

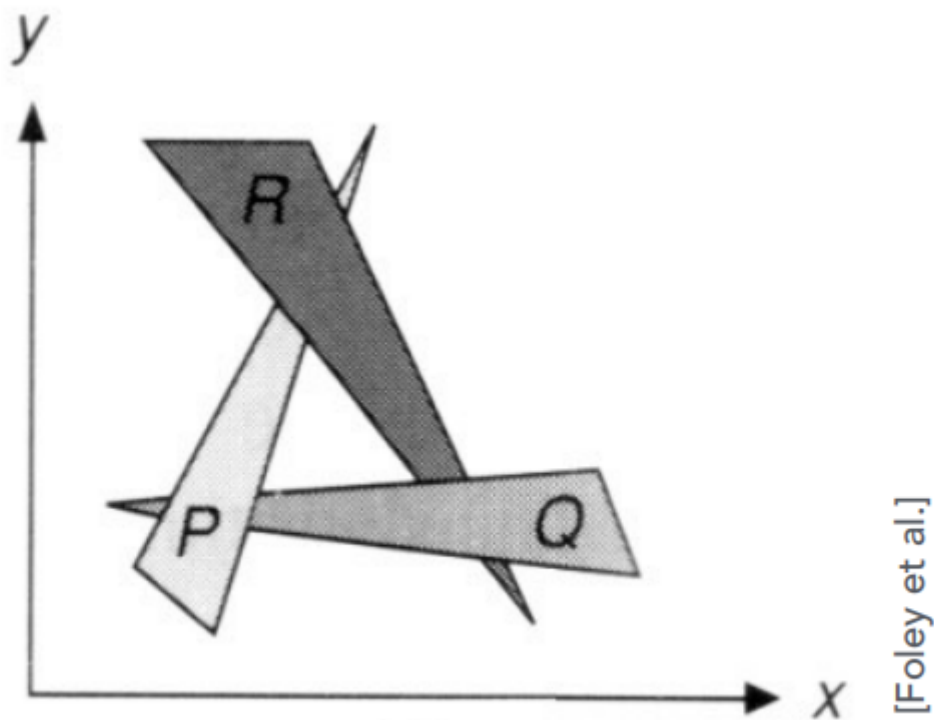
Lecture 07

1. Visibility / occlusion 可视性/遮挡

在绘制从特定位置的视图时，需要正确处理不同远近位置（即 z 轴上的位置不同）的关系。以下有两种算法。

1. 画家算法

- 由油画画家在绘画过程中由远及近的绘画过程而来的
- 按物体深度（通俗的说是离摄像机的远近）来从远到近的绘制
- **问题**：需要有物体深度做参考，而对某些相互遮挡的物体无法得出深度关系，比如下图
-



2. Z-Buffer

- 按像素为单位获取深度，对每个像素操作时，遍历所有包含该像素的三角形，并只记录其中最浅的深度（因为所有深度更大的三角形会被遮挡）。
- Z-Buffer 的伪代码
 - 其中需要两个数组 `framebuffer[]` 存颜色，`depthbuffer[]` 存深度

