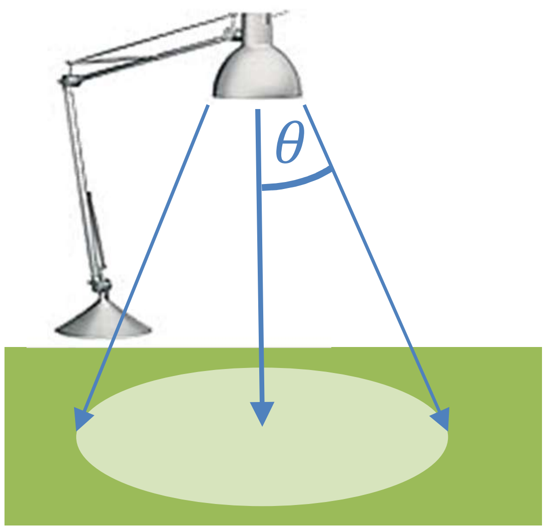
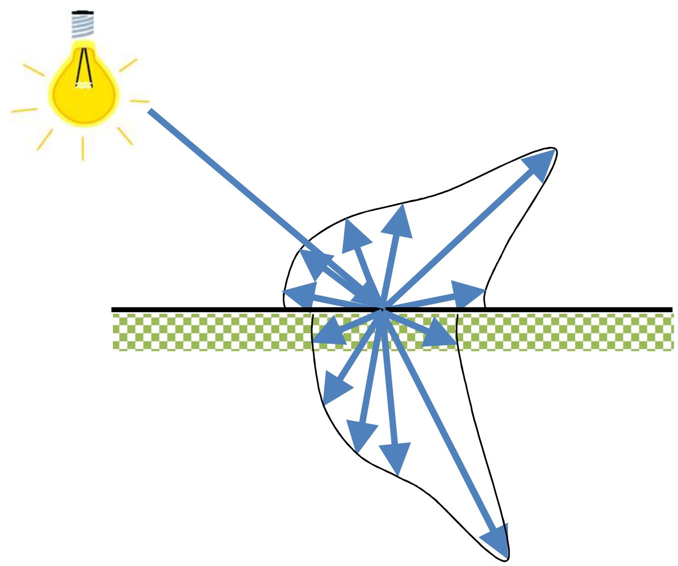
Lighting


Autodesk Maya, Arnold, Nuke
Naveen Kumar - CG Lighting Artist | Blog : nvncgartist.blogspot.in

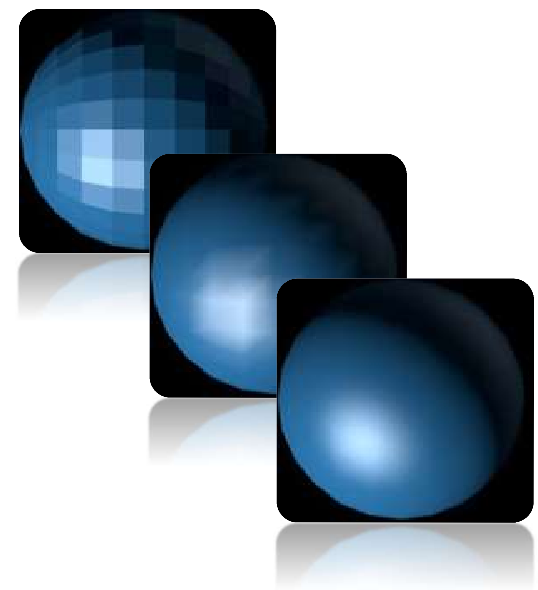
Light Sources



Reflection Models

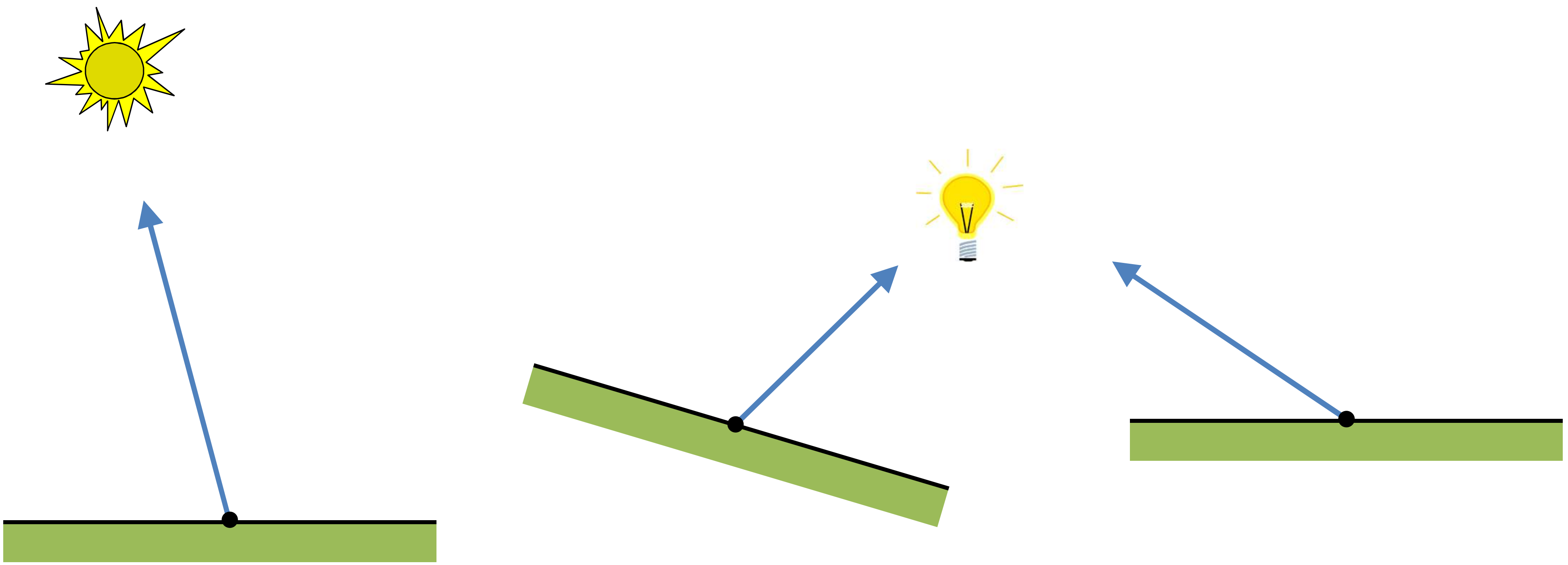


Shading Models



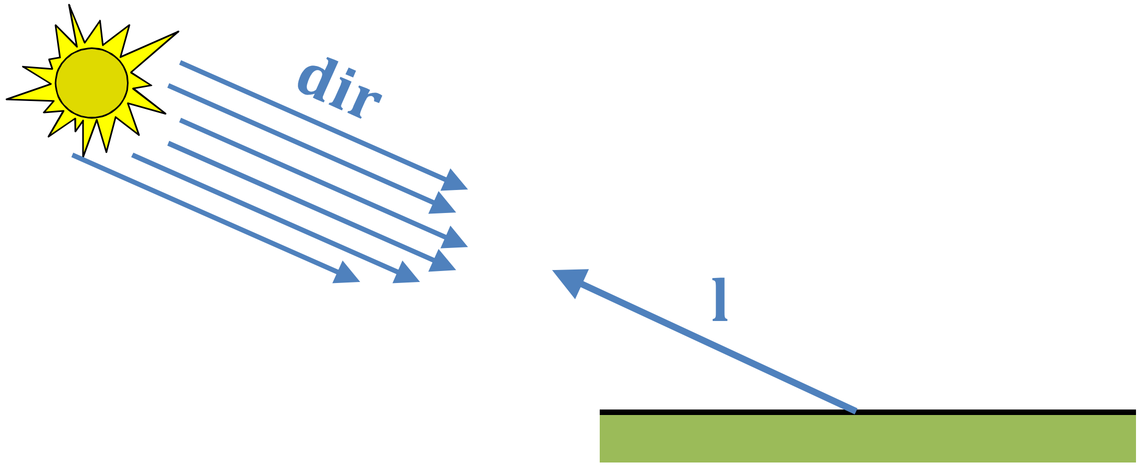
Light Sources

- Where is the light coming from?



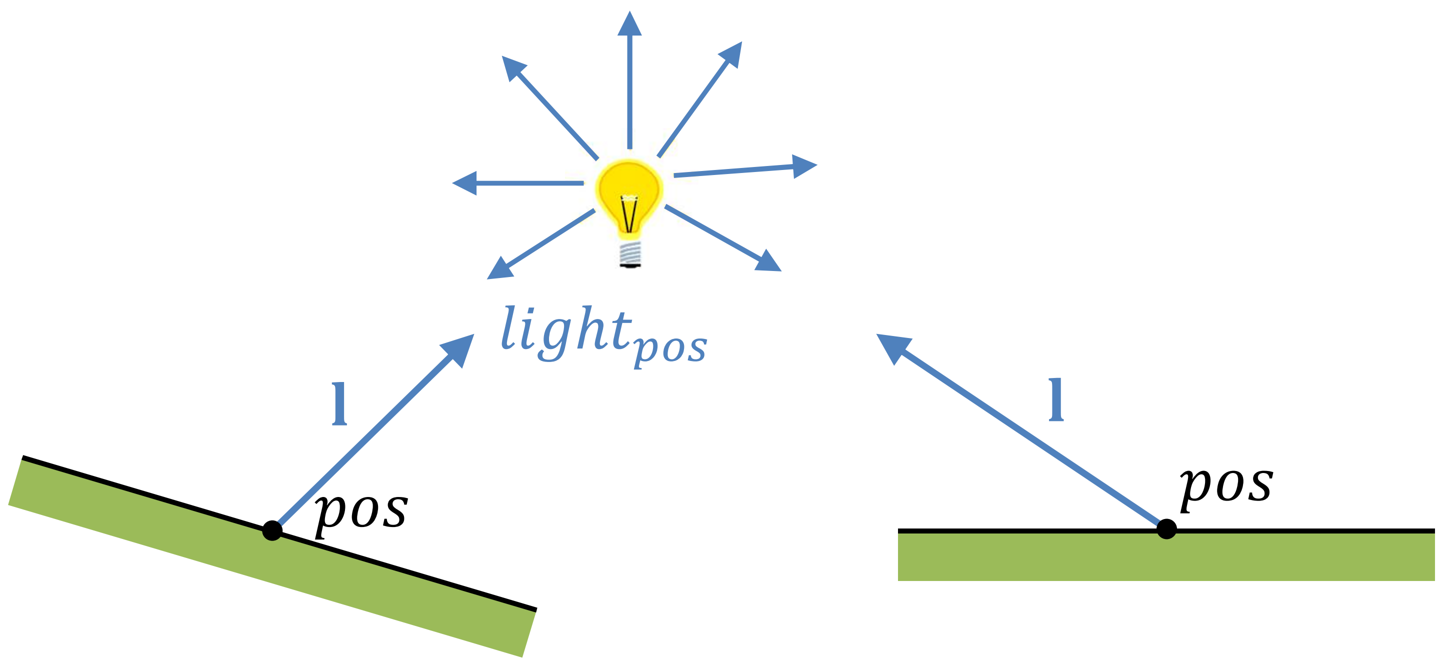
Directional Light

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



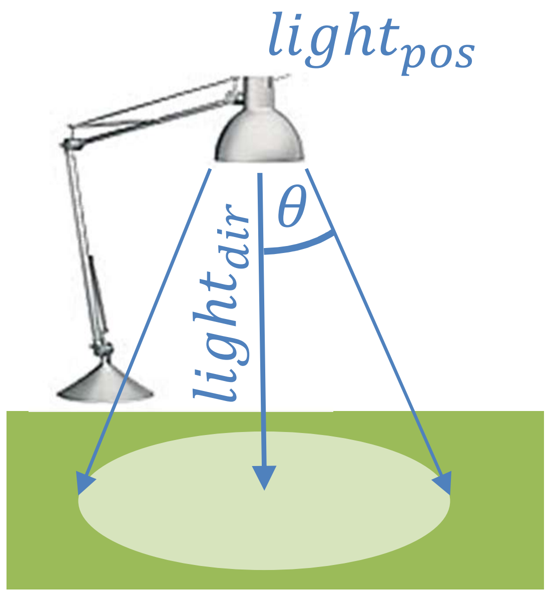
Point Light

- Has a certain position in space $\mathbf{l} = \text{normalize}(light_{pos} - pos)$



Spot Light

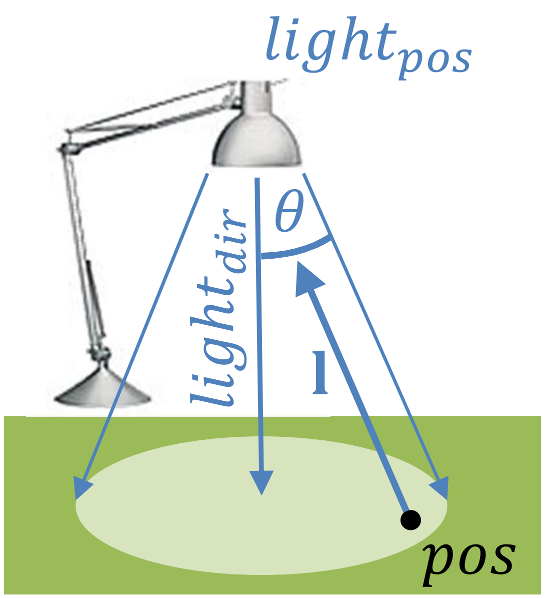
- Has a certain position and cone in space
- Cone can be specified by opening angle and central direction



