第10讲：光线追踪

中国海洋大学 周佳然

上次课程内容

- 几何处理(Geometry Processing)
  - 几何处理的常见任务
  - 网格细分
  - 网格简化
- 阴影贴图（Shadow Mapping）

OCEAN UNIVERSITY OF CHINA

本次课程内容

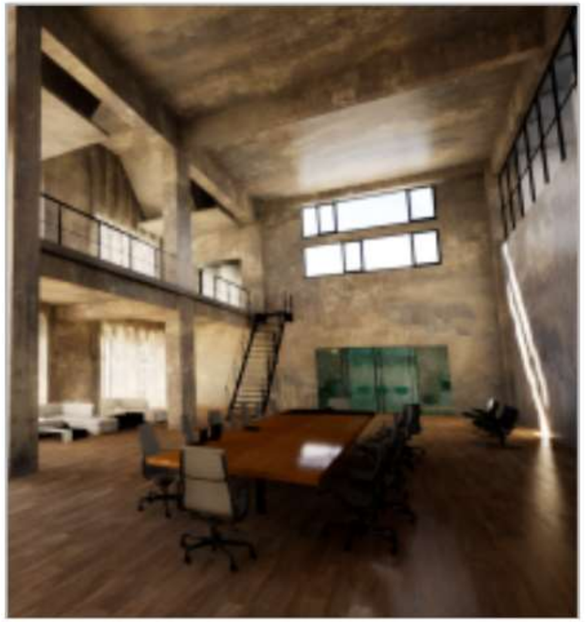
- 光线追踪(Ray Tracing)
  - ➢ 为什么要做光线追踪?
  - ➢ 基本的光线追踪算法：Whitted-Style Ray Tracing
  - ➢ 技术细节：如何求交？如何加速求交计算？
- 实验5发布
- 实验3提交截止
- 实验1-3报告提交截止
- 期中综述报告提交截止

为什么要做光线追踪?

- 光栅化不能很好地处理全局的效果
- ➢ （软）阴影
- ➢ 尤其是当光线在场景中多次弹射时

软阴影

光泽反射
（Glossy Reflection）



间接光照

为什么要做光线追踪?

