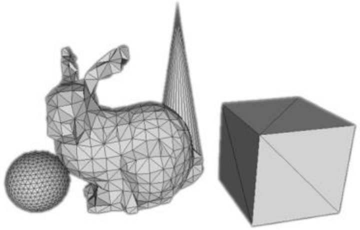
Introduction to Computer Graphics



Ray Tracing
光线跟踪

第七章 光线跟踪

光线跟踪基本原理

Whitted-Style 光线跟踪

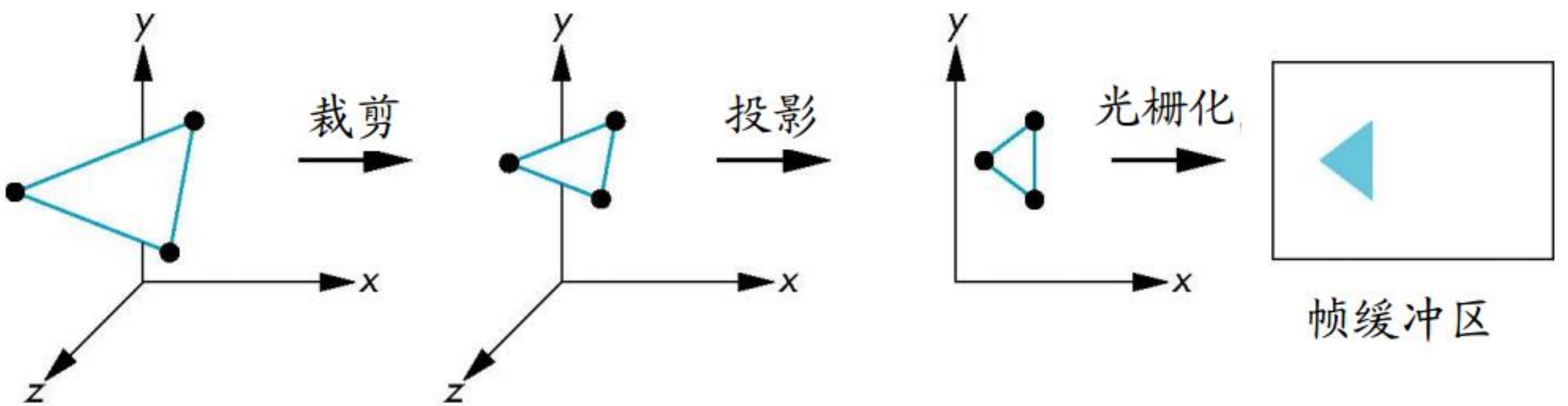
光线跟踪实现细节

光线跟踪进阶话题

两类绘制方式

- 对于每个对象，确定它所覆盖的像素，并用对象的状态确定像素的明暗值
  - 流水线方法
  - 必须跟踪深度值

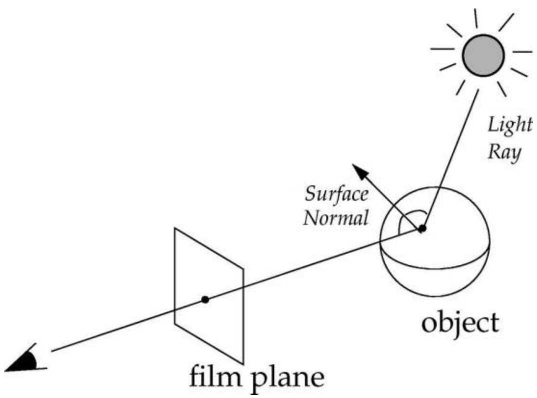
```
for (each_object)
    render(object)
```



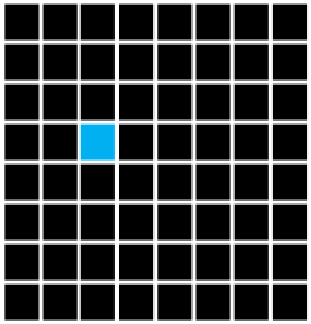
两类绘制方式

- 对于每个像素，确定投影到这个像素的离观察者最近的那个对象，从而基于该对象计算像素的明暗值
  - 光线跟踪框架

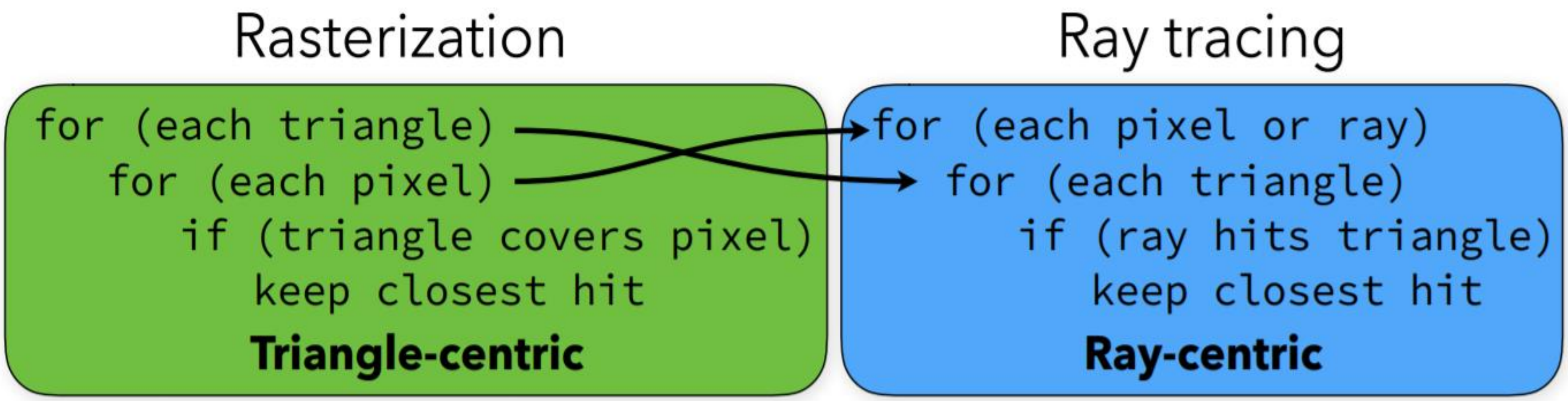
for (each_pixel)
assign_a_color(pixel)



(ray intersection rendered to screen)

