

# 虚拟环境中多细节层次模型自动生成算法

潘志庚 马小虎 石教英

(浙江大学 CAD & CG 国家重点实验室 杭州 310027)

**摘要** 虚拟环境是限时计算和限时图形绘制技术的典型应用. 本文对用于限时图形绘制的多细节层次模型表示进行研究, 提出了基于三角形网格简化的多细节层次模型自动生成算法. 该算法多次遍历现有三角形网格模型的每一个顶点, 使用局部几何和拓扑特征移去满足简化标准的顶点, 对移去顶点后产生的多边形区域进行局部三角化. 多次执行上述过程, 结果形成不同细节层次的三角形网格模型. 文中给出的实例说明了该算法的有效性.

**关键词** 虚拟环境, 细节层次, 限时计算, 网格简化, 约束三角化.

虚拟环境是一种高度逼真地模拟人在自然环境中视、听、动等行为的人机界面技术.<sup>[1]</sup>它是一种对时间性要求很强的应用, 是一种典型的限时计算/限时图形绘制应用问题.<sup>[2]</sup>例如, 在 walk-through 类的虚拟环境应用中, 计算必须在指定时间内完成, 否则头盔显示器中显示的图象将会与实际不符, 从而导致出错. 限时计算与传统的计算概念不一样(传统的计算正确性与完成计算的时间无关), 它要求计算必须在规定的时间内完成, 否则认为计算结果是错误的.

实现限时图形绘制主要可采用如下方法<sup>[3]</sup>: (1) 采用不同细节层次 LoD(level of detail) 的场景模型; (2) 采用并行处理技术; (3) 使用不同复杂度的绘制算法. 要在限定的计算时间内绘制出复杂的场景, 可以采用多个 LoD 来描述同一场景, 并根据给定时间选择某个 LoD 和某种绘制算法进行并行绘制, 以满足限定的目标帧时间约束, 同时达到最佳的显示效果.

给每个物体提供多个不同 LoD 的描述是交互式图形应用中获得高帧速率的一种重要途径, 而且, 一个物体具有多个 LoD 表示有许多优点: 第 1, 如果物体仅覆盖屏幕的一个小的区域, 那么在绘制时通常不需要使用物体的精细模型, 使用一个粗的模型可以大大减少绘制图象的时间. 第 2, 当绘制图象时, 采用不同的 LoD 表示, 通常可避免子采样. 第 3, 物体的特征可以按物体的不同 LoD 表示来分类, 这种特征识别方法最近在图象处理和模式识别中也

• 本文研究得到国家自然科学基金和浙江省自然科学基金资助. 作者潘志庚, 1965年生, 博士, 副研究员, 主要研究领域为分布式图形, 限时计算, 虚拟环境和多媒体. 马小虎, 1964年生, 博士生, 讲师, 主要研究领域为虚拟环境, 限时计算和优化方法. 石教英, 1937年生, 教授, 博士导师, 浙江大学 CAD & CG 国家重点实验室主任, 主要研究领域为虚拟环境, 多媒体和科学计算可视化.

本文通讯联系人: 潘志庚, 杭州 310027, 浙江大学 CAD & CG 国家重点实验室

本文 1996-03-04 收到修改稿

得到了应用,然而,手工创建物体的不同 LoD 描述是极其费时的,因此,自动生成不同 LoD 模型显得相当重要.本文正是在这样的研究背景下,提出了一种基于三角形网格简化的 LoD 模型自动生成算法.

本文第 1 节讨论多 LoD 模型的自动生成技术,第 2 节是实现技术细节和实例,最后是结论.

