多分辨率BSP树的生成及应用*

陶志良, 成迟薏, 潘志庚, 石教英

(浙江大学CAD&CG国家重点实验室,浙江 杭州 310027)

E-mail: chengcy@cad.zju.edu.cn

http://www.cad.zju.edu.cn

摘要: CAD和科学可视化应用要求能够让用户对复杂场景进行交互式显示或者处理,因此,多细节层次技术被广泛应用于图形系统中以提高处理效率.通过构造一种新的BSP(binary space partitioning tree)树形式来加速多细节层次模型的绘制,这种技术已成功地应用于所开发的多分辨率模型编辑系统.同时,详细描述了新的BSP树结构的构造和绘制过程.

关键词: BSP树;多分辨率模型;实时绘制;网格简化

中图法分类号: TP391 **文献标识码**: A

计算机辅助设计和科学可视化应用要求能够让用户对复杂场景进行交互式显示或者处理.但是,这些场景的复杂程度往往超过了当前图形系统的交互显示能力,而且现在模型大小的增长速度也远远超过了硬件绘制能力的增长[1].

高度复杂的多边形网格模型非常不利于存储、传输和绘制[2],目前,如何减少要绘制的多边形的算法可以粗略地分成两类,一类是可见性裁剪,另一类就是使用多细节层次(level of detail,简称LoD模型).

所谓LoD模型方法,即为每个物体建立多个近似模型,不同模型对物体的细节描述其程度是不同的,细节描述越精确,模型越复杂.根据物体在屏幕上所占区域的大小及用户视点等因素,为各物体选择不同的LoD模型,从而减少所需显示的多边形数目,许多LoD技术已经被成功地嫁入各种虚拟现实和科学计算可视化的应用中[3].

空间二分树(binary space partitioning tree,简称BSP树),是一种非常高效的排序和分类数据结构,通常利用BSP树结构可以辅助绘制复杂的多边形场景,通过从后往前或者从前往后的次序一一绘制场景对象,自动完成场景绘制中的消隐.它的应用范围非常广泛,从隐藏面消除和光线跟踪层次结构到实体造型和机器人运动策划等应用都使用了BSP树结构[4~6].

本文在详细阐述LoD模型和BSP树的各自特点的基础上,讨论如何利用BSP树结构来加速绘制动态LoD模型,并提出一种新的支持多分辨率模型绘制的多分辨率BSP树结构,它具有结构简单、适应性强、绘制效率高等特点.这种技术已经应用于我们开发的多分辨率模型编辑系统中.

本文第1节简要介绍BSP树的构造和特点,说明BSP树结构在多细节层次网格模型绘制中的局限性,并提出一种新的支持多分辨率模型绘制的BSP树结构(multi-resolution binary space

* 收稿日期: 1999-04-12; 修改日期: 1999-09-29

基金项目: 国家自然科学基金资助项目(60083009)

作者简介: 陶志良(1974-),男,浙江绍兴人,硕士生,主要研究领域为虚拟现实,计算机图形;成迟薏(1976-),女,江苏南通人,硕士生,主要研究领域为虚拟现实,计算机图形;潘志庚(1965-),男,江苏淮阴人,博士,研究员,博士生导师,主要研究领域为虚拟现实,分布式图形;石教英(1932-),男,浙江宁波人,教授,博士生导师,主要研究领域为多媒体计算,虚拟环境,科学计算可视化.

partitioning tree,简称 MRBSP).第 2 节在 MRBSP 结构的基础上提出进一步的优化过程.第 3 节给出实验结果及其分析以及相应的图例.最后是结论.

1 多分辨率 BSP 树及其生成

1.1 BSP 树简介及其与多细节层次模型的结合

BSP 树是一种标准的二叉树,一般用来在 n 维空间中进行对象(polytopes)排序和搜索.整个 BSP 树在应用中表示整个场景,其中每个树节点分别表示一个凸子空间.每个节点里面包含一个"超平面(hyperplane)"作为二分空间的分割平面,该节点的两个子节点分别表示被分割成的两个子空间.另外,每个节点还可以包含一个或者多个几何对象.

