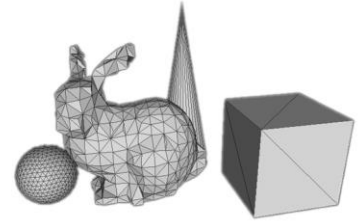


Introduction to Computer Graphics



Shading 1

第五章 视图(Viewing)

本章节主要解决相机的设置问题

- 视图概念(了解)
 - 平行投影与透视投影的区别
 - 各自有何优缺点
 - 特殊的平行投影：正交投影
 - 三视图，常用于CAD领域
 - 透视投影的分类
 - 灭点：三灭点、两个灭点、一个灭点

第五章 视图(Viewing)

- OpenGL中的视图设置（掌握）
 - 描述相机方位的函数、矩阵
 - 能够实现gluLookAt
 - 当up向量不平行于投影平面时应该怎么办？
 - 描述相机投影的函数、矩阵
 - 能够实现glOrtho [要掌握规范化矩阵的推导过程]
 - 能够实现glFrustum
 - 能够实现gluPerspective

第五章 视图(Viewing)

- 回顾、总结图形流水线，尤其是顶点变换（掌握）
 - 输入点的坐标，能够自行算出此点在窗口中的位置
 - 经过模型视图变换->投影变换->裁剪(暂时不用) ->视口变换（要自己实现此过程！）
 - 介绍了OpenGL的gluProject函数
 - 对于每个顶点，除了计算它的位置之外，我们还要计算它的颜色，这就是本节课要介绍的shading技术（明暗着色）

第六章 明暗着色(Shading)

本章主要介绍光源作用在物体上呈现出不同明暗效果的物理过程。

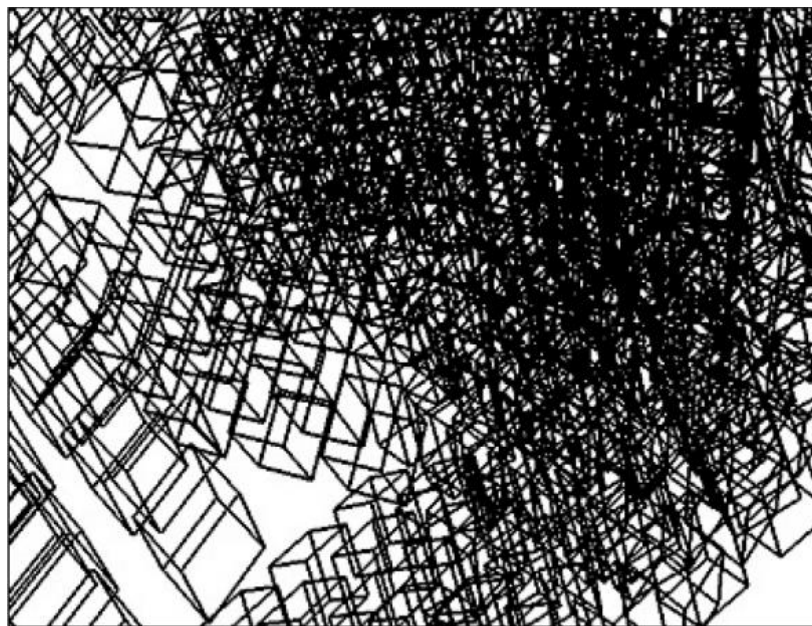
- 基本概念
- Phong光照明模型
- 多边形明暗处理
- OpenGL明暗处理

第一部分 基本概念

- 主要内容
 - 为什么需要明暗着色处理
 - 光源类型
 - 点光源
 - 聚光灯
 - 远距离光源
 - 环境光
 - 表面类型
 - 镜面
 - 漫反射面
 - 透明面

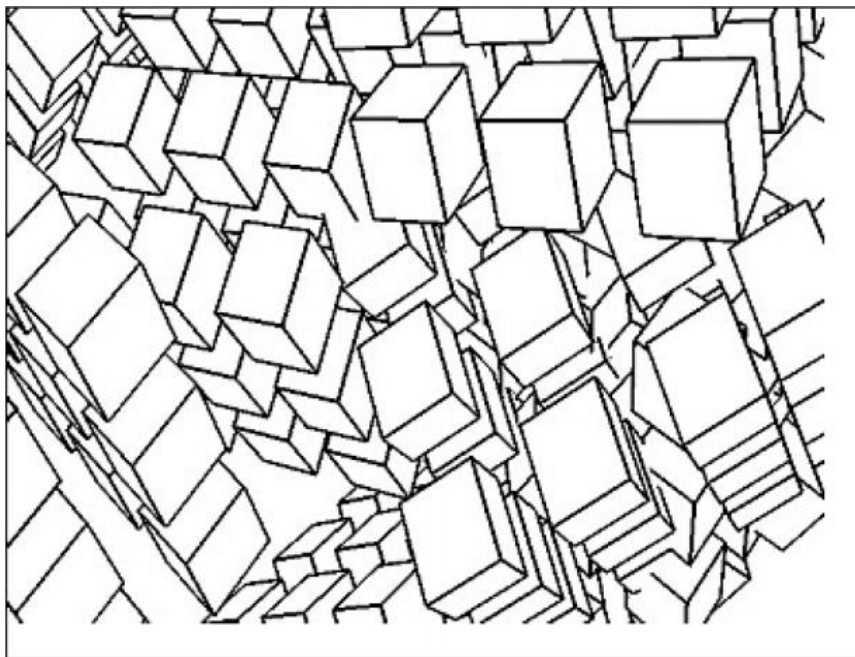
多边形网格模型

- 在计算机图形学中的模型通常是由多边形网格构成的
- 右图以线框形式显示一个场景，由540个立方体组成



隐藏面消除

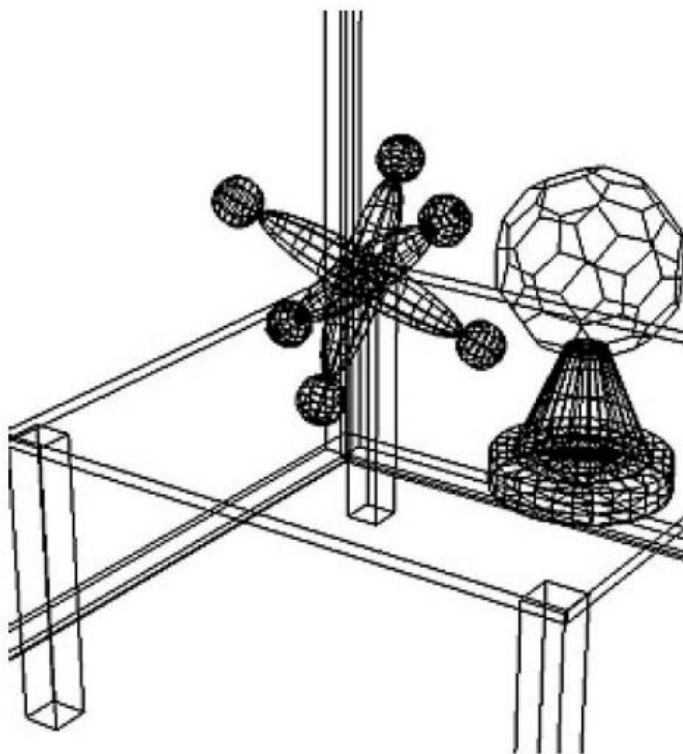
- 使多边形网格显得真实的第一种方法是隐藏面消除
- 右图显示的是前一页场景经隐藏面消除后的结果
- 但结果仍显得平面



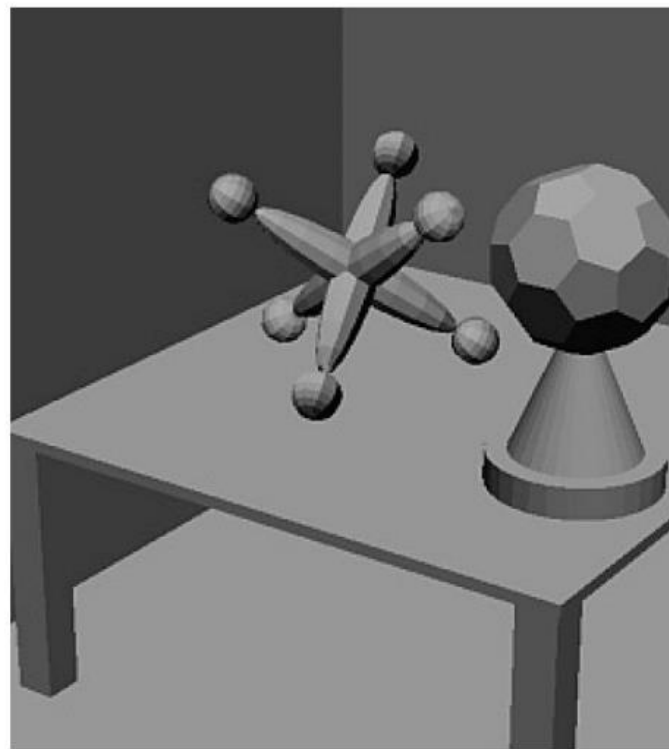
模拟光照

- 在隐藏面消除后，为了使对象看起来更真实，应当模拟光照在物体的状态，即应当通过计算确定表示对象的像素的适当状态
- 在这种计算中应充分根据对象表面的状态，光源的位置以及视点的位置
- 需要计算每帧图像中各个像素的颜色亮度，而不是由用户直接指定

线框模型 VS 简单明暗处理



线框模型

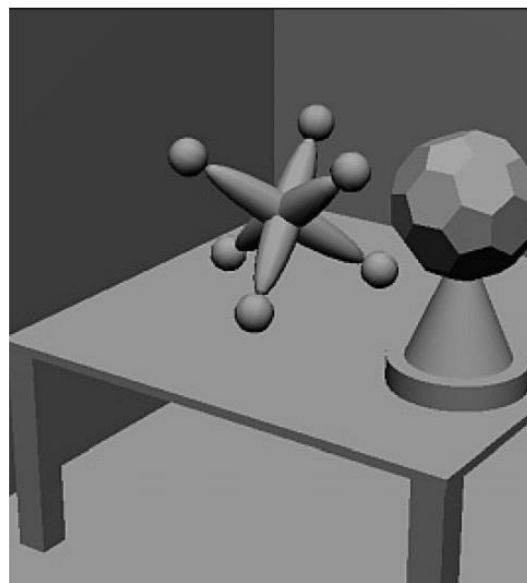
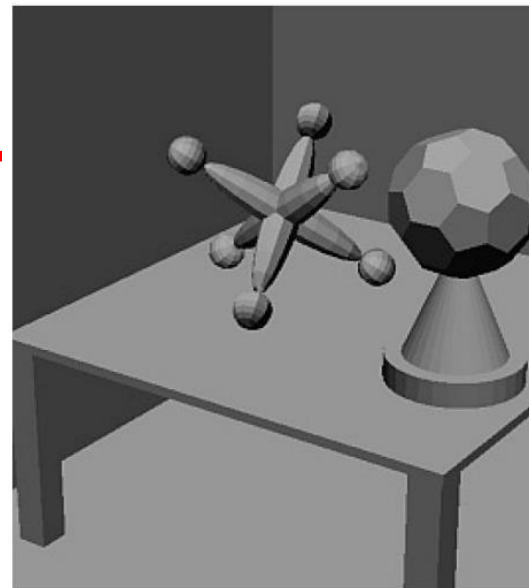