Spot Light

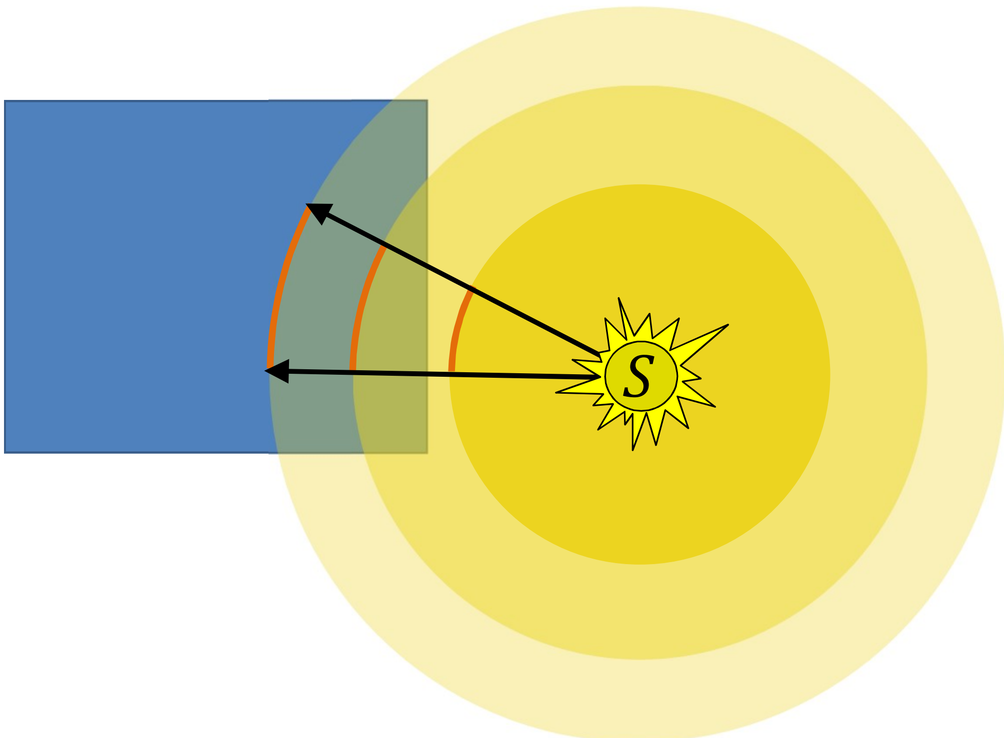
- Point is in cone iff

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



Point Light Attenuation

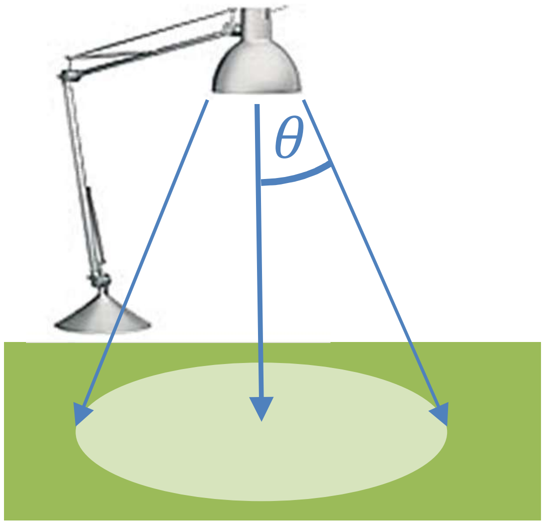
- Point light energy emitted is distributed across surface of a sphere
- Further object receive less energy per surface area
- $S = \frac{S_{intensity} * S_{color}}{4\pi r^2}$
- Inverse-square law



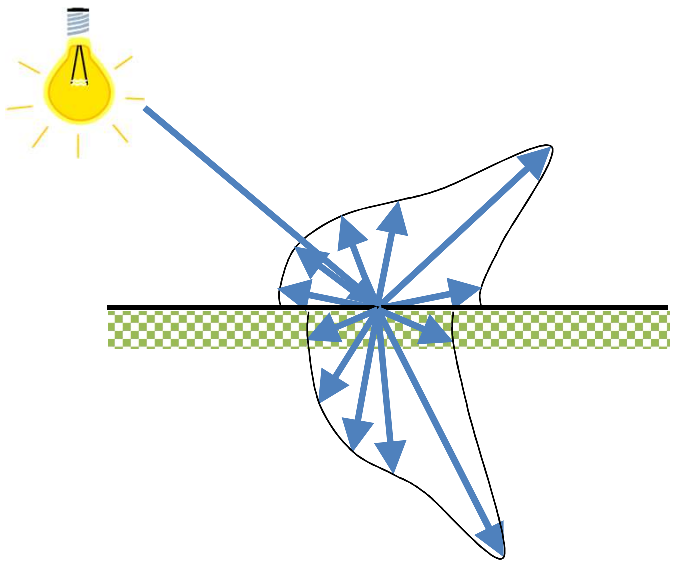
Multiple Light Sources

- Have a linear light response
- $Total\ light = \sum Light_i$

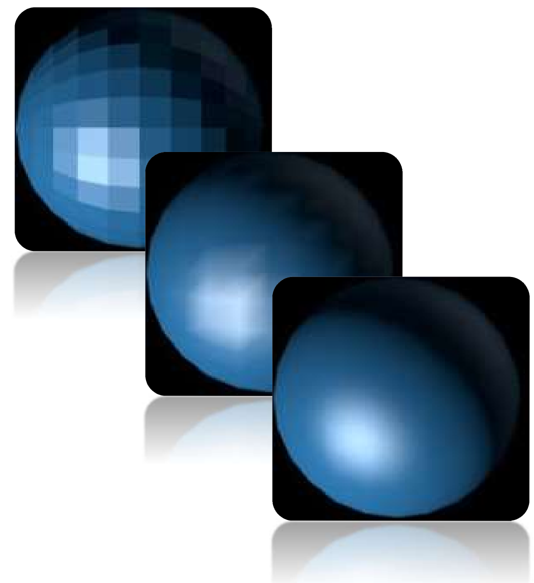
Light Sources



Reflection Models



Shading Models

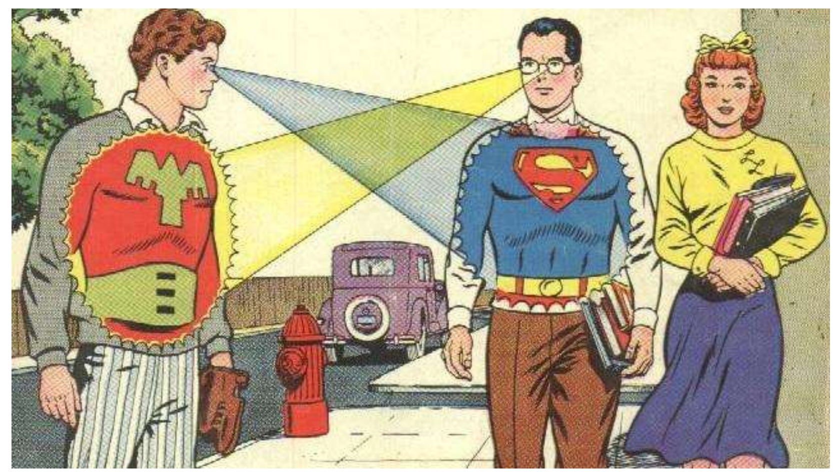


Light-Material Interaction

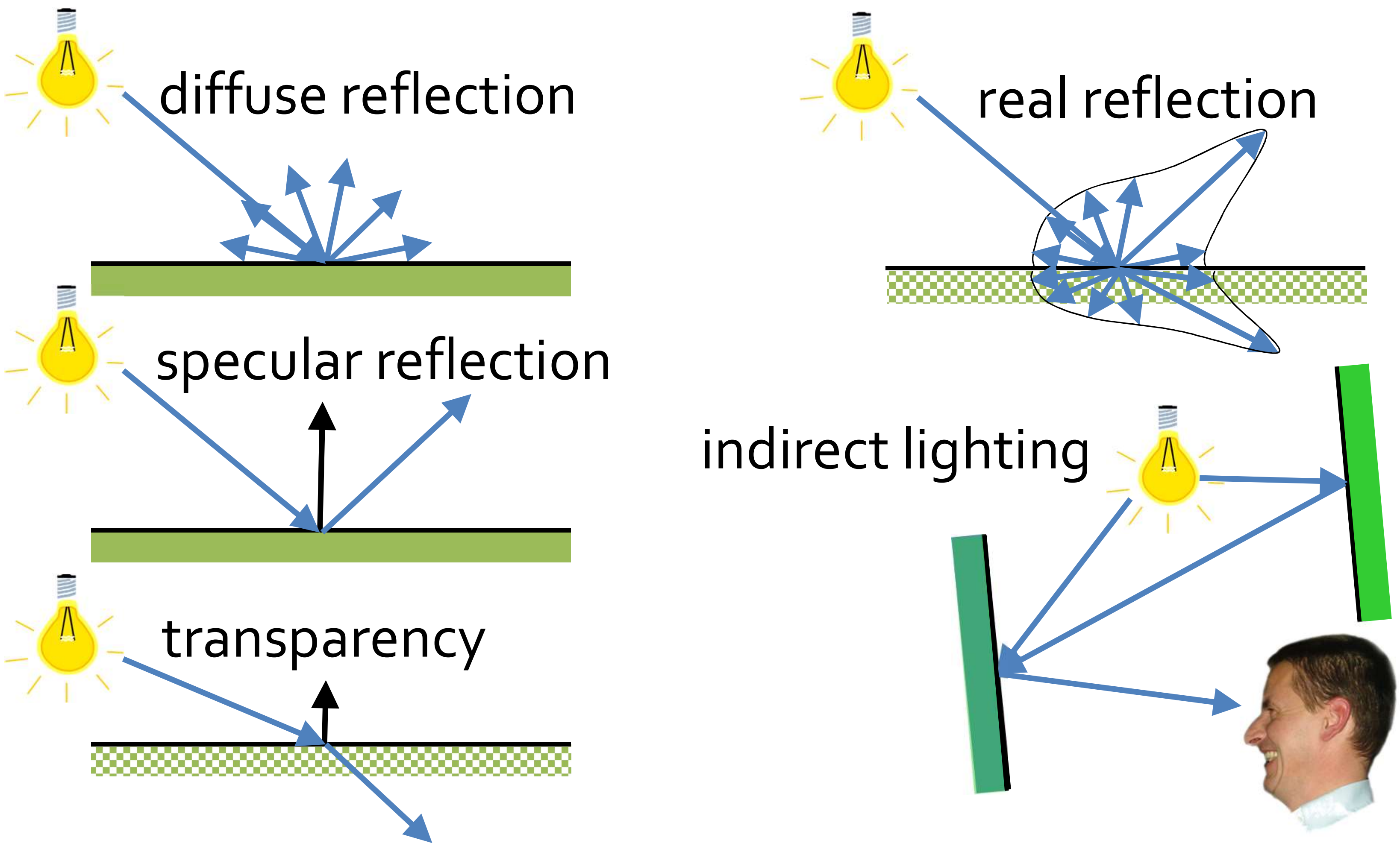
Reflection/Illumination/Lighting model

- How is light reflected by a surface?
- What is the resulting color?
- What properties can we simulate with a given model?
- What lighting effects can we create?