- 光栅化：速度快（实时），但生成质量相对较差

为什么要做光线追踪？

- 光线追踪：很准确，生成质量高，但速度慢（离线）

中國海洋大學 OCEAN UNIVERSITY OF CHINA

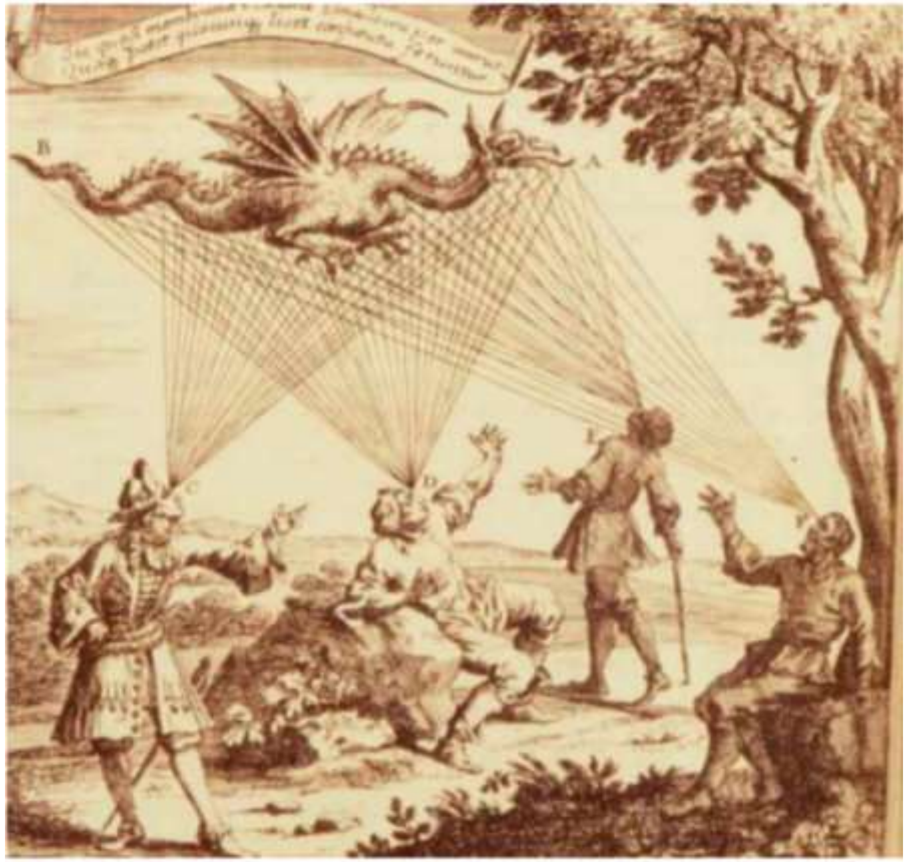
基本的光线追踪算法

- 关于光线的几点假设：

➢ 光沿直线传播（尽管这是错误的）

➢ 光线交叉时不会相互碰撞（虽然这也是错误的）

➢ 光线从光源传播到眼睛（物理上，路径是可逆的）

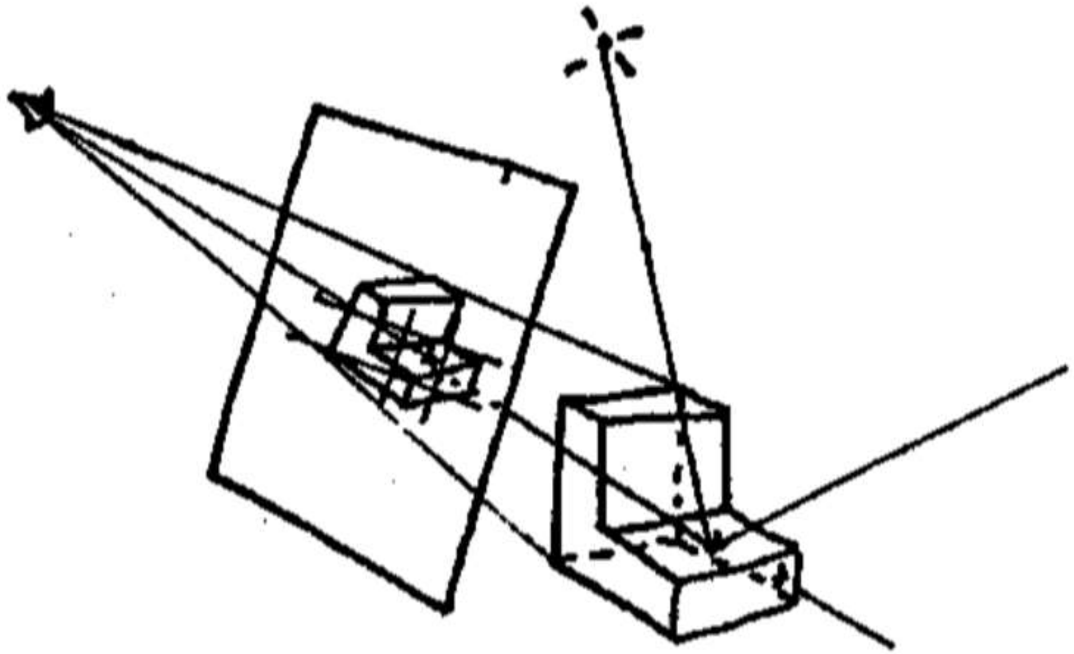


Eyes send out “feeling rays” into the world

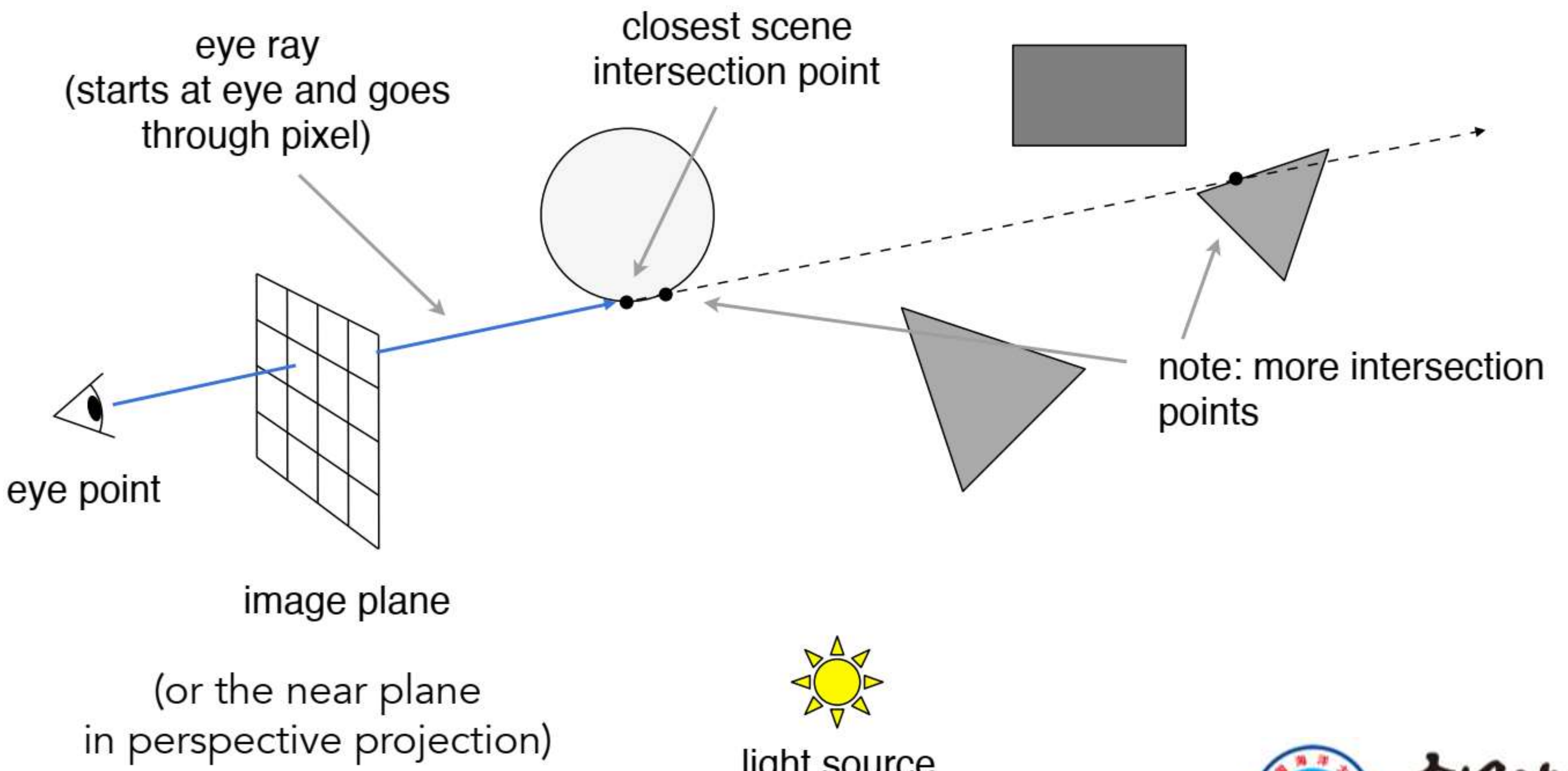
中國海洋大學 OCEAN UNIVERSITY OF CHINA

光线投射（Ray Casting）算法

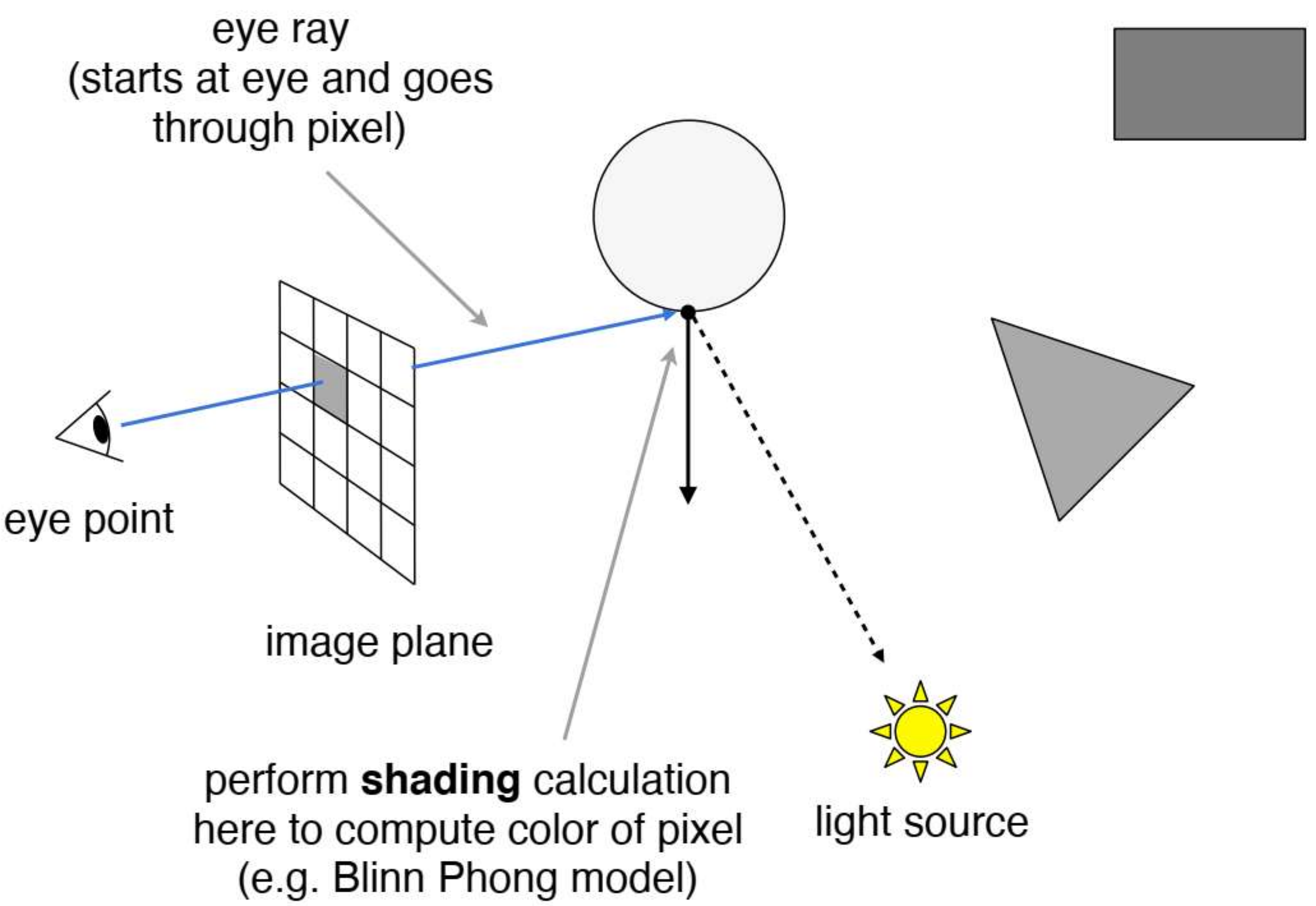
- 由Appel于1968年提出
  - 通过每个像素投射一条光线来生成图像
  - 通过向光源发送光线来检查阴影(表面可见性判别)