为了说明简单,下面,我们以二维空间为例来描述 BSP 树的原理和构造,并且假定只处理平行于 X 和 Y 轴的线段对象.给定一个二维空间 X-Y(如图 1 所示),首先选择第 1 次空间分割的分割"超平面",这里选择为一条平行于 X 轴的直线作为初始 BSP 树的根节点,如图 1(b)所示.以后每次分割都在各自的子空间里选择一条合适的分割线,不断地把子空间分割成更小的空间,直到满足一定的收敛条件为止.BSP 树结构中每个节点的两个子节点分别表示背向"超平面"法向量方向的子空间和正对"超平面"法向量方向的子空间.

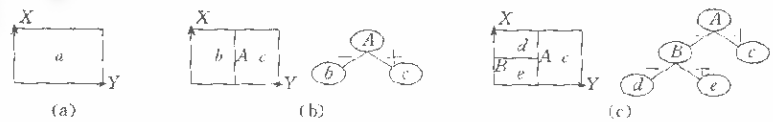

Fig. 1 The construction of BSP tree in 2D
图 1 二维空间BSP树的分割及其构造过程

BSP 树结构十分适合静态几何场景的交互式显示,因为 BSP 树的构造可以作为一个预处理过程,并且由于场景 BSP 树的构造与观察视点无关,从任意视点出发的场景绘制都不需要重新构造新的 BSP 树,因此可以在线性时间内完成整个绘制.假定已经得到一棵表示一个 3D 多边形场景的 BSP 树和当前观察视点 e,就能通过按一定次序遍历 BSP 树绘制.先绘制与 e 所在子空间不同的那个子空间内的对象,接着绘制根节点内的对象,最后绘制距离 e 近的那个子空间对象;在每个子空间内递归重复上述步骤,直到 BSP 树遍历结束.BSP 树的遍历方式与观察视点相关,而构造过程与视点无关.因此,当视点 e 改变之后,BSP 树不需要重新构造.如果场景是动态的,存在模型的增减,则原先生成的 BSP 树就无效了,需要重新构造 BSP 树.

为了减少显示多边形数目而应用多细节层次模型的场景实际上是一种动态场景.随着视点的改变,应用更换相应的 LoD 模型,从而改变了整个场景,就需要重建 BSP 树.如果系统应用静态 LoD 模型,预先计算物体若干个层次的模型,我们就可以同时生成各自的 BSP 树.当场景使用某种 LoD 模型时就调用相应的 BSP 树绘制,Wiley 在文献[7]中利用这种方法处理地形数据和体绘制,取得比较好的结果.

相比之下,视点相关动态 LoD 模型具有更出色的适应性和绘制质量[2,8,9].与静态 LoD 模型不同,由于模型根据视点实时生成,并且整个模型由多个相异层次组合而成,三角形之间的组合种类成千上万,根本无法枚举所有可能的 LoD 模型,也不可能预先构造所有的 BSP 树结构,因此,一旦场景视点改变,我们就需要重新建立整个场景的 BSP 树结构,而 BSP 树构造过程的建立非常耗时,每次重建必定引起整个绘制过程的效率下降.为了解决这个问题,我们提出了模型的多分辨率 BSP 树,以适用于动态 LoD.

1.2 分辨率 BSP 树产生背景及粘合多分辨率模型(GMM)介绍

先简要介绍场景中动态 LoD 模型的生成过程.在原始网格模型简化时,记录删除的三角形数据,保存在相关的顶点信息中,建立一个顶点层次结构.显示时根据视点在顶点层次结构上作顶点分裂操作或者顶点合并操作,同时增加或者移去相应的三角形[2,10].我们可以设定约束条件:(1)简化后产生的新顶点和三角形都重新标识,与原来的三角形信息区分开;(2)顶点分裂时所有的邻接顶点和三角形必须与该顶点生成时完全一致[8],这样生成的顶点层次结构包含了一个固定的三角形集合.