平面
明暗处理

经过了简单光照处理
和隐藏面消除

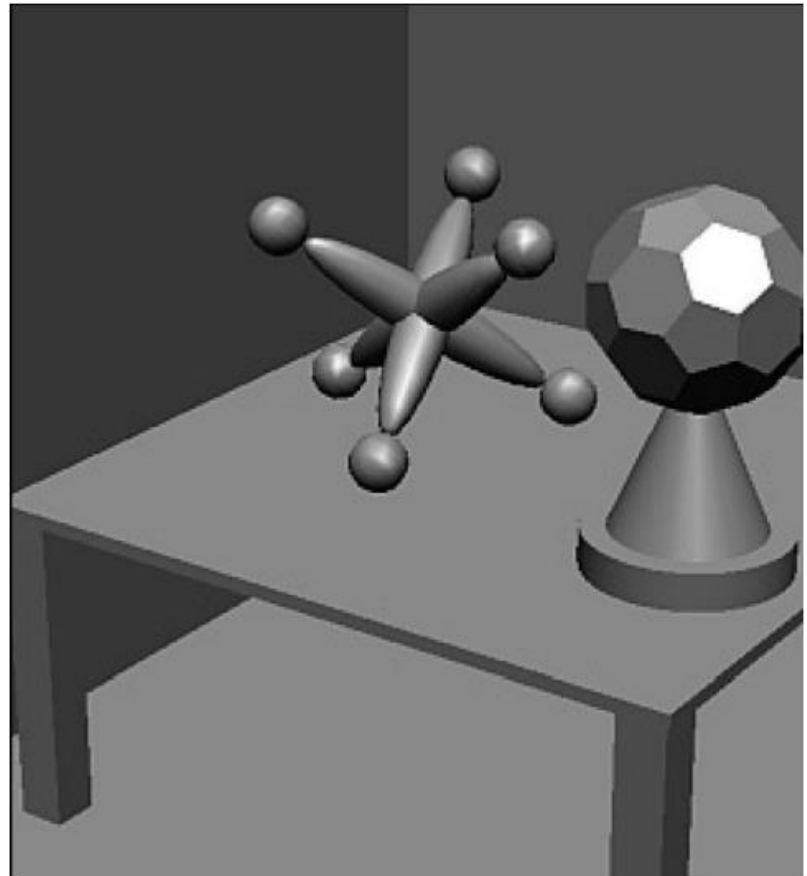
光滑明暗处理

- 如果多边形网格表示的对象为光滑对象，那么显示的结果应当反映这种光滑性
- 在计算了每个顶点处的亮度后，应用线性插值计算出内部的亮度--称为Gouraud明暗处理算法