光线投射算法:eye rays



光线投射算法:shading pixels



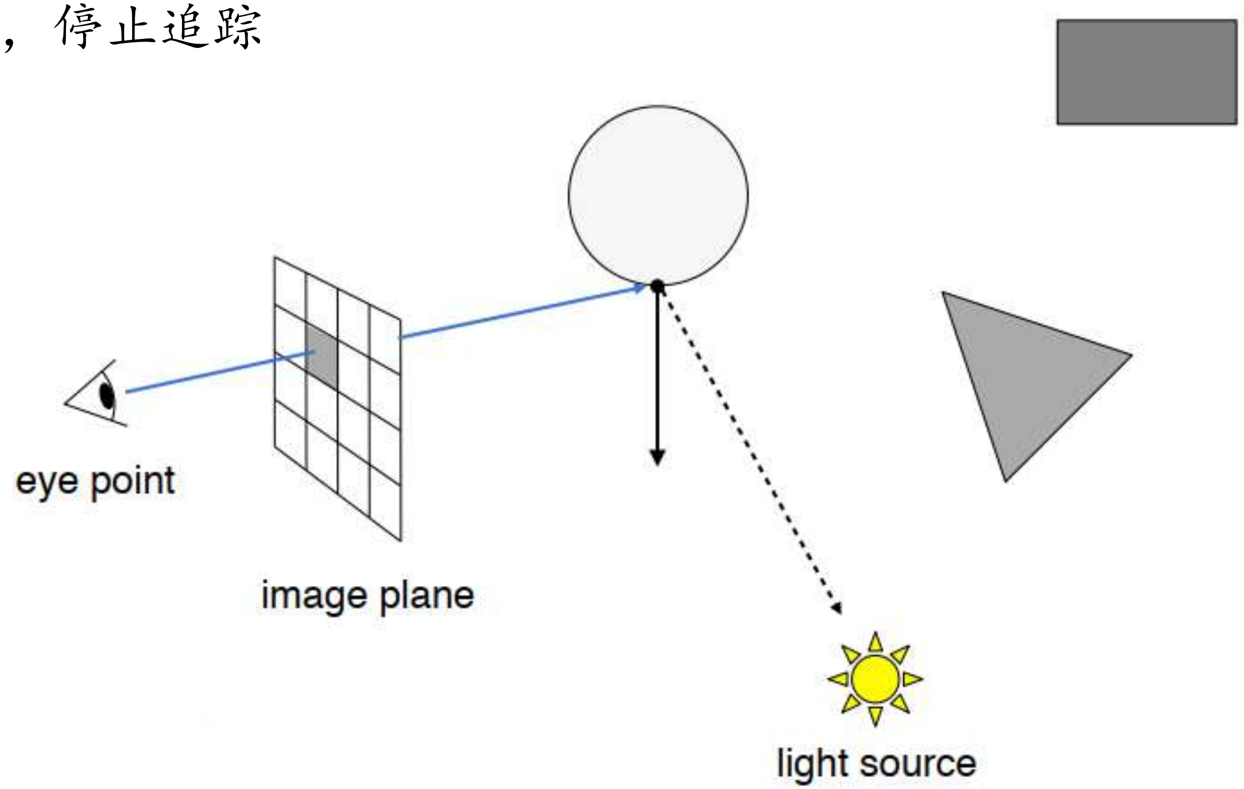
OCEAN UNIVERSITY OF CHINA

光线投射算法

- 基本思想

➢ 从视点或像素出发，仅对穿过像素的光线反向跟踪

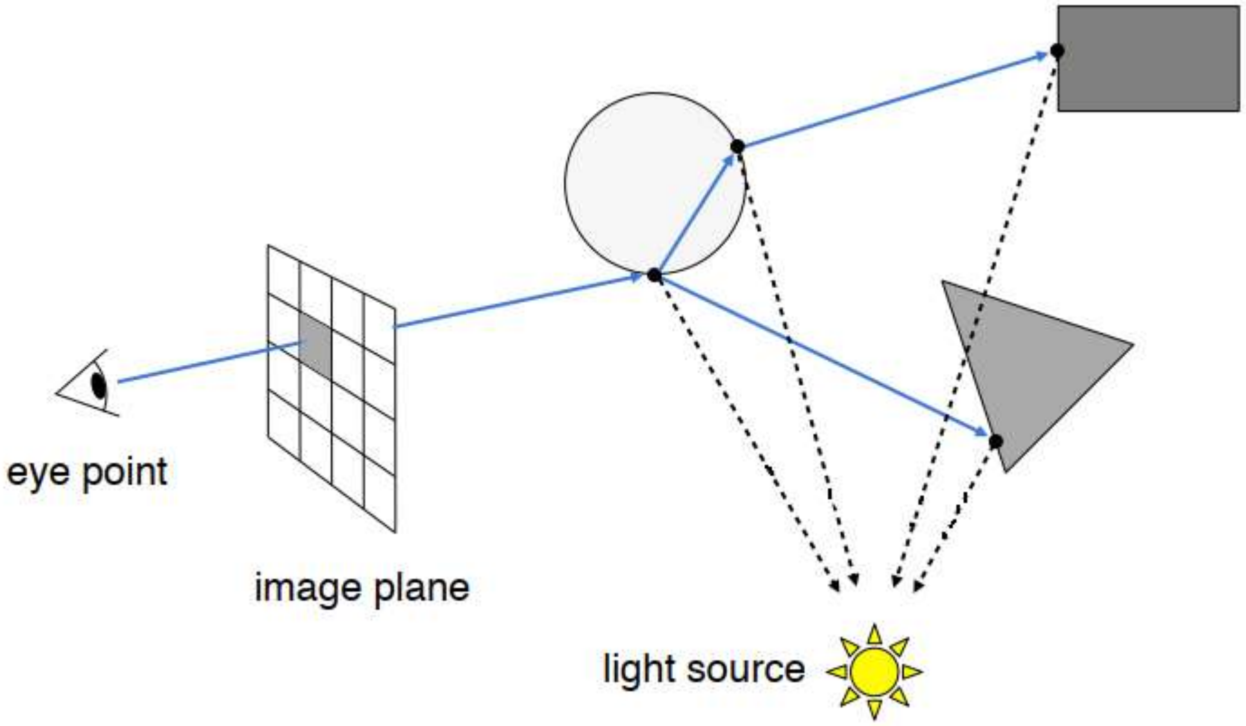
➢ 当光线路径到达一个离视点最近的可见的不透明物体的表面(即为屏幕上该像素对应的可见面)时，停止追踪



Whitted-Style光线追踪算法

- 递归的光线追踪（Recursive Ray Tracing）

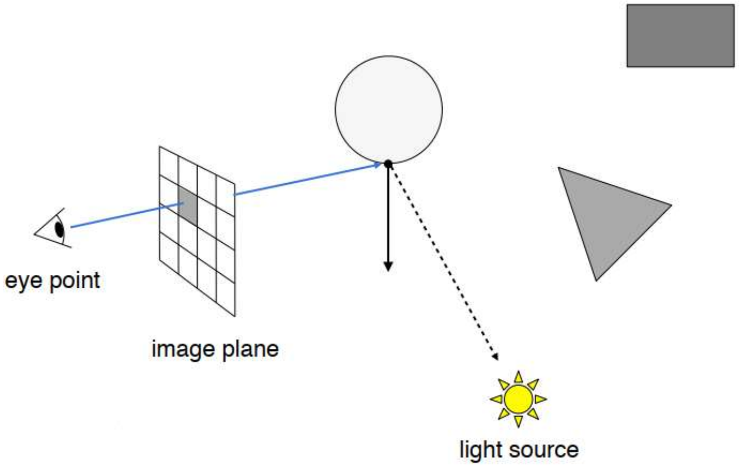
➢ 由Kay和Whitted于1979年提出，它不仅为每个像素寻找可见面，还跟踪光线在场景中的反射和折射，并计算它们对总光强的贡献

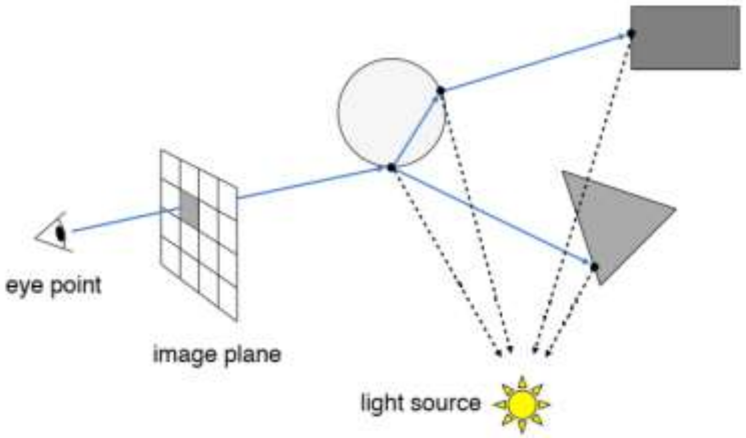


中國海洋大學 OCEAN UNIVERSITY OF CHINA

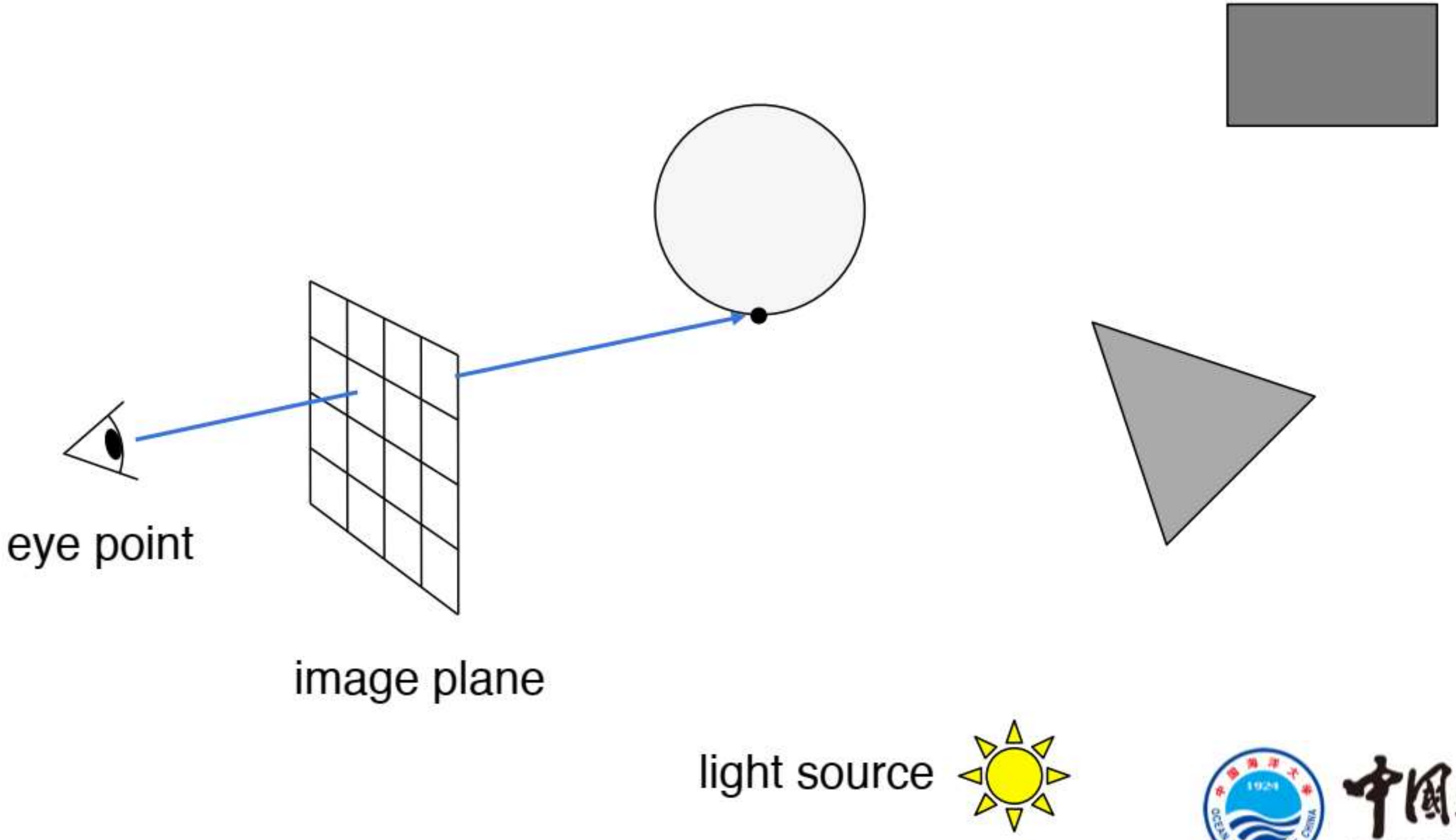
Whitted-Style光线追踪算法

- 光线投射算法：被追踪的光线仅从每个像素到离它最近的景物为止
- Whitted-Style光线追踪算法：通过追踪多条光线在场景中的路径，以得到多个景物表面所产生的反射和折射影响
- **Whitted-Style 光线追踪算法是光线投射算法的延伸**

