

简化球体的 BSP 剖分结构的快速碰撞检测

于晓霞,沈志刚

(中山职业技术学院 计算机工程系, 广东 中山 528404)

摘要: 在动态场景中, 碰撞检测遇到的最明显的问题就是需要对 N 个物体对进行两两求交测试, 其时间复杂度达到 $O(N^2)$ 。提出了一种简化球体的 BSP 剖分结构的快速碰撞检测算法。首先用一种调度算法估计 BSP 树开始失衡的地方, 再用一种策略来选择分割面, 新结构在表示动态场景中不需要重新构建树, 而是在保持树的平衡和合理的高度的同时通过自身的调节来达到更新。

关键词: 碰撞检测; 动态场景; BSP 树

中图分类号: TP18 **文献标识码:** A

Detection of simplified sphere BSP-trees structure for broad-phase collision

YU Xiao Xia, SHEN Zhi Gang

(Department of Computer Engineer, Zhongshan Polytechnic College, Zhongshan 528404, China)

Abstract: The broad-phase step of collision detection in scenes composed of n moving objects is a challenging problem because enumerating collision pairs has an inherent $O(N^2)$ complexity. In this paper we propose a new structure based on BSP tree for representing scenes composed of thousands of moving objects. An scheduling algorithm evaluates locations where the BSP-tree becomes unbalanced, uses several strategies to alter cutting planes, and defer updates based on their re-structuring cost. We show that the tree does not require a complete re-structuring even in highly dynamic scenes, but adjusts itself while maintaining desirable balancing and height properties.

Key words: collision detection; dynamic scenes; BSP-trees

在动态场景的模拟中, 当场景中物体的个数超过 2 个, 碰撞检测遇到的最明显的问题就是需要对所有 N 个物体进行两两求交检测, 其时间复杂度达到 $O(N^2)$ 。为此, 加快这种检测速度通常是用两步算法^[1]。所谓两步法, 就是在第一步(也称初步检测阶段), 首先将多数明显不相交的物体对进行快速排除, 找出潜在的相交区域或潜在的相交物体对; 然后在第二阶段根据已经确定的潜在的相交区域或潜在的相交物体对几何做进一步的相交测试, 这一步也称详细检测阶段。在处理动态场景初步检测时, BSP 树算法在快速排除明显不发生碰撞的物体不够理想, 为此本文提出了一种基于 BSP 结构的快速碰撞检测算法。该算法是在 BSP 树结构的更新操作时, 定义了 5 种操作算子, 并通过一种调度策略执行算子, 保持 BSP 树结构的平衡。

1 BSP 树的构建

BSP 树是最常用的空间剖分技术, Fuchs 于 1980 年

首次将 BSP 技术中剖分平面的定侧性质应用于多边形场景的剖分^[2], BSP 树是一个用来对 n 维空间内多面体进行排序和查找操作的标准二叉树。这个树代表了整个场景, 每一个叶节点代表着场景的 1 个凸集多边形集合, 每一个非叶节点包含一个分割平面, 这个分割平面将 1 个子场景分成 2 个更小的场景。BSP 树的构造是这样一个过程, 已有一个子场景, 用这个场景中 1 个多边形所在的平面从内部分割这个场景, 结果又得到 2 个子场景然后继续循环分割它们, 直到所有子场景都是一个凸集多边形集合为止。BSP 树就是一个二叉树, 叶节点存储场景多边形, 非叶节点存储分割面。图 1 所示, 为二维平面上的 BSP 树结构剖分。

BSP 在计算碰撞检测时有巨大的优势, 它能够很容易定位运动物体在 BSP 树的具体位置, 即场景中的具体位置。定位完成后, 只有很少数目的多边形需要检测, 即每一帧里只检测物体是否穿过那些组成它所在叶节

