CS4620/5620: Lecture 15

Programmable Shading

Announcements

- HW 1 back
- HW 2 out
  - Due next Friday
  - Due date?

Pipeline overview

you are here → APPLICATION

COMMAND STREAM

3D transformations; shading → VERTEX PROCESSING

TRANSFORMED GEOMETRY

conversion of primitives to pixels → RASTERIZATION

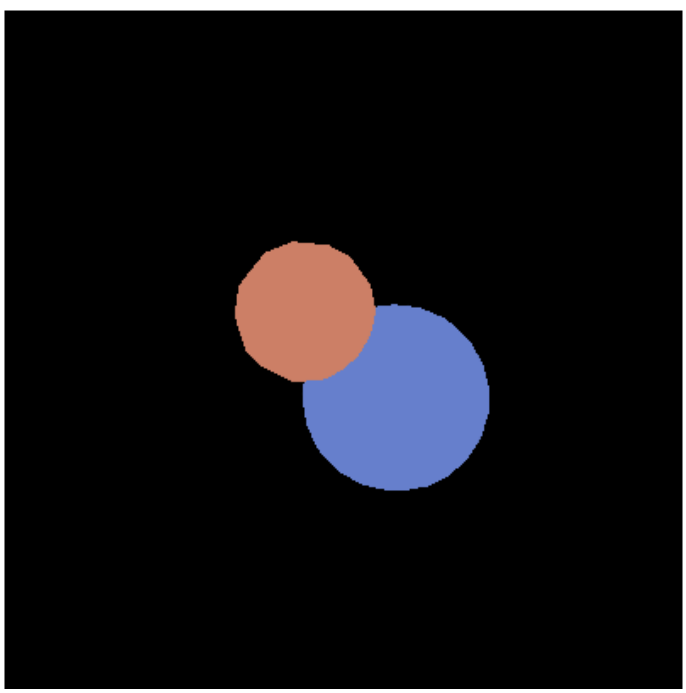
FRAGMENTS

blending, compositing, shading → FRAGMENT PROCESSING

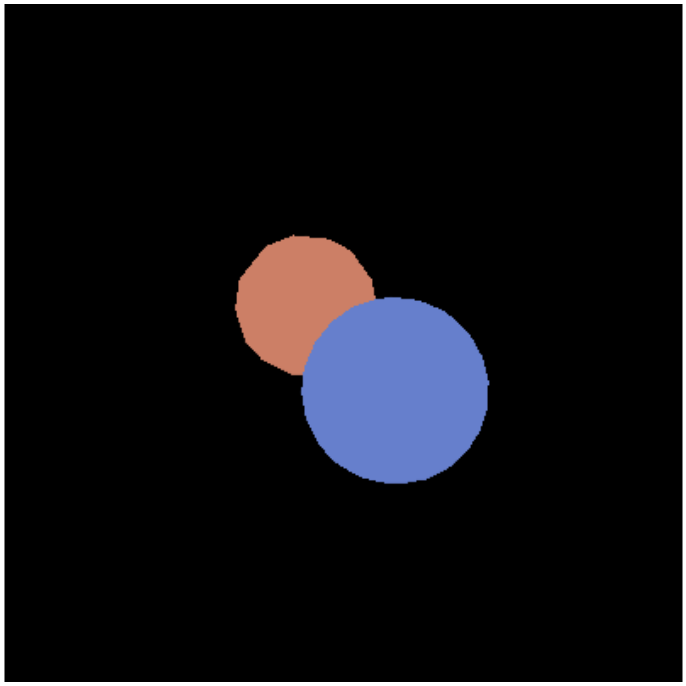
FRAMEBUFFER IMAGE

user sees this → DISPLAY

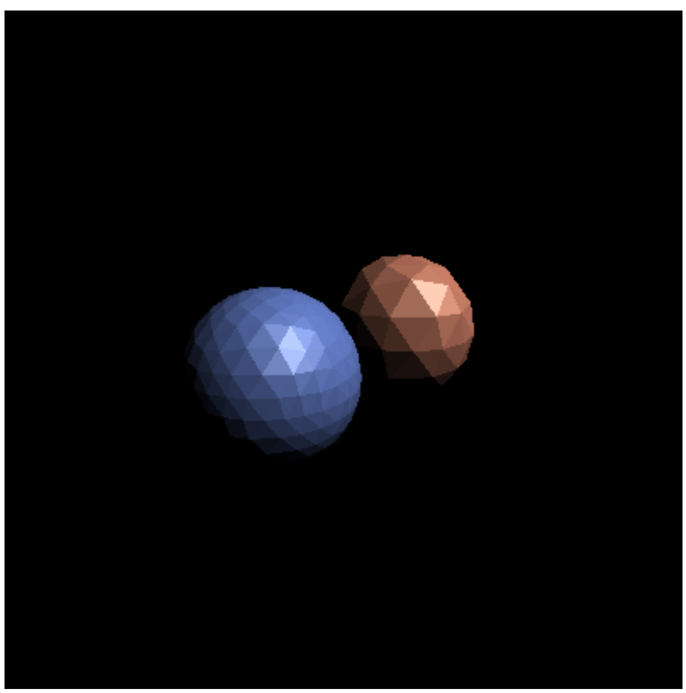
Result of minimal pipeline (no z test)



