Basic Shading Algorithms

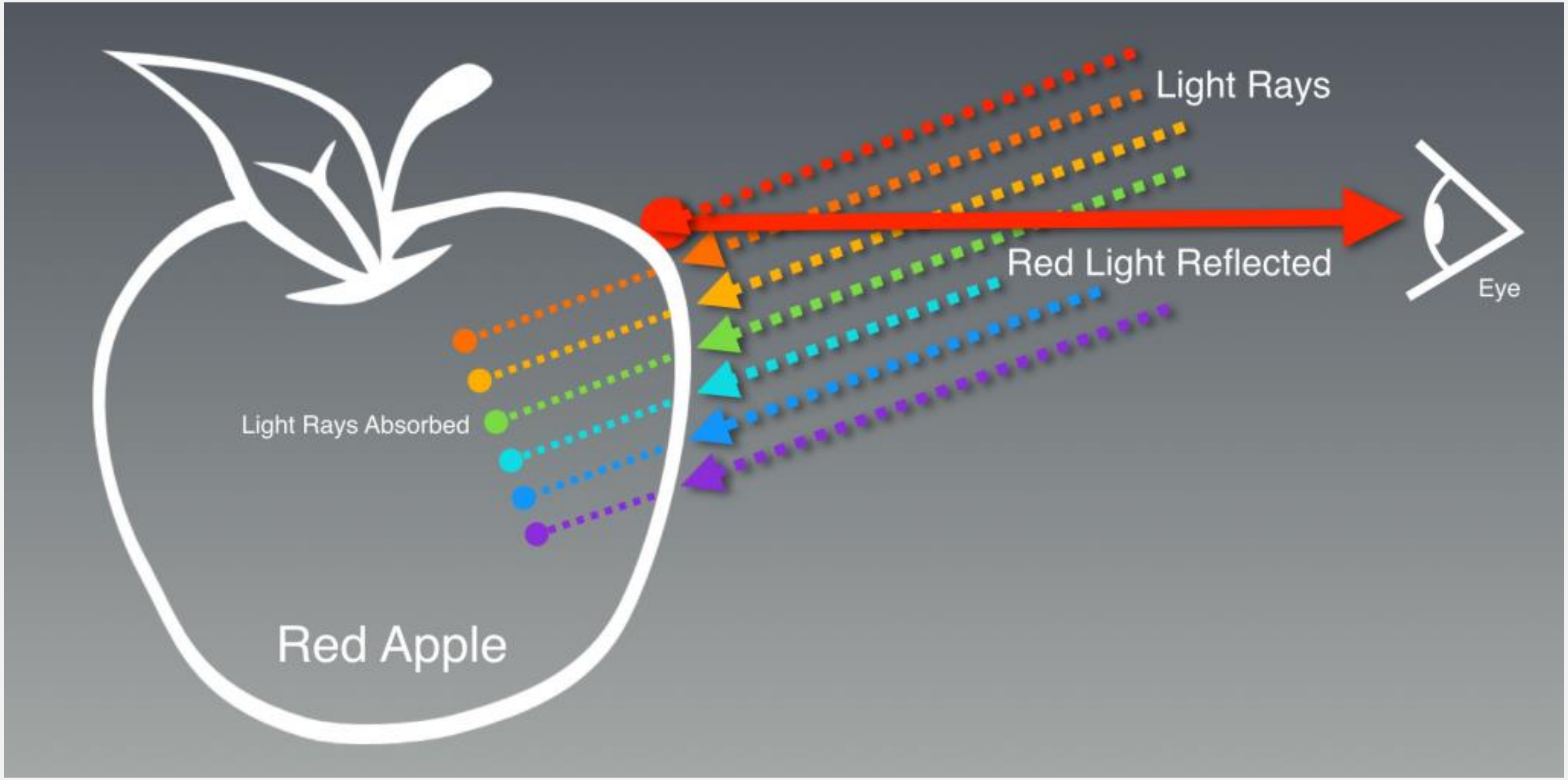
2017 Fall

National Cheng Kung University

Instructor: Min-Chun Hu 胡敏君

Shih-Chin Weng 翁士欽

How Do We Perceive Color?

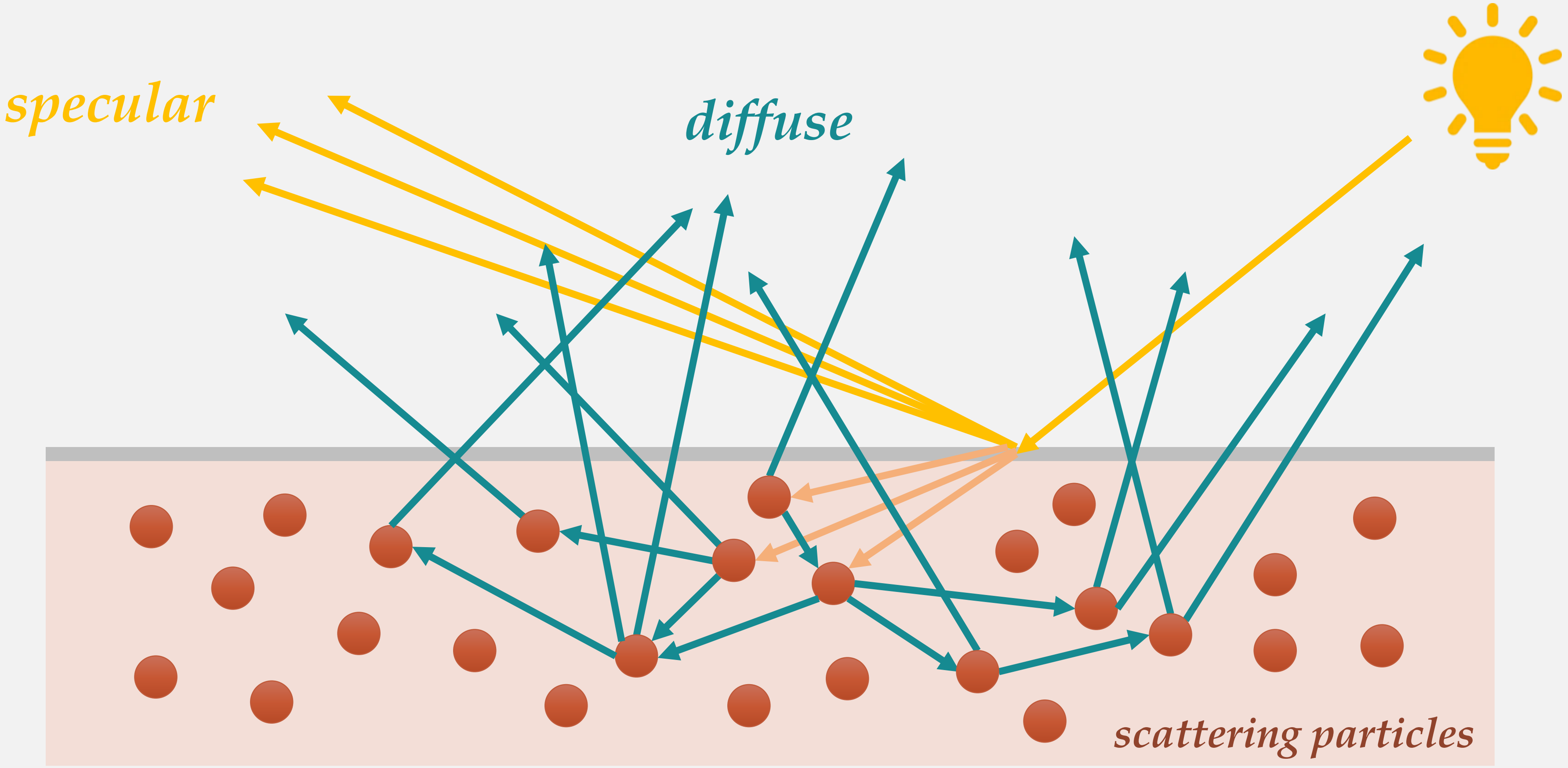


[figure from PBR Guide Vol. 1 by Allegorithmic]

Key Ingredients for Visual Appearance

- Light (illuminant)
  - Emittance spectrum (i.e. spectral power distribution)
  - Position, direction, shape, attenuation, etc.
- Object (material)
  - Geometric structure (smooth or rough)
  - Absorption or scattering
  - Reflectance, index of refraction, etc.
- Sensor (camera)
  - Tone mapping: convert physical quantity to color appearance

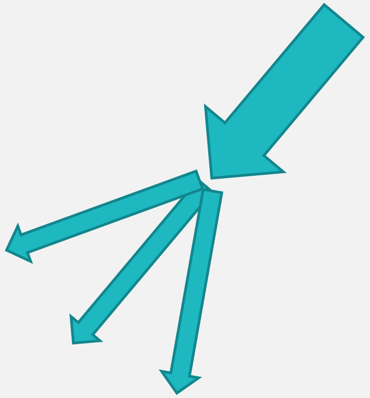
Light-material Interactions



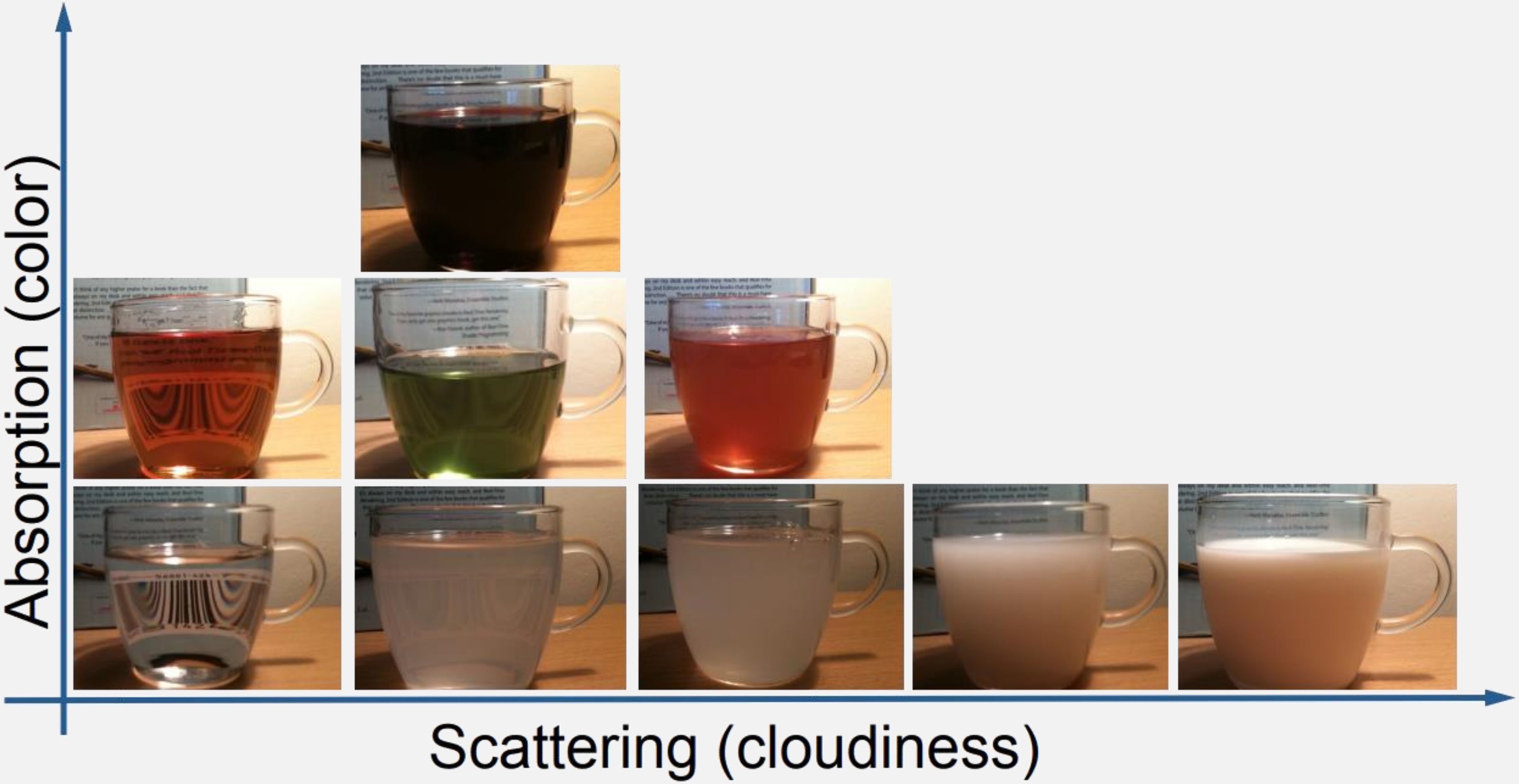
Absorption & Scattering

- Absorption (transparency)
  - Light intensity decreases as it changes into other form of energy (ex. heat)
    - The color of reflected light is changed
  - Light direction doesn't change

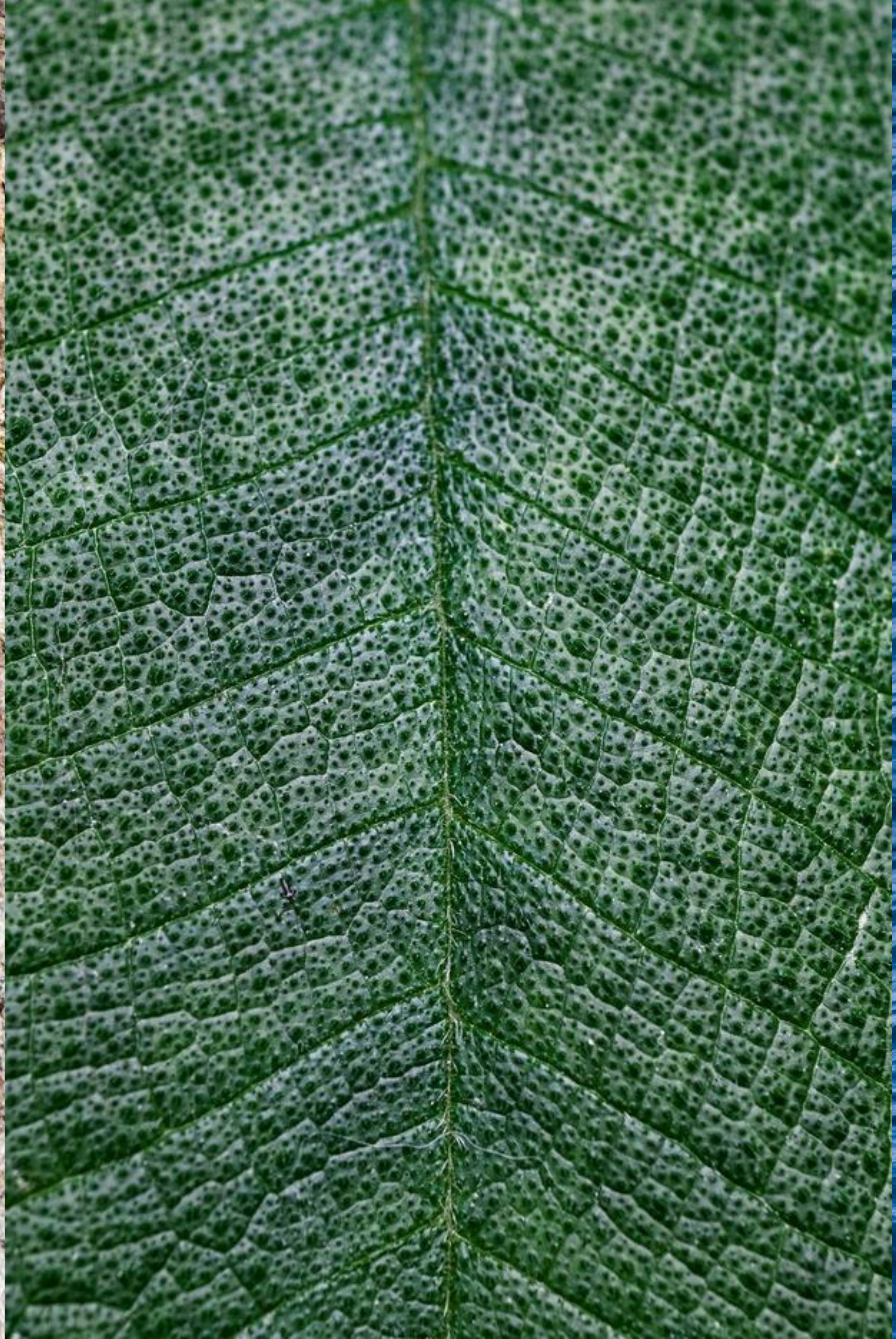
- Scattering (translucency)
  - Light direction changes randomly according to the material properties
  - Light intensity remains the same



Absorption & Scattering (Cont.)