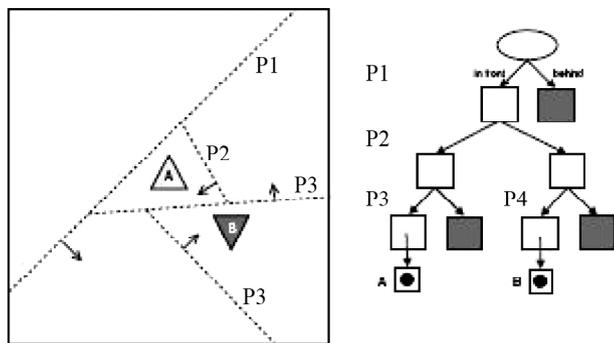


图1 二维平面上的BSP树结构

点的多边形^[3-4]。

2 简化球体BSP剖分结构的分割面选取策略

当创建一个BSP树时, 决定是否需要一个平衡树, 也就是说树的左右分支的深度的差异不应该太大, 由于每一次分割都会产生新的多边形, 因此应尽量减少分割的次数。如果在BSP树的创建过程中产生了太多的新多边形, 显示卡就要花更多的时间来处理多边形的渲染, 使一个不平衡的树将用更长的时间来进行树的遍历。因此在选择分割面时, 应该根据下面的标准(p 为分割面):

population(p)由分割面所生成节点所包含物体的数目。

balance(p)被分割面分割成的左右子树深度的比率。

redundancy(p)横跨分割面的物体的数目。

即在选择分割面时, 需要综合考虑以上3个条件, 选择balance最大、redundancy最少的分割面。

分割面的选取方法一般采用直接选择平面法, 是指从表示场景的多边形集合中随机选出一个多边形, 假定该多边形在空间范围内无限延展, 则平面将空间分成了2个子空间。在各自2个子空间内, 原有的场景多边形分别位于其中, 对于某一子空间, 再从其中的多边形集合中任意选取1个, 作为该子空间的超平面。这一过程递归进行, 直到最终的子空间中只有唯一的景物为止。图2给出了这一过程的图形示意。

图2中, 把各线段看作是垂直于纸面的平面, 箭头为该平面的法向量, 箭头所指方向为该平面的前面(可见一侧), 其相反方向即为该平面的后面(不可见一侧)。图2(a)给出了初始的场景多边形集合以及每个多边形的法向量, 在图2(b)中, 首先选取面1为分割面, 将空间分为A、B两个子空间; 在B子空间中选取面4为分割面, 将B分为C、D两个子空间; 在D子空间中选取面2为分割面, 将D分为E、F两个子空间。至此, 整个空间划分完毕, 每个子空间中只含有一个景物。

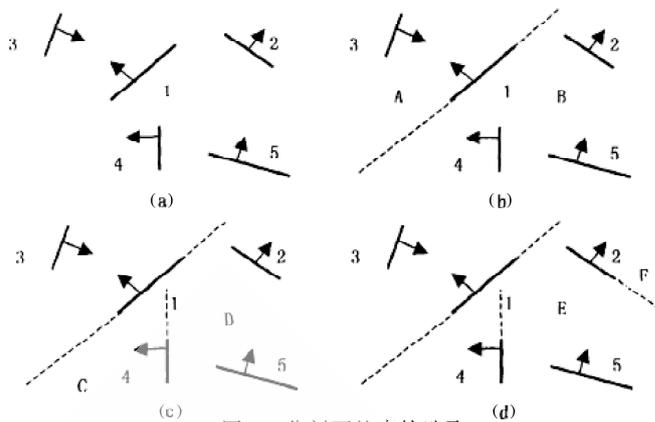


图2 分割面的直接选取

3 简化球体BSP剖分结构的更新操作

由于要保持动态物体在动态场景中的高效率的运动和一定的搜索特性, 导致了BSP树结构的改变。如果重新构造树, 额外的开销将严重影响碰撞检测的效率, 因此, 希望在原来树的基础上通过局部更新得到新的树。在这种情况下, 更新树的过程时间开销就相当重要, 如果更新树所花费的时间和完全重建差不多, 更新就没有什么意义了。更新操作必须能够实时完成。本文采用的方法是根据动态物体的物理位置把它插入到BSP树的相应节点下, 用BSP树逻辑操作修改BSP树的结构。下面将介绍是如何被调度来执行更新的。

3.1 分裂操作算子

当叶节点包含物体数量(population)大于给定的叶节点包含物体数量的阈值时, 将该节点向下分裂, 如图3所示。这样, 一个新的分割面产生1个节点, 同时生成两个左右子树。对于产生内部节点的分割面的选取按照上一部分提到的分割面的选择方法。对物体在候选分割面法线上的投影进行分类支配着这个操作的开销, 为了减少这种开销只考虑这些物体1个子集。

