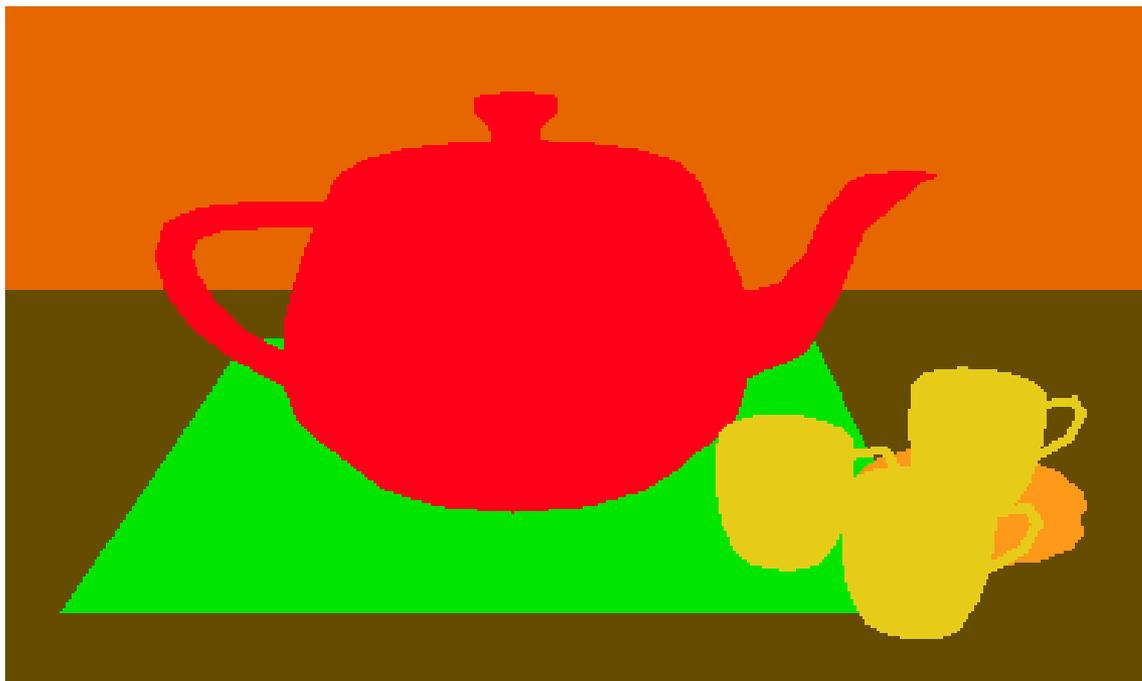


光照和色调

计算机图形学——原理、方法及应用（第4版），潘云鹤、童若锋、耿卫东、唐敏、童欣，高等教育出版社，2022。

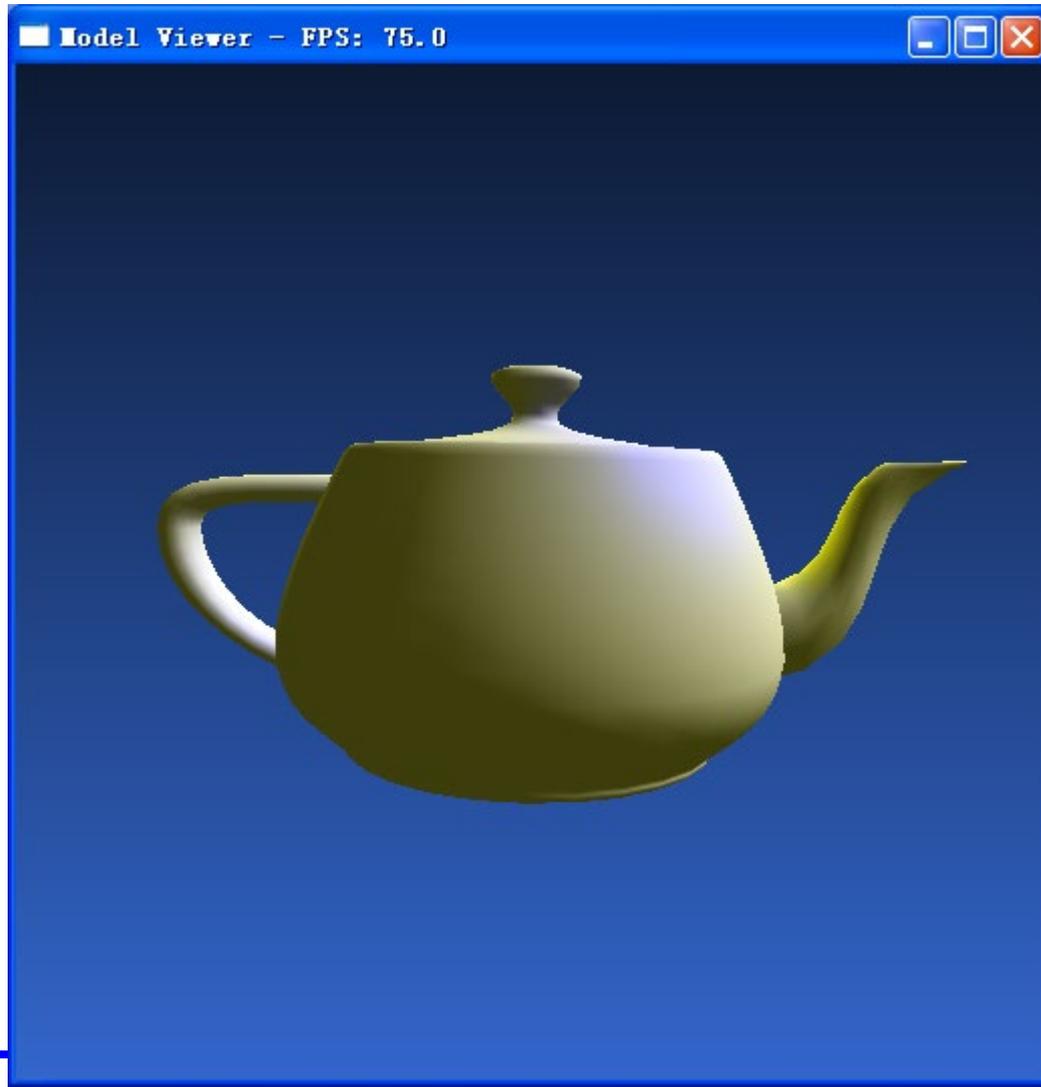
照明 & 色调

没有光源 ... 我们看不见场景的内容!



没有色调, 物体看上去不是三维的!

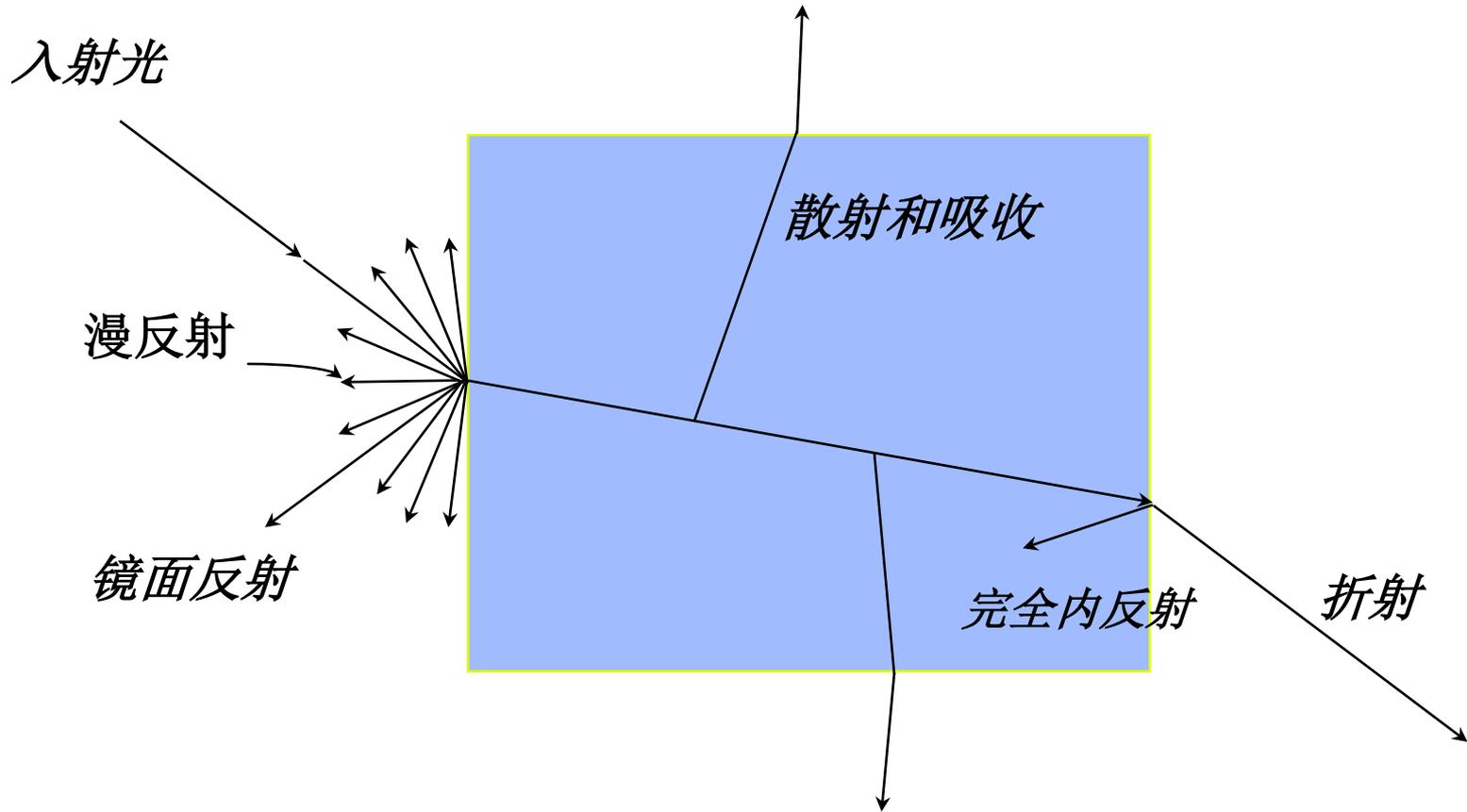
演示 - 多光源



光照

- **光照** 是所有光线击中特定表面特定位置的物理现象的完整描述
- 物体某点的**颜色** 由离开该点的光的物理属性所决定
- 在已知**光照** 和表面某点**物理属性**的情况下，我们可以计算出离开该点的光的物理属性
- 为了生成具有真实感的图像，我们需要理解光是如何与物体表面相互作用的

