

# 中华博士 园地

这是本刊特为海内外正在就读和学成立业的博士、博士后青年学者们开辟的一片科普园地. 深学浅著是一门德识、慧学、素质修养的学问. 你们的新知识、新调研、新观察、新目光、新展望, 能够用尽可能深入浅出、通俗流畅的语言, 汇报给祖国人民、家乡父老子弟乡亲们吗? 中华博士园地, 乃耕耘忠孝之地, 科教兴国、民族昌盛之地. 要用慈母听得懂的语言, 写出你们的心声!

中图法分类号: TP391.4 文章编号: 1006-8961(2002)01-0097-06

## 基于 LOD 实时图形绘制和加速技术

淮永建 郝重阳

(西北工业大学电子与信息工程研究所西安虚拟现实工程技术研究中心, 西安 710072)

### 0 前言

三维几何模型和场景的交互可视化是计算机图形、科学计算可视化、虚拟现实技术和 3D 游戏中很重要的研究主题. 许多应用领域(诸如驾驶模拟、战场可视化、建筑漫游、虚拟现实、远程会诊、数字城市和游戏娱乐等)都要求场景有很高的显示帧速率, 以使得用户能对场景中的对象进行实时交互, 进而获得对场景的视觉理解和感知.

大家知道, 复杂三维场景可视化对系统图形绘制处理能力要求很高, 需要专用的图形芯片来进行硬件加速, 因为可视化应用中所绘制的场景日益复

杂, 且要求很高的真实感, 即便是最高端的图形工作站也不能满足实时绘制的需求, 所以需要采用一些算法对场景绘制进行简化加速. 目前, 通常采用可见性剔除和多边形简化技术来对大的多边形场景绘制进行加速, 其中, 可见性剔除方法是通过采用视锥体裁剪技术来剔除掉视点之外的场景多边形, 以加速场景绘制; 多边形简化是用较少的多边形数来近似表示初始模型, 以减少所绘制的多边形数目. Clark 提出的多层次细节(LOD)模型简化技术<sup>[1]</sup>是目前多边形简化方法中研究最热的一个领域. LOD 简化的目的是为了对多边形模型或场景进行多分辨率表示, 即应用中, 依据模型和视点的距离, 通过选择适当分辨率的模型表示来进行绘制. 如果模型离视点较远, 且在屏幕空间的投影区域覆盖较少的象素, 则用粗糙的模型来进行表示; 相反, 模型离视点较近, 则采用精细的模型进行绘制(如图 1 所示).

### 1 LOD(Levels of Detail)

LOD 简化多边形的目的, 不是为了从初始模型

淮永建 1970 年生, 博士研究生. 研究方向为计算机图形图象、多媒体技术和虚拟现实技术等.

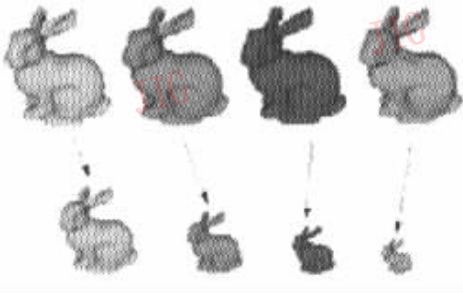


图1 兔子模型 LOD 表示

中移去粗糙的部分,而是为了保留重要的视觉特征来生成简化的模型,其理想的结果应是一个初始模型序列的简化<sup>[2]</sup>,这样简化模型才可以应用于不同的实时加速.为了实现连续帧的绘制,LOD 简化方法必须均衡考虑模型的细节度和绘制速率.

LOD 简化算法需要某种启发式准则,以选择相关的近似误差估计来控制简化,下面讨论简化中应考虑的各种条件准则.

