

一种基于混合层次包围盒的快速碰撞检测算法

刘晓东, 姚兰, 邵付东, 王颖

(西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 提出了一种基于混合层次包围盒(BVH)的快速碰撞检测算法, 它将固定方向凸包(FDH)包围盒与沿坐标轴方向的包围盒(AABB)相结合, 吸收了基于图像的碰撞检测算法的优点, 利用了图形硬件的计算能力. 算法在预处理阶段为待测物体的凸块构造 FDH 包围盒树, 然后在逐步求精的过程中遍历该树并进行初步碰撞检测, 最后通过建立 AABB 包围盒快速设置视域参数, 在对凸块绘制的同时进行精确求交, 实时得到碰撞检测的结果. 实验结果表明, 所提算法可以较好地解决形状复杂刚体的实时碰撞检测问题, 检测速度比经典算法 RAPID 和 RECODE 分别提高了约 17% 和 20%.

关键词: 实时检测; 层次包围盒; 固定方向凸包; 刚体碰撞

中图分类号: TP391 **文献标识码:** A **文章编号:** 0253-987X(2007)02-0141-04

Fast Collision Detection Algorithm Based on Mixed BVH

Liu Xiaodong, Yao Lan, Shao Fudong, Wang Ying

(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A fast collision detection algorithm based on mixed hierarchical bounding volumes (BVH) is proposed, which combines the quality of compactness of the fixed direction hull (FDH) with the simplicity of AABB, absorbs the merits of image-based collision detection algorithm, and utilizes the computing capabilities of graphic hardware. In the preliminary period the FDH tree of the objects is established. Then the tree is traversed and the original detection is carried out simultaneously. Subsequently, the AABB of the object is established to set the view parameter, and the detection result could be obtained in real time when the convex pieces of the objects are drawn. The experimental results demonstrate that the proposed algorithm works well in the collision detection for complexly shaped rigid-body, and the speed of the collision detection is improved by about 17% and 20% compared to the traditional algorithms RAPID and RECODE.

Keywords: real-time detection; hierarchical bounding volumes; fixed direction hull; rigid-body collision

随着虚拟现实、分布式交互仿真等技术的兴起, 碰撞检测问题开始成为研究的热点. 碰撞检测算法在空间采样方式上可划分为基于物理空间和基于图像空间两大类, 在前一类算法中, 基于层次包围盒树的方法是研究的热点. 具有代表性的包围盒有沿坐标轴方向的包围盒(AABB)^[1]、方向包围盒(OBB)^[2]以及固定方向凸包(FDH)^[3]. 利用包围盒

可以将碰撞检测问题简化, 但是紧密性好的包围盒计算相对复杂, 简单性好的包围盒精确度不够, 这两类包围盒均难以同时满足速度和精确度的要求. 基于图像空间的碰撞检测算法^[4-5]能有效利用图形硬件的绘制加速功能, 但是这种方法在对物体的所有凸块进行绘制时没有经过预先的筛选, 在一定程度上影响了检测效率.

本文给出了一种将 FDH、AABB 层次包围盒和图像空间相结合的碰撞检测算法,在不影响碰撞检测精确性的前提下,它的检测速度比以往的算法有较大的提高。

1 算法实现

整个算法分为预处理、逐步求精、精确求交 3 个阶段。算法的输入为三维空间中的 2 个物体,输出为 2 个物体的碰撞检测结果,即 True(相交)或 False(不相交)。

1.1 预处理阶段

预处理阶段的主要工作是将物体对的每个对象分解为凸块,构造凸块的 FDH 层次包围盒树,并对凸块进行三角形带压缩编码,以便于提高后 2 个阶段的效率。工作步骤描述如下。

