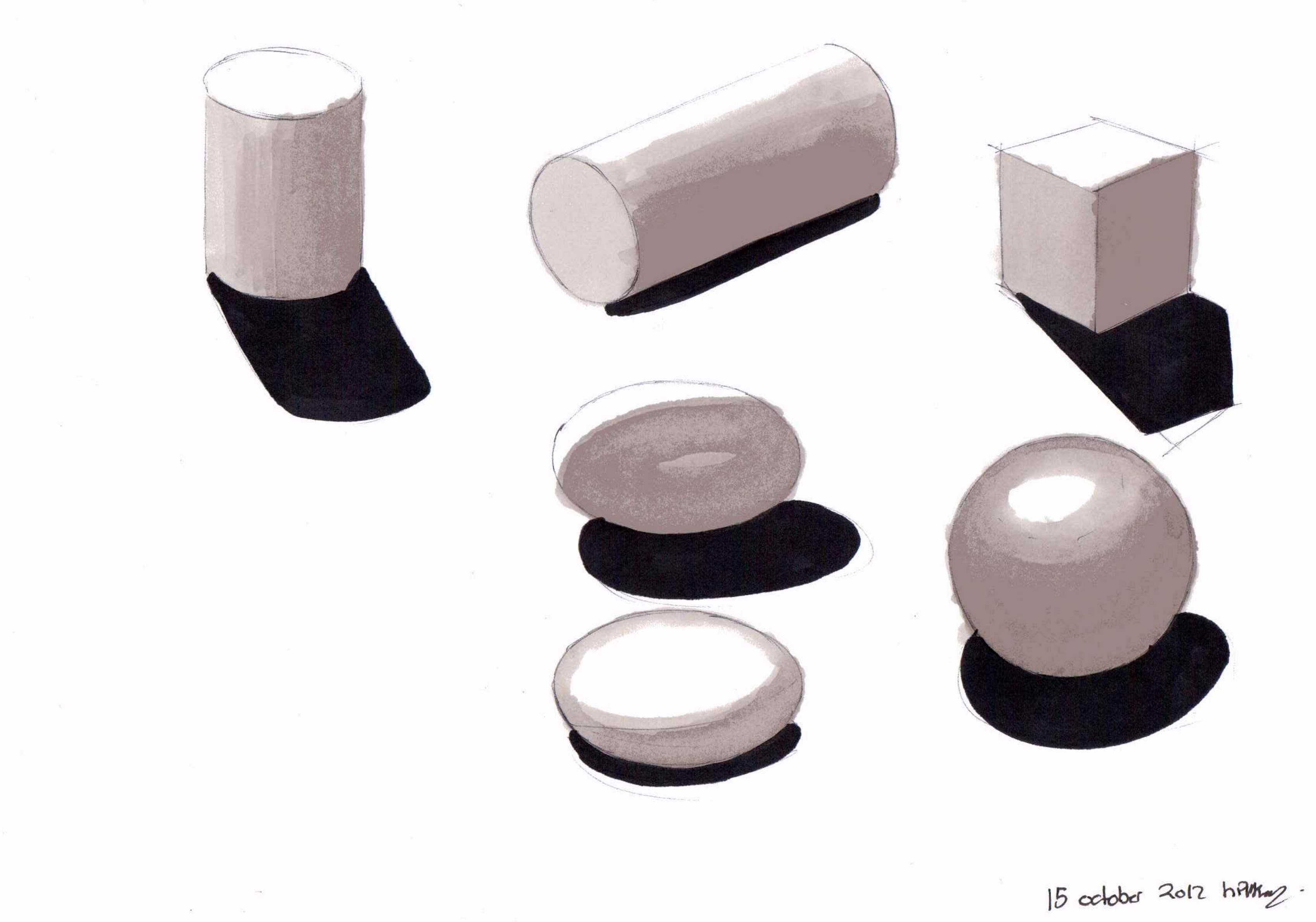


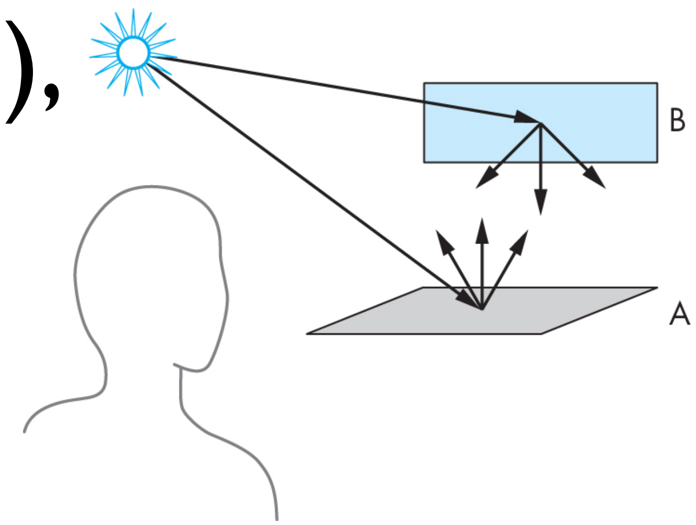
Lighting and Shading

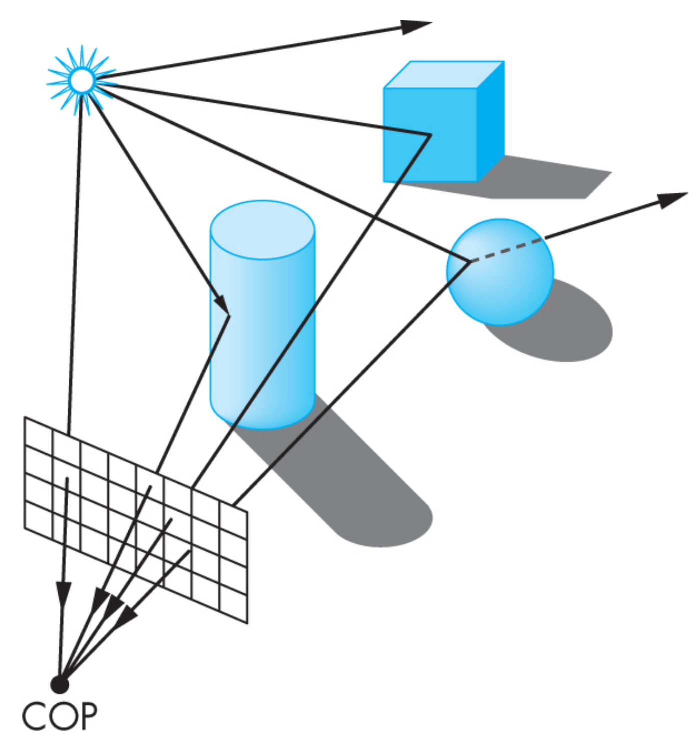
Introduction to Computer Graphics
CSE 533 / 333

Ojaswa Sharma

Light and Matter

- A surface can emit light (self-emission), or reflect light from other sources
- The recursive scattering of light between surfaces accounts for subtle shading effects. This can be expressed in the *rendering equation.*
  - Physically based models
  - Empirical models

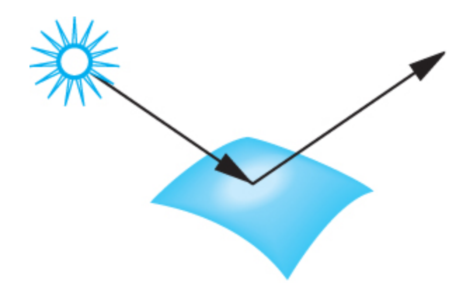




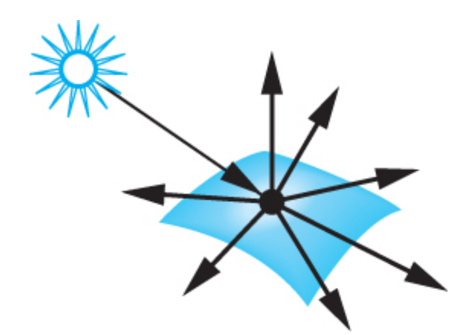
Light and Matter

Surface types based on light-matter interaction:

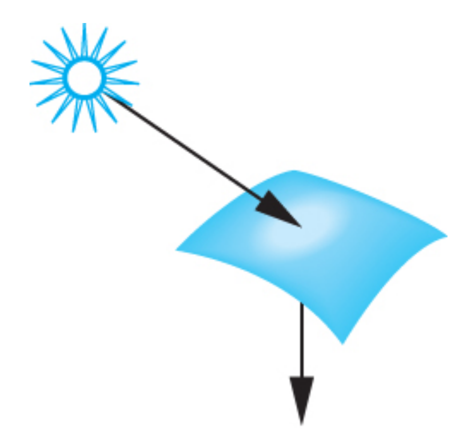
- Specular surfaces: shiny; most of the light scattered in a narrow range of angles close to angle of reflection