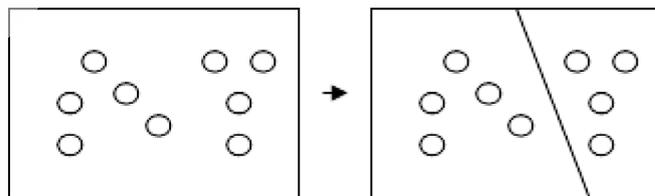


图3 分裂操作

3.2 移除分裂操作算子

这个操作通过用存储在BSP树中的叶节点的信息来降低因分类而导致的分裂操作的开销。

在BSP树的构造过程中, 每个叶节点将被整个放入表中。也就是从根节点(内节点)开始在节点上沿着它的父分割面的法线放置物体。在这张表中可以找出另一个法线方向与父分割面法线方向相平行的分割面。如果这个新的分割面满足分割面选择标准, 就可以用移除分裂操作来代替分裂操作, 如图4所示。

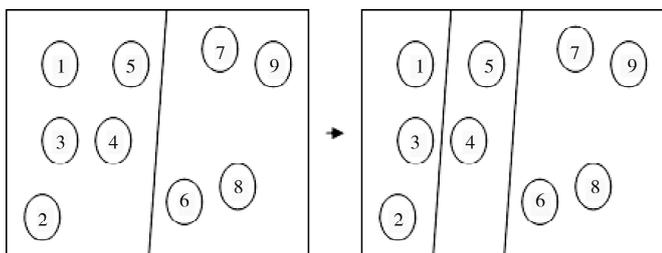


图4 移除分裂操作

3.3 合并操作算子

合并操作算子的作用是：当一个叶节点连同它的兄弟所含物体数量之和还没超过阈值时，将它们向上合并。而且当由于物体的运动使分割面分割的区域出现空值时，它负责重新移动此分割面，(如图5)。

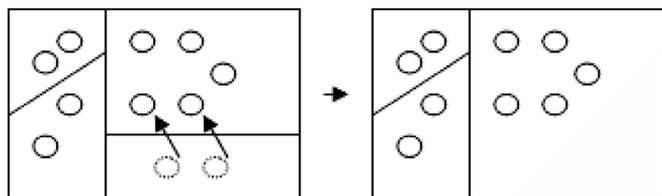


图5 合并操作

所有存储在合并节点的子树的叶节点的物体的有序表需要合并成一个。左右子树的节点自下而上合并，并且叶子节点代替原来的节点，如图6所示。每一个表需要沿着新的方向分类。当合并节点包含物体数量很少时，合并节点的子树通常也很小。

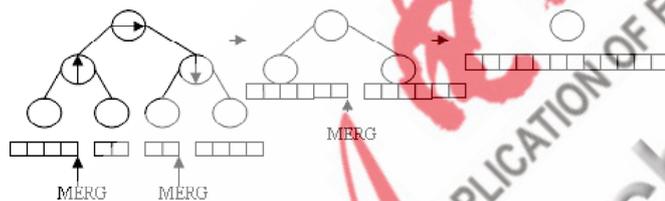


图6 表的合并自下而上

3.4 平衡操作算子

平衡操作算子的作用是：在原来树的基础上通过最小的改变来修复开始变得不平衡的节点。它适合于左右子树的深度差异很大的节点，即1个节点的1个子树所含物体数量很多而另一个子树几乎是空的。在这种情况下，利用平衡操作来取代合并子树操作，并通过移动分割面重新建立平衡，如图7所示。如果这个移动的分割面满足选择分割面的标准，可利用平衡操作来修改这个节点。

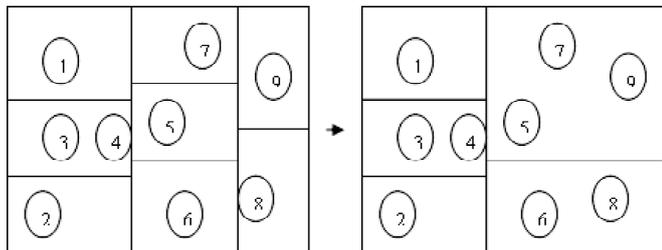


图7 平衡操作

3.5 交换操作算子

交换操作算子的作用是：它适合于当平衡操作失败于重新建立平衡时的情况，在合并操作之前使用交换操作。交换操作删除不平衡的节点，用包含物体数量多的子树的根节点代替删除的不平衡节点结构的调整所需要的只是将包含物体数量少的子树插入，但这个操作很快，因为这个子树几乎是空的，如图8所示。

