第13章　求交测试方法

金小刚　**Email: jin@cad.zju.edu.cn**

*浙江大学**CAD&CG**国家重点实验室*

紫金港校区 信息与控制大楼506

图形学中经常需要进行求交测试...

- 用鼠标单击屏幕上的物体进行物体拾取
- 判断两个物体是否相碰撞
- 光线跟踪绘制中光线与物体的求交
- 。。。

- 一般来说，求交测试属于图形流水线中的应用层。

OpenGL提供的拾取方法

- 将所有的物体绘制到一个比较小的拾取窗口，所有覆盖在该窗口上的物体都会返回一个表，同时附有每个物体的最大和最小z深度值。
- 优点：不仅可以拾取面，而且可以拾取线和点。

求交测试 vs OpenGL提供的拾取

- 但求交测试具有一些OpenGL拾取方法不具有的优点。
- **Case Study**：判断一条光线是否穿过一个三角形？
- **OpenGL拾取**：首先需要用拾取窗口对该三角形进行裁剪，再判断是否选中该三角形。
- 求交测试：直接进行光线与三角形之间的求交测试要比OpenGL快得多，而且可以计算出交点位置、法向、纹理坐标等数据。*独立于具体的API*。

视域四棱锥裁剪(View Frustum Culling)

- 是一种将视域四棱锥外的物体进行快速剔除的方法。
- 用于判断一个包围盒
  - 是否完全在视域四棱锥内
  - 是否完全在视域四棱锥外
  - 部分在在视域四棱锥内

基于层次包围盒的拾取方法

- 拾取问题可以采用层次包围盒来解决。
- 首先，计算出视点到用户拾取象素位置之间的光线。
- 然后，用递归方法测试这条光线是否与场景图中的包围盒相交。如果光线与某个包围盒无交，则可以忽略该包围盒所在的子树。否则，继续对该包围盒的子包围盒进行上述测试，直到包含几何体的叶子节点。
- 最后，将光线与叶子节点包含的所有图元进行求交测试。

碰撞检测算法中的图元

- 碰撞检测算法通常建立在层次结构上，涉及判断两个图元是否发生碰撞。
- 图元类型：
  - 三角形
  - 球体
  - 轴对齐包围盒 (**Axis-Aligned Bounding Box，AABB**)
  - 有向包围盒(**Oriented Bounding Box, OBB**)
  - 离散有向多面体 (**Discrete Oriented Polytope，*k*-DOP**)

求交测试结果

- **A与B是否相交**
  - **A**完全在**B**内（或者**B**完全在**A**内）
  - **A与B**完全分离
  - **A与B**相交
- 相交信息
  - 精确的相交位置
  - 与相交位置点的距离
  - 交点处的法向、纹理坐标等信息

硬件加速拾取方法

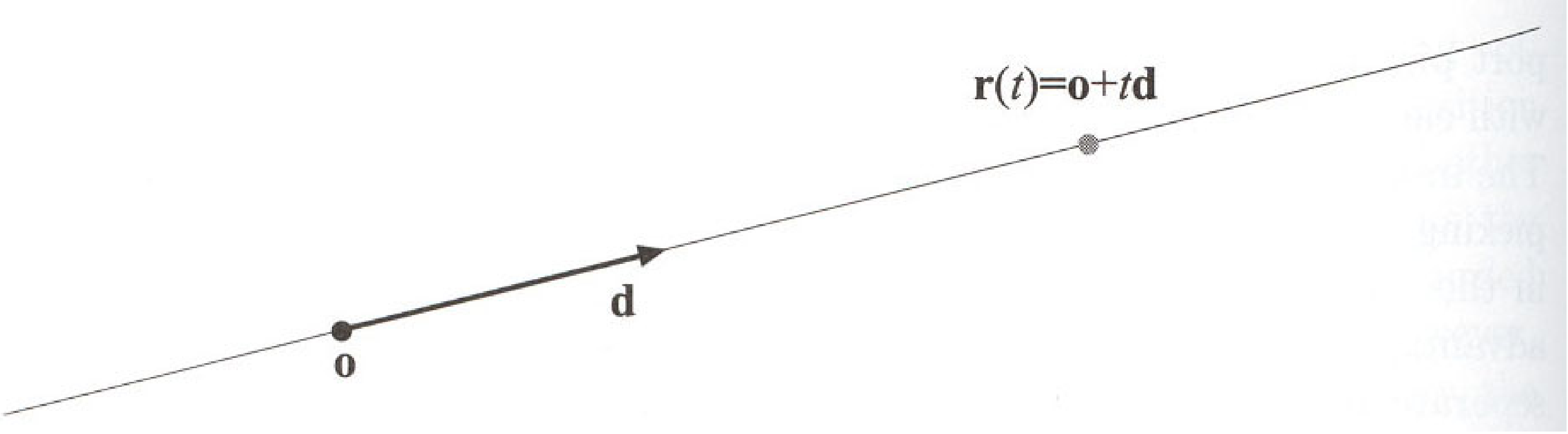
- **Hanrahan和Haeberli提出的拾取方法**
  - 将每个多边形用一种唯一的颜色作为标识，然后把整个场景以没有光照的方式绘制到缓冲器。
  - 形成的图象以离屏的方式进行存贮。当用户点击到一个象素时，通过颜色标识就可以很快地确定相应的多边形。
- 缺点：必须用一个单独的通道来绘制整个场景

硬件加速拾取方法

- 借助颜色缓冲器找到点在三角形中的相对位置
  - 把三角形三个顶点的颜色分别设为红**(255,0,0)**、绿**(0,255,0)**、篮**(0,0,255)**，用**Gouraud Shading**绘制每个三角形。
  - 假设拾取的颜色象素为**(23,192,40)**，则红色顶点贡献了**23/255**，绿色顶点贡献了**192/255**，篮色顶点贡献了**40/255**（实际上这些值就是拾取点相对于三角形的*面积坐标*）。利用这些信息可以找到拾取点在三角形中的真实位置及**(u,v)**坐标。
  - 类似，通过把三角形顶点法向的每个分量从**[-1,1]**变换到**[0,1]**，可把顶点法向变换为**RGB**颜色，利用**Gouraud Shading**绘制后，就可以从颜色缓冲器中读出任意点的法向。

定义和工具

- 射线及其参数



一条射线及其参数：**r**(t)=**o**+t **d**，其中**o**为原点，**d**是射线方向(归一化为单位矢量)，t 为参数

表面的定义

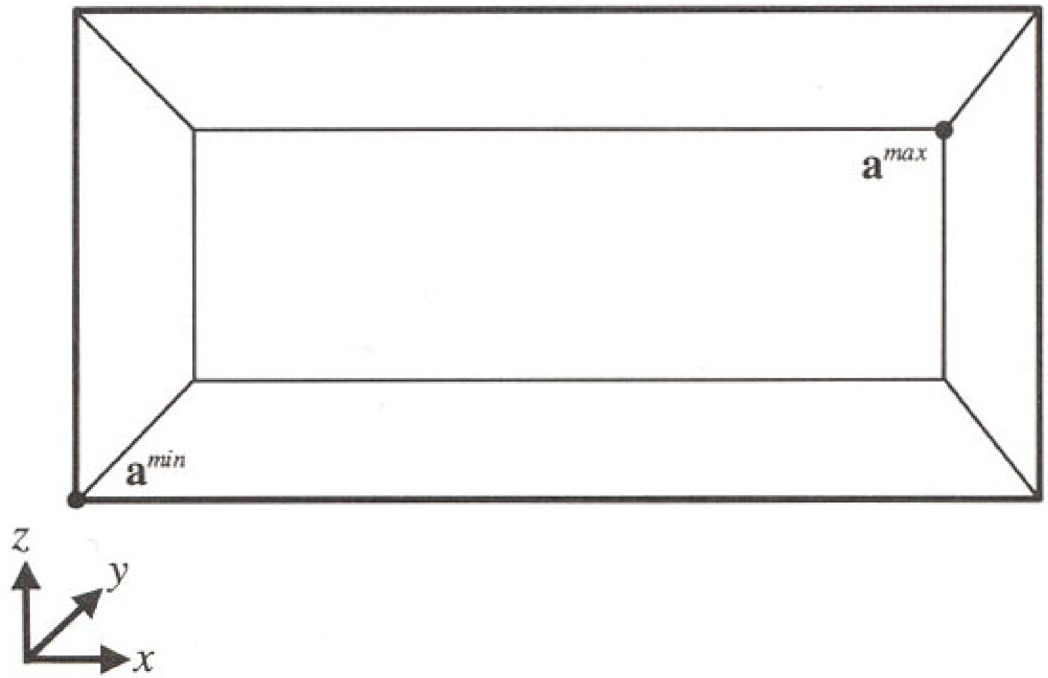
- 隐式表面 $f(\mathbf{P}) = f(p_x, p_y, p_z) = 0$
  - 例子: 中心在原点,半径为r 的球

$$p_x^2 + p_y^2 + p_z^2 - r^2 = 0$$

- 显式表面
  - 三角形▲$\mathbf{v}_0\mathbf{v}_1\mathbf{v}_2$表示为显示形式

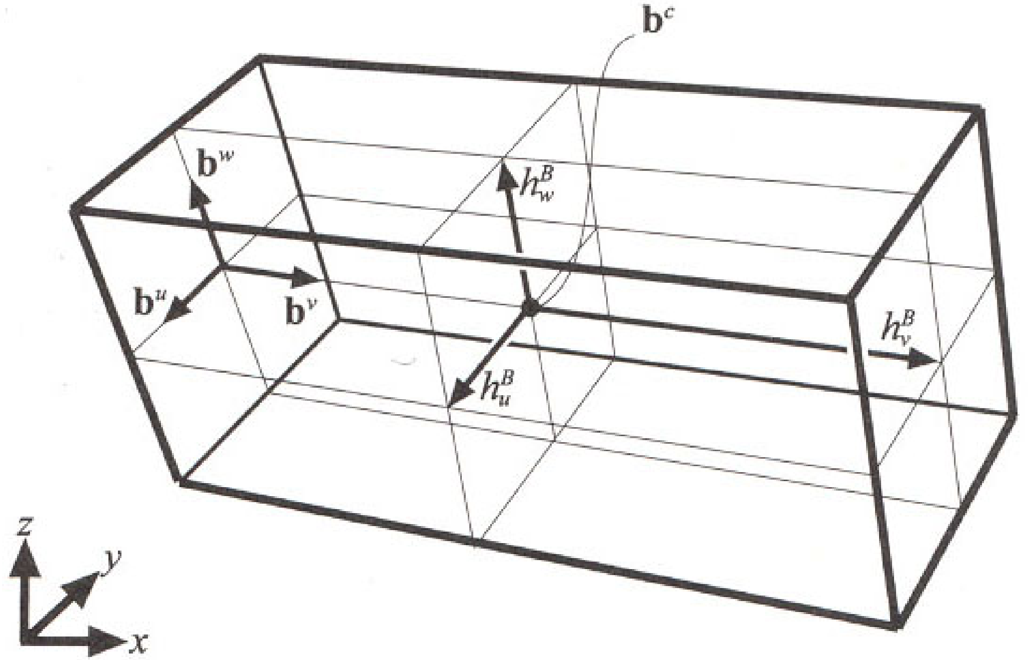
$$\mathbf{t}(u,v) = (1-u-v)\mathbf{v}_0 + u\mathbf{v}_1 + v\mathbf{v}_2$$

AABB包围盒的定义



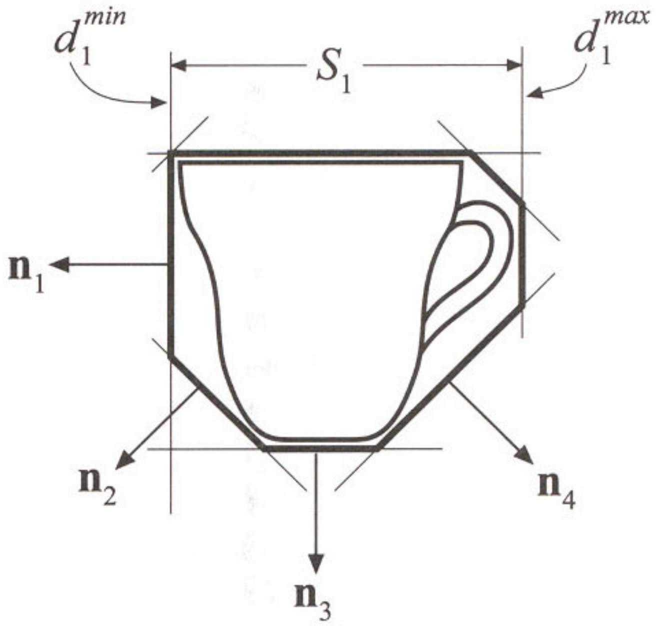
- 轴对齐包围盒(**Axis-Aligned Bounding Box, AABB**)
  - 是一个表面法向与坐标轴方向一致的长方体。
  - 可以用两个顶点$\mathbf{a}^{min}$和$\mathbf{a}^{max}$来表示物体A的AABB，其中 $\mathbf{a}_i^{min}<\mathbf{a}_i^{max}$, $i \in \{x, y, z\}$

OBB包围盒的定义



- 有向包围盒(**Oriented Bounding Box, OBB**)
  - 是一个表面法向两两垂直的长方体。是一个可以任意旋转的**AABB**
  - 可以用它的中心点$\mathbf{b}^c$、三个归一化向量$\mathbf{b}^u$、$\mathbf{b}^v$、$\mathbf{b}^w$以及半边长h_u^B 、h_v^B 、h_w^B来描述。

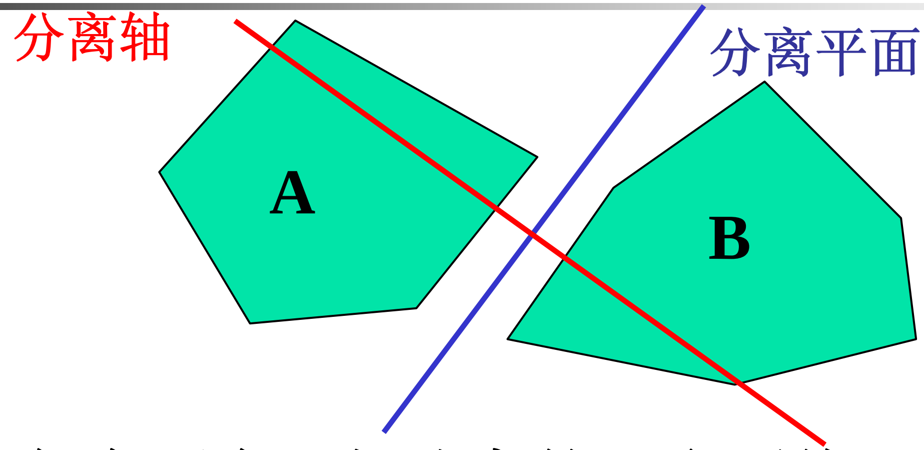
离散有向多面体*k*-DOP的定义



一个茶杯的二维**8-DOP**示例，包括所有法向和零号平板。

- 由$k/2$(*k*是偶数)个归一化法向$\mathbf{n}_i$来定义$(1 \leqslant i \leqslant k/2)$，每个$\mathbf{n}_i$有两个相关标量值$d_i^{\min}$和$d_i^{\max}$，其中$d_i^{\min} < d_i^{\max}$。每个三元组$(\mathbf{n}_i, d_i^{\min}, d_i^{\max})$描述一个平板层$S_i$，表示两个平面之间的空间。这两个平面是$\mathbf{n}_i.\mathbf{x}+d_i^{\min}=0$和$\mathbf{n}_i.\mathbf{x}+d_i^{\max}=0$。所有平板层的交集为***k*-DOP**的实际空间。