光与物体的相互作用



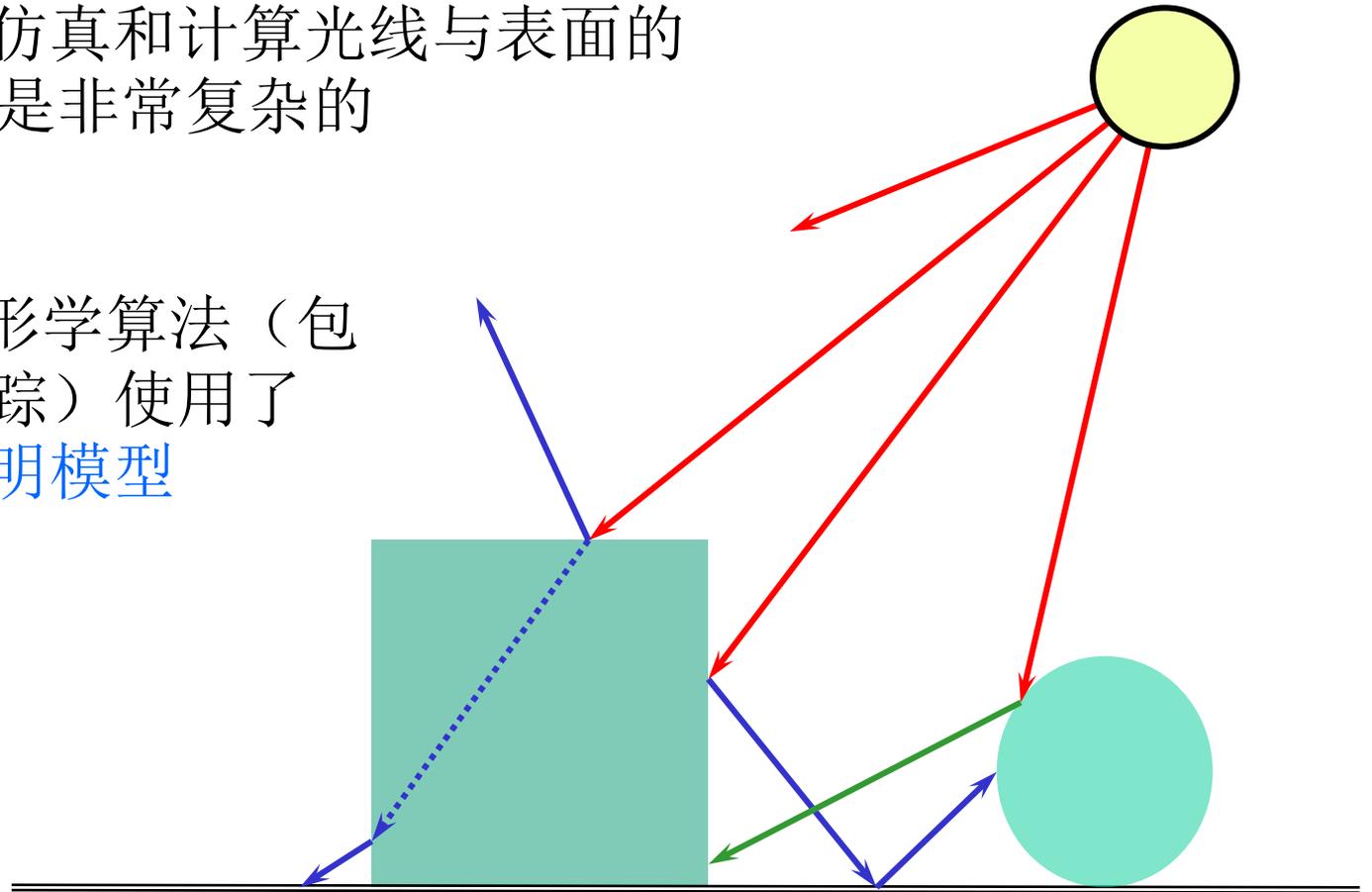
光的相互作用

- 两种重要的光照现象
 - 光与材质边界的交互作用
 - 光通过材质的散射和吸收
- 材质边界由环境中的表面组成
- 光在边界处发生反射和透射，对于不透明材质，光在边界处被吸收。

场景中的光传播

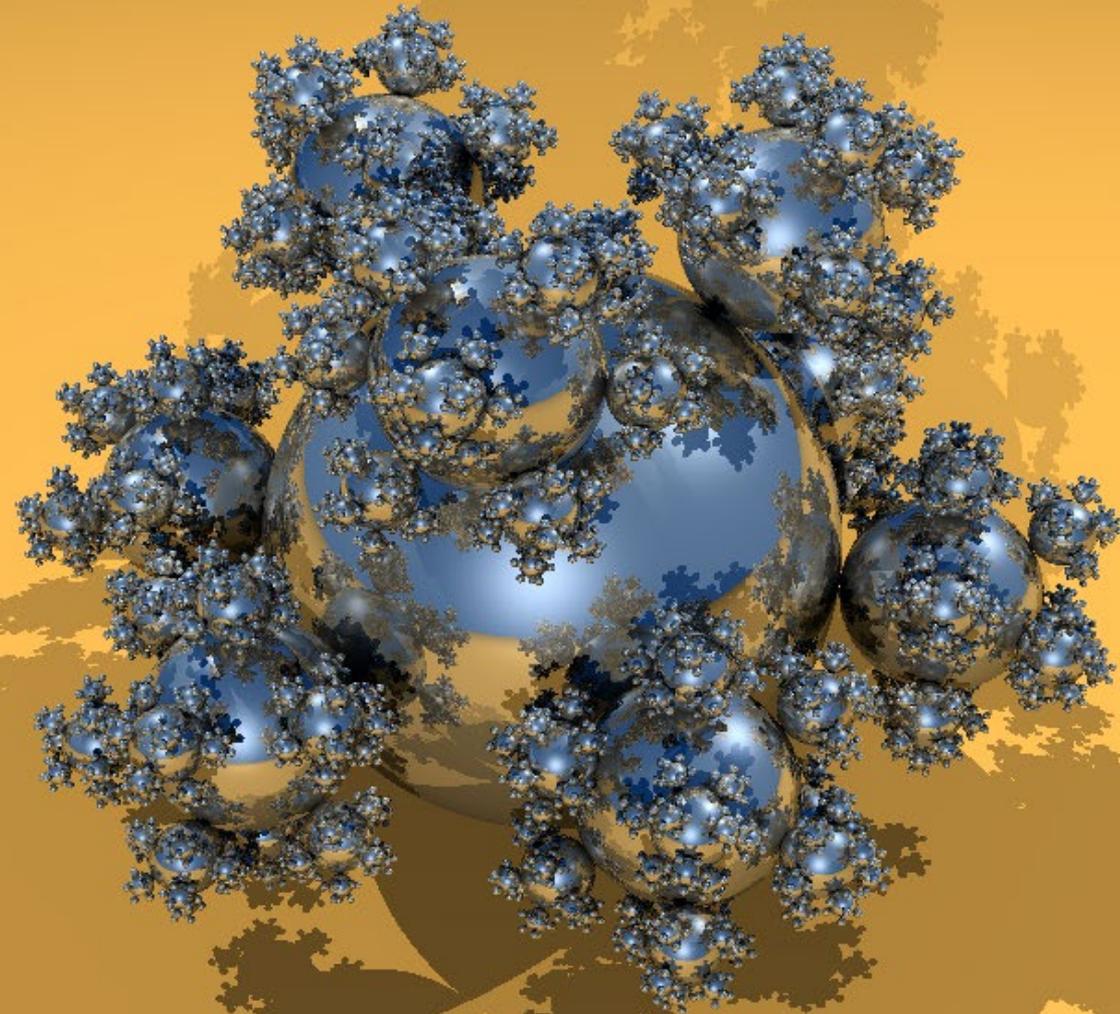
物理真实的仿真和计算光线与表面的交互作用是非常复杂的

绝大部分图形学算法（包括光线跟踪）使用了近似的**照明模型**



光照模型

- 表面上的一点可被以下照明：
 - *局部光*: 直接来自光源的光
 - *全局光*: 反射自或穿透自身或其他表面的光
- *光照模型* 体现了决定表面某点颜色的因素
- *光照模型* 使用了局部光和全局光来计算某点的颜色



反射模型

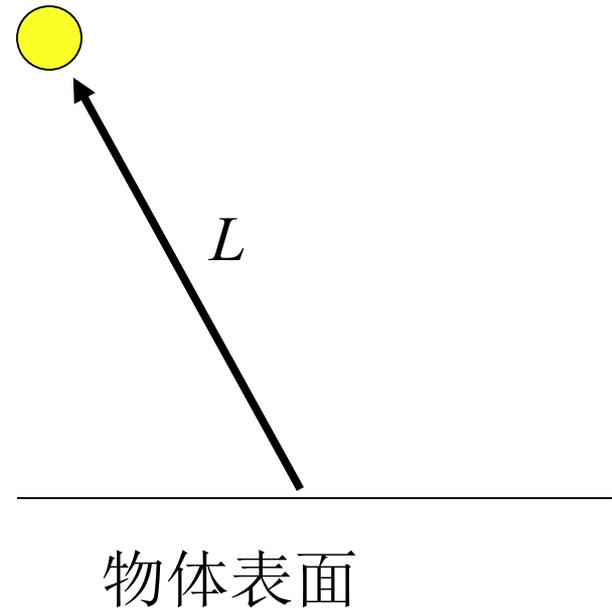
- 图形学最初使用的**光照模型**大部分是近似模型
- 主要目的是对光与物体间的相互作用进行建模，并且希望同时具有真实感和快速

简单反射模型

- 是最简单的光照模型
- 有如下假定：
 - 只有**局部光照**，没有全局光照
 - 光在物体表面只有**反射**，不会穿透物体
 - 没有传播性介质，即物体在真空中

