Android 手持设备游戏中的碰撞检测算法研究

罗 军[1]，李绍文[2]
(1. 桂林电子科技大学 电子工程与自动化学院，广西 桂林 541004；
2. 桂林电子科技大学 网络中心，广西 桂林 541004)

摘 要：针对 Android 手持终端中复杂游戏场景的碰撞检测需求，提出了一种基于包围球和 AABB 的实时碰撞检测算法。该算法针对不同的虚拟对象构建不同的包围盒，并将改进后的包围盒投影排序分组方法应用其中。将该算法与使用包围盒投影排序分组方法的包围球算法与 AABB 算法比较，实验表明，该算法在保持更高精度的前提下仍能满足复杂场景中实时碰撞检测的要求。

关键词：Android；碰撞检测；包围球；AABB
中图分类号：TP391.9 **文献标识码**：A **文章编号**：1674-7720(2013)14-0035-03

Research on collision detection algorithms in games of Android PDA

Luo Jun[1], Li Shaowen[2]
(1. Department of Electronic Engneering and Automation, Guilin University of Electronic Technology, Guilin 541004, China;
2. Department of Network Center, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: A real-time collision detection algorithm, which is based on sphere bounding box and AABB, is presented to content the requirement of complex scenes of games in Android PDA. Difference bounding box are built for difference virtual objects in the algorithm, and the improved method of bounding box projected in axis to sort and group is applied. Experiments proved, this algorithm completely meet the requirement of real-time collision detection as well as the sphere bounding box and AABB algorithms that also used the improved algorithm, but is more accurate.

Key words: Android; collision detection; sphere bounding box; AABB

随着移动互联网的高速发展和快速普及，以及以 ARM Cortex-A8、Cortex-A9 为代表的高端嵌入式芯片的飞速发展，智能手机、平板电脑等手持 PDA 已经进入每一个人的生活，并且其性能越来越强大。基于 Android 的高端消费类电子产品更是以其开源性和通用性受到广大消费者的青睐。游戏无疑是这些手持设备的一个重要应用，而且不是传统的简单 2D 游戏，而是越来越复杂的 3D 游戏，甚至是移植的 PC 经典游戏，如实况足球、极品飞车、CS 等。为了使游戏更加逼真，场景中对象间的实时碰撞检测[1]是一个必须要解决的问题。但是当前手持设备的 3D 游戏规模越来越庞大，场景越来越复杂，如何快速精确地实现移动手持设备中复杂游戏场景的实时碰撞检测已经成为当前的研究热点之一。

在游戏场景中，一般采用各种包围盒算法来替代对象进行相交测试，但假设游戏场景中有 D 个动态对象和 S 个静态对象，如果直接用两对象的包围盒进行碰撞检测，则必须经过 C_D^2+DS 次检测才能最终确定整个场景的碰撞情况，当 D 和 S 都较大时，这将是一个庞大的计算量，从而严重影响游戏场景的实时性和真实性。基于 Lin-Canny[2-4]算法的 I-Collide[4-5]算法库是一种经典的实时碰撞检测算法，它使用一维区间排序法快速排除不相交的对象对，从而大大减少了精确求交的计算量。由于游戏场景对实时性的要求比较高，本文在 I-Collide 算法库及其改进算法的基础上提出了一种基于包围球[1]和轴对称包围盒(AABB)[1]的快速碰撞检测算法。

1 算法描述

1.1 I-Collide 算法库及其改进算法概述

三维空间的包围盒间相交测试，常常是把包围盒投影到坐标轴上转化为一维或二维来处理，即通过降低维度来提高检测的效率。基于 AABB 的经典 I-Collide 算法库采用的是一维空间排序法，即将 AABB 投影到某一坐标轴上，如图 1 所示，然后使用插入排序法对投影列表

排序确定包围盒的交叠情况，从而将没有相交的包围盒快速略去。I-Collide 算法库比以往的其他碰撞检测算法更加高效，但它忽略了碰撞检测的局部性，将排序后的所有包围盒两两进行相交测试。董峰等[2]对该排序算法进行了改进，使用二维投影排序的方法，即将所有物体的包围盒投影到 $x-y$ 坐标面，然后用矩形排序算法进行筛选，最后验证交迭的矩形对在 z 轴上的相交情况。王晓荣[5]又进行了改进，她不仅把包围盒投影到二维 $x-y$ 平面，而且采用了效率更高的希尔排序法对投影序列实时排序，并将坐标轴划分为一系列包含相同数目的投影子段(如果投影区间数不能被均匀划分，让最后一个子段的投影区间数小于前面每个子段中的投影区间数，且前面的每个投影子段包含的投影区间数应相同)。

