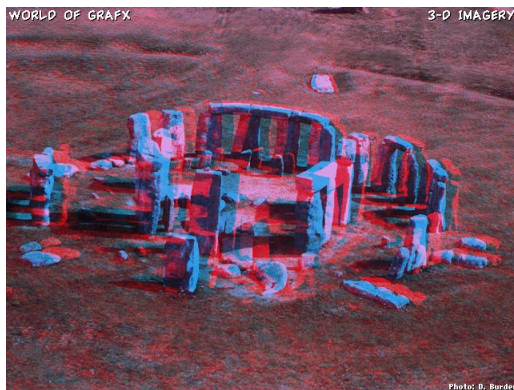
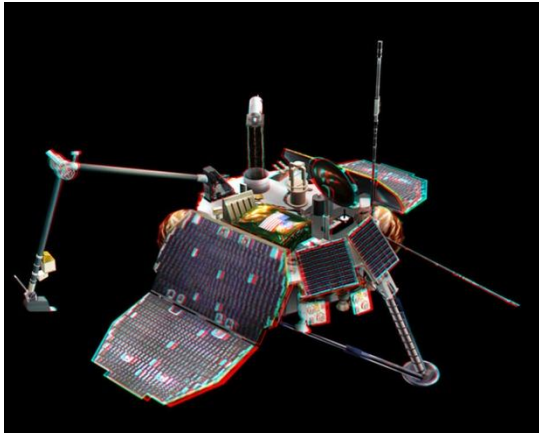

第十二章 光照模型与真实感图形

□ 主要内容:

- 本章主要讲述简单光照明模型，整体光照明模型及其区别
 - Phong模型，环境光，漫反射，镜面反射
 - 均匀着色方法，Gouraud着色方法，Phong着色方法，
 - 物体的表面细节，纹理映射，法向扰动法
 - 光线跟踪算法
 - 绘制真实感图形的流程图，细节多边形。
-

□ 掌握要点：

- 掌握简单光照明模型与整体光照明模型的特点和区别；
 - 掌握Phong模型，包括环境光，漫反射，镜面反射三分量的意义和计算，包括如何产生彩色，多个点光源条件的计算；
 - 掌握简单的多边形着色方法，包括均匀着色方法，Gouraud着色方法，Phong着色方法及它们的特点；
 - 了解如何产生物体的边面细节，纹理的概念和分类
 - 了解如何产生阴影和透明效果；
 - 了解整体光照明模型和光线跟踪算法；
-



From 2D
to 3D

在三维软件中绘图步骤

- 建立几何模型
 - 对每个面片设定材质和贴图
 - 设立光照
 - 设置摄像机位置，方向，焦距等参数
 - 绘制
-

12.1 简单光照明模型

- 当光照射到物体表面时，光线可能被吸收、反射和透射。
 - 被物体吸收的部分转化为热，反射、透射的光进入人的视觉系统，使我们能看见物体。
 - 为模拟这一现象，我们建立一些数学模型来替代复杂的物理模型，这些模型就称为明暗效应模型或者**光照明模型**。三维形体的图形经过消隐后，再进行明暗效应的处理，可以进一步提高图形的真实感。
-

光照明模型的早期发展情况：

- 1963年...
 - 1967年，Wylie等人第一次在显示物体时加进光照效果[WYLI67]。Wylie认为：物体表面上一点的光强，与该点到光源的距离成反比。
 - 1970年，Bouknight在Comm.ACM上发表论文，提出第一个光反射模型[BOUK70]，指出物体表面朝向是确定物体表面上一点光强的主要因素，用Lambert漫反射定律计算物体表面上各多边形的光强，对光照射不到的地方，用环境光代替。
-

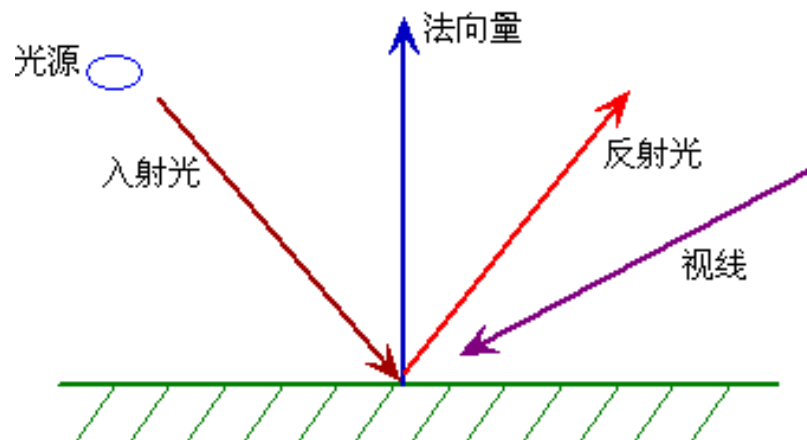
光照明模型的早期发展情况：

- 1971年，Gourand在IEEE Trans. Computers上发表论文[GOUR71]，提出漫反射模型加插值的思想。对多面体模型，用漫反射模型计算多边形顶点的光亮度，再用增量法插值计算。
 - 1975年，Phong在Comm. ACM上发表论文[PHON75]，提出图形学中第一个有影响的光照明模型。
 - Phong模型虽然只是一个经验模型，但是其真实度已达到可以接受的程度。
-

光照模型相关知识

□ 光的传播

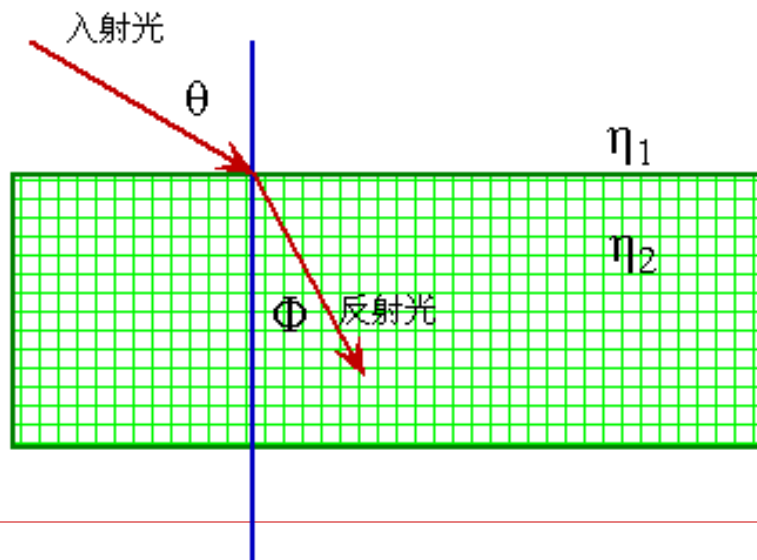
- 在正常的情况下，光沿着直线传播，当光遇到介质不同的表面时，会产生反射和折射现象，而且在反射和折射的时候，遵循反射定律和折射定律。
- **反射定律：**入射角等于反射角，且反射光线、入射光线与法向量在同一平面上。



□ 折射定理：折射角与入射角满足

$$\frac{\eta_1}{\eta_2} = \frac{\sin \varphi}{\sin \theta}$$

且折射线在入射线与法线构成的平面上。



□ **能量关系：**在光的反射和折射现象中，能量是守恒的，能量的分布情况满足这样的一个式子：
 $I_i = I_d + I_s + I_t + I_v$ 。其中：

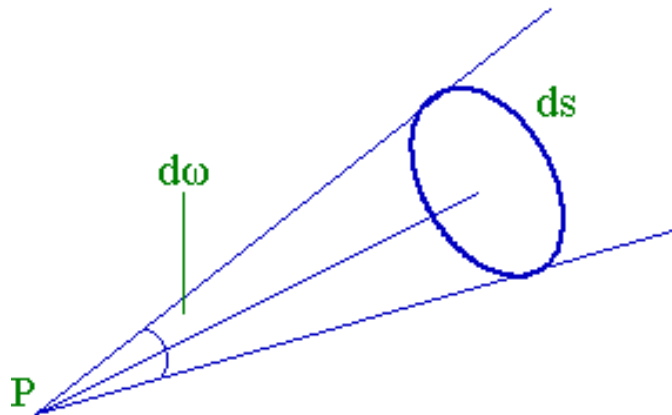
- I_i 为入射光强，由直接光源或间接光源引起；
 - I_d 为漫反射光强，由表面不光滑引起；
 - I_s 为镜面反射光强，由表面光滑性引起；
 - I_t 为透射光，由物体的透明性引起；
 - I_v 为被物体所吸收的光，由能量损耗引起。
-

□ 光的度量

■ 立体角：面元 ds 向点源 P 所张的立体角 $d\omega$ 为：

$$d\omega = \frac{ds}{r^2}$$

其中 r 为点光源到面元中心的垂直距离。



□ 点发光强度

- 光通量：单位时间内通过面元 ds 的光能量，记为 dF 。
- 发光强度：点光源在某个方向上的发光强度，定义为该方向上单位立体角内的光通量，即：

$$I = \frac{dF}{d\omega} = \frac{dF}{ds} \cdot r^2$$

- 各向同性的点光源，在各个方向上单位立体角内通过的光通量相等，即在各个方向上发光强度相等。设发光强度为 I ，则点光源向外辐射的整个光通量为球立体角内的光通量

$$F = \int_0^{4\pi} I d\omega = 4\pi \cdot I$$

简单光照明模型与整体光照明模型

- 光照到物体表面时，物体对光会发生反射(Reflection)、透射(Transmission)、吸收(Absorption)。
 - 首先我们先来介绍对于光反射现象的研究。
 - **简单光照明模型**模拟物体表面对光的反射作用。光源被假定为点光源，反射作用被细分为镜面反射(Specular Reflection)和漫反射(Diffuse Reflection)。
 - **简单光照明模型**只考虑物体对直接光照的反射作用，而物体间的光反射作用，只用环境光(Ambient Light)来表示
 - **Phong光照明模型**就是这样的一种模型。
-

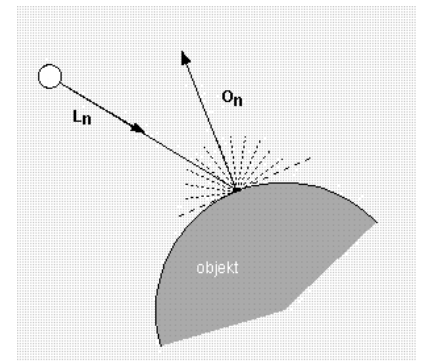
§ 12.1.2 Phong光照模型

□ 理想漫反射

- 当光源来自一个方向时，漫反射光均匀向各方向传播，与视点无关，它是由表面的粗糙不平引起的，漫反射光的空间分布是均匀的。
- 记入射光强为 I_p ，物体表面上点 P 的法向为 N ，从点 P 指向光源的向量为 L ，两者间的夹角为 θ ，由Lambert余弦定律，则漫反射光强为：

