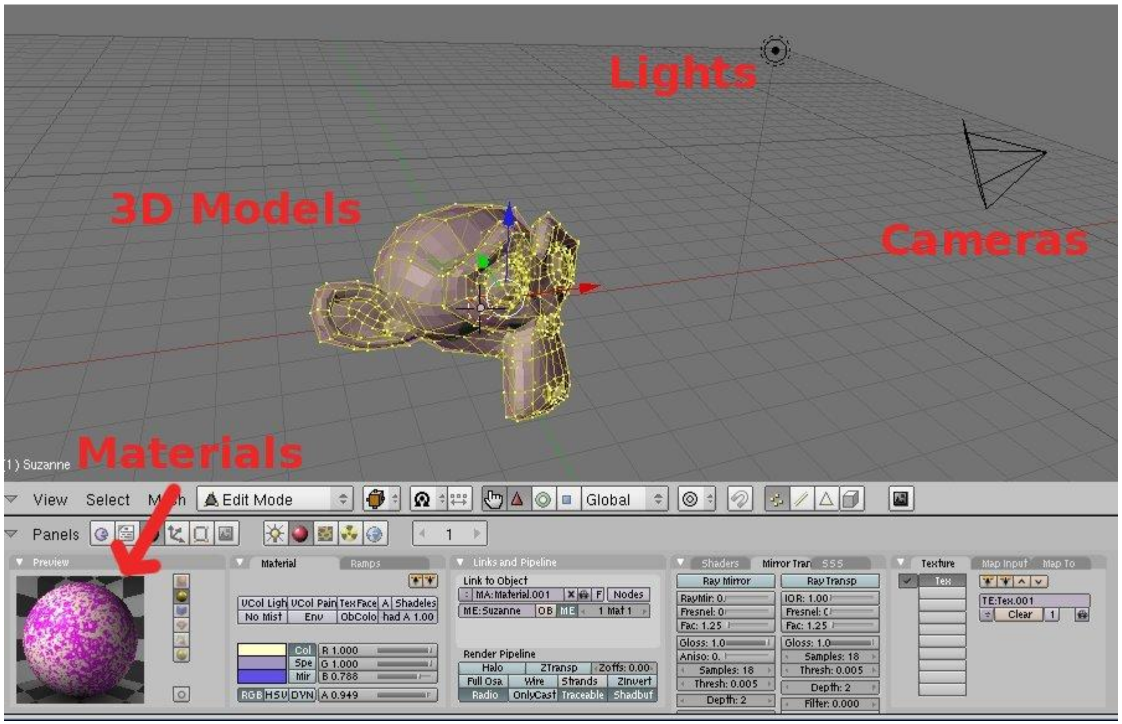
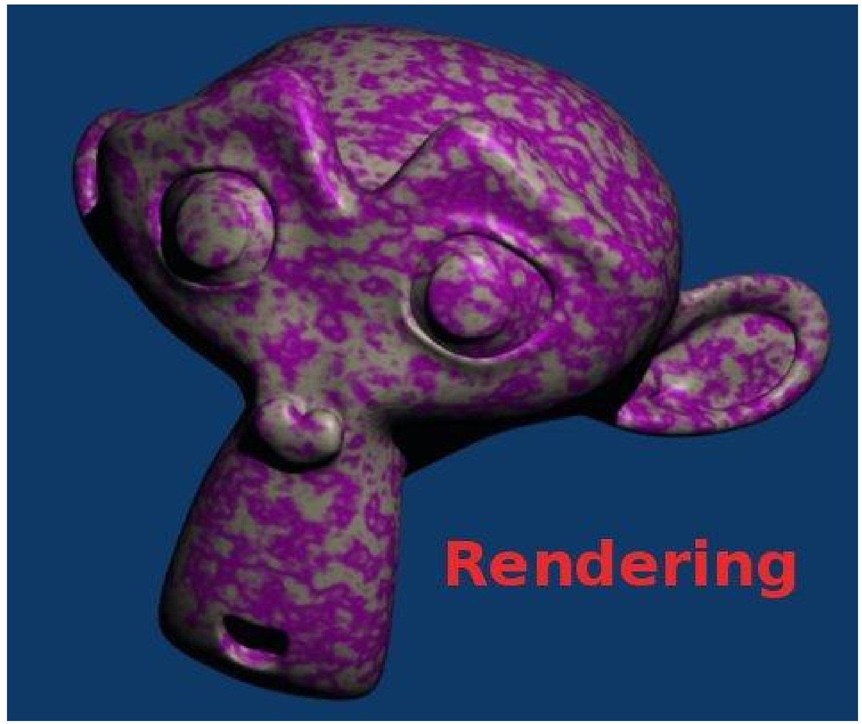
3: illumination

1. Project geometry onto the screen frame
2. Rasterize triangles
3. Visibility test
4. Compute pixel color





Illumination

Which color shall we display in each pixel ?

⇒ Depends on the local amount of light coming back to the eyes

⇒ So it depends on :

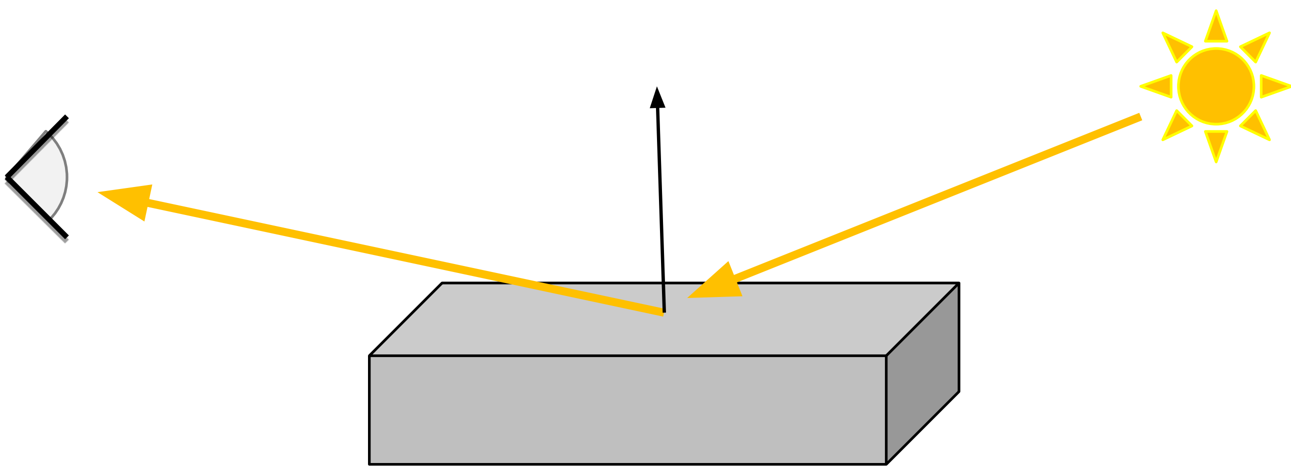
- where the surface element is in 3D
- its orientation w.r.t. lights & camera
- the material the surface is made of





