

基于复合层次包围盒的实时碰撞检测研究

朱元峰, 孟 军, 谢光华, 马文娟

(大连理工大学计算机科学与工程系, 辽宁 大连 116024)



摘 要: 针对各种层次包围盒的特点, 提出了基于球状包围盒(Sphere)结构与有向包围盒(OBB)的复合层次包围盒的碰撞检测方法。将复合包围盒树分为 X、Y、Z 层, X 层节点为球状包围盒(Sphere)结构, Y 层为 OBB-Sphere 混合结构, Z 层为 OBB 结构。上层节点间的相交测试采用 Sphere 与 Sphere 或者 OBB 的方式, 以此快速排除不可能相交的物体; 下层节点间的测试采用 OBB 与 OBB 的方式保证精确的判定距离较近的物体的碰撞状态。实验通过与 OBB 算法的性能比较, 证明了复合包围盒能有效地应用于复杂场景的实时碰撞检测。

关键词: 碰撞检测; 层次包围盒; 球形包围盒; 有向包围盒

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X(2008)02-372-06

Research on Real-Time Collision Detection Based on Hybrid Hierarchical Bounding Volume

ZHU Yuan-feng, MENG Jun, XIE Guang-hua, MA Wen-juan

(Department of Computer Science and Engineering, Dalian University of Technology, Dalian 116024, China)

Abstract: Analyzing the performance of various hierarchical bounding volumes (HBVs), a fast algorithm for real-time collision detection (CD) was proposed using a structure of hybrid hierarchical bounding volume (HHBV) based on OBB and sphere BV. The HHBV tree for CD falls into three successive layers in a top-down manner, which are Layer X using sphere as its tree node, Layer Y using OBB-Sphere BV and Layer Z using OBB. The intersection test uses sphere vs. sphere or sphere vs. OBB method for fast overlap test in upper layers, eliminating the possibility of collision between objects far apart; then it accurately determines the contact status among objects in closer proximity by OBB vs. OBB test in lower ones. Experiments were compared with other competitive algorithm (OBB) to analyze the performance. Results show the algorithm is as efficient in precision as OBB, and faster in complex virtual environment.

Key words: collision detection; hierarchical bounding volumes; sphere bounding boxes; OBB

引 言

碰撞检测作为虚拟现实一项重要的技术, 在实现虚拟场景所应提供的沉浸性、交互性和构想性等方面扮演了重要角色, 并再次成为研究热点。碰撞检测的基本任务就是确定两个或多个物体之间是否发生接触或穿透^[1]。

如果对虚拟环境中组成物体的所有 n 个基本几何元素进行两两相交测试, 复杂度为 $O(n^2)$, 其几何复杂性使碰撞检测算法的计算复杂度大大提高^[2]。针对复杂虚拟场景的特点, 已有的经典算法, 例如 BSP 树、K-DOP^[3]、OBB、Sphere、AABB、FDH^[4]等在检测速度和精度上各有侧重, 但是不能很好的权衡两者之间的关系。根据不同的碰撞概率采用不同的包围盒; 根据不同的碰撞检测精度要求采用不同的检测深度^[5], 就能提高整个实时系统的检测效率。把碰撞检测分为预处理阶段和实时检测阶段。在预处理阶段为待检测对象建立具有不同的包围盒模型节点的包围盒树(BVT), 即上层

(X层)用简单的球(Sphere)包围盒, 中间层(Y层)用混合的包围盒结构(OBB-Sphere), 下层(Z层)用有向包围盒(OBB)。相距较远不可能碰撞的对象可以被上层或者中层的Sphere-Sphere检测快速排除掉; 对那些上层或者中间层可能相交的对象, 进一步进行OBB-Sphere包围盒的相交测试; 最后对于物体的紧密性已经很好的可能重叠的包围盒, 则采用精确的OBB包围盒相交测试来确定碰撞状态。

本方法实时检测速度的代价是Sphere与OBB的折中, 具有Sphere包围盒在复杂场景中能迅速排除不可能相交的物体对的特性, 同时又能对需要进行精确碰撞检测的物体对进行准确碰撞检测, 在提高实时检测效率的同时, 使包围盒的紧密性仍能接近OBB, 体现了复合层次包围盒的优势。