光照向量

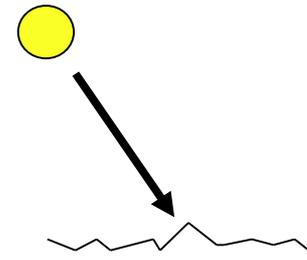
- 我们把从物体表面点指向光源的向量定义为光照向量
- 注意从物体到光源！



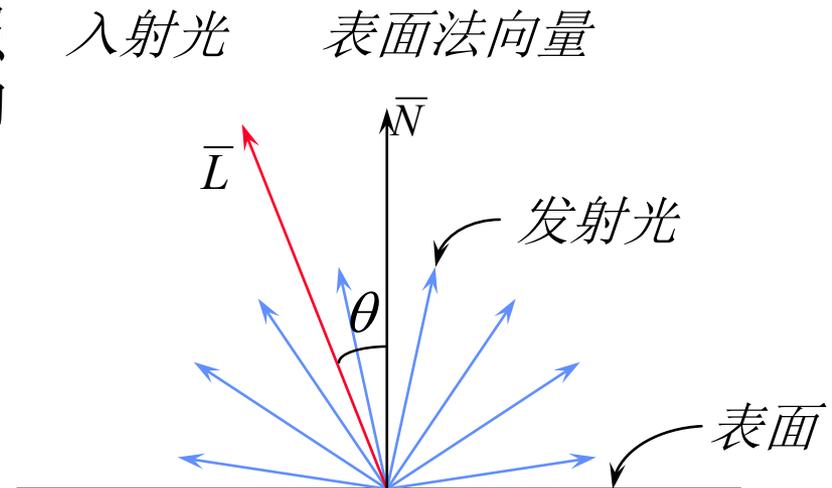
漫反射

- 漫反射 也被称为 **Lambertian** 反射，刻画了物体的粗糙表面，例如粉笔、织物
- 从物体表面某点反射出的光照强度在各个方向相等
- 发射光的亮度依赖于光照向量和物体表面法向量的夹角 θ
- 该亮度与观察方向无关

$$I_d = I_p k_d \cos(\theta)$$

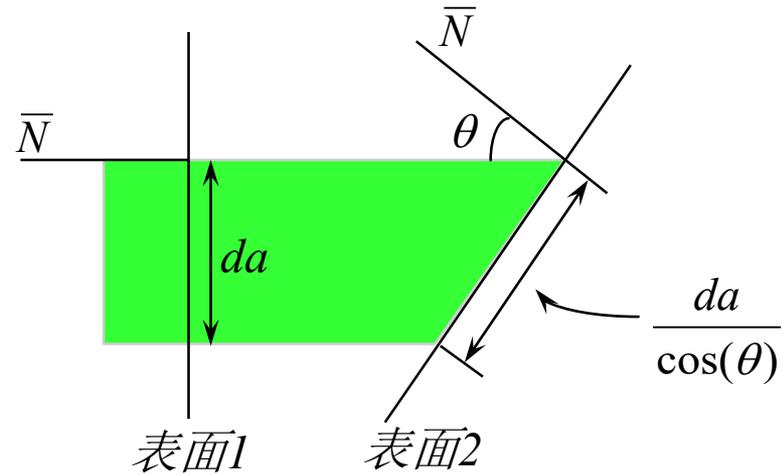


粗糙表面的放大图



漫反射

- 如果入射光柱的截面积为 da ，并且与照射表面的夹角为 θ ，则光柱的接收面积为 $da/\cos(\theta)$
- 物体表面接收到的光能与 $\cos(\theta)$ 成比例
- 漫反射方程为 $I_d = I_p k_d \cos(\theta)$ ，这里 I_p 是点光源的强度， k_d 是材质的漫反射系数，取值在0-1之间。 θ 需要在 0° 和 90° 之间，该点才会被照亮。

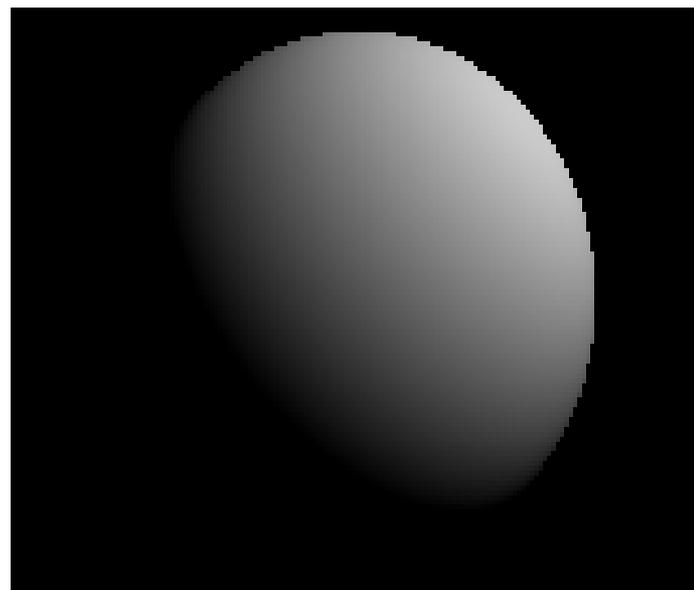
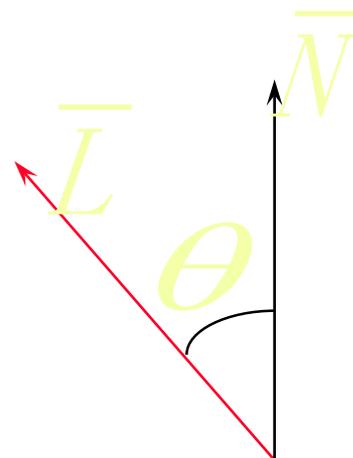


漫反射

- 如果 \bar{N} 和 \bar{L} 都是单位向量，则漫反射方程可以如下改写：

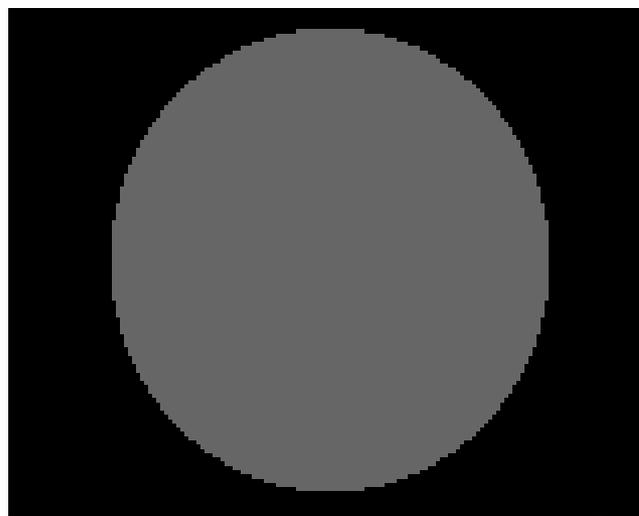
$$I_d = I_p k_d (\bar{N} \cdot \bar{L})$$

- 仅有漫反射模型，物体表面没有被照明的部分是黑色的。



环境光

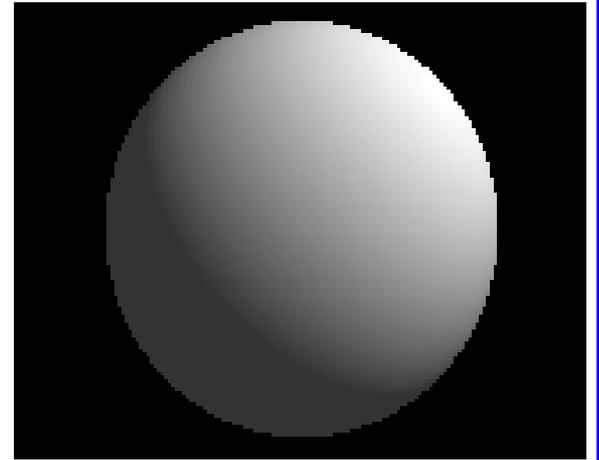
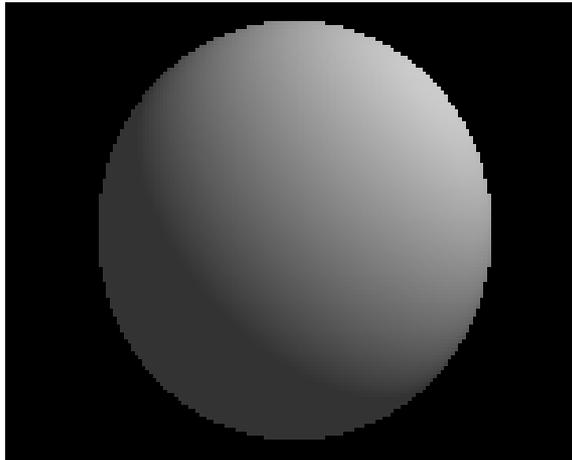
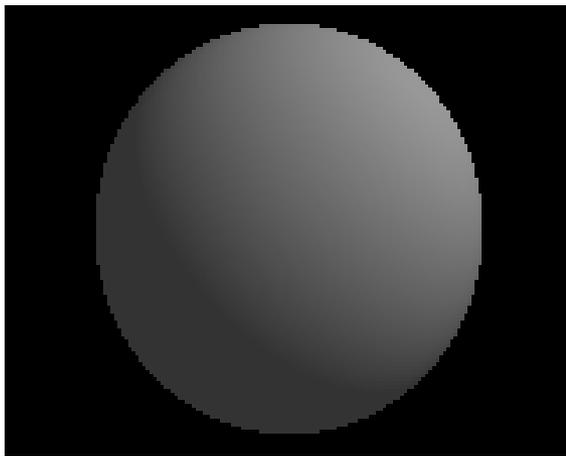
- 在现实世界，即使没有被光源直接照射到的部分也不会是完全黑色的
- 这些部分被全局光照所照明，即光被周围环境多次反射照明
- 这部分光照通常用一个常量环境光刻画
- 环境光照为 $I_a k_a$ ，这里 I_a 是环境光的强度， k_a 是材料的环境光反射系数，取值在0/1之间



