第 34 卷 第 5 期 计 算 机 工 程 2008 年 3 月
Vol.34 No.5 Computer Engineering March 2008
·工程应用技术与实现· 文章编号：1000—3428(2008)05—0231—03 文献标识码：A 中图分类号：TP391.72

基于面向对象八叉树的虚拟漫游碰撞检测

王功明[1,2]，郭新宇[1]，赵春江[1]，王纪华[1]
(1. 国家农业信息化工程技术研究中心，北京 100097；2. 中国科学院计算技术研究所，北京 100080)

摘 要：借助面向对象概念，根据层数和叶结点链表个数上限来构建存储场景物体信息的高效八叉树。每个叶结点指向记录对应空间区域内物体信息的链表，每个空间物体信息链表按照其对象大小降序排列。在逐步求精阶段用降序包围球遍历链表进行检测，然后根据凸多面体剖分算法，使用主从 MPI 模式并行处理以实现精确碰撞检测。该方法利用物体空间位置关系进行碰撞检测，毋需存储大量空间物体三角面片。在基于粒子系统的土壤可视化漫游中的运行结果表明，该方法精度高、实时性好，具有一定的研究和应用价值。

关键词：面向对象；八叉树；碰撞检测；包围球；凸多面体剖分；粒子系统

Virtual Walkthrough Collision Detection Based on Object-oriented Octree

WANG Gong-ming[1,2], GUO Xin-yu[1], ZHAO Chun-jiang[1], WANG Ji-hua[1]
(1. National Engineering Research Center for Information Technology in Agriculture, Beijing 100097;
2. Institute of Computing Technology, Chinese Aademy of Sciences, Beijing 100080)

【Abstract】By virtue of object-oriented concept, the efficient octree that stores object's information in scene is constructed based on number of layers and amount of leaf node link tables. Every leaf node pointed link table that records object's information is sorted by its size in corresponding special area. The descent order encircle sphere algorithm is used to detect in seeking exactness phase. Then the principal and subordinate Message Passing Interface(MPI) pattern are parallel applied to realize accurate collision detection according to dividing a convex polyhedron to tetrahedrons algorithm. Compared with the traditional octree algorithm, this method does collision detection by using object's spatial position relation instead of storing lots of spatial object's triangle faces. It is applied in the soil visual walkthrough based on particle system. The result of this application proves that it has high precision and good real-time performance. It is researchful and can be applied to some extent.

【Key words】object-oriented; octree; collision detection; encircle sphere; dividing a convex polyhedron to tetrahedrons; particle system

虚拟现实是近年来研究热点，它利用专门软硬件设备，使人随心所欲在场景漫游，产生身临其境的感觉[1]。在构造三维交互漫游系统中，碰撞检测必不可少，否则 2 个或多个物体同时占有同一空间区域，违反现实世界观察事实[2]，破坏虚拟环境浸入性，降低漫游真实感。八叉树基于空间剖分存储场景物体信息，碰撞检测时可快速剔除场景无关物体，提高效率。层次包围体树是比较简单的空间结构表示方法，能进行粗略的碰撞检测[2]。基于凸体剖分的碰撞检测将凸体分解为多个简单四面体，然后利用MPI并行处理进行多个四面体之间碰撞检测，既降低复杂度，又提高速度，同时坚实的凸多面体剖分算法理论又保证较高精度[3]。本文综合上述方法和面向对象概念，构成面向对象八叉树进行碰撞检测，并在基于粒子系统的土壤可视化漫游系统中应用。

1 面向对象八叉树基本数据结构

普通八叉树和面向对象八叉树都用空间分解法建树并用叶结点保存信息。区别是前者保存物体外表面三角面片，存储量大；后者保存对象信息[4]，用几何求交来碰撞检测，存储量小、精度高。

1.1 基本结点

图 1 为面向对象八叉树基本结构，结点 R 为根结点，结点 M 为非叶结点，R 和 M 间虚线表示 M 是 R 子孙，结点 R 可能还有其他子孙，为突出结点 M 故省略。

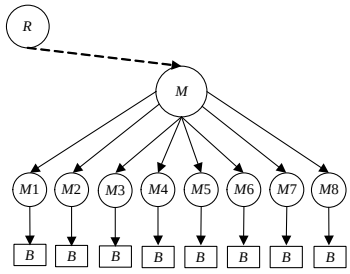

图 1 八叉树基本结构

图 1 中结点 R 为整个空间区域，结点 M 为还可继续分解子区域，该区域被分解成 8 个不再分解子区域，用 $M1 \sim M8$ 表示，B 为叶结点链表，记录区域内物体信息。