1 层次包围盒算法概述

比较常见的包围盒有轴向包围盒AABB(axis-aligned bounding boxes)、包围球(spheres)、有向包围盒OBB(oriented bounding boxes)等等。基于层次表达的刚体的包围盒碰撞检测算法代价可以用公式(1)表示:

$$T = N_v \times C_v + N_p \times C_p + N_u \times C_u \quad (1)$$

其中, T 是碰撞检测的总的代价方程; N_v 是参与重叠测试的包围盒对的数目; C_v 是一对包围盒重叠测试的代价; N_p 是测

收稿日期: 2006-11-06

修回日期: 2007-03-16

作者简介: 朱元峰(1982-), 男, 辽宁大连人, 硕士生, 研究方向为计算机图形图像、虚拟现实技术; 孟军(1964-), 女, 辽宁大连人, 副教授, 研究方向为数据库技术、数据挖掘、虚拟现实技术; 谢光华(1983-), 男, 江西萍乡人, 硕士生, 研究方向为虚拟现实技术; 马文娟(1983-), 女, 山东德州人, 硕士生, 研究方向是虚拟现实技术。

试基本几何元素对的数目; C_p 是测试一对基本几何元素对的代价; N_u 为当对象发生旋转后需更新包围盒数目; C_u 是更新一个包围盒所需的代价。根据代价函数,下面对几种常见包围盒的特点进行简述。

1.1 轴向包围盒

轴向包围盒(AABB)是应用最早的包围盒,构造和相交测试都比较简单,存储空间小,但紧密性较差,会导致层次二叉树增加更多节点,从而进行很多不必要的检测,影响检测效率。AABB包围盒重叠测试比较简单;当物体旋转之后需要对AABB更新,并可能进行包围盒的重新计算。由于建树和更新速度较快,通常用于非刚体的碰撞检测^[6]。

1.2 球形包围盒

球形包围盒(Sphere)用一个球体代表整个对象,重叠测试只要计算球心与另一物体的距离是否大于半径即可;另外在物体运动时,球形BV不需要重建,只要在原球体的基础上加上一个物体平移的向量即可。球形包围盒在明显分离的物体间检测有很高的性能,然而对于一般空间分布不均匀的物体,如细长物体, Sphere的紧密性是最差的。若要提高精确度则需要增加建树的深度,相应地增加了运算量^[7]。

1.3 有向包围盒

有向包围盒(OBB)被定义为包含该对象相对于坐标轴方向的任意最小的正六面体。它的最大特点是方向任意性,因此它可以根据被包围对象的形状特点尽可能紧密地包围对象。但同时它的相交测试变得更复杂,其关键在于寻找最佳方向,并确定在该方向上包围对象的包围盒的最小尺寸。OBB能有效的提高系统的检测效率,但在解决系统的实时性检测问题时还稍显不足,可以在其基础上进行改进^[8]。

2 混合层次包围树算法描述

2.1 算法流程概述

在预处理阶段,构建混合层次包围树HBVT,按照自顶向下的顺序对物体进行剖分,建立层次二叉树,具体分为三层结构,分别为上层(X层)的Sphere包围盒,中间层(Y层)的OBB-Sphere混合包围盒,下层(Z层)的OBB包围盒。根据包围树自顶向下的节点所对应的包围盒的紧密性,以及精度与速度的需要,调整X、Y、Z的层数。在实时处理阶段,根据不同的碰撞概率以及对碰撞检测精度要求的不同,采用不同的检测方法和检测深度,以提高整个实时系统的检测效率和适用性。根据预处理阶段所构建的HBVT,在遍历非叶子节点时需要进行三种包围盒的检测,即Sphere-Sphere、OBB-Sphere、OBB-OBB;当遍历两棵树到达它们的叶节点时,则要对叶节点内的图元集合进行相交测试,从而判断相交与否。HBVT核心算法CDTraverse()流程见图1,设 a_i 、 b_i 为A、B两棵树上的节点序号; $box(a_i)$ 、 $box(b_i)$ 为节点号对应的包围盒; Layer X、Y、Z分别是X、Y、Z层; CDFlag是碰

撞的标志; $CDSphere_Sphere()$ 为Sphere-Sphere检测方式的实现函数, $CDSphere_OBB()$ 为OBB-Sphere检测方式的实现函数, $CDOBB_OBB()$ 为OBB-OBB检测方法的实现函数。