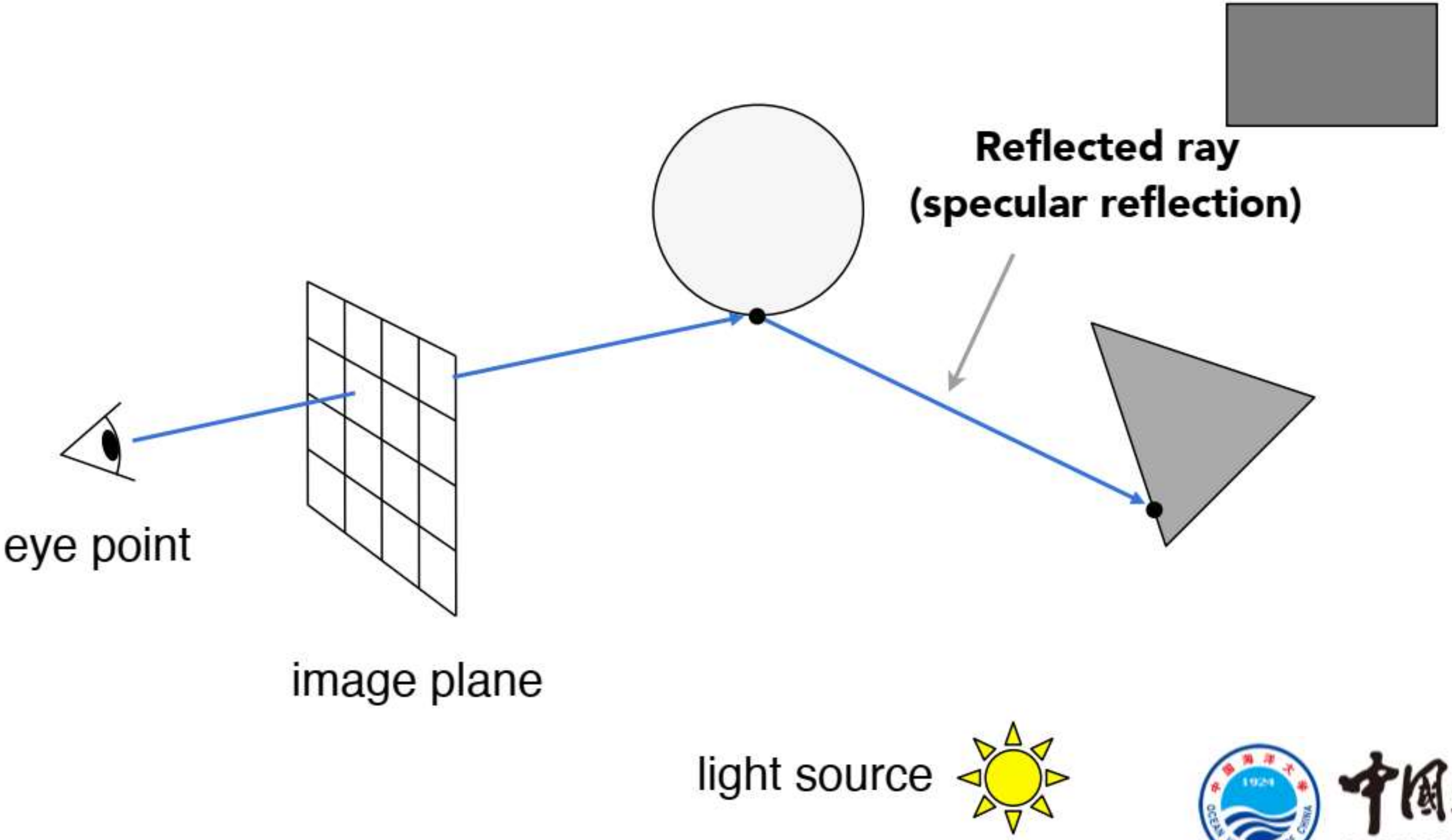




Whitted-Style光线追踪算法

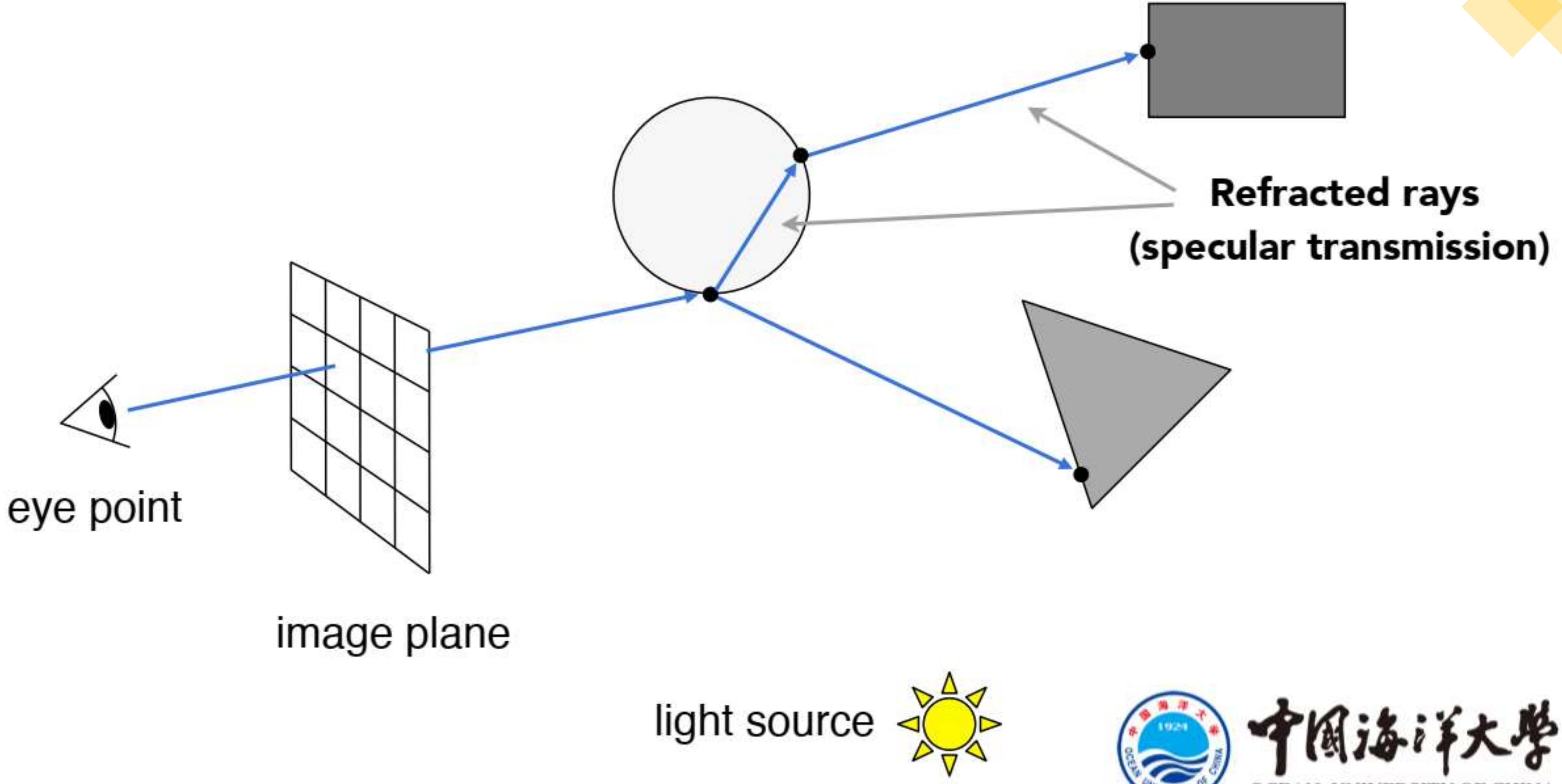


Whitted-Style光线追踪算法

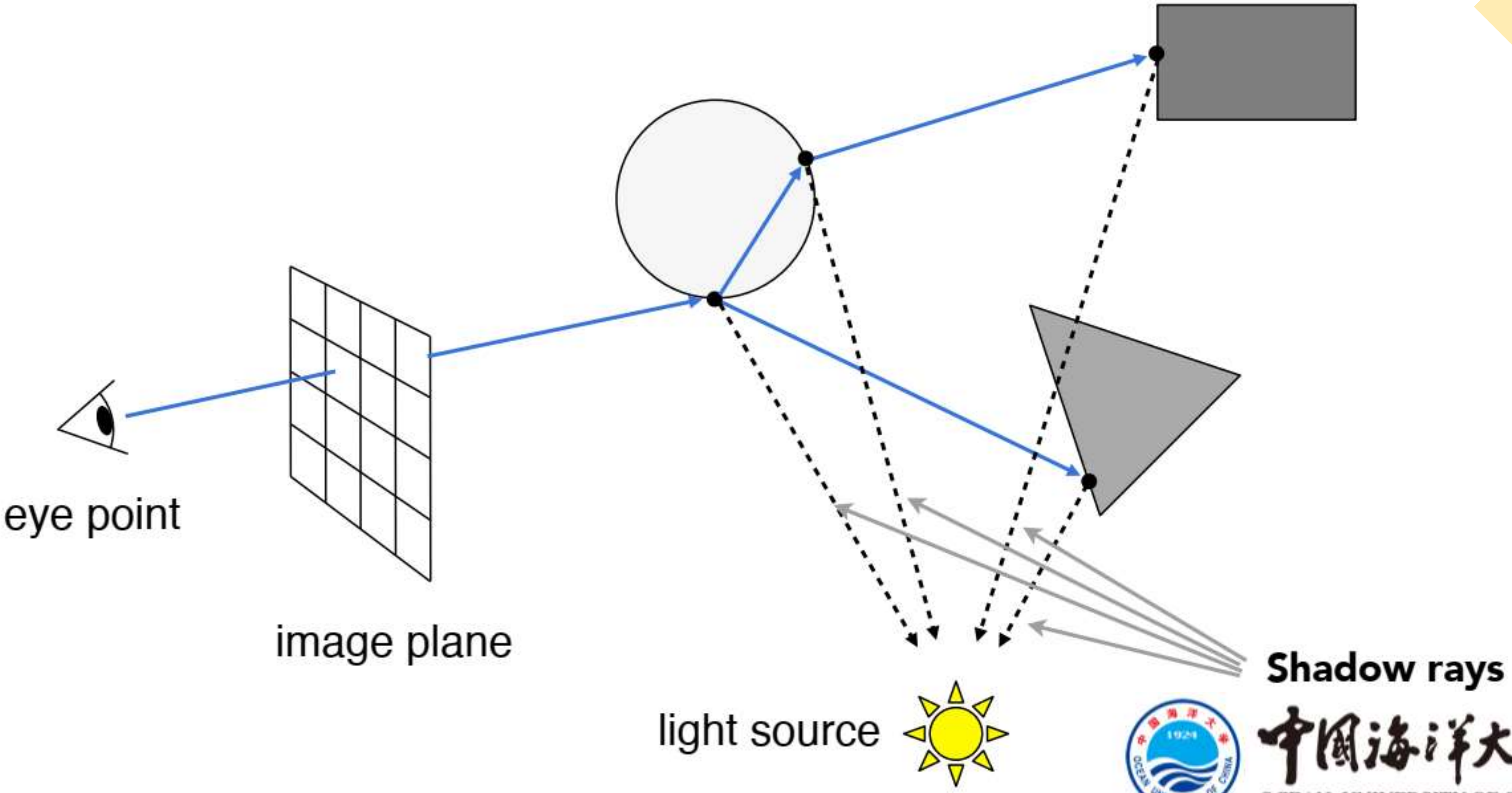


OCEAN UNIVERSITY OF CHINA

Whitted-Style光线追踪算法



Whitted-Style光线追踪算法



中國海洋大學 OCEAN UNIVERSITY OF CHINA

Whitted-Style光线追踪算法

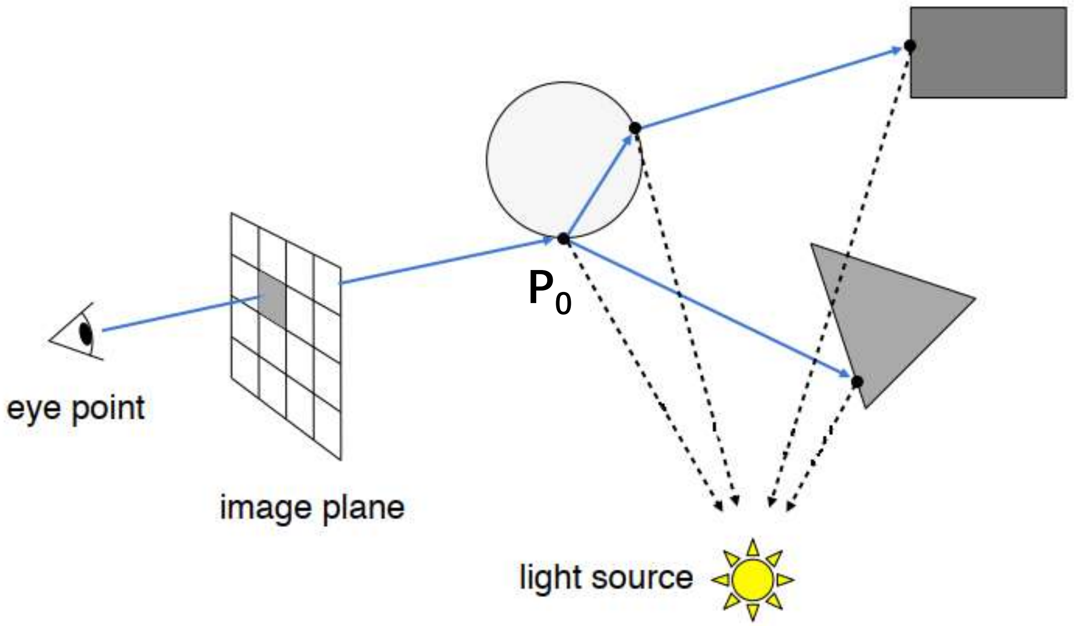
- 基本思想

➢ 沿着到达视点的光线的相反方向追踪

➢ 经过屏幕上一像素点找出与视线所交的物体表面点P_0

➢ 继续追踪，找出影响P_0点光强的所有的光源

➢ 算出P_0点上精确的光照强度



Whitted-Style光线追踪算法

- 本质上是一个递归算法

➢ 每个像素的光强度必须综合各级递归计算的结果才能获得

- 结束条件

➢ 光线与光源相交

➢ 光线与漫反射表面相交

➢ 被追踪的光线对第一个交点处的贡献趋近于0

Whitted-Style光线追踪算法

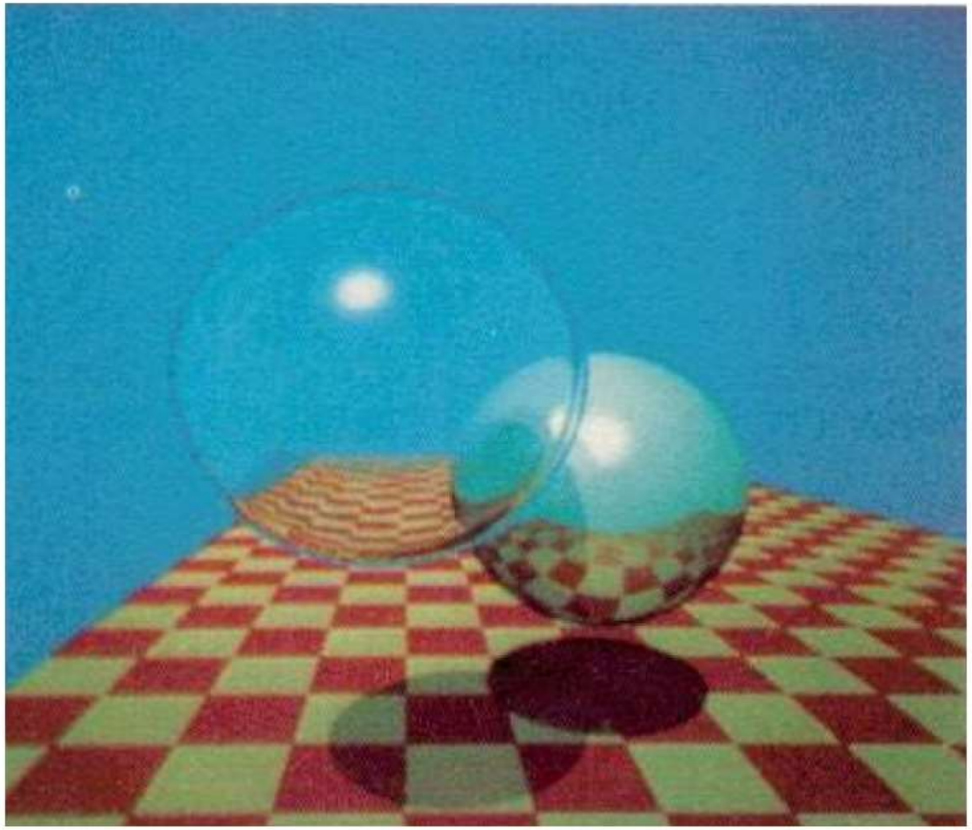
- 时间：

  VAX 11/780(1979) 74m

  PC(2006) 6s

  GPU(2012) 1/30s

Q:复杂图形的时间花费？