分离轴定理(Separating Axis Theorem, SAT)



- 对于任意两个互相分离的凸多面体A和B来说，存在一条分离轴，其中这两个多面体在这条轴上具有一定间隔而且在轴上的投影也是互相分开的。这条轴正交于下列其一(也就是由一个平面分开，该平面平行于下列其中一个平面)：
  - A的一个面
  - B的一个面
  - 多面体的一条边

注：这里指的广义上的凸多面体(包括线段和三角形等)

- 相交测试的优化方法
  - 排除检测：通过在早期进行一些简单的计算来判断射线或物体是否与其它物体完全分开
  - 投影：将三维物体投影到最优的一个正交平面上(xy, xz或yz)，然后在二维平面上进行处理。
- ε问题
  - 由于计算精度问题，相交测试常常会用到非常小的数字，用ε来表示。
  - 选择ε的值：具体问题具体分析。

包围盒的创建

- 给定一组物体，选择合适的包围盒非常重要，因为可以将相交测试的代价最小化。
- 任意一条射线与凸体相交的概率与该物体的表面积大小成正比。因为排除过程通常比直接计算射线与物体的交点要快，所以将表面积最小化可以有效提高相交算法的效率。
- 在碰撞检测算法中，最好将包围体的体积最小化。

AABB与k-DOP的创建

- **AABB**的创建
  - 沿每条轴线方向分别取给定多边形组中顶点的最大值和最小值，就可得到**AABB**。

- **k-DOP**的创建
  - 将顶点投影到**k-DOP**的每条法线方向$\mathbf{n}_i$上，然后将投影的极大值和极小值分别保存在$d_i^{\min}$和$d_i^{\max}$中，这两个值定义了该方向上最紧密的平板层。所有这些值共同定义了一个最小的**k-DOP**。

包围球体的创建

- 算法很多，取决于速度和质量之间的折中。
- 恒时单遍算法(**Constant Time Single Pass**)
  - 先为多边形组构建一个**AABB**，然后用该**AABB**的中心和对角线来构造球体包围体。
- *改进*：从**AABB**的中心出发，将它作为球体包围盒的中心，然后再次遍历所有的顶点，找到距离最远的顶点，最后将最远距离最为球的新半径。
- **Ritter**的近似最优包围盒方法：先找到**x**、**y**、**z**轴线上最远和最近的顶点，然后从这三组顶点中找出距离最大的一组顶点，取这组的中心为球心，取它们之间距离的一半为半径。然后，遍历其它顶点，计算顶点到球心之间的距离**d**，如果这个顶点在球体半径**r**以外，则将球心朝着这个顶点方向移动**(d-r)/2**，同时将半径大小修正为**(d＋r)/2**。重复上述过程，直到把不在这个球体内的顶点和旧的球体全部包含在新的球体内。

包围球体的创建

- **Welzl的方法**：找到一组定义球体的支撑点，当发现一个顶点在当前球体的外面时，将这个顶点位置添加到支撑点集合中（同时删除旧的支撑点），计算新的球体。再次遍历随机分布的顶点列表，直到球体包含所有的顶点。

  该算法的复杂度对于一组随机分布的顶点来说是线性的，并能保证生成一个最优的包围球体。

OBB的创建

- **Gottschalk方法：**
  - 思想：首先从物体的凸包计算出一个方向，然后找到紧密贴合物体的OBB。
  - 凸包计算方法：QuickHull方法。计算复杂度为O(*n*log*n*)。
  - 方向计算方法：计算凸包的协方差矩阵，求其特征向量并将其归一化就是所球包围盒的方向向量。
  - OBB的中心和半边长计算方法：将凸包上的点投影到方向向量上，并求每个方向上的最大值和最小值。每个方向最大值和最小值的平均值决定了该方向的中心，最大值减去最小值的一半为该方向的半边长。

OBB的创建

- **Eberly的方法：**
  - 原理：利用最小化技术来计算体积最小的**OBB**。是一种迭代方法，其收敛速度取决于长方体的初始估计。
  - 优点：不需要凸包计算。
  - 设已经有一个估计的长方体，三条轴线分别为$\mathbf{a}^u$, $\mathbf{a}^v$, $\mathbf{a}^w$, 将所有点在这三条轴线上进行投影，可以找出$\mathbf{a}^u$方向上的最小值$k^u_{\min}$和最大值$k^u_{\max}$。同样可以找出$k^v_{\min}$和$k^v_{\max}$、$k^w_{\min}$和$k^w_{\max}$。利用这些值计算长方体的中心：

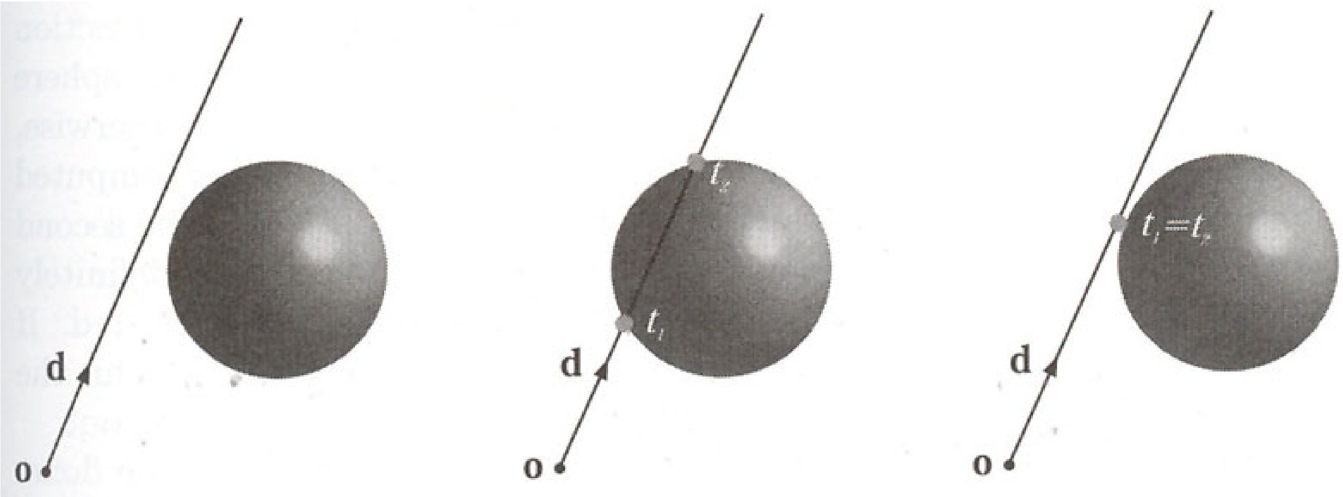
$$\mathbf{a}^c = \frac{k^u_{\min} + k^u_{\max}}{2}\mathbf{a}^u + \frac{k^v_{\min} + k^v_{\max}}{2}\mathbf{a}^v + \frac{k^w_{\min} + k^w_{\max}}{2}\mathbf{a}^w$$

    长方体的半边长可表示为: $h_l=(k^u_{\min}+k^u_{\max})/2,\ l\in\{u,v,w\}$。再迭代。
  - 初始长方体的计算：对所有可能的方向进行采样，选择长方体最小的一组轴线最为迭代计算的初始值。

求交测试的重要规则

- 这些规则可以使得相交测试更快速、稳定、精确。
  - 尽可能利用简单的比较和计算来排除或确认各种相交类型，从而避免进一步的计算。
  - 尽可能充分利用上次的测试结果。
  - 如果使用了多种排除或确认测试，试着改变它们之间的内部顺序，这有可能产生一个更有效的测试结果。
  - 尽量搁置开销量大的计算，如三角函数、平方根、除法等。
  - 降低维数。（如把三维降为二维、二维降为一维）。
  - 尽量找出在相交测试之前可预先完成的计算。
  - 当相交测试比较复杂时，尽量先利用物体的包围球进行初步排除。
  - 上机实践进行计时比较，计时要采用真实的数据和测试环境。
  - 尽量使程序具有比较好的鲁棒性。能处理各种特殊情况，对浮点精度误差不敏感。

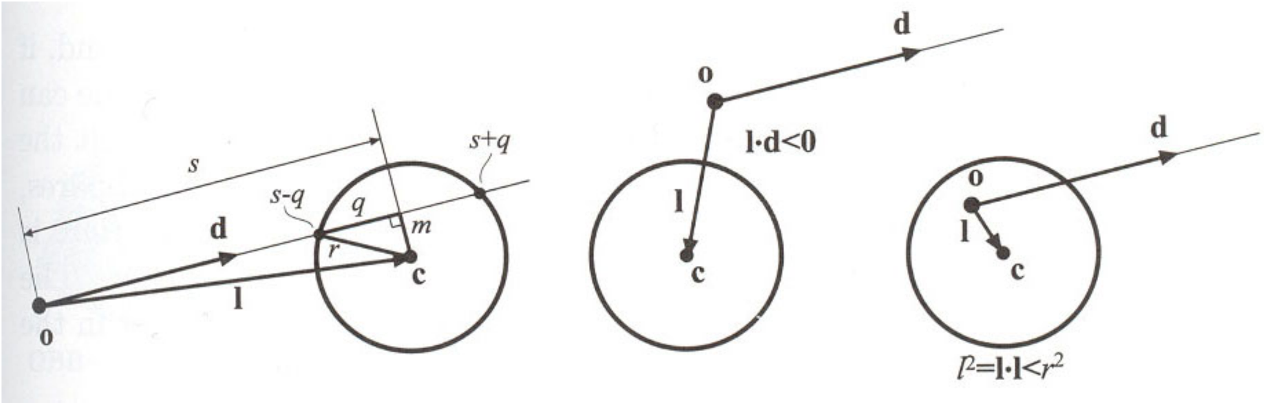
射线/球体相交测试



没有交点　　两个交点　　一个交点

- 数学方法
  - 把射线的参数方程代入球体的隐式方程，得到一个一元二次方程，利用方程的判别式进行无根判别和根的计算。

射线/球体相交测试



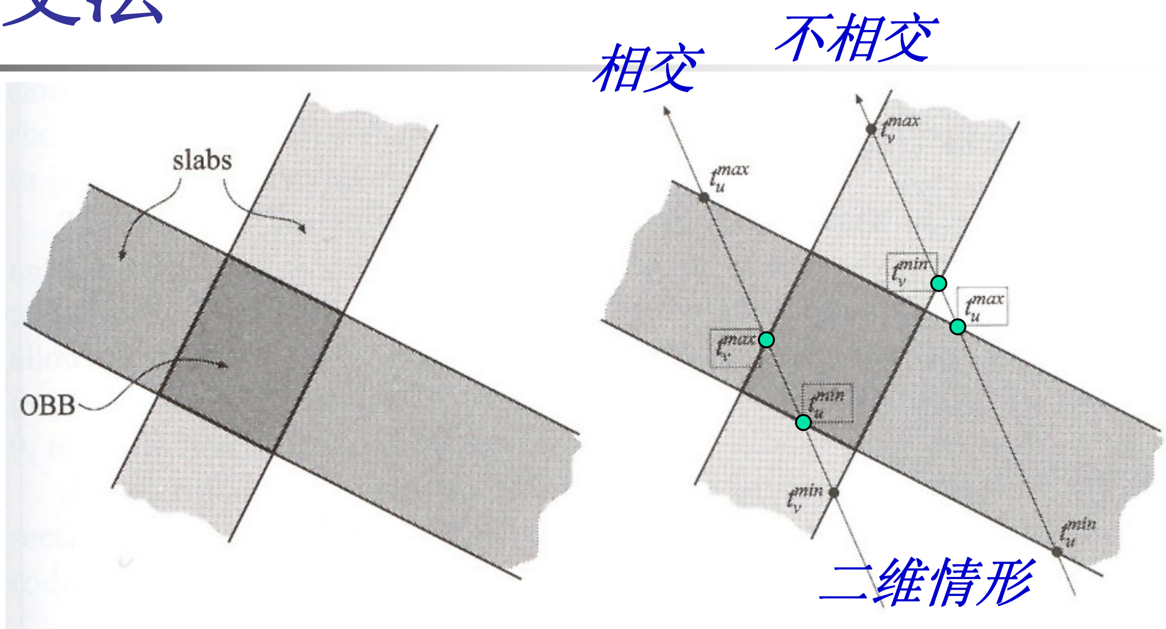
- 优化方法
  - 不需要对射线原点**o**后的相交进行计算。
    - 计算射线原点**o**到球体中心的向量**l=c-o**，其平方长度为l^2＝**l.l**。如果$l^2 < r^2$，则说明射线原点位于球体内部（见上图右），从而可以肯定这条射线肯定与球体相交。如果只需知道它们之间是否会相交，则结束，否则继续。
    - 计算**l**在射线方向的投影：s=**l.d**。如果s < **0**而且射线原点位于球体之外，则球体位于射线原点的后面，没有交点。**(**见上图中**)**
    - 利用勾股定理计算球心与投影点之间的平方距离：$m^2=l^2-s^2$。如果$m^2 > r^2$, 则射线与球体无交，否则肯定有交点。

射线/长方体相交测试

- 平板层相交法
  - 可以处理AABB、OBB、K-DOP等情况
- Woo算法
  - 只能处理AABB。但在某些体系结构下可以计算得更快一些。
- 线段/长方体重叠检测法
  - 基于分离轴定理，只能处理线段与AABB之间的相交问题。