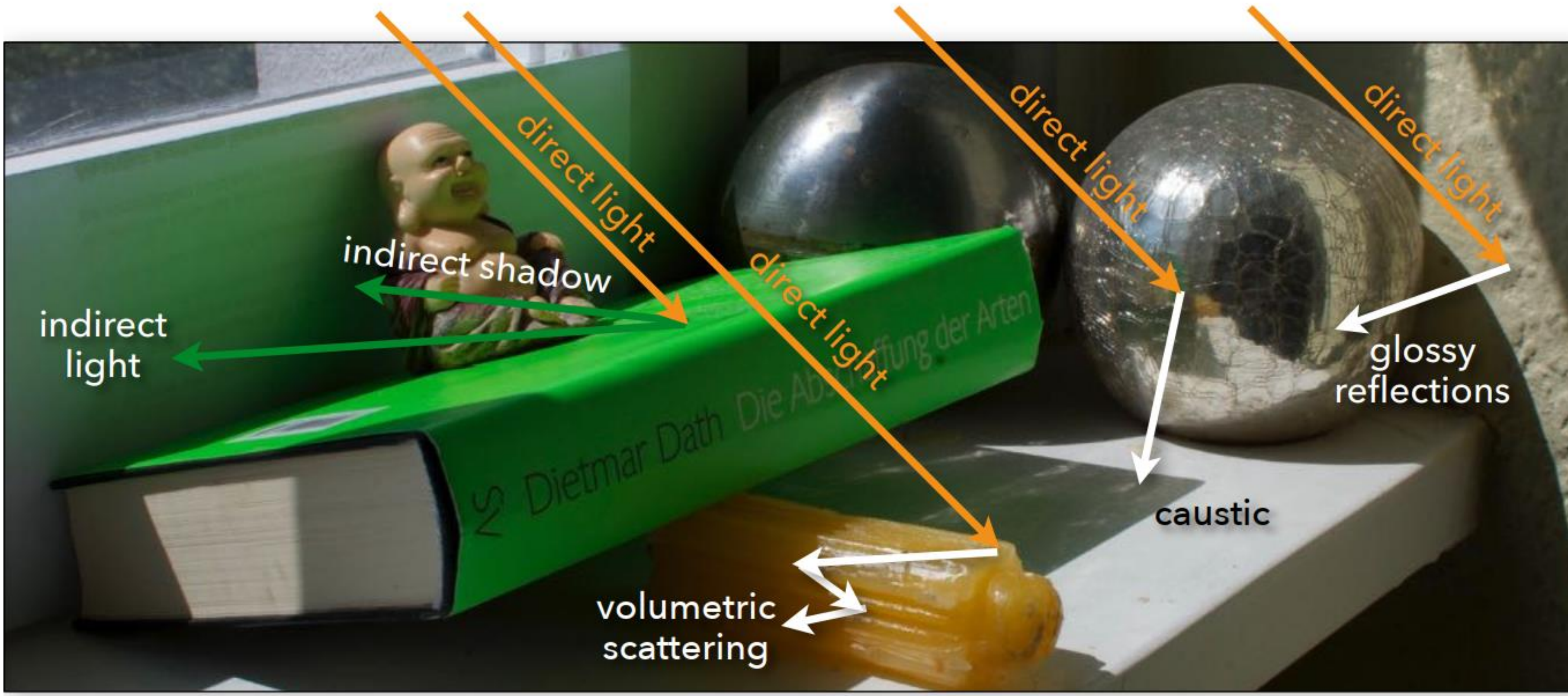


两类绘制方式



Rasterization
Ray tracing
for (each triangle)
for (each pixel)
if (triangle covers pixel)
keep closest hit
Triangle-centric
for (each pixel or ray)
for (each triangle)
if (ray hits triangle)
keep closest hit
Ray-centric

真实世界中的光线传播

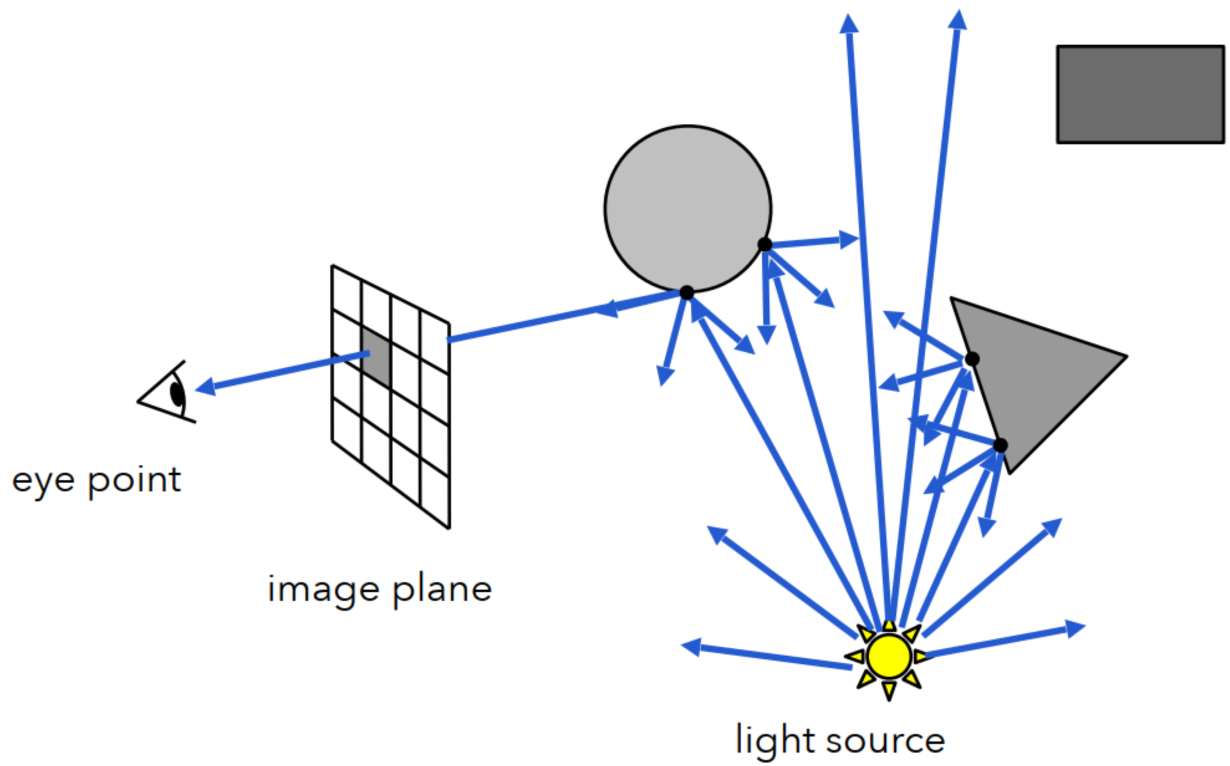


After [Ritschel et al 2011]