$$I_d = I_p * K_d * \cos(\theta) , \theta \in (0, \pi/2) ,$$

K_d 是与物体有关的漫反射系数, $0 < K_d < 1$



-
- 当 L 、 N 为单位向量时，上式也可用如下形式表达：

$$I_d = I_p K_d * (L \cdot N)。$$

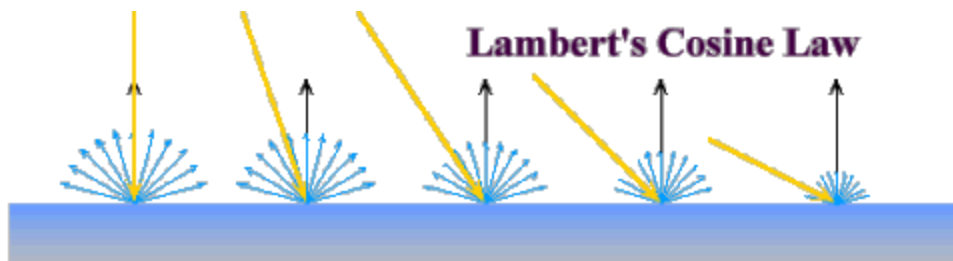
- 在有多个光源的情况下，可以有如下的表示：

$$I_d = K_d \sum I_{pi} * (L_i \cdot N)。$$

- 考虑：入射角为何时，点光源才照亮面片？入射角为何时，光亮度为0？入射角为何时，反射光强度最大？

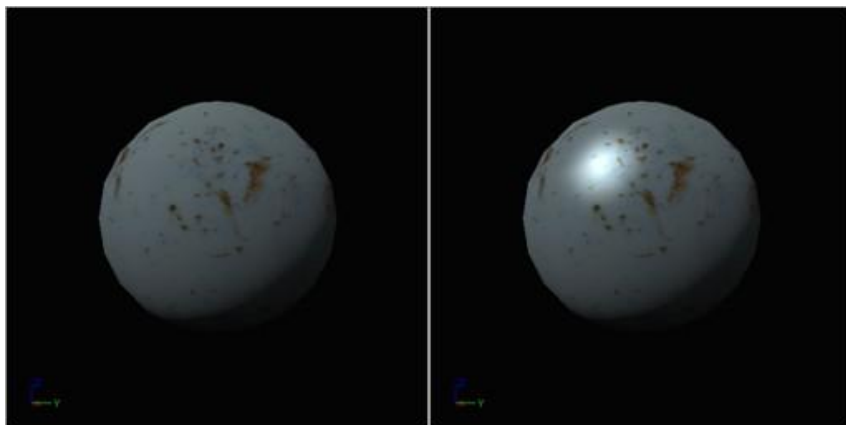


-
- 在RGB颜色模型下，漫反射系数 Kd 有三个分量 Kdr, Kdg, Kdb 分别代表RGB三原色的漫反射系数，通过调整它们，可以改变物体的颜色。
 - 同样，我们也可以把入射光强 I 设为三个分量 Ir, Ig, Ib ，通过这些分量的值来调整光源的颜色。



□ 镜面反射光

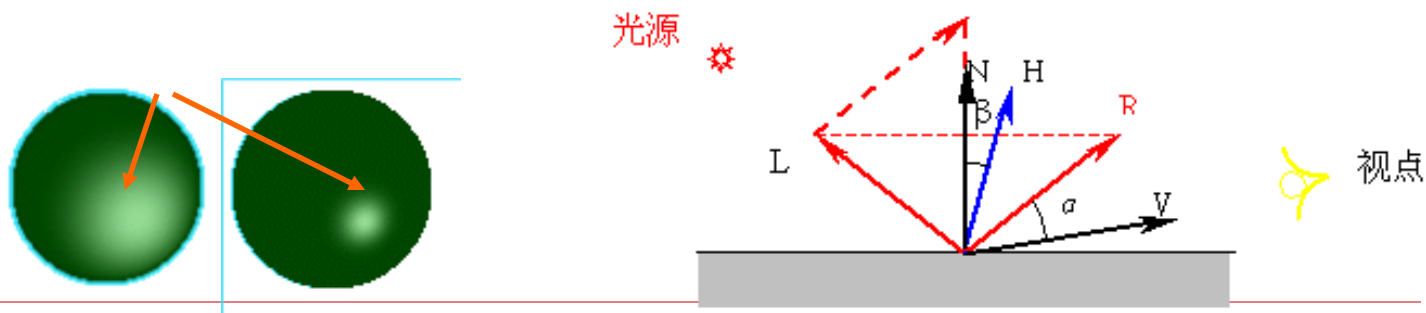
- 对于理想镜面，反射光集中在一个方向，并遵守反射定律。对一般的光滑表面，反射光集中在一个范围内，且由反射定律决定的反射方向光强最大。
- 对于表面上同一点来说，从不同位置所观察到的镜面反射光强是不同的。



- Phong模型中，镜面反射光强可表示为：

$$I_s = I_p * K_s * \cos^n(\alpha), \alpha \in (0, \pi/2)$$

- 其中 K_s 是与物体有关的镜面反射系数， α 为视线方向 V 与反射方向 R 的夹角， n 为反射指数，反映了物体表面的光泽程度，一般为1~2000，数目越大物体表面越光滑。镜面反射光将会在反射方向附近形成很亮的光斑，称为高光现象。



- 同样，将 V 和 R 都格式化为单位向量，镜面反射光强可表示为：

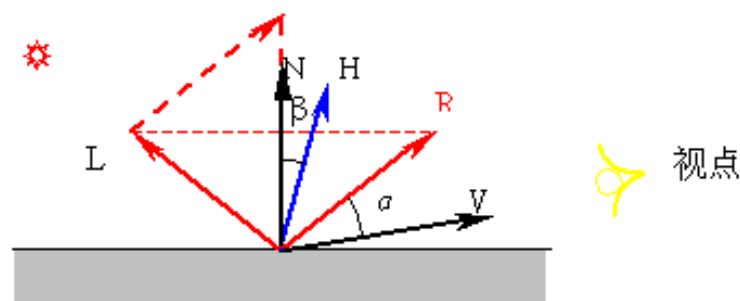
$$I_s = I_p \cdot K_s (R \cdot V)^n ,$$

- 其中 R 的计算方式如下：

$$R = N \cdot 2 \cos \theta - L = 2N(N \cdot L) - L$$

- 对于多个光源的情形，
镜面反射光强可表示为：

$$I_s = K_s \sum_{i=1}^m [I_{pi} \cdot (R_i \cdot V)^n]$$



□ 环境光

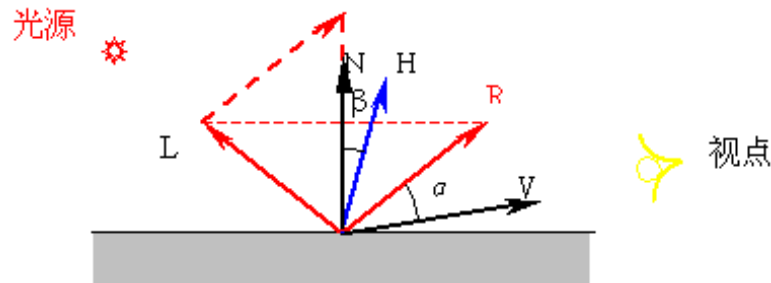
- 环境光是指光源间接对物体的影响，是在物体和环境之间多次反射，最终达到平衡时的一种光。我们近似地认为同一环境下的环境光，其光强分布是均匀的，它在任何一个方向上的分布都相同。例如，透过厚厚云层的阳光就可以称为环境光。
 - 在简单光照明模型中，我们用一个常数来模拟环境光，用式子表示为： $I_e = I_a \cdot K_a$ 。其中： I_a 为环境光的光强， K_a 为物体对环境光的反射系数。
-

□ Phong光照模型

- 综合上面介绍的光反射作用的各个部分，**Phong**光照模型有这样的一个表述：由物体表面上一点 P 反射到视点的光强 I 为环境光的反射光强 I_e 、理想漫反射光强 I_d 、和镜面反射光 I_s 的总和，即：

$$I = I_a K_a + I_p K_d (L \cdot N) + I_p K_s (R \cdot V)^n$$

a: ambient
d: diffuse
s: specular



□ Phong光照明模型的实现

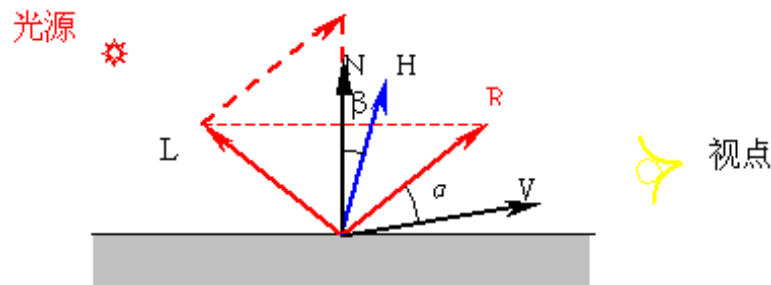
- 在用Phong模型进行真实感图形计算时，对物体表面上的每个点 P ，均需计算光线的反射方向 R ，再由 V 计算 $(R \cdot V)$ 。为减少计算量，我们可以作如下假设：

- 光源在无穷远处。即光线方向 L 为常数；

- 视点在无穷远处，即视线方向 V 为常数；

- 用 $(H \cdot N)$ 近似 $R \cdot V$ ，这里 H 为 L 和 V 的平分向量。 $H = \frac{L + V}{|L + V|}$

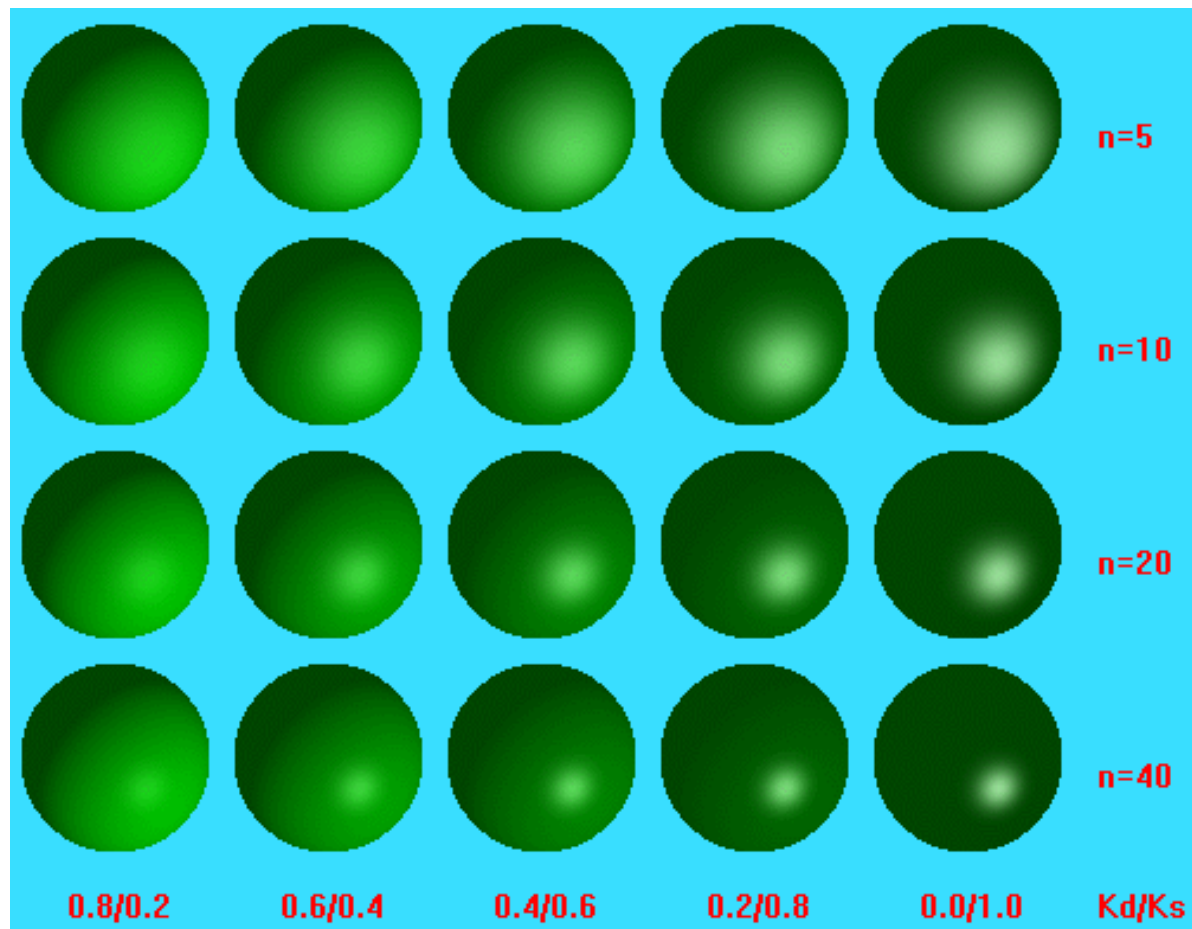
- 在这种简化下。由于对所有的点只需计算一次 H 的值，节省了计算时间。



-
- 结合RGB颜色模型，Phong光照模型最终有如下的形式：

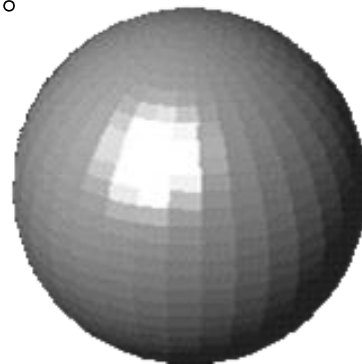
$$\begin{cases} I_r = I_{ar}K_{ar} + I_{pr}K_{dr}(L \cdot N) + I_{pr}K_{sr}(H \cdot N)^n \\ I_g = I_{ag}K_{ag} + I_{pg}K_{dg}(L \cdot N) + I_{pg}K_{sg}(H \cdot N)^n \\ I_b = I_{ab}K_{ab} + I_{pb}K_{db}(L \cdot N) + I_{pb}K_{sb}(H \cdot N)^n \end{cases}$$

