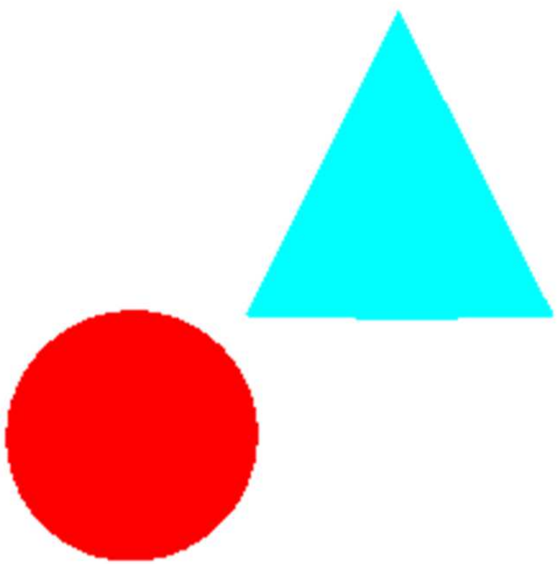
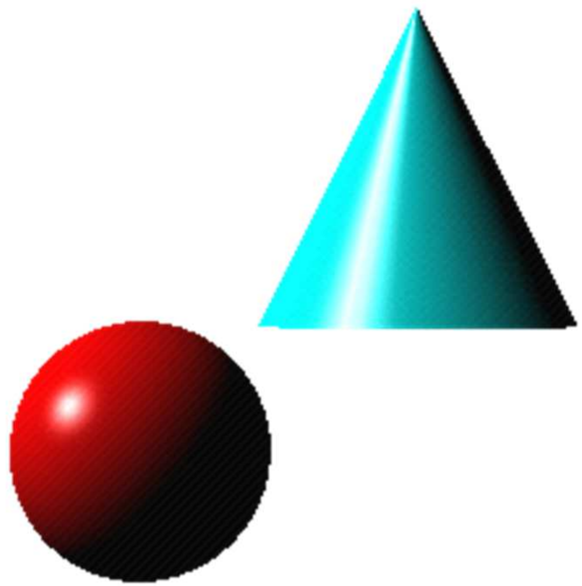
Lighting

Lighting principles

- Simulates how objects reflect light
  - Material composition of object
  - Light's color and position





Models

Color model

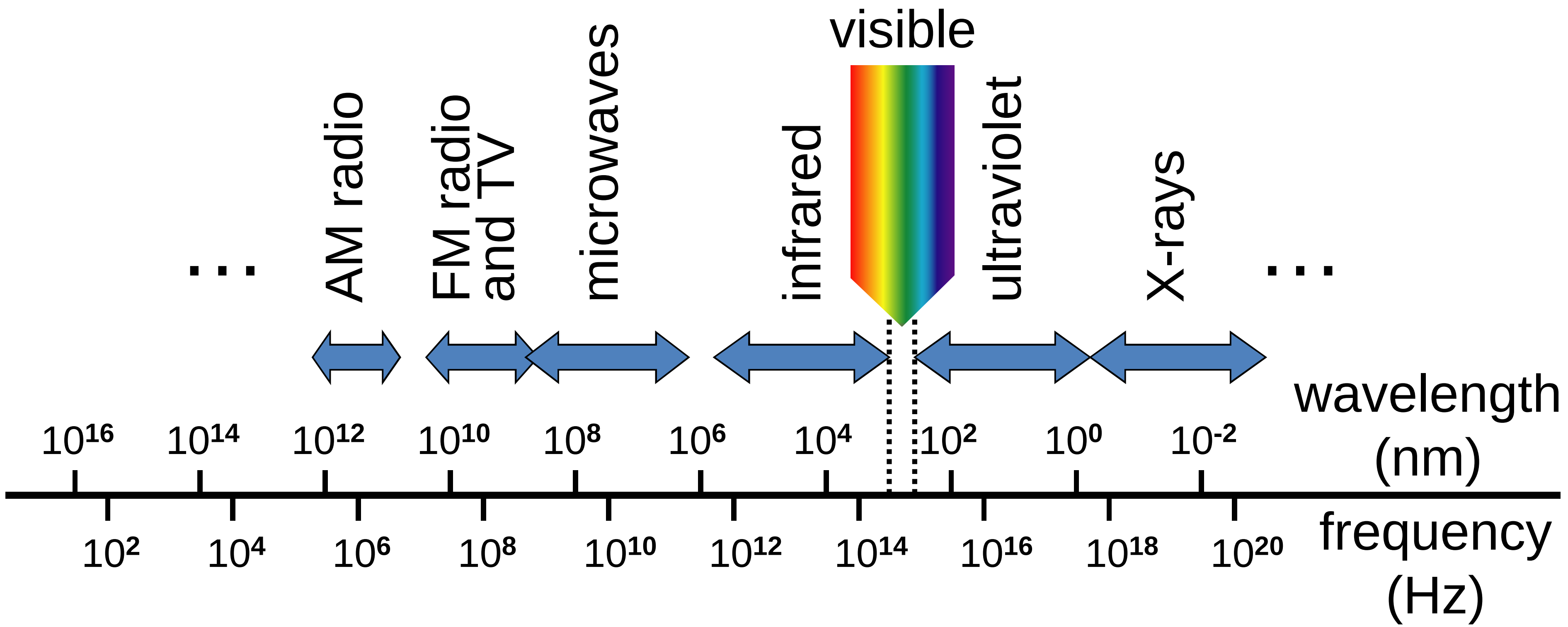
Light sources

Reflection model

Shading model

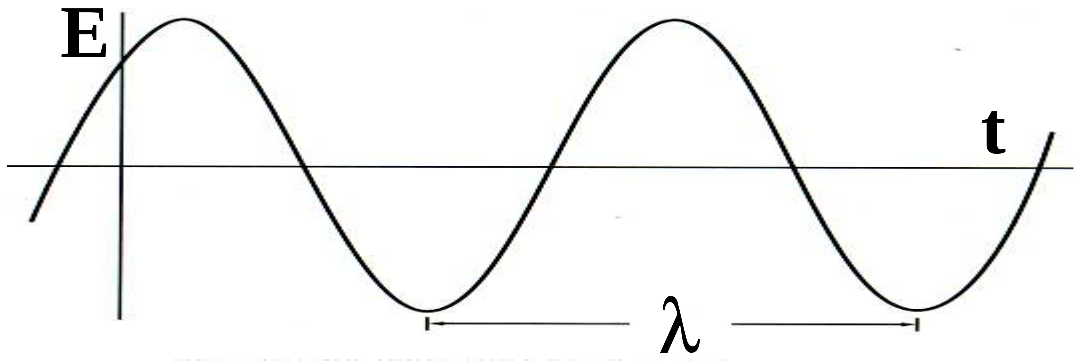
What is Light?

- frequency band of electromagnetic spectrum
- red border: 380 THz ≈ 780 nm
- violet border: 780 THz ≈ 380 nm

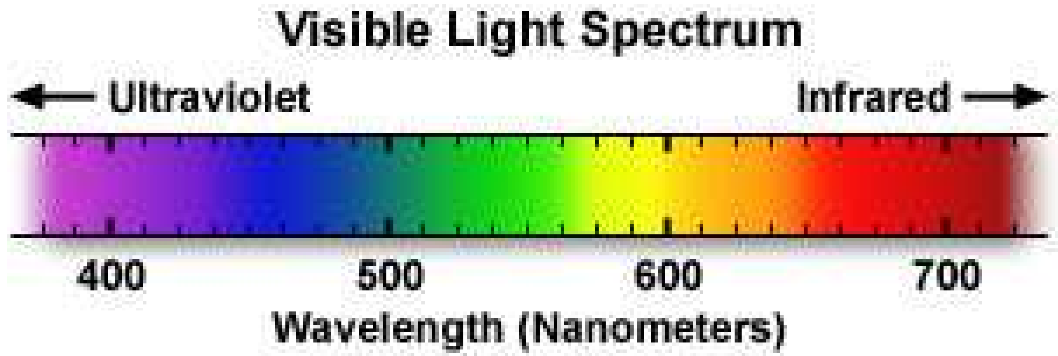


Light - An Electromagnetic Wave

- light is electromagnetic energy
- monochrome light can be described either by frequency f or wavelength λ
- $c = \lambda f$ (c = speed of light)



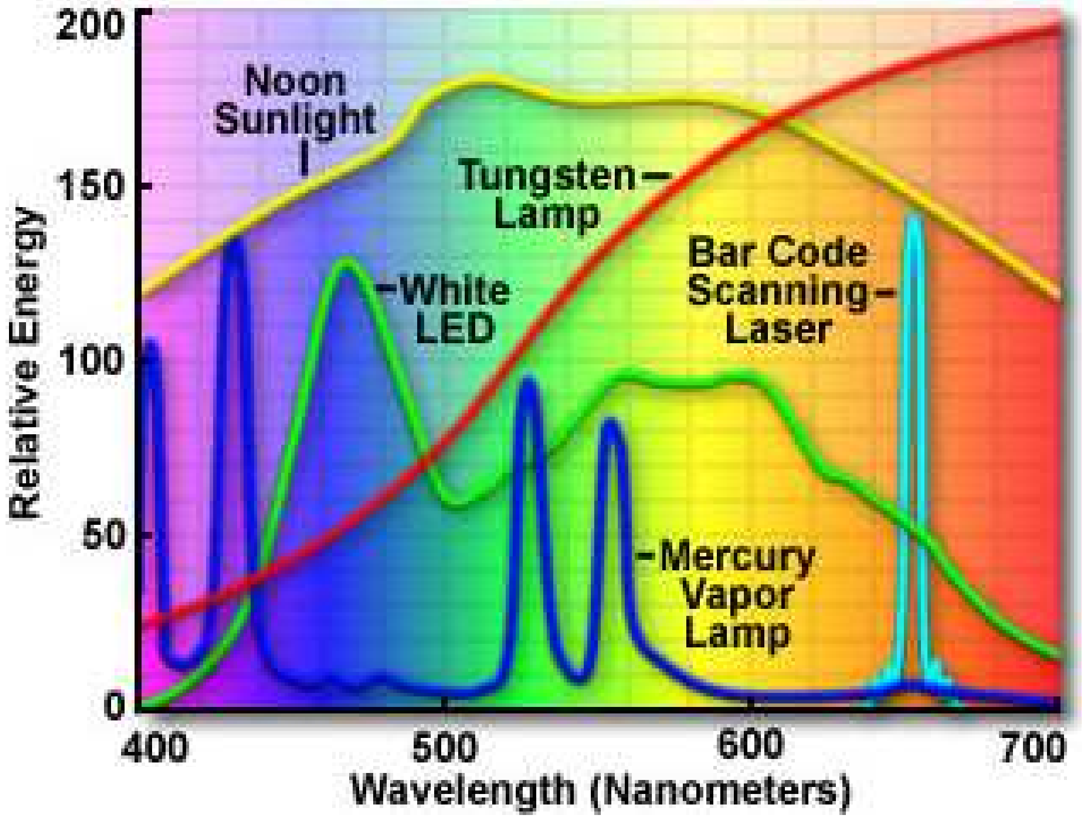
- shorter wavelength equals higher frequency



- red ≈ 700 nm
- violet ≈ 400 nm

Light – Spectrum

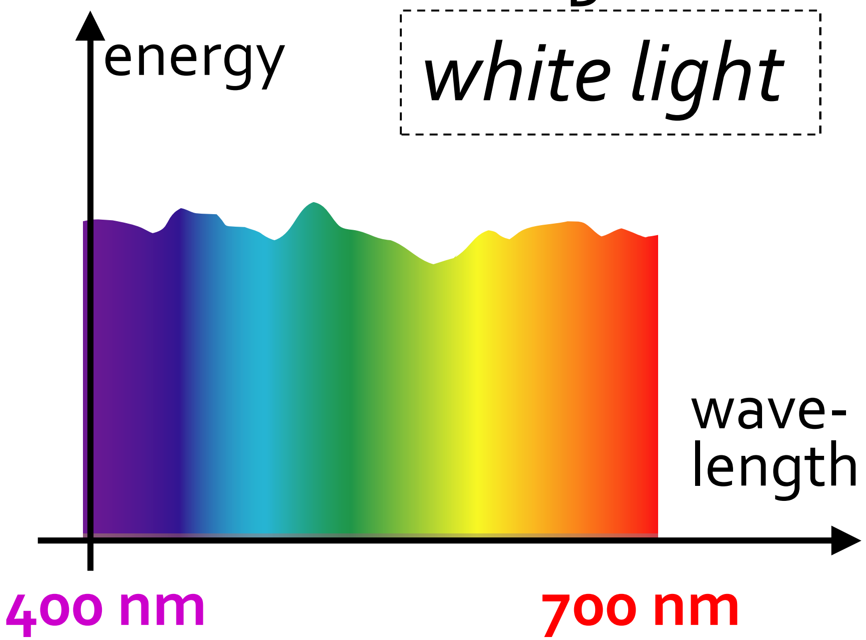
- normally, light mixture of different frequencies
- distribution of wavelength intensities is called ***spectrum***

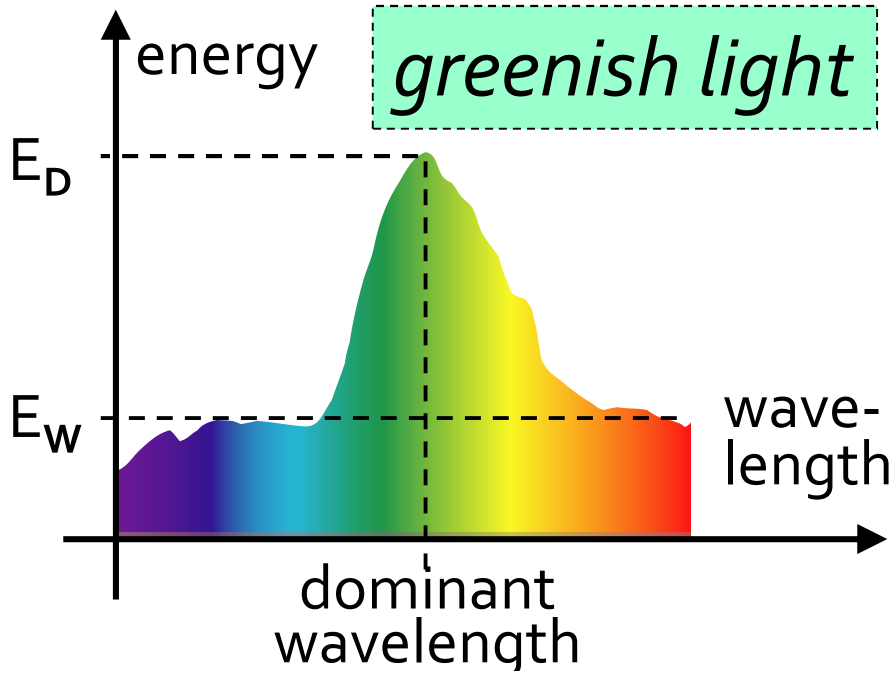


Dominant Wavelength | Frequency

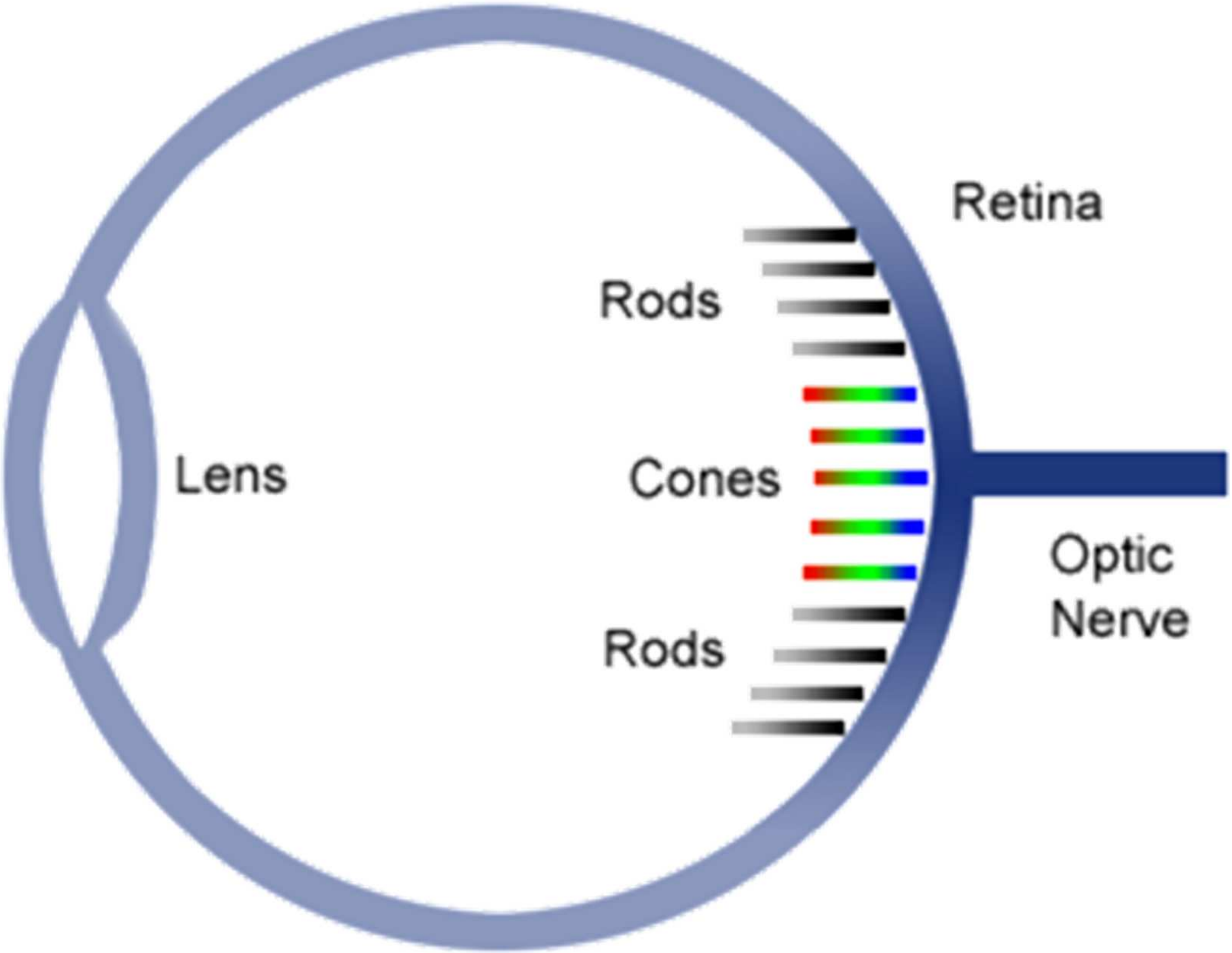
- dominant wavelength | frequency (hue, color)
- brightness (area under the curve)
- purity $\dfrac{E_D - E_W}{E_D}$

