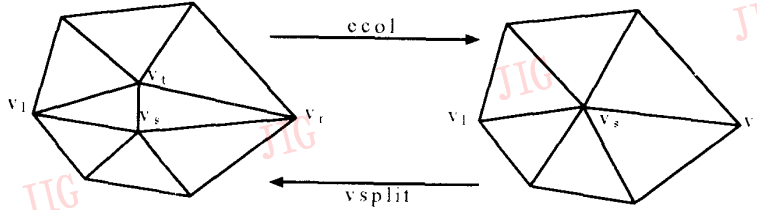


图 1 边折叠与点分裂的互逆过程

2.3 边折叠后新顶点的生成和误差的度量

简化模型 M^0, M^1, M^2, \dots 必须与原网格尽量相似, 这取决于边折叠的顺序和边折叠后生成的新点的位置。如何选择合适的边进行折叠及如何生成新的顶点, 有一个选择误差度量标准的问题。本文的算法以点到平面的距离为误差度量标准。

设对边 $\{s, t\} \in K$ 进行边折叠, 则与边 $\{s, t\}$ 相关的三角形集合 $\text{star}\{\{s, t\}, K\}$ 构成了原网格模型上的一个区域。设边折叠后生成的新点的位置 v_h 为 $(x_h, y_h, z_h, 1)^T$, 我们定义这次边折叠带来的误差 $\Delta(s, t)$ 为顶点到三角形集合 $\text{star}\{\{s, t\}, K\}$ 中每个三角形所在面的距离的平方和^[5]:

$$\Delta(s, t) = \sum_{p \in \text{star}\{\{s, t\}, K\}} (p^T V_h)^2 \quad (1)$$

其中 $p = (a, b, c, d)^T$ 表示三角形集合 $\text{star}\{\{s, t\}, K\}$ 中的每个三角形所在面的平面方程 $ax + by +$

$cz + d = 0$, 且有 $a^2 + b^2 + c^2 = 1$ 。式 (1) 可以变换成如下的形式:

$$\Delta(s, t) = \sum_{p \in \text{star}\{\{s, t\}, K\}} (V_h^T p)(p^T V_h) = \sum_{p \in \text{star}\{\{s, t\}, K\}} V_h^T (pp^T) V_h = V_h^T \left(\sum_{p \in \text{star}\{\{s, t\}, K\}} K_p \right) V_h \quad (2)$$

其中 K_p 是 4×4 的对称矩阵, 称为三角形的误差矩阵, 它的定义如下:

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (3)$$

将 $\text{star}\{\{s\}, K\}$ 中的所有三角形对应的误差矩阵求和, 就得到了关于顶点 s 的误差矩阵 Q_s 。在对边 $\{s, t\}$ 折叠时, 将顶点 $\{s\}$ 与顶点 $\{t\}$ 的误差矩阵 Q_s, Q_t 相加, 就近似得到了该次边折叠的误差矩阵 Q , 即可

求出该次边折叠的误差 $\Delta(s, t) = \mathbf{V}_k^T \mathbf{Q} \mathbf{V}_k \cdot \mathbf{V}_k$ 的位置可有多种选择,为了使简化模型与原网格尽量相似,我们取 \mathbf{V}_k 的位置使 $\Delta(s, t)$ 最小^[5]。

2.4 误差代价队列

对于每条可以进行边折叠的边,都可以按上述算法求出其误差代价 Δ , Δ 越小,表明对该边进行折叠的折叠效果越好。对于每一条可以进行边折叠的边,根据 Δ 值的大小,按从小到大的顺序将其链成一条候选链,即误差代价队列。每次边折叠时,都对误差代价队列列首的边进行折叠,修改折叠点的误差矩阵,并重新计算折叠以后的误差代价队列,为下次边折叠作准备。误差代价队列的引入,使连续分辨率的细节层次模型的建立成为可能,并能支持网格的递进传输。

2.5 算法总结

根据上面的讨论,本文提出的递进网格的生成方法可用下面的算法描述:

步骤 1: 读入初始网格 $M = M^0$ 。

步骤 2: 对 M 中可以进行边折叠的边,计算误

差代价 Δ , 并根据 Δ 值建立误差代价队列。

步骤 3: 取误差代价队列列首的边进行折叠,记录折叠信息,修改折叠点的误差矩阵,并重新计算相关的折叠边的误差代价,重排误差代价队列,转步骤 3,直至误差代价队列为空或获得用户满意的简化网格为止。