### 1.1 LOD 连续性

利用 LOD 进行实时绘制时,不同 LOD 模型转换时所引起的图象跳跃,是简化中急需解决的问题,因此在实时绘制时,需要从 LOD 简化序列中选择合适的 LOD 模型.如果连续的 LOD 模型之间相差仅一个或两个多边形,则转化时可保持模型的视觉连续性.任意网格多分辨率简化算法<sup>[3]</sup>和递增网格法<sup>[4]</sup>都提出了连续 LOD 模型生成的数据结构,它们虽使得不同 LOD 模型转化时能光滑改变,但由于实际应用中,要考虑存储和计算需求,仅生成有限层次细节度的 LOD 模型,且不同层次细节间的多边形数差异较大,转化时会引起图象的跳跃,因此层次细节光滑技术在简化中显得至关重要.

### 1.2 形状保留

由于简化算法必须尽可能保留模型的形状和表面特征,因此算法必须首先找出模型的特征信息(如平面曲率、尖点和特征边),然后才能通过融合平坦的区域和线性变化的特征边来简化模型.如今,大多数算法是用特征边折叠来简化模型<sup>[4,5]</sup>或采用合并曲率较小的相邻面<sup>[2,6]</sup>来简化模型,并通过门限值来控制简化.这种合并相邻面的方法,门限值可定义为相邻平面法向量的角度值,即超过门限值的平面不予合并,且门限值越大,简化程度越高.

### 1.3 近似误差估计

为了控制简化过程,每一步简化均应评价模型局部的近似误差,如有的算法<sup>[4]</sup>使用局部的误差估计或

者距离准则<sup>[3]</sup>(顶点到平面的距离)来评判简化误差;而有的算法<sup>[2,6]</sup>则采用几何体来限制简化程度.

LOD 简化算法中,目前急需解决的问题是:(1) LOD 模型的自动生成;(2) LOD 模型转换时的图象跳跃;(3)同一场景不同的层次细节区域间如何保持网格的连续性,而不出现裂缝;(4)简化中,如何保留模型的表面特征信息.

## 2 基于 LOD 几何模型绘制

多边形几何模型简化中,生成 LOD 模型的主要方法有:细分法、采样法和删减法.

### 2.1 细分算法

细分算法首先用非常简单的基模型表示初始模型,然后迭代细分基模型,这样每一步就向模型的局部区域增加越来越多的细节,直到细分的模型满足用户定义的误差需求时,则算法终止.

Eck 提出的算法是首先用基于小波的方法来生成 LOD 模型<sup>[7]</sup>;然后通过改变小波系数来光滑 LOD 模型转化,以减少图象“跳跃”.该方法虽可处理无折叠多边形网格模型,但不能准确地处理具有折痕和突变边模型.它是一种自适应的细分法,其在简化中保留了模型的拓扑结构,并能依据顶点到平面的距离来测试近似误差.为了将 3D 网格参数化成平面三角形,需进行谐函数映射.由于该算法数学上的精确表述,可使简化模型的近似度很高,因而具有很大的吸引力,但计算量太大,且基于小波有效网格的提取代价也很大.

### 2.2 采样法

采样算法是通过随机抽取一些点来初始化多边形模型,再通过较少的多边形数来重建初始模型,其在采样点处满足误差估计.该算法用户虽可以控制采样点数,但不能控制最终近似的初始多边形模型.

Turk 提出的算法可光滑处理没有尖边和不连续的曲面模型<sup>[8]</sup>,是一种基于多边形数的简化算法,它允许用户指定简化模型的顶点数量,但不能修改算法提供的逼真度测试.该算法是首先随机将给定的顶点数分布在模型表面,然后根据顶点间的斥力,对顶点进行重分布;接着,对模型重新进行三角形剖分,再依据误差准则来迭代移去初始顶点,并对留下的洞重新进行三角形划分;最后通过几何渐变法对 LOD 模型转化进行光滑处理.

Hoppe 提出的网格优化法是一种复杂的采样处

理算法<sup>[9]</sup>. 该算法通过一系列的样本点来重建三维模型, 并将简化问题当作优化的过程来处理, 其简化模型的顶点数量和近似误差可通过能量最小函数来确定. 除了初始网格的顶点外, 该算法首先通过随机放置一些样本点来测试模型的近似度; 接着, 随机选取一条边执行边折叠、边分割、边交换 3 种操作; 最后通过调整顶点位置, 以使模型的能量函数最小. 由于算法能产生很高逼真度, 因此非常适合 CAD 模型简化, 并可处理表面模型突变的区域, 但其最大的缺点是模型简化速度太慢.

