

# 整体光照模型

## 一、 为什么需要整体光照模型？

- n 简单和局部光照模型不能很好地模拟光的折射、反射和阴影等，也不能用来表示物体间的相互光照影响。
- n 整体光照模型是更精确的光照模型，主要有光线跟踪和辐射度两种方法。

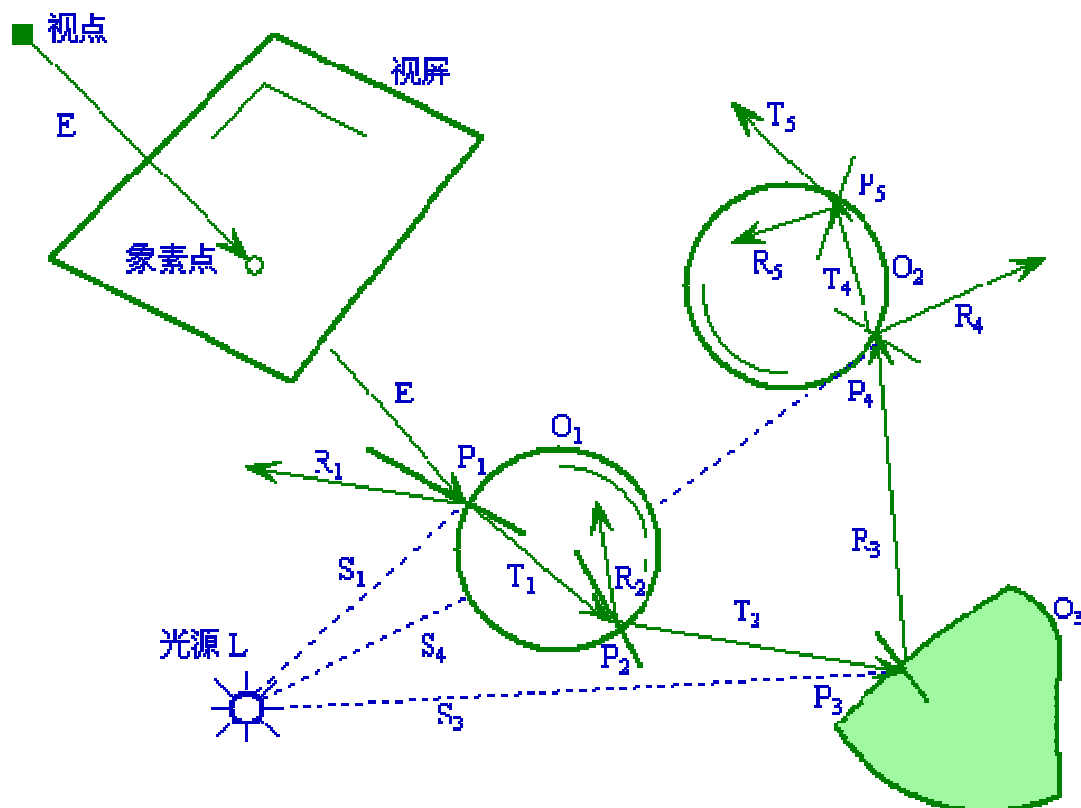
## 一、 为什么需要整体光照模型？

- n 简单和局部光照模型不能很好地模拟光的折射、反射和阴影等，也不能用来表示物体间的相互光照影响。
- n 整体光照模型是更精确的光照模型，主要有光线跟踪和辐射度两种方法。

