Lighting & Shading

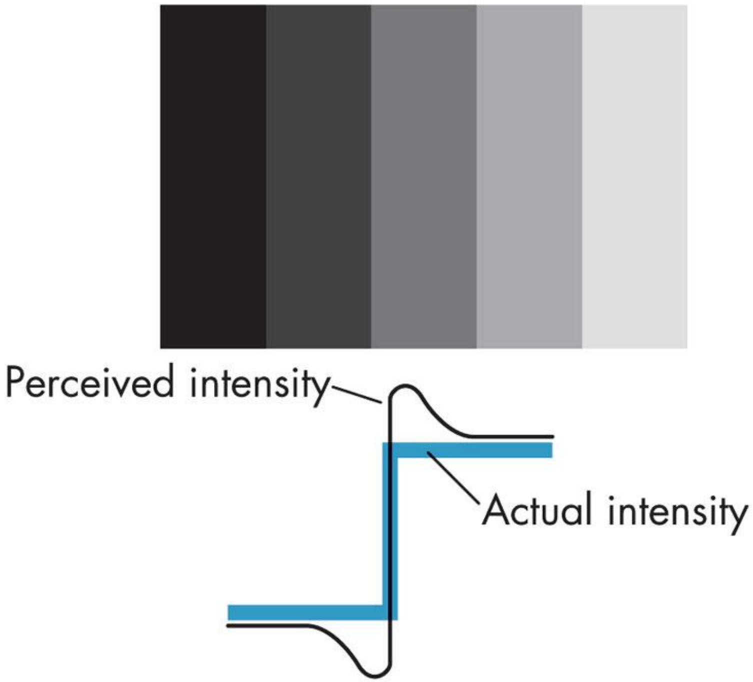
Fall 2025
11/6/2025
Kyoung Shin Park
Computer Engineering
Dankook University

Shading

- Shading is the process of determining the colors of the pixels in a primitive.
  - **Flat shading** fills the polygon face with the first vertex color
  - **Gouraud shading (smooth shading)** interpolates the polygon face with the colors at each vertex
  - **Phong shading** interpolates the polygon normal

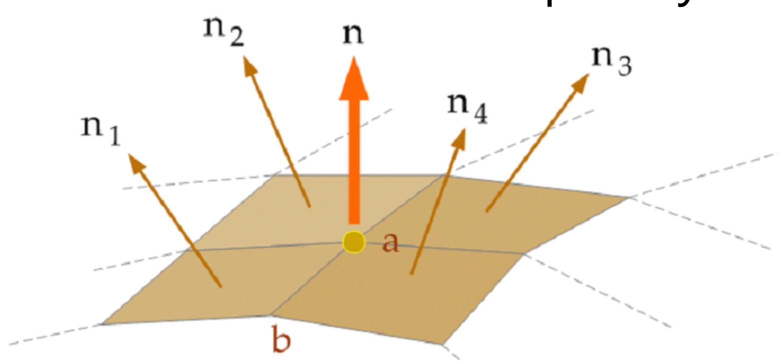
Flat Shading

- Flat shading
  - The color of the entire given polygon painted with the same color.
  - Also called constant shading, facet shading
  - Mach band effect in flat shading - an optical phenomenon from edge enhancement due to lateral inhibition of the retina



Gouraud Shading

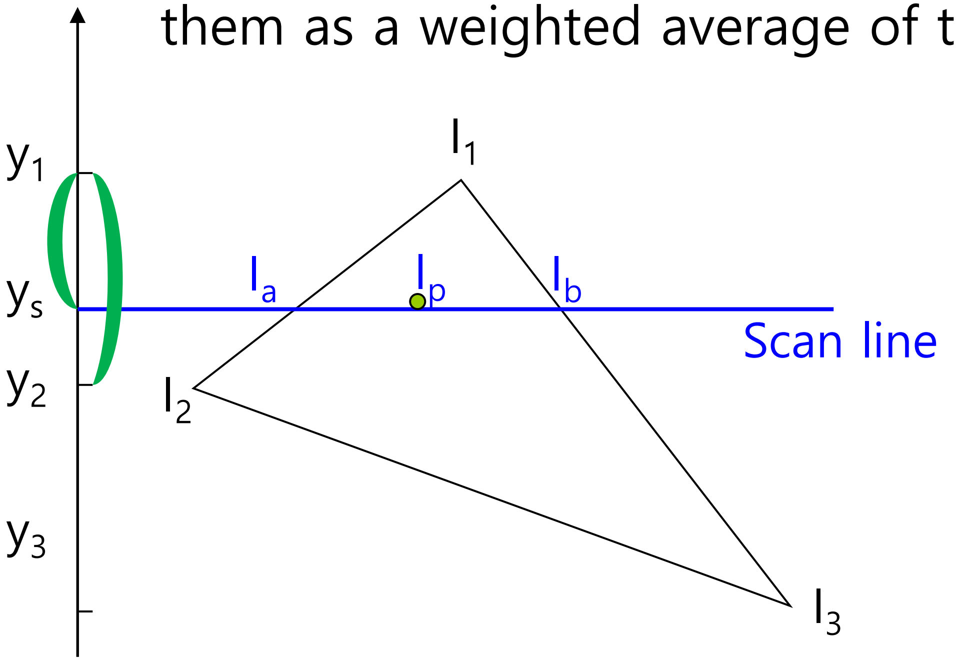
- Fills the inside of a polygon with different colors.
- Interpolating vertex color
  - Requires the normal vector of the vertex, which is calculated by averaging the normal vectors of the adjacent surface.
  - Linear interpolation of inner surface color from vertex color.
- Does not take specular light into account
  - This is because the actual vertex normal vector and the approximate calculated normal vector do not completely coincide.



$$n = \frac{n_1 + n_2 + n_3 + n_4}{|\, n_1 + n_2 + n_3 + n_4 \,|}$$

Gouraud Shading

- Gouraud shading interpolates color
  - Using the brightness intensity of the starting point and the ending point, and it calculates the brightness intensity between them as a weighted average of the two.



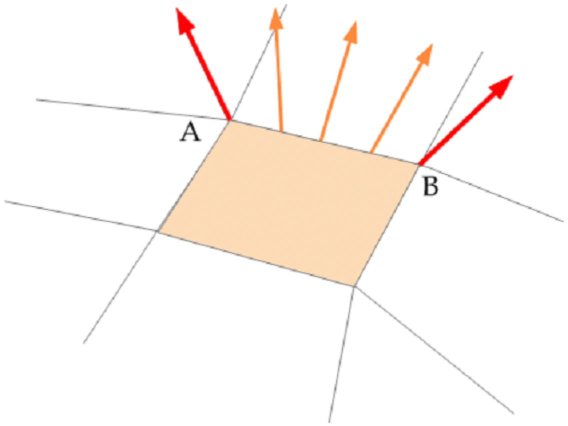
$$I_a = I_1 - (I_1 - I_2)\frac{y_1 - y_s}{y_1 - y_2}$$

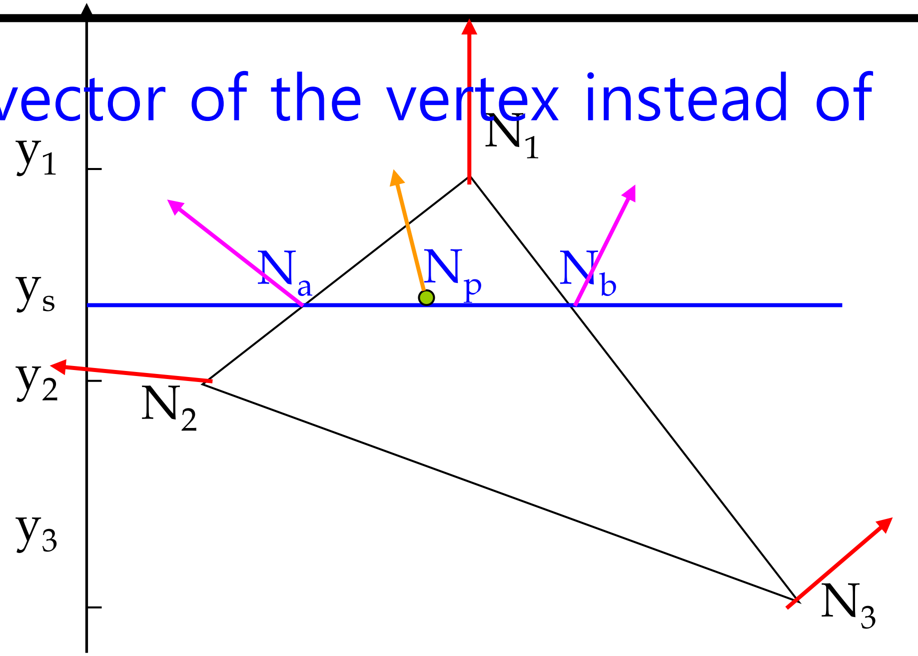
$$I_b = I_1 - (I_1 - I_3)\frac{y_1 - y_s}{y_1 - y_3}$$

