扇形与圆盘相交测试浅析



本文以文字方式节录魔方学堂《游戏常用几何学应用及优化》课程的部分内容。本文先简单介绍游戏中几何相交测试的应用,讨论如何用数学方式表示各种形状,然后用例子解释分离轴定理的应用,最后讨论如何实现动作游戏中常用的扇形攻击范围与圆形受击范围的攻击判定。阅读本文需要基本的线性代数(矢量运算)知识,代码范例以 Unity 的 C# 脚本实现。

(题图 Photo by Jim Wang)

1. 引言

几何形状的相交测试(intersection test)判断两个几何形状是否相交,在游戏中常应用于游戏性编程(游戏对象触碰区域、攻击判定、拾取三维物体)、物理碰撞测试、图形可见性测试等,可以说是游戏编程的基础之一。若要判断大量的游戏对象是否互相相交,通常会使用一些空间剖分的方法,去降低平均情况下的运行复杂度,但本文将集中讨论两个几何形状的相交测试。

虽然游戏引擎(如 Unity)一般已内置相交测试功能,但是其支持的形状有限。解决方法之一是用一些近似方式拟合所需的形状,例如用网格或多个矩形来拟合扇形,然而,更理想的做法是直接求解,除了更精确,性能也可能更优。本文最后介绍的扇形与圆盘相交测试,也是沿自项目组提出的一个具体问题,而 Unity 并没有直接支持。因此,无论是游戏引擎开发者,或是前台/后台游戏程序员,也应该对几何相交问题有一些基本了解。

2. 形状表示方式

在数学上,相交是指两个集合有共通的元素。在几何上,我们可以把形状定义为点的集合,若两个形状有共点,就是表示它们相交。

2.1 圆形和圆盘



已关注

$$\|\mathbf{x} - \mathbf{c}\| = r$$

方程左右侧都是正数,可以把左右平方,那么这个矢量形式也可以展开成二元方程:

$$\|\mathbf{x} - \mathbf{c}\|^2 = r^2$$

$$\Leftrightarrow (\mathbf{x} - \mathbf{c}) \cdot (\mathbf{x} - \mathbf{c}) = r^2$$

$$\Leftrightarrow (x - c_x)^2 + (y - c_y)^2 = r^2$$

除此之外,我们也可以用参数式来表示圆形上的点:

$$x(\theta) = c_x + r \cos \theta$$

 $y(\theta) = c_y + r \sin \theta$

如果我们希望圆形包含圆形内的面积,这种形状在数学上称为圆盘(disk),可用不等式表示,有开闭之分:

$$\|\mathbf{x} - \mathbf{c}\| < r$$
$$\|\mathbf{x} - \mathbf{c}\| \le r$$

2.2 线段和胶囊

再举另一个常见的形状作为例子。二维直线(line)是二元一次方程,可表示为多种方式:

• 点斜式: $y - y_0 = m(x - x_0)$

• 斜截式: y = mx + b

• 两点式: $\frac{y-y_0}{y_1-y_0} = \frac{x-x_0}{x_1-x_0}$

• 截距式: $\frac{x}{a} + \frac{y}{b} = 1$

• 一般式: Ax + By + C = 0

• 参数式: $\mathbf{x}(t) = \mathbf{x}_0 + t\mathbf{u}$

然而,数学上斜率和截距可能是无穷大,而程序中需要表示任何直线,因此含斜率和截距的表示方式都不太适合。另一方面,三维直线的一般式需要用两个平面的共线来表示,在程序中会有退化问题(即两个平面不相交或是共面)。所以,无论是二维、三维或更高维,最容易表示直线的方法是**参数式**,即用直线上的其中一点与方向矢量来表示。

线段(line segment)可表示为两个点之间的线性插值。我们也可以沿用上面的参数式,设 ${\bf u}={\bf x}_1-{\bf x}_0$,并限制 $t\in[0,1]$:





