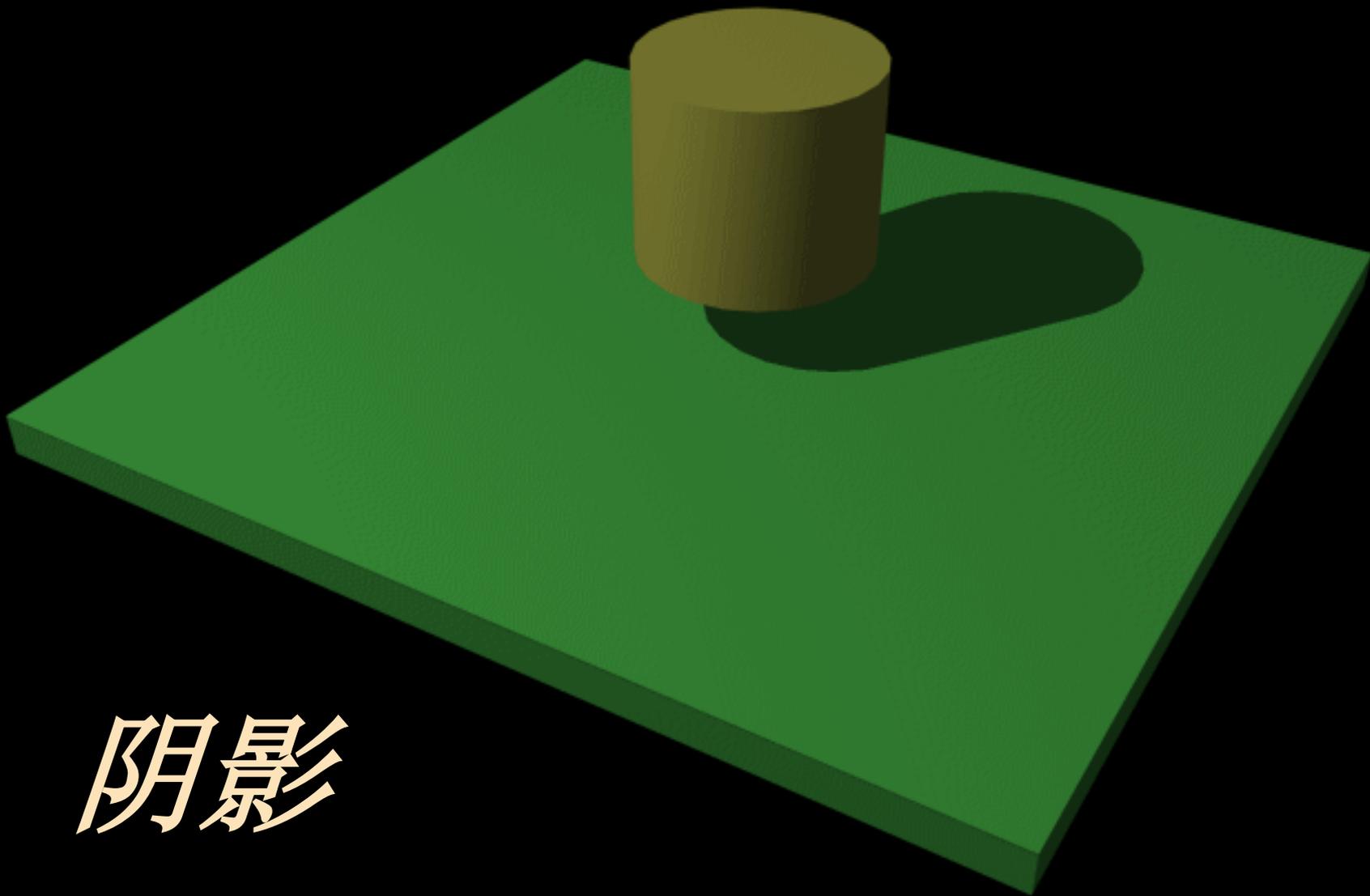


整体光照模型

阴影, 光线跟踪

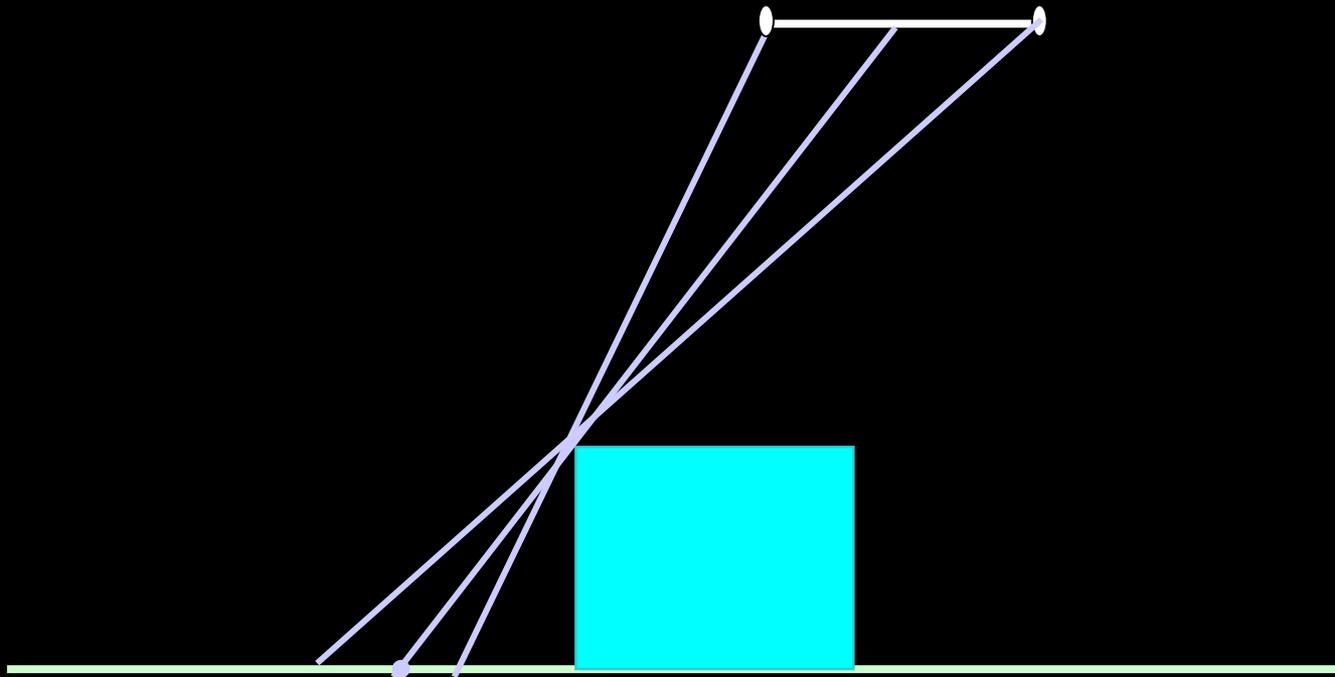


阴影

阴 影

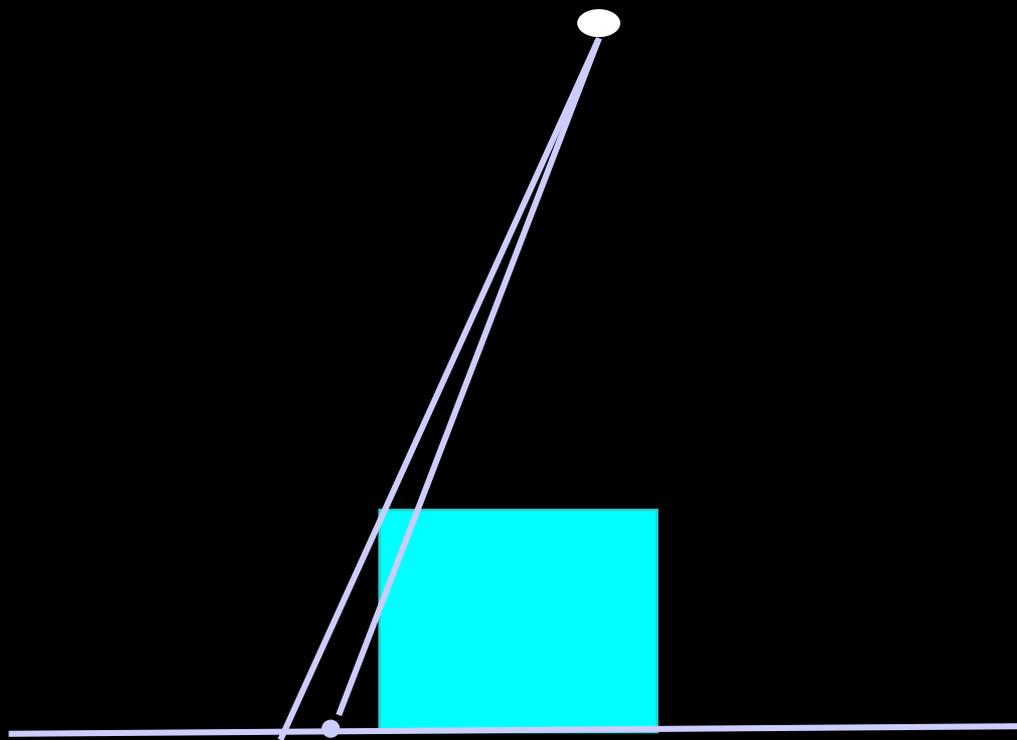
- 阴影消除了物体漂浮在地面上的感觉
- 阴影强调光源的变化方向
- 阴影可能是锐边或软边，本影和半影，具体取决于光源的形状及其与光源的距离