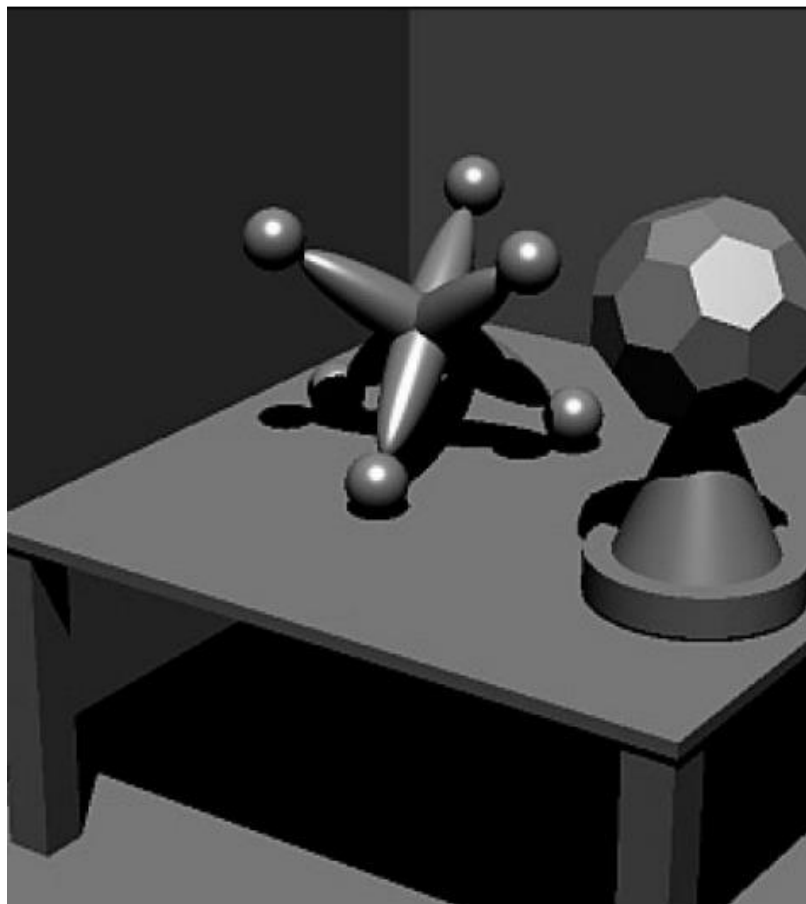
镜面光

- 为了得到更真实的效果，可以加入镜面光的效果



阴影

- 在加入阴影后，可以进一步提高图像的真实感



纹理映射

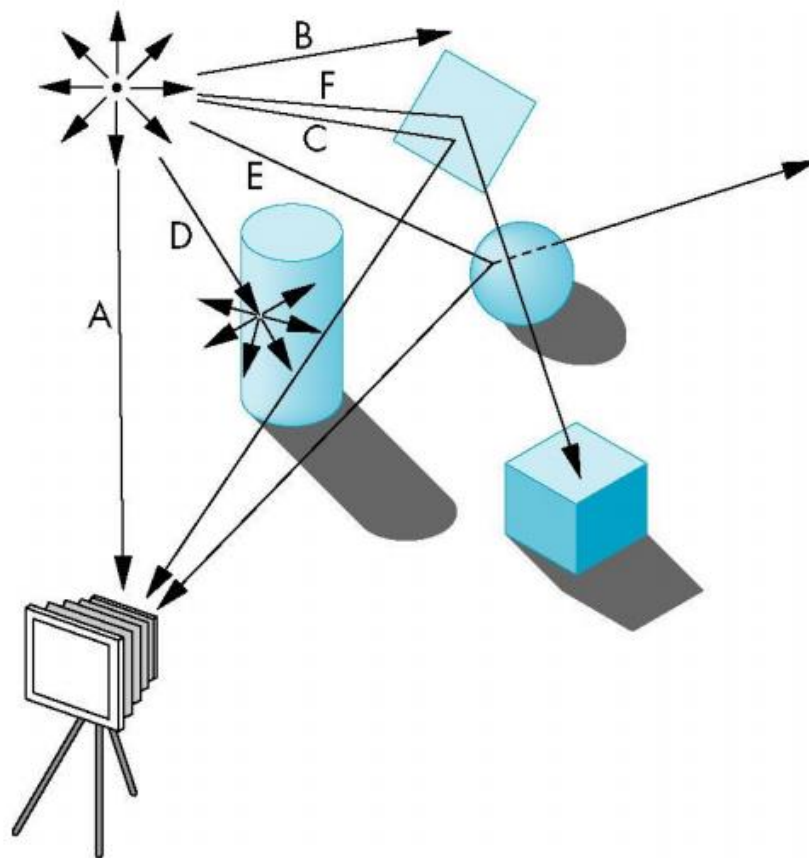
- 在对象表面上加上纹理效果，会使图像的真实感增强一大步



其他增加真实感的绘制方法

- 光线跟踪

- 计算复杂
- 容易实现，生成的图形中正确地反映阴影、镜面反射以及透明的效果



- 辐射度方法

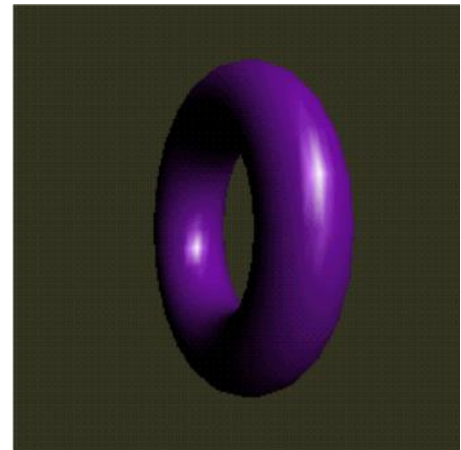
- 基于能量守恒

为什么需要明暗处理

- 假设用多边形网格建立了球面的模型，其颜色采用glColor定义，那么得到的结果为



- 而我们希望为

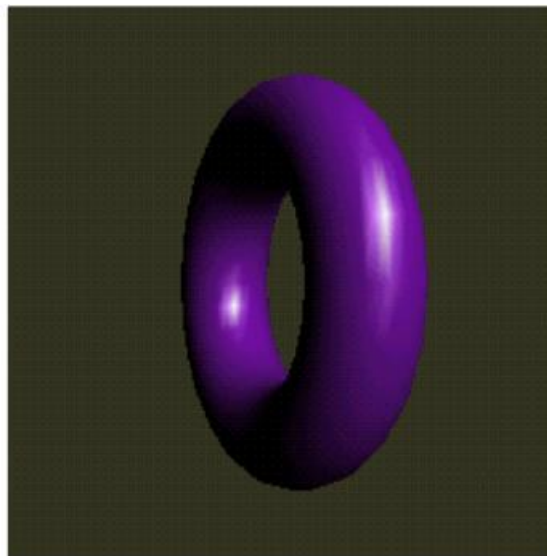


明暗处理

- 为何实际中球的图像类似于

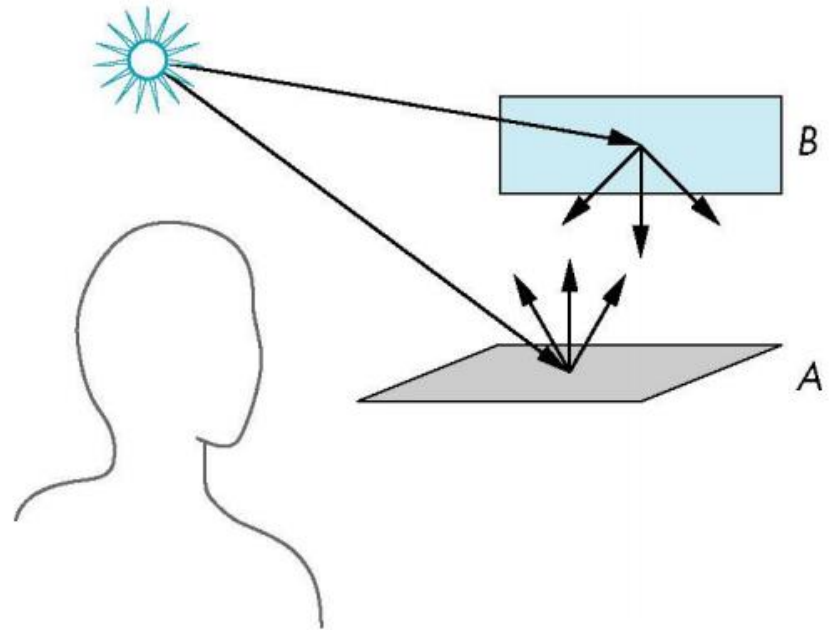


- 光与材料的交互作用导致每点有不同的颜色或者明暗效果
- 这时需要考虑
 - 光源
 - 材料属性
 - 观察者位置
 - 曲面定向

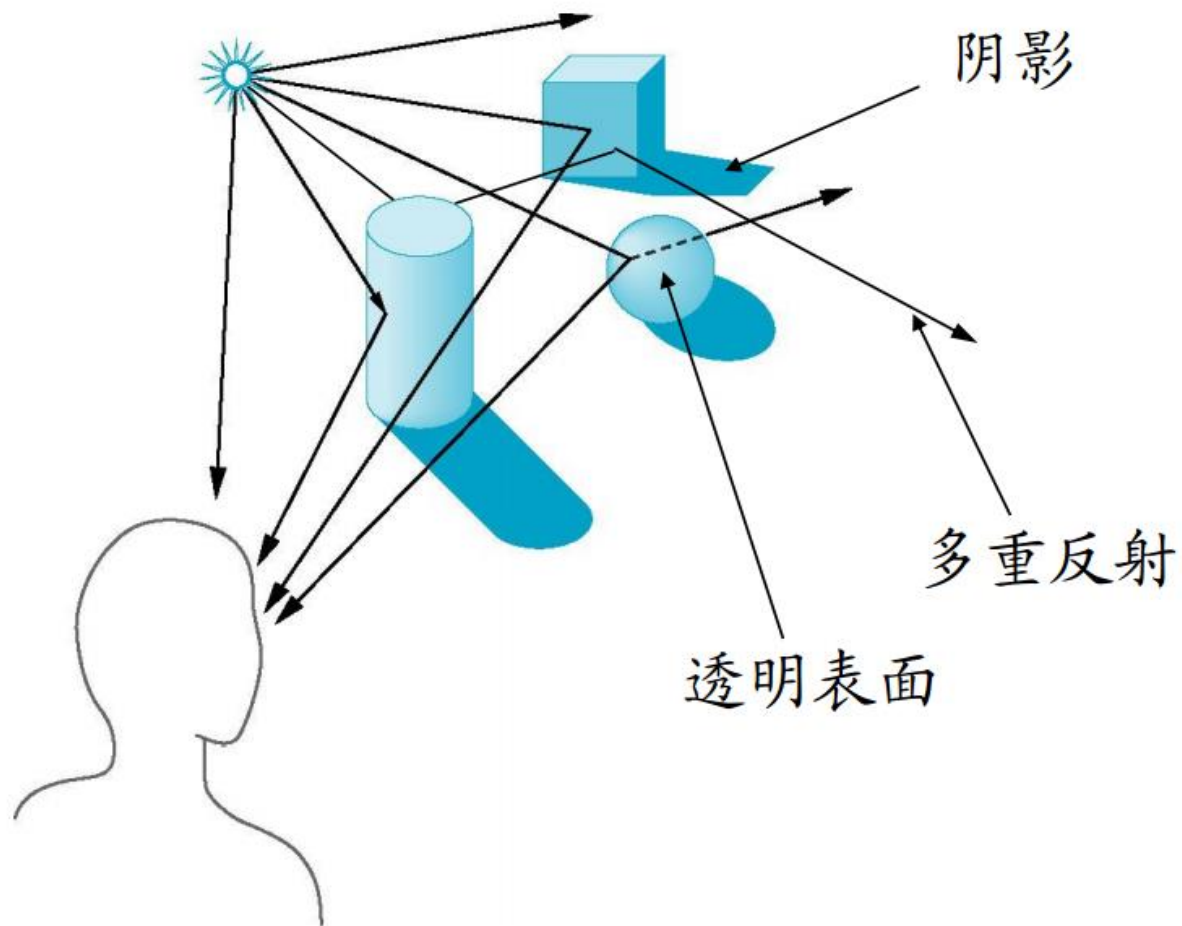


散射

- 光照射到A
 - 有些被反射
 - 有些被吸收
- 反射光中有些射到B
 - 有些被反射
 - 有些被吸收
- 反射光中又有些射到A,



全局效果

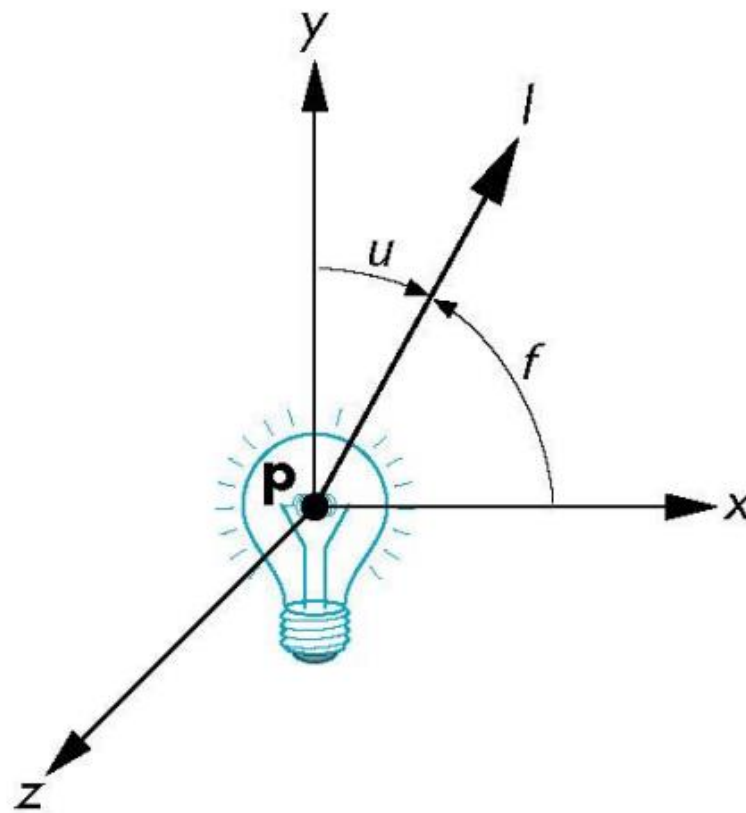


局部与全局光照

- 正确的明暗处理需要全局计算，即包含了所有的对象和光源
 - 这与流水线模型不兼容，在这个体系中对每个多边形单独进行明暗处理（局部光照）
- 然而在计算机图形学中，特别是在实时图形应用中，如果所得结果看起来可以的话，这种局部计算是可以接受的
 - 存在许多方法逼近全局效果

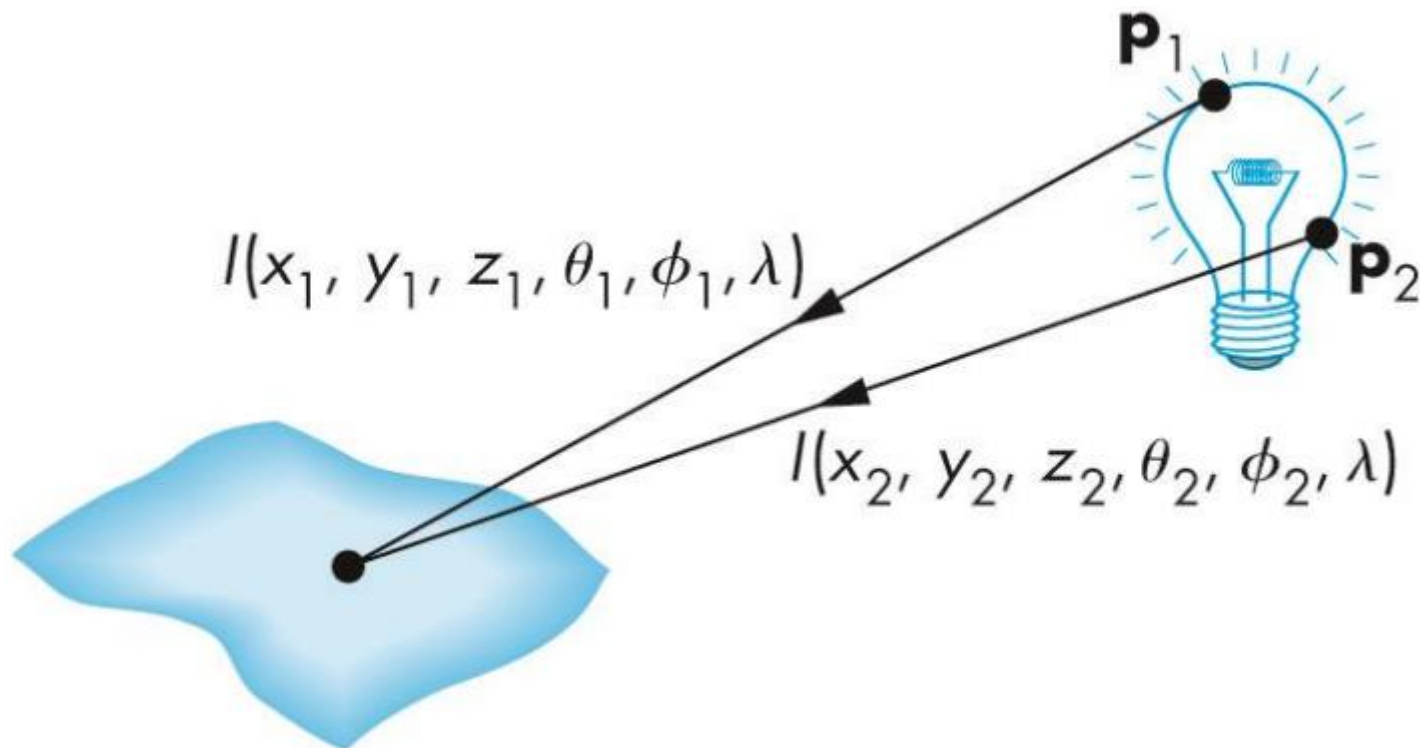
光源的描述

- 光线从光源表面离开的方式有两种
 - 自发射
 - 反射
- 在表面上任一点 (x, y, z) 所发出的光可以用发射方向 (θ, φ) 和波长 λ 的光强度来描述
- 光源模型：照明函数 $I(x, y, z, \theta, \varphi, \lambda)$