[figure from *"Physics and Math of Shading"*, Naty Hoffman]

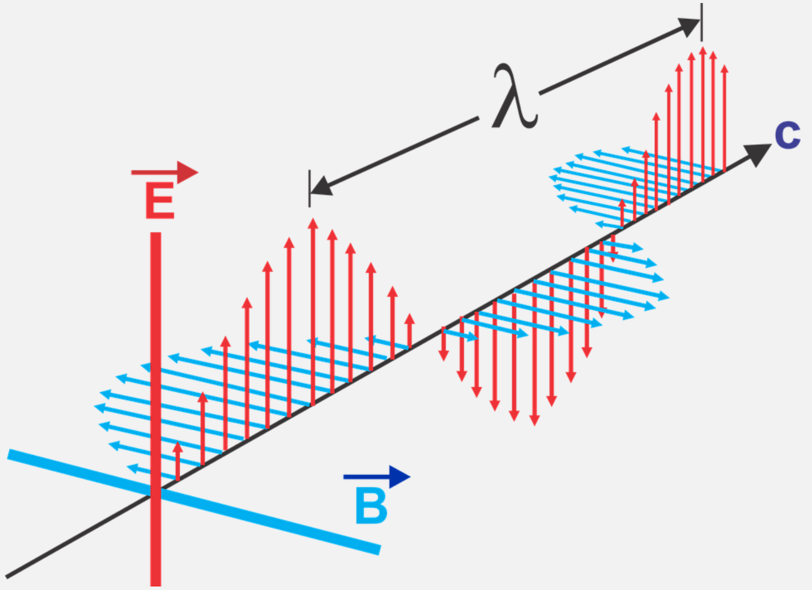



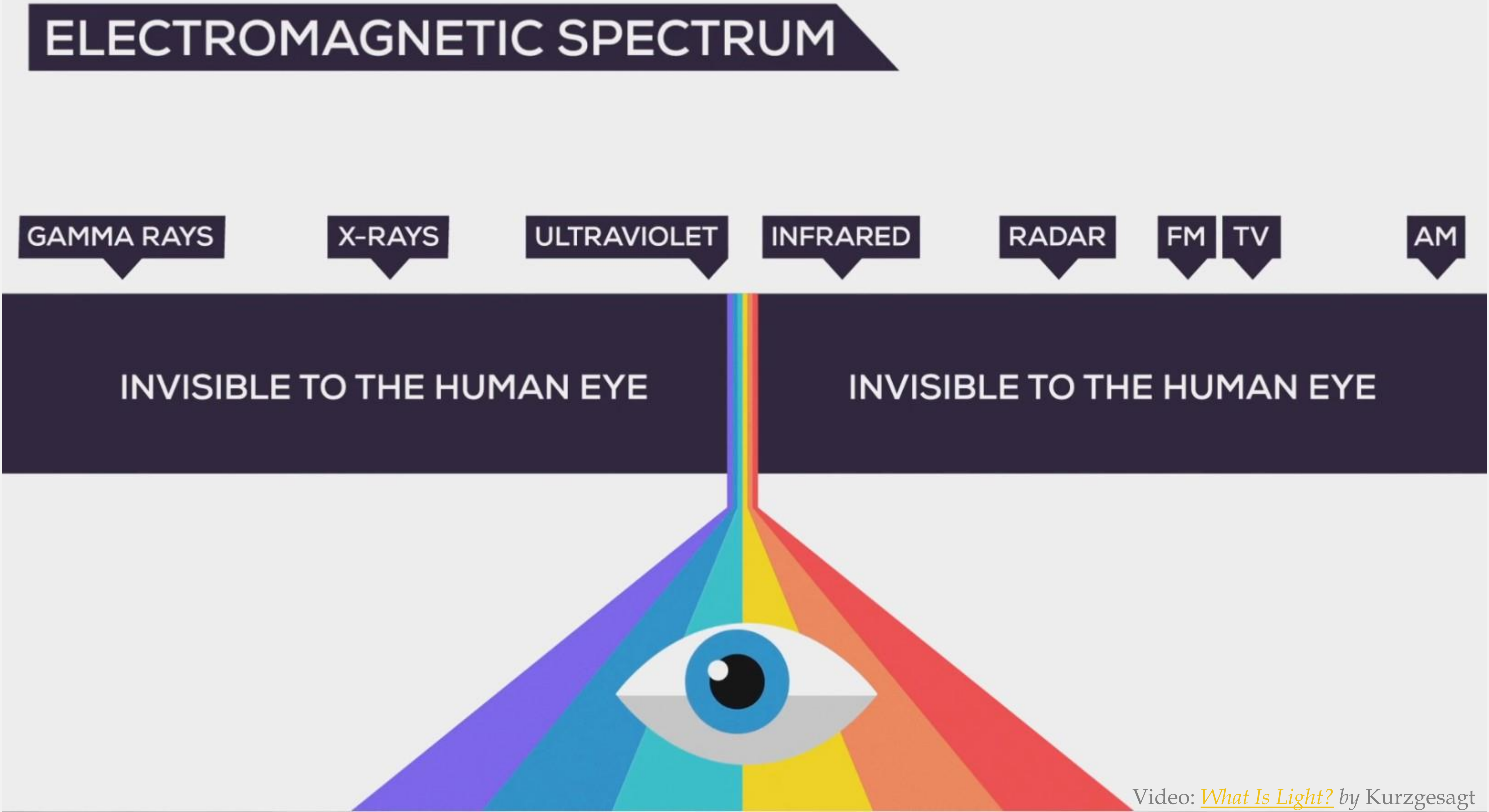


Firebird

What is Light?

- Light is a form of electromagnetic radiation
  - Emitted when atoms or molecules drop from higher state to lower one
  - Travels through vacuum at speed $c \approx 3 \times 10^{8}\ m/s$
- Wave-particle duality
- Energy of a single photon $Q = \frac{hc}{\lambda}$
- No mass at all
- Do not need a medium to transport





Video: *What Is Light?* by Kurzgesagt

Radiometry

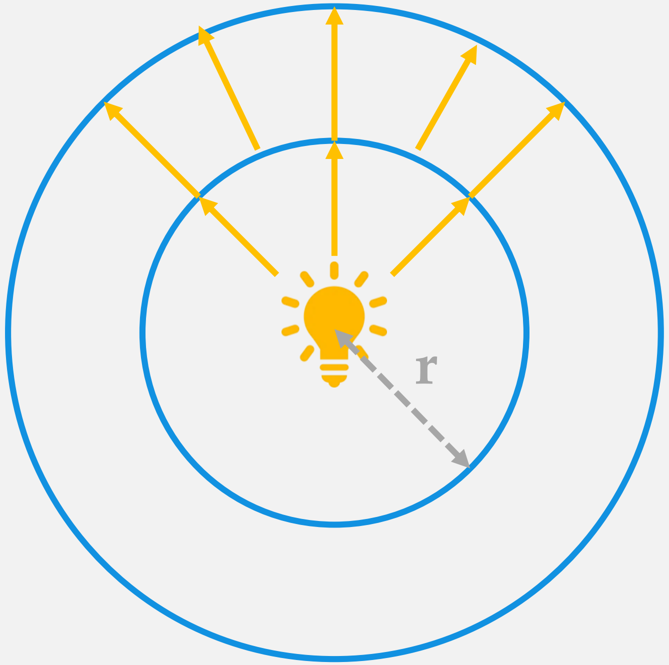
Radiant flux $\Phi = \frac{\mathrm{dQ}}{\mathrm{dt}}$ (J/sec)

The total amount of energy passing through a region of surface per unit time

Irradiance $E = \frac{d\Phi}{dA}$

Pre area incoming flux at a surface

Radiant Exitance or Radiosity $M = B = \frac{d\Phi}{dA}$

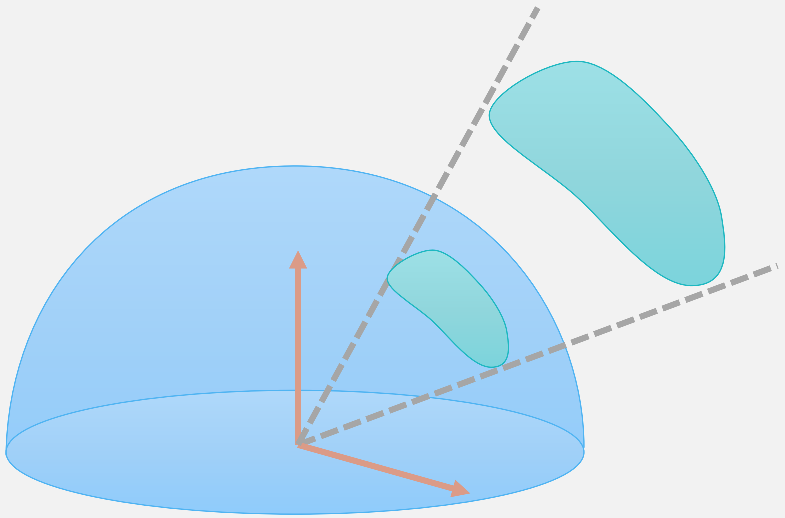


$$\boxed{E = \frac{\Phi}{4\pi r^2}}$$

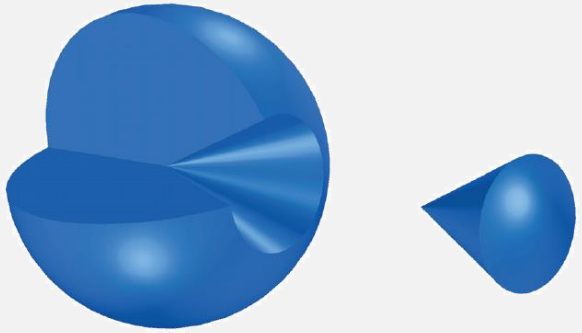
*the total amount Φ measured at **inner** and **outer** sphere is **the same** (equals to the radiant flux of the point light)*

Solid Angle

$$\Omega = \frac{A}{r^2}$$



- A: the total surface area on the sphere
- r: the radius of the sphere

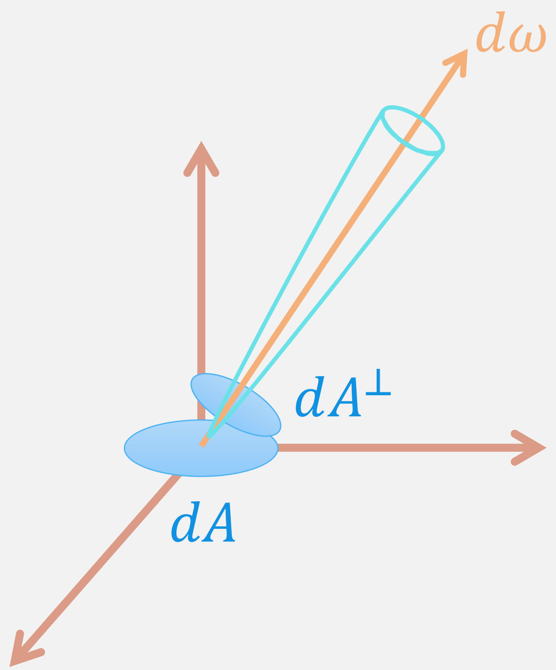


- The total area on a unit sphere subtended by the object
  - Concept: a set of directions
- Measured in steradians (sr)
- Often denoted as ω

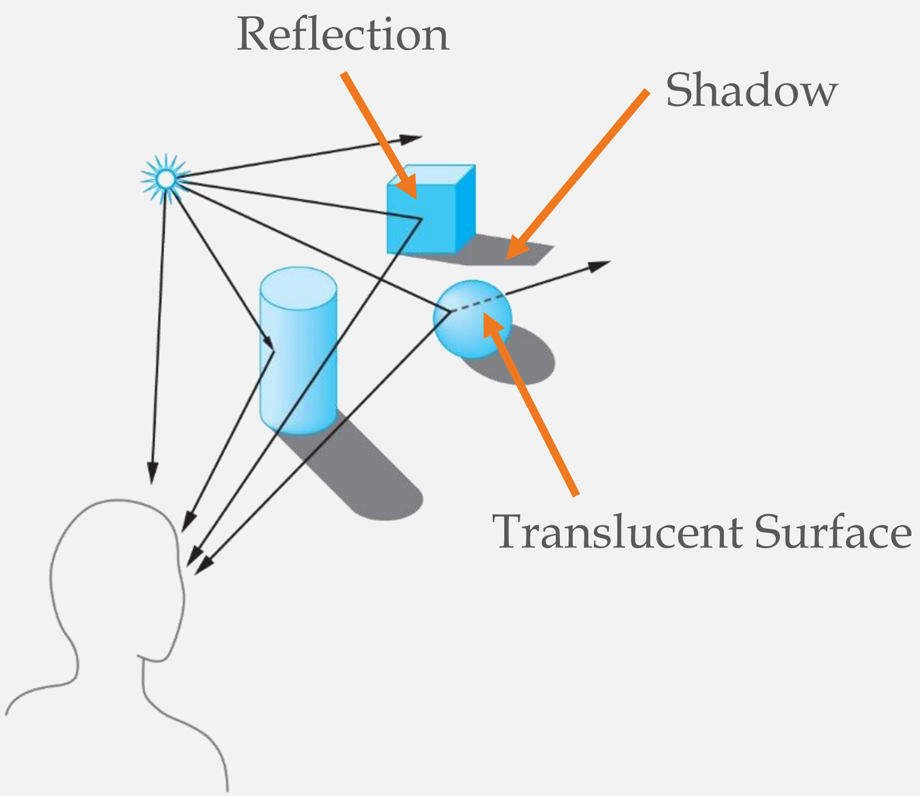
Radiance

$$L = \frac{d^2 \overbrace{\boxed{\Phi}}^{\text{flux}}}{d\underbrace{\boxed{\omega}}_{\text{solid angle}} d\boxed{A^{\perp}}} = \frac{d^2\Phi}{d\omega d\boxed{A \cos\theta}}$$

solid angle projected area



The density of photons passing near x and traveling in directions near ω.
It measures the intensity of light along a single ray

