

# 基于 Kinect 和 Unity 的虚拟手抓取碰撞检测算法

赵美荣, 田卫萍, 熊风光, 韩 雯

(中北大学 计算机与控制工程学院, 太原 030051)

**摘要:** 针对目前虚拟现实碰撞检测算法效率低精度差的问题, 提出了一种改进的层次方向包围盒(OBB)算法; 首先通过位置关系判断虚拟手是否在物体包围球的邻近区域, 然后用 OBB 和八叉树算法进行详细的碰撞检测, 最后利用离散点到虚拟手简化面的矢量计算法实现精确的碰撞检测; 实验结果表明, 随着三维物体基元数目的增多, 这种由粗略到精确递进的检测方式极大地提高了碰撞检测的效率和精度, 具有可行性; 该算法适用于任何复杂场景中刚体结构模型的碰撞检测, 在运行时候不存在滞后情况, 显示流畅, 而且碰撞检测精度高, 完全能够满足虚拟环境实时性和精确性的要求。

**关键词:** 抓取; 碰撞检测; 有向包围盒; 简化面; 包围球

## Collision Detection Algorithm of Virtual Hand Grasping Based on Kinect and Unity

Zhao Meirong, Tian Weiping, Xiong Fengguang, Han Xie

(Computer and Control Engineering, North University of China, Taiyuan 030051, China)

**Abstract:** For the problem of the low efficiency and the poor precision of collision detection algorithm in the virtual reality at present, a new improved hierarchical oriented bounding box (OBB) is developed. First, the the algorithm use positional relationship to judge whether virtual hand is in the nearby place of object's bounding box, then use the OBB and octree algorithm to perform the detailed collision detection. At last, the tangent plane vector calculation method which is from discrete point to tangent plane of virtual hand is used to realize the accurate collision detection. Experimental results show that, with the increasement of the object primitives, the efficiency and accuracy is greatly improved by means of this progressive mode from coarse to precise, and it is feasible. The algorithm is applicable to collision detection of rigid model in any complex scenes, it does not exist lag and shows smooth when the system is running, and it's accuracy is high. It can be fully able to meet the requirement of real-time and accuracy in the virtual environment.

**Keywords:** grasp; collision detection; OBB (Oriented Bounding Box); simplified plane; sphere box

## 0 引言

虚拟现实是利用计算机仿真技术为用户创造虚拟三维世界的人机交互技术, 虚拟手交互是其中最重要的交互模式。在虚拟手交互中, 为了更好地保证交互的真实性和沉浸感, 虚拟手和三维模型之间实时精确的碰撞检测是研究的重要内容。

层次包围盒算法是一种常用碰撞检测算法<sup>[1-2]</sup>。传统的包围盒算法存在计算效率低、精度差的缺点, 针对以上问题, 前人进行了一定的改进研究。文献 [1-2] 提出了基于八叉树的层次 OBB 算法, 这种方法在一定程度上能提高算法的效率, 但还是存在显示不流畅、运行滞后的现象。文献 [3] 提出利用空间剖分法进行视域的快速剔除, 但是该方法只适用于分布均匀的稀疏环境<sup>[4]</sup>, 对于大规模复杂场景效率低下。

本文通提出了一种新的碰撞检测算法。首先通过位置关系

判断虚拟手是否在物体包围球的邻近区域, 进而快速找到潜在的相交区域。其次进行精确碰撞检测, 提出一种离散点矢量计算法, 解决三维物体由于基元不同而给相交测试带来的问题; 从面到面的计算简化为点到面的计算, 计算量减少。

## 1 基于八叉树的层次包围盒算法

### 1.1 包围盒算法

包围盒算法用体积比原物体略大且几何特性简单的包围盒来近似描述复杂的三维模型<sup>[5]</sup>。沿坐标轴的包围盒(AABB)计算量少, 紧密性差; 包围球算法测试简单, 计算量比 AABB 大, 而且紧密性更差。这两种算法都会导致大量的冗余。方向包围盒(OBB)计算量大, 但紧密性非常好, 能成倍地减少相交测试的包围盒的数目。除此之外, 当三维物体发生旋转时, 只要改变 OBB 的基底即可<sup>[6]</sup>。

### 1.2 八叉树层次结构

八叉树是将三维数据集划分到不同包围体层次的树状数据结构。它的每个节点表示一个长方体的体积元素, 每个节点或者有八个子节点, 或者没有子节点。

假设根节点是第 0 层, 那么每 i 层最多有  $8^i$  个节点, 如果这棵八叉树有 m 层, 那么这棵树最多有  $(\sum_{i=0}^m 8^i)$  个节点。