光源

- 一般的光源是很难处理的，因为我们需要对于在光源表面上的所有点进行光强积分



光源的颜色

- 光源不但发射出不同量的不同频率的光，而且它们的方向属性随着频率也可能不同
 - 真正的物理模型将非常复杂
- 人的视觉系统是基于三原色理论的
- 在大多数应用中，可以用三种成分——红、绿、蓝——的强度表示光源
 - 光亮度(luminance)函数为 $\mathbf{L} = [L_r, L_g, L_b]$

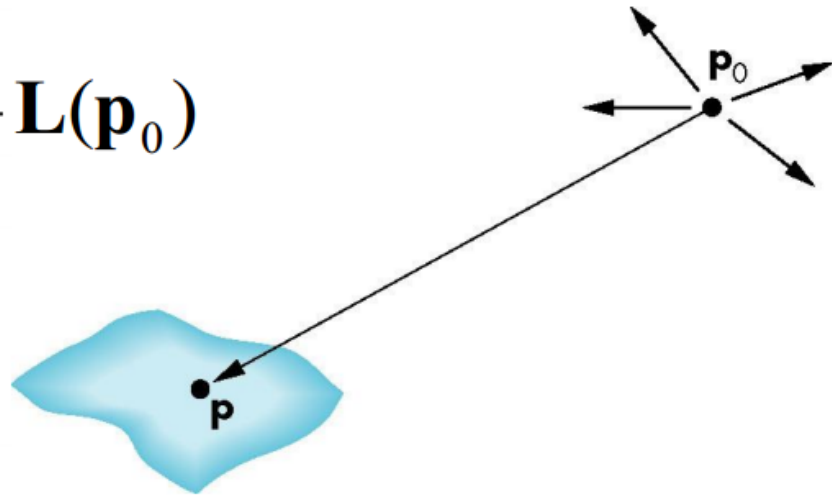
光源类型

- 四种基本类型
 - 点光源
 - 聚光灯
 - 无穷远光源
 - 环境光

点光源

- 由位置和颜色表示
- 理想的点光源向各个方向发射光线的强度相等
- 点光源的亮度函数 $\mathbf{L}(\mathbf{p}_0) = [L_r(\mathbf{p}_0), L_g(\mathbf{p}_0), L_b(\mathbf{p}_0)]$
- 点 \mathbf{p} 接受的光强反比于光源与点的距离平方

$$\mathbf{L}(\mathbf{p}, \mathbf{p}_0) = \frac{1}{|\mathbf{p} - \mathbf{p}_0|^2} \mathbf{L}(\mathbf{p}_0)$$

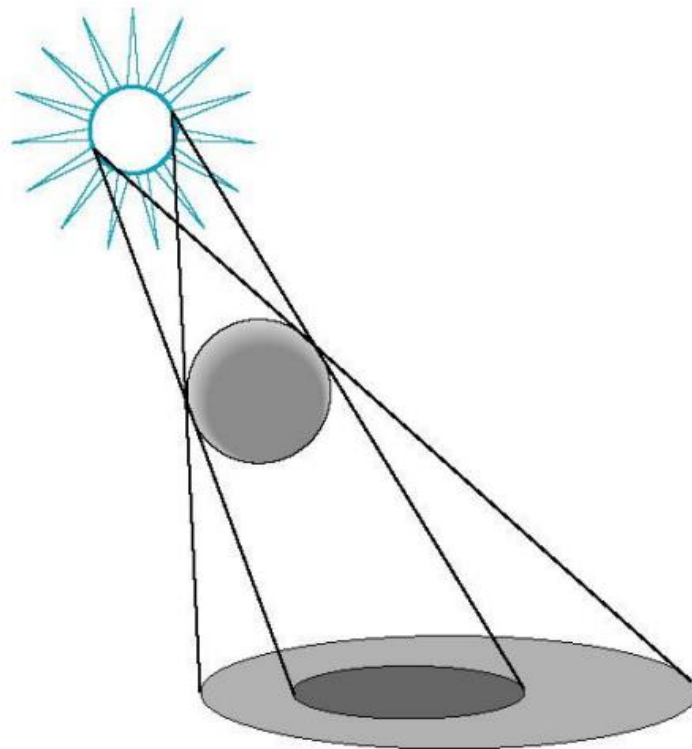


点光源的应用

- 在计算机图形学中大量应用点光源，是因为它易于使用
- 但不能很好地反映物理现实
 - 只有点光源的场景得到的图像中对比度较高；对象显得要么很亮，要么很暗
 - 而真实的光源由于尺寸较大，因此场景的结果比较柔和

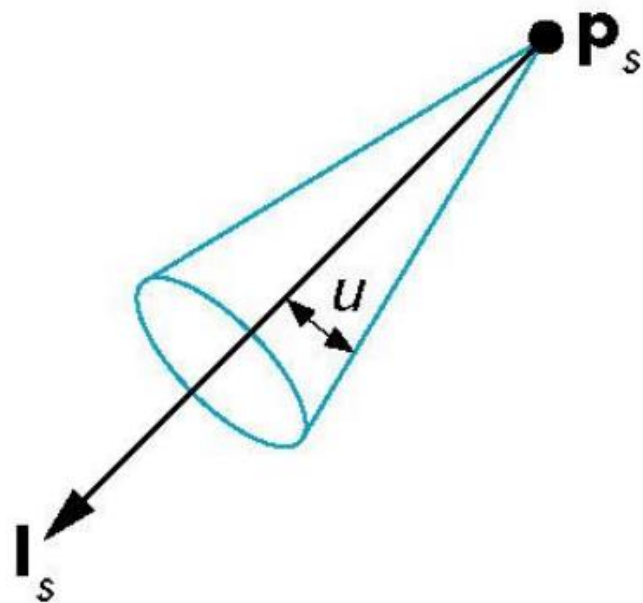
真实光照

- 完全在阴影中的区域称为本影(umbra)
- 部分在阴影中的区域称为半影(penumbra)
- 可以在点光源中加入环境光降低高对比度的问题



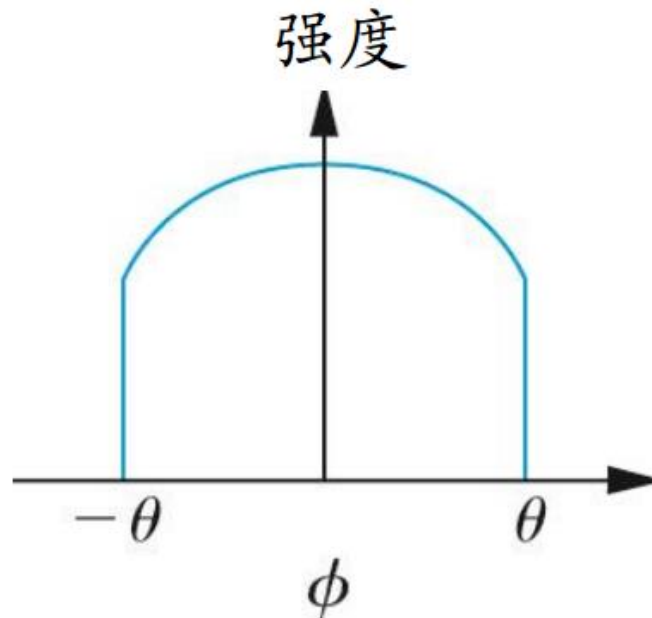
聚光灯

- 聚光灯(spotlight)具有一个比较的窄的照明范围,通常为圆锥形半无穷区域
 - 可以给点光源加上一定的限制得到
 - 锥的顶点在 \mathbf{p}_s , 而中心轴方向为 \mathbf{l}_s
 - 如果中心轴和母线的夹角 $u = 180^\circ$, 聚光灯成为点光源



更真实的聚光灯

- 光亮度在照明锥内具有一定的分布
 - 通常绝大多数光集中在照明锥的中心轴附近
 - 照明强度是光源到表面上某点向量 \mathbf{s} 与中心轴夹角 ϕ 的函数