步骤 4: 输出简化网格 $M^0 = M'$ 和边折叠记录的序列。

3 算法实现的相关问题及实验结果

3.1 顶点的分类

对于一些拓扑结构比较复杂的网格,初始网格 $M = M^0$ 与递进网格 $(M^0, vsplit_0, vsplit_1, \dots, vsplit_{n-1})$ 之间并不是一一映射的关系,如图 2 所示。在图中,由于简化网格在点 t 可能有一个面上翘,因此相同的 M^0 和 $vsplit(s, l, r, A = (V_s, V_t))$, 有两种精细网格 M 与之对应。这种一对多的映射,给添加 $vsplit$ 记录带来了混乱。

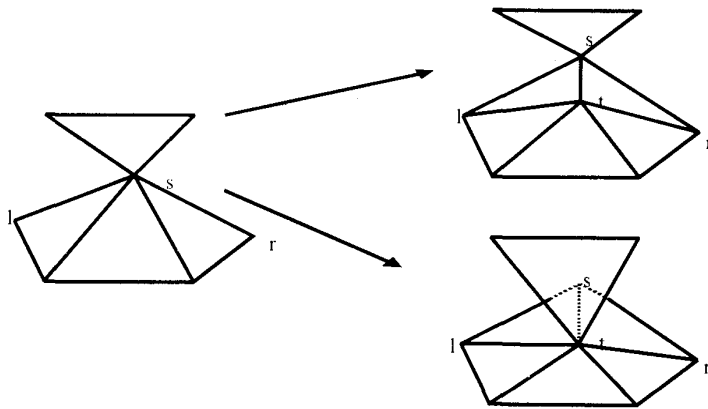


图 2 点分裂的情况可能不唯一

根据上面的分析可知,边折叠与点分裂并不是一一对应的。为了保证重构精细网格时的唯一性,我们把顶点分为以下类:边界顶点、非边界点和不规则点。对由不规则点组成的边,不允许进行边折叠;对边界顶点和非边界点组成的边,还要进行自相交的检查。

另外,为了解决这种二义性问题,我们提出了一种新的全可逆递进网格模型,支持特定细节层次模型的快速恢复。

3.2 自相交的检查

边折叠有可能改变网格的拓扑结构,使本来没有自相交的网格产生自相交现象(如图 3 所示,对边 $\{s, t\}$ 折叠后,共有 3 个面在边 $\{s, r_2\}$ 相交)。为了避免这种情况,我们采用以下两种方法加以控制。

(1) 在计算边折叠的误差代价之前,首先检查该次边折叠 $\{i, j\}$ 是否合法,合法性用 $legal(i, j)$ 函数定义,即必须满足以下 3 个约束条件。

- 对所有既与点 $\{i\}$ 相关又与点 $\{j\}$ 相关的点 $\{k\}$, ($\{i, k\} \in K, \{j, k\} \in K$), $\{i, j, k\}$ 必须是 K 的一个面 ($\{i, j, k\} \in K$)。

- 若 $\{i\}, \{j\}$ 均是边界顶点, 则 $\{i, j\}$ 必是边界边。

- 若 $\{i\}, \{j\}$ 均不是边界顶点, 则 K 中至少包含 4 个顶点; 若 $\{i\}, \{j\}$ 中至少有一个是边界顶点, 则 K 中至少包含 3 个顶点。

(2) 定义一二面角的阈值 β , 在已计算出边折叠新生成的点的坐标 V_k 后, 对 $\text{star}\{\{i, j\}, K\}$ 中所有的边检查其相关的二面角, 若二面角小于阈值 β , 则拒绝该边折叠, 不把它放入误差代价队列中。

以上两种方法, 计算量不大, 又能有效的控制网格的简化操作而不破坏网格的拓扑形状。

3.3 边界特征的保持

在计算顶点 $\{i\}$ 的误差矩阵时, 如果 $\{i\}$ 是边界顶点, 为保持原始网格的边界特征, 我们在与顶点 $\{i\}$ 相关的三角形集 $\text{star}\{\{i\}, K\}$ 中的每条边界边上作

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, 把这些平面的误差矩阵也包括在顶点 $\{i\}$ 的误差矩阵中。实践证明这种方法对保持原始网格的边界特征是非常有效的。

3.4 实验结果

该算法在 SUN SPARC 工作站上用 C 语言实现, 并对兔子、机枪、球冠等不同类型的几组网格数据进行了测试。实验结果表明, 用该算法生成的递进网格能完全保留原网格的信息, 又大大减小了存储量。其中, 简化网格的简化率一般能达到 90% 以上, 并能很好的保持初始网格的形状, 特别是边界特征, 把边折叠记录的序列逐次添加到简化网格上, 则能完全恢复初始网格。