事实上 C. Cignoni 提出的粘合多分辨率模型(glued multi resolution model,简称 GMM)[9]就符合这样的条件.GMM 撇开了"层次"的概念,引入了"超三角剖分"的思想,把在三维空间内具有邻接关系的、出现在层次树的不同层上(即具有不同分辨率的网格)的三角形编码在一起,如图 2 所示.GMM 的适用范围很广,适合于采用任何简化或精化算法生成的网格.

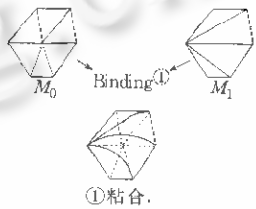

Fig. 2 Generation of GMM
图2 GMM模型的生成

GMM 生成算法在简化过程中对所有变化的三角形重新编号,简化结束后产生了一个三角形集合,包含各细节层次网格的所有三角形:

步骤 1. 读入初始网格 M,并构造初始网格中的顶点、三角形和 GMM;

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,将新生成的顶点、三角形加入 GMM 中,同时修改 GMM 中相关的三角形的出生误差和死亡误差;

步骤 4. 重排误差代价队列,误差代价队列非空,转步骤 3;

步骤 5. 输出多分辨率网格的 GMM 模型.

动态多细节层次模型的生成过程可以看成是每次从这个特殊的三角形集合中选择符合视点条件误差的、数目精确的以及边界相互吻合的一个三角形子集,最后得到的三角形集合就是所求的网格模型.

我们已经开发了一个粘合多分辨率模型的编辑系统,为了解决粘合多分辨率模型的绘制问题,我们将构造用于粘合多分辨率模型的 BSP 树结构,称其为 multi-resolution binary space partitioning tree,简称 MRBSP.MRBSP 树同样适用其他符合约束条件的动态 LoD 模型.

1.3 MRBSP 树的结构及生成

这里,我们使用 BSP 辅助绘制 GMM 模型.我们希望对同一个模型,当其分辨率变化时,系统不随三角形对象的改变而重新构造 BSP 树,而只需利用预先生成的 MRBSP 来完成绘制工作.

首先,我们在原来的 BSP 树节点数据结构基础上增加一个域 faceindex,表示该节点所包含的多边形对象由哪个三角形分割产生.在构造该模型的 MRBSP 树结构时,与传统 BSP 树构造方式不同,我们把整个三角形集合看做是一个完整的网格构造 MRBSP 树结构,并且在每个节点内记录相关的三角形信息 faceindex.下面列出的是 MRBSP 树节点内的多边形对象数据结构.

```
typedef struct faceTag {
    COLOR     color;          //多边形颜色
    VERTEX    *vhead;         //多边形顶点链
    PLANE     plane;          //平面方程
    int       faceindex;      //相关三角形信息
```

```
    struct      faceTag *fnext;                    //下一个多边形结构
} FACE;         //MRBSP树节点内的多边形对象描述信息
```

这样,我们已经把GMM中所有细节层次将用到的三角形用一个MRBSP树结构在模型绘制之前建立起来,表示一个整体.普通的BSP只建立在3维几何空间(x,y,z)内,我们可以把MRBSP树看成是建立在包含离散时间坐标t的一个四维空间(x,y,z,t)内,如果我们假设传统的BSP树结构具有形式$bspf$,那么MRBSP树可以表示成

$$\begin{aligned} &mrbspf(t)=\delta_1(t)bspf_1+\delta_2(t)bspf_2+\ldots+\delta_{m-1}(t)bspf_{m-1}+\delta_m(t)bspfm,\\ &mrbspf(t)=\sum_{i=1}^{m}\delta_i(t)bspf_i, \end{aligned} \tag{1}$$

其中$t=1,2,\ldots,m$,并且有

$$\delta_j(t)=\begin{cases}1 & t=j;\\ 0 & t\neq j,\end{cases} \tag{2}$$

其中m表示该模型所有可能的粘合细节层次模型个数,参数t表示分辨率参数,因为最后生成的多分辨率模型不具有统一的细节层次,所以也可能用一个单一的参数来表示,这里,式(1)只用来说明MRBSP的形式意义,并没有严格的数学意义.