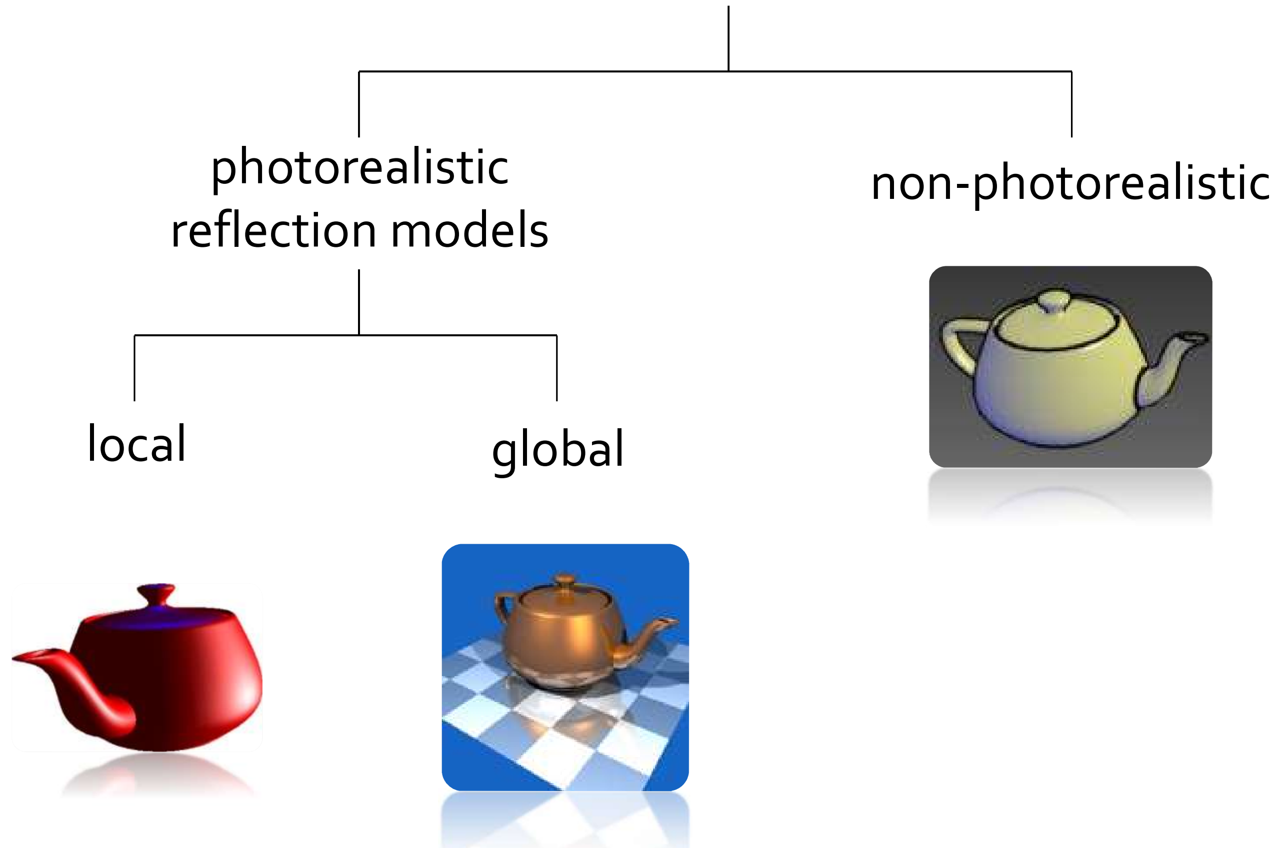
Glossy → Matte



Surface lighting effects

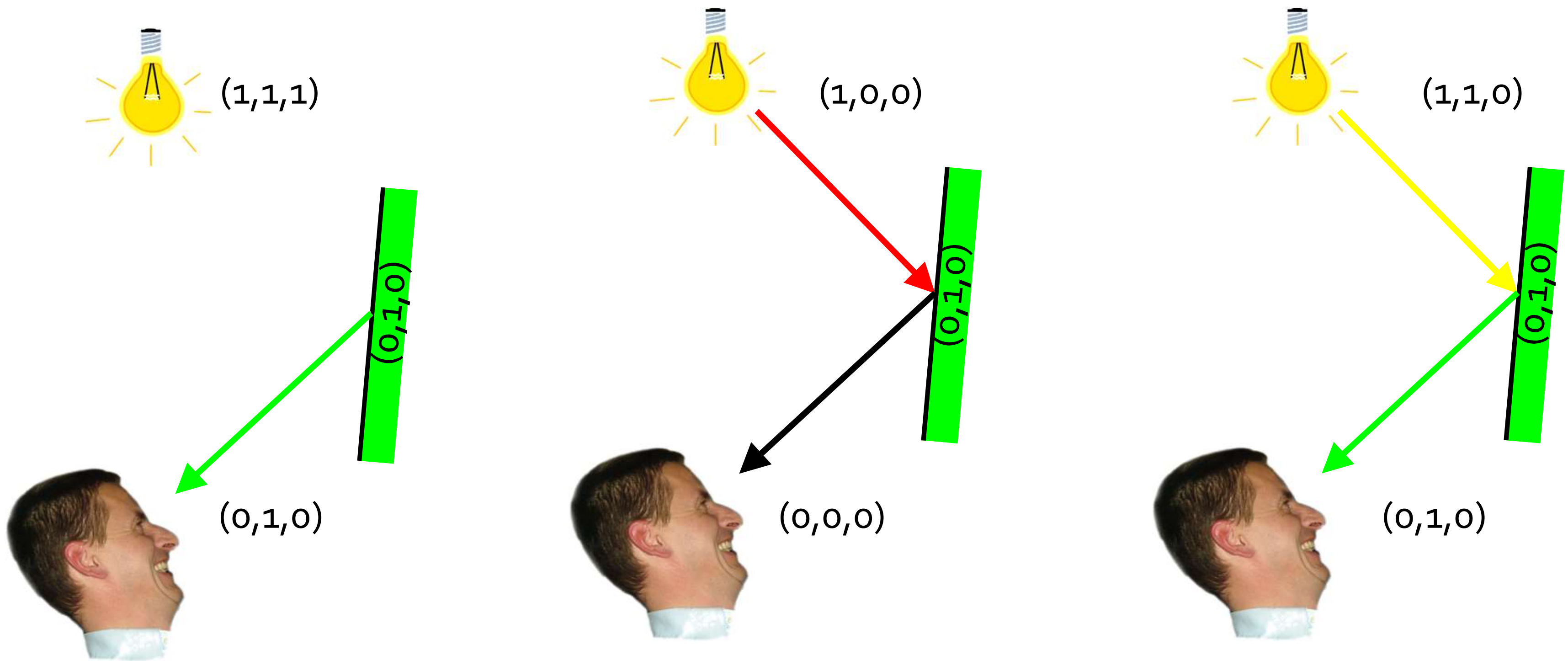


Lighting Models



Light-Material Interaction

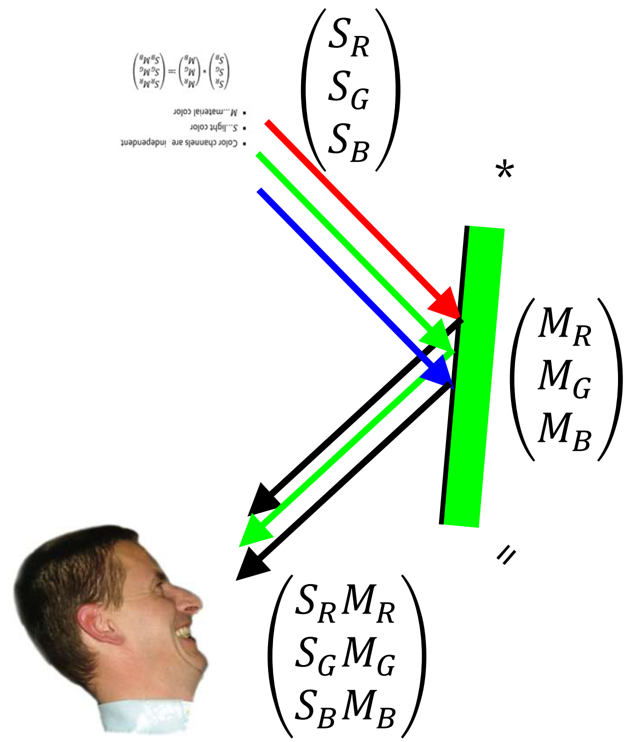
- Color channels are independent (not true for advanced phenomena, like fluorescence, phosphorescence, interference, relativistic Doppler effect)



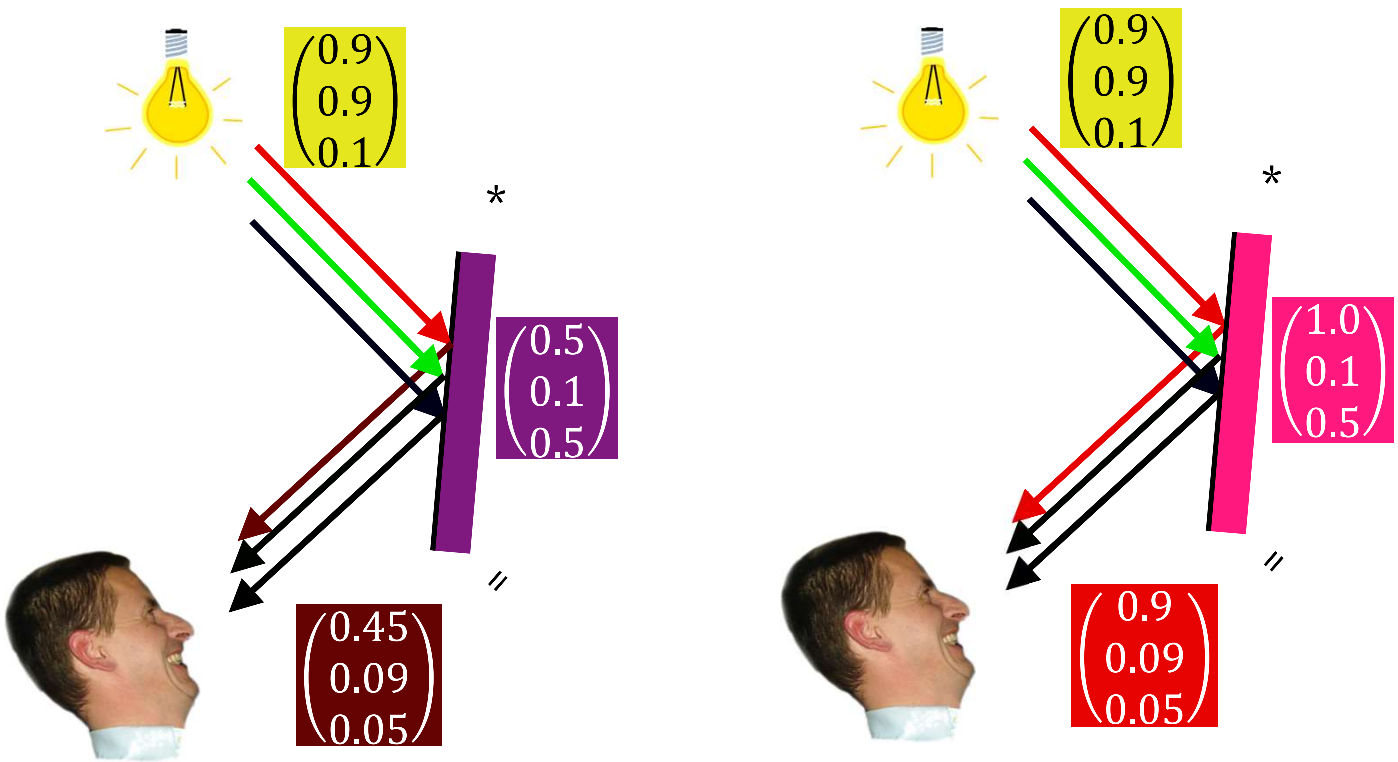
Color Multiplication – “*” Operator

- Color channels are independent
- 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



Phong Illumination Model

component … lighting model component

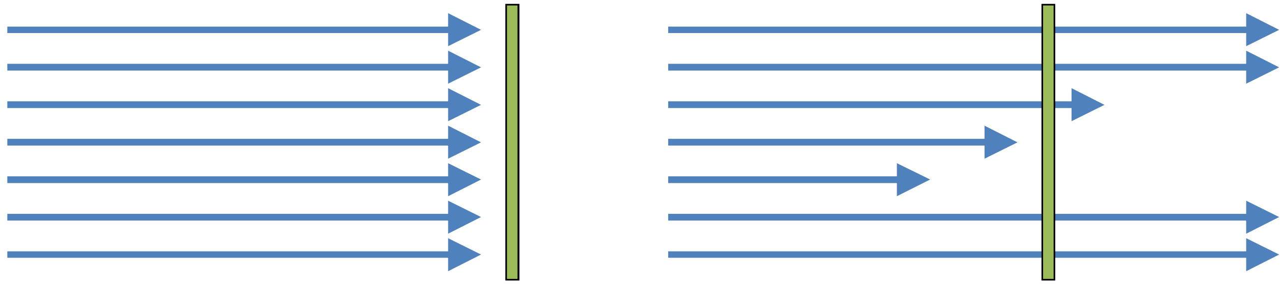
Ambient Light Reflection

- Background light
  - No direction because scattered so often
  - "Color inside shadow"
  - Approximation of global diffuse lighting effects
- S_{amb}…ambient light color
- M_{amb}…ambient material color

$$ambient = M_{amb} * S_{amb}$$

Diffuse Light Reflection

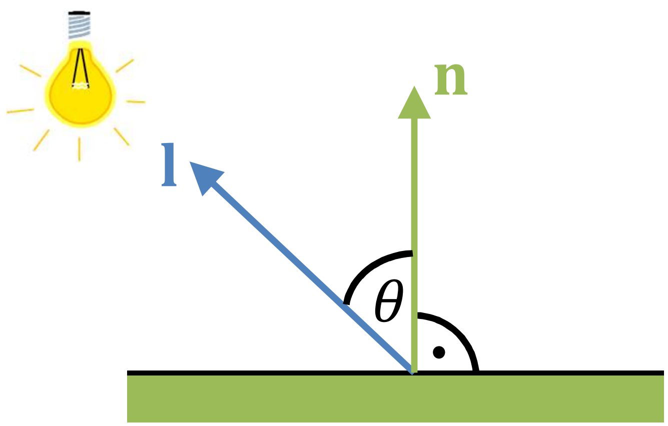
- The flatter light falls on a surface,
  the darker it will appear
- Ideal diffuse reflectors (Lambertian reflectors)
- Brightness depends on orientation of surface



ıal and
ıe light

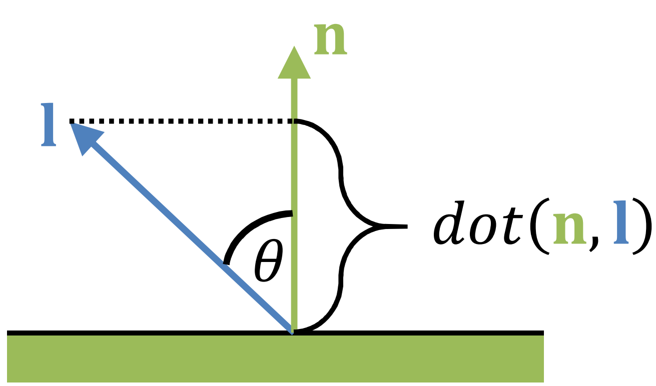
Lamberts Law

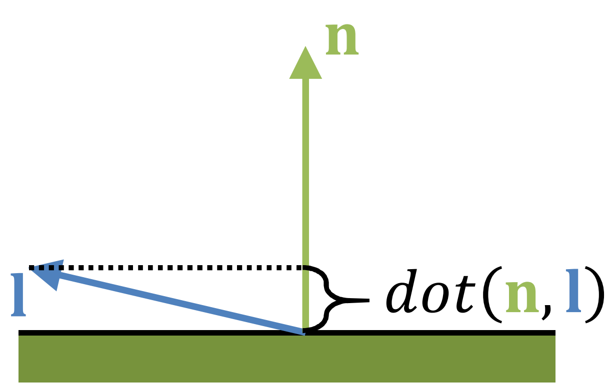
- Diffuse brightness is dependent on angle between
  n … surface normal and
  l … direction to the light



Lamberts Law

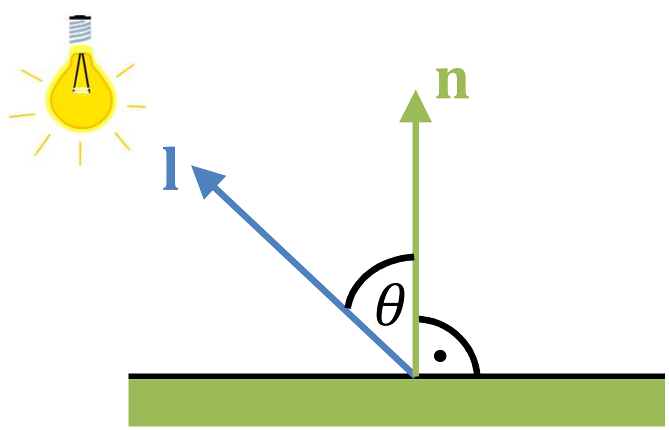
- Diffuse brightness is dependent on angle between
  $\mathbf{n}$ … surface normal and
  $\mathbf{l}$ … direction to the light
- $\cos\theta = dot(\mathbf{n}, \mathbf{l})$





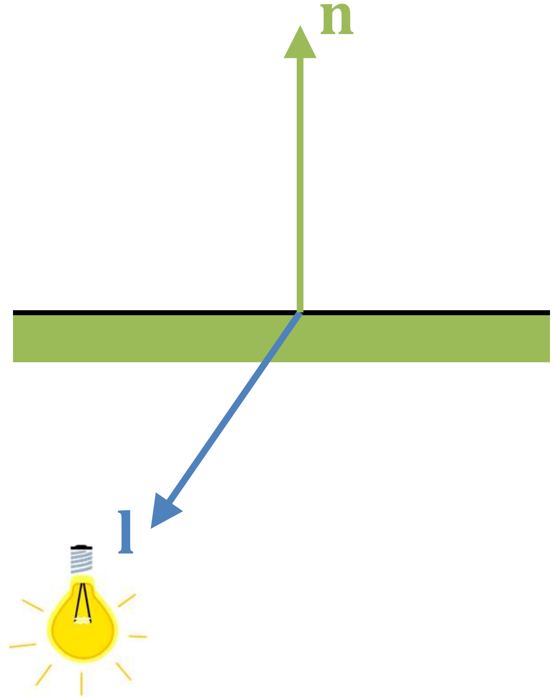
Diffuse Light Reflection

- S_{diff}...diffuse light color
- M_{diff}... diffuse material color
- $diffuse = M_{diff} * S_{diff} \cdot \mathrm{dot}(\mathbf{n}, \mathbf{l})$