E_D...dominant energy density
E_W...white light energy density



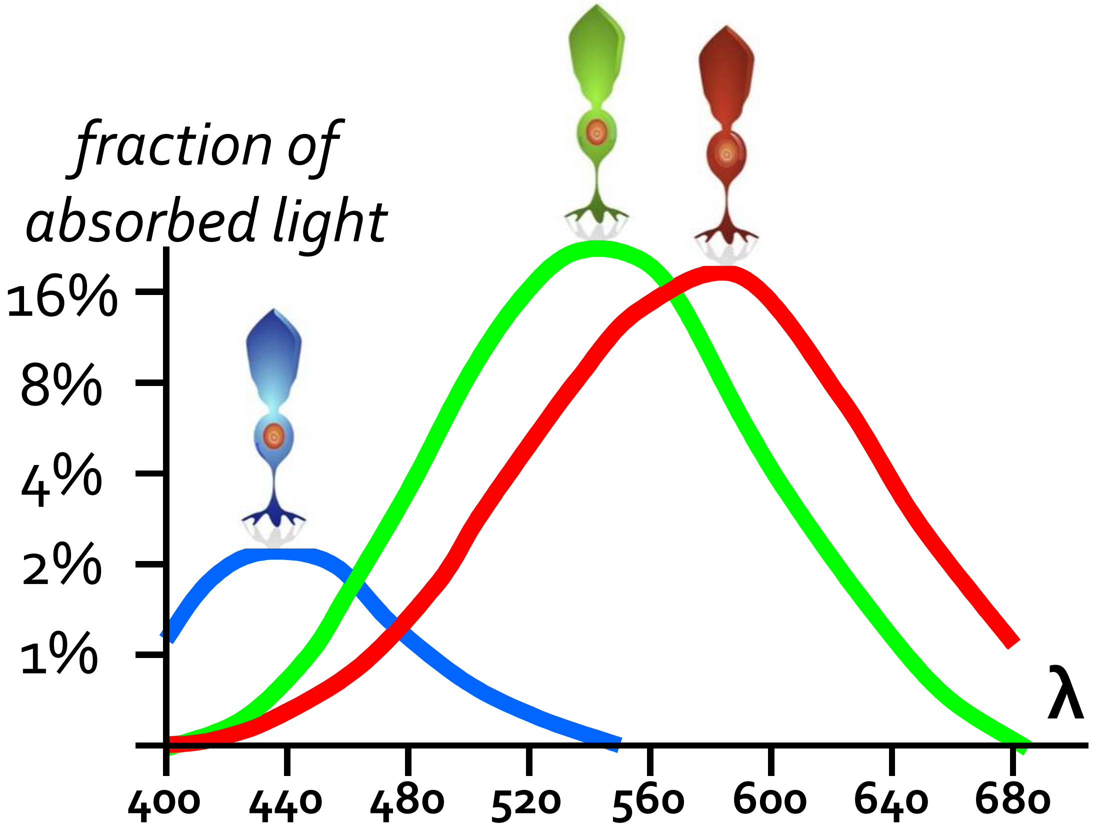


The Human Eye



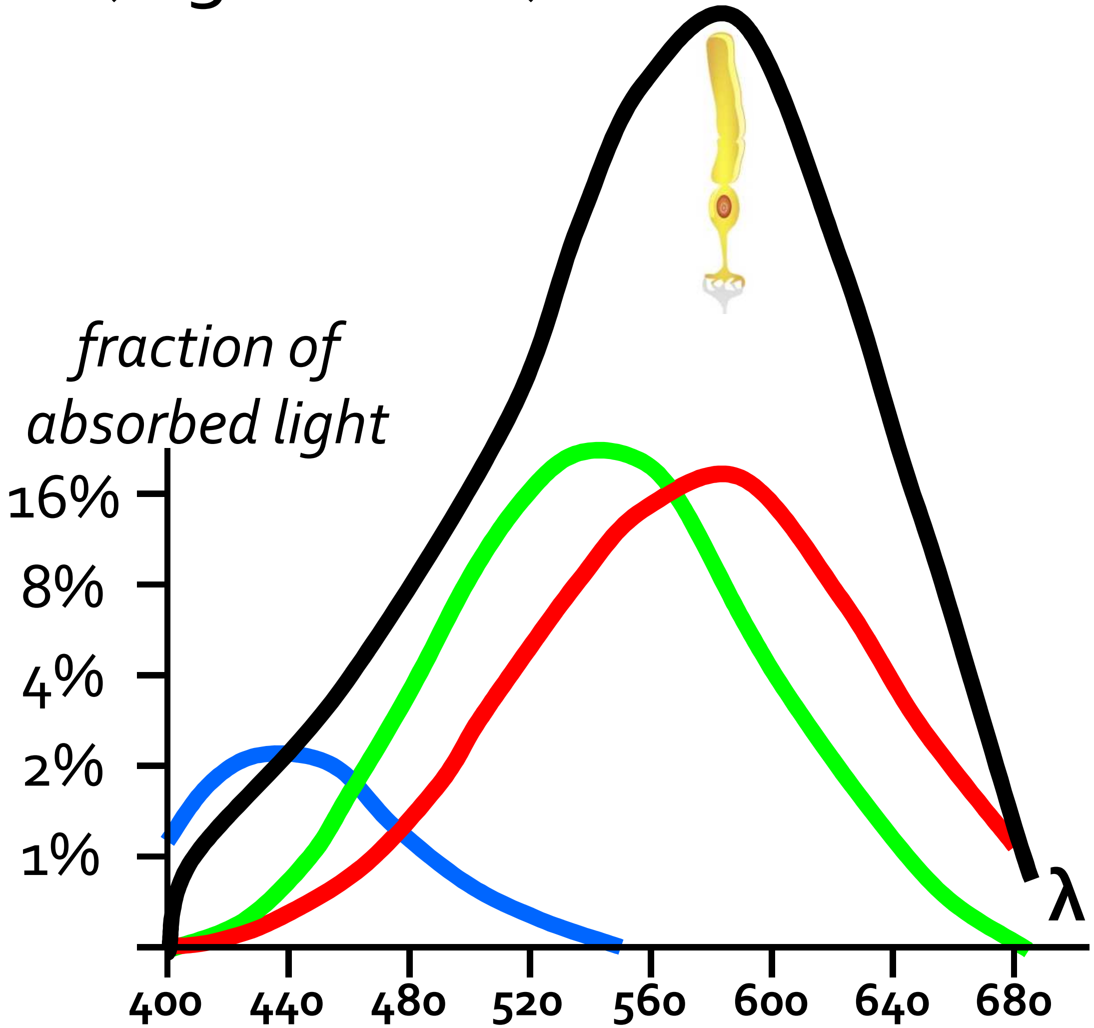
The Human Eye – Cones

- 3 types
- different wavelength sensitivities:
  - red
  - green
  - blue



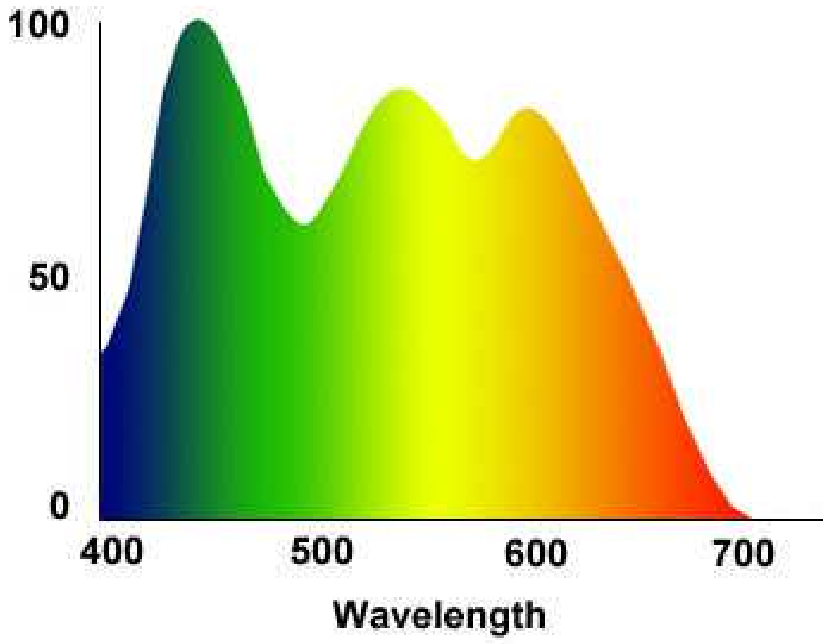
The Human Eye – Rods

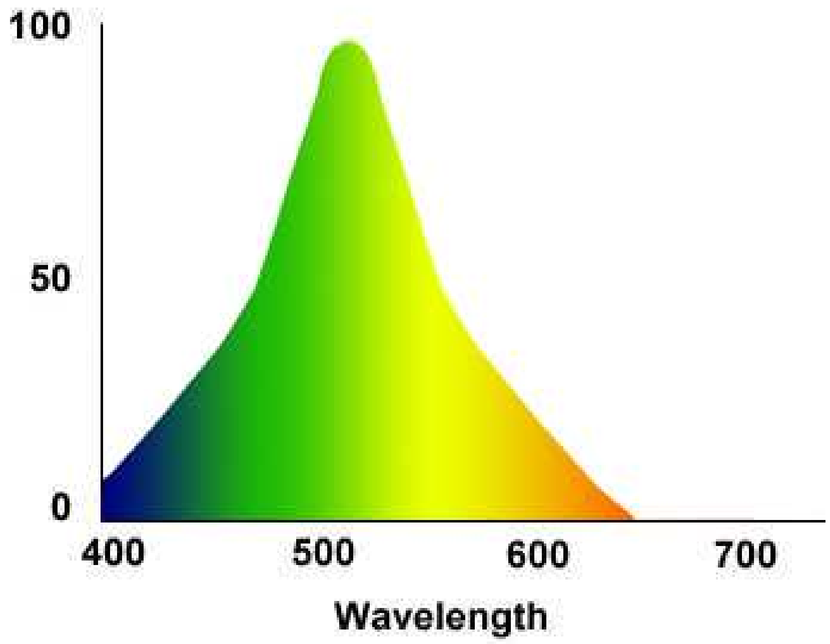
- For less intense light (night vision)
- Peripheral vision



The Human Eye - Adaptation

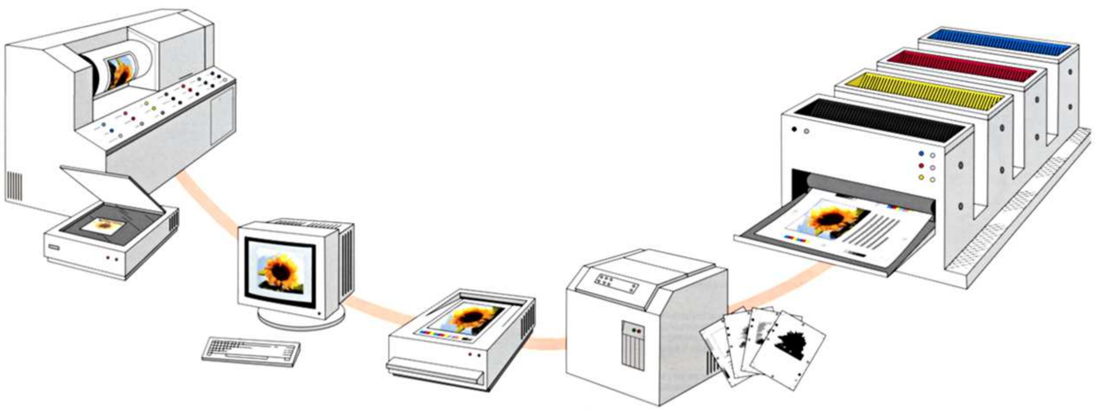
- daylight-adapted human eye
- dark-adapted human eye



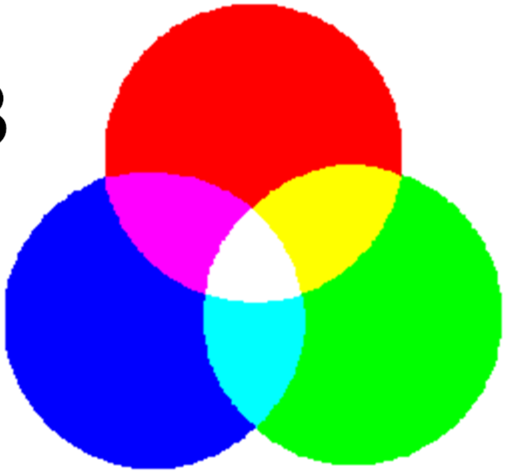


Color Models (Spaces)

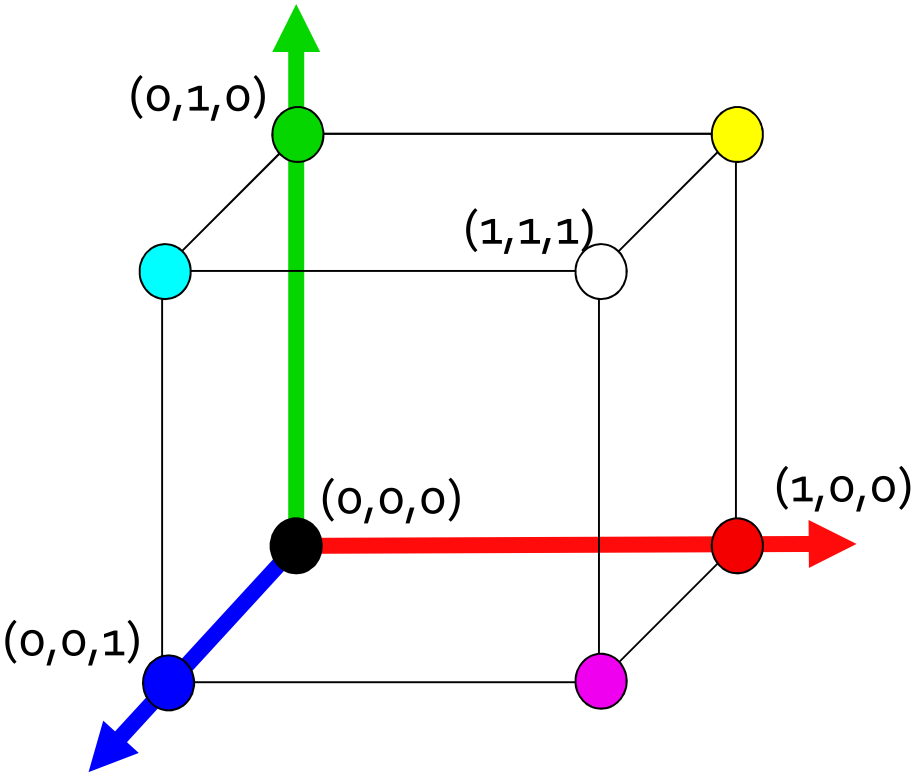
- **Color Metric Spaces** (CIE XYZ, L*a*b*)
  - Measure absolute values and differences
- **Device Color Spaces** (RGB, CMY, CMYK)
  - Device specific
- **Color Ordering Spaces** (HSV, HLS)
  - Find colors according to some criterion



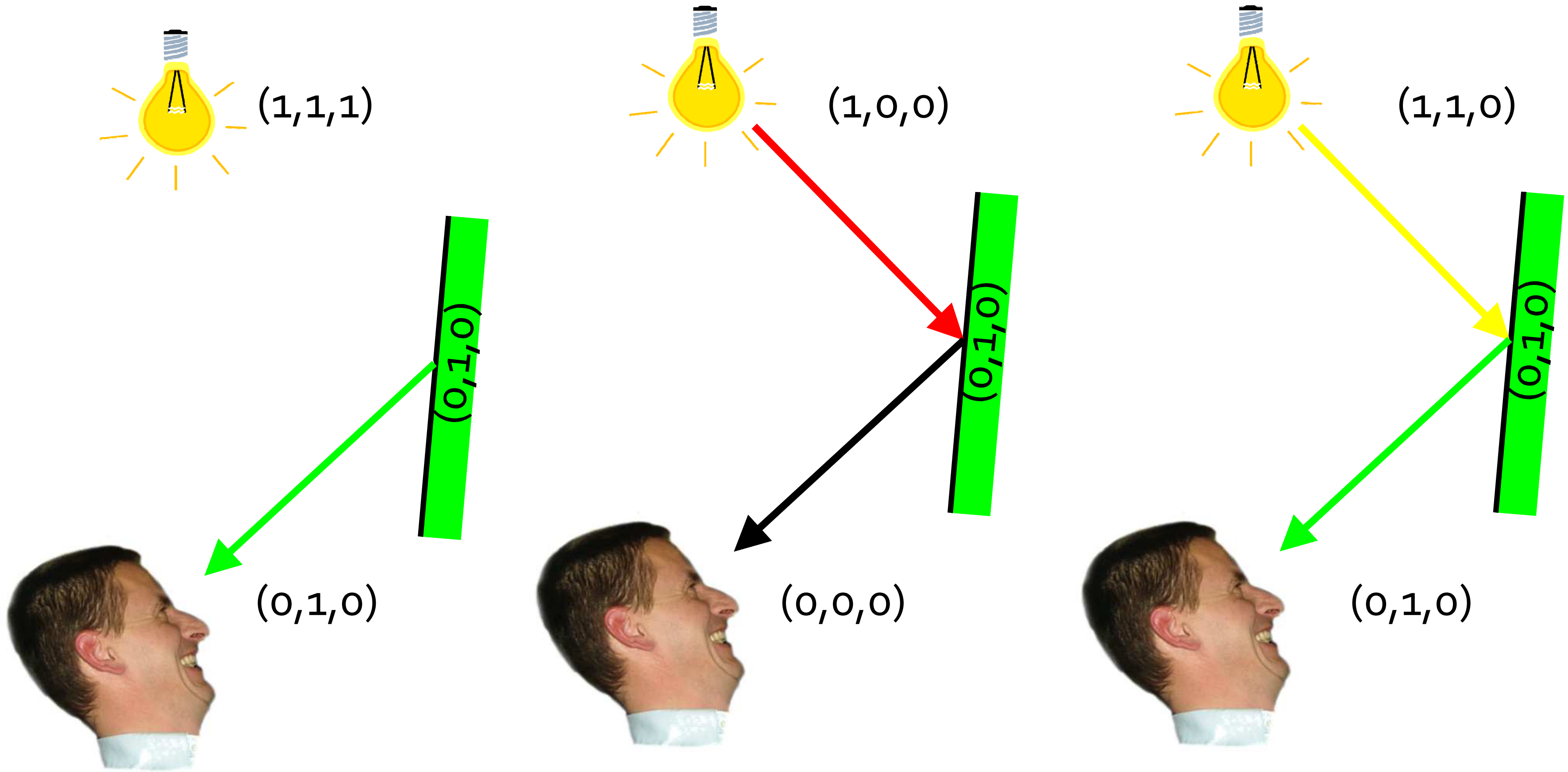
Color model – RGB



- C = RGB
- Additive Color Model
- 0 <= RGB <= 1
- Channels independent
  - Calculations / channel