-
- Phong光照模型是真实感图形学中提出的第一个有影响的光照模型，生成图象的真实度已经达到可以接受的程度；但是在实际的应用中，由于它是一个经验模型，还具有以下的一些问题：
 - 用Phong模型显示出的物体象塑料，没有质感；
 - 环境光是常量，没有考虑物体之间相互的反射光；
 - 镜面反射的颜色是光源的颜色，与物体的材料无关；
 - 镜面反射的计算在入射角很大时会产生失真等。
 - 在后面的一些光照模型中，对上述的这些问题都作了一定的改进。
-



§ 12.1.3 多边形着色方法

- 在上一节介绍的Phong光照明模型中，由于光源和视点都被假定为无穷远，最后的光强计算公式就变为物体表面法向量的函数。
- 对于当今流行的显示系统中用多边形表示的物体来说，它们中的每一个多边形由于法向一致，因而多边形内部的像素的颜色都是相同的，而且在不同法向的多边形邻接处，不仅有光强突变，而且还会产生**马赫带效应**(人类视觉系统夸大具有不同常量光强的两个相邻区域之间的光强不连续性)。



-
- 为了保证多边形之间的光滑过渡，使连续的多边形呈现匀称的光强，我们将介绍**多边形着色方法**。
 - **多边形着色方法**的基本思想是在每一个多边形的顶点处计算合适的光照明强度或参数，然后在各个多边形内部进行均匀插值，得到多边形的光滑颜色分布。
 - 它包含两个主要的算法：双线性光强插值和双线性法向插值，又被分别称为**Gouraud**明暗处理和**Phong**明暗处理。
-

□ 双线性光强插值(Gouraud明暗处理)

- 双线性光强插值是由Gouraud于1971年提出的，又被称为Gouraud明暗处理，它先计算物体表面多边形各顶点的光强，然后用双线性插值，求出多边形内部区域中各点的光强。

□ 它的基本算法描述如下：

1. 计算多边形顶点的平均法向。
 2. 用Phong光照明模型计算顶点的平均光强。
 3. 插值计算离散边上的各点光强。
 4. 插值计算多边形内域中各点的光强。
-

□ 下面我们分别介绍算法中的每一个步骤。

1. 顶点法向计算

- 我们用与顶点相邻的所有多边形的法向的平均值近似作为该顶点的近似法向量。假设顶点 A 相邻的多边形有 k 个，法向分别为 N_1, N_2, \dots, N_k ，我们取顶点 A 的法向为：

$$N_a = \frac{1}{k}(N_1 + N_2 + \dots + N_k)$$

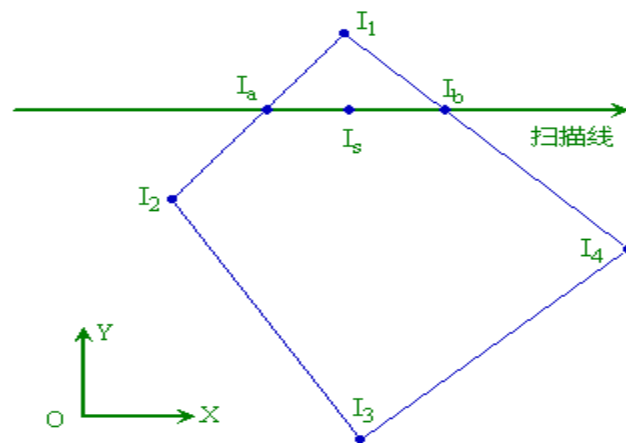
2. 顶点平均光强计算

- 在求出顶点 A 的法向 N_a 后，我们可以用Phong光照明模型计算在顶点处的光亮度。但是在Gourand提出明暗处理方法时，Phong模型还没有出现，他们采用的是：

$$I = I_a K_a + I_p K_d (L \cdot N_a) / (r + k)$$

3. 光强插值

- 用多边形顶点的光强进行双线性插值，可以求出多边形上各点和内部点的光强。在这个算法步骤中，我们把线性插值与扫描线算法相互结合，同时还用增量算法实现各点光强的计算。算法首先由顶点的光强插值计算各边的光强，然后由各边的光强插值计算出多边形内部点的光强，如图所示。



-
- 双线性光强插值的公式如下：

$$I_a = \frac{1}{y_1 - y_2} [I_1(y_s - y_2) + I_2(y_1 - y_s)]$$

$$I_b = \frac{1}{y_1 - y_4} [I_1(y_s - y_4) + I_4(y_1 - y_s)]$$

$$I_s = \frac{1}{x_b - x_a} [I_b(x_b - x_s) + I_a(x_s - x_a)]$$

- 如果我们采用增量算法，当扫描线 ys 由 j 变成 $j+1$ 时，新扫描线上的点 $(xa, j+1)$ 和 $(xb, j+1)$ 的光强，可以由前一条扫描线与边的交点 (xa, j) 和 (xb, j) 的光强作一次加法得到：

$$I_{a,j+1} = I_{a,j} + \Delta I_a,$$

$$I_{b,j+1} = I_{b,j} + \Delta I_b,$$

$$\Delta I_a = (I_1 - I_2) / (y_1 - y_2),$$

$$\Delta I_b = (I_1 - I_4) / (y_1 - y_4),$$

-
- 而在一条扫描线内部，横坐标 xs 由 xa 到 xb 递增，当 xs 由 i 增为 $i+1$ 时，多边形内的点 $(i+1, ys)$ 的光强可以由同一扫描行左侧的点 (i, ys) 的光强作一次加法得到，即：

$$I_{i+1,s} = I_{i,s} + \Delta I_s ,$$

$$\Delta I_s = \frac{1}{x_b - x_a} (I_b - I_a)$$

双线性光强插值(Gouraud方法)评价

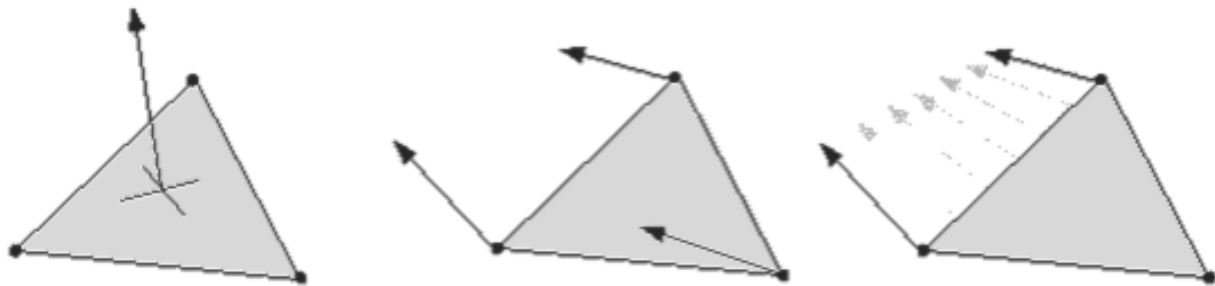
- 在双线性光强插值中，计算速度比以往的简单光照模型有了很大的提高，同时解决了相邻多边形之间的颜色突变问题，产生的真实感图象颜色过渡均匀，图形显得非常光滑，这是它的优点，
 - 但是，由于采用光强插值，它的镜面反射效果不太理想，而且相邻多边形的边界处的马赫带效应不能完全消除。
-

双线性法向插值(Phong方法)

- Phong提出的双线性法向插值以时间为代价，可以部分解决上述的弊病。双线性法向插值将镜面反射引进到明暗处理中，解决了高光问题。
 - 与双线性光强插值相比，该方法有如下特点：
 - 保留双线性插值，对多边形边上的点和内域各点，采用增量法。
 - 对顶点的法向量进行插值，而顶点的法向量，用相邻的多边形的法向作平均。
 - 由插值得到的法向，计算每个象素的光亮度。
 - 假定光源与视点均在无穷远处，光强只是法向量的函数。
-

□ Phong明暗处理的一般步骤:

- 计算每个多边形顶点的平均单位法矢量;
- 对多边形顶点的法矢量进行双线性插值, 得到多边形内每个点的法矢量;
- 根据光照模型沿每条扫描线计算多边形内各点对应的投影像素的光强度值。



-
- 双线性法向插值的式子与光强插值的式子基本类似，只不过是把其中的光强项用法向量项来代替，有如下的插值公式：

$$N_a = \frac{1}{y_1 - y_2} [N_1(y_s - y_2) + N_2(y_1 - y_s)]$$

$$N_b = \frac{1}{y_1 - y_4} [N_1(y_s - y_4) + N_4(y_1 - y_s)]$$

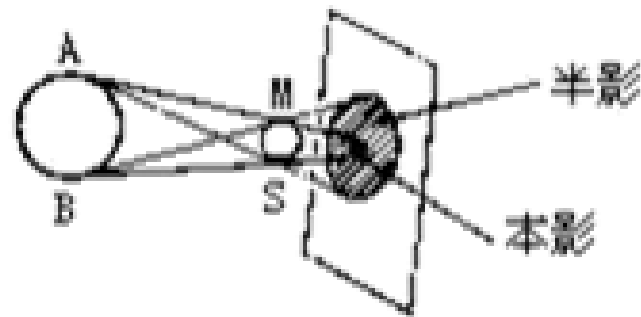
$$N_s = \frac{1}{x_b - x_a} [N_b(x_b - x_s) + N_a(x_s - x_a)]$$

- 同时，增量插值计算的公式也与光强插值公式相似，我们只要用法向代替光强即可，在这里我们就不在列出详细的公式了。
-

-
- 双线性光强插值可以有效的显示漫反射曲面，它的计算量小；而双线性法向插值与双线性光强插值相比，可以产生正确的高光区域，但它的计算量要大的多。
 - 这两个插值算法的增量式光照明模型本身也都存在着一些缺陷，具体表现为：用这类模型得到的物体边缘轮廓是折线段而非光滑曲线；由于透视的原因，使等间距扫描线产生不均匀的效果；插值结果决定于插值方向，不同的插值方向会得到不同的插值结果等。
-

§ 12.1.4 阴影的生成

- 阴影是现实生活中一个很常见的光照现象，它是由光源被物体遮挡而在该物体后面产生的较暗的区域。
- 在真实感图形学中，通过阴影可以反映出物体之间的相互关系，增加图形的立体效果和真实感。
- 我们知道，阴影的区域和形态与光源及物体的形状有很大的关系，在本小节中，我们只考虑由点光源产生的阴影，即阴影的**本影部分**。



-
- 当我们知道了物体的阴影区域以后，我们就可以把它结合到前面介绍的简单光照明模型中去：对于物体表面的多边形，如果在阴影区域内部，那么该多边形的光强就只有环境光那一项，后面的那几项光强都为零，否则就用正常的模型计算光强。
-

-
- 从原理上讲，计算阴影的本影部分是十分清楚简洁的：从阴影的产生原因上看，有阴影区域的物体表面都无法看见光源，我们只要把光源作为观察点，那么我们在前面介绍的任何一种隐藏面消除算法可以用来生成阴影区域。
-