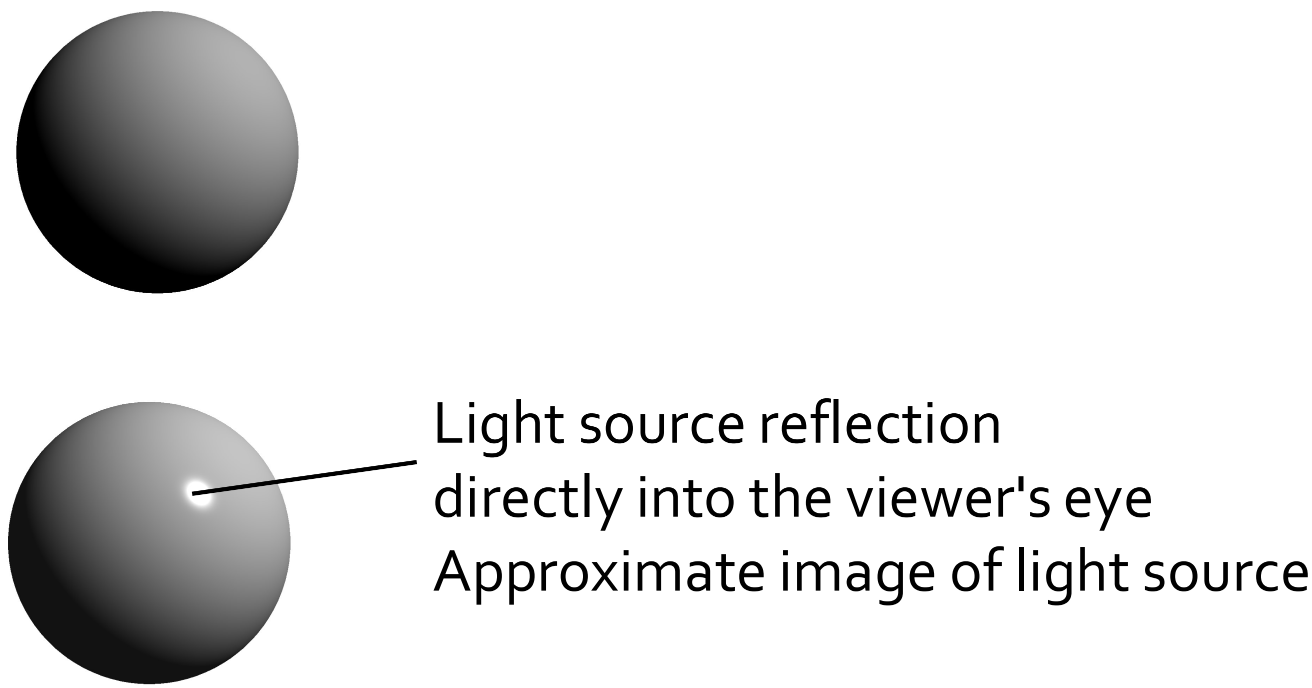
Light from Behind

- Should be ignored
- $diffuse = M_{diff} * S_{diff} \cdot \max(0, \mathrm{d}ot(\mathbf{n}, \mathbf{l}))$

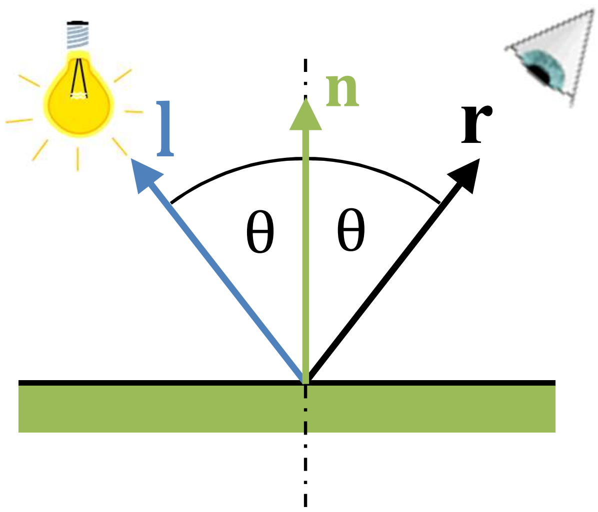


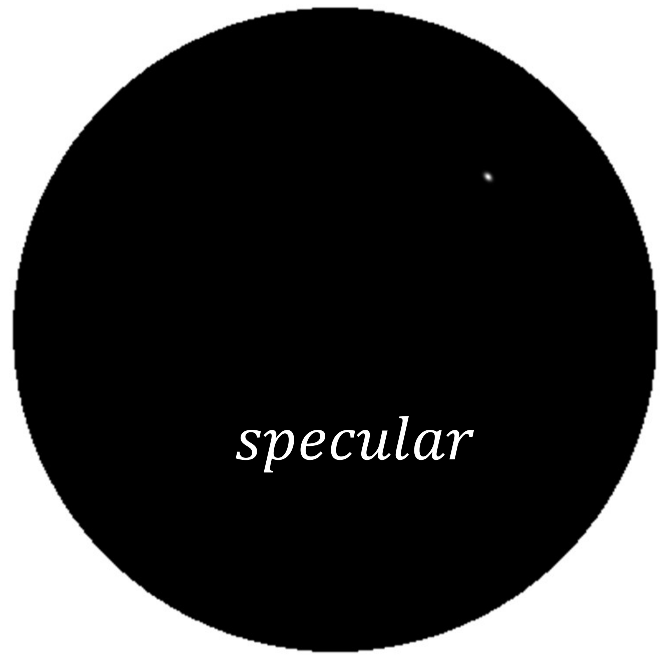
Effect of Ambient + Diffuse

Specular Highlights



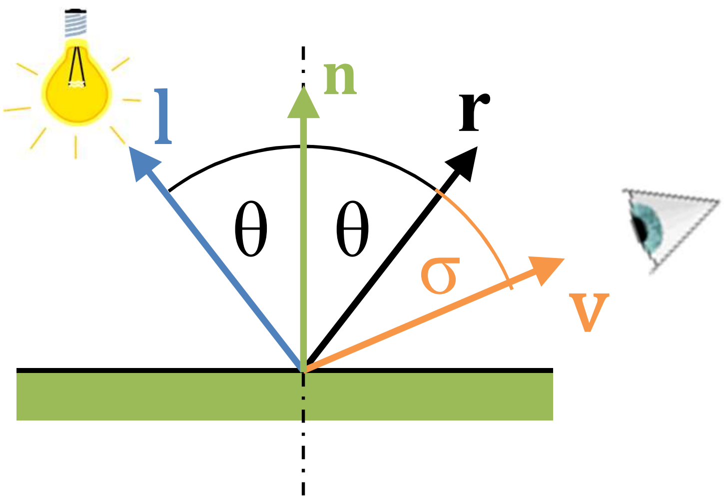
Mirror Reflection

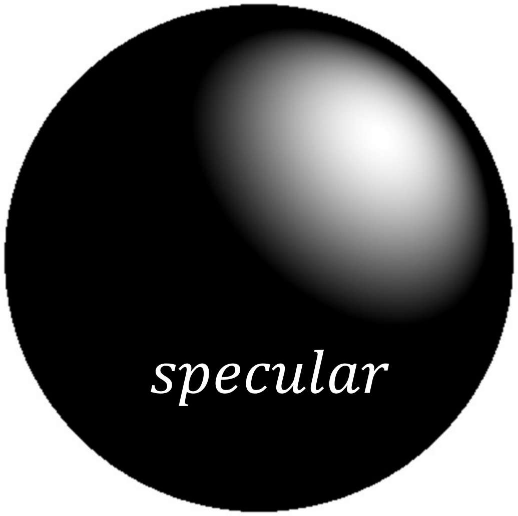




Empirical Specular Reflection

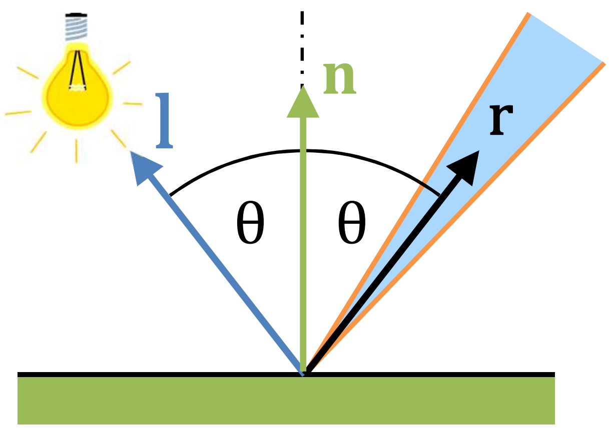
- S_{spec}… specular light color
- M_{spec}… specular material color
- $specular = M_{\text{spec}} * S_{\text{spec}} \cdot \cos \sigma$



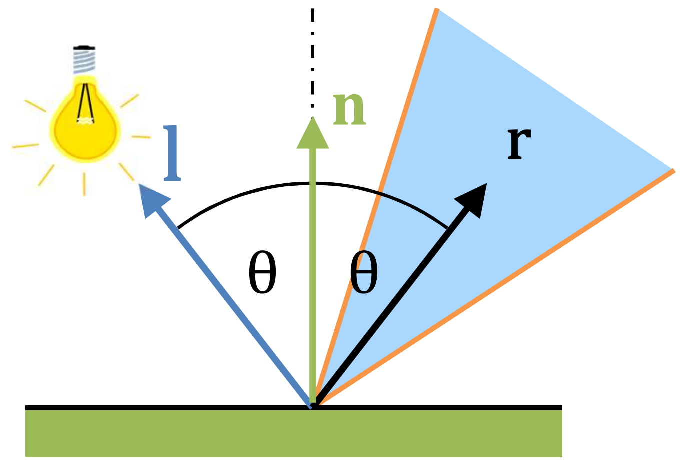


Specular Reflection Coefficient – p = Shininess

- $specular = M_{\mathrm{spec}} * S_{\mathrm{spec}} \cdot \cos^{\mathrm{p}} \sigma$

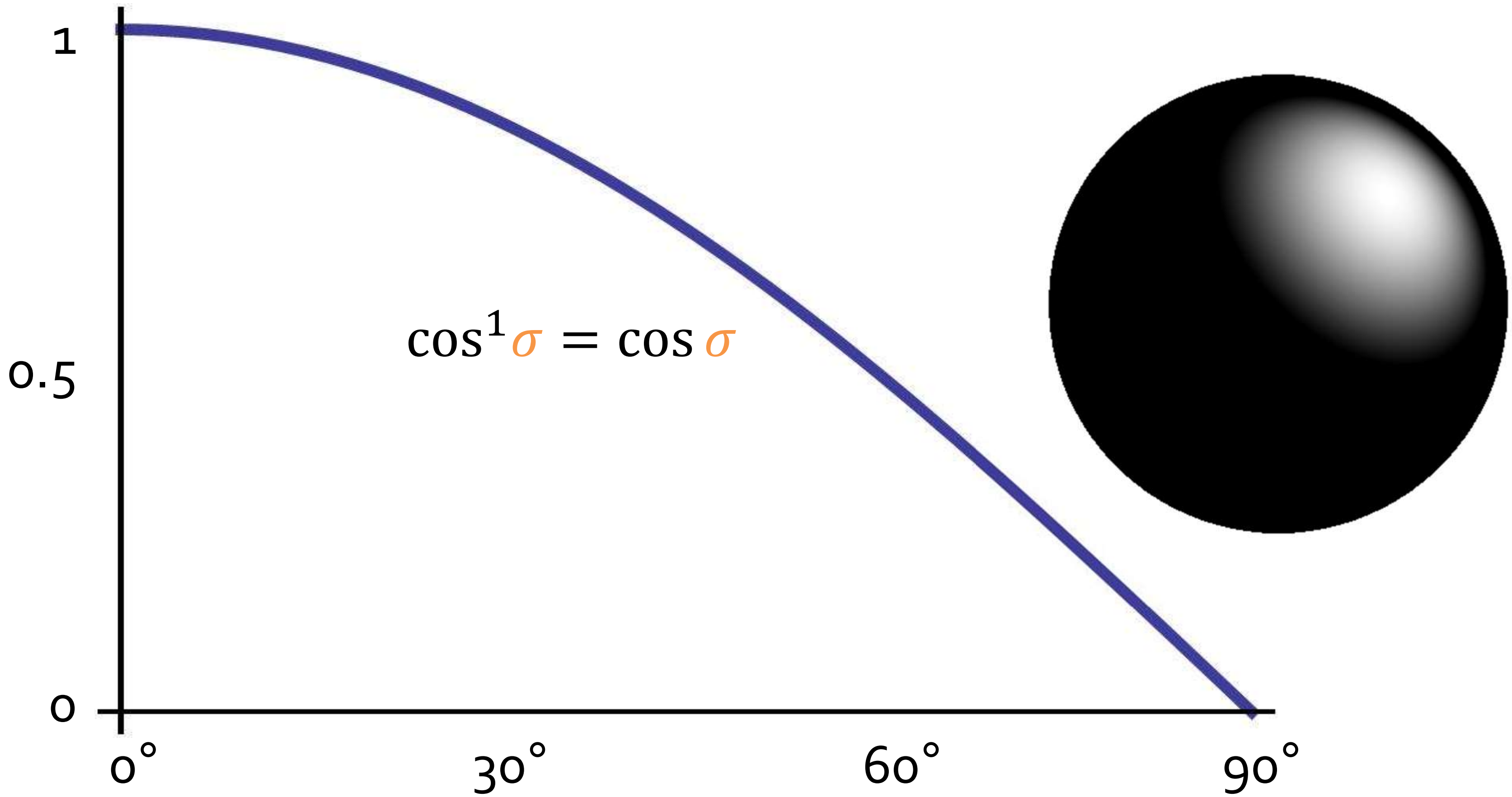


shiny surface
(large p)