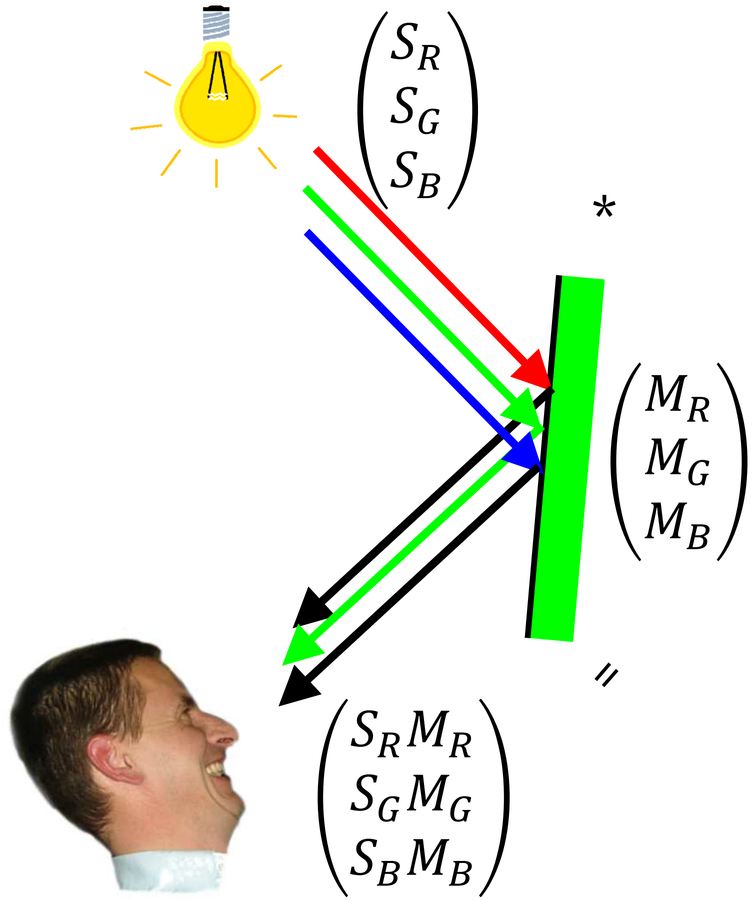
Color Multiplication



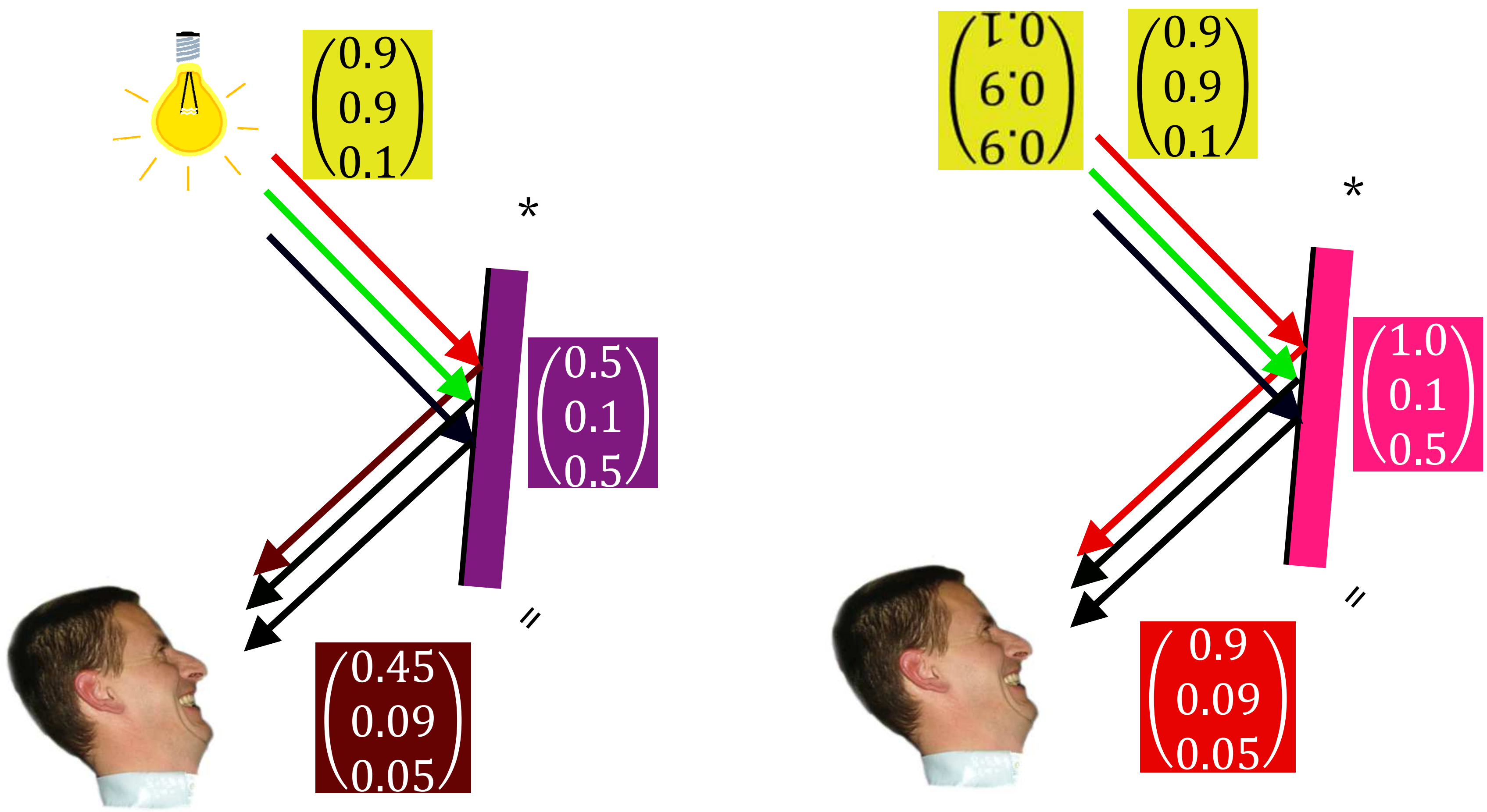
Color Multiplication - “*” Operator

- Color channels have independent light interaction
- S…light color
- M…material color

$$\begin{pmatrix} S_R \\ S_G \\ S_B \end{pmatrix} * \begin{pmatrix} M_R \\ M_G \\ M_B \end{pmatrix} := \begin{pmatrix} S_R M_R \\ S_G M_G \\ S_B M_B \end{pmatrix}$$



Color Multiplication - "*" Operator



Models

Color model

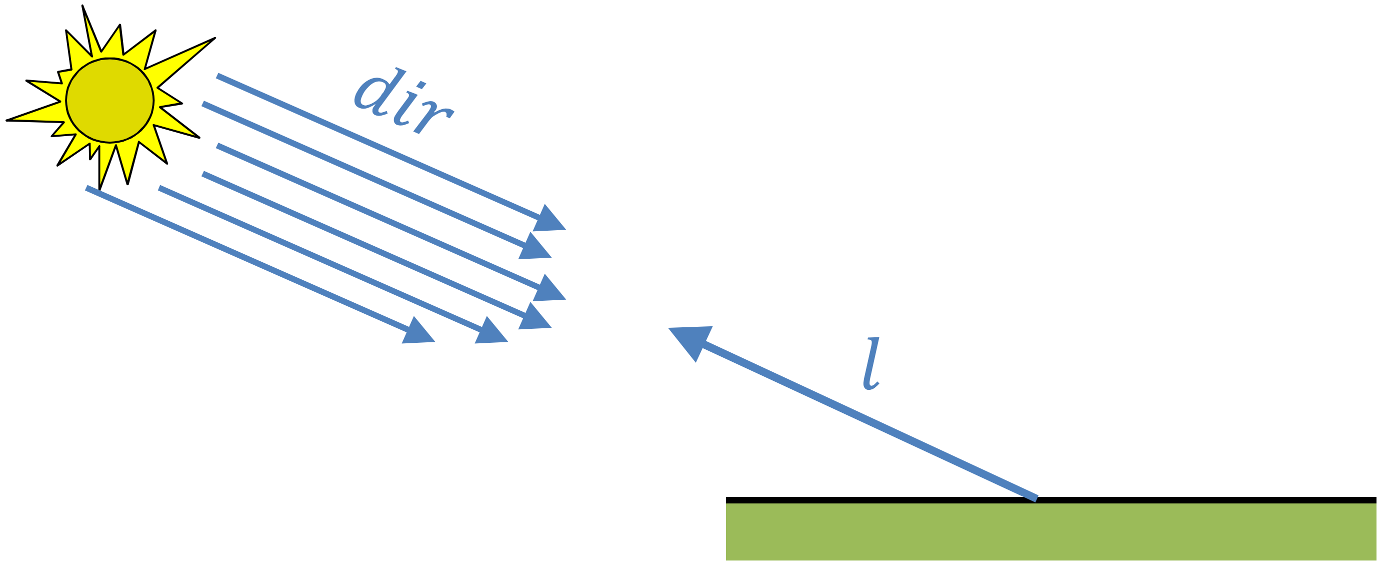
Light sources

Reflection model

Shading model

Directional Light

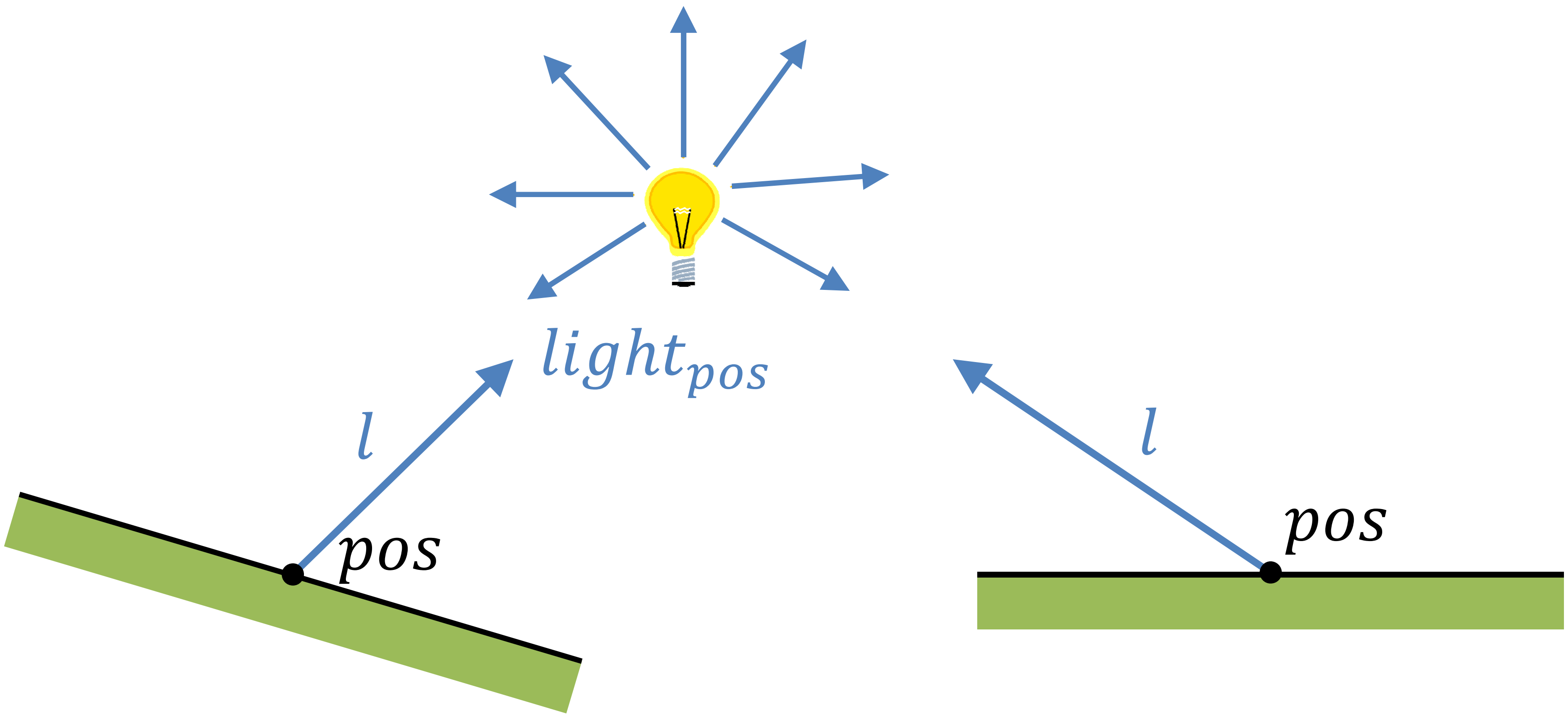
- Light source is infinitely far away
- Light rays are parallel, like sun
- l … direction to the light $= -dir$



Point Light

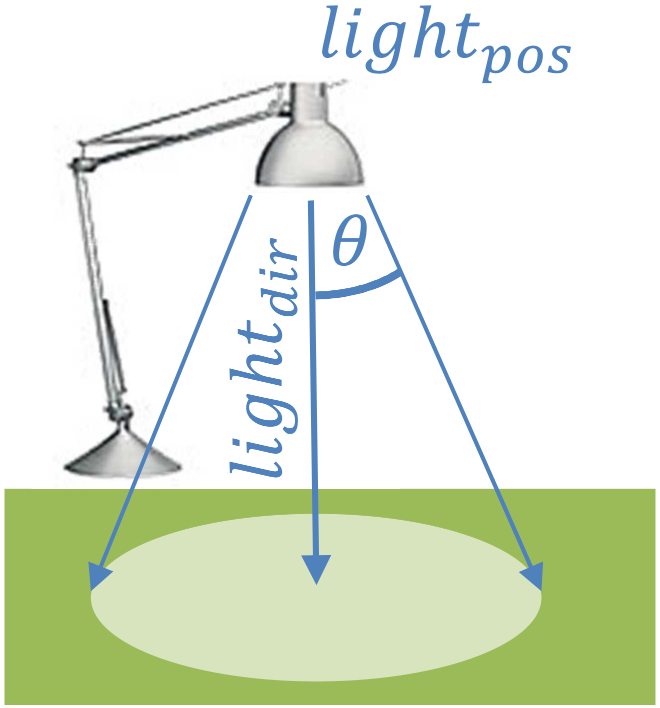
- Has a certain position in space

$$l = \text{normalize}(light_{pos} - pos)$$



Spot Light

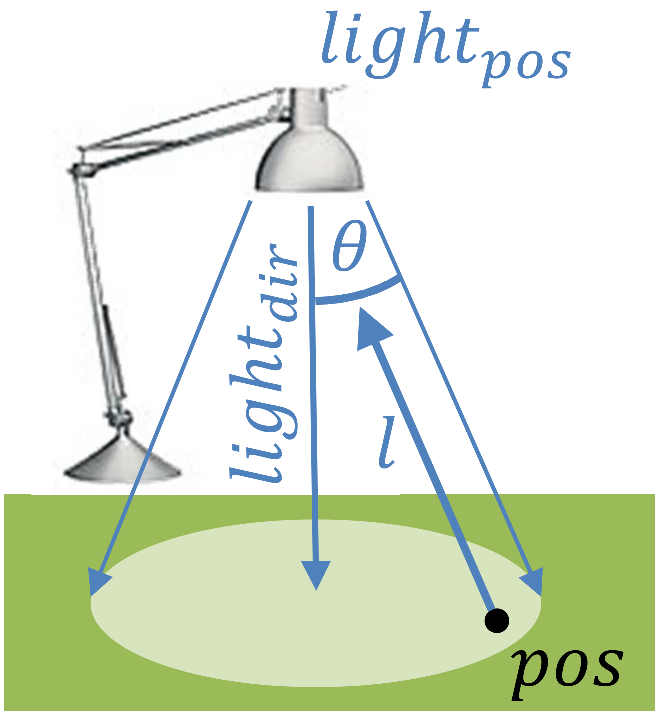
- Has a certain position and cone in space
- Cone can be specified by opening angle and central direction



Spot Light

- Point is in cone iff

$$\cos^{-1} dot(l, -light_{dir}) < \theta$$



Models

Color model

Light sources

Reflection model

Shading model

Reflection/Illumination model

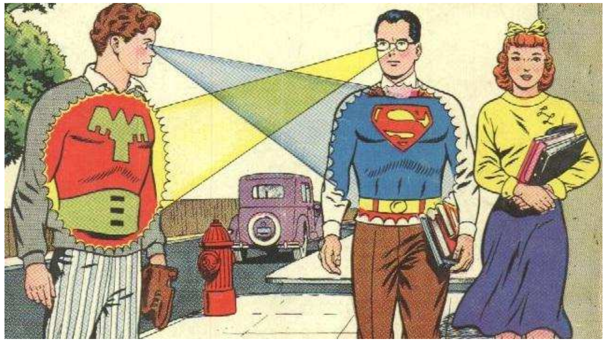
- Input
  - Surface properties: Glossiness
  - ...

- Output

Glossy → Matte

Reflection/Illumination model

- Input
  - Surface properties: Opaque
  - …

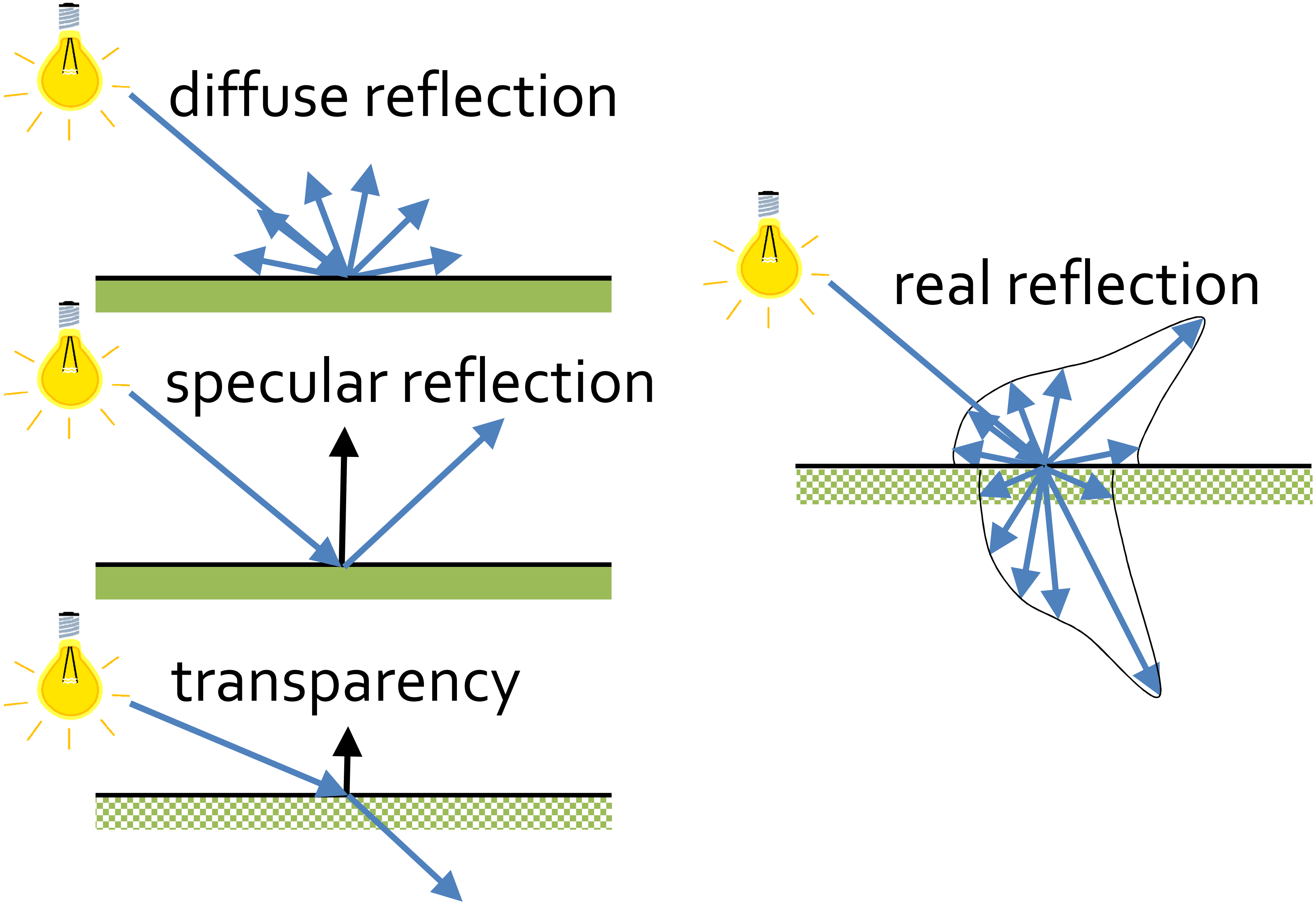


- Output

Reflection/Illumination model

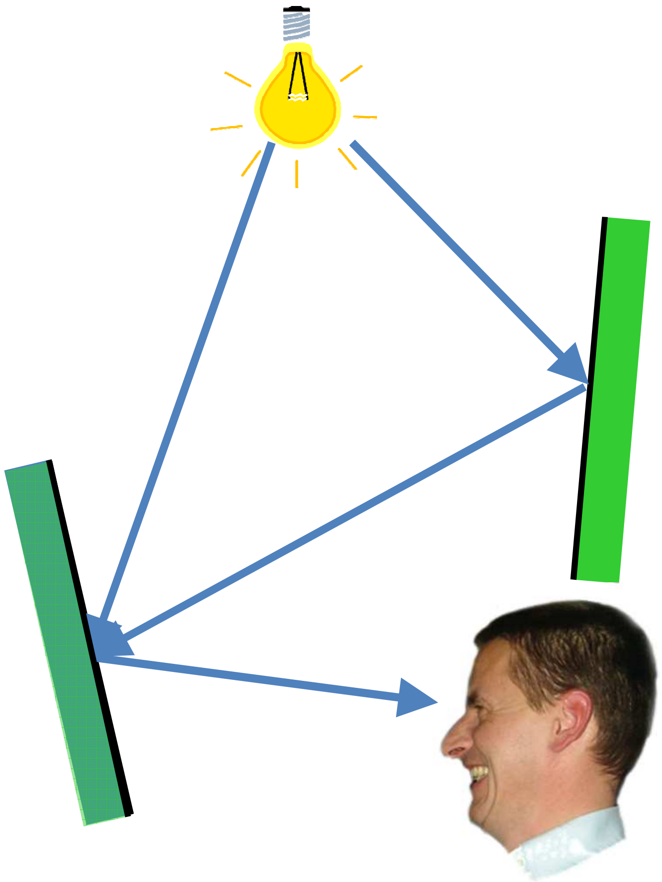
- Input
  - Surface properties
    - Glossiness
    - Opaqueness
    - Reflection, absorption
  - Background lighting conditions
- Output
  - Simulate range of surface lighting effects
  - Calculate intensity / color values