照明方程

- 加入了环境光，则漫反射的照明方程如下：

$$I = I_a k_a + I_p k_d (\bar{N} \bullet \bar{L})$$



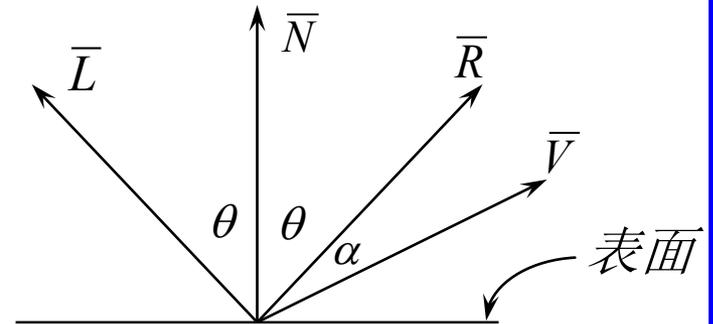
$$k_a = 0.2, k_d = 0.4$$

$$k_a = 0.2, k_d = 0.6$$

$$k_a = 0.2, k_d = 0.8$$

镜面反射

- **镜面反射** 在光亮表面展现，例如塑料
- 对于完美镜面，光只在单一方向 \bar{R} 进行反射。对于观察者，只有其观察方向 \bar{V} 与反射方向完全重合时，即 $\alpha = 0$ 时，才能接收到反射光
- 而自然界的光亮表面会在不同方向上进行各向异性的反射
- 这种现象被 **Phong光照模型** 所刻画

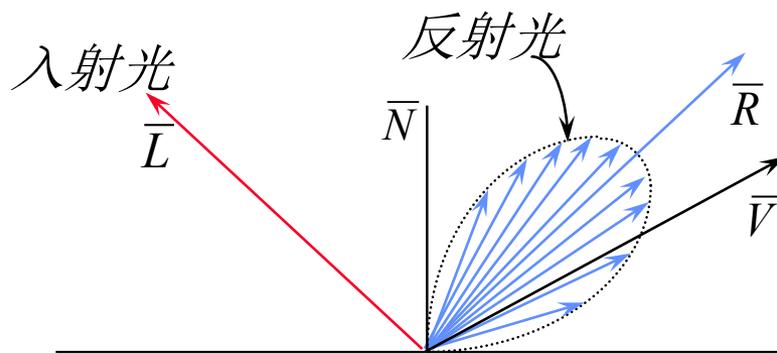


Phong光照模型

- Phong光照模型由以下两种反射组成：
 - 漫反射
 - 镜面反射
- 因此，照明效果依赖于表面属性和光源自身
 - 光源属性
 - 材质属性

Phong 光照模型

- 由Phong Bui-Tuong (1975)提出，是一种广泛应用的非完美反射模型
- 考虑了光亮物体的镜面反射，假定最大镜面反射出现在 $\alpha = 0$ 的方向上
- 光照计算依赖于观察方向
- 反射光强度由以下几项组成：
 - 环境光分量
 - 漫反射分量
 - 镜面反射分量

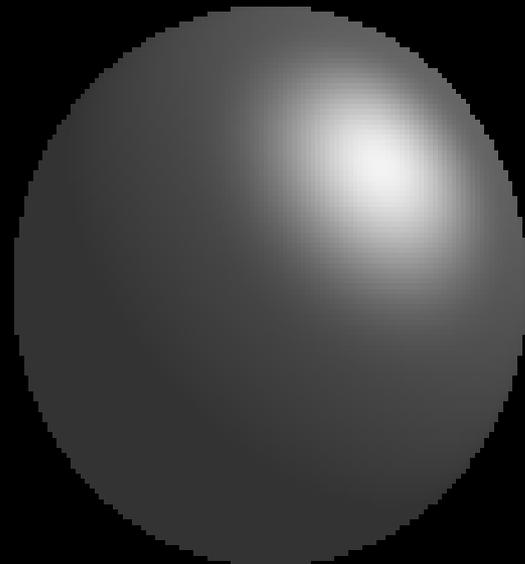


Phong 光照模型

- 光照方程的最简形式如下：

$$I = I_a k_a + I_p k_d (\bar{N} \cdot \bar{L}) + I_p k_s (\bar{R} \cdot \bar{V})^n$$

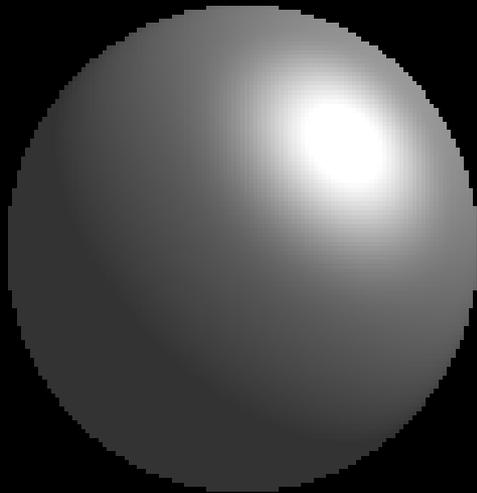
- 这里， k_s 是材质的 *镜面反射系数*，取值在0 和 1之间， n 是 *镜面反射指数*，是物体的材质属性
- \bar{R} 是反射光向量
- \bar{V} 是观察向量



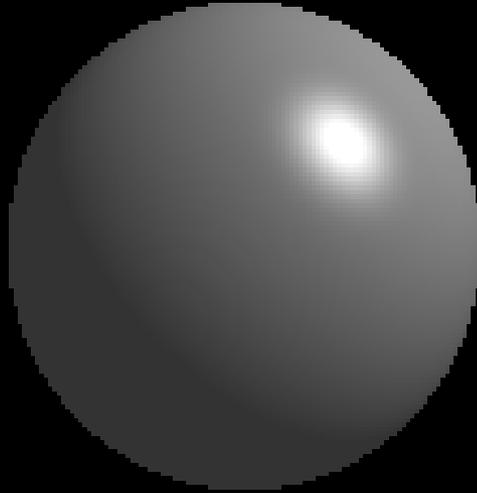
$$k_a = 0.2, \quad k_d = 0.2, \quad k_s = 0.6, \quad n = 4 \quad \llcorner$$

Phong 光照模型

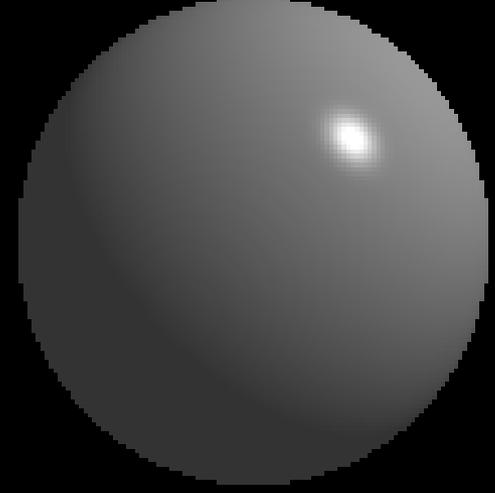
- 镜面反射指数 n 控制了表面的光洁程度. 当 n 增加时, 物体更加光滑, 高光区域也聚焦