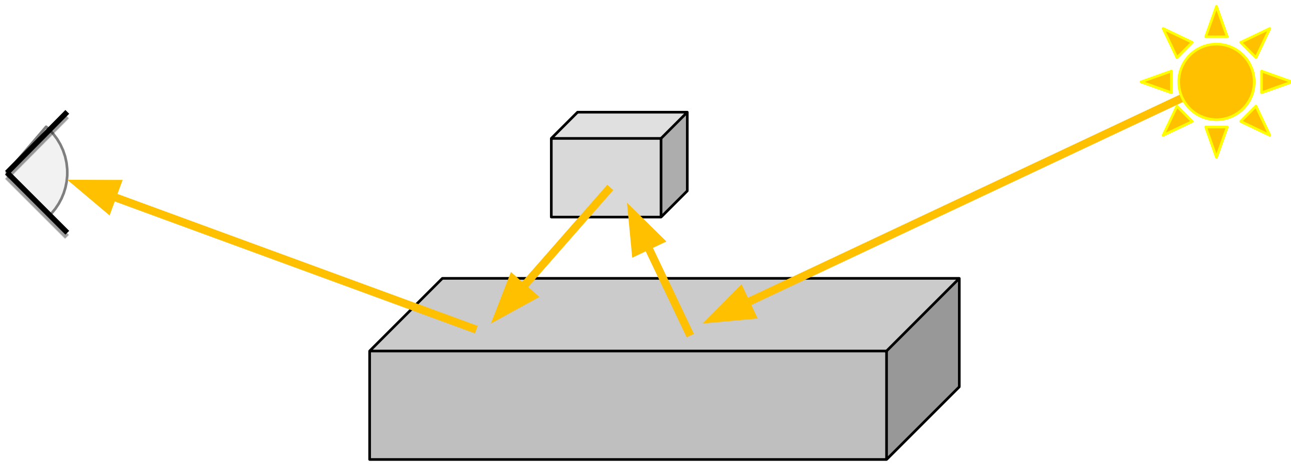
Illumination of an object

- 2 components :
  - Direct illumination from light sources



Illumination of an object

- 2 components :
  - Direct illumination from light sources
  - Indirect illumination
    - all objects become secondary sources



Illumination of an object

- The reflectance equation [Kajiya 1986]:

$$L(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_n} \rho(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell})(\mathbf{n} \cdot \boldsymbol{\ell}) \; L(\mathbf{p} \leftarrow \boldsymbol{\ell}) \; d\boldsymbol{\ell}$$

Outgoing radiance

Color at point “p” of the surface towards the camera eye “e”

Illumination of an object

- The reflectance equation [Kajiya 1986]:

$$L(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_n} \rho(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell})(\mathbf{n} \cdot \boldsymbol{\ell}) \; L(\mathbf{p} \leftarrow \boldsymbol{\ell}) \; d\boldsymbol{\ell}$$

Integral over the hemisphere

means: consider all possible light directions around the point “p”

Illumination of an object

- The reflectance equation [Kajiya 1986]:

$$L(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_n} \rho(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell})(\mathbf{n} \cdot \boldsymbol{\ell})\ L(\mathbf{p} \leftarrow \boldsymbol{\ell})\ d\boldsymbol{\ell}$$

Incoming radiance

The light energy/color in the given direction

Illumination of an object

- The reflectance equation [Kajiya 1986]:

$$L(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_n} \rho(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell})(\mathbf{n} \cdot \boldsymbol{\ell}) \; L(\mathbf{p} \leftarrow \boldsymbol{\ell}) \; d\boldsymbol{\ell}$$

Surface orientation

more energy for surfaces perpendicular to the light

Illumination of an object

- The reflectance equation [Kajiya 1986]:

$$L(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega_n} \rho(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell})(\mathbf{n} \cdot \boldsymbol{\ell}) \; L(\mathbf{p} \leftarrow \boldsymbol{\ell}) \; d\boldsymbol{\ell}$$

BRDF (Bidirectionnal Reflectance Distribution Function)

amount of energy towards the eye "e"
given the light "l" and point "p"

Illumination of an object

- Drastic simplifications for real-time app:

$$L(\mathbf{p} \rightarrow \mathbf{e}) = \sum_k \rho(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell}_k)\ (\mathbf{n} \cdot \boldsymbol{\ell}_k)\ L(\mathbf{p} \leftarrow \boldsymbol{\ell}_k)$$

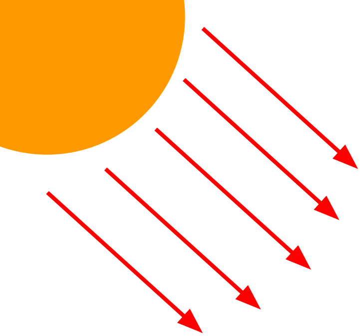
Each light

Illumination of an object

- Drastic simplifications for real-time app:

$$L(\mathbf{p} \rightarrow \mathbf{e}) = \sum_k \rho(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell}_k)\ (\mathbf{n} \cdot \boldsymbol{\ell}_k)\ L(\mathbf{p} \leftarrow \boldsymbol{\ell}_k)$$

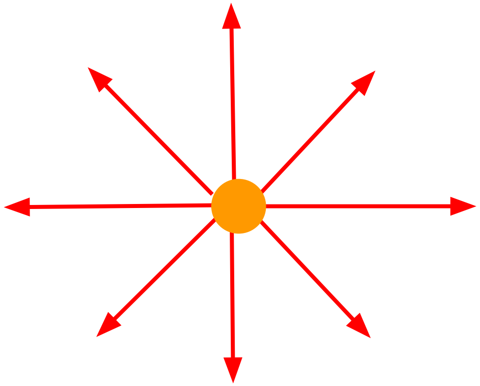
How to represent lights?



How to represent lights?

- Directional light

$$L(\mathbf{p} \leftarrow \boldsymbol{\ell}) = L$$

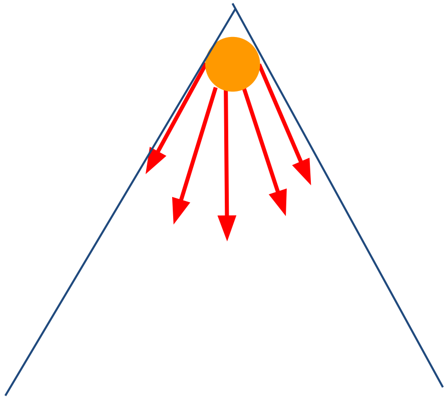


How to represent lights?

- Directional light $$L(\mathbf{p} \leftarrow \boldsymbol{\ell}) = L$$

- Point light $$L(\mathbf{p} \leftarrow \boldsymbol{\ell}) = L/r^2$$

$$r = ||\mathbf{p} - \mathbf{p}_{\ell}|| \qquad \boldsymbol{\ell} = \frac{\mathbf{p} - \mathbf{p}_{\ell}}{r}$$



How to represent lights?