Spheres and Checkerboard, T. Whitted, 1979

Whitted-Style光线追踪算法

- 需要从视点出发通过屏幕发射大量的光线并对它们一一进行追踪

- 光线与物体的求交计算

➢ 每一条射线要与所有的物体求交，然后再对所得的全部交点排序，才能确定可见点

➢ 75%以上的工作量用于求交计算

- 光线追踪算法实用的关键

➢ 线与面求交算法的效率

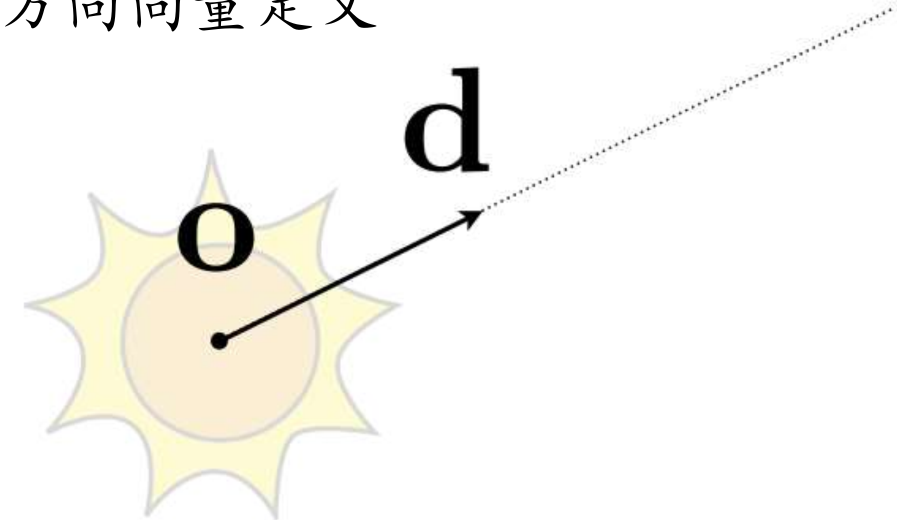
中國海洋大學 OCEAN UNIVERSITY OF CHINA

Q&A

中國海洋大學
OCEAN UNIVERSITY OF CHINA

光线的定义

- 光线由原点和一个方向向量定义



- 光线上一点的计算公式:

$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d} \qquad 0 \le t < \infty$$

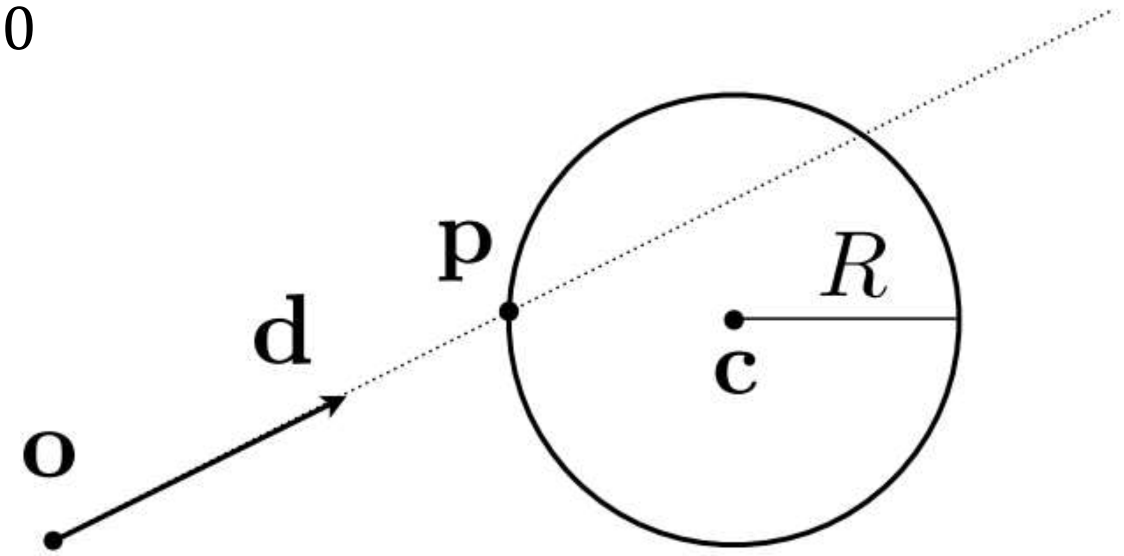
point along ray　“time”　origin　(normalized) direction

光线与球面求交

- 光线上一点：$\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d} \qquad 0 \leq t < \infty$
- 球面上一点p：$(\boldsymbol{p} - \boldsymbol{c})^2 - R^2 = 0$