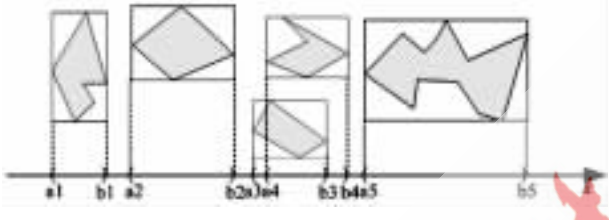

图 1 投影到 x 轴的包围盒序列

1.2 算法改进

随着虚拟场景的实时更新，如果每次都用希尔排序法对所有包围盒实时排序，会消耗大量的时间，从而降低算法的效率。如图 2 所示，本文对此进行了改进，使用了排序效率更高的堆排序法对某一轴上的投影序列排序，并将初次划分后投影子段(该文中称之为“组”)的边界固定，之后将运动对象更新后包围盒的投影值与之前包围盒所在组的边界值进行比较，如果仍在边界之内，则继续留在本组内，否则，与下一组的边界值比较，依次类推，直到找到新的归属组；然后在每组内使用堆排序法对投影值进行实时排序，快速排除没有交叠的对象对；最后，对交叠的对象对(其中至少有一个对象是动态的)进行相交测试。其实，在很短的时间间隔内包围盒的位移量是有限的，因此，包围盒分组时往往只要与本组及相邻组进行比较即可。这样随着场景的实时更新，很可能某些组所包含的对象全部都是静态的，因而整个组都可以不用进行相交测试了。即：相交测试只在有动态对象存在的组内进行，这大大提高了检测的效率。

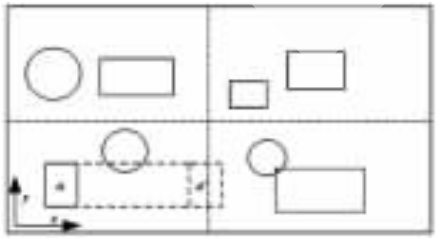

图 2 包围盒投影排序分组

该算法需要注意以下问题：(1)当某一包围盒的一部分已进入新的投影子段，而另一部分仍然留在原投影子段时(如图 2 中虚线框图所示)，包围盒在两个投影子段所属组都需参与碰撞检测。并且当某两个相交对象在坐标轴划分阶段都被分为两个部分、当对两个对象左边部分对应的 x 轴和 y 轴子列表进行相交测试时，已经知道两对象重叠，算法就不再对两个对象的右边部分的投影进行相交测试，从而减少了算法的执行时间。(2)如果场景中有体积非常大的物体，在执行碰撞检测之前必须先将物体分解为体积较小的子块，再将各子块分别进行碰撞测试。

1.3 新算法描述

由于游戏场景中几何物体的形状各异，不同对象用不同的包围盒来近似模拟时会出现截然不同的效果，不合理使用包围盒会使碰撞检测效率大幅降低。例如，一个球体如果用 AABB 来模拟，会出现较大的误差；而一个长条形物体用包围球模拟，碰撞检测的精度也会很差。一种好的碰撞检测系统应该根据不同对象的形状特征提供不同的解决方案。因此，本文用包围球来模拟近似球状的对象，其余的对象都用 AABB 代替参与相交测试，并将上述改进后的包围盒投影排序分组算法应用在该算法中，从而快速完成场景的碰撞检测。

整个算法的执行过程是：首先对包围球和 AABB 投影排序并分组，这样可以迅速排除不可能发生碰撞的对象对；然后，对各组中的潜在碰撞检测集进行相交测试，即最终的碰撞检测就转化为球与球的相交测试、球与 AABB 的相交测试以及 AABB 间的相交测试。

2 包围盒间的相交测试

包围球之间的相交测试很容易实现，只需根据球心之间的距离和两球的半径之和的差值就可以判断相交情况。两个 AABB 相交，则它们在 3 个坐标轴上的投影也必然相交，因此，AABB 间的相交测试完全可以转化为坐标轴上投影区间的相交测试，只要有一个轴上的投影不相交，就可以直接判定两包围盒不相交，从而结束相交测试。

包围球与 AABB 的相交测试，如果使用常规的几何分析方法会非常复杂。一类简单可行的方法是求取包围球到 AABB 的最近距离点对，但如果使用常规的暴力遍历算法来求解几何体之间的最近点对，效率太低。而 Lin- Canny 算法求解凸包间的最近点对是非常高效的。Lin-Canny 算法高效实现的关键是引入了 Voronoi 域的思想，通过遍历 AABB 的 Voronoi 特征域获取球心在某一特征(顶点、边或面)上的投影点，这个投影点就是 AABB 到球心的最近点，再求出这个最近点到球心的距离并与包围球的半径(实际使用平方值)进行比较，如果这个距离小于半径，则包围球与 AABB 相交；否则，不相交。

如图 3 所示，假设 P 为球心位置，A 为 AABB，A 包围盒中顶点 Q 的 Voronoi 域、边 MQ 的 Voronoi 域和面 S 的 Voronoi 域分别如图中不同颜色所示。则 A 上到 P 的最近点可以通过下面的方法获得：在某一轴上将 P 限定