$$I_p = I_b - (I_b - I_a)\frac{x_b - x_p}{x_b - x_a}$$

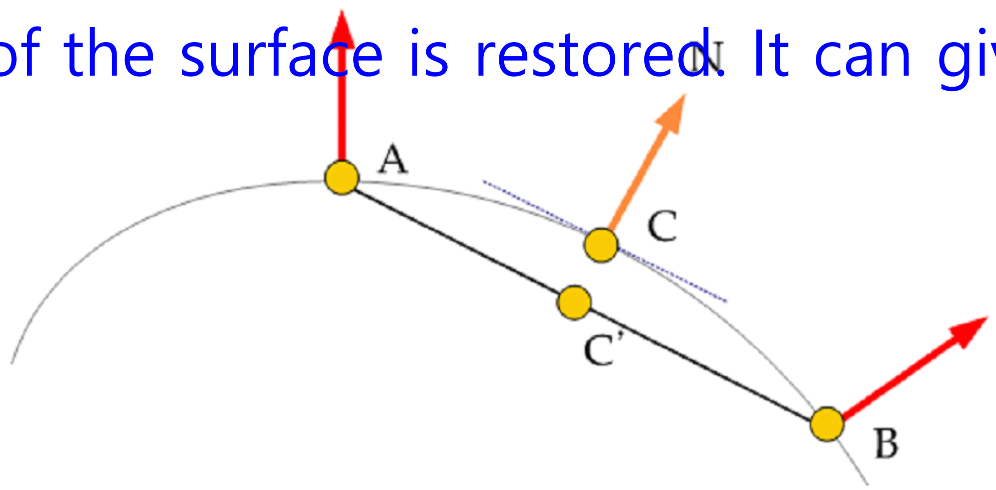
Phong Shading

- Interpolate the normal vector of the vertex instead of the color of the vertex

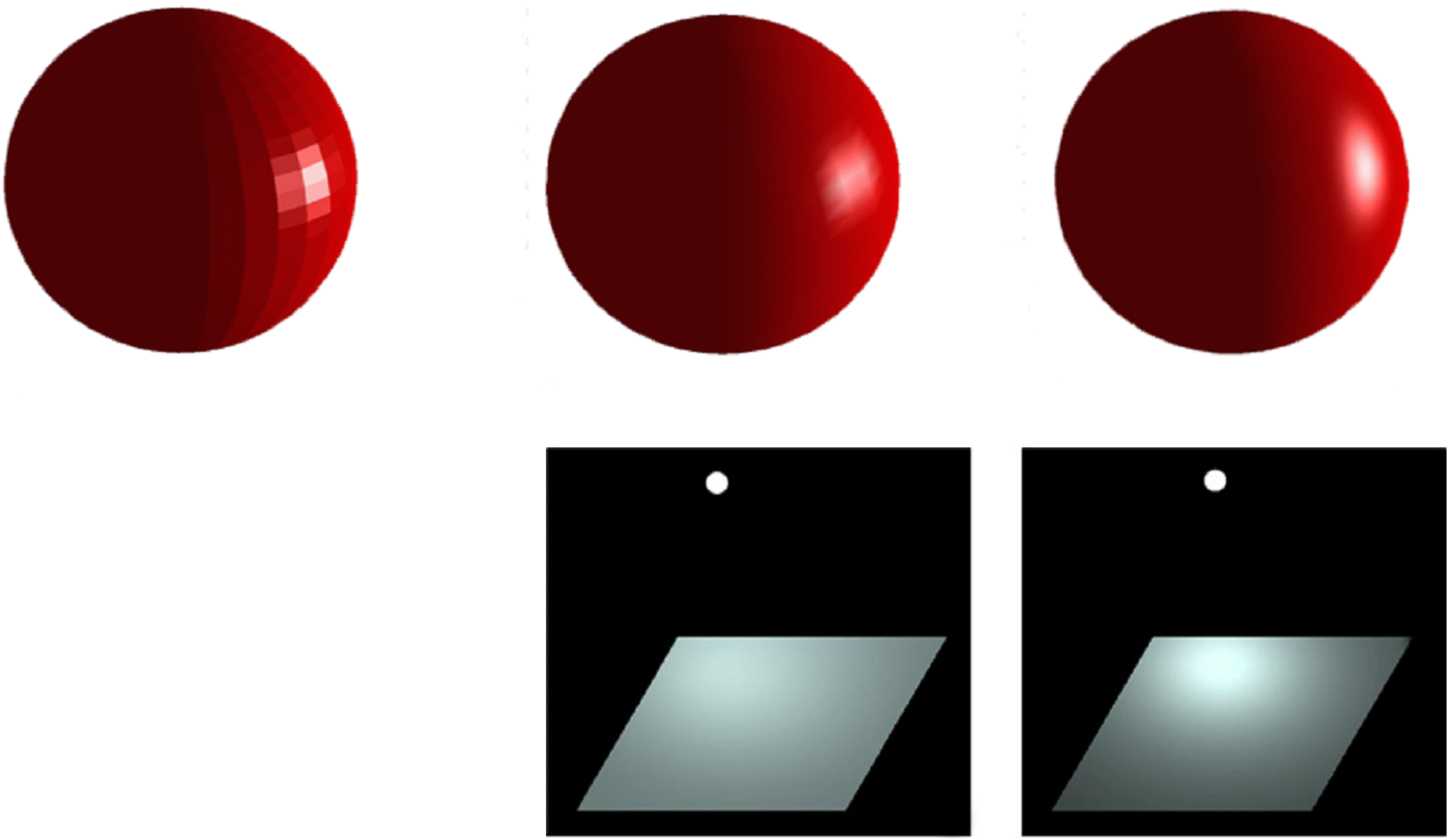




- The slope of the surface is restored. It can give specular light.