Q:光线与球面求交?

$$(\boldsymbol{o} + t\boldsymbol{d} - \boldsymbol{c})^2 - R^2 = 0$$



光线与球面求交

- 求解光线与球面的交点

$$(\boldsymbol{o} + t\boldsymbol{d} - \boldsymbol{c})^2 - R^2 = 0$$

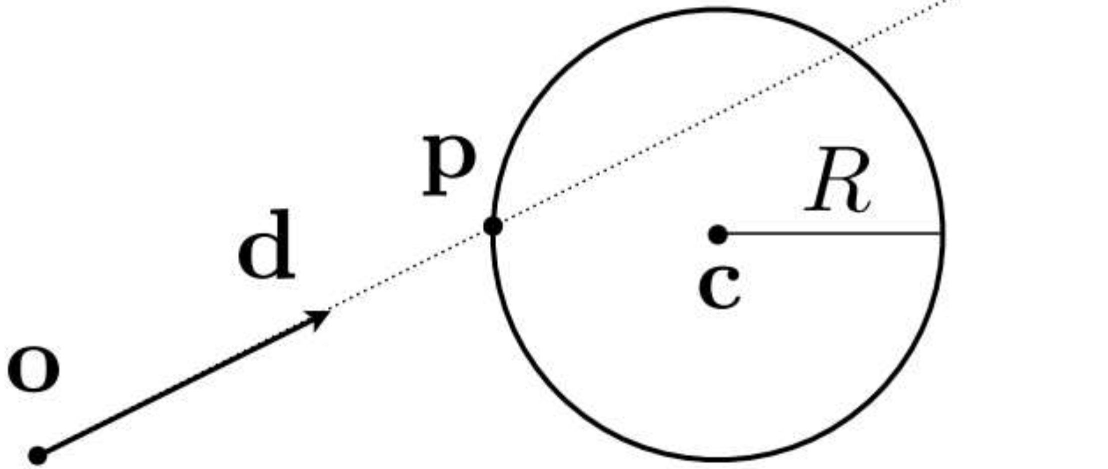
$$at^2 + bt + c = 0，\text{其中}$$

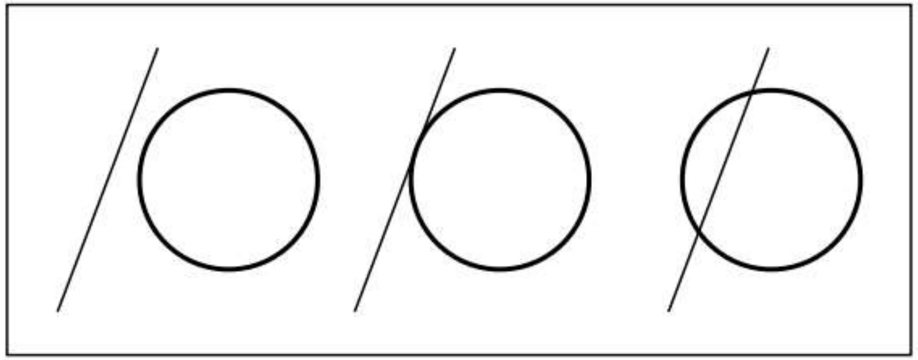
$$a = \boldsymbol{d} \cdot \boldsymbol{d}$$

$$b = 2(\boldsymbol{o} - \boldsymbol{c}) \cdot \boldsymbol{d}$$

$$c = (\boldsymbol{o} - \boldsymbol{c}) \cdot (\boldsymbol{o} - \boldsymbol{c}) - R^2$$

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$





光线与隐式曲面求交

- 光线上一点：$\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d} \qquad 0 \le t < \infty$
- 隐式曲面上一点p：$f(\boldsymbol{p}) = 0$

Q:光线与隐式曲面求交?