光线跟踪基本原理

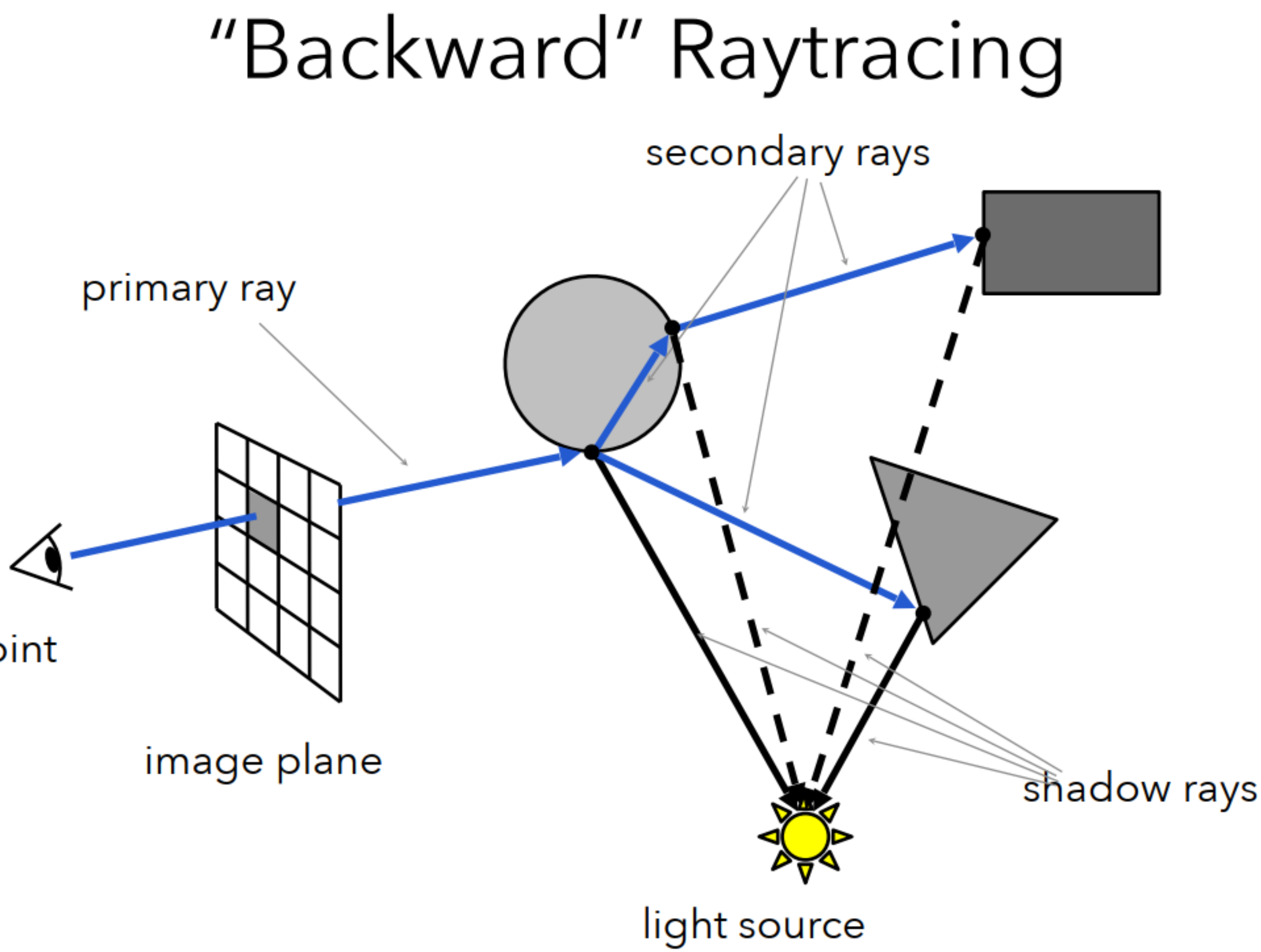
- 光线跟踪思路

"Forward" Raytracing



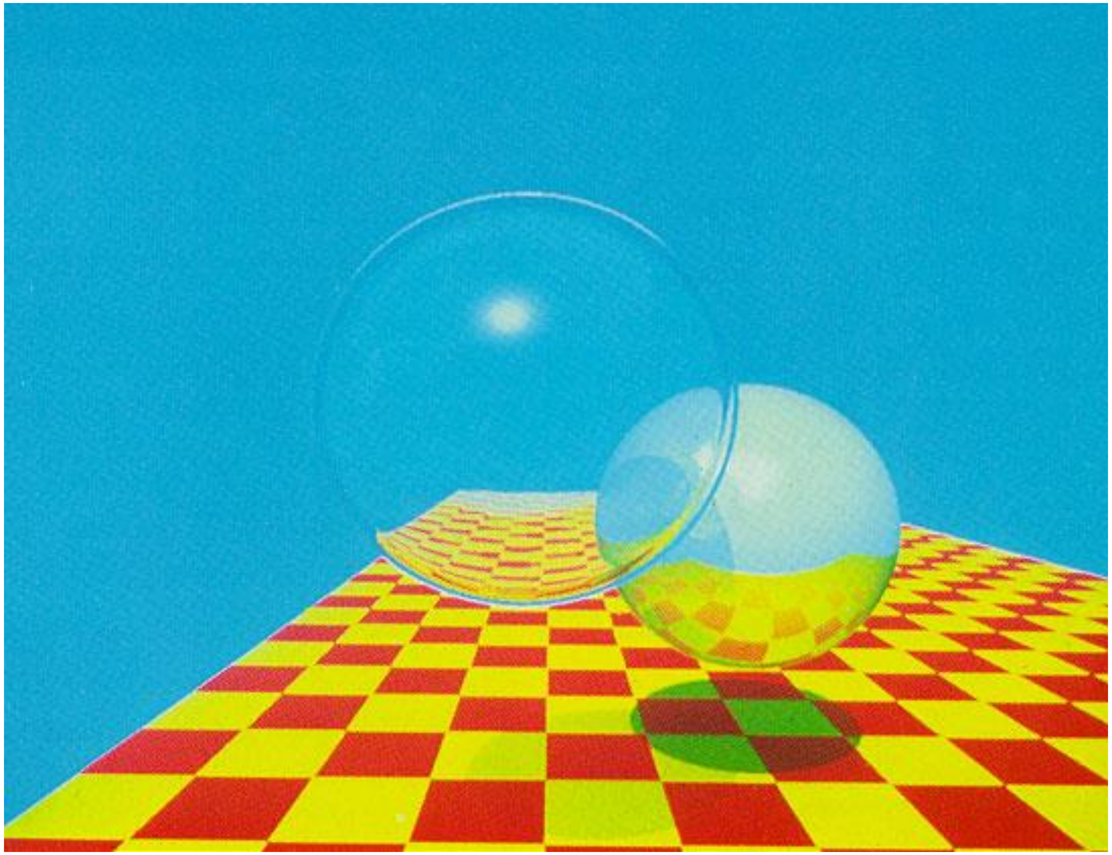
光线跟踪基本原理

- 光线跟踪思路

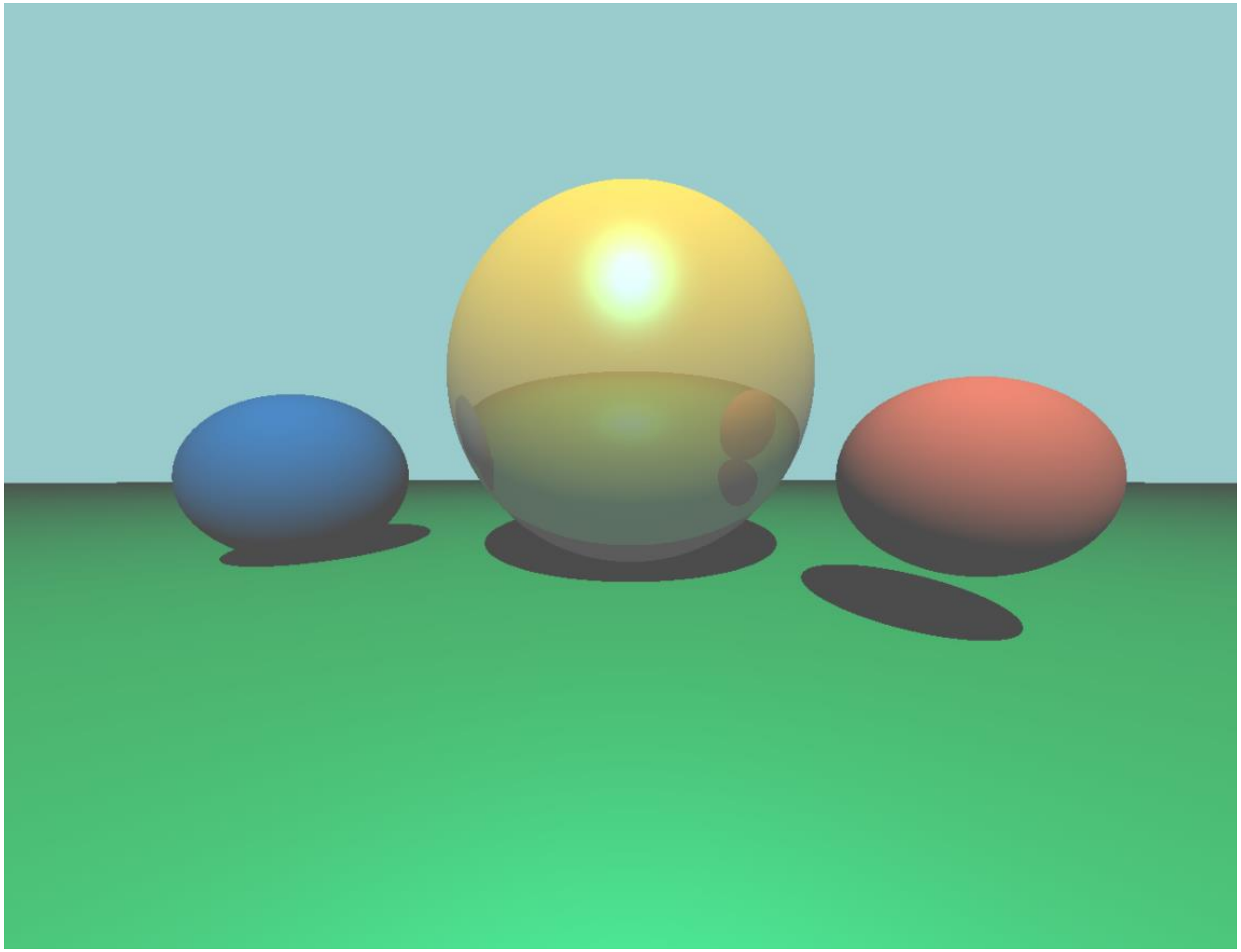


光线跟踪效果