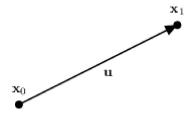


图 2.1:线段表示方式

上一节提到,圆形是与圆心等距的点集,那么,与线段等距的点集是一个什么形状?我们若定义点和线段的距离是指两者最近的距离,则我们可以表示胶囊(capsule)形状。三维的胶囊体也可以这样表示。

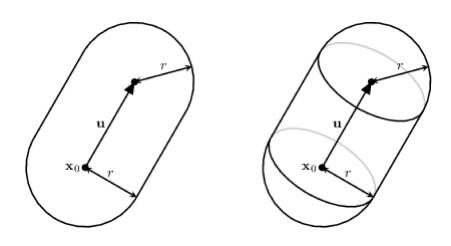


图 2.2: 二维胶囊形和三维胶囊体

2.3 矩形

最简单的矩形是各面与主轴对齐的(不能任意旋转),通常称为轴对齐包围盒(axis-aligned bounding box, AABB)。AABB 通常以每轴的最小、最大范围定义,也可理解为两个极点 ${\bf a}_{\min}$ 和 ${\bf a}_{\max}$ 。

$$x_{\min} \le x \le x_{\max}$$
 $y_{\min} \le y \le y_{\max}$

但除了极点来定义外,也可以使用对称形式,以中心点 \mathbf{c} 和半长度(half-extent)矢量 \mathbf{h} 定义。

$$|x - c_x| \le h_x$$
 $|y - c_y| \le h_y$

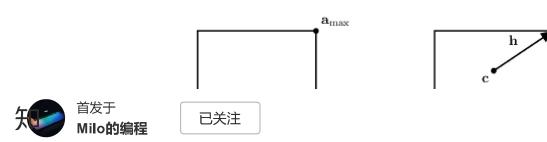


图 2.3: 两种 AABB 的表示方式

如果矩形不与轴对齐,就称为定向包围盒(oriented bounding box, OBB)。OBB需要加上定向信息,在二维中只需要加上一个角度,而在三维中则可选择 3×3 矩阵或单位四元数等方式表示定向。

3. 几何相交测试

知道如何用数学表示一些形状之后,我们便可以判断它们是否相交。如果两个形状都是用方程来表示,只需联合它们的方程,判断它们是否有解,就能得知它们是否相交。那些解就是两个形状的共点。

然而,有一类称为凸集的形状,是特别容易做这种判断的。我们先定义凸集。

3.1 凸集

凸集 (convex set) 是点的集合,集合中任意两点间的线段都在该集合中。用数学可以这么表示:

$$\forall \mathbf{x}, \mathbf{y} \in S, \forall t \in [0, 1], t\mathbf{x} + (1 - t)\mathbf{y} \in S$$

下图是凸集和非凸集的例子。注要圆盘是凸集,但圆形不是(圆周上任意两点连成的线段中的点都不在圆周上)。

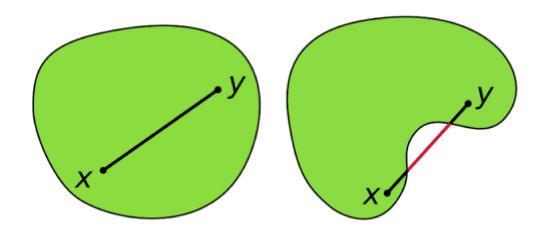


图 3.1: 凸集与非凸集的例子

3.2 分离轴定理



已关注

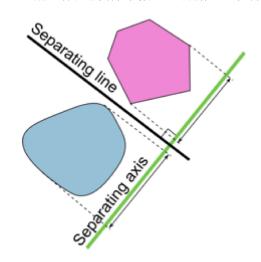


图 3.2:分离轴定理

那么,要判断两个形状是否相交,我们只需要找出可能的分离轴,然后判断那些轴是否能把两个形状分离。若所有可能的分离轴都不能分离它们,就可以确认它们是相交的。

3.3 圆盘与圆盘的相交测试

作为首个例子,圆盘与圆盘可能是最简单的相交测试。传统上我们或会考虑若两圆心距离短于 半径之和,那么两圆盘便相交。若换以 SAT 的方式来思考,圆盘是凸集,而唯一可能分离它 们的分离轴是两圆心连成的轴,如下图。如果该轴不能分离两个圆盘,两个圆盘便相交。