所以，基本结点如下：

(1)非叶结点：表示能继续分解的空间网格，如图 1 结点 R 和 M ，数据结构为

标志	网格大小	网格中心	8个子结点指针

基金项目：国家“十一五”科技支撑计划基金资助项目(2006BAD10A07)；北京市优秀人才培养计划基金资助项目

作者简介：王功明(1981－)，男，高级工程师、在职博士研究生，主研方向：计算机图形学，科学计算可视化；郭新宇，副研究员、博士；赵春江、王纪华，研究员、博士

收稿日期：2007-03-28 **E-mail：**zhaocj@nercita.org.cn

其中，标志字段表示该区域是否可继续分解。

(2)叶结点链表头结点：表示不能继续分解的空间网格，如图 1 结点 $M1 \sim M8$，数据结构为

父结点指针	叶结点链表首地址	叶结点个数

其中，叶结点链表首地址指向该空间网格中的物体链表。

(3)叶结点链表内结点：表示空间物体，如图 1 链表 B 中结点，数据结构为

中心	半径	类型	下一物体指针

为减少无用碰撞检测，链表中物体按半径降序排列。

1.2 面向对象八叉树层数控制

每个结点表示一空间区域，区域满足约束条件就不分解，结点指向该区域物体信息链表；否则区域一分为八，该结点产生 8 个子结点，每个结点继续上述过程，直至约束条件为真。常用约束条件为：(1)层数限为定值[5]；(2)层数不限，限定叶结点链表中结点个数[4,6]。上述 2 种方法都有缺陷，前者易于实现，但物体分布不均时，视点在物体较多区域碰撞检测耗时大。后者可弥补此缺陷，但实现困难。

因此，综合方法(1)和方法(2)形成下述层数控制方案：(1)视漫游场景为一边长 M 空间立方体；(2)不计物体个数和空间分布时八叉树期望层数为 T，八叉树自上而下分 T 层，场景随之分解，每个叶结点链表表示边长 $M/2^T$ 空间立方体内物体分布；(3)叶结点链表最多允许 P 个结点；(4)逐个检查每个叶结点链表，结点个数超过 P 则分裂，链表为空则删除。

对于给定场景区域，M 易知，确定 T 和 P 就可构建高效八叉树。碰撞检测步骤为：(1)自上而下查找物体所在空间结点；(2)与该空间网格内物体逐个碰撞检测。这 2 步消耗时间之和为碰撞检测总时间，步骤(1)时间复杂度为 $O(T)$，步骤(2)时间复杂度为 $O(P/2)$。T 较大增加空间分支存储量，P 较大增加链表元素比较量，所以两者尽量相等。

设 $T = P/2 = Q$，N 为场景物体总数，T 和 P 必须是满足 $8^Q(2Q) > N$ 的最小整数。

1.3 面向对象八叉树生成

由 1.2 节可知，空间物体分布不一定均匀，稠密区域叶结点链表长度可能超过 P，结点要分解；稀疏区域可能连续好几个区域都无物体，结点要合并。

1.3.1 八叉树结点分解与合并

叶结点链表头结点中的父结点指针在结点分裂时确定父结点，叶结点个数判断是否分裂，初值为 1，插入前先加 1，检验是否分裂，若分裂则分裂，否则按序插入。

如图 2 所示，分裂时先得到 M 父结点指针，然后生成非叶结点 N 作为后继插入，由空间位置分解法则确定 N 个子结点位置并插入，最后释放 M。结点合并时从根结点向下遍历，若结点 $M8$ 个分支结点均空，则删除 M。

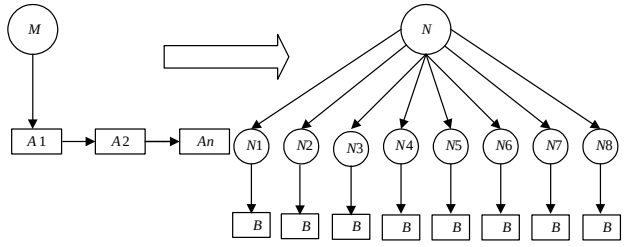

图 2 八叉树结点的分解

1.3.2 八叉树生成

(1)生成由非叶结点构成的 Q 层八叉树。

(2)遍历场景每个物体，由其中心位置 (X,Y,Z)，从根结点向下遍历至 (X,Y,Z) 所在结点，叶结点链表头指针 M 指向该结点，非叶结点指针 R 指向其父结点。