步骤 1: 对 2 个物体分别进行表面凸分解,得到一组凸片集合,并针对每一凸片构建与其相应的凸块。本文采用对偶图搜索方法和增量凸片算法来搜索凸片集合^[6]。

步骤 2: 采用自顶向下的策略将凸块集合组织为层次二叉树。从凸块集出发,以全集作为根结点,利用基于全集信息递归对这个结点进行划分以形成其子结点,直至到达叶结点。

步骤 3: 为层次树中每个凸块构建一个 FDH 包围盒。具体方法如下:① 选定固定方向集 D_k , 本文取 $D_{14} = \{\pm(1,0,0), \pm(0,1,0), \pm(0,0,1), \pm(1,1,1), \pm(1,-1,1), \pm(1,1,-1), \pm(1,-1,-1)\}$;② 计算凸块集 S 中的顶点与固定方向集 D_{14} 中各个方向的最大点积,得到凸块集的 FDH 包围盒。

步骤 4: 对层次二叉树上的所有节点的凸块进行三角形带压缩编码。采用 Isenburg 的启发式搜索法^[7],一次性地对层次二叉树中的每个结点凸块的三角形网格进行处理,得到比较理想的三角形带,并对这些三角形进行压缩编码。

1.2 逐步求精阶段

建立好 FDH 树之后,算法在逐步求精阶段会同时遍历环境对象和活动对象的层次凸块二叉树,检测凸块节点所对应的 FDH 包围盒是否相交。

1.2.1 层次二叉树的遍历 遍历算法在预处理阶段已为物体对的每个对象各构造了一个层次二叉树,层次树的每个结点表示一个由 FDH 包围盒包围的凸块。因此,检测 2 个物体是否相交可通过同时递归遍历它们的层次二叉树来确定,遍历的过程主

要由一个递归调用函数 $\text{TraverseTree}(V_E, V_F)$ 来实现。遍历算法如图 1 所示,其中 V_E 为环境对象包围盒树中的当前结点, V_F 为活动对象包围盒树的当前结点,该过程的输入为环境对象包围盒树的根结点 E 和活动对象包围盒树的根结点 F , S_E 和 S_F 分别为结点 V_E 和 V_F 所对应的凸块, $b(S_E)$ 和 $b(S_F)$ 分别为它们的包围盒。

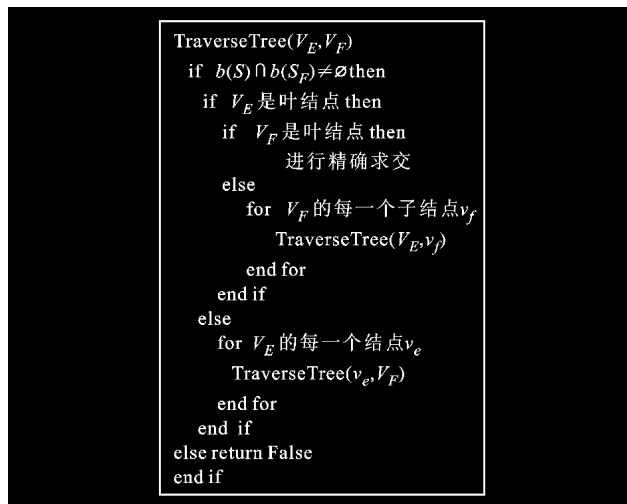


图1 层次二叉树的遍历算法

1.2.2 包围盒间的相交测试 由于所有的 FDH 都由同一个固定方向集合 D_k 中的 $k/2$ 个方向相反的向量对来定义,故一个 FDH 完全可以在 $k/2$ 个方向轴的投影区间上定义。对于 FDH 的相交测试,可采用区间重叠测试法^[3],即:如果 2 个包围盒在 $k/2$ 个固定轴上的某个轴的投影区间不重叠,就可以判定这 2 个包围盒不重叠,包围盒的曲面也必定不相交;如果在所有方向上的投影区间都重叠,认为 2 个包围盒相交。

1.3 精确求交阶段

算法在逐步求精阶段检测到 2 个凸块包围盒相交后,需要进一步断定这 2 个凸块是否相交,这个过程就是精确求交。在这一阶段,算法首先基于 AABB 包围盒设置视域参数,再进行凸块相交测试,得到碰撞检测结果。