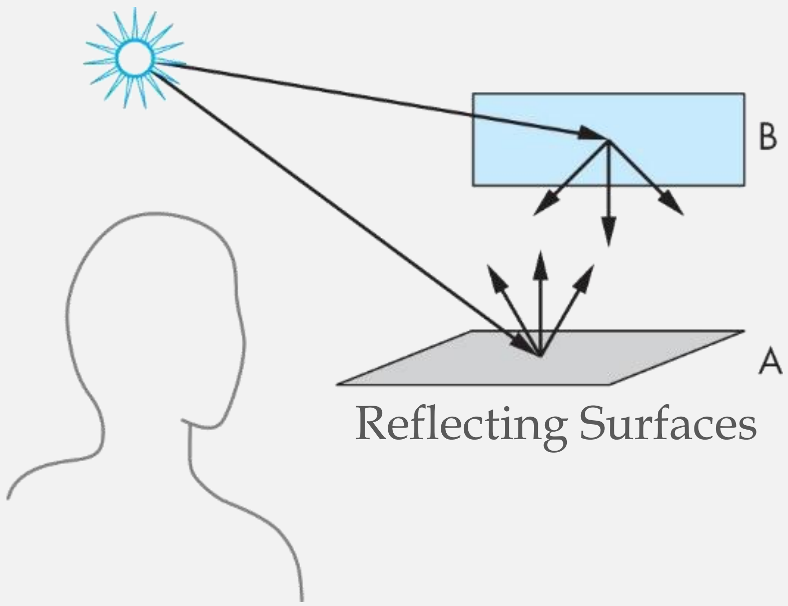


Shading

The process of computing radiance L_o along with ***the view ray v***, ***material properties*** *and* ***light sources***.

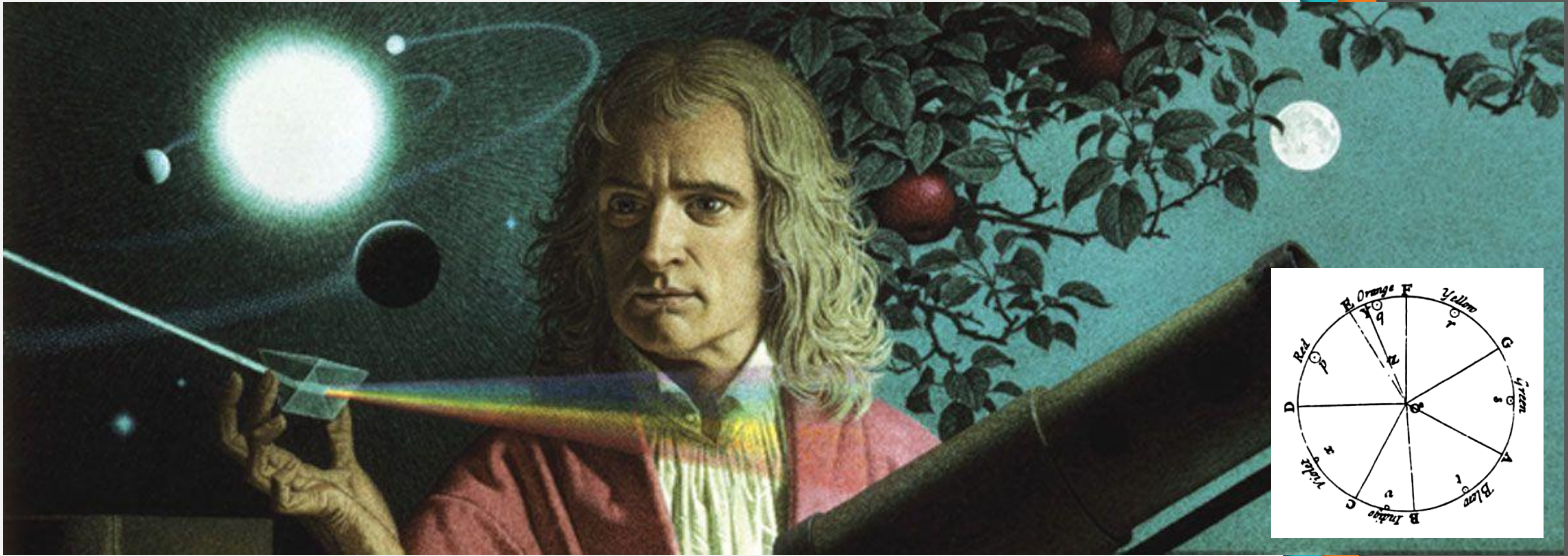
Light Transport

- The radiance of a pixel is determined by multiple interactions among light sources and surfaces
  - Light strikes A
    - Some scattered
    - Some absorbed
  - Some of scattered light strikes B
    - Some scattered
    - Some absorbed
  - Some of this scattered light strikes A and so on



- These interactions can be seen as a recursive process

Light exists in the physical world, but color **ONLY** exists in the eye and the brain!!



https://www.davidstreams.com/mis-apuntes/isaac-newton-ficha-cientifico/

Illumination and Shading

- Radiance is a physical quantity, while the ‘color’ is perceptual sensation
  - In shading process, color vector (r, g, b) is used to store the value of radiance
    - Those values can exceed 1
- Object may absorb some energy of incoming light and caused the reflected light has different spectral power distribution
  - This could be interpreted as ‘surface color’ if the incoming light is white

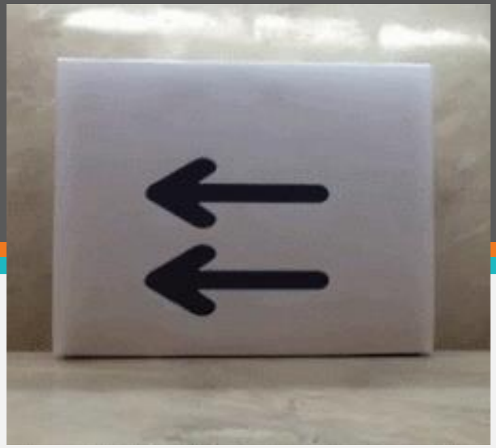
Geometric Optics

- **Assumption**
  - the wavelength of light is *much smaller than* the scale of interacted object
- Light travels
  - in straight lines
  - instantaneously through a medium
- Light is **NOT** influenced by gravity or magnetic fields
  - No diffraction, dispersion
  - But the movie “Interstellar” does simulate the light bent by gravity!!

Snell’s Law

Photo by Gabriel Gurrola

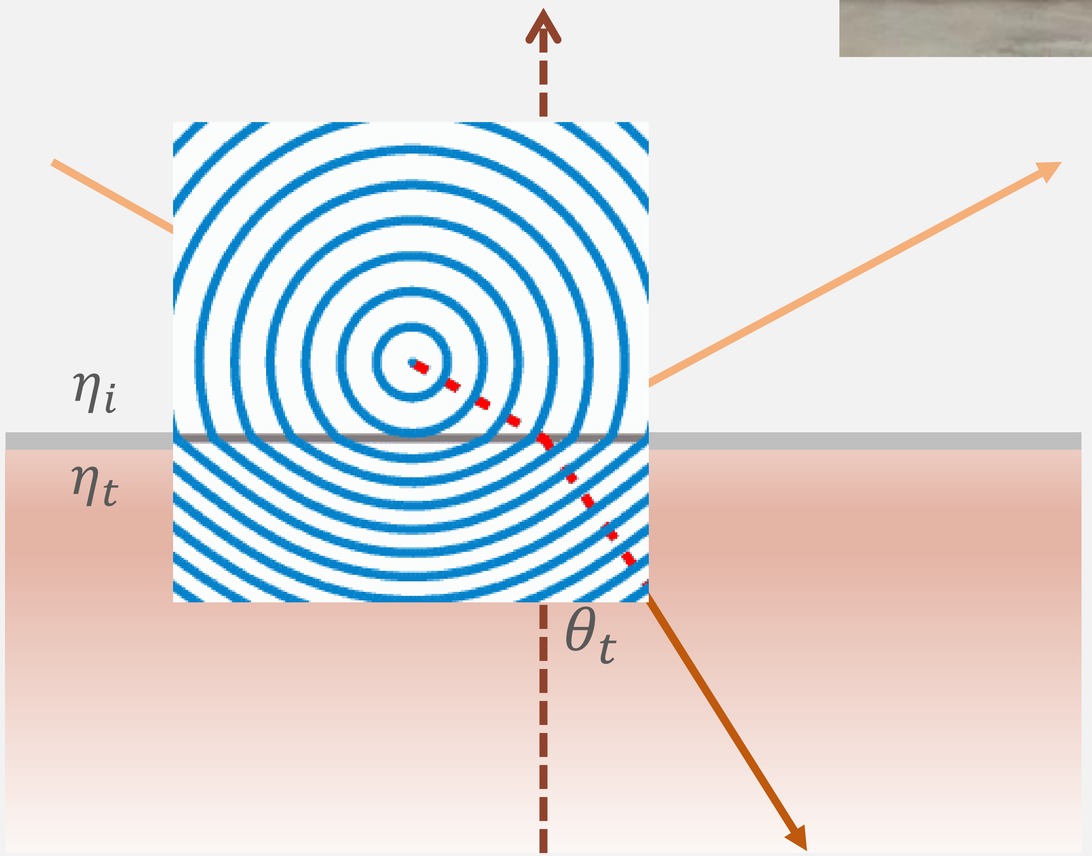
Snell's Law



$$\sin\theta_i\,\eta_i = \sin\theta_t\,\eta_t$$

Index of Refraction (IOR): η

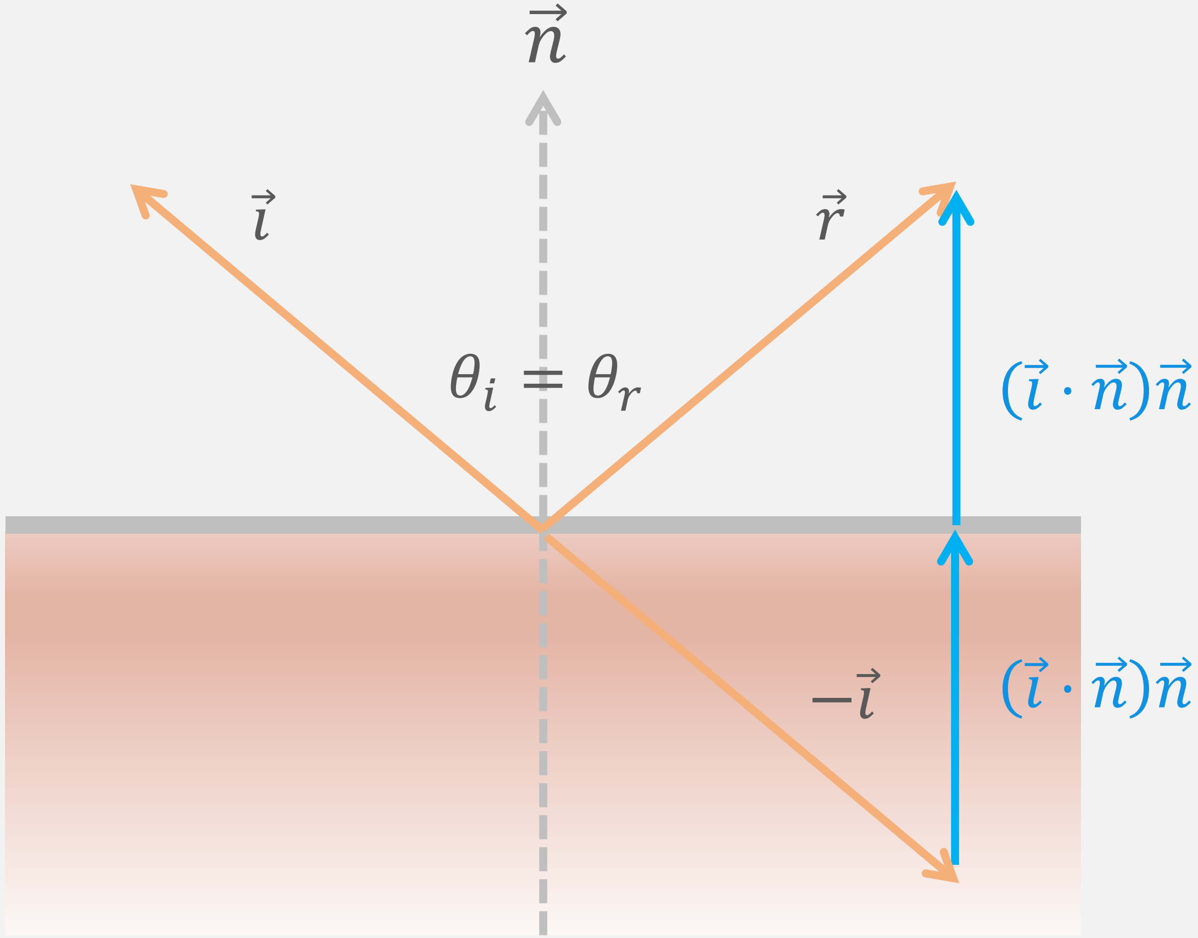
$$\eta(\lambda) = \frac{c_0}{v_p(\lambda)}$$