- Directional light $$L(\mathbf{p} \leftarrow \boldsymbol{\ell}) = L$$
- Point light $$L(\mathbf{p} \leftarrow \boldsymbol{\ell}) = L/r^2$$
- Spot light $$L(\mathbf{p} \leftarrow \boldsymbol{\ell}) = \frac{(\mathbf{s}_{\ell} \cdot \boldsymbol{\ell})^e L}{r^2}$$

$$r = ||\mathbf{p} - \mathbf{p}_{\ell}|| \qquad \boldsymbol{\ell} = \frac{\mathbf{p} - \mathbf{p}_{\ell}}{r}$$

Illumination of an object

- Drastic simplifications for real-time app:

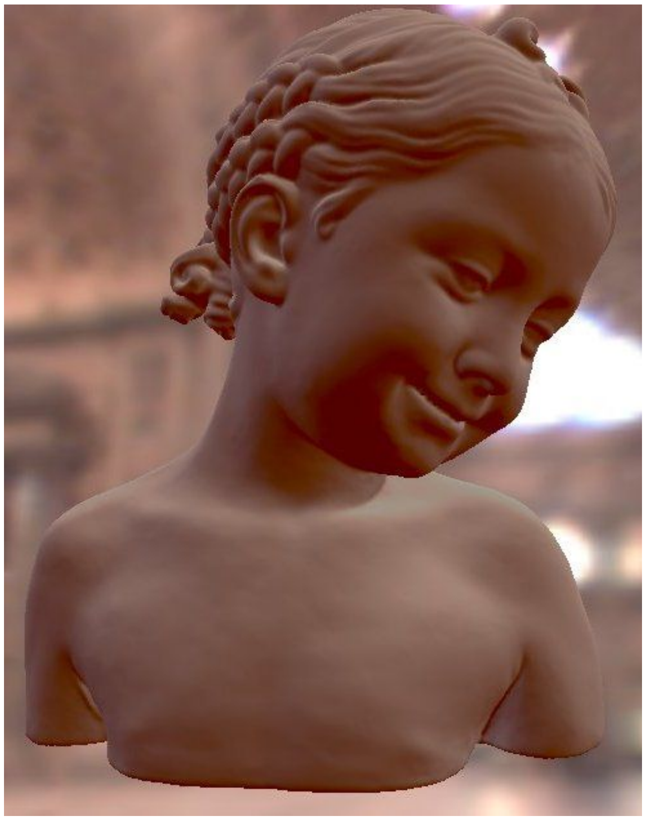
$$L(\mathbf{p} \rightarrow \mathbf{e}) = \sum_k \rho(\mathbf{p}, \mathbf{e}, \boldsymbol{\ell}_k)\,(\mathbf{n} \cdot \boldsymbol{\ell}_k)\; L(\mathbf{p} \leftarrow \boldsymbol{\ell}_k)$$

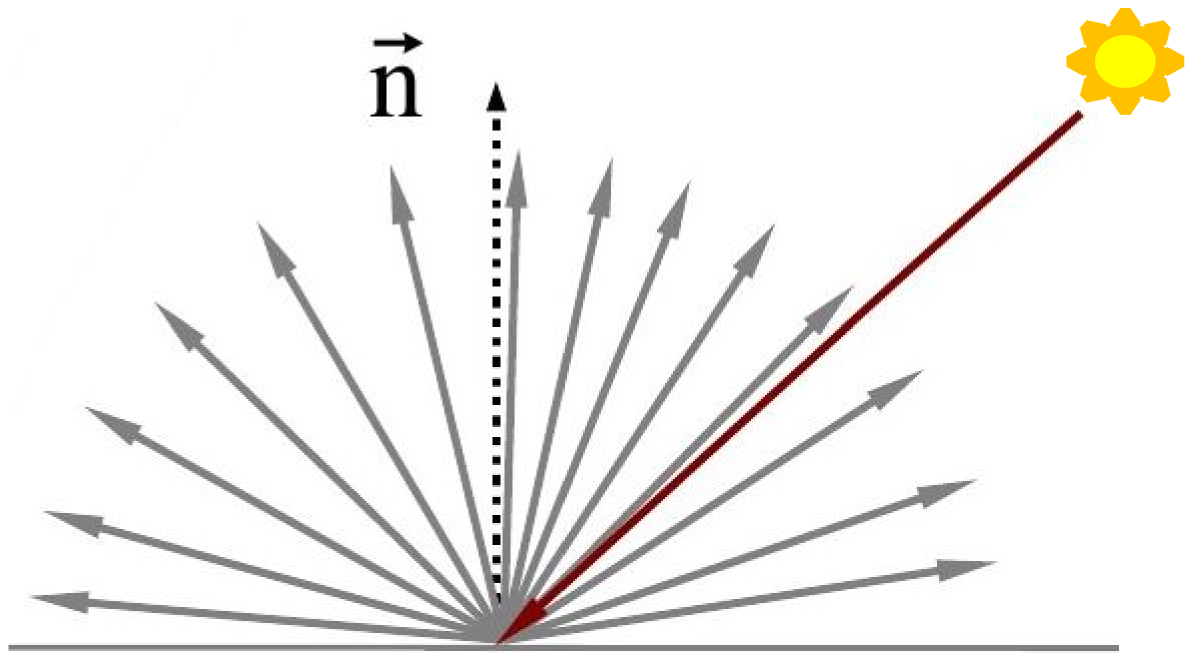
how does it work?

Why does the material impact how much energy comes to the eye?