- Whitted 1980



光线跟踪效果



光线跟踪效果

- PBRT

光线跟踪效果

- PBRT

光线跟踪效果

- 渲染？照片？

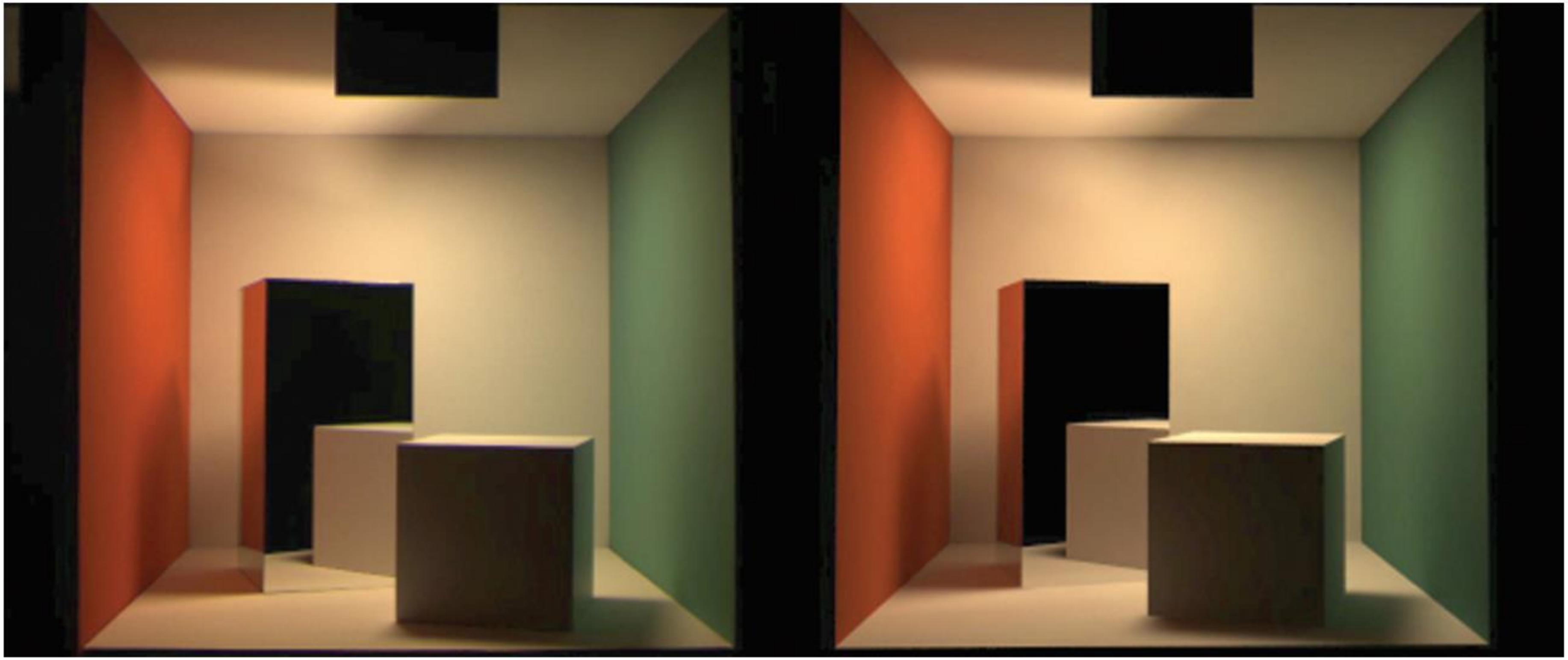
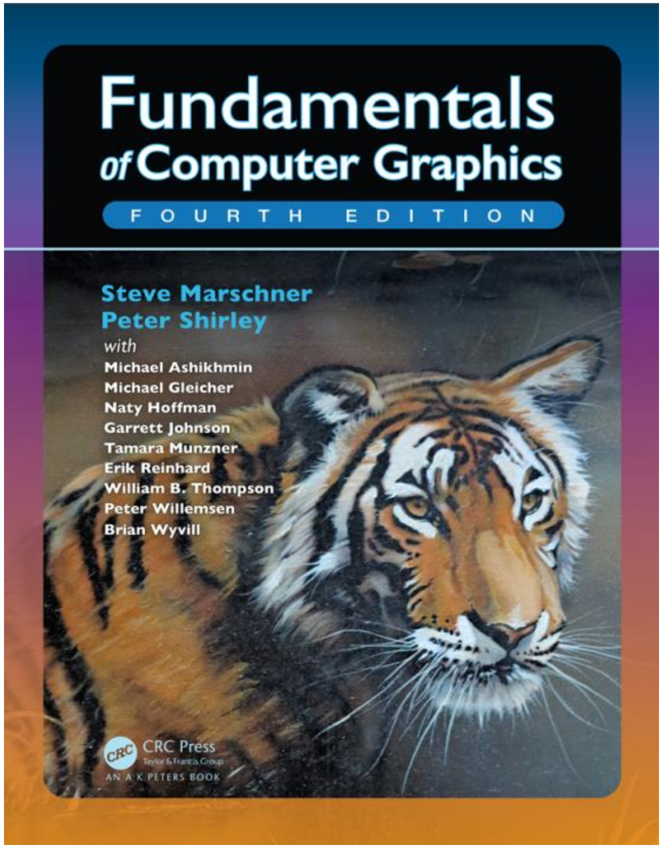
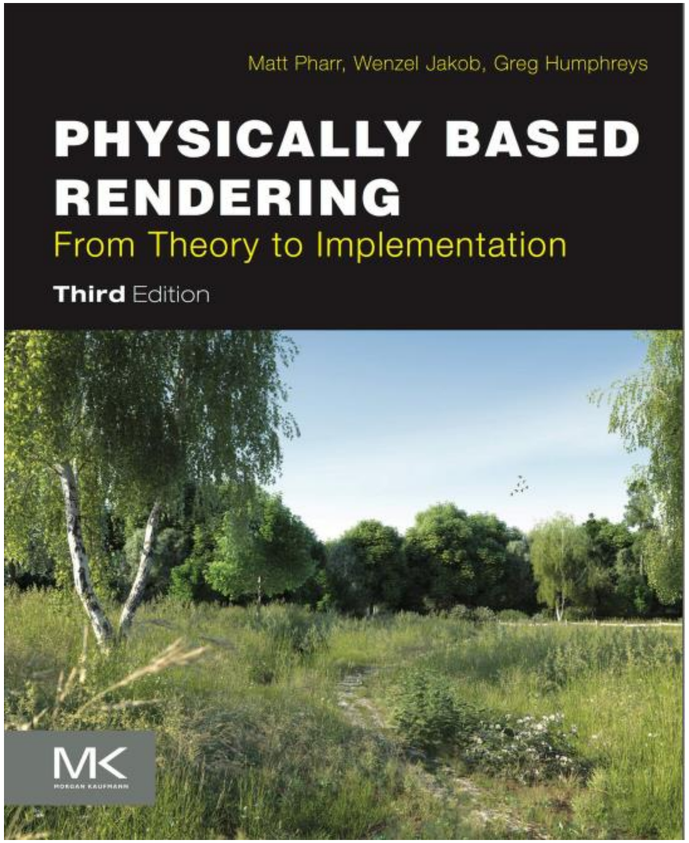