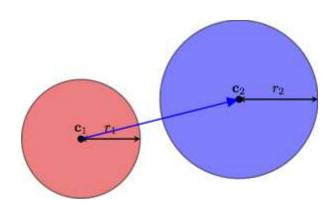


图 3.3:圆盘与圆盘的相交测试

在此例子中,套用 SAT 并没有造成实质上的差异,但对于其它形状,例如矩形,就非常有用。

3.4 圆盘与胶囊相交测试

圆盘和胶囊的相交测试其实和上一个例子也很相似。它们唯一的分离轴就是圆心和胶囊线段之间最短距离方向的轴。



已关注

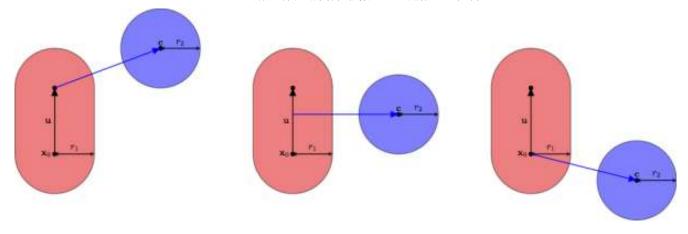


图 3.4:圆盘与胶囊相交测试

因此,这个相交问题可以转化为点与线段的求最短距离问题(下图中的 d),然后再比较该距离是否少于半径之和。

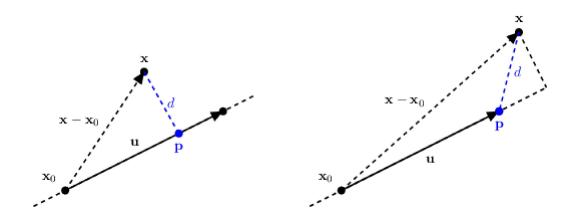


图 3.5:点 \mathbf{x} 与线段 $\mathbf{x}_0 + t\mathbf{u}$ 的最短距离为 d

首先 , 计算点 \mathbf{x} 在 $\mathbf{x}_0 + t\mathbf{u}$ 上的投影 :

$$t = \frac{(\mathbf{x} - \mathbf{x}_0) \cdot \mathbf{u}}{\|\mathbf{u}\|^2}$$

把 t 限制在 [0,1] , 得最近点 \mathbf{p} , 然后用最近点求最短距离:

$$d = \|\mathbf{x} - (\mathbf{x}_0 + \min(\max(t, 0), 1)\mathbf{u})\|$$

如同圆盘与圆盘的相交测试,我们可以只比较距离的平方,从而除去开方运算。

// 计算线段与点的最短平方距离
// x0 线段起点
// u 线段方向至末端点
// x 任意点



已关注

```
}

// 胶囊与圆盘相交测试

// x0 胶囊线段起点

// u 胶囊线段方向至末端点

// cr 胶囊半径

// c 圆盘圆心

// r 圆盘半径

static bool IsCapsuleDiskIntersect(
    Vector2 x0, Vector2 u, float cr,
    Vector2 c, float r)

{
    return SegmentPointSqrDistance(x0, u, c) <= (cr + r) * (cr + r);
}
```

3.5 圆盘和AABB相交测试

在圆盘和AABB的相交测试中应用 SAT,会发现有两种分离轴:

- 2个坐标轴方向
- 4个顶点与圆心的方向

我们可以简单地检查这6个分离轴之中,是否有任何一个能把两形状分离。

但是,我们可以利用矩形的对称性质减少分离轴,方法是把 AABB 的中心 c 作为坐标系原点,然后把圆心反射至第一象限:

$$\mathbf{v} = \begin{bmatrix} |p_x - c_x| & |p_y - c_y| \end{bmatrix}$$

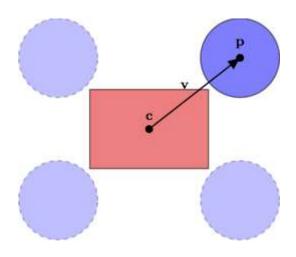


图 3.6: 利用绝对值运算把虚线的圆盘都反射至第一象限



已关注

写文章 。。。

$$\mathbf{u} = \left[\max(v_x - h_x, 0) \quad \max(v_y - h_y, 0) \right]$$

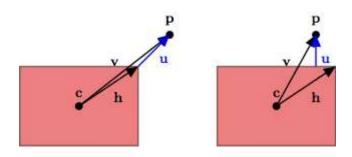


图 3.7: 减去半长度后,把负值分量设为 0,便得点与AABB的最短矢量 \mathbf{u}

最后用 u 作为唯一分离轴,若不能分离便判断为相交:

```
\|\mathbf{u}\|^2 \le r^2

// c AABB的中心
// h AABB的半长度
// p 圆盘的圆心
// r 圆盘的墨仑
static bool IsAabbDiskIntersect(Vector2 c, Vector2 h, Vector2 p, float
        Vector2 v = Vector2.Max(p - c, c - p); // = Abs(p - c);
        Vector2 u = Vector2.Max(v - h, Vector3.zero);
        return u.sqrMagnitude <= r * r;
}
```

这个算法最早见于[2]。

以上的 3 个例子都可以简单地扩展至三维,当中的圆盘→球体(ball)、胶囊→胶囊体、矩形→长方体。

4. 扇形与圆盘相交测试

经过以上的讨论,我们了解到,实现几何相交测试的方法是先定义形状的表示方式,然后使用 SAT 找出所有可能出现的分离轴,再判断分离轴是否可分离两个形状。

接下来我们讨论一个动作游戏项目中的具体需求,就是扇形攻击范围与圆形受击范围的攻击判定。这是较少在相关书籍记载的一种测试,而且 Unity 也没有直接提供。

4.1 扇形



已关注

扫掠角度,我们选择使用其中间方向的单位矢量 $\hat{\mathbf{u}}$ 和半角 θ ,如下图。

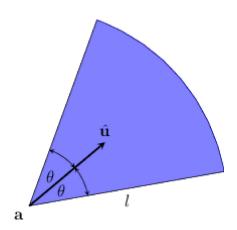


图 4.1:扇形表示方式

然而当 $\theta > \frac{\pi}{2}$ (如吃豆人的形状),这便不是一个凸集,不能使用 SAT。以下我们先考虑 $\theta \leq \frac{\pi}{2}$ 的情况。

4.2 分离轴

我们发现,扇形与圆盘的分离轴共有3种:

- 扇形圆心和圆盘圆心的方向(扇形的圆弧部分)
- 扇形两边的法线
- 扇形 3 个顶点和圆盘圆心的方向

直接应用 SAT 的话,就是使用这 5 个方离轴测试能否把两个形状分离。不过,这可以进一步优化。

4.3 对称性优化

之前在AABB与圆盘相交测试的例子里揭示,利用对称性可以简化一些相交测试。扇形也是一个对称形状,其对称轴刚好为 $\hat{\bf u}$ 。我们可以把 $\hat{\bf u}$ 设为扇形局部坐标系的 x 轴,把该轴向左旋转90度则为 y 轴。把圆盘圆心 $\hat{\bf c}$ 变换至这个局部坐标系(为了简单起见使用了行矩阵):

$$\mathbf{c}' = (\mathbf{c} - \mathbf{a}) \begin{bmatrix} u_x & -u_y \\ u_y & u_x \end{bmatrix}$$