Illumination of an object

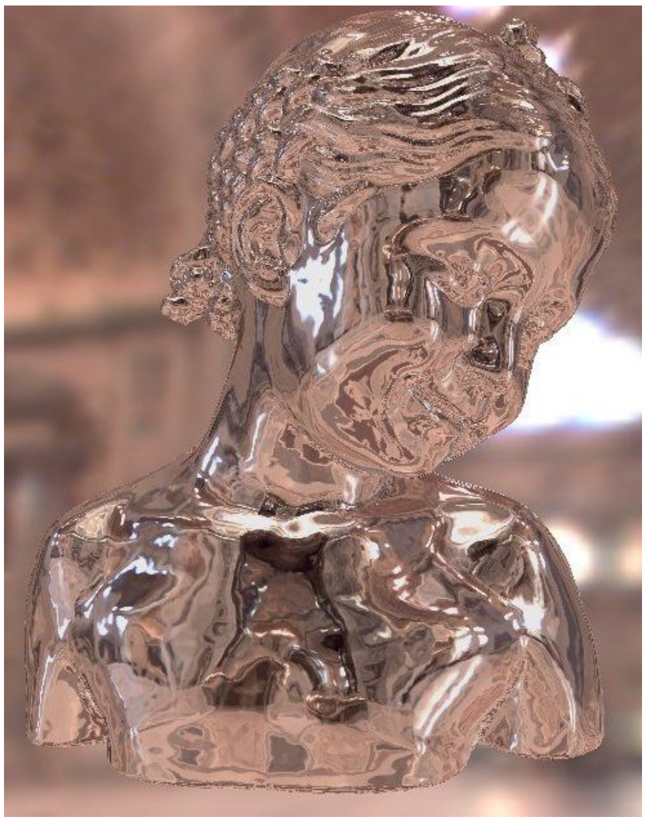
- Diffuse (matte) surface

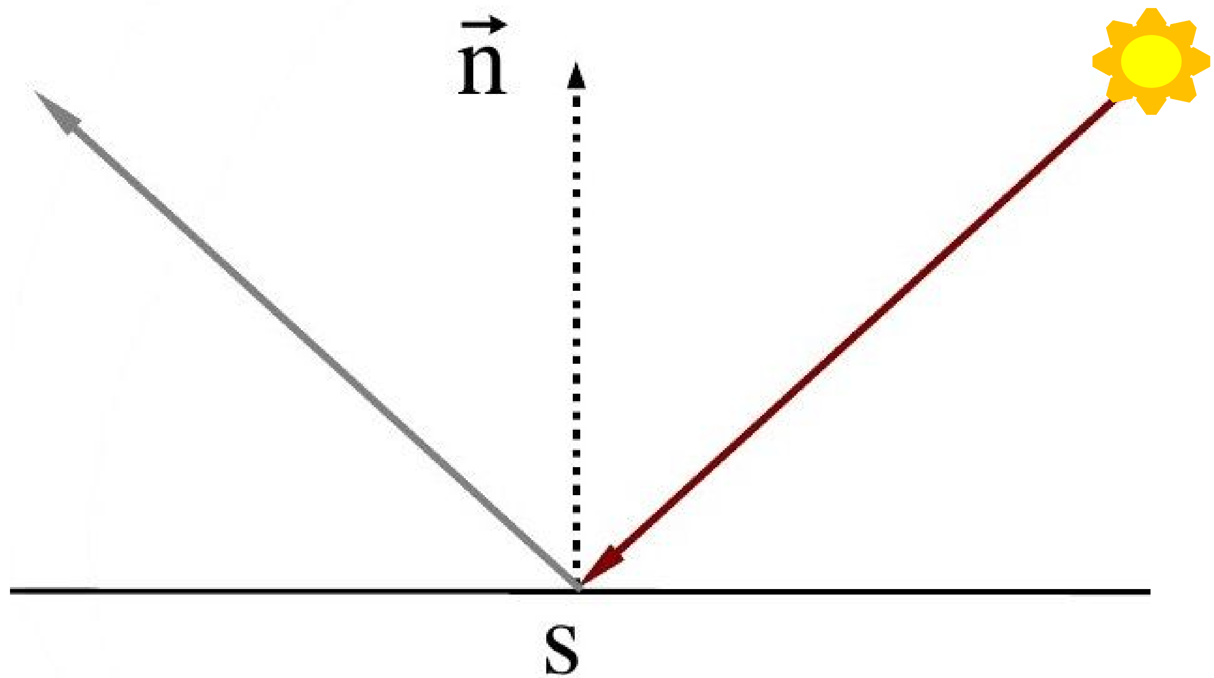




Illumination of an object

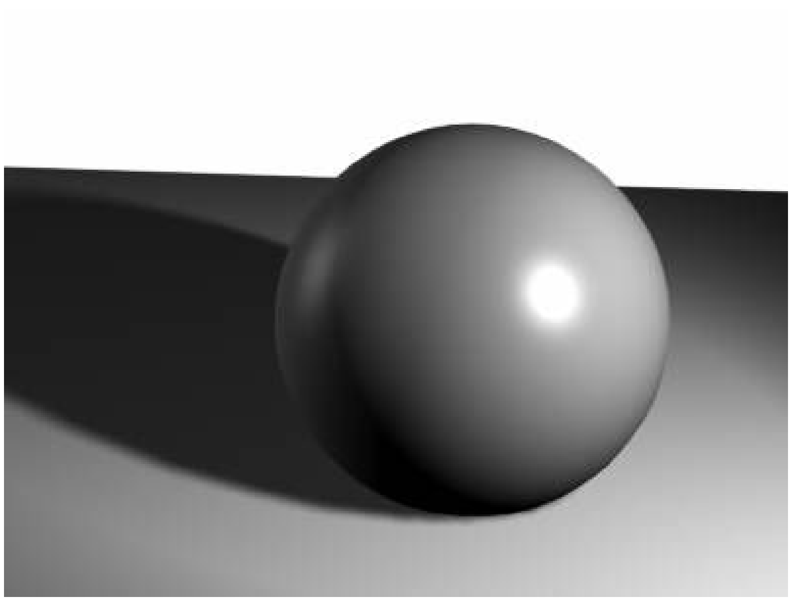
- Specular (mirror) surface

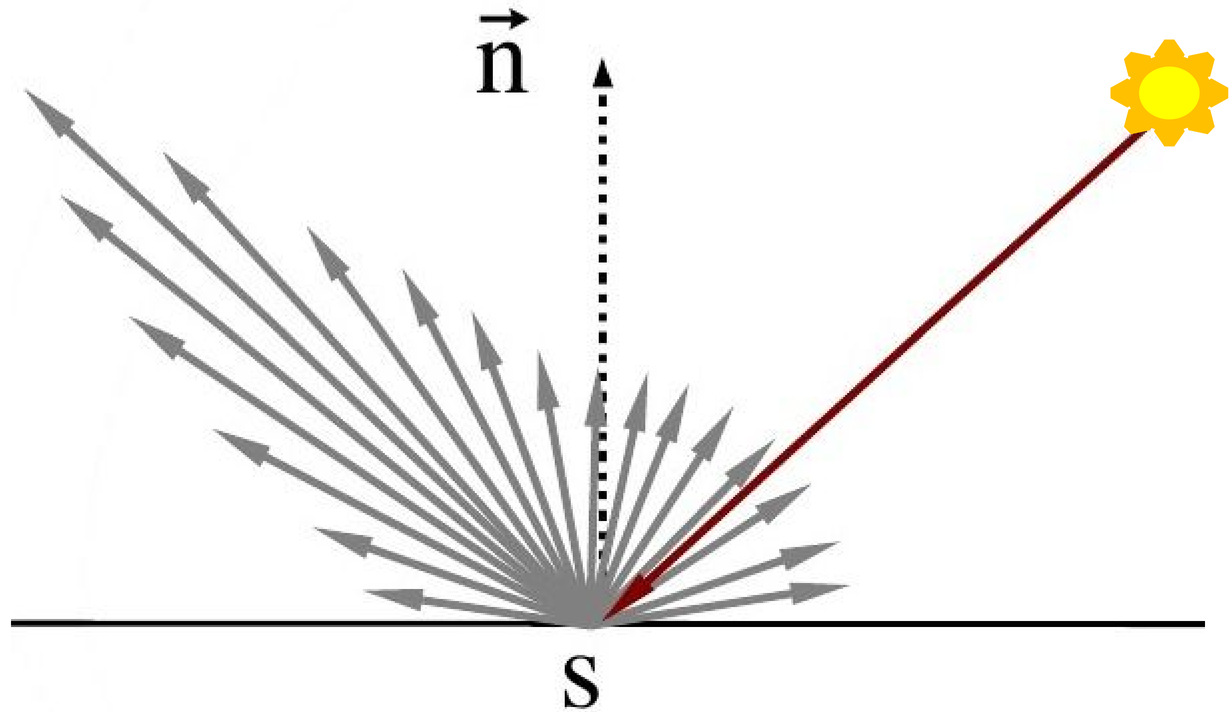




Illumination of an object

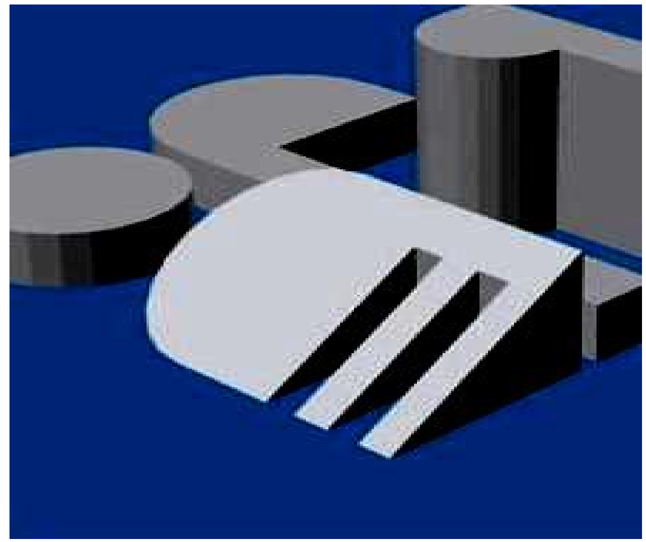
- In real-life: a combination of both





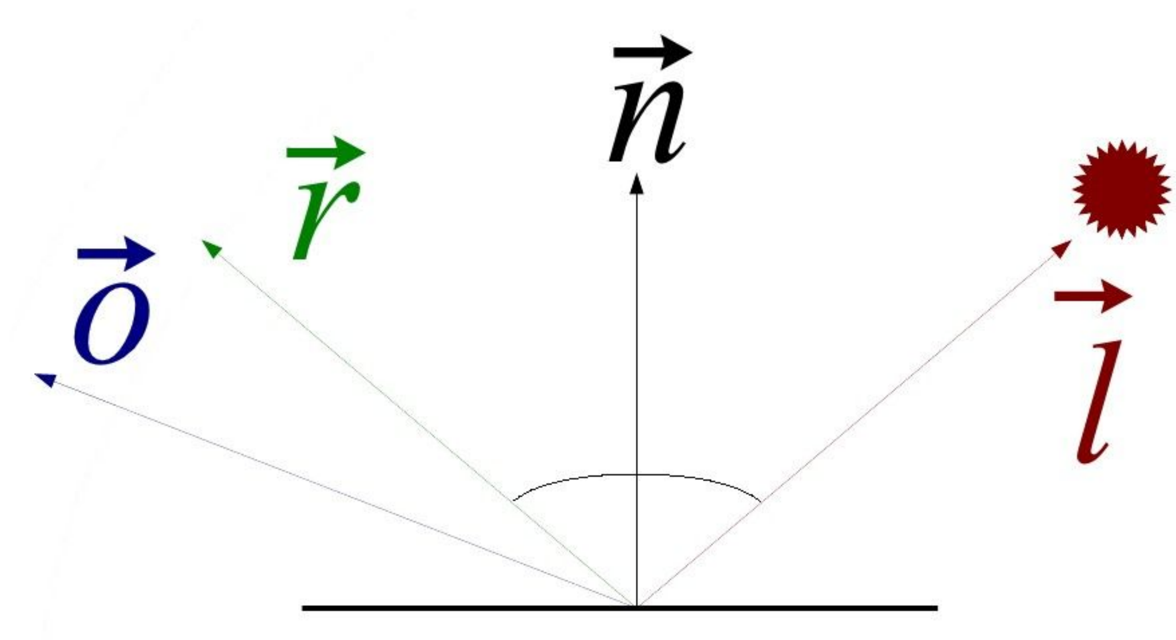
Phong's local illumination model

