



虚拟现实技术

Virtual Reality Technology

金枝

中山大学智能工程学院 2019秋季课程



本堂课内容

- 复习上堂课内容
- 虚拟现实的关键技术——碰撞检测技术
 - 1 碰撞检测技术的意义
 - 2 碰撞检测技术内容
 - 3 面向凸体的碰撞检测
 - 4 基于层次包围盒的碰撞检测



本堂课内容

- 复习上堂课内容
- 虚拟现实的关键技术——碰撞检测技术
 - 1 碰撞检测技术的意义
 - 2 碰撞检测技术内容
 - 3 面向凸体的碰撞检测
 - 4 基于层次包围盒的碰撞检测



复习题

填空题

1. 请简述立体声、立体环绕声和三维虚拟声音的差别。

立体声音有左右声道之分，但就整体效果而言，立体声音来自听者面前的某个平面；立体环绕声保留原信号的声源方向感，并伴随产生围绕感和扩展感的音响效果，立体环绕声可使空间声源由线扩展到整个水平面；三维虚拟声音是来自围绕听者双耳的一个球形中的任何地方，即声音出现在头的上方、后方或者前方。



复习题

填空题

1. 请简述立体声、立体环绕声和三维虚拟声音的差别。
2. 三维虚拟声音的两个主要特征：_____和_____。

全向三维定位特性

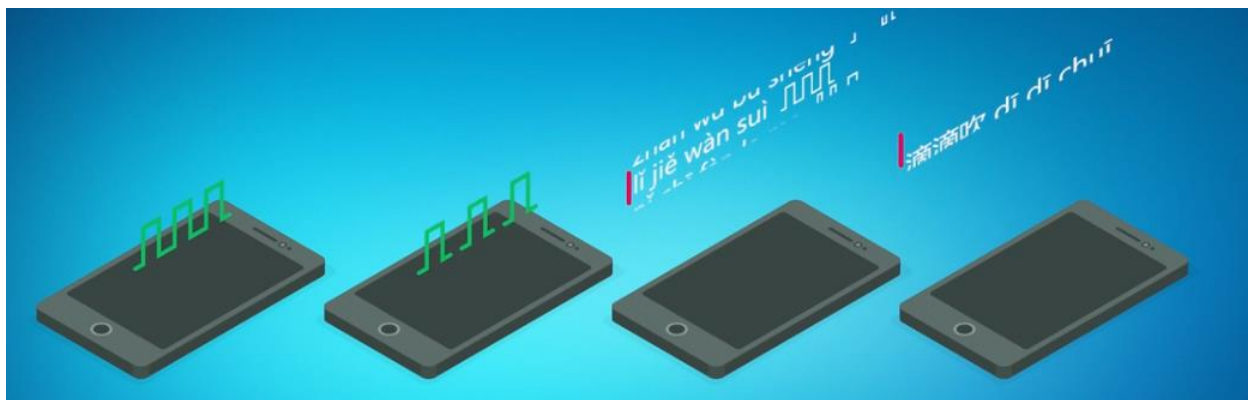
三维实时跟踪特性



复习题

填空题

1. 请简述立体声、立体环绕声和三维虚拟声音的差别。
2. 三维虚拟声音的两个主要特征：_____和_____。
3. 请简述语音识别技术。





复习题

填空题

1. 请简述立体声、立体环绕声和三维虚拟声音的差别。
2. 三维虚拟声音的两个主要特征：_____和_____。
3. 请简述语音识别技术。
4. 在虚拟现实领域中较为常用的交互技术主要有：
等：____、____、____和_____。

手势识别、面部表情识别、眼动跟踪、语音识别



本堂课内容

- 复习上堂课内容
- 虚拟现实的关键技术——碰撞检测技术
 - 1 碰撞检测技术的意义
 - 2 碰撞检测技术内容
 - 3 面向凸体的碰撞检测
 - 4 基于层次包围盒的碰撞检测



1 碰撞检测技术意义

- 碰撞检测是虚拟现实中另一个重要的部分，它的主要作用是检测游戏中各种物体的物理边缘是否产生了碰撞。游戏的效果必须在一定程度上符合客观世界的物理规律，如地心引力、加速度、摩擦力、惯性和碰撞检测。
- 碰撞检测是虚拟现实中不可回避的问题之一，只要场景中的物体在移动就必须判断是否与其他物体相接触。





2 碰撞检测技术内容

- 碰撞检测的基本任务是确定两个或多个物体彼此之间是否有**接触或穿透**，并给出**相交部分的信息**。碰撞检测之所以重要，是因为现实世界中，两个或多个物体不可能同时占有同一空间区域。如果物体之间发生了穿透，用户会感觉不真实，从而影响VR体验的沉浸度。
- 由于碰撞检测的基本问题是物体的**求交**，直观的算法是两两检测场景物体之间的位置关系。对于复杂的三维场景，显然复杂度为 $O(N^2)$ 检测算法无法满足VR实时性的要求。**设计高效的碰撞检测算法是编程的难点。**



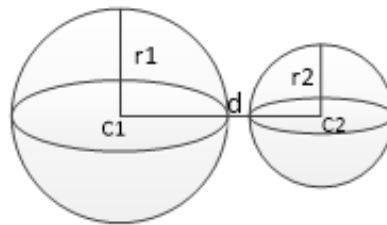
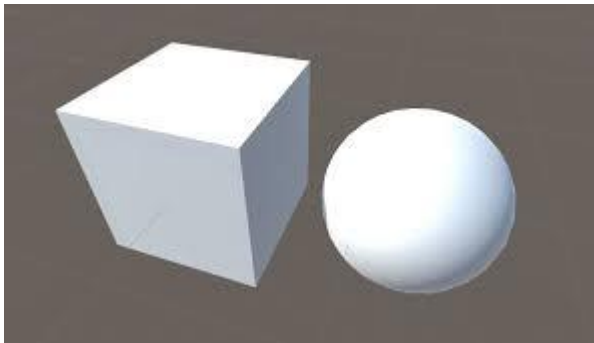
本堂课内容

- 复习上堂课内容
- 虚拟现实的关键技术——碰撞检测技术
 - 1 碰撞检测技术的意义
 - 2 碰撞检测技术内容
 - 3 面向凸体的碰撞检测
 - 4 基于层次包围盒的碰撞检测