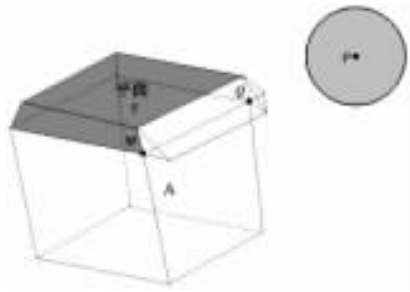

图 3 包围盒 A 的某一 Voronoi 顶点域、边域和面域

在 A 的边界上。存在 3 种限定 P 的情况：(1)如果 P 点位于 A 的某表面的 Voronoi 域，则截取操作将 P 限定在 A 的该表面上；(2)如果 P 点位于 A 的某顶点的 Voronoi 域，则截取操作将视该顶点为所求点；(3)如果 P 点位于 A 的某边的 Voronoi 域，则截取操作为该边上的正交投影。其伪代码实现如下：

```
    /*p 为球心，a 为 AABB*/
Point  findNearestDistancePoint(Point  p,
                    AABBBox  a)
{
    Point  q;                                   /* 所求点 q*/
    for(int  i=0;i<3;i++)
    {
        float  v=p[i];
        if(v 小于 a 在某一轴上投影的最小值)
            v=这个最小值;
        if(v 大于 a 在某一轴上投影的最大值)
            v=这个最大值;
        q[i]=v;
    }
    end  for;
    return  q;                                  /* 返回 q 点 */
}
```

3 实验结果及分析

在小米 M1 手机上实现该算法，其系统为 Android OS 4.0；CPU 为高通骁龙 MSM-8260 (双核)1.5 GHz；GPU 为 Adreno220；RAM 为1 GB；图形 API 接口为OpenGL ES 2.0。首先，在多个包含不同对象个数的场景中，将改进算法与参考文献[5]中的算法(只进行包围盒相交测试，省去了叶子节点的相交测试)进行比较，结果如图 4 所示。再将改进后的包围盒投影排序分组方法应用到包围球算法和 AABB 算法中，并与本文提出的新算法进行对比分析，结果如表 1 所示。

由于用堆排序法替代了希尔排序法，并将静态包围盒的分组固定，只实时更新动态包围盒的组别，随着包围盒数量的不断增加，改进算法比参考文献[5]中的算法效率更高。尤其是当场景越来越复杂时，改进算法完成碰撞检测所需的时间并没有明显地增加，具有更加优越的实时性能。 通常游戏场景所需的最低帧率一般为

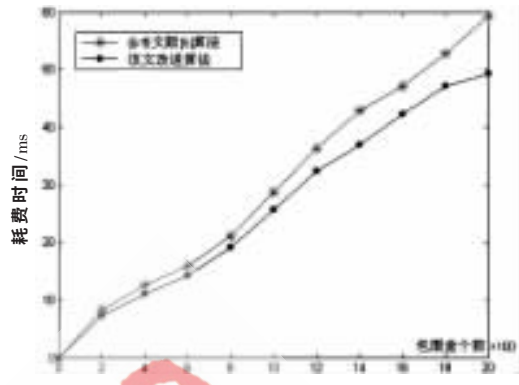

图 4 算法耗费时间比较

表 1 各算法所需碰撞检测时间

包围盒个数	检测时间/ms 包围球	AABB	该文算法
200	6.8	6.9	7.2
400	10.4	10.9	11.1
600	13.3	13.6	14.3
800	18.0	18.8	19.1
1 000	23.7	24.8	25.7
1 200	30.2	30.9	32.3
1 400	34.3	35.2	36.9
1 600	39.7	40.3	42.2
1 800	45.4	46.2	47.1
2 000	47.1	47.6	49.2

20 f/s~30 f/s。本文算法在保证了更高精度的前提下仍能与包围球算法和 AABB 算法一样满足复杂场景的实时性需求。

针对 Android 手持终端的复杂游戏场景的需求，本文在 I-Collide 算法库及其改进算法的基础上，提出了一种基于包围球和 AABB 层次包围盒的碰撞检测算法。把传统的基于 AABB 的投影排序法应用在该算法中并进行优化，从而排除了大量不可能相交的对象对，提高了算法的实时性。但该算法没有对变形体或旋转体的包围盒重构提出快速的解决方案，有待在以后的研究中解决这些问题。随着嵌入式芯片的高速发展和实际应用的需要，以后的研究中还应该考虑使用 OBB[1]或 k-DOPs[1]等更加紧密的包围盒算法或基于 GPU 的碰撞检测算法[6]及其他的精确碰撞检测算法来满足各种场合的苛刻需求。

参考文献

[1] 邹益胜，丁国富，许明恒，等.实时碰撞检测算法综述[J].计算机应用研究，2008,25(1)：8-12.

[2] 董峰，王同洋.虚拟环境中的快速碰撞检测算法[J].计算机工程与应用，2003,39(8)：66-67.

[3] Lin Ming, CANNY J. A fast altorithm for incremental distance calculation[C].Proceedings of the 1991 IEEE International Conference on Robotics and Automation，1991:1008-1014.