Surface lighting effects



Surface lighting effects

reflections from other surfaces



Lighting Model

photorealistic
reflection models

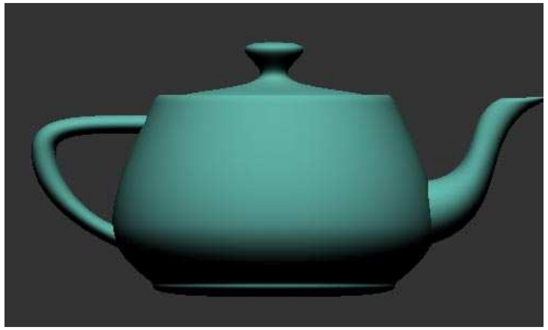
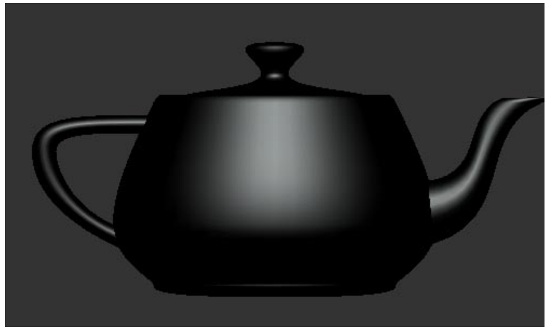
non-photorealistic

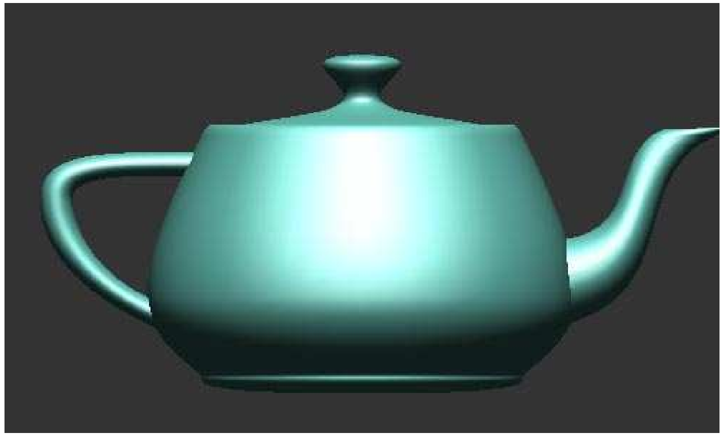
local
- ambient
- diffuse
- specular

global
- raytracing
- radiosity

Local lighting model – hack!

- $\text{Color} =$
  $$ambient \quad + \quad diffuse \quad + \quad specular$$
- <component> … lighting model component


+

+


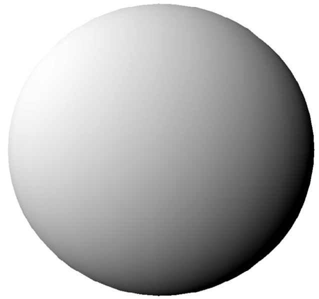
=


Ambient Light Reflection

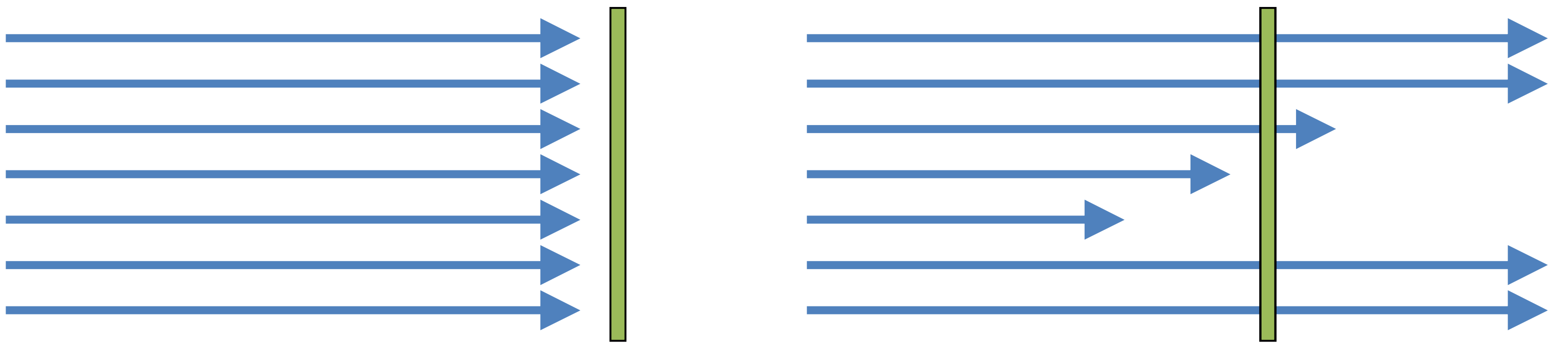
- Background light
  - No direction because scattered so often
  - "Color inside shadow"
  - Approximation of global diffuse lighting effects
- $S_{ambient}$...background light color
- M...material color

$$ambient = M * S_{ambient}$$

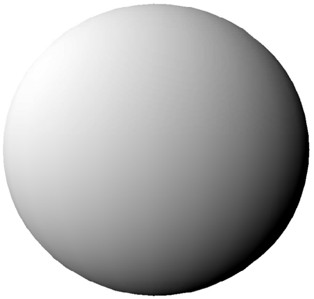
Diffuse Light Reflection



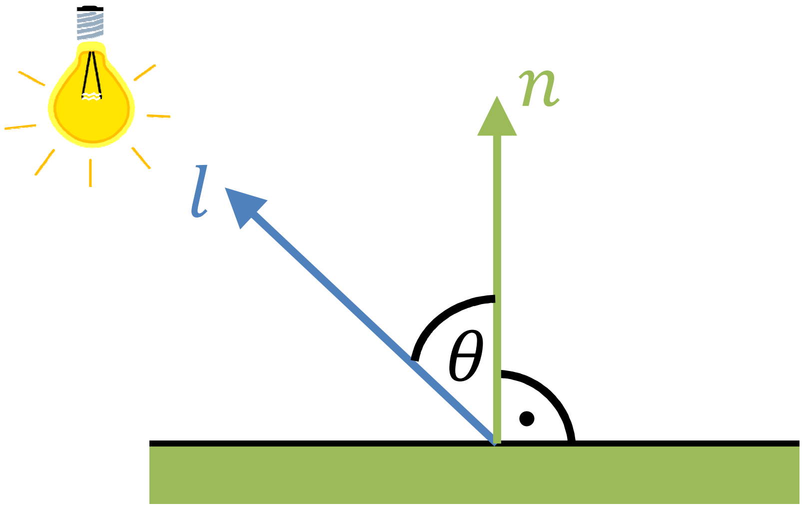
- The flatter light falls on a surface,
  the darker it will appear
- Ideal diffuse reflectors (Lambertian reflectors)
- Brightness depends on orientation of surface



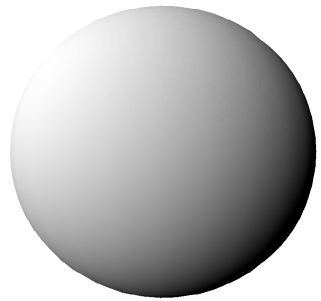
Diffuse Light Reflection



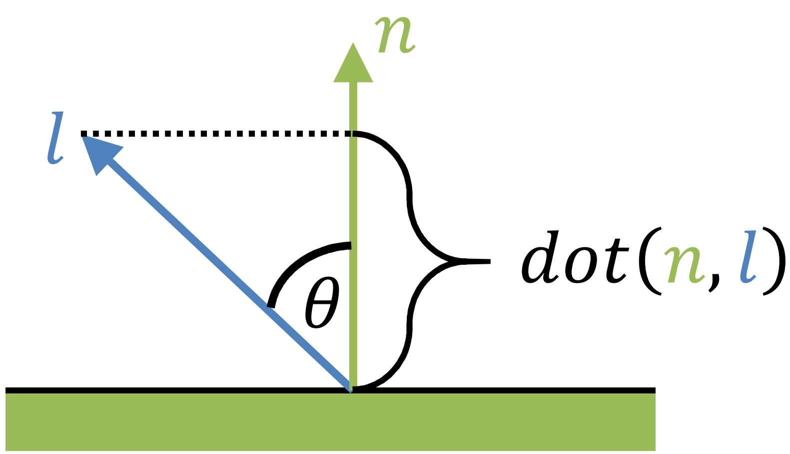
- Diffuse brightness is dependent on angle between
  n … surface normal and
  l … direction to the light

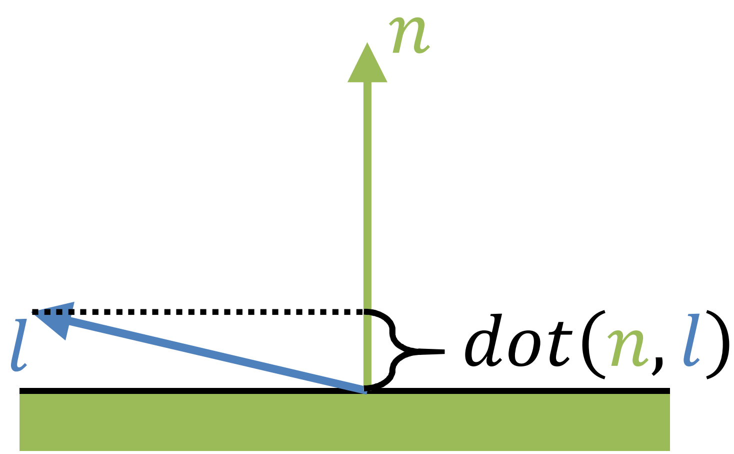


Diffuse Light Reflection

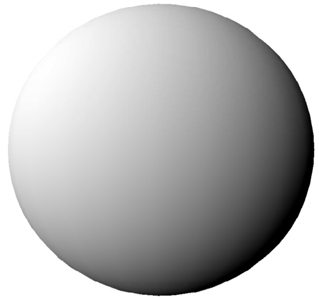


- Diffuse brightness is dependent on angle between
  n ... surface normal and
  l ... direction to the light
- $\cos\theta = dot(n, l)$