Image courtesy Sumant Pattanaik and the Cornell Program of Computer Graphics.

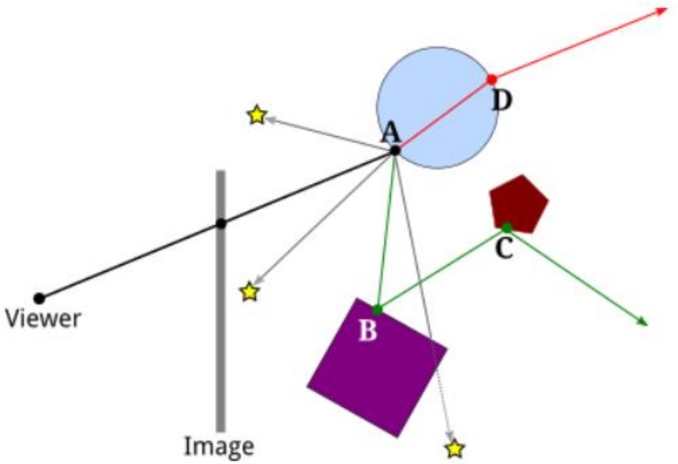
光线跟踪参考资料


Fundamentals
of Computer Graphics
FOURTH EDITION
Steve Marschner
Peter Shirley
with
Michael Ashikhmin
Michael Gleicher
Naty Hoffman
Garrett Johnson
Tamara Munzner
Erik Reinhard
William B. Thompson
Peter Willemsen
Brian Wyvill
CRC Press
AN A K PETERS BOOK


Matt Pharr, Wenzel Jakob, Greg Humphreys
PHYSICALLY BASED
RENDERING
From Theory to Implementation
Third Edition
MK

光线跟踪算法

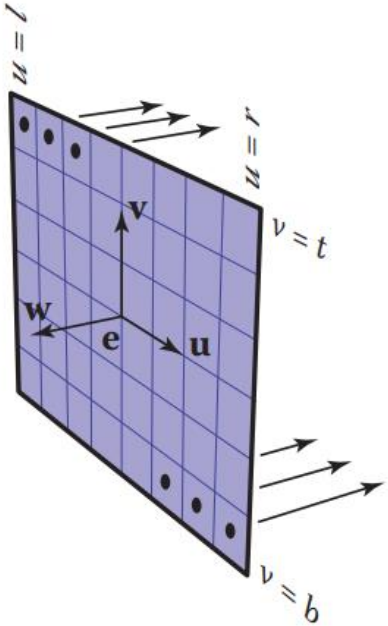
- 光线跟踪算法框架
  - Viewing (Primary) Ray
  - Secondary Ray
  - Shadow Ray



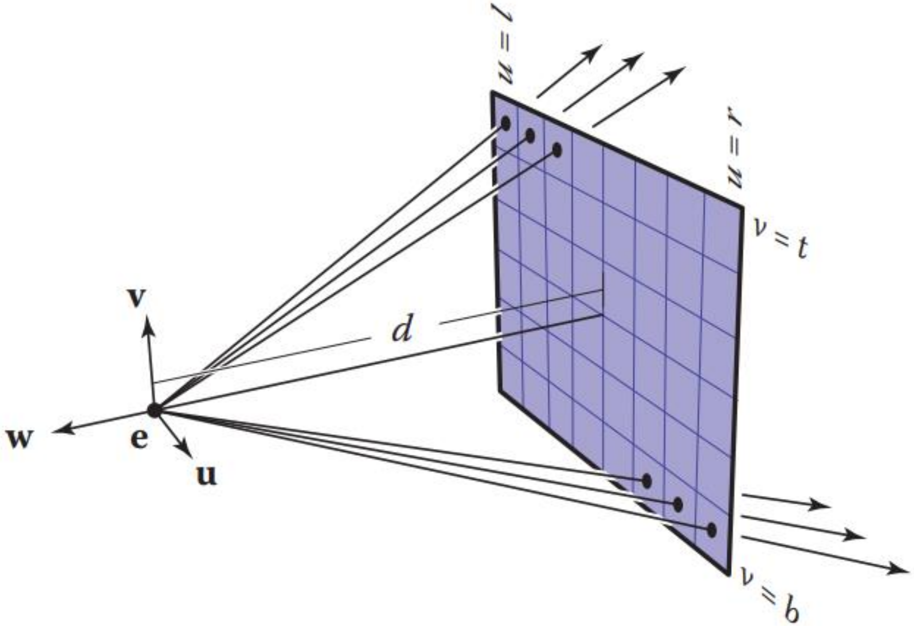
for each pixel **do**
  compute viewing ray
  if (ray hits an object with $t \in [0, \infty)$) **then**
    Compute **n**
    Evaluate shading model and set pixel to that color
  else
    set pixel color to background color