□ 阴影多边形算法

- 1978年，Atherton等人提出了阴影多边形算法 [ATHE78]，在这个算法中第一次提出用隐藏面消除技术来生成阴影。把光源设为视点，这样物体的不可见面就是阴影区域，利用隐藏面消除算法就可以把可见面与不可见面区别开来。
 - 相对光源可见的多边形被称为阴影多边形，而不可见面就是非阴影多边形，这样非阴影多边形就处在物体多边形的阴影区域中。
-

□ 阴影域多面体算法

- 在物体空间中，按照阴影的定义，若光源照射到的物体表面是不透明的，那么在该表面后面就会形成一个三维的多面体阴影区域，该区域被称为阴影域（**Shadow Volume**），实际上，阴影域是一个以被光照面为顶面，表面的边界与光源所张的平面系列为侧面的一个半开三维区域，任何包含于阴影域内的物体表面必然是阴影区域。
 - 在我们的透视变换生成图象的过程中，屏幕视域空间常常是一个四棱椎，用这个四棱锥对物体的阴影域进行裁剪，那么裁剪后得到的三维阴影域就会变成封闭多面体，我们称其为阴影域多面体。
-

-
- 通过这种方法得到物体的阴影域多面体后，我们就可以利用它们来确定场景中的阴影区域，对于场景中的物体，只要与这些阴影域多面体进行三维布尔交运算，计算出的交集就可以被定为物体表面的阴影区域。
 - 该算法中涉及大量的复杂三维布尔运算，对于场景中的每一个光源可见面的阴影域多面体都要进行求交运算，算法的计算复杂度是相当可观的。因而这个算法关键是如何有效的判定一个物体表面是否包含在阴影域多面体之内。
 - **Crow**于1977年提出了这样的算法[CROW77]来生成阴影。
-

□ 其它方法

- 实际上，现有的整体光照明模型如光线跟踪算法和辐射度算法都可以很好的处理阴影的生成问题。将在后面的章节详细讨论。

Fresnel光照模型

□ 光的电磁理论

- 我们知道，光波是电磁波的一种，自然光是非偏振光，由自然光照射到物体表面产生的反射光及其反射率系数 ρ 由如下Fresnel公式确定：

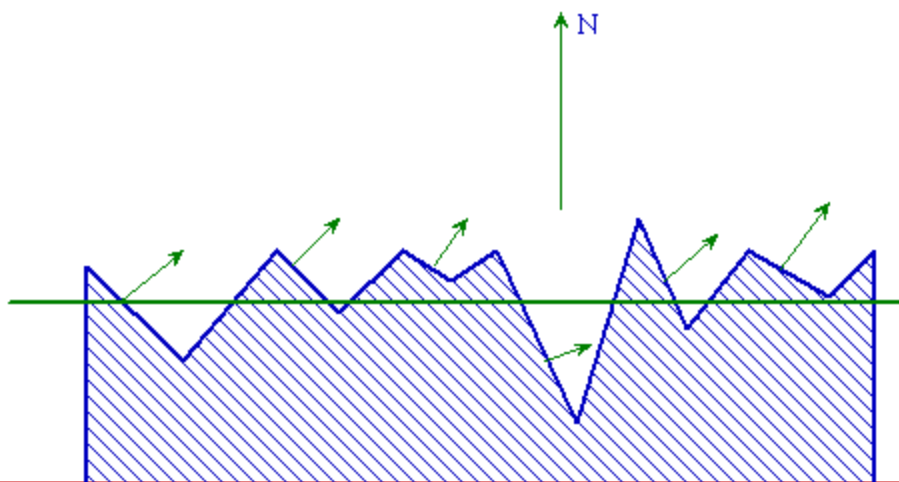
- 其中 θ 是入射角，若发生反射的物体表面两侧的折射率分别为 n_1, n_2 ，那么 ψ 满足这样的一个式子
- $$\rho = \frac{1}{2} \left[\frac{\tan^2(\theta - \psi)}{\tan^2(\theta + \psi)} + \frac{\sin^2(\theta - \psi)}{\sin^2(\theta + \psi)} \right]$$

$$\sin \psi = \frac{n_1}{n_2} \sin \theta$$

-
- Fresnel公式表明，物体的反射率不仅与光线的入射角有关，也物体的折射率有关
 - 由于折射率是入射光的波长的函数，故反射率也是波长的函数，我们将其记为 $\rho(\theta, \lambda)$ 。
-

□ 微平面理论

- 从微观情况看来，相对于很小的光波长来说，物体表面都是十分粗糙不平的。
- 微平面理论将粗糙物体表面看成是由无数个微小的理想镜面组成。这些平面朝向各异，随机分布，如图所示。对于每一个微平面，只有在它的反射方向上才有反射光，而在其他方向上都没有光出现。



-
- 每一微平面是理想镜面，它的反射率可以用上述 **Fresnel** 公式计算，而粗糙表面的反射率与表面的粗糙度有关。
 - 当表面完全光滑时，反射光只有镜面反射光，随着粗糙程度的增加，反射光中镜面反射部分减少，漫反射的部分增加，直到该表面最后完全成为漫反射面。
-

-
- 对于一个实际物体表面，它的反射率可用下式计算：

$$DG\rho(\theta, \lambda)$$

- 其中，**D**为微平面法向的分布函数，表示了物体表面的粗糙程度因子；**G**为由于微平面的相互遮挡或屏蔽而使光产生的衰减因子。

- 对于微平面法向的分布函数**D**，Torrance和Sparrow采用Gauss分布函数模拟微平面法向的分布：

$$D = ke^{-(a/m)^2}$$

- 其**k**为常系数，**a**为微平面的法向与平均法向的夹角，即 $(N \cdot H)$ ；**m**为微平面斜率的均方根

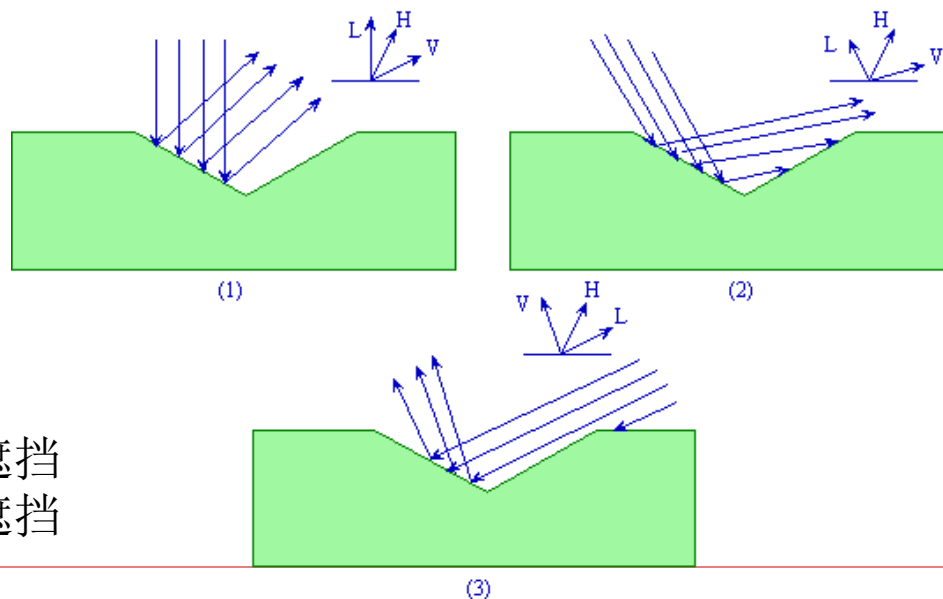
$$m = \sqrt{\frac{m_1^2 + m_2^2 + \dots + m_n^2}{n}}$$

-
- 分布函数**D**也可采用Berkmann分布函数模拟微平面的分布。 即：

$$D = \frac{1}{m^2 \cos^4 \alpha} e^{-\tan^2 \alpha / m^2}$$

- 微平面法向的分布函数**D**表示微平面的法向与平均法向的夹角为***a***的微平面占整个微平面的比例。**m**越小，表面越光滑。
 - 与之相比较，在简单光照明模型中 **$\cos^n a$** 也可看作一种微平面法向的分布函数。
-

- 由于微平面的相互遮挡或屏蔽而使光产生的衰减因子 G 在局部光照明模型中也是可以反映物体表面的粗糙程度的，对于不同角度的入射光，光路被微平面遮挡或屏蔽的情况有三种。



- (1) 没有遮挡
- (2) 反射光被遮挡
- (3) 入射光被遮挡

-
- 对于光路没有遮挡或屏蔽的情况，我们就令 $G=1$;
 - 而对于部分反射光被屏蔽的情况，我们有

$$G_m = \frac{2(N \cdot H)(N \cdot V)}{(N \cdot H)}$$

- 对于部分入射光被遮挡的情况，衰减因子为：

$$G_s = \frac{2(N \cdot H)(N \cdot L)}{(V \cdot H)}$$

- 上面两个式子中， N 是物体表面的法向， H 是微平面的法向， L 是入射光方向， V 是观察方向。
 - 这些符号在本章中都是一致的。
-

-
- 在实际的应用中，我们把上述三种情况衰减因子的最小值作为该微平面的衰减因子：

$$G = \text{Min}\{1, G_m, G_s\}$$

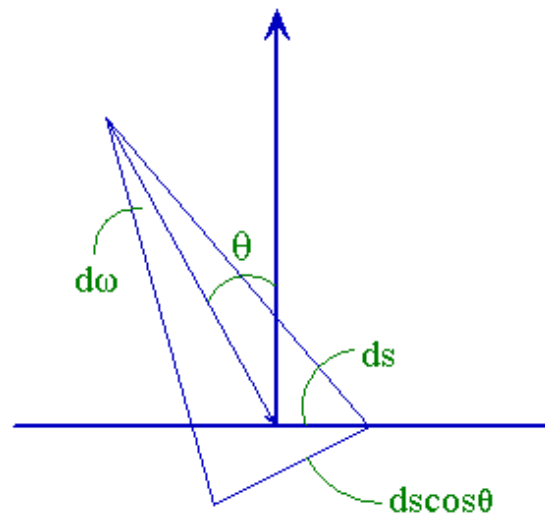
12.2.2 Cook-Torrance光照模型

- Cook和Torrance于1981年提出了这个局部光照明模型 [COOK81]，我们采用其符号，用 R_{bd} 表示物体表面对入射自然光的反射率系数，写成反射光的光强 I_r 与单位时间内单位面积上的入射光能量 E_i 的比，就是

$$R_{bd} = \frac{I_r}{E_i}$$

- 其中，入射光能量 E_i 可用入射光的光强 I_i 和单位面积向光源所张的立体角 $d\omega$ 表示为：

$$E_i = I_i \cos\theta \cdot d\omega = I_i (N \cdot L) d\omega$$



□ 于是有如下的一个关系：

$$I_r = R_{\theta d} I_i (N \cdot L) d\omega$$

□ 而反射率系数可写成漫反射率和镜面反射率的代数和，即

$$R_{\theta d} = K_d R_d + K_s R_s$$

■ 式中，

$K_d + K_s = 1$ ，分别是漫反射系数和镜面反射系数

$R_d = R_d(\lambda)$ 为物体表面的漫反射率，受入射光波的影响；

$R_s = \frac{DG\rho(\theta, \lambda)}{\pi(N \cdot L)(N \cdot V)}$ 为物体表面的镜面反射率，是在上面讨论的理论基础上综