- 没有阴影的照明场景可能会令人困惑并影响真实感

本影和半影



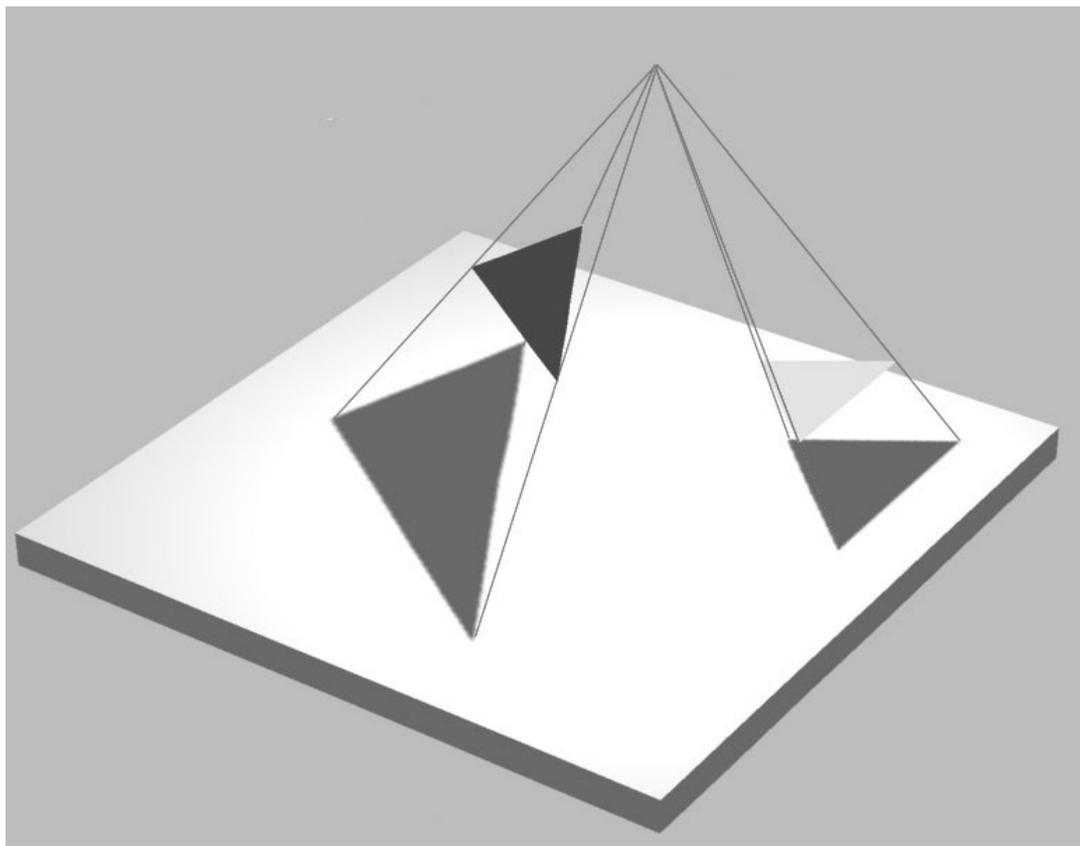
$$I = I_a k_a + \sum_{1 \leq i \leq m} I_{pi} [k_d (\bar{N} \cdot \bar{L}_i) + k_s (\bar{R}_i \cdot \bar{V})^n]$$