光线的生成

- 给定相机中心，成像平面左下角，成像平面竖直方向，成像平面水平方向



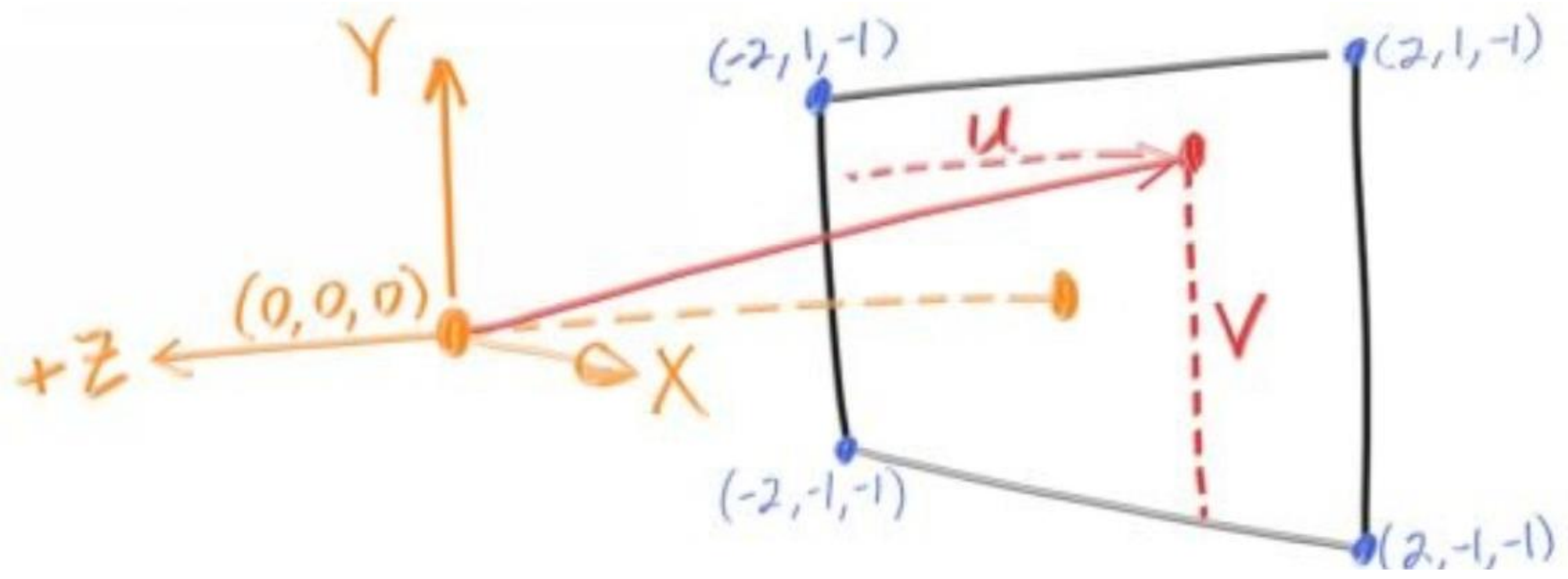
Parallel projection
same direction, different origins



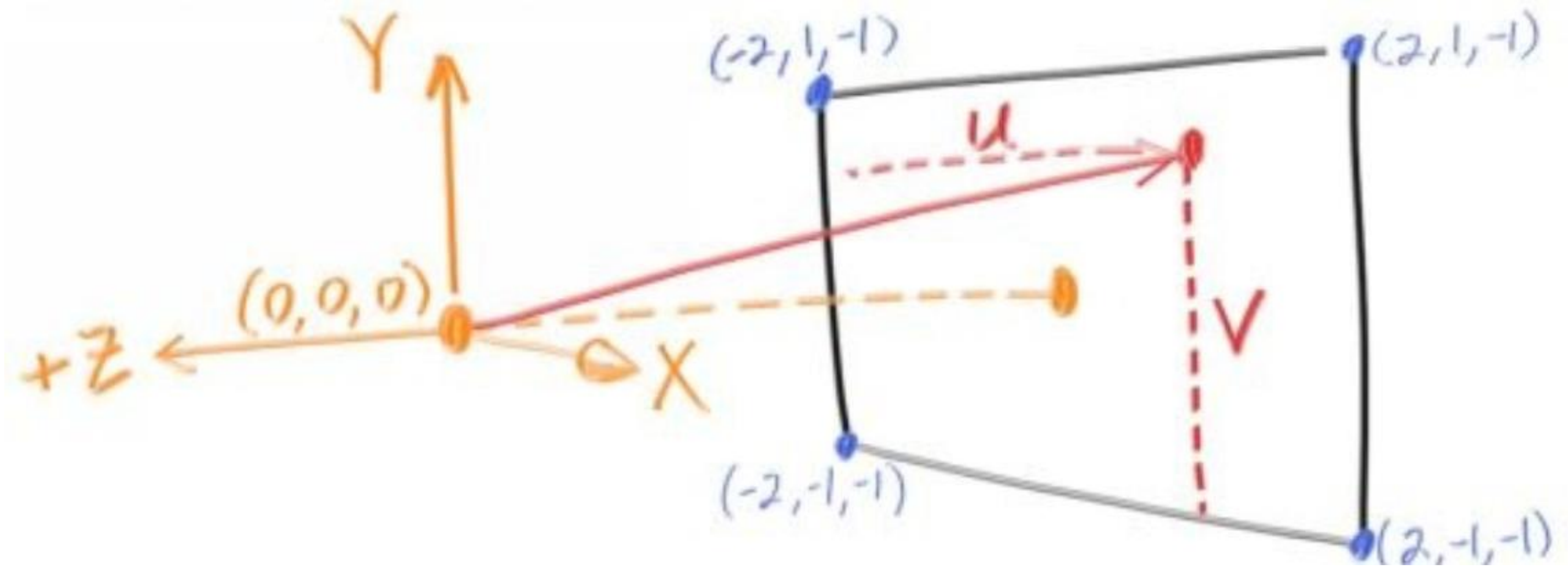
Perspective projection
same origin, different directions

光线的生成

- 如何与像素坐标联系起来?



光线的生成



光线与场景求交

- 光线与球面求交
  - 光线方程：

$$\mathbf{p}(t) = \mathbf{e} + t\mathbf{d}$$

  - 球面方程：

$$(\mathbf{p} - \mathbf{c}) \cdot (\mathbf{p} - \mathbf{c}) - R^2 = 0$$

  - 带入求解：

$$(\mathbf{e} + t\mathbf{d} - \mathbf{c}) \cdot (\mathbf{e} + t\mathbf{d} - \mathbf{c}) - R^2 = 0.$$

光线与场景求交

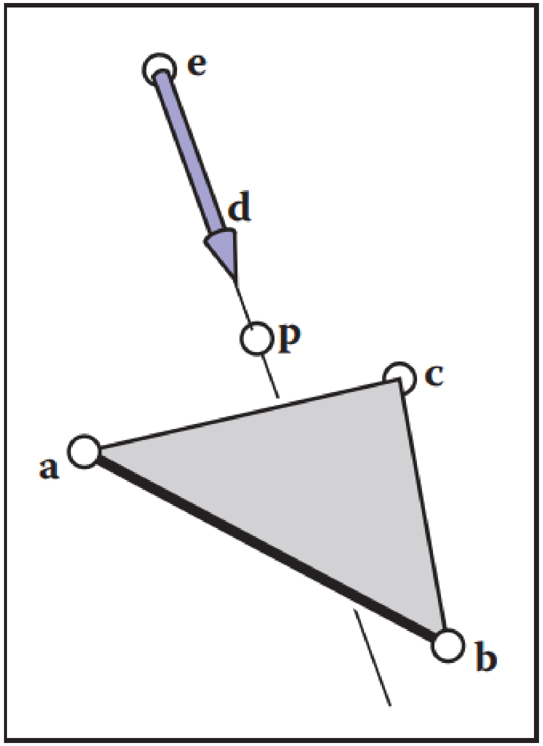
- 光线与球面求交

$$(\mathbf{d}\cdot\mathbf{d})t^2 + 2\mathbf{d}\cdot(\mathbf{e}-\mathbf{c})t + (\mathbf{e}-\mathbf{c})\cdot(\mathbf{e}-\mathbf{c}) - R^2 = 0$$

$$t = \frac{-\mathbf{d}\cdot(\mathbf{e}-\mathbf{c}) \pm \sqrt{(\mathbf{d}\cdot(\mathbf{e}-\mathbf{c}))^2 - (\mathbf{d}\cdot\mathbf{d})\left((\mathbf{e}-\mathbf{c})\cdot(\mathbf{e}-\mathbf{c}) - R^2\right)}}{(\mathbf{d}\cdot\mathbf{d})}.$$

