

# 8 真实感图形绘制

---

- 基本概念
- 简单光照模型
- 基于简单光照模型的多边形绘制
- 透明处理
- 整体光照模型与光线跟踪
- 模拟景物表面细节

## 8.1 基本概念

---

- 真实感图形绘制：通过综合利用数学、物理学、计算机以及心理学等知识在计算机图形输出设备上绘制出能够以假乱真的美丽景象。
- 光强（度）：描述物体表面朝某方向辐射光的颜色，它既能表示光能大小又能表示其色彩组成的物理量。

# 基本概念

---

- 光照模型（**Illumination model**），也称明暗模型，主要用于物体表面某点处的光强度计算。
  - 简单的光照模型
  - 复杂的光照模型

# 基本概念

---

## □ 真实感图形绘制过程

根据假定的光照条件和景物外观因素，依据一定的光照模型，计算可见面投射到观察者眼中的光强度大小，并将它转换成适合图形设备的颜色值，生成投影画面上每一个像素的光强度，使观察者产生身临其境的感觉。

# 基本概念

---

## □ 真实感图形绘制步骤

- 在计算机中进行场景造型;
- 进行取景变换和透视变换;
- 进行消隐处理;
- 进行真实感图形绘制。

## 8.2 简单光照模型

---

- 简单光照模型
- 环境光
- 漫反射光
- 镜面反射光
- 光强衰减
- 颜色处理

# 简单光照模型

---

- 简单光照模型中只考虑反射光的作用。
- 反射光由环境光、漫反射光和镜面反射光三部分组成。

# 环境光 (Background Light)

---

- 特点：照射在物体上的光来自周围各个方向，又均匀地向各个方向反射。
- P点对环境光的反射强度为

$$I_e = I_a K_a$$

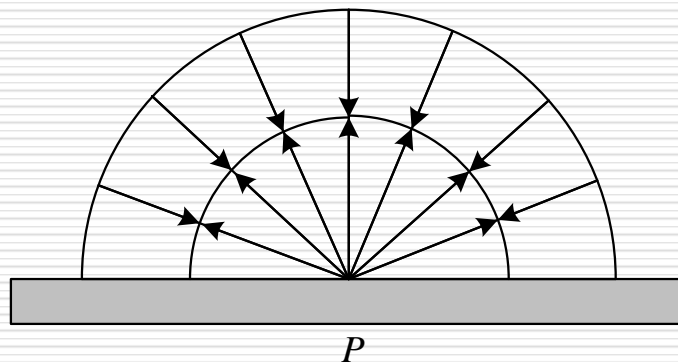


图8.1 环境光的反射



# 漫反射光 (Diffuse Reflection)

- 一个粗糙的、无光泽的表面呈现为漫反射。
- 特点：光源来自一个方向，反射光均匀地射向各个方向。
- 由Lambert余弦定理可得点P处漫反射光的强度为：

$$I_d = I_p K_d \cos \theta, \theta \in [0, \frac{\pi}{2}]$$

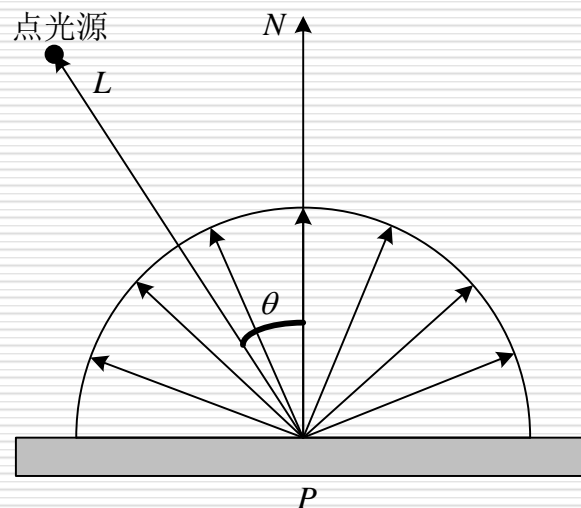


图8.2 漫反射

# 漫反射光 (Diffuse Reflection)

□ 若 $L$ 和 $N$ 都已规格化为单位矢量，则有

$$I_d = I_p K_d (L \cdot N)$$

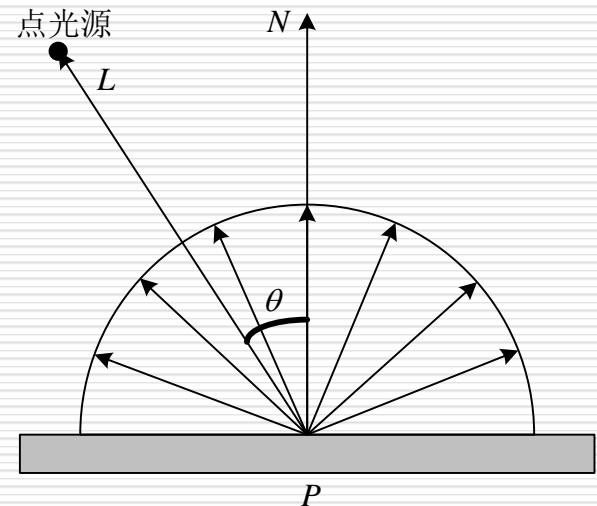


图8.2 漫反射

# 漫反射光 (Diffuse Reflection)

---

## □ 对于彩色

$$I_p = (I_{pR}, I_{pG}, I_{pB})$$

$$I_{dR} = I_{pR} K_{dR} (L \cdot N)$$

$$I_{dG} = I_{pG} K_{dG} (L \cdot N)$$

$$I_{dB} = I_{pB} K_{dB} (L \cdot N)$$

## □ 对于多个漫反射光源

$$I_d = \sum_{i=1}^n I_{p,i} K_d (L_i \cdot N)$$

# 镜面反射光

- 镜面反射遵循反射定律，入射光和反射光分别位于表面法矢的两侧。
- 如果观察者正好处在P点的镜面反射方向上，就会看到一个比周围亮得多的高光点。

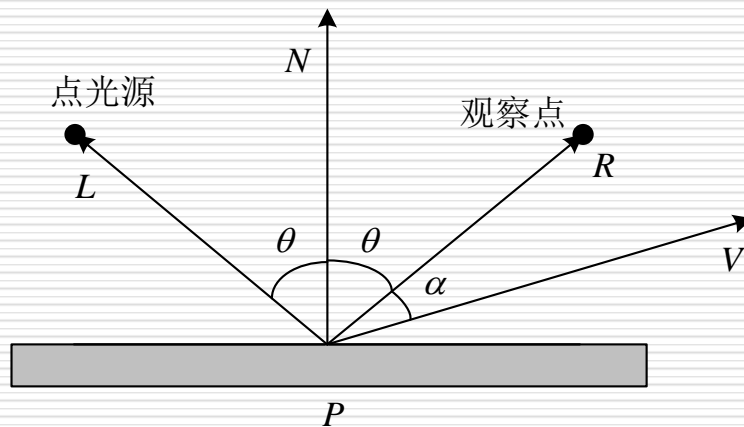


图8.3 镜面反射

# 镜面反射光

□ 镜面反射情况由Phong模型给出：

$$I_s = I_p K_s \cos^n \alpha$$

□ 若 $R$ 和 $V$ 已规格化为单位矢量，则：

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

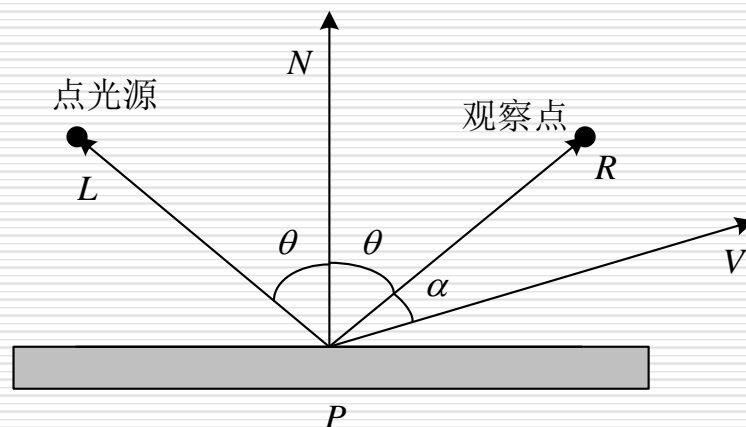


图8.3 镜面反射

# 物体表面光强计算

---

- 从视点观察到物体上任一点P处的光强度I应为环境光反射光强度 $I_e$ 、漫反射光强度 $I_d$ 以及镜面反射光的光强度 $I_s$ 的总和：

$$\begin{aligned} I &= I_e + I_d + I_s \\ &= I_a K_a + I_p K_d (L \cdot N) + I_p K_s (R \cdot V)^n \end{aligned}$$

# 光强衰减

---

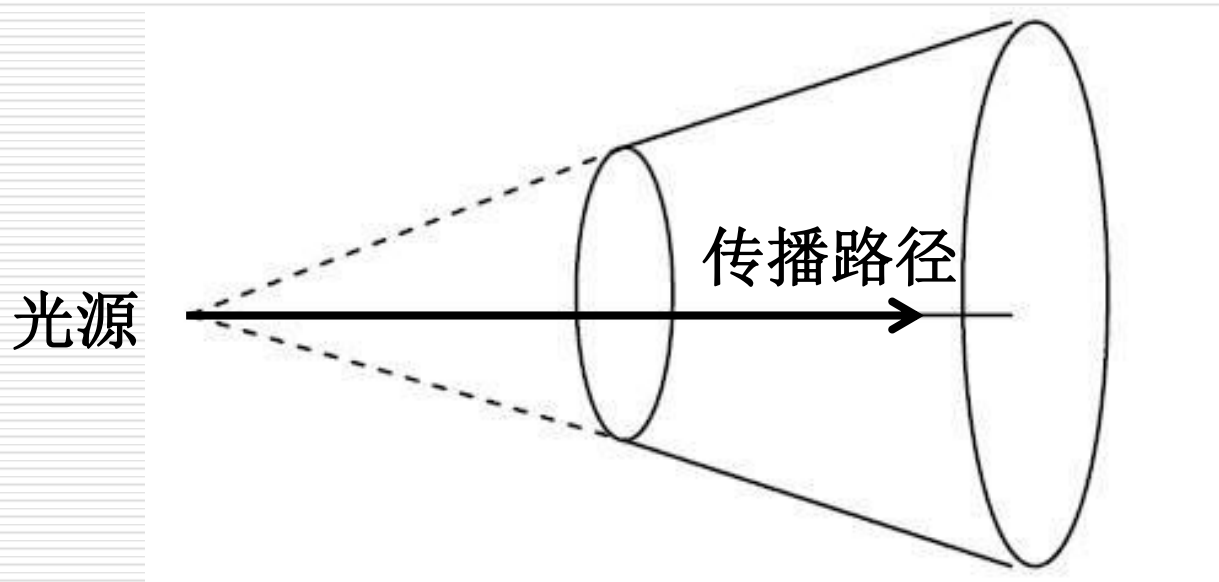


图8.4 镜面反射

# 光强衰减

---

- 光在传播的过程中，其能量会发生衰减。光照模型中必须考虑光强衰减，否则会影响生成图形的真实效果。
- 光强的衰减可以采用常数衰减、一次函数衰减和二次函数衰减等。