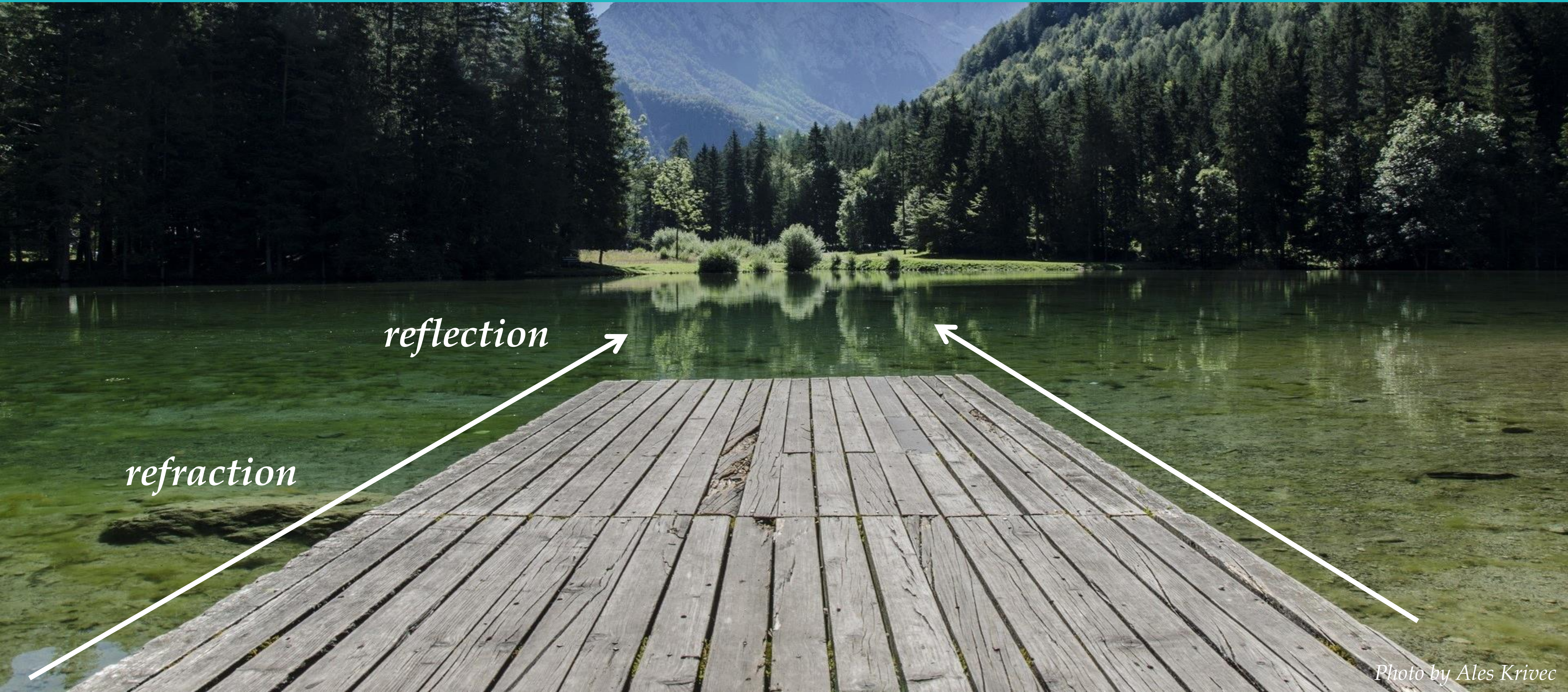
https://media.giphy.com/media/26FPuuFM8UlQIMymA/giphy.gif
https://en.wikipedia.org/wiki/Snell%27s_law#/media/File:Snells_law_wavefronts.gif

Perfect Specular Direction

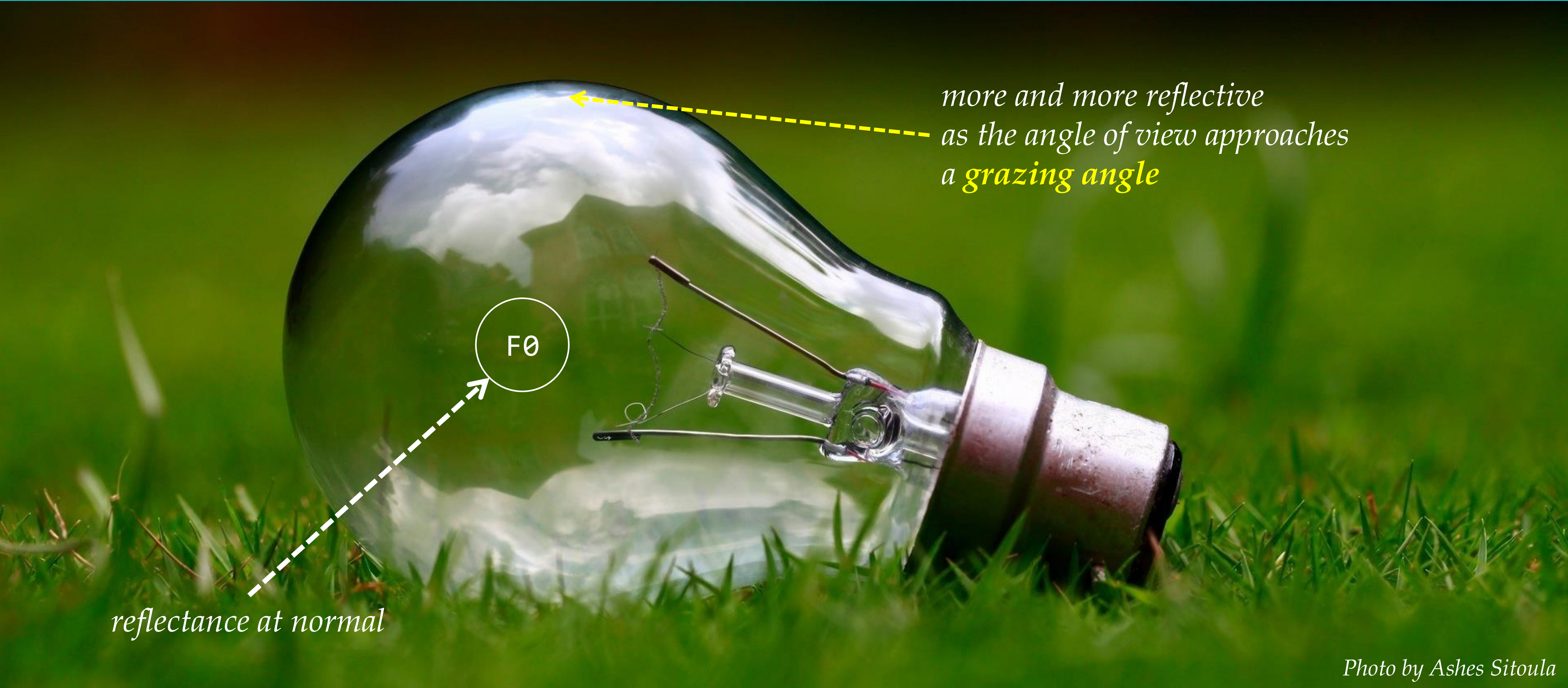
$$\boxed{\vec{r} = -\vec{\imath} + 2(\vec{\imath} \cdot \vec{n})\vec{n}}$$



Fresnel Effect



Fresnel Effect

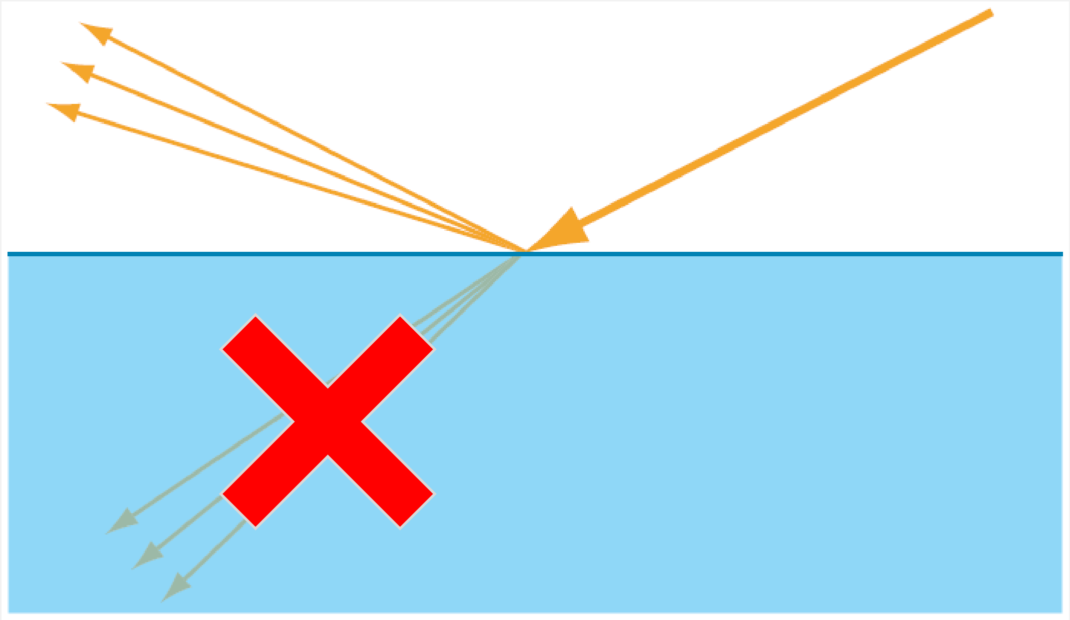


Fresnel

- Fresnel reflectance
  - The amount of reflected light w.r.t. the viewing angle
- Relates the ratio of reflected and transmitted energy as a function of
  - Incident direction
  - Polarization
  - Material properties

Metals

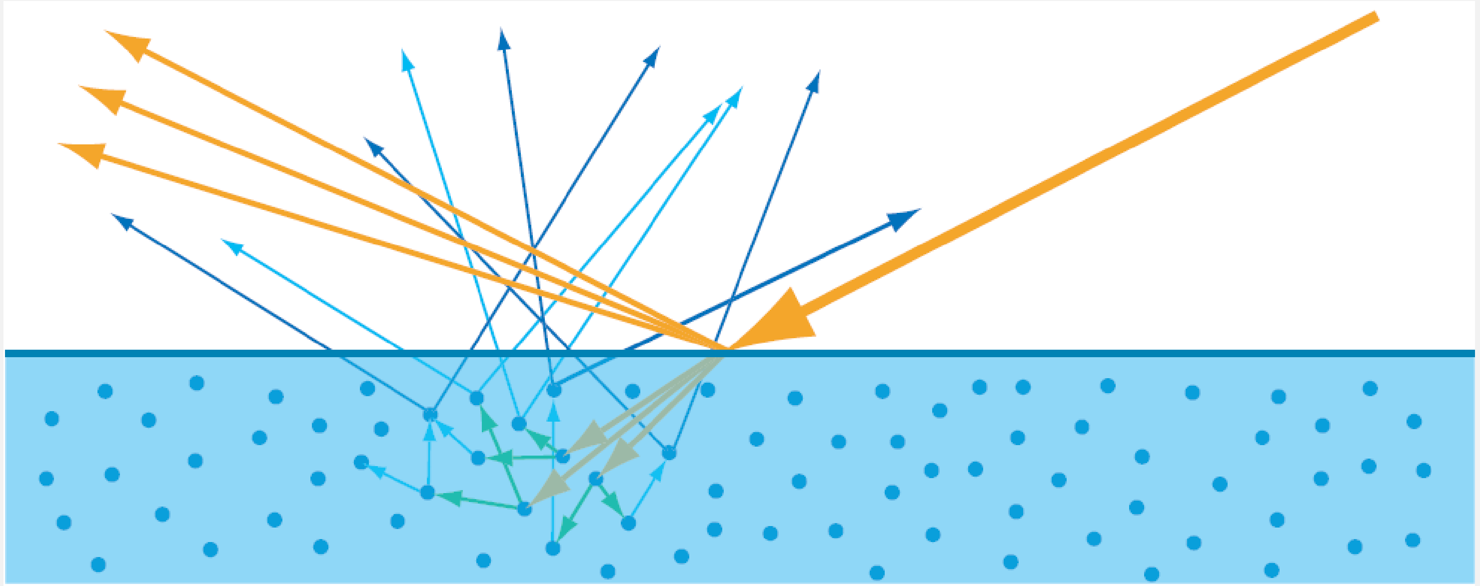
- IOR strongly depends on the wavelength
- Immediately absorbs refracted lights (i.e. no refraction)
  - The reflected lights would change their color



[figure from *Real-time Rendering*, 3/e]

Non-Metals (Insulators, Dielectrics)

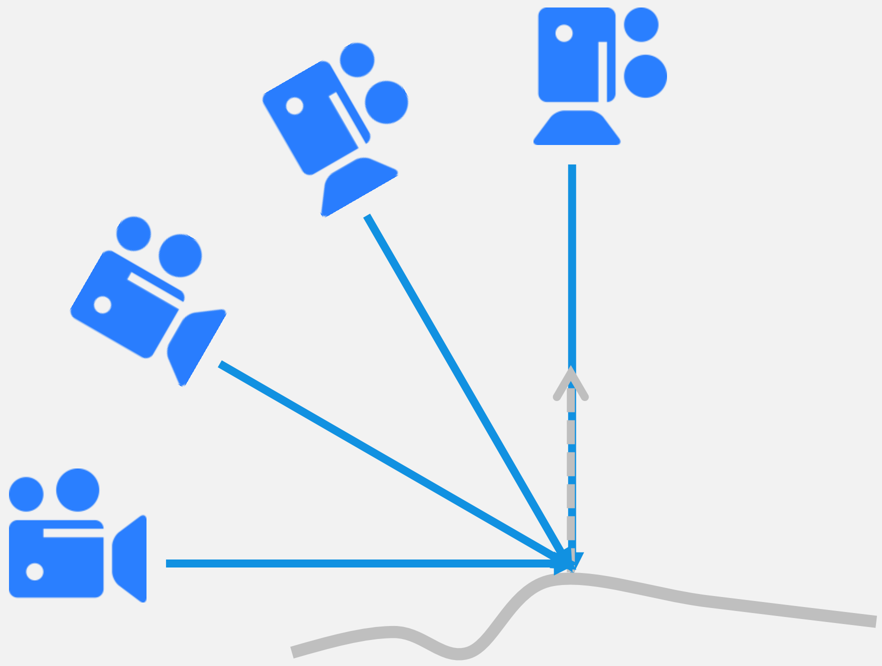
- Refracted light undergoes scattering and/or absorption, often re-emerging from the surface
  - Only reflect 4~10% of incoming light in average
  - The reflection intensity is independent on the wavelength