- Opaque faces only, processed one by one
- Direct lighting from the sources
- A constant « ambiant » term
    - \+ Diffuse shading
    - \+ Specular shading



Phong's local illumination model

[Phong CACM 1975]



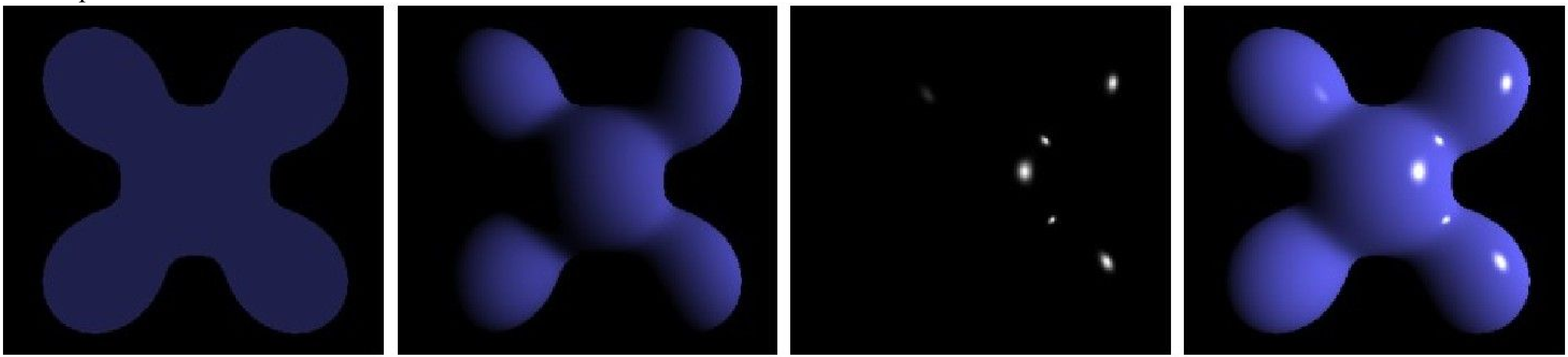
$$L(\mathbf{p} \to \mathbf{e}) = K_a + \sum_s [K_d(\mathbf{n} \cdot \boldsymbol{\ell}) + K_s(\mathbf{r} \cdot \mathbf{e})^n] I_s$$

ambiant diffuse specular

Phong's local illumination model

[Phong CACM 1975]