# 光强衰减

---

## □ 常用的二次衰减函数

$$f(d) = \min(1, \frac{1}{c_0 + c_1 d + c_2 d^2})$$

$$\begin{aligned} I &= I_e + I_d + I_s \\ &= I_a K_a + \sum_{i=1}^n f(d_i) I_{p,i} K_d (L_i \cdot N) + \sum_{i=1}^n f(d_i) I_{p,i} K_s (H_i \cdot N)^n \end{aligned}$$

# 颜色处理

---

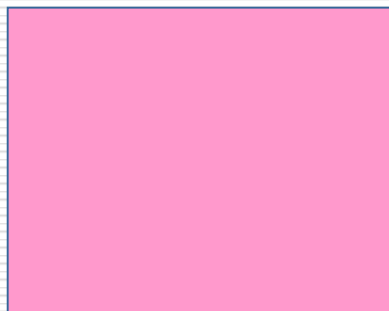
- 选择颜色模型 (**color model**)
  - 面向硬件的颜色模型: **RGB**、**CYM**
  - 面向视觉感知的颜色模型: **HSI**
- 为颜色分量指定光照模型

# 颜色模型

---

## □ 颜色的三要素

- 明度（Brightness）：色彩的明暗程度，可以量化的光源亮度



- 明度对纯度产生不可忽视的影响

# 颜色模型

---

## □ 颜色的三要素

- 色调（Hue）：颜色的相貌，只和颜色的波长有关。明度和纯度变化时，波长不变。

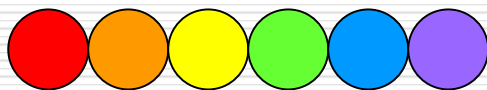


# 颜色模型

---

## □ 颜色的三要素

■ 饱和度（ Saturation ）：色彩的纯度，光的颜色接近光谱色的程度



纯度最高的颜色



黑、白、灰没有纯度

图8.5 饱和度

# 颜色模型

## □ RGB颜色模型

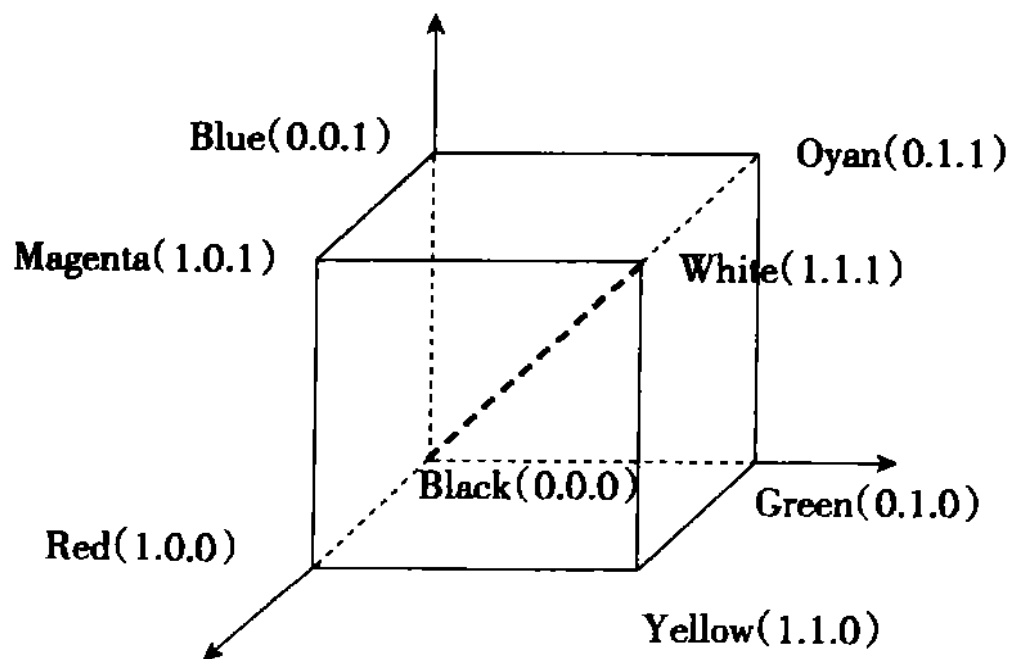


图8.6 RGB颜色空间

# 颜色模型

---

## □ YUV模型

- 亮度信号（Y），U、V代表两个色差分量
- 在亮度分量（Y）的分辨率中的差错比色度值（U、V）中的差错更为严重

$$Y = 0.30R + 0.59G + 0.11B$$

$$U = (B - Y) \times 0.493$$

$$V = (R - Y) \times 0.877$$

# 颜色模型

## □ HSV模型

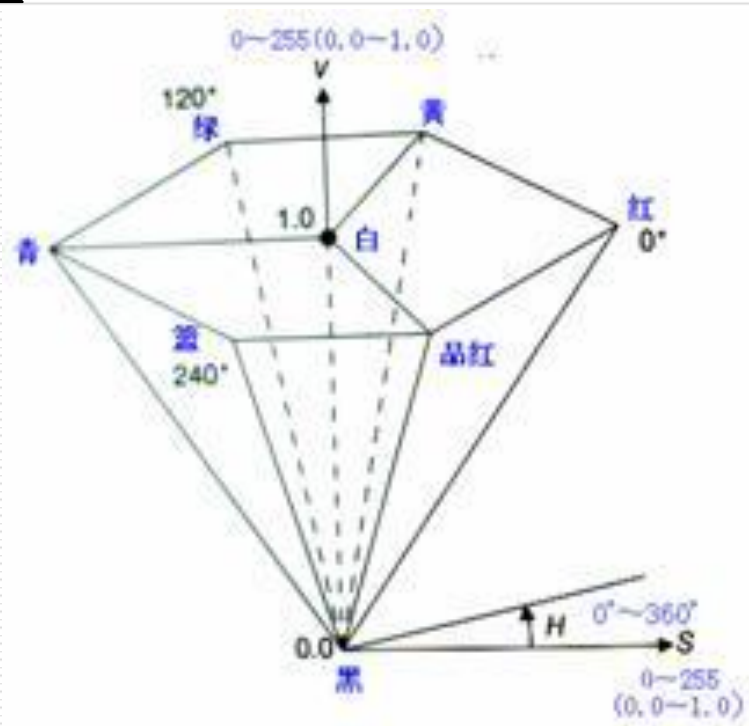


图8.7 HSV颜色空间



# 颜色

---

## □ 以RGB颜色模型为例

■ 环境光强度:  $I_a = (I_{aR}, I_{aG}, I_{aB})$

■ 入射光强度:  $I_p = (I_{pR}, I_{pG}, I_{pB})$

■ 环境光反射系数:  $K_a = (K_{aR}, K_{aG}, K_{aB})$

■ 漫反射系数:  $K_d = (K_{dR}, K_{dG}, K_{dB})$

■ 镜面反射系数:  $K_s = (K_{sR}, K_{sG}, K_{sB})$

# 颜色

---

## □ 光强计算公式

$$I_R = I_{aR}K_{aR} + \sum_{i=1}^n f(d_i)I_{pR,i}K_{dR}(L_i \cdot N) + \sum_{i=1}^n f(d_i)I_{pR,i}K_{sR}(H_i \cdot N)^n$$

$$I_G = I_{aG}K_{aG} + \sum_{i=1}^n f(d_i)I_{pG,i}K_{dG}(L_i \cdot N) + \sum_{i=1}^n f(d_i)I_{pG,i}K_{sG}(H_i \cdot N)^n$$

$$I_B = I_{aB}K_{aB} + \sum_{i=1}^n f(d_i)I_{pB,i}K_{dB}(L_i \cdot N) + \sum_{i=1}^n f(d_i)I_{pB,i}K_{sB}(H_i \cdot N)^n$$

# 颜色

---

## □ 光强计算公式

$$\begin{bmatrix} K_{aR} \\ K_{aG} \\ K_{aB} \end{bmatrix} = K_a \begin{bmatrix} S_{dR} \\ S_{dG} \\ S_{dB} \end{bmatrix}, \quad \begin{bmatrix} K_{dR} \\ K_{dG} \\ K_{dB} \end{bmatrix} = K_d \begin{bmatrix} S_{dR} \\ S_{dG} \\ S_{dB} \end{bmatrix}, \quad \begin{bmatrix} K_{sR} \\ K_{sG} \\ K_{sB} \end{bmatrix} = K_s \begin{bmatrix} S_{sR} \\ S_{sG} \\ S_{sB} \end{bmatrix}$$

## 8.3 基于简单光照模型的多边形绘制

---

- 恒定光强
- Gouraud明暗处理
- Phong明暗处理

# 恒定光强Flat明暗处理

---

## □ 只用一种颜色绘制整个多边形

- 光源在无穷远处，则多边形上所有点的 $L \cdot N$ 为常数，衰减函数也是一个常数。
- 视点在无穷远处，则多边形上所有点的 $V \cdot R$ 为常数。
- 多边形是景物表面的精确表示，即不是一个含曲线面景物的近似表示。