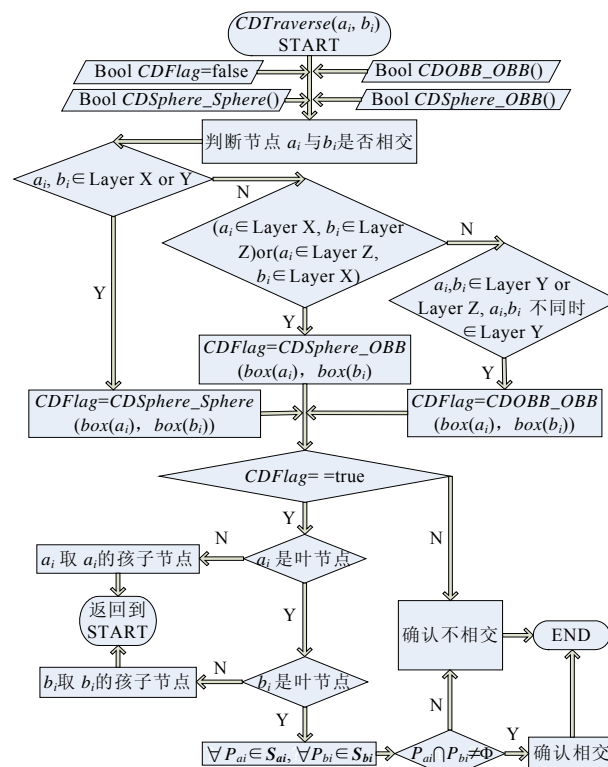


图1 HBVT遍历核心算法CDTraverse流程图

2.2 HBVT检测算法的选取

当遍历A物体的X、Y层节点与B物体的X、Y层节点时,采用Sphere-Sphere检测方式,实现函数为CDSphere_Sphere()。因为这种检测方式简单快速,适于快速排除距离较远的物体。当遍历A物体X层的盒子与B物体X层时,包围A物体X层的盒子与B物体X层的盒子都为Sphere,包围物体的紧密性最差,所以直接用Sphere-Sphere检测方法;当遍历A物体X、Y层的盒子与B物体Y层时,由于二者紧密性都不是很好,利用Y层具有混合结构的特点,取Y层的Sphere结构作为参与碰撞检测的包围盒,用Sphere-Sphere的方法进行检测。

当遍历A物体的X层节点与B物体的Z层的节点时使用OBB-Sphere,实现函数为CDSphere_OBB()。因为当检测A物体的X层与B物体的Z层时,由于一个盒子比较紧密,另一个很紧密,所以采用OBB-Sphere的检测方式。若这时采用Sphere-Sphere则需要进一步向树的底层遍历,相比每次少花费的检测时间,而且可能导致即使遍历到了系统默认的最底层,依然无法精确判定是否相交;倘若用OBB-OBB,因为一次检测OBB-OBB最多进行15次(每个OBB各个面的法矢共6个,每个OBB的3个法矢之间的差积形成的9个法矢,共15个法矢)分离轴的测试,而OBB-Sphere最多只要进行3次测

试,而此时A物体的包围盒不够精确,可能在每一次的重叠检测中花费更多的时间。

当A物体的Y层或Z层与B物体的Z层作重叠检测时使用OBB-OBB检测方法,实现函数为 $CDOBB_OBB()$ 。利用Y层具有混合结构的特点,这里取用Y层的OBB结构作为参与碰撞检测的包围盒,虽然Y层的盒子不是非常紧密的包围A物体的元素,但是相比先前的X层的逼近状态已经比较接近A物体的元素了,而且又由于B物体现在已经被判定在Z层可能相交,对应的OBB包围盒在经历X, Y层的紧缩后,已经可以提供相当紧缩的OBB盒了,所以采用OBB-OBB来进行检测。当A物体的Z层与B物体的Z层进行检测的时候,由于2者的OBB包围盒都经过了前面(X+Y)层的剔除,已经能够相当紧密的贴近物体了,所以也使用OBB-OBB检测。

当遍历两棵树到达它们的叶节点时,则要对叶节点内的三角形面片集合进行相交测试。如果任意一对面片相交,那么判断两物体相交,否则继续遍历叶节点其它面片直到结束,如果结束时仍没有相交的图元对,则两物体不相交。