顶点聚类算法是将顶点集分割成顶点束集合, 并将位于同一束内的顶点用单一的顶点代替. 该算法虽运算速度快、简化效率高, 但简化的近似质量差, 并改变了模型拓扑结构. Rossignac 提出了最简单的均匀顶点聚类算法<sup>[10]</sup>. 在该算法中, 顶点集用均匀网格的边界盒来分割, 每个单元新的顶点通过基于边长度的计算准则来获得, Low 对该算法又进行了改进<sup>[11]</sup>, 它不仅提高了图形的近似质量和简化速度, 并能处理所有的三维模型输入(包括自相交和退化的几何模型), 与 Rossignac 提出的算法<sup>[10]</sup>不同之处是它采用改进的系统来初始化顶点权值, 即对重要的顶点采用非均匀的单元细分, 由于这种浮动的单元聚类法对网格单元尺寸的改变不敏感, 因此用户可根据需求来指定网格单元的大小, 以控制模型简化程度.

### 2.3 删减法

删减法主要通过几何移去模型单元来进行简化处理. 几何移去方法包括顶点移去法、边折叠法、边收缩法和三角形删减法.

#### 2.3.1 顶点移去法

顶点移去法是在每一步简化处理中, 选择一个顶点移去, 并将所有与顶点相关的面从模型中移去, 而将留下的洞重新进行三角形网格划分. 以前的顶点移去算法采用相当保守的近似误差估计, 而 Kobbelt 和 Soucy 提出了更精确的误差估计方法<sup>[12,13]</sup>, 如局部 Hausdorff 距离误差估计方法就是将初始模型顶点与邻近近似表面对应起来, 再用这些点和关联的面来定义近似误差. 该算法无论从空间上, 还是从时间上来说, 虽都是一种很有效的简化算法, 但不能较好保留物体表面的光滑度, 即其简化处理后物体表面将变得十分粗糙.

#### 2.3.2 边折叠法

边折叠法中最有代表性的是 Hoppe 提出的累

进网格法<sup>[4]</sup>, 也是引起人们普遍关注的 LOD 简化方法, 因为这种边折叠能生成连续的 LOD 模型. 该算法通过边折叠操作来简化模型, 其折叠后不仅产生一新顶点, 且每一次简化均移去了两个面、一条边和一个顶点. 该算法的优点是考虑了模型表面的颜色、纹理、表面法向量等特征信息, 并考虑了如何通过将这些信息引入到模型的能量函数中来控制模型的简化, 因此简化速度快, 可应用到实时 LOD 模型之间的光滑转化. 该算法不仅保留了模型的几何信息, 而且其他重要特征信息(颜色、纹理和拓扑结构)也得到了保留. 最近, Hoppe 又将该算法改进为实时视相关的累进网格法<sup>[14]</sup>, 该算法的特点是考虑了视点的改变, 即依据视点变化来选择适当的 LOD 模型.

#### 2.3.3 迭代的顶点对收缩法

Hoppe, Garland 和 Lindstrom 采用的基于顶点对的迭代收缩算法<sup>[15~17]</sup>是目前模型简化的一种主要方法. 该迭代收缩算法的主要优点是通过收缩产生的层次结构可有效地生成多分辨率表面模型, 并能用于实时视相关的 LOD 模型简化. 这种顶点对收缩可描述为:  $(v_i, v_j) \rightarrow v$ , 并采用以下 3 步来简化表面:

- (1) 移动顶点对  $v_i, v_j$  到  $v$ ;
- (2) 用  $v_i$  代替所有出现的  $v_j$  顶点;
- (3) 移去顶点  $v_j$  和所有退化的面

Garland 通过迭代的顶点对收缩算法<sup>[16,17]</sup>来简化模型, 该方法是利用二次曲面误差估计来度量简化模型的近似度, 并用一种更有效的平面集来隐含表示近似误差. 尽管该算法牺牲了近似模型的准确性, 但它能迅速地产生高质量的近似效果, 因此是目前简化速度最快的模型简化算法. Lindstrom 提出的算法<sup>[18]</sup>与大多数算法不同之处在于, 它不需要初始模型的形状信息, 可直接根据近似度来简化模型, 并采用基于体保留的线性约束法来选择边收缩和目标顶点位置. 该算法主要优点是内存消耗少, 因此解决了多层次 LOD 模型的存储问题.