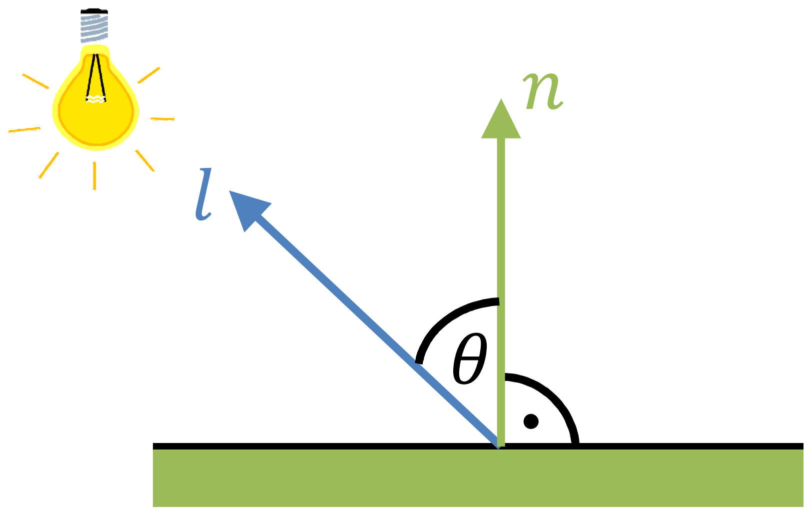




Diffuse Light Reflection

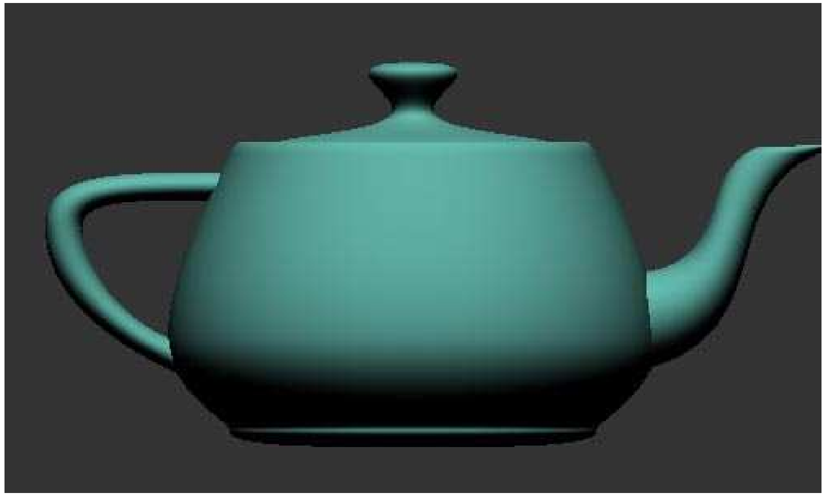


- S…light color
- M…material color
- $diffuse = M * S \cdot \mathrm{dot}(n, l)$



Lambertian (Diffuse) Reflection

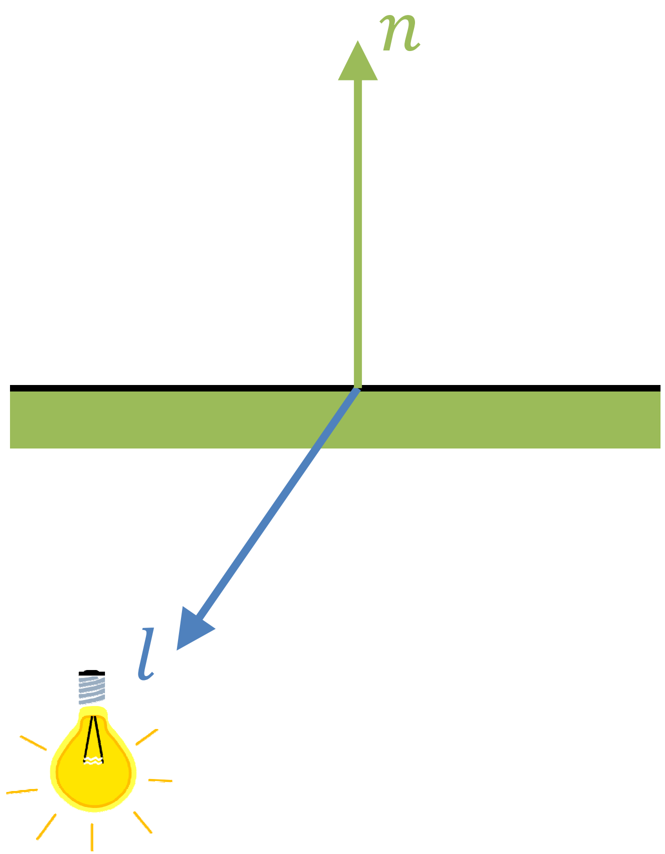
$diffuse$



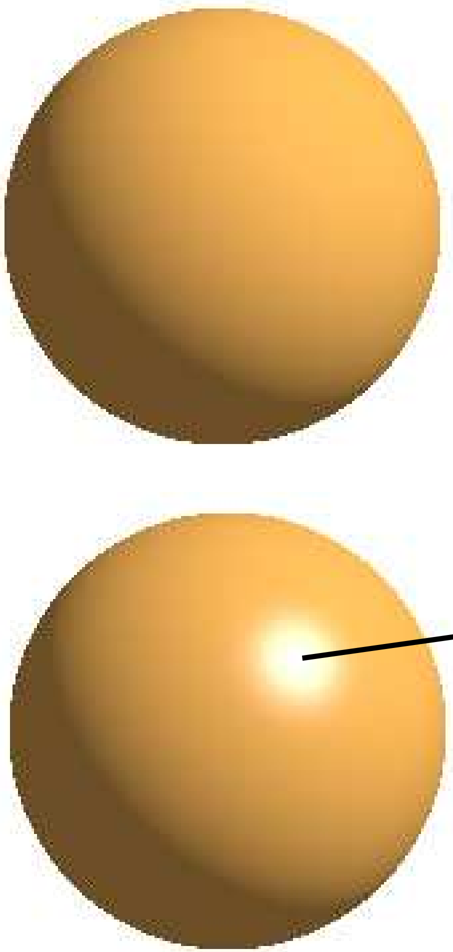
$ambient + diffuse$

Light from Behind

- Should be ignored
- $diffuse = M * S \cdot \max(0, \mathrm{d}ot(n, l))$



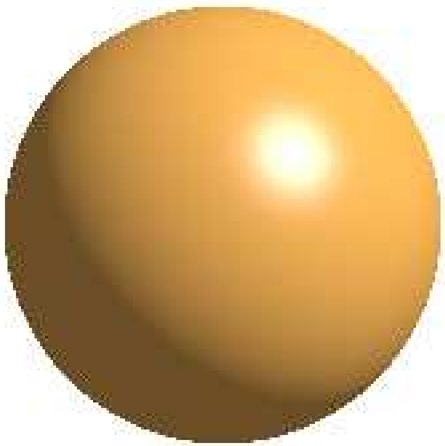
Specular Highlights

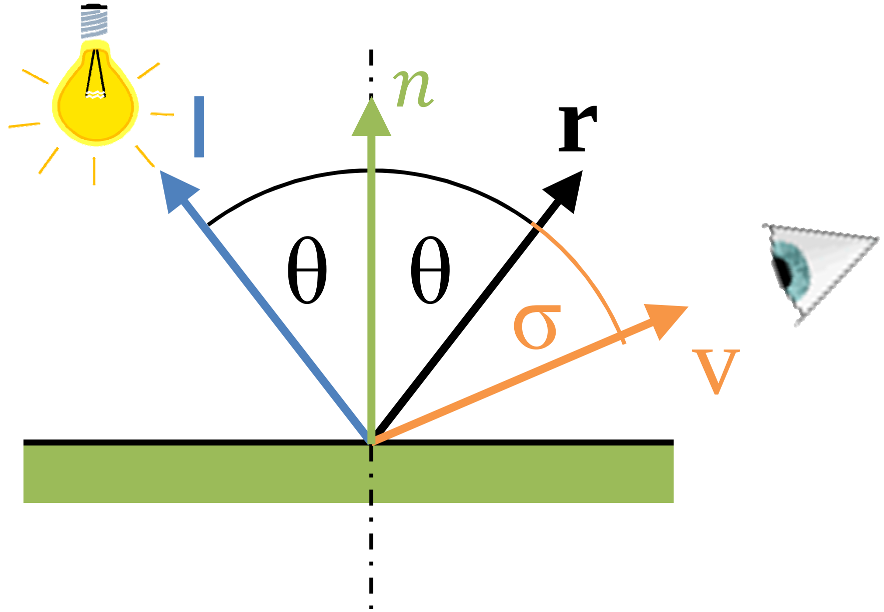


Light source reflection
directly into the viewer's eye

Specular Reflection Model

- Reflection of incident light around specular-reflection angle

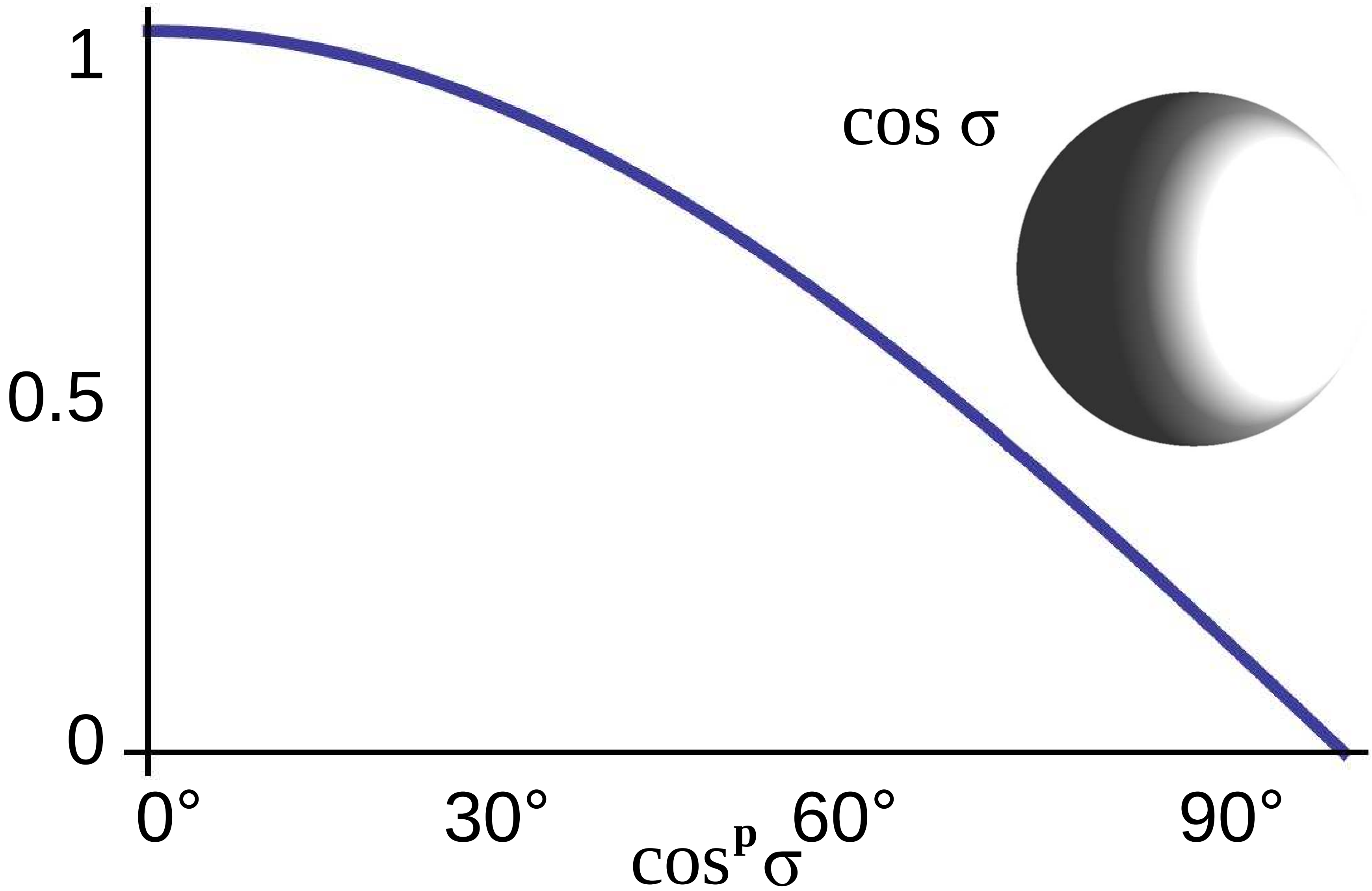




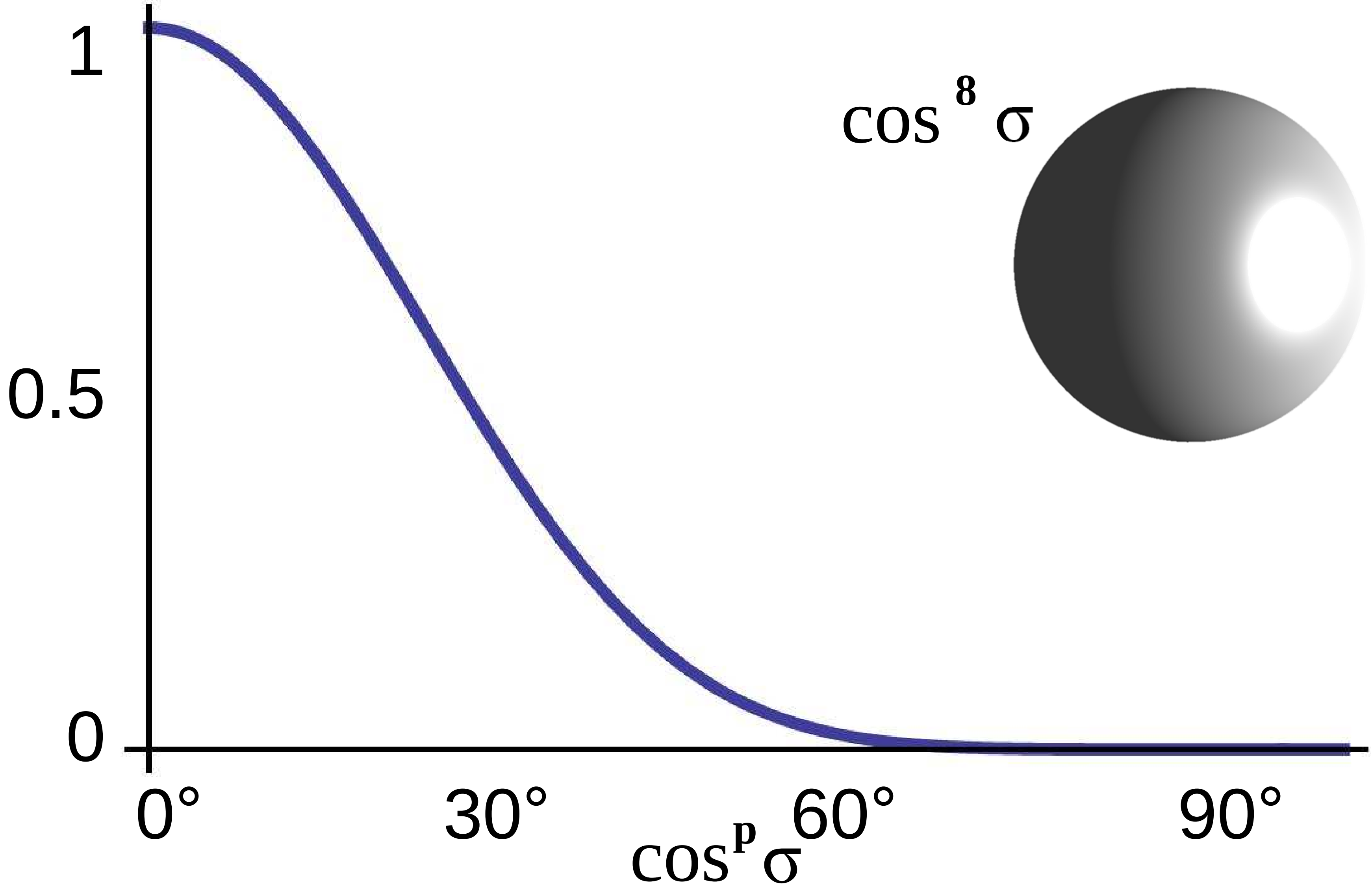
- Empirical Phong model

$$L_{spec} = M*S \cdot \cos^p \sigma$$

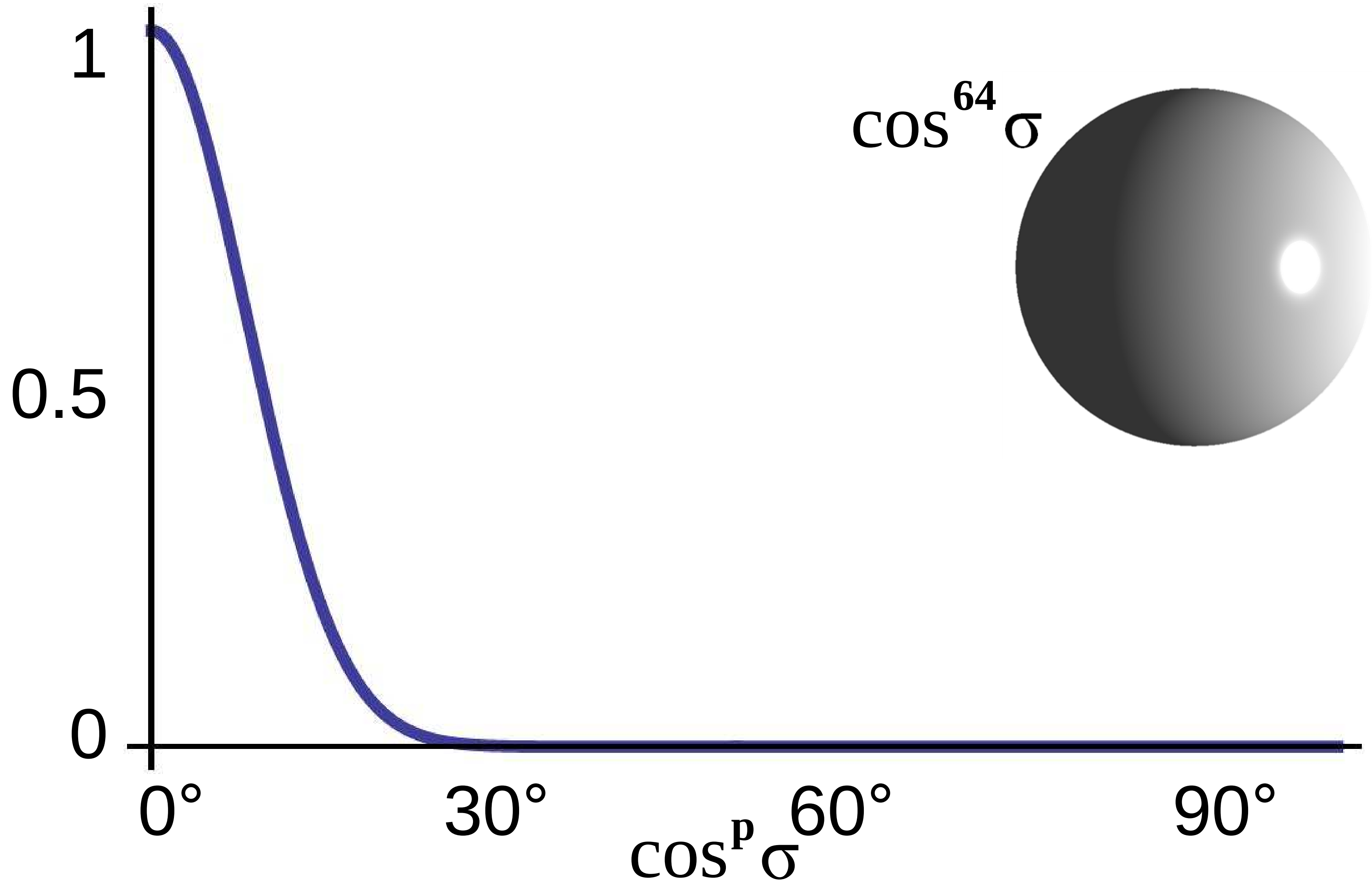
Specular Reflection Coefficient p



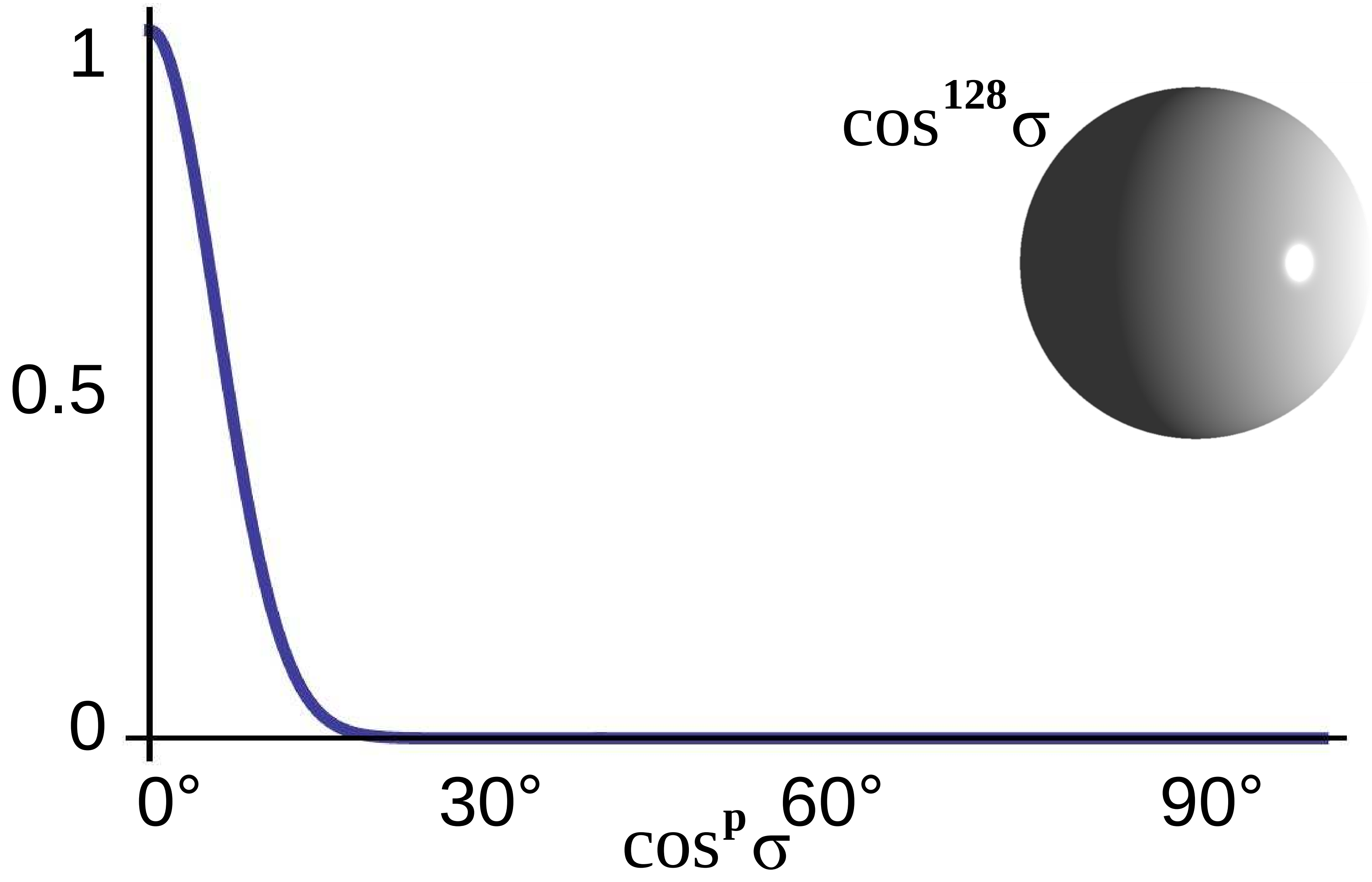
Specular Reflection Coefficient p



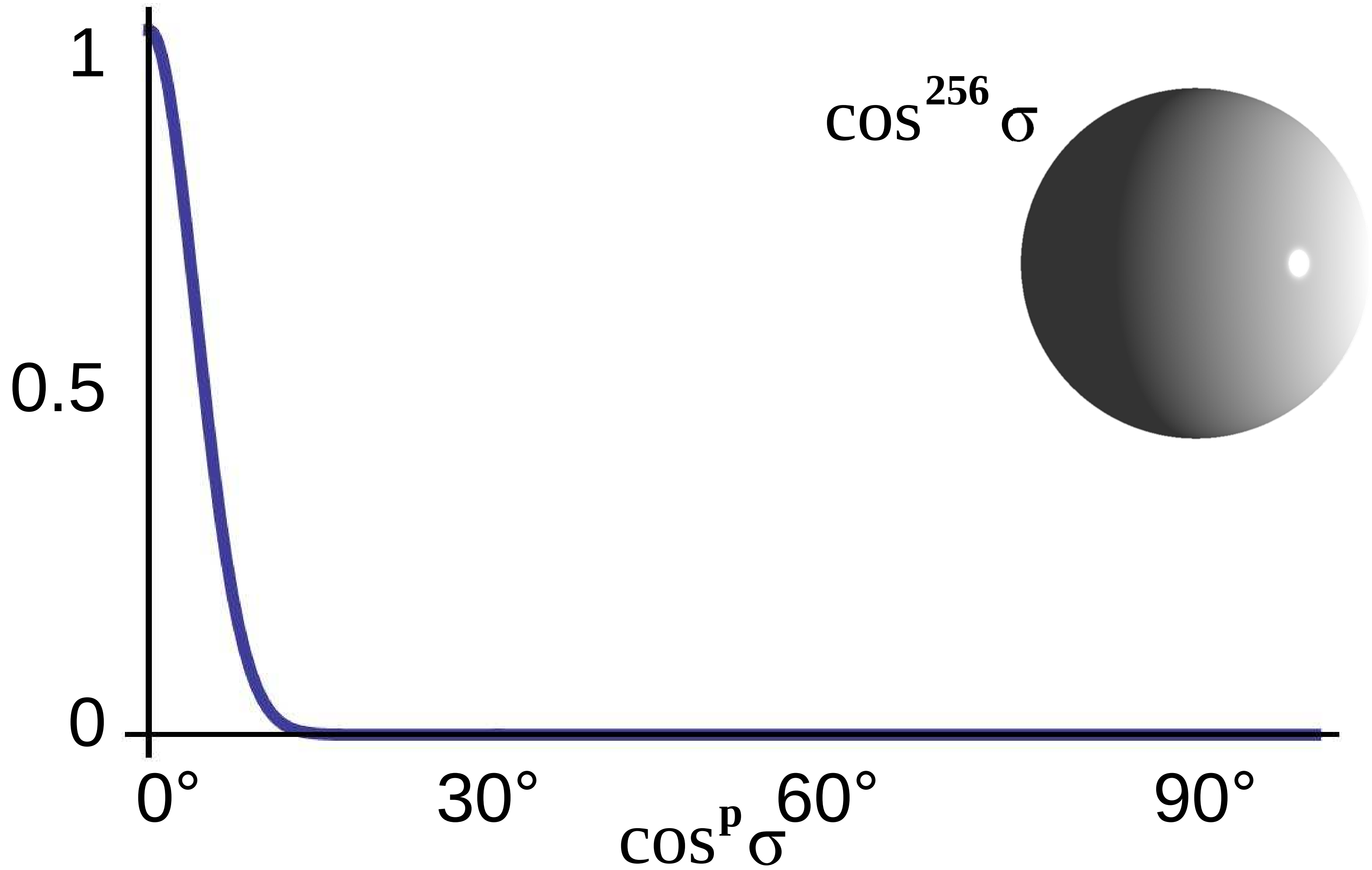
Specular Reflection Coefficient p



Specular Reflection Coefficient p



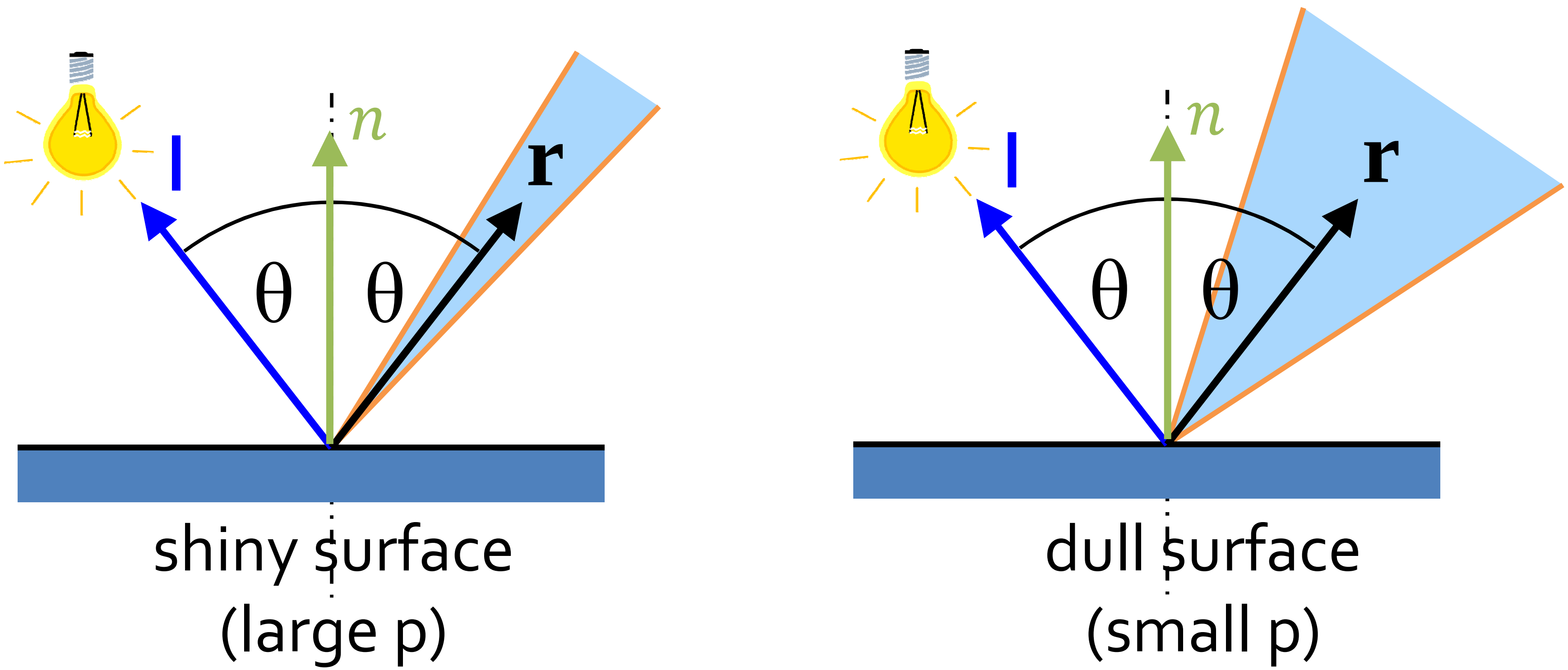
Specular Reflection Coefficient p



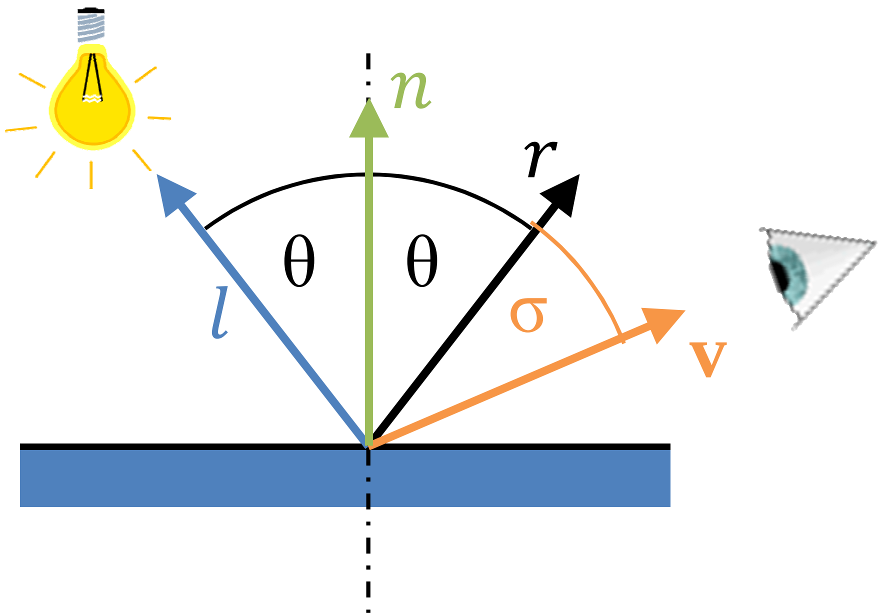
Specular Reflection Coefficient

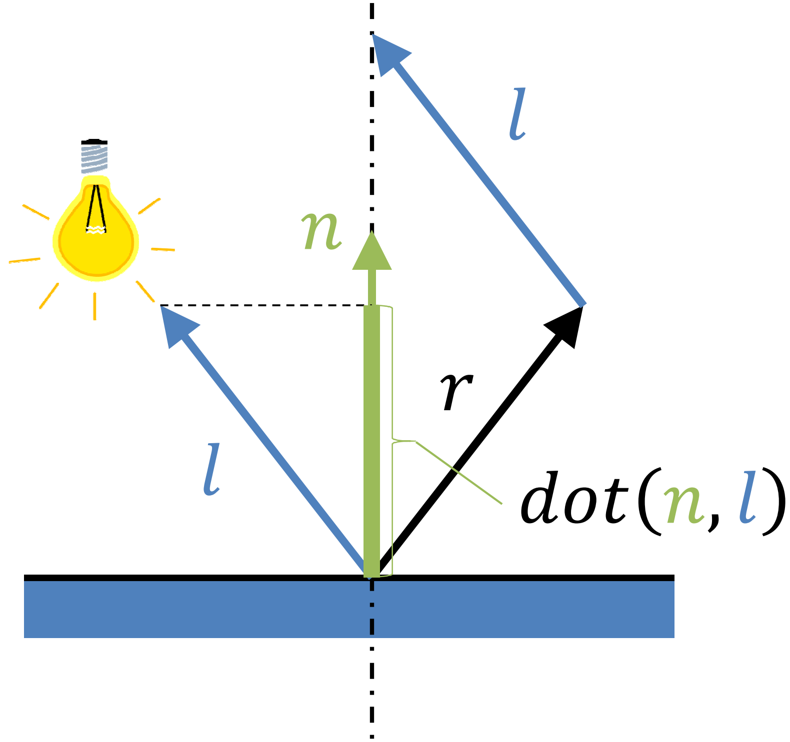
- Empirical Phong model
  - p large $\Rightarrow$ shiny surface
  - p small $\Rightarrow$ dull surface

$$L_{spec} = M*S \cdot \cos^p \sigma$$



Specular reflection