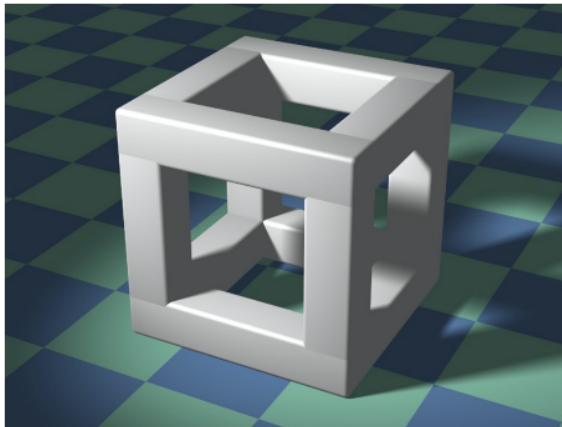
- z 值永远为正，越小代表越浅

```

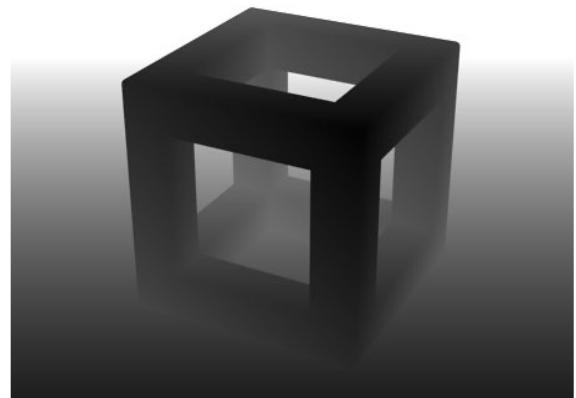
for (each triangle T)
    for(each sample (x,y,z) in T)
        if( $z < zbuffer[x,y]$ ) //更浅的深度
            framebuffer[x,y] = rgb; //更新颜色
            zbuffer[x,y] = z; //更新深度
        else
            //do nothing

```

- Z-Buffer 渲染图与深度图



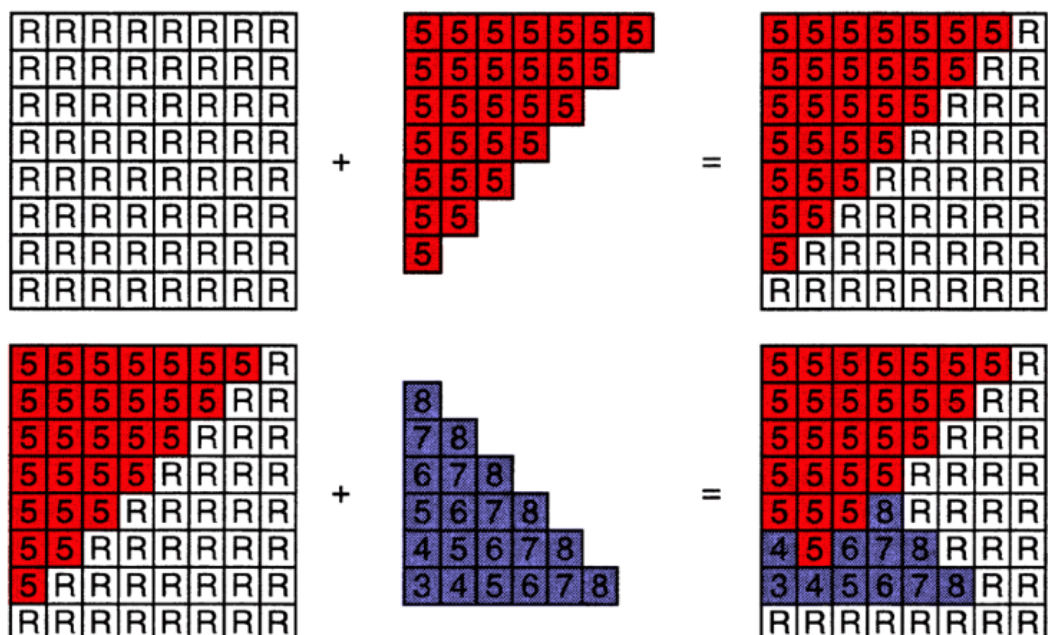
Rendering



Depth / Z buffer

Image source: Dominic Alves, flickr.

