

碰撞检测技术在三维交互漫游系统中的应用

和莉,刘惠义

(河海大学 计算机及信息工程学院,江苏 南京 210098)

摘要:针对提高交互漫游系统中动态的视点与静态的场景对象之间的碰撞检测性能进行了研究,将碰撞检测算法分成两步:第一步采用包围盒进行碰撞的粗略检测;第二步采用相交测试法进行碰撞的精确检测,并采用限时计算的思想对整个碰撞检测加以优化。实验结果表明,该算法加快了碰撞检测的速度,能较好地适应3D物体碰撞检测准确、快速的要求。

关键词:三维交互漫游;碰撞检测;包围盒;相交测试

中图分类号:TP391.41

文献标识码:A

文章编号:1673-629X(2006)06-0092-03

Application of Collision Detection in Three-Dimensional Walkthrough System

HE Li, LIU Hui-yi

(School of Computer Science and Engineering, Hohai University, Nanjing 210098, China)

Abstract: Collision detection is the key issue of three-dimensional walkthrough system. This paper proposes a two-step method for the collision detection between active observer and static virtual environment of three-dimensional walkthrough system. The first step uses bounding box to get a rough detection. The second step uses intersection test to get accurate collision detection. The collision detection is optimized by applying the thought of time-limited computing. The experiment results show that the algorithm speeds up the detection.

Key words: three-dimensional walkthrough; collision detection; bounding box; intersection test

0 引言

碰撞检测是构造三维交互漫游系统的不可或缺的一个重要部分,它可以使用户以更自然的方式与系统中的场景对象进行交互。如果没有碰撞检测,当一个对象碰到另一个对象时,往往会“穿墙而过”,这在现实中是不存在的。因此,构造三维交互漫游系统时,必须能够实时、精确地判断场景中物体之间是否发生碰撞。

碰撞检测的研究目标是如何在很高的实时交互要求下完成对大量复杂物体的碰撞检测,其具体工作包括两部分:检测到有碰撞;计算出碰撞发生的位置。目前,有多种方法可以实现碰撞检测技术^[1]。由于碰撞检测算法对具体碰撞场景非常敏感,因此没有一种算法可以针对所有的情形都具有很好的性能^[2]。文中从交互漫游角度出发,针对动态的视点与静态的场景对象之间的碰撞,研究将包围盒与相交测试法相结合的碰撞检测算法,并加以优化。

1 算法的基本思想

包围盒方法是目前碰撞检测的一个常用方法,其最大

的好处是可以实现快速碰撞检测。但在很多实际应用的交互系统中,要想做到自然交互光靠包围盒进行碰撞检测是不够的。当要证明两个对象并不相交时,利用包围盒是非常有效的;但是当两个对象的包围盒相交时,并不能保证这两个对象一定是相交的,因为包围盒仅仅是对象边界的一个简单粗略的表示。因此,基于包围盒的碰撞检测是非常粗略而不精确的。

精确的碰撞检测对提高真实性、增强沉浸感有着至关重要的作用,而三维交互漫游系统自身的复杂性和实时性又对碰撞检测提出了更高的要求。因此,为了实现精确检测,在实际应用中将算法分成两步:第一步采用包围盒方法排除不可能碰撞的元素;第二步把视点及其方向确立为射线,采用射线与三角面相交测试法,进行精确检测并准确确定碰撞点的位置。

1.1 包围盒碰撞检测法

沿坐标轴的包围盒(AABB, axis-aligned bounding boxes)^[3]在碰撞检测的研究历史中使用的时间比较长,使用的范围较广。所给的对象被定义为包含该对象且各边平行于坐标轴的最小的六面体。图1给出了一个使用AABB包围盒的例子,它具有顶点 a^{\min} 和 a^{\max} 而且轴为标准基轴。给定对象E的AABB的计算十分简单,只需分别计算组成对象的基本几何元素集合中各个元素的顶点的X坐标、Y坐标和Z坐标的最大值以及最小值;AABB

收稿日期:2005-09-15

作者简介:和莉(1981-),女,云南丽江人,硕士研究生,研究方向为虚拟现实、计算机图形学;刘惠义,博士,副教授,研究方向为计算机图形学、虚拟现实、模式识别与智能系统。