3 预处理

BV树,即 $BVT(S)$,其中 S 是对象的基本图元的集合,通常是在预处理阶段构建。由包围盒算法的代价函数知,预处理时间并不包含在代价函数之内,所以这就允许在预处理中做一些优化从而使实时碰撞检测的速度得到提高。在某些复杂场景中,为了能使BVT的碰撞检测速度提高,允许在预处理中用很长的时间建树,然而在小型交互的应用中,用户对于几分钟的预处理时间是无法忍受的。预处理计算和存储的初始化信息越多,实时碰撞检测就可能越快,因此在不增加实时碰撞检测时间的前提下,优化预处理是必要的。

3.1 BV的选择

复合HBV的宗旨是在实时检测时尽可能快地排除不可能碰撞的物体,而树的顶层又是每次遍历的开始,顶层BV测试的次数最多,所以我们希望树的顶层节点的重叠检测比较简单,更新也尽量简单。从而使相距较远的物体在简单快速的检测中即被排除掉。

而靠得较近的物体,经过BV树的遍历,它们的BV会越来越紧密地贴近原物,粗略的算法已经不能满足要求,越来越需要精确的检测,从而使靠得较近但没有接触的物体也被排除掉,这就需要紧密性好的BV。基于上述构建复合BVT的核心思想,根据复合BVT各层的需求,考察各种BV的特点,然后选择各层的BV。

Sphere包围盒是最简单的一种,它的优点是旋转不变性,更新快速,且彼此间距离的测量和重叠测试都很简单,缺点是球形包围盒紧密性差。而这正适合复合HBV顶层的要求,顶层命名为X层。

由于OBB根据被包围对象的形状特征尽可能紧密地包

围对象,其紧密性很好。但实物体的重叠检测很耗时,物体运动时,OBB的平移操作与Sphere是一样的计算量,但OBB的旋转的更新计算量较大,需要对OBB三个方向轴进行旋转变换计算,所以更适合BV树的底层(Z层)遍历。

考虑到BV树自顶向下的对物体进行剖分后,各层包围盒的紧密程度的渐变性,所以在X层与Z层之间插入了中间层Y层,Y层使用OBB-Sphere的结构,用来解决从X层到Z层重叠检测的复杂度提升过快的问题。

3.2 树的度的选择

建立层次结构时,通常都希望能减小树的深度从而在遍历树的时候,从根到叶子经历很少的步数,完成搜索。而树的度,确定了所有节点中最大的子节点的个数。在节点数 N 确定的情况下,树的宽度与深度成反比。权衡树的深度与宽度的关系,我们选择二叉树,原因有二:首先,在节点剖分时,2个比分成2个以上的子集所需的剖分和建立新的子节点的花费要少;其次,分析数据证明二叉树优于 s 树($s>2$),特别是考虑到平衡树。证明如下:若 $s\geq 2$,则在遍历树的每个节点时要测试 s 个子节点中至多 $(s-1)$ 个节点,所以从根到叶子的查找所耗费的总时间量与 $f(s)=(s-1)*\log_2 n$ 成比例,其中 n 为叶子节点数。在区间 $s \in (1, +\infty)$ 上, $f'(s)>0$,即 $f(s)$ 递增,因此,当 $s=2$ 时, $f(s)$ 有最小值。

3.3 HBV构建顺序

对于 $BVT(S)$,可以按自顶向下的顺序,也可以按自底向上的顺序。自底向上方法为开始于输入原物集,并将原物集中的元素作为叶子,然后根据元素空间位置等信息进行递归的分组,直至达到一个根节点包含所有的原物集合 S 。自顶向下的方法为开始于近似于 S 的一个节点,然后把物体分割成几个部分(对于二叉树,分割成两部分),每个部分用一个包围盒包围住,这样再对每一个包围盒递归分割进行下去,直到每一个包围盒不能再分割为止,这时到达叶节点。选用技术成熟的自顶向下方法构建BV树。

3.4 剖分原则与策略

为了确定节点是否剖分,应该满足以下原则:划分后的子节点包含的片元个数都要小于父节点所含个数,否则没有继续划分的意义;划分所得的节点深度小于事先规定的深度,因为树的深度过大会增加检测的时间,造成进行无价值的精确剖分;每个节点所含片元个数大于预先给定的最小个数,因为划分不必过细,满足要求即可。