## 二、光线跟踪基本原理 (Ray Tracing)

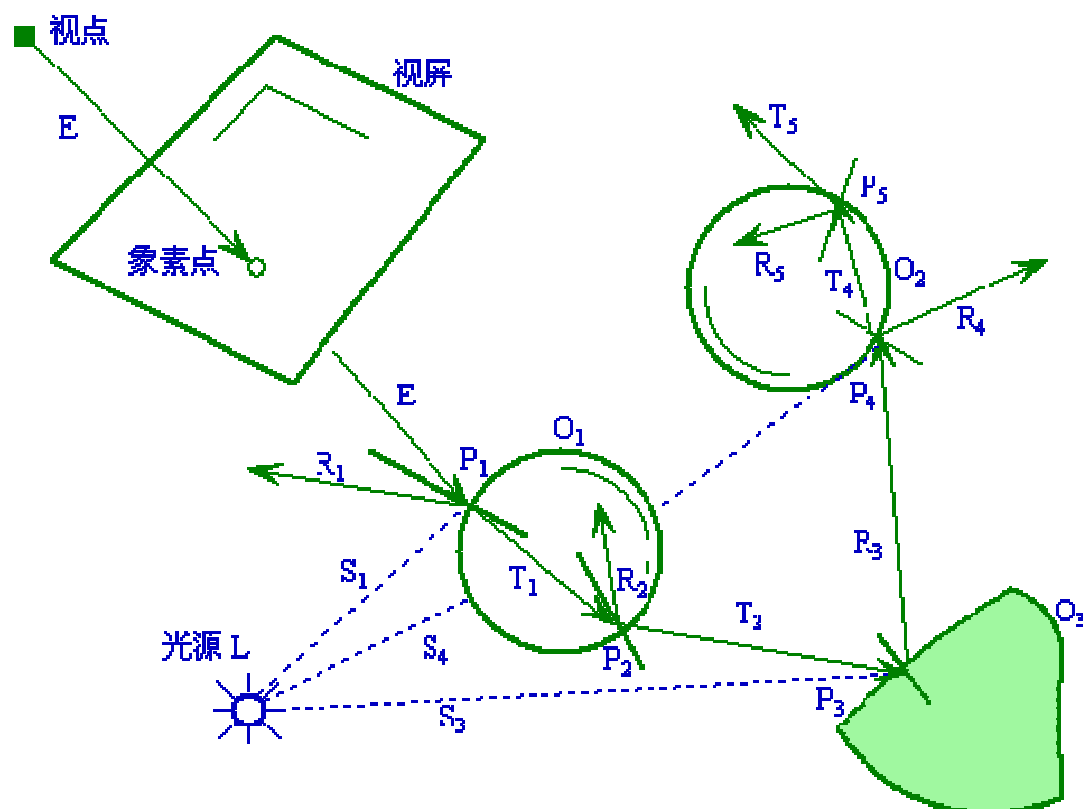
- n 光线跟踪算法是真实感图形学中的主要算法之一，该算法具有原理简单、实现方便和能够生成各种逼真的视觉效果等突出的优点，综合考虑了光的反射、折射、阴影等。