$$L_{spec} = M*S\cdot(\mathbf{v}\cdot\mathbf{r})^{p}$$

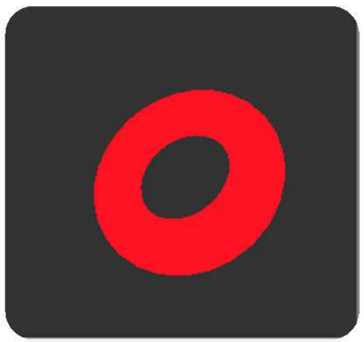
$$r + l = 2 \cdot dot(n, l) \cdot n$$

$$r = 2 \cdot dot(n, l) \cdot n - l$$

Complete reflection model

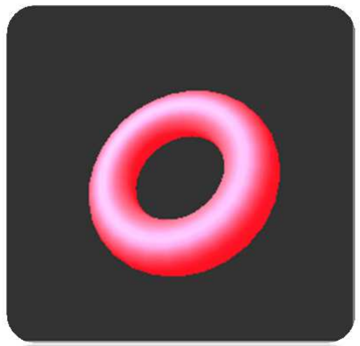
$$
\begin{aligned}
I &= I_a\, k_a + I\,(k_d \cos\theta + k_s \cos^p \alpha) \\
&= I_a\, k_a + I\,(k_d\,(\mathit{N} \bullet \mathit{L}) + k_s\,(\mathit{R} \bullet \mathit{V})^p)
\end{aligned}
$$

ambient



diffuse

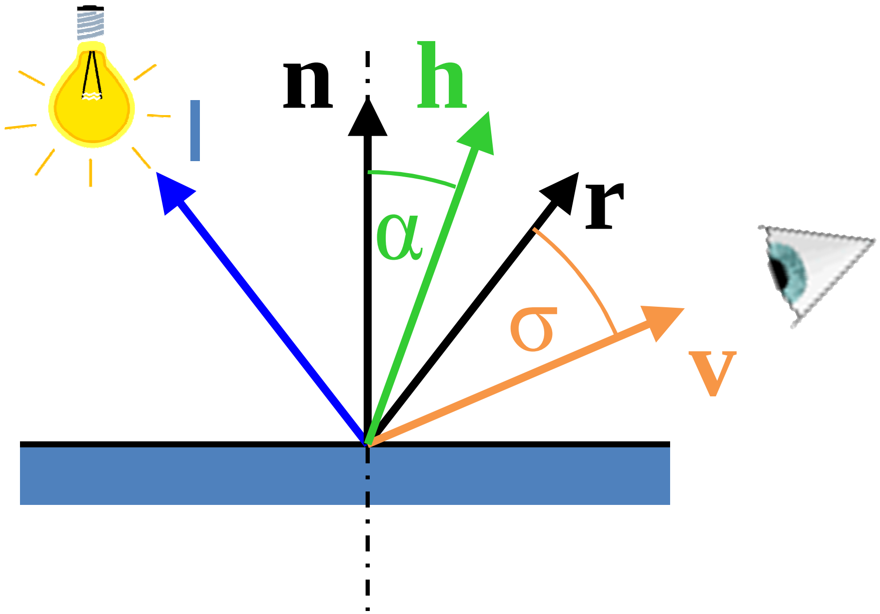
specular



Blinn-Phong

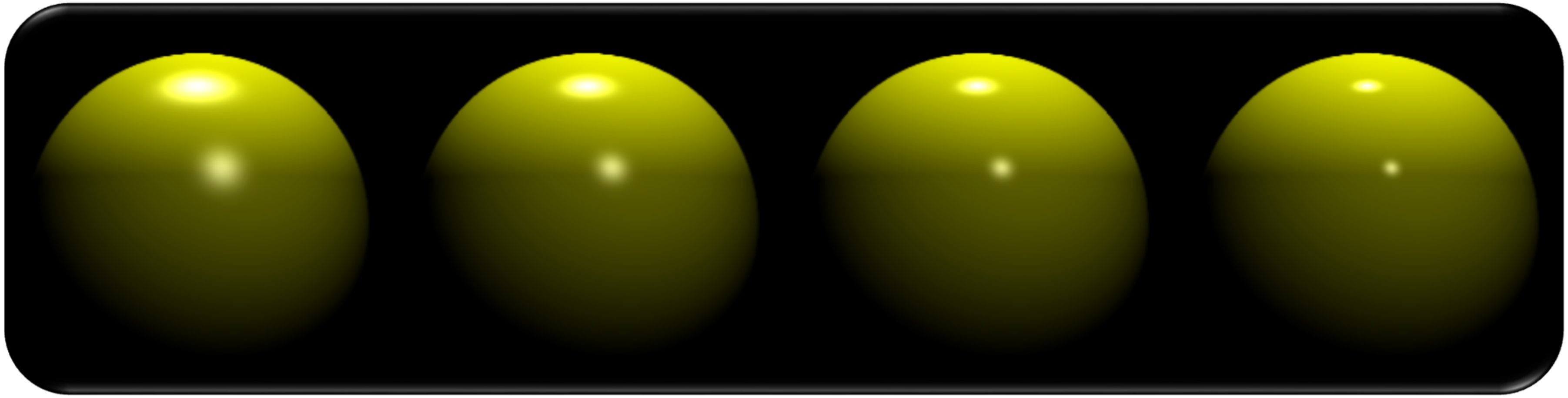
- Halfway vector $\mathbf{h}$

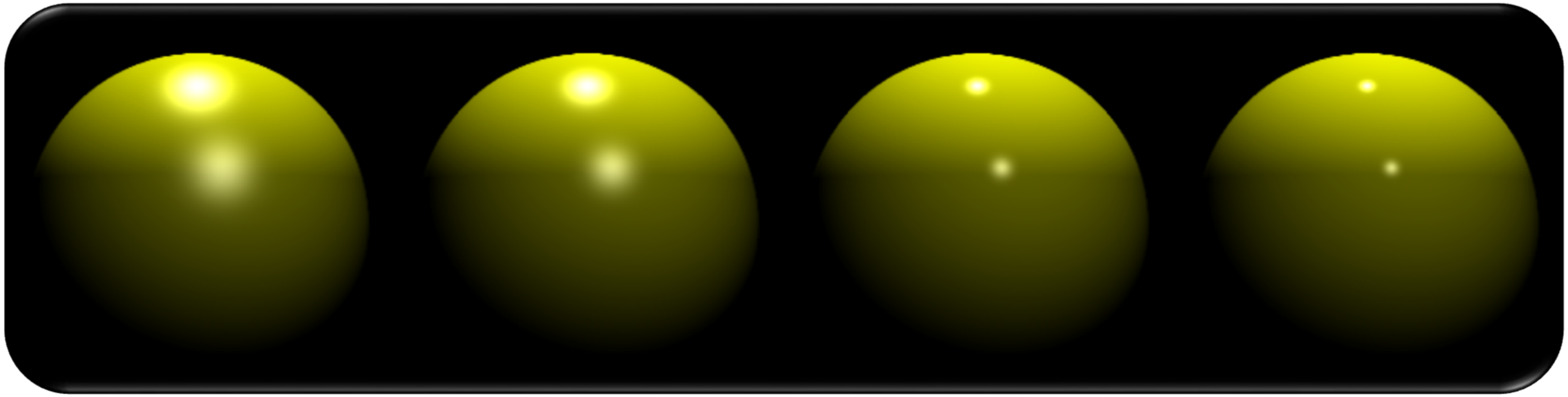
$$L_{spec} = k_s \cdot I \cdot (\mathbf{v} \cdot \mathbf{r})^p \quad \rightarrow \quad L_{spec} = k_s \cdot I \cdot (\mathbf{n} \cdot \mathbf{h})^p$$



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

Phong vs Blinn-Phong





Models

Color model

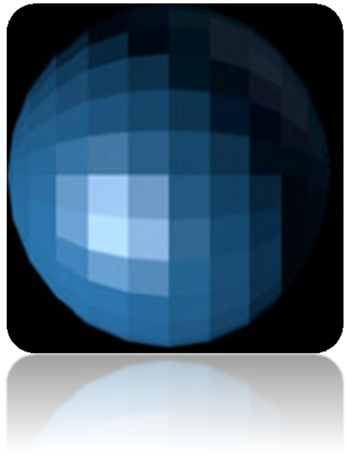
Light sources

Reflection model

Shading model

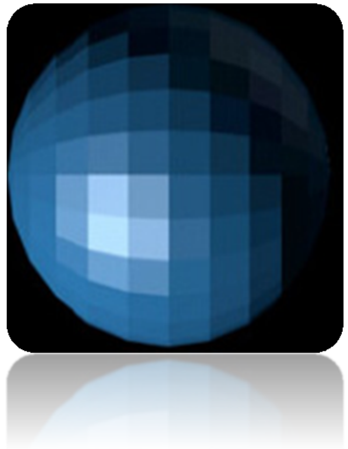
Shading Models

Shading model



- Shading ≠ shadows (shadowing)
- Coloring / shading the model
- When to evaluate lighting model

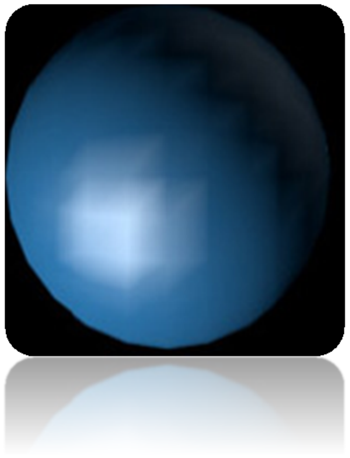
Flat-shading

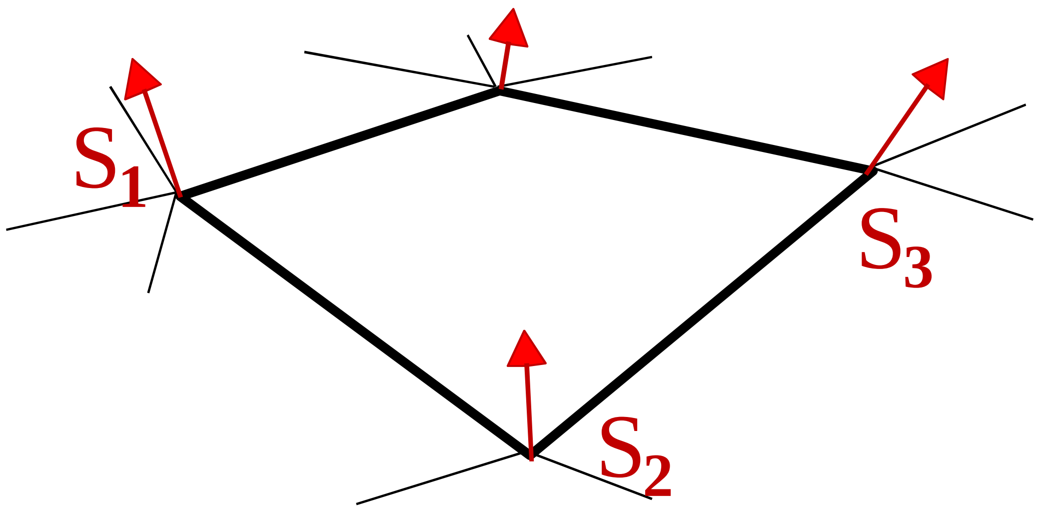


- 1 color for the mesh (polygon)
- Really fast
- Really ugly
- *If an object really <u>is</u> faceted, is this accurate?*
- No:
  - Point light sources
    - Direction to light varies across the facet
  - Specular reflectance
    - Direction to eye varies across the facet