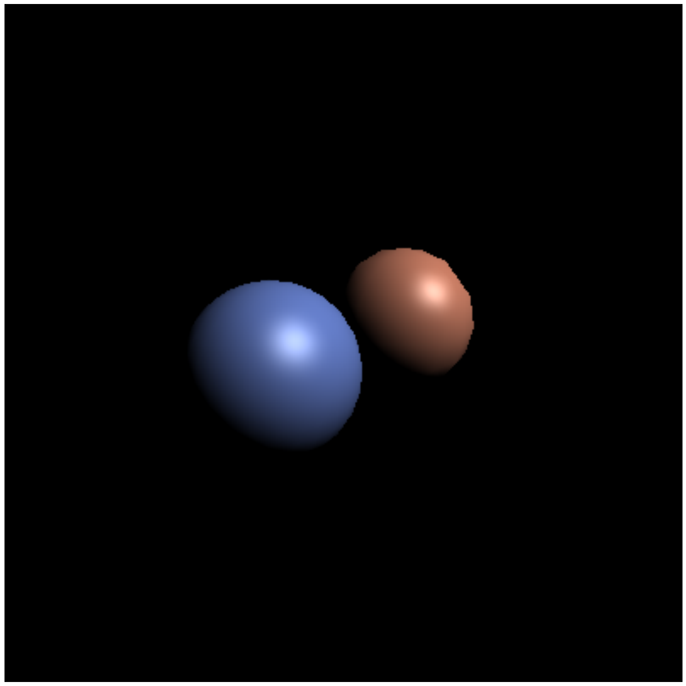
Result of z-buffer pipeline



Result of flat-shading pipeline



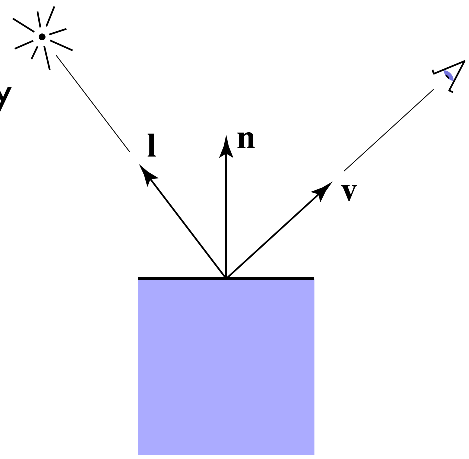
Result of Phong shading pipeline



How to achieve shading?

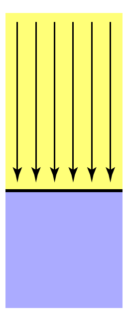
Shading

- Compute light reflected toward camera
- Inputs:
  - eye direction
  - light direction and light intensity (for each of many lights)
  - surface normal
  - surface parameters (color, shininess, …)

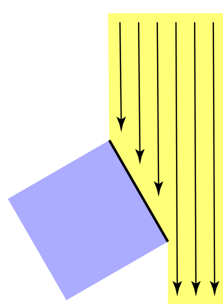


Diffuse reflection

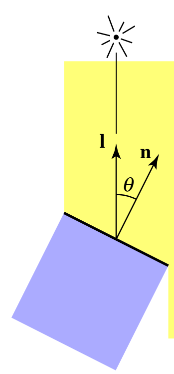
- Light is scattered uniformly in all directions
  - the surface color is the same for all viewing directions
- Lambert's cosine law



Top face of cube receives a certain amount of light



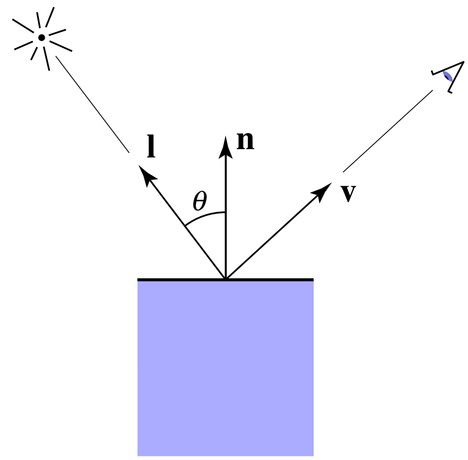
Top face of 60° rotated cube intercepts half the light