平板层相交法

平板层(Slab)为两个相互平行的平面。



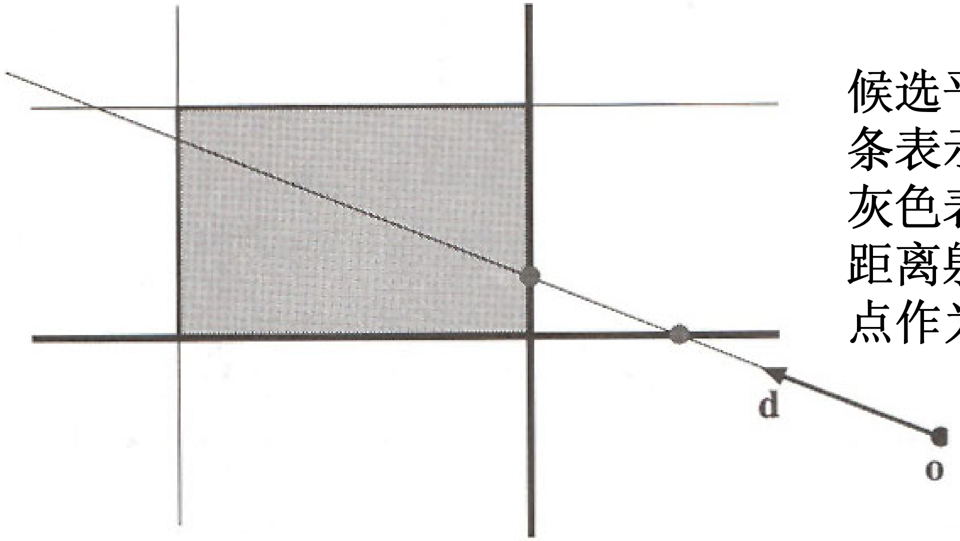
- 将长方体看成3个平板层。计算射线与平板层的交点。对于每个平板层，存在一个最小的t值和最大的t值，分别记做t_i^{min}和t_i^{max}，$i\in\{u,v,w\}$，计算

$$t_{\min}=\max(t_u^{min}, t_v^{min}, t_w^{min})$$

$$t_{\max}=\min(t_u^{max}, t_v^{max}, t_w^{max})$$

然后就可以进行巧妙测试：如果$t_i^{min}\leqslant t_i^{max}$，射线与长方体相交，否则不相交。

Woo算法



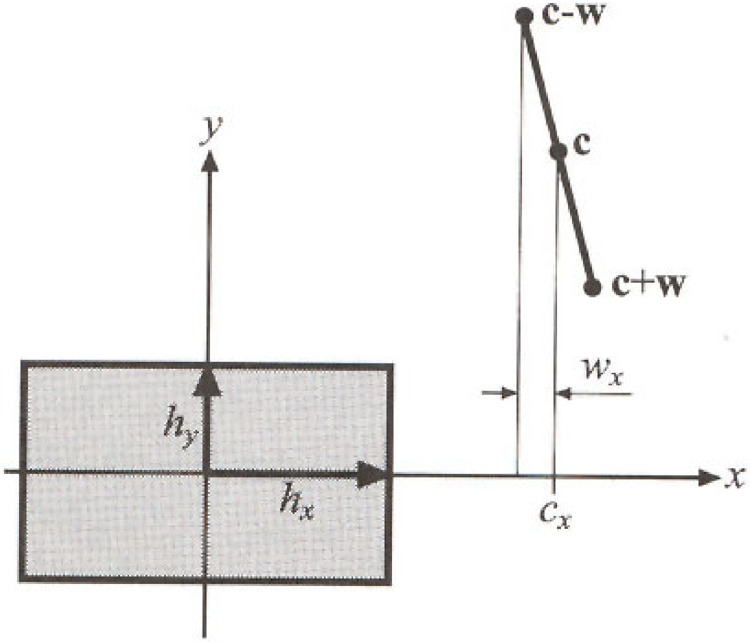
候选平面的方向朝前，用粗线条表示，与射线之间的交点用灰色表示。由于左边的相交点距离射线原点最远，因此将该点作为潜在相交点。

- 针对射线与AABB之间的相交检测，**Woo**提出了一些巧妙的优化方法。
- 基本思想：在组成**AABB**的**6**个平面中选出**3**个候选平面。对于每一对互相平行的平面来说，忽略其背向平面而不作进一步考虑。计算射线与候选平面的相交距离(**t**值)，其中的最大值就可能对应一个相交情形。如果找到潜在的相交情形(*相交点位于AABB的相应面上*)，再计算出实际的相交点。

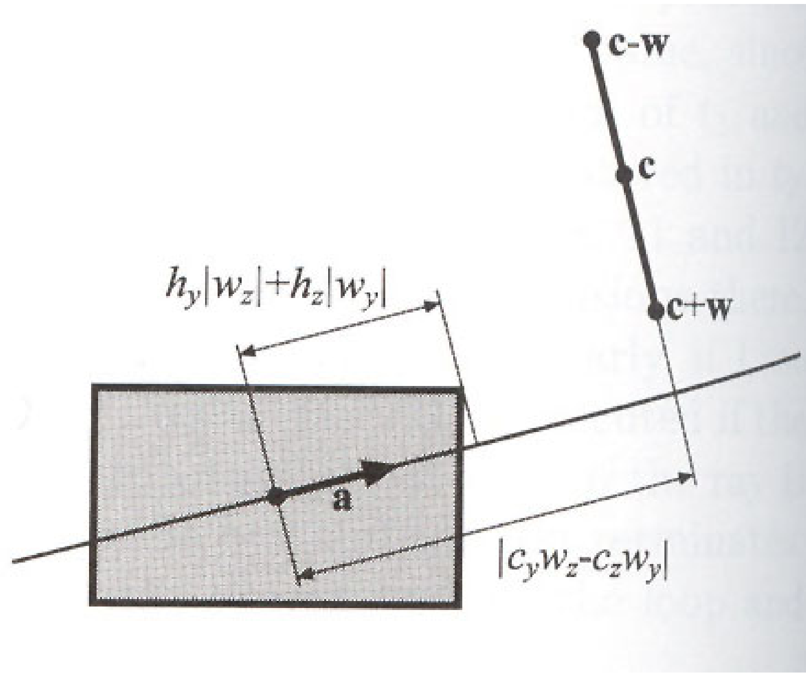
线段/长方体重叠检测

- 利用分离轴定理判断一条线段是否与一个**AABB**彼此重叠。
- 线段与射线不同，其长度是有限的，设线段由一个中点**c**和一个半向量**w**定义。对线段和**AABB**进行平移以确保**AABB**的原点位于**(0,0,0)**。长方体的大小为$(2h_x, 2h_y, 2h_z)$
- 一个长方体有**3**个轴需要进行测试，分别为**(1,0,0), (0,1,0), (0,0,1)**。**(3**个轴**)**
- 线段方向与长方体的轴线进行叉积可以形成**3**个需要测试的轴线。**(3**个轴**)**
- 例如, 对**(1,0,0)** 进行相交测试**:** $|c_x|>|w_x|+h_x$**,** 若为真，则不相交。对叉乘轴**a=w×(1,0,0)=**$(0, w_z, -w_y)$的测试如下：若 $|c_y w_z - c_z w_y| > h_y |w_z| + h_z |w_y|$为真，则不相交。

线段/长方体重叠检测



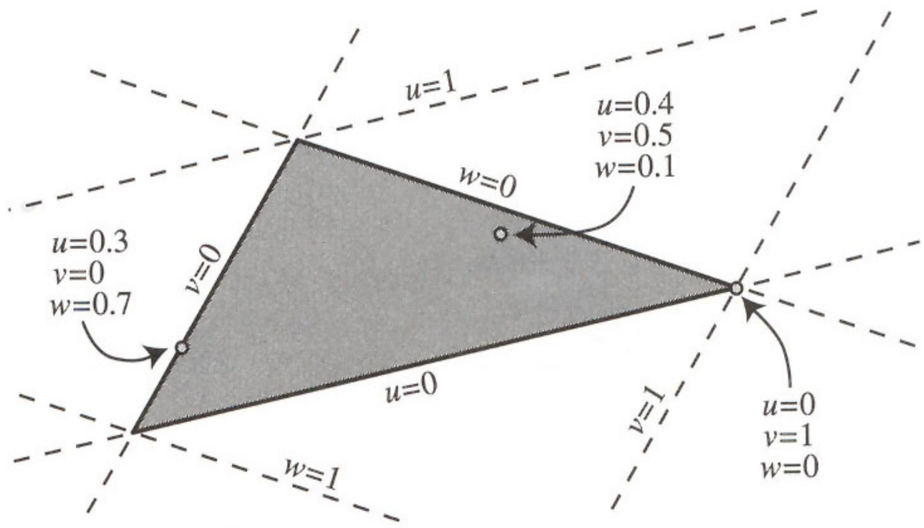
用x轴进行线段与长方体之间的相交测试。



用a轴进行相交测试。

射线/三角形相交测试

- 原理：首先计算出射线与三角形所在平面的交点，然后将交点和三角形顶点投影到一个轴对齐平面（**xy, yz**或**xz**）上，并使得投影三角形面积最大化。
- 将三维问题变为二维问题，从而只需判断二维点是否在二维三角形内部。
- 三角形上的点$\mathbf{t}(u,v)$可以由下面的公式给出：
  $\mathbf{t}(u,v) = (1-u-v)\mathbf{v}_0 + u\mathbf{v}_1 + v\mathbf{v}_2$，其中$(u,v,w)$为重心坐标



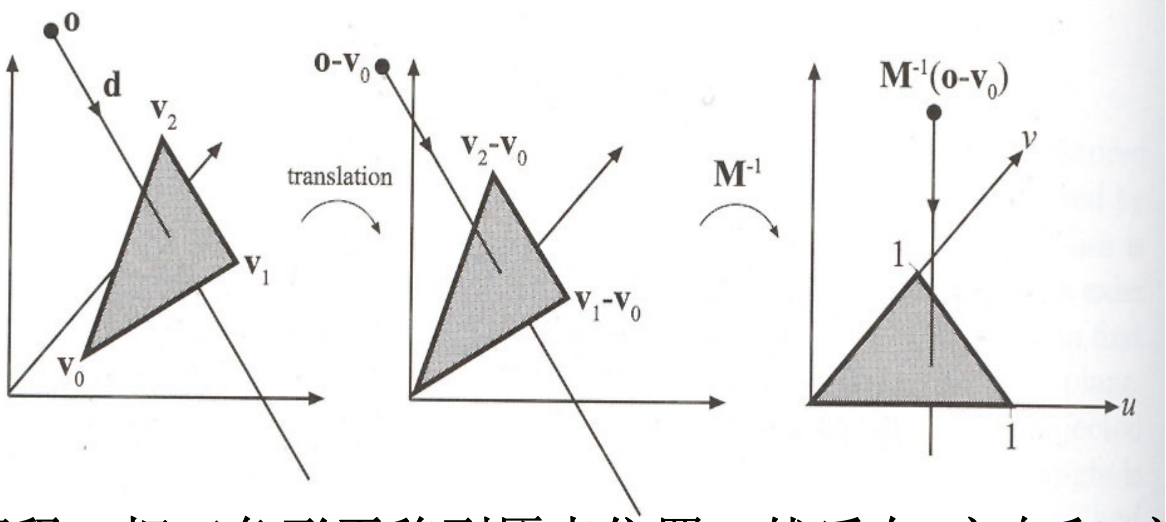
三角形的重心坐标

射线/三角形相交测试

- 射线$\mathbf{r}(t)$与三角形$\mathbf{t}(u,v)$之间的相交计算等同于对方程$\mathbf{r}(t) = \mathbf{t}(u,v)$的求解，得到线性方程组:

$$\begin{pmatrix} -\mathbf{d} & \mathbf{v}_1 - \mathbf{v}_0 & \mathbf{v}_2 - \mathbf{v}_0 \end{pmatrix} \begin{pmatrix} t \\ u \\ v \end{pmatrix} = \mathbf{o} - \mathbf{v}_0$$

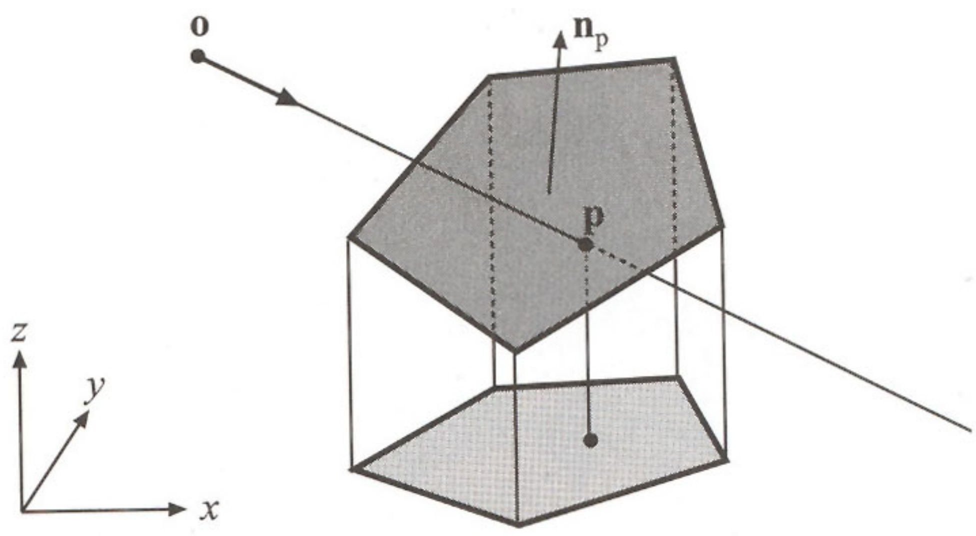
从而得到交点的重心坐标(u, v)和 t 值。



几何解释：把三角形平移到原点位置，然后在y方向和z方向将这个三角形变换为单位三角形，同时将射线方向与x轴方向对齐。

射线/多边形相交测试

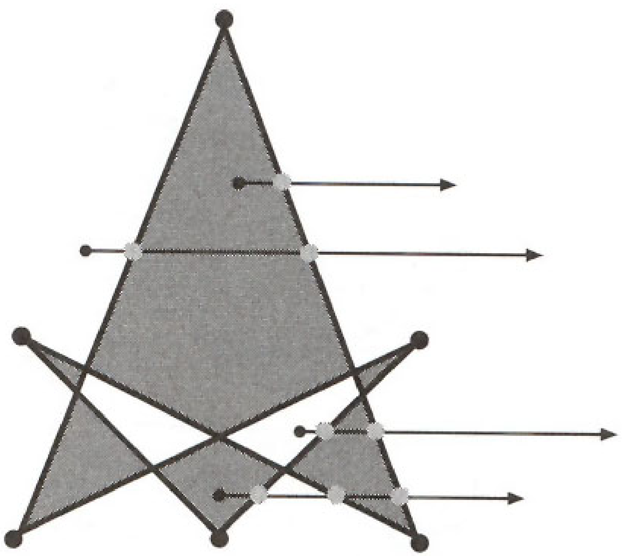
- 首先计算射线与多边形所在平面方程的交点
- 将交点和多边形的所有顶点投影到xy, yz或xz平面上（使得投影面积最大：利用平面的法向来判断，若x分量最大，则舍弃x分量，投影到yz平面），从而将三维降为二维。
- 在二维空间，判断交点是否包含在二维的多边形中。