1.3.1 基于 AABB 的视域参数设置 为了提高精确求交阶段的效率,算法为 2 个凸块重新建立 AABB 包围盒,因为 AABB 简单,易于为绘制过程设定相应的视域参数,而 FDH 的方向轴过多,不利于设定这些参数。

AABB 包围盒可用各个坐标轴上的最小、最大值所限定的区域来表示^[1],即

$$R = \{(x, y, z) \mid l_x \leq x \leq u_x, l_y \leq y \leq u_y, l_z \leq z \leq u_z\} \quad (5)$$

式中: l_x, l_y, l_z 分别为 x, y, z 轴上的最小值, u_x, u_y, u_z 分别为 x, y, z 轴上的最大值, 这些值均为区域内点的边界。

绘制所用到的视域参数包括相机位置、视线方向、视口大小和投影矩阵, 而本文的投影方式采用平行投影。设置视域参数的步骤如下。

步骤 1: 分别为 2 个凸块建立 AABB 包围盒。

步骤 2: 在 3 个坐标轴中, 将 AABB 包围盒的重叠区间最长的一个坐标轴方向确定为视线方向, 剩余的 2 个重叠区间构成一个矩形, 将这个矩形作为视口。

步骤 3: 依据视线方向、视口大小和 AABB 包围盒的尺寸来设定相机位置。相机在视线方向的坐标值为 $-\infty$, 在其他 2 个方向的坐标值与视口中心坐标值相同。

例如, 建立 2 个 AABB 包围盒, 假定它们在 z 轴上重叠区间最长, 则 z 方向被设定为视线方向, 而 x 和 y 方向上的重叠区域被设定为视口, 如图 2 所示。根据 2 个包围盒在 x, y 方向上的最大、最小值, 可以得到视口的范围。

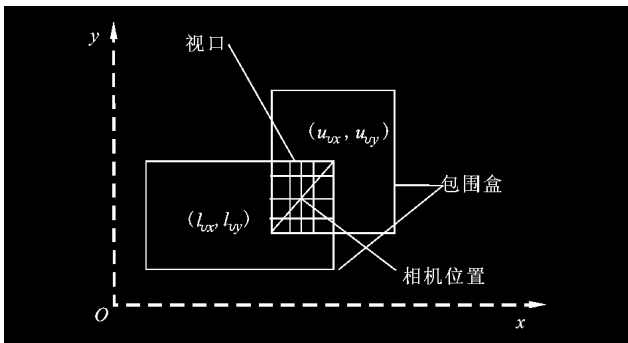


图2 视域参数设置

设视口 $\gamma = \{(x, y) | l_x \leq x \leq u_x, l_y \leq y \leq u_y\}$, 则相机位置由坐标 $(C_1, C_2, -\infty)$ 来确定, 其中

$$C_1 = (l_x + u_x) / 2$$

$$C_2 = (l_y + u_y) / 2$$

利用 FDH 的相交检测结果建立 AABB 包围盒, 可以迅速实现视域参数的设定, 从而达到加快绘制过程的目的。

1.3.2 凸块之间的相交测试 2 个凸块之间的相交测试是在图形硬件绘制过程中实时进行的, 主要用到了深度缓存和模板缓存。本文通过比较重叠像素的深度值来修改模板缓存值, 再根据模板缓存值来确定凸块是否相交。

假设 z 轴为视线方向, 用 $R_{xy}(x)$ 表示物体在 xOy 平面上的垂直投影区域, 用 $I_z(x)$ 表示物体在 z 轴上所占的区间。那么, 2 个凸块 A, B 发生碰撞时

当且仅当下式成立^[5]

$$R_{xy}(A) \cap R_{xy}(B) \neq \emptyset, I_z(A) \cap I_z(B) \neq \emptyset$$

对 A, B 而言, 在某像素上的 z 轴区间的重叠情况有 8 种。当只比较 A 的正面与 B 的前、后面之间的位置关系时, 可进一步将 8 种情况归纳成为 4 类^[8], 并且可用模板缓存值 0~3 进行区分, 如图 3 所示。