作为一种统一的表示方式,当场景应用了新的细节模型时,就可以立即根据场景分辨率分布系数t得到当前的BSP树$mrbspf(t)=bsp_t$,而不必重新生成新的BSP树.

1.4 多分辨率BSP树的遍历

从式(1)我们知道,当前误差范围的场景BSP树可以用$mrbsp(t)$准确地表示,绘制时系统应用该MRBSP树进行隐藏面消除.MRBSP树的遍历过程与传统的BSP树遍历方法类似,不同之处在于MRBSP并不对所有节点上的多边形对象都作处理,而只绘制满足当前视点条件和分辨率系数为t的多边形.实验中我们仍然使用如下递归方法实现MRBSP树的遍历.

步骤1.如果达到收敛条件或者MRBSP树为空,则转步骤6;

步骤2.判断视点与当前节点分割平面的相对位置;

步骤3.遍历表示与视点相异子空间的MRBSP子树,转步骤1;

步骤4.判断分辨率层次,绘制合适的当前节点;

步骤5.遍历表示包含视点位置子空间的MRBSP子树,转步骤1;

步骤6.该遍历结束.

2 MRBSP树的局限和改进

2.1 MRBSP树的局限性

第1节介绍了一种关于动态多细节层次模型的统一BSP结构——MRBSP树,根据这种方式,我们可以给整个动态场景建立一个单一的BSP树结构,不论场景中观察者的视点如何改变,模型使用了什么样的网格细节层次,应用都不需要重新建立新的BSP树.

虽然MRBSP树结构具有上述稳定的特点,但是这种树结构也存在明显的局限性.MRBSP通过对GMM统一进行空间分割,而GMM三角形数目随着简化的程度而加大,一般为原始网格模型三角形数目的3~5倍,所以对场景复杂度的影响并不大.然而,因为我们选择模型本身包含的三角形作为分割平面,这些不同层次的三角形相互切割产生了大量的多边形碎片,使生成的MRBSP

树异常庞大.例如,在简化过程中,如图3所示,同一局部区域内前后被两个层次的网格覆盖,这两个网格之间必然存在相互交割的现象,而每次这样的简化操作都会带来不同层次之间的重叠,所以最后使生成的三角形的碎片随着网格三角形的数目呈非线性增长,不仅降低了MRBSP树的构造效率,也影响了场景绘制时间.

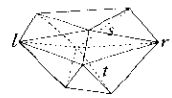

Fig. 3 Different levels of detail overlaping in the same region generates fragments

图3 同一区域两个细节层次的网格相互叠加分割,产生许多碎片

2.2 包围盒与MRBSP树

在场景BSP树的构造过程中产生了许多独立的子空间,每个子空间由一个子BSP树结构独立地表示.实际上,场景中的模型之间并不都存在很强的空间相关性,模型之间还存在一定的间隔,因此我们可以首先把场景内的模型根据空间位置相互隔开,使得模型之间互不分割,就可以在MRBSP树的构造过程中减少三角形碎片的产生.

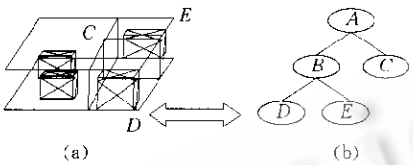

Fig. 4 Several separate objects in the same scene are divided into independent scenes by the subdivision surfaces and sub MRBSP is constructed

图4 选择合适的分割平面把场景中相互独立的物体隔开,分别构造了MRBSP,得到场景MRBSP

许多规则场景,例如建筑物室内结构和2D场数据等,可以方便地使用类似四叉树或者八叉树的规则分割方式进行场景分割.分割后产生的每个规则子空间分别用单独的MRBSP树结构表示空间内物体的动态LoD模型.该场景MRBSP的示意图如图4所示.同时,我们也可以根据需要把单个模型进一步分割,直到满足一定的条件使得生成的MRBSP子树不会非常庞大为止.