## 3 基于 LOD 场景绘制

### 3.1 静态场景简化

目前已有许多算法用来对地形和以多边形网格表示的高度场进行简化. 这些算法企图用给定的顶点数、定义的误差边界或保留关键特征的方式来对场景进行简化, 并采用均匀网格或不规则三角形网

格来表示地景,即用层次细分或网格减少的方法来简化场景,以生成多分辨率 LOD 场景,如 Heckbert 提出的一种通用的多分辨率 LOD 地景选择方法<sup>[19]</sup>、Eck 提出的一种通用不规则网格表面细分方法<sup>[3]</sup>以及 Stadt 提出的一种基于小波分析的地形 LOD 层次细节模型生成方法<sup>[21]</sup>。

### 3.1.1 实时连续的 LOD 高度场绘制

Lindstrom 提出了一种高复杂的多边形表面实时层次细节模型简化和显示算法<sup>[22]</sup>。该算法用均匀网格表示地形,还引入了层次二叉树技术来生成地形的 LOD 模型,并用一屏幕空间的门限变量来表示投影图象的最大误差。该算法基于块来选择适当的 LOD 层次,当视点改变时,则依据视距,通过对网格块 LOD 模型进行从下往上的顶点删减来获得高帧速率地形绘制。虽然这种基于块的层次细节增量更新,充分考虑了帧间和对象空间的相关性,但块的自下往上的顶点删除却限制了算法的性能。

### 3.1.2 高度场连续 LOD 模型的实时生成

Lindstrom 提出的算法<sup>[22]</sup>在实时生成连续不同 LOD 层次模型转化时,会产生图象跳跃,即当观察接近细节丰富的区域时,由于与某一视距相一致的细节层次会突然出现,从而引起图象跳跃,为此, Rottger 提出了通过几何渐变来平滑 LOD 模型转化的算法<sup>[23]</sup>,该算法采用自上而下的二叉树数据结构来生成三角形网格,并采用无附加代价估计法来进行不同 LOD 模型的几何形变。该算法在进行顶点删除时,不仅考虑了视点距离,而且考虑了表面模型的粗糙度,当采用自上而下的策略来进行顶点删除时,由于对每一帧图象绘制只需访问整个数据集的局部顶点,因此对大的高度场数据集也可产生很高的帧速率。另外,二叉树数据结构还考虑了有效的视窗裁减,从而使内存的使用被限制在需要高度场数据、纹理映象的空间。

### 3.1.3 实时优化的自适应地形生成(ROAM)

Duchaineau 提出了一种在所需的帧速率下,通过保持动态视相关的三角形网格和纹理映射来生成高质量图象的方法<sup>[24]</sup>。该方法用优化灵活的视相关误差来度量生成的三角形网格,以产生确定的误差边界,由于该算法基于三角形数目来对场景进行简化,并考虑了帧间的相关性,因而可获得很大的帧速率显示。与文献<sup>[22]</sup>、<sup>[23]</sup>算法不同,ROAM 算法采用二叉树方法来表示场景,当增加和移去顶点时,即可通过分裂和融合三角形二叉树结构的方法来生成

连续的 LOD 场景。另外,还可采用双队列的方法来对算法进行优化,并驱动分裂和融合操作,以保持网格的连续性。

### 3.1.4 任意多边形环境视相关的简化

Luebke 采用分级动态简化法(HDS),并用视相关方法来对大的多边形场景进行自适应的简化<sup>[26]</sup>。HDS 通过顶点对收缩操作建立的顶点层次树结构称为顶点树,这种顶点树提供了一种整个场景的连续 LOD 表示方法。实时绘制时,HDS 遍历顶点树,对场景中任何有效的多边形进行绘制,并通过有效多边形来确定屏幕空间误差门限、轮廓边界和三角形概算等。对于给定的场景,HDS 可用任何多边形简化算法来生成顶点树,HDS 的默认算法是采用二叉树结构来分割空间,然后将二叉树节点中的顶点融合到该空间最重要的顶点上。这样通过该简化算法可将场景中不连接的部分融合到一起,以产生理想的模型集近似效果。