八叉树层次结构可以将三维模型的各小部分进行有效管

收稿日期:2015-01-19; 修回日期:2015-03-09。

**基金项目:** 国家科技支撑计划基金(2013BAH45F02)国家自然科学基金项目(61379080)。

**作者简介:** 赵美荣(1989-), 女, 山西吕梁人, 硕士研究生, 主要从事仿真与可视化方向的研究。

韩 雯(1964-), 女, 山西吕梁人, 博士, 教授, 主要从事仿真与可视化、智能信息处理方向的研究。

理,通过树状结构的遍历方式对各节点实施相交测试,从而无需对所有的基元进行两两相交测试,可以减少碰撞检测的次数。

### 1.3 层次包围盒算法

将包围盒算法和八叉树层次结构结合起来,即为基于八叉树的层次包围盒算法。该算法基本思想是:首先用包围盒来近似描述复杂的三维模型,然后构造树状的层次结构来逼近这个模型。在对两个三维模型进行碰撞检测的时候,先对它们的根节点进行相交测试,若发生干涉,对它们的子节点进行详细的碰撞检测,否则无需对其子节点进行进一步地检测。求包围盒的交比求原物体的交简单得多,这样就快速地排除了许多不相交的部分,提高了算法的效率。

## 2 改进的碰撞检测算法

传统的基于八叉树的 OBB 算法首先使用空间剖分法将整个虚拟场景划分为若干个单元,每个物体模型占据一个或者连续的几个单元,碰撞检测只是对占据了一个单元的物体进行检测。然后利用 OBB 包围盒包围复杂的物体,将物体包围盒作为根节点构造八叉树层次结构。在进行碰撞检测时候,首先判断两物体包围盒树的根节点是否相交,若相交,再对它们的子节点进行相交测试。如果两物体包围盒的叶子节点发生了干涉,则需判断叶子节点中包含的物体模型的基元是否相交,对这些基元进行两两相交测试。

在均匀分布的稀疏环境中利用空间剖分法可以进行快速剔除掉不相交的物体<sup>[7]</sup>,但对于大规模复杂场景空间耗费过大<sup>[8]</sup>,而且计算效率低。包围盒树叶子节点中包含的基元之间的相交测试是基于面片与面片进行的测试,计算量大,消耗时间多。因此系统在运行时候会出现滞后、显示不流畅的现象。

### 2.1 虚拟手和物体包围球位置关系的比较

给虚拟场景中的每个物体模型添加一个包围球,通过判断虚拟手是否在静态物体包围球的邻近区域来进行快速有效的预先排除。具体的算法如下:

1) 在 unity 中给每个物体模型添加一个包围球,在检视面板中查找到该包围球的位置坐标和半径大小<sup>[9]</sup>。检视面板中物体包围球的位置坐标是局部坐标,需将其转化为世界坐标。

2) 将物体包围球的位置坐标与虚拟手的位置坐标进行比较,假设物体包围球位置坐标为  $(x_1, y_1, z_1)$ ,半径为  $r$ ;虚拟手的位置坐标为  $(x_2, y_2, z_2)$ ,判定条件为:

$$(|x_1 - x_2| \leq r) \parallel (|y_1 - y_2| \leq r) \parallel (|z_1 - z_2| \leq r)$$

当条件满足时,虚拟手与该物体可能发生了碰撞,需进行下一步详细的检测;否则可判断不可能发生碰撞。

与传统的空间剖分法相比,使用该算法既能节省存储空间,又可以减少计算量,因此可快速地找到虚拟手与静态物体模型潜在的相交区域。

### 2.2 为虚拟手添加 OBB

首先以虚拟手的手掌面为标准建立参考坐标系  $oxyz$ :  $x$ 、 $y$  所在的面要与手掌面相切,那么  $z$  矢量就是  $x$ 、 $y$  所在面的法向量,这样既可确定虚拟手 OBB 的方向矢量  $x$ 、 $y$ 、 $z$ 。然后要确定 OBB 的大小:OBB 的边界要与虚拟手保持一致<sup>[10]</sup>。

### 2.3 物体模型的八叉树描述

由于碰撞检测是在物体模型的八叉树节点和虚拟手的

OBB 包围盒之间进行,为了简化碰撞检测,将八叉树节点描述为 OBB 数据的形式,描述结构如下:

```
Structure Octant{
    Index code;
    Center point;
    Lengths of 3 edges;
}
```

八叉树上的节点有3类:第一类是白节点,它所对应的长方体中不包含对象中的内容;第二类是灰节点,它所对应的长方体部分被对象所占据;第三类是黑节点,它所对应的长方体完全被对象所占据,如图1所示,在层次划分的时候,只对灰节点进行划分。