先针对点光源考察阴影生成方法

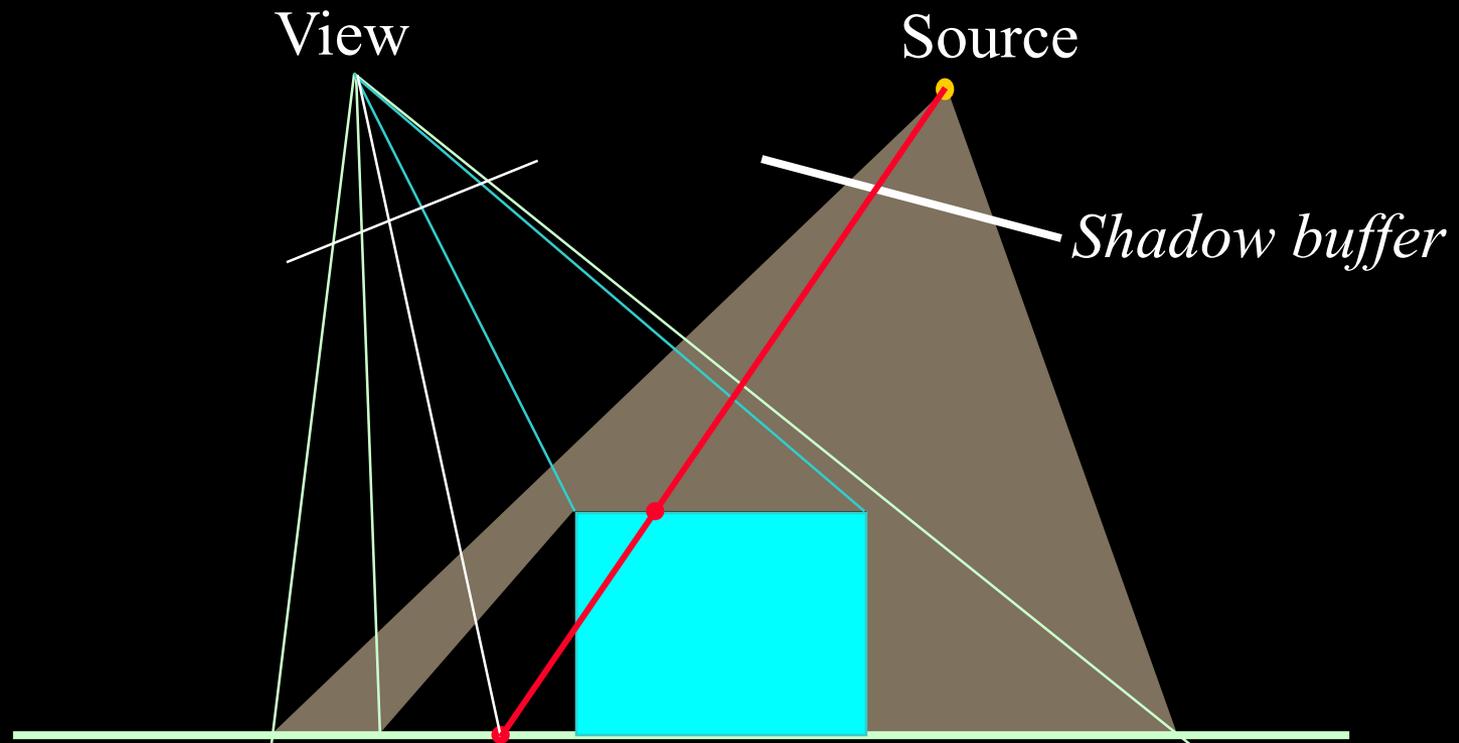


方法1: 阴影体

- 对场景中的每个多边形，生成一个影域棱台, 位于该棱台内的点都处于阴影中。
- 方法的缺点是计算量极大。



方法2: 阴影缓存法 (*Shadow buffer*)



$$I = I_a k_a + \sum_{1 \leq i \leq m} I_{pi} [k_d (\bar{N} \cdot \bar{L}_i) + k_s (\bar{R}_i \cdot \bar{V})^n]$$