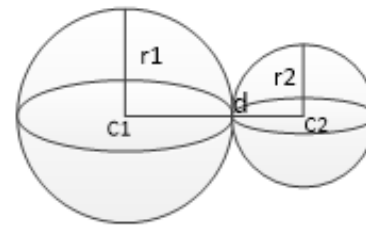


3 面向凸体的碰撞检测

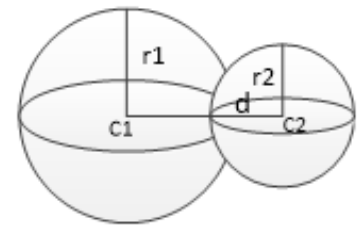
- 面向凸体的碰撞检测算法大体上又可以分为两类：
 - 基于特征的碰撞检测算法；
 - 基于单纯形的碰撞检测算法；



(a) $d > r_1 + r_2$ 不相交



(b) $d = r_1 + r_2$ 相切



(c) $d < r_1 + r_2$ 相交



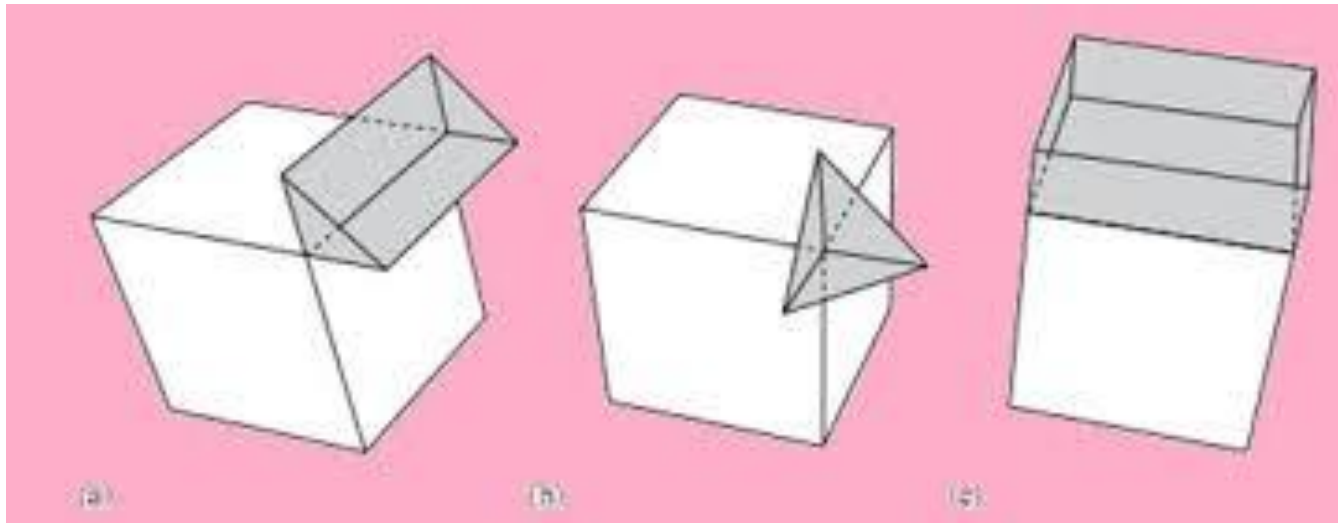
3.1 基于特征的碰撞检测算法

- 顶点、边和面成为多面体的特征。基于特征的碰撞检测算法主要通过判别两个多面体的顶点、边和面之间的相互关系进行他们之间的相交检测—— **Lin-Canny算法**。
- Lin-Canny算法的主要用途即是求解两个凸包几何体间的**最近距离（或最近点对等）**。它的实现基础首先是将几何体元进行分类，即：点、边（三维线段）、面（有限面片），进而将两个凸包中的几何体元组成基本的几何体元对：**点-点、点-边、点-面、边-边、边-面、面-面**。如此一来，对于两个几何体之间的最近点对可能出现的位置即归结到了上述六种体元对之间。