合考虑影响反射光强的因素而得出的。

□ 因此，我们的局部光照明模型最后表示为：

$$I_r = I_a K_a + I_i (N \cdot L) d\omega (K_d R_d + K_s R_s)$$

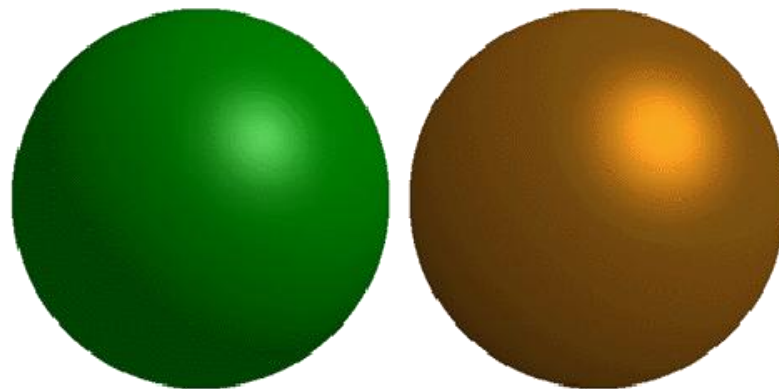
- 式中， I_r 为直接光照下物体表面表现出来的反射光强； $I_a K_a$ 的定义与简单光照明模型相同，表示环境光的影响；式子中的最后一项就是我们前面主要讨论的考虑了物体表面性质的反射光强度量。

□ 与上一节介绍的简单光照模型的比较，本节讨论的局部光反射模型有如下的一些优点：

- （1）局部光照明模型是基于入射光能量导出的光辐射模型，而简单光反射模型基于经验，显然前者更具有理论基础。
 - （2）局部光照明模型的反射项以实际物体表面的微平面理论为基础，反映表面的粗糙度对反射光强的影响，而简单光照模型只以 $(N*H)n$ 近似，前者的模拟更精确。
 - （3）局部光照明模型的高光由Fresnel定律，根据材料的物理性质决定颜色，而简单光照模型只以高光颜色与材料无关。
-

-
- (4) 简单光照模型在入射角接近90度时会产生失真现象，而在局部光照明模型中可以很好的改进这一点。
 - (5) 用简单光照模型生成的物体图象，看上去象塑料，显示不出磨亮的金属光泽，而在局部光照明模型中，反射光强的计算考虑了物体材质的影响，就可以模拟金属的光泽。

图中左边的球用Phong光照明模型显示
右边的球用局部光照明模型显示
两者在高光域有明显的区别



§ 12.3 光透射模型

- 通过对前面的光照明模型的讨论，我们知道无论是简单光照明模型还是局部光照明模型，它们只考虑了由光源引起的漫反射现象和镜面反射现象，而对于光的透射现象都没有处理，这显然不能满足真实感图形学的要求，
 - 本节中将介绍一些典型的光透射模型，把对透射现象的模拟引入到真实图形学中。
-

§ 12.3 光透射模型

- 对于透明或半透明的物体，在光线与物体表面相交时，一般会产生反射与折射，经折射后的光线将穿过物体而在物体的另一个面射出，形成透射光。如果视点在折射光线的方向上，就可以看到透射光。
 - 反射光可以用简单光照明模型或局部光照明模型计算，而对透射光的计算，1980年Whitted提出了一个光透射模型[WHIT80]：Whitted模型，并第一次给出光线跟踪算法的范例，实现了Whitted模型。1983年，Hall在此进一步给出Hall光透射模型[HALL83]，考虑了漫透射和规则透射光。
-

§ 12.3.1 透明效果的简单模拟

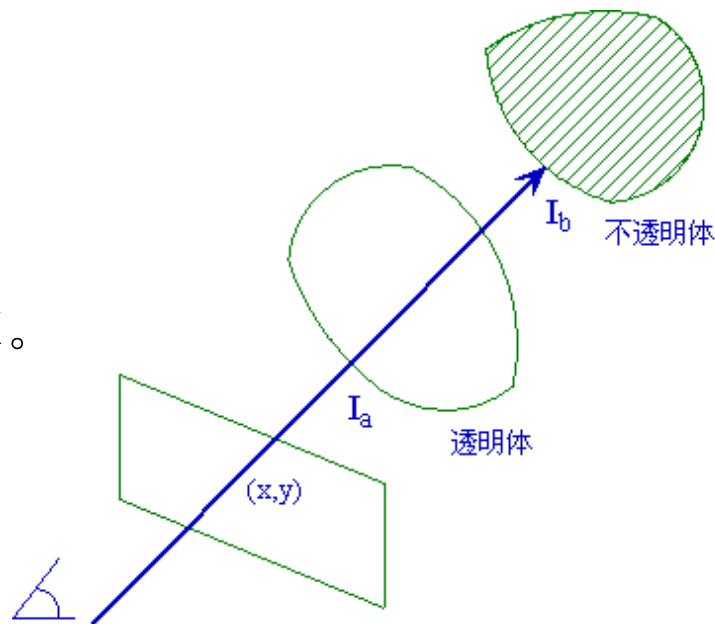
- 由于透明物体可以透射光，因而我们可以透过这种材料看到后面的物体，由于光的折射通常会改变光的方向，要在真实感图形学中模拟折射，需要较大的计算量，
 - 在Whitted和Hall提出光透射模型之前，为了能够看到一个透明物体后面的东西，就有一些透明效果模拟的简单方法。
-

-
- 颜色调和法，该方法不考虑透明体对光的折射以及透明物体本身的厚度，光通过物体表面是不会改变方向的，故可以模拟平面玻璃。
 - 设 t 是物体的透明度， $t=0$ 表示物体是不透明体； $t=1$ 表示物体是完全透体。
 - 我们可以看到物体后面的背景和其他物体，这些物体的前后位置可以通过隐藏面消除算法计算出来。实际上，我们最终所看到的颜色，是物体表面的颜色和透过物体的背景颜色的叠加。
-

- 如图所示，设过象素点 (x,y) 的视线与物体相交处的颜色(或光强)为 I_a ，视线穿过物体与另一物体相交处的颜色(或光强)为 I_b ，则象素点 (x,y) 的颜色(或光强)可由如下颜色调和公式计算：

$$I = tI_b + (1-t)I_a$$

- I_a 和 I_b 可由简单光照明模型计算。
- 由于未考虑透射光的折射以及透明物体的厚度，颜色调和法只能模拟玻璃的透明或半透明效果。



§ 12.3.2 Whitted 光透射模型

- 在简单光照明模型的基础上，加上透射光一项，就得到Whitted光透射模型：

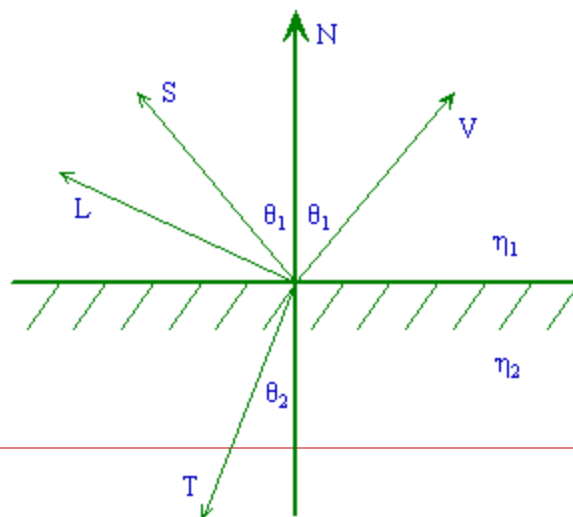
$$I = I_a K_a + I_p K_d (L \cdot N) + I_p K_s (H \cdot N)^n + I_t K_t'$$

- 其中 I_t 为折射方向的入射光强度， K_t' 为透射系数，为0-1之间的一个常数，其大小取决于物体的材料。
- 如果该透明体又是一个镜面反射体，应再加上反射光一项，以模拟镜面反射效果。于是得到Whitted 整体光照模型：

$$I = I_a K_a + I_p K_d (L \cdot N) + I_p K_s (H \cdot N)^n + I_t K_t' + I_s K_s'$$

- 这里， I_s 为镜面反射方向的入射光强度； K_s' 为镜面反射系数，为0-1之间的一个常数；其大小同样取决于物体的材料。
-

- 需要说明的是，所谓的折射方向和镜面反射方向都是相对于视线而言的，它们实际上是视线在折射方向和反射方向的入射光的方向，但方向与光传播的方向相反。如图所示，**S**是视线**V**的镜面反射方向，**T**是**V**的折射方向。在简单光照明模型的情况下，折射光强和镜面反射光强可以认为是折射方向上和反射方向上的环境光的光强。



-
- 用Whitted模型计算光照效果，剩下的关键问题就是计算反射与折射方向。即已知视线方向 V ，求其反射方向 S 与折射方向 T 。然后可由求出反射与折射方向上与另一物体的交点。关于上面的问题可以用几何光学的原理来解决。
 - 给定视线方向 V 与法向方向 N ，视线方向 V 的反射方向 S 可以由下式计算

$$S = 2N(N \cdot V) - V$$

-
- 那么在给定视线方向 V 与法向方向 N 以后，如何求 V 的折射方向 T 呢？首先我们令 V, N, T 均为单位向量， η_1 是视点所在空间的介质折射率， η_2 物体的折射率。根据折射定律，入射角和折射角有如下关系：

$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{\eta_1}{\eta_2} = \eta$$

- V, N, T 共面。
-

□ Whitted的折射方向计算公式:

$$T = k_f(N - V') - N$$

■ 其中, $k_f = 1 / \sqrt{\eta^2 |V'|^2 - |N - V'|^2}$, $V' = \frac{V}{N \cdot V'}$, 计算所得的 T 为非单位向量。

□ Heckbert的计算公式:

$$T = -\frac{1}{\eta}V - (\cos\theta_2 - \frac{1}{\eta}\cos\theta_1)N$$

■ 其中, $\cos\theta_2 = \sqrt{1 - \frac{1}{\eta^2}(1 - \cos^2\theta_1)}$, $\cos\theta_1 = N \cdot V$, 计算所得的 V 为单位向量。

§ 12.3.3 Hall光透射模型

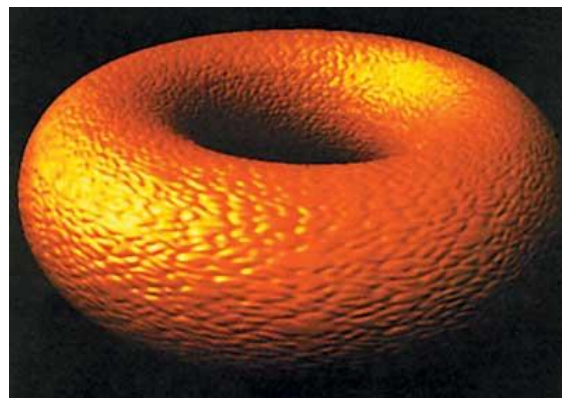
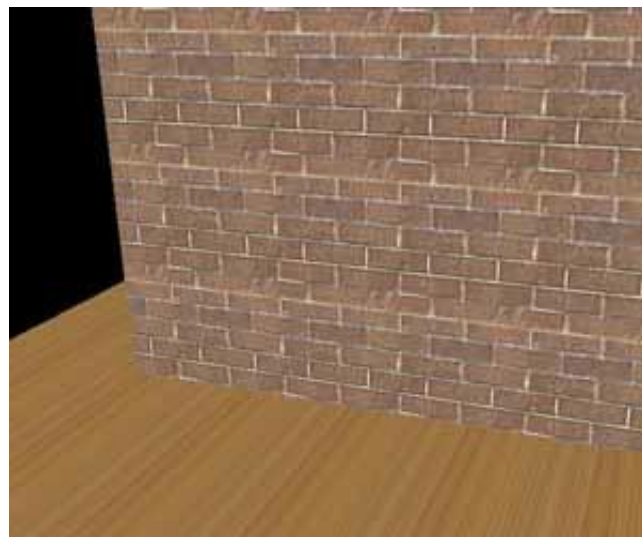
- Hall光透射模型是在Whitted光透射模型的基础上推广而来的，它能够模拟透射高光的效果，实际上，就是在Whitted模型的光强计算中加入光源引起的规则透射分量，同时还可以处理理想的漫透射。
-

§ 12.4 纹理及纹理映射

- 用前面几节中介绍的方法生成的物体图象，往往由于其表面过于光滑和单调，看起来不真实。
 - 现实世界中的物体，其表面通常有它的表面细节，即各种纹理。
 - 通过颜色色彩或明暗度变化体现出来的表面细节，这种纹理称为颜色纹理。
 - 另一类纹理则是由于不规则的细小凹凸造成的，例如桔子皮表面的皱纹，称为几何纹理。
 - 可以用纹理映射的方法给计算机生成的图象加上纹理。
-