## 二、光线跟踪基本过程



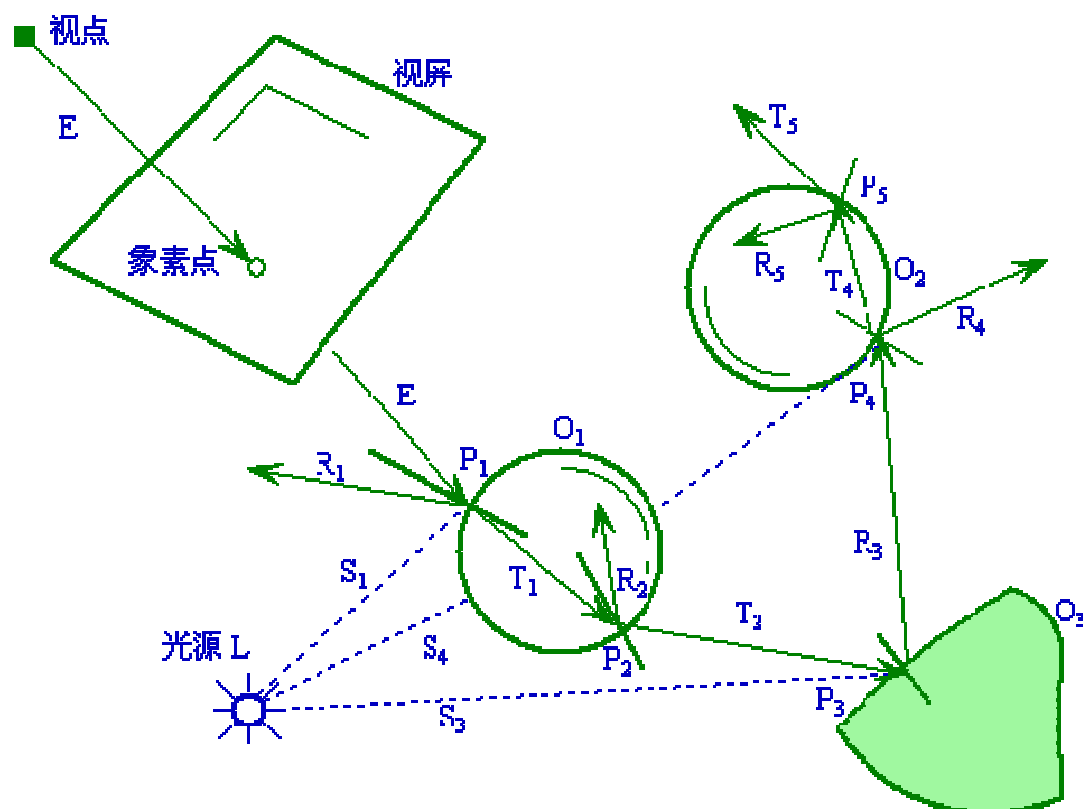
在这个场景中，  
有一个点光源 $L$ ，两  
个透明体 $0_1$ 与 $0_2$ ，  
一个不透明体 $0_3$ 。

## 二、光线跟踪基本过程



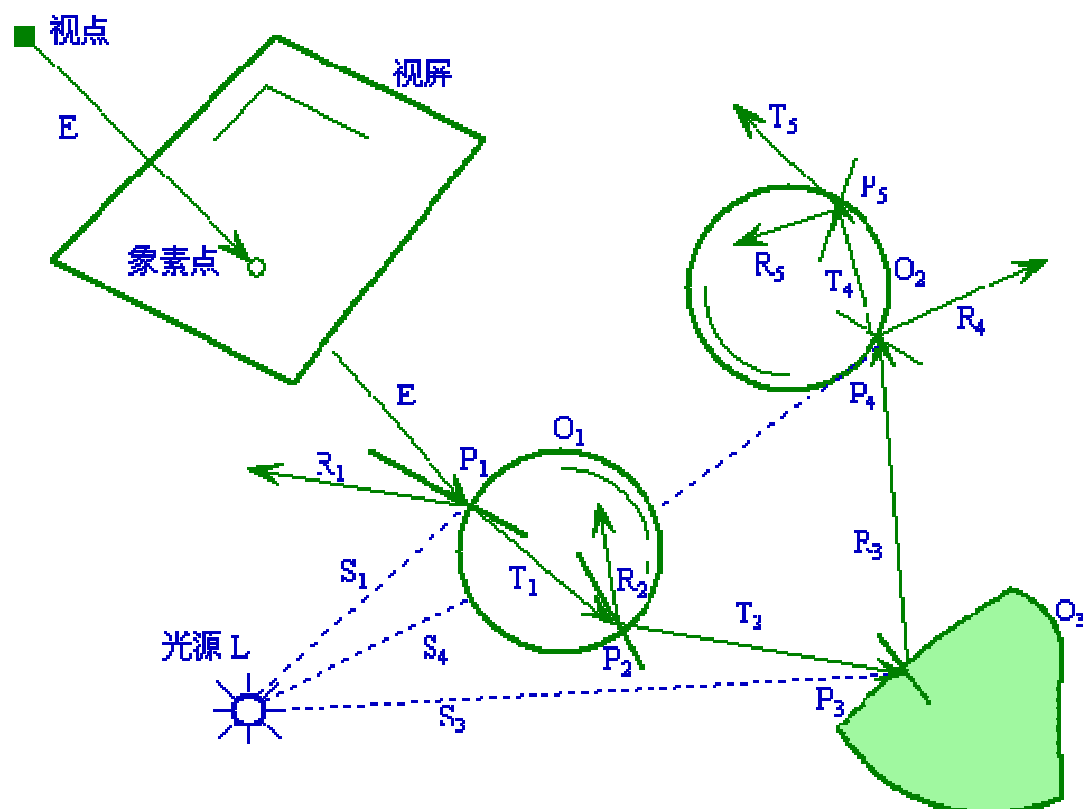
首先，从视点出发经过视屏一个像素点的视线 $E$ 传播到达球体 $O_1$ ，交点为 $P_1$ 。从 $P_1$ 向光源 $L$ 作一条阴影测试线 $S_1$ ，可以发现其间没有遮挡的物体，那么就用局部光照模型计算光源对 $P_1$ 在其视线 $E$ 方向上的光强，作为该点的局部光强；