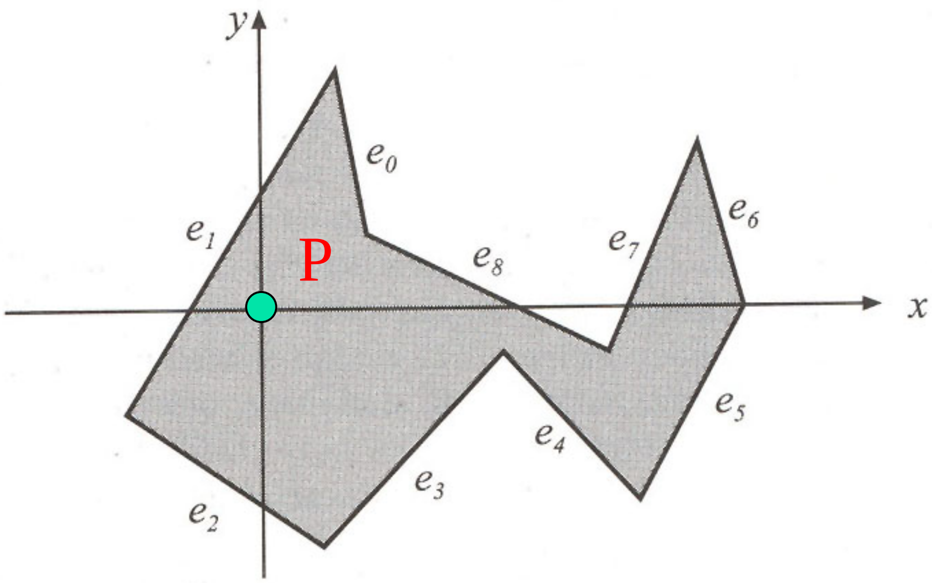


多边形顶点和交点p在xy平面上的正交投影（在xy平面上的投影面积最大）

点是否在二维多边形内部的 Crossing Test

- 基于Jordan 定理
  - 一条射线从某点出发，若与多边形的交点个数为奇数个，则点在多边形内部，否则在多边形的外部。





自交的凹多边形：*并非所有被多边形包围的部分就一定是多边形的内部。*

加速方法：将多边形平移-**P**, 这样通过计算与**x**轴正向的交叉点数量就可以确定**P**是否位于多边形内部。

平面/长方体相交测试

- 要确定长方体与平面Π: **n.x**+d=0之间是否相交，一种方法是将长方体的所有顶点代入该平面方程。如果所有的结果有正有负（或者为零），则表明这些顶点位于平面的两侧，因此存在相交关系。

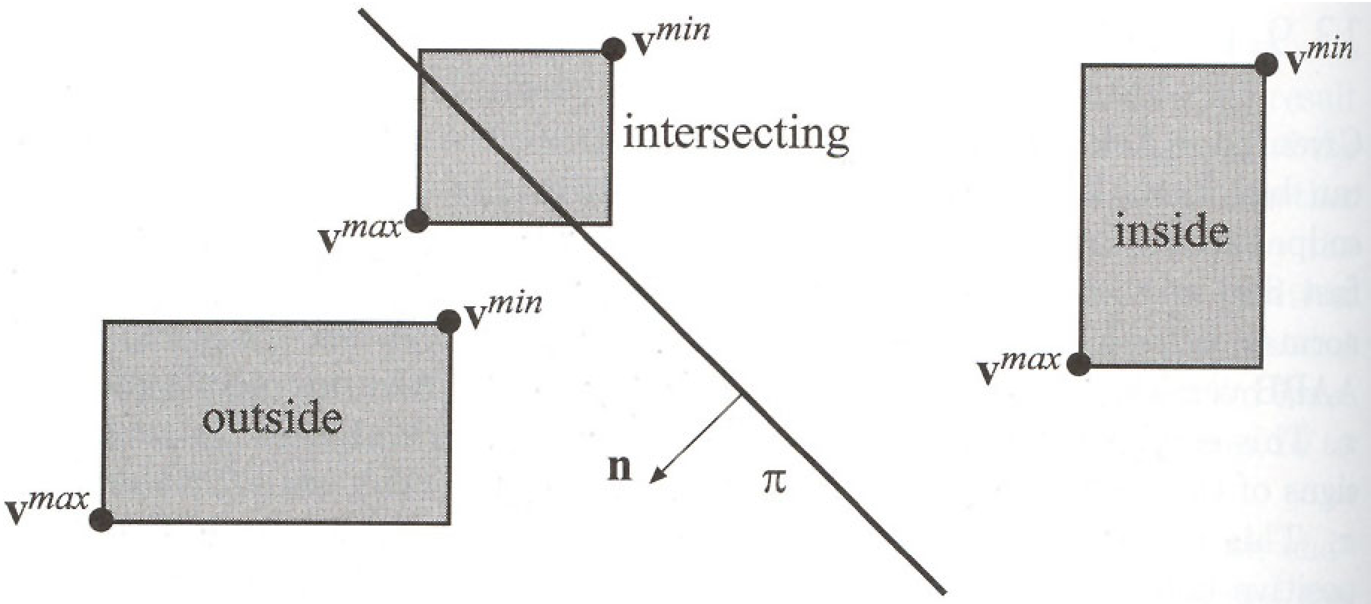
- 针对**AABB**和**OBB**有更快的方法。

- 基本思想：只需要将两个顶点代入平面方程，这两个顶点构成了这个长方体的对角线。所取的对角线通过长方体的中心，而且*与平面的法向**n**对齐程度最好。*

平面/ AABB相交测试

- 给定一个由b^{min}和b^{max}定义的AABB，记做*B*，基于它可以生成4条不同的对角线，这些对角线都通过*B*的中心。
  - 首先找出哪条对角线与这个平面的法向最接近。
  - 然后将这条对角线的两个顶点代入平面方程中，从而可以测试出每个顶点位于平面的哪一侧。
  - 如果求解的结果符号相反，或者至少一个为0，则**AABB**与平面相交。
  - *注意*: 如果**AABB**位于平面的正半空间或负半空间，则无交。如果v^{min}位于平面的正半空间，则v^{max}一定位于正半空间。

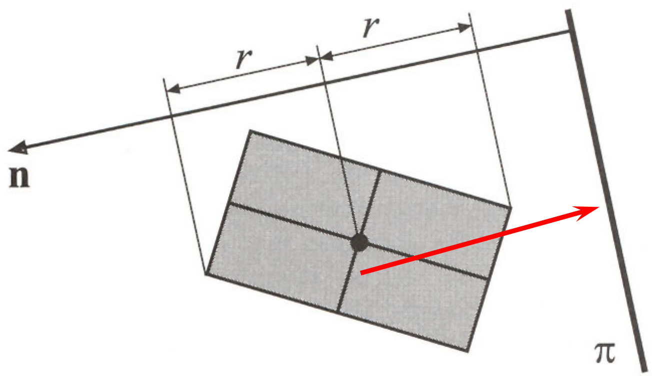
平面/ AABB相交测试



3个**AABB**对于给定平面的**v**^{min}和**v**^{max}，如果这对顶点位于平面的同一侧，则**AABB**与平面不相交，否则相交。注意：可先检测**v**^{min}，如果发现它与平面法向位于同一侧，则说明**AABB**“在外面”（即位于该平面的正半空间）。

平面/ OBB相交测试

- 问题的关键：如何确定v^{min}和v^{max}，然后应用与AABB相同的方式进行相交测试。
- 方法1：为了找到这两个点，可以对平面法向进行变换，使得它位于OBB的坐标系统中，从而转化为AABB情形。
- 方法2：将OBB投影到平面法线上来实现。



将OBB投影到平面的法线方向上，把OBB在法线上的半长记为r，如果从OBB中心到平面的距离大于r，则无交。

“三角形/三角形”相交测试

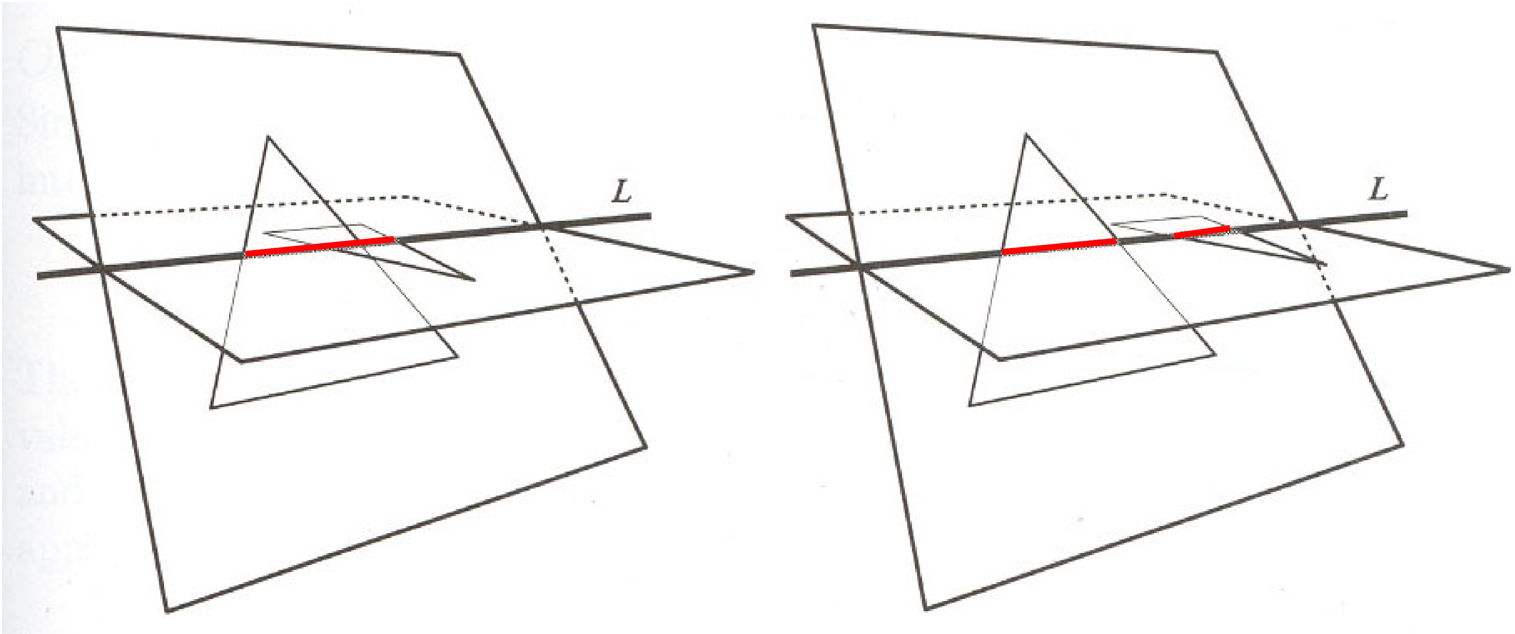
- 三角形是图形硬件最重要的绘制单元。在碰撞检测算法的最底层，一般会有一个程序来判断两个三角形是否相交。
- 通常仅关心它们之间是否会发生碰撞，而不需要具体的交点信息。
- 给定两个三角形$\boldsymbol{T_1}=\triangle \mathbf{u_0u_1u_2}$和$\boldsymbol{T_2}=\triangle \mathbf{v_0v_1v_2}$，它们分别位于平面$\Pi_1$和$\Pi_2$上，需要判断它们之间是否相交
  - 区间重叠法
  - **ERIT**方法

区间重叠法(Interval Overlap Method)

- 早期排除
  - 计算平面方程Π_2，并把三角形T_1的顶点代入该平面方程，得到三角形T_1的顶点到平面Π_2的符号距离
  - 若所有的距离不等于0，且这些值具有相同的符号(说明三角形T_1位于平面Π_2的一侧)，则不相交。
  - 对三角形T_2和平面方程Π_1，进行同样的测试。

这两种早期排除测试可以节省很多计算量！！！

- 判断交线上是否含有两个三角形的公共部分
  - 如果所有的距离都等于零，则这两个三角形共面，需单独处理。（投影到二维空间进行处理）
  - 否则，平面Π_1和Π_2之间的相交部分应该是一条直线，可表示为$\mathbf{l}=\mathbf{o}+t\mathbf{d}$。
  - 可能发生的情形分两种，见下图



