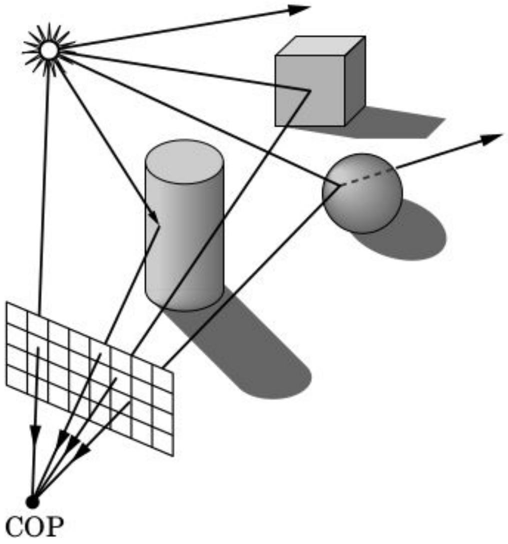
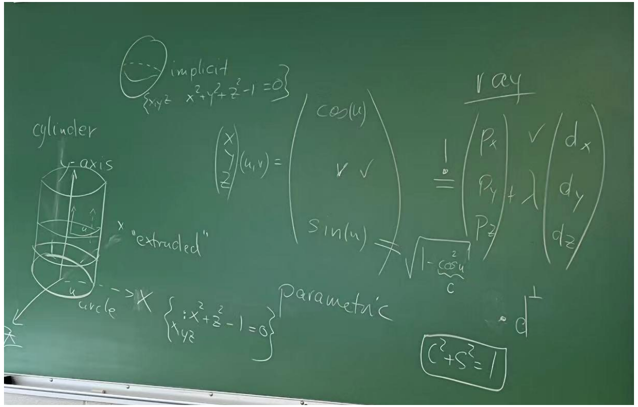
Illumination & Lighting

Ray Tracing: not supported by openGL
Path from light source to object to observer

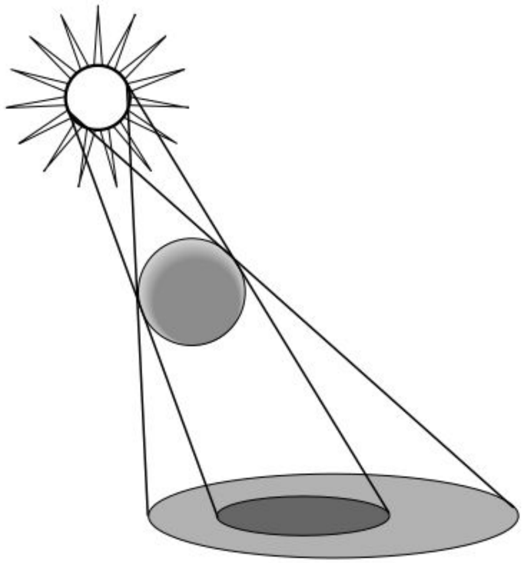


Ray-object intersection reduces to root finding



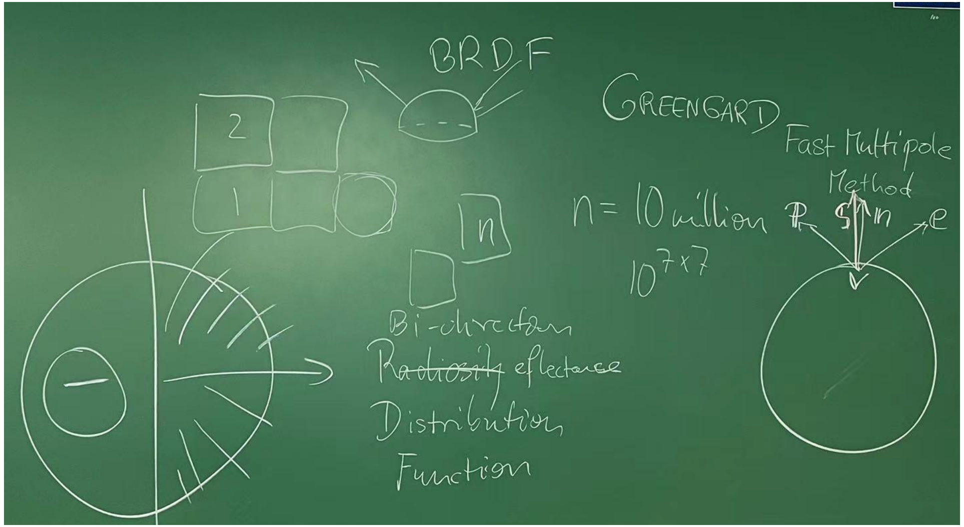
Illumination & Lighting

Interreflection: soft shadows, color bleeding, umbra, penumbra, shadows



Global Illumination

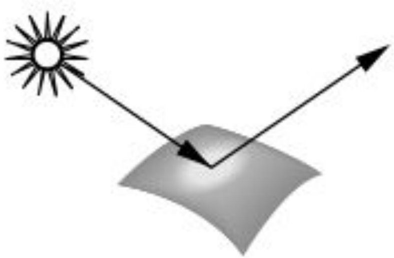
energy preservation

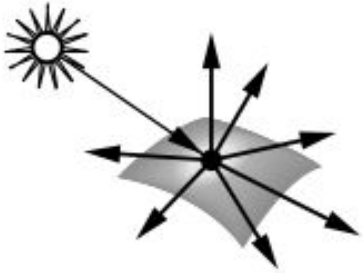


OpenGL's approximation of global illumination and ray tracing

Spectrum→ RGB (no refraction, incadescence)
Radiosity in→ sum over light sources
(no soft shadows,color bleed)
BRDF→ ambient,diffuse,specular
Radience→ intensity