聚光灯的照明强度函数

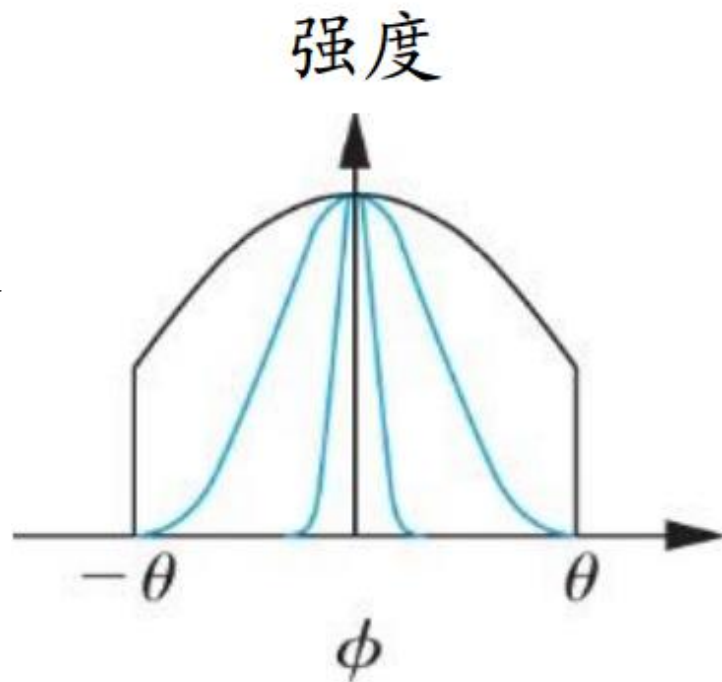
- 可以采用各种方式定义强度函数

- 通常定义为 $\cos^e \phi$

- 指数 e 确定光强度衰减的快慢

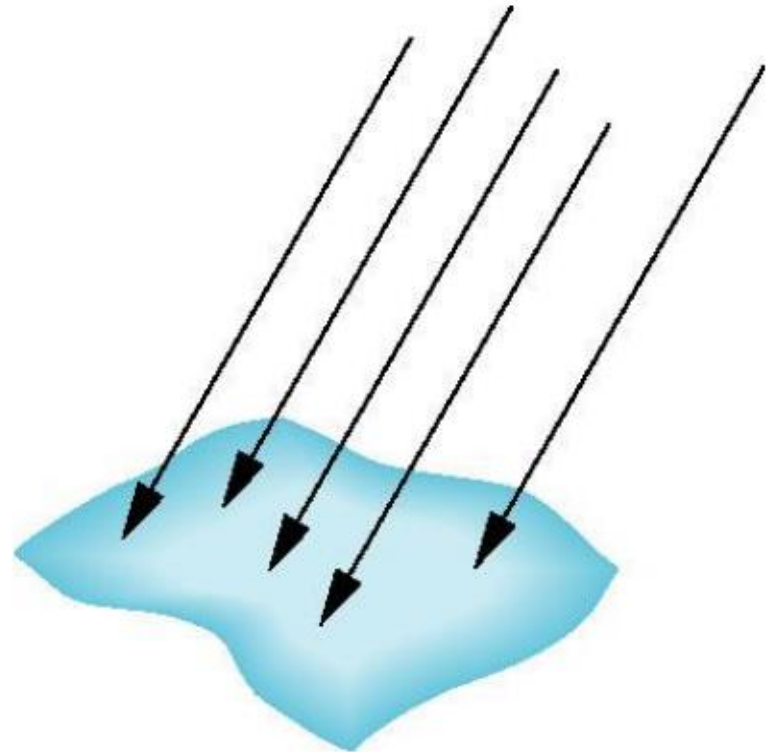
- 之所以采用余弦函数，是因为在这种情况下，非常容易计算出来它的值

$$\cos \phi = \mathbf{s} \cdot \mathbf{l}$$



无穷远光源

- 在光照计算中需要从表面上的点指向光源的方向向量
 - 入射方向为上述向量的相反向量，而且需要单位化
- 如果光源在无穷远，该向量为常数
 - 光线为平行线：平行光源
 - 太阳光是无穷远光源的典型代表



环境光

- 模拟均匀光照：在场景中每个点都具有相同的亮度
- 严格意义上说，环境光也是来自于某个光源，但由于在光照计算中进行了某些简化，需要用环境光来模拟多次反射后的效果
- 环境光亮度是由 $\mathbf{L}_a = [L_{ar}, L_{ag}, L_{ab}]$ 确定的，在每点的值完全相同
 - 但各个表面对环境光的反射是不同的

光线与材料的相互作用

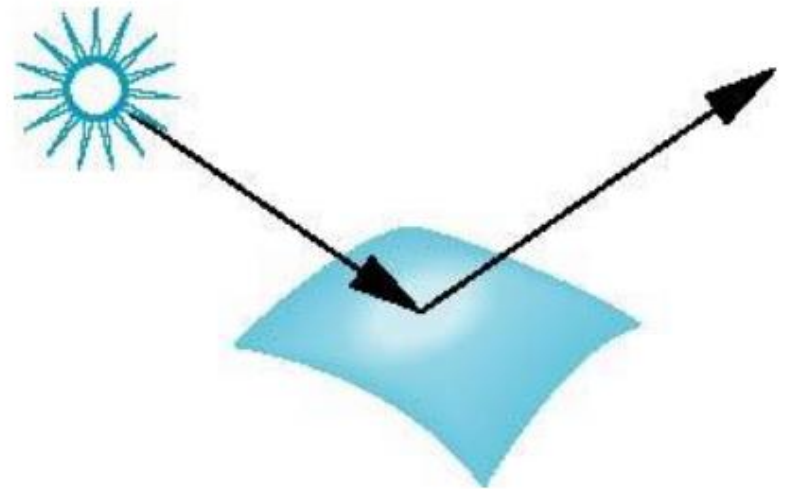
- 照射在对象上的光线部分被吸收，部分被散射（反射）
- 如果对象是透明的，有些光被折射
- 反射部分的多少确定对象的颜色与亮度
 - 对象表面在白光下看起来是红的，就是因为光线中的红色分量被反射，而其它分量被吸收
- 反射光被反射的方式是由表面的光滑程度和定向确定的

对象表面

- 如果对象表面光滑，对象显得明亮；如果表面粗糙，那么就显得暗淡
- 表面有三种类型
 - 镜面 (specular surfaces)
 - 漫反射面 (diffuse surfaces)
 - 透明面 (translucent surfaces)

镜面

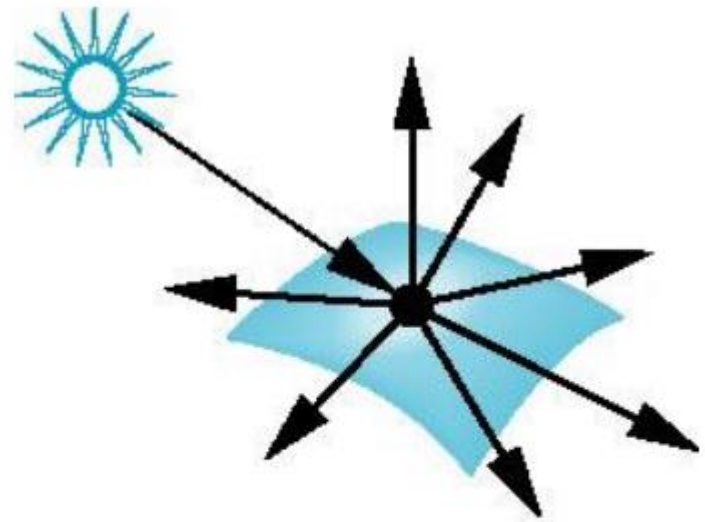
- 表面显得明亮，因为绝大多数光集中在严格镜面反射方向的周围
- 镜子是理想镜面模型
 - 入射光除部分被吸收外，全部以单一角度反射



光滑表面

漫反射面

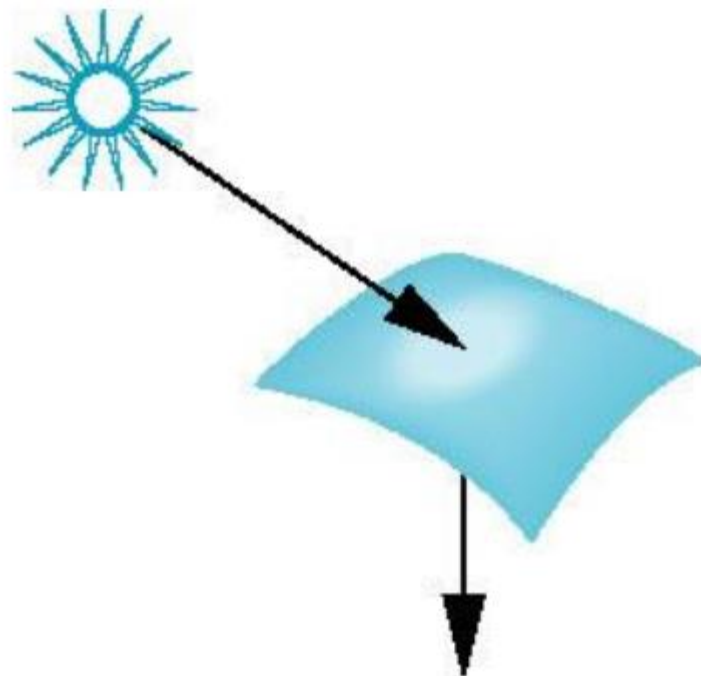
- 特征是反射光线散射到各个方向
- 例如：涂有粗糙或无光涂料的墙面
- 理想漫反射面：把光线均匀散射到各个方向



漫反射面

透明面

- 有些光可以进入表面，
从对象的另一处出来
- 例如：玻璃与水中的折射
- 这时也会有部分光被表面反射



透明面

三种情形的表面

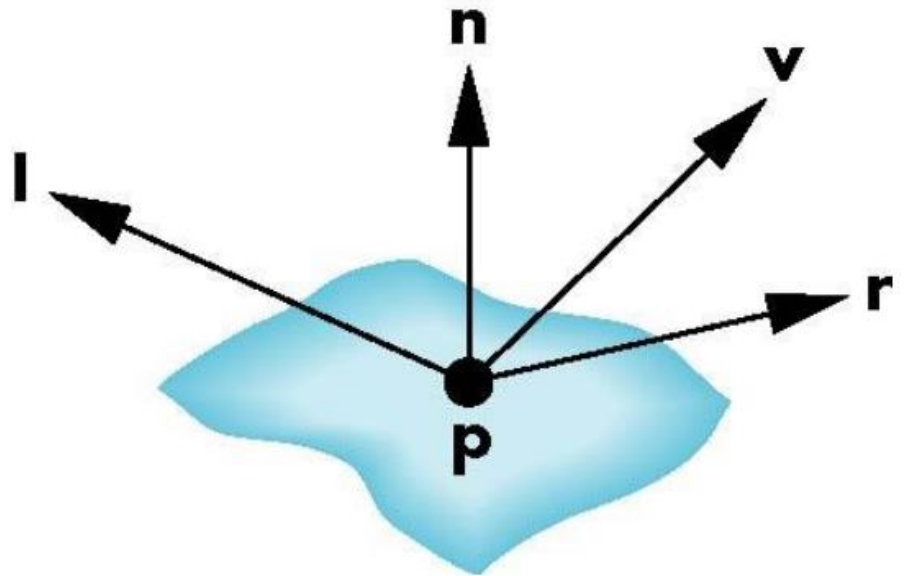
- 在三维场景中的每个对象的表面可以是上述三种情形中的任一种，也可以是其中两种或三种的综合
- 每种情形所占比例由对象表面的性态确定

第二部分 Phong光照模型

- 主要内容
 - Phong光照模型
 - 模型
 - 向量计算

Phong光照模型

- 可以快速计算的局部光照模型
- 有三类分量
 - 漫反射光
 - 镜面反射光
 - 环境光
- 使用四个向量
 - 入射光方向 \mathbf{l}
 - 视点方向 \mathbf{v}
 - 法向 \mathbf{n}
 - 理想反射方向 \mathbf{r}