- Diffuse surfaces: matte; incoming light is scattered in all directions



- Translucent surfaces: refractive; allow light to penetrate through the surface

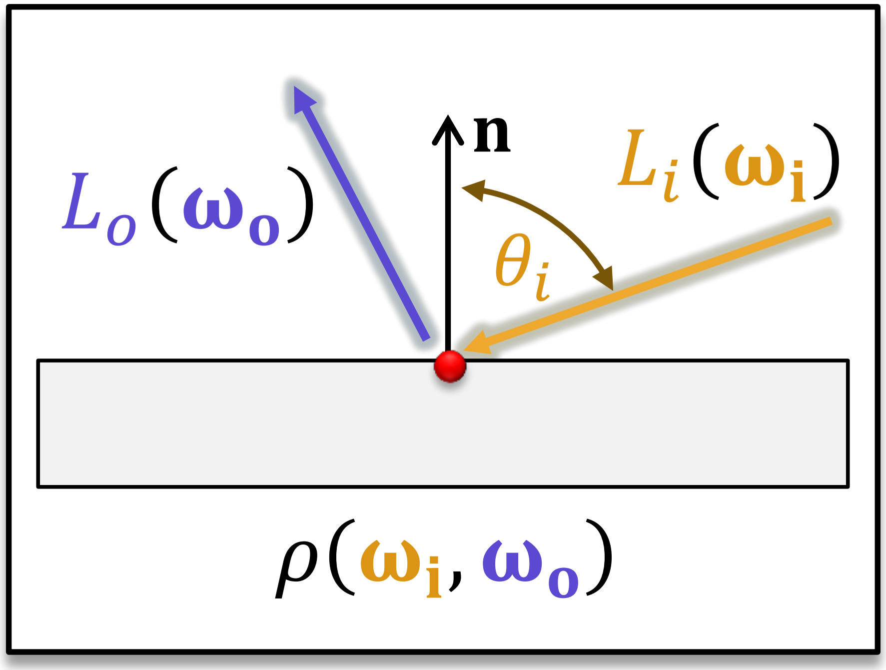


Physically based models

a.k.a. Global Illumination

Definition

$$\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_\mathbf{o}) = \frac{d}{d\boldsymbol{\omega}_i} \frac{L_o(\boldsymbol{\omega}_\mathbf{o})}{L_o(\boldsymbol{\omega}_\mathbf{i}) \cdot \cos\theta_i}$$



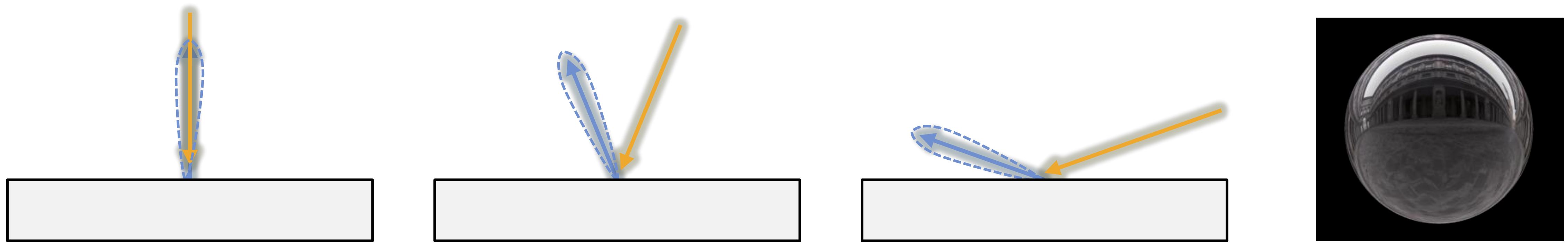
BRDF

- Ratio between
  - Outgoing radiance for angle $\boldsymbol{\omega}_\mathbf{o}$
    and
  - Incoming irradiance*) for angle $\boldsymbol{\omega}_i$
- Density w.r.t. $\boldsymbol{\omega}_i$
  - Integrate over $\boldsymbol{\omega}_i$ to obtain outgoing radiance

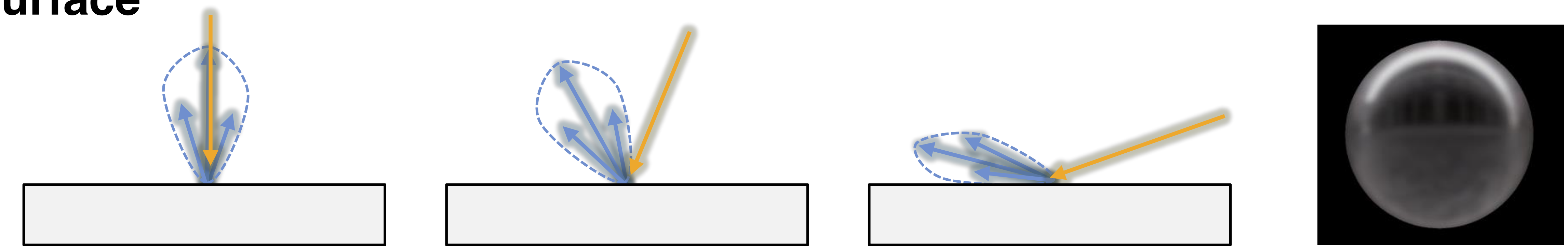
*) **irradiance:** incoming cosine term included
convenient because this is not material specific

Bidirectional Reflectance Distribution Function (BRDF)

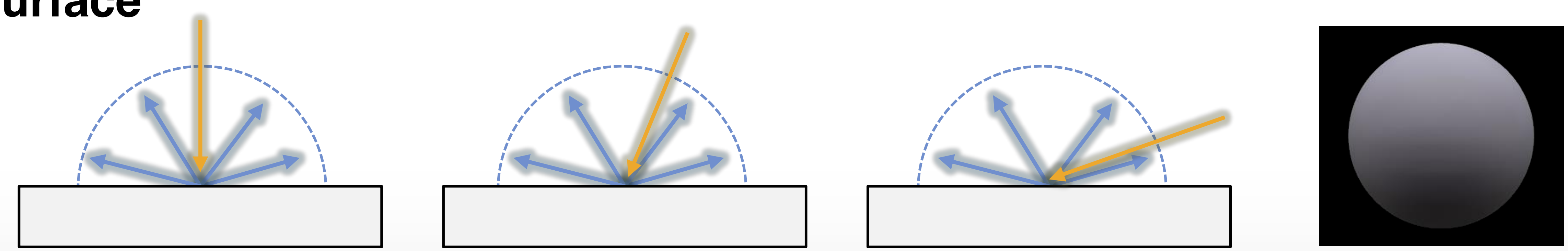
mirror



glossy surface



diffuse surface



Properties of the BRDF

BRDF properties

- Positivity:

$$\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_\mathbf{o}) \geq 0$$

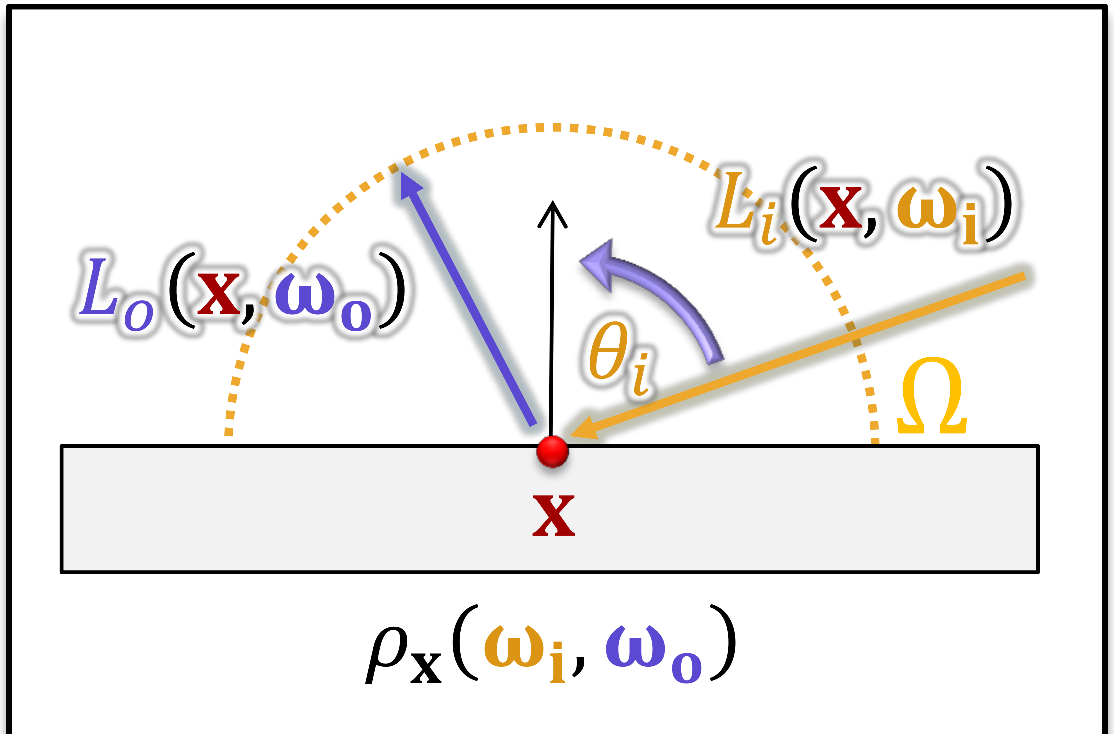
- Helmholtz reciprocity:

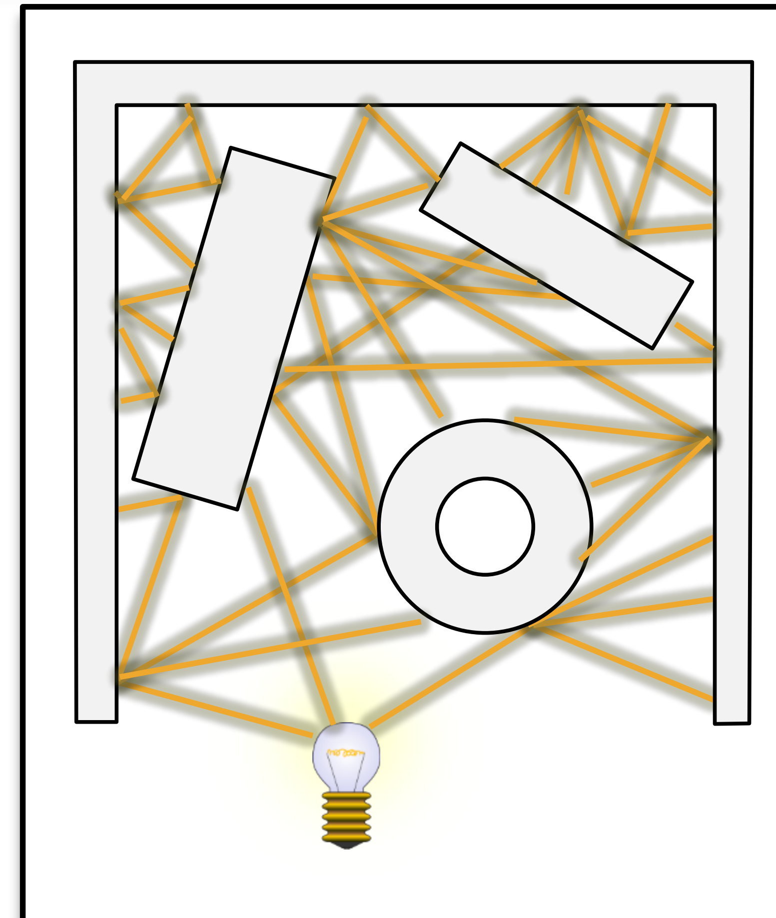
$$\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_\mathbf{o}) = \rho(\boldsymbol{\omega}_\mathbf{o}, \boldsymbol{\omega}_i)$$

- Energy conservation:

$$\forall \boldsymbol{\omega}_i : \int_\Omega \rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_\mathbf{o}) \cos\theta_o \, d\boldsymbol{\omega}_\mathbf{o} \leq 1$$

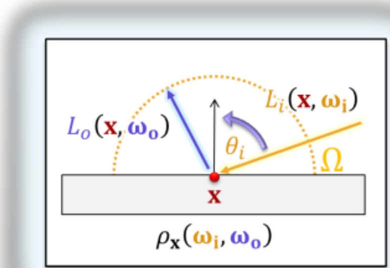
Rendering Equation





Rendering Equation

$$L_o(\mathbf{x}, \boldsymbol{\omega}_\mathbf{o}) = \underbrace{E_o(\mathbf{x}, \boldsymbol{\omega}_\mathbf{o})}_{\text{emission}} + \underbrace{\int_{\omega_i \in \Omega} \left[L_i(\mathbf{x}, \boldsymbol{\omega}_\mathbf{i}) \cdot \rho_\mathbf{x}(\boldsymbol{\omega}_\mathbf{i}, \boldsymbol{\omega}_\mathbf{o}) \cdot \cos \theta_\mathrm{i} \right] d\omega_i}_{\text{reflection}}$$



Structure

Rendering equation

$$L_o(\mathbf{x}, \boldsymbol{\omega}_\mathbf{o}) = E_o(\mathbf{x}, \boldsymbol{\omega}_\mathbf{o}) + \int_{\omega_i \in \Omega} L_i(\mathbf{x}, \boldsymbol{\omega}_\mathbf{i}) \cdot \rho_\mathbf{x}(\boldsymbol{\omega}_\mathbf{i}, \boldsymbol{\omega}_\mathbf{o}) \cdot \cos\theta_\mathrm{i} \; d\omega_i$$

Abstract notation

- Unknown function *L* is related to linear operation on itself + emission function *E*
- Discrete:
  - Functions = vectors, linear operators = matrices
  - $L = E + \mathbf{K} \cdot L$
  - Linear system of equations
- Rendering: solving linear systems of equations

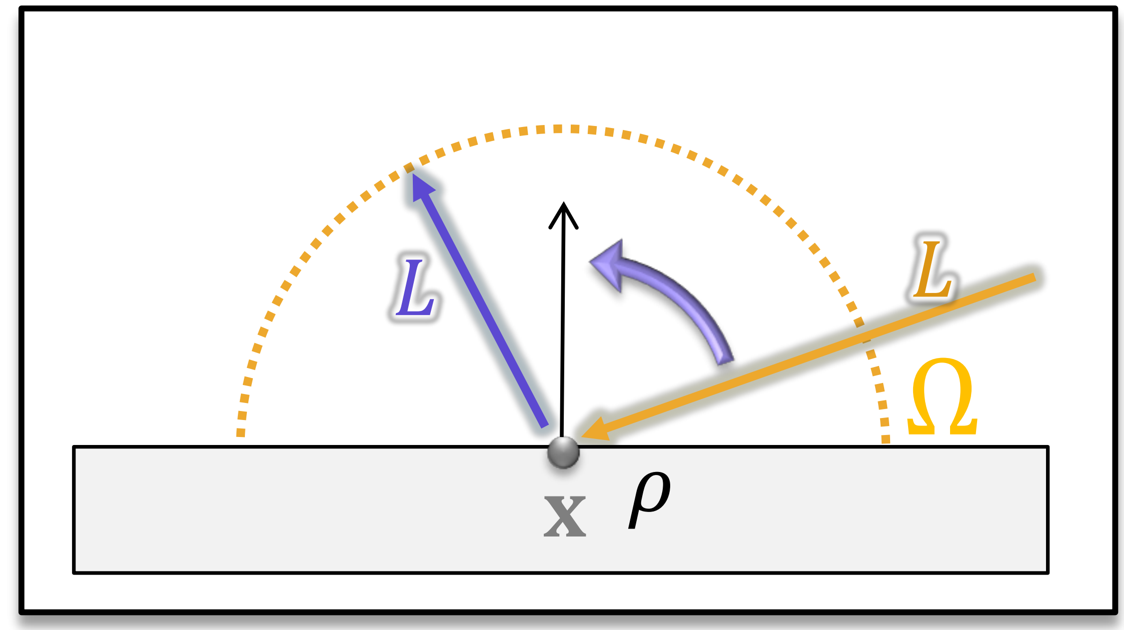
Operator Notation

Solving the equation

- Transport operator $\mathbf{K}$
  - Linear operator
- Rendering equation:

$$L = E + \mathbf{K}L$$

$$\Rightarrow \quad L = (1 - \mathbf{K})^{-1}E$$



Operator $\mathbf{K}$:
One light bounce

Operator Notation

Rendering equation

$$L = E + \mathbf{K}L$$

$$\Rightarrow \quad L = (1 - \mathbf{K})^{-1} E$$

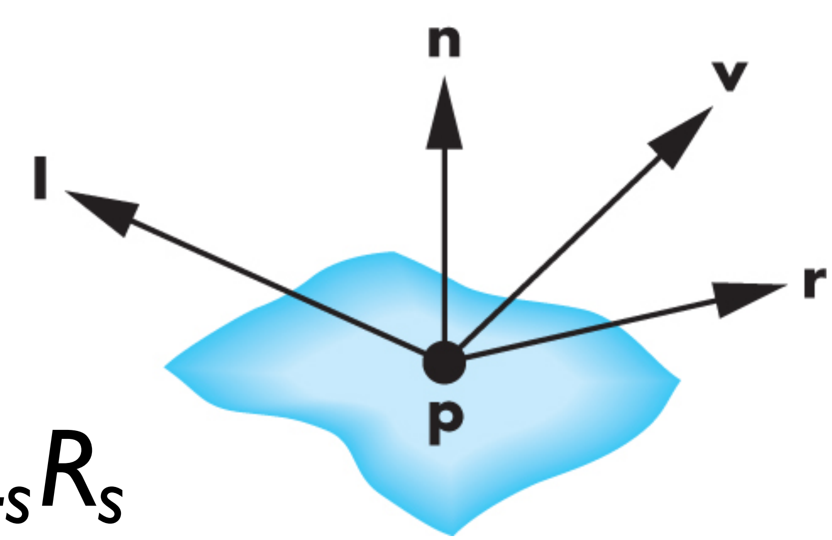
Formal Solution: Neumann Series

$$\begin{aligned} L &= (1 - \mathbf{K})^{-1} E \\ &= \sum_{i=0}^{\infty} \mathbf{K}^i E \\ &= E + \mathbf{K}E + \mathbf{K}^2 E + \mathbf{K}^3 E + \cdots \end{aligned}$$

Approximate models

Phong Reflection Model

- Blinn-Phong reflection model
- Efficient to compute and a close-enough approximation to reality
- Supports three types of material interactions:
  - Ambient
  - Diffuse
  - Specular



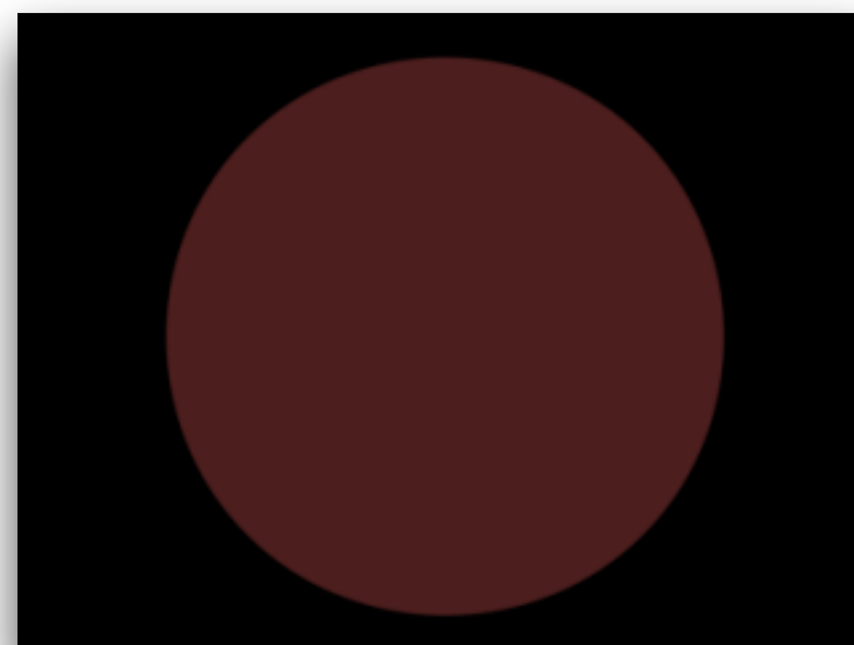
- $I = I_a + I_d + I_s = L_a R_a + L_d R_d + L_s R_s$