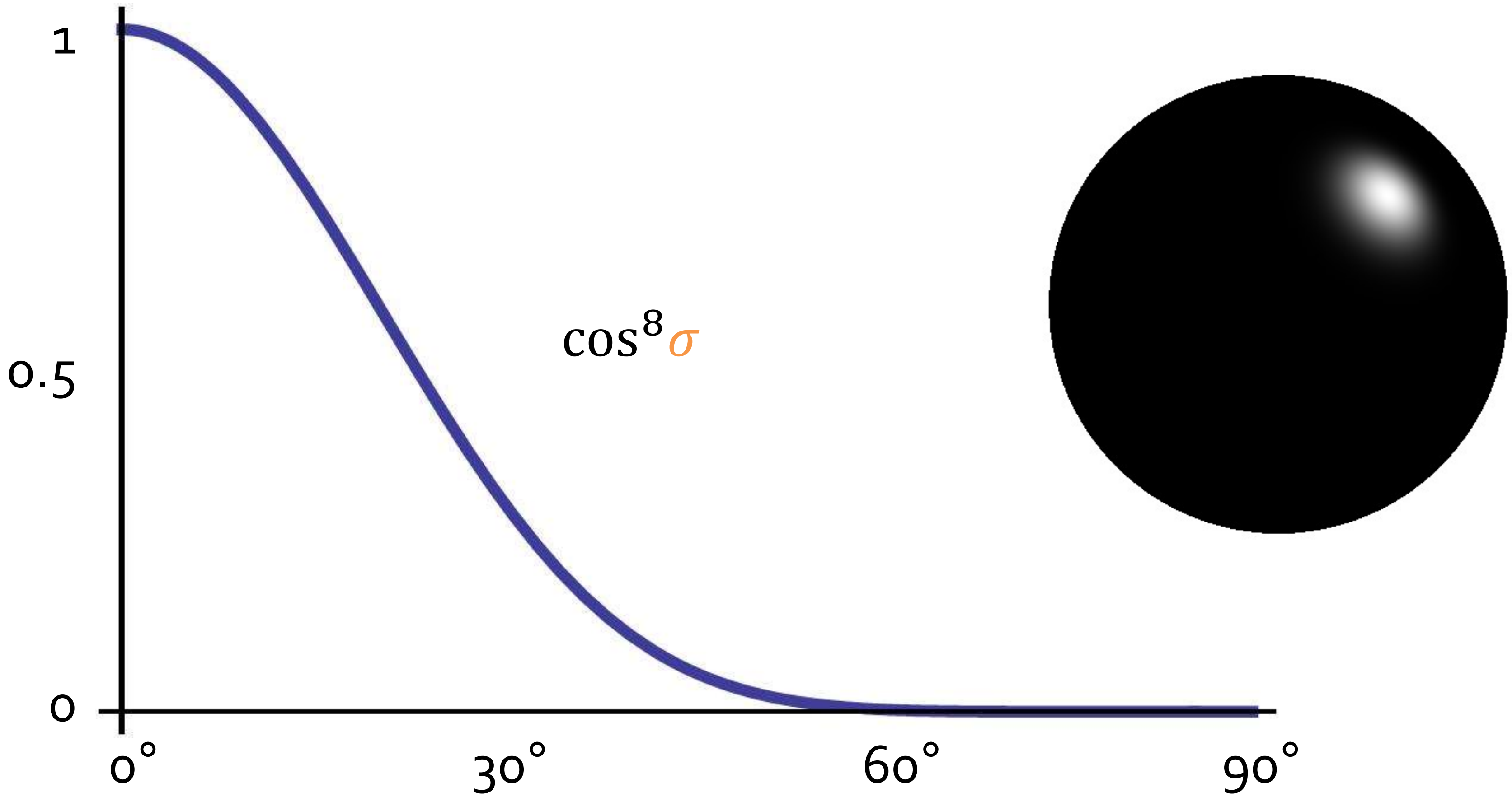


dull surface
(small p)

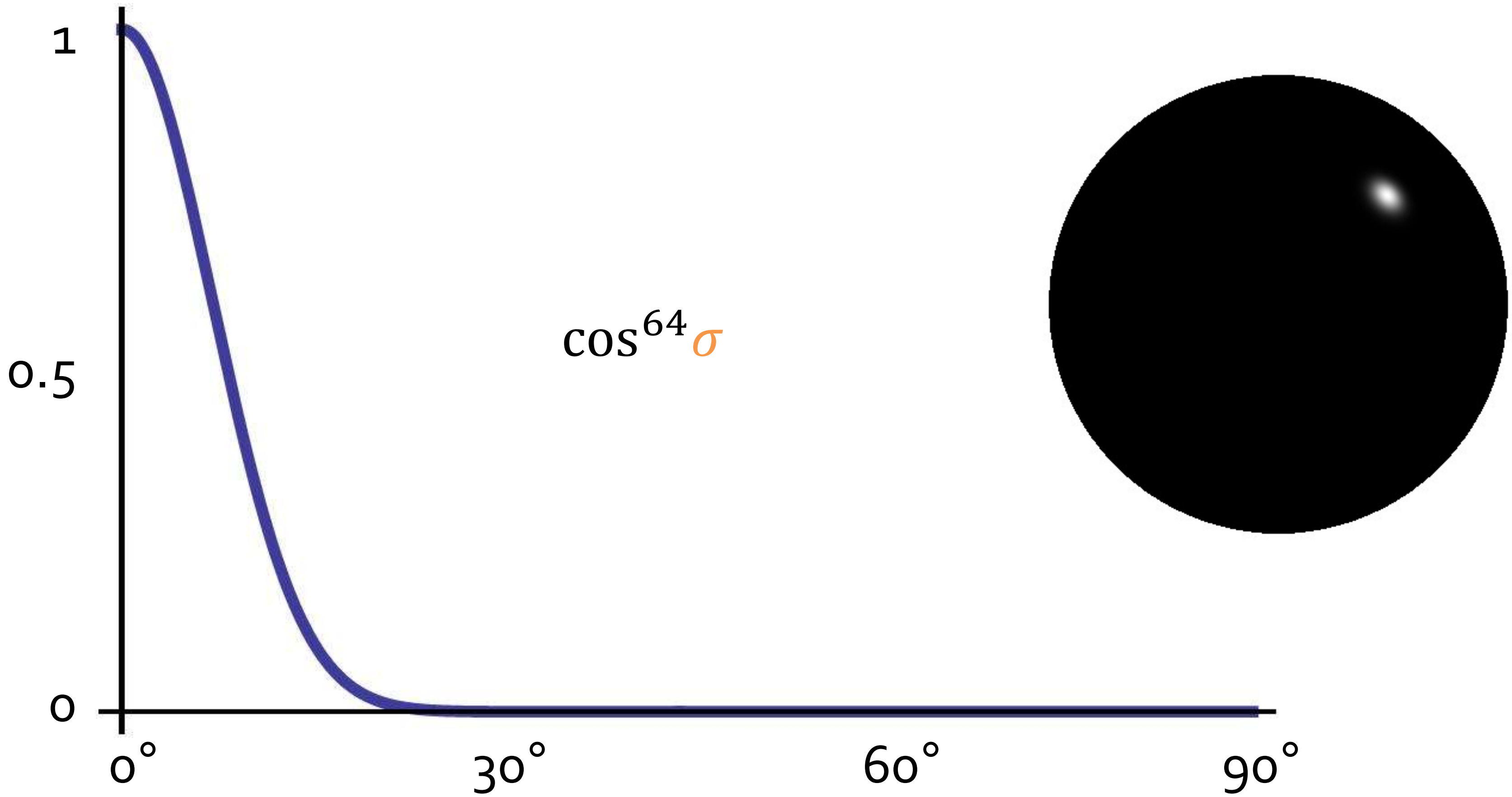
Specular Reflection Coefficient $\mathrm{p} = 1$



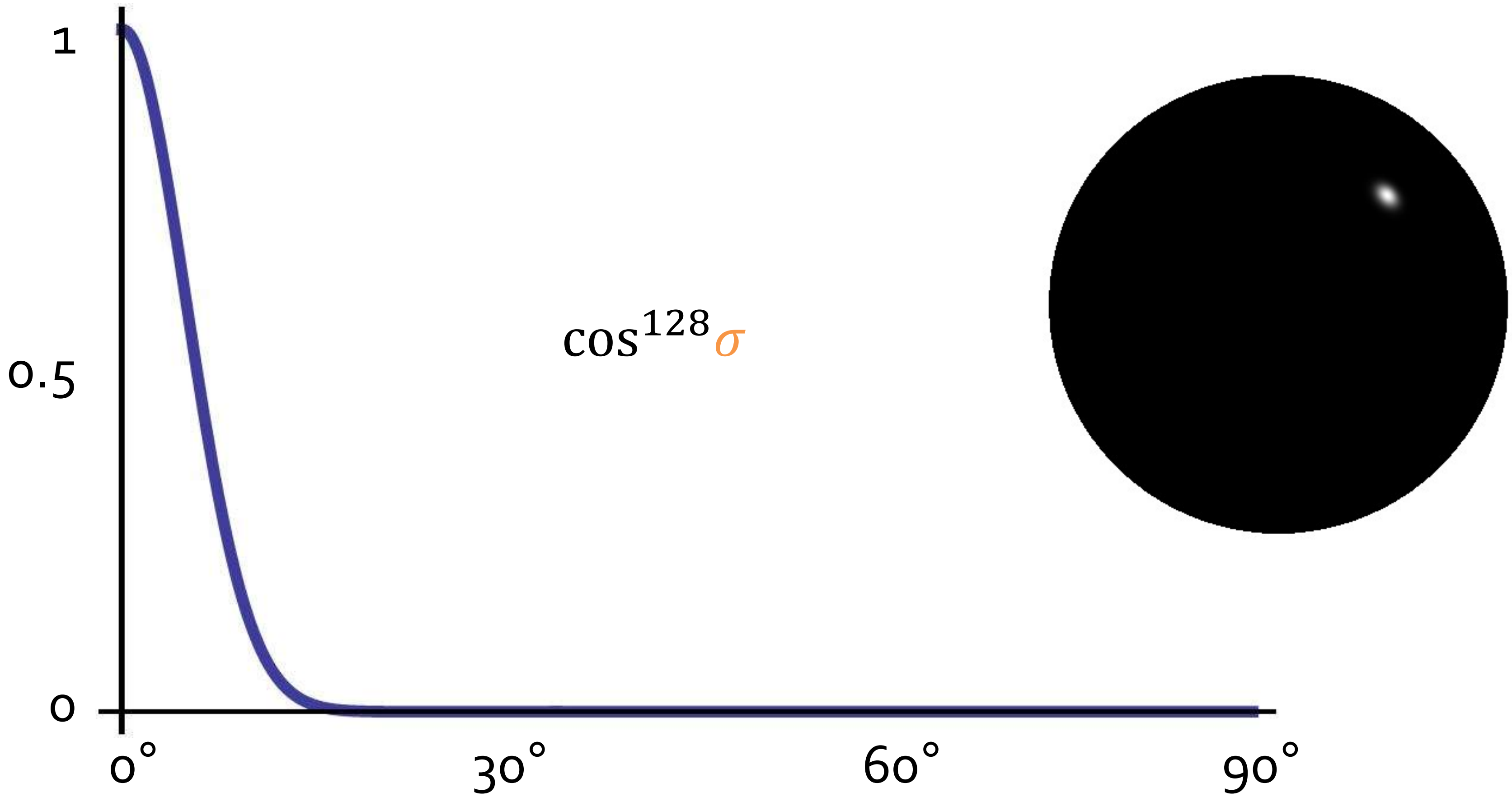
Specular Reflection Coefficient $p = 8$



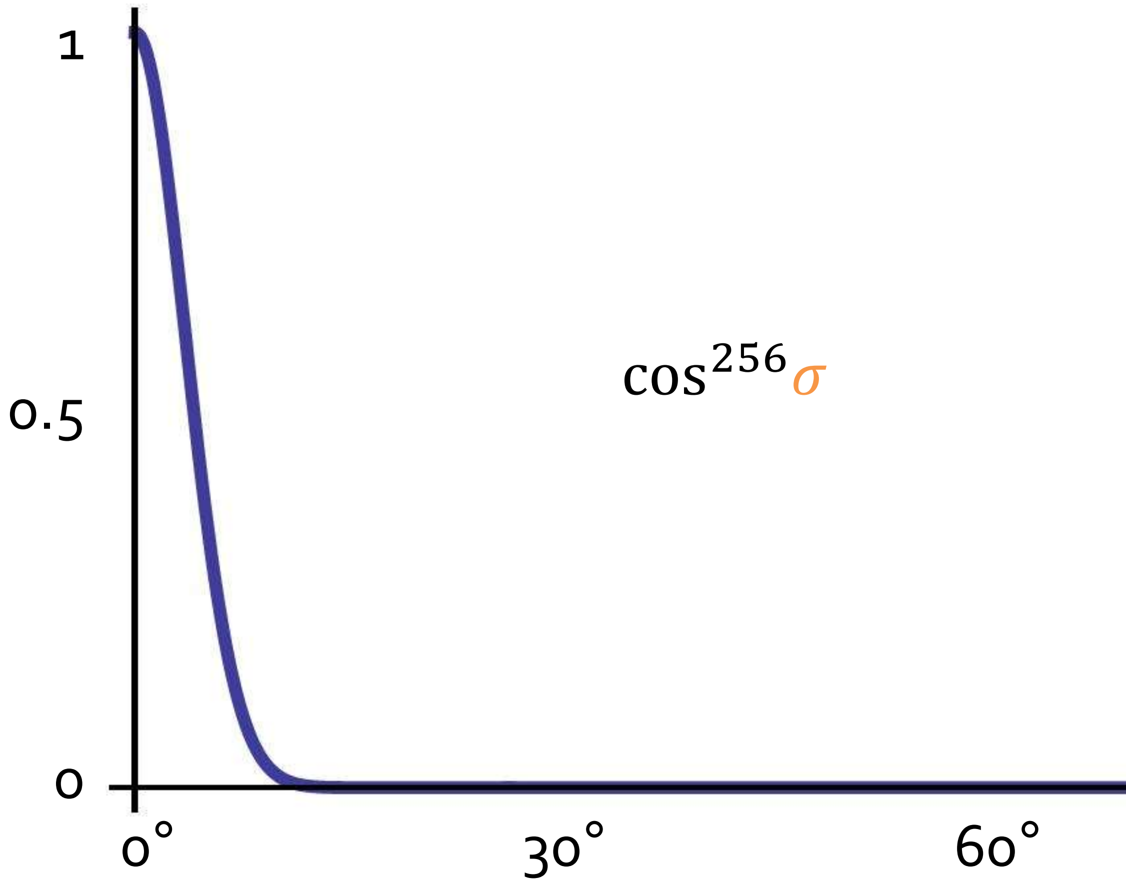
Specular Reflection Coefficient $p = 64$