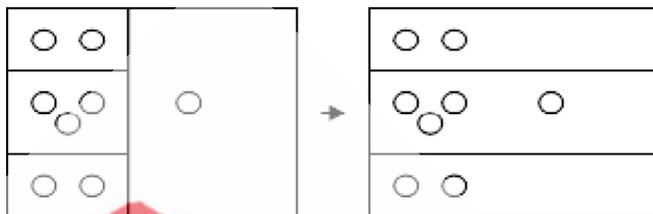


图8 交换操作

4 简化球体BSP剖分结构更新操作调度策略

BSP树结构的更新不同于物体位置的更新，因为树结构更新的时间开销很大，而且结构更新的好坏会导致碰撞检测对数目的增减，如果结构更新失败就会导致碰撞检测对数目的增加。定义的更新是在规定时间间隔内达到一个折中，即在更新的时间开销和树的高度之间。结构操作执行的时间顺序是至关重要的。更新BSP树结构的结构操作的调度策略说明如下：

平衡操作是通过调整分割面来改善平衡，但是它会产生改变物体分布的副作用。这可能影响其他操作，在某些情况下可能变得不必要了。因此，在其他操作之前应该先考虑平衡操作，如图9所示。

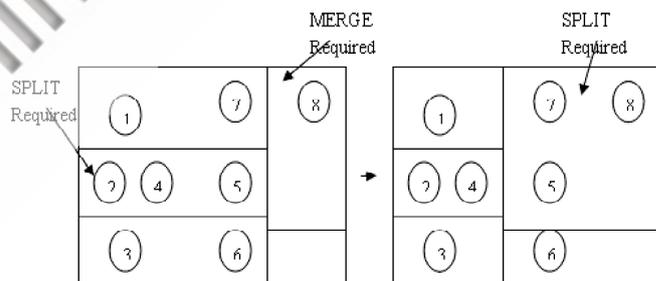


图9 平衡操作对分裂合并的影响

分裂操作是非常关键的，因为它是要把1个大的存放物体信息的顺序表分成2个小表，以降低更新顺序表的代价。首先是集中在所包含物体数量小于阈值的叶节点上，也就是通过对场景中的物体数量动态调整叶节点。而移动分裂操作是在分裂操作之前执行，也就是只有当移动分裂操作失效时才开始执行分裂操作。

当平衡操作失效时，交换操作开始执行。这个操作也如同平衡操作一样会产生副作用，通常导致小数量的物体移动。

合并操作是消除包含物体数量小于阈值的叶节点。这个操作不是很重要，因为对于小的顺序表的维持是

技术与方法 Technique and Method

很快的，而且额外的开销是保持几乎是空的节点和最小限度的叶节点。只有在模拟的更新周期期间有自由的时间，这个操作才会被执行。

更新 BSP 树结构的结构操作调度策略如下：

开始遍历根节点为 N 的 BSP 树：

- (1) If 节点 N 是不平衡节点；
- (2) then 评估平衡操作算子；
- (3) If 平衡算子有效执行平衡操作；
- (4) Else 调用交换算子；
- (5) 停止遍历；
- (6) End if ；
- (7) 检查移动分裂操作，分裂操作，合并操作和调度时间表；
- (8) if 所有事件已调度，停止遍历；
- (9) Else 重复执行 1 ~ 8 步遍历节点 N 的每个子树；
- (10) End if ；
- (11) 根据调度时间表估算被执行的事件数目；
- (12) 执行移动分裂操作；
- (13) 执行分裂操作；

(14) 执行交换分裂操作；

(15) 执行合并分裂操作。

5 实验结果与分析

为了验证 BSP 算法的性能，与 QT Quad(Tree), Spatial Hash(SH)、LO Loose(Octrees), SP Sweep-and-(Prune)这 4 种算法进行了一系列比较实验。所有的实验都是在 PIV2.5GHz、512MB 内存的 PC 机上进行的。

每个模拟实验按每秒 25 步被评估 4 min。实验评估包括执行效率(FPS)，碰撞检测对的数目，初步碰撞检测阶段的时间，更新的时间(物体位置的更新时间 and 数据结构的更新时间)。

实验物体落入有限制的区域中，然后跟区域中的其他物体相互碰撞，这是利用了动态和静态物体之间的空间分割方法。因为静态物体对于提高执行速度是至关重要的。这种情景最适合 BSP。BSP 的碰撞检测时间最少，QT 次之。SP 的碰撞检测时间最坏但是更新时间跟 BSP 差不多。LO 更新时间最佳。SH 的更新时间最坏。