Ambient Reflection

- Any point whose normal faces away from the light will appear black
- In reality, indirect illumination allows us to see parts hidden from direct light
- An ambient term is added to compensate: $R_a = k_a$

$$I_a = k_a L_a,$$

$$0 \leq k_a \leq 1$$

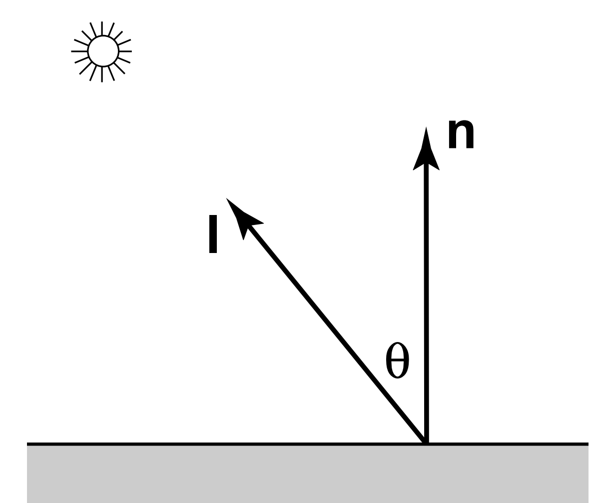


Source: http://cse.csusb.edu

Diffuse Reflection

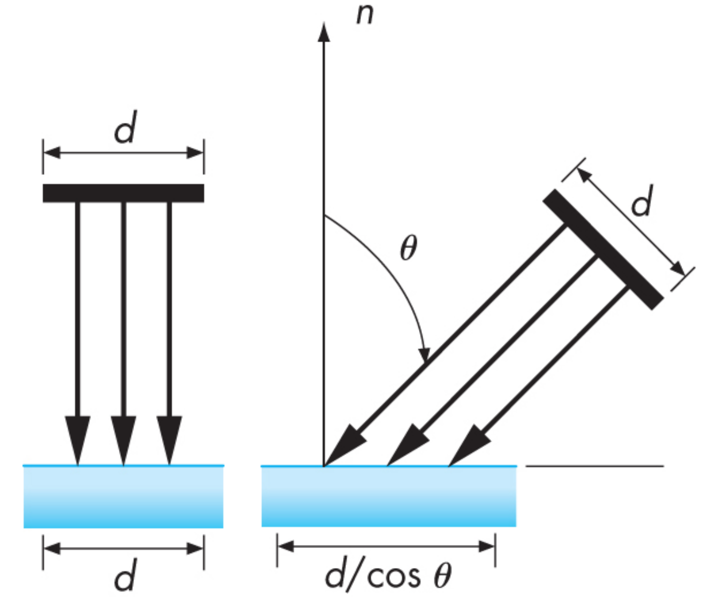
- *Lambertian* objects scatter incoming light in all directions
- No change in color with change in viewpoint
- Lambert's cosine law: color *c* of a surface is proportional to the angle between the surface normal and the direction of light source

$$c \propto n \cdot l$$

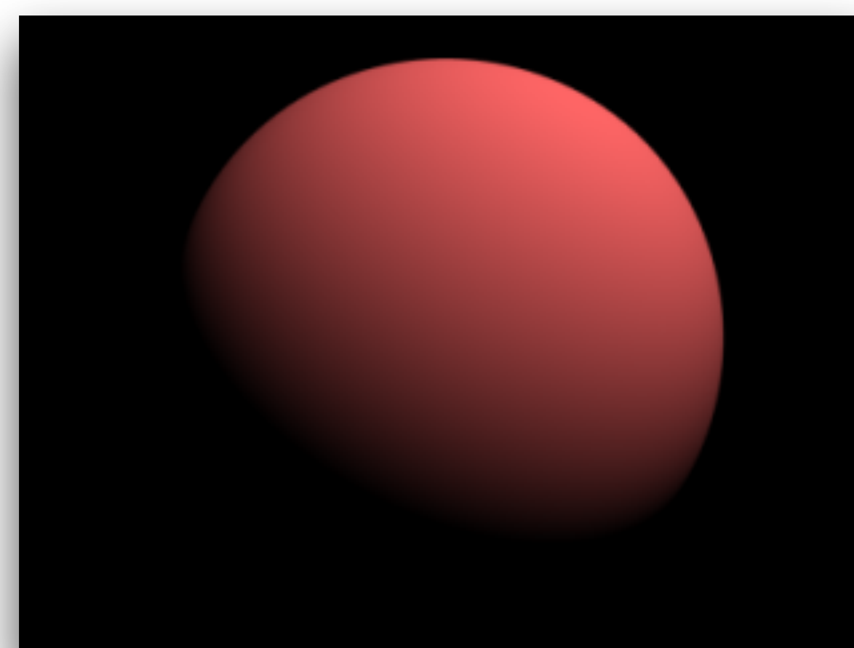


Diffuse Reflection

- Lambert's law states that:
  $R_d \propto \cos(\theta)$
  where, $\cos(\theta) = l \cdot n$



- Reflection coefficient k_d represents fraction of incoming diffuse light that is reflected



Source: http://cse.csusb.edu

- $I_d = k_d \max((l \cdot n)L_d, 0)$

Light Source Attenuation

- energy decreases as an inverse square with distance
- Therefore for diffuse lighting,

$$I_d = k_d \, f_{att} \, \max((l \cdot n) L_d, 0)$$

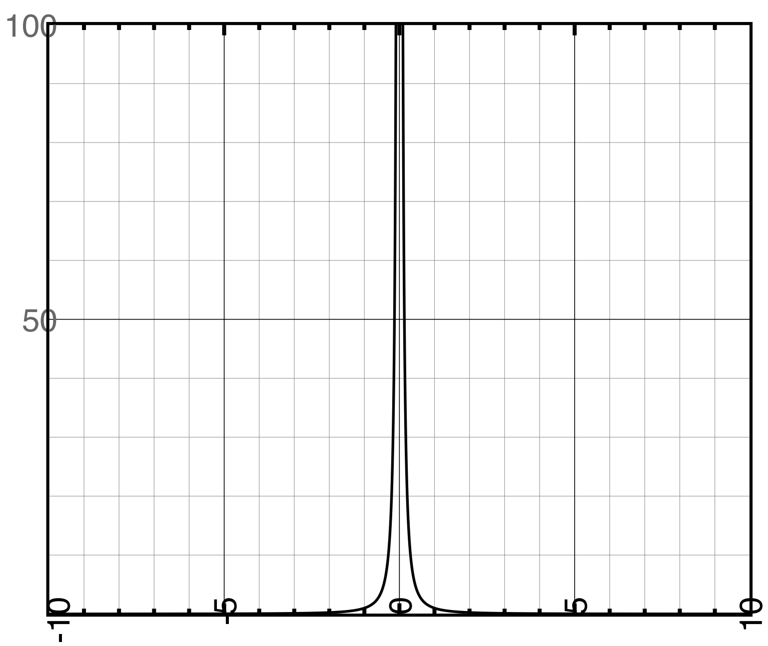
- A suitable attenuation function is:

$$f_{att} = \min\left(\frac{1}{a + bd + cd^2}, 1\right)$$

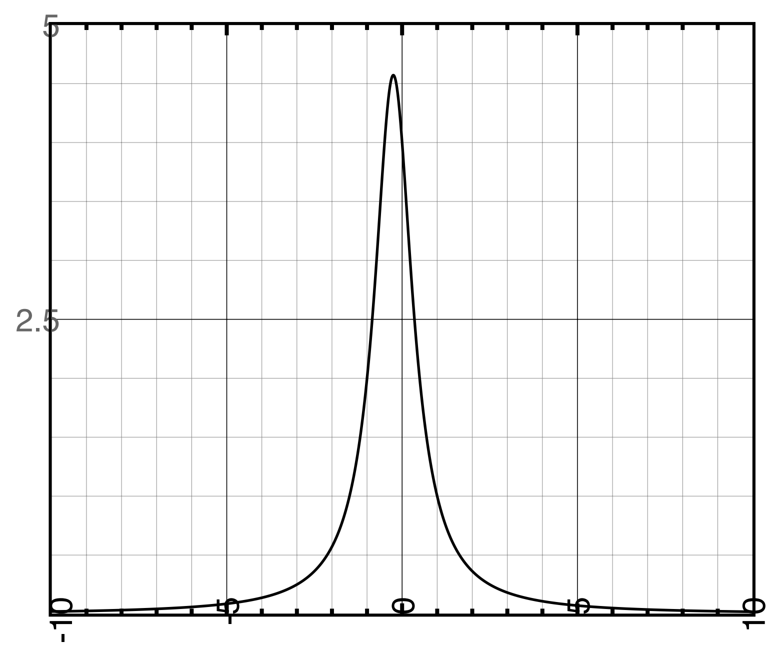
where, d is distance from light source to surface