# 恒定光强

---

## □ 特点

- 每个多边形只需计算一次，速度快
- 由于相邻多边形的法矢量不同，计算出的颜色也不相同，整个景物表面的颜色过渡不光滑，在边界处产生不连续的变化，呈块状效应

# Gouraud明暗处理

---

- **Gouraud**明暗处理方法，又称为亮度插值明暗处理，它通过对多边形顶点颜色进行线性插值来绘制其内部各点，其步骤为：
  - 计算每个多边形顶点处的平均单位法矢量；
  - 对每个顶点根据简单光照模型来计算其光强；
  - 在多边形表面上将顶点强度进行线性插值。

# Gouraud 明暗处理

---

## □ 计算顶点的平均单位法矢量

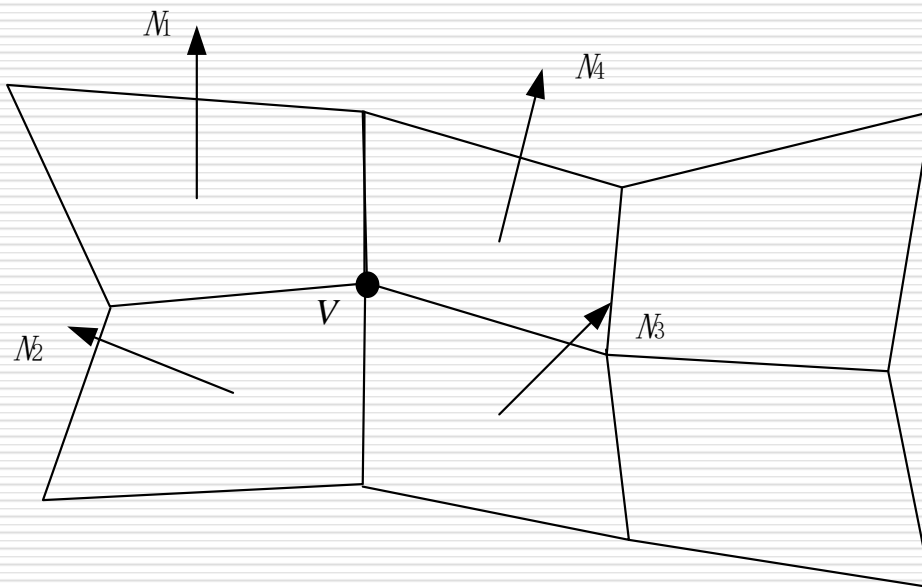
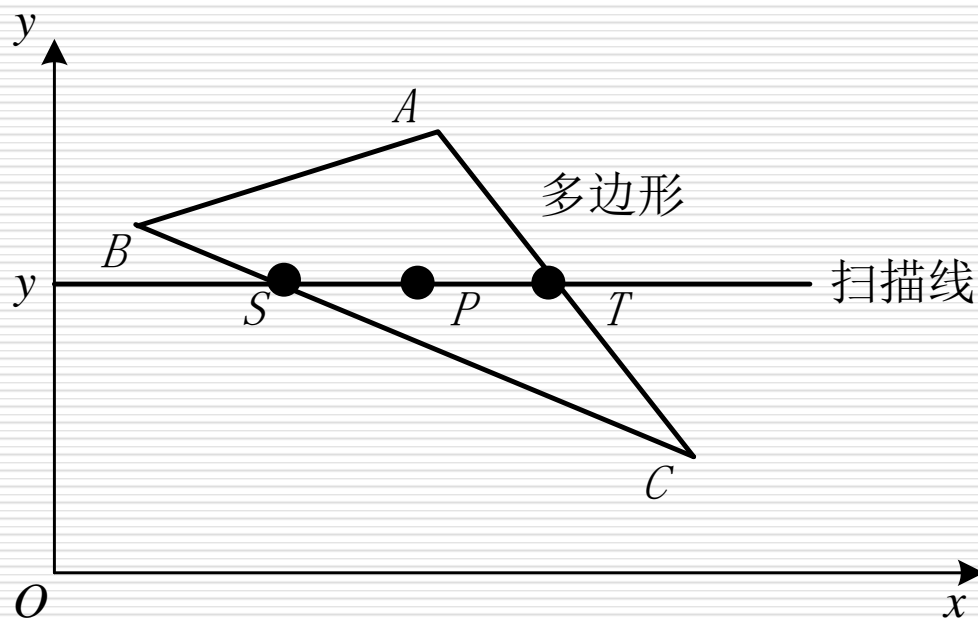


图8.8 Gouraud 明暗处理的顶点法矢量计算



# Gouraud 明暗处理

## □ 双线性插值方法



$$I_S = \frac{y_S - y_C}{y_B - y_C} I_B + \frac{y_B - y_S}{y_B - y_C} I_C$$

$$I_T = \frac{y_T - y_C}{y_A - y_C} I_A + \frac{y_A - y_T}{y_A - y_C} I_C$$

$$I_P = \frac{x_T - x}{x_T - x_S} I_S + \frac{x - x_S}{x_T - x_S} I_T$$

图8.8 Gouraud 明暗处理的双线性插值

# Gouraud 明暗处理

## □ 利用连贯性

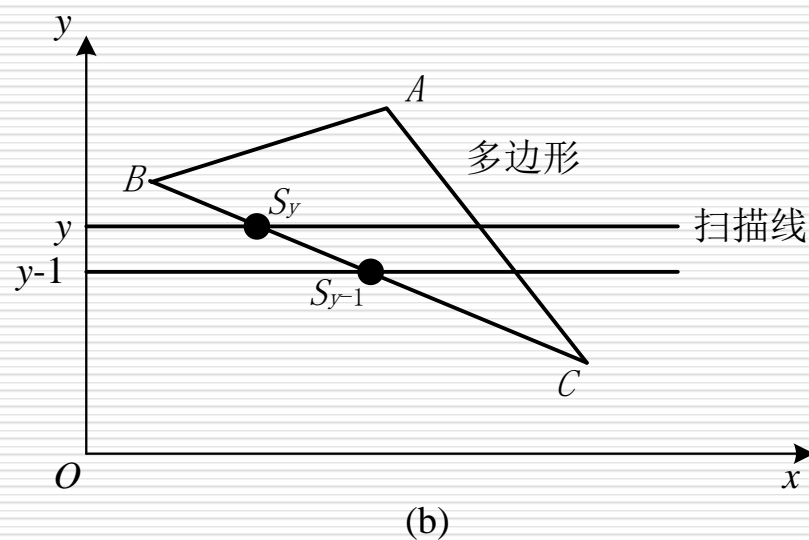
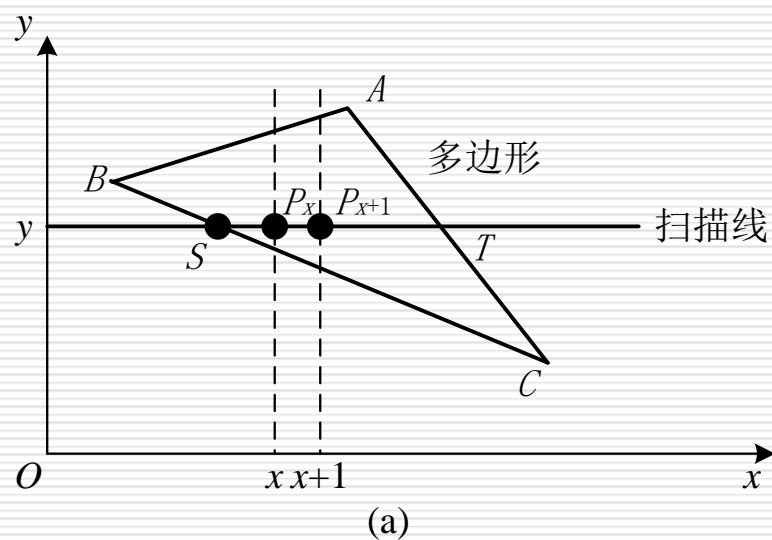


图8.9 Gouraud 明暗处理的双线性插值

# Gouraud明暗处理

---

- 优点：能有效的显示漫反射曲面，计算量小
- 缺点
  - 会造成表面上出现过亮或过暗的条纹，称为马赫带（**Mach\_band**）效应
  - 高光有时会异常
  - 当对曲面采用不同的多边形进行分割时会产生不同的效果

# Phong明暗处理

---

- **Phong**明暗处理方法，又称为法矢量插值明暗处理，它对多边形顶点的法矢量进行插值以产生中间各点的法矢量，其步骤为：
  - 计算每个多边形顶点处的平均单位法矢量；
  - 用双线性插值方法求得多边形内部各点的法矢量。
  - 最后按光照模型确定多边形内部各点的光强。

# Phong明暗处理

## □ 矢量双线性插值方法

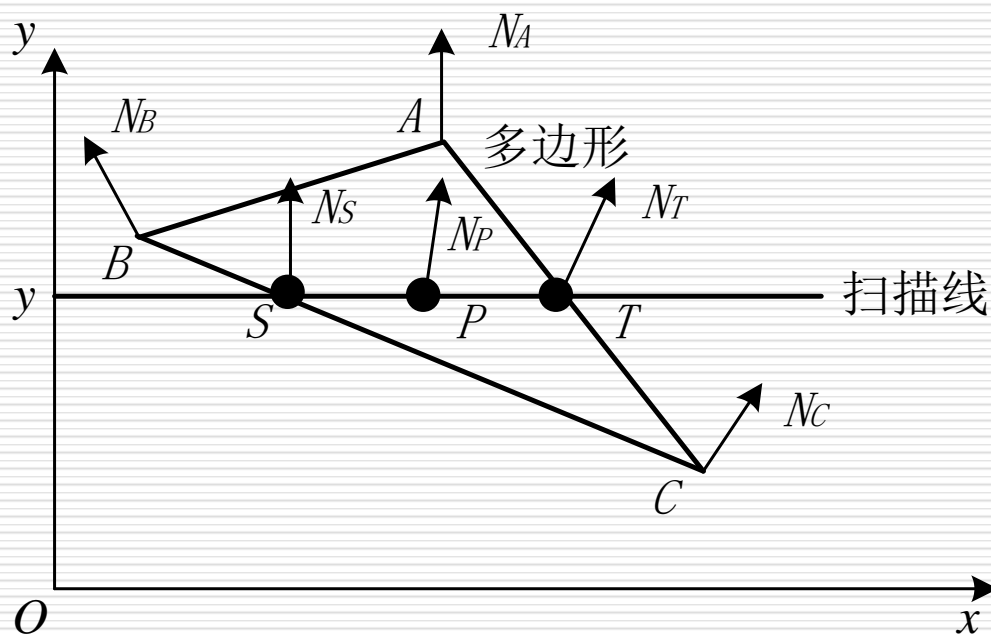


图8.10 Phong明暗处理的矢量双线性插值

# Phong明暗处理

---

## □ 优点

- **Phong**着色方法绘制的图形比**Gouraud**方法更真实
- 高光区域的扩散
- 产生正确的高光区域

# Phong明暗处理

---

## □ 缺点

- **Phong**着色方法计算量远大于**Gouraud**着色方法
- 在处理某些多边形分割的曲面时，**Phong**算法还不如**Gouraud**算法好