## 1 多 LoD 模型自动生成

在计算机图形学应用领域,经常采用多边形网格(特例为三角形网格)来描述物体模型<sup>[4,5]</sup>,但用这种方法描述复杂物体时,就需要成千上万的多边形才能刻画出复杂物体的细节,结果导致庞大的物体模型.由于绘制时间和存储量与多边形的数目成正比,因此,过于庞大的物体模型通常是不实用的.特别地,在虚拟环境中绘制复杂场景时,为了达到图形的实时生成,可采用多个 LoD 来描述场景中的物体.<sup>[6]</sup>

### 1.1 相关研究工作

在一些应用领域中,研究人员对 LoD 方法进行了探讨.<sup>[7~10]</sup>在商用造型软件 Software systems 的 ModelGEN<sup>[7]</sup>或 Coryphaeus Software 公司的 Designers Workbench<sup>[8]</sup>能在一定程度上处理 LoD 节点,但是不提供生成多 LoD 的工具,文献[9]在数字地形图生成和显示应用中使用了 LoD 方法,而文献[10]则对 CAD 应用中多分辨率场景模型的生成方法进行了研究.在把 LoD 方法用在虚拟环境方面,德国 Darmstadt 图形研究所作了一些初步研究工作,针对特定的造型软件,实现了一种交互式多 LoD 场景模型生成工具<sup>[6]</sup>,但其所需的计算量很大,处理速度较慢.本文针对一般的多边形网格模型,使用网格简化方法进行多 LoD 模型自动生成方法的研究.

### 1.2 基于网格简化的多 LoD 模型生成算法

对给定的场景模型 S,基于网格简化来生成多 LoD 模型的方法可用算法 1 描述.

#### 算法 1.

步骤 1:确定多 LoD 模型中要求的细节层次个数  $N$ ,设层次分别为  $L_i, i=1, 2, \dots, N$ .

步骤 2:对于不同的层次  $L_i$ ,计算网格简化算法的终止条件(如要删除的顶点个数).

步骤 3:对于每一个所需的层次  $L_i$  进行网格简化(参见算法 2),所生成的场景模型保存到多 LoD 模型库中.

步骤 4:结束.

由于三角形网格在场景描述时用得比较普遍,因而下面先讨论三角形网格简化算法,然后再推广到一般的多边形情形.

### 1.3 三角形网格简化算法

为了下面的叙述方便和清楚起见,首先引入一些基本概念.

#### 1.3.1 基本概念

定义 1. 空间中一组三角形,沿公共边及在顶点处相邻接,把这样的一组三角形定义为三角形网格(TM),TM 可以由顶点集合  $V = \{V_1, V_2, \dots, V_n\}$  和三角形集合  $T = \{T_1, T_2, \dots, T_m\}$  所组成的二元组  $(V, T)$  来表示.

定义 2. 对 TM 中任一顶点  $V_i$ ,有 1 个或多个三角形使用了  $V_i$ ,把这些三角形定义为与

顶点  $V_i$  相关的三角形  $T_{ik}$ , 简称相关三角形 ( $0 < i \leq n, 0 < k \leq m$ ).

定义 3. 把  $TM$  边界上的顶点定义为边界顶点, 其余为非边界顶点.

定义 4. 对  $TM$  中任一顶点  $V_i$ , 与  $V_i$  相关的所有三角形  $T_{ik}$  按逆时针方向排列, 则构成有序三角形环(或半环), 简记为  $TL_i$ , 相应的顶点则构成有序顶点环(或半环), 简记为  $VL_i$ . 对边界顶点, 则有 1 个有序顶点半环围绕该顶点; 对非边界顶点, 则有 1 个有序顶点环围绕该顶点.

定义 5. 若与顶点  $V_i$  相关的三角形  $T_{ik}$  能构成有序三角形环  $TL_i$ , 并且与顶点  $V_i$  相关的三角形包含在有序三角形环  $TL_i$  中, 那么把  $V_i$  称为简单顶点. 否则把  $V_i$  称为复杂顶点.

定义 6. 设与简单顶点相关的有序三角形环中每个三角形的法向量为  $\vec{n}_i$ , 中心为  $\vec{x}_i$ , 面积为  $A_i$ , 那么由式(1)定义的法向量  $\vec{n}$  和中心  $\vec{x}$  构造的平面定义为三角形环的平均平面.

$$\vec{N} = \frac{\sum \vec{n}_i A_i}{\sum A_i}, \quad \vec{n} = \frac{\vec{N}}{|\vec{N}|}, \quad \vec{x} = \frac{\sum A_i \vec{x}_i}{\sum A_i} \tag{1}$$

### 1.3.2 算法描述

三角形网格简化算法包括 3 个主要步骤: (a) 计算可删除顶点; (b) 顶点删除; (c) 进行局部三角化, 具体方法可用算法 2 描述.