间的相交测试也比较简单,两个 AABB 相交可归纳为,当且仅当它们在 3 个坐标轴上的投影区间均重叠。

定义 AABB 包围盒的 6 个最大值和最小值,则分别确定了它在 3 个坐标轴上的投影区间,因此, AABB 包围盒间的相交测试最多只需要 6 次比较运算。

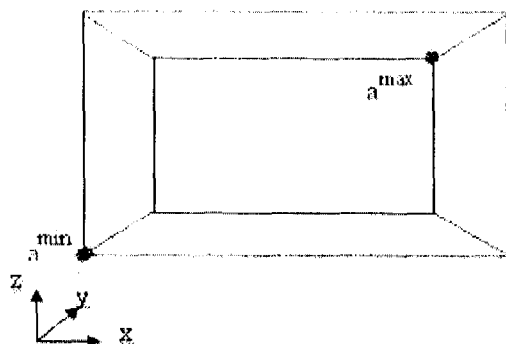


图 1 一个三维 AABB 的图例

具体实现的伪代码如下:

```
Struct TZone
{
    float max;
    float min;
}

Struct TBoundingBox
{
    TZone X;
    TZone Y;
    TZone Z;
}

int TestofCollision ( TBoundingBox BoundingBox, TBoundingBox Camera)
{
    if (BoundingBox.X, BoundingBox.Y) 和 (Camera.X, Camera.Y) 相交
        if (BoundingBox.Y, BoundingBox.Z) 和 (Camera.Y, Camera.Z) 相交
            if (BoundingBox.X, BoundingBox.Z) 和 (Camera.X, Camera.Z) 相交
                return 1; // 两个包围盒相交
            return 0; // 两个包围盒不相交
    }
}
```

1.2 射线与三角面相交测试法

在计算机图形学中,经常会用到相交测试。目前有很多关于“线/体”的相交测试方法^[2],由于三维环境中任何物体的表面都可以被剖分成三角形,因此可以采用射线与三角面相交测试方法,首先计算出射线与三角形所在平面的交点,然后将相交点和三角形顶点投影到一个轴对齐平面(xy, yz 或者 xz)上,这样三角形面积就会最大化。通过如此操作,就可以将这个三维问题变成二维问题,随后只需要判断二维点是否在二维三角形内部即可。

(1)射线的定义。

一条射线 $R(t)$ 通常由一个原点 O 和一个方向向量 n 定义,为了方便起见, n 一般是归一化的,也就是说 $\|n\|$ 可以将射线如下表达:

$$R(t) = O + t \times n \quad (1)$$

作为一个变量,标量 t 用来在射线上生成不同的点,当 t 值小于零时,生成的点在原点的反方向上(不属于这条射线);而当 t 值大于零时,生成的点就在原点的正方向上。此外,由于对射线方向进行了归一化,因此 t 值在射线方向上生成的点距离原点有 t 个单位长度。

(2) 射线与平面的相交。

由平面上的一个点和一条法线已经足够定义一个平面。例如,如果有一个点 $(0,0,0)$ 和一条法线 $(0,1,0)$,实际上就已经定义了一个平面,即 xoz 平面。一个平面可由以下的矢量来描述:

$$Xn \cdot X = d \quad (2)$$

矢量 X 是平面上的一点,矢量 Xn 表示该平面的法线, d 描述了从坐标系的原点到平面的距离。设射线与平面相交于点 P ,则 P 必定同时满足方程(1)和(2),解这个方程组,得

$$t = (Xn \cdot P - Xn \cdot O) / (Xn \cdot n) \quad (3)$$

将式(3)代入式(1)即得到碰撞发生的位置。但是还有几个特殊情况需要考虑:如果 $Xn \cdot n = 0$,表明射线和平面是平行的,将不会有撞击点。如果是负数,表明撞击点是在射线的起始点的后面,也就是沿着射线后退的方向才能撞到平面,这只能说明射线和平面没有交点。具体实现的代码如下:

```
int TestIntersect(CVector3 PNormal, CVector3 RDirection,
CVector3 Intersection, CVector3 RStartpoint)
{
    //求得平面法线和射线方向的点积
    double DotResult = PNormal.dot(RDirection);
    //判断射线是否和平行
    //判断一个浮点数是否为0,也即在一个很小的数的正负区间内即可认为该浮点数为0
    if ((DotResult < e) && (DotResult > -e))
        return 0;
    //求得从射线的起点到碰撞点的距离
    double d;
    d = (PNormal.dot(Intersection) - PNormal.dot(RStartpoint)) / DotResult;
    //如果 d 小于 0 表明碰撞点在射线的反方向上,这只能表明两者没有相撞
    if (d < -e)
        return 0;
    return 1;
}
```

(3) 判断交点是否位于三角形的内部。

判断一个点是否在三角形内部,可以通过图 2 所示的方法,即如果该点与三角形两个顶点的夹角 α, β 和 γ 之和

等于 360° , 则该点位于三角形的内部。

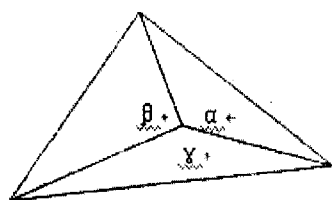


图 2 点在三角形内部的判断

1.3 算法优化

经过包围盒做粗略检测,快速排除了很多不相交的物体,碰撞检测的速度已大有提高。但仍有进一步提高的可能性。当碰撞的发生概率是小概率事件时,可以有效地减少计算次数以及每次计算的复杂度。因此,对以上提出的两步碰撞检测法可进一步引入限时计算的思想^[4],采取设定可变时间片长度的方法。