$n = 4$

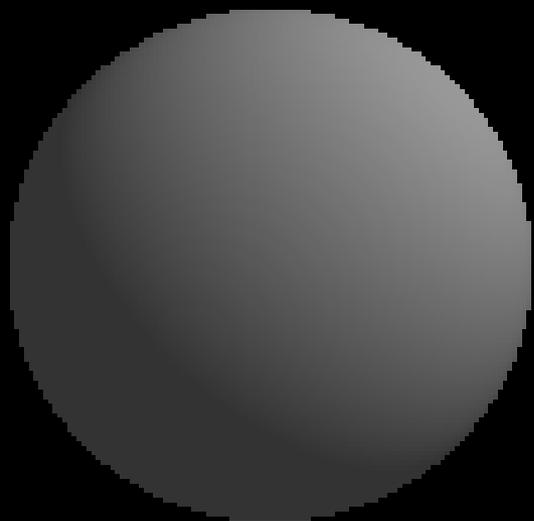


$n = 16$

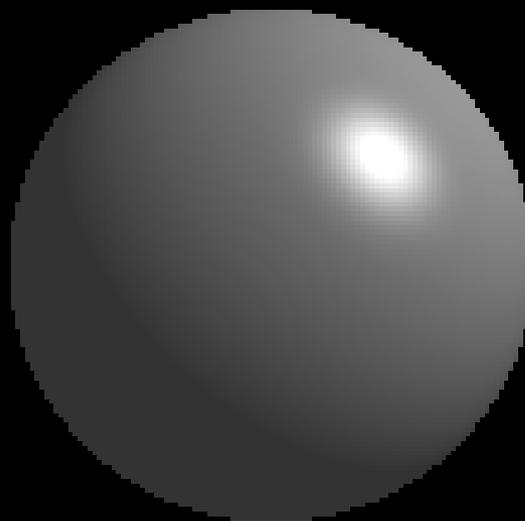


$n = 64$

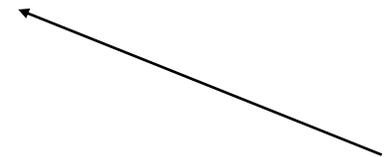
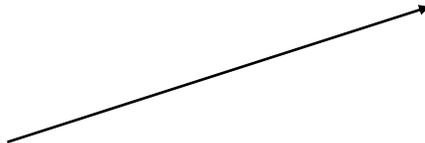
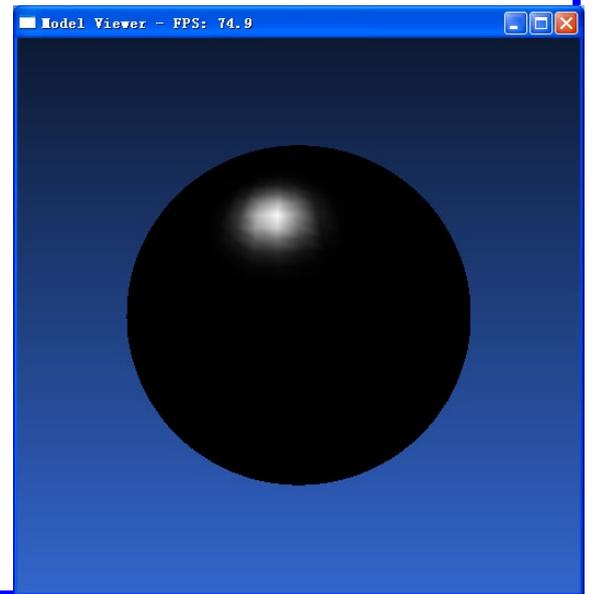
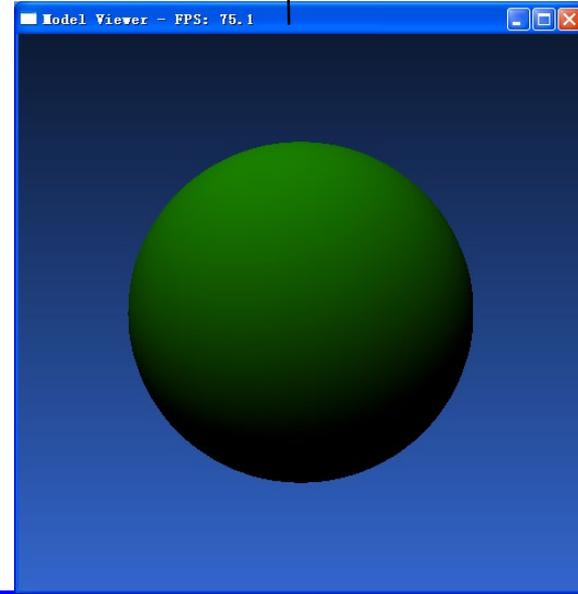
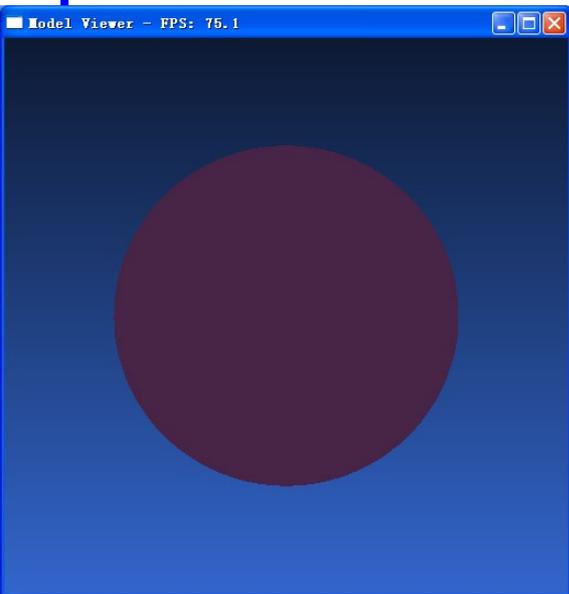
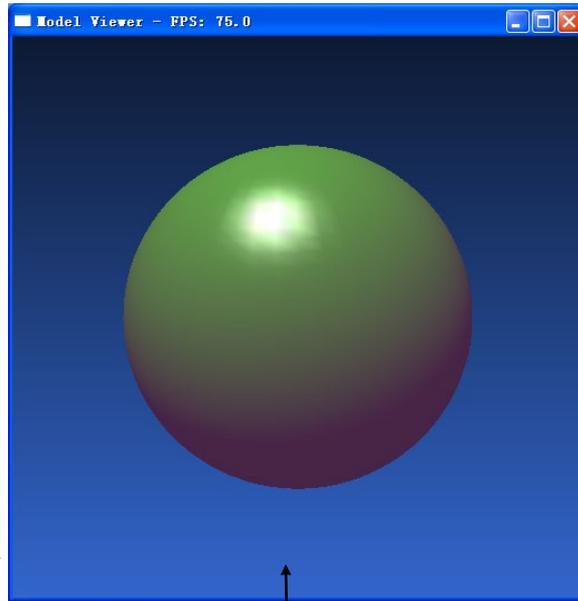
光照模型对比



漫反射光照模型



Phong光照模型



多光源照明

- 对于多光源的情况，需要对每个光源的贡献进行累加，于是 m 个光源的光照方程如下：

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

- 进行累加后，可能会出现 I 超出像素可展示最大值的情况：
 - 通过调整材质属性进行避免
 - 将 I 进行最大值截断
 - 将每个像素值除以 I 的峰值

彩色物体

- 物体的颜色由环境光反射系数和漫反射系数决定
- 镜面反射系数并不由颜色所决定
- 照明方程变成以下三个方程式：

$$I_r = I_{ar}k_{ar} + I_{pr}[k_{dr}(\bar{N} \bullet \bar{L}) + k_{sr}(\bar{R} \bullet \bar{V})^n]$$

$$I_g = I_{ag}k_{ag} + I_{pg}[k_{dg}(\bar{N} \bullet \bar{L}) + k_{sg}(\bar{R} \bullet \bar{V})^n]$$

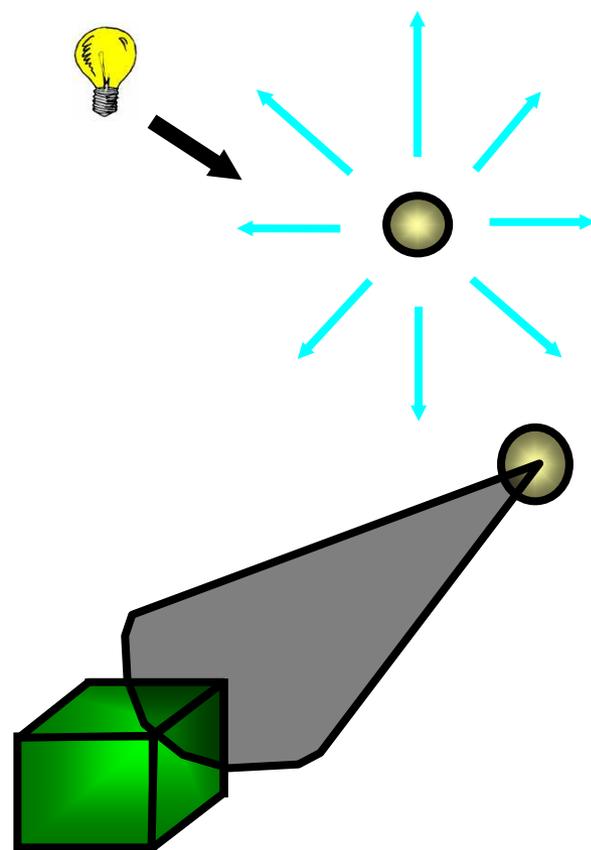
$$I_b = I_{ab}k_{ab} + I_{pb}[k_{db}(\bar{N} \bullet \bar{L}) + k_{sb}(\bar{R} \bullet \bar{V})^n]$$

- 或者缩写为以下形式：

$$I(r, g, b) = I_a k_a(r, g, b) + I_p [k_d(r, g, b)(\bar{N} \bullet \bar{L}) + k_s(\bar{R} \bullet \bar{V})^n]$$

OpenGL 照明模型

- OpenGL 照明模型较为简单
- 光源类型包括：
 - 环境光源 – 光被多次反射以至于看不出来自何处，对各个方向的照明强度相同
 - 点光源 – 朝所有方向发射光线，相对于场景中的物体，大小可忽略
 - 筒光源 – 在一定角度范围内发射光线



OpenGL中的光源

- 大部分OpenGL实现支持场景中最多有8个光源
 - 每个光源可以有一个漫反射分量和一个镜面反射分量
 - 每个光源也可以有环境光分量 (光照在许多表面, 但是无法判断其来源)
 - 光源由以下宏来访问:
GL_LIGHT0, GL_LIGHT1, ..., GL_LIGHT7
 - 可以通过调用函数“glLightfv”来设置光源的属性

OpenGL中的光源

- glLightfv(光源编号, 属性, 值数组)
 - 光源编号 = GL_LIGHT*i* 之一
 - 属性, 以下值之一:
 - GL_AMBIENT
 - GL_DIFFUSE
 - GL_SPECULAR
 - GL_POSITION = 光源的位置
 - 值数组 = 属性值

色调

- 色调是使用某个光照模型决定表面所有像素颜色的过程
- 最简单的方法为：
 - 判断表面每个像素是否可见
 - 计算表面法向量
 - 使用光照模型计算光照强度或颜色
- 这个方法过于昂贵
- 色调算法可用于对特定曲面表示进行高效计算

色调模型

- 色调模型给出了一种使用光照模型来计算表面所有像素颜色的技术
- 多边形网格通常被用于复杂曲面表示
- 几何信息只在多边形的顶点处可获得
- 插值型色调模型可用于大幅度提升效率

Constant 色调

- 这是最简单的色调模型，也被称为 *faceted 色调* 或 *flat 色调*
- 一个多边形只计算一个光照强度值
- 光照模型对每个多边形只使用一次
- 基于以下假定：
 - 光源在无穷远处，使得 $\bar{N} \bullet \bar{L}$ 在多边形内部为常量
 - 观察者在无穷远处，使得 $\bar{R} \bullet \bar{V}$ 在多边形内部为常量
 - 使用了多边形来表示模型表面

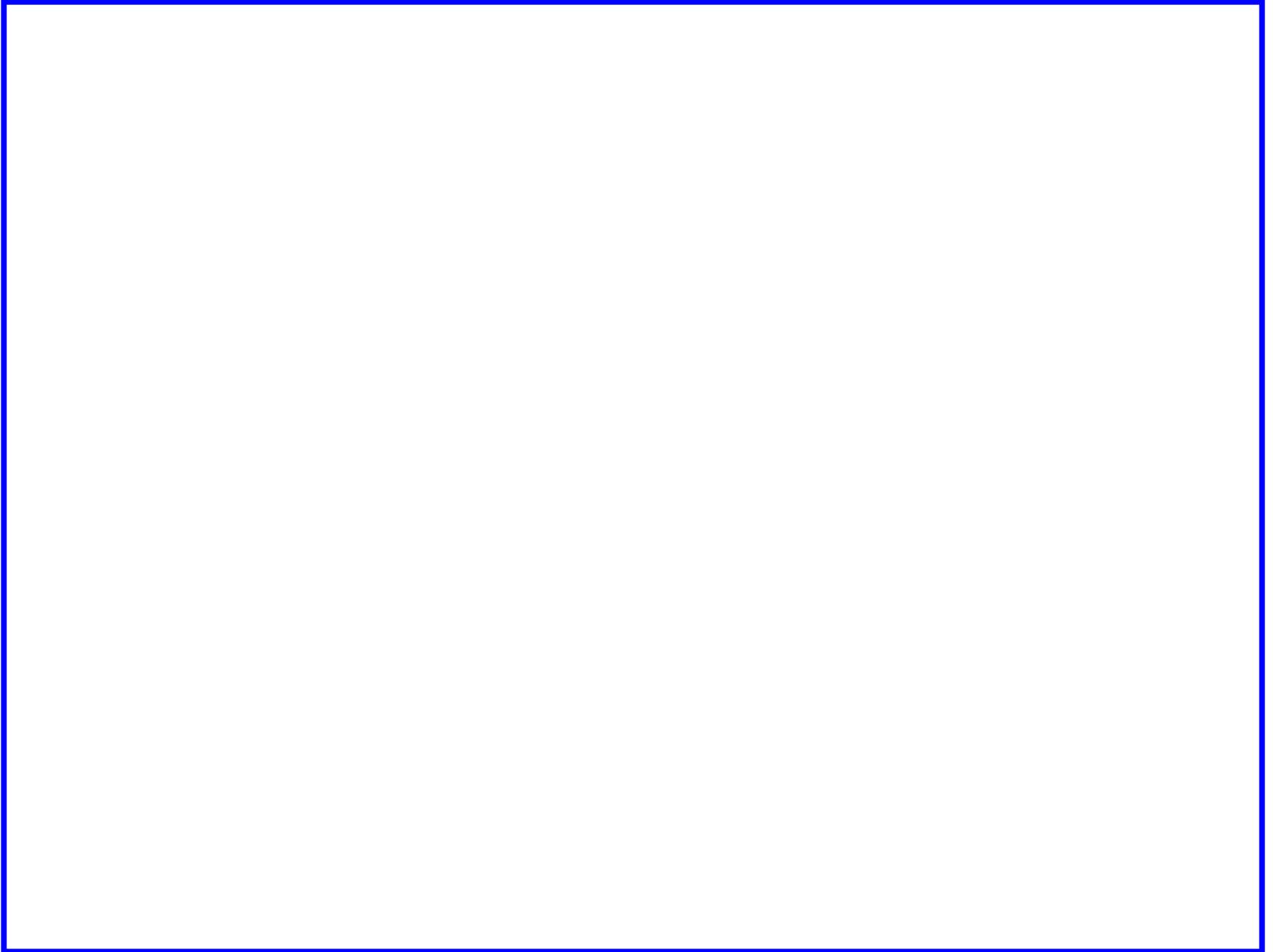
平面着色 (flat shading)