- Z-Buffer 算法示例



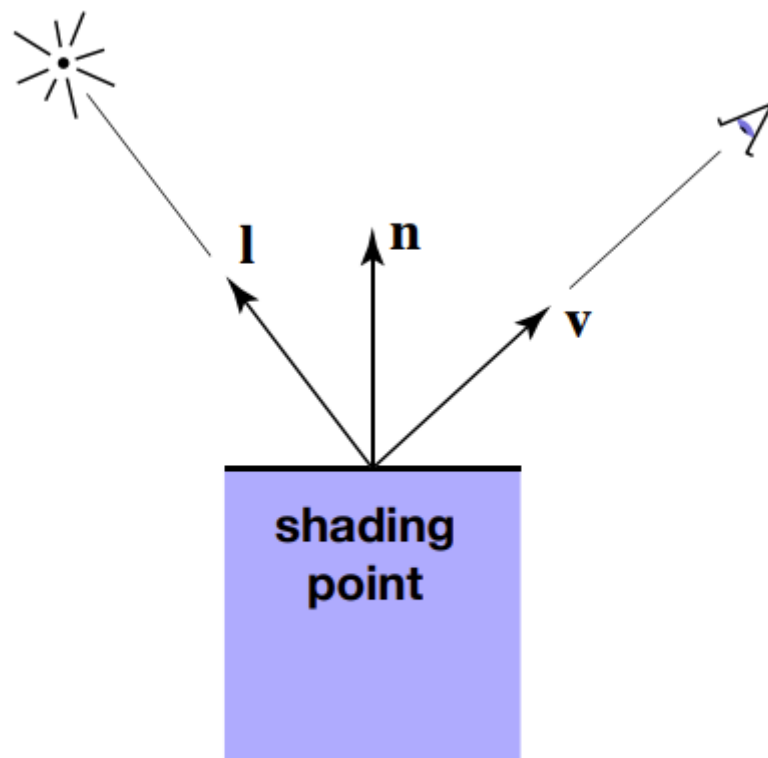
2.Shading 着色

1.Blinn-Phong Reflection Model

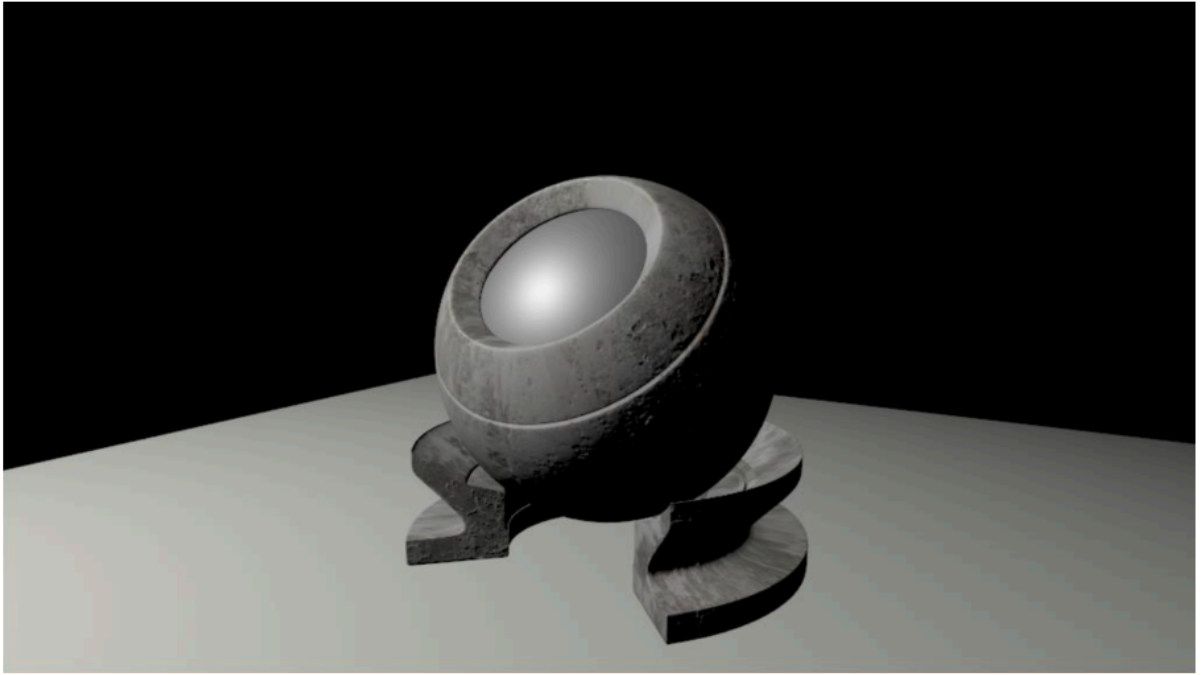
- Blinn-Phong\ Reflection\ Model 为一种着色模型
- 光亮照亮分为三部分
 - Specular highlights 高光/镜面光
 - Diffuse reflection 漫反射
 - Ambient lighting 环境光照，即光源无法直接照到的地方，需要经过光线的反射