Light Source Attenuation

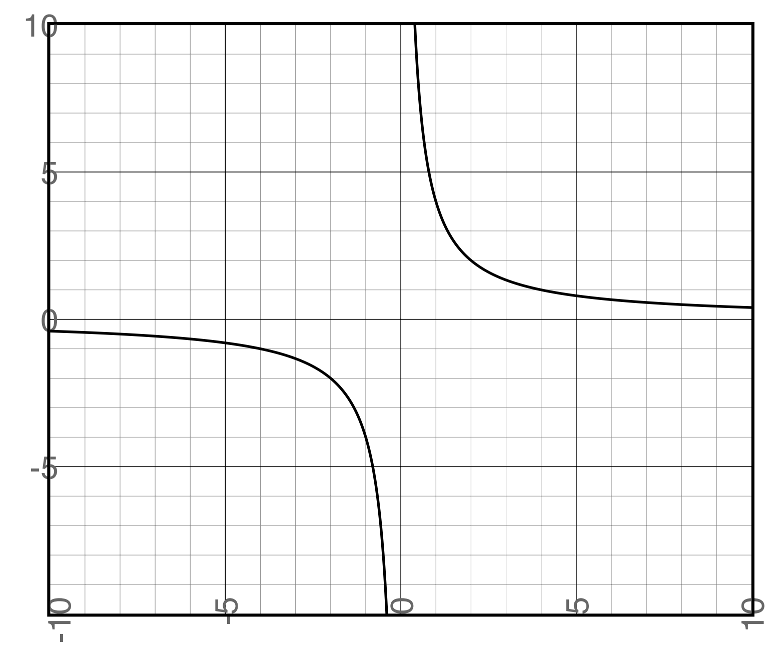
$$f_{att}(d) = \frac{1}{a + bd + cd^2}$$



$a = 0, b = 0, c = 1$



$a = 0.25, b = 0.25, c = 0.5$



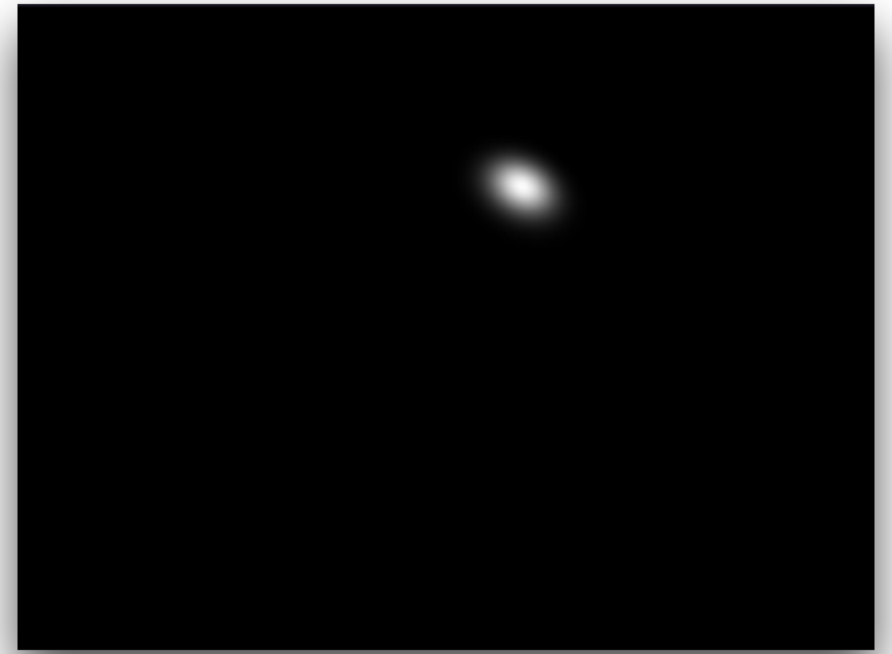
$a = 0, b = 1, c = 0$

Specular reflection

- Phong lighting model adds a specular reflection term to the lighting equation
- The amount of light that the viewer sees depends on the angle ϕ between *r* (reflected direction) and *v* (view vector)
- The Phong model uses:

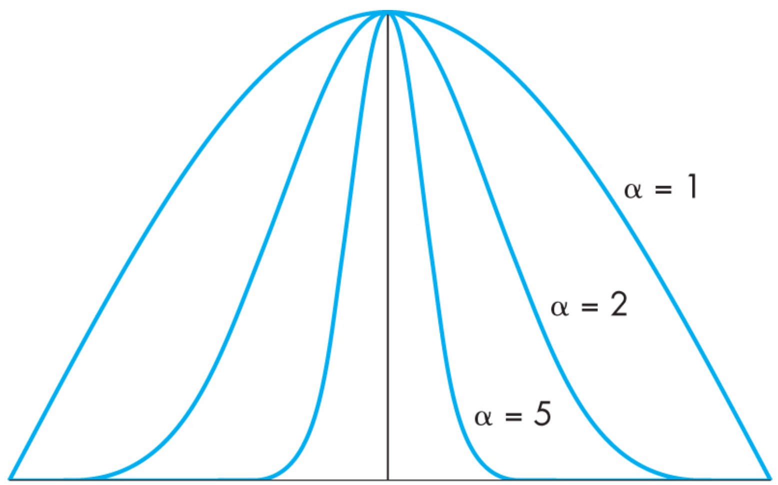
$$I_s = k_s L_s \max\left((r \cdot v)^{\alpha}, 0\right)$$

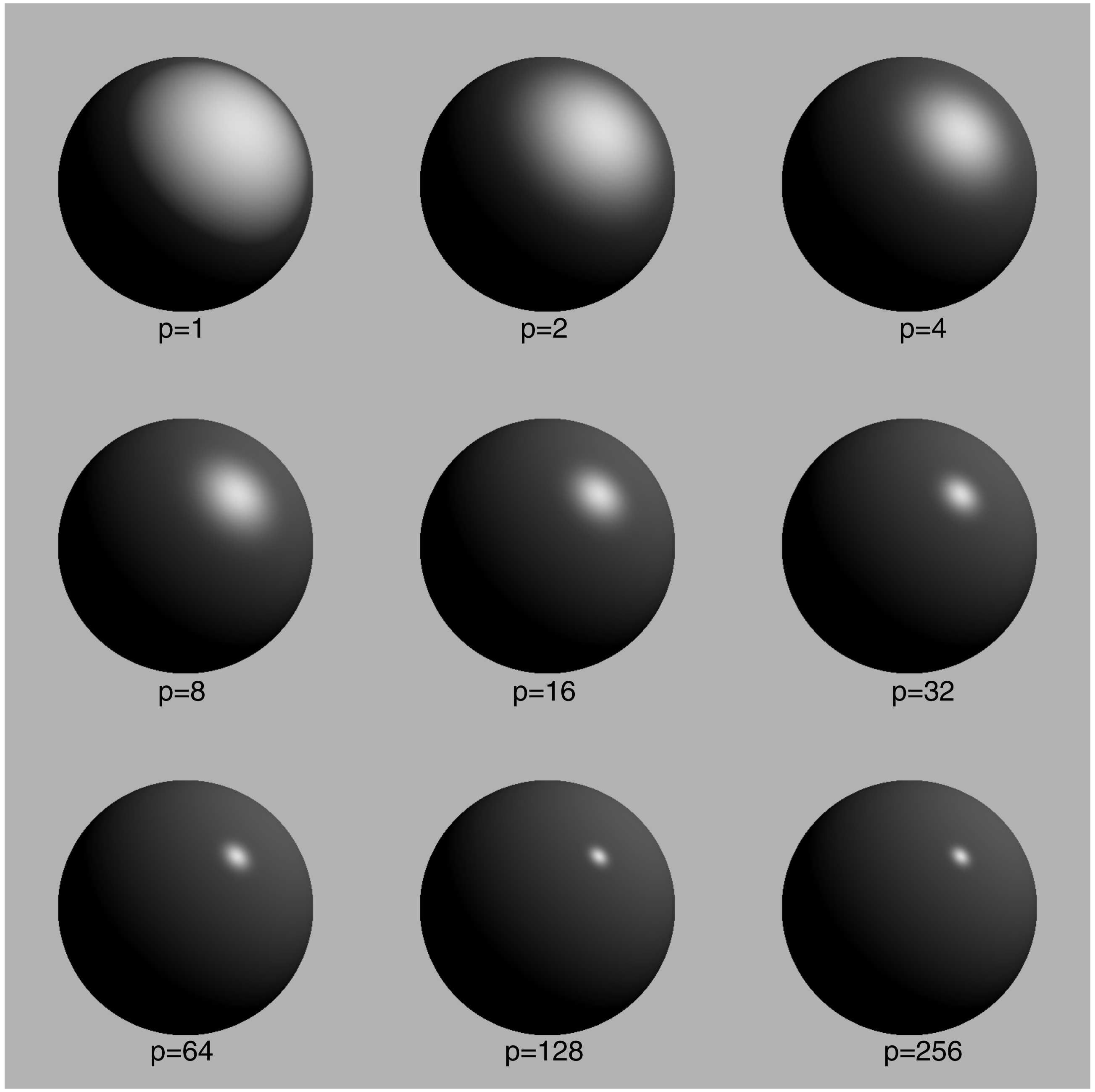
α is the *shininess* coefficient



Source: http://cse.csusb.edu

Shininess Coefficient

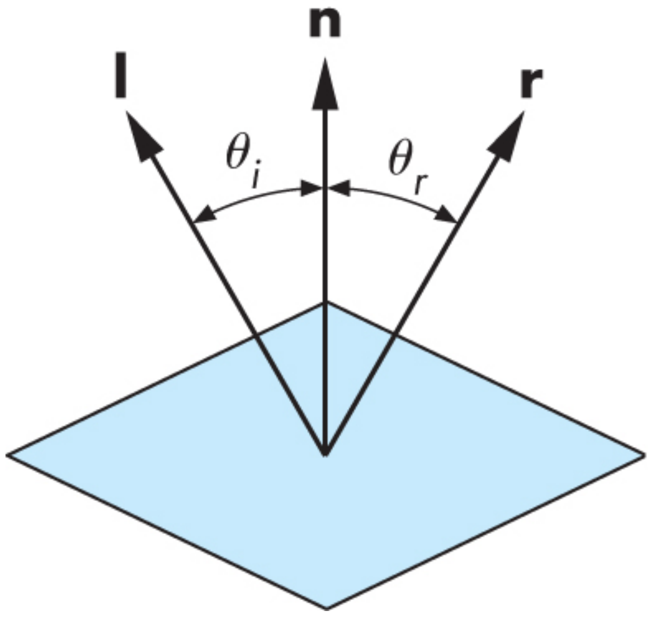




Source: Nate Robins

Reflected Vector

- Two fundamental conditions:
  1. Angle of incidence is equal to the angle of reflection
  2. At a point *p* on the surface, the incident ray *i*, the normal *n* at *p*, and the reflected ray *r* lie in the same plane