## 8.4 透明处理

---

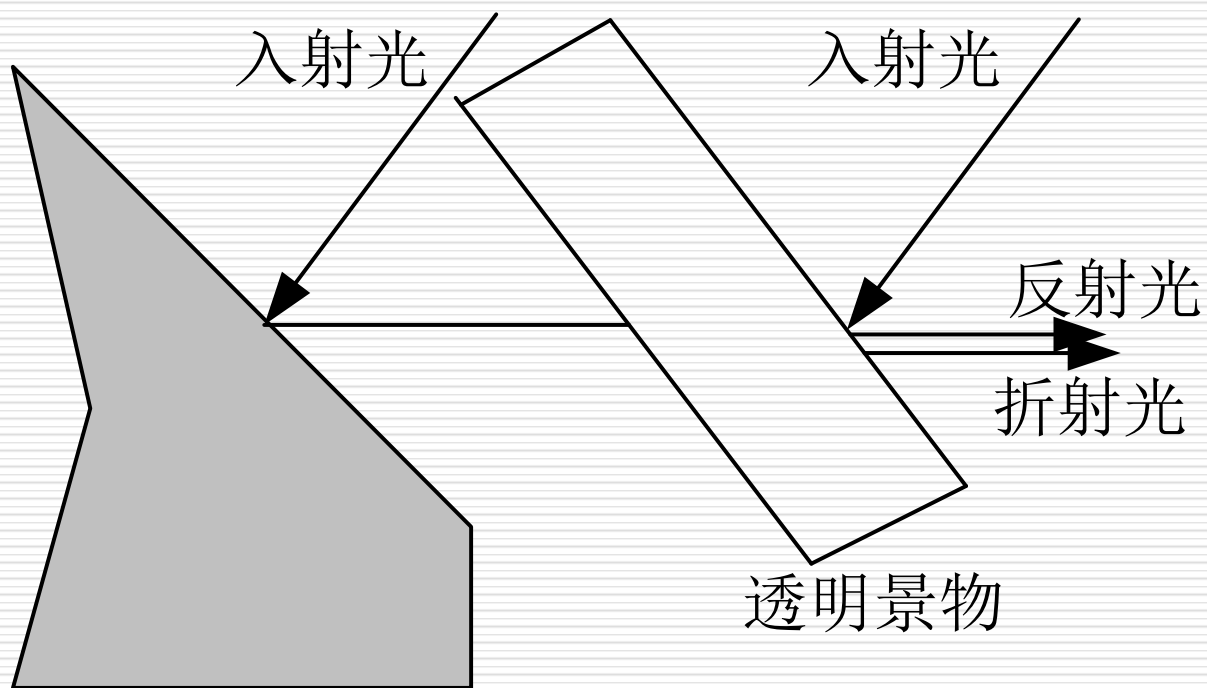


图8.11 透明表面的光强包括反射光和折射光



# 透明处理

$$T = \left( \frac{\eta_i}{\eta_r} \sin \theta_i - \cos \theta_r \right) N - \frac{\eta_i}{\eta_r} L$$

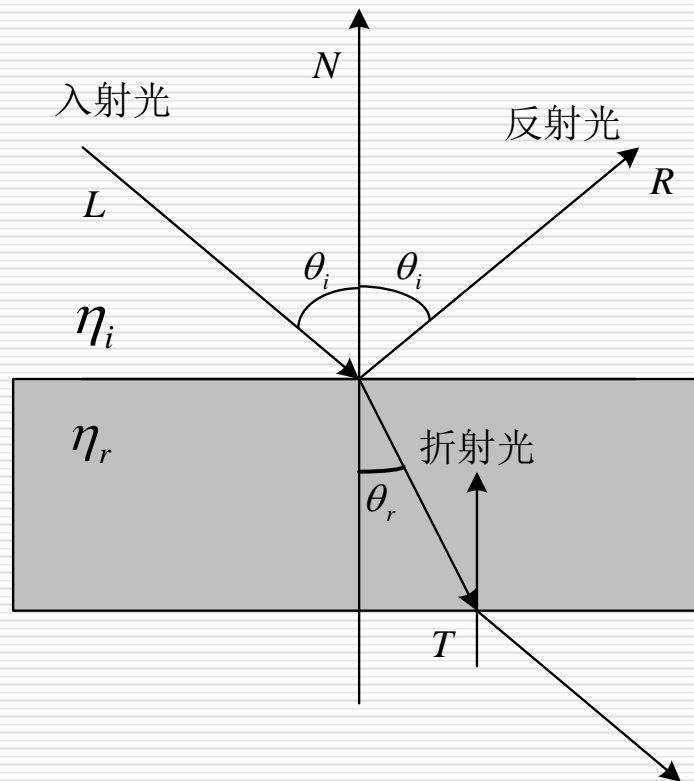


图8.12 光的折射

# 透明处理

---

## □ 基本的透明模型

- 不考虑折射导致的路径平移
- 假定各个对象间的折射率不变，则折射角总等于入射角
- 对于较薄的多边形表面，可以产生合理的透明效果。

# 透明处理

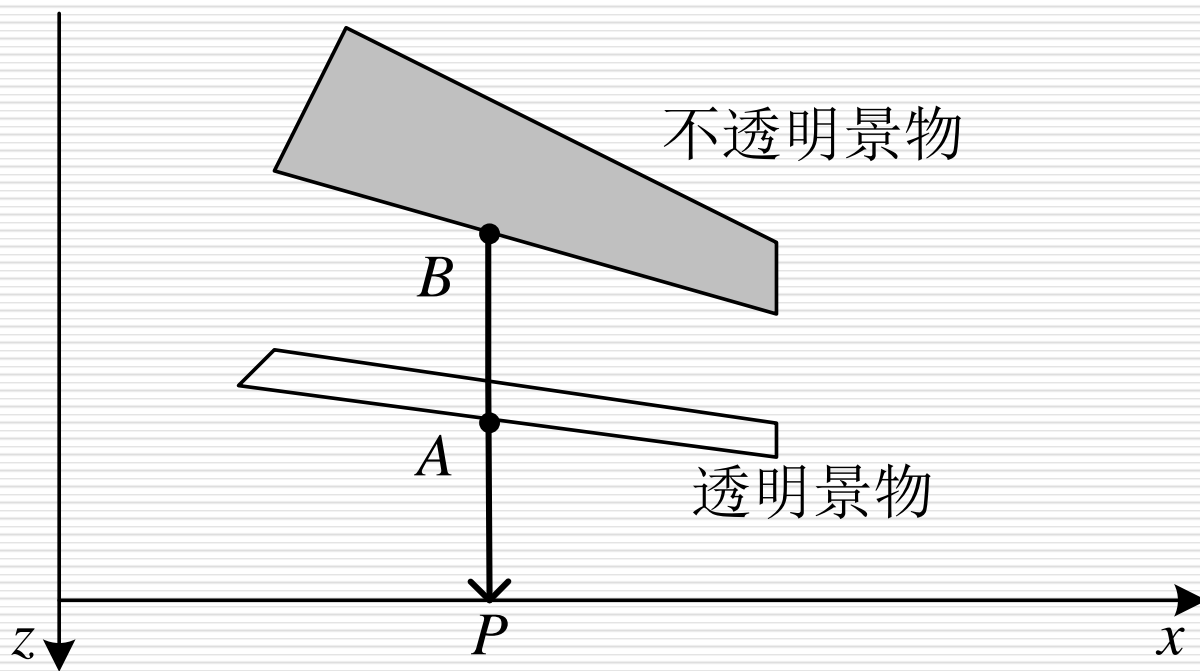


图8.13 简单的透明处理

# 透明处理

---

## □ 基本的透明模型

### ■ 透明系数 $k$

$$I = (1 - k)I_{\text{反射}} + kI_{\text{透射}}$$

### ■ $(1 - k)$ 称为不透明因子

## 8.5 产生阴影

---

- 阴影是由于物体截断了光线而产生的
- 如果光源位于物体一侧的话，阴影总是位于物体的另一侧，也就是与光源相反的一侧。
- 从视点以及从光源看过去都是可见的面不会落在阴影中
- 从视点看过去是可见的，而从光源看过去是不可见的面，肯定落在阴影之内

# 产生阴影

---

## □ 产生具有阴影的图形绘制算法

- 将视点移到光源位置，用多边形区域排序消隐算法，将多边形分成两大类：向光多边形和背光多边形。
- 将视点移到原来的观察位置，对向光多边形和背光多边形进行消隐，并选用一种光照模型计算多边形的亮度，就可得到有阴影效果的图形。