#### 算法 2.

步骤 1: 对  $TM$  中每一顶点  $V_i$ , 求与  $V_i$  相关的三角形  $T_{ik}$ , 作步骤 2 到步骤 7.

步骤 2: 使用上面给出的定义, 判定顶点  $V_i$  是否为边界顶点、复杂顶点或简单顶点.

步骤 3: 如果  $V_i$  为复杂顶点, 那么不删除  $V_i$ , 置该顶点的删除标志为假, 转步骤 7; 如果  $V_i$  为边界顶点, 那么使用点到直线的距离  $d$  作为简化标准, 其中直线为连接 2 个产生边界的顶点所形成的直线, 转步骤 4; 如果该顶点为简单顶点, 那么使用点到平均平面的距离  $d$  作为简化标准, 其中平均平面根据定义 6 求得, 转步骤 5.

步骤 4: 设形成直线的 2 个顶点为  $s$  和  $e$ , 候选顶点为  $v$ , 那么点到直线的距离  $d$  可由下面的公式求出, 并转步骤 6.

$$\text{令 } \vec{\mu} = \vec{sv}, \vec{\omega} = \vec{se}, \text{ 并设 } |\vec{\mu}| = a, |\vec{\omega}| = b \text{ 则: } c = \frac{\vec{\mu} \cdot \vec{\omega}}{b}, \quad d = \sqrt{a^2 - c^2} \tag{2}$$

步骤 5: 设平均平面的单位法向量和中心分别为  $\vec{n}$  和  $\vec{x}$ , 那么顶点  $v$  到平均平面的距离  $d$  可由下面的公式求出, 并转步骤 6.

$$d = |\vec{n} \cdot (\vec{v} - \vec{x})| \tag{3}$$

步骤 6: 如果距离  $d$  小于指定的简化标准阈值, 那么置该顶点的删除标志为真, 否则, 置该顶点的删除标志为假.

步骤 7: 对删除标志为真的顶点, 进行顶点删除及局部三角化操作(参见算法 3).

步骤 8: 重复以上步骤, 可以相应地调整简化标准, 直到终止条件满足为止.

### 1.4 局部三角化

如果删除一个顶点以及与该顶点相关的三角形, 那么由与该顶点相关的有序顶点环(或半环)所形成的多边形区域必须进行三角化. 由于每个有序顶点环都是星型的, 可以使用基于递归分割的三角化方法.<sup>[4]</sup> 这种方法计算比较复杂, 速度较慢, 为此我们提出一种简化方法, 其基本思想是: 对与简单顶点相关的有序顶点环, 通过投影到平均平面上, 从而把一个三维三角化问题转化为一个二维带约束的局部三角化问题. 实践表明, 这种方法不仅实现简

单,而且比较有效,具体方法可用算法 3 描述.

### 算法 3.

步骤 1:求出有序顶点环中每一顶点在平均平面上的正交投影,并用循环单向链表保存顶点,顺序连接投影点得到一平面多边形,转步骤 2.

步骤 2:如果该多边形是非自交的,则转步骤 3,否则,表示该环不能三角化,那么产生该环的候选顶点不能移去,转步骤 7.

步骤 3:计算出多边形顶点链表中每一顶点的凸、凹性.

步骤 4:在链表中顺序取出 3 个结点  $P, Q, R$ ,若  $Q$  为凸点,并且由  $P, Q, R$  对应的顶点所构成的三角形不包含多边形上其它顶点,那么,保存该三角形并从链表中删除  $Q$  结点,转步骤 5,否则重复步骤 4.

步骤 5:若链表中还存在 3 个以上的结点,则转步骤 3,否则转步骤 6.

步骤 6:由链表中最后 3 个结点构成 1 个三角形.

步骤 7:结束.