本系统选择了自顶向下的方式构建BV树,且树的度为2,需要不断地把一个三角形集合 S 一分为二,剖分的关键是分割平面的选择,即分割点和分割平面法面矢量的选择。考虑到复合HBV的BV构建复杂性以及BV的复合连接,这里应尽量减少剖分的时间,使用最简单的方法。这里采用技术相对成熟的OBB盒的剖分方法来剖分物体,根节点就是物体的

点集。

首先,为了达到树的平衡,将分割点选在所有元素集合的中心。分割点选好后就是分割面法线的选择了。对于OBB的可能存在的分割轴,会与OBB包围盒某根方向轴(共3根)平行。设将要被分割的三角形集合中的三角形面片数为 N ,以第 i 条方向轴为分割轴时,左子树的三角形面片数为 L_i ($i=1, 2, 3$)。如果对于任意 L_i , L_i 和节点深度都不满足剖分原则,则停止对此节点划分,此节点位数的叶子节点;倘若满足剖分原则,则计算 $d_i = |N/2 - L_i|$ ($i=1, 2, 3$)的最小值,以此确定最小值对应的 i 为分离轴。然后就需要把一个三角形集合分配到左右子树,策略是:把三角形中心作为判断点,如果三角形中心在分割平面的左侧,那么该三角形划分到左节点,否则划分到右节点,如果三角形中心在分割平面上,根据左右节点数的多少,把三角形分到较少的节点上,使得二叉树尽量平衡。判断中心点 v 在面左右的方法是:点与面上任一点的连线所决定的向量 $\vec{v_0v}$ 与面的法线 \vec{n} 点乘,根据结果正负决定该点在面的左右,如果结果为正,则该点在面的左边,为负则在右边,若为零,则点在平面上。

3.5 复合HBV树的构建过程

对于一对给定的物体, X层采用Sphere结构, Y层采用OBB-Sphere结构, Z层采用OBB结构,但是考虑到复合建树的技术可行性, X和Y层节点的包围盒生成均采用OBB-Sphere混合结构。这种结构是在OBB包围盒建好之后,在OBB外接一个sphere,其中心就是OBB的中心,半径通过 $R = \sqrt{a^2 + b^2 + c^2}/2$ 计算得出, a 、 b 、 c 分别是OBB的边长;利用一次矩(均值)和二次矩(协方差矩阵)统计量来计算OBB节点的位置和方向。设第 i 个三角形的顶点矢量为 p^i , q^i 和 r^i ,包围盒包围的三角面片数为 n 。包围盒的中心位置如公式(2):

$$m = \frac{1}{3n} \sum_{i=1}^n (p^i + q^i + r^i) \quad (2)$$

协方差矩阵元素计算如公式(3):

$$C_{jk} = \frac{1}{3n} \sum_{i=1}^n \overline{p_j^i p_k^i} + \overline{q_j^i q_k^i} + \overline{r_j^i r_k^i}, 1 \leq j, k \leq 3 \quad (3)$$

其中, $\overline{p^i} = p^i - m$, $\overline{q^i} = q^i - m$, $\overline{r^i} = r^i - m$,利用数值的方法解出协方差矩阵的特征向量并单位化,由于 C 是一个实对称矩阵,所以矩阵 C 的特征向量相互垂直,可以作为包围盒的方向轴。把将要包围的几何体的顶点向方向轴上投影,找出各方向轴的投影区间,各投影区间的长度就是所求OBB盒的相应边长 a 、 b 、 c 。X层与Y层节点建成之后,按照前文(章节3.4)提到的剖分策略生成各自的子节点。

对于Z层节点的建立,由于采用了OBB包围盒,所以继续利用上文提到的方法,即用一次矩(均值)和二次矩(协方差矩阵)统计量来计算包围盒的位置和方向;并利用(章节3.4)提到的剖分策略生成Z层节点的子节点。

由于包围盒的中心位置是各三角形顶点简单的平均,如

果构成几何体的三角形的尺寸不均匀,这样计算出来的包围盒就会向着三角形尺寸小且密集的部分偏移。这样一方面使得构造的BVT不平衡,影响碰撞检测的效率,另一方面这对协方差矩阵也有影响,从而导致包围盒方向向量的计算不准确,使得包围盒的紧密性受到影响。这里采用的是给每一个三角形加权^[9],选用三角形的面积作为权值的办法来计算各层包围盒的中心,可以有效避免上面提到的问题。