## 8.6 整体光照模型与光线跟踪

---

- 整体光照模型
- **Whitted**光照模型
- 光线跟踪算法

# 整体光照模型与光线跟踪

## □ 物体表面入射光的构成

- 光源直接照射
- 其它物体的反射光
- 透射光

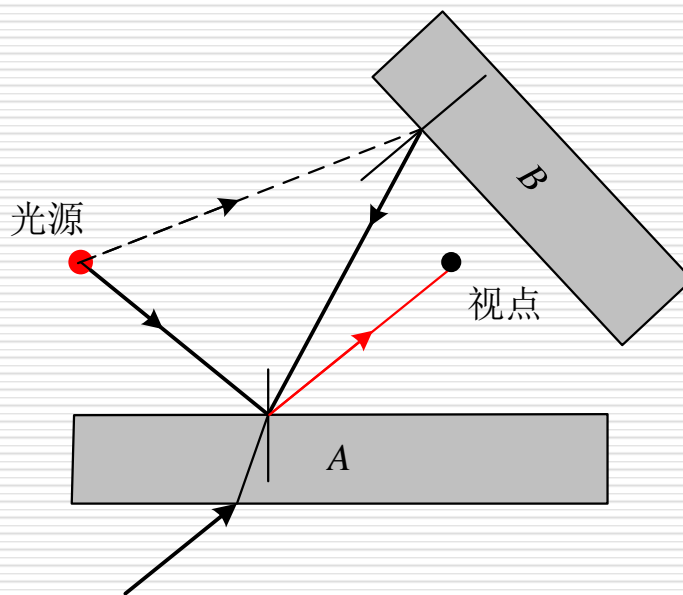


图8.17 物体A表面的入射光



# 整体光照模型

---

- 一个完整的光照明模型应该包括由光源和环境引起的漫反射分量、镜面反射分量、规则透射分量以及漫透射分量等。
- 仅考虑由光源引起的漫反射分量和镜面反射分量，而环境反射分量则简单地用一常数来代替，这类光照模型称为局部光照模型。
- 能同时模拟光源和环境照明效果的光照模型称为整体光照模型。

# Whitted光照模型

□ **Whitted**在简单光照模型中增加了环境镜面反射光和环境规则透射光，以模拟周围环境的光投射在景物表面上产生的理想镜面反射和规则透射现象。

$$I = I_{local} + K_s I_s + K_t I_t$$

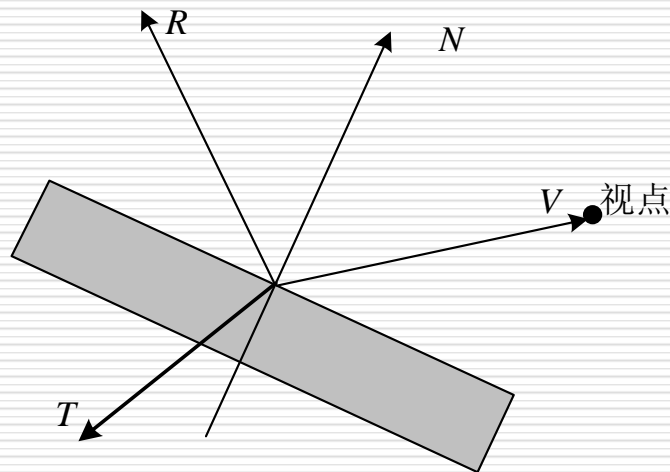


图8.18 物体表面的镜面反射和投射

# Whitted光照模型

---

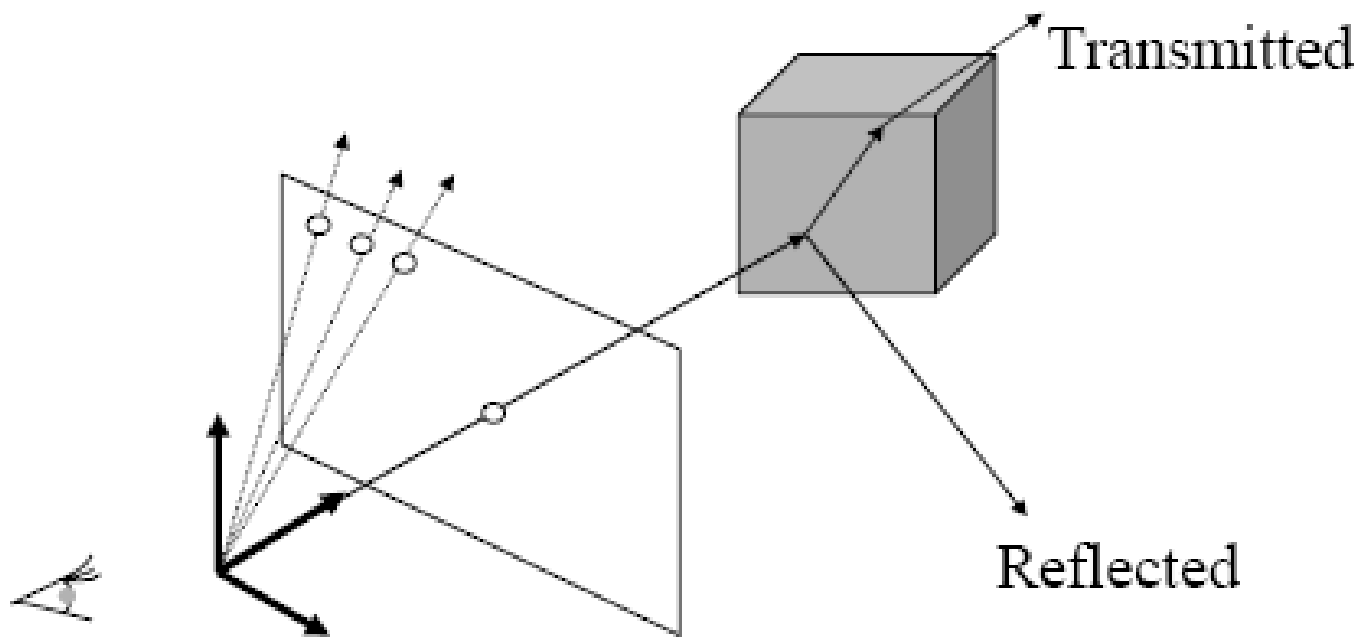


图8.19 整体光照模型

# 光线跟踪算法

- 光线跟踪（**Ray Tracing**）方法基于几何光学的原理，通过模拟光的传播路径来确定反射、折射和阴影等。

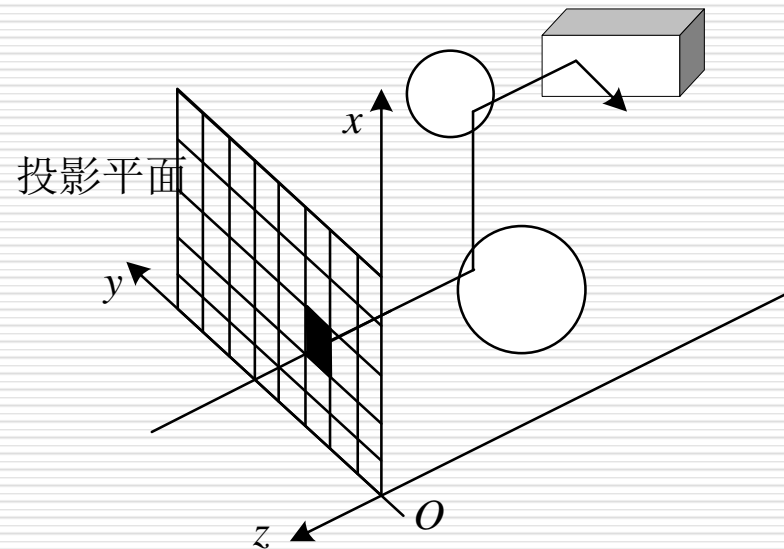


图8.20 光线跟踪算法

# 光线跟踪算法

---



图8.21 光线跟踪算法

# 光线跟踪算法

---

- 迄今为止最为成功的生成真实感图形算法之一
  - 算法简单
  - 生成的图形真实感强
  - 计算量大
- 其前身是光线投射（**Ray Casting**）算法