## 二、光线跟踪基本过程



在反射光线 $R_1$ 方向上，没有再与其他物体相交，那么就设该方向的光强为0，并结束这条光线方向的跟踪。然后对折射光线 $T_1$ 方向进行跟踪，计算该光线的光强贡献。

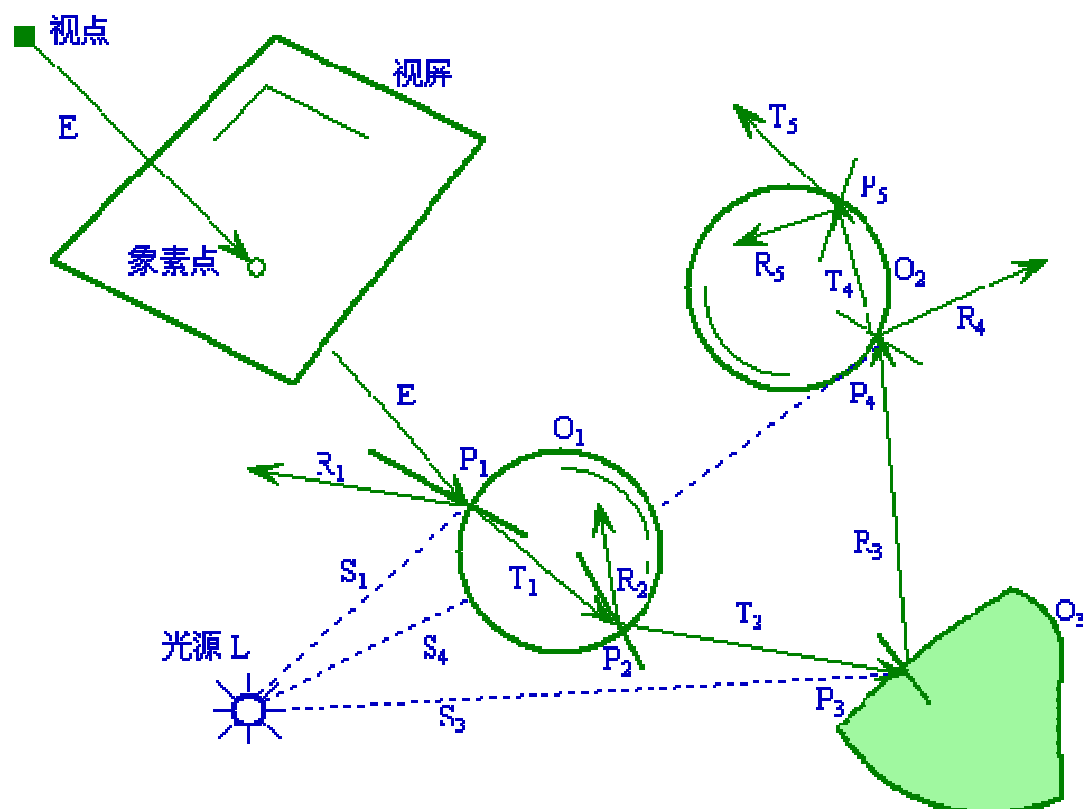
## 二、光线跟踪基本过程



折射光线 $T_1$ 在物体 $O_1$ 内部传播，与 $O_1$ 相交于点 $p_2$ ，由于该点在物体内部，假设它的局部光强为0。该点处同时产生了反射光线 $R_2$ 和折射光线 $T_2$ ，在反射光线 $R_2$ 方向，可以继续递归跟踪下去计算它的光强。而对折射光线 $T_2$ 则继续进行跟踪。

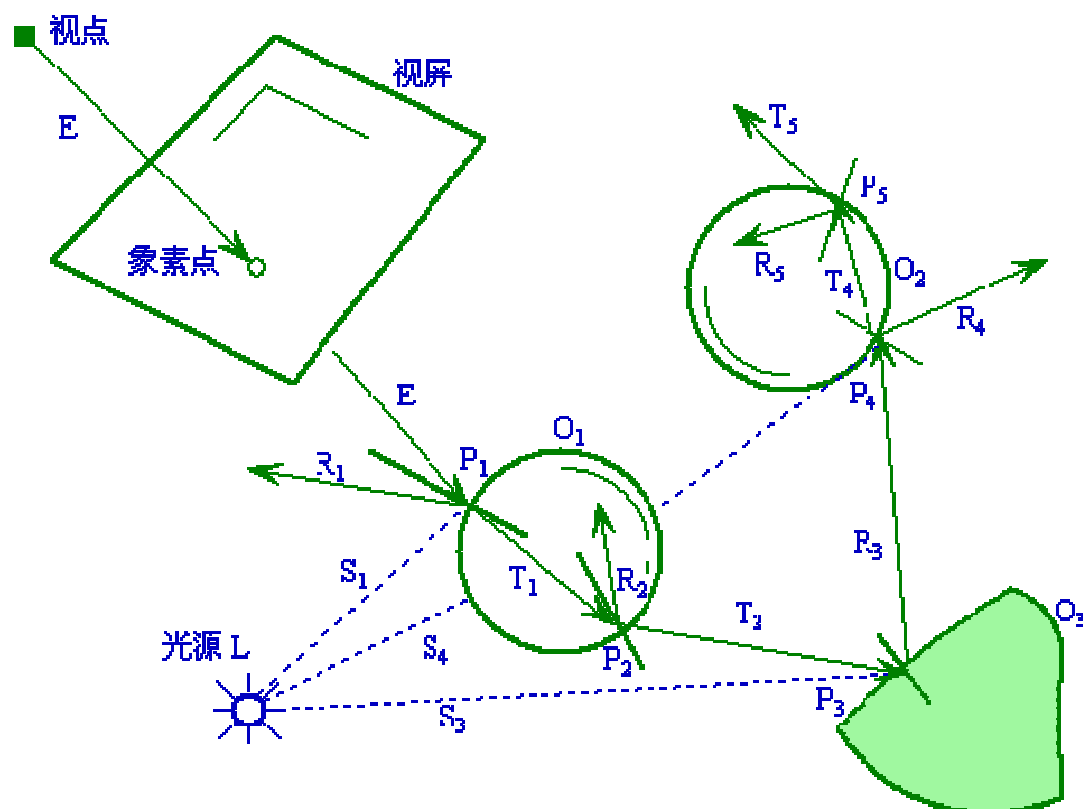


## 二、光线跟踪基本过程



$T_2$ 与物体 $O_3$ 交于点 $p_3$ ，作 $P_3$ 与光源 $L$ 的阴影测试线 $S_3$ ，没有物体遮挡，正常计算该处的局部光强。由于该物体是非透明的，可以继续跟踪反射光线 $R_3$ 方向的光强，结合局部光强得到 $P_3$ 处的光强。