图 4 是用我们的方法生成的递进网格图例(人头模型), 这里只给出了 3 个连续的细节层次模型(后一个模型与前一个模型相比, 是多了 600 次顶点分裂操作)。

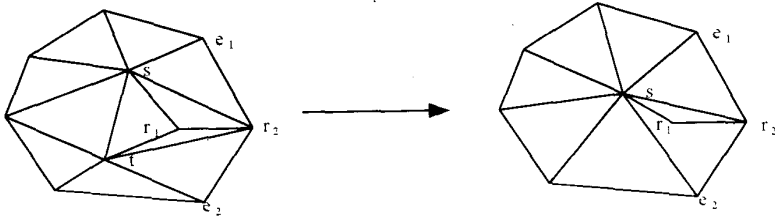
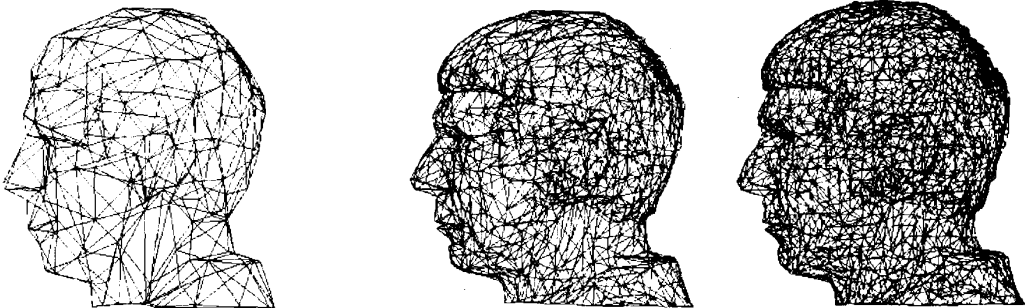


图 3 自相交现象



(a) 178 个顶点/340 个面

(b) 979 个顶点/1931 个面

(c) 1779 个顶点/3525 个面

图 4 递进网格图例

3.5 算法比较

为了与 H. Hoppe 所提出的递进网格生成算法进行比较, 我们也在机器上实现了该算法。表 1 给出

了我们的方法与 H. Hoppe 方法的时间对比。从表中可以看出, 用我们的方法生成递进网格的速度明显快于 H. Hoppe 方法(所有数据都是在 hp720 工作站上测得的)。

表1 两种算法的时间比较

	原模型 (三角形数)	生成递进网格的基 网格(三角形数)	Hoppe, H 算法所 用时间(秒)	本文算法所用 时间(秒)
人头(带头发)	3 525	340	3 593.5	10.8
人头(不带头发)	1 355	128	1 367.5	2.4
机枪	8 210	704	8 483.4	52.9
球冠	180	16	240.2	0.2
茶壶	3 751	289	3 596.6	11.9

参考文献

4 结束语

递进网格表示法能高效的存储网格信息,支持多细节层次场景的自动生成和光滑过渡,为虚拟现实和交互式可视化的研究提供了新的方法。

在研究递进网格及其生成算法时,除了实现了基于边折叠的递进网格生成方法外,我们还提出了基于顶点删除的累进网格构造方法。今后的研究主要是以下4个方面:(1)寻找更高效的、适用于二维流形和非流形物体的网格简化方法,并用它们来生成递进网格。(2)希望不仅能保持网格的几何形状,还能保持物体的外观属性,如颜色和纹理等。(3)将该算法与视点和绘制算法结合起来,形成与视点相关的累进网格,从而支持实时图形绘制。(4)能对递进网格形式进行编辑。

致谢:在本文所述的研究工作过程中,课题组的陶志良、周昆等同学给予了必要的帮助,在此表示感谢。

- 1 潘志庚,马小虎,石教英. 多细节层次模型自动生成技术综述. 中国图象图形学报,1998,3(9):754~759.
- 2 Hoppe H. Progressive Meshes. In SIGGRAPH'96,1996,49~108.
- 3 Hoppe H. T et al. DeRose, Mesh Optimization. In SIGGRAPH'93,1993:19~26.
- 4 陶志良,潘志庚,石教英. 基于能量优化的网格简化算法及其应用. 软件学报,1997,8(12).
- 5 Garland M, Heckbert P S. Surface simplification using quadric error metrics. In SIGGRAPH'97,1997.