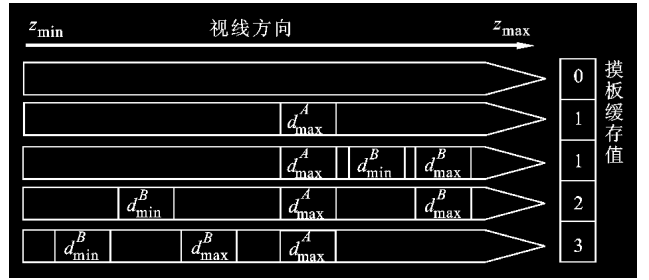


图3 A正面与B的位置关系

凸块相交测试步骤如下。

步骤 1: 在已设置的视域下, 采用背面剔除的方式来绘制 A , 将 A 在投影区域内的所有像素的模板缓存值置 1, 并将其最大深度值 $d_{\max}^A$ 写入相应的像素深度缓存。绘制时, 通过实时解码绘制以三角形带编码保存的凸块。

步骤 2: 将 B 分正面和背面进行 2 次三角形带解码绘制。每次绘制时, 检测 B 的投影区域满足模板缓存值为 1 的像素点, 并将 B 上该像素点的深度值 d^B 与当前深度缓存中的值进行比较, 若前者小于后者, 则该像素的模板缓存值加 1。

步骤 3: 查询视域中所有的像素模板缓存值, 若均小于或等于 1, 则 A, B 不相交, 返回 False; 若发现至少有一个值为 2, 则 A, B 相交, 返回 True; 其他情况, 执行步骤 4。

步骤 4: 将 A 和 B 的角色互换, 重复步骤 1~步骤 3, 最终确定 A, B 是否相交。

2 算法复杂度分析

下面从构造 FDH 包围盒树的时间开销和对存储空间的要求进行比较分析。

对于不同的包围盒类型, 计算复杂度主要取决于其构造过程和相交测试的复杂度。本文算法采用的 FDH 包围盒, 固定方向集取 D_{14} , 所以构造凸块集合 S 的 FDH 的时间复杂度为 $O(14n)(n=|S|)$, 而每对 FDH 间的相交测试需要 14 次比较运算。OBB 包围盒在构造时需要计算凸块集合 S 中所有元素的协方差矩阵、特征向量以及顶点在 3 个特征向量上的最大值, 而且 2 个 OBB 的相交测试需要

15次比较运算、60次加减运算、81次乘法运算和24次绝对值运算^[2],其时间开销大于本文所用的FDH.

FDH树对存储空间的要求比较低,对树中的每一个结点,只需存储定义FDH的表示固定方向的浮点数,这里共有14个方向(需112个字节)、存储指向子结点的2个指针(共8个字节)和指向父结点的1个指针(4个字节),而对叶结点还需保留1个指向相应的基本几何元素的指针(4个字节).一棵完全的FDH树,共有 n 个叶结点和 $n-1$ 个内部结点,故本文算法中的FDH树所需存储空间为 $256n$ 个字节,而OBB在各结点上的包围盒方向的任意性使得它需要更多的存储空间($336n$ 个字节)^[2].

采用C++,OpenGL/GLUT在PC机(PIV 3.0 GHz,内存1 GB,显卡MSI RADEON X600 PRO,显存128 MB)上实现了本文提出的基于混合BVH的碰撞检测算法.图4为测试场景中虚拟手的运动.

通过实验数据比较了本文算法、基于OBB包围盒的碰撞检测算法(RAPID)^[2]、基于图像的碰撞检测算法(RECODE)^[4],它们在不同情况下的平均碰撞检测时间见表1.从表中可以得出,在速度上本文

算法比RAPID、RECODE分别提高了约17%和20%.根据对算法的描述和分析得知,速度提高的主要原因有以下几个方面.