光源模型

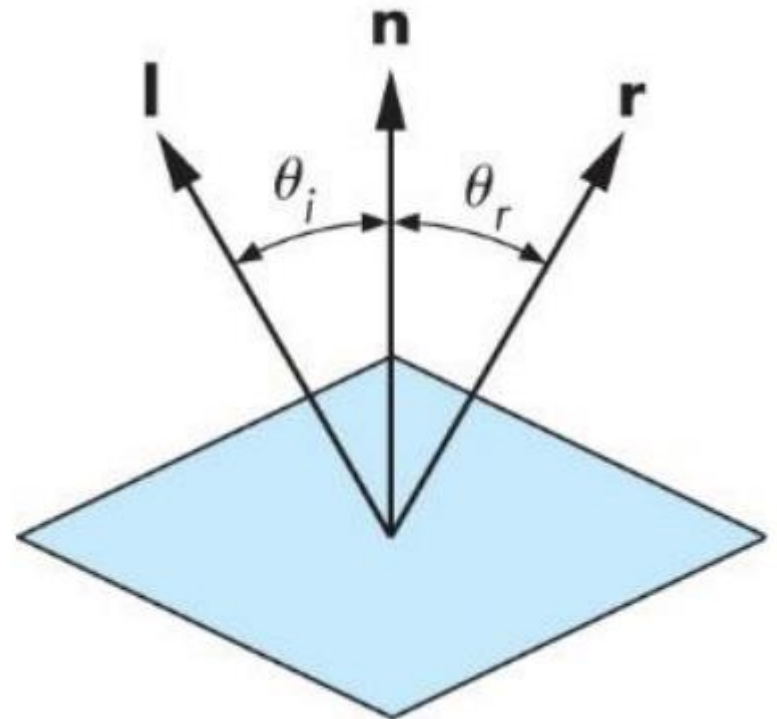
- Phong模型假设点光源的三原色都有各自的环境光、漫反射光和镜面反射光
 - 用局部光照模型去模拟本质上全局的光照效果
 - 需要9个系数来描述光源在表面上点 \mathbf{p} 处的光照属性，即入射光属性：
 - $L_{ar}, L_{ag}, L_{ab}, L_{dr}, L_{dg}, L_{db}, L_{sr}, L_{sg}, L_{sb}$
 - 这里没考虑距离衰减因素

反射系数

- 光照模型的前提是已知某点的入射光，通过某种方法求得在该点的反射光
- 假设光源的漫反射项中红光分量为 L_{dr} ，点 p 的反射率为 R_{dr} ，则光源在该点光强的贡献值为 $R_{dr} L_{dr}$
 - R_{dr} 与材料属性、表面朝向、光源方向以及光源与观察者距离有关
 - 9个反射系数

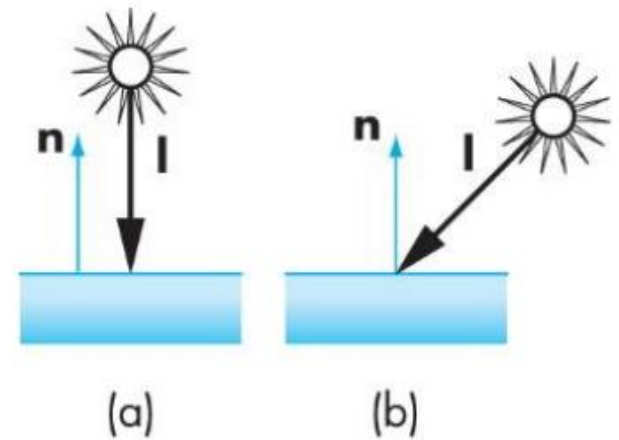
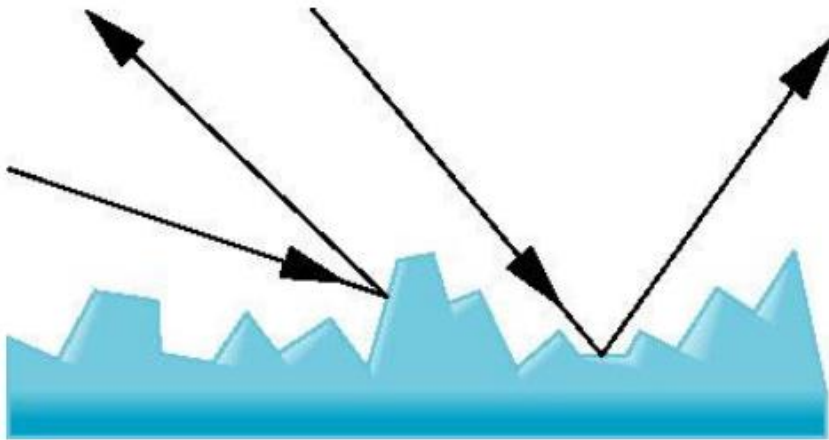
理想反射

- 法向由局部定向确定
 - 单位化 $|\mathbf{n}| = 1$
- 入射角 = 反射角
 $\theta_i = \theta_r$
- 三个向量 \mathbf{l} , \mathbf{n} , \mathbf{r} 必须共面
 - 均为单位向量
$$\mathbf{r} = 2(\mathbf{l} \cdot \mathbf{n})\mathbf{n} - \mathbf{l}$$



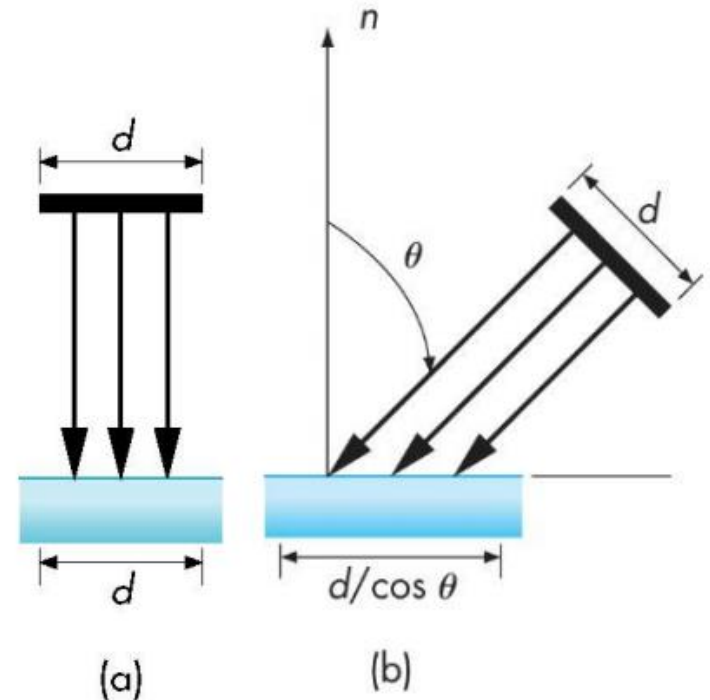
朗伯(Lambertian)表面

- 真正的漫反射
- 光向各个方向均匀地散射
- 模拟粗糙表面



漫反射光强

- 反射光的比例正比于入射光的竖直分量
 - 即反射光 $\approx \cos \theta_i$
 - 如果向量为单位向量，则 $\cos \theta_i = \mathbf{l} \cdot \mathbf{n}$
 - 存在三个系数 k_{dr} , k_{dg} , k_{db} 分别相应于每种颜色的光反射的比例
- 对每种光，漫反射光强 $I_d = k_d (\mathbf{l} \cdot \mathbf{n}) L_d$
 - $(\mathbf{l} \cdot \mathbf{n})$ 常采用 $\max(\mathbf{l} \cdot \mathbf{n}, 0)$ 的形式



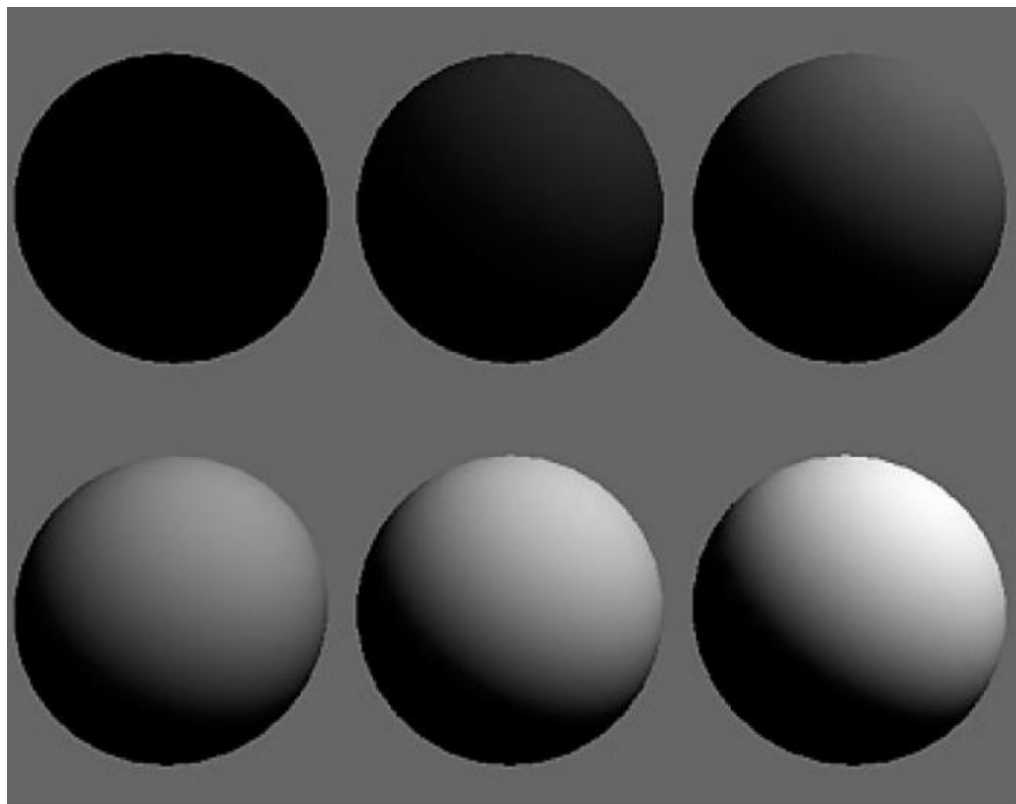
特点

- 粗糙表面之所以可见，主要是来自于表面的漫反射
- 基于简单Lambertian漫反射模型所生成的对象图形显得比较暗淡
 - 因为此时相当于假定在视点处有一个点光源，那些光源无法直接照到的地方显得较黑
 - 在真实的场景中，对象也接受到其它对象反射过来的光

