

基于包围盒与空间分解的碰撞检测算法^{*})

李建波 潘振宽 孙志军

(青岛大学信息工程学院 青岛266071)

摘 要 本文提出了一种基于包围盒方法与空间分解方法相结合的碰撞检测算法,用于解决变形体的碰撞检测问题。该算法首先用包围盒来快速判断物体之间是否相交,如果相交则进一步用空间分解法来定位相交的区域,在此阶段用哈希表的数据结构来保存物体的几何信息。与其他碰撞检测算法相比较,本算法不仅能够较大地节省空间,而且时间复杂度也比较低。除此之外,本算法不仅能够找出发生碰撞的基本几何元素对,而且还能够精确地找出碰撞点。

关键词 碰撞检测,包围盒,空间分解,哈希表,AABB

The Collision Detection Algorithm Based on Combination of Bounding Volumes and Space Division

LI Jian-Bo PAN Zhen-Kuan SUN Zhi-Jun

(Information Engineering College, Qingdao University, Qingdao 266071)

Abstract This paper proposes a new collision detection algorithm that combines bounding volume boxes method and space division method. It can solve the collision detection problems between deformable objects. First, the algorithm uses bounding volume boxes to determine if objects are intersecting. If they do, it further uses space division to locate the intersection areas and it uses the data structure called hash table to store the geometry information of objects on this phase. Comparing to other collision detection algorithm, the algorithm this paper proposes not only has the advantage that it can save the storage space well but also its time complexity is comparatively low. Besides, it can find not only the colliding elementary geometry entity pairs but also the precise colliding points.

Keywords Collision detection, Bounding volume boxes, Space division, Hash table, AABB

1 引言

碰撞检测是虚拟现实应用中的一个核心问题。譬如,在一个虚拟场景中,为了模拟的真实性,我们必须避免两个物体的互相穿透(interpenetration)。此外为了给出真实的物理现象或反应,不仅要判断物体之间是否发生碰撞,而且还要在发生碰撞的情况下精确地定位发生碰撞的位置。如果一个虚拟场景对精度要求不高,可以只求出相交的基本几何元素对,但如果是一个类似于虚拟手术这样的对精度要求非常高的系统,必须能够求出碰撞点。本文的碰撞检测算法能够精确地求出碰撞点。

随着计算机硬件处理能力的不断增强,对处理复杂场景以及细节渲染方面提出了更高的要求。其中的一点就表现在当虚拟场景中包含变形对象时,例如人体的器官软组织或布料。尽管在碰撞检测领域国内外学者做了大量的工作,但是这些科研工作大部分都集中于解决虚拟场景中刚体之间的碰撞检测。近几年碰撞检测的热点集中在变形体方面,尤其是解决变形体的自相交问题以及拓扑结构改变的情形。

相对于刚体对象,变形对象表现在组成变形体对象的顶点之间的相对位置发生变化或者是在拓扑结构改变时导致基本几何元素的增加或减少。这就导致了适用于刚体对象的层次包围盒方法不能直接用于变形对象上,因为此时包围盒树需要实时地重新构建或者更新。而重新构建整棵包围盒树的时间耗费在一个实时的仿真环境中是不能接受的,因此在层次包围盒方法中必须避免整棵树的重新构建。由此 Van den Bergen^[1]利用 AABB 的特殊属性提出了自底向上更新的方法来避免重新构建整棵树的巨大耗费。这种方法假设的前提是每个树的内部节点所包含的基本几何元素的个数是不变

的。因此当发生大尺度变形时,包围盒的紧密性就比较差并且包围盒与包围盒之间有非常大的重叠区域,如图1所示。Van den Bergen 发布了称为 Solid 的碰撞检测库。但 Solid 碰撞检测库不能解决变形体的自相交问题,也不适合解决变形体剧烈变形的情况。