2.3 MRBSP树的分割及绘制

我们开发的多分辨率模型编辑系统可以按照用户的需要构造相应分辨率的网格模型,其中主要的用户参数是网格误差值E_m,该参数表示当前网格与原始网格之间的逼近程度.例如,下面图6中的人头模型,其原始模型的E_m为0,最简模型的E_m为30000,理论上该粘合多分辨率模型编辑系统能够产生E_m介于其中的任意近似网格.为了加快MRBSP树的生成和绘制,我们根据网格模型的误差范围分割粘合多分辨率三角形集合,每个子集合各自生成一个子MRBSP树,绘制时根据当前模型的误差值选择相应的子MRBSP树进行遍历,如图5所示.

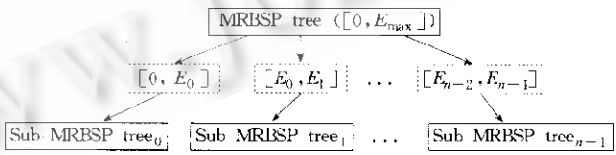

Fig. 5 Sub-Division of MRBSP tree

图5 MRBSP树的分割

GMM中每个三角形都标注了所处的误差范围[*birth*,*death*).所有三角形误差范围的合并为多分辨率模型的误差选择范围$[0,E_{max}]$.下面给出带分割的MRBSP树构造算法:

步骤1.将$[0,E_{max}]$指数*分割为m个子集$E[i]$,并设相应三角形集合$TS[i]=\varnothing$;

步骤2.对每个三角形$T\in GMM$,如果$Error(T)\cap E[i]\neq\varnothing$,则$TS[i]=TS[i]\cup\{T\}$;

* 网格简化生成的模型与原始模型之间的误差成倍上升,因此为了取得较好的效果,我们使用指数增长分割误差范围,我们取底数的经验值为10.

步骤 3. 对每个三角形集合 $TS[i]$, $i\in[1..m]$, 进行步骤 4~8 操作, 构造相应的 MRBSP 树 MRBSP[i];

步骤 4. 如果 $TS[i]$非空, 设 $TS=TS[i]$;

步骤 5. 如果 TS 非空, 从 TS 选择一个空间分割平面 P;

步骤 6. 对每个三角形 $F\in TS[i]$, 计算 F 和 P 之间的相对位置关系, 如果重合, 把 F 加入当前节点; 如果 F 与 P 不相交, 根据位置分别将 F 加入当前节点的前节点或者后节点.

步骤 7. 如果 F 与 P 相交, 先用 P 分割 F 为 $F1$ 和 $F2$ 两个三角形, 并且为其记录 GMM 三角形序号, 然后再把 $F1$ 和 $F2$ 分别放入相应的前后节点;

步骤 8. 分别设 TS 为当前节点的前后节点, 转步骤 5 递归执行步骤 5~8 操作; 完成后转步骤 3;

步骤 9. 输出 MRBSP 树结构.

我们首先将$[0,E_{max}]$指数分割为有限个子集, 接着通过计算每个三角形的误差范围与误差子集的交集非空来判断该三角形是否属于这个误差子集, 实际上, 一个三角形可以属于不同的误差子集. 然后以误差子集为单位分别构造相应的 MRBSP 树, 因为我们通过一定的非线性规则划分$[0,E_{max}]$, 所以最后生成的各个 MRBSP 树之间的大小比较平衡, 使得最终绘制时间比较均匀.

最后, 当应用选择了模型的某个细节层次后, 通过该细节模型的误差系数和所属的误差子集来选择相应的 MRBSP 树, 然后绘制这个细节层次的模型. 这种优化方法能大大加快粘合多分辨率模型 MRBSP 树的构造和绘制. 参见表 1, 实验中我们取 MRBSP 树的分割数为 4.