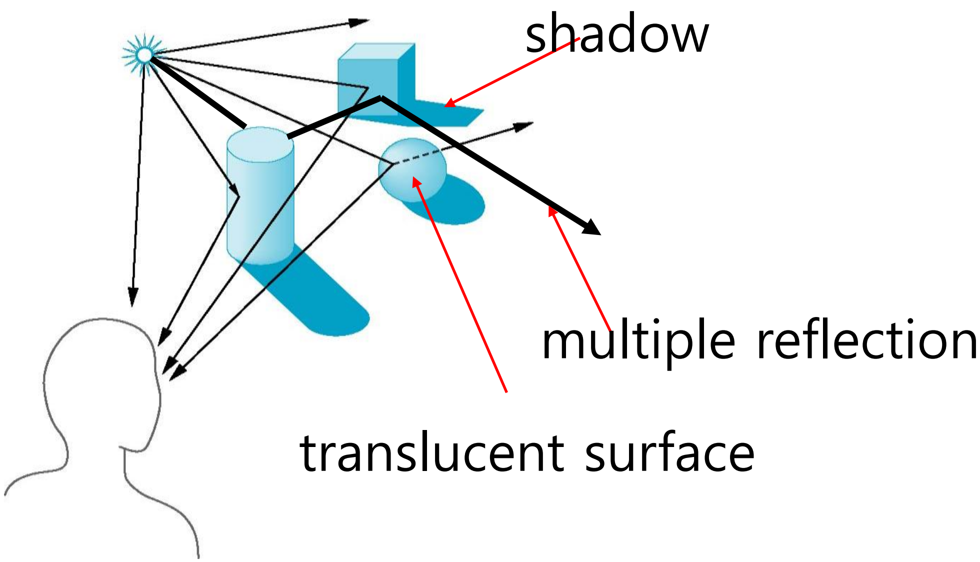


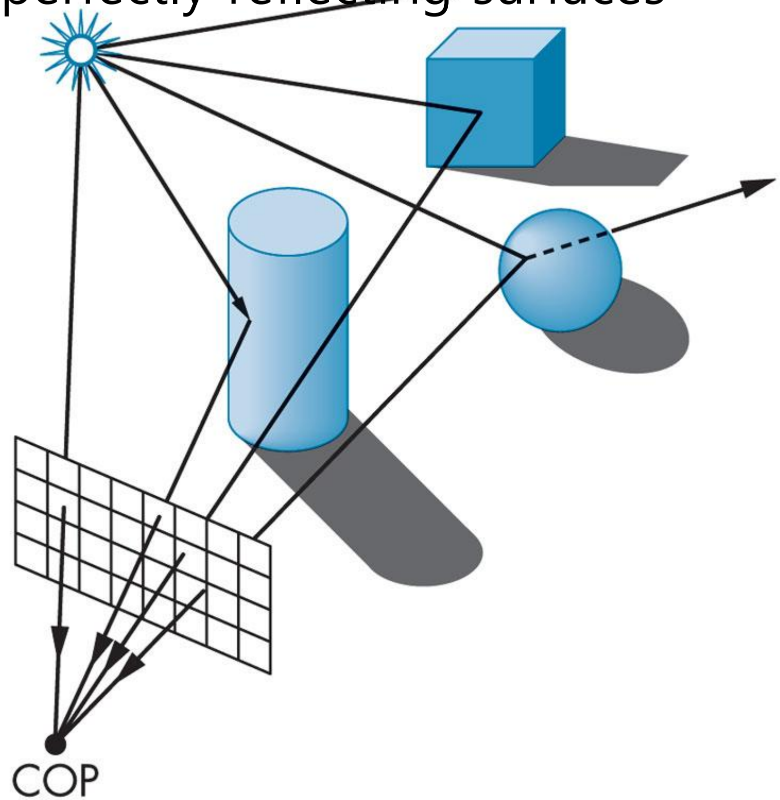
Flat, Gouraud, and Phong Shading



Lighting Model

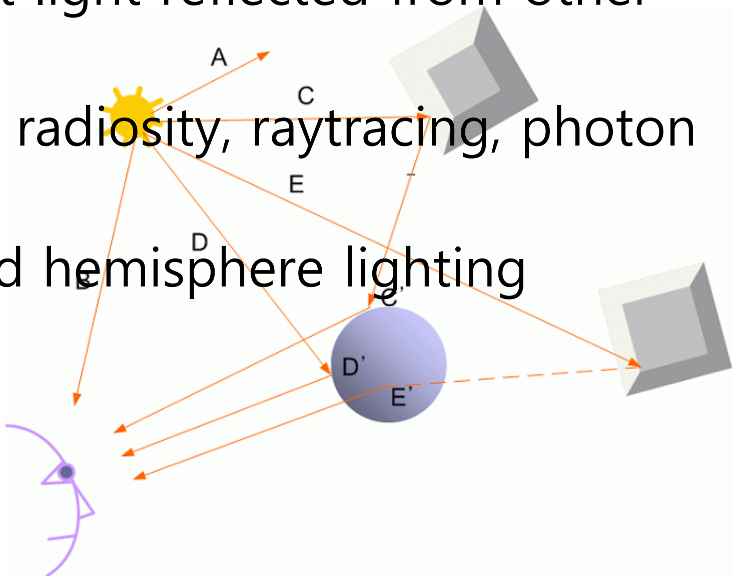
- The infinite scattering and absorption of light can be described by the rendering equation
  - Cannot be solved in general
  - Ray tracing is a special case for perfectly reflecting surfaces





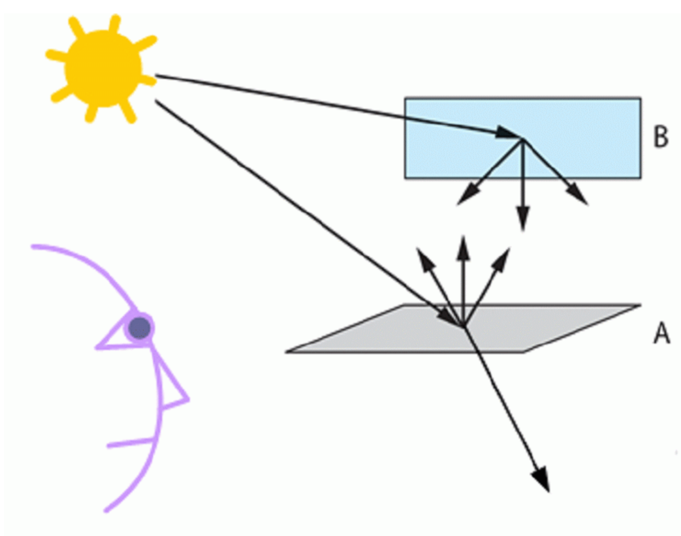
Lighting Model

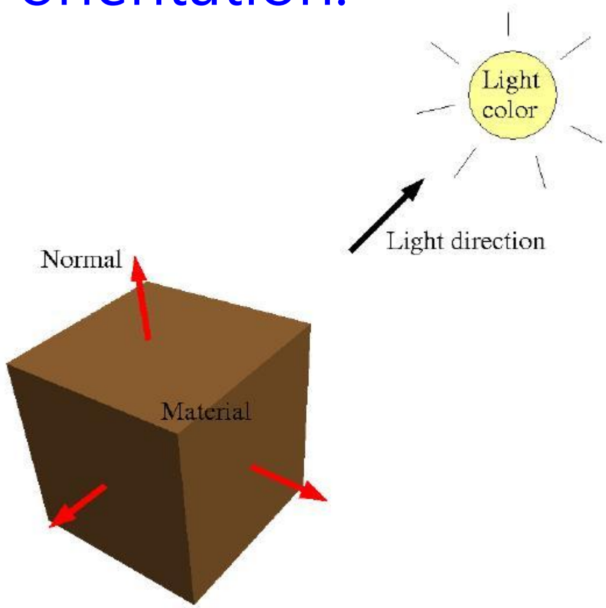
- Direct Illumination Model
  - Light model that deals with the light that points on the surface of an object receive directly from all light sources in the scene
  - It is mainly used in traditional real-time rendering because of low computation
  - Phong reflection model
- Global Illumination Model
  - Light model that considers incident light reflected from other objects
  - Global illumination model includes radiosity, raytracing, photon mapping, etc
  - Real-time GPU programming-based hemisphere lighting



Lighting

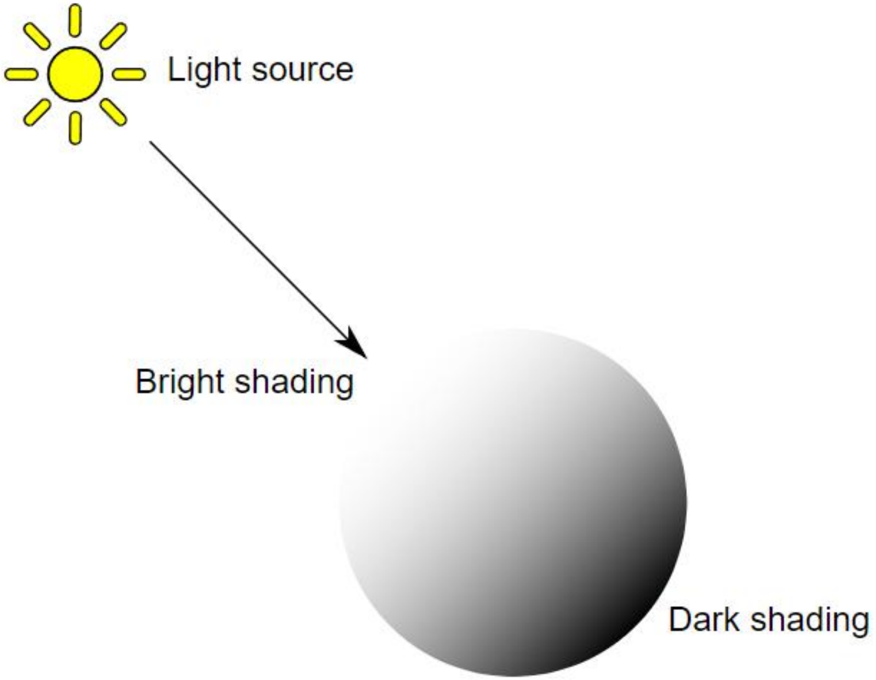
- Light starts at the **lighting source**
- The light strike on the surface, it
  - **Absorption**
  - **Reflection**
  - **Transmission or Refraction**
- Shading is determined by light source color, material properties, viewer location, surface orientation.