# 光线投射算法步骤

- 从视点出发通过该像素中心向场景发出一条光线，并求出该条光线与场景中物体的全部交点。

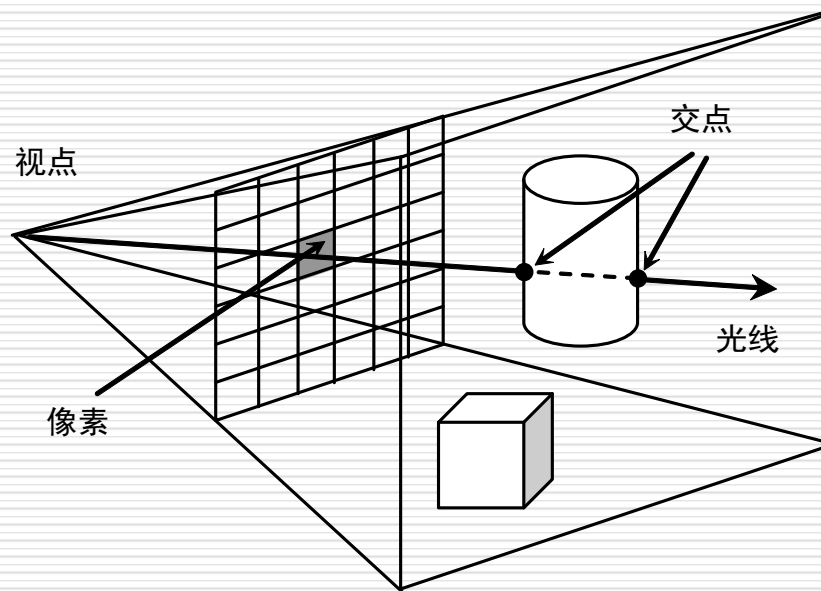


图8.22 光线投射算法

# 光线投射算法步骤

- ❑ 将各交点沿光线方向排序，获得离视点最近交点
- ❑ 依据局部光照明模型计算该交点处的光亮度，并将所得光亮度值赋给该像素

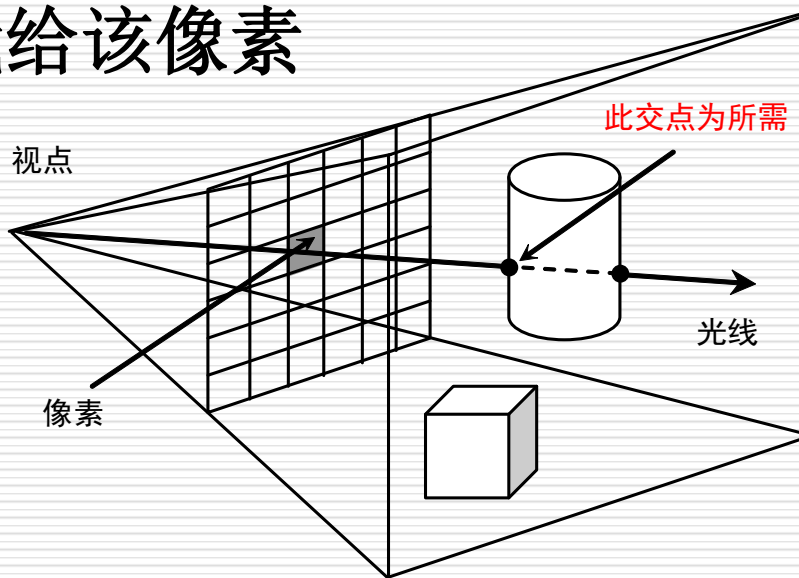


图8.23 光线投射算法



# 光线跟踪算法步骤

---

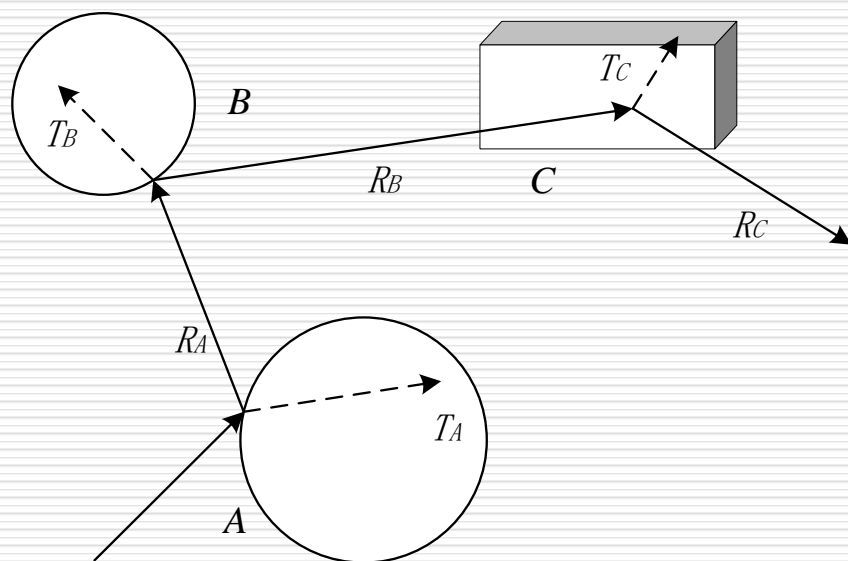
- 从视点出发通过该像素中心向场景发出一条光线 **R**，并求出 **R** 与场景中物体的全部交点；获得离视点最近交点 **P**；并依据局部光照明模型计算 **P** 处颜色值  **$I_{local}$** （光线投射）；
- 在 **P** 处沿着 **R** 镜面反射方向和透射方向各衍生一条光线，若点 **P** 所在表面非镜面或不透明体，则无需衍生出相应光线。

## 光线跟踪算法步骤

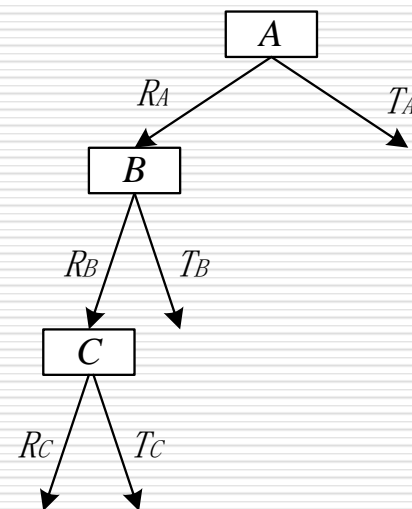
---

- 分别对衍生出的光线递归地执行前面步骤，计算来自镜面反射方向和透射方向上周围环境对点**P**光亮度的贡献 **$I_s$** 和 **$I_t$** ;
- 依据**Whitted**光照明模型即可计算出点**P**处的光亮度，并将计算出的光亮度赋给该像素。

# 光线跟踪算法步骤

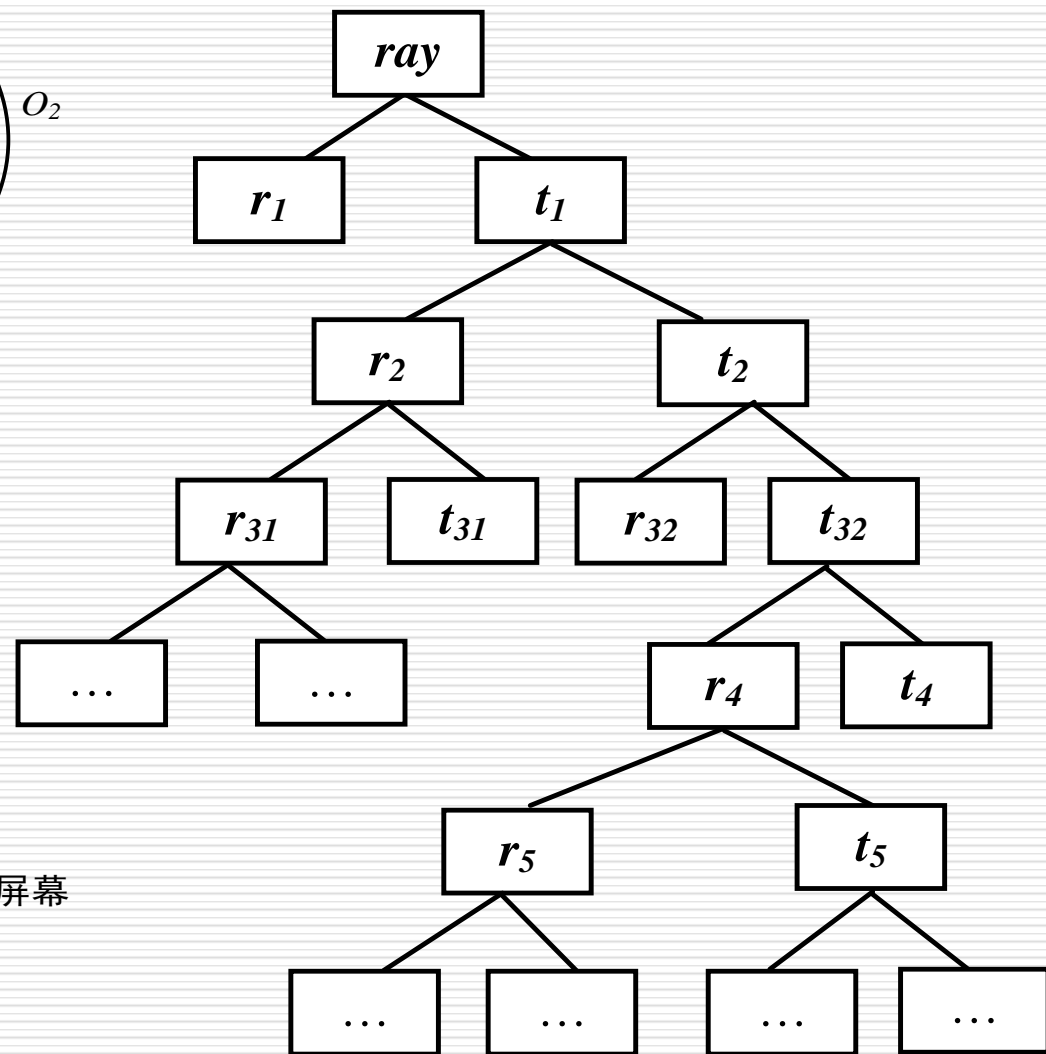
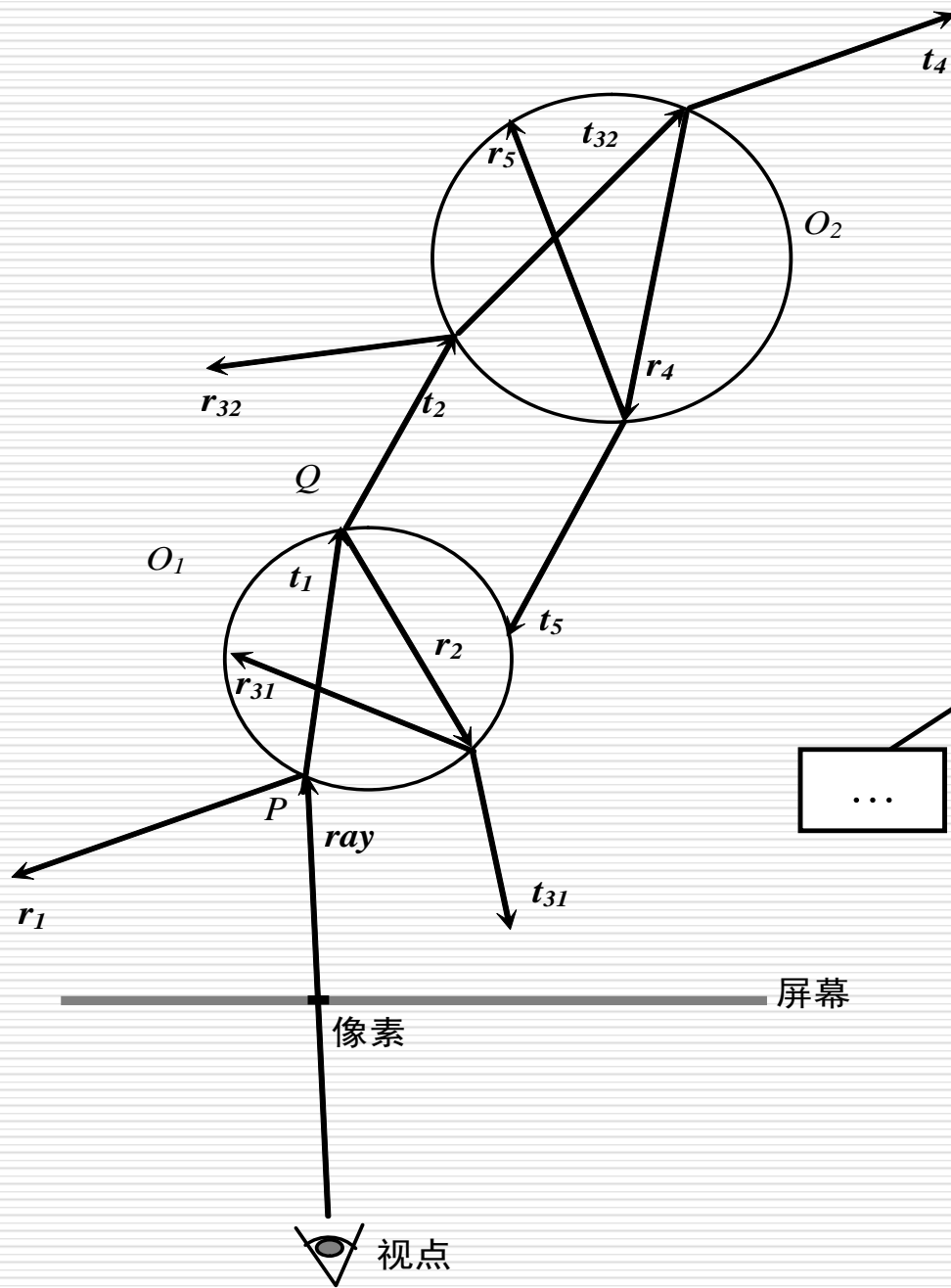