3.1 基于特征的碰撞检测算法

- LC的高效实现中，另外一个关键的概念即是**三维Voronoi区域**（3D Voronoi Area, VA），它是二维Voronoi区域的升维扩展。



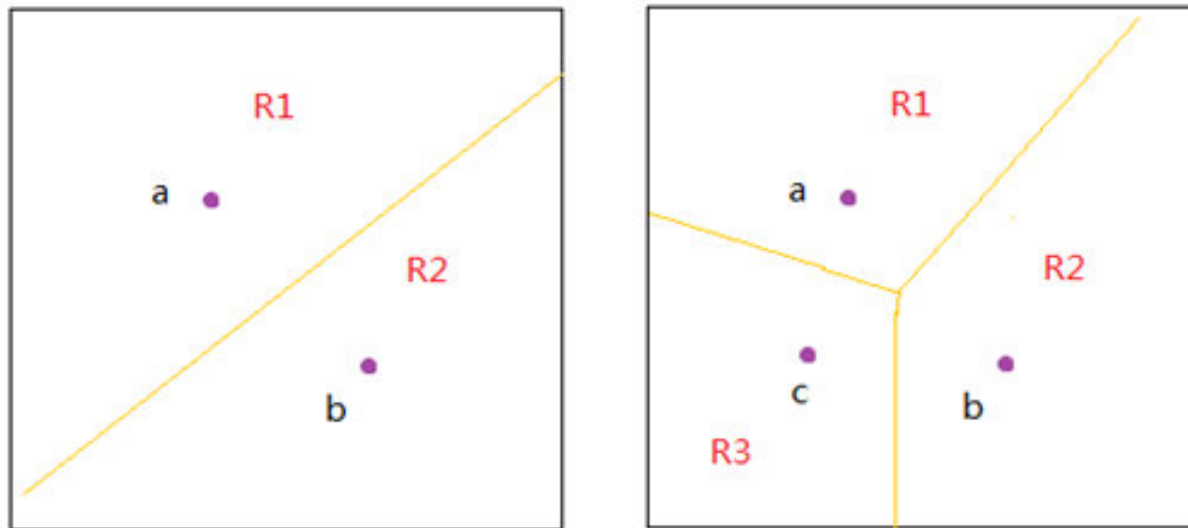
一个立方体特征的Voronoi区域 a)边 b)点 c)面

注：Voronoi为泰森多边形



3.1 基于特征的碰撞检测算法

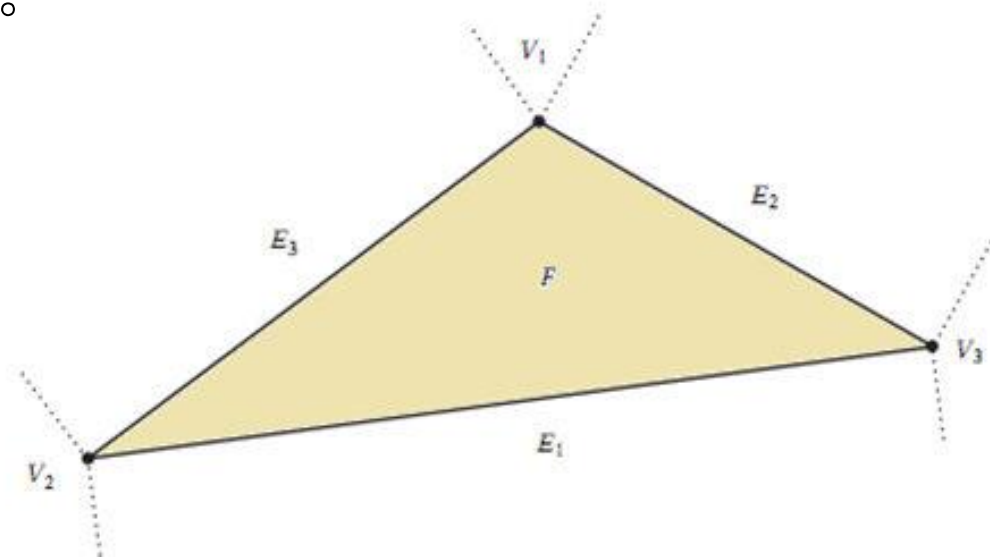
- 什么是Voronoi区域呢？
- 假定在平面上有一个点集合 S ， a 是 S 中的一个点，则点 a 的Voronoi区域定义为：离点 a 的距离比离 S 中其它点的距离都要近的平面上的点集合。如图1中所示，点 a 的Voronoi区域是 $R1$ ，点 b 的Voronoi区域是 $R2$ ，点 c 的Voronoi区域是 $R3$ 。





3.1 基于特征的碰撞检测算法

- 在碰撞检测算法中，对Voronoi区域的概念稍有扩展。
- 给定一个多面体 P ，它的特征包括：顶点、边、面以及多面体本身（多面体内部空间），则特征 F 的Voronoi区域定义为：**空间中**离特征 F 比其它任何特征都近的点的集合（不包括多面体内部）。





3.1 基于特征的碰撞检测算法

- 得到了凸包中的每个基本几何体元所对应的VA之后，即可进行基于VA的LC算法。
 1. 点-点：对于来自不同几何体的两个顶点，判断其是否是最近点对。如果这两个点组成了最近点对，那么它们之间必定满足：**A处于B所对应的VA域内；B处于A所对应的VA域内。**
 2. 点-边：对于由点和边组成的体元对，**首先求解出点到该边（即线段）间的最近点对，接着判断该最近点对是否是原始凸包的最近点对。**假设点A到边E上的最近点为P，则判断标准为：**A处于E所对应的VA域内；P处于A所对应的VA域内。**如果条件满足，则可以得到由边点计算而来的最近点对（A,P）。



3.1 基于特征的碰撞检测算法

- 3. 点-面，此种情况需要先求解出点到有限面上的最近点对，接着进行VA检验。假设A到面F上的最近点为P：A处于F所对应的VA域内；P处于A所对应的VA域内。

有了上述三种基于体元对间的VA判断以后，基本上相当于已经实现了LC算法，对于其它三种基本体元对的判断，均是做相应的处理之后转为上述三种情况中的一种，并接着进行算法的判断操作，直至得到最终的解。



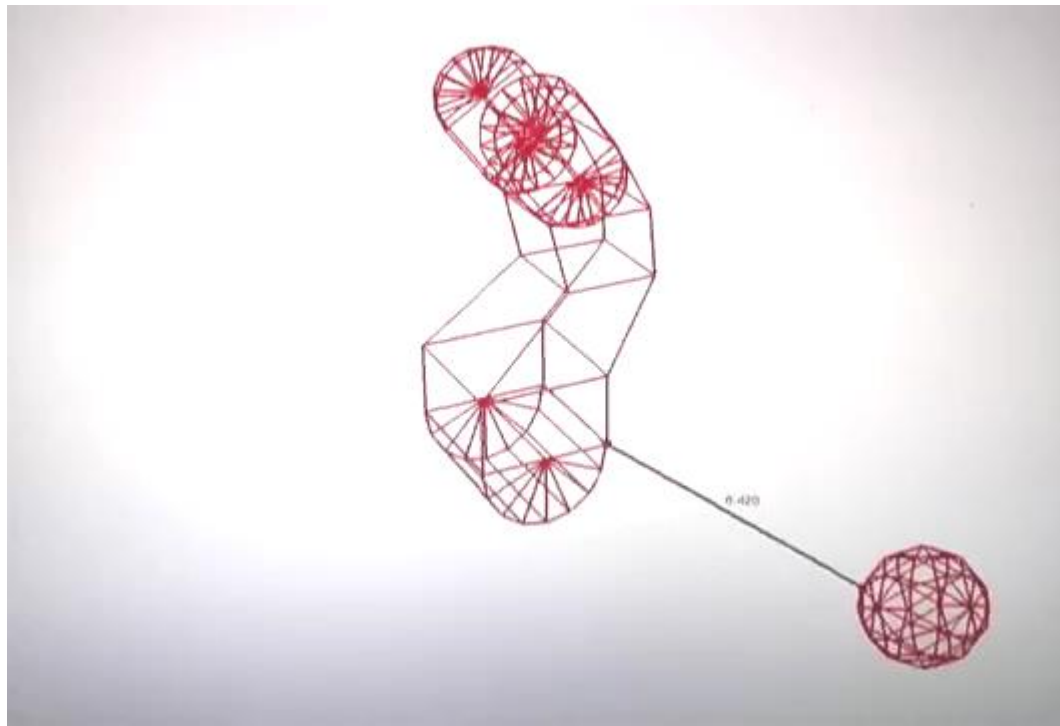
3.1 基于特征的碰撞检测算法

4. **边-边**: 首先, 求解出边-边 (线段-线段) 间的最近点对, 这里要保证得到的最近点对均处于线段之上。进而分别以各个点与另外一条边为体元对, 做情况2的判断。
5. **边-面**: 同样, 求解出边-面间的最近点对, 类同于情况4归结为前3种情况中的一种后继续进行判断。
6. **面-面**: 一般情况下来讲, 对于两个凸包间的最近点对分别出现于个集合的面片上的情况较少, 因而在此处的判断方法同样是做类型变换。先判断是否平行, 若两个面片不平行或是平行而无重叠, 则**求出两个面片之间的最近边对**来组成新的几何体元对接着进行之后的检测。