## 3.2 动态场景简化

目前对动态场景简化的研究不多,其主要原因是随着场景中对象的运动,场景简化的复杂度不断增加,因此简化难度较大。Rohlf 用传统的 LOD 技术来简化动态场景<sup>[27]</sup>,当物体移动时,用该技术可对场景图进行修改。目前大多数对动态场景的研究主要集中于对象运动后边界体层次的更新和空间的分割上。

### 3.2.1 可视化场景中最优的 BSP 树空间分割算法

Torres 通过引入 6 层的 BSP 树结构来简化动态场景<sup>[28]</sup>。BSP 的第 1 层结构由用户定义的约束平面组成;第 2 层平面集用来将场景中的模型完全分割开,但在有些情况下,模型完全分割开是不可能的;第 3 层平面集可通过牺牲模型的方法将其分裂;第 4 层平面集称为包裹平面,可用来将模型包装起来,以防止通过牺牲模型的方法将其分裂,该算法用二等分平面法来平衡 BSP 树内部的对象;第 5 层的平面集则尽量将对象的多边形分裂成两个等密度的区域;最后,第 6 层的 BSP 树结构包括牺牲对象的多边形。这样,当场景中模型运动时,仅仅局部 BSP 树需要更新,即通过仔细选择顶层的动态 BSP 树结构,就可以有效地更新 BSP 树。

### 3.2.2 计算动态改变的 BSP 树

当对场景中的对象进行移动、插入和删减操作时,Chrysanthou 提出了一种有效的 BSP 树更新方法<sup>[29]</sup>,即对于模型的运动,可将其从 BSP 树中删减、

然后重新将模型插入到树中来处理。该算法用滤波的方法将模型插入到 BSP 树中,插入步骤从 BSP 树的根节点开始。如果多边形被插入到 BSP 树节点多边形的前方,则该算法循环遍历节点前方;如果插入到节点多边形的后方,则循环遍历节点后方的子节点;如果循环到一叶节点,则新的子节点创建多边形插入;如果被插入的多边形与节点多边形共享一平面,则将其增加到该节点。这里,多边形的删减,可通过如下方法来处理:如果删减多边形是一叶节点,则删除该节点;如果多边形与其他多边形共享一节点,则删除该多边形;如果删除多边形的节点仅有一个子节点,则用子节点代替该节点;最后,如果删减多边形有两个子节点,则用大的子节点代替该节点。由于这些技术允许增量更新 BSP 树,因此与重新计算 BSP 树相比,效率更高。

### 3.2.3 基于 HLOD 大的静态和动态场景的快速显示

Erikson 提出了一种新的静态和动态场景快速显示方法<sup>[30,31]</sup>。该算法给定一几何数据集,采用场景图表示几何体,并自动计算场景图节点的层次细节模型。为了有效地简化场景,特别是场景多边形的数量,以达到实时显示的目的,首先必须计算局部场景图的分层层次细节模型(HLOD);然后当场景中物体运动时,该算法就能实时更新与动态场景改变对应的 HLOD 模型。更新时,它首先更新受物体运动影响的 HLOD 模型的误差边界,然后依据空间近似度,重新聚集场景图节点,最后增量更新场景图边界体层次。这样,一旦场景图被修改,则将 HLOD 需重新计算的节点插入到队列中,再采用简化处理方法计算 HLOD 模型。该算法不仅可有效地处理有限模型运动的动态场景简化,而且该算法还支持基于图象质量的绘制和基于目标帧频率的绘制两种绘制模式。本算法由于利用了显示列表绘制 LOD 模型和 HLOD 模型,从而加速了大的静态和动态场景的绘制,进而满足了实时显示的需要。