两个三角形及其各自所在的平面。

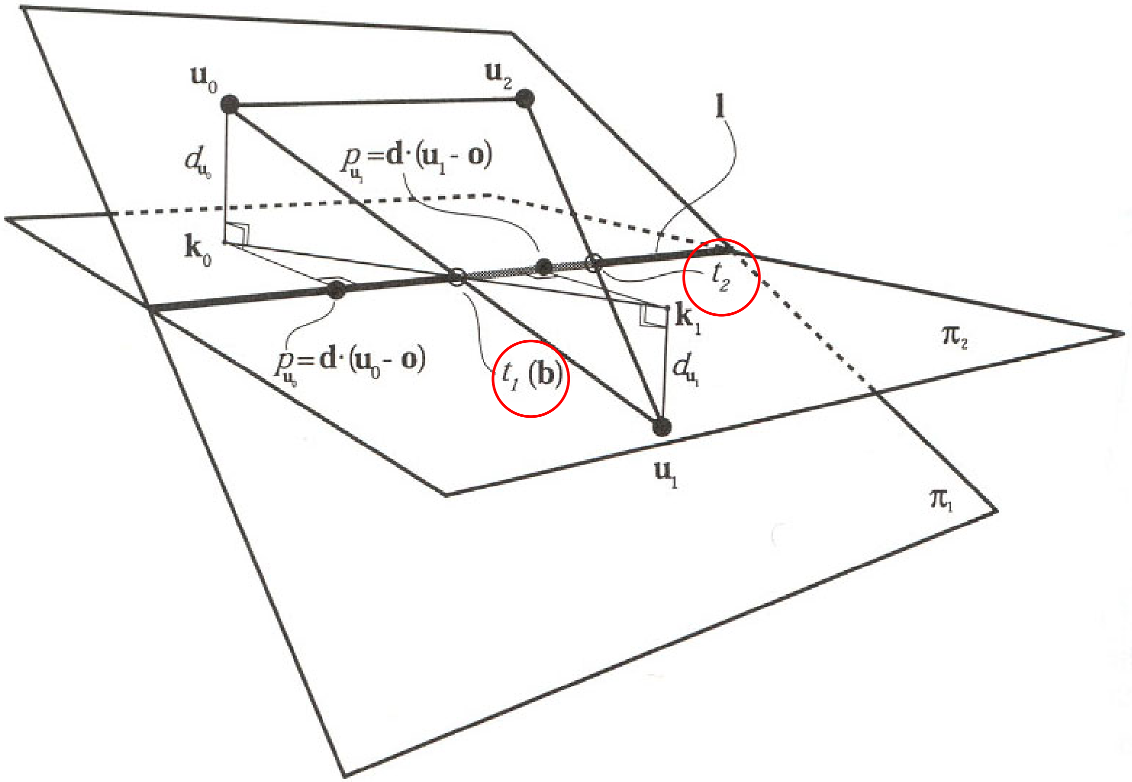
左图：与两个重叠三角形一样，沿直线L上的区间也相互重叠

右图：三角形之间不相交，直线上的区间也没有重叠

相交区间计算

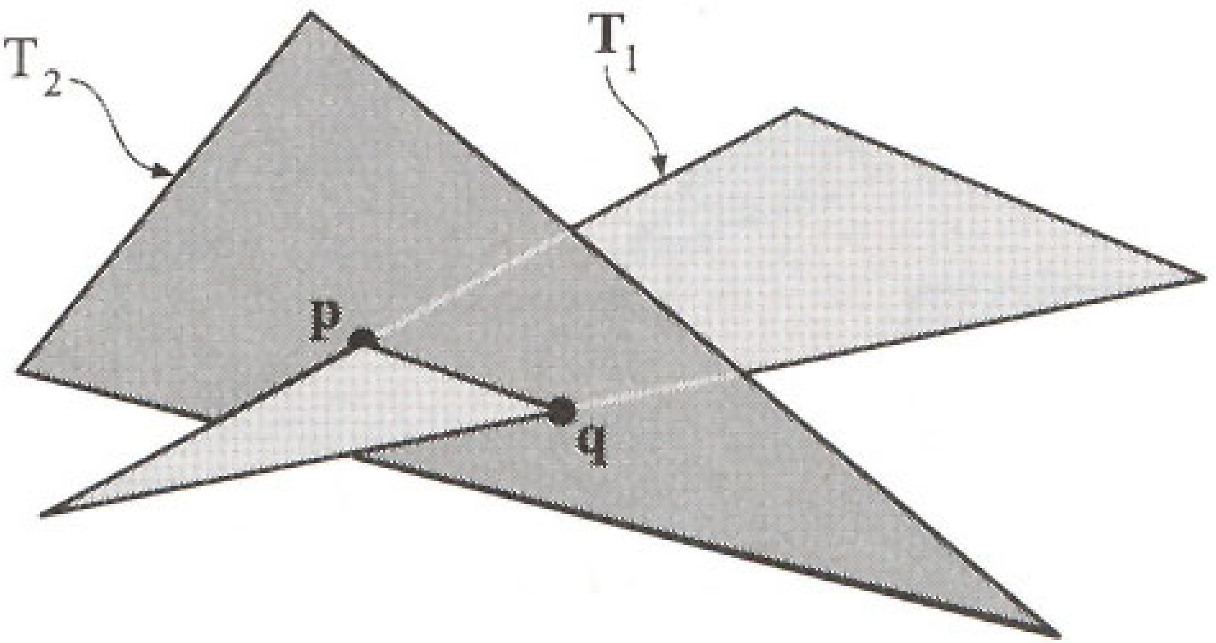
- 假设需要计算$\mathbf{T}_1$和$\mathbf{l}$之间的相交区间，假设三角形的两个顶点$\mathbf{u}_0$和$\mathbf{u}_2$位于平面Π_2的同侧，而另外一个点$\mathbf{u}_1$位于平面的另一侧。
- 利用几何关系可以算出直线$\mathbf{l}$与边$\mathbf{u}_0\mathbf{u}_1$之间的交点t_1，直线$\mathbf{l}$与边$\mathbf{u}_1\mathbf{u}_2$之间的交点t_2。$[t_1, t_2]$为三角形$\mathbf{T}_1$和$\mathbf{l}$之间的相交区间。
- 同理可以算出三角形$\mathbf{T}_2$和$\mathbf{l}$之间的相交区间。
- 如果区间相互重叠，则这两个三角形一定相交。否则不相交。
- 还可以在利用优化处理来减少计算量

两个三角形之间的相交示意图



$\mathbf{u}_i$是三角形$\mathbf{T}_1$的顶点， Π_1和Π_2是三角形$\mathbf{T}_1$和$\mathbf{T}_2$所在的平面，$d_{\mathbf{u}i}$是从$\mathbf{u}_i$到平面Π_2的有向距离，$\mathbf{k}_i$是$\mathbf{u}_i$在平面Π_2上的投影点，$p_{\mathbf{u}i}$是$\mathbf{u}_i$在两个平面交线$\mathbf{l}$上的投影点。

ERIT方法



首先计算三角形T_1和T_2的两个交点**p**和**q**，如果**p**与**q**之间的线段完全包含在**T2**中，或者与**T2**的某条边相交，则这两个三角形相交，否则不相交。

算法描述

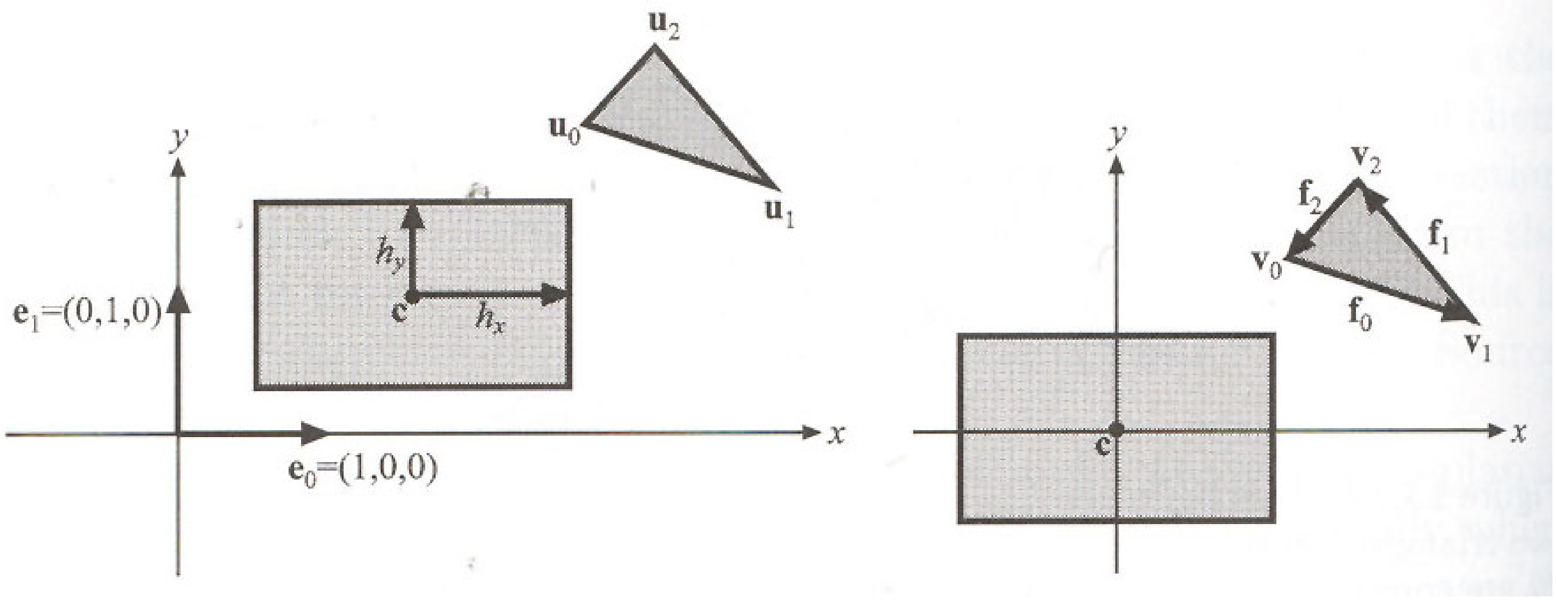
1. 计算出三角形T_2所在平面的方程
2. 如果三角形T_1上的所有点均在三角形T_2的同一侧，则不相交。
3. 如果两个三角形共面，则使用区间重叠法进行处理。
4. 计算出三角形T_1和平面Π_2之间的交线，这条线位于平面Π_2上。
5. 如果这条线段与三角形T_2相交，或者完全包含在T_2中，则T_1和T_2相交，否则不相交。

“三角形/长方体”重叠测试

- 可用于
  - 体素空间的创建
  - 三角形与长方体的碰撞检测
  - 多边形与规范化视域体(**Canonical View Volumes**）的相交测试

- 我们将介绍**Akenine-Moller**提出的三角形/**AABB**的重叠测试方法（对**OBB**需要先变换）
- 原理：基于分离轴定理

“三角形/长方体”重叠测试中的符号表示



左图：为长方体和三角形在坐标系中原始位置

右图：对长方体和三角形进行了平移，使得长方体的中心与坐标系的原点重合

三角形/AABB的重叠测试

- 基于分离轴定理，对下面13条轴线进行测试
  - [3个测试] AABB的三个轴e_0=(1,0,0), e_1=(0,1,0), e_2=(0,0,1), 分别是AABB的3条法线
  - [1个测试] 三角线的法线n。只对与三角形法向最接近的*长方体对角线的两个顶点*进行测试
  - [9个测试] AABB的三个轴与三角形三条边的叉积

  - 在测试中，一旦找到分离轴，算法即可中止，返回没有相交信息。如果通过所有测试，则三角形与长方体重叠。

BV/BV相交测试

- 完全包含物体的封闭体称为包围体(**BV**)。**BV**的主要目的是为了提供更加简便的相交测试和更加有效的预先排除。

- 例如，如果要测试两辆汽车是否相撞，可先计算各自的**BV**，然后测试两个**BV**是否发生重叠。如果没有重叠，则可以确保这两辆汽车不会相撞(这是最常见的情况)，而不必对汽车上的每个图元进行测试，从而节省计算量。

- 最常见的包围体：球体、**AABB**、***k*-DOP**、**OBB**

球体/球体相交测试

- 计算出两个球体球心距离
- 如果这个距离大于两个球体的半径之和，则不相交；否则，相交。

球体/长方体相交测试：

- 主要思想：在AABB上找到距离球心最近的一个点
- 对AABB的每条轴进行一次一维测试，判断球心坐标的在该轴线的分量是否在AABB的边界之内。
- 若在边界之内，则相交。如果在AABB的边界之外，则计算球心与这个长方体的距离（这是一个减法计算），然后再计算出这个距离的平方。
- 在三条轴线全部计算完之后，将这些距离的平方和相加与球体半径的平方进行比较。
- 如果平方距离和小于球体半径平方，则最近点位于球体内部（相交）；否则不相交。

- 将球心坐标变换到OBB的轴上去，即可处理OBB

AABB/AABB相交测试

- 因**AABB**与轴线对齐，所以只需要两个点就可以描述这个体。由于**AABB**非常简单，因此通常将它应用于碰撞检测算法中。
- 思想：每个轴上区间是否相交

Bool AABB_intersect(A,B)

returns({OVERLAP, DISJOINT}) ;

for each i $\in$ **{x,y,z}**

if($a_i^{\min} > b_i^{\max}$ or $b_i^{\min} > a_i^{\max}$ **)**

return(DISJOINT) ;

return(OVERLAP) ;

k-DOP/*k*-DOP相交测试

- 一个***k*-DOP**是一个凸多面体，各个面由***k***个固定的法向确定，其中法向以外的半空间不作为包围体一部分来考虑。
- **AABB**实际上是一个特殊的**6-DOP**。当***k***值增大时，包围体越来越接近凸包，从而更接近物体。
- **Klosowski**等指出，对于适当的***k***值，两个***k*-DOP**相交测试速度要比两个**OBB**的相交测试速度快一个数量级。
- 假设两个***k*-DOP**分别为**A**和**B**，需要对它们进行相交测试，这时需对每一对平板层（S_i^A, S_i^B）进行重叠测试，而$S_i=S_i^A\cap S_i^B$是一个非常容易解决的一维区间重叠测试。
- 如果在任何情况下S_i为空集，则这两对平板层相互分离，测试过程结束。如果所有S_i均不为空集，则可以确定这两个***k*-DOP**相交。

k-DOP/k-DOP相交测试算法描述

- KDOP_Intersect($d_1^{A,min}$, ...,$d_{k/2}^{A,min}$,$d_1^{A,max}$, ... , $d_{k/2}^{A,max}$,

  $d_1^{B,min}$, ..., $d_{k/2}^{B,min}$,$d_1^{B,max}$, ... , $d_{k/2}^{B,max}$)

  returns ({OVERLAP, DISJOINT});

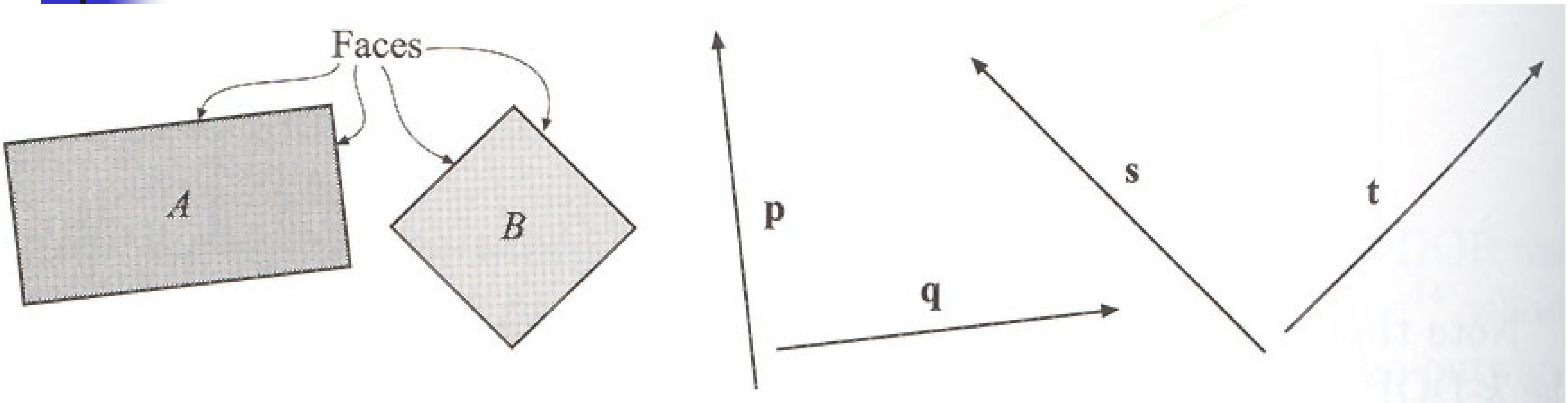
  for each i$\in${1,2, ... ,k/2}

  if ($d_i^{B,min}>d_i^{A,max}$ or $d_i^{A,min}>d_i^{B,max}$)

  return (DISJOINT);

  return (OVERLAP) ;