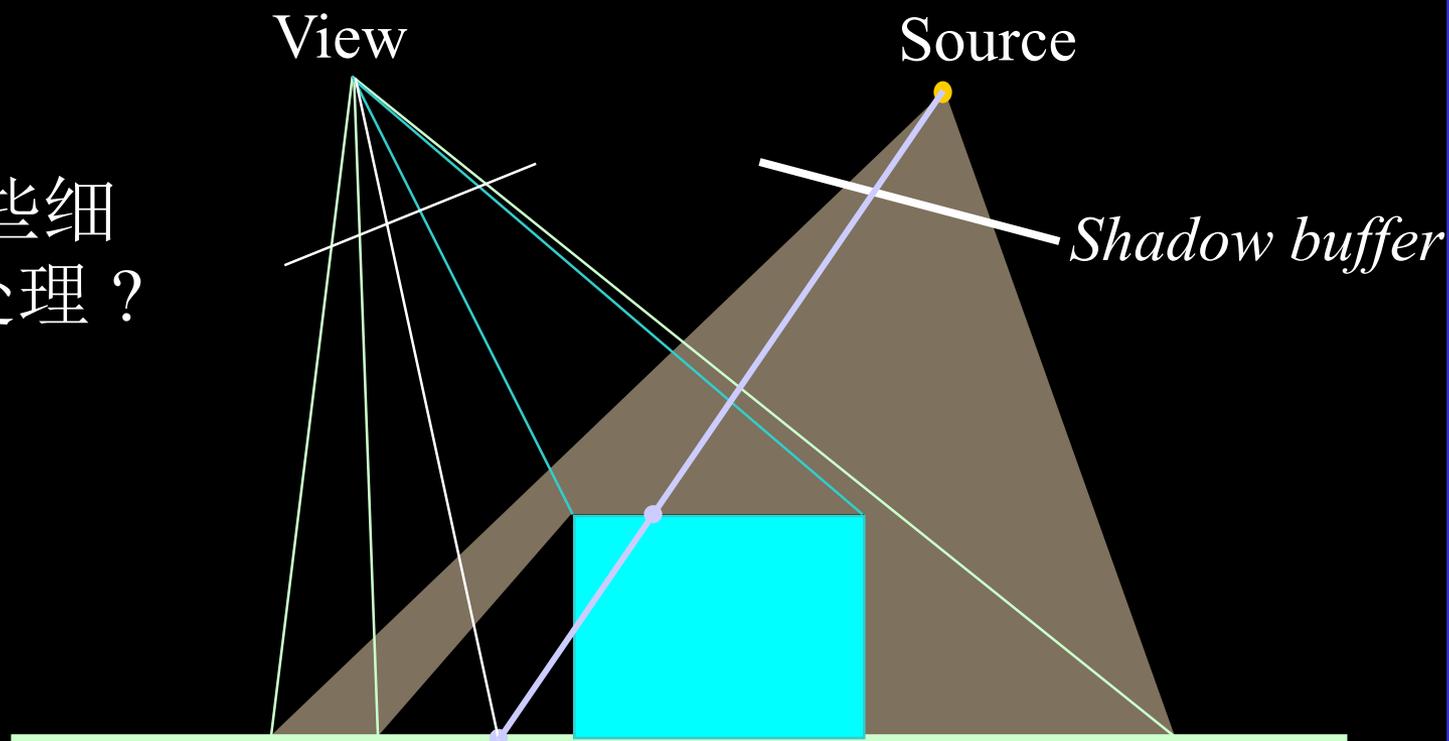
Shadow buffer

- 是最简单的阴影计算方法之一，可轻松与基于 z-buffer 的渲染器集成。
- 使用阴影是从光源观看时不可见的对象的那些部分的想法。
- 该算法有两个步骤：

Step1 将光源视为视点应用观察变换将所有点变换到光源坐标系。使用 z-buffer 算法渲染场景，但只有深度信息存储到Shadow-buffer 中。

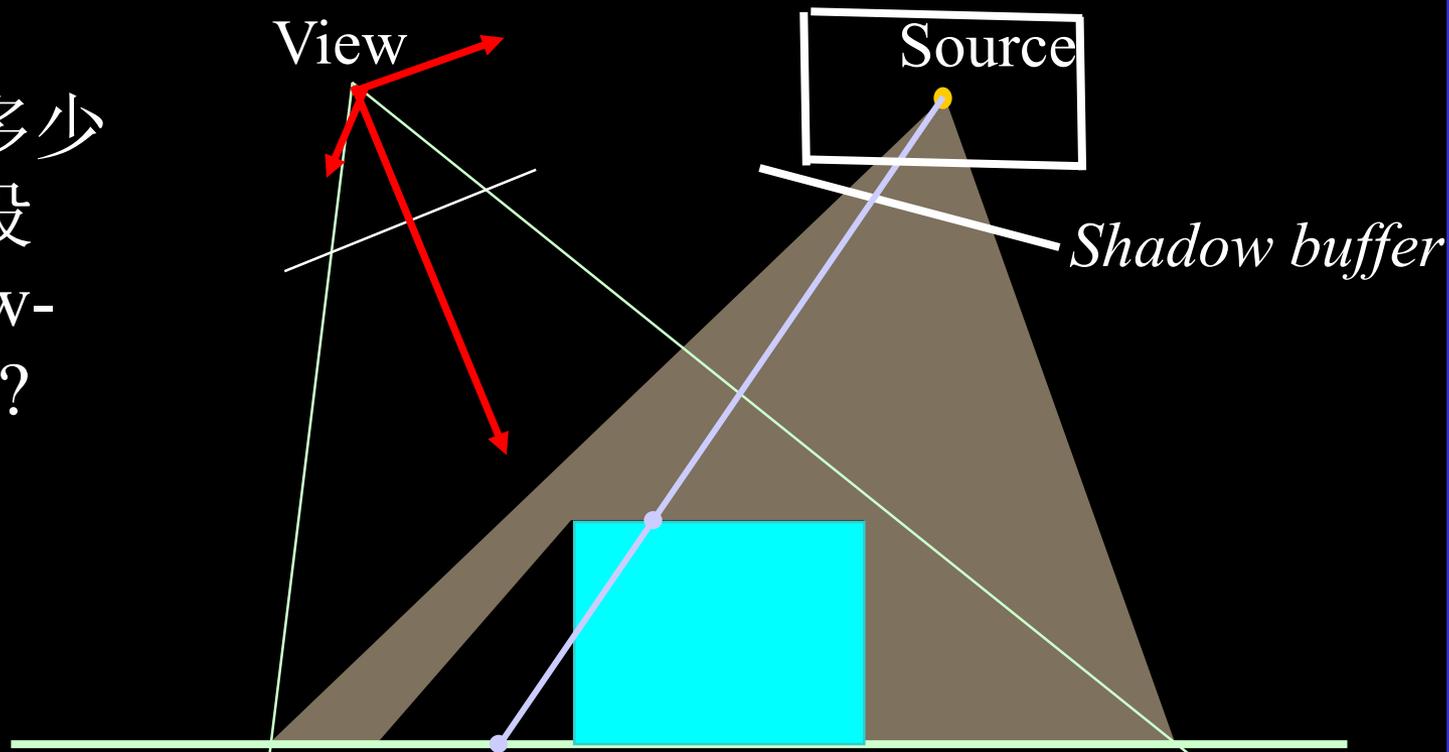
Step2 使用 z-buffer 算法从实际视点渲染场景，如果点 (x, y, z) 可见，将其变换到光源坐标系，设他在光源坐标系下的坐标为 (x_1, y_1, z_1) ，索引Shadow-buffer中 (x_1, y_1) 的深度值，将其与 z_1 进行比较，如果 z_1 大于该深度值，则 (x, y, z) 处于阴影中。