3.1 基于特征的碰撞检测算法

- LC算法的实现主要以上述几种基本体元之间的情形做为基本情形进而进行**渐进式的近点对求解**，而且每两次基本体元对之间的转换必定使得求得到有效距离变得更小，直到求出最终的最近点对时得到最小的凸包间距离。





3.2 面向单纯形的碰撞检测算法

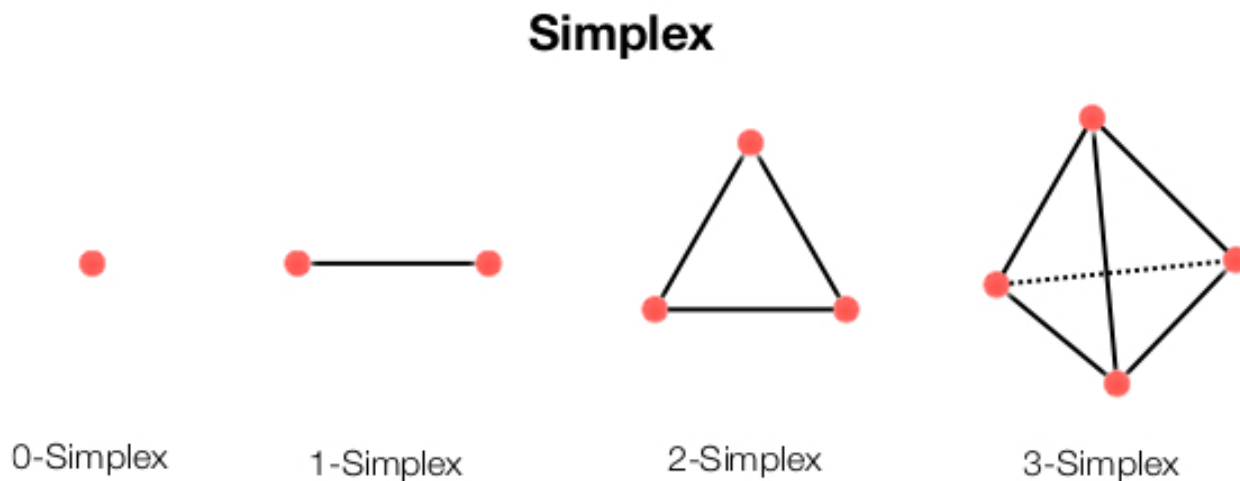
- GJK算法
- 单纯形 (Simplex)
- 闵可夫斯基 (Minkowski) 和&闵可夫斯基差



3.2.1 单纯形 (Simplex)

- 单纯形指的是点，线段，三角形或四面体。

例如，0-simplex是一个点，1-simplex是一个线段，2-simplex是一个三角形，而3-simplex是一个四面体。



copyright haroldserrano.com

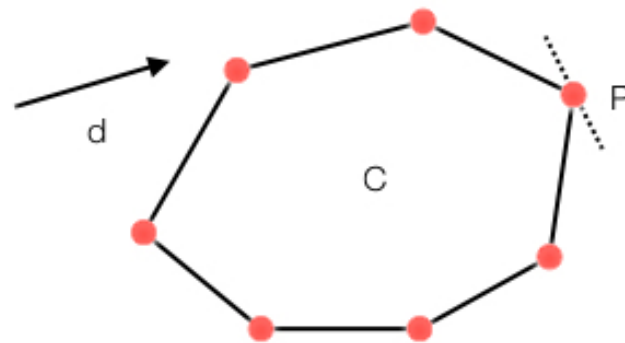


3.2.2 支持点 (Supporting Point)