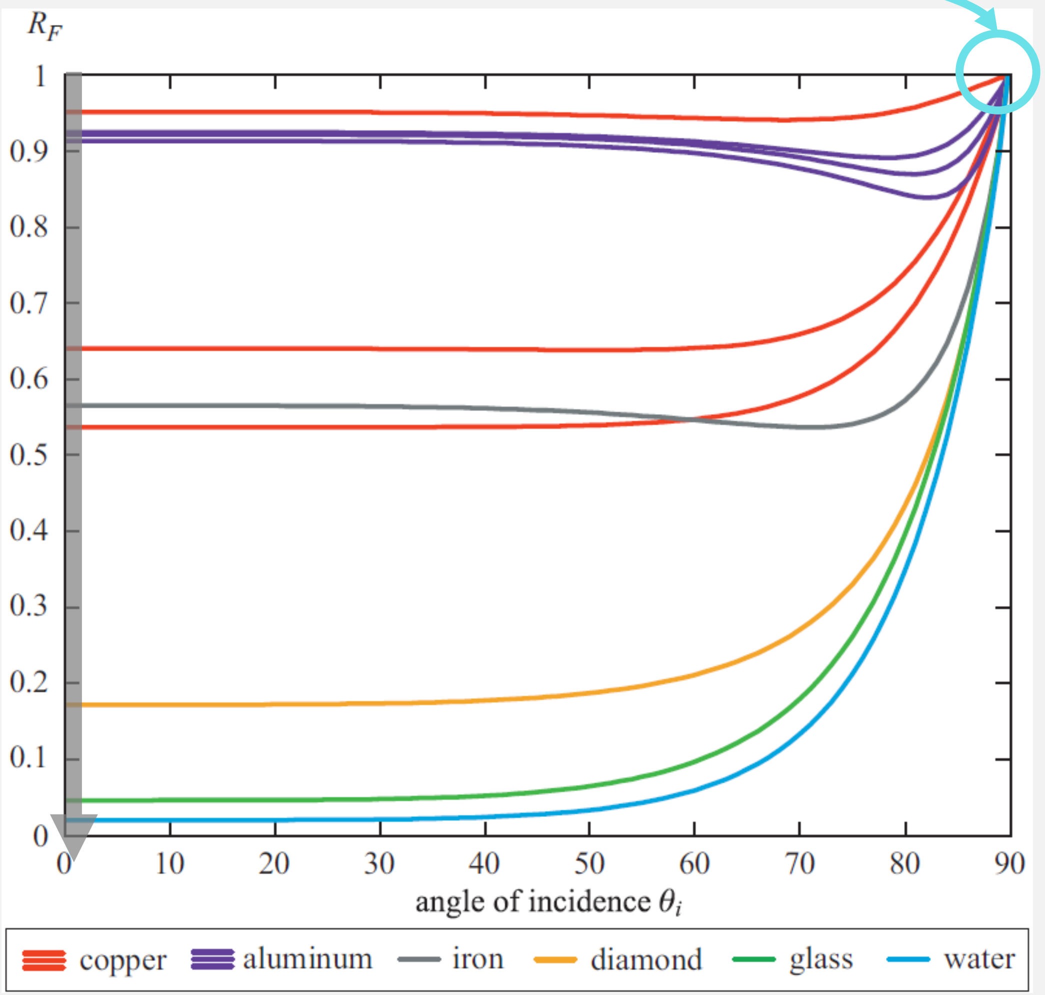


[figure from *Real-time Rendering*, 3/e]

Fresnel Reflectance

*Reflection goes to **100%** at grazing angle!*





[figure from *Real-time Rendering*, 3/e]

Fresnel Reflectance (Cont.)

$$\mathrm{F}_{\mathrm{Schlick}} = \mathrm{F}_0 + (1 - \mathrm{F}_0)(1 - \overline{\cos\theta_i})^5$$

Dielectric

$$\mathrm{r}_{\parallel} = \frac{\eta_t \cos\theta_i - \eta_i \cos\theta_t}{\eta_t \cos\theta_i + \eta_i \cos\theta_t}$$

$$\mathrm{r}_{\perp} = \frac{\eta_i \cos\theta_i - \eta_t \cos\theta_t}{\eta_i \cos\theta_i + \eta_t \cos\theta_t}$$

$$\mathrm{F_r} = \frac{1}{2}\left(r_{\parallel}^2 + r_{\perp}^2\right)$$

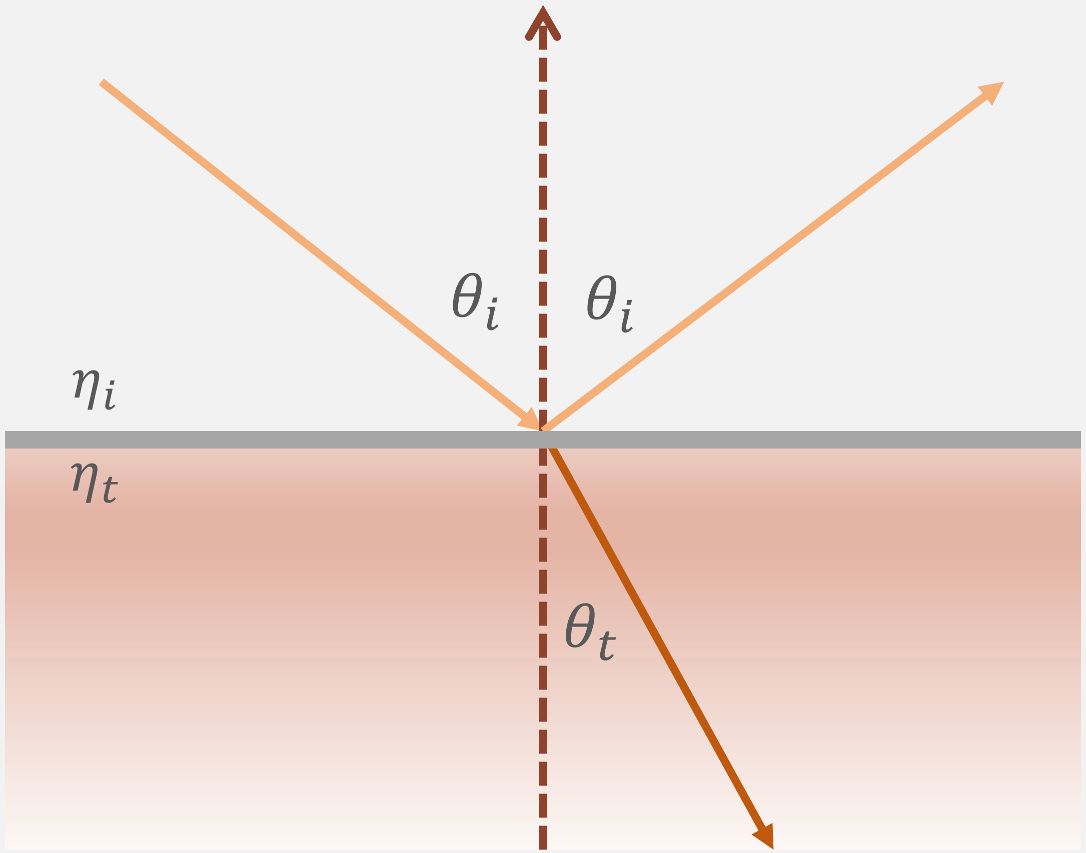
for unpolarized light

Conductor

$$\mathrm{r}_{\parallel}^2 = \frac{(\eta^2 + k^2)\cos^2\theta_i - 2\eta\cos\theta_i + 1}{(\eta^2 + k^2)\cos^2\theta_i + 2\eta\cos\theta_i + 1}$$

$$\mathrm{r}_{\perp}^2 = \frac{(\eta^2 + k^2) - 2\eta\cos\theta_i + \cos^2\theta_i}{(\eta^2 + k^2) + 2\eta\cos\theta_i + \cos^2\theta_i}$$

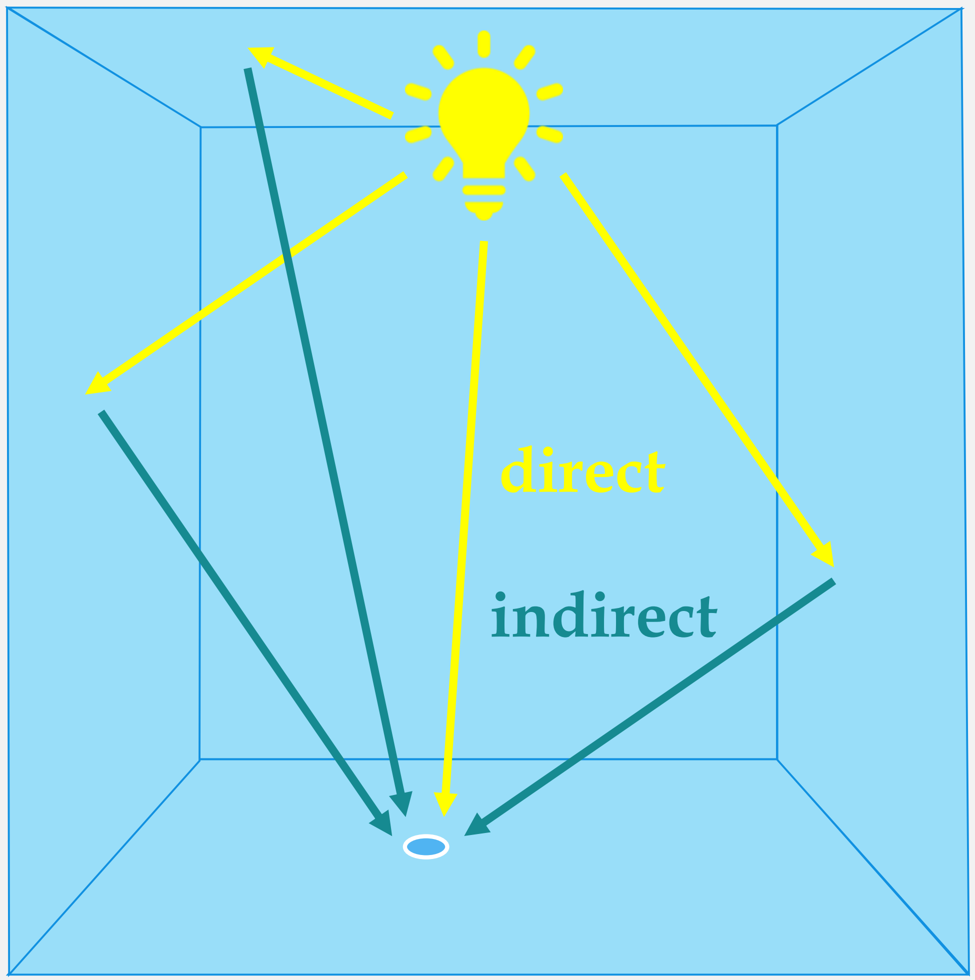
$$\tilde{\eta}(\lambda) = \eta(\lambda) + ik(\lambda)$$



Shading Pixels

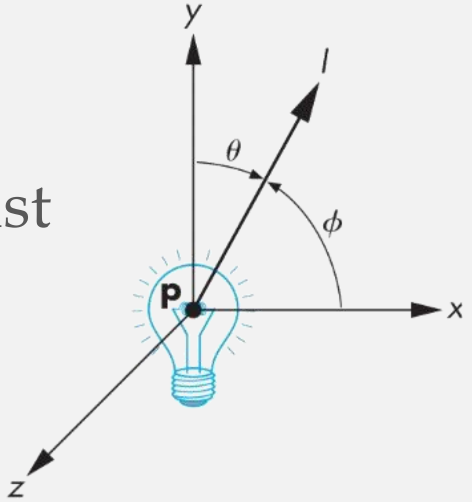
- **Local reflection model**
  - Compute the radiance of direct lighting at **shading point p** with
    - local frame (normal, tangent and binormal)
    - light and view directions
    - lights' properties: emittance spectrum, energy attenuation, etc.
    - material's reflection/refraction properties
- **Shading algorithms (interpolation techniques)**
  - Compute the radiance of a **pixel**
    - Apply interpolative values to local reflection model

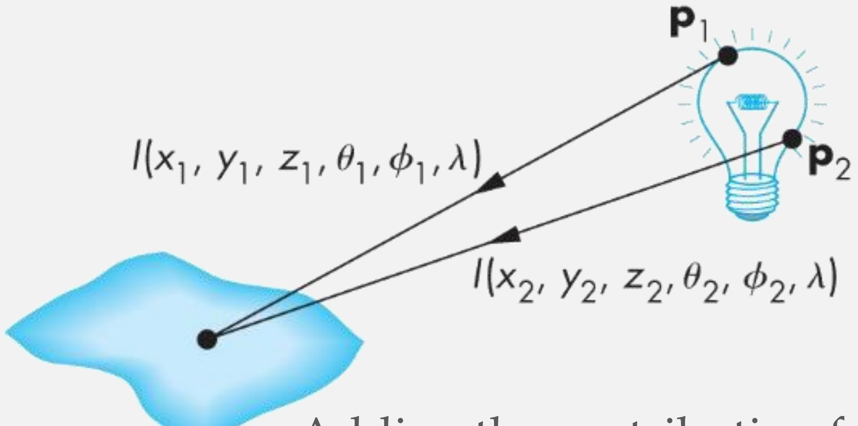
Direct & Indirect Lighting



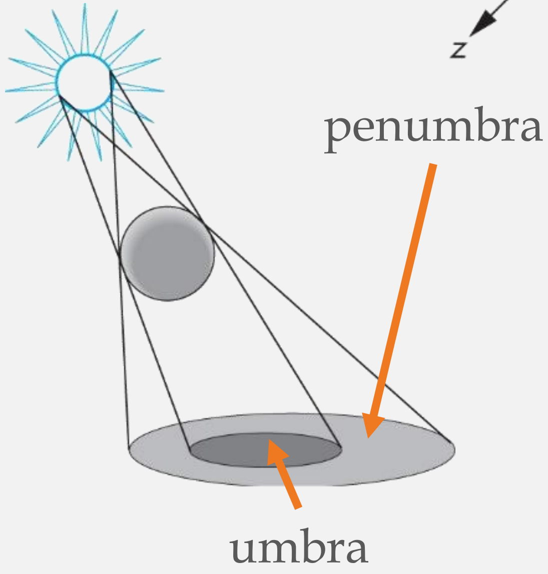
Light Sources

- A general light source can be characterized by a six-variable illumination function $I(x, y, z, \theta, \phi, \lambda)$
- General light sources are difficult to simulate because we must integrate light coming from all points on the light source.
- The intensity of the color source: $I = \begin{bmatrix} I_r \\ I_g \\ I_b \end{bmatrix}$



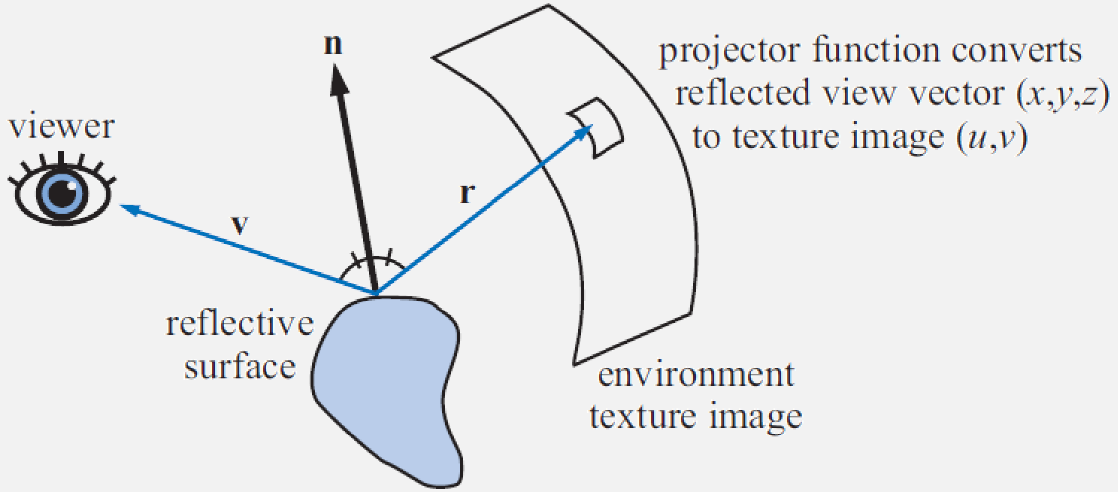


Adding the contribution from a source.