还有哪些细
节需要处理？



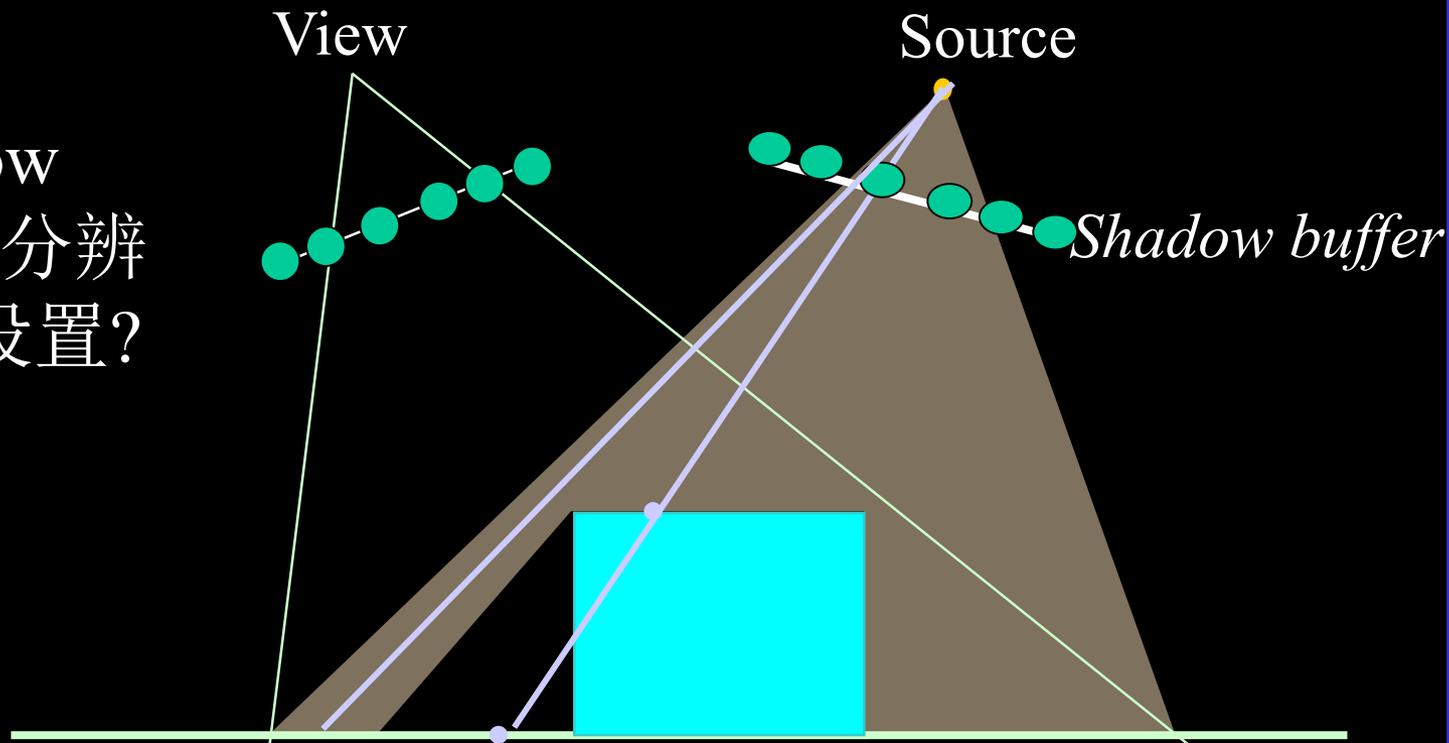
$$I = I_a k_a + \sum_{1 \leq i \leq m} I_{pi} [k_d (\bar{N} \cdot \bar{L}_i) + k_s (\bar{R}_i \cdot \bar{V})^n]$$

需要在多少
方向设
Shadow-
buffer?



$$I = I_a k_a + \sum_{1 \leq i \leq m} I_{pi} [k_d (\bar{N} \cdot \bar{L}_i) + k_s (\bar{R}_i \cdot \bar{V})^n]$$

Shadow
Buffer的分辨
率如何设置?



$$I = I_a k_a + \sum_{1 \leq i \leq m} I_{pi} [k_d (\bar{N} \cdot \bar{L}_i) + k_s (\bar{R}_i \cdot \bar{V})^n]$$

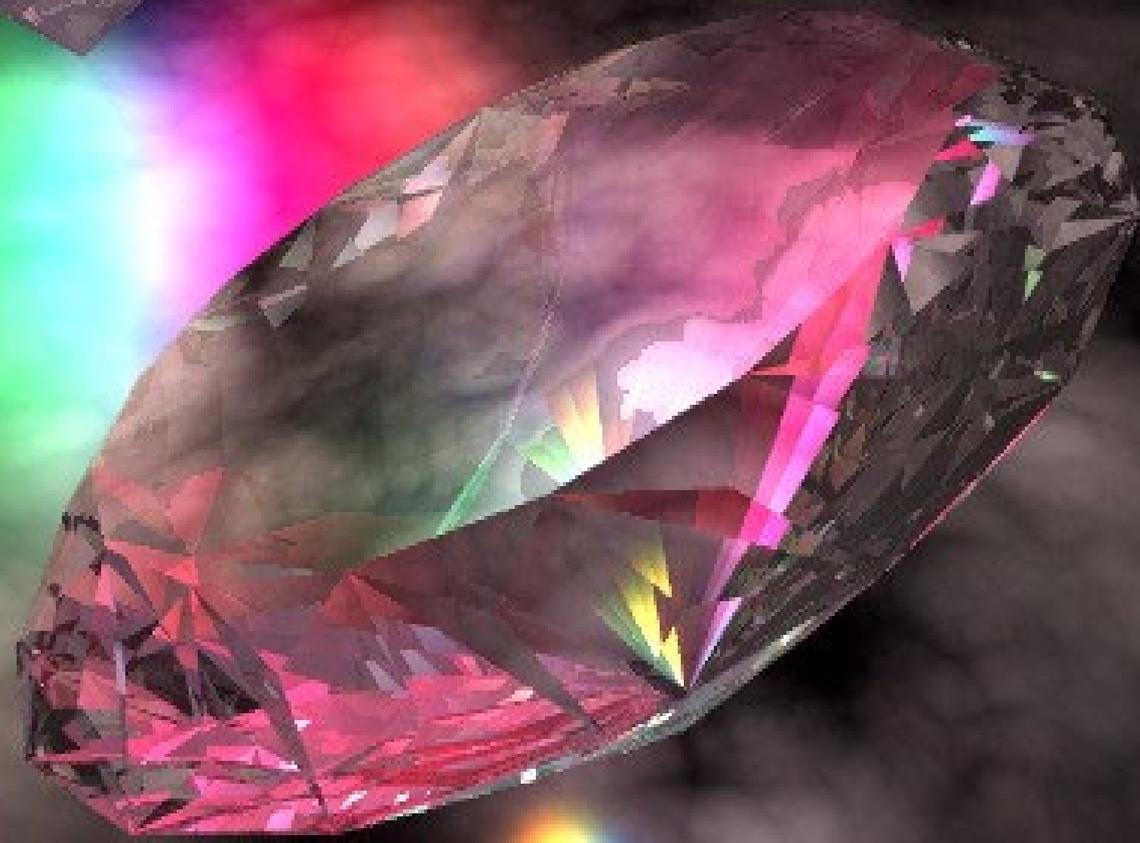
整体光照模型

目标: 计算物体表面的光强时不仅考虑光源的直接影响，还要考虑其他物体反射过来的光。

两类方法:

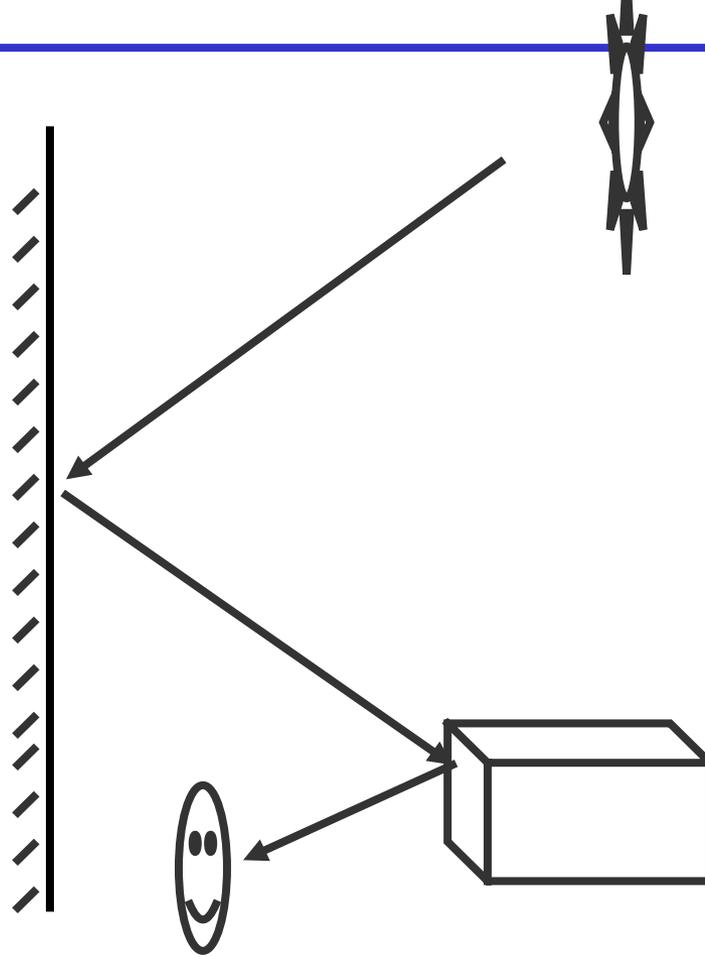
1. **光线跟踪:** 只考虑镜面反射
2. **辐射度:** 只考虑漫反射

光线跟踪

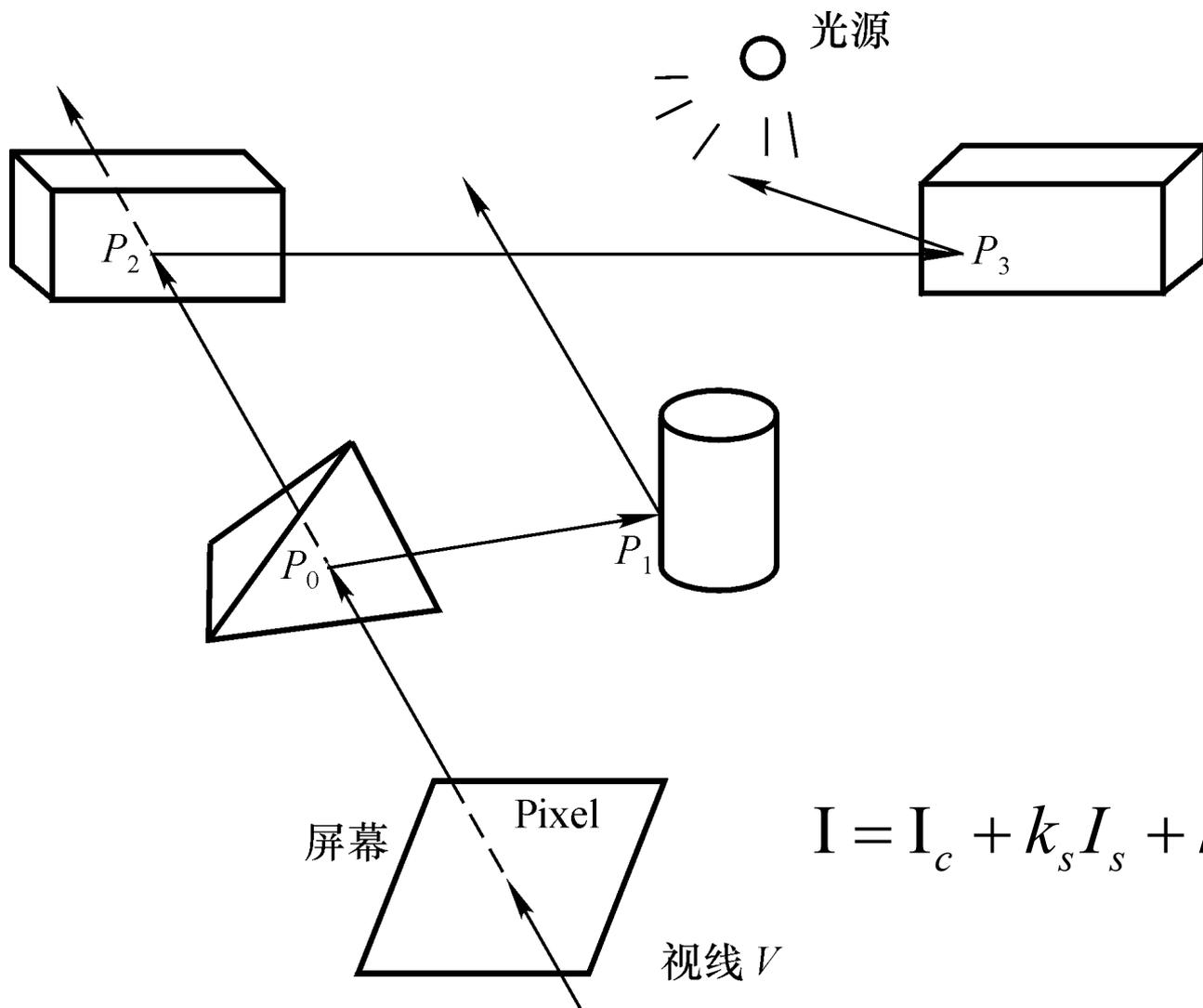


光线跟踪

- 光线跟踪是一种真实感显示技术。具有下述功能：
 - 隐藏面去除
 - 光源光照
 - 其他物体镜面反射引起的光照
 - 阴影



Witted 模型: $I = I_c + k_s I_s + k_t I_t$



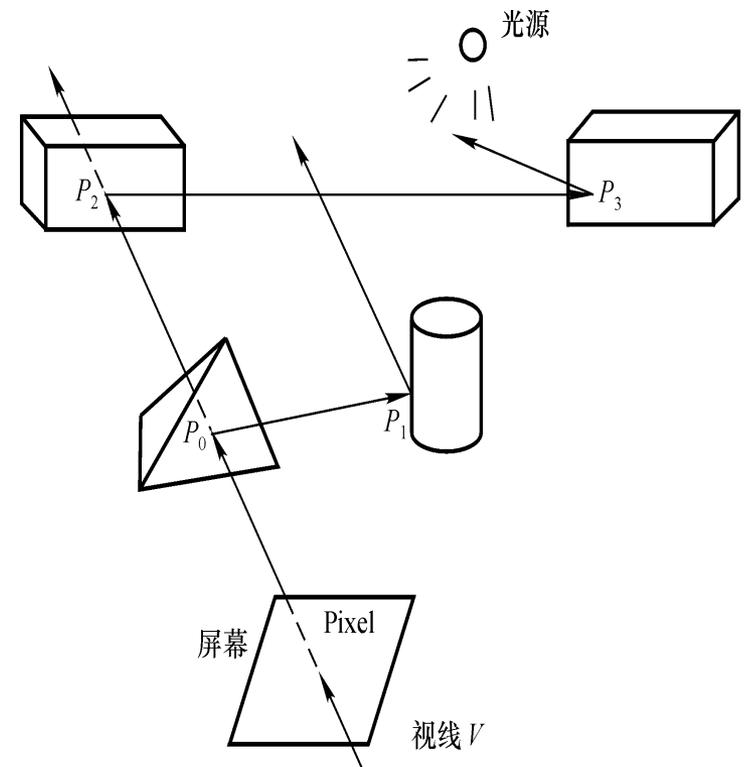
$$I = I_c + k_s I_s + k_t I_t$$

$$I = I_c + k_s I_s + k_t I_t$$

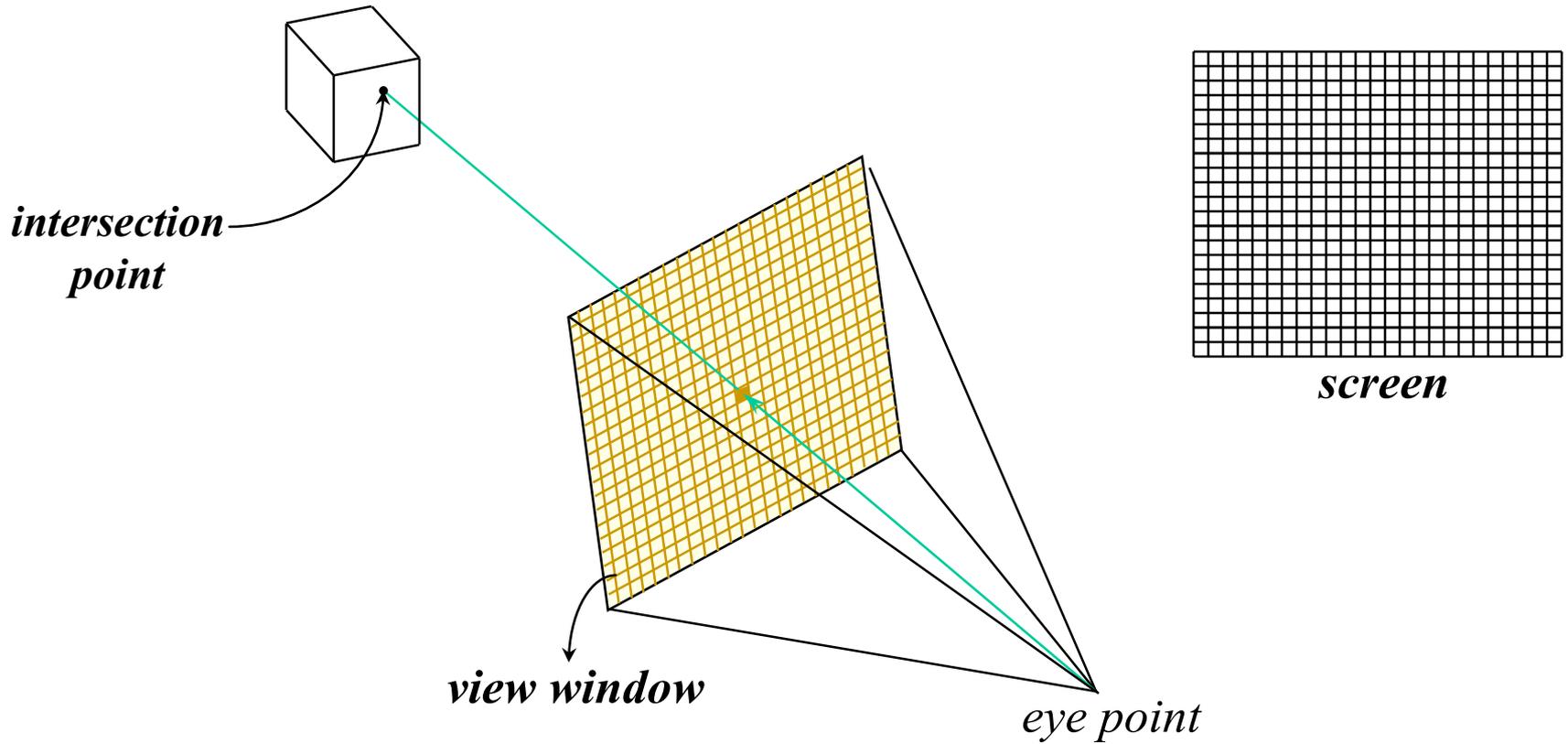
```

a_raytrace(Vector3D ray, int level, Color *I)
{Color Ilocal, Is, It; Int Is_inter; Vector3D vl, vr, p, Normal;
Inter_scene(ray, scene, &Is_inter, &p, &Normal, &vl, &vr, face);
If (Is_inter) {
    Calculate_Local_I(&Ilocal, face, p, Normal);
    If (Level < plvl) {
        a_raytrace(vl, int level+1, &Is);
        a_raytrace(vr, int level+1, &It);
        *I = Ilocal + face->ks*Is + face->kt*It;
    }
    else
        *I = Ilocal;
}
Else
    *I = Background;
}