## 二、光线跟踪基本过程



反射光线 $R_3$ 的跟踪与前面的过程类似，算法可以递归地进行下去。重复上面的过程，直到光线满足跟踪终止条件。这样最终可以得到视屏上一个像素点的光强，也就是它相应的颜色值。

# 光线跟踪 (Ray Tracing)

## 一、光线跟踪怎么停止？

n 在算法应用的意义上，可以有以下几种终止条件。

- u 该光线未碰到任何物体
- u 该光线碰到了背景
- u 光线在经过许多次反射和折射以后，就会产生衰减，光线对于视点的光强贡献很小
- u 光线反射或折射次数即跟踪深度大于一定值

## 二、光线跟踪伪代码

光线跟踪算法的函数名为RayTracing（），光线的起点为start，方向为direction，光线的衰减权值为weight，初始值为1，算法最后返回光线方向上的颜色值color。

对于每一个像素点，第一次调用RayTracing（），可以设起点start为视点，而direction为视点到该像素点的射线方向。

```

RayTracing (start,, direction, weight, color)
{
    if (weight<MinWeight)
        color = black
    else
    {
        计算光线与所有物体的交点中离start最近的点;
        if (没有交点)
            color = black
        else
        {
             $I_{local}$  = 在交点处用局部光照模型计算出的光强;
            计算反射方向R;
            RayTracing (最近的交点, R, weight*Wr, Ir)
            计算折射方向T;
            RayTracing (最近的交点, T, weight*Wt, It) ;
            color =  $I_{local}$  + KrIr + KtIt;
        }
    }
}