Table 1 Statistics of rendering each LoD model of MRBSP tree with different divisions

表 1 应用不同分割的 MRBSP 树的细节层次模型绘制的统计数据

Model①	Partition number of MRBSP②	Construction time (ms)③	Number of nodes in MRBSP④	Construction time of different LoD levels (ms)/number of triangles⑤				
				Level⑥ 1 (180)	Level 2 (120)	Level 3 (60)	Level 4 (25)	Level 5 (10)
Hemisphere⑦	1	13 280	26 836	90	90	85	75	65
	2	13 930	20 306	70	70	55	60	≈0
	3	13 429	14 464	60	50	15	≈0	≈0
	4	10 636	6 930	25	20	6	≈0	≈0

①模型, ②MRBSP 树分割数, ③构造时间(毫秒), ④MRBSP 节点数,
⑤不同细节层次的三角形数目和绘制时间(毫秒), ⑥层次, ⑦球冠.

3 实验结果

因为 MRBSP 树结构已应用于我们开发的多分辨率模型编辑环境中, 表 2 和表 3 列出了应用 MRBSP 树绘制多边形模型的基本实验数据, 表 2 中普通 BSP 实验模型为原始模型, 而 MRBSP 树对包含各个细节层次模型的 GMM 三角形集合进行处理, 比较时 MRBSP 选择误差为 0 的模型绘制, 即原始模型. 从中可以看到, 在多分辨率模型场景中, 虽然构造 MRBSP 树比构造普通的 BSP 树要花费更多的时间, 单独绘制时间也更多, 但是因为实时绘制时 MRBSP 树不需要重新构造, 而普通 BSP 树因为模型的改变需要重新构造, 所以 MRBSP 树的最终处理时间要大大少于应用普通的 BSP 树.

Table 2 Construction of MRBSP tree and rendering result compared with general BSP tree

表 2 MRBSP 树的构造及绘制结果及其与 BSP 树的比较

BSP tree①	Model②	Number of triangles③	Construction time (ms)④	Rendering time (ms)⑤	Rendering in real-time (ms)⑥
General BSP tree (orginal model)⑦	Molecule⑧	7 344	336 343	280	336 623
	Sphere⑨	1 680	1 012	30	1 042
	Human head⑩	1 355	7 711	50	7 760
	Hemisphere⑪	180	40	5	45
MRBSP tree (GMM)⑫	Molecule	31 658	995 311	741	741
	Sphere	7 225	681 831	80	80
	Human head	5 779	137 437	180	180
	hemisphere	736	10 625	31	31

①BSP 树,②模型,③模型三角形数目,④构造时间(毫秒),⑤绘制时间(毫秒),⑥实时绘制时间(毫秒),⑦普通 BSP 树(原始模型),⑧分子模型,⑨球体模型,⑩人头模型,⑪球冠模型,⑫MRBSP 树(GMM).

图 6 中原始模型有 1 355 个三角形,如图 6(a)所示,直接绘制所有三角形会产生形状错乱,需要进行消隐处理. 图 6(b)~(f)为应用预先生成的 MRBSP 树快速绘制的 5 个分辨率模型,各个模型的三角形数目分别为 1 355、875、412、173 和 71. 粘合模型共有 5 779 个三角形. 其他详细数据见表 2 和表 3.