各层节点的剖分标准统一使用章节3.4提到的标准, X、Y、Z的层数预先做好设置,其比例根据不同场景复杂度的需要进行调整,以此自顶向下建立HBV树。

4 实时检测

4.1 BV重叠测试

本算法在实时检测阶段,需要进行的重叠测试有3种,即Sphere-Sphere、OBB-Sphere、OBB-OBB。

4.1.1 Sphere-Sphere检测

只要计算两球心间的距离是否大于两球半径之和,采用它们的平方形式,如公式(4):

$$|o_1 o_2|^2 > r_1^2 + r_2^2 + 2r_1 r_2 \quad (4)$$

其中,设 o_1 、 o_2 为球心,则 $|o_1 o_2|$ 为球心间距离, r_1 、 r_2 分别为两个球半径。

4.1.2 OBB-OBB检测

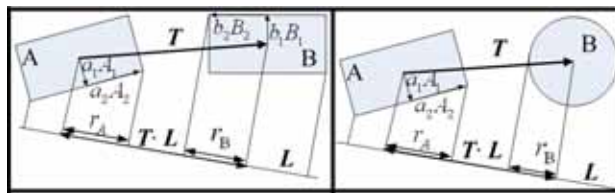
如图2(a)所示,假设有两个OBB盒子包围盒A和B, a_i 、 b_i 分别是A、B的边长一半 ($i=1, 2, 3$); A_i 、 B_i 分别是A、B的轴单位向量 ($i=1, 2, 3$); T 是A的中心与B的中心的距离, L 是平行于分离轴的单位向量。 r_A 为A的三个半径在 L 方向上的投影之和, r_B 为B的三个半径在 L 方向上的投影之和。只要比较 $|T \cdot L|$ 与 $(r_A + r_B)$ 的和就能判断二者是否相交。由上述已知条件,则有公式(5):

$$r_A = \sum_{i=1}^3 |a_i A_i \cdot L|, r_B = \sum_{i=1}^3 |b_i B_i \cdot L| \quad (5)$$

因此,只要判断公式(6),就能对A、B两个物体是否相交进行判断。其中 L 为15个分离轴中的一条。

$$|T \cdot L| > r_A + r_B = \sum_{i=1}^3 |a_i A_i \cdot L| + \sum_{i=1}^3 |b_i B_i \cdot L| \quad (6)$$

当前者大于后者,则两个物体不相交;否则继续计算A、B在其他14个分离轴的投影,并利用上述公式进行比较,直到15个轴的投影都不满足上式,则判定A、B物体相交。



(a) 有向盒间的碰撞

(b) 有向盒与球的碰撞

图2 不同包围盒间的重叠测试

4.1.3 OBB-Sphere检测

如图2(b)所示, 假设有两个盒子, 分别是OBB包围盒A和Sphere包围盒B, a_i 是A半径 ($i=1,2,3$), r 是B的半径; A_i 是A的轴单位向量($i=1,2,3$); T 是A的中心与B的中心的距离, L 是平行于分离轴的单位向量。 r_A 、 r_B 分别为A、B的三个半径在 L 方向上的投影之和, 如公式(7):

$$r_A = \sum_{i=1}^3 |a_i A_i \cdot L|, \quad r_B = r \quad (7)$$

只要比较 $|T \cdot L|$ 与 $(r_A + r_B)$ 的和, 就能比较出2者是否相交, 因此只要判断公式(8), 就能对A、B两个物体是否相交进行判断。其中 L 为OBB包围盒三个分离轴中的一条。

$$|T \cdot L| > r_A + r_B = \sum_{i=1}^3 |a_i A_i \cdot L| + r \quad (8)$$

例如当前者大于后者, 则两物体不相交; 否则继续计算A、B在其他2个分离轴的投影, 并利用上述公式进行比较, 直到3个轴的投影都不满足上式, 则判定A、B物体相交。

4.2 运动物体HBV的更新

在构建HBV时, 物体的BV在最初建造时也是在模型坐标系统里, 所以在BV检测前也必须将它们转换到同一个坐标系统中, 这种转换称为BV的更新。有两种方法使它们在同一个坐标系统里, 一种是将两BV都转换到世界坐标系统, 另一种是将一个BV转换到另一个模型坐标系统中。为了避免将两个BV都作变换, 本系统选择第二种策略。