Ambient Illumination

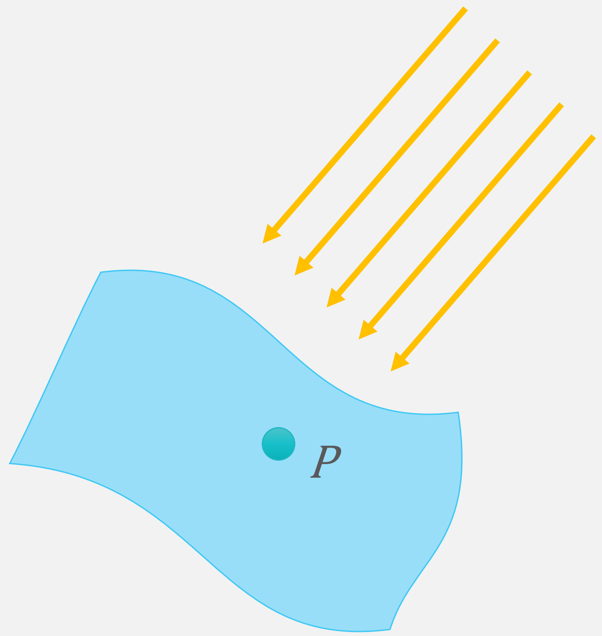
- Ambient light
  - Indirect lighting usually smooth and has low-frequency in their signal
  - Add constant color to fake the indirect lighting
- Environment light map
  - Lookup incoming radiance from reflection direction



[figure from *Real-time Rendering*, 3/e]

Directional Light

- Modeled by light direction and intensity I_L
- Assume the light source is far away (also called distant light)
  - Thus the light direction to each shading point is constant
- Often used to model sun light
  - No intensity attenuation

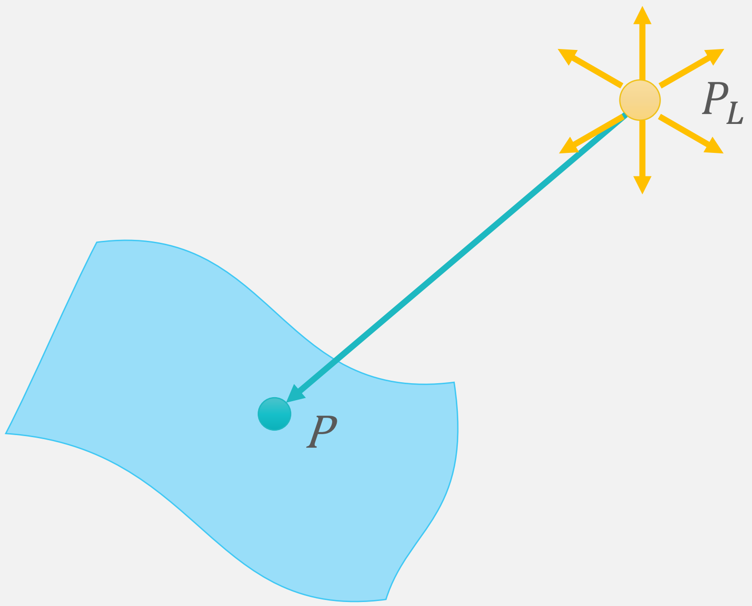


Point Light

- Defined by position P_L and intensity I_L
- Emits *constant* intensity in all directions
- Scenes rendered with only point sources tend to have high contrast
  - Can be solved by adding ambient light

$$E_L = I_L f_{dist}(r), \qquad r = |P_L - P|$$

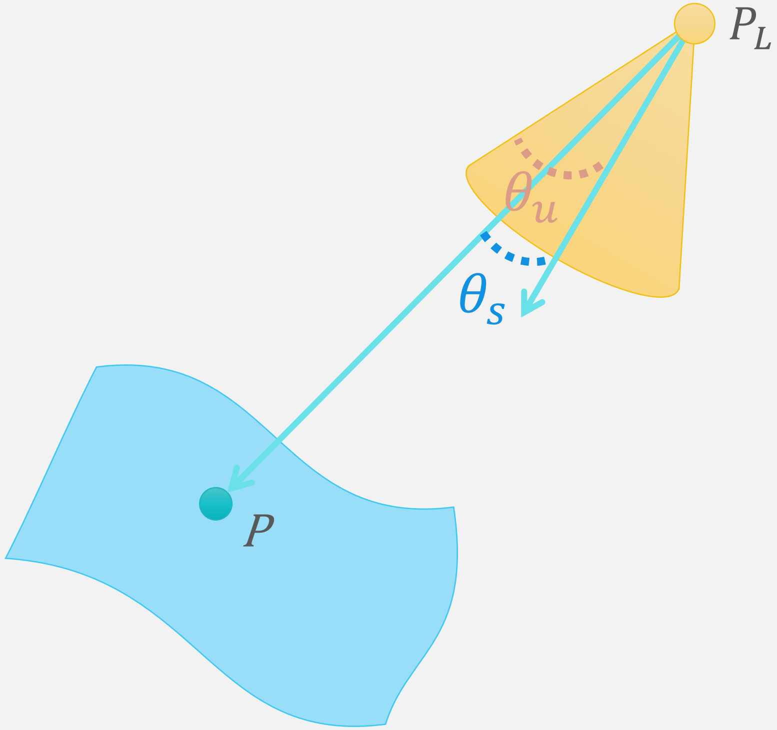
$$f_{dist}(r) = \frac{1}{r^2} \quad or \quad \frac{1}{k_C + k_L d + k_Q d^2}$$



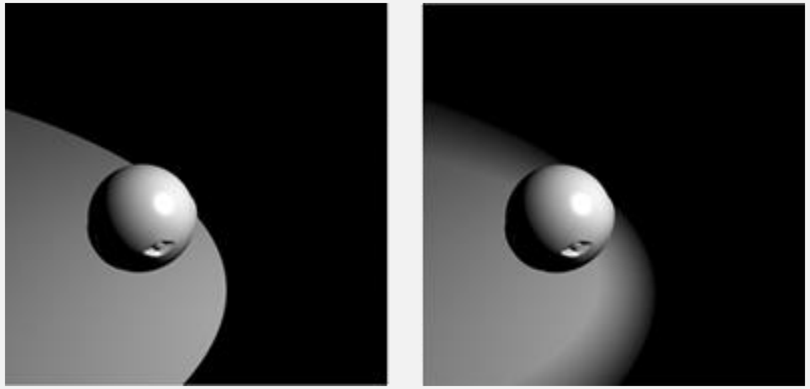
Spotlight

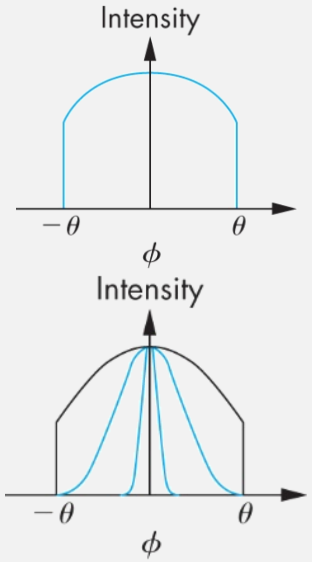
- Restrict light from ideal point source with a cone
  - Modeled by position P_L, direction, cone angle θ_s & intensity I_L

$$I_L(l) = \begin{cases} I_L(\cos\theta_s)^{\alpha} & if\ \theta_s \leq \theta_u \\ 0 & if\ \theta_s > \theta_u \end{cases}$$

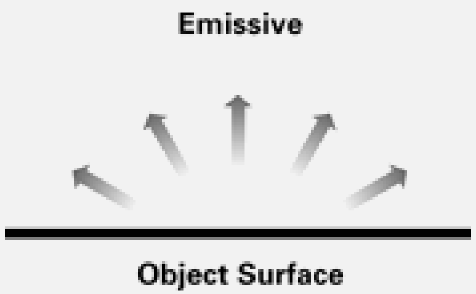


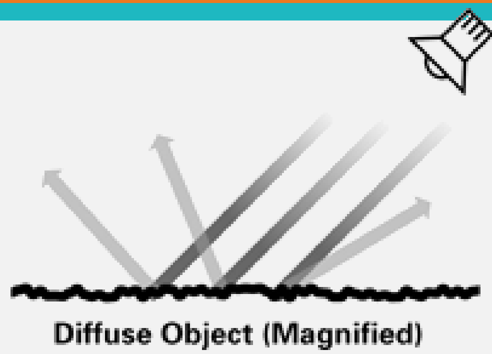
  - Attenuation of a spotlight



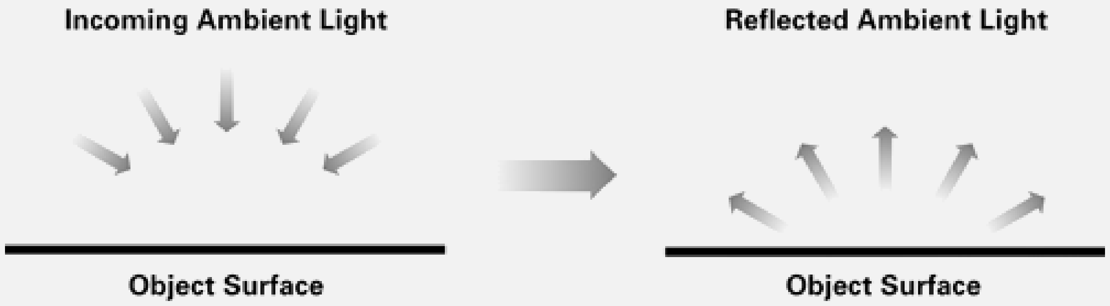


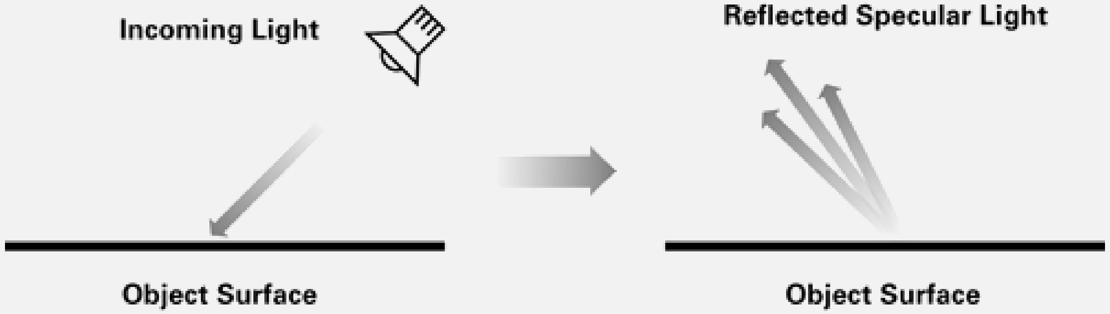
Basic Shading Model





surface color = emissive + ambient + diffuse + specular

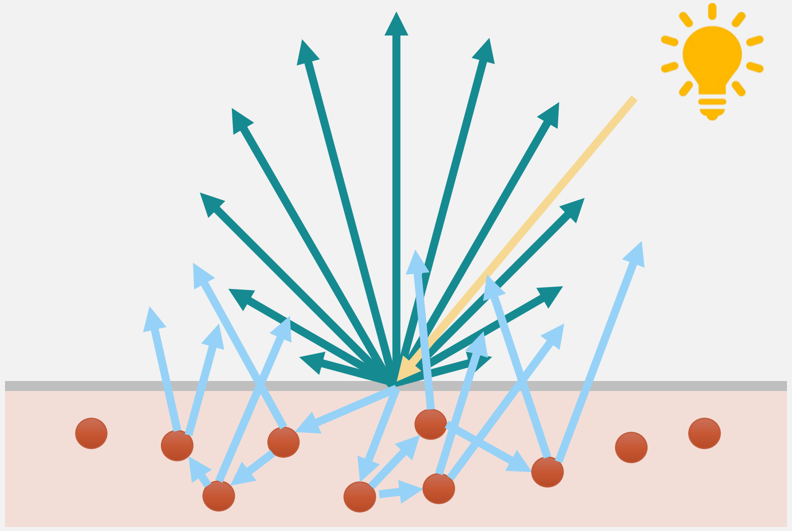




[figures from *"The Cg Tutorial"*]

Diffuse Material

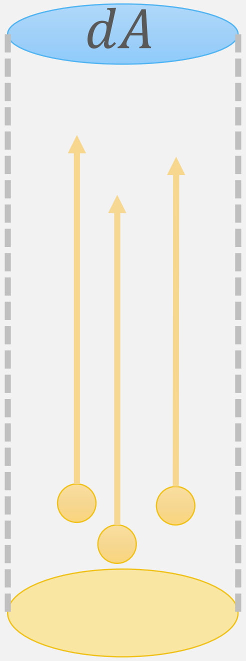
- Often described as having dull or matte appearance
- Albedo
  - the ratio of total amount of light reflected off of a surface relative to the light incident on the surface
  - ranges from 0 (no reflectance) to 1 (100% reflectance)

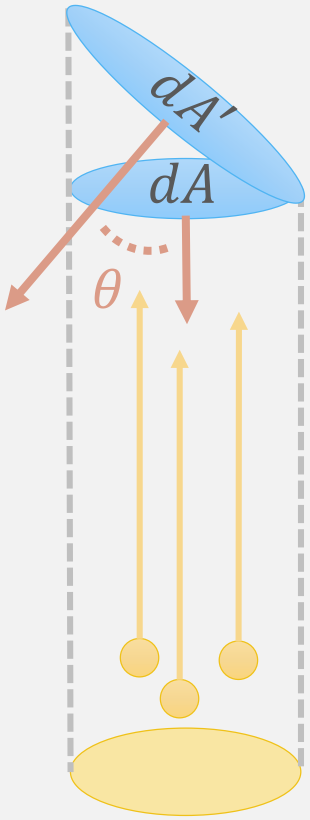


scattering particles

Lambert's Cosine Law

$$\boxed{E = \frac{d\Phi}{dA}}$$





$$dA = dA' \cos\theta$$

$$E_1 = \frac{d\Phi}{dA}$$

$$E_2 = \frac{d\Phi}{dA'} = \frac{\cos\theta \, d\Phi}{dA} = E_1 \cos\theta$$

Lambertian Diffuse Model

- Models an ideal diffuse reflector
  - Scatters incoming light equally in all directions
  - The surface color appears the same from any viewing direction with the same grazing angle