- 在凸的物体中，支撑点是给定方向上最远的点。

例如，在下图中，d方向的支持点为P。

Supporting Point



copyright haroldserrano.com



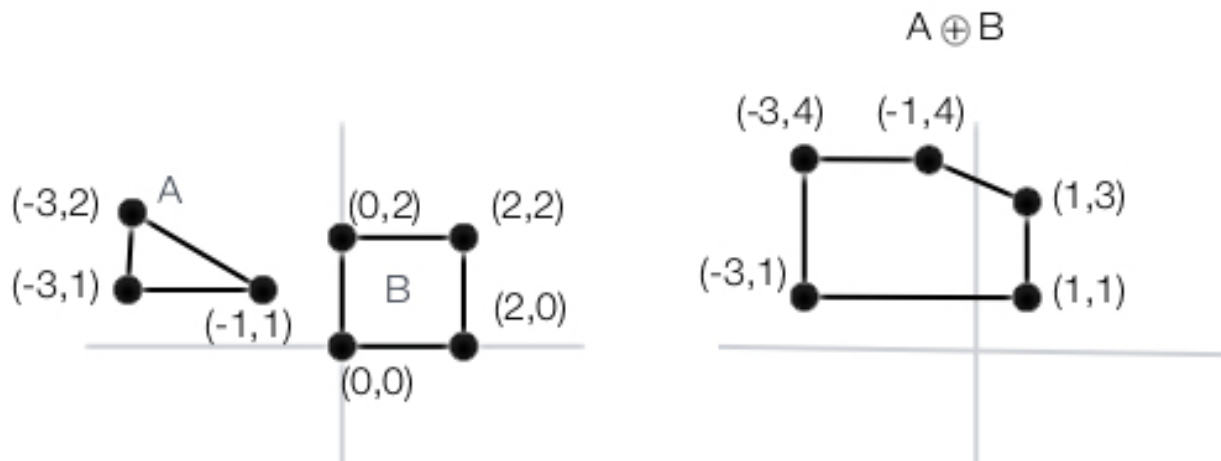
3.2.3 闵可夫斯基 (Minkowski) 和

- 闵可夫斯基和：是物体1上的所有点和物体2上的所有点的（坐标）和集。用公式表示就是：

$$A + B = \{a + b \mid a \in A, b \in B\}$$

- 如果两个物体都是凸体，它们的闵可夫斯基和也是凸体。

Minkowski Sum





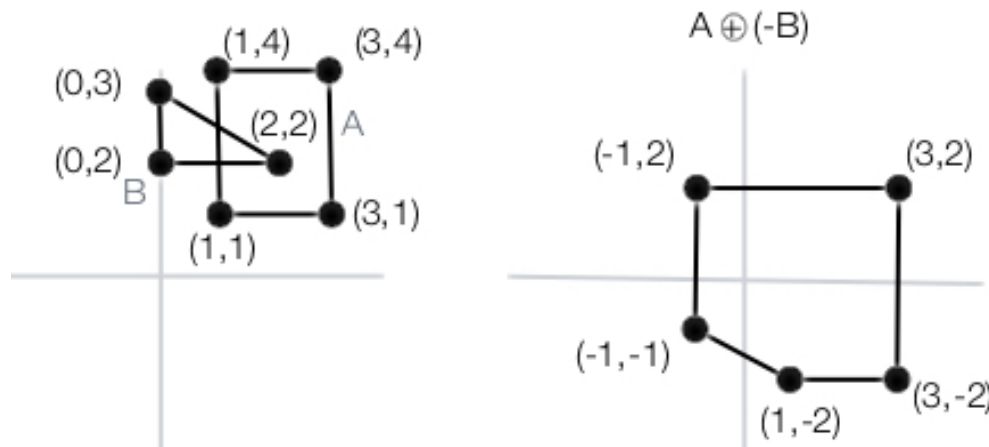
3.2.3 闵可夫斯基差

- 闵可夫斯基差：是物体1上的所有点和物体2上的所有点的（坐标）差。用公式表示就是：

$$A - B = A + (-B) = \{a + (-b) \mid a \in A, b \in B\} = \{a - b \mid a \in A, b \in B\}$$