Reflected Vector

l, n, r: unit vectors

$$\theta_i = \theta_r \implies \cos(\theta_i) = \cos(\theta_r)$$

Angle condition

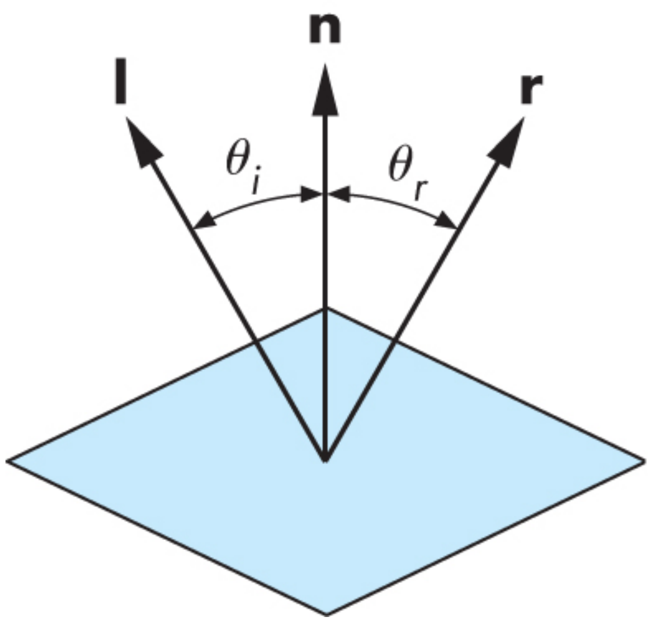
$$\cos(\theta_i) = l \cdot n = \cos(\theta_r) = n \cdot r$$

$$r = \alpha l + \beta n$$

Coplanarity

$$n \cdot r = \alpha l \cdot n + \beta = l \cdot n \qquad (1)$$

$$1 = r \cdot r = \alpha^2 + 2\alpha\beta l \cdot n + \beta^2 \qquad (2)$$



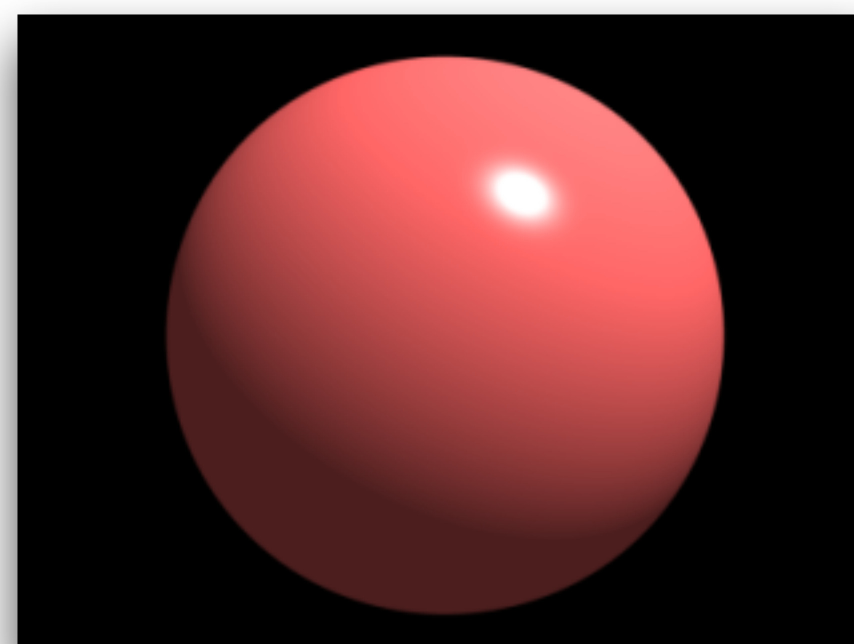
$$\boxed{r = 2(l \cdot n)n - l}$$

Phong Reflection Model

- The Phong model, including the distance term is

$$I = \frac{1}{a + bd + cd^2} \left(k_d L_d \max(l \cdot n, 0) + k_s L_s \max((r \cdot v)^\alpha, 0)\right) + k_a L_a$$

- The diffuse and specular terms are computed and added for each light source

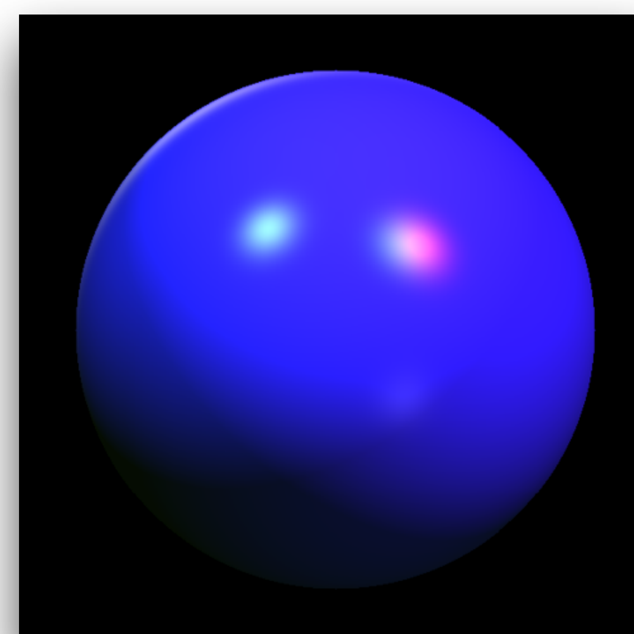


Source: http://cse.csusb.edu

Multiple Light Sources

- Multiple light sources will result in separate diffuse and specular reflections summed up as:

$$I = \sum_{l=1}^{n} \left(f_{att} \left(k_d L_d^l \max(l \cdot n, 0) + k_s L_s^l \max((r \cdot v)^\alpha, 0) \right) \right)$$
$$+ k_a L_a$$

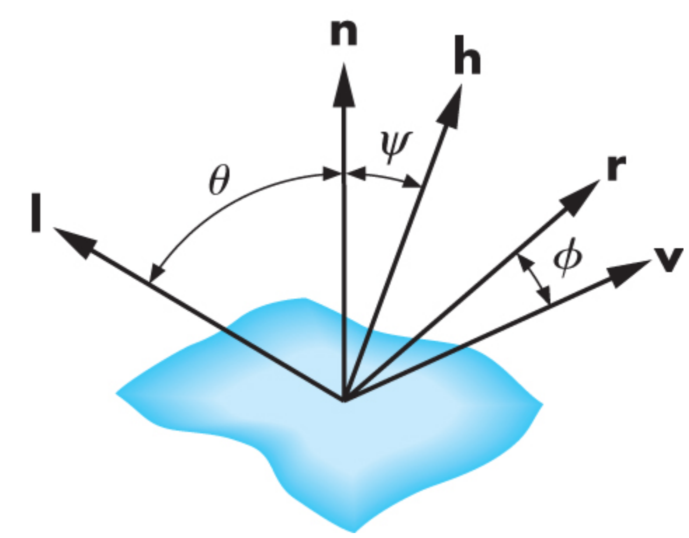


Source: http://www.kysdesign.com

Modified Phong Model

- With the Phong model, the reflected vector *r* needs to be computed at every point on surface
- To avoid this expensive computation, the *halfway* vector *h* can be computed:

$$h = \frac{l + v}{\|l + v\|}$$

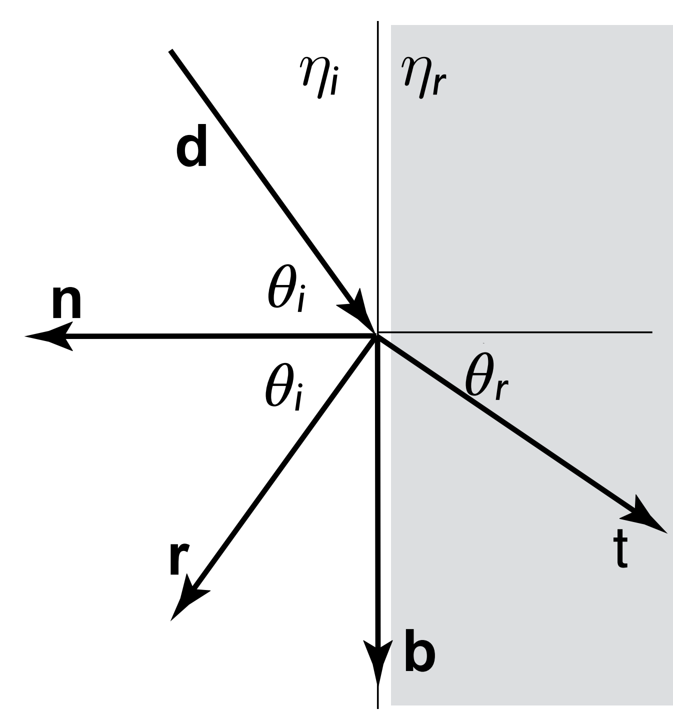


- Replace $r \cdot v$ with $n \cdot h$
- Phong highlights will be smaller than before

Blinn, James F. "Models of light reflection for computer synthesized pictures." ACM SIGGRAPH Computer Graphics. Vol. 11. No. 2. ACM, 1977.