Lighting

- Light Components
  - Lights (L)
  - Materials
  - Viewer (V)
  - Vertex Normals (N)

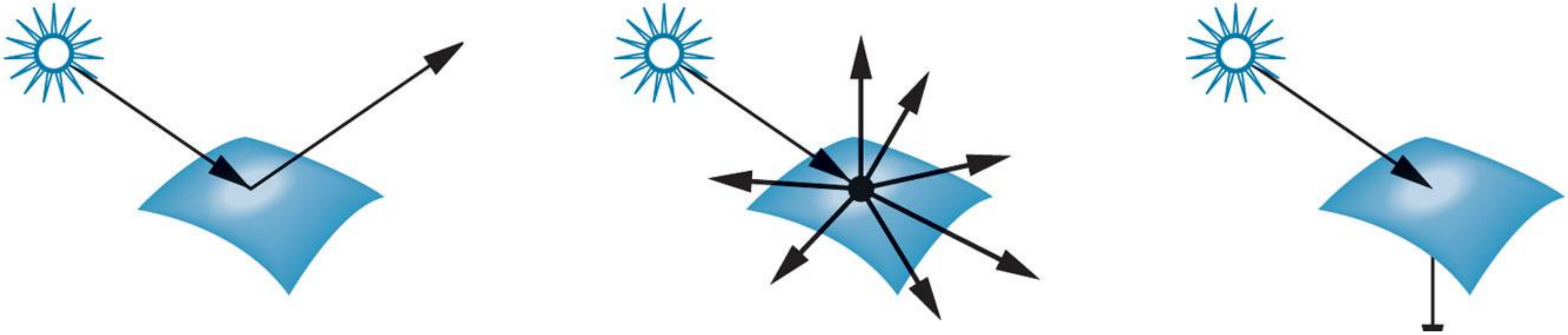


Light and Material Interaction

- **When using lighting**, we no longer specify vertex colors ourselves; rather, **we specify materials and lights**, and then, apply a lighting equation, which computes the vertex colors for us based on light/material interaction.
- **Materials** can be thought of as the properties that determine how light interacts with an object.
- We model **lights** by an additive mixture of red, green, and blue light (RGB); we can simulate many light colors.

Light-Material Interactions

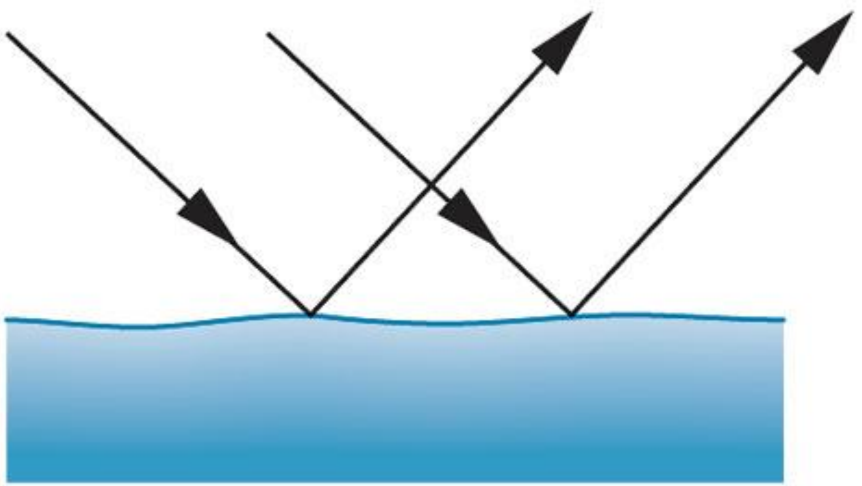
- Specular surface
  - The smoother a surface, the more reflected light is concentrated in the direction a perfect mirror would **reflected** the light.
- Diffuse surface
  - A very rough surface **scatters light in all directions**.
- Translucent surface
  - In a translucent surface, some light penetrates the surface and exist to other locations on the object **(refraction)**



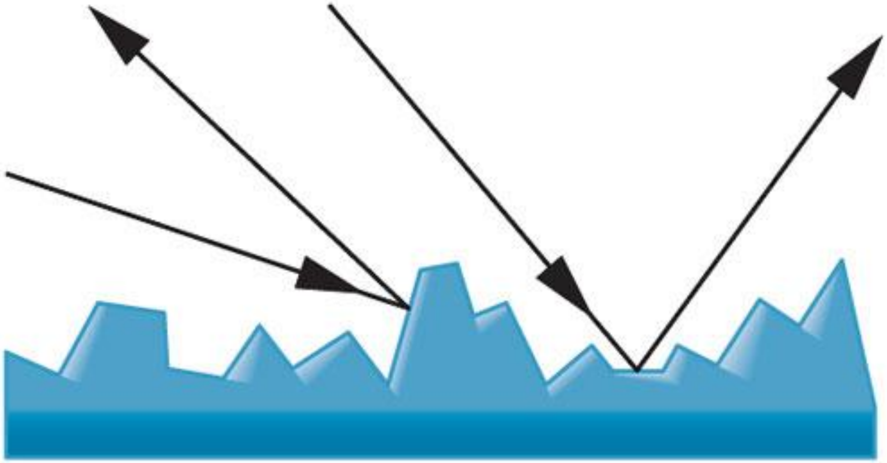
smooth surface rough surface translucent surface

Light-Material Interactions

- Perfectly Specular surface
  = very smooth surface



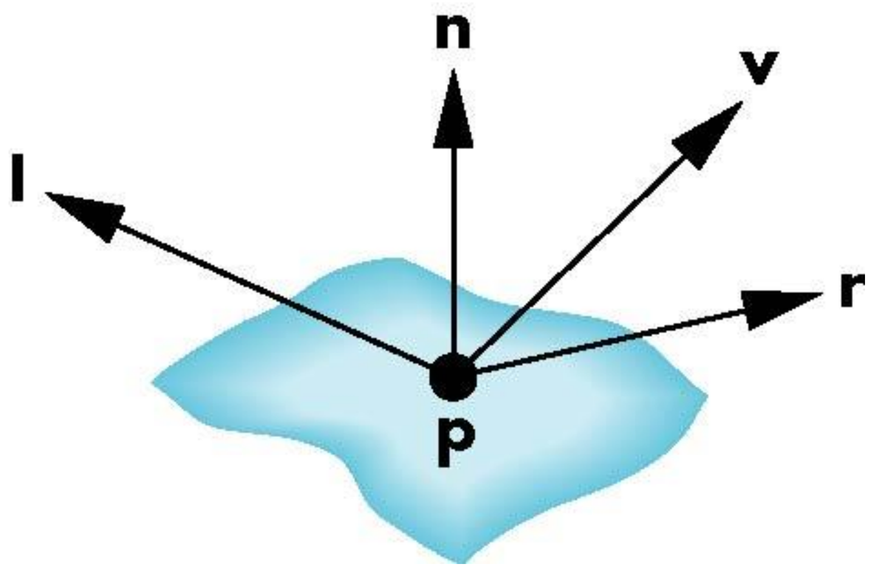
- Perfectly Diffuse surface
  = very rough surface



Angle of Incidence

- Angle of Incidence
  - The angle between the light source vector and the normal vector

$$N \cdot L = \|N\|\|L\|\cos\theta = (1)(1)\cos\theta = \cos\theta$$



Angle of Reflection

- Angle of Reflection
  - The angle of incident and the angle of reflection are the same.