```



光线跟踪





Credits: Mike Miller using Pov-Ray



Canon

AE-1
PROGRAM

No. 213987
Nikkor-XR7
1=28-80 mm
1:3.9/4.9

Credits: Mike Miller using Pov-Ray

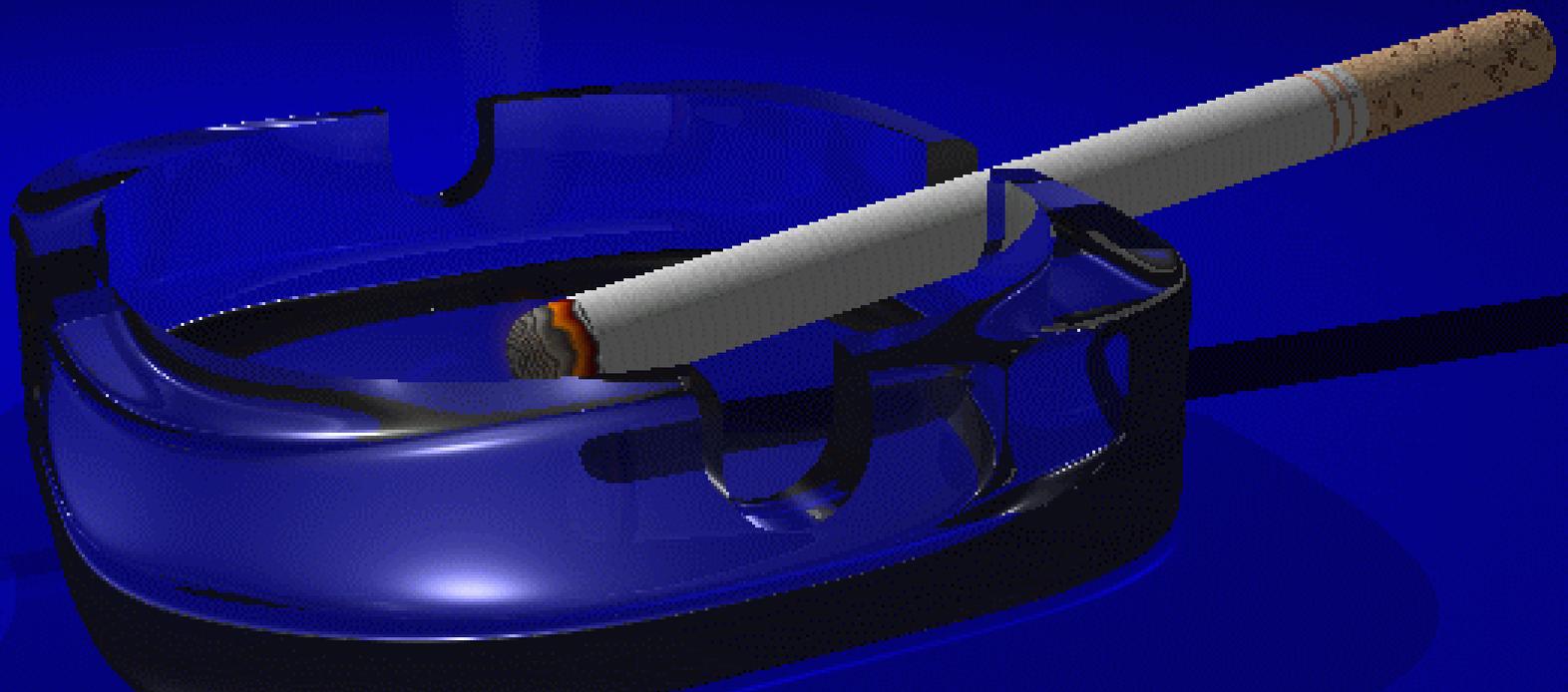
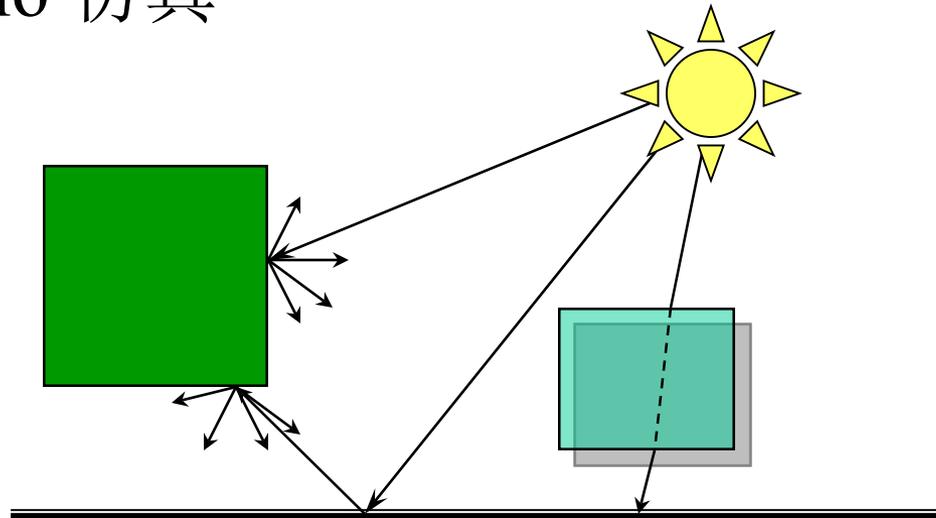


Image generated using Pov-Ray (Mike Miller)

粒子跟踪 Particle Tracing

- 整体光照模型
- 光的粒子性
- Monte Carlo 仿真



THANK YOU