实验测试结果分别如图 10、图 11、图 12、图 13 所示。

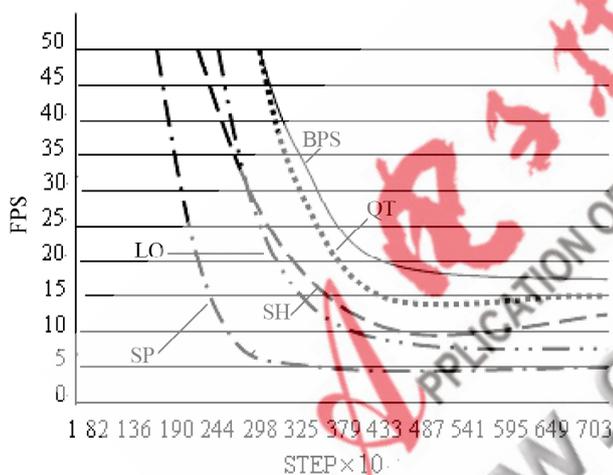


图 10 执行效率

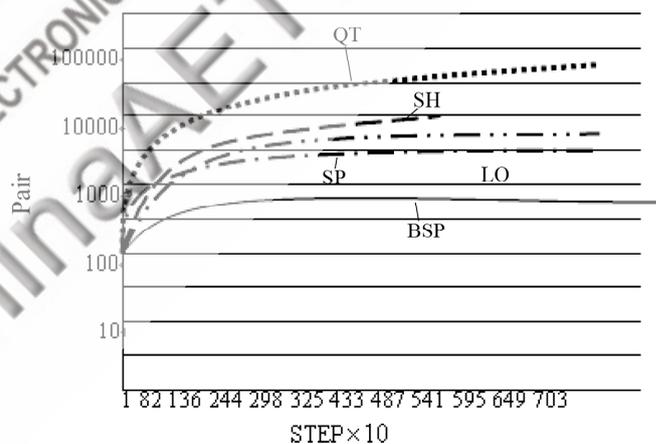


图 12 碰撞检测对数目

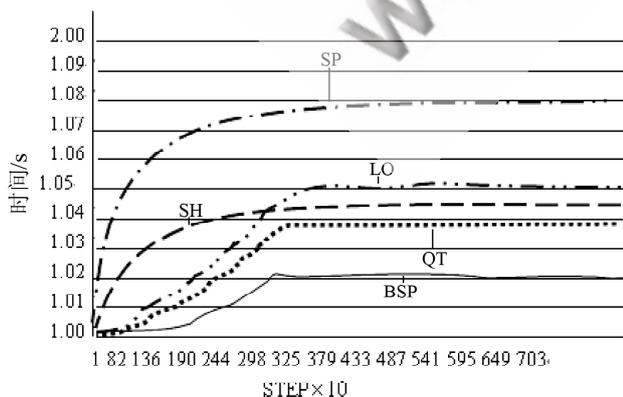


图 11 碰撞检测时间

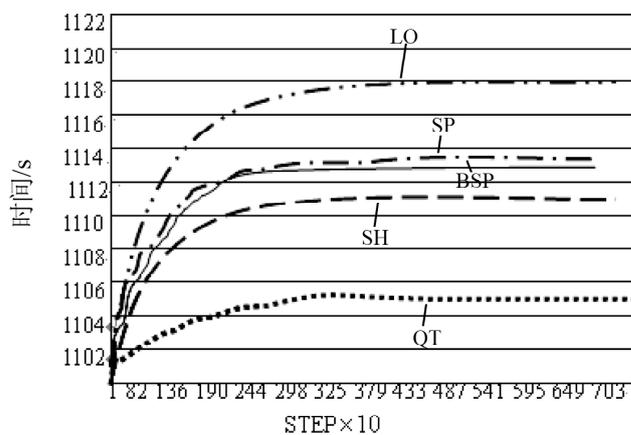


图 13 更新时间

(下转第 75 页)

技术与方法 Technique and Method

线路继续供电；即使两路电源均出现故障，大功率的UPS电源仍可继续延时供电。通过先进的冗余设计确保系统安全运行。服务器采用了双机热备方式，同时采用了大容量磁盘阵列、磁带机；每半小时对系统数据自动完成备份，同时可实时根据需要进行数据备份。