OpenGL's approximation of global illumination and ray tracing





Ambient (global energy)
background glow,
equal scattering

Specular (Phong)
laser beam, mirror

Diffuse (Lambertian)
nature, equal scattering
(but still directional light
source)

OpenGL lighting model

$$\text{intensity} := \text{emission}_m + \text{ambient}_l \cdot \text{ambient}_m + \sum_{\text{lights}} \frac{1}{k_0 + k_1 d + k_2 d^2} \cdot \text{spot}_\ell \cdot \ldots$$

Distance to camera

$$\left(\text{ambient}_\ell \cdot \text{ambient}_m + \max\{\frac{\mathbf{p}-\mathbf{v}}{d} \bullet \mathbf{n}, 0\}\text{diffuse}_\ell \cdot \text{diffuse}_m \ldots\right.$$

Distance to light

$$\left. + \quad \max\{\mathbf{s} \bullet \mathbf{n}, 0\}^{\text{shininess}} \text{specular}_\ell \cdot \text{specular}_m \right)$$

where

m	=material	ℓ	=light source	l		=lighting model
$\mathbf{v}$	=vertex	$\mathbf{n}$	=normal	$\mathbf{p}$		=light position
$\mathbf{e}$	=eye position	d	$:= \|\mathbf{p}-\mathbf{v}\|$	$\mathbf{s} := \frac{\mathbf{s}'}{\|\mathbf{s}'\|}$		$\mathbf{s}' := \frac{\mathbf{p}-\mathbf{v}}{\|\mathbf{p}-\mathbf{v}\|} + \frac{\mathbf{e}-\mathbf{v}}{\|\mathbf{e}-\mathbf{v}\|}$

Distance to light

Here specular_ℓ, specular_m etc. are scalars.

Formula applies separately to RGB

Lights are objects affected by model-view transformations.

OpenGL Lighting

http://www.opengl-tutorial.org/beginners-tutorials/tutorial-8-basic-shading/

Given a unit sphere, where is the highlight (= point of highest intensity)?
Compute this for some choice of e and p. (Reduce to plane through 0, e, p since n lies in that plane.)

screenshot?

Translucency

If vertex v_j has
opaqueness value α_j and intensity i_j
is drawn before v_{j+1} then the intensity is

$$\alpha_0 \mathbf{i}_0 + (1-\alpha_0)(\alpha_1 \mathbf{i}_1 + (1-\alpha_1)(\ldots))$$

Given a unit sphere, where is the highlight (= point of highest intensity)?
Compute this for some choice of e and p. (Reduce to plane through 0, e, p since n lies in that plane.)

Computing Normals

Surface in *implicit* representation $p(\mathbf{x}) = p(x, y, z) = 0$.

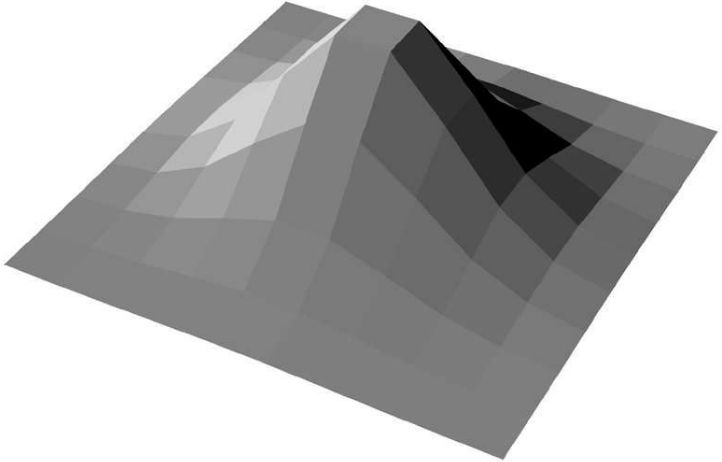
The normal direction is the (normalized) gradient $\nabla p = \begin{bmatrix} \frac{\partial}{\partial x} p \\ \frac{\partial}{\partial y} p \\ \frac{\partial}{\partial z} p \end{bmatrix}$

Surface in *parametric* representation $\mathbf{x}(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix}$.

The normal direction is $\frac{\partial \mathbf{x}}{\partial u} \times \frac{\partial \mathbf{x}}{\partial v}$. To obtain the normal, normalize the normal direction to length 1.

Blackboard Examples: $p(\mathbf{x}) = x^2 + y^2 + z^2 - 1$, $\mathbf{x}(u, v) = \begin{bmatrix} \cos(u)\cos(v) \\ \cos(u)\sin(v) \\ \sin(u) \end{bmatrix}$

Polygon Shading



Flat
Gouraud : averaged vertex color using barycentric weights.
Phong: averaged vertex normal (and other lighting factors)

Dithering, fog, blur