§ 12.4.1 纹理的概述

- 从根本上说，纹理是物体表面的细小结构，它可以是光滑表面的花纹、图案，是颜色纹理，这时的纹理一般都是二维图象纹理。
- 纹理还可以是粗糙的表面(如桔子表面的皱纹)，它们被称为几何纹理，是基于物体表面的微观几何形状的表面纹理，一种最常用的几何纹理就是对物体表面的法向进行微小的扰动来表现物体表面的细节。



□ 纹理映射是把我们得到的纹理映射到三维物体的表面的技术。对于纹理映射，我们需要考虑以下三个问题：

1. 考察简单光照明模型，我们需要了解，当物体上的什么属性被改变，就可产生纹理的效果。我们先给出简单光照明模型的式子：

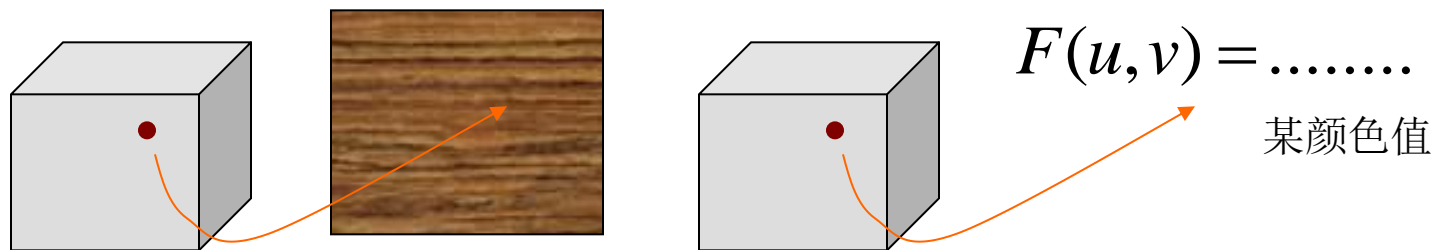
$$I = I_a K_a + K_d I_d (N \cdot L) + K_s I_s (N \cdot H)^n$$

□ 通过分析上面的式子及结合我们前面的介绍，我们可以改变的物体属性有：漫反射系数（改变物体的颜色），物体表面的法向量。通过这些变化，我们就可以得到纹理的效果。

-
- 对于全局光照明模型，可用于产生景物表面的纹理效果的参数更多：
 - 表面颜色，即表面的漫反射系数。
 - 表面法向扰动。即扰动景物表面的法向来产生表面的凹凸纹理。
 - 镜面反射和漫反射。即通过镜面反射和漫反射，可以将物体周围的环境映射到景物的表面上，这种纹理映射技术称为环境映射。
 - 透明度。用纹理函数表示景物的透明度，来表现雕花玻璃这类透明物体的纹理图案。
-

2. 在真实感图形学中，我们可以用如下的两种方法来定义纹理：

- 图象纹理：将二维纹理图案映射到三维物体表面，绘制物体表面上一点时，采用相应的纹理图案中相应点的颜色值。
- 函数纹理：用数学函数定义简单的二维纹理图案，如方格地毯。



3. 在定义了纹理以后，我们还要处理如何对纹理进行映射的问题。

- 对于二维图象纹理，就是如何建立纹理与三维物体之间的对应关系；而对于几何纹理，就是如何扰动法向量。

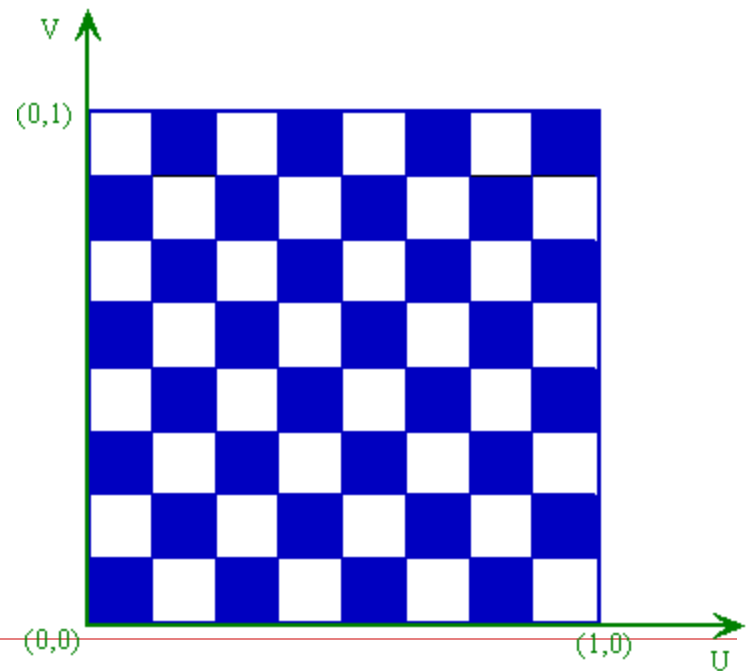
-
- 纹理一般定义在单位正方形区域($0 \leq u \leq 1, 0 \leq v \leq 1$)上，称为纹理空间(对于图像纹理和函数纹理来说...).
 - 理论上，定义在纹理空间上的任何函数可以作为纹理函数，而在实际上，往往采用一些特殊的函数，来模拟生活中常见的纹理。对于纹理空间的定义方法有许多种，下面是常用的几种：
 - 用参数曲面的参数域作为纹理空间(二维)
 - 用辅助平面、圆柱、球定义纹理空间(二维)
 - 用三维直角坐标作为纹理空间(三维)
-

§ 12.4.2 二维纹理域的映射

- 在纹理映射技术中，最常见的纹理是二维纹理。
- 通过映射将这种纹理变换到三维物体的表面，形成最终的图象。我们给出一个二维纹理的函数表示：

$$g(u, v) = \begin{cases} 0 & [u \times 8] \times [v \times 8] \text{ 为奇数} \\ 1 & [u \times 8] \times [v \times 8] \text{ 为偶数} \end{cases}$$

- 它的纹理图象模拟国际象棋上黑白相间的方格。



-
- 二维纹理还可以用图象来表示，用一个 $M \times N$ 的二维数组存放一幅数字化的图象，然后把该二维图象映射到三维的物体表面上。
 - 为了实现这个映射，就要建立物体空间坐标 (x, y, z) 和纹理空间坐标 (u, v) 之间的对应关系，一般用投影后几何变换的方法来实现。

$$(x, y, z) \longleftrightarrow (u, v) ?$$

-
- 平面映射（任何线性变换）
 - 柱面映射
 - 球面映射



□ 圆柱面纹理映射

- 问题：已知圆柱面上一点坐标，如何取得其纹理坐标。即得到纹理映射函数(可以由圆柱面的参数方程推出)。

- 如果参数方程如下所示：
$$\begin{cases} x = \cos(2\pi u) & 0 \leq u \leq 1 \\ y = \sin(2\pi v) & 0 \leq v \leq 1 \\ z = v \end{cases}$$

- 那么，对给定圆柱面上一点 (x, y, z) ，可以用下式反求参数：

$$(u, v) = \begin{cases} (y, z) & \text{如果 } x = 0 \\ (x, z) & \text{如果 } y = 0 \\ (\frac{\sqrt{x^2 + y^2} - |y|}{x}, z) & \text{其它} \end{cases}$$

§ 12.4.3 三维纹理域的映射

- 前面介绍的二维纹理域映射对于提高图形的真实感有很大的作用，但是，由于纹理域是二维的，图形场景物体一般是三维的，这样在纹理映射的时候是一种非线性映射，在曲率变化很大的曲面区域就会产生纹理变形，极大的降低了图象的真实感，
 - 对于二维纹理映射，对于一些非正规拓扑表面，纹理连续性不能保证。
-

§ 12.4.3 三维纹理域的映射

- 假如在三维物体空间中，物体中每一个点 (x, y, z) 均有一个纹理值 $t(x, y, z)$ ，其值由纹理函数 $t(x, y, z)$ 唯一确定，那么对于物体上的空间点，就可以映射到一个纹理空间上了，而且是三维的纹理函数，这是三维纹理提出来的基本思想。
 - 三维纹理映射的纹理空间定义在三维空间上，与物体空间是同维的，在纹理映射的时候，只需把场景中的物体变换到纹理空间的局部坐标系中去即可。
-

□ 以木纹的纹理函数为例，来说明三维纹理函数的映射，它是通过空间坐标 (x, y, z) 来计算纹理坐标 (u, v, w) ：

- 首先求木材表面上的点到木材中心的半径 R ， $R = \sqrt{u^2 + v^2}$
对半径进行小的扰动：

$$R = R + 2\sin(20 * \alpha)$$

- 然后对Z轴进行小弯曲处理， $R = R + 2\sin(20 * a + w/150)$ ，
 - 最后根据半径 R 用下面的伪码来计算color值作为木材表面上点的颜色，就可以得到较真实的木纹纹理：
-

```
{  
    grain = R MOD 60 ; /* 每隔60一个木纹 */  
    if ( grain < 40 )  
        color = 淡色 ;  
    else  
        color = 深色 ;  
}
```



§ 12.4.4 几何纹理

- 为了给物体表面图象加上一个粗糙的外观，我们可以对物体的表面几何性质作微小的扰动，来产生凹凸不平的细节效果，就是几何纹理的方法。
-

-
- 定义一个纹理函数 $F(u, v)$ ，对理想光滑表面 $P(u, v)$ 作不规则的位移，具体是在物体表面上的每一个点 $P(u, v)$ ，都沿该点处的法向量方向位移 $F(u, v)$ 个单位长度，这样新的表面位置变为：

$$\tilde{P}(u, v) = P(u, v) + F(u, v) * N(u, v)$$

- $N(u, v)$ ：法向量

- 扰动后的向量单位化，用于计算曲面的明暗度，可以产生貌似凹凸不平的几何纹理。
-

§ 12.5 整体光照明模型

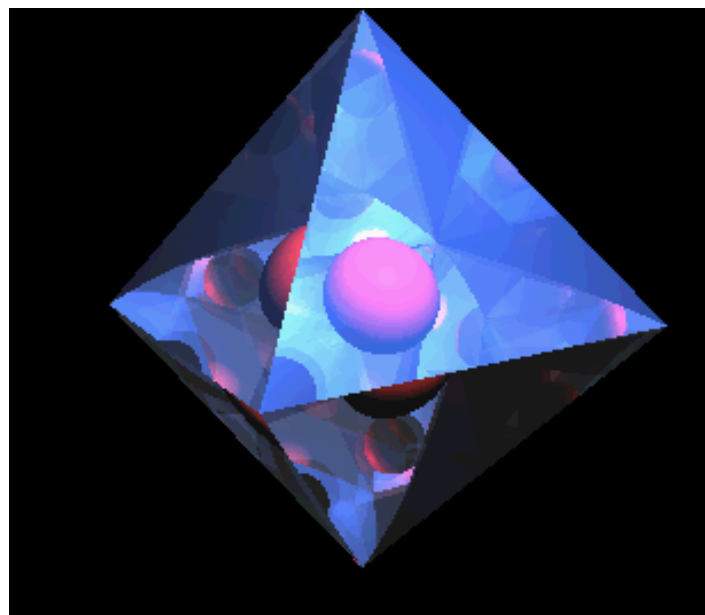
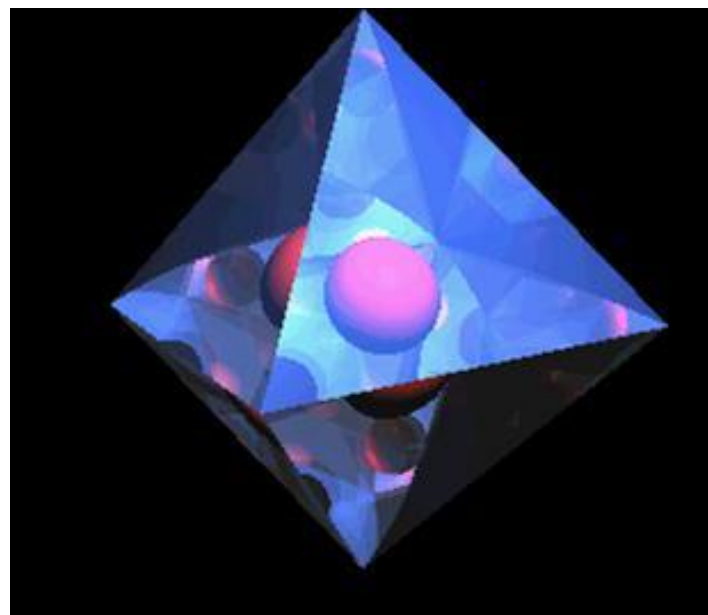
- 简单光照明模型虽然可以产生物体的真实感图象，但它们都只是处理光源直接照射物体表面的光强计算，不能很好的模拟光的折射、反射和阴影等，也不能用来表示物体间的相互光照明影响；
 - 基于简单光照明模型的光透射模型，虽然可以模拟光的折射，但是这种折射的计算范围很小，不能很好的模拟多个透明体之间的复杂光照明现象。
 - 对于上述的这些问题，就必须要有有一个更精确的光照明模型。
-