(3)若 M 非空，转(4)；否则生成叶结点链表头结点，M 指向该结点，然后生成表示该物体的结点 T，设置各结点属性，转(6)。

(4) M 所指结点中叶节点个数增加 1，超过 $2Q$ 转(5)，否则生成表示该物体的结点 T，按降序插入叶结点链表，转(6)。

(5)结点分裂，生成表示该空间物体结点 T，把 T 插入适当位置。

(6)按照(2)到(5)逐个处理场景其他物体，直至所有物体处理完为止。

(7)从根结点向下进行结点合并。

任一非叶结点有 8 个分支，表示空间细分后 8 个方向，它们指向 1.1 节(1)和(2) 2 种结点。若为这 2 种结点创建不同分支指针，首先增加空间复杂度；其次涉及结点类型判断，不利于八叉树生成、删除、检索等操作。所以，用 void *指针表示分支结点，该指针没有固定指向类型，通过强制类型转换能指向不同数据类型。

2 基于面向对象八叉树虚拟漫游碰撞检测

2.1 碰撞检测阶段划分

面向对象八叉树碰撞检测阶段为：(1)初步检测阶段，自上而下遍历，检测目标空间有无物体，无则不碰撞，否则转下一阶段；(2)详细检测阶段，分为 2 个层次：逐步求精层和精确求交层，前者用降序排列包围球和目标物体作粗略碰撞检测，若目标物体不在任一包围球内则无碰撞，否则用基于凸多面体剖分的并行精确碰撞检测算法进行碰撞检测。上述阶段划分如图 3，每一阶段都排除大量非碰撞情况，减轻后继阶段处理负担，提高实时性。

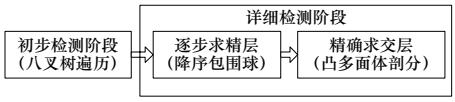

图 3 碰撞检测阶段划分

2.2 基于八叉树初步碰撞检测

先确定目标物体所处空间区域，若物体比较大，可能要占据多个空间区域。分别检测这些空间区域是否有物体。关键代码为

```
if (CheckArea->GetStatus())
//查找空间区域粒子链表
{
if (x<=area_x&&y>=area_y&&z>=area_z)//前方左上分支
CheckList=(ParticleList*)CheckArea->m_pOctreeNodes[TOP_LEFT_FRONT];
else
…
if (CheckList==NULL)//无粒子
return false;
…}
```

2.3 基于降序排列包围球逐步求精碰撞检测

某一空间区域有多个物体，物体越大，和目标物体发生碰撞几率越高，所以物体按大小降序排列，然后用包围球作粗略碰撞检测。

关键代码为

```
while(CheckNode)
{//包围球
if(sqrt((x-CheckNode->GetPosition().x)*(x-CheckNode->GetPosit
```

```
ion().x)+(y-CheckNode->GetPosition().y)*(y-CheckNode->GetPosition
().y)+(z-CheckNode->GetPosition().z)*(z-CheckNode->GetPosition().z
))<(CheckNode->GetRadius()+5.0))
    return true;//内部
    CheckNode=CheckNode->next;
    }
    return false;//外部
```

2.4 基于凸多面体剖分并行精确碰撞检测

由文献[3]知，从任一凸多面体某一凸顶点出发，使用Delaunay三角剖分[7]，经过若干次递归可把该凸多面体分解成一系列四面体。用该方法把目标物体和场景待检测物体分解成一系列四面体，然后用文献[8]中方法计算目标和待检测物体之间距离，为加快速度，用主从MPI模式[3]并行处理。关键代码为

```
//判断顶点是否在由三顶点构成三角形内
bool InsideTriangle(Vcetor vIntersection,Vector Triangle[3])
{
const double MATCH_FACTOR=0.99;//误差
double Angle=0.0;//初始化角度
Vector vA,vB;//临时矢量
for(int i=O;i<3;i++)
{
vA=Triangle[i}-vIntersection;
//下一顶点与交点之间矢量
vB=Triangle[(i+l)%3]-vIntersection;
//两矢量之间夹角，循环中累加
Angle+=AngleBetweenVector(vA,vB);
   }
   //角度>=360 度，点位于三角形内
if(Angle>=(MATCH_FACTOR*2*PI)
return true;
return false:
}
```