再把 $c_y < 0$ 反射至 $c_y > 0$ 的方向:



已关注

这样,第二和第三种分离轴便变成:

- 扇形左边的法线
- 扇形 2 个顶点(圆心和左边的顶点)与 P 的方向

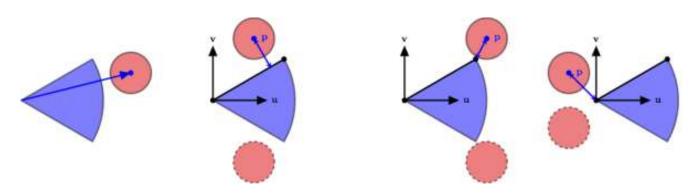


图 4.2:利用对称性优化的步骤

我们还发现,仅当 \mathbf{p} 位于 θ 范围之外,我们才需要检查上述的测试。所以,按照分离轴测试从简单至复杂排序,这个相交测试的过程为:

- 1. 如果扇形圆心和圆盘圆心的方向能分离,两形状不相交。
- 2. 计算出扇形局部空间的 P。
- 3. 如果 $p_x > ||p|| \cos \theta$, 两形状相交。
- 4. 求左边线段 $tl \left[\cos \theta \sin \theta\right]$ 与圆盘是否相交。

```
// 扇形与圆盘相交测试
// a 扇形圆心
// u 扇形方向(单位矢量)
// theta 扇形扫掠半角
// 1 扇形边长
// c 圆盘圆心
// r 圆盘半径
static bool IsSectorDiskIntersect(
   Vector2 a, Vector2 u, float theta, float 1,
   Vector2 c, float r)
{
   // 1. 如果扇形圆心和圆盘圆心的方向能分离,两形状不相交
   Vector2 d = c - a;
   float rsum = l + r;
   if (d.sqrMagnitude > rsum * rsum)
       return false;
```



已关注

```
float py = Mathf.Abs(Vector2.Dot(d, new Vector2(-u.y, u.x)));

// 3. 如果 p_x > //p// cos theta, 两形状相交
if (px > d.magnitude * Mathf.Cos(theta))
    return true;

// 4. 求左边线段与圆盘是否相交
Vector2 q = l * new Vector2(Mathf.Cos(theta), Mathf.Sin(theta));
Vector2 p = new Vector2(px, py);
return SegmentPointSqrDistance(Vector2.zero, q, p) <= r * r;
}
```

由于对称性而作的优化,实际上我们只检测半个扇形,即使 $\theta > \frac{\pi}{2}$,半个扇形还是一个凸集。因此,我们可以把 $\theta \leq \frac{\pi}{2}$ 这个限制除去,支持任意扫掠角度的扇形。

4.4 其他潜在优化

以上的实现还有不少可优化的地方,这里只是简单列出两项:

- 1. $p_x>\|p\|\cos\theta$ 中需要开方来计算模,但这里不可以简单地把左右侧平方,因为在 $\theta>\frac{\pi}{2}$ 的时候, $\cos\theta$ 可能是负数,而 p_x 也可能是负数。可以分别处理这四种符号组合来避免开方运算。
- 2. 对于游戏来说,扇形的位置和方向可能是比较动态的,但半径和扫掠角度可能是较静态的。在静态的情况下,我们可以预计算相关的表达式,例如 $\cos\theta$ 、 $l\left[\cos\theta \sin\theta\right]$ 、 $(l+r)^2$ 等。

5. 总结

本文以一些例子说明基本的几何形状相交原理,也导出扇形与圆盘相交测试的方法,并加以优化。除了本文所提及的形状,游戏中通常还会用到射线、多边形 / 多面体、椭圆 / 椭球体、平截头体、圆柱体等二维 / 三维形状,只要是凸集,都可使用 SAT 作为基础去解决相交问题,读者可以参考[3]中更多的例子。除了这种解析式方法处理相交问题,还可以用一些数值方法,例如著名的 GJK 距离算法[4]。使用这个算法可以避免为 n 种形状实现 n(n-1) 种相交测试,而只需为每种形状提供一个函数。然而,为了性能和精确性起见,一些简单形状的相交测试还是值得专门实现及优化的。