```

### 三、光线跟踪缺点

- n 光线跟踪方法由于要进行大量的求交运算，且每一条射线都要和所有的物体求交，因此效率很低，需要耗费大量的计算时间。
- n 光线跟踪方法可以进行加速。

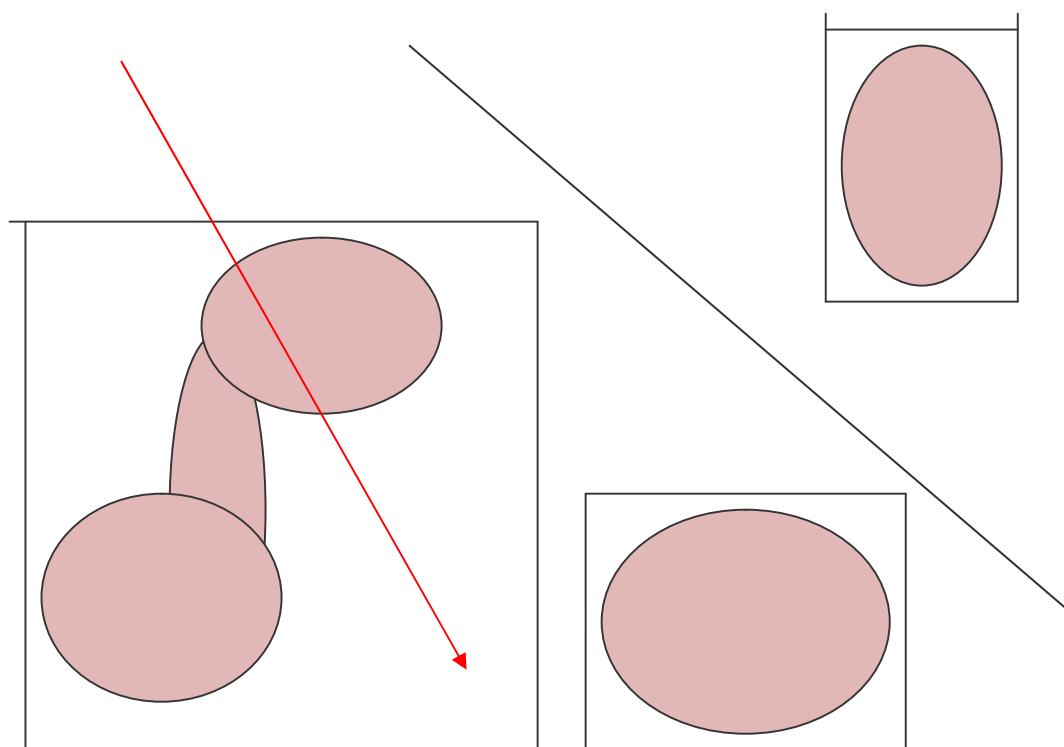
## 四、光线跟踪加速

- ü 提高求交速度：针对性的几何算法、...
- ü 减少求交次数：包围盒、空间索引、...
- ü 减少光线条数：颜色插值、自适应控制、...
- ü 采用广义光线和采用并行算法等