计算 $f(\boldsymbol{o} + t\boldsymbol{d}) = 0$的正实数根

$x^2+y^2+z^2=1$

$(R-\sqrt{x^2+y^2})^2+z^2=r^2$

$(x^2+\frac{9y^2}{4}+z^2-1)^3 = x^2z^3+\frac{9y^2z^3}{80}$

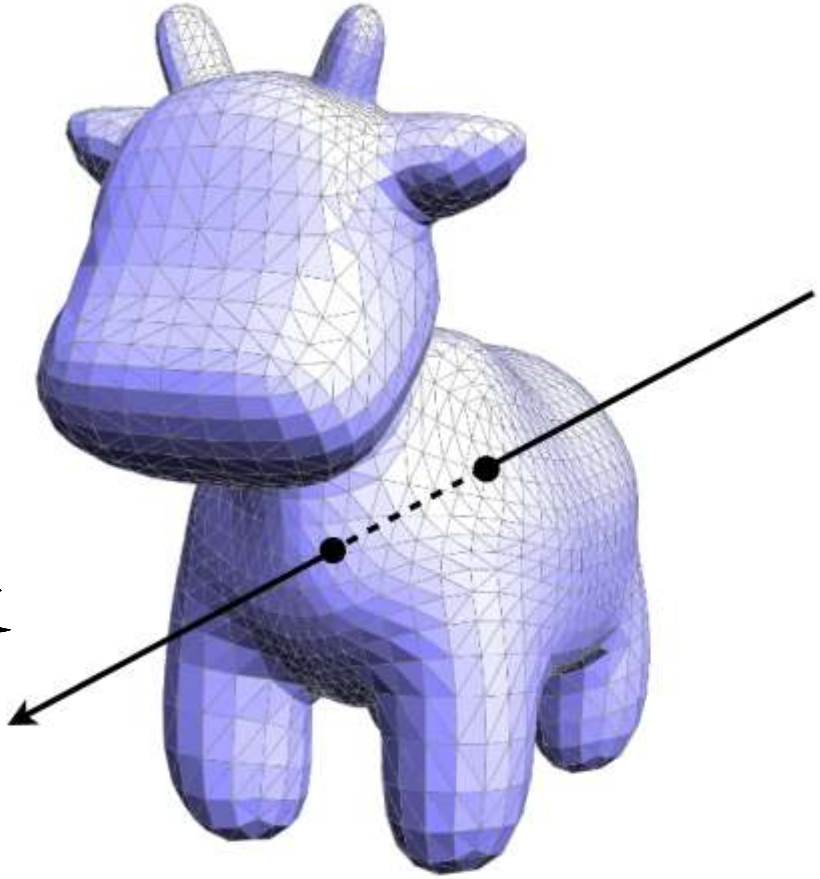
光线与三角网格求交

Q:为什么要计算光线与三角网格的交点?

- 渲染：可见性、阴影、光照的计算
- 几何：内部/外部测试

Q:如何计算?

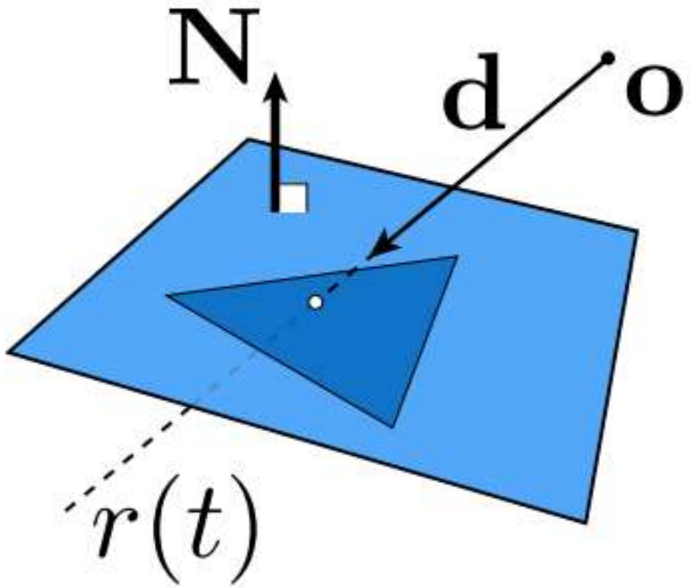
- 简单直接的思路：光线与每一个三角形求交
- 速度慢（如何加速？）
- 注意：只考虑交点数为0或1



光线与三角形求交

- 考虑三角形所在的平面
  - 光线与平面求交
  - 检查交点是否在三角形内

- Möller Trumbore算法
  - 直接计算光线与三角形的交点

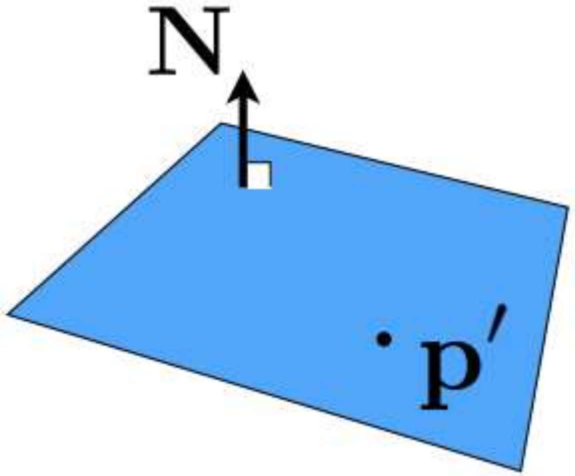


OCEAN UNIVERSITY OF CHINA

光线与平面求交

- 平面由平面的法向量与平面内的一点定义
- 平面方程：

$$\underset{\text{all points on plane}}{\mathbf{p}} : (\mathbf{p} - \underset{\text{one point on plane}}{\mathbf{p}'}) \cdot \underset{\text{normal vector}}{\mathbf{N}} = 0$$



光线与平面求交

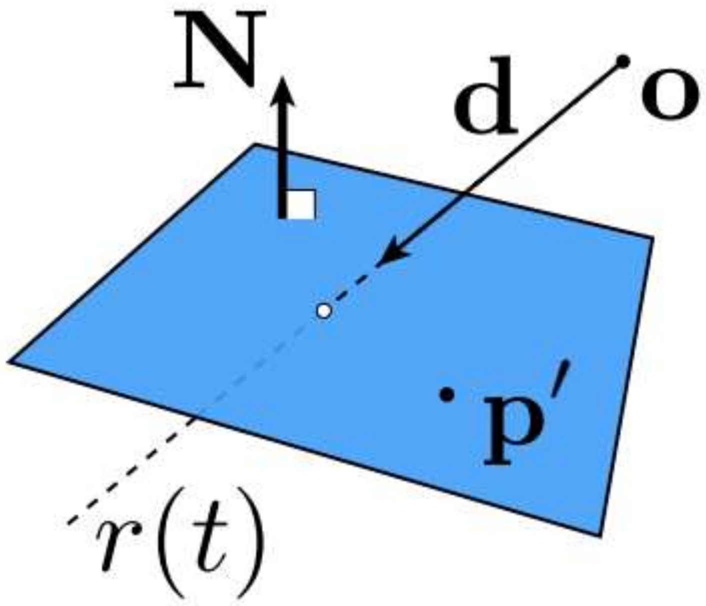
- 光线上一点：$\boldsymbol{r}(t)=\boldsymbol{o}+t\boldsymbol{d}$ 　　$0\le t<\infty$
- 平面上一点p：$(\boldsymbol{p}-\boldsymbol{p}')\cdot\boldsymbol{N}=0$

- 求解交点：

$$(\boldsymbol{p}-\boldsymbol{p}')\cdot\boldsymbol{N}=(\boldsymbol{o}+t\boldsymbol{d}-\boldsymbol{p}')\cdot\boldsymbol{N}=0$$

$$t=\frac{(\boldsymbol{p}'-\boldsymbol{o})\cdot\boldsymbol{N}}{\boldsymbol{d}\cdot\boldsymbol{N}}$$

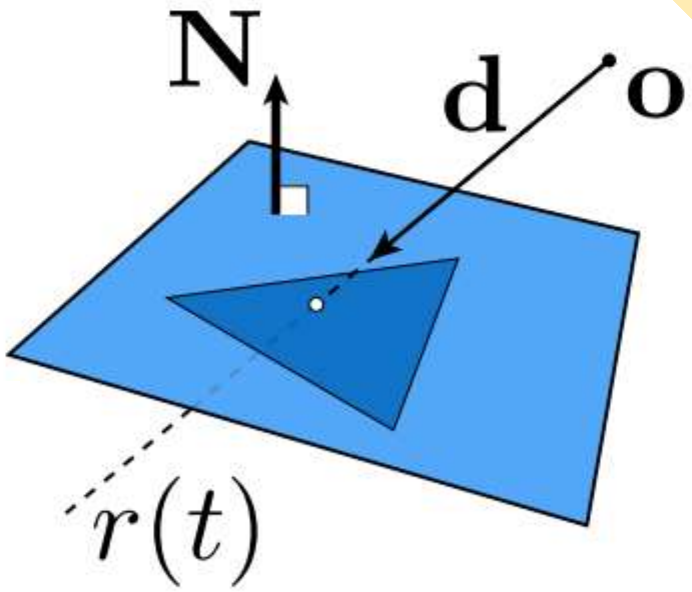
检查t是否满足$0\le t<\infty$



OCEAN UNIVERSITY OF CHINA

Möller Trumbore算法

- 一种计算光线与三角形交点的更快速的方法
- 利用重心坐标



$$\vec{\mathbf{O}} + t\vec{\mathbf{D}} = (1 - b_1 - b_2)\vec{\mathbf{P}}_0 + b_1\vec{\mathbf{P}}_1 + b_2\vec{\mathbf{P}}_2$$

$$\begin{bmatrix} t \\ b_1 \\ b_2 \end{bmatrix} = \frac{1}{\vec{\mathbf{S}}_1 \bullet \vec{\mathbf{E}}_1} \begin{bmatrix} \vec{\mathbf{S}}_2 \bullet \vec{\mathbf{E}}_2 \\ \vec{\mathbf{S}}_1 \bullet \vec{\mathbf{S}} \\ \vec{\mathbf{S}}_2 \bullet \vec{\mathbf{D}} \end{bmatrix}$$