参考



已关注

- [2] Arvo, "A Simple Method for Box-Sphere Intersection Testing", Graphics Gems, pp. 247-250, 1993. tog.acm.org/resources/G...
- [3] Schneider, Philip, and David H. Eberly. Geometric tools for computer graphics. Morgan Kaufmann, 2002.
- [4] Gilbert, Elmer G., Daniel W. Johnson, and S. Sathiya Keerthi. "A fast procedure for computing the distance between complex objects in three-dimensional space." Robotics and Automation, IEEE Journal of 4.2 (1988): 193-203.

「真诚赞赏,手留余香」

赞赏

还没有人赞赏,快来当第一个赞赏的人吧!

计算机图形学 游戏引擎 计算几何



△分享 ① 举报











文章被以下专栏收录



Milo的编程

进入专栏

17 条评论



写下你的评论



无聊时的自娱自乐



已关注

000

3 个月前



弦无音

就喜欢这种纯干货,赞milo大大,从来不废话扯淡

3 个月前



你好

写的真的很用心,收藏

3 个月前



竹山塘一代小白

话说游戏中扇形攻击使用递进的圆半径来实现怎么样(dota1是这样实现的,是因为API太挫)

3 个月前



YKZDY

简洁易懂,m大大有空再讲讲OBB与OBB相交的怎么样,之前项目中遇到我最后直接使用的坐标轴做分离轴。相当于是一个充分非必要判断。再哪个项目里也够用了倒是...

3个月前 1 赞



杨睿涵

向大神学习!

3 个月前



江流

可以请教你关于 triangle-box 交叉检测的问题吗?

3个月前



Milo Yip (作者)回复 江流

② 查看对话

你是微博上问的哪位么?刚看到。其实都是 SAT 的应用。边与边的分离轴在于该两边的差积。

3 个月前



已关注

是的,都是我,抱歉在很多地方留了言。那么我先去找找 SAT 的文章读一读吧。可以的话,能私信个联系方式吗?以后遇到图形学方面的问题,我还想请教你 3 个月前



江流 回复 Milo Yip (作者)

② 查看对话

David Eberly 的 Intersection of Convex Objects: The Method of Separating Axes 这篇文章 也挺有用的,加到参考资料里怎么样?

我同时有一个问题:

2D 空间中,物体的每条边可以方便地投影到分离轴上;但 3D 空间中,如果边和分离轴不共面,那么怎么把这条边投影到分离轴上?

3 个月前

1 2

2 下一页

推荐阅读

如何优雅地判断 N 个布尔值是否全部相等?

原问题:https://www.zhihu.com/question/52881172,原来是指定 Jav aScript 的。原答案:return arr.indexOf(!arr[0])<0; (题图 Photo by... 查看全文 >

Milo Yip · 3 个月前

发表于 Milo的编程

程序员可以学画画吗?

我大约十年前(2006年末)和太太一起去学过画画。我在初中没上过美术课(当时选了设计与工艺课,即是学金工木工),所以应该算是零基础吧。(…查看全文>

Milo Yip · 3 个月前

发表于 Milo的编程

-个测量工具的优劣?

3知者明。--《道德经》人事(HR)决策都属于重大决策。比如选拔,如何从多个人中找到最合适的人选对企业的发展至关重要。对于个人来说,是否被选中关乎着



已关注

去怡保,一天吃八顿!

说起马来西亚,大家都知道吉隆坡、槟城、马六甲、沙巴,却甚少有人提起过这样一座城市:李安导演大名鼎鼎的《色·戒》、阿牛执导的《初恋红豆... 查看全文》

穷游锦囊·3天前·编辑精选

发表于 对世界上瘾