In general, light per unit area is proportional to cos θ = **l • n**

Lambertian shading

- Shading independent of view direction



illumination from source

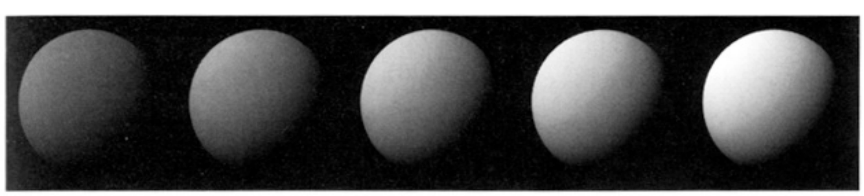
$$L_d = k_d\, I \max(0, \mathbf{n} \cdot \mathbf{l})$$

diffuse coefficient

diffusely reflected light

Lambertian shading

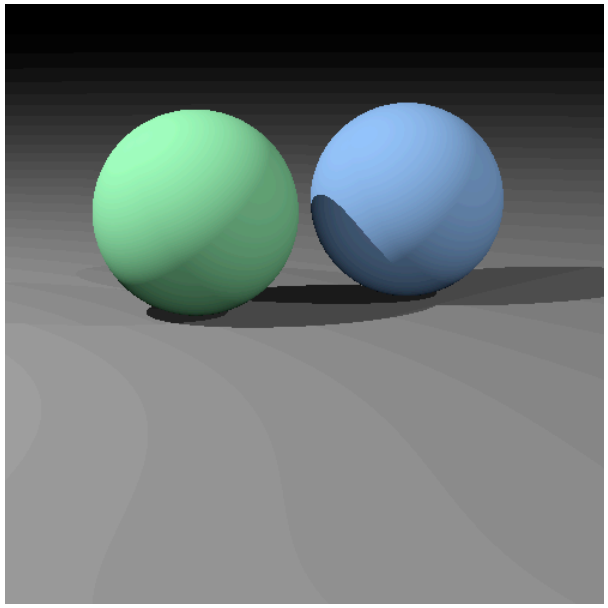
- Produces matte appearance



$k_d \longrightarrow$

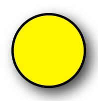
[Foley et al.]

Diffuse shading

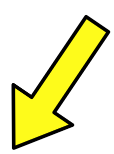


Light

- Local light
  – Position

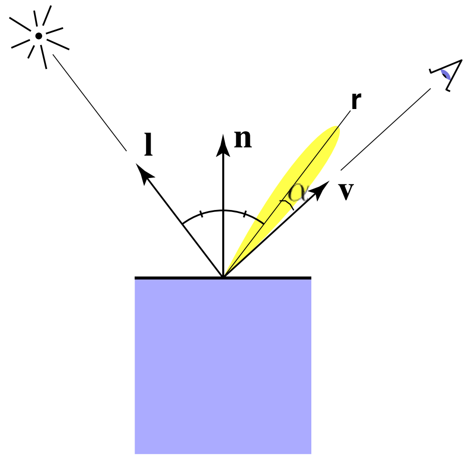


- Directional light (e.g., sun)
  – Direction, no position



Specular shading (Phong)

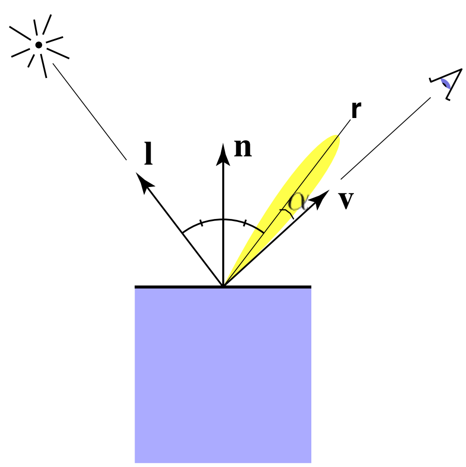
- Intensity depends on view direction
  - bright near mirror configuration
  - measure "near" by dot product of unit vectors



$$cos(\alpha) = \mathbf{v}.\mathbf{r}$$

Specular shading (Phong)