时间步长问题的实质是计算频率问题^[5],原因是在有些时间片中物体根本不可能发生碰撞。时间步长方法的目标是避免不必要的时间片中的计算。当碰撞无法发生时,就增大时间步长。当空间中的碰撞发生频率比较低时,这种方法可以节省大量计算时间。

2 实验结果

基于 VC++ 6.0 和 OpenGL,开发了泰兴东收费站室外三维交互漫游系统,如图 3 所示。整个场景采用 3D 模型,共有 29833 个三角面。在系统中运用文中上述算法进行了碰撞检测,取得了很好的效果,说明该算法是可行和有效的。

3 结束语

关于碰撞检测的研究,目前已经有很多很好的实现算法。对于不同的问题,往往采用不同的检测碰撞的算法。

(上接第 91 页)

RTX51 Tiny 使得程序的编写和调试变得简单起来。上述交通灯的程序如果不采用操作系统,将要编写大概 200 行的源程序,而采用了 RTX51 Tiny 后,程序还不到 50 行。当程序越复杂时,它作为操作系统精简代码的优点就更加明显。然而,由于资源的限制,它缺乏一般操作系统所必要的支持,没有功能强大的软件包,用户得自己编写驱动程序。

这些应用特点及程序的运行充分说明了 RTX51 Tiny 操作系统代码精简和运行可靠的基本特点^[5]。

当然,一个嵌入式工作者要面对的不仅仅是决定采用什么操作系统和开发工具的问题,还必须学会适应一个可能完全外语化的编程环境。并且还不得不处理如系统资源限制、硬件设备驱动、存储分配等一系列问题。

常用算法会带来很大的计算开销,效率不高。文中将两种方法结合,把碰撞检测的过程分成粗略检测和精确检测两步进行,并采用限时计算的思想加以优化,提高了碰撞检测的效率。

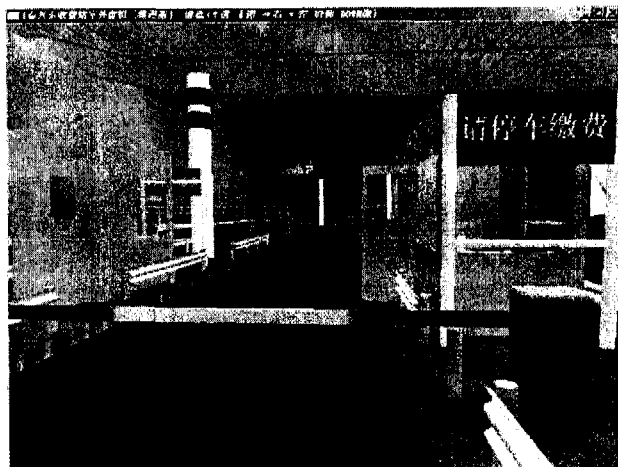


图 3 泰兴东收费站室外三维交互漫游系统

参考文献:

- [1] 王志强,洪嘉振,杨辉.碰撞检测问题研究综述[J].软件学报,1999,10(5):545-551.
- [2] Akenine-Moller T, Haines E. 实时计算机图形学(第2版)[M]. 曹建涛译. 北京:北京大学出版社,2004.
- [3] Van den Bergen G. Efficient collision detection of complex deformable models using AABB trees[J]. Journal of Graphics Tools, 1997, 2(4): 1-13.
- [4] Hubbard P M. Collision detection for interactive graphics applications[J]. IEEE Trans on Visualization and Computer Graphics, 1995, 1(3): 218-230.
- [5] 石教英. 虚拟现实基础及实用算法[M]. 北京:科学出版社, 2002.

参考文献:

- [1] Stepner D, Rajan N, Hui D. Embedded Application Design Using a Real-Time OS[A]. Proceedings of the 1999 36th Annual Design Automation Conference (DAC)[C]. New Orleans, LA, USA: IEEE, 1999. 151-156.
- [2] 李仕涌, 谭南林. 多任务操作系统在嵌入式系统开发中的应用[J]. 北方交通大学学报, 2002, 26(4): 79-82.
- [3] 赵学军. 单片机实时嵌入式操作系统微内核的设计[J]. 桂林电子工业学院学报, 2002, 22(3): 76-79.
- [4] 刘天泉, 黄海, 王树青. RTX51 在运动控制系统中的应用[J]. 机电工程, 2004, 21(1): 30-32.
- [5] 李小文, 万家富, 梁慧冰. $\mu\text{C}/\text{OS}-\text{II}$ 在 MCS-51 系列中的应用[J]. 微计算机应用, 2003, 24(5): 274-277.