- 这是最快的色调算法，因为计算非常少
- 如果投影在屏幕上的多边形非常小（比如说一个像素大），那么结果和其他插值型技术一样好
- 通常用于生成场景的粗略预览

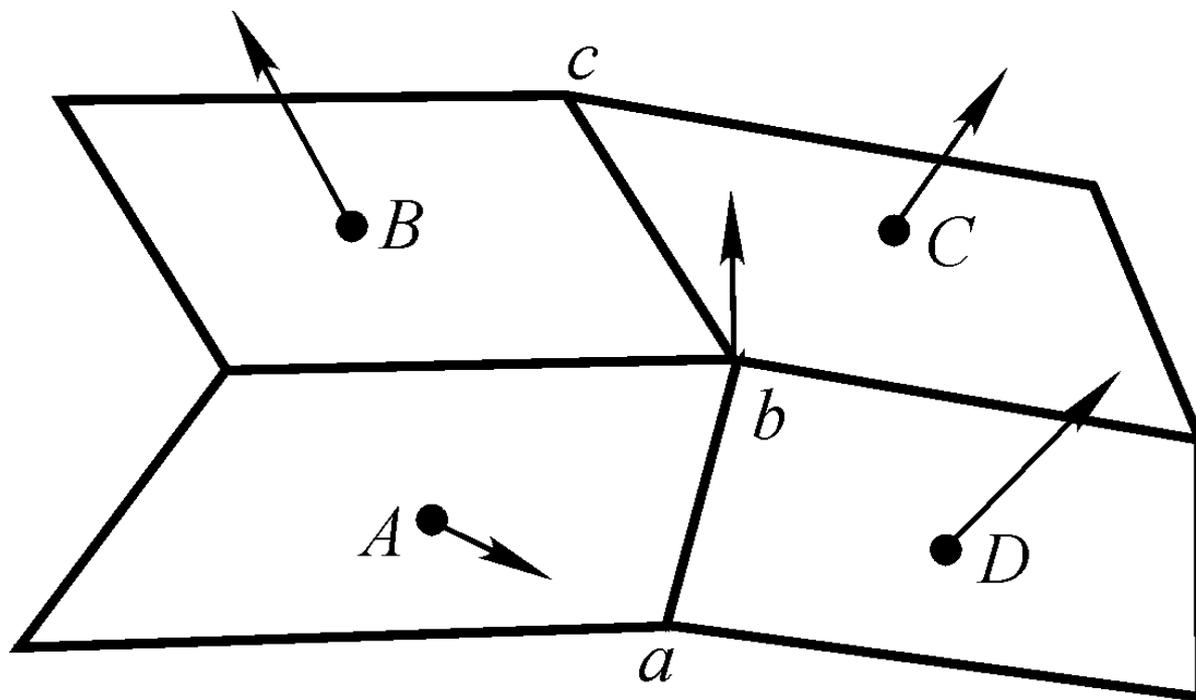


OpenGL使用了第一个顶点的法向量来计算单个多边形的颜色：

```
glShadeModel (GL_FLAT) ;
```



着色的法向计算



Gouraud 着色 (Gouraud shading)

- 这是一种插值型着色算法，也被称为 *强度插值着色* 或 *颜色插值着色*
- 由以下几步组成：
 - 首先计算顶点的法向量：通过对共享该顶点的所有面的法向量求和平均得到
 - 使用顶点法向量和光照模型来计算每个顶点的光照强度
 - 对每个多边形，内部像素的光照强度通过对顶点光照强度进行线性插值计算得到

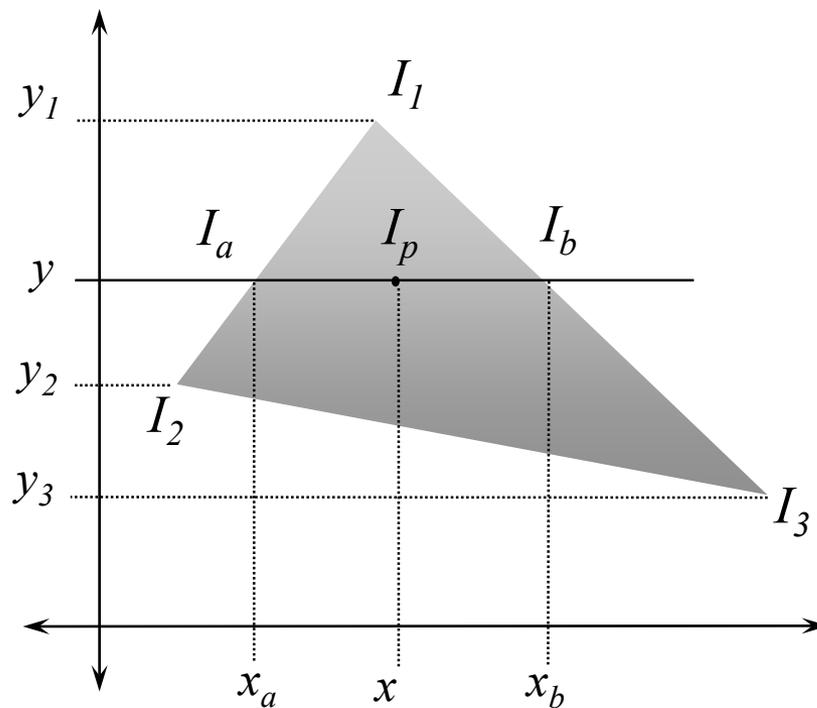
Gouraud 着色

- 某内部点光照强度 I_p 如下计算：

$$I_a = I_1 + (I_2 - I_1) \frac{y - y_1}{y_2 - y_1}$$

$$I_b = I_1 + (I_3 - I_1) \frac{y - y_1}{y_3 - y_1}$$

$$I_p = I_a + (I_b - I_a) \frac{x - x_a}{x_b - x_a}$$



Gouraud 着色

- 当高光区域没有落在顶点上时，可能会造成丢失

```
glShadeModel (GL_SMOOTH) ;
```



Phong 着色 (Phong shading)

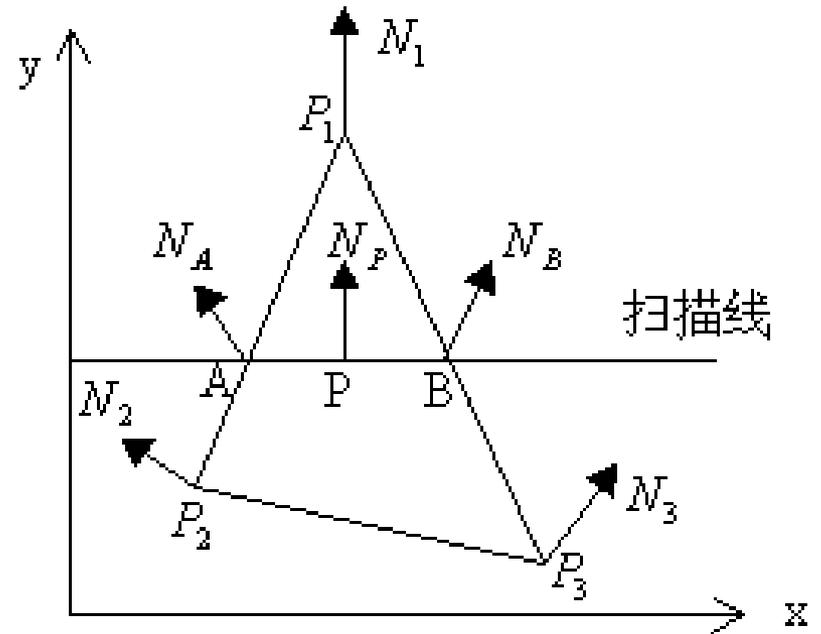
- 这是一种插值型着色算法，也被称为*法向量插值着色*由以下几步组成：
 - 首先计算顶点的法向量：通过对共享该顶点的所有面的法向量求和平均得到
 - 对每个多边形，内部像素的法向量通过对顶点法向量进行线性插值计算得到
 - 插值得到的法向量代入到光照模型计算光照强度
- 镜面反射可以较好的捕捉到
- 法向量插值的过程和Gouraud 着色中的光照强度插值完成相同