- 如果两个物体重叠或者相交，它们的闵可夫斯基差肯定包括原点。

Minkowski Difference





3.2.4 GJK算法

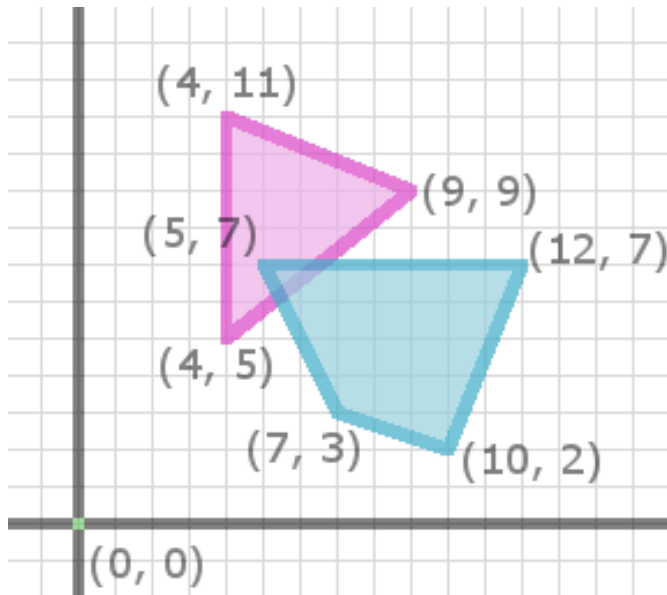


图1: 两个凸体相交

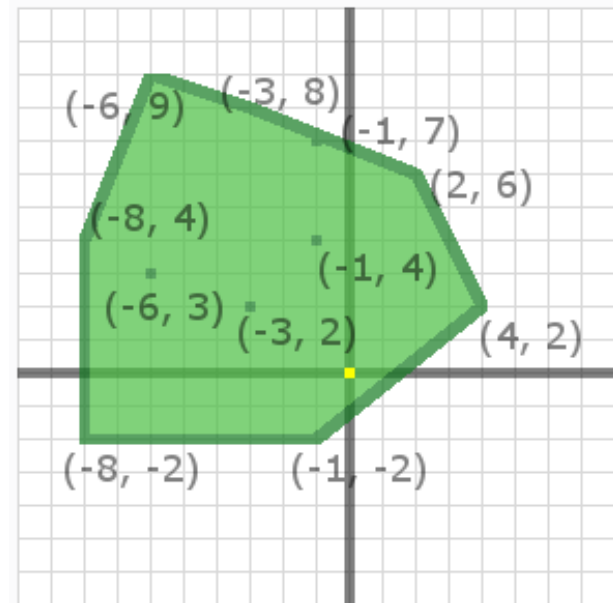


图2: 明可夫斯基差

我们不需要显式计算物体之间的闵可夫斯基差，只要知道它们的闵可夫斯基差是否包含原点就ok了。如果包含原点，物体之间就相交，否则，则不相交。



3.2.4 GJK算法

- 我们可以在闵可夫斯基差形成的物体内**迭代形成**一个单纯形，并使这个单纯形尽量包围原点。如果这个单纯形包含原点，显然闵可夫斯基差形成的物体必然包括原点。

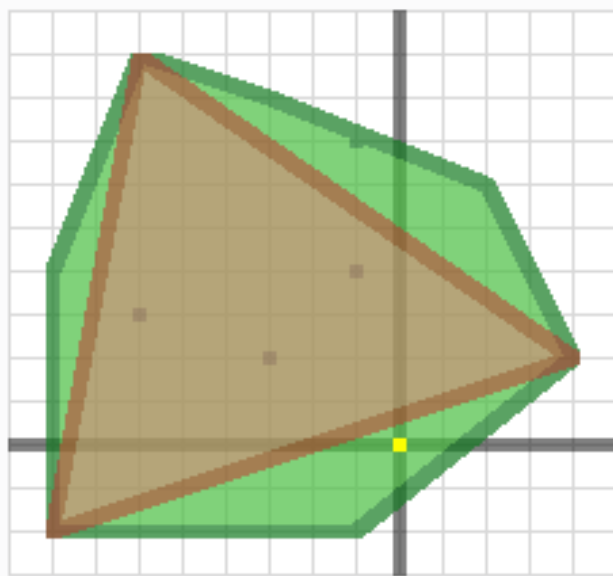


图3: 单纯形例子

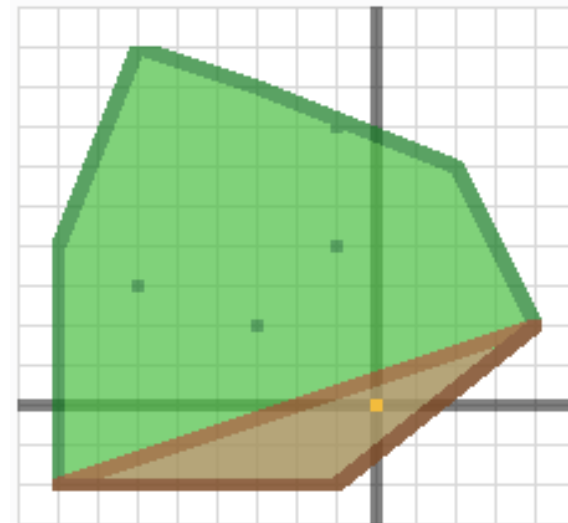


图4: 包含原点单纯形的例子



3.2.4 GJK算法

- 我们可以在闵可夫斯基差形成的物体内迭代的形成一个单纯形，并使这个单纯形尽量包围原点。如果这个单纯形包含原点，显然闵可夫斯基差形成的物体必然包括原点。

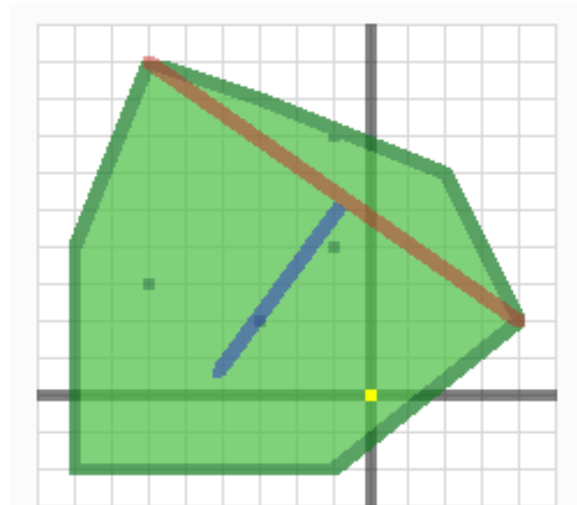


图 5: 第一次迭代

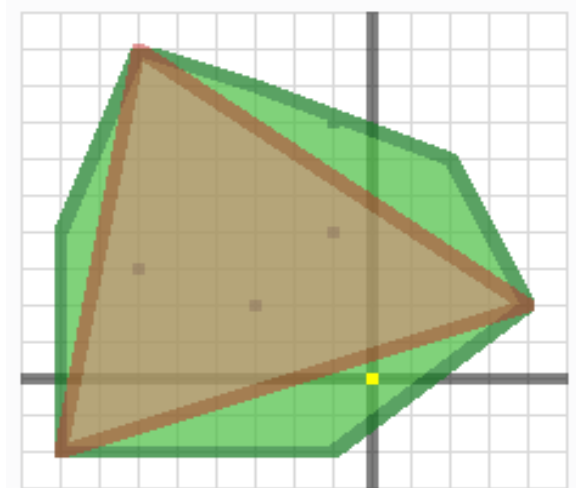


图 6a: 第二次迭代，产生新的单纯形

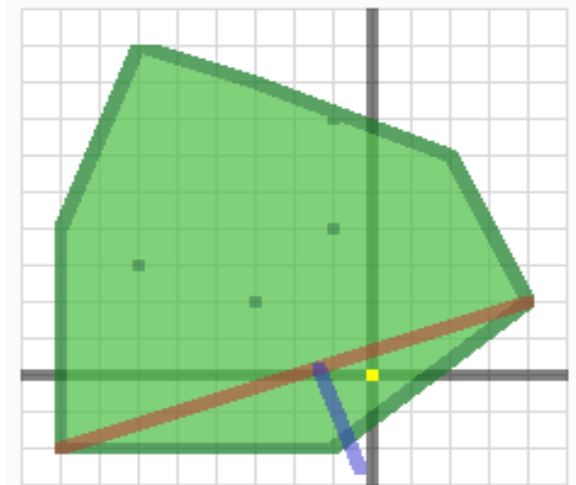
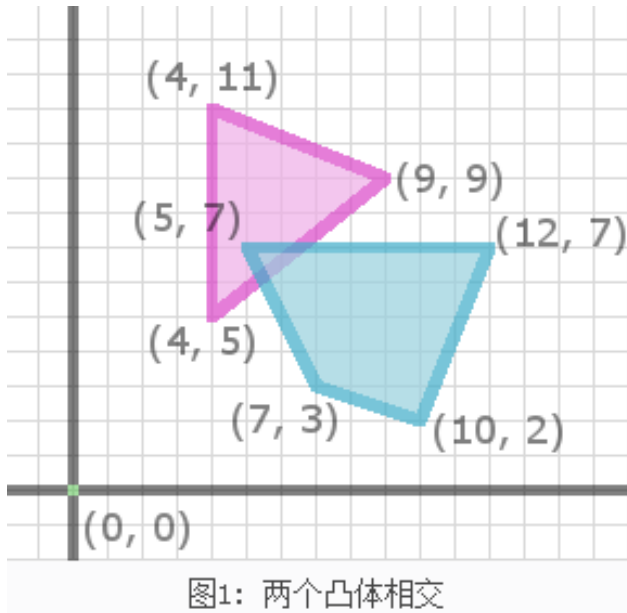