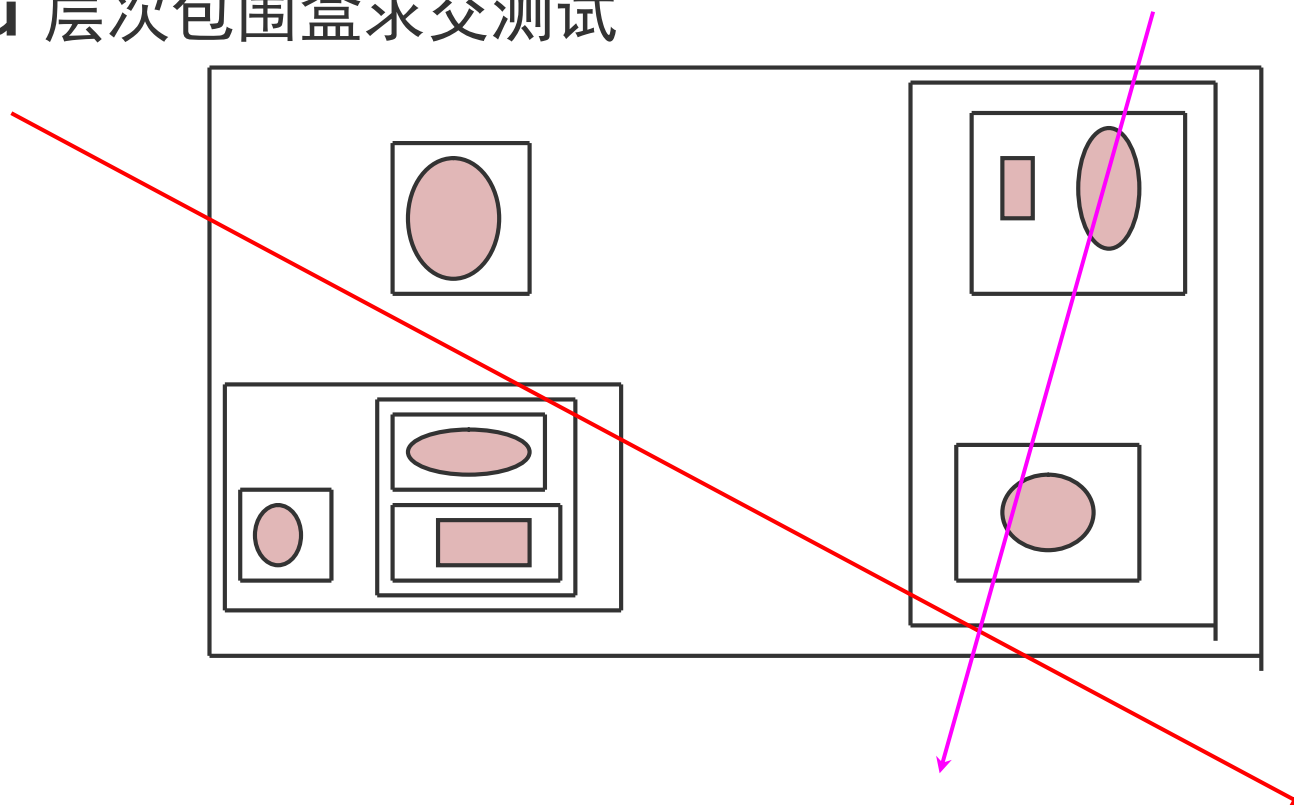
## 四、光线跟踪加速

### ü 包围盒求交测试



## 四、光线跟踪加速