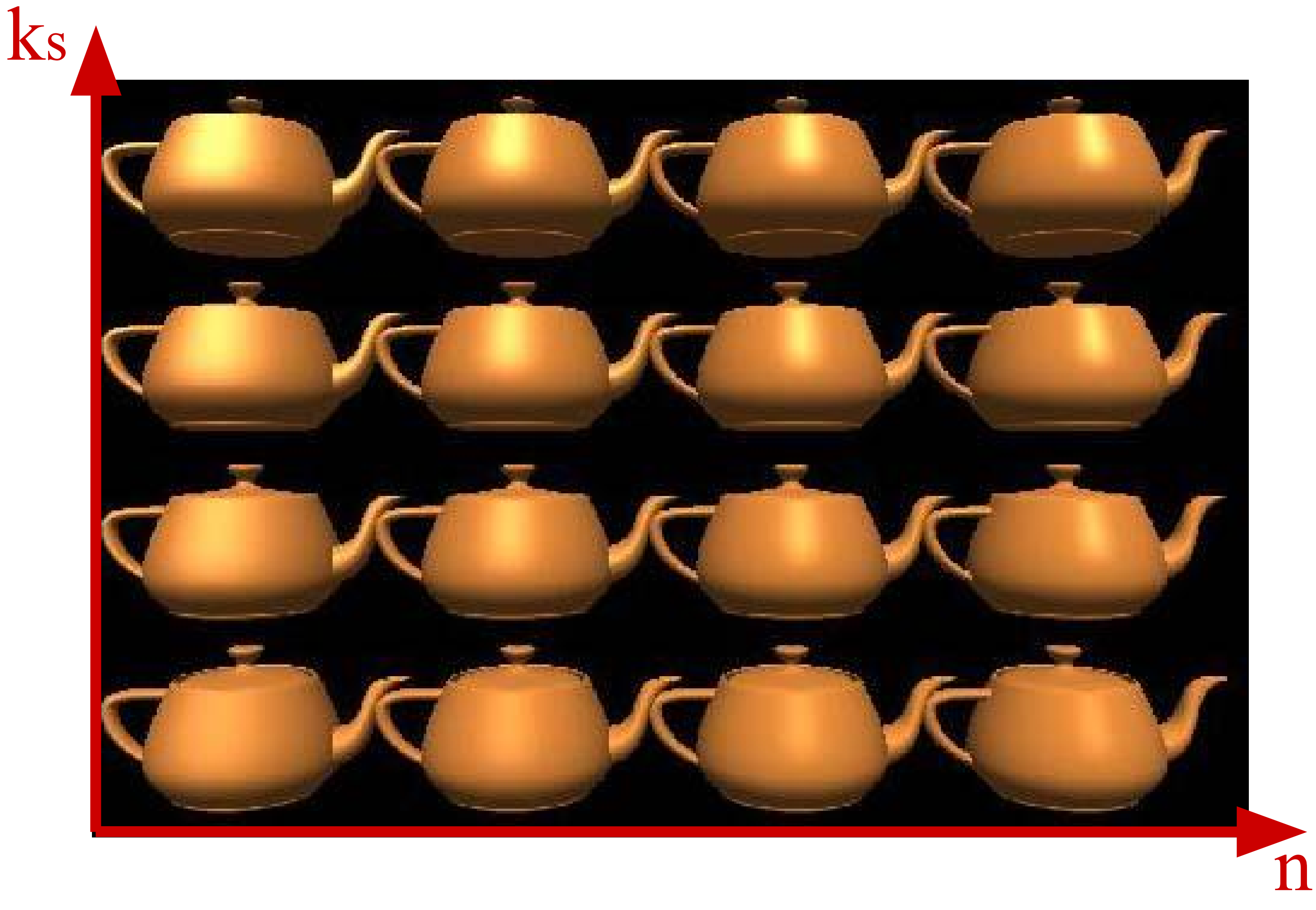
wikipedia



$$L(\mathbf{p} \rightarrow \mathbf{e}) = K_a + \sum_s [K_d(\mathbf{n} \cdot \boldsymbol{\ell}) + K_s(\mathbf{r} \cdot \mathbf{e})^n] I_s$$

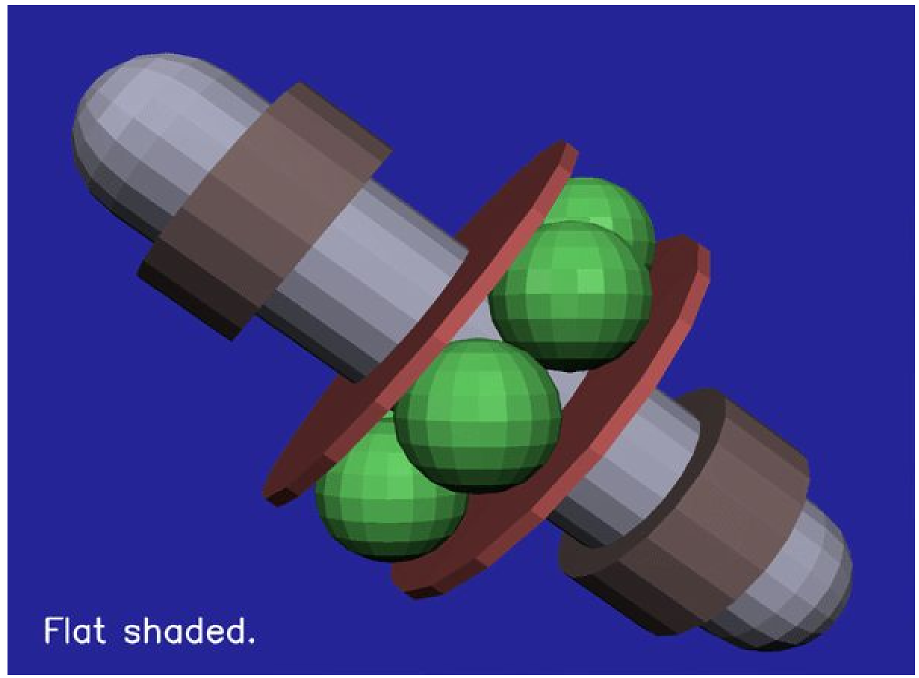
ambiant diffuse specular

Phong's local illumination

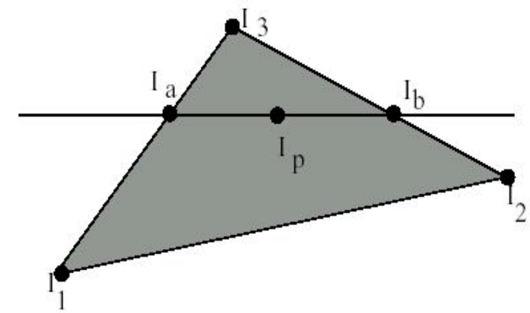


Direct application (Flat Shading)

- A single normal by face
  - Uniform colors!