Gouraud shading

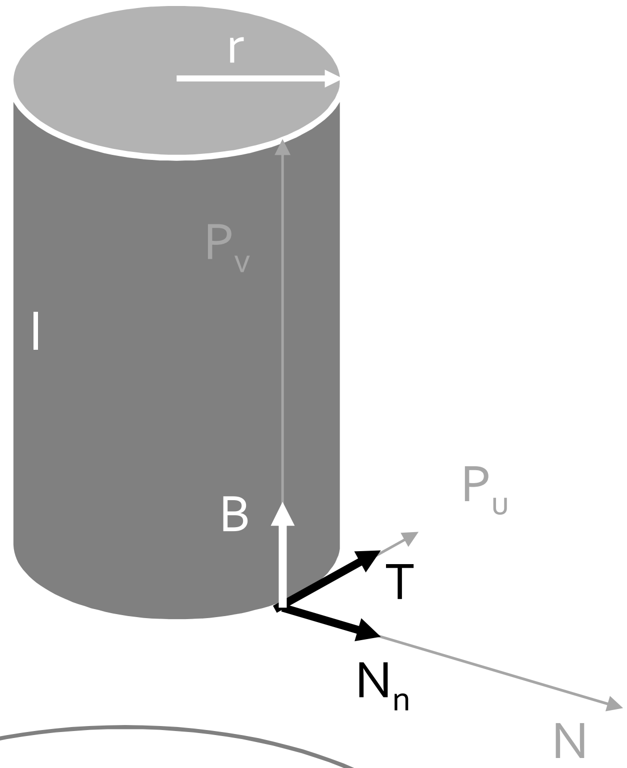
1. Calculate the normal vector for each vertex
2. Calculate the intensitity for each vertex

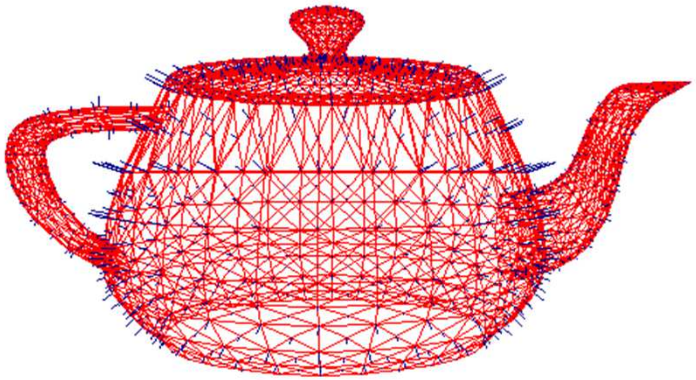


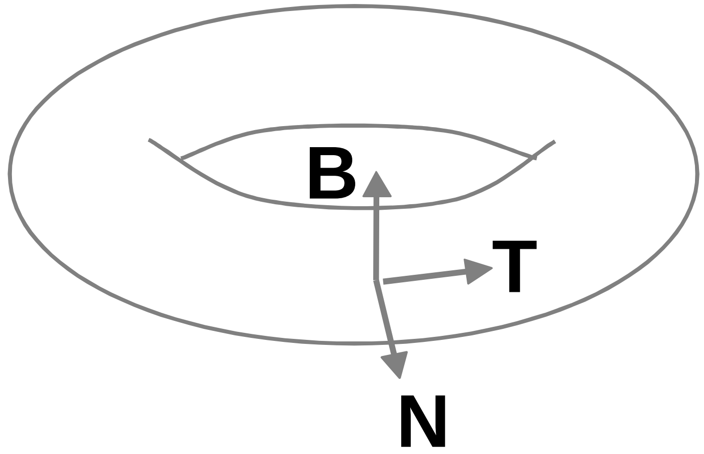


Vertex Normals

- Vertex normals may be
  - Provided with the model
    - Artist
    - 3d program
  - Computed from first principles
    - Mathematic description of model

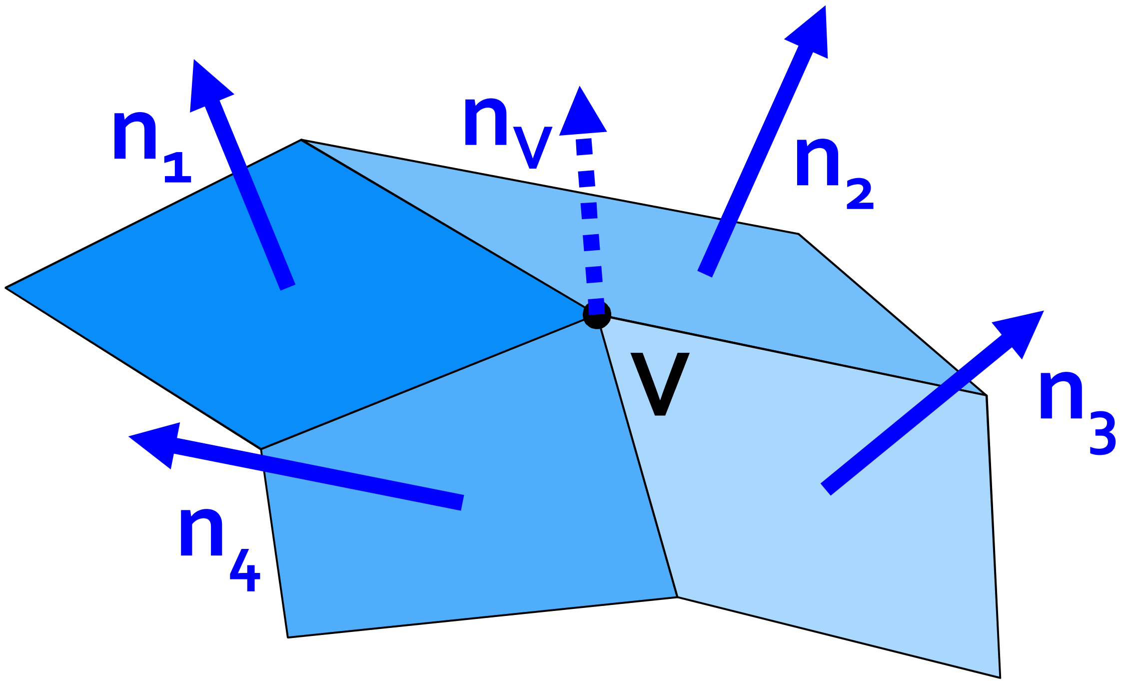






Vertex Normals

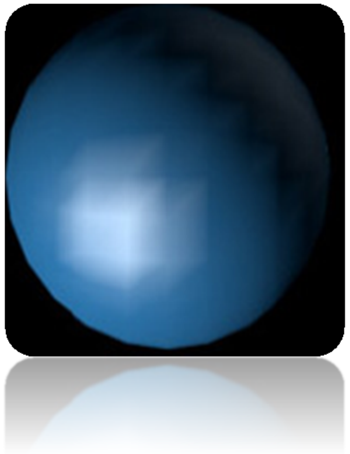
- Approximated by averaging the normals of the facets that share the vertex

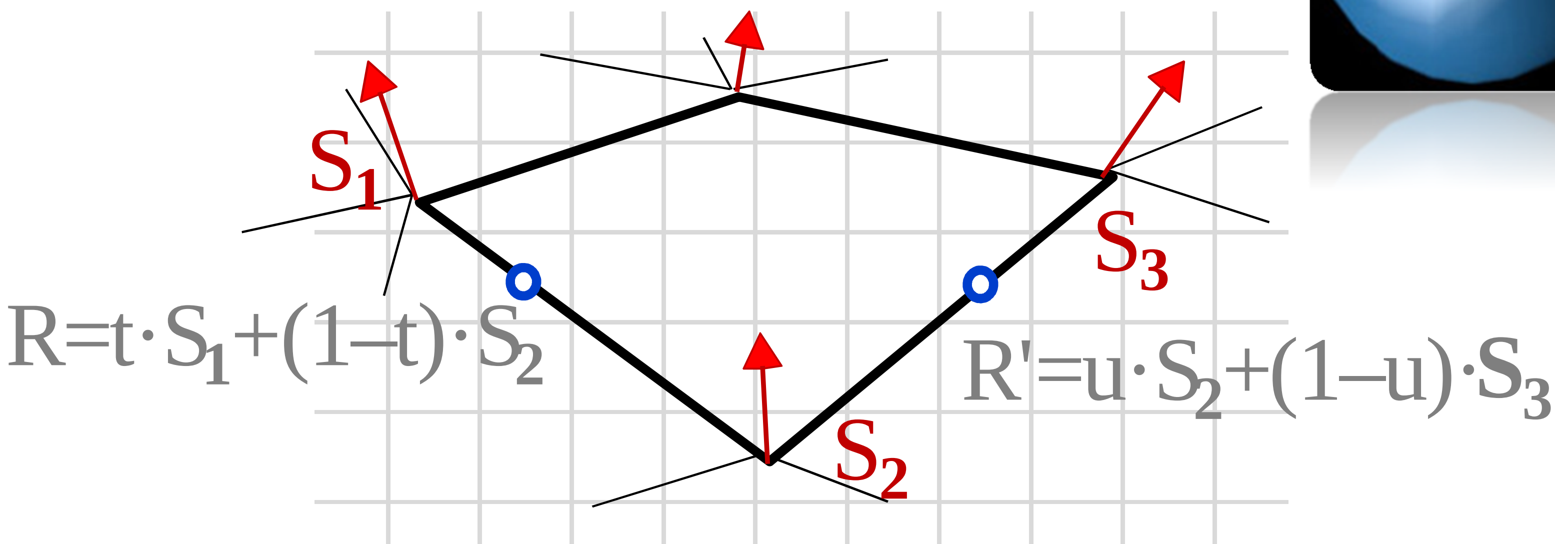


$$\mathbf{n}_V = \frac{\sum_{k=1}^{N} \mathbf{n}_k}{\left\| \sum_{k=1}^{N} \mathbf{n}_k \right\|}$$

Gouraud shading

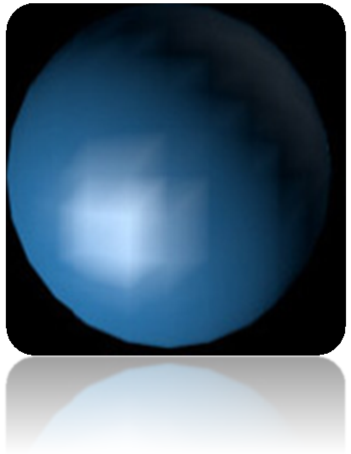
1. Calculate the normal vector for each vertex
2. Calculate the intensitity for each vertex
3. Color interpolation along edges

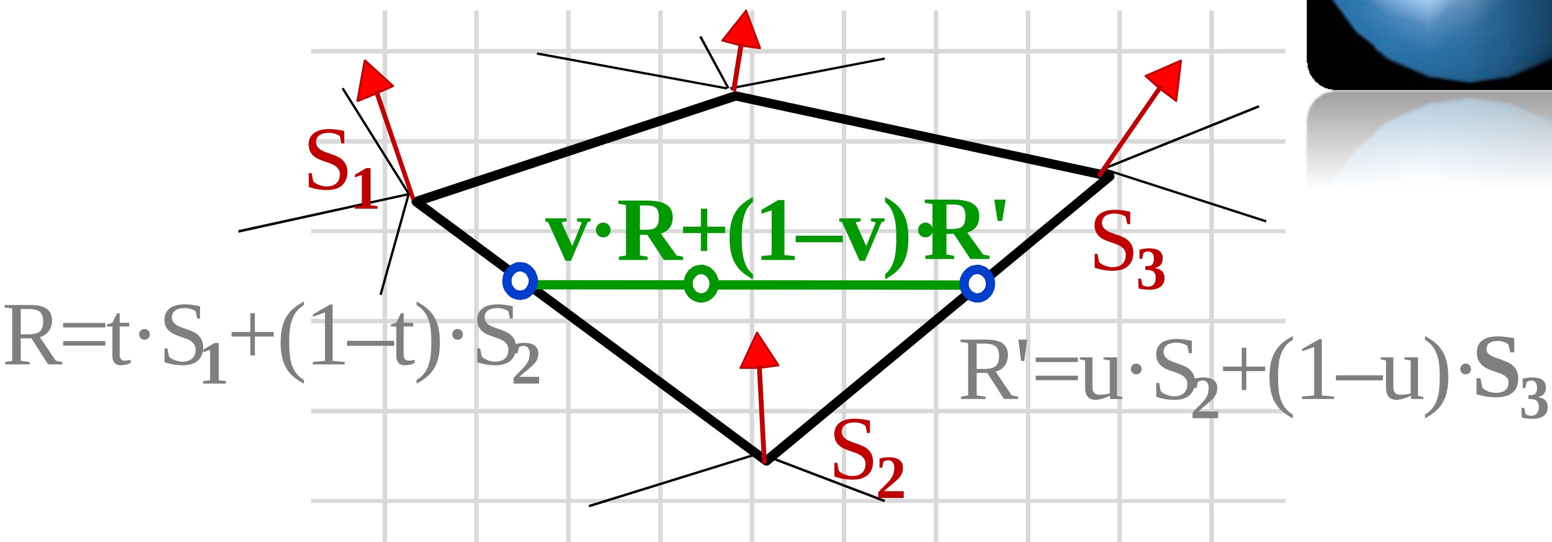




Gouraud shading

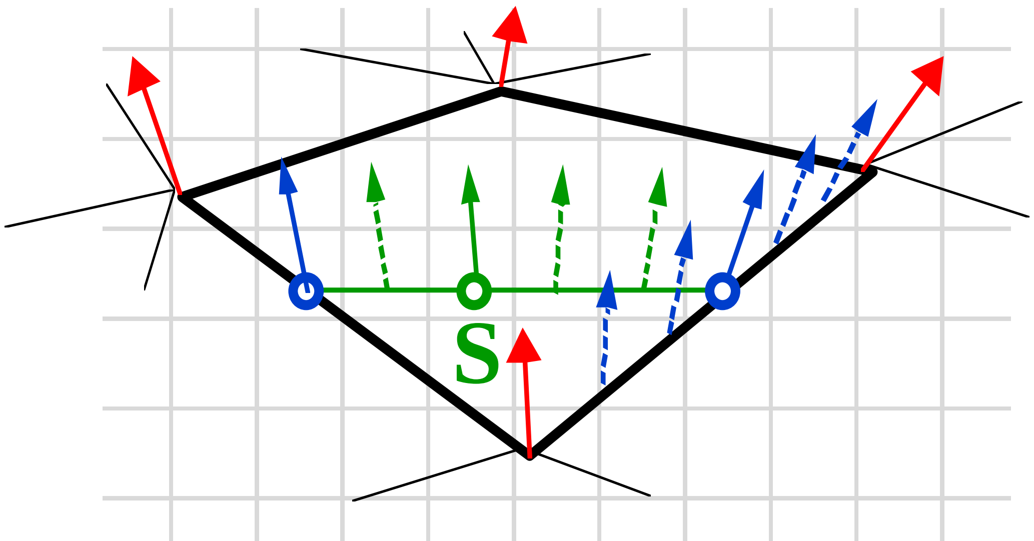
1. Calculate the normal vector for each vertex
2. Calculate the intensitity for each vertex
3. Color interpolation along edges
4. Color interpolation along scanline

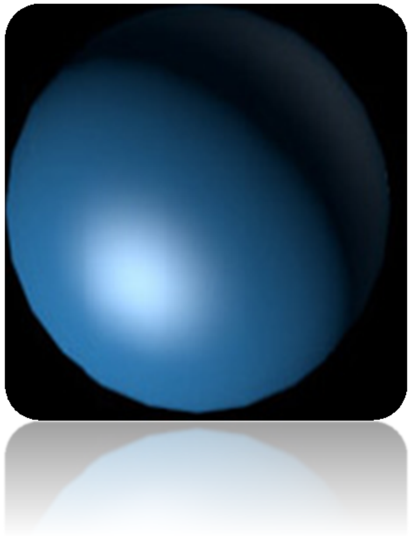




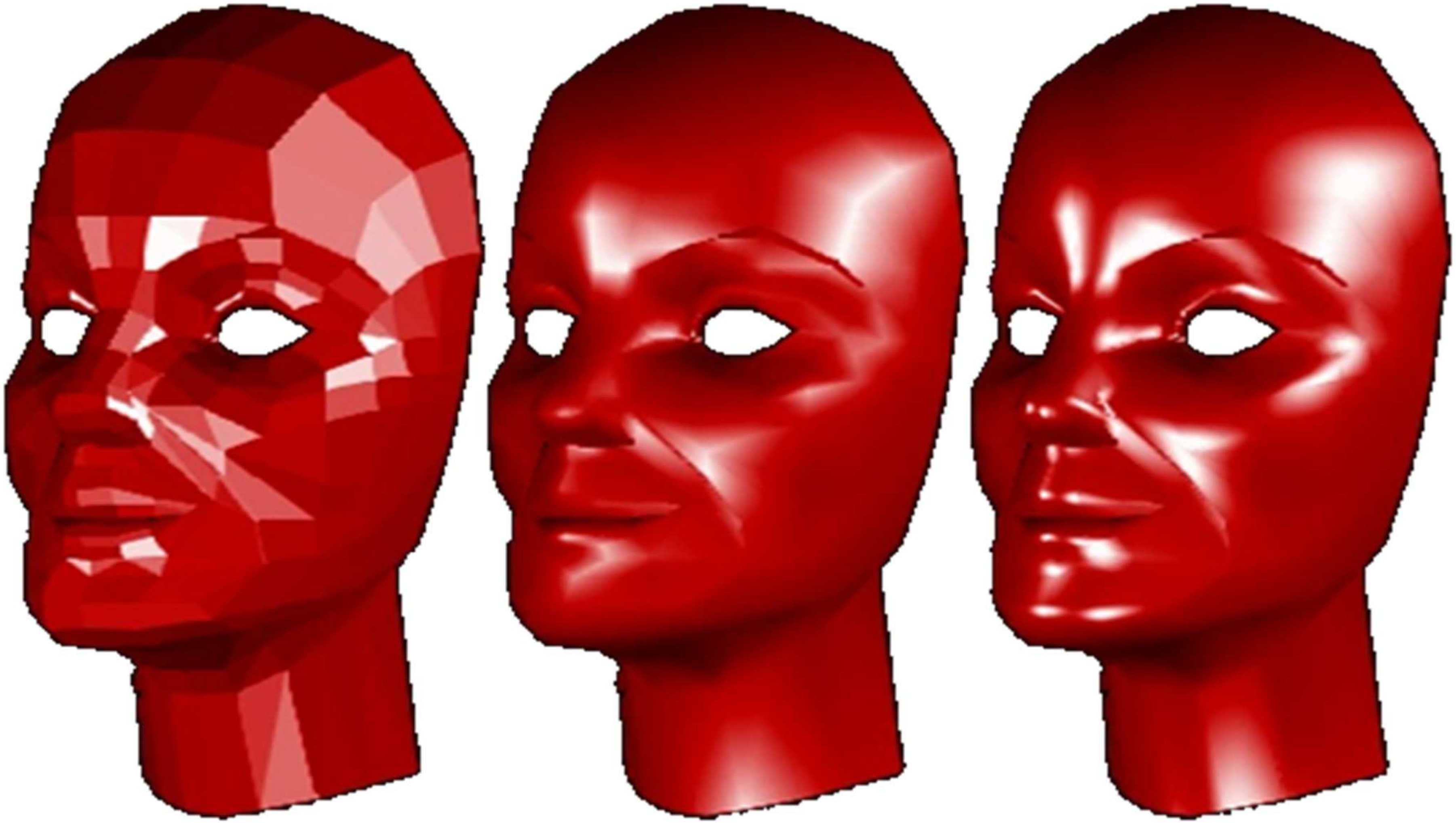
Phong Shading Model

1. Normal Vector for each vertex
2. Normal vectors are interpolated along the edge
3. Normal vectors are interpolated along the scanline
4. Calulate the intensity using the normal vectors