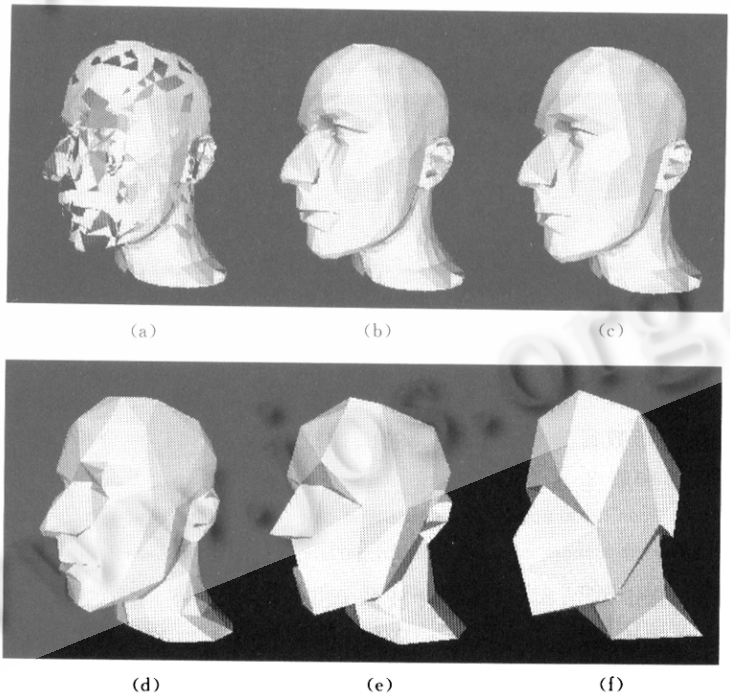

Fig. 6 Examples of different multi-resolution models—human head generated by multi-resolution edit system

图 6 Multi-Resolution 编辑系统生成的具有不同分辨率的人头模型

作为一个预处理过程,MRBSP 的构造一般比普通 BSP 树慢几十倍或上百倍,但是表 2 中分子模型显示,当模型三角形数目和表面复杂度增加时两者的构造时间会很接近. 对不同复杂度和不同类型的模型应用 MRBSP 的结果并不相同,只要我们选择适当的误差分割数目和切割平面就可以获得较为理想的效果.

Table 3 Time performance of rendering different multi-resolution models with MRBSP tree
表 3 MRBSP 树绘制不同分辨率模型的时间统计

Model①	Parameter②	Level③(b)	Level (c)	Level (d)	Level (e)	Level (f)
Molecule④	Number of triangles⑤	7 344	3 938	1 822	800	364
	Rendering time (ms)⑥	741	540	330	150	120
Sphere⑦	Number of triangles	1 680	840	422	192	84
	Rendering time (ms)	80	68	50	27	20
Human head⑧	Number of triangles	1 355	875	412	173	71
	Rendering time (ms)	180	160	120	70	20
Hemisphere⑨	Number of triangles	180	112	60	25	10
	Rendering time (ms)	31	30	15	≈0	≈0

①模型,②参数,③层次,④分子,⑤三角形数目,⑥绘制时间(毫秒),⑦球体,⑧人头,⑨球冠.

图 6 和图 7 分别是用人头模型和分子模型进行实验得到的绘制结果.实验环境为:联想主板,Intel Pentium-2,主频 350M,128M 内存,Windows NT 4.0 操作系统.

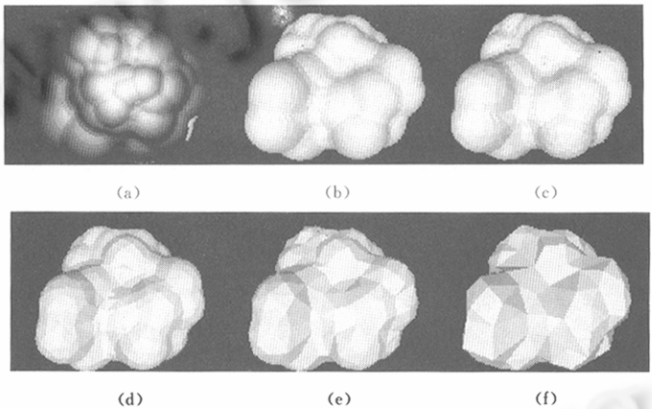

Fig. 7 Examples of different multi-resolution models——molecule generated by multi-resolution edit system
图 7 Multi-Resolution 编辑系统生成的具有不同分辨率的分子模型

原始模型有 7 344 个三角形,图 7(b)~(f)为应用预先生成的 MRBSP 树快速绘制的 5 个分辨率模型,各个模型的三角形数目分别为 7330、3938、1822、800 和 364.粘合模型共有 31 658 个三角形,其他详细数据见表 2 和表 3.