## 1.5 多边形网格简化

用多边形网格来描述物体,目前仍然是计算机图形学中模型表示最广泛采用的形式.其原因有 2 个:①图形工作站可以快速地绘制多边形;②有许多技术可以把一给定的模型转换成多边形数据集.因而,如何构造一种自动生成方法,来生成同一物体的不同细节层次的多边形模型显得尤为重要.解决这一问题有 2 种方案:①对 1.3 节中介绍的三角形网格简化算法稍作修改,就可直接应用于多边形网格简化;②可把初始多边形网格转换成三角形网格,然后直接调用 1.3 节介绍的算法,生成同一物体的不同细节层次的三角形网格模型.把 1 个多边形剖分成若干个三角形,可采用算法 3 的基本思想来实现.受篇幅所限,不再给出详细描述.

## 2 实现与实例

尽管三角形网格简化算法的基本思想比较简单,但要实现一个有效的网格简化算法必须解决以下 2 个问题.①由于要存取和存储大量的数据,所以要设计高效的数据结构;②三角化算法要设计得简单有效.

### 2.1 数据结构

数据结构必须包含每个顶点的坐标信息,每个三角形的定义信息(用 3 个顶点序号来表示).另外,由于经常要使用围绕每个顶点的有序三角形环,因此,还需要保存使用每个顶点的三角形表.所以,三角形网格简化算法所使用主要数据结构为:(a) 顶点表;(b) 三角形定义表;(c) 使用每个顶点的有序三角形环所组成的表.

### 2.2 实现时的一些不同策略

在实现时,顶点删除和局部三角化可按不同策略交替或顺序执行.

(I)对任一候选顶点,估算其距离标准  $d$ ,如果  $d$  小于给定的阈值,则删除该顶点及与其相关的所有三角形,再对由有序顶点环所围成的区域进行三角化,重复以上过程,并适当增大阈值,直到某个结束条件满足为止.

(II)对所有的候选顶点,估算其距离标准  $d$ ,如果  $d$  小于给定的阈值,则置该顶点删除标志为真,重复以上过程,并适当增大阈值,直到某个结束条件满足为止,然后,一次性删除

所有的删除标志为真的顶点,再调用相应的三角化算法,得到简化了的三角形网格模型.

无论是对策略(I)还是对策略(II),若某一候选顶点满足删除阈值,那么该顶点为可删除顶点,但接下来的操作有2种不同方案:①把与该删除顶点相邻接的顶点设置为不可删除的点,直到下一次循环改变阈值为止;②直接处理下一个顶点.实践表明方案①比方案②能保持更多的特征信息.

### 2.3 实 例

我们把基于三角形网格简化的多LoD模型生成算法应用于不同的三角形网格模型,得到了同一物体的多个不同的三角形网格模型,参见图1和图2.

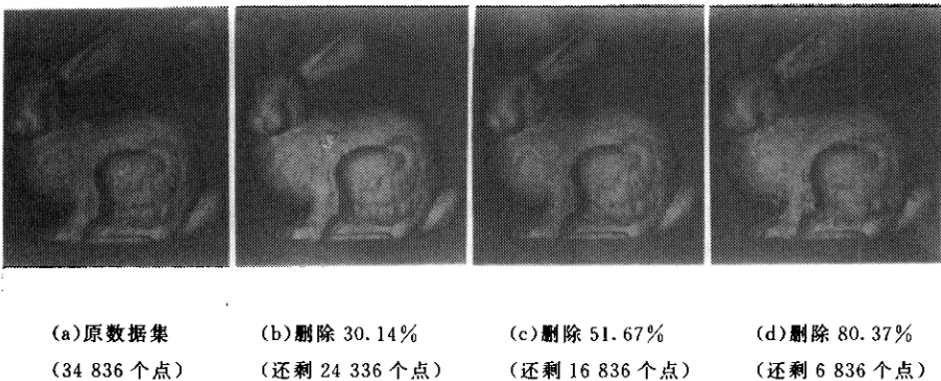


图 1 算法生成的多LoD(例1)