图 6b: 第二次迭代，新的单纯形以及方向



3.2.4 GJK算法



起始

$$d = (1, 0)$$

$$p1 = (9, 9);$$

$$p2 = (5, 7);$$

$$p3 = p1 - p2 = (4, 2);$$

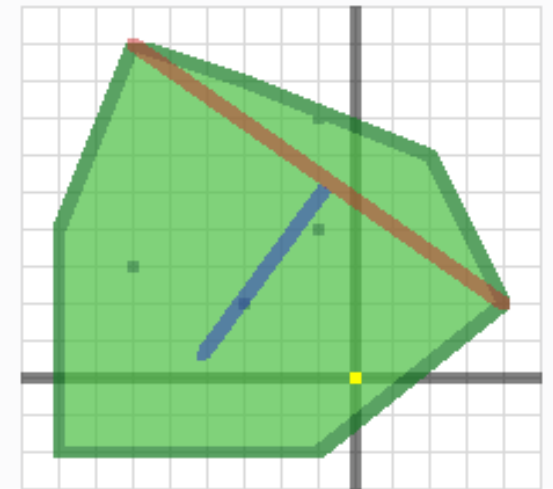
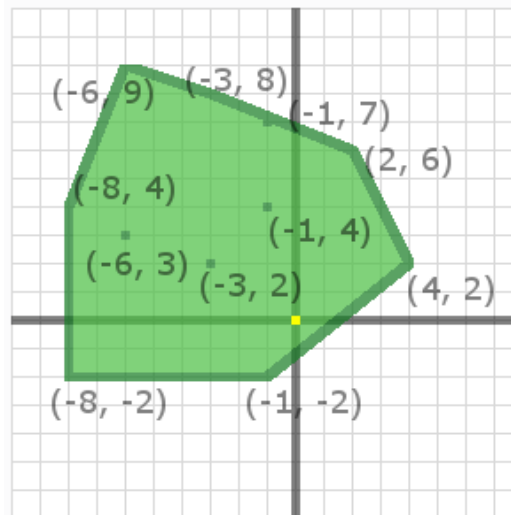
第一次迭代

$$d = (0, 1)$$

$$p1 = (4, 11);$$

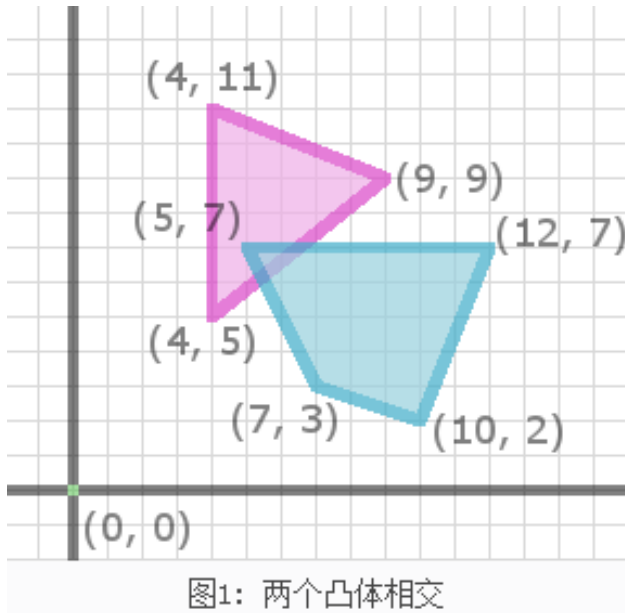
$$p2 = (10, 2);$$

$$p3 = p1 - p2 = (-6, 9);$$





3.2.4 GJK算法



第二次迭代

$$d = (-1, 0)$$

$$p1 = (4, 5);$$

$$p2 = (12, 7);$$

$$p3 = p1 - p2 = (-8, -2);$$

第三次迭代

$$d = (0, -1)$$

$$p1 = (4, 5);$$

$$p2 = (5, 7);$$

$$p3 = p1 - p2 = (-1, -2);$$

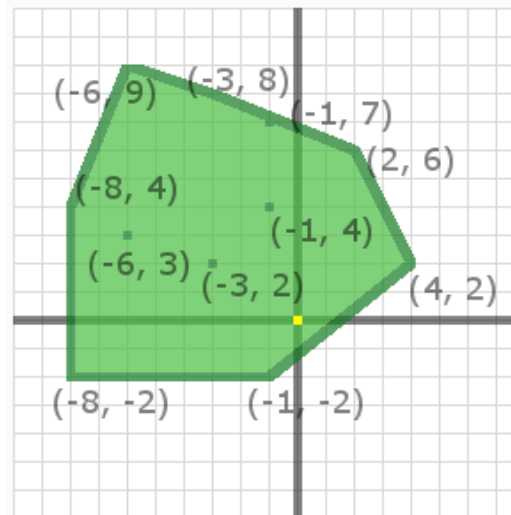
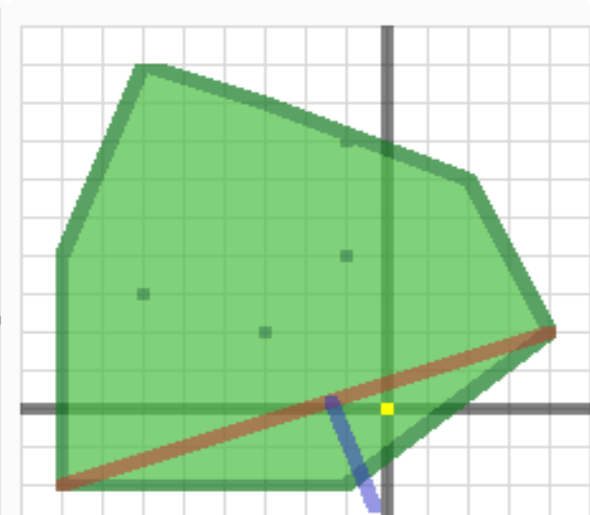