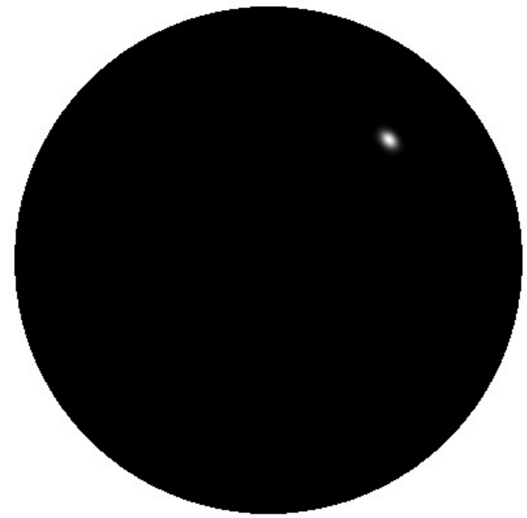


Specular Reflection Coefficient $\mathrm{p} = 128$

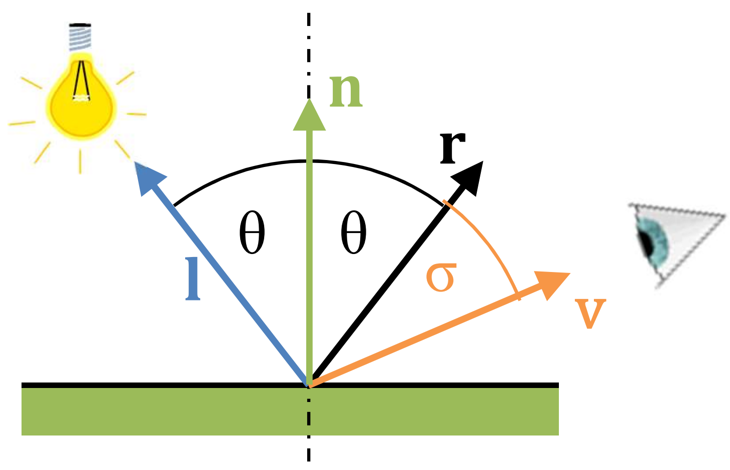


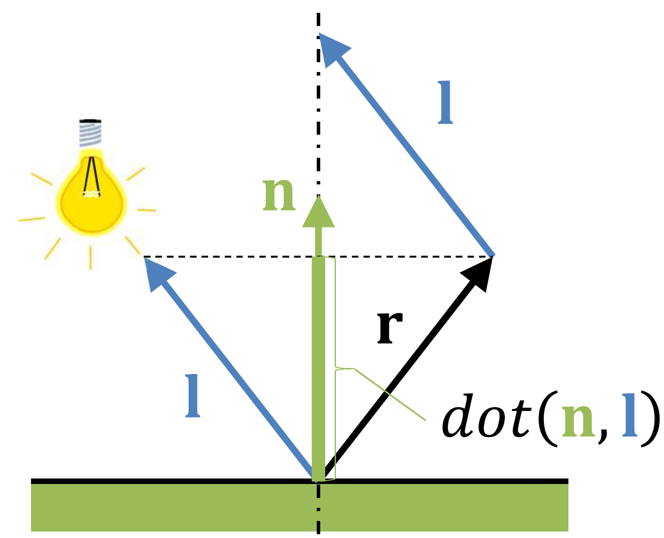
Specular Reflection Coefficient $\mathrm{p} = 256$





Specular Reflection Calculation





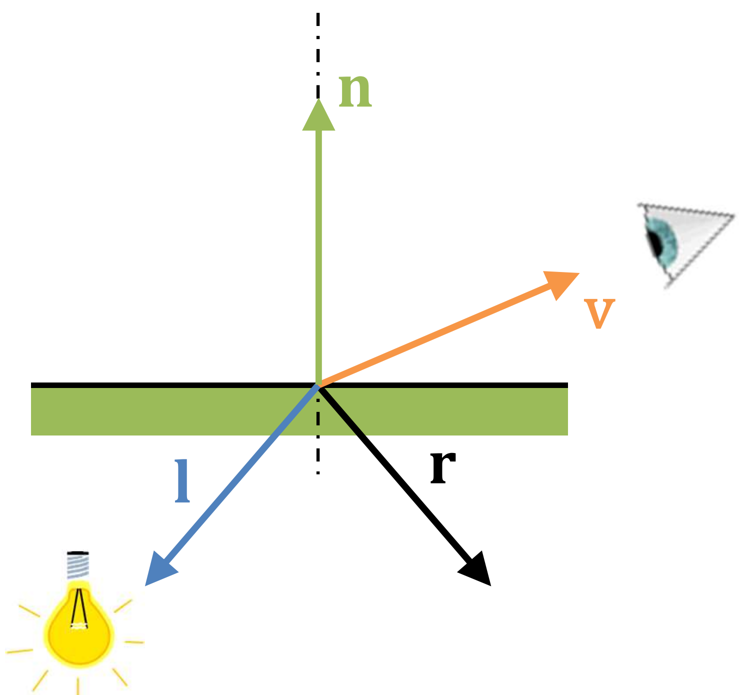
$$specular = M_{\text{spec}} * S_{\text{spec}} \cdot \cos^{\text{p}} \sigma$$
$$= M_{\text{spec}} * S_{\text{spec}} \cdot \mathrm{d}ot(\mathbf{r}, \mathbf{v})^{\text{p}}$$

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

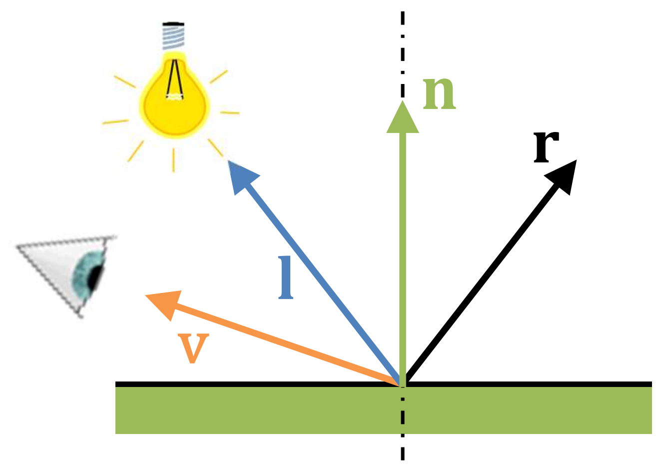
$$\mathbf{r} = 2 \cdot dot(\mathbf{n}, \mathbf{l}) \cdot \mathbf{n} - \mathbf{l}$$

Do not apply Specular when

$0 > dot(\mathbf{n}, \mathbf{l})$



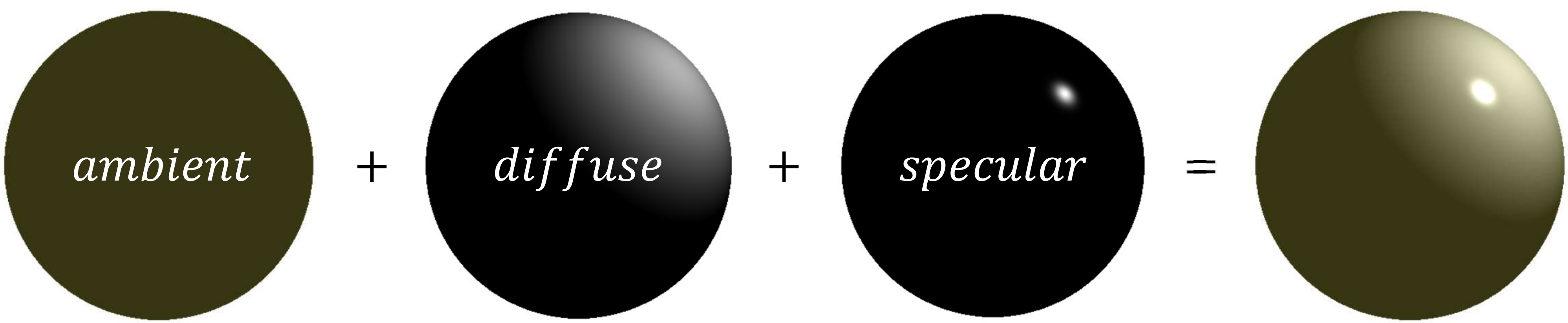
$0 > dot(\mathbf{r}, \mathbf{v})$



Phong Illumination Model

$$ambient + diffuse + specular =$$

$$= M_{amb} * S_{amb} + M_{diff} * S_{diff} \cdot \mathrm{d}ot(\mathbf{n}, \mathbf{l}) + M_{\mathrm{spec}} * S_{\mathrm{spec}} \cdot \mathrm{d}ot(\mathbf{r}, \mathbf{v})^{\mathrm{p}}$$