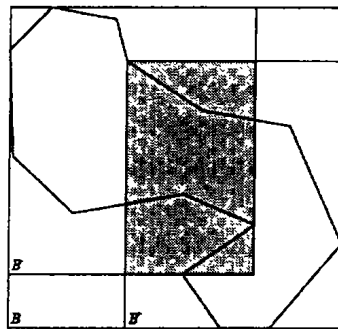


图1 重叠的 AABB 包围盒

Yashifumi Kitamura^[2]等人提出了一种用于解决复杂场景下的变形体的碰撞检测算法。该算法首先求出每个物体的 AABB 包围盒,然后对这些包围盒进行相交测试并求出这些包围盒之间的相交区域(在相交的前提下)。下一步针对每个相交的包围盒所对应物体的基本几何元素,判断它们是否与包围盒相交区域相交。如果相交,对这些基本几何元素建立二叉树的数据结构,求出每个空间网格内相交的基本几何元素对。此方法的缺点是采用二叉树的数据结构实现比较复杂。

J. Mezger^[3]等人在模仿布料仿真时对层次包围盒结构作了很多优化。它采用的包围盒类型是 k-DOPS^[4]。由于 k-DOPS 在最坏情况下只需 k/2 次相交测试,因此 J. Mezger 等人采用的是四叉树(4-ary)。这样减少相交测试时递归调用的

^{*})山东省自然科学基金项目,基金号:Y2003G01。

深度,从而减少了内存消耗。另外它还采用了“向量圆锥”(normal cones)来解决自相交问题。

Matthias Teschner^[5]等人提出了用一种优化的哈希表来解决变形体的自相交问题。本文的第二阶段采用的数据结构也是哈希表,但实现细节不同。另外,本文还增加了第一阶段即用 AABB 包围盒来快速排除不相交的物体,来加快碰撞检测的效率。近年来基于图形硬件的图像-空间技术(image-space technique)^[6~8]被提出用于解决变形体的碰撞检测。

2 本文算法

2.1 算法流程

首先给出算法的流程图,见图2,随后具体讲解算法的细节。

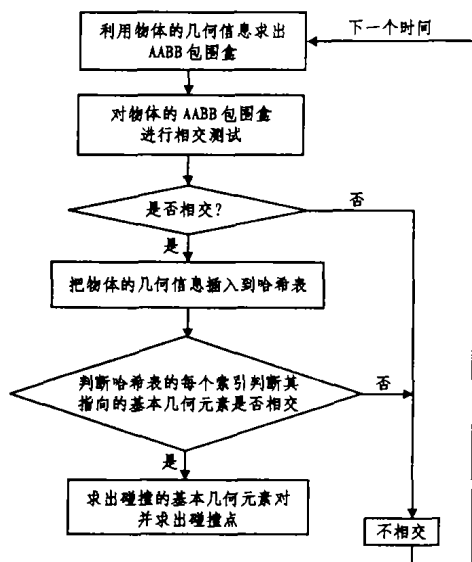


图2 算法的流程图

由于此算法的背景是虚拟手术仿真,因此假设只有两个物体,分别为手术刀与人体组织。在第一阶段首先分别求出两个物体的 AABB 包围盒,注意不是建立包围盒树,这也是本文题目用包围盒而不是层次包围盒的方法。如果它们的 AABB 包围盒不相交,则可以立即判断手术刀与人体组织不发生碰撞。如果 AABB 包围盒相交,则进入第二阶段即空间分解法,这也是此算法的关键部分。

本文的空间分解法不是对整个空间进行分解,而是对一个局部的空间进行分解(假定两个物体运动的范围不会超出此空间)。记此空间为 S ,把 S 划分成边长为 $size$ ($size$ 大小的确定后面会论述)的空间网格(正方体)。下一步须对这些网格进行编号,这里取三个整数 (i, j, k) 依次代表 x, y, z 轴来唯一地标识一个网格。紧接着需要将基本几何元素映射到空间网格中去,直接映射可能比较复杂,我们采取的是一种近似方法,即先求出基本几何元素的 AABB 包围盒(计算量非常小)。由于 AABB 实际上是一个矩形,它可以由主对角线上的两个点唯一标识,因此我们记这两个点为 p_1, p_2 。只要求出这两个点所占据的空间网格,则此基本几何元素所占的空间网格可以非常容易地求出。设 p_1, p_2 的坐标分别为 $(x_1, y_1, z_1), (x_2, y_2, z_2)$ 。 p_1 所在的空间网格为 (i_1, j_1, k_1) , i_1, j_1, k_1 可以由下式求出。 $i_1 = \lfloor x_1/size \rfloor, j_1 = \lfloor y_1/size \rfloor, k_1 = \lfloor z_1/size \rfloor$,同理 p_2 所占据的空间网格 (i_2, j_2, k_2) 也可以按同样的方法求出。这样基本几何元素所占据的空间网格就是 (i_1, j_1, k_1) 和 (i_2, j_2, k_2) 之间所有的连续网格。