“OBB/OBB”相交测试

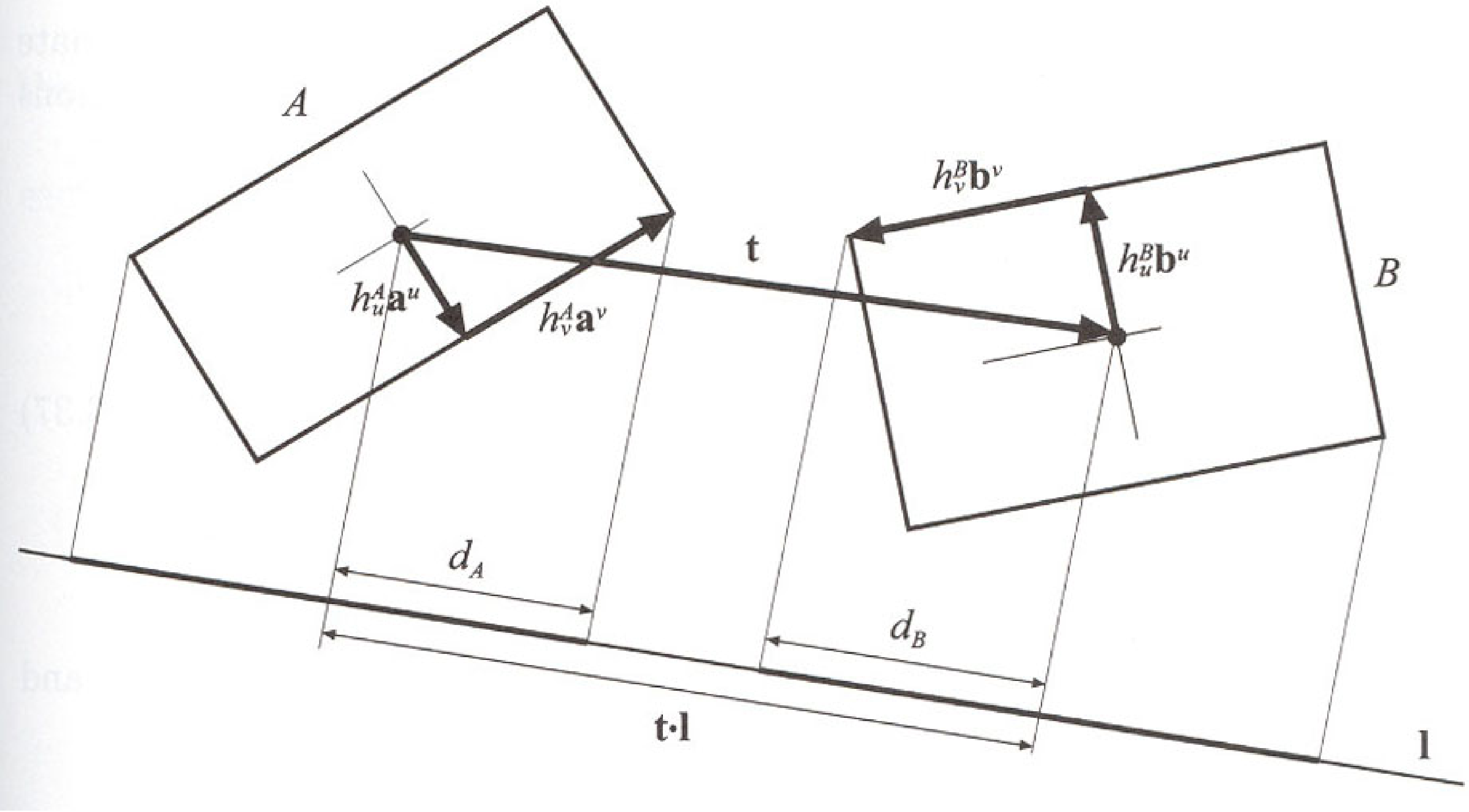


二维形式

为了判断两个**OBB**是否发生重叠，可使用分离轴定理。因为分离轴应与**A**和**B**的表面正交，可选**p, q, s, t**为分离轴，然后将**OBB**投影到这些轴上。如果两个**OBB**在所有这些轴上的投影相互重叠，则两者相交，否则不相交。

因此，为了知道两个**OBB**是否重叠，只需要找到一条可以使**A**和**B**的投影不重叠的轴线即可。在本例中，**q轴**是唯一能将这两个投影分开的轴。

“OBB/OBB”相交测试



分离轴定理示意。由于**A**和**B**在轴线**l**上的投影的“半长”没有重叠，因此两个**OBB**不相交。

“OBB/OBB”相交测试

- 对于两个**OBB(A**和**B)**, 根据分离轴原理，需要对15条轴线进行测试：**3**条取自**A**的表面，**3**条取自**B**的表面，另外**9**条取自**A**和**B**边的组合。

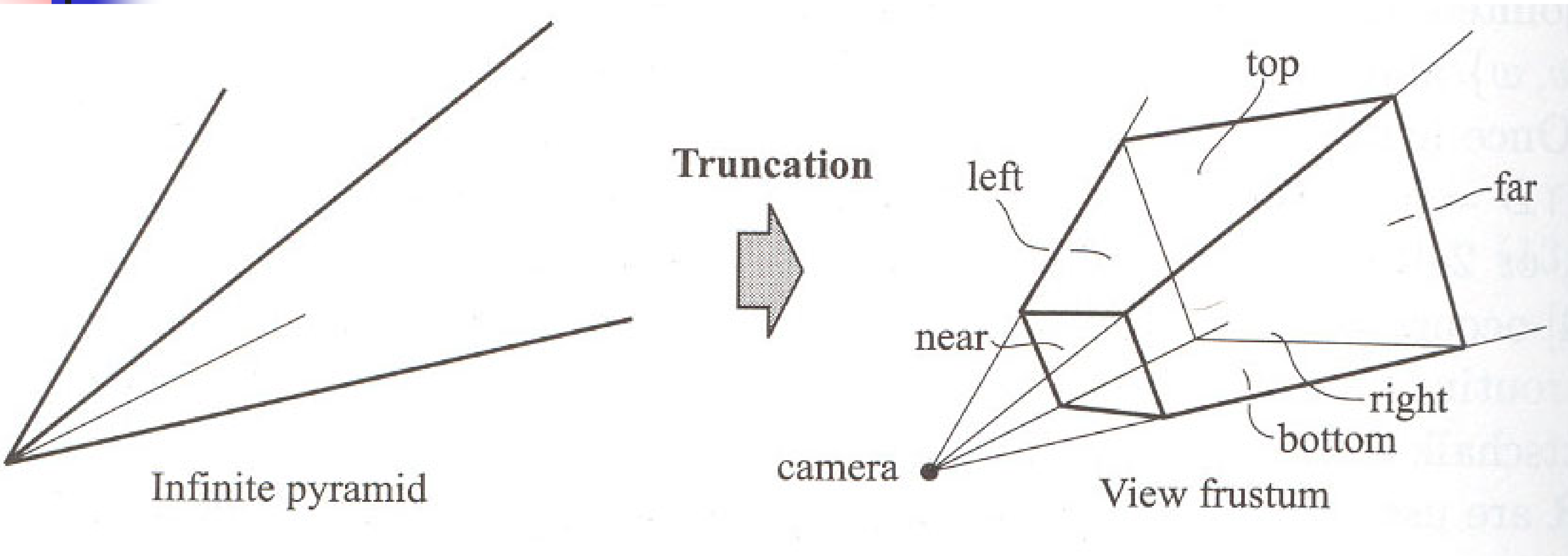
- 定理：当且仅当**l**是一条分离轴时，在这条轴线上的投影才彼此分离，不重叠：

$$|\mathbf{t.l}| > d_A + d_B$$

- 如果在这**15**个测试中有一个成立，则这两个**OBB**不相交。

- 最好在开始真正的**OBB/OBB**测试之前，用包围球进行快速排除。

视域四棱锥的相交测试



无限金字塔　　　　　　　　　　视域四棱锥

- 三种可能: **(1).** 在外面 **(2).** 在里面 **(3).** 相交

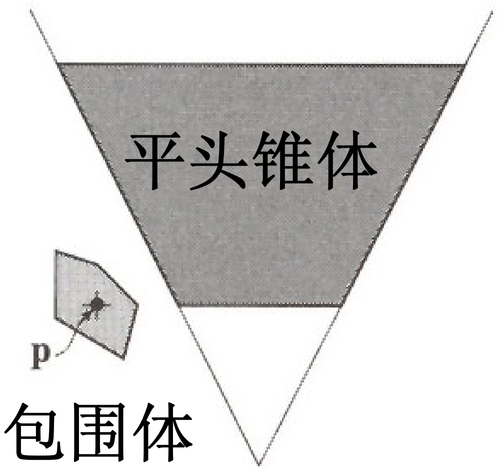
- 如果确定一个BV完全在一个视域四棱锥的外面，则无需对这个BV所包含的子树进行遍历，也无需绘制所包含的几何体。

- 如果BV完全在视域四棱锥的里面，则不需要对它的子树进行任何视域四棱锥/BV相交测试，必须绘制其中所包含的所有元素。

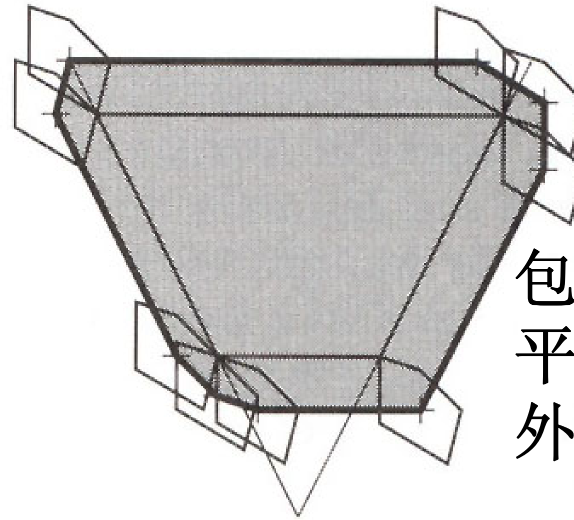
- 对于部分可见的BV来说，需要对BV子树与视域四棱锥进行进一步的递归相交测试。

先介绍一般物体与平头锥体的相交测试方法

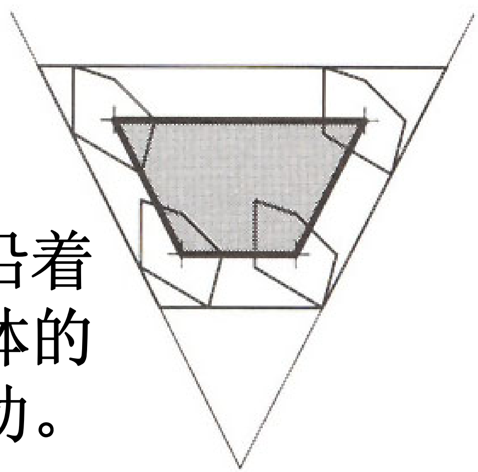
- 基本思想：将**BV**/平头锥体的相交测试转化为点/体的相交测试。
- 首先选取一个与**BV**相关的点，然后沿着平头锥体的外部移动**BV**，在不发生重叠的情况下尽可能靠近平头锥体。在移动过程中，记下这个**BV**相关点的轨迹，从而形成一个新的体。
- 类似，在平头锥体的内部移动**BV**，尽可能靠近平头锥体，可以得到比较小的平头锥体，其各个面与初始锥体的各个面相互平行。
- 如果**P**点位于深灰色体外，则**BV**就在平头锥体的外部；如果**P**位于深灰色区域内，则**BV**与平头锥体相交；如果**P**位于浅灰色区域内，则**BV**位于平头锥体内部。



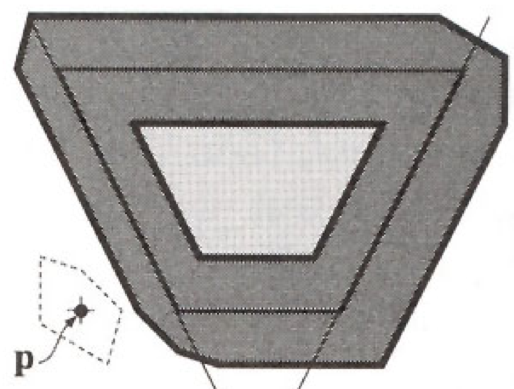
包围体



包围体沿着平头锥体的外部移动。

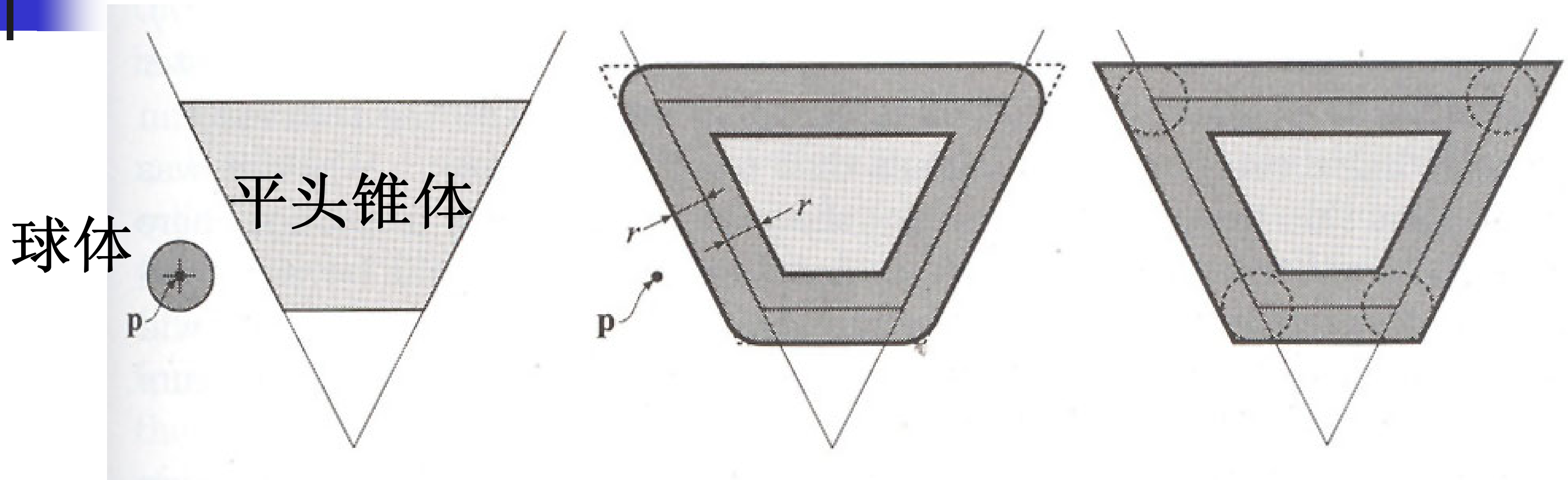


包围体沿着平头锥体的内部移动。



可以将平头锥体/**BV**的相交测试变换为点**P**在内外体之间的测试。

平头锥体/球体相交测试



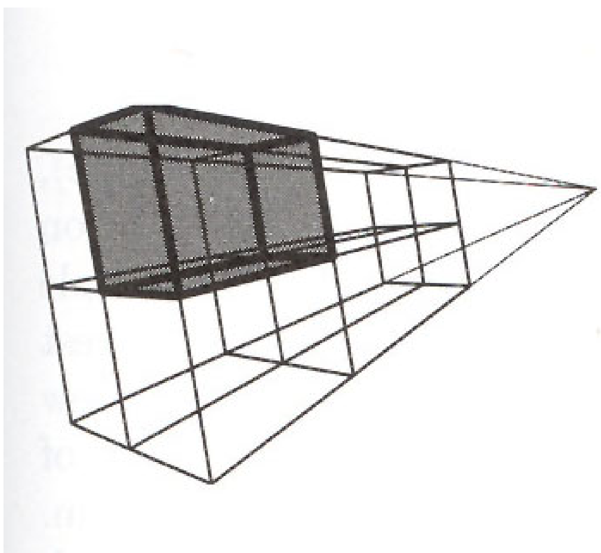
将平头锥体/球体之间的相交测试转换为检测点**P**位于中图所示的深灰色体还是浅灰色体中。