Where:

$$\vec{\mathbf{E}}_1 = \vec{\mathbf{P}}_1 - \vec{\mathbf{P}}_0$$
$$\vec{\mathbf{E}}_2 = \vec{\mathbf{P}}_2 - \vec{\mathbf{P}}_0$$
$$\vec{\mathbf{S}} = \vec{\mathbf{O}} - \vec{\mathbf{P}}_0$$
$$\vec{\mathbf{S}}_1 = \vec{\mathbf{D}} \times \vec{\mathbf{E}}_2$$
$$\vec{\mathbf{S}}_2 = \vec{\mathbf{S}} \times \vec{\mathbf{E}}_1$$

Cost = (1 div, 27 mul, 17 add)

Q:如何确定交点是否在三角形内？

光线追踪的效率

- 简单的光线场景求交
  - 光线与每一个三角形求交
  - 找到最近点（最小t值）

- 问题
  - 规模：像素数*三角形数（*弹射数）
  - 非常慢！需要加速方法！

光线追踪的效率

三角形数目：1070万

光线追踪的效率

三角形数目：2000万

光线追踪的加速技术

Q:如何提升光线场景求交的效率?

- 包围盒：包围目标的简单形体

➢ 如果光线与包围盒不相交，那它与目标物体也不相交

➢ 先测试光线与包围盒是否相交，如果相交再做光线与物体的求交计算

➢ 避免盲目求交

光线与包围盒求交

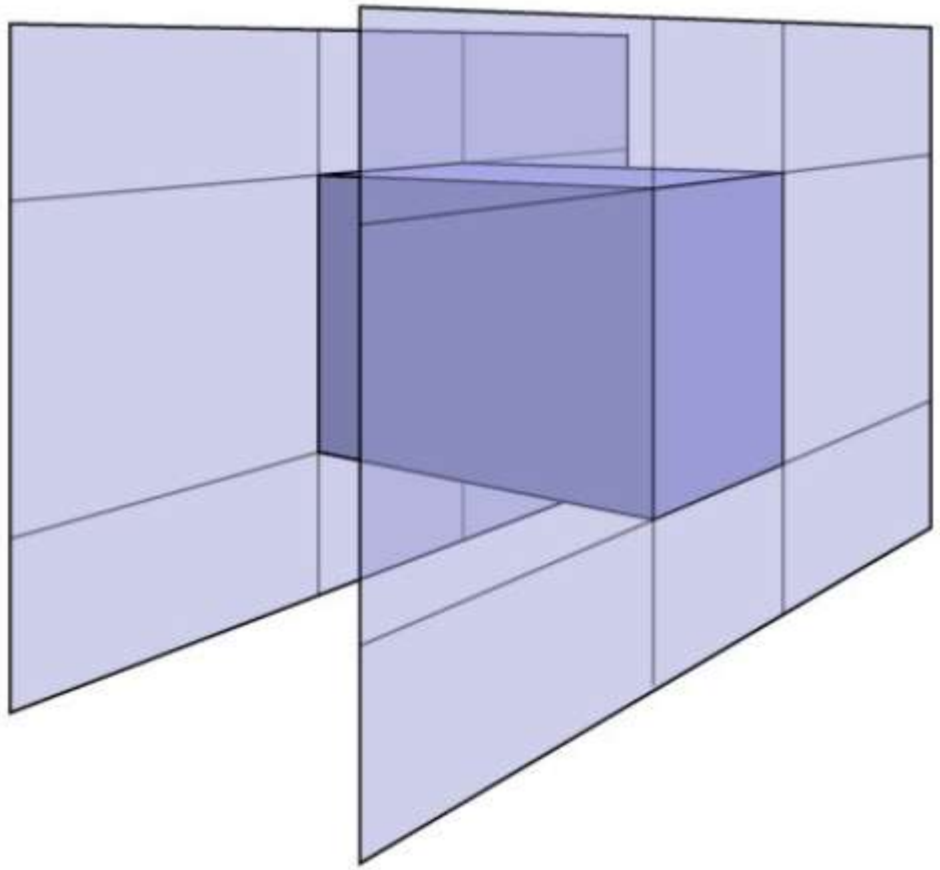
Q:回忆之前在哪里使用过包围盒技术?

- 这里对包围盒（volume）的理解:

➢ 三对平板的交集

- 轴对齐包围盒（AABB）

➢ Axis-Aligned Bounding Box



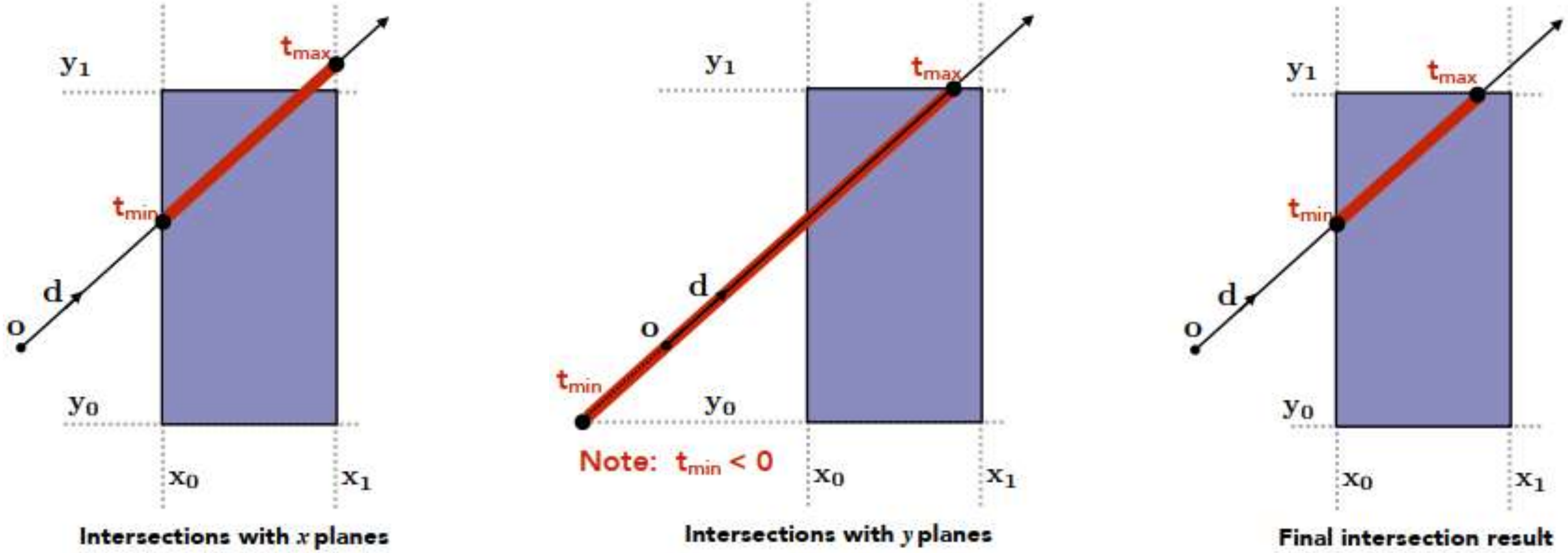
OCEAN UNIVERSITY OF CHINA

光线与轴对齐包围盒求交

- 二维的简单示例：

➢ 分别计算光线与两对平板的交点

➢ 求线段的交集，确定最后结果



光线与轴对齐包围盒求交

- 推广到三维：
  - 分别计算光线与三对平板的交点
  - 求线段的交集，确定最后结果

- 主要思想
  - 仅当光线进入到所有成对平板时，光线进入到包围盒内
  - 光线离开任意一对平板时，即离开了包围盒
  - 对每一对平板，计算t_{min}和t_{max}
  - 对包围盒， $t_{enter}=\max\{t_{min}\}$， $t_{exit}=\min\{t_{max}\}$

中國海洋大學 OCEAN UNIVERSITY OF CHINA

光线与轴对齐包围盒求交

Q:如何确定有交点?

Q:如果有$t_{enter} < t_{exit}$可以确定有交点吗?

- 注意光线是一条射线，还需要考虑

➢ 如果t_{enter}和t_{exit}都非负，说明有交点;

➢ 如果有$t_{exit} < 0$, 说明包围盒在光线“后面”，无交点;

➢ 如果有$t_{exit} \geq 0$并且$t_{enter} < 0$, 说明光源在包围盒内，有交点

- 综上，光线与轴对齐包围盒相交，当且仅当$t_{enter} < t_{exit}$并且$t_{exit} \geq 0$

Q&A