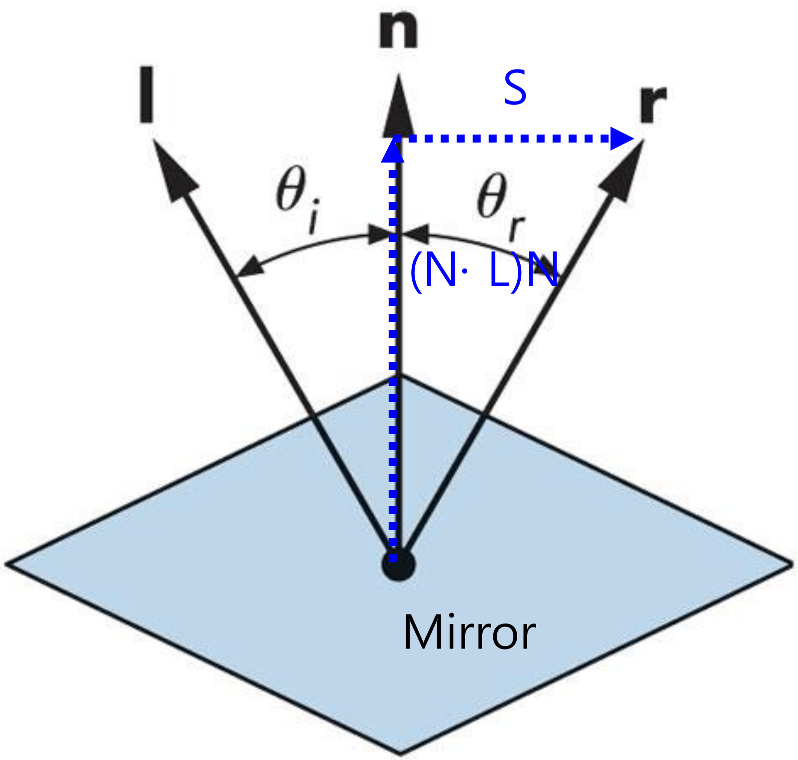
$$\theta_i = \theta_r$$

$$R = (N \cdot L)N + S$$

$$L = (N \cdot L)N - S$$

$$\Rightarrow S = (N \cdot L)N - L$$

$$\Rightarrow R = 2(L \cdot N)N - L$$



Projecting L onto $\mathrm{N} = (\mathrm{L} \cdot \mathrm{N})\mathrm{N}$ when $\|\mathrm{N}\| = 1$

Indices of Refraction

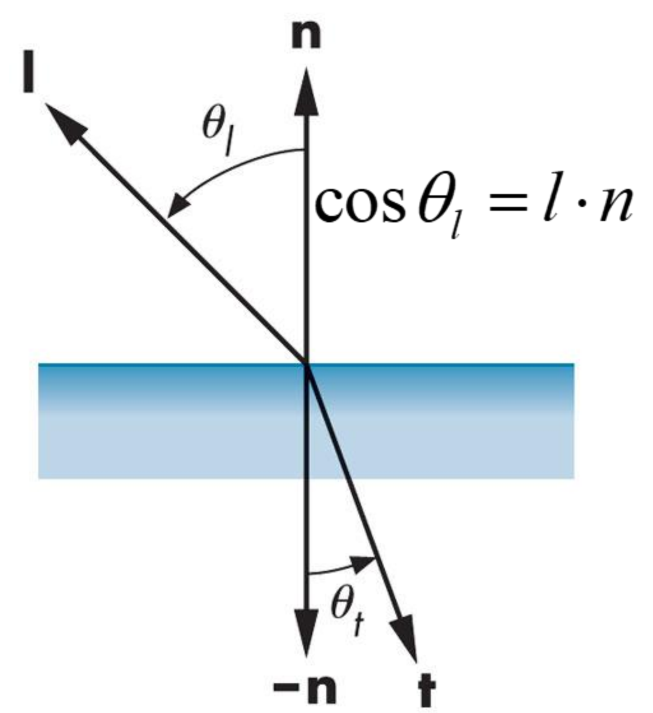
- Refraction
  - η_l, η_t = the indices of refraction of two materials
- Snell's law:

$$\frac{\sin\theta_l}{\sin\theta_t} = \frac{\eta_t}{\eta_l} = \eta$$

$$\eta = \eta_t / \eta_l$$

$$\cos\theta_t = \sqrt{1 - \frac{1}{\eta^2}\left(1 - \cos^2\theta_l\right)}$$

$$T = -\frac{1}{\eta}L - \left(\cos\theta_t - \frac{1}{\eta}\cos\theta_l\right)N$$



Perfect light transmission

Lighting Component

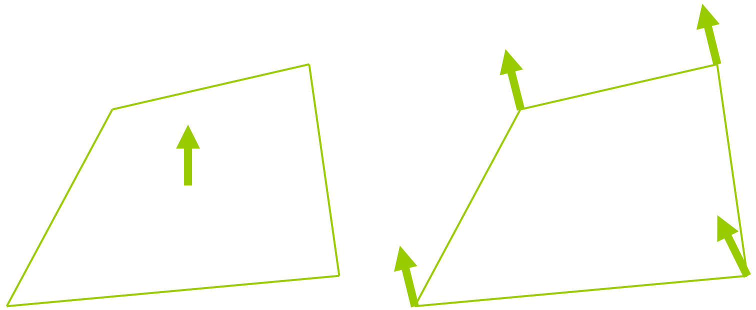
- Light
  - Ambient light – Ambient light comes from no particular direction. It reflects equally in all directions.
  - Diffuse light – The basic lighting effects is diffuse lighting. The intensity of diffuse lighting depends on the orientation of the surface relative to the light source. Diffuse light is reflected equally in all directions.
  - Specular light – Specular light is the light that is directly reflected off the surface to the camera. Its intensity depends on the orientation of the surface relative to camera, as well as to the light source.

Materials

- Material properties define how a surface reflects light. Basically, they represent the surface color.
- *When lighting is active, the material is used instead of color.*
- Material
  - Surface diffuse/ambient/specular reflections
  - Emissive material (to make object appeared to be self-luminous)
  - Sharpness of specular reflection (higher value, smaller highlights)

Vertex Normal

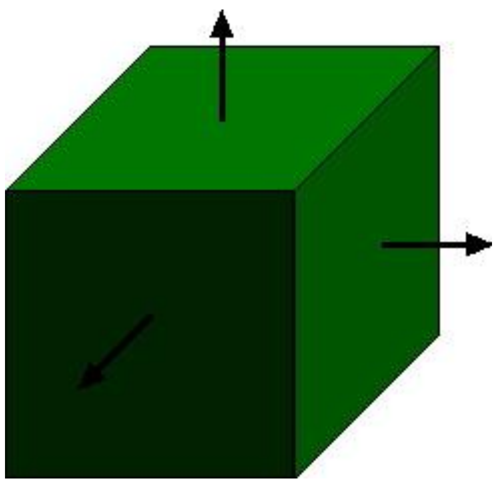
- Normal
  - Lighting computation uses vertex normals
- Vertex structure
  - Use normals instead of colors

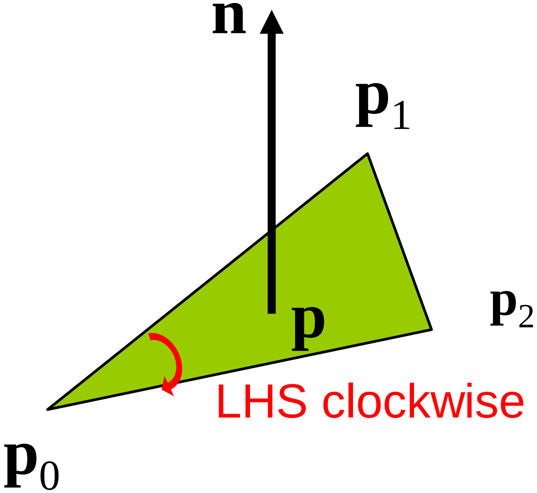


Face normal Vertex normal

Face Normal

- Face Normal
  - Compute the normal vector of a triangle consisting of vertex p0, p1, p2





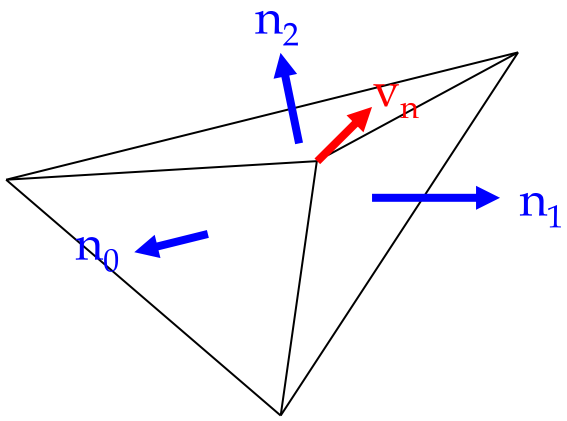
$$\mathbf{n} \cdot (\mathbf{p} - \mathbf{p}_0) = 0$$