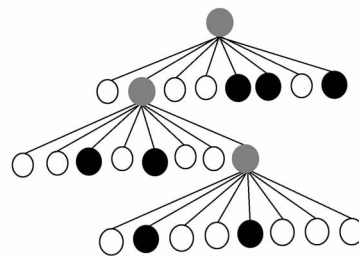


图1 八叉树结构

假设根节点的三边长分别为:

$$L_{x, \max}, L_{y, \max}, L_{z, \max}$$

则第  $i$  层节点的  $x$ 、 $y$ 、 $z$  轴的最大值为:

$$Li_{x, \max} = L_{x, \max} (0.5)^i$$

$$Li_{y, \max} = L_{y, \max} (0.5)^i$$

$$Li_{z, \max} = L_{z, \max} (0.5)^i$$

### 2.4 虚拟手的 OBB 和物体八叉树节点的碰撞检测

该碰撞检测采用分离轴定理的相交测试方法。两个 OBB 之间存在 15 个分离轴,若在这 15 个分离轴上两包围盒的投影区间均重叠,则两包围盒相交,否则就不相交。

在相交测试中,如果物体模型的八叉树层次结构中某节点与虚拟手 OBB 未相交,那就不需再对其子节点进行检测。但如果该节点和虚拟手包围盒相交,则需遍历它所有的子节点进行更深层次的相交测试。

### 2.5 矢量算法进行精确的碰撞检测

通过以上步骤检测出物体包围盒树中某叶节点和虚拟手包围盒相交,选取该叶节点上一些离散的点作为可能发生碰撞的点并记录下来。

在虚拟手抓取中,只有虚拟手的手掌面与物体接触才能进行抓取操作,因此将虚拟手简化为由多个对象的切平面组成的模型<sup>[10]</sup>。利用 2.1 中建立虚拟手 OBB 包围盒的方法得到手掌的切平面,同理得到各手指段的切平面。然后使用 unity 中的地面网格覆盖各切平面,避免虚拟手和模型之间的穿透。具体的矢量算法如下:

假设  $A$  为物体上一个可能发生碰撞的点,  $O_i$  为各切平面的中心点,  $AO_i$  为从  $A$  到  $O_i$  的向量,  $N$  是各切平面的法向量,  $d$  为  $A$  到切平面的距离,允许的碰撞检测误差范围为  $[-a, a]$  ( $a=0.1 \text{ mm}$ )。则

$$AO_i = A(x, y, z) - O_i(x, y, z)$$

$$d = \frac{N \cdot AO_i}{|N|}$$

$$\begin{cases} d \leq a & \text{发生碰撞} \\ d > a & \text{未发生碰撞} \end{cases}$$

### 3 实验结果与分析

本文采用 Unity 和 Kinect 作为开发平台。利用外部设备 Kinect 采集到的人手数据和 Kinect 提供给 Unity 3d 引擎的 SDK<sup>[12]</sup>, 建立虚拟手模型。最后通过 Kinect 2.0 将真实人手与场景中的虚拟手映射起来<sup>[13]</sup>。图 2 给出了虚拟手与物体模型发生碰撞的过程: 图 2 (a) 为虚拟手处于自由状态, 未与物体模型发生碰撞; 图 2 (b) 为虚拟手与物体模型发生了碰撞, 碰撞位置的颜色发生变化。

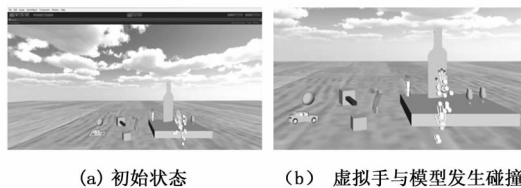


图 2 虚拟手与物体模型的碰撞实例

图 2 中可以看到, 实验效果和视觉感受都符合用户的意图, 满足精度要求。利用这种改进算法, 提高了算法的效率, 满足虚拟现实实时性的要求。

本文算法利用 Matlab 仿真, 仿真结果如图 3 所示。由图可知, 随着基元数目的增多, 改进的算法效率明显高于原始算法, 当基元达到两万多个时, 改进算法效率比原有算法提高了 25% 左右。