## 4 研究展望

在实时可视化领域中,表面简化算法和 LOD 简化技术已取得了许多研究成果,有些成果已成功应用于商业化的建模和视景生成软件,其中,多边形表面简化是提高图形绘制效率,增强交互性的一种有效的实时绘制方法,尤其是 LOD 简化技术是目前研究的热点。在简化算法中,目前第 1 个趋势是越

来越多的研究人员采用边折叠的方法,这是由于序列边折叠方法可生成连续的 LOD 模型的缘故;同时人们也越来越多地采用拓扑结构来改变简化算法,以提高简化效率;第 2 个趋势是在简化中越来越多的算法均尽可能保留初始模型的特征(如纹理、光照、形状和表面法向量等表面特征),以便建立基于几何和表面特征相混合的简化方法;第 3 个趋势是在简化中,充分考虑了视点的变化和方向,也就是说,研究随视点改变的模型简化方法也是研究的主流。另外,由于所绘制的场景越来越复杂,大量多边形的绘制和存储将影响实时绘制性能,因此采用特殊的数据结构,以便在显示时,实时地生成视相关、连续的 LOD 模型也是未来研究的主流,而且该方法占用的存储空间也小。

目前加速图形绘制方法主要有基于图象的绘制方法和基于几何模型的绘制方法两大类。其中,基于图象的绘制方法不仅可生成与照片一样真实感强的图象,而且具有绘制速率与场景复杂度无关的优点,其缺点是不能满足交互显示的目的;而基于几何模型的绘制方法虽交互性较好,但绘制速率与场景的复杂度有关,即场景越复杂,绘制时间越长。另外,基于图象和几何模型相混合的实时加速绘制方法也是目前的研究热点,该方法对于视点远的场景采用多层次图象方法进行绘制,而对于视点近的模式则采用基于几何模型的 LOD 模型进行绘制。

## 参考文献

- 1 Clark J. Hierarchical geometric models for visible surface algorithms [J]. Communications of the ACM, 1976, 19 (10): 547~554.
- 2 Cohen J, Wright W. Simplification envelopes[A]. In: Proceedings of SIGGRAPH '96[C], Held in New Orleans, Louisiana, August 1996;119~128.
- 3 Eck M, DeRose T, Duchamp T *et al.* Multiresolution analysis of arbitrary meshes. In: Proceedings of SIGGRAPH '95[C], Held in Los Angeles, California, August 1995: 173~180.
- 4 Hoppe H. Progressive meshes [A]. In: Proceedings of SIGGRAPH '96 [C], Held in New Orleans, Louisiana, August 1996;99~108.
- 5 Algorri M-E, Schmitt F. Mesh simplification[A]. In: Proceedings of EuroGraphics'96[C], 1996: 77~86.
- 6 Guezic A. Surface simplification inside a tolerance volume[R]. Technical Report RC 20440, IBM Research Division, T. J. Watson Research Center.
- 7 Eck M, DeRose T, Duchamp T *et al.* Multiresolution analysis of