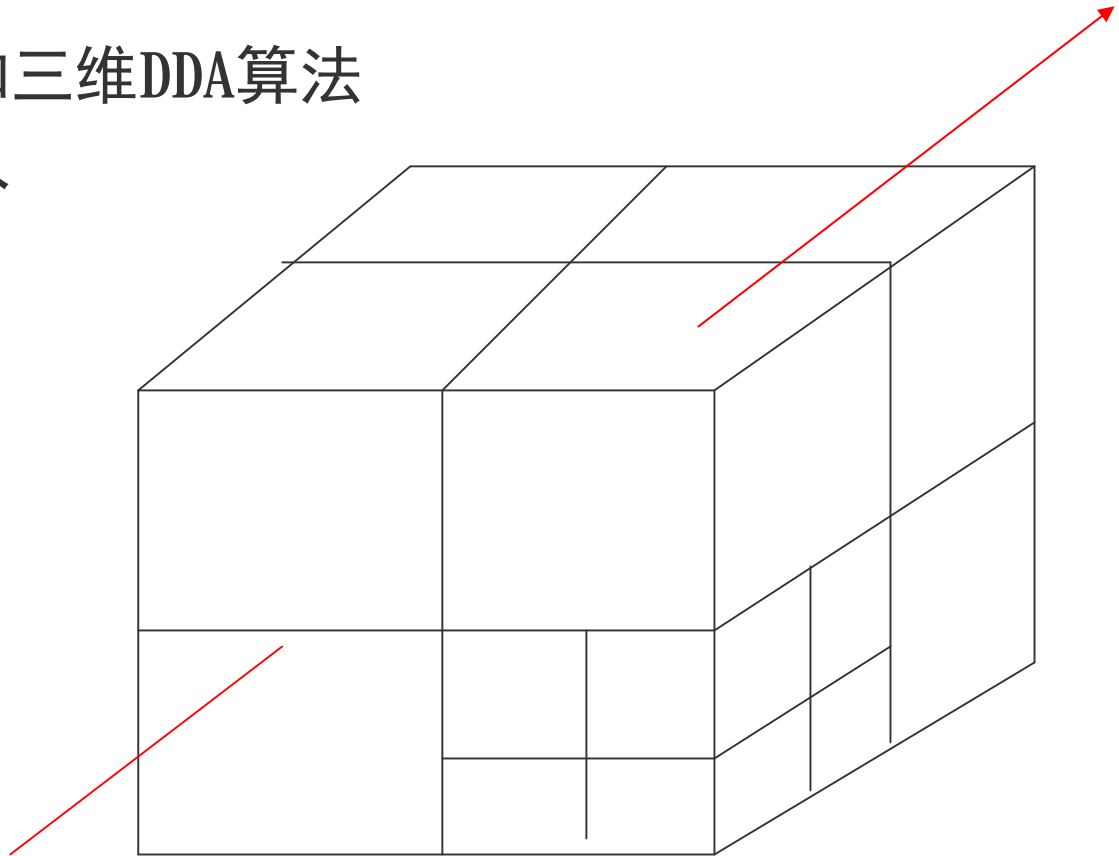
### ü 层次包围盒求交测试



## 四、光线跟踪加速

ü 空间网格剖分和三维DDA算法

ü 空间八叉树剖分



## 五、光线跟踪场景渲染