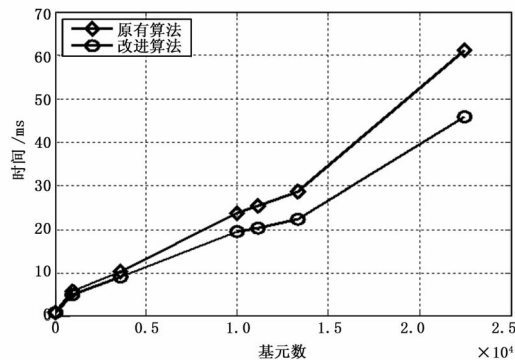


图 3 算法效率比较

### 4 结束语

基于虚拟手抓取物体的特性, 本文提出了一种基于八叉树

的层次包围盒碰撞检测的改进算法。首先为了简单快速地找到可能发生干涉的区域, 通过比较物体包围球与虚拟手的位置, 剔除掉不可能与虚拟手发生碰撞的物体, 对这些物体就不再进行详细的碰撞检测, 弥补了空间剖分法仅适用于均匀分布稀疏环境的不足。然后利用层次包围盒算法实现详细的碰撞检测, 遍历八叉树的时候采用宽度优先的算法, 提高了碰撞检测的效率。最后在精细碰撞检测阶段提出一种新的方法, 运用离散点与虚拟手简化面的矢量算法实现精确碰撞检测, 计算量少, 效率高。将本文提出的改进算法应用于复杂的刚性结构模型的仿真系统中取得了满意的效果。该方法也可应用于三维游戏中物体的碰撞检测和虚拟装配等领域。

#### 参考文献:

- [1] 周俊玮, 万 宇, 万旺根, 等. 一种基于八叉树的 OBB 包围盒碰撞检测方法 [J]. 计算机应用与软件, 2009, 04: 75-77.
- [2] 周艳飞, 王子牛. 基于 OBB 与八叉树的数控加工碰撞干涉算法的研究 [J]. 贵州大学学报 (自然科学版), 2012, 05: 68-71.
- [3] 刘 林, 贾庆浩, 熊志勇. 基于工程语义的虚拟装配序列规划 [J]. 机械设计与制造, 2013, 08: 44-47.
- [4] 李蕴奇. 基于可变形物体的碰撞检测算法 [J]. 吉林大学学报 (信息科学版), 2012, 03: 321-326.
- [5] Wang Y, Hu Y J, Fan J C, et al. Collision detection based on bounding box for NC machining simulation [J]. Physics Procedia, 2012, 24.
- [6] 姜光焱. 基于包围盒的碰撞检测算法的研究及应用 [D]. 成都: 电子科技大学, 2012.
- [7] Loeiz G, Schwartzman S C, Maud M, et al. Fast collision detection for fracturing rigid bodies. [J]. IEEE Transactions on Visualization and Computer Graphics, 2013.
- [8] Tapie L, Mawussi B, Bernard A. Topological model for machining of parts with complex shapes [J]. Computers in Industry, 2012, 5.
- [9] 姚翠莉, 袁 璠, 彭飞翔, 等. Kinect 在 Unity 平台上的开发实例 [J]. 计算机光盘软件与应用, 2014, 12: 68-69, 72.
- [10] 黄务莎. 基于虚拟现实的油泵车虚拟培训系统的研究与实现 [D]. 南京: 南京大学, 2013.
- [11] Wang X C, Fang Z G, Xu J, et al. Grasping of virtual hand based on Data Gloves [A]. Proceedings of 2010 4th International Conference on Intelligent Information Technology Application [C]. 2010.
- [12] 张 获. 应用领域的探讨 [J]. 物流工程与管理, 2012, 06: 39-41.
- [13] 陈文锋. 灵巧虚拟手自然交互研究 [D]. 杭州: 浙江大学, 2014.
- [8] 邱联奎, 刘启亮, 雷文龙. 基于背景减除与三帧差分相融合的运动检测 [J]. 合肥工业大学学报 (自然科学版) ISTIC PKU, 2014 (5).
- [9] 谭家政, 刘 勇, 邱芹军. 改进的 HSV 阴影去除算法研究 [J]. 物联网技术, 2014 (1): 84-86.
- [10] 卢章平, 孔德飞, 李小蕾, 等. 背景差分与三帧差分结合的运动目标检测算法 [J]. 计算机测量与控制, 2013, 21 (12): 3315-3318.

(上接第 2075 页)

- [5] 庞首颜, 张元胜. 基于三帧差分及 Canny 算子的运动目标提取 [J]. 重庆工商大学学报 (自然科学版), 2013, 30 (5).
- [6] 徐克虎, 王天召, 陈金玉, 等. 动态背景下的运动目标检测定位算法研究 [J]. 计算机测量与控制, 2013, 21 (12): 3319-3321.
- [7] 莫 林, 廖 鹏, 刘 勋. 一种基于背景减除与三帧差分的运动目标检测算法 [J]. 微计算机信息, 2009 (12): 274-276.