光线与场景求交

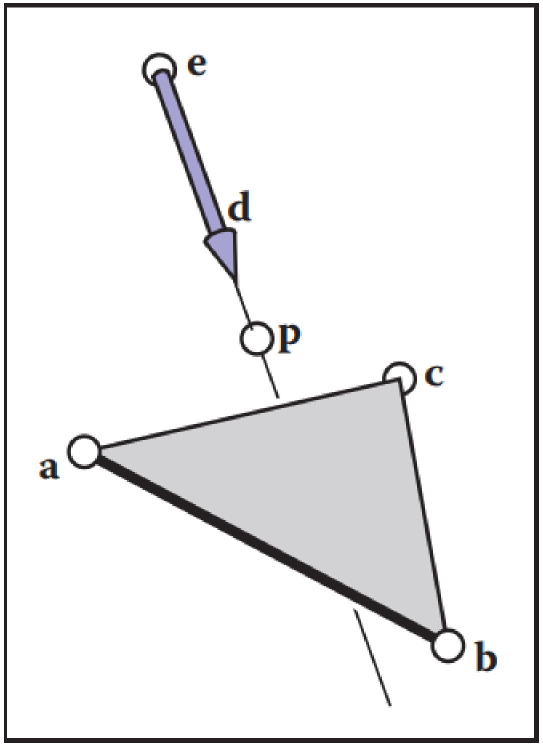
- 光线与三角形求交



$$\mathbf{e} + t\mathbf{d} = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a})$$

光线与场景求交

- 光线与三角形求交



$$x_e + t x_d = x_a + \beta(x_b - x_a) + \gamma(x_c - x_a),$$
$$y_e + t y_d = y_a + \beta(y_b - y_a) + \gamma(y_c - y_a),$$
$$z_e + t z_d = z_a + \beta(z_b - z_a) + \gamma(z_c - z_a).$$

$$\begin{bmatrix} x_a - x_b & x_a - x_c & x_d \\ y_a - y_b & y_a - y_c & y_d \\ z_a - z_b & z_a - z_c & z_d \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \\ t \end{bmatrix} = \begin{bmatrix} x_a - x_e \\ y_a - y_e \\ z_a - z_e \end{bmatrix}$$

光线与场景求交

- 光线与三角形求交

$$\beta = \frac{\begin{vmatrix} x_a - x_e & x_a - x_c & x_d \\ y_a - y_e & y_a - y_c & y_d \\ z_a - z_e & z_a - z_c & z_d \end{vmatrix}}{|\mathbf{A}|},$$

$$\gamma = \frac{\begin{vmatrix} x_a - x_b & x_a - x_e & x_d \\ y_a - y_b & y_a - y_e & y_d \\ z_a - z_b & z_a - z_e & z_d \end{vmatrix}}{|\mathbf{A}|},$$

$$\mathbf{A} = \begin{bmatrix} x_a - x_b & x_a - x_c & x_d \\ y_a - y_b & y_a - y_c & y_d \\ z_a - z_b & z_a - z_c & z_d \end{bmatrix}$$

$$t = \frac{\begin{vmatrix} x_a - x_b & x_a - x_c & x_a - x_e \\ y_a - y_b & y_a - y_c & y_a - y_e \\ z_a - z_b & z_a - z_c & z_a - z_e \end{vmatrix}}{|\mathbf{A}|},$$

像素着色

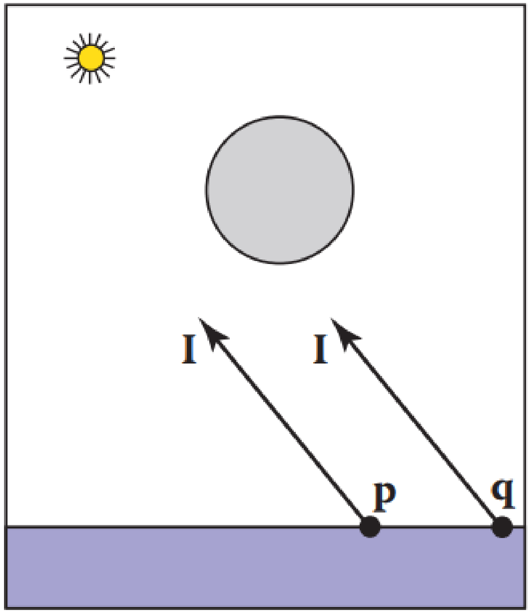
- Whitted着色模型

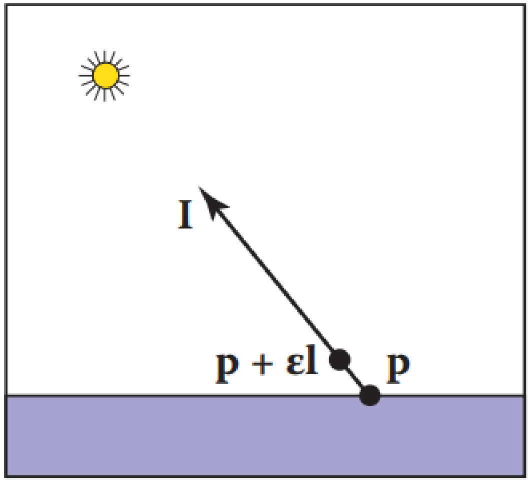
$$I_\lambda = \underbrace{L_{a\lambda}k_aO_{a\lambda}}_{ambient} + \sum_{lights} f_{att}L_{p\lambda}[\underbrace{k_dO_{d\lambda}(\vec{n}\bullet\vec{l})}_{diffuse} + \underbrace{k_sO_{s\lambda}(\vec{r}\bullet\vec{v})^n}_{specular}] + \underbrace{\underbrace{k_sO_{s\lambda}I_{r\lambda}}_{reflected} + \underbrace{k_tO_{t\lambda}I_{t\lambda}}_{transmitted}}_{recursive\ rays}$$

- 前三部分为直接光照
- 后两部分为间接光照

阴影

- Shadow ray





阴影

function raycolor(ray $\mathbf{e} + t\mathbf{d}$, real t_0, real t_1)

hit-record rec, srec

if (scene$\rightarrow$hit($\mathbf{e} + t\mathbf{d}$, t_0, t_1, rec)) **then**

  $\mathbf{p} = \mathbf{e} + (\text{rec}.t)\,\mathbf{d}$

  color $c = \text{rec}.k_a\ I_a$

  if (not scene$\rightarrow$hit($\mathbf{p} + s\mathbf{l}$, ϵ, ∞, srec)) **then**

    vector3 $\mathbf{h} = \text{normalized}(\text{normalized}(\mathbf{l}) + \text{normalized}(-\mathbf{d}))$