一卡通是数字化校园的基础工程，是推行数字化校园的有力载体。一卡通的内容将随需求愈加充实，其构成亦愈加丰富。一卡通的功能可以渗透到学校的教学、科研、管理和后勤等多方面应用，并随着科技进步和管理内容的增多而加以深化，具有广阔的发展前景。但是，一卡通涉及的是使用者的多方利益，必须考虑集成中的各种风险，只有慎重考虑，充分平衡各方利益和权力，稳妥实施，才能取得预期的效果。

(收稿日期：2008-11-20)

(上接第68页)

射、尾罩侧推和点火过程仿真，场景真实，符合该导弹发射的实际。同时，效果流畅，人机交互自然，再配以音响产生实时音效，有一种身临其境的感觉，更增加了直观效果。导弹武器发射环境视景仿真系统使得导弹实装操作、训练和观摩简单易行，实施方便。

本文在计算机软硬件平台上，应用虚拟仿真技术，构建了导弹发射环境视景仿真系统。虚拟视景系统具有高效性、可控性、无破坏性、使用灵活性、系统运转费用低等特点，能够突破传统训练方式的局限，既对号手进行导弹操作模拟训练，取代实弹训练中的部分科目和内容，又改变了传统的人教人、师带徒的落后教学方式，提高了训练质量，同时，对下一步导弹虚拟训练系统的开发具有很好的借鉴意义。虚拟导弹训练系统

参考文献

- [1] 张升平. 数字化校园之一卡通的建设. 重庆工商大学学报(自然科学版), 2008, 24(1).
- [2] 卢璐, 姚水林. 基于校园网的一卡通系统设计和应用. 科技情报开发与经济, 2006, 16(24).
- [3] 段智敏等. 基于一卡通的数字化校园资源整合研究与实现, 计算机工程与科学, 2008, 30(1).
- [4] 裴沛. 基于一卡通模型的数字化校园的构建研究. 金卡工程, 2006(11).
- [5] 林葱, 蔡秀珊. 校园一卡通系统的总体设计和实现, 科技情报开发与经济, 2007, 17(3).

的开发应用为导弹部队提高和掌握导弹武器系统操作、迅速形成战斗力提供了新的方法。

参考文献

- [1] 吴家铸, 刘华峰, 程志全, 等. 视景仿真技术用度应用. 西安: 西安电子科技大学, 2001.
- [2] 刘玉海, 李瑞, 张锡恩. 导弹虚拟操作训练系统的设计与实现. 飞航导弹, 2002, (8).
- [3] 和平鸽工作室. OpenGL高级编程与可视化开发系统开发篇(第2版). 北京: 中国水利水电出版社, 2007.
- [4] 吴宾, 毕丽蕴. OpenGL编程实例与技巧. 北京: 人民邮电出版社, 1999.
- [5] WRIGHTR S, LIPCHAK J B. OpenGL 超级宝典(第2版). 徐波, 译. 北京: 人民邮电出版社, 2005.

(收稿日期：2008-11-21)

(上接第72页)

本文所介绍的方法主要目的是当碰撞检测对数目增加时，在计算碰撞检测的代价和在空间数据结构中的结构更新代价两者之间建立一个折中的方法。实验结果表明，在初步碰撞检测阶段与SH、SP和LO方法相比，本文的方法是很有有效的。虽然在碰撞检测对数目上与QT相似，但与QT相比，文中的方法对静态物体导致的空间分割模拟或者是高度集中的环境中产生太多碰撞的模拟是比较好的。文中各个更新BSP树结构的结构操作和调度策略的设计可以更好地重新利用存储在树中的信息来重新定位分裂面和稳定的执行效率。初步碰撞检测只是这种方法的一个应用，用这种方法来加快其他几何体的碰撞是今后要研究的另一个方面。

参考文献:

- [1] HUBBARD P M. 1995. Collision detection for interactive graphics

applications. IEEE Transactions on Visualization and Computer Graphics, 1995, 1(3): 124-133.

- [2] FUCHS H, KEDEM Z M, NAYLOR B F. H1 On visible surface generation by a priori tree structure [J]. Computer Graphics, 1980, 14(3): 124-133.
- [3] AR S, MONTAG G, TAL A. Deferred, self-organizing BSP trees. Computer Graphics Forum, 2002, 21(3): 269-278.
- [4] JAMES D L, PAID K. BD-Tree: Output-sensitive collision detection for reduced deformable models. ACM Transactions on Graphics (SIGGRAPH), 2004, 23(3).

(收稿日期：2008-11-19)