Transparent Surfaces

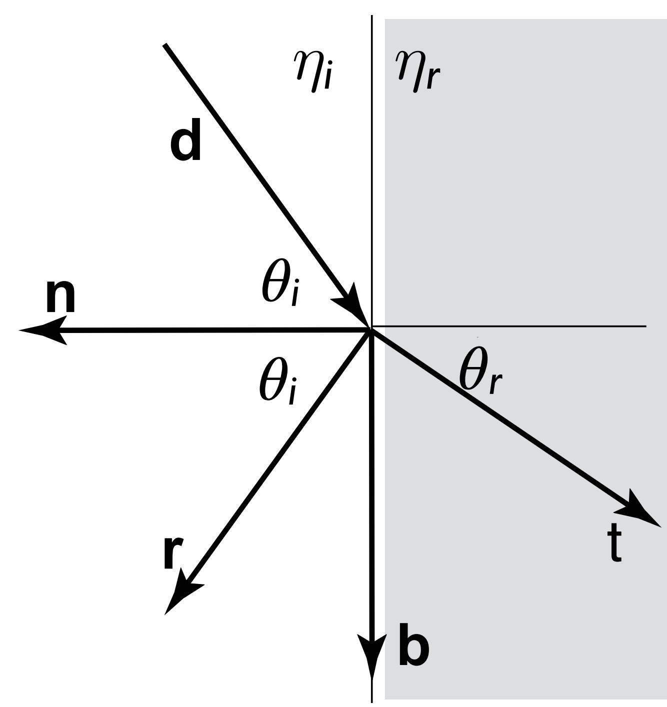
- Both diffuse and specular transmission can take place at the surfaces of a transparent object
- When a beam of light is incident on a transparent material, part of it is reflected and part is transmitted through



Snell's Law

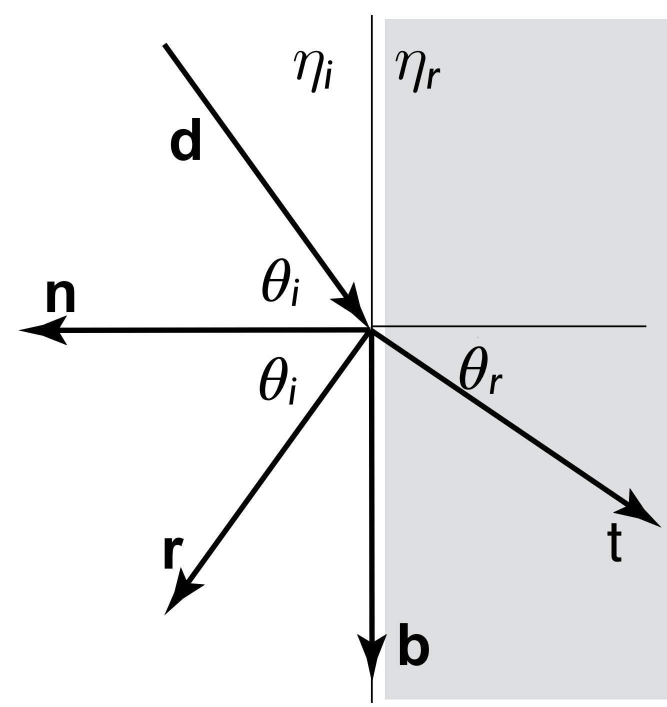
- Ratio of sines of the angles of incidence and refraction is equivalent to the reciprocal of the ratio of the indices of refraction

$$\frac{\sin(\theta_i)}{\sin(\theta_r)} = \frac{\eta_r}{\eta_i}$$



Light Refraction

- Refraction occurs whenever a light ray moves through the boundary between materials
- Law of refraction: the incident ray *d*, refracted ray *t* and the normal *n*, all lie in the same plane



Light Refraction

- Let n and b be basis vectors

Coplanarity

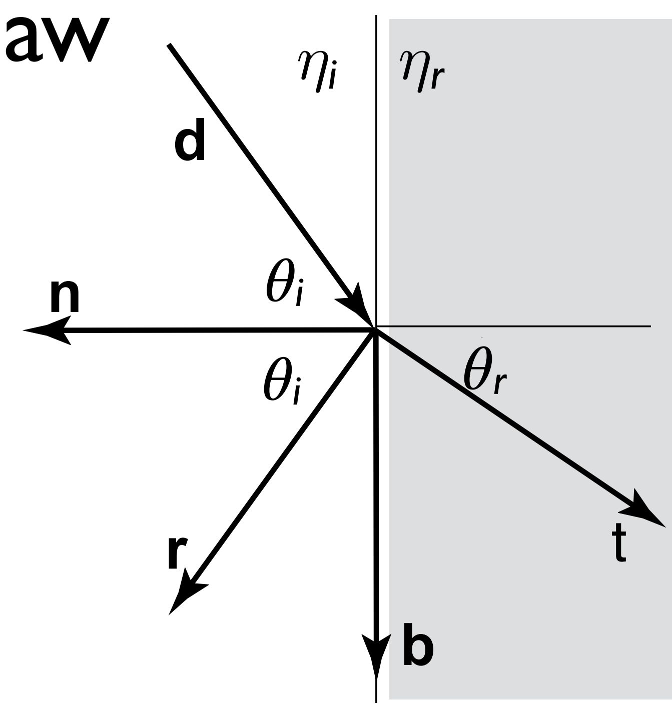
$$t = \sin\theta_r\, b - \cos\theta_r\, n$$

$$d = \sin\theta_i\, b - \cos\theta_i\, n \implies b = \frac{d + n\cos\theta_i}{\sin\theta_i}$$

Snell's law

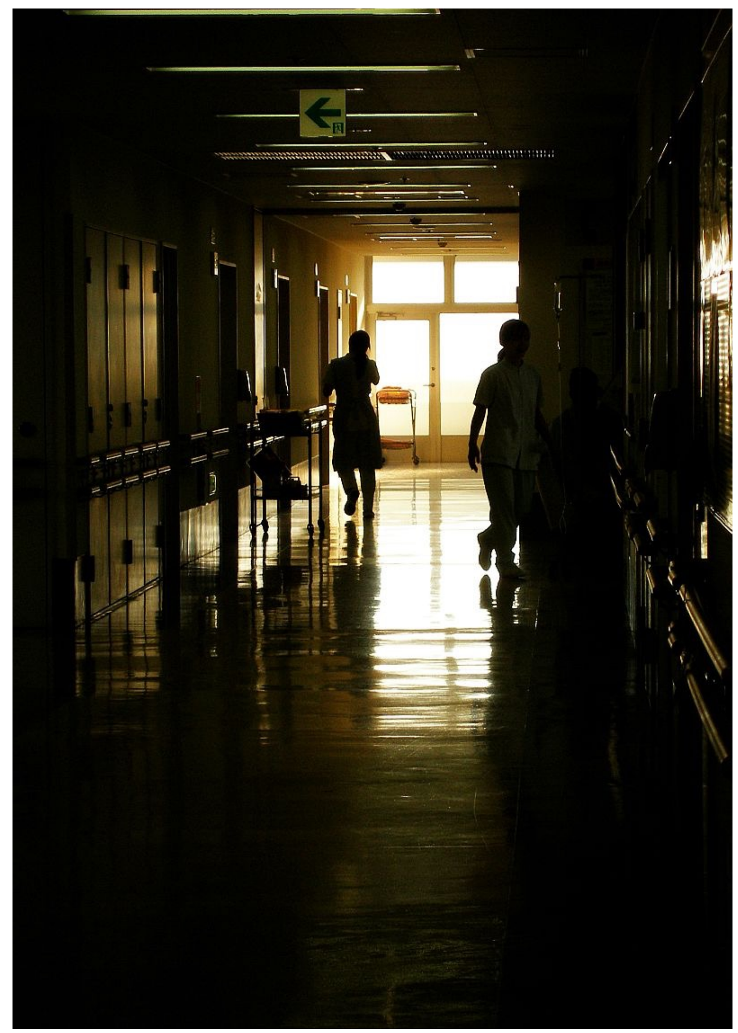
$$t = \frac{\eta_i(d + n\cos\theta_i)}{\eta_r} - n\cos\theta_r$$

$$= \frac{\eta_i(d - n(d \cdot n))}{\eta_r} - n\sqrt{1 - \frac{\eta_i^2(1 - (d \cdot n)^2)}{\eta_r^2}}$$



Basic Transparency Model

- Reflectivity of a dielectric varies with the incident angle according to *Fresnel equations*



Sources:
Slim, Chance Payne, http://sfdm.ca.scad.edu/faculty/mkesson/shared/chance_payne/index.html
https://en.wikipedia.org/wiki/Fresnel_equations

Basic Transparency Model

- Reflectivity of a dielectric varies with the incident angle according to *Fresnel equations*