Gouraud's shading

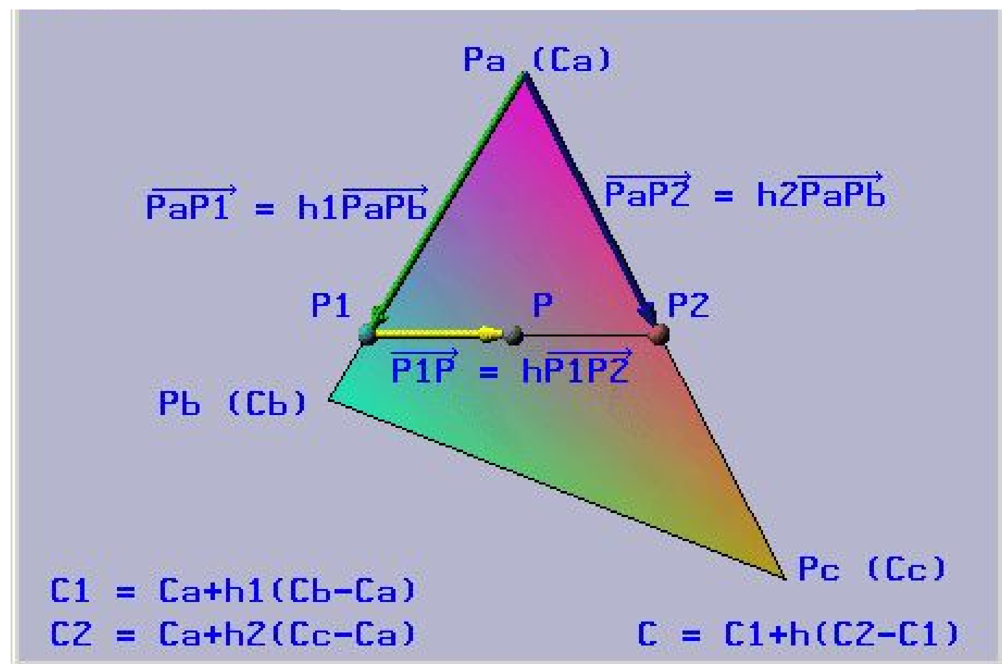


Input: normals at vertices

Color

- Compute illumination at vertices from the normals
- Interpolate illumination

bi-linear interpolation →



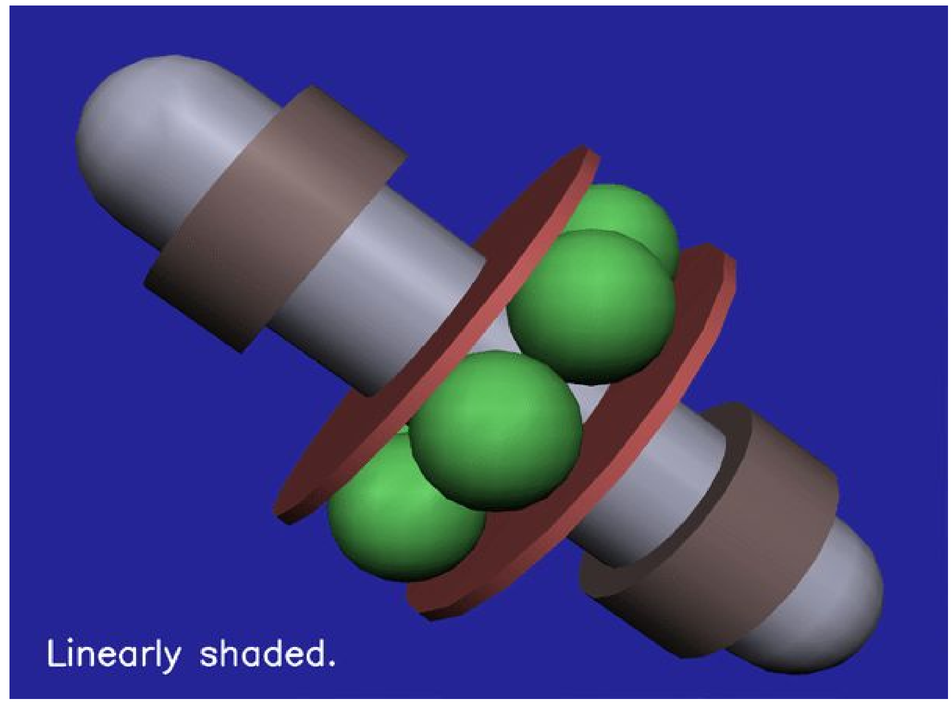
Gouraud's shading

Results

- Illusion of smoothness!
- Faces seen on silhouettes

Exercise (TD) :

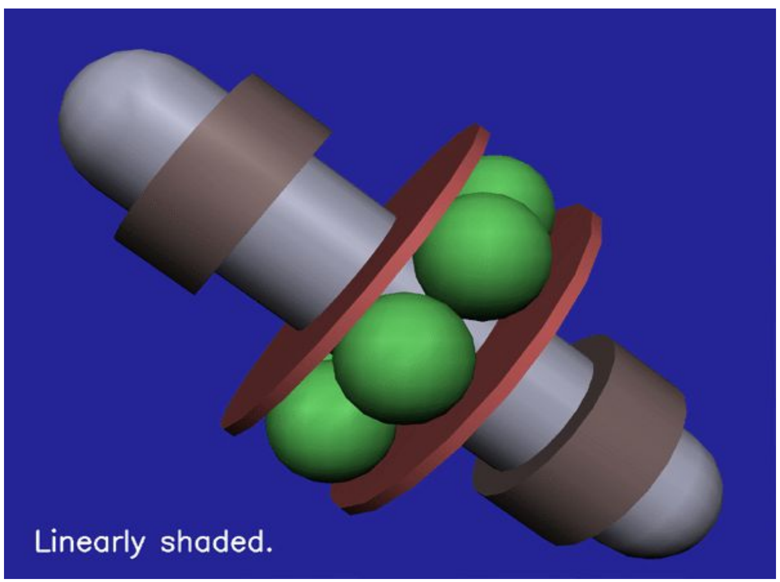
- Is it correct? Why?
- Main visual problems?

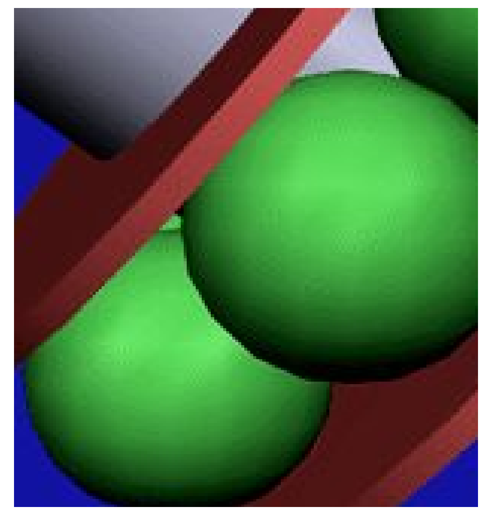


Solution: Gouraud's shading

→ Reflections can be missed if no vertex on the spot

→ They are propagated to large regions due to interpolation

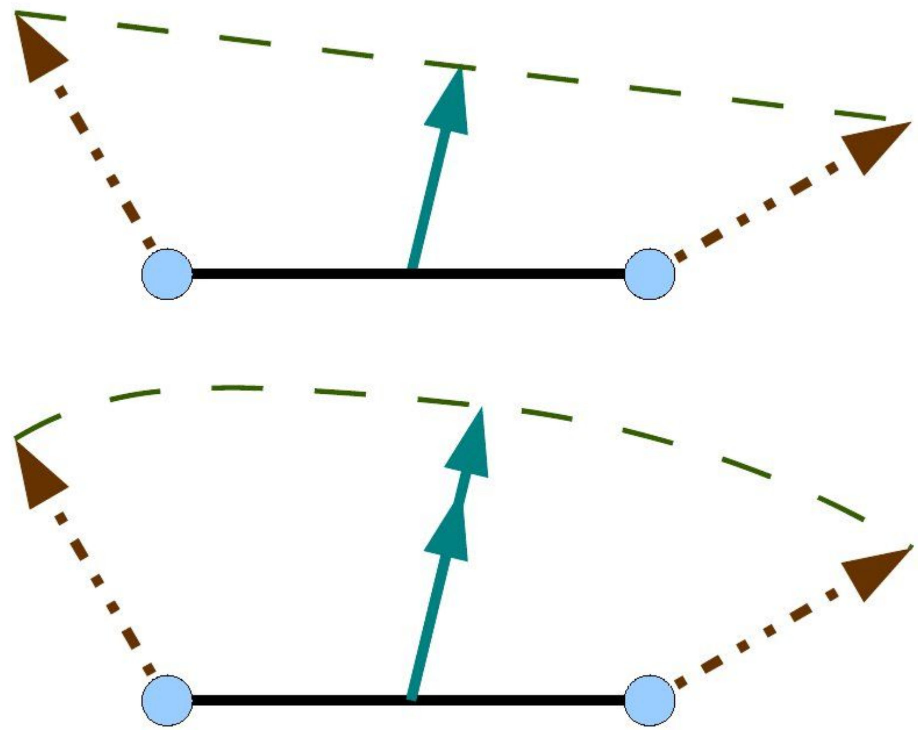




Phong's shading

- A normal N by vertex
- Interpolate normals
  (bilinear in x,y,z)
- Re-normalize!
- Illumination at each pixel

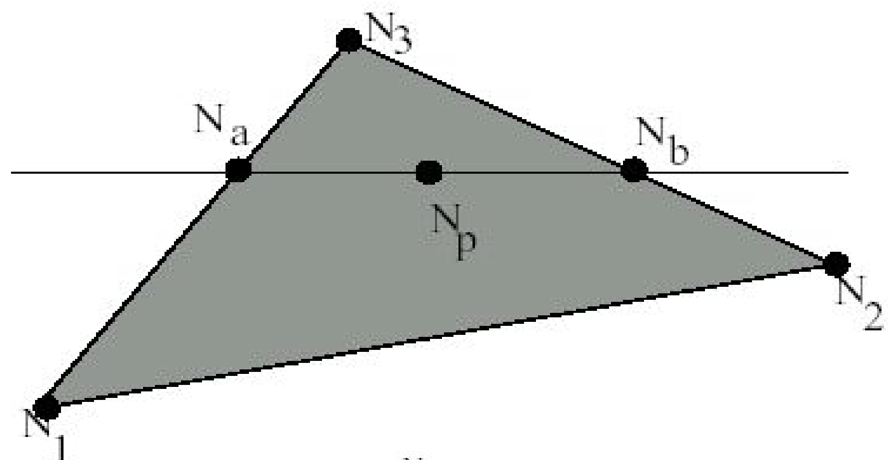
Better than Gouraud!