Lighting Models

photorealistic reflection models

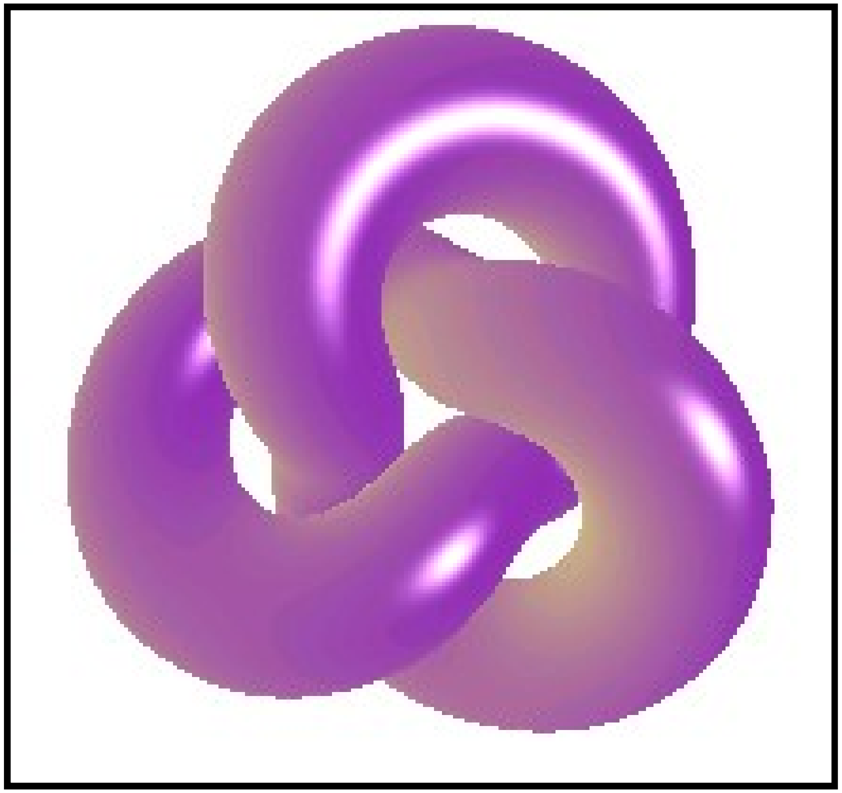
local

global

non-photorealistic

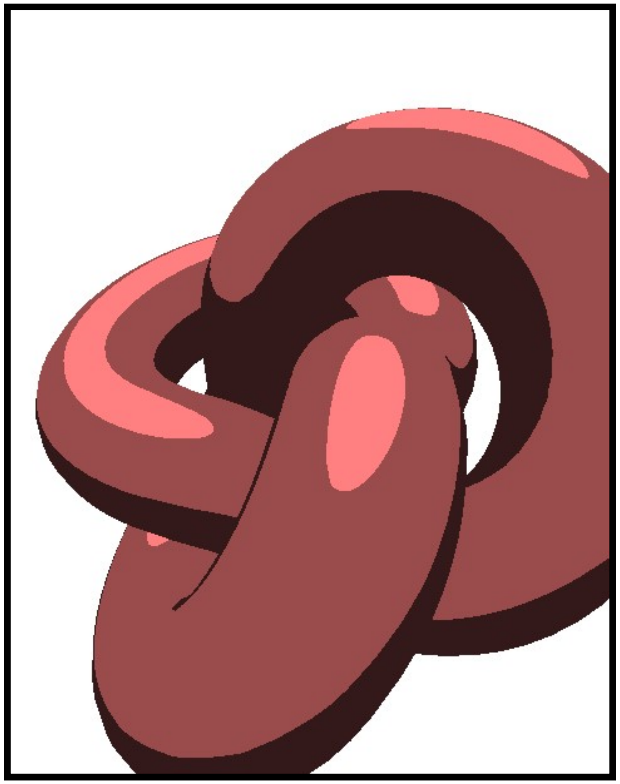
Gooch

- Blend between a cool and a warm color



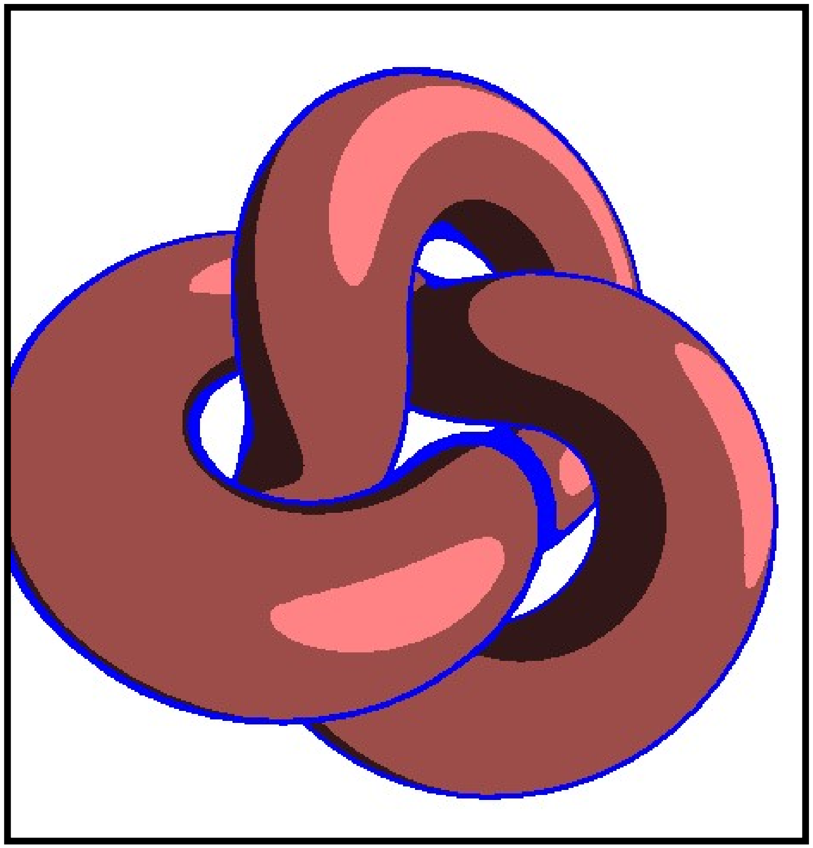
Toon shading

- Discrete color steps for diffuse

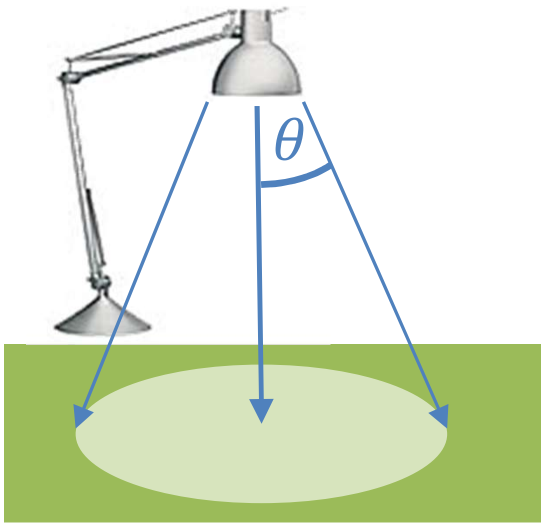


Cel shading

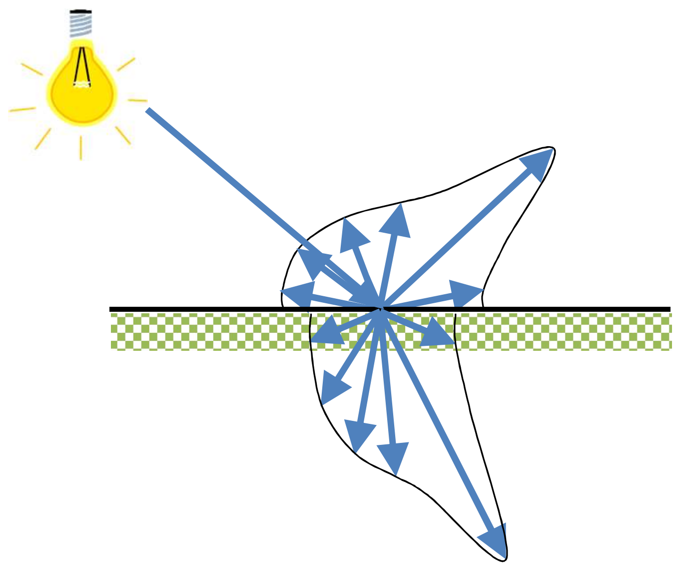
- Detect edges and color them



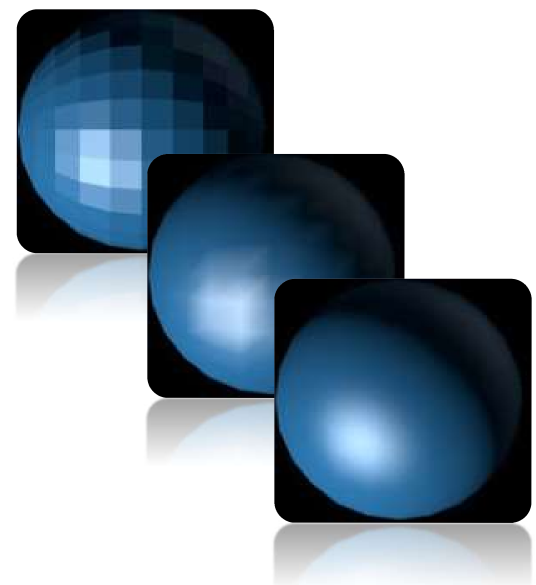
Light Sources



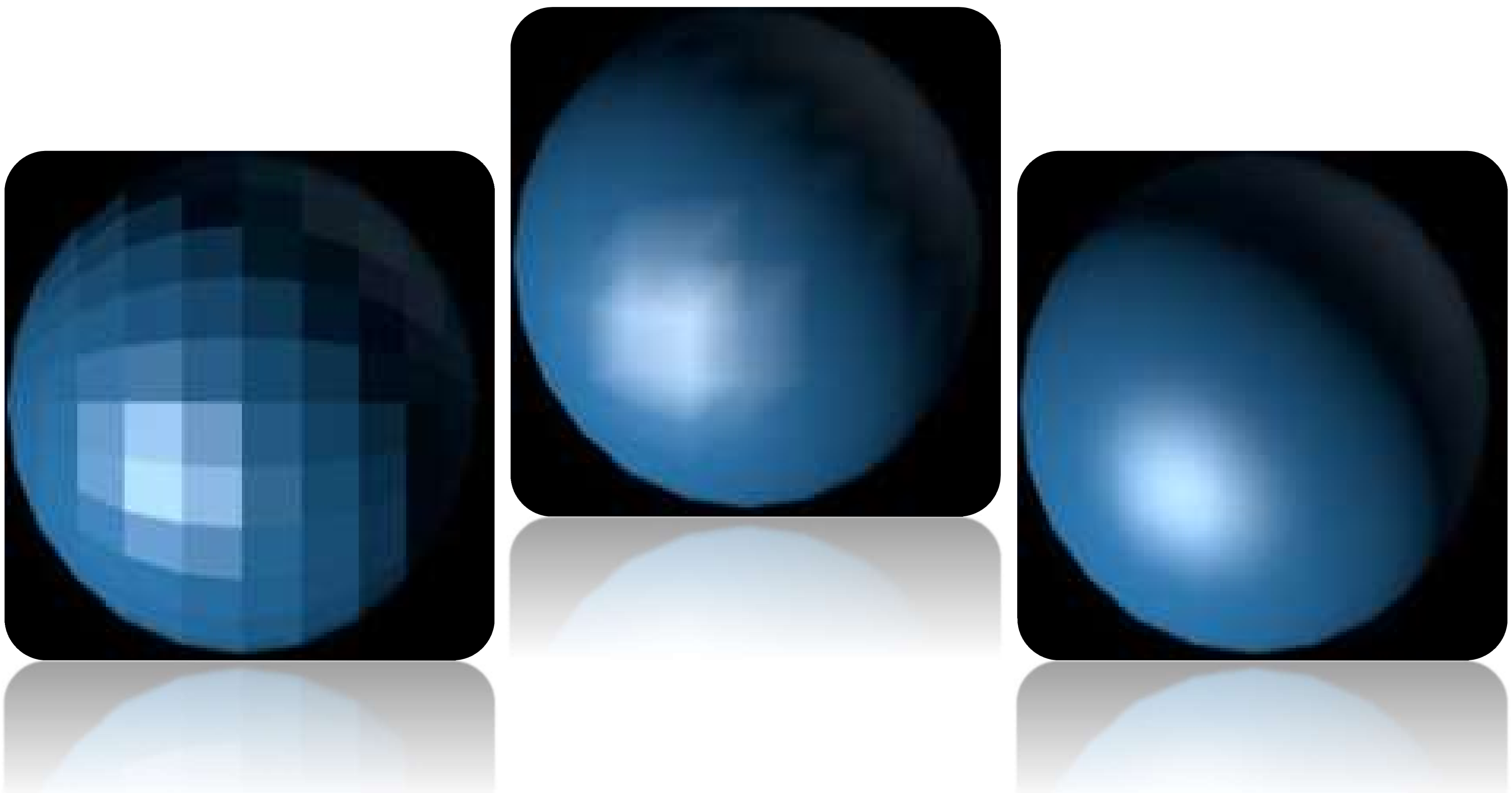
Reflection Models



Shading Models



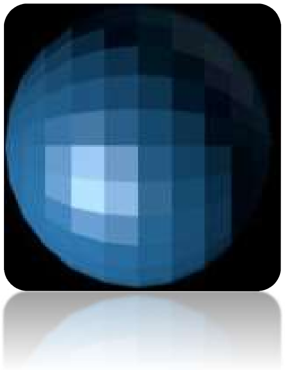
Shading Models



Shading model

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

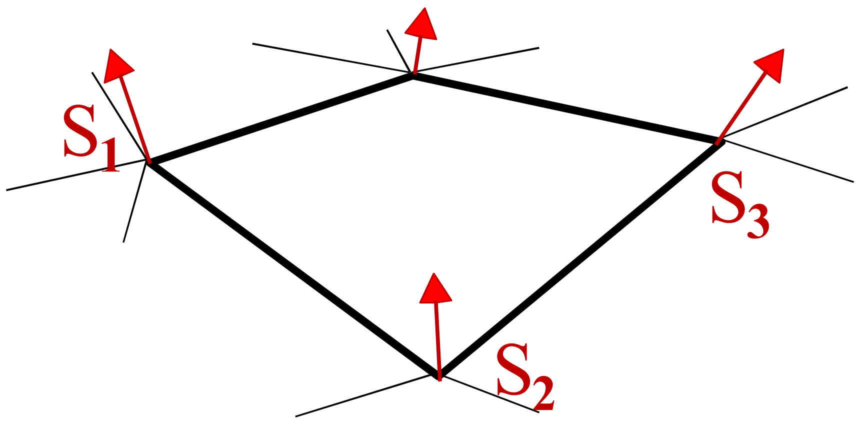
Flat-shading



- 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

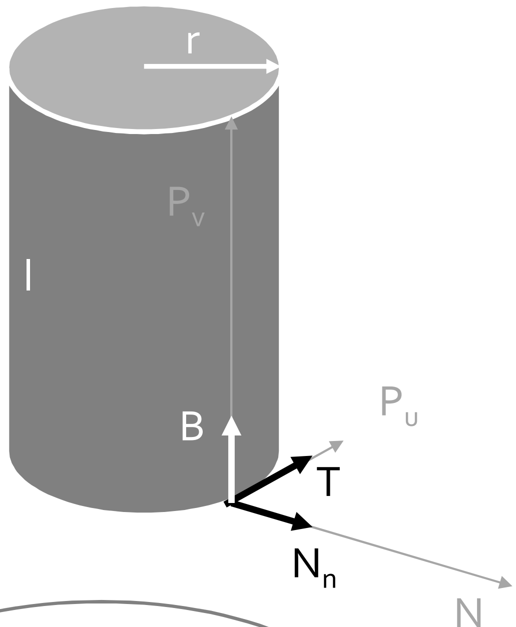
Gouraud shading

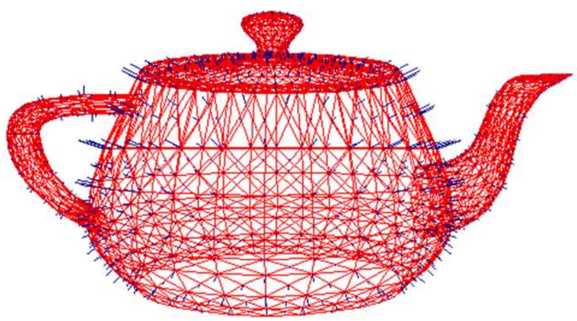
1. Calculate the normal vector for each vertex
2. Calculate the intensitity for each vertex

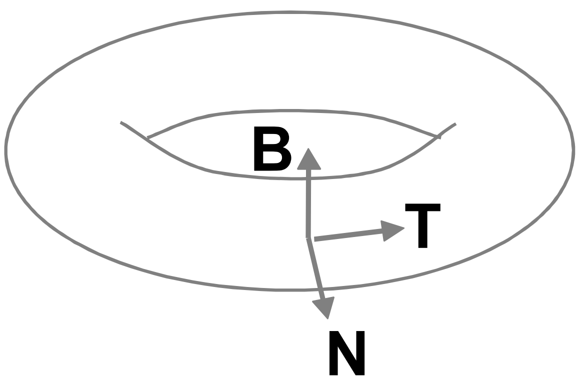


Vertex Normals

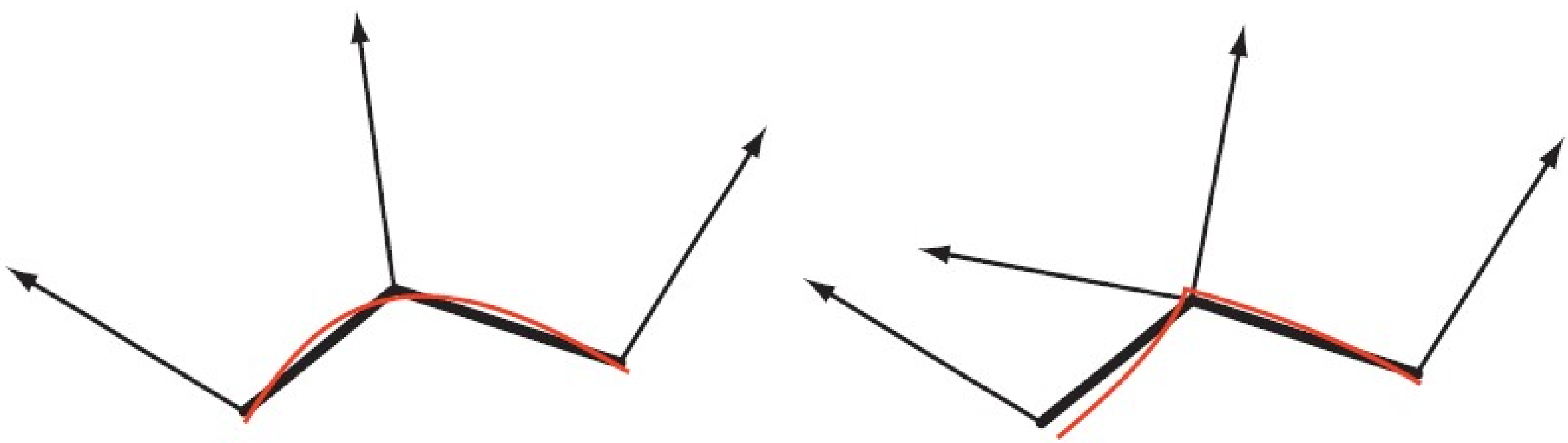
- Vertex normals may be
  - Provided with the model
    - Artist
    - 3d program
  - Computed from first principles
    - Mathematic description of model