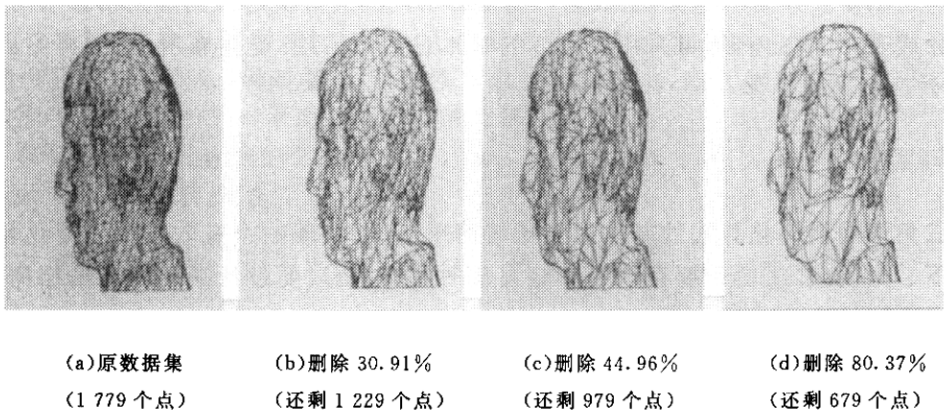


图 2 算法生成的多LoD(例2)

### 3 结 论

这里介绍的多LoD模型自动生成算法,不仅可以用于自动构造虚拟环境中复杂场景的多个不同细节层次模型,以达到实时图形绘制的目的,而且可以用作交互式可视化系统的过滤器,大大简化需要实时显示的物体模型,以达到交互式可视化的目的.

另外,本文所介绍的多LoD表示方法比文献[6]中的方法自动化程度高,不需要交互操作,而且它不依赖于所应用的数据集.进一步的研究工作包括:(1)更为有效的数据结构的构造;(2)网格简化的逼近程度度量;(3)其它三角化方法的选用.

致谢 在本课题的研究过程中,郑文廷同志参加了部分基础研究工作;张明敏同志提供了供观察用的绘图程序,在此一并表示感谢。

### 参考文献

- 1 石教英. 虚拟环境(VE)技术及其应用. 全国第1届虚拟环境研讨会论文集, 杭州, 1994. 1~20.
- 2 潘志庚. 分布式图形处理的理论与应用研究[博士论文]. 浙江大学, 1993.
- 3 Pan Zhigeng, Zhang Mingmin, Zheng Wenting *et al.* Time-critical computing in virtual environment. Proceedings of CAD/Graphics'95, Wuhan, Oct. 1995. 1078~1083.
- 4 Schroeder W J, Zarge J A *et al.* Decimation of triangle meshes. *Computer Graphics*, 1992, **26**(2):65~70.
- 5 Turk Greg. Re-tiling polygonal surface. *Computer Graphics*, 1992, **26**(2):55~64.
- 6 Peter Astheimer, Maria-luise poche. Level-of-detail generation and its application in virtual reality. Proceedings of the VRST'94 Conference, World Scientific, 1995. 299~309.
- 7 Software Systems Inc. ModelGen's users guide. 1993.
- 8 Coryphaeus Software Inc. Designer's workbench introduction. 1994.
- 9 Schroder F, Rodbach P. Managing the complexity of digital terrain models. *Computer & Graphics*, 1994, **18**(6): 775~783.
- 10 Rossinac J, Borrel P. Multiresolution 3D approximation for rendering complex scenes. In: Falcindieno eds., *Modeling in Computer Graphics*, Springer, 1993. 455~465.

## THE AUTOMATIC GENERATION ALGORITHM FOR MODELS AT MULTIPLE LEVELS OF DETAIL IN VIRTUAL ENVIRONMENT

Pan Zhigeng · Ma Xiaohu · Shi Jiaoying

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

**Abstract** Virtual environment is a typical application of time-critical computing or time-critical rendering. This paper addresses the level of detail representing models in time-critical rendering. The authors present an automatic generation algorithm for models at multiple levels of detail based on triangle mesh simplification. The algorithm makes multiple passes over an existing triangle mesh, using local geometry and topology to remove vertices that pass a distance or angle criterion. The holes left by the vertex removal are patched using a local triangulation process. Examples illustrate efficiency of the algorithm.

**Key words** Virtual environment, levels of detail, time-critical computing, mesh simplification, constrained triangulation.