成迟意 1976年生。浙江大学计算机系本科四年级学生。



潘志庚 1965年生。1990年毕业于南京大学计算机系,获硕士学位,1993年毕业于浙江大学计算机系,获博士学位。现为浙江大学CAD & CG国家重点实验室研究员,研究方向为分布式图形、虚拟现实和多媒体计算技术。



石教英 1937年生。教授,博士生导师。现任中国图象图形学会副理事长,研究方向为虚拟现实、多媒体和科学计算可视化。

A High-speed Generation Algorithm for Progressive Meshes

Cheng Chiyi, Pan Zhigeng, Shi Jiaoying

(State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027)

Abstract Progressive meshes can store mesh information efficiently and generate continuous level of detail (LOD) models. They support multiresolution representation, progressive transmission and selective refine-

ment. A high-speed algorithm for generating progressive meshes based on edge collapses is presented in this paper. This algorithm provides an efficient error management for the process of mesh simplification. The speed of the algorithm is very fast and the boundary features are well preserved.

Keywords Virtual reality, Mesh simplification, Progressive Meshes, Level of Detail Models

倾全力于亚洲市场艾尔莎分公司升级

在图形处理和数字通信两大领域以傲人科技领先的德商艾尔莎,出于亚太市场的广泛需求及巨大的市场发展潜力,日前将其亚太分公司转为独立子公司经营形态,并正式更名为艾尔莎国际科技股份有限公司(ELSA Asia Inc.)。新公司的总理由原分公司的张国龙先生担任。至此,艾尔莎国际科技股份有限公司将以全新的面貌出现,为亚洲客户提供更加尽善尽美的服务。

德商艾尔莎于 1997 年初进入亚太市场,开始阶段即以高档 3D 图形加速卡深受众多知名电脑厂商的青睐,宏基、康柏、戴尔、力捷,以及中国电脑第一厂商联想电脑相继在其产品中采用 ELSA 图形卡。在专业市场上,ELSA 图形卡更以其强大的性能优势受到专业人士的好评。这些优势包括,支持 3D Studio MAX、AutoCAD、Autodesk Mechanical Desktop、EDS/Unigraphics、I-DEAS、Microstation、Microcadam Helix、Pro-ENGINEE、Pro/JR、Softimage 3D、ALLPLAN、Hi-CAD、Open Inventor、SolidWorks、Sense8、Logocad、MSC/NASTRAN 以及其它许多流行软件。

自今年第二季度,为提高公司运作效率,增强公司自身的财务独立性和自主性,同时也为适应亚洲市场瞬息万变的

形势,开始实施“心有国际观,行有本地念”的服务理念,选择了子公司的经营形态。子公司的成立,除了能继续实现对客户的承诺,不断提供给客户高质量的产品外,还能为客户提供更快更好的支持服务,并与经销合作伙伴保持长期而良好的关系。

艾尔莎国际科技股份有限公司新任总经理张国龙先生告诉记者:“未来新公司的亚洲市场营销策略是,陆续引进高质量的全系列产品,满足亚太区各国市场对图形处理产品和解决方案的需求;同时,还将建立完善的技术支持体系,使客户充分感受到艾尔莎支持服务的便利。”

张先生还透露,除了已经推向亚太市场的、为图形处理专业人士量身订制的具备 OpenGL 超强加速性能的新产品 ELSA GLoria 2D/3D 系列图形卡外,还将通过其中国经销代理商中科多媒体公司推出针对多媒体及娱乐市场的 ELSA VICTORY Erazor 图形卡产品和 ELSAvision 视频会议系统。据了解,ELSA VICTORY Erazor 支持 Direct 3D 的卓越性能,更能为基于 Windows 95 操作系统的图形应用带来令人耳目一新的感受,同时为电脑提供视频输入/输出功能;ELSAvision 则能让用户实现天涯若毗邻的梦想。

欢迎订阅《通信学报》

《通信学报》是中国通信学会主办,反映我国通信科学水平,交流国内外通信科技新成果;促进学术进步和人才成长,探索新理论,新技术的学术性刊物。主要栏目有学术论文、技术报告、综述、短文、学术通信、研究简报、新技术展望等。

在无线电电子学、电信技术类中文核心期刊中,名列第三位。

邮发代号:2-676	主办单位:中国通信学会
国外代号:M395	出版:人民邮电出版社
定价:15.00 元/期;180.00 元/年	编辑:《通信学报》编辑部
电话:(010)67129323 67129324(传真)	地址:北京市崇文区夕照寺街 14 号
电子邮件:pptphxb@public.bta.net.cn	邮编:100061