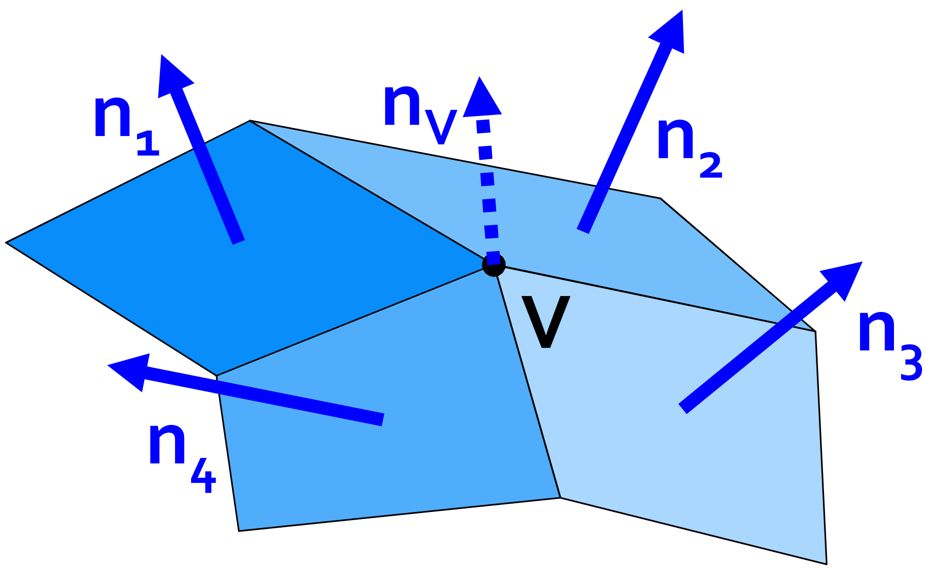


Sharp Edges and Normals



Vertex Normals

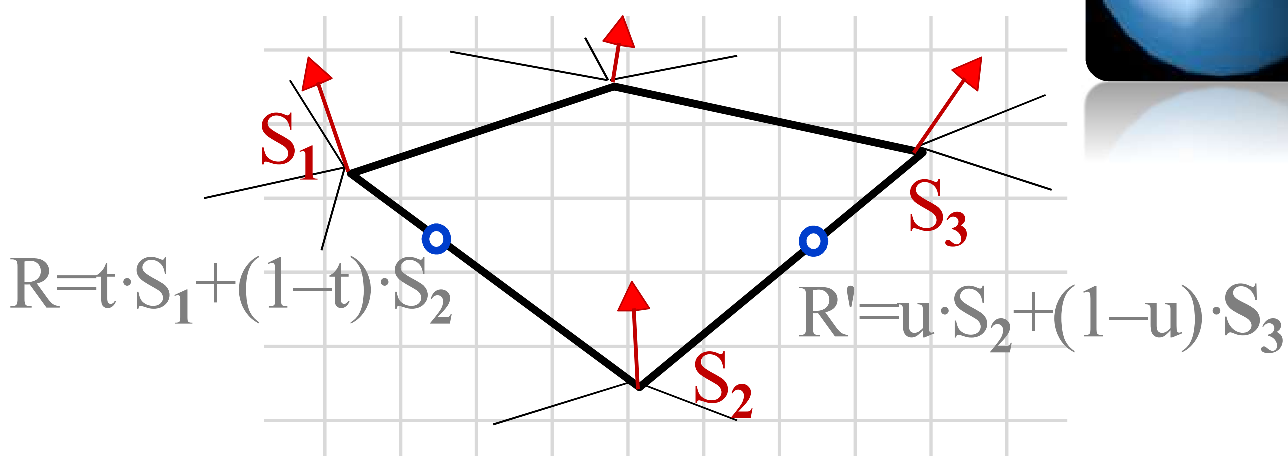
- Approximated by averaging the normals of the facets that share the vertex



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

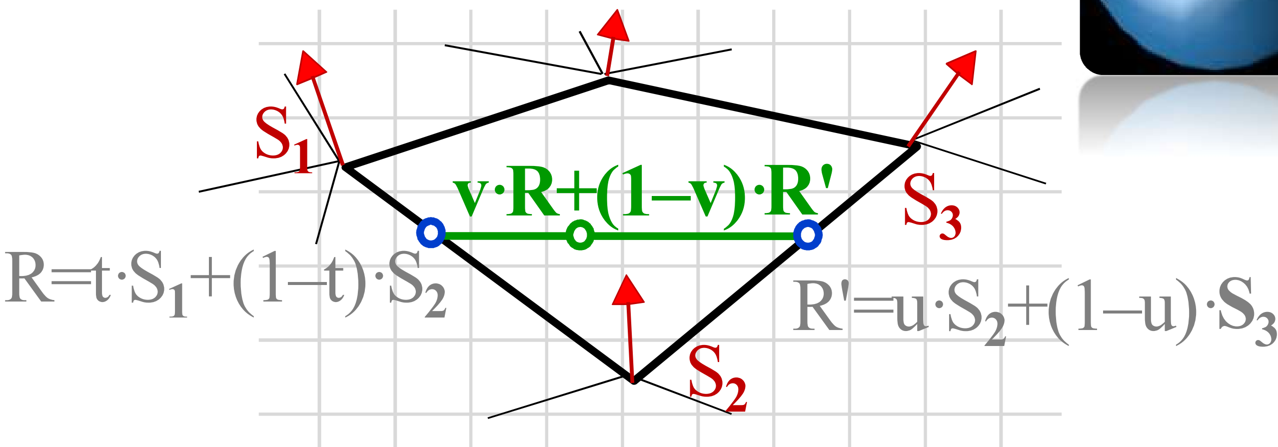
Gouraud shading

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



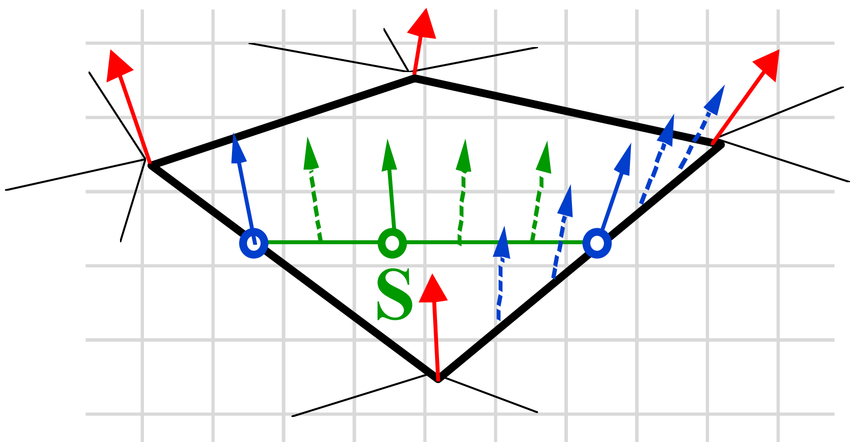
Gouraud shading

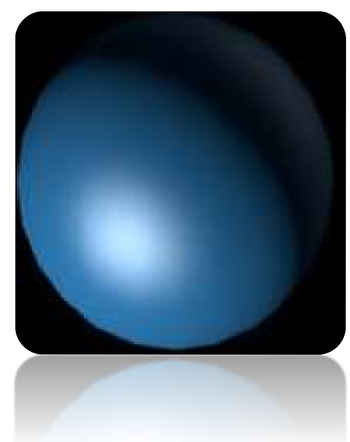
1. Calculate the normal vector for each vertex
2. Calculate the intensitity for each vertex
3. Color interpolation along edges
4. Color interpolation along scanline



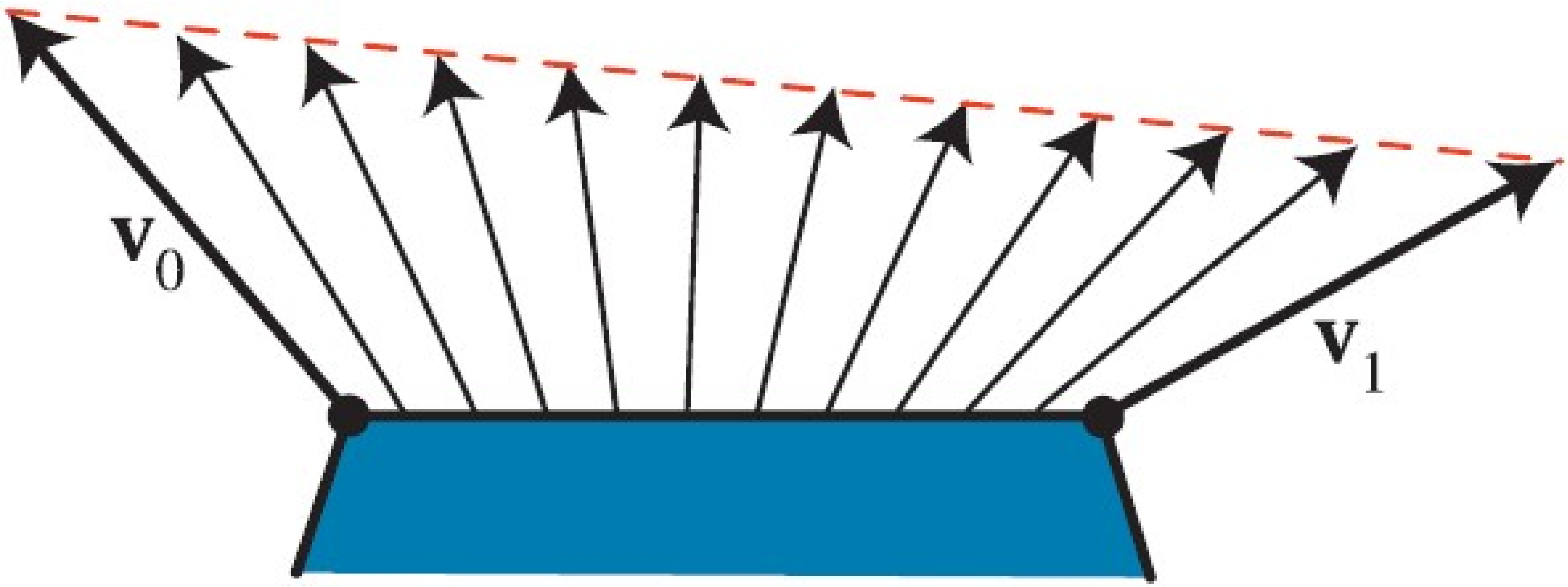
Phong Shading Model

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

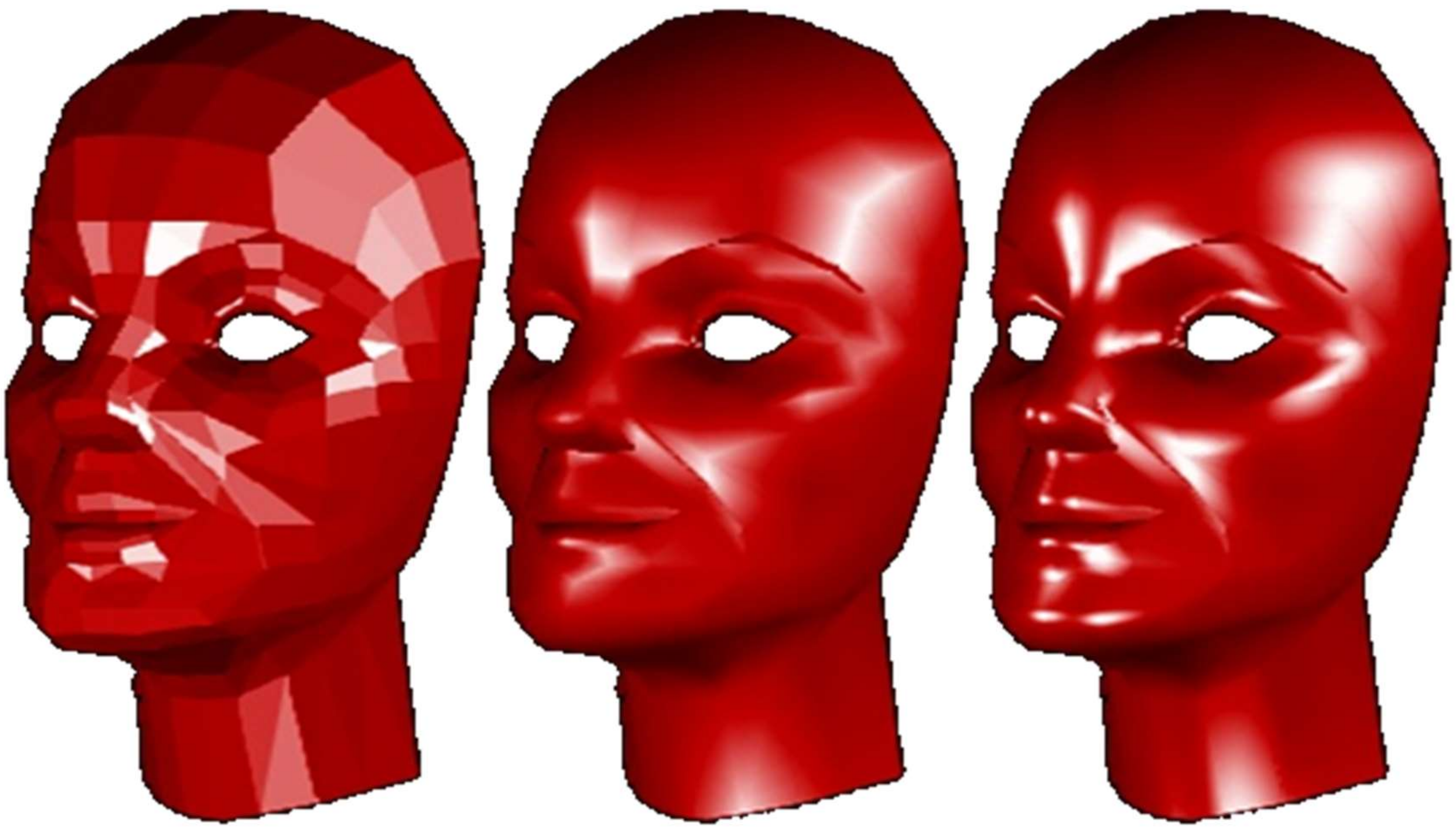




Normal Interpolation



Flat / Gouraud / Phong Comparison



Transforming Normals

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