漫反射参数的影响

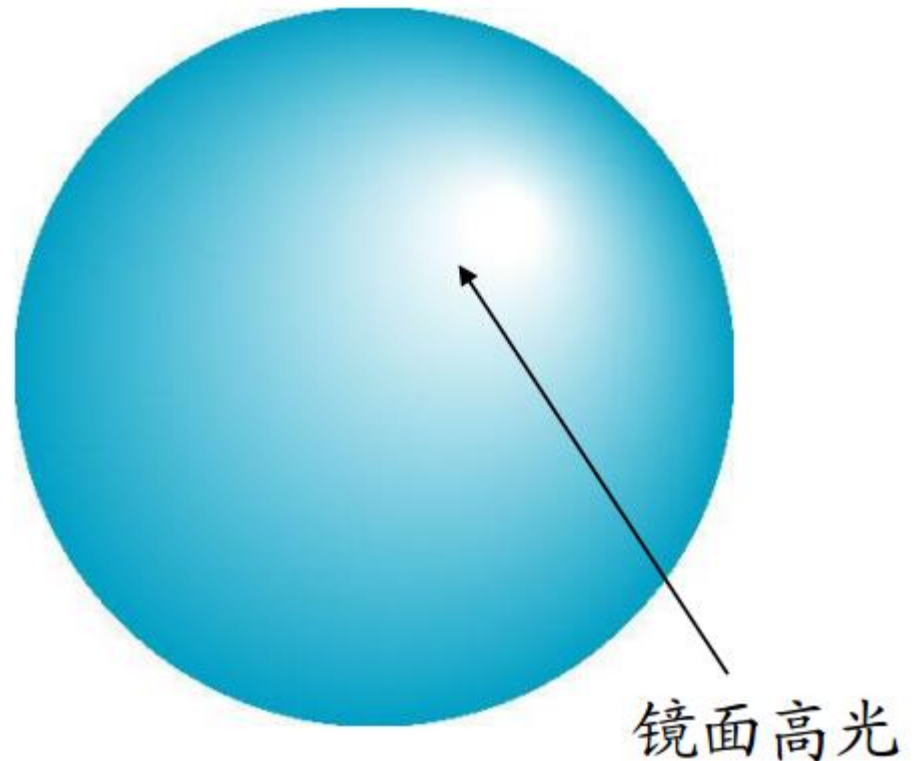
漫反射系数依次
为0.0, 0.2, 0.4,
0.6, 0.8, 1.0

光强为1.0



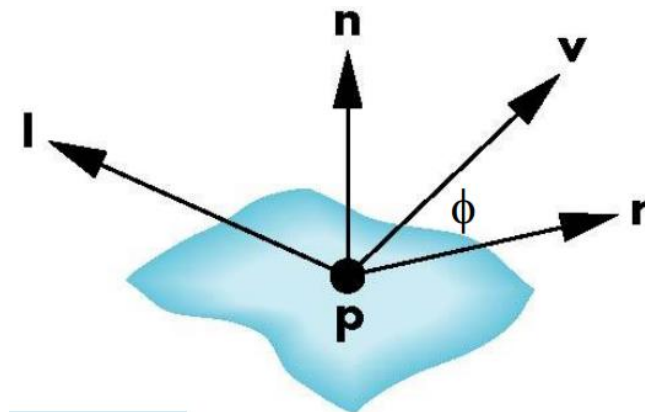
镜面光

- 大多数曲面既不是理想的漫反射型曲面，也不是真正的镜面(理想反射)
- 光滑表面之所以显出镜面高光，是因为入射光反射后，绝大多数集中在理想反射方向周围



镜面光模型

- **n**: 法向量
- **l**: 入射光方向
- **r**: 理想反射方向
- **v**: 视点方向
- ϕ : **r**与**v**的夹角, $\cos \phi = \mathbf{v} \cdot \mathbf{r}$

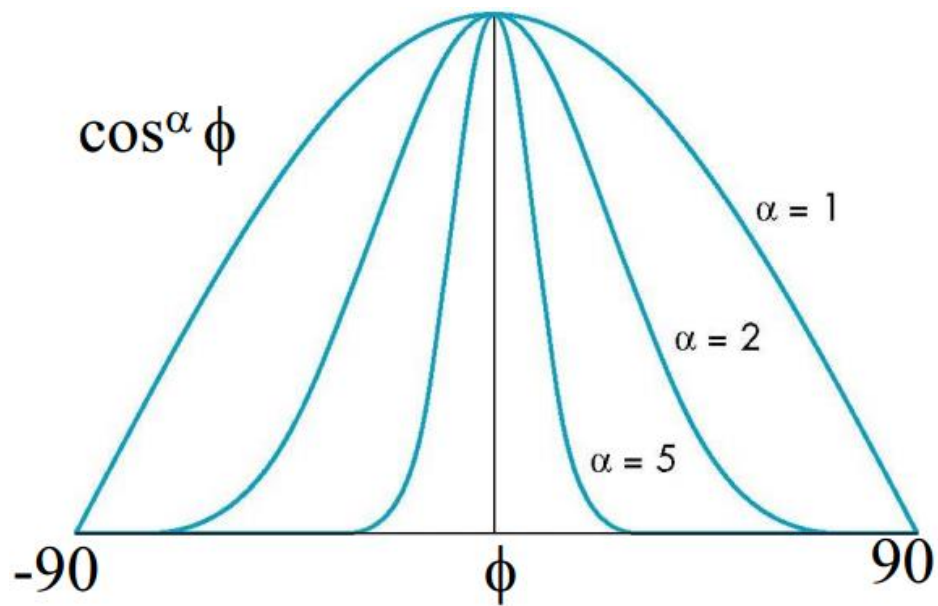


$$I_s \sim k_s L_s \cos^{\alpha} \phi$$

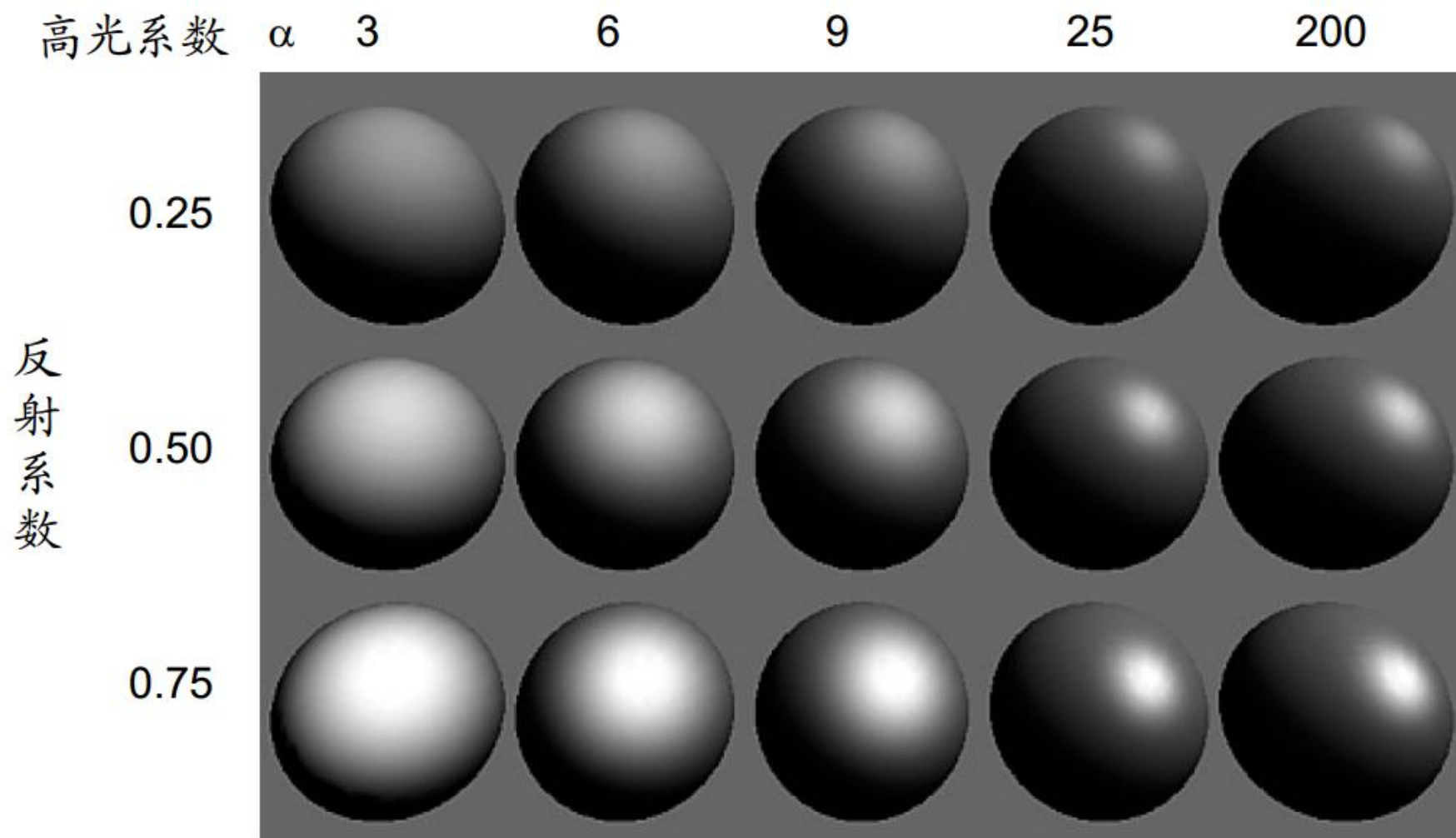
反射光强 反射系数 入射光强 高光系数

高光系数

- a 的值介于 100 到 200 之间，那么对应于金属材料
- a 的值介于 5 到 10 之间，材料类似于塑料
- a 趋向无穷大时，镜子



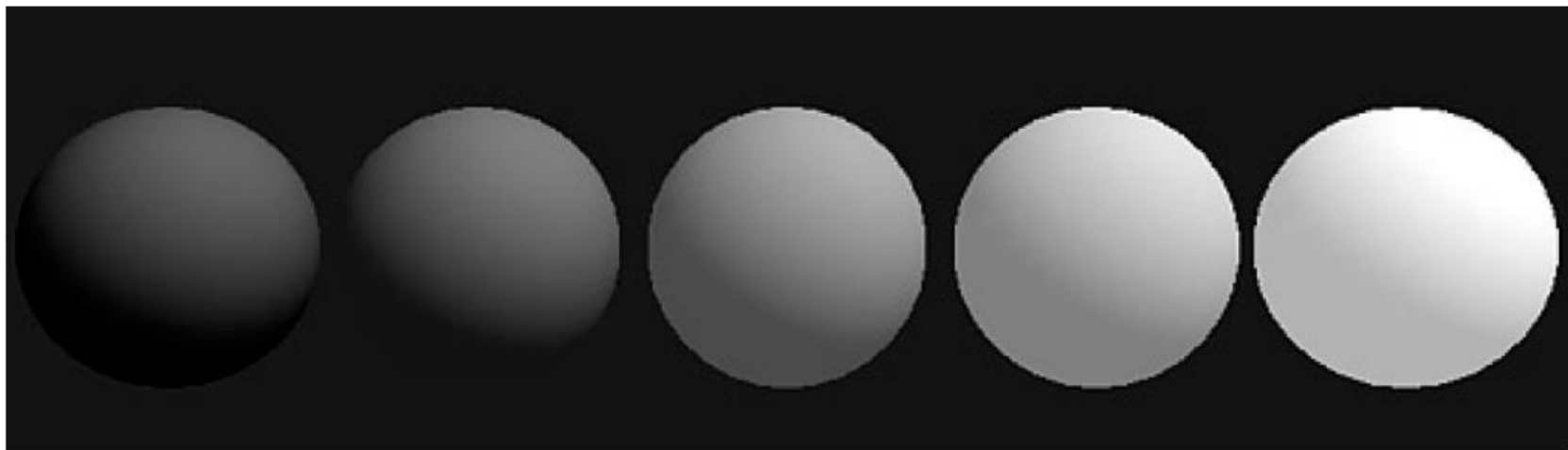
镜面反射参数的影响



环境光

- 环境光是由于在场景中（大的）光源与对象间的多次相互作用而导致的
- 环境光的量与颜色依赖于光源的颜色和对象的材料属性
- 向漫反射和镜面反射项中添加上 $I_a = k_a L_a$ 项
 - k_a : 反射系数
 - L_a : 环境光强