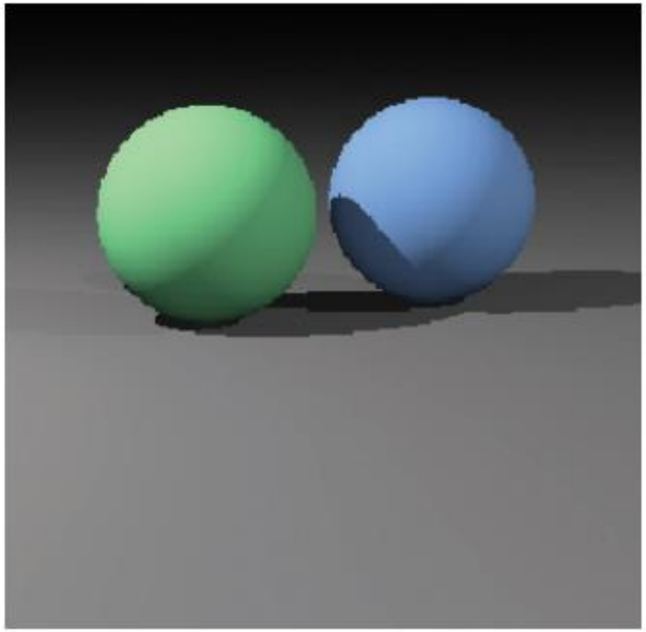
    $c = c + \text{rec}.k_d\, I \max(0, \text{rec}.\mathbf{n} \cdot \mathbf{l}) + (\text{rec}.k_s)\, I\, (\text{rec}.\mathbf{n} \cdot \mathbf{h})^{\text{rec}.p}$

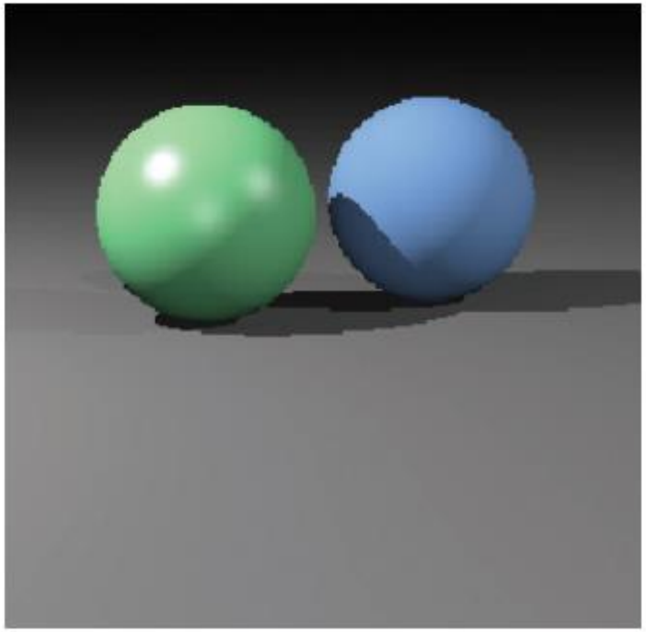
  return c

else

  return background-color

阴影

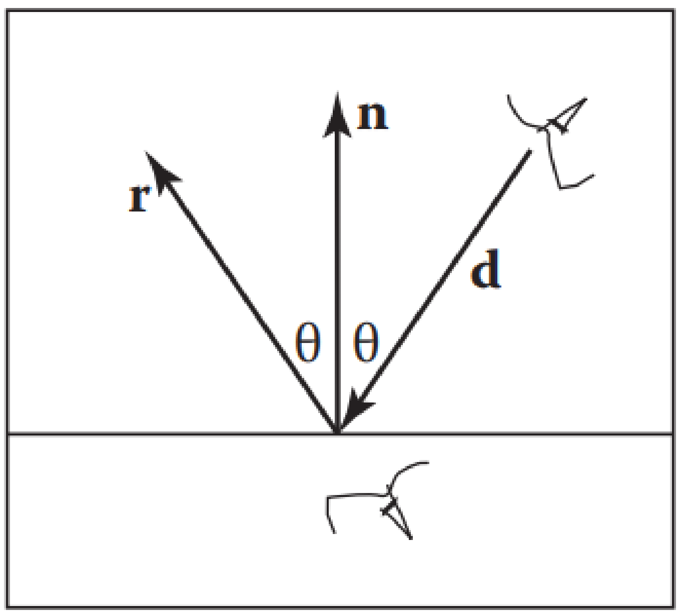


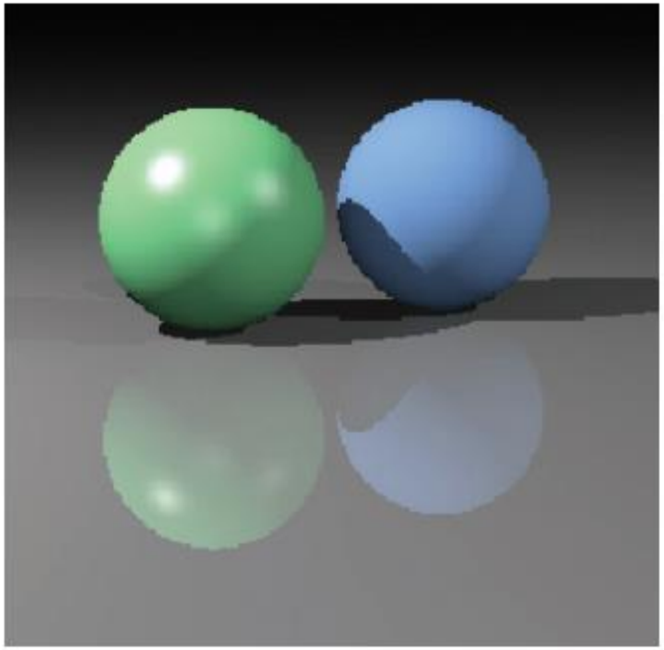


反射

$$\mathbf{r} = \mathbf{d} - 2(\mathbf{d} \cdot \mathbf{n})\mathbf{n}.$$

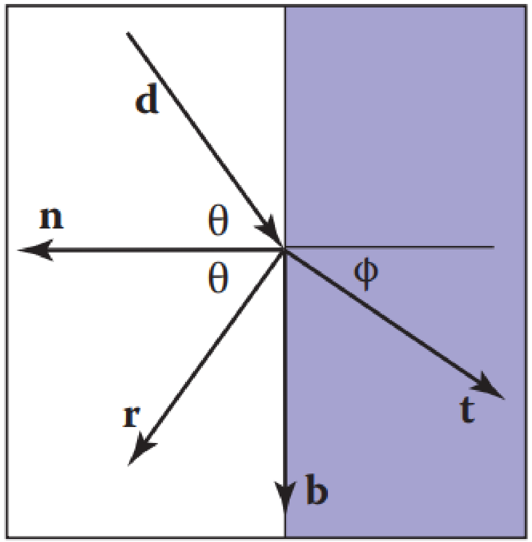
$$\text{color } c = c + k_m \text{raycolor}(\mathbf{p} + s\mathbf{r}, \epsilon, \infty)$$





折射

- Snell's Law



$$n \sin\theta = n_t \sin\phi.$$

$$\begin{aligned}\mathbf{t} &= \frac{n\left(\mathbf{d}+\mathbf{n}\cos\theta\right)}{n_t} - \mathbf{n}\cos\phi \\ &= \frac{n\left(\mathbf{d}-\mathbf{n}(\mathbf{d}\cdot\mathbf{n})\right)}{n_t} - \mathbf{n}\sqrt{1-\frac{n^2\left(1-(\mathbf{d}\cdot\mathbf{n})^2\right)}{n_t^2}}\end{aligned}$$

- 根号内小于零?