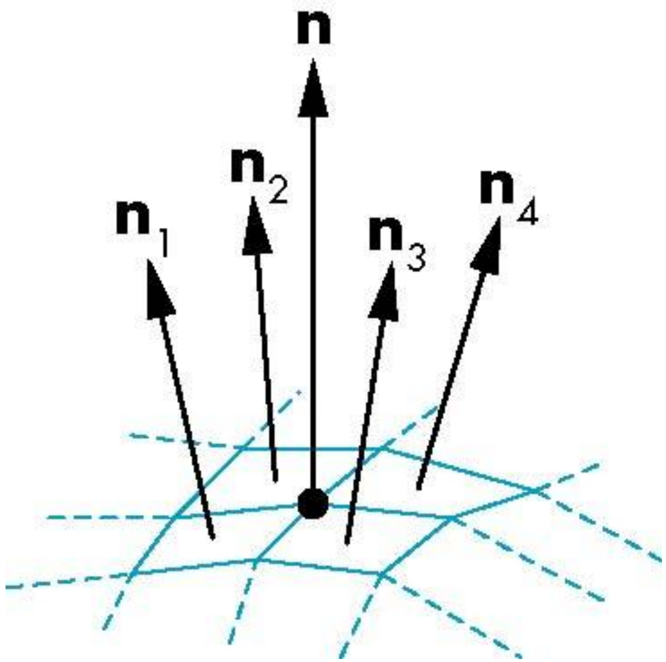
$$\mathbf{n} = (\mathbf{p1} - \mathbf{p}0) \times (\mathbf{p2} - \mathbf{p}0)$$

normalize $\mathbf{n} \leftarrow \mathbf{n} / |\mathbf{n}|$

Vertex Normal

- Vertex Normal
  - Vertex normal calculation using adjacent faces

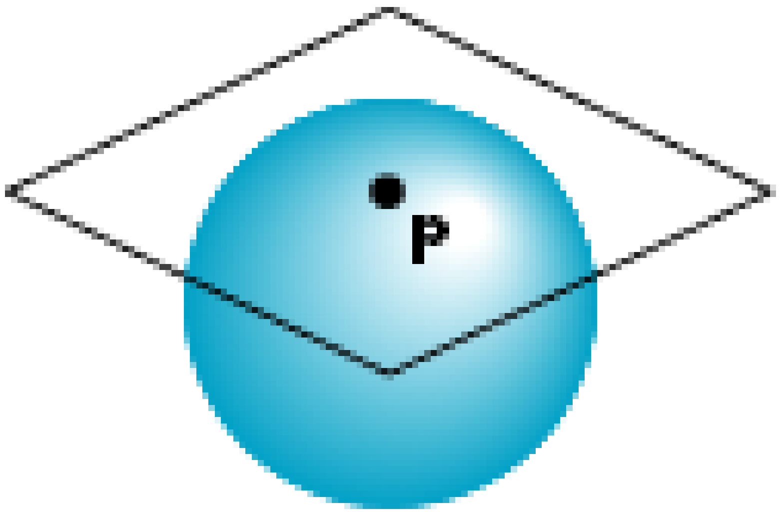




$$n = \frac{n_1 + n_2 + n_3 + n_4}{|\, n_1 + n_2 + n_3 + n_4 \,|}$$

Normal to Sphere

- Implicit function of Sphere:
  - $f(x,y.z)=0$
- Unit Sphere:
  - $f(\mathbf{p})=\mathbf{p}\cdot\mathbf{p}-1 = 0$
- Sphere Normal
  - $n = [\partial f/\partial x, \partial f/\partial y, \partial f/\partial z]^T = \mathbf{p}$



Ambient Lighting

- Ambient lighting
  - **Ambient light**, also known as environmental light, is **light that is present all around the Scene** and doesn't come from any specific source object.
  - It can be an important contributor to the overall look and brightness of a **scene**.

$$Ambient\ Reflection = K_a I_a$$

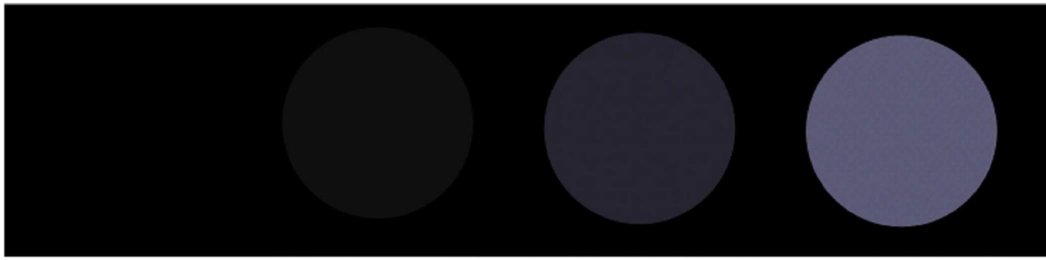
I_a: ambient light intensity

K_a: ambient light coefficient ($0 <= K_a <= 1$)

$$(L_a \otimes M_a)$$

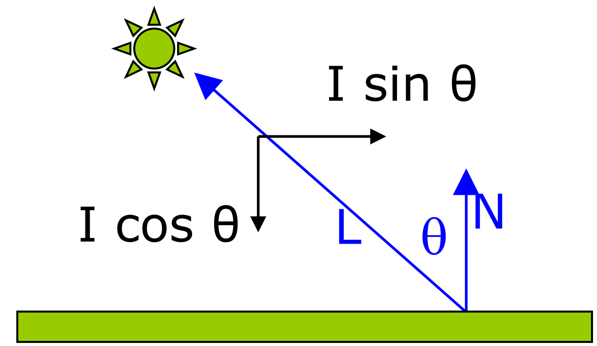
L_a: ambient light color

M_a: ambient material color



Diffuse Lighting

- Lambert's Cosine Law
  - θ between the normal and light vector
  - Maximum intensity when the normal and light vector are perfectly aligned (θ = 0)
  - f(θ) = max(cosθ, 0) = max(L • N, 0)



$$Diffuse\ Reflection \propto \cos\theta$$

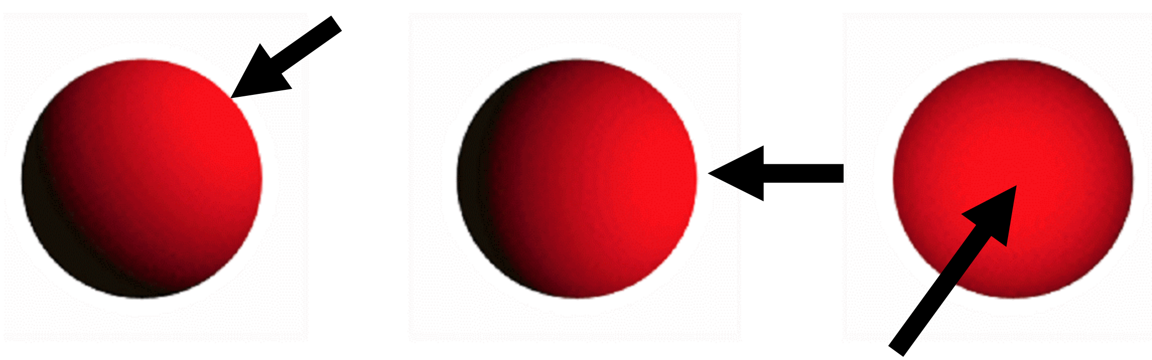
$$Diffuse\ Reflection = K_d I_d \cos\theta = K_d I_d (N \bullet L)$$