为了防止对每个空间网格进行判断并且为了方便查询,

这里采用了哈希表的数据结构。由于哈希表是一维表,每个索引实质上存储的是基本几何元素的指针链表的头指针,因此我们必须把三维的空间网格映射到一维的哈希表中。设 h 为哈希表的索引,空间网格为 (i, j, k) ,哈希表的大小为 $length$ (后面会讲到它的取值)。定义哈希函数为 $h = (p_1 \times i + p_2 \times j + p_3 \times k) \% length$,其中 p_1, p_2, p_3 为三个大素数,在我们的算法中 $p_1 = 2000689, p_2 = 8000981, p_3 = 5123693$ 。这样做的目的是使空间的网格尽可能地均匀映射到哈希表中去。当两个基本几何元素相交时,它们必定在某个相同的空网格发生碰撞,这样它们必定映射到同一个哈希表的索引中去。因此当我们逐个扫描哈希表的索引时,就可以把所有的基本几何元素对求出,并且这种方法能够很好地解决自相交问题。最后对相交的基本几何元素运用 T. Möller^[9]所提出的算法找出精确的碰撞点。

设计哈希表时有三点问题需要考虑,第一是哈希函数能否决定空间网格均匀映射到哈希表中去。第二是哈希表的大小,设置得太小,会使太多的基本几何元素映射到同一网格中;设置得太大,既浪费空间,又在扫描哈希表的整个索引时明显耗时,进而影响碰撞检测的效率。在本算法中取哈希表的大小为 $3n$,其中 n 为基本几何元素的个数。在实验中证明 $3n$ 是一个比较理想的数值。三是空间网格的大小,由于实验采用的基本几何元素是四面体,本文取 $size$ 为四面体各个边长的平均值。

2.2 哈希表的更新

当手术刀和人体器官发生位移时需要对哈希表进行更新。在这里分两种情况,第一是拓扑结构不发生改变的情况,第二是当拓扑结构发生改变的情况。在第一种情况下,由于在相邻的时间片只有少量基本几何元素发生位移,并且位移量通常比较小,因此不能对整个的哈希表进行更新,那样就使时间耗费太大而不能满足碰撞检测的实时性。本文算法中是在基本几何元素中加了一个布尔变量 $isChanging$,用于指示基本几何元素所在的空间网格是否发生改变。因此在下一个离散时间段首先计算每个基本几何元素的 AABB 包围盒(很短时间可以完成),当包围盒所占据的空间网格不发生改变时就不需要进行更新,此时 $isChanging$ 设置为 $false$ 。相反当占据的空间网格发生改变时,此时 $isChanging$ 设置为 $true$ 。在这种情况下必须首先把基本几何元素在上一时间片断中在哈希表中的信息去除掉,这可以通过哈希函数首先找到哈希表的索引,然后可以沿着此索引所指向的链表来快速地找出此基本几何元素并删除。在哈希表设计比较好的情况下,一个索引所指向的链表长度通常很短,因此这一步在很短时间即可完成。下一步需要把此基本几何元素在下一个时间片断的信息插入到哈希表中去,由于此时已经求出下一时间片断此基本几何元素所占据的空间网格,因此首先利用哈希函数查找其在新位置下的索引,然后为了加快时间把此基本几何元素插入到该索引指向链表的头部。这样在一个时间片断内只有极少量的基本几何元素在哈希表中的位置需要更新。