右图是对中图所示体的一种合理近似，如果球体中心位于圆角外边，而且同时位于所有外平面内部，那么即使这个球体位于平头锥体的外面，也会错误地认为相交。但可以提高测试效率！

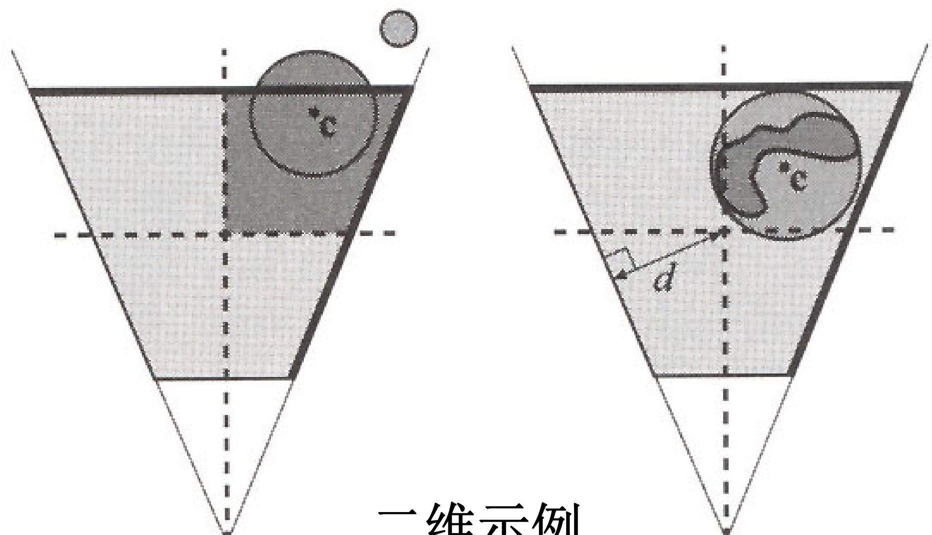
平头锥体/球体相交测试具体实现

- 对于平头锥体的6个面，计算从球心到该平面的有向距离(只需将球心坐标代入平面方程)。
- 如果这个距离大于球体的半径r，则球体位于平头锥体的外部;
- 如果到所有6个平面的距离都小于－r，则球体位于平头锥体的内部。
- 否则，球体与平头锥体相交。

平头锥体/球体相交测试的加速



粗线为仅需考虑的平面



二维示例

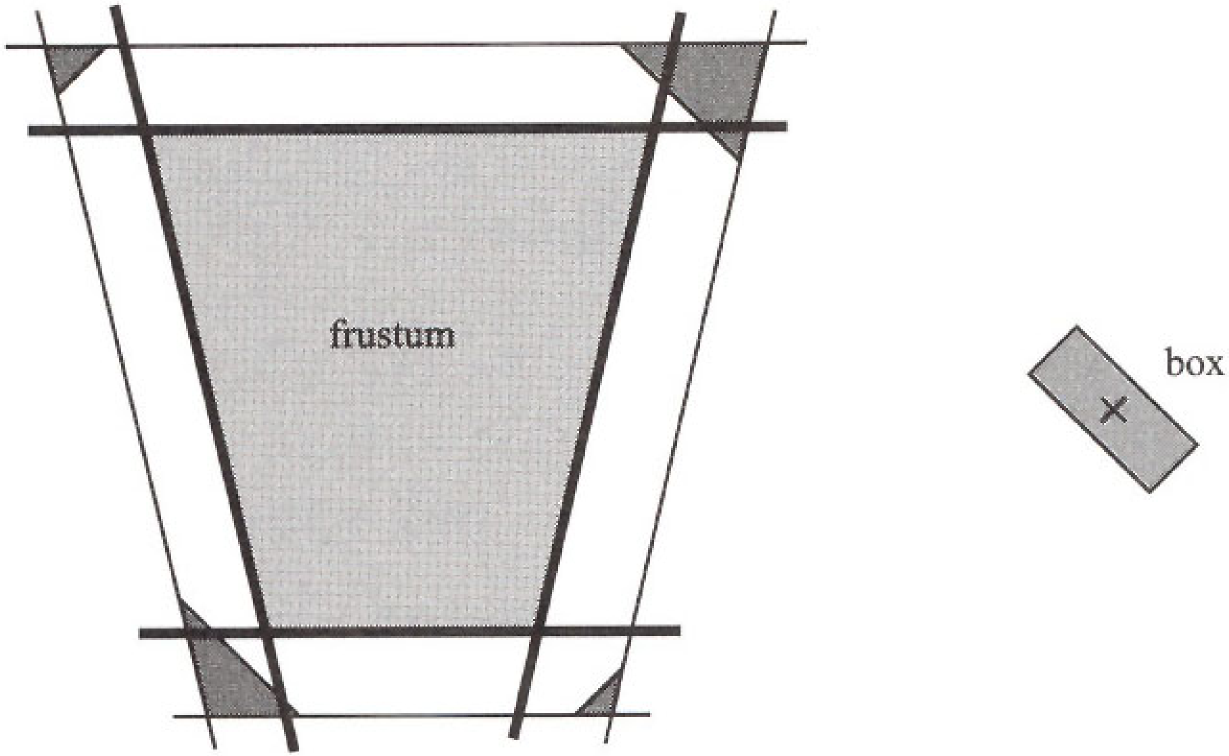
把三维平头锥体分割成**8**个八分体，仅需对球心所在的八分体的**3**个外平面进行测试。

限制：从平头锥体中心到平头锥体各个平面的最小距离d一定要大于这个物体最小包围球的半径。

平头锥体/长方体相交测试

- 原理：分别对平头锥体的**6**个平面与**OBB**或者**AABB**包围盒进行测试。
- 如果包围盒的所有顶点均位于一个平面的外部，就可以确定这个包围盒一定位于平头锥体的外部。
- 具体实现：只需要测试包围盒中的两个顶点，就可以确定包围盒位于平面的哪个半平面，或者是否与该平面相交。

平头锥体/长方体相交测试



当长方体中心位于图中的深灰色区域时，有可能错误地认为这个长方体与平头锥体相交，而事实上它位于平头锥体的外部。

平头锥体平面的获取

- 为了进行视域四棱锥裁剪，需要知道平头锥体的6个不同的平面方程。
- 假设视域变换为**V**，投影矩阵为**P**，则它们的合成变换为**M**＝**PV**，从而可以将点**s**（其中$s_w=1$）变换为**t**＝**Ms**。由于经过透视投影，**t**可能无法满足$t_w\neq 1$，因此需要将**t**所有分量除以t_w，从而得到一个点**u**，同时满足$u_w=1$。对于视域四棱锥内部的所有点，有

$$-1\leq u_i \leq 1, \quad i\in x ,y, z$$

也就是说，点**u**位于一个单位立方体内部。

- 从上述式子可以得到平头锥体的平面！

公式推导

- 单位立方体左平面的右侧，有$-1 \leqslant u_x$，展开得

$$-1 \le u_x \Leftrightarrow -1 \le \frac{t_x}{t_w} \Leftrightarrow t_x + t_w \ge 0$$

$$\Leftrightarrow (\mathbf{m}_0 \bullet \mathbf{s}) + (\mathbf{m}_3 \bullet \mathbf{s}) \ge 0 \Leftrightarrow (\mathbf{m}_0 + \mathbf{m}_3) \bullet \mathbf{s} \ge 0$$

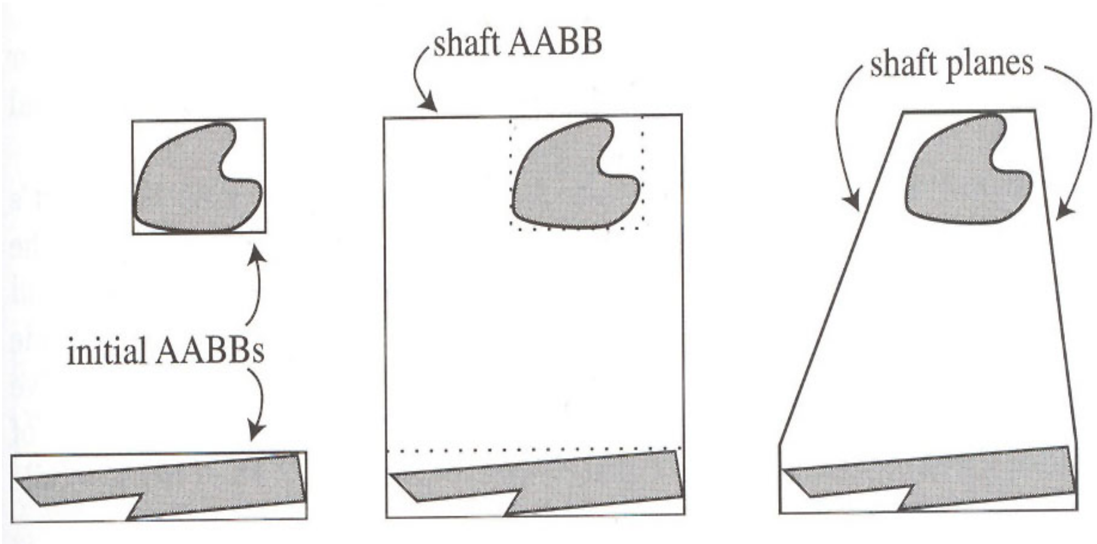
- 其中$\mathbf{m}_i$表示矩阵**M**的第i列。为了使平面法向朝平头锥体外，需要将上述方程取反，故左平面为：

$$-(\mathbf{m}_3 + \mathbf{m}_0) \bullet (x, y, z, 1) = 0$$

- 其它平面可类推。

Shaft/长方体和Shaft /球体相交测试

- 应用背景：在遮挡裁剪、动态相交测试时，需要找出两个AABB之间存在的对象。
- 两个AABB之间的体连同AABB本身，称为一个Shaft，所涉及的相交测试称为Shaft裁剪(Shaft Culling)。



在满足Shaft定义的情况下，通过增加平面将不在这两个初始AABB之间的空间体去除。

Shaft/长方体和Shaft /球体相交测试

- 为了对**Shaft**和另外一个**AABB**（或者球体）进行相交测试，首先要对**Shaft的AABB**和图元进行相交测试，然后依次对每个平面和图元进行相交测试。
- 如果图元在**AABB**或者任何一个平面外部，则中止测试（图元与**Shaft**不相交）。
- 如果图元完全在所有平面的内部，则图元一定在**Shaft**内部。
- 否则，图元和**Shaft**相交。

“线/线”相交测试

- 两条二维直线的相交测试
  - 假设两条直线分别为

$$\mathbf{r}_1(s) = \mathbf{o}_1 + s\mathbf{d}_1, \text{和} \mathbf{r}_2(s) = \mathbf{o}_2 + s\mathbf{d}_2$$

  - 则交点为：

$$s = \frac{(\mathbf{O}_2 - \mathbf{O}_1) \bullet \mathbf{d}_2^{\perp}}{\mathbf{d}_1 \bullet \mathbf{d}_2^{\perp}},$$

$$t = \frac{(\mathbf{O}_1 - \mathbf{O}_2) \bullet \mathbf{d}_1^{\perp}}{\mathbf{d}_2 \bullet \mathbf{d}_1^{\perp}}$$

两条二维线段的相交计算

- 假设两条线段分别为

$$\mathbf{r}_1(s)=\mathbf{p}_1+s(\mathbf{p}_2-\mathbf{p}_1),\text{和}\mathbf{r}_2(s)=\mathbf{q}_1+s(\mathbf{q}_2-\mathbf{q}_1)$$

- 则交点为：

$$s=\frac{-\mathbf{c}\bullet\mathbf{a}^{\perp}}{\mathbf{b}\bullet\mathbf{a}^{\perp}}=\frac{\mathbf{c}\bullet\mathbf{a}^{\perp}}{\mathbf{a}\bullet\mathbf{b}^{\perp}}=\frac{d}{f},$$

$$t=\frac{\mathbf{c}\bullet\mathbf{b}^{\perp}}{\mathbf{a}\bullet\mathbf{b}^{\perp}}=\frac{e}{f},$$

$$\text{其中}\mathbf{a}=\mathbf{q}_2-\mathbf{q}_1,\mathbf{b}=\mathbf{p}_2-\mathbf{p}_1,\mathbf{c}=\mathbf{p}_1-\mathbf{q}_1$$

伪代码

```
if (f > 0)
    if( d < 0 or d > f ) return NO_INTERSECATION
else
    if( d > 0 or d < f ) return NO_INTERSECATION
```