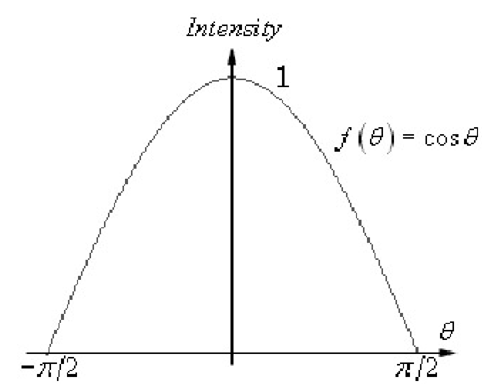
Light vector

Normal vector

I_d: diffuse light intensity

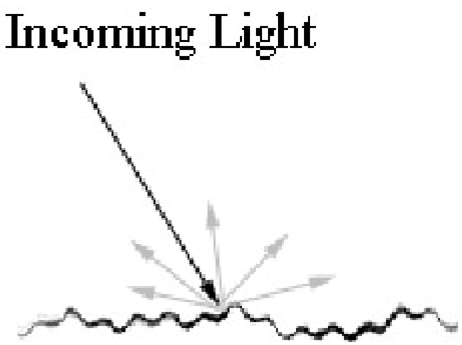
K_d: diffuse light coefficient





Diffuse Lighting

- Diffuse lighting calculation (viewpoint independent)



$$\max(L \bullet N, 0) \bullet (L_d \otimes M_d)$$

L_d: diffuse light color

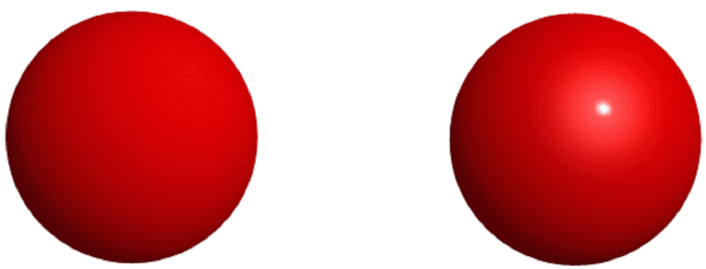
M_d: diffuse material color

L: light vector

N: vertex normal vector

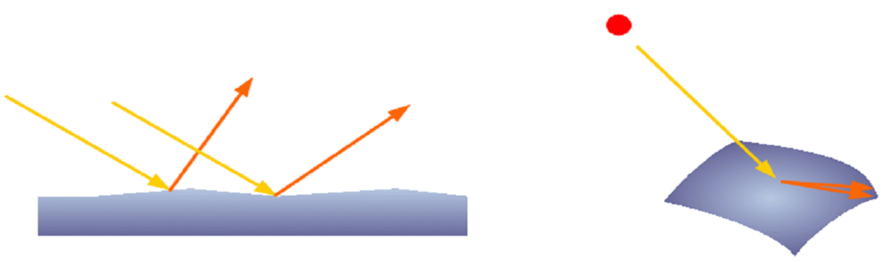
Specular Lighting

- Specular light is a light reflected off a smooth surface.
  - Specular reflection
  - The color of the light source, not the color of the object

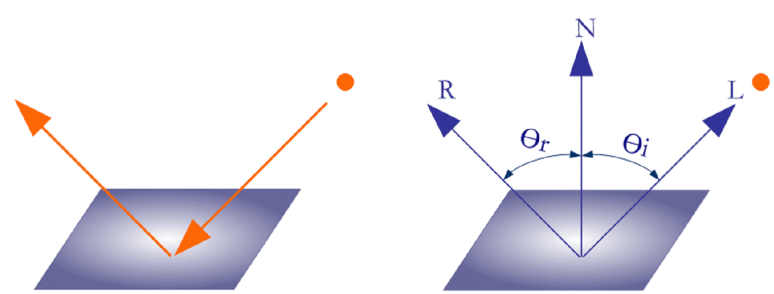


(a) diffuse (b) diffuse + specular

  - Specular light is visible when the viewpoint is in the exact opposite direction.



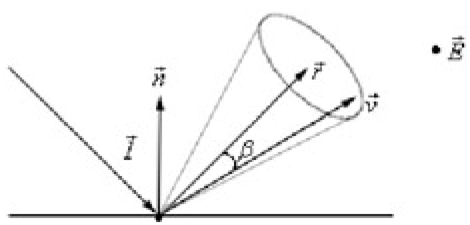
(a) (b)



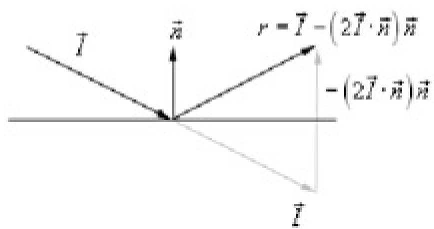
(a) (b)

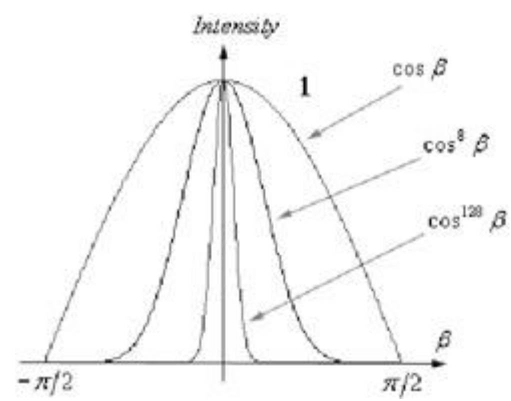
Specular Lighting

- Specular lighting calculation (viewpoint dependent)



$$(\max(V \bullet R, 0))^{p} \bullet (L_s \otimes M_s)$$



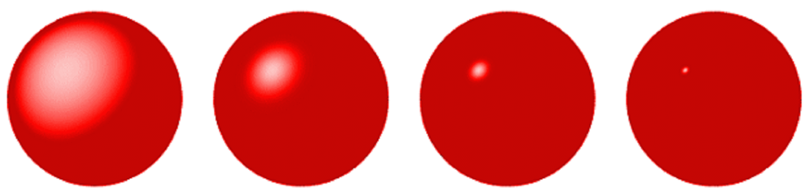


L_s: specular light color

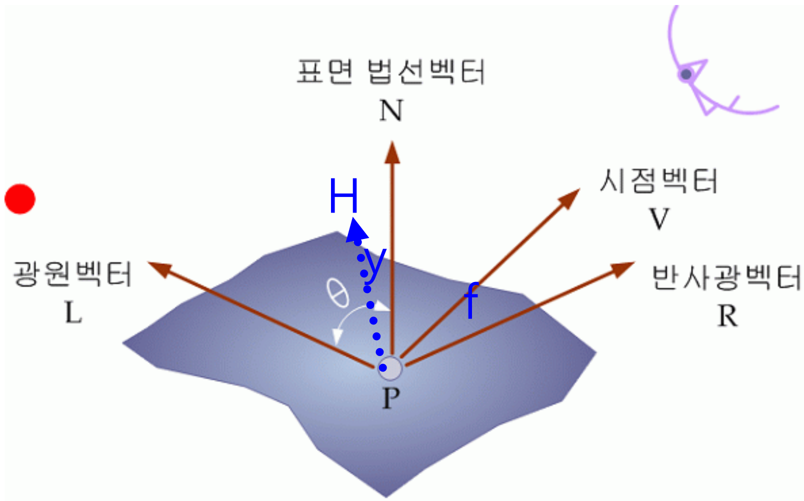
M_s: specular material color

V: view vector

R: light reflection vector, R = L – (2L • N)N



Halfway Vector [Blinn]



$$\text{중간각 } H = \frac{L+V}{\|L+V\|}$$

Shininess

$$\text{Specular Reflection} = K_s I_s (\cos\psi)^n = K_s I_s (N \bullet H)^{n'}$$

Halfway vector

Normal vector

$$where\ 2\psi = \phi$$

$$When\ N \bullet L > 0, H = 1$$

$$When\ N \bullet L \le 0, H = 0$$

Light Attenuation

- Light attenuation
  - Light intensity weakness as a function of distance based on the inverse square law.