(a) 光线在景物中反射和折射



(b) 二叉光线跟踪树

图8.24 光线跟踪及光线跟踪树



# 光线跟踪算法-实例

---

□ 开源光线跟踪软件**POV-Ray**简介

**<http://www.povray.org>**

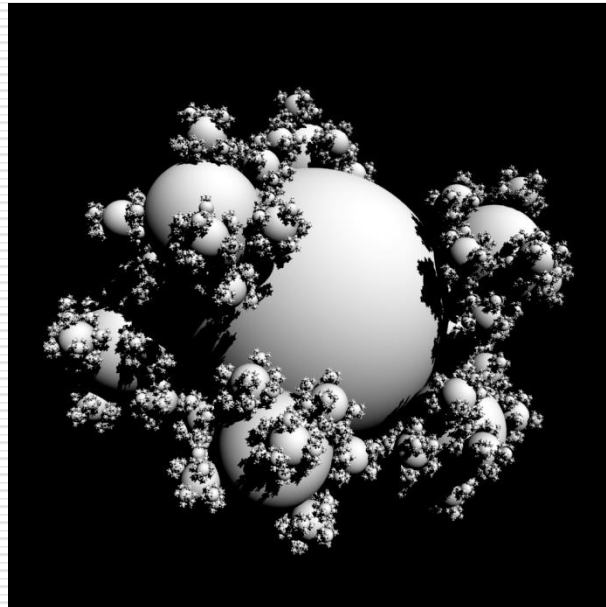


# 光线跟踪算法-实例

---

□ Berger-Perrin T. The sphere flake, in 100 lines of c code.

<http://ompf.org/ray/sphereflake/>



# 实时光线跟踪API- OpenRT

---



# 实时光线跟踪API- OpenRT

---





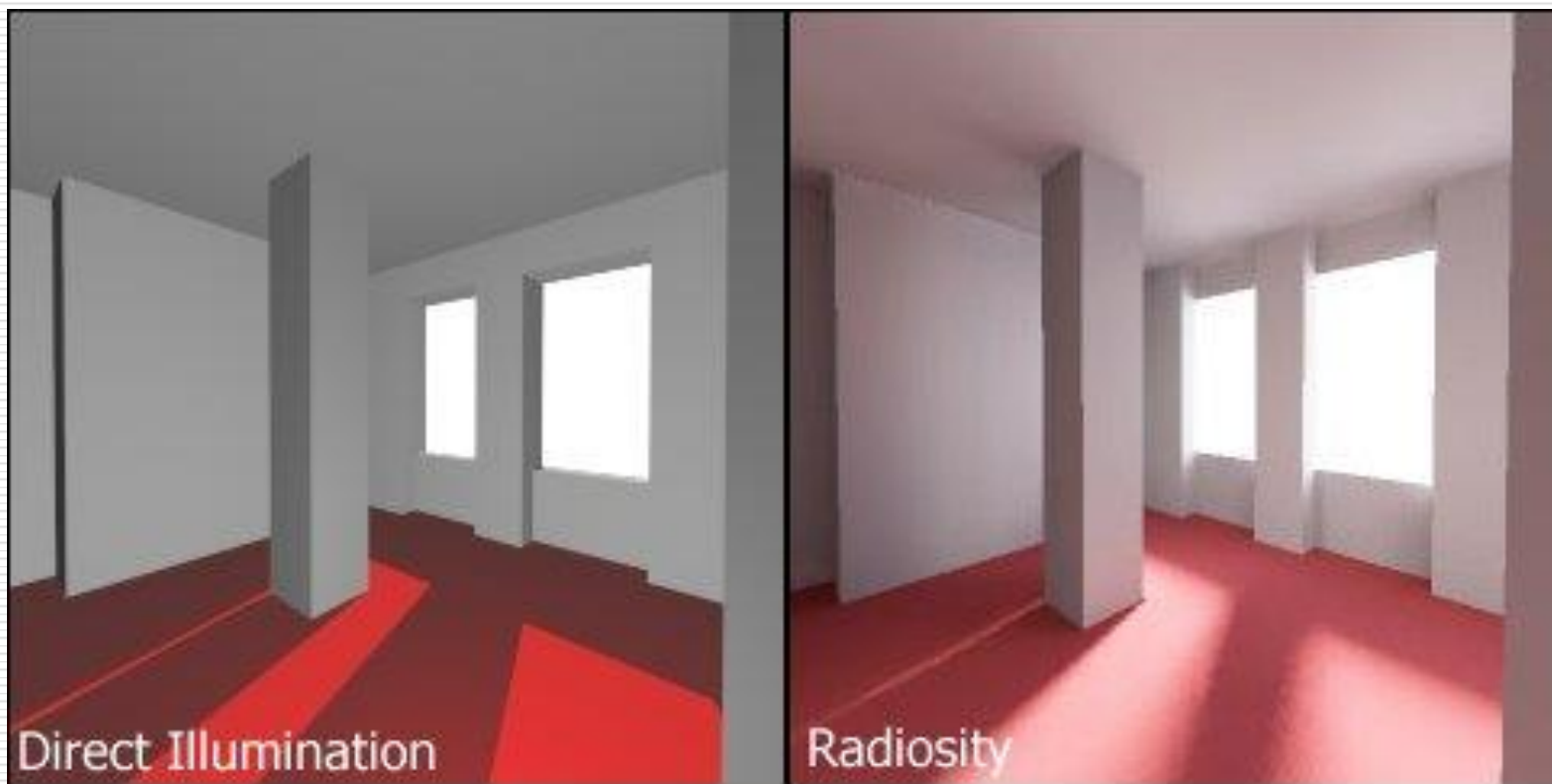
# 实时光线跟踪API- OpenRT

---



# 辐射度算法

## □ 直接照明与辐射度效果对比



# 辐射度算法

---

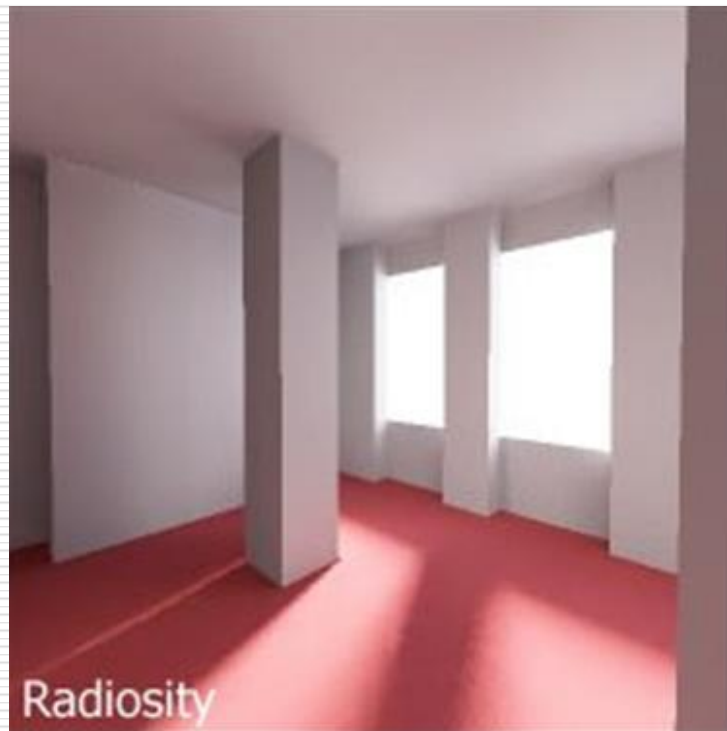
## □ 直接照明

- 聚光灯：能产生阴影，放置于窗外。
- 泛光：缺少泛光，房间内所有可见但未被光源直接照射的表面会是全黑的。
- 点光源：不产生阴影，可减弱泛光产生的“平板”效果。



# 辐射度算法

- 采用辐射度算法绘制，场景中仅包含一个面光源，即放置于窗外的天空。与直接照明效果相比，主要区别在于：
  - 房间有发光效果
  - 地板上软影清晰可见，房间周围可观察到细微的光效。
  - 地毯上的红色辉映到灰墙上，产生淡淡的温暖的感觉。