□ 整体光照明模型：

- 它是相对于局部光照明模型而言的。
 - 在现有的整体光照明模型中，主要有：
 - 光线跟踪
 - 辐射度
 - 上两种方法是当今真实感图形学中最重要两个图形绘制技术，在**CAD**及图形学领域得到了广泛的应用。
-

□ 光线跟踪算法

- 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。
 - 在真实感图形学对光线跟踪算法的研究中，早在**1968年Apple A**研究隐藏面消除算法时[APPL68]，就给出了光线跟踪算法的描述。**1979年Kay和Greenberg**的研究考虑了光的折射[KAY79]。**1980年Whitted**提出了第一个整体光照**Whitted**模型[WHIT80]，并给出一般性光线跟踪算法的范例，综合考虑了光的反射、折射透射、阴影等。
-

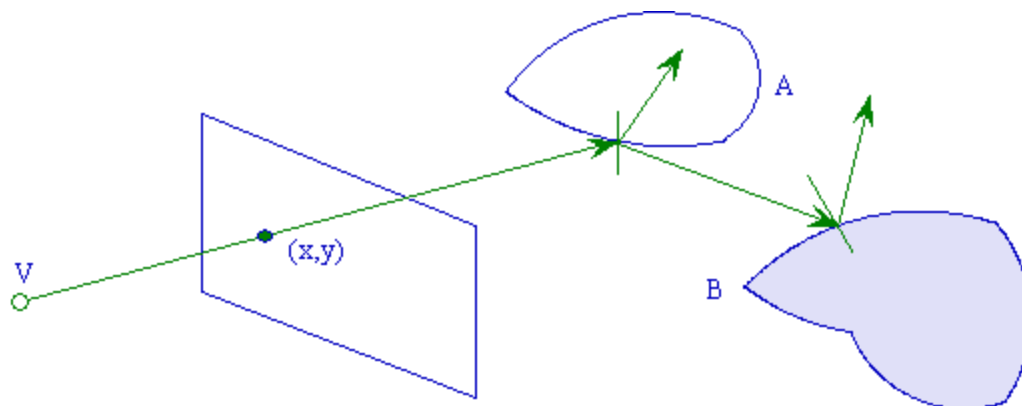


□ 光线跟踪的基本原理

- 由光源发出的光到达物体表面后，产生反射和折射，简单光照明模型和光透射模型模拟了这两种现象。
 - 在简单光照明模型中，反射被分为理想漫反射和镜面反射光，在简单光透射模型把透射光分为理想漫透射光和规则透射光。
 - 由光源发出的光称为直接光，物体对直接光的反射或折射称为直接反射和直接折射，相对的，把物体表面间对光的反射和折射称为间接光，间接反射，间接折射。这些是光线在物体之间的传播方式，是光线跟踪算法的基础。
-

-
- 最基本的光线跟踪算法是跟踪镜面反射和折射。
从光源发出的光遇到物体的表面，发生反射和折射，光就改变方向，沿着反射方向和折射方向继续前进，直到遇到新的物体。
 - 实际光线跟踪算法的跟踪方向与光传播的方向是相反的，是视线跟踪。
-

-
- 为了详细介绍光线跟踪算法，我们先给出四种射线的定义与光强的计算方法。
 - 在光线跟踪算法中，我们有如下的四种光线：
 - 视线是由视点与象素 (x, y) 发出的射线
 - 阴影测试线是物体表面上点与光源的连线
 - 反射光线
 - 折射光线

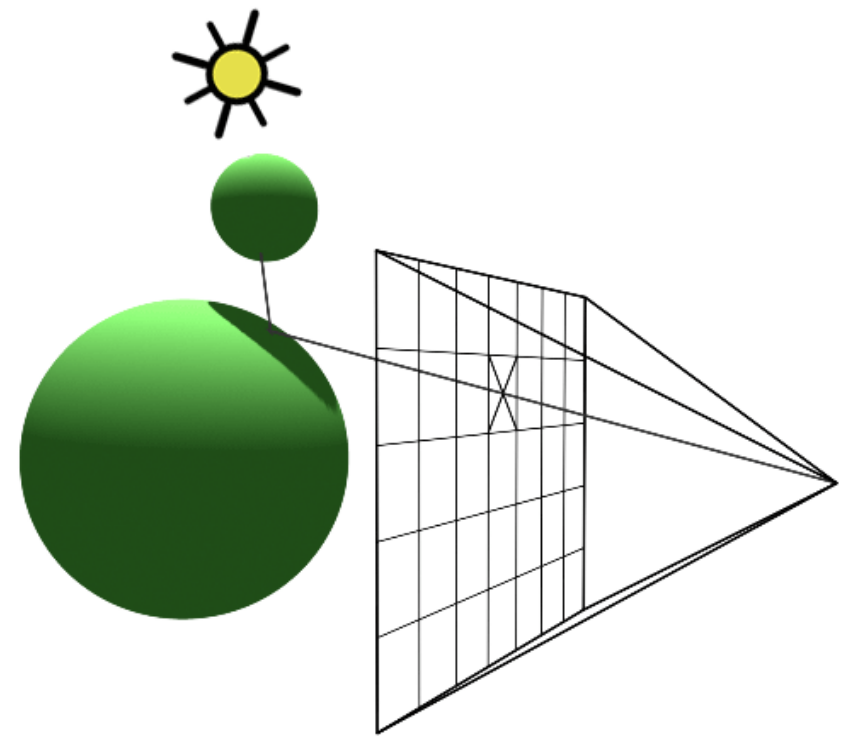
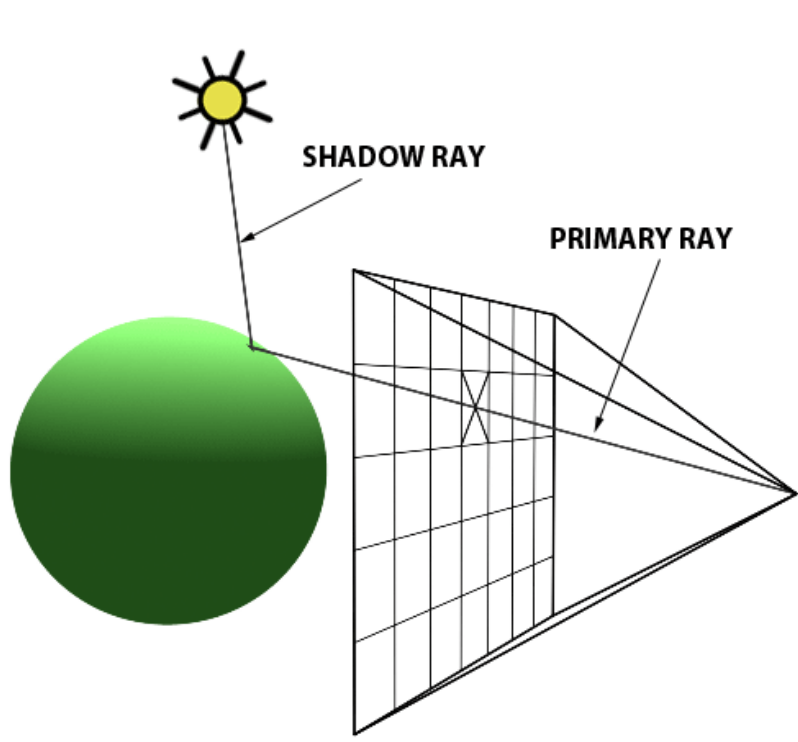


□ 当光线 V 与物体表面交于点 P 时，点 P 分为三部分，把这三部分光强相加，就是该条光线 V 在 P 点处的总的光强：

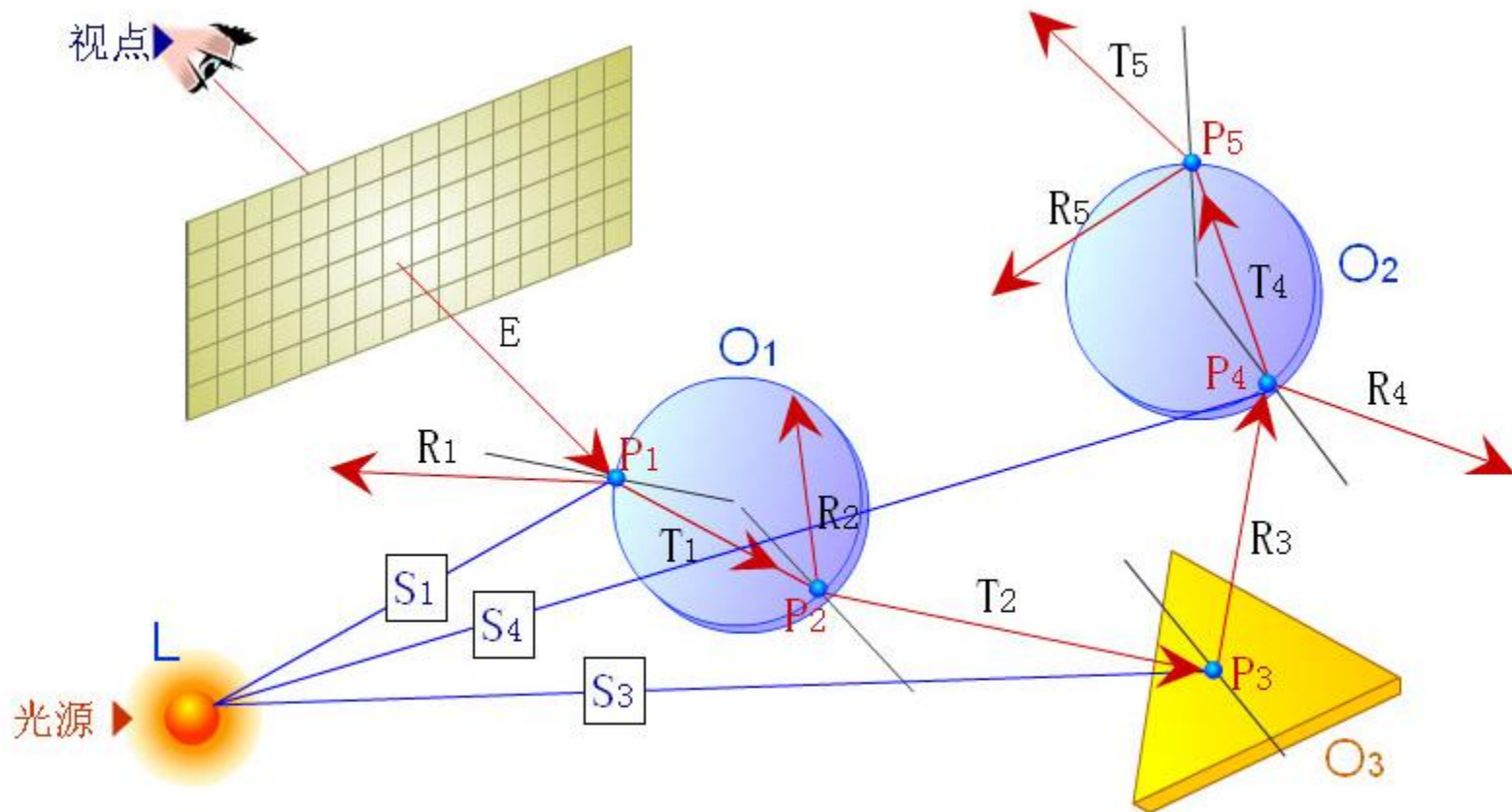
- 由光源产生的直接的光线照射光强，是交点处的局部光强，可以由下式计算：

$$I = I_a K_a + \sum_i I_{pi} [K_{ds} (L_i \cdot N) + K_s (H_{si} \cdot N)^{n_s}] \\ + \sum_j I_{pj} [K_{dt} (-N \cdot L_j) + K_t (N \cdot H_{vj})^{n_t}]$$