Phong 着色

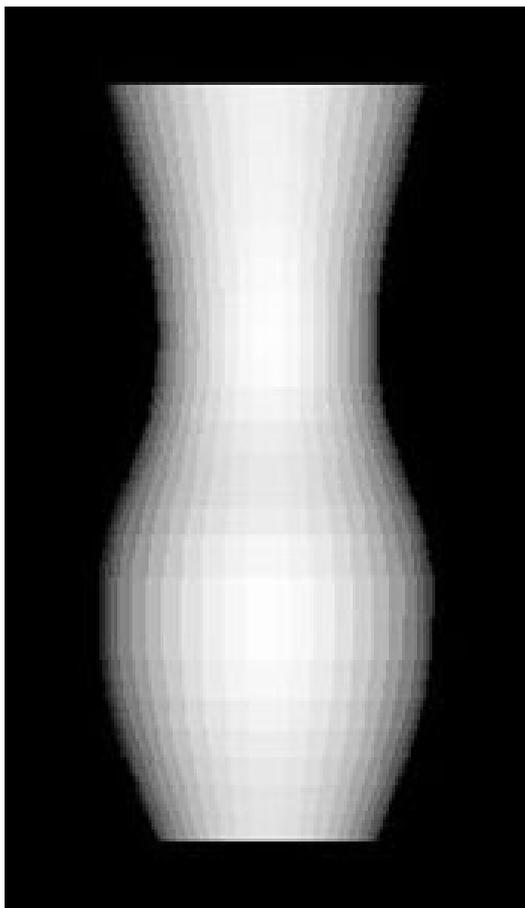
$$N_A = \frac{y_A - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y_A}{y_1 - y_2} N_2$$

$$N_B = \frac{y_B - y_3}{y_1 - y_3} N_1 + \frac{y_1 - y_B}{y_1 - y_3} N_3$$

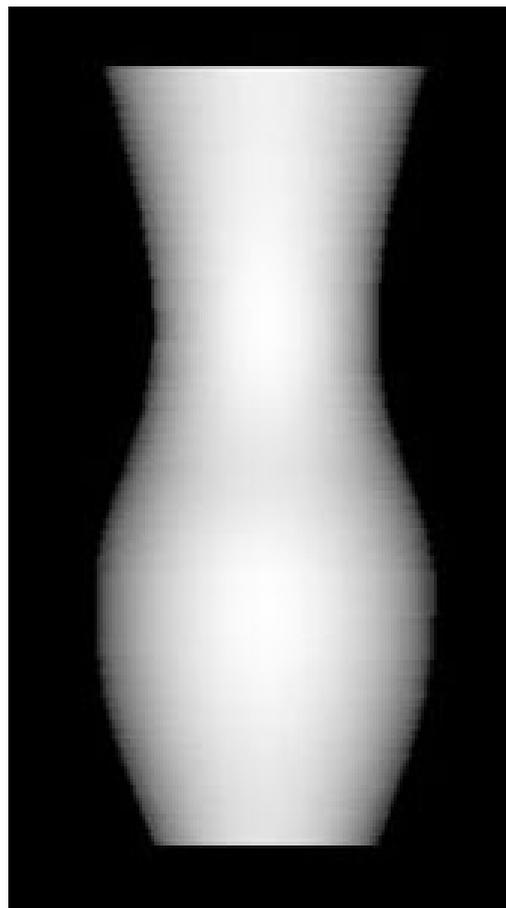
$$N_P = \frac{x_B - x}{x_B - x_A} N_A + \frac{x - x_A}{x_B - x_A} N_B$$



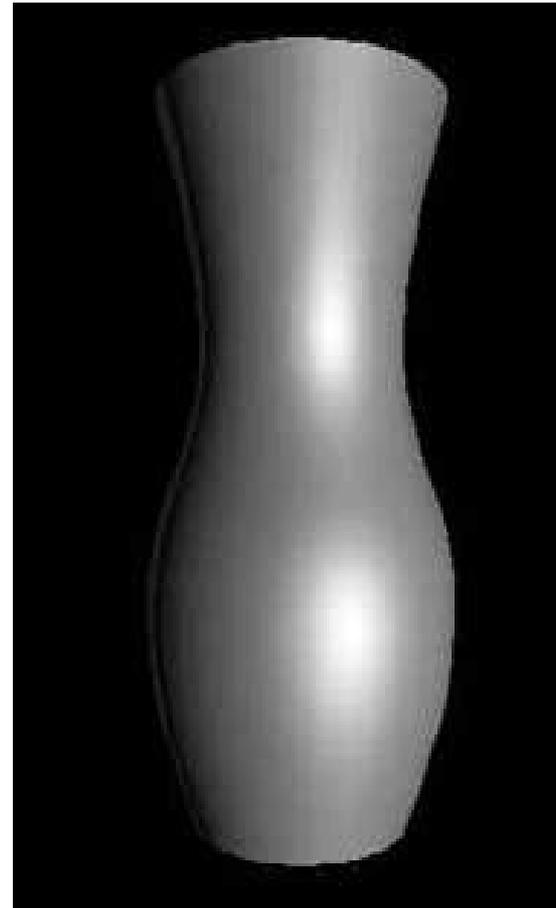
Flat 着色



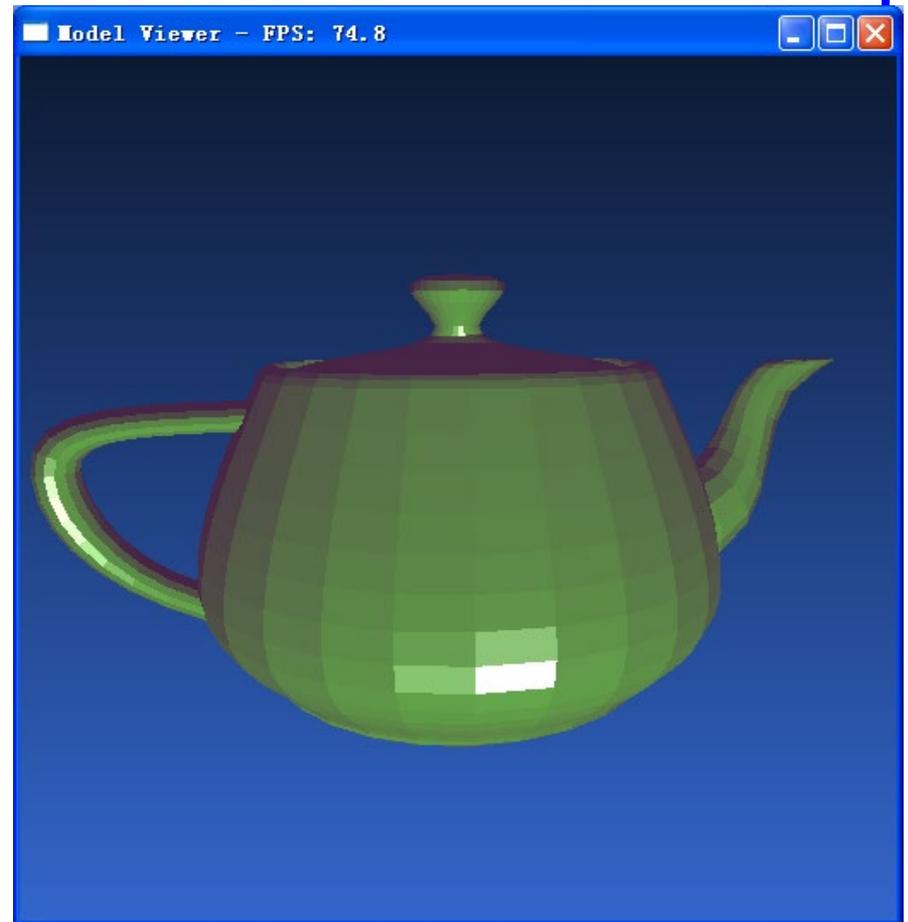
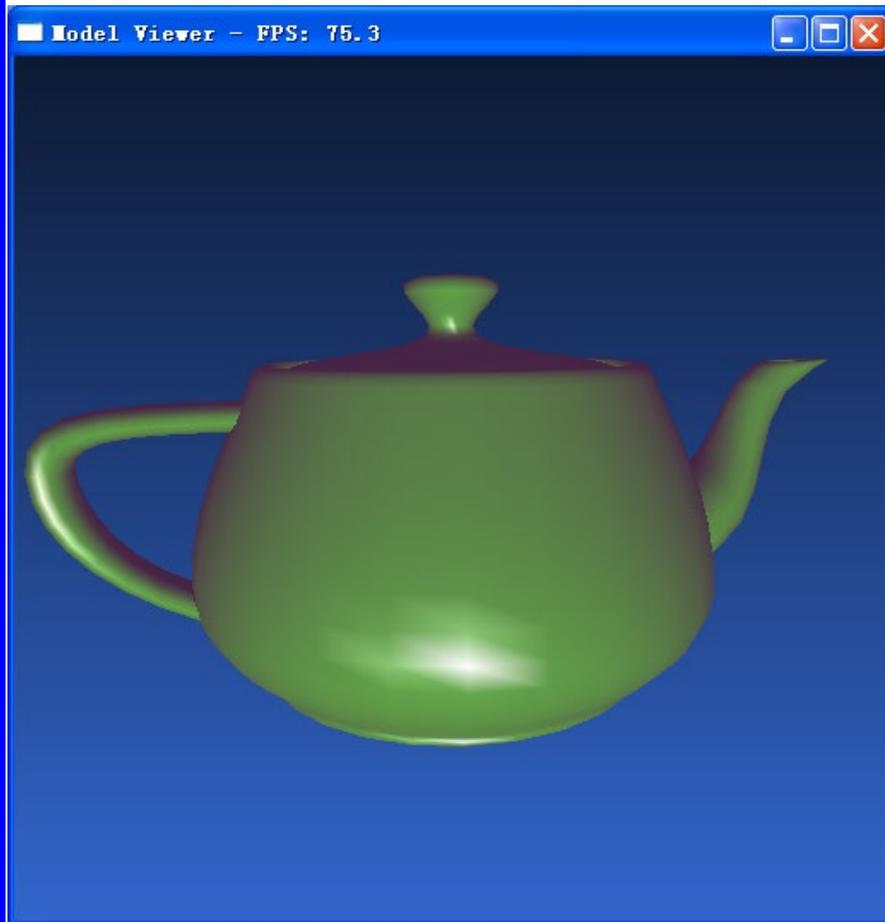
Gouraud 着色