环境光的影响



向漫反射光中加入不同的环境光的效果

两种光强都是1.0, 漫反射系数为0.04

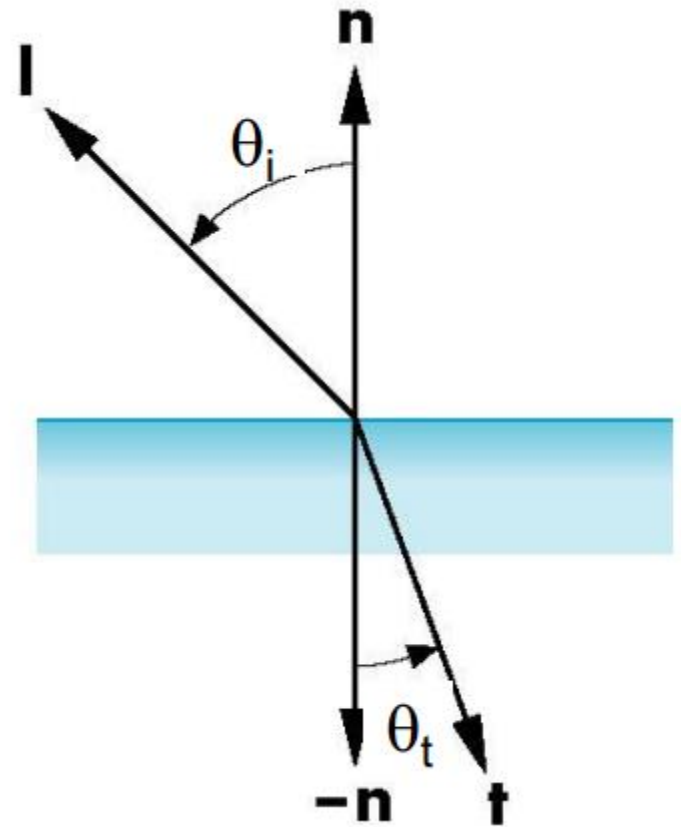
环境光反射系数依次为0.0, 0.1, 0.3, 0.5, 0.7

折射光

- Snell定律

$$\frac{\sin \theta_i}{\sin \theta_t} = \frac{\eta_t}{\eta_i}$$

η_t, η_i 分别表示两种物质的
折射系数

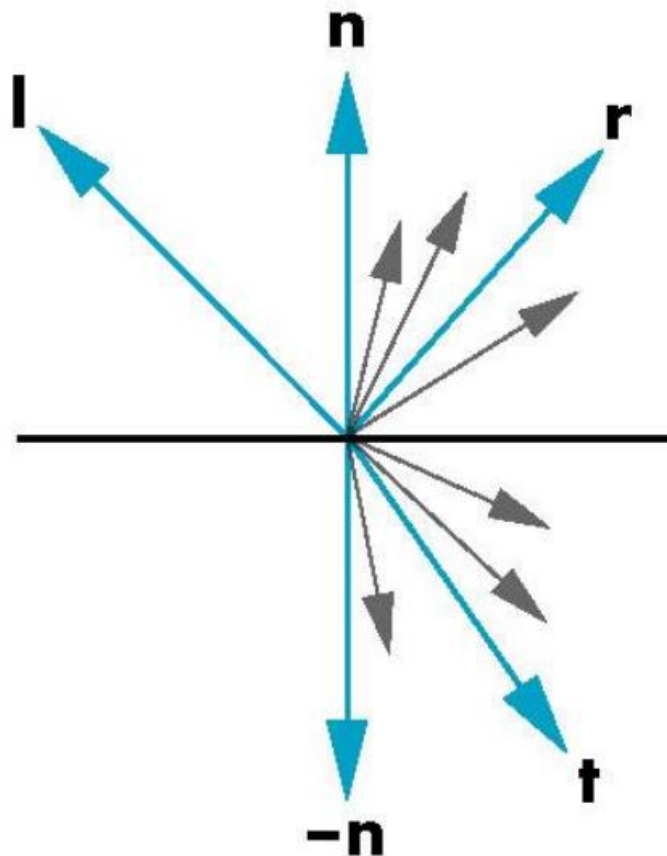


常见材料的折射率

真空	1	空气	1.0003
水	1.33	酒精	1.36
玻璃	1.5~1.6	石英	1.54
红宝石	1.77	钻石	2.42

多种光的综合

- 在简单光照模型中，漫反射光、镜面光和折射光同时并存
- 即使在折射光中也有一部分光类似于漫反射光向各个方向发射



距离项

- 从点光源到达对象表面的光强反比于两者之间距离的平方
- 向漫反射项和镜面项中添加形式为 $1/(a + b d + c d^2)$ 的二次距离衰减因子，其中 d 表示距离
- 常数与线性项起到柔和点光源的效果

光源

- 在光照模型中，把每个光源的结果叠加在一起
- 每个光源具有不同的漫反射、镜面和环境光项，从而充分发挥各自最大的灵活性，虽然这种处理并没有任何物理上的理由
- 对三原色中的红绿蓝分量分别处理
- 因此每个点光源有九个系数
 - $L_{dr}, L_{dg}, L_{db}, L_{sr}, L_{sg}, L_{sb}, L_{ar}, L_{ag}, L_{ab}$

材料属性

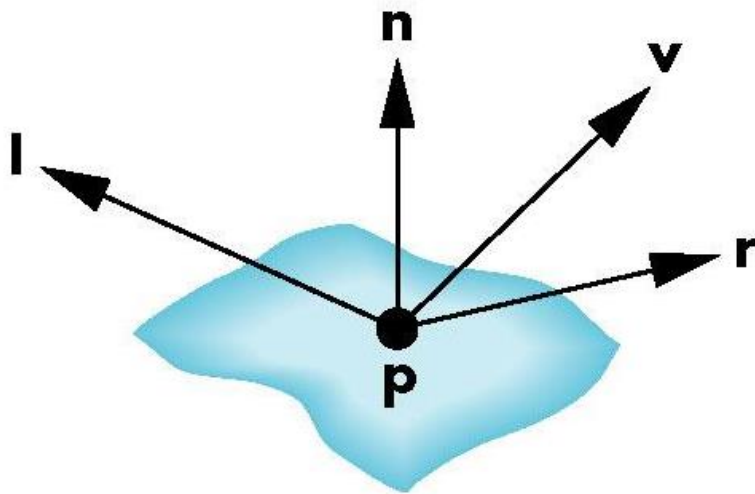
- 材料属性与光源属性相匹配
 - 九个反射系数
 - k_{dr} , k_{dg} , k_{db} , k_{sr} , k_{sg} , k_{sb} , k_{ar} , k_{ag} , k_{ab}
 - 高光系数 a

把各种分量叠加在一起

- 对于每个光源和每种颜色成分， Phong光照模型可以表示为(这里未放入距离项)

$$I = k_d L_d \max(\mathbf{l} \cdot \mathbf{n}, 0) + k_s L_s \max((\mathbf{v} \cdot \mathbf{r})^\alpha, 0) + k_a L_a$$

- 对每个颜色分量， 把所有光源贡献的值加在一起

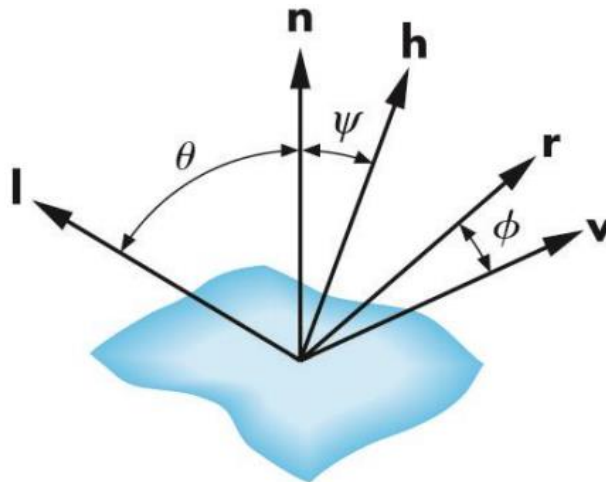


改进的Phong模型

- 在Phong模型中， 镜面光项有一个问题，因为它需要为每个顶点计算一个新的反射向量和视点向量
- Blinn利用中值(halfway)向量给出了一个近似， 从而使得效率更高

中值向量

- 中值向量 \mathbf{h} 是 \mathbf{l} 和 \mathbf{v} 的中值单位向量，即 $\mathbf{h} = (\mathbf{l} + \mathbf{v}) / |\mathbf{l} + \mathbf{v}|$
- \mathbf{n} 和 \mathbf{h} 的夹角 ψ 称为中值角(haflway angle)
 - 当 \mathbf{v} 位于 \mathbf{l} 、 \mathbf{n} 和 \mathbf{r} 所在平面时，可以证明 $2\psi = \phi$



改进的Phong模型

- 镜面项用 $(\mathbf{n} \cdot \mathbf{h})^\beta$ 代替 $(\mathbf{v} \cdot \mathbf{r})^\alpha$
 - 参数 β 恰当选取，以匹配光洁度 α
- 当 $\mathbf{l}, \mathbf{n}, \mathbf{v}$ 共面时，中值角 ψ 就是 \mathbf{r} 和 \mathbf{v} 的夹角 ϕ 的一半
- 由此得到的模型称为改进的Phong模型或者Blinn光照模型
 - 在OpenGL标准中实现

向量计算

- \mathbf{l} 和 \mathbf{v} 由应用程序指定
- 可以从 \mathbf{l} 和 \mathbf{n} 计算 \mathbf{r} ($\mathbf{r} = 2 (\mathbf{l} \cdot \mathbf{n}) \mathbf{n} - \mathbf{l}$)
- 问题就剩下如何确定 \mathbf{n}
 - 对于简单曲面, \mathbf{n} 可以被求出。例如平面、球面等
 - 对于复杂曲面, 可以对其局部邻域几何进行分析。
(这里不做介绍, 一般在模型文件中都会有normal这一项)

OpenGL法向定义

- OpenGL里， 顶点法向用如下函数定义

```
glNormal3f(nx, ny, nz);  
glNormal3fv(pointer_to_normal);
```

- 法向是状态值