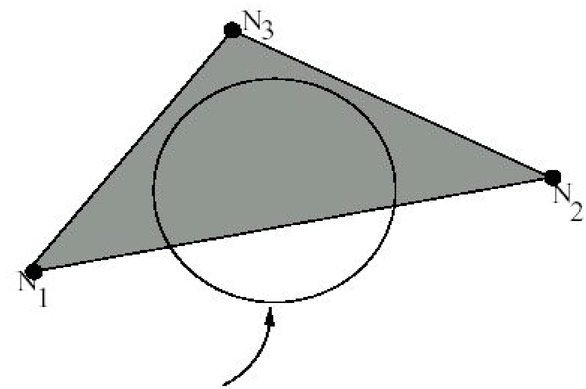


Phong's shading

- A normal N by vertex
- Interpolate normals
  (bilinear in x,y,z)
- Re-normalize!
- Illumination at each pixel

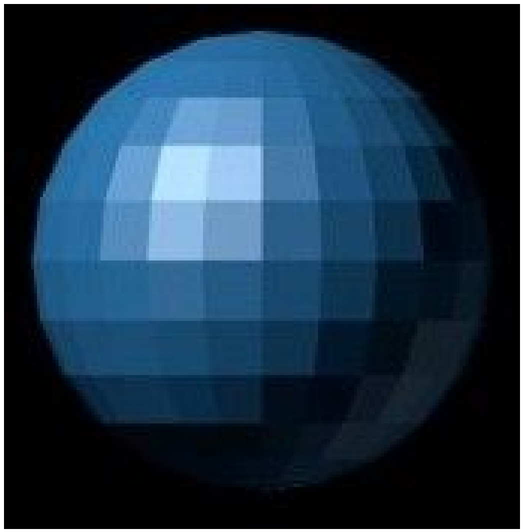
Better than Gouraud!



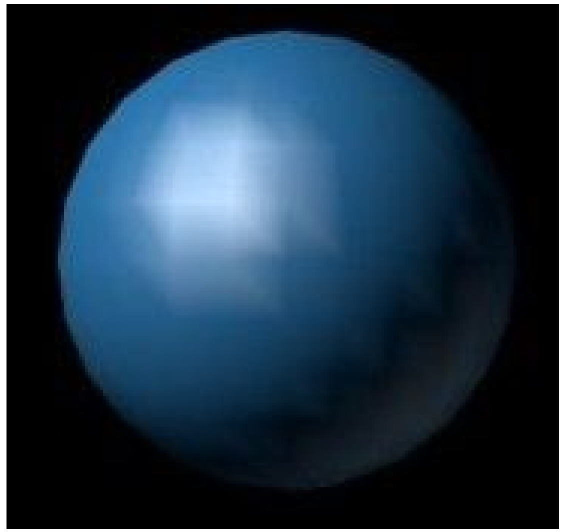


Specular reflexion

Phong's shading



Flat



Gouraud

Phong

*** Phong shading != Phong model ***

TD: Projective rendering versus Realism

Projective rendering + Phong

vs

Realistic rendering

Exercise 1:

Which effects are missing?

Solution 1.
Projective rendering versus Realism

Missing effects

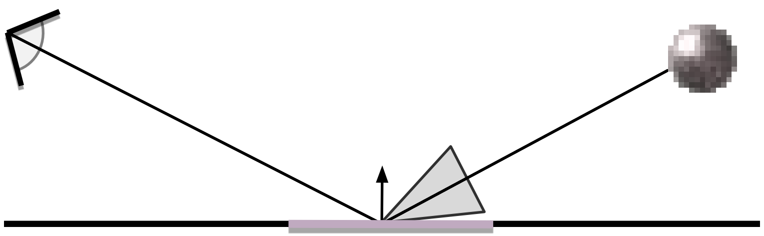
- Cast shadows
- Transparency
- Refraction
- Extended light sources

2. Add reflections to projective rendering?
Case of a planar mirror

Exercise: How would you add a mirror?

- Propose different solutions
- Discuss the advantages and drawbacks

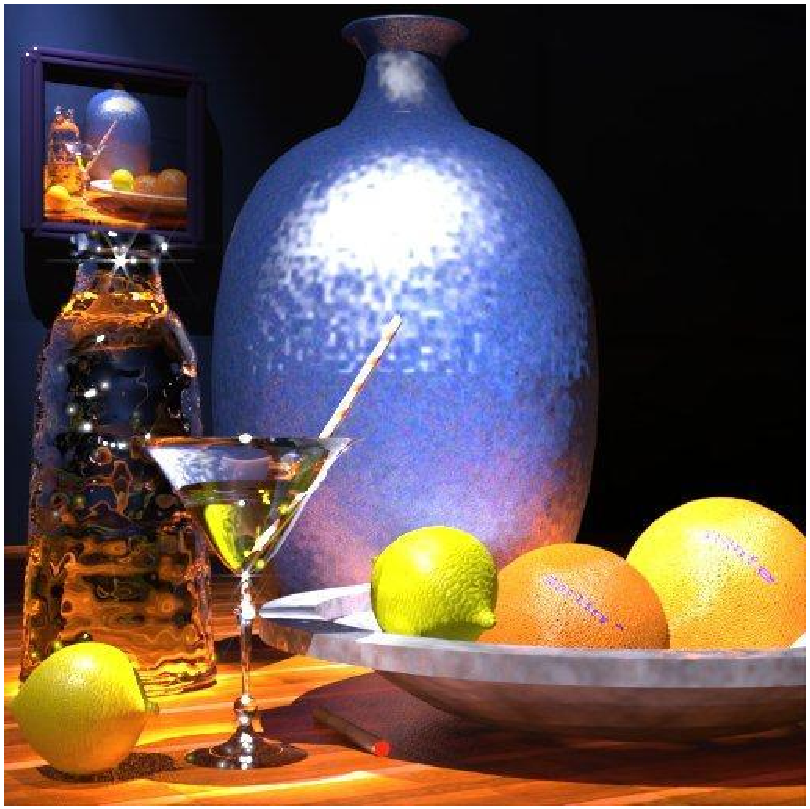
Correct? Costly? Moving cameras?..



Solution 2. Add reflections to projective rendering?
Case of a planar mirror

- **Solution A**
  1. Replace the mirror by a window
  2. Place a symetric copy of the scene behind
- **Solution B**
  1. Compute rendering from the mirror in reflection direction
  2. Use the resulting image as a “texture”

More effects in the advanced rendering course… and next year!



Ray-tracing

Global illumination