2.Shading is Local 局部着色

- 局部着色，即对每个点做着色
- 计算某个点的着色需要 I （光线方向）， n （表面点法线）， v （观察方向），以及表面特征（color 物体本来的颜色，shininess 物体是光滑或粗糙）
-



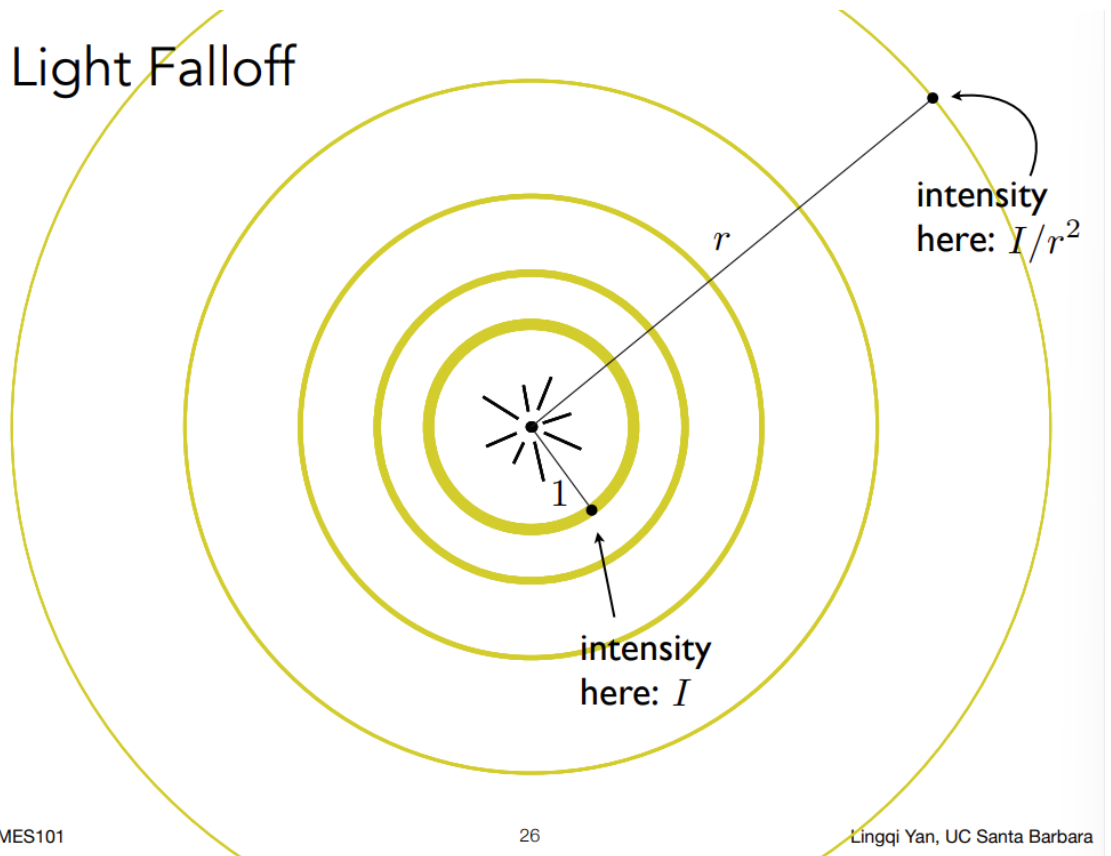
- 着色≠阴影，即着色过程中 **不会生成阴影**。



3.漫反射

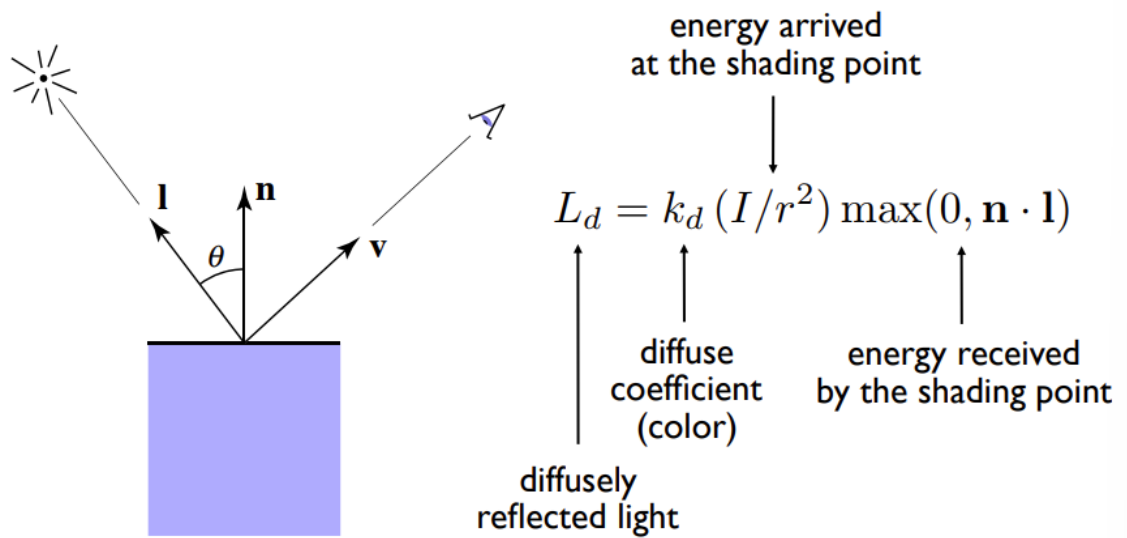
- 当光线传播到物体表面后，其产生的光是向四周均匀出射的，也即 **表面颜色不随观察角度而变化**
- Light Falloff
 - 强度：定义离点光源距离为1处的强度为 I