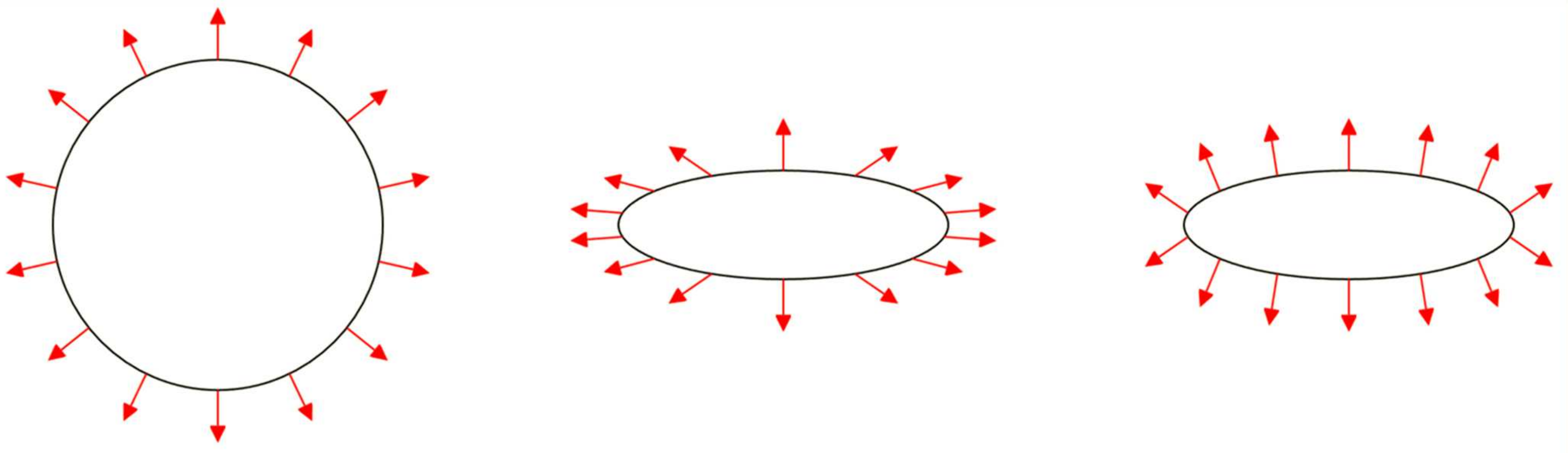


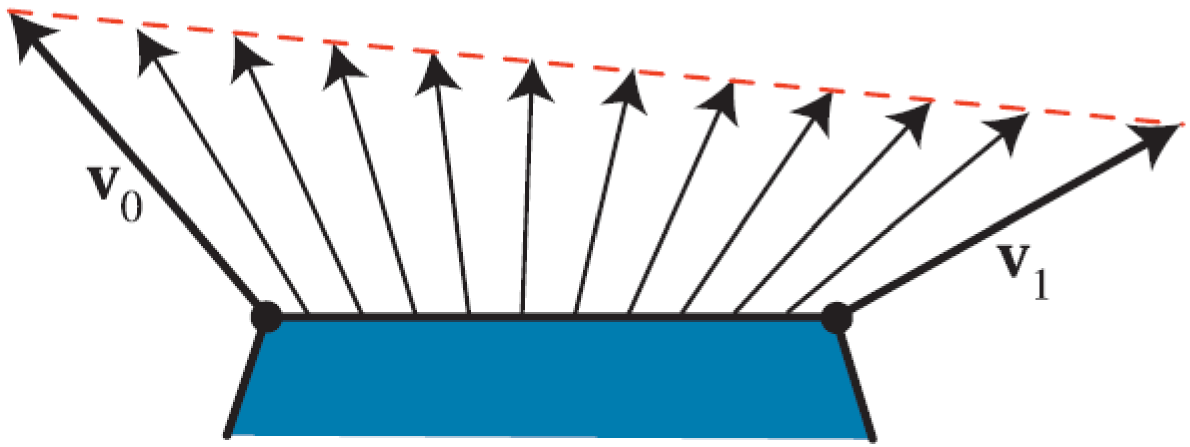
Flat / Gouraud / Phong Comparison



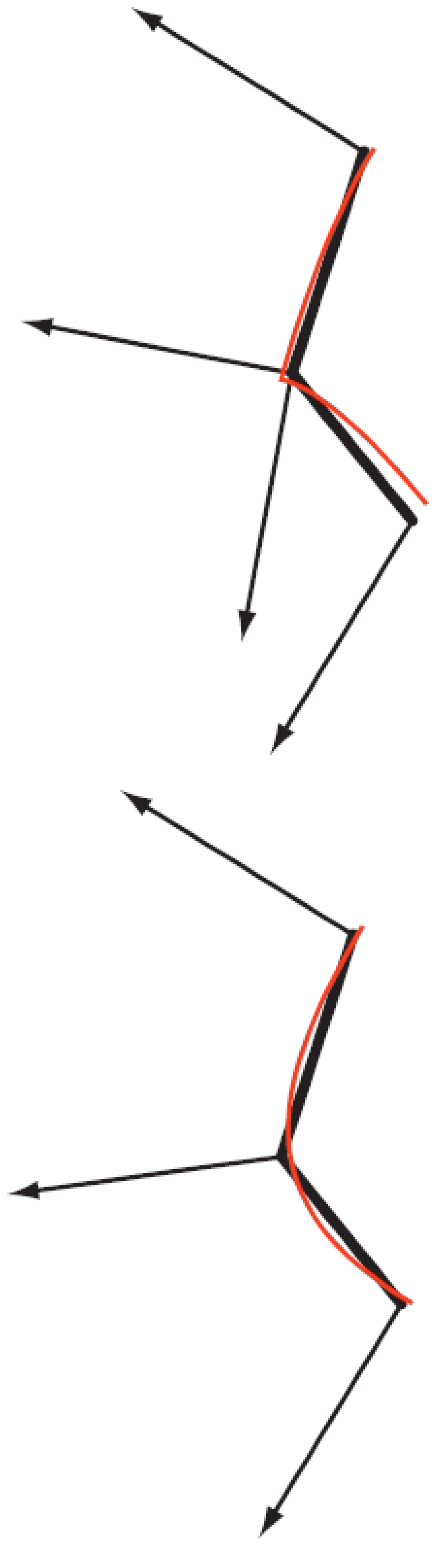
Transforming Normals

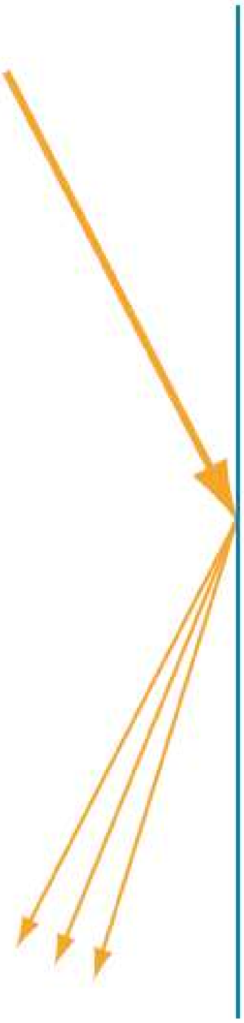
- Differential scaling changes shape and normals
- If **M** transforms points,
  then $(\mathbf{M}^T)^{-1}$ transforms normals

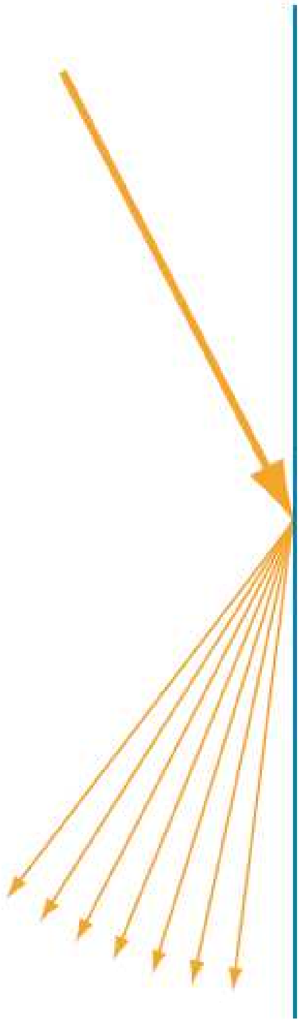


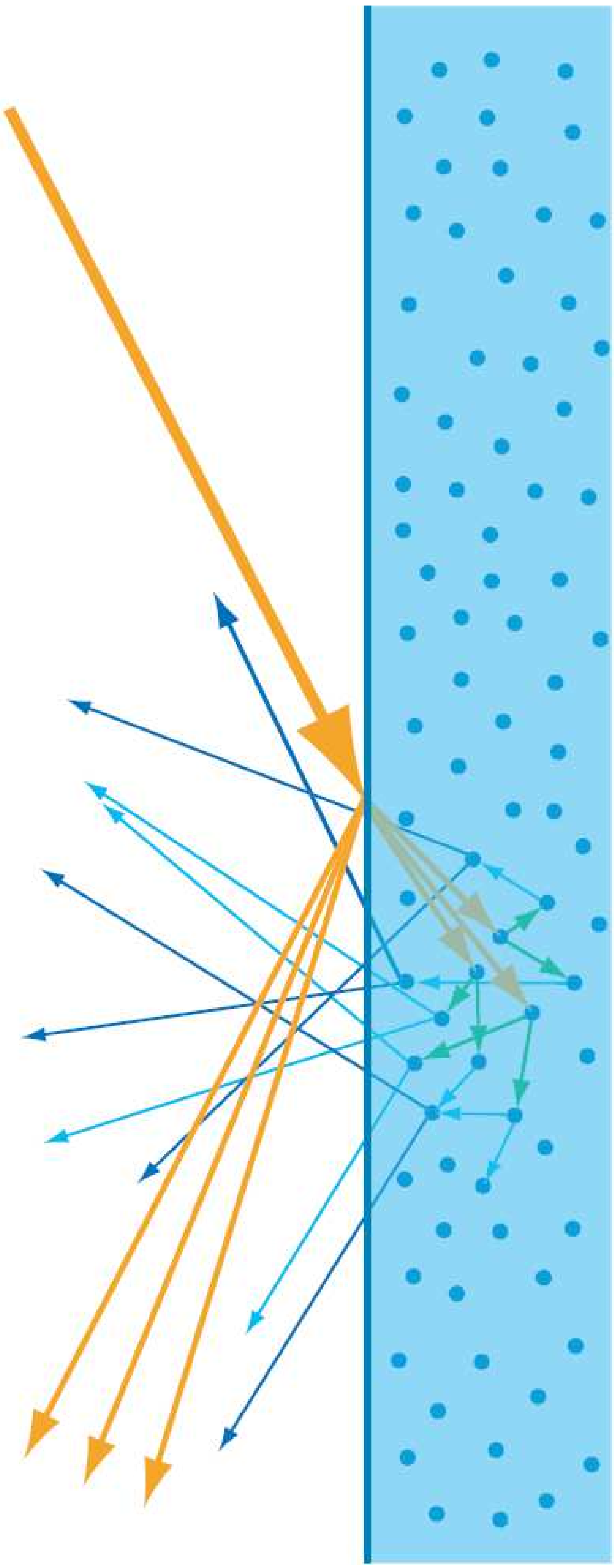


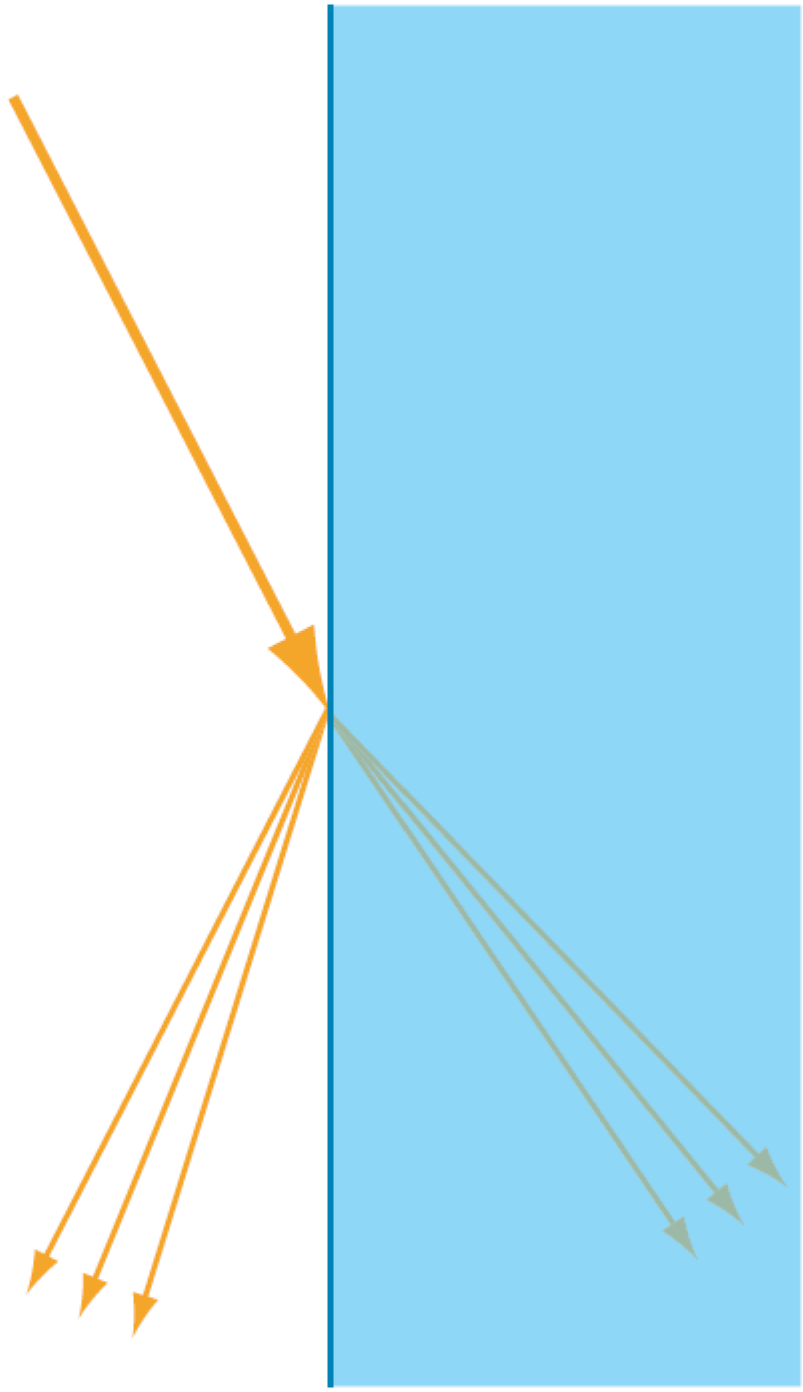
$\mathbf{v}_0$
$\mathbf{v}_1$

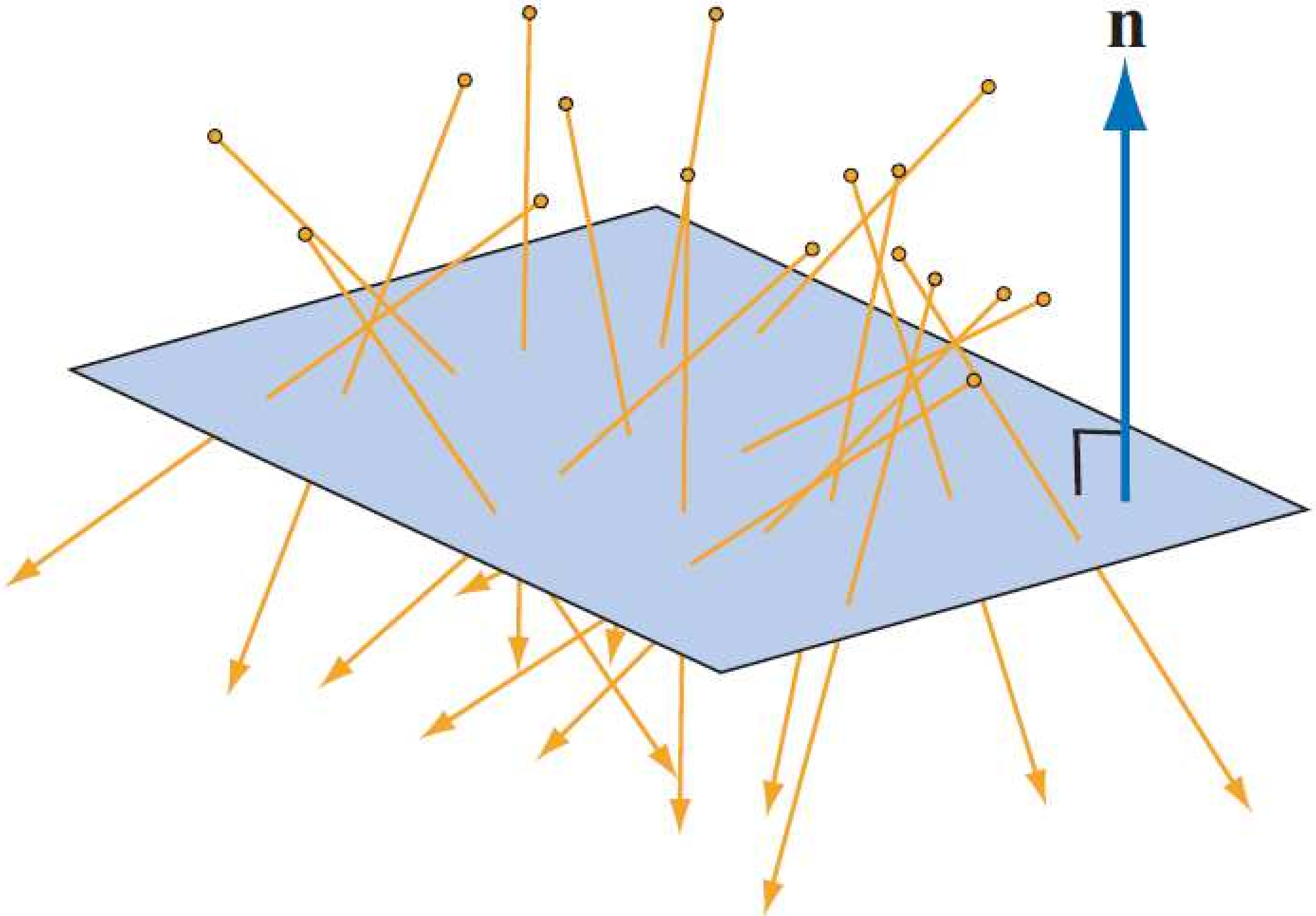











n