三个平面之间的相交测试

- 给定3个平面，法向量为$\mathbf{n}_i$, 平面上各有一个任意点$\mathbf{p}_i$，则其交点为：

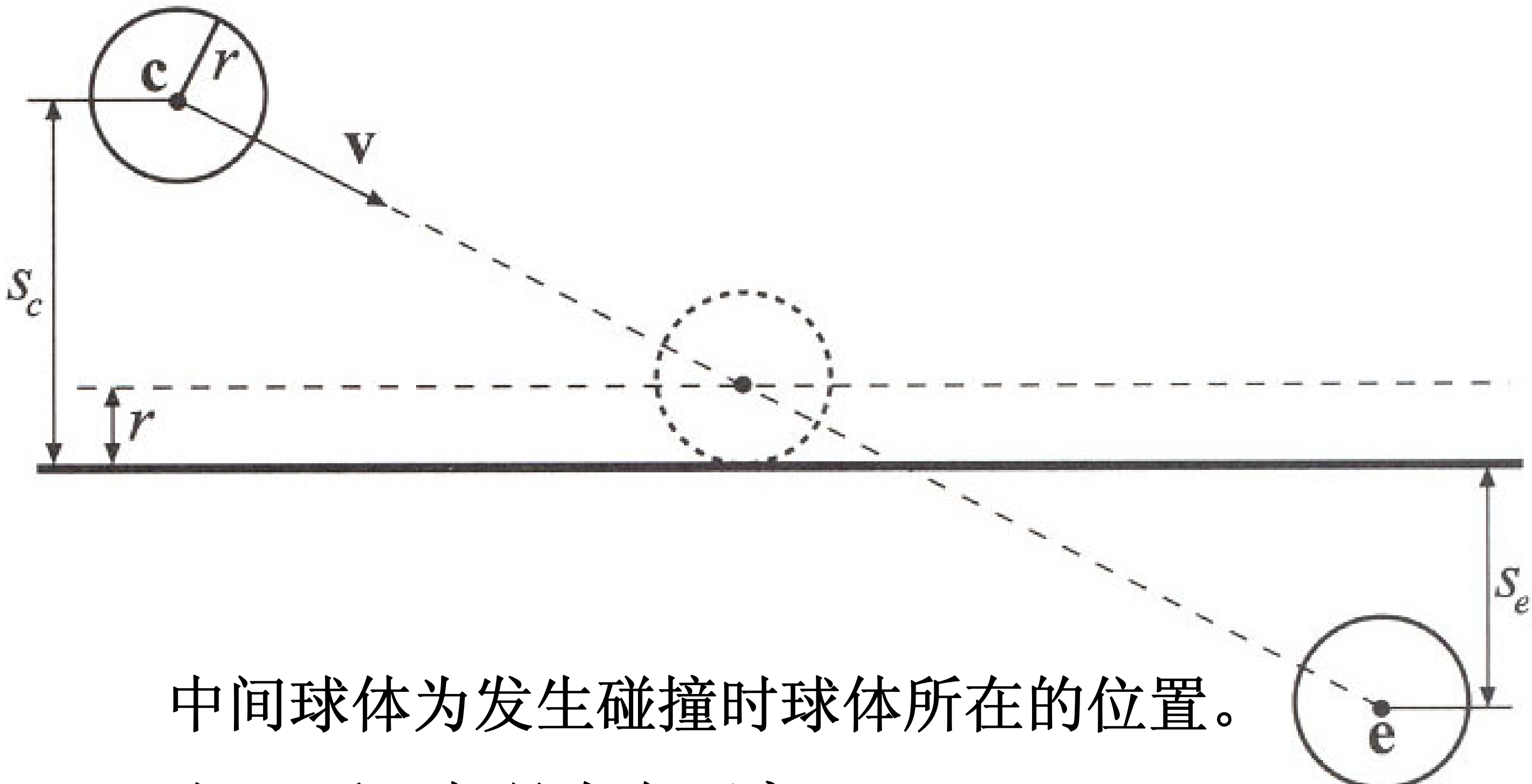
$$\mathbf{p} = \frac{(\mathbf{p}_1 \bullet \mathbf{n}_1)(\mathbf{n}_2 \times \mathbf{n}_3) + (\mathbf{p}_2 \bullet \mathbf{n}_2)(\mathbf{n}_3 \times \mathbf{n}_1) + (\mathbf{p}_3 \bullet \mathbf{n}_3)(\mathbf{n}_1 \times \mathbf{n}_2)}{\begin{vmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3 \end{vmatrix}}$$

- 当三个法向行列式为0时，有平面平行，无交。

动态相交测试

- 动机
  - 在动态场景中，物体在时间轴上是连续运动，用离散的方式(t，t+△t, …)，很可能漏掉相交情形。虽然可以通过加密时间间隔进行相交测试，但仍会错过部分相交测试，而且还会增加计算负荷。因此，需要设计动态相交测试来处理这种情况。
- 动态相交测试的运动相对原则
  - 运动是相对的。
  - 假设A和B均在运动，则可看成A是运动的（当然速度需改变），B是静止的。

“球体/平面”动态相交测试



中间球体为发生碰撞时球体所在的位置。

注：s_c和s_e都是有向距离

"球体/平面"动态相交测试

- 假设球体的中心为**c**，半径为r，在△t内物体匀速直线运动，则球心在一条直线上。

- 计算球心和平面之间的有向距离s_c和s_e

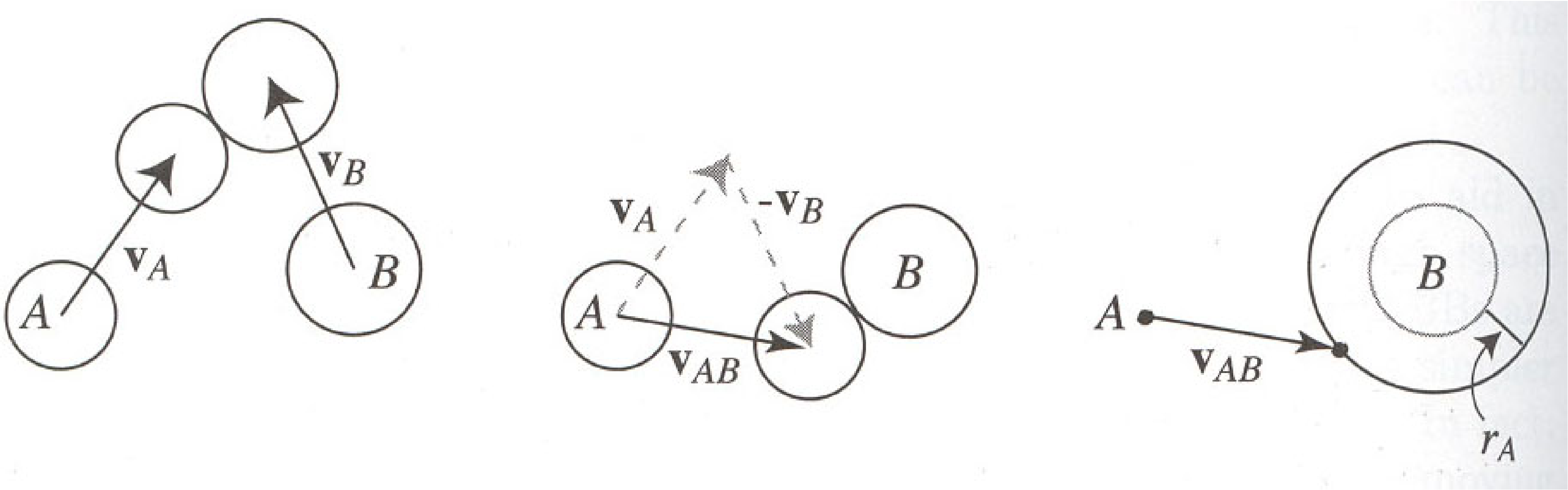
- 如果球心位置在平面的同一侧(即$s_c s_e>0$)，而且$|s_c|>r$和$|s_e|>r$，则球体不会与平面相交。
- 否则需要计算相交的准确时刻和球体位置。

$$t=\frac{s_c-r}{s_c-s_e}$$

“球体/球体”相交测试

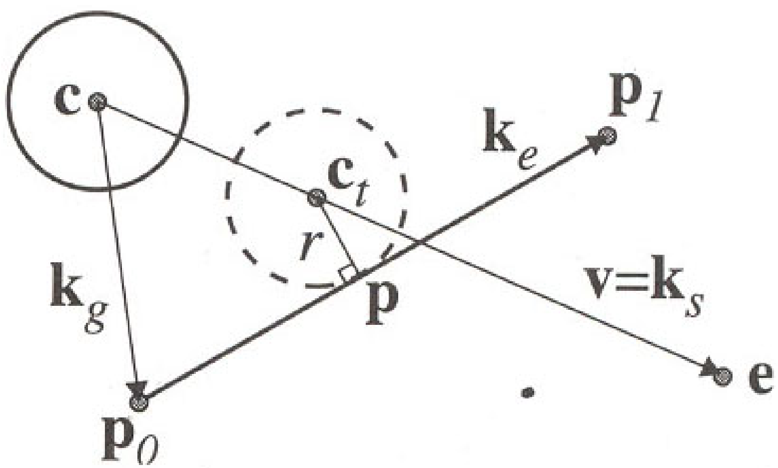
- 两个球体间的相交测试等同于一条射线与静止球体之间的相交测试。（见下页图）
- 首先，使用相对运动原则，把球体**B**看成是静止的。
- 其次，借用平头锥体与球体之间的相交测试方法进行测试。这里，可以使一个球在另外一个球的表面移动，从而得到一个新的球体，半径为两个球体的半径之和。
- 假设球体**B**是新的球体且处于静止状态，半径是两个球体之和，则球体**A**就是沿着直线运动的一个点。

“球体/球体”相交测试

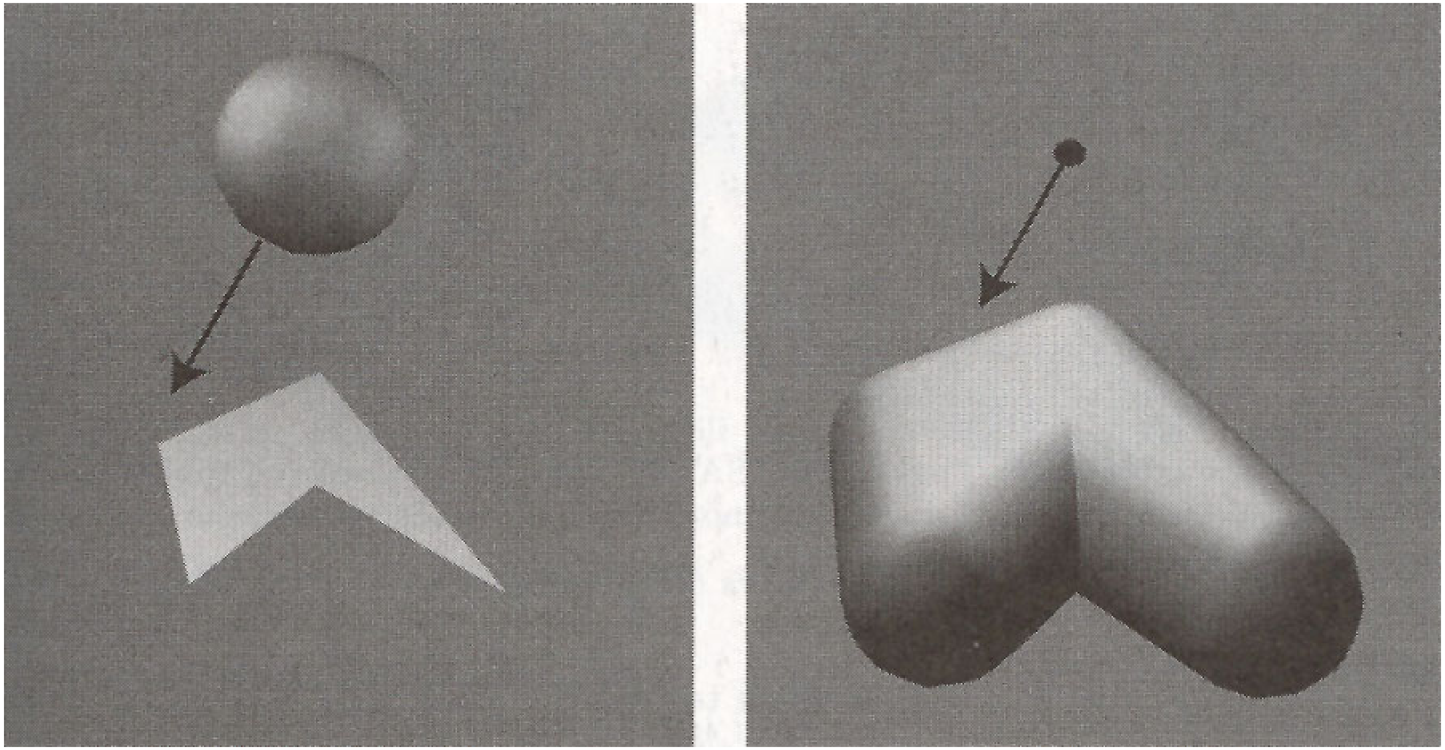


“球体/多边形”动态相交测试

- 利用前面的平面与球体的相交方法，测试球体是否和多边形所在的平面相交，如果不相交，则结束。
- 否则，计算球体首次与平面相交的时刻，然后检查交点是否在多边形内部，如果是，则相交；否则需要进一步的检查球体是否与多边形的边或点相交。



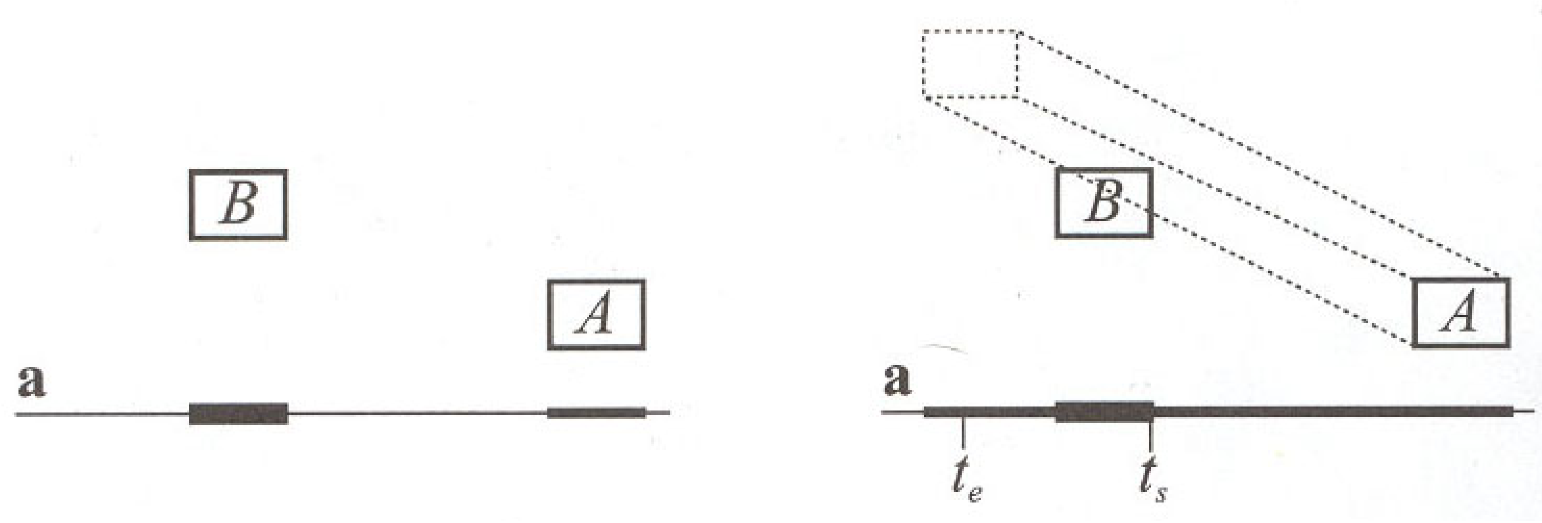
等价处理法



左图：一个球体正向一个多边形移动

右图：一条射线指向一个膨胀的多边形，这两种测试是等价的。

动态分离轴测试方法



左图：对于轴**a**的分离轴定理**(SAT)**，其中**A**和**B**在该轴上没有重叠。

右图：对于轴**a**的**SAT**，其中**A**是移动的，在运动过程中对**A**和它在轴**a**上的区间投影进行跟踪，可以看出两个物体在轴**a**上是重叠的。