 - 而将传播过程视作一个球壳， I 反比于 r （传播距离）
 -

Light Falloff

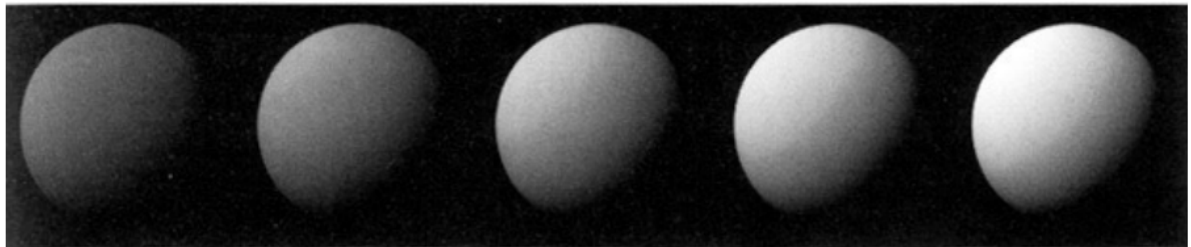


- Lambert's cosine law, 光线与物体表面的夹角决定了物体表面吸收能量(光线)的多少。

•



- $$L_d = k_d (I/r^2) \max(0, \mathbf{n} \cdot \mathbf{l})$$
- k_d 漫反射系数, I/r^2 光强, $\max(0, \mathbf{n} \cdot \mathbf{l})$ 设置从物体背面射过来的光线对物体表面影响为0
- 随 k_d 的增大, 物体接收到的 L_d 逐渐变大



$k_d \longrightarrow$