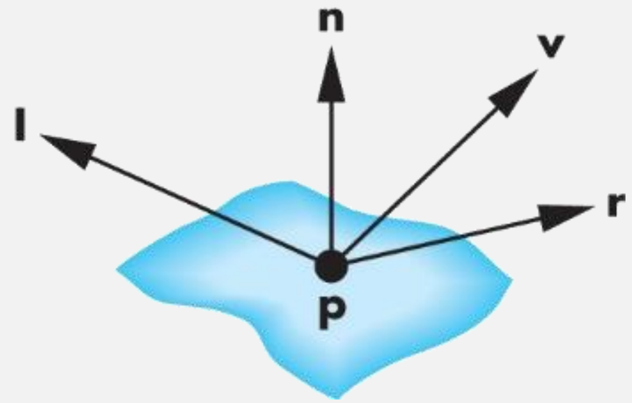
$$L_d = K_d \max(\vec{n} \cdot \vec{l}, 0)$$

const

Lambert's cosine law

Phong Reflection Model

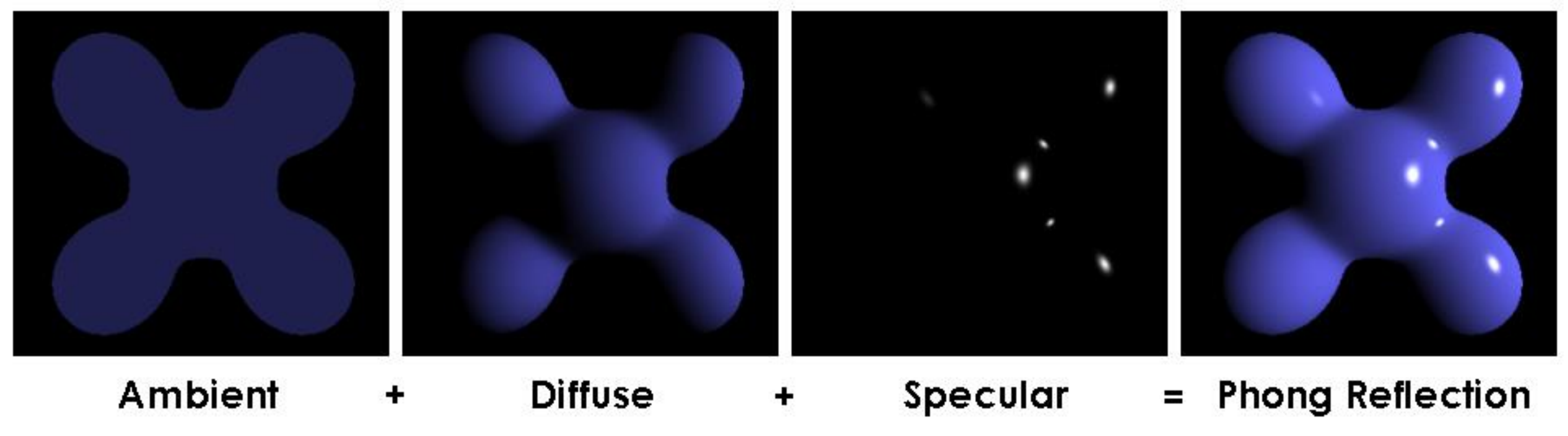
- An empirical reflection model
- Uses four vectors to calculate a color for a *shading point* $\boldsymbol{p}$
  - n : the normal at $\mathbf{p}$
  - v : the vector from $\mathbf{p}$ to the viewer
  - l : the vector from $\mathbf{p}$ to the light source
  - r : the perfect mirror reflection vector of $\mathbf{l}$
- Includes three components:
  - Ambient
  - Diffuse: Lambertian reflection
  - Specular



Phong Reflection Model (Cont.)

reflected light = ambient light + diffuse component + specular component

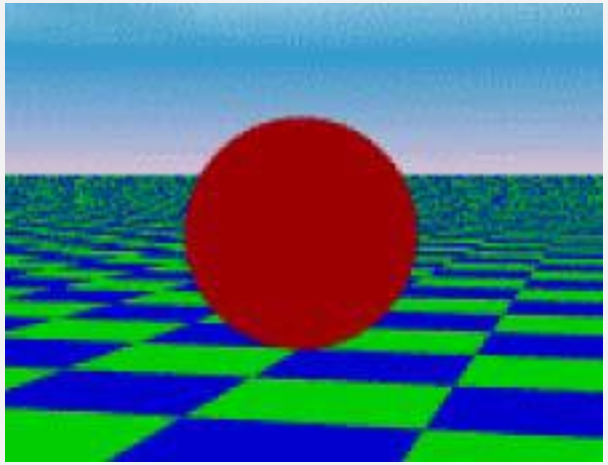
$$L_o(v) = f_{amb}L_{amb} + \sum_{k=1}^{n}\left(f_{diff}(l_k, v) + f_{spec}(l_k, v)\right) \otimes L_{i,k}$$



Ambient

- The intensity of ambient light (I_a) is the same at every point on the surface
  - Some of L_a is absorbed and some is reflected according to the reflection coefficient

$$I_{ra} = k_{ra} \cdot L_{ra}$$
$$I_{ga} = k_{ga} \cdot L_{ga} \qquad \Rightarrow \mathbf{I_a} = \mathbf{k_a} \otimes \mathbf{L_a}$$
$$I_{ba} = k_{ba} \cdot L_{ba}$$



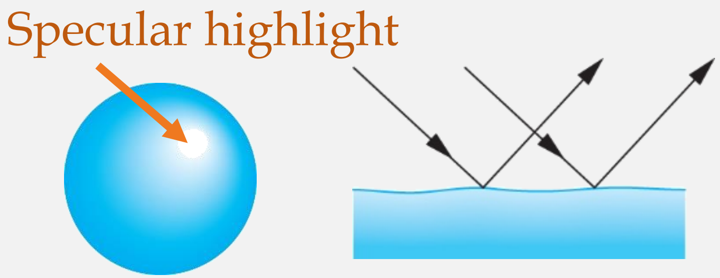
[fig w, w/o amb shading]

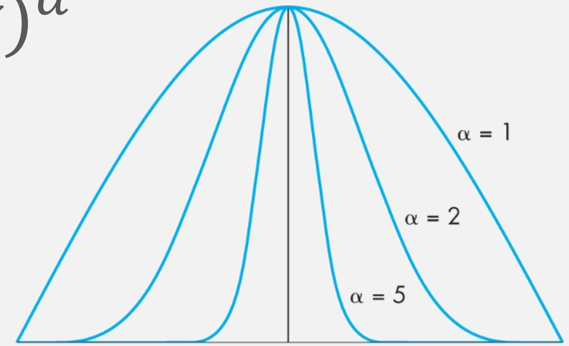
Specular Reflection

- Most surfaces are neither ideal diffusers nor perfectly specular (ideal reflectors)
- Incoming light is reflected in directions concentrated close to the perfect reflection direction
- The amount of light the viewer sees depends on the angle between **r** and **v** :

$$I_s = k_s L_s \cos^{\boxed{\alpha}} \phi = k_s L_s (\mathbf{r} \cdot \mathbf{v})^{\alpha}$$

shininess coefficient



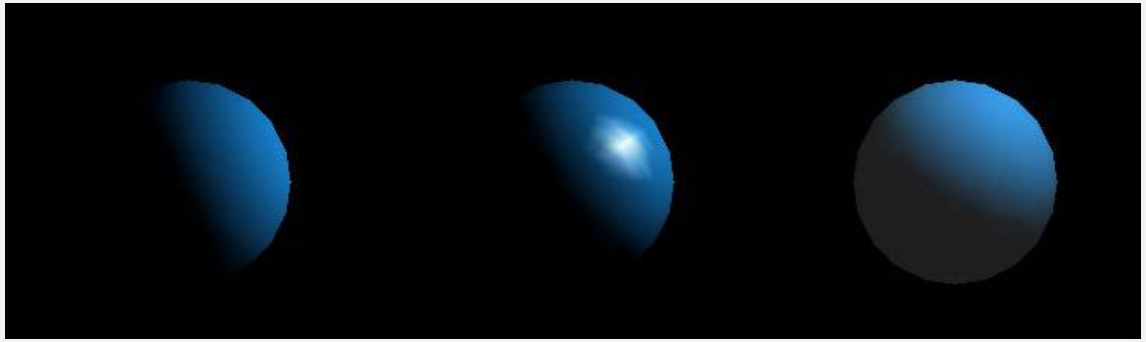


Phong Reflection Model (Cont.)

- For **each light source** and **each primary component**:

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

- Coefficients:
  - 9 coefficients for each point light source
  - 9 absorption coefficients (surface color)
  - 1 shininess coefficient



Diffuse

Diffuse + Specular

Diffuse + Ambient

Blinn-Phong Reflection Model

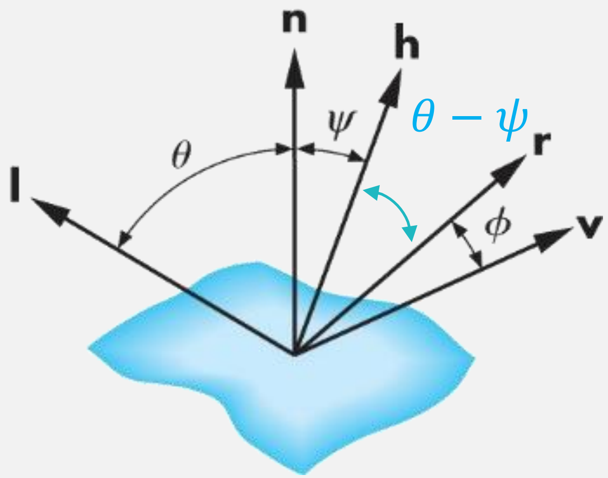
- Problem:
  - In the specular component of the Phong model, it requires the calculation of a new reflection vector **r** and view vector **v** for each vertex

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

- Blinn suggested an approximation using the halfway vector that is more efficient

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

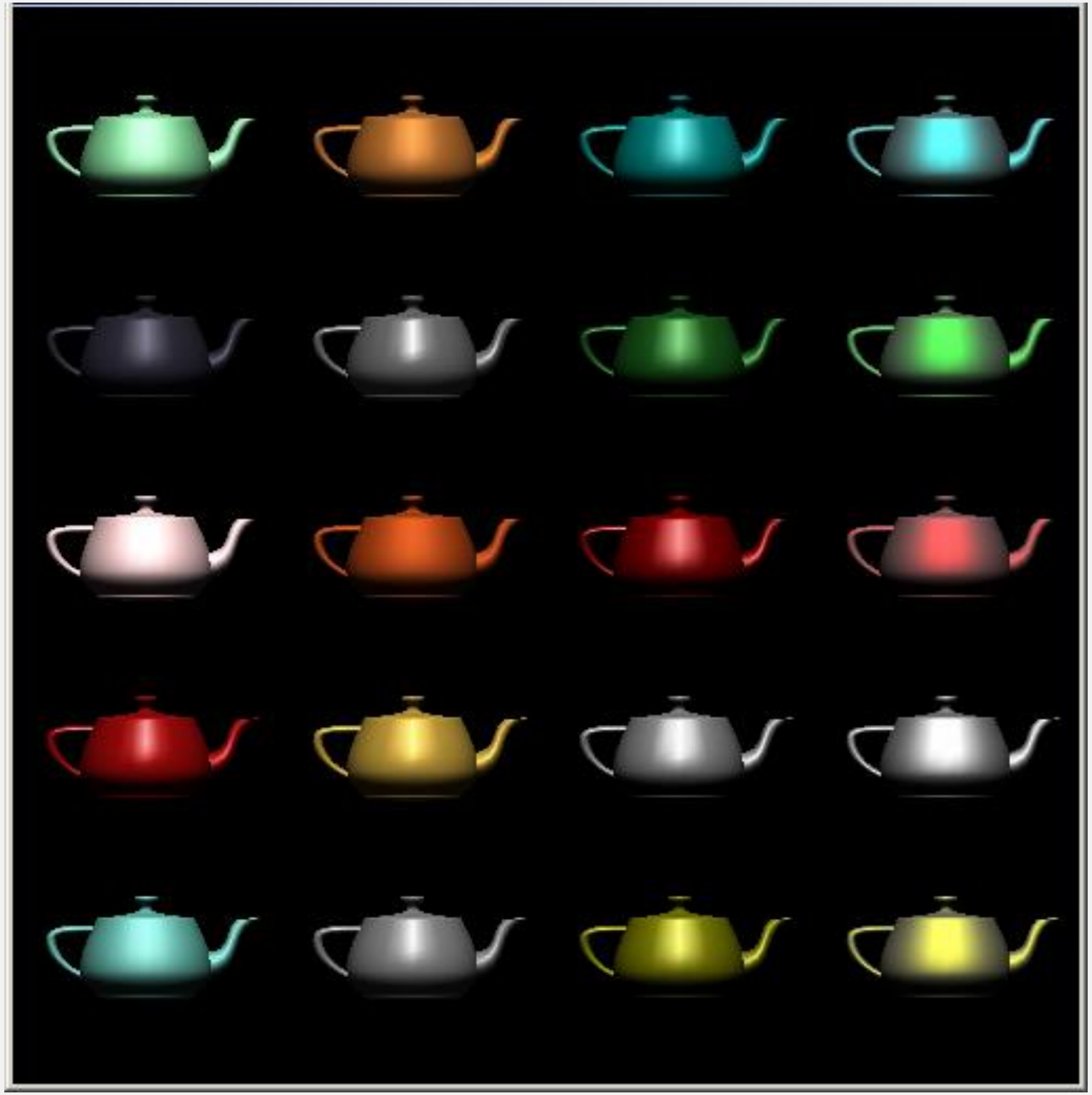
  - Replace $(\mathbf{r} \cdot \mathbf{v})^{\alpha}$ by $(\mathbf{n} \cdot \mathbf{h})^{\alpha'}$



if vectors are coplanar:

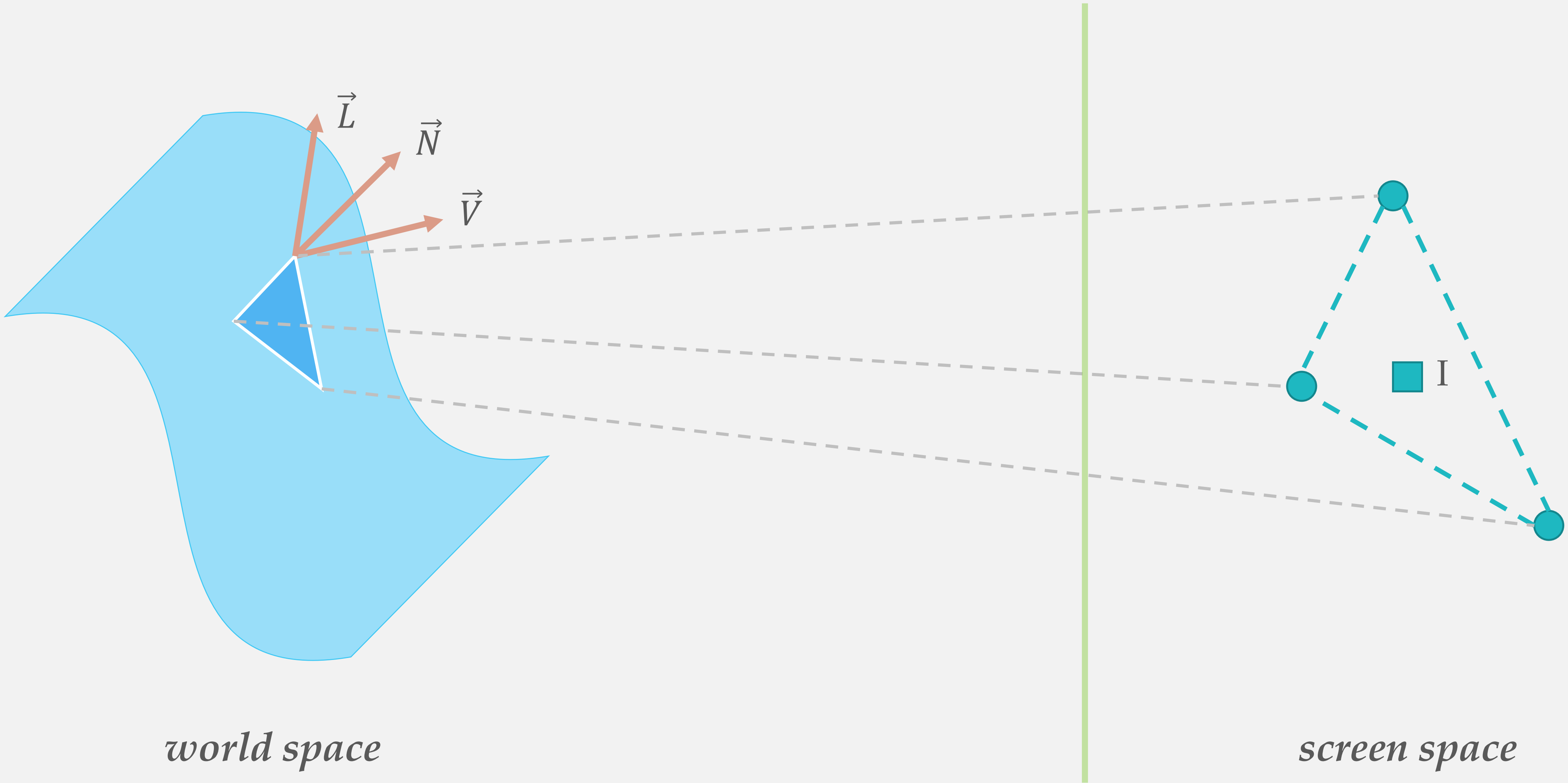
$$\theta + \psi = \theta - \psi + \phi$$
$$\Rightarrow 2\psi = \phi$$

Examples (Modified Phong Model)



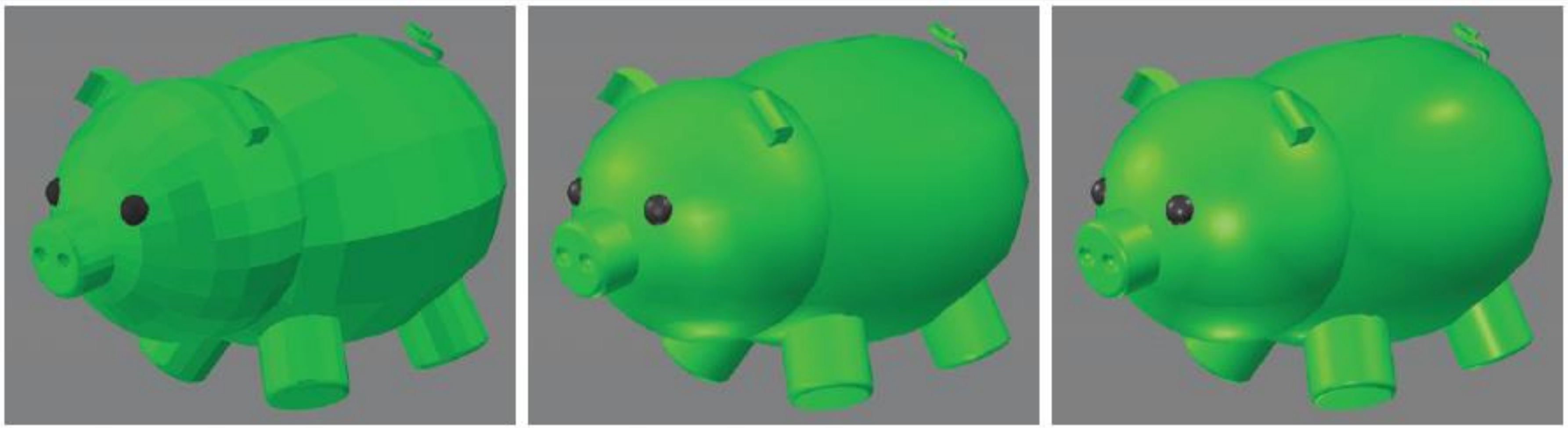
Teapot with different parameters

Local Reflection Model & Shading Algorithms



Shading Algorithms

- Interpolation techniques to fill color within a polygon.
  - Flat shading (per-primitive)
  - Gouraud shading (per-vertex)
  - Phong shading (per-pixel)

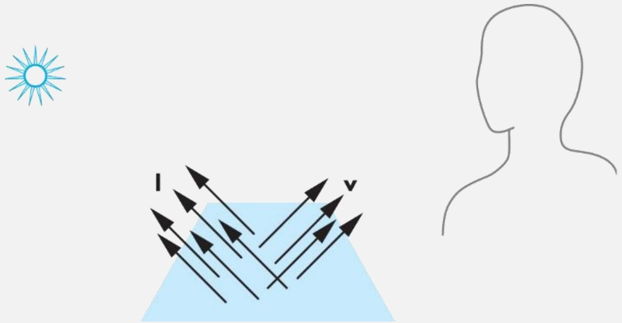


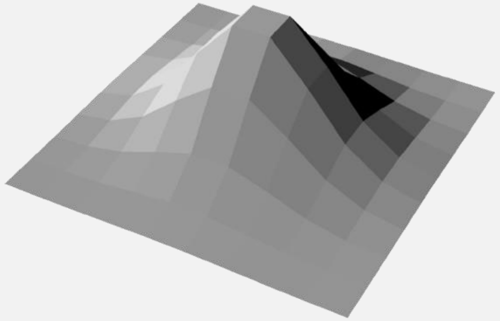
flat shading　　Gouraud shading　　Phong shading

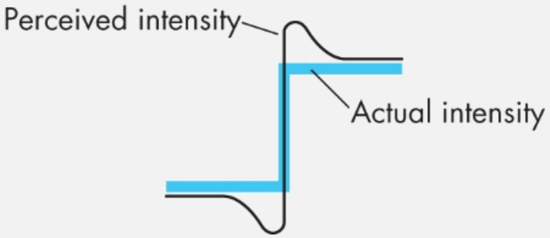
[*Real-time Rendering*, 3/e]

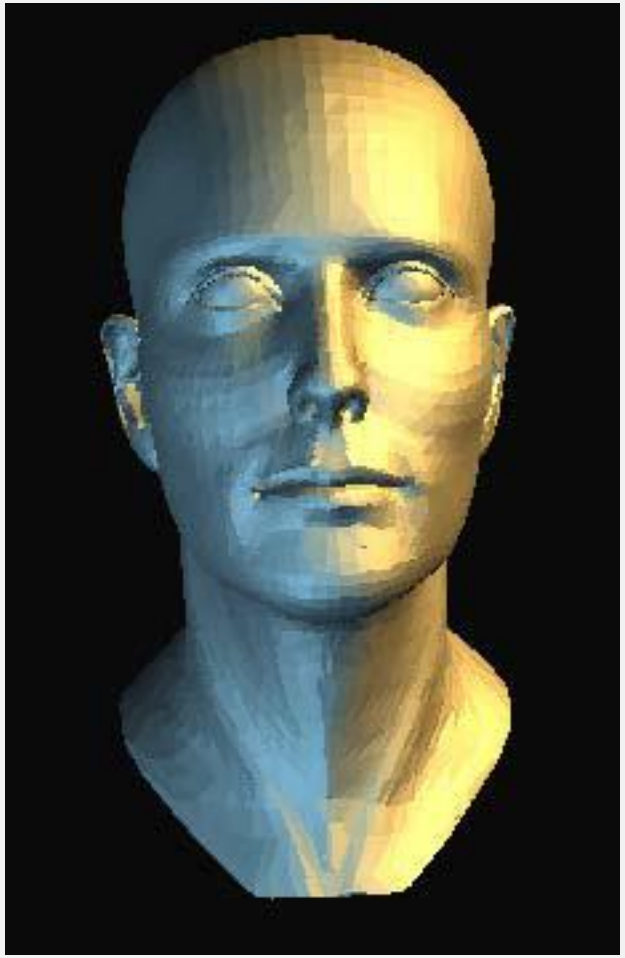
Flat/Constant Shading

- Flat or constant shading
  - Assume **l**, **n**, **v** are constant for a polygon.
- Shading calculation: only once for each polygon









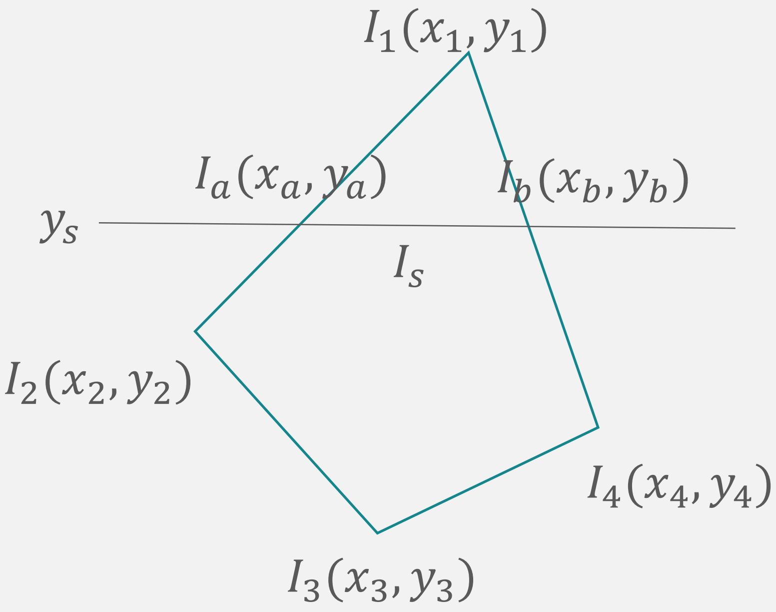
Smooth/Gouraud Shading

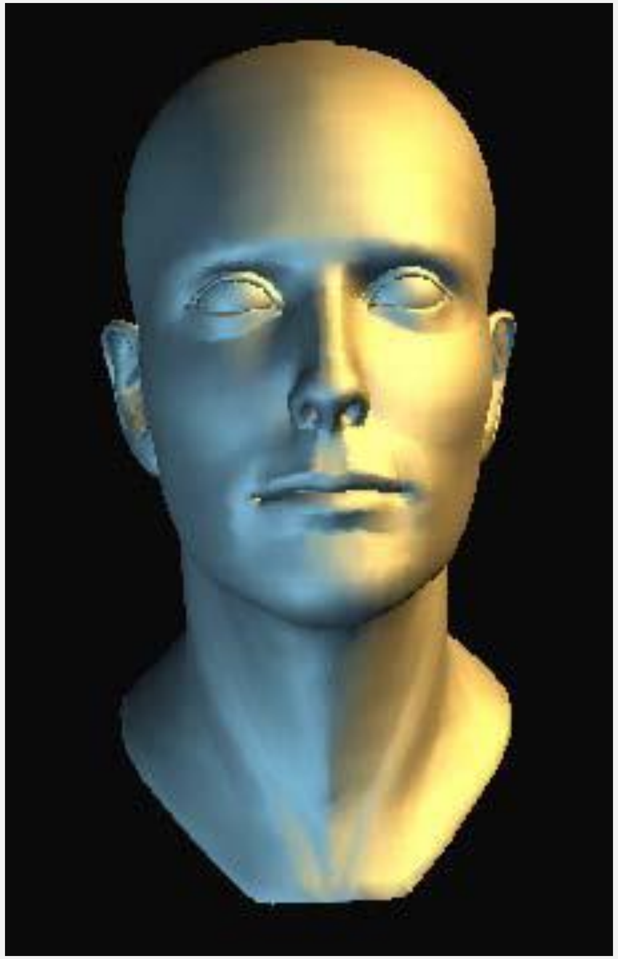
- Find average normal at each vertex
- Apply lighting model at each vertex
- Interpolate vertex shades across each polygon

$$I_a = \frac{1}{y_1 - y_2}\left[I_1(y_s - y_2) + I_2(y_1 - y_s)\right]$$

$$I_b = \frac{1}{y_1 - y_4}\left[I_1(y_s - y_4) + I_4(y_1 - y_s)\right]$$

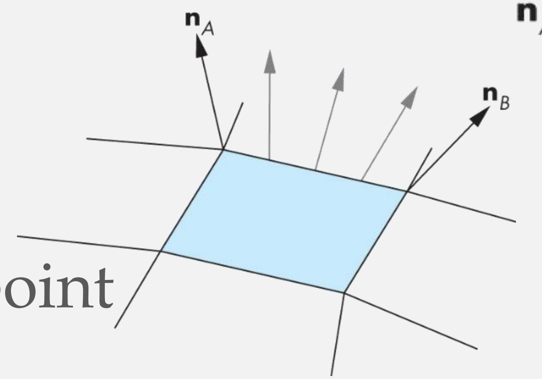
$$I_s = \frac{1}{x_a - x_b}\left[I_a(x_b - x_s) + I_b(x_s - x_a)\right]$$

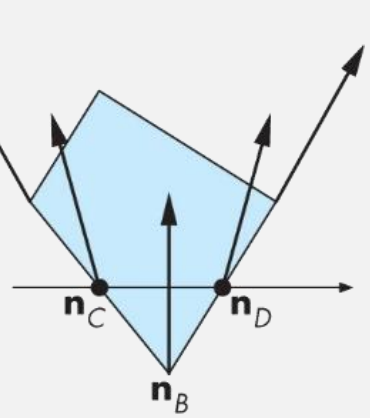




Phong Shading (Per-fragment Shading)

- Find vertex normals
- Interpolate vertex normal across edges
- Interpolate vertex normal of each interior point
- Calculate shade for each point





$$n_C(\alpha) = (1-\alpha)n_A + \alpha n_B$$

$$n_{in}(\alpha,\beta) = (1-\beta)n_C + \beta n_D$$