3 应用实例

3.1 基于粒子系统土壤可视化漫游

土壤是植物生长基础，构建土壤可视化模型，对于研究植物生长发育、获取土壤结构参数等具有重要意义[9]。构成土壤主体的土壤颗粒具有分形特征，和粒子系统中粒子相似，大量粒子无规则运动为稳态运动，可模拟土壤。由文献[10]可知，土壤结构体形状分 10 种，但屑粒状粒子可看作多种其他小粒子构成的团聚体，故粒子基本形状有 9 种。

根据表 1 的设置，图 4 表示由这些粒子生成的三维可视化土壤。为了开展基于土壤的生化反应可视化，需深入内部可视化漫游，以便发现问题，及时修正模型。本系统场景有大量粒子，为提高漫游真实感和浸入性要碰撞检测。用八叉树存储粒子，粒子为凸体，诸阶段碰撞检测，精度和效率上能满足要求，流程如图 5。

表 1 土壤粒子形状设置

序号	形状	描述	实现形状
0	片状	表面平滑	扁平长方体
1	鳞片状	表面弯曲	扁平椭球上半部
2	棱柱状	边角明显无圆头	上下底面均为正六边形的柱体
3	柱状	边角较明显无圆头	细高长方体
4	棱块状	边角明显多面体状	正 20 面体
5	团块状	边角浑圆	正 8 面体
6	核状	边角尖锋紧实少孔	扁平椭球体
7	粒状	浑圆少孔	球体
8	团粒状	浑圆多孔	球体及周围均匀分布相切 8 个球

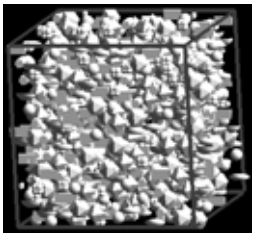

图 4 三维土壤

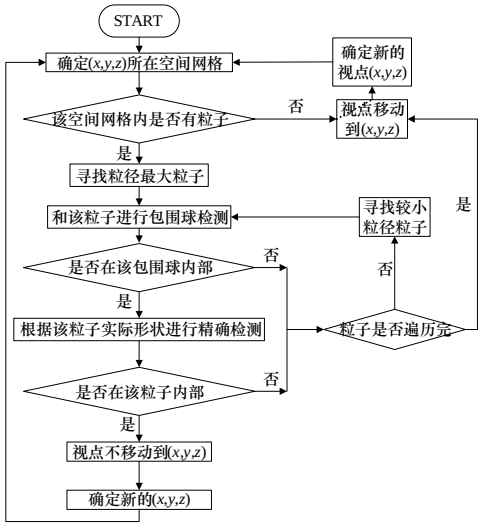

图 5 基于八叉树碰撞检测流程

3.2 漫游实现

采用文中碰撞检测方法，建立八叉树，基于第三人称视角模式，在普通 PC 机上(处理器 P4 2.4 GHz，显卡 NVIDIA GFORCE FX 5200，内存 512 MB)利用标准 VC+OPENGL 架构完成可视化漫游碰撞检测，真实感和流畅性都比较好。如图 6 所示，左边居中小方块表示第三人称视角物体，它和左边居中团粒状粒子发生碰撞，系统弹出对话框予以提示。

图 6 漫游中的碰撞检测

场景粒子数量不同时，系统运行帧频对比如表 2，由此可知，本系统实时性好，能满足一般实际应用需要。

表 2 漫游帧频对比

粒子数	帧频/(帧·s^{-1})
500	25.0
800	14.6
1 000	9.2
1 500	4.5

4 结束语

本文构建面向对象八叉树储存虚拟场景中的物体信息，并且综合八叉树、降序排列包围球、凸多面体剖分等方法逐阶段碰撞检测，并在基于粒子系统的可视化漫游中得到应用，精度和效率都比较好。该方法在物体信息可获取、位置固定的大场景漫游中具有一定的研究和应用价值。

（下转第 239 页）