物体相对另一物体做运动, 可分解成平动和绕自身的转动。当物体相对另一物体作平移运动时, 它的BV树仍然有效, 只要在所有的方向上加上一个物体平移的向量即可, 混合HBV在平动的更新中, 相比Sphere和OBB并不多花费时间。下面将分析物体旋转的时候的更新。

对于X层更新, 相比纯粹的OBB方式具有很大的优势, 因为X层的更新, 是对球形结构进行检测, 因此只对球形BV进行更新即可, 而包围球在旋转变换中不需要任何操作。

对于Y层更新, 虽然Y层具有混合结构, 所以需要混合结构的两种结构(Sphere和OBB)同时更新。由于球体的旋转不变性, 总的旋转更新时间跟OBB旋转更新是一样的。

对于Z层更新, 由于采用了纯粹的OBB结构, 所以仅需要按照正常的OBB更新操作即可。由于OBB包围的是刚体, 所以OBB的形状不会改变, 即大小不变, 只要对OBB的三个方向轴进行相应的旋转变换即可。

对于更新中的优化, 在物体运动的每一步, 并没有必要更新所有的BV, 如果上一层的BV是分离的, 则下一层的BV就不会参加干涉测试, 也就不需要更新了, 这样相比纯粹OBB算法, 就能将更新Y层混合结构所多花费的更新球心的时间缩得更小, 还可同时执行BVT的更新和复合BVT的遍历。比如, 在搜索与A物体某一节点可能发生碰撞的B物体的节点的时候, 排除A物体的节点与B物体的节点相交的可能性后, 就可以通过布尔数据类型标识B物体的节点, 使该节点不进行接下来的更新操作。

5 实验结果与数据分析

本系统(HBVHS混合包围盒层次系统)采用AMD 64bit 工作站 (AMD 64bit 3000+ CPU, NVIDIA 6800GT 显卡和2 GB DDR400 内存), VC 8.0进行开发。下面给出茶壶模型碰撞检测的图像采样及数据采样。本实验描述的是一个茶壶向另一个茶壶作匀速直线相对运动, 并发生碰撞的简略过程, 其中每个茶壶由16256个面、8257个顶点组成。表1显示了HBVHS(混合包围盒层次系统)在两个茶壶模型运动到不同的碰撞位置时系统各项参数值的采样情况。图3(a)、(b)、(c)、(d)分别为HBVHS四个运行截图, 发生碰撞的简略过程。

使用RAPID系统是衡量刚体碰撞检测效率的有效尺度之一, HBVHS集成了RAPID系统的OBB核心算法模块, 这样就实现了在同一个碰撞场景中, 在得到相同正确的检测结果

表1 不同的重叠数目对应不同的参数值

重叠数目	每秒帧数(fps)	RAPID 耗时(毫秒)	HBVHS 耗时(毫秒)	速度提升倍数
0	209.72	1.01	4.77	0.2137
28	152.58	13.02	6.55	2.030854
116	153.45	24.93	6.52	3.823658
238	143.71	16.31	6.96	2.343968
364	101.32	23.74	9.87	2.405191
496	86.74	32.74	11.53	2.839676
694	68.99	33.13	14.50	2.28478
822	60.06	46.15	16.65	2.771996
936	52.35	48.05	19.10	2.515883
1061	44.56	55.88	22.44	2.490184
1282	31.49	60.30	31.76	1.898586

果(相同的相交面数)的前提下,计算出混合层次结构的算法和OBB算法的运算消耗。图4显示了HBVHS和RAPID系统在检测到不同重叠数目时场景绘制的计算时间(毫秒),由图知HBVHS在各种检测的复杂度下变化较为平稳,在最差的情况下,HBVHS计算时间约为31.76毫秒;在同等情况下OBB算法与本算法计算时间的差值随着碰撞复杂性的增加

而增大,相应的本算法的检测速度的提升也就越明显,这证明本算法相对OBB算法在复杂场景中执行效率的提升。

图5显示了HBVHS在检测到不同碰撞值时场景绘制的流畅程度(FPS),避免动作不流畅的最低要求是30FPS,HBVHS在碰撞检测的重叠测试较多的情况下,仍能保证场景绘制速度在30 FPS以上,能够满足实时检测的要求。