- Schlick approximation may be used to approximate

  $$R(\theta) = R_0 + (1 - R_0)(1 - \cos\theta)^5$$

  Where R_0 is the reflectance at normal incidence

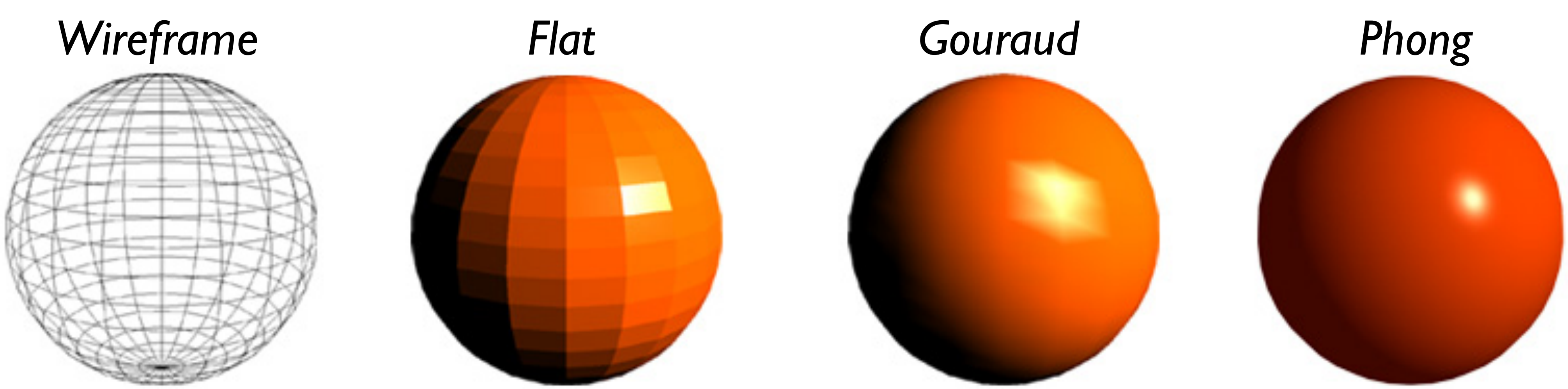
  $$R_0 = \left(\frac{\eta_r - 1}{\eta_r + 1}\right)^2$$

- Thus, $I = (1 - R(\theta))I_{trans} + R(\theta)T_{refl}$

Polygon Rendering Methods

Different shading methods may be employed during rendering of a tessellated model

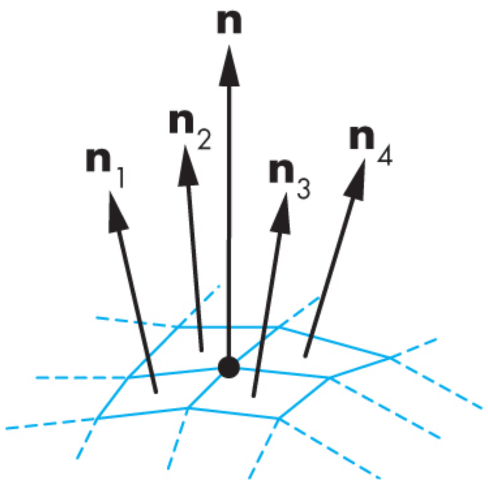
1. Flat shading
2. Gouraud shading
3. Phong shading

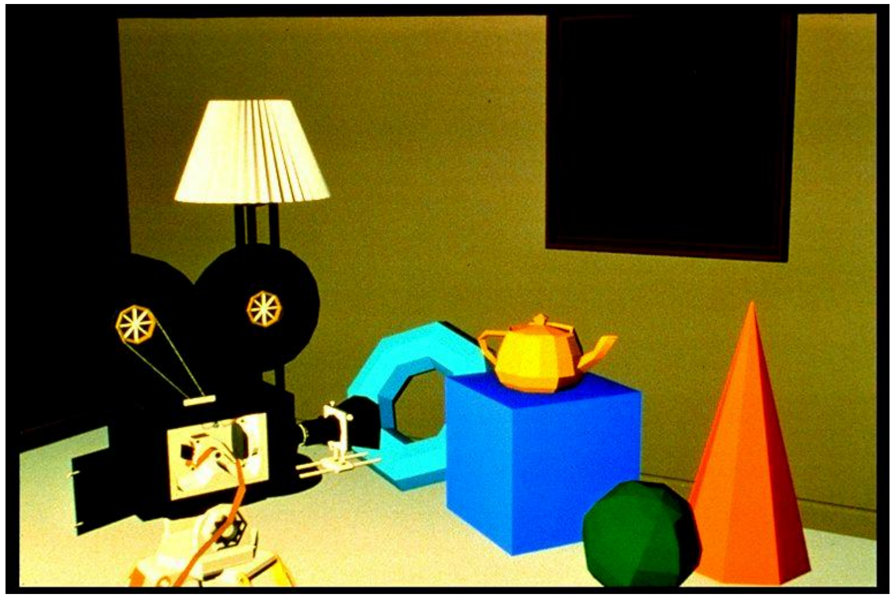


Source: Digital Media For Artists - Schattierung, www.dma.ufg.ac.at

Flat Shading

- All points within a face of the model are assigned the same colour
- All vertices of a face have same normal
- Gives a faceted appearance to the model

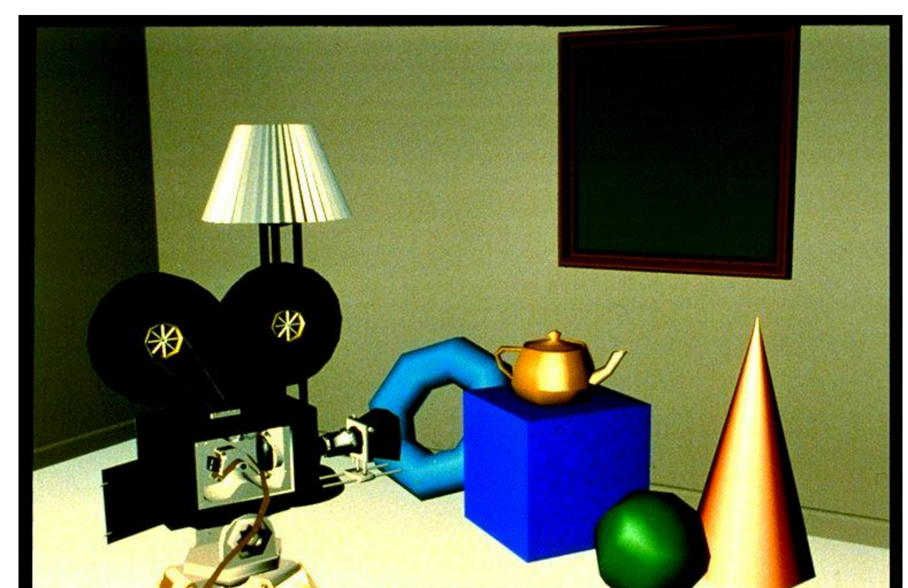




Source: SIGGRAPH

Gouraud Shading

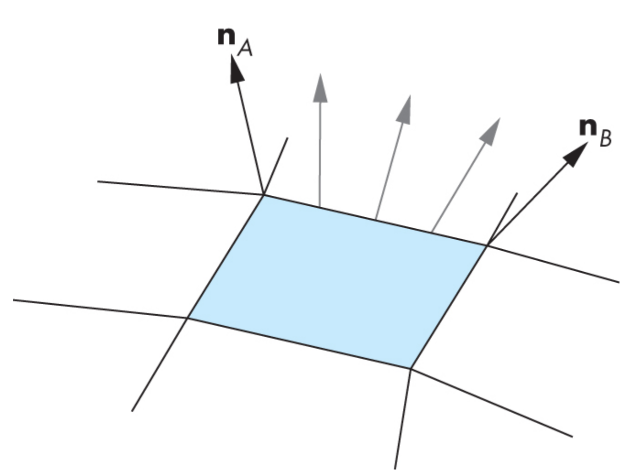
- Also known as *intensity interpolation*, linearly interpolates vertex intensities across the polygon faces
- All vertices of a face have different normals
  - One way to compute vertex normals is to average out face normals

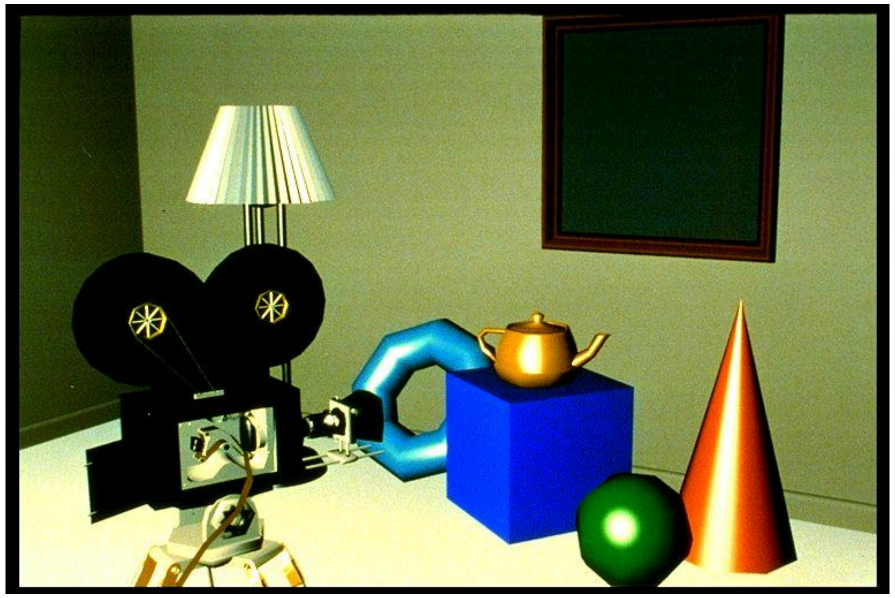


Source: SIGGRAPH

Phong Shading

- Also known as normal vector interpolation, interpolates normal vectors instead of intensity values
- All points within a face have different normals (interpolated from vertices)





Reading

- FCG: 10, 13.1
- ICG: 5 (refer to chapter notes ICG_Lighting_and_Shading_notes.pdf)

ICG: Interactive Computer Graphics, E. Angel, and D. Shreiner, 6th ed.
FCG: Fundamentals of Computer Graphics, P. Shirley, M. Ashikhmin, and S. Marschner, 3rd ed.
CG: Computer Graphics, principles and Practice, J. F. Hughes, et al.