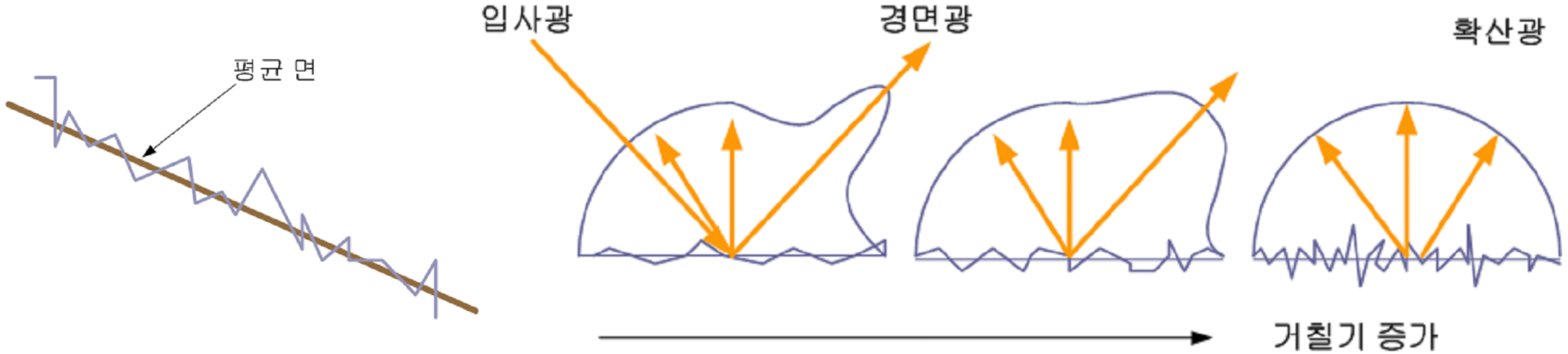
$$I(d) = \frac{I_0}{d^2}$$

$$I(d) = \frac{I_0}{a_0 + a_1 d + a_2 d^2}$$

$$d = \|S - P\| (i.e., the\ distance\ between\ P\ and\ S)$$

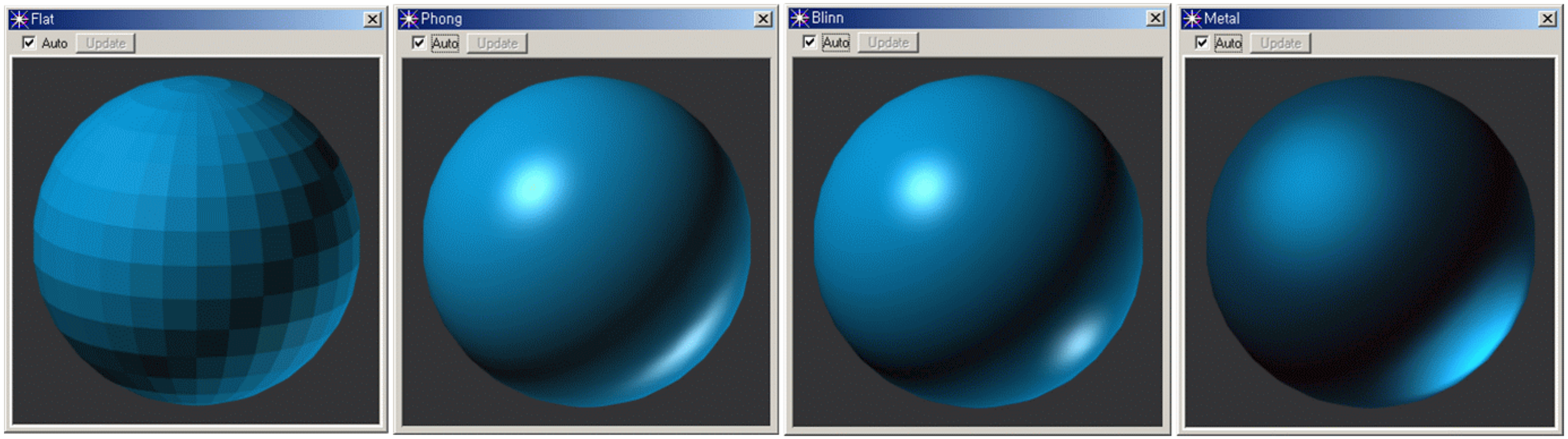
Microfacet Model

- Modeling the roughness of the surface
  - Based on the direction of the average plane
  - Controls the curvature or shape of microsurfaces using a parameter called surface roughness



Microfacet Model

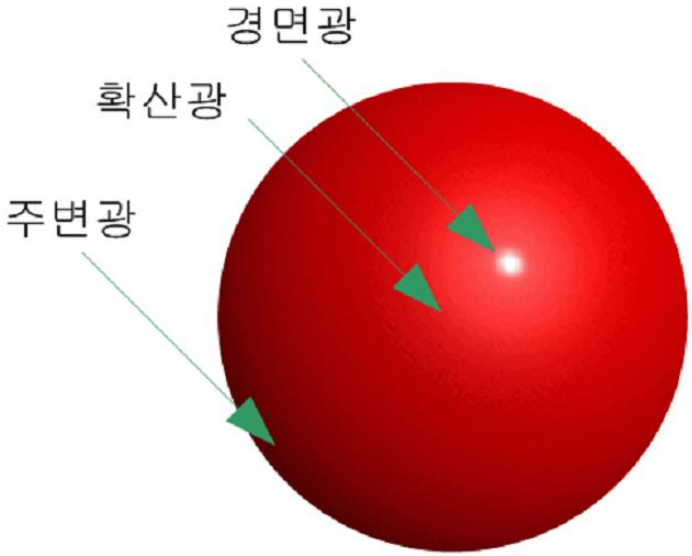
- Flat, Phong, Blinn, Cook-Torrance Shading



- Blinn
  - Similar to Phong. The specular light component spreads more gently.
- Cook-Torrance (Metal shading)
  - Advantageous for subtle specular light on metal surfaces
  - Phong model: plastic material

$$I_{specular} = K_s I_s (\hat{N} \cdot \hat{H})^{\beta}$$

Direct Illumination Model



$$I = K_a I_a + K_d I_d (N \bullet L) + K_s I_s (R \bullet V)^n$$

Ambient reflection

Diffuse reflection

Specular reflection

Light Attenuation

- Adjust the brightness intensity according to the distance between the light source and the object

$$I = K_a I_a + f_{att}(d)\left\{K_d I_d (N \bullet L) + K_s I_s (N \bullet H)^n\right\}$$

$$f_{att}(d) = \frac{1}{d^2}$$

$$f_{att}(d) = \frac{1}{k_0 + k_1 d + k_2 d^2}$$

$$f_{att}(d) = \min\left(\frac{1}{k_0 + k_1 d + k_2 d^2}, 1\right)$$

Multiple Light Sources

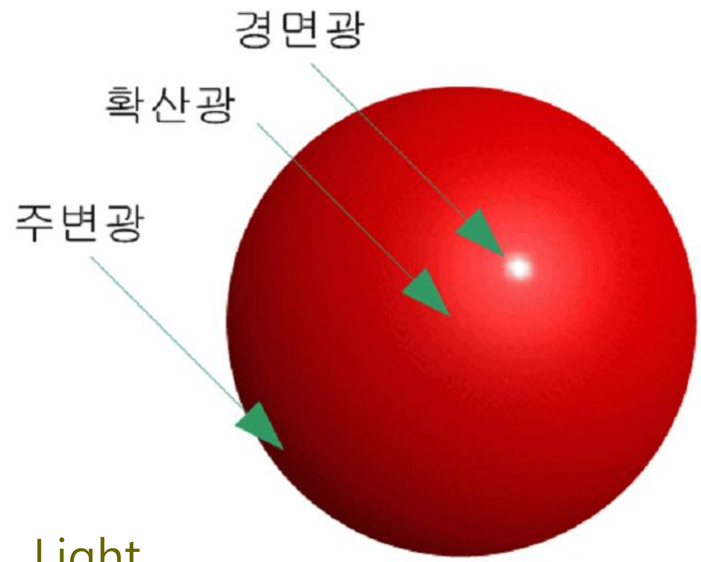
- Multiple light sources

$$I = K_a I_a + \sum_{i=0}^{m-1} f_{att}(d)\left\{K_d I_d (N \bullet L) + K_s I_s (N \bullet H)^n\right\}$$

- Emissive illumination $I_e = E$
  - Certain objects not only reflect light but also emit light, which is called emissive lighting. Simply add the color of the emitted light.

$$I = K_a I_a + \sum_{i=0}^{m-1} f_{att}(d)\left\{K_d I_d (N \bullet L) + K_s I_s (N \bullet H)^n\right\} + E$$

Direct Illumination Model



Multiple lights

Light attenuation

$$I = K_a I_a + \sum_{i=0}^{m-1} f_{att}(d)\{K_d I_d (N \bullet L) + K_s I_s (N \bullet H)^n\} + E$$

Ambient reflection

Diffuse reflection

Specular reflection

Emissive light

OpenGL uses the modified Phong model (Blinn model) with Halfway vector.