4 结 论

本文扩展了基本的 BSP 树结构,提出一种新的 BSP 树结构——MRBSP,使它可以应用在动态多细节层次的绘制中实现隐藏面消除功能.MRBSP 能够提供多细节层次场景多边形正确的前后次序,无需因为视点的改变而重新构造场景的 BSP 树结构.由于 BSP 树的固有特性,MRBSP 树的节点数目因为三角形碎片的产生而增加,降低了构造效率.据此我们作了一定的优化,首先将场景中的分离物体分开,并根据网格模型的误差范围分割粘合多分辨率三角形集合,为每个子集合各自生成一个子 MRBSP 树,绘制时根据当前模型的误差值选择相应的子 MRBSP 树进行遍历,取得较理想的效果.另外,MRBSP 树的构造是一个预处理过程,构造完成后,可以达到实时绘制的要求.

References:

[1] Aliaga, D., Cohen, J., Wilson, A., *et al*. A framework for the real-time walkthrough of massive models. Technical

Report, UNC TR# 98-013, 1998.

[2] Hoppe, H. View-Dependent refinement of progressive meshes. Proceedings of the SIGGRAPH'97: Computer Graphics. 1997. 189～198.

[3] Pan Zhi-geng, Ma Xiao-hu, Shi Jiao-ying. Summarization of automatic generation of LoD models in virtual reality. Transactions of Image and Graphics of China, 1998,3(9):754～759 (in Chinese).

[4] Chysemthou, Y., Slater, M. Computing dynamic changes to BSP trees. Computer Graphics Forum (Eurographics'92 Proceedings), 1992,11(3):321～332.

[5] Thibault, W., Naylor, B. Set operations on polyhedra using BSP trees. Computer Graphics (Proceedings of the SIGGRAPH'87), 1987,26(3):153～162.

[6] Paterson, M., Yao, F. Binary partitions with applications to hidden surface removal and solid modeling. In: Proceedings of the 5th Annual Symposium on Computational Geometry. 1989.

[7] Wiley, C., Szygenda, S., Fussell, D., *et al*. Multi-Resolution BSP trees applied to terrain. Transparency, and General Objects. http://www.arlut.utexas.edu/~atc/doc/gi97wiley/ GI97.camera.html.

[8] Xia J., Varshney, A. Dynamic view-dependent simplification for polygonal models. In: Proceedings of the Visualization'96. IEEE, 1996. 327～334.

[9] Cignoni, P., Montani, C., Rocchini, C., *et al*. Resolution modeling. Technical Report, C97-02. CNUCE-C. N. R., Pisa, Italy, 1997.

[10] Tao Zhi-liang, Pan Zhi-geng, Shi Jiao-ying. Generation of dynamic LoD models in complex scene. Transactions of Image and Graphics of China, 1998,3(12):1032～1035 (in Chinese).

附中文参考文献：

[3] 潘志庚,马小虎,石教英. 虚拟现实中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.

[10] 陶志良,潘志庚,石教英. 复杂场景中动态简化层次的构造. 中国图象图形学报,1998,3(12):1032～1035

Generation and Applications of a Multi-Resolution BSP Tree

TAO Zhi-liang, CHENG Chi-yi, PAN Zhi-geng, SHI Jiao-ying

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

E-mail: chengcy@cad.zju.edu.cn

http://www.cad.zju.edu.cn

Received April 12, 1999; accepted September 29, 1999

Abstract: Computer aided design applications and scientific visualization often need user-steered interactive display of very complex environments. Instead of rendering original complex models directly, multiple levels of detail (LoD) models are widely applied to graphics system at run time to achieve a great speedup. In this paper, a new BSP tree framework is presented that can incorporate dynamic LoD models. This framework is designed for a multi-resolution model editing system. The tree construction algorithm and traversal routine for the multi-resolution BSP tree are discussed in detail.

Key words: BSP tree; multi-resolution modeling; real-time rendering; mesh simplification