当拓扑结构改变时又分为两种情况^[10],一是基本几何元素的减少(比如虚拟手术中人体组织的灼烧),二是某个基本几何元素破裂成新的基本几何元素导致基本几何元素数目的增加(比如虚拟手术中切割某组织导致该组织的破裂)。在本文所采用的哈希表的数据结构中理论上能够解决拓扑结构改变的情况。哈希表支持三种最基本的操作^[11],分别是查询(search),插入(insert),删除(delete)。其中插入和删除操作在最坏情况下的时间复杂度为 $O(1)$,而查询操作当哈希表的大小与所要存储的元素个数成正比时(本文中为3),查询操作在

平均情况下为 $O(1)$ 。当第一种情况发生即基本几何元素的减少对应着对哈希表的两个操作查找以及删除,即首先在哈希表中查找到该基本几何元素然后从链表中删除它。很明显可以在 $O(1)$ 常数时间完成。当第二种情况发生即当一个基本几何元素分裂成有限个基本几何元素时,对应着三个哈希表操作查找以及删除和插入,即首先在哈希表中找到该要分裂的基本几何元素,然后从其所在的链表中删除,最后把其新生成的基本几何元素插入到相应位置,同样也只需 $O(1)$ 常数时间就能完成。

2.3 复杂度分析

复杂度分析包括时间复杂度分析和空间复杂度分析,并给出它和层次包围盒方法的比较。假设所有的基本几何元素个数为 n ,则本算法中哈希表的长度取为 $3n$,设哈希表的每个索引所指链表的平均长度为 p 。利用聚集分析(aggregate analysis),则在每个时间片断,需要检测的基本几何元素对的个数为 $3n \times \binom{p}{2}$,当哈希表设计较好的情况下, p 是一个较小

的常数,因此 $\binom{p}{2}$ 也是一个常数,而检测基本几何元素是否碰撞所需的时间为 $O(1)$,因此时间复杂度为 $\Theta(n)$ 。而在层次包围盒方法中(假设采用的是二叉树数据结构),基本几何元素为 n 意味着叶子节点数为 $n^{[12]}$,分析表明,二叉树的平均深度为 $O(\sqrt{n})$,由于需要检测的基本几何元素都位于叶子节点,因为需要检测的基本几何元素的次数与二叉树的深度成正比,因此层次包围盒法的平均时间复杂度为 $O(\sqrt{n})$ 。在平均情况下要优于本文的算法,但是 $\Theta(n)$ 的时间复杂度在理论和实践中完全可以接受。

在空间复杂度分析中设所采用的机器为32位长,此时即指针占内存4个字节。在本文的算法中哈希表的索引存储着一个指向链表的指针,因此共占去的字节数为 $4 \times 3n = 12n$ (其中 n 为基本几何元素的个数, $3n$ 为哈希表的长度)。每个链表节点由两部分组成,一部分指向基本几何元素,另一部分指向下一链表节点,因此链表部分共占去的字节数为 $(4+4) \times n = 8n$ 。因此本算法所占去的空间总共为 $20n$ 。在层次包围盒方法中所占去的内存空间随所采用的包围盒类型的不同而不同,其中最节省空间的是包围球。当采用包围球方法时,因为它只需存储球的中心和球的半径,设采用的是 float 类型,中心 x, y, z 坐标共占据12字节,半径占据4个字节,因此基本的包围球占据16个字节的内存空间。当基本几何元素的个数为 n 时,一棵满二叉树共有 $2n-1$ 个节点,其中叶子节点的个数为 n ,内部节点的个数为 $n-1$ 。当节点的类型为内部节点时,节点内部除了存储基本的包围球外,还有两个指向左右孩子的指针,因此共占据的字节数为 $(n-1) \times (16+8) = 24(n-1)$ 。当节点的类型为叶子节点时,节点内部除了存储基本的包围球外,还有一个指向基本几何元素的指针,因此共占据的字节数为 $(16+4) \times n = 20n$ 。因此整棵树所占据的内存空间为 $24(n-1) + 20n$,当 n 较大时约等于 $44n$ 。因此在空间复杂度方面,本文算法所占据的内存空间要远小于层次包围盒方法,从某种程度上抵消了在时间复杂度方面的劣势。