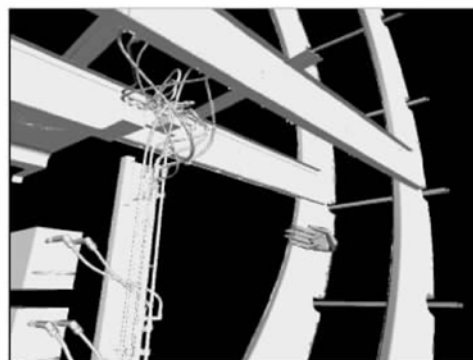


图4 测试场景中虚拟手的运动

(1)采用了复杂度较低、效率较高的FDH包围盒进行碰撞初判.

(2)采用了最简单的AABB包围盒来快速设置视域参数.

(3)采用的三角形条带化技术提高了绘制速度.

(4)利用图形硬件的计算能力,使得凸块相交测试能够在绘制过程中实时进行.

表1 平均碰撞检测的时间比较

算法	平均碰撞检测时间/ms									
RECODE	7.021	13.24	18.62	25.90	33.54	42.69	52.10	60.85	68.44	80.23
RAPID	6.816	12.32	16.58	23.96	32.06	40.87	50.63	57.74	67.05	79.65
本文算法	6.120	10.56	11.69	17.25	24.54	32.76	41.35	49.36	62.25	70.96
三角形面片数	2 000	4 000	6 000	8 000	10 000	12 000	14 000	16 000	18 000	20 000

3 结束语

精确的碰撞检测对提高虚拟环境的真实性、增强虚拟环境的沉浸感有着至关重要的作用,而提高速度是虚拟环境中碰撞检测的核心问题.本文主要围绕提高碰撞检测速度展开研究,并提出了一种新的基于混合BVH的快速碰撞检测算法.该算法充分利用了FDH包围盒紧密性好、AABB包围盒简单性好的特点和图形的硬件加速功能,大大地提高了碰撞检测的速度,它可应用于形状复杂的刚体精确碰撞检测,而在对碰撞检测的实时性要求较高的场合中应用效果尤其明显.

参考文献:

[1] Cohen J D, Lin M C, Manocha D, et al. I-COOLIDE:

an interactive and exact collision detection system for large-scale environments [C] // Proceedings of ACM Interactive 3D Graphics Conference. New York: ACM, 1995: 189-196.

[2] Gottschalk S, Lin M C, Manocha D. OBB-tree: a hierarchical structure for rapid interference detection [C] // Computer Graphics Proceedings. New York: ACM, 1996: 171-180.

[3] 魏迎梅, 吴泉源, 石教英. 碰撞检测中的固定方向凸包围盒的研究 [J]. 软件学报, 2001, 12(7): 1056-1063.

Wei Yingmei, Wu Quanyuan, Shi Jiaoying. Research on fixed direction hull bounding volume in collision detection [J]. Journal of Software, 2001, 12(7): 1056-1063.

[4] Baciú G, Wong S K W, Sun H. RECODE: an image-

- based collision detection algorithm [J]. Journal of Visualization and Computer Animation, 1999, 10(4): 181-192.
- [5] 范昭炜, 万华根, 高曙明. 基于图像的快速碰撞检测算法 [J]. 计算机辅助设计与图形学学报, 2002, 9(14): 805-809.
- Fan Zhaowei, Wan Huagen, Gao Shuming. A fast collision detection algorithm in image space [J]. Journal of Computer Aided Design & Computer Graphics, 2002, 9(14): 805-809.
- [6] Ehmann S, Lin M C. Accurate and fast proximity queries between polyhedra using convex surface decomposition [C] // Proceedings of the Eurographics Conference. Manchester, UK: Blackwell Publishers for Eurographics Assoc., 2001: 500-510.
- [7] Isenburg M. Triangle strip compression [C] // Proceedings of Graphics Interface Conference. Waterloo, Canada: Canadian Human-Computer Communication Society, 2000: 197-204.
- [8] 霍滨焱. 基于图像空间的碰撞检测算法 [D]. 哈尔滨: 哈尔滨工程大学计算机科学与技术学院, 2005.
- (编辑 苗凌)