# 辐射度算法

---

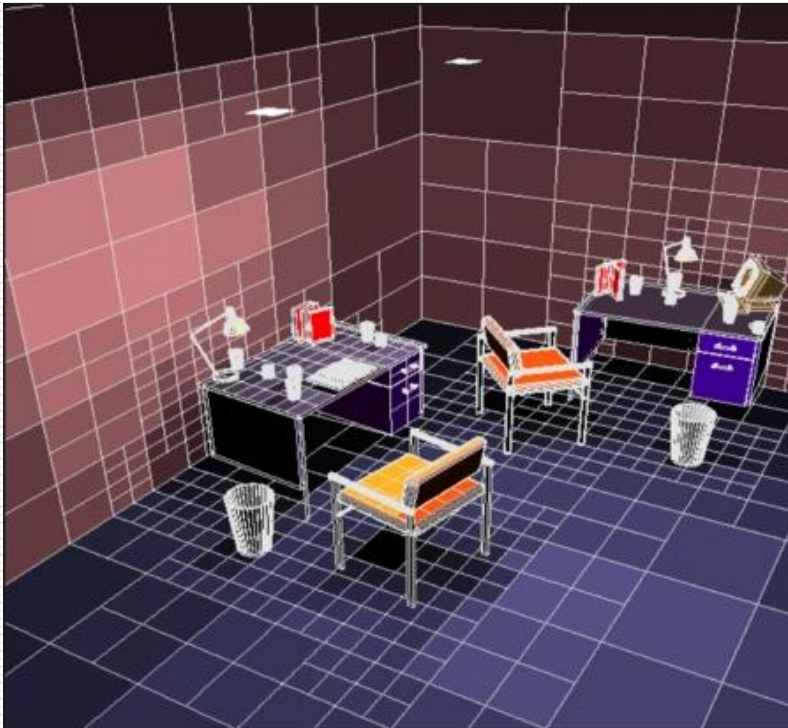
## □ 原理

- 光是一种辐射能，在一个封闭环境中，场景中的光能经过表面之间的反射和透射，最终达到平衡状态
- 场景中各表面的光亮度实际上是场景中光能分布的反映。

□ 辐射度：单位时间内从物体单位表面积向外辐射的光能。

# 辐射度算法

---



## 8.7 模拟景物表面细节

---

- 用多边形模拟表面细节
- 纹理的定义和映射
- 凹凸映射

## 多边形模拟表面细节

---

- ❑ 简单地模拟景物表面细节的方法是用多边形，称为表面图案多边形，来模拟纹理的结构和模式。
- ❑ 处理时，首先根据待生成的颜色纹理构造表面图案多边形，然后将表面图案多边形覆盖到物体的表面上。



# 多边形模拟表面细节

- 颜色纹理：通过颜色色彩或明暗度的变化体现出来的表面细节。
- 几何纹理：由于不规则的细小凹凸造成的。

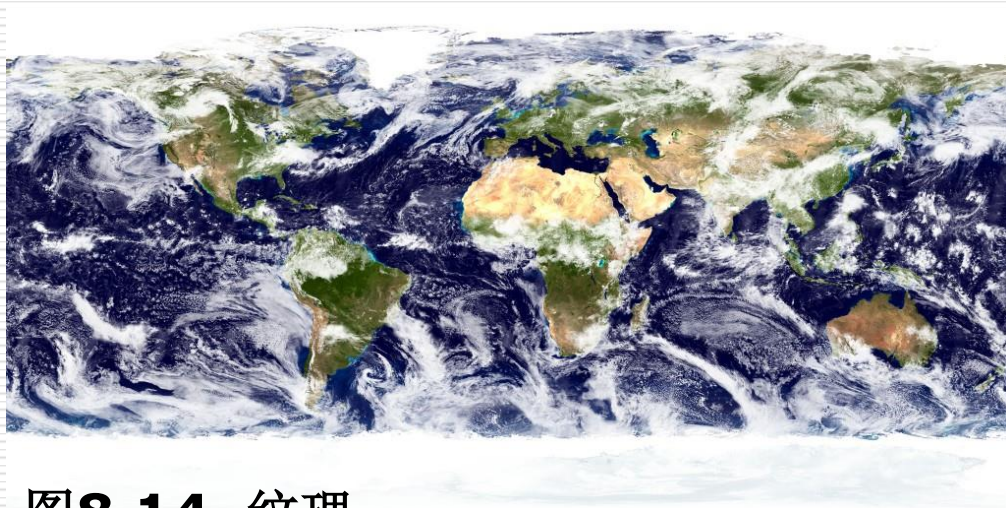


图8.14 纹理

# 纹理映射和定义

---

## □ 纹理映射（**Texture Mapping**）

- 预先定义纹理模式
- 建立物体表面的点与纹理模式的点之间的对应关系
- 当物体表面的可见点确定之后，以纹理模式的对应点参与光照模型进行计算，就可把纹理模式附到物体表面上

# 纹理映射和定义

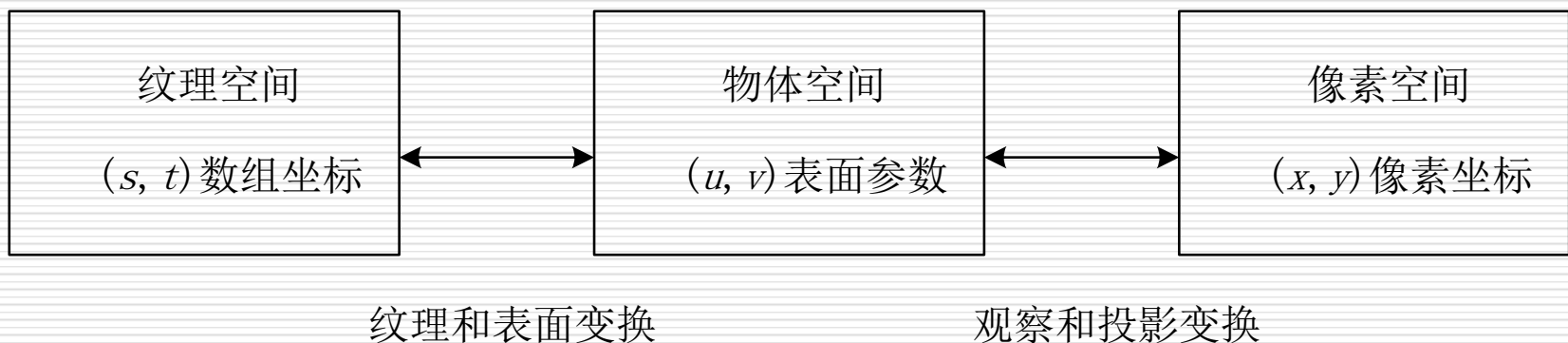
---

## □ 纹理映射的方法

- 将纹理模式映射至物体表面，然后再映射至投影平面
- 将像素区域映射至物体表面再映射至纹理空间

# 纹理映射和定义

---



**图8.15** 纹理映射中纹理空间、物体空间和像素空间的变换

# 纹理映射和定义

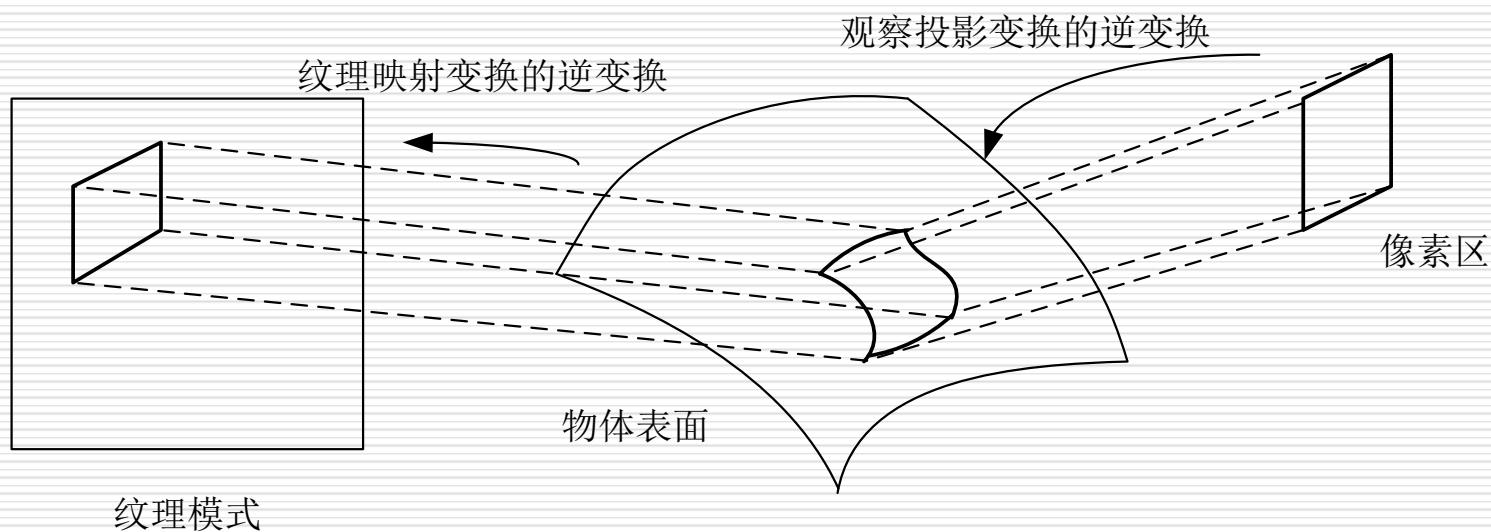


图8.16 由像素空间向纹理空间的映射

## 凹凸映射

---

- 1978年，**Blinn**提出了一种无需修改表面几何模型，即能模拟表面凹凸不平效果的有效方法——凹凸映射技术（**Bump Mapping**）。
- 假定用 $\mathbf{P}(\mathbf{u}, \mathbf{v})$ 表示一个参数曲面上的点，可以通过计算得到该点处的表面法矢量为：

$$N_P = P_u \times P_v$$

$\mathbf{P}_u$ 与 $\mathbf{P}_v$ 为 $\mathbf{P}$ 关于参数 $\mathbf{u}$ 和 $\mathbf{v}$ 的偏导数

## 凹凸映射

---

- 在景物表面每一采样点处沿其法矢量附加一微小增量，从而生成一张新的表面  $Q(u, v)$ ，它可表示为：

$$Q(u, v) = P(u, v) + b(u, v)n = P(u, v) + b(u, v) \frac{N}{|N|}$$

式中  $b(u, v)$  为用户定义的扰动函数

## 凹凸映射

---

- 扰动函数 **$b(u, v)$** 可以任意选择
  - 简单的网格图案
  - 字符位映射
  - **Z**缓存器图案
  - 随意手描图案
  - 用一个二维 **$u, v$** 查询表列出离散点处的值，用双线性插值方法获得