图3 茶壶间发生碰撞的效果图

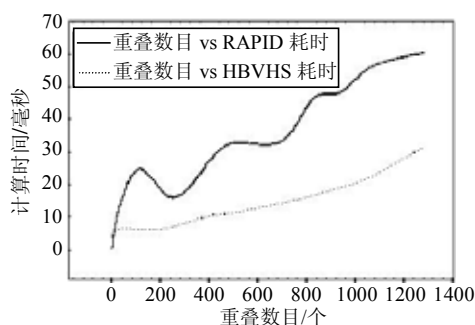


图4 不同碰撞数目下的场景绘制的计算时间

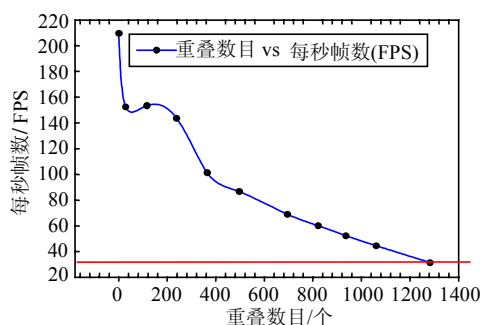


图5 不同碰撞数目下的场景绘制的流畅程度

6 结论

在复合HBV中, X层节点采用球形BV, 通过计算球心距离即可完成检测, 而物体平移时球形BV不需要更新。当距离逐渐靠近, 需要OBB-Sphere进行更准确的检测。OBB-Sphere重叠测试依靠的是分离轴理论, 只是OBB算法中的一个有向长方体变成了球体, 最多需要计算三次分离轴判定的计算。当两个物体更加靠近时, 用到OBB的重叠检测算法, 虽然有一些优化的方法, 然而代价依然比较高。通过调节X, Y, Z的层数, 以及融入一些比较成熟的优化策略, 可获得更令人满意的测试结果。

参考文献:

- [1] M Lin, S Gottschalk. Collision Detection between Geometric Models: A Survey [C]// The Proceedings of IMA Conference on Mathematics of Surfaces, Birmingham, UK, 1998. Birmingham: Eurographics Association, 1998: 37-56.
- [2] 郑轶, 宁汝新, 刘检华, 等. 虚拟装配环境下快速碰撞检测方法的研究 [J]. 系统仿真学报, 2005, 17(9): 2167-2170.(ZHENG Yi, NING Ru-xin, LIU Jian-hua. Research on Fast Collision Detection Method in Virtual Assembly Environment[J]. Journal of System

Simulation, 2005, 17(9): 2167-2170.)

- [3] 周云波, 闫清东, 李宏才. 虚拟环境中碰撞检测算法分析[J]. 系统仿真学报, 2006, 18(S1): 103-107. (ZHOU Yun-bo, YAN Qing-dong, LI Hong-cai. Collision Detection Algorithms Analysis in Virtual Environment [J]. Journal of System Simulation, 2006, 18(S1): 103-107.)
- [4] 魏迎梅, 吴泉源, 石教英. 虚拟环境中的碰撞检测方法 [J]. 计算机工程与科学, 2001, 23(2): 44-47.
- [5] 马登武, 叶文, 李瑛. 基于包围盒的碰撞检测算法综述 [J]. 系统仿真学报, 2006, 18(4): 1058-1064. (MA Deng-wu, YE Wen, LI Ying. Survey of Box-based Algorithms for Collision Detection [J]. Journal of System Simulation, 2006, 18(4): 1058-1064.)
- [6] Larsson T, Möller TA. Collision detection for continuously deforming bodies [C]// The Proceedings of Eurographics 2001, Manchester: Eurographics Association, 2001: 325-333.
- [7] Palmer I J, Grimsdale RL. Collision Detection for Animation Using Sphere-trees [J]. Computer Graphics Forum(S0167-7055), 1995, 14(2): 105-116.
- [8] 章勤, 黄侃, 李光明. 一种基于 OBB 的碰撞检测算法的改进 [J]. 华中科技大学学报, 2003, 31(1): 46-48.
- [9] 陈学文, 丑武胜, 刘静华, 等. 基于包围盒的碰撞检测算法研究 [J]. 计算机工程与应用, 2005, 41(5): 46-47.