- arbitrary meshes[A]. In: Proceedings of SIGGRAPH '95[C], Held in Angles, California, August 1995:173~182.
- 8 Turk G. Re-tiling polygonal surfaces[A]. In: Proceedings of SIGGRAPH '92 [C], Held in Chicago Illinois, July 1992: 55~64.
  - 9 Hoppe H, DeRose T, Duchamp T *et al.* Mesh optimization[A]. In: Proceedings of SIGGRAPH '93[C], Held in Anaheim, California, August 1993:19~26.
  - 10 Rossignac J, Borrel P. Multi-resolution 3D approximations for rendering complex scenes[J]. *Geometric Modeling in Computer Graphics*, 1993,5(2):455~465.
  - 11 Low K, Tan T. Model simplification using vertex-clustering [A]. In: Proceedings of ACM Symposium on Interactive 3D Graphics '97[C], ACM SIGGRAPH, April 1997:75~82.
  - 12 Kobbelt L, Campagna S, Seidel H. A general framework for mesh decimation[A]. In: Proceedings of Graphics Interface '98 [C], 1998: 43~50.
  - 13 Soucy M, Laurendeau D. Multiresolution surface modeling based on hierarchical triangulation[J]. *Computer Vision and Image Understanding*, 1996,63(1):1~14.
  - 14 Hoppe H. View - dependent refinement of progressive meshes [A]. In: Proceedings of SIGGRAPH '97[C], Held in Los Angles, California, August 1997:189~198.
  - 15 Popovi J, Hoppe H. Progressive simplicial complexes[A]. In: Proceedings of SIGGRAPH '97[C], Held in Los Angles, California, August 1997:217~224.
  - 16 Garland M, Heckbert P. Surface simplification using quadric error metrics[A]. In: Proceedings of SIGGRAPH '97[C], Held in Los Angles, California, August 1997:209~216.
  - 17 Garland M, Heckbert P. Simplifying surfaces with color and texture using quadric error metrics[A]. In: Proceedings of IEEE Visualization '98 Conference [C], Held in Research Triangle Park, North Carolina, October 1998:263~269,542.
  - 18 Lindstrom P, Turk G. Fast and memory efficient polygonal simplification[A]. In: Proceedings of IEEE Visualization '98 Conference[C], Held in Research Triangle Park, North Carolina, October 1998:279~286.
  - 19 Heckbert P, Garland M. Multiresolution modeling for fast rendering[A]. In: Proceedings of Graphics Interface '94[C], Held in Alberta, Canada, May 1994:43~50.
  - 20 Evans F, Skiena S, Varshney A. Optimizing triangle strips for fast rendering [A]. In: Proceedings of IEEE Visualization '96 Conference[C], Held in San Francisco, California Carolina, October 1996:319~326.
  - 21 Stadt O, Gross M, Gatti R. Fast multiresolution surface meshing [A]. In: Proceedings of IEEE Visualization '95 Conference [C], Held in Atlanta Georgia, October 1995:135~142.
  - 22 Lindstrom P, Koller D. Real-time, continuous level of detail rendering of height fields[A]. In: Proceedings of SIGGRAPH '96 [C], Held in New Orleans, Louisiana, August 1996: 109~118.
  - 23 Röttger S, Heidrich W, Seidel H. Real-time generation of continuous levels of detail for height fields[R]. Technical Report 13/1997, Universität Erlangen-Nürnberg.
  - 24 Duchaineau M. Wolinski M. ROAMing terrain: real-time, optimally adapting meshes [A]. In: Proceedings of IEEE Visualization '97 Conference[C], Held in Phoenix Arizona, October 1997:81~88.
  - 25 Xia J, El-Sana J, Varshney A. Adaptive real-time level-of-detail-based rendering for polygonal models [J]. *IEEE Transactions on Visualization and Computer Graphics*, 1997,3(2):171~183.
  - 26 Luebke D, Erikson C. View-dependent simplification of arbitrary polygonal environments [A]. In: Proceedings of SIGGRAPH '97[C], Held in Los Angles, California, August 1997: 199~208.
  - 27 Rohlf J, Helman J. IRIS performer: A high performance multi-processing toolkit for real-time 3D graphics[A]. In: Proceedings of SIGGRAPH '94[C], Held in Orlando, Florida, July 1994: 381~394.
  - 28 Torres E. Optimization of the binary space partitioning algorithm (BSP) for the visualization of dynamic scenes[A]. In: Proceedings of Eurographics '90[C], 1990:507~518.
  - 29 Chrysanthou Y, Slater M. Computing dynamic changes to BSP trees[A]. In: Proceedings of Eurographics '92[C], 1992:321~332.
  - 30 Erikson C, Manocha D. GAPS: General and automatic polygonal simplification[A]. In: Proceedings of ACM Symposium on Interactive 3D Graphics '99[C], ACM SIGGRAPH, April 1999: 79~88,225.
  - 31 Erikson C, Manocha D. Hierarchical levels of detail for fast display of large static and dynamic environments [R]. UNC-CH Technical Report TR00-012.