- 反射方向上由其它物体引起的间接光照光强，由 $IsKs'$ 计算， Is 通过对反射光线的递归跟踪得到；
 - 折射方向上由其它物体引起的间接光照光强，由 $ItKt'$ 计算， It 通过对折射光线的递归跟踪得到。
-



光线跟踪例子



-
- 我们可以看出，光线跟踪算法实际上是光照明物理过程的近似逆过程，这一过程可以跟踪物体间的镜面反射光线和规则透射，模拟了理想表面的光的传播。
 - 虽然在理想情况下，光线可以在物体之间进行无限的反射和折射，但是在实际的算法进行过程中，我们不可能进行无穷的光线跟踪，因而需要给出一些跟踪的终止条件。在算法应用的意义上，可以有以下几种终止条件：
 - 该光线未碰到任何物体。
 - 该光线碰到了背景。
 - 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（小于某个设定值）。
 - 光线反射或折射次数即跟踪深度大于一定值。
-

□ 伪码

■ RayTracing(start, direction, weight, color)

```
{  
    if ( weight < MinWeight )  
        //衰减, 光线对于视点的光强贡献很小了  
        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 )  
            //碰到背景了  
            color = black;  
    }
```

```
else
{
    Ilocal = 在交点处用局部光照模型计算出的光强;
    计算反射方向 R; //如果有
    RayTracing(最近的交点, R, weight*Wr, Ir);
    计算折射方向 T; //如果有
    RayTracing(最近的交点, T, weight*Wt, It);
    color = Ilocal + KsIr+ KtIt;
}
}
```

□ 光线跟踪算法的加速

- 自适应深度控制
 - 包围盒及层次结构
 - 三维**DDA**算法
 - 空间八叉树剖分技术
-

□ 辐射度方法

- 辐射度方法是继光线跟踪算法后，真实感图形绘制技术的一个重要进展。
 - 尽管光线跟踪算法成功地模拟了景物表面间的镜面反射、规则透射及阴影等整体光照效果，但由于光线跟踪算法的采样特性，和局部光照模型的不完善性，该方法难于模拟景物表面之间的多重漫反射效果，因而不能反映色彩渗透现象。
-

-
- 1984年，美国Cornell大学和日本广岛大学的学者分别将热辐射工程中的辐射度方法引入到计算机图形学中，用辐射度方法成功地模拟了理想漫反射表面间的多重漫反射效果。
 - 经过十多年的发展，辐射度方法模拟的场景越来越复杂，图形效果越来越真实。与前几章介绍的光照模型与绘制方法有所不同，辐射度方法基于物理学的能量平衡原理，它采用数值求解技术来近似每一个景物表面的辐射度分布。由于场景中，景物表面的辐射度分布与视点选取无关，辐射度方法是一个视点独立(View independent)的算法，使之可广泛应用于虚拟环境的漫游(walkthrough)系统中。
-

§ 12.6 实时真实感图形学技术

- 我们已经详细介绍了各种光照明模型及它们在真实感图形学中的一些应用方法，它们可以很好的模拟出现实世界中的复杂场景，所生成的真实感图象可以给人以高度逼真的感觉。
 - 但是，用这些模型生成一幅真实感图象都需要较长的时间，因此不能满足某些需要实时图形显示的任务要求。例如交互的科学计算可视化系统，虚拟现实系统，游戏系统(LAG)。
 - 对于生成真实感图形学的实时性要求很高的场合，就必须采用实时真实感图形学技术。
-

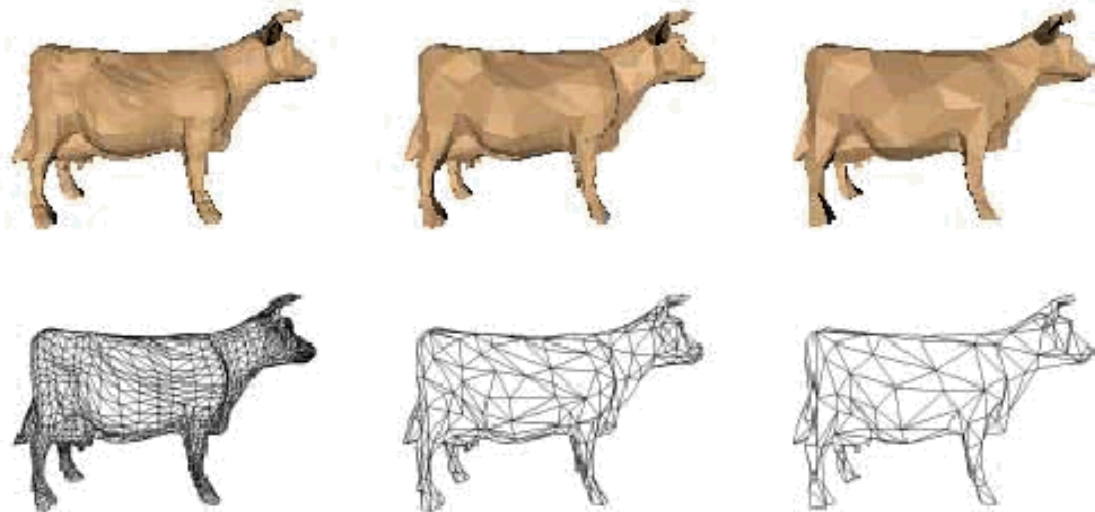
-
- 实时真实感图形学技术是在当前图形算法和硬件条件的限制下提出的在一定的时间内完成真实感图形图象绘制的技术。一般来说，它是通过损失一定的图形质量来达到实时绘制真实感图象的目的。
 - 目前主要是通过降低显示三维场景模型的复杂度来实现，这种技术被称为层次细节(**LOD: Level of Detail**)显示和简化技术，是当前大多数商业实时真实感图形生成系统中所采用的技术。
 - 在最近的几年中，又出现了一种全新思想的真实感图象生成技术—基于图象的绘制技术(**Image Based Rendering**)，它利用已有的图象来生成不同视点下的场景真实感图象，生成图象的速度和质量都是以前的技术所不能比拟的，具有很高的应用前景。
-

□ 层次细节显示和简化

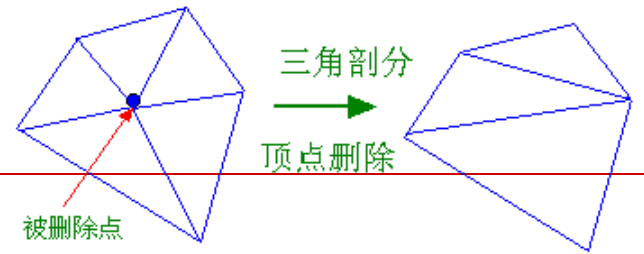
- 从需要绘制的三维场景本身入手。
 - 在当前的真实感图形学中，需要绘制的三维场景的复杂度都非常高，一个复杂的场景可能会包含几十甚至几百万个多边形，要实现对这种复杂场景的实时真实感图形绘制是很困难的。
 - 一种自然的想法就是通过减少场景的复杂度，来提高图象绘制的速度。层次细节显示和简化技术就是在这种背景下提出来的。
-

-
- 当场景中许多面片在屏幕上的投影小于一个像素，我们可以合并这些可见面而不损失画面的视觉效果。
 - 多面体网格的简化：在不影响画面视觉效果的前提下，通过逐次简化景物的表面细节来减少场景的几何复杂性，从而提高绘制算法的效率。
-

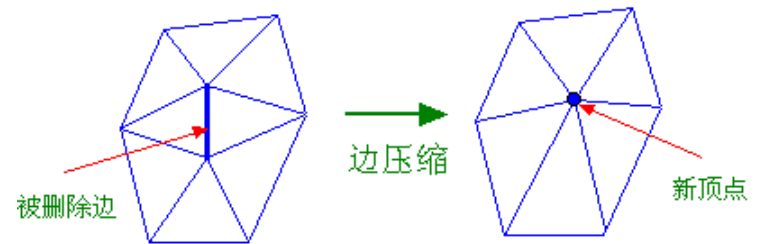
-
- 该技术通常对一个原始多面体模型建立几个不同逼近程度的几何模型。与原模型相比，每个模型均保留一定层次的细节，当从近处观察物体时，我们采用精细的模型，而当从远处观察物体时，则采用较粗糙的模型，这是层次细节显示和简化技术的基本原理。



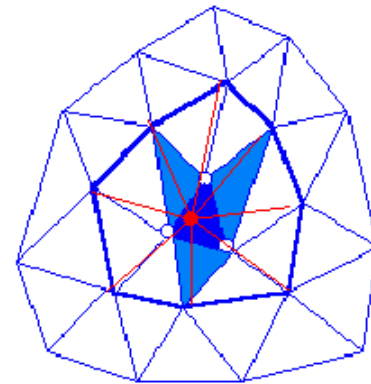
-
- 需要注意的是，当视点连续变化时，在两个不同层次的模型之间就存在一个明显的跳跃，有必要在相邻层次的模型之间形成光滑的视觉过渡，即几何形状过渡，使生成的真实感图象序列是视觉光滑的。
-



(A) 顶点删除操作



(B) 边压缩操作



(C) 面片收缩操作

□ LOD模型的建立方法:

- 顶点删除操作[SCHR92], 删除网格中的一个顶点, 然后对它的相邻三角形形成的空洞作三角剖分。
- 边压缩操作[HOPP93], 把网格上的一条边压缩为一个顶点, 与该边相邻的两个三角形的退化(面积为零), 而它的两个顶点融合为一个新的顶点。
- 面片收缩操作[GIEN97], 把网格上的一个面片收缩为一个顶点, 该三角形本身和与其相邻的三个三角形都退化, 而它的三个顶点收缩为一个新的顶点。

-
- 层次细节显示和简化技术是实时真实感图形学技术中应用比较多的一个技术。
 - 通过这种技术，可以较好的简化场景的复杂度，
 - 采用不同分辨率的模型来显示复杂场景的不同物体，使在生成的真实感图象质量损失很小的情况下，来实时的产生真实感图象，满足某些关键任务的实时性要求。
-

-
- LOD通过减少场景的几何复杂程度，也就是减少真实感图形学算法需要渲染的场景面片数目，来提高绘制真实感图象的效率，达到实时的要求。
 - 随着计算机水平的发展，人们可以得到高度复杂的三维场景，对于这种场景，我们即使对其层次简化到一定的程度，它的复杂度仍然很高，而不能被现有水平的计算机实时处理，同时我们又不能把场景简化程度很大，这样会导致图象质量的严重降低，而失去真实感图象的最初目的。
-

-
- 因此：需要一种能够对高度复杂的场景进行实时真实感图形绘制的技术，而且要求这种技术可以在普通的计算机上应用。最近几年，满足这种要求的技术才开始出现，那就是基于图象的绘制技术（**Image Based Rendering**）。
 - 基于图象的绘制技术与前面的真实感图象算法完全不同，它是从一些预先生成好的真实感图象出发，通过一定的插值、混合、变形等操作，生成不同视点处的真实感图象。
-

-
- 基于一些预先生成的图像（或环境映照）来生成不同视点的场景画面。



END