3 实验分析

实验采用的是虚拟手术中手术刀和人体组织手臂发生碰撞的情形见图3。在实验中采用的基本几何元素为四面体,采用体模型的好处是可以给出精确的物理响应。在实验中精确实现了当手臂发生变形时手术刀和手臂发生碰撞的情况并给出了碰撞结果,在碰撞结果中不仅求出了发生碰撞的基本几

何元素对,而且还求出了碰撞点。碰撞点的信息放在一个 collide.txt 的生成文件中,为虚拟手术中后面的切割阶段提供信息。但是本实验暂时还没有实现当拓扑结构改变的情形,这将在以后完善。

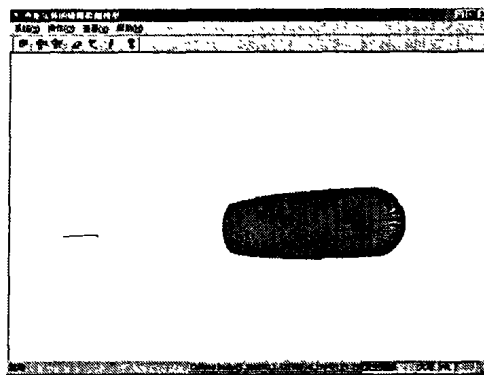


图3 虚拟手术实验

结束语 本文提出了一种基于包围盒方法和空间分解法相混合的碰撞检测算法,即首先用物体的 AABB 包围盒来快速排除不相交的物体对,然后对可能发生的物体运用空间分解法来精确地求出发生碰撞的区域。运用空间分解法的优点是很容易地解决自相交问题,并且在空间分解法中采用了一种灵活的数据结构即哈希表,利用哈希表所支持的三种操作在理论上能够解决发生碰撞时拓扑结构改变的情形。本文解决了变形体发生变形时和刚体发生碰撞的情况,并精确地求出了碰撞点,为下一阶段的切割奠定了基础。但本文暂时还未实现当物体的拓扑结构改变时发生碰撞的情况,因此下一步研究的重点在于结合恰当的切割算法来实现拓扑结构改变时的碰撞检测问题。

参考文献

- 1 van den Bergen G. Efficient Collision Detection of Complex Deformable Models using AABB Trees. Journal of Graphics Tools, 1997, 2(4): 1~14
- 2 kitamura Y, Smith A, Takemura H, Kishino F. A Real-Time Algorithm for Accurate Collision Detection for Deformable Polyhedral Objects. Presence, 1998, 7(1): 36~52
- 3 Mezger J, Kimmmerle S, Hierarchical E O. Hierarchical. Techniques in Collision Detection for Cloth Animation. Journal of WSCG11, 2003, 2: 322~329
- 4 Klosowski J T, Held M, Mitchell J S B, et al. Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs. IEEE Transaction on Visualization and Computer Graphics, 1998, 4(1): 21~36
- 5 Teschner M, Heideleberger B, Muller M, et al. Optimized Spatial Hashing for Collision Detection of Deformable Objects Munish. Germany, Nov. 2003. 19~21
- 6 Baci G, Wong W S K. Hardware assisted self-collision for deformable surfaces. In: Proc. of ACM Symposium on Virtual Reality Software and Technology, ACM Press, 2002, 13: 129~136
- 7 Heidelberger B, Teschner M, Gross M. Detection of collisions and self-collisions using image-sapce techniques. In: Proc. of WSCG'04, 2004. 13~15
- 8 Heidelberger B, Teschner M, Gross M. Real-time volumetric intersections of deforming objects. In: Proc. of Vision, Modeling, Visualization VMV'03, 2003. 461~468
- 9 Möller T. A fast triangle-triangle intersection test. Journal of Graphics Tools, 1997, 2(2): 25~30
- 10 阎丽霞. 虚拟手术关键技术研究: [浙江大学博士论文]. 2001
- 11 Cormen T H, Leiserson C E, Rivest R L. Clifford Stein Introduction to Algorithm MIT Press second edition, 2001. 221~251
- 12 Weiss M A. Data Structure and Algorithm Analysis in C. Addison Wesley Press, 1997. 95~96