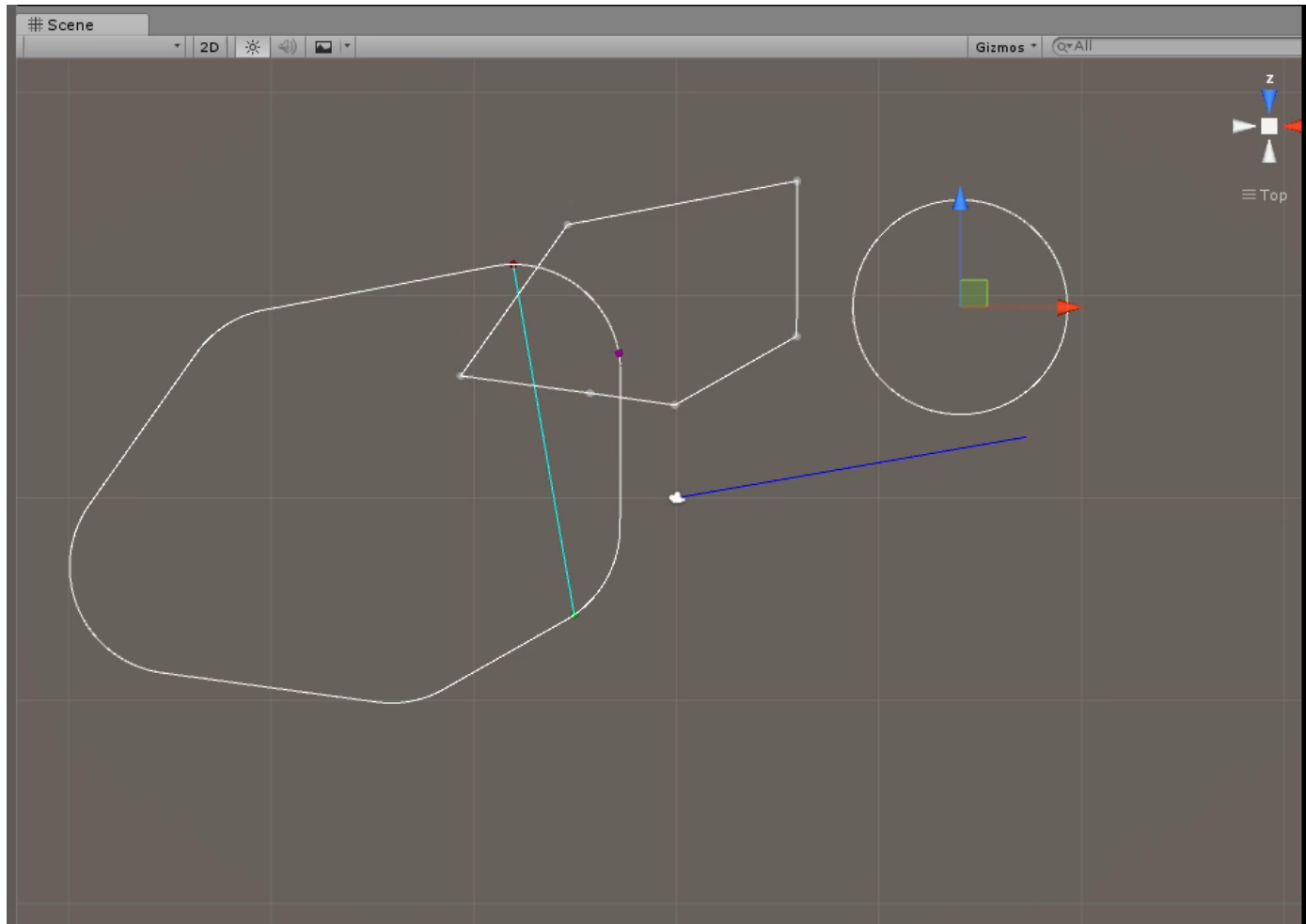
图2: 明可夫斯基差





3.2.4 GJK算法

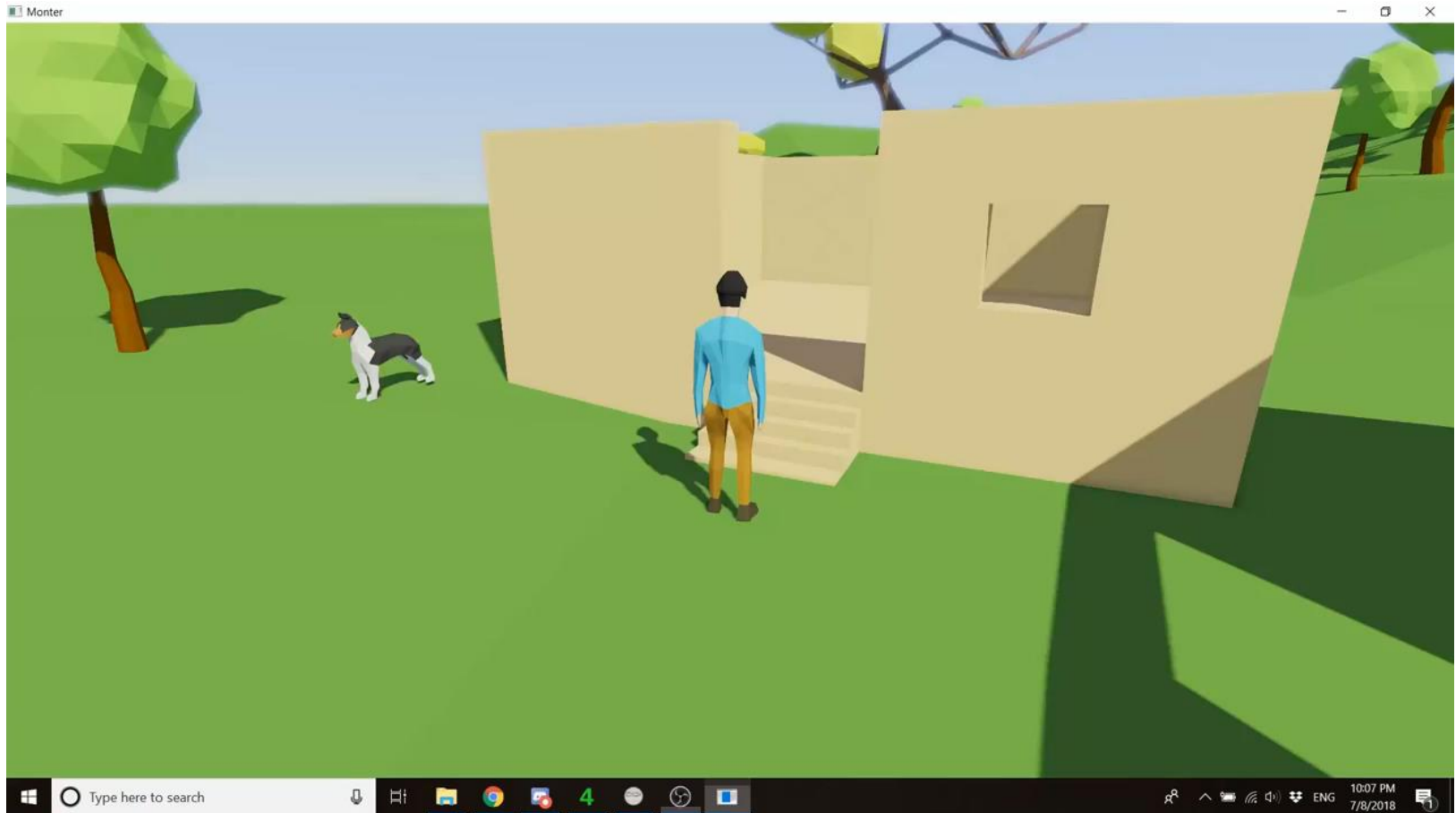
- GJK算法的可视化





3.2.4 GJK算法

- 游戏中GJK算法的碰撞检测和反响





Thanks !