Phong 着色

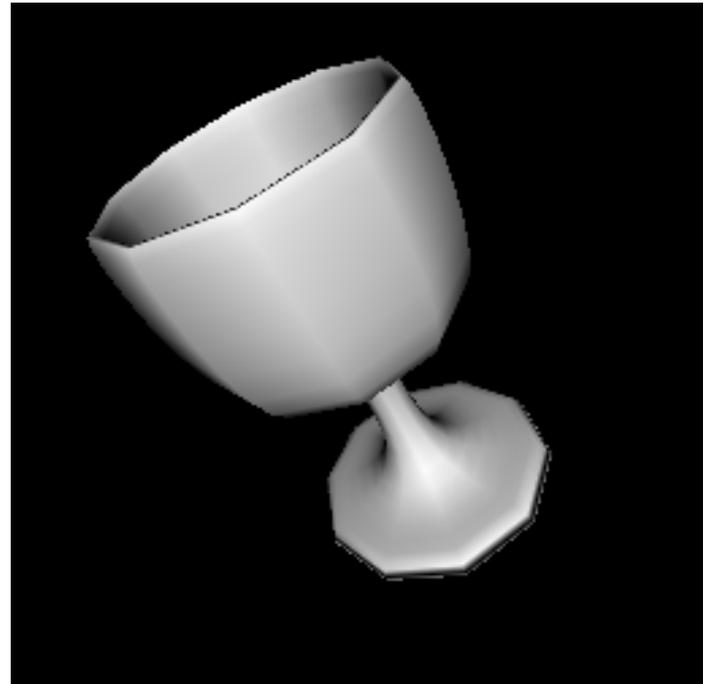


Demo – OpenGL着色模型



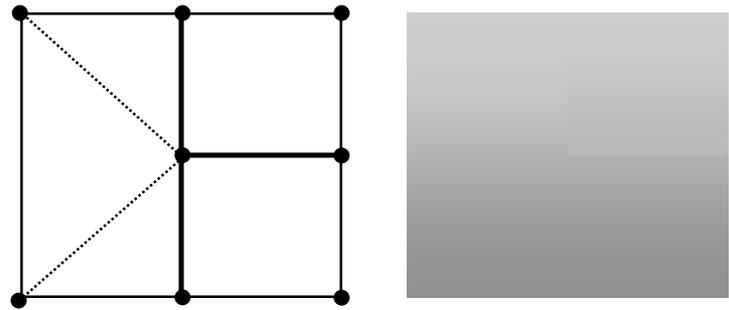
插值型着色算法的问题

- 多边形轮廓线：网格的轮廓线始终是多边形
- 解决思路：
 - 对整个表面进行细分
 - 只沿着轮廓线进行细分
(视线相关)

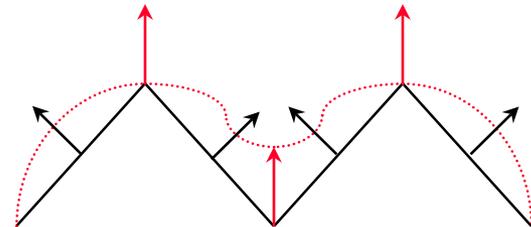


插值型着色算法的问题

- 共享顶点做成的问题
 - 避免这些情况

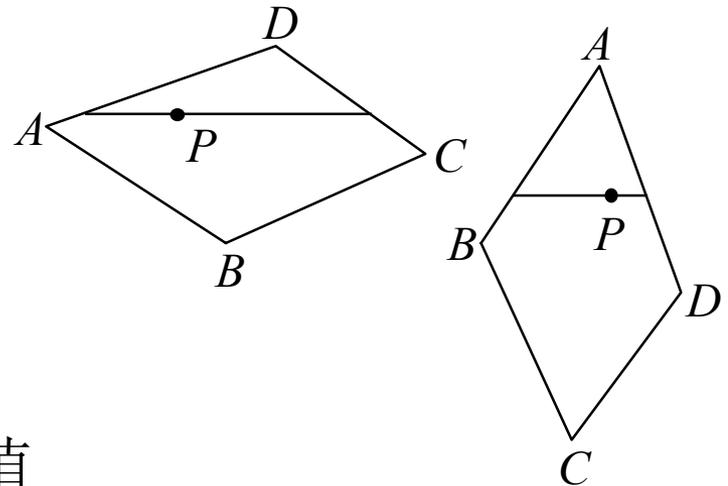
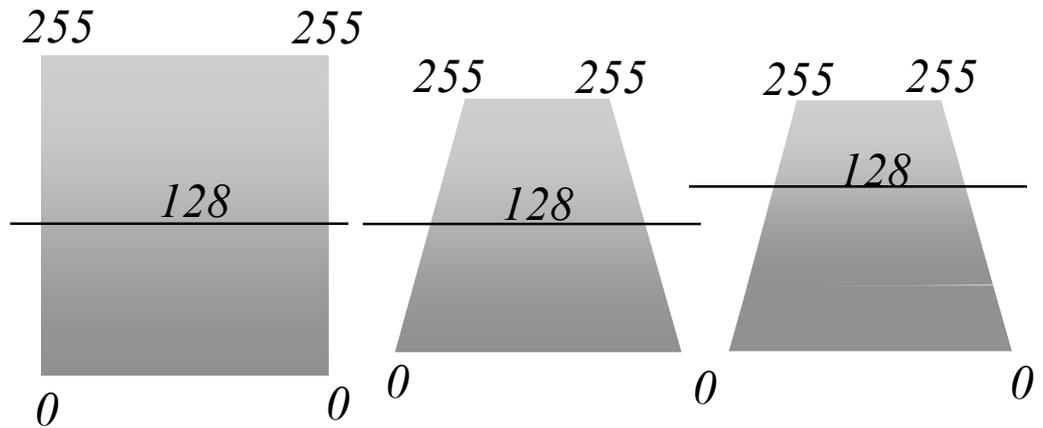


- 未表示的顶点法向量
 - 在计算法向量前对多边形进行细分



插值型着色算法的问题

- 透视扭曲
- 解决:
 - 细化生成较小的多边形
 - 在世界坐标系进行插值
- 朝向相关
- 解决:
 - 将多边形分解为三角形
 - 在对象空间进行旋转不变插值



OpenGL中的照明

OpenGL中指定照明的步骤:

- 定义物体每个顶点的法向量
- 创建、定位、激活一个或多个光源
- 选择一个照明模型
- 定义场景中物体的材质属性

指定OpenGL材质

通用形式: `glMaterialf(face, parameter, value);`
`glMaterialfv(face, parameter, *array);`

`face` 可以是 `GL_FRONT`, `GL_BACK`, `GL_FRONT_AND_BACK`

`parameter` 为:

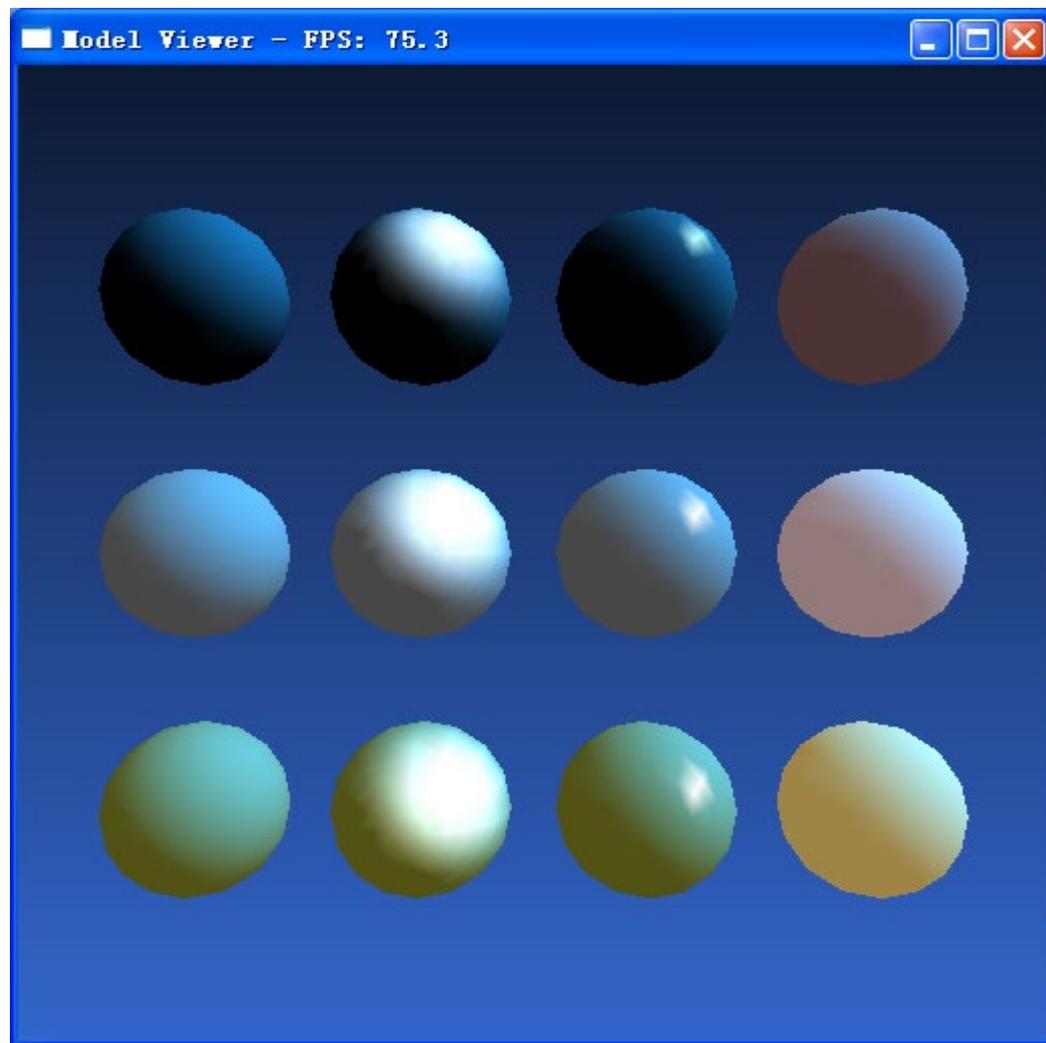
`GL_AMBIENT` 材质的环境光反射系数. (0.2, 0.2, 0.2, 1.0)

`GL_DIFFUSE` 材质的漫反射系数. (0.8, 0.8, 0.8, 1.0)

`GL_SPECULAR` 材质的镜面反射系数. (0.0, 0.0, 0.0, 1.0)

`GL_SHININESS` 材质的镜面反射指数. 0.0

Demo – 多种材质



谢谢！