- Intensity depends on view direction
  - bright near mirror configuration



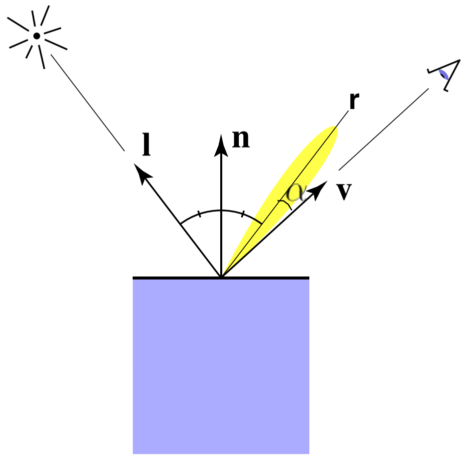
$$L_s = k_s I \max(0, cos\alpha)^n$$

$$cos(\alpha) = \mathbf{v}.\mathbf{r}$$

$$L_s = k_s I \max(0, \mathbf{v}.\mathbf{r})^n$$

Reflected direction

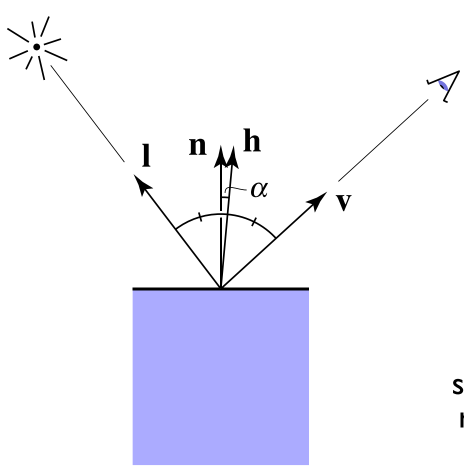
- Intensity depends on view direction
  - reflects incident light from mirror direction



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

Specular shading (Blinn-Phong)

- Close to mirror ⇔ half vector near normal



$$\mathbf{h} = \mathrm{bisector}(\mathbf{v}, \mathbf{l}) = \frac{\mathbf{v} + \mathbf{l}}{\|\mathbf{v} + \mathbf{l}\|}$$

$$L_s = k_s\, I \max(0, \cos\alpha)^n = k_s\, I \max(0, \mathbf{n} \cdot \mathbf{h})^n$$

specularly reflected light

specular coefficient

Phong model—plots

- Increasing *n* narrows the lobe

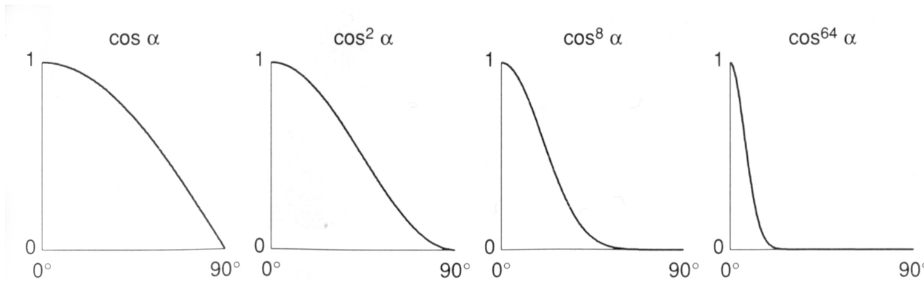
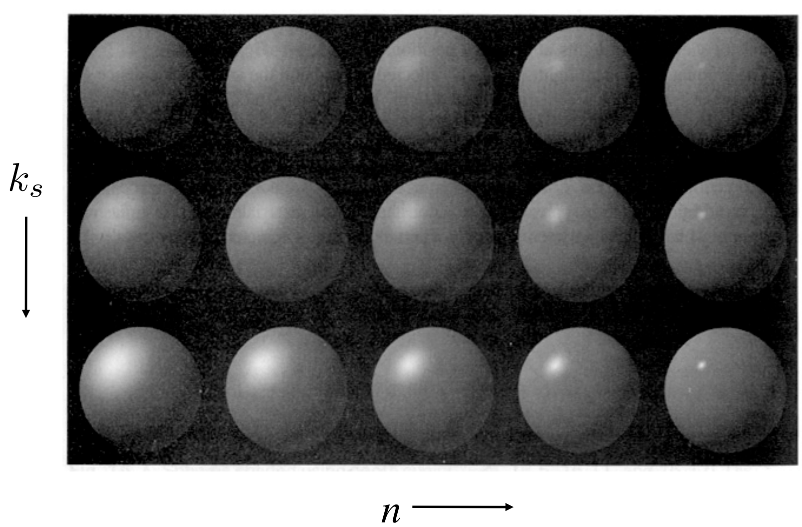


Fig. 16.9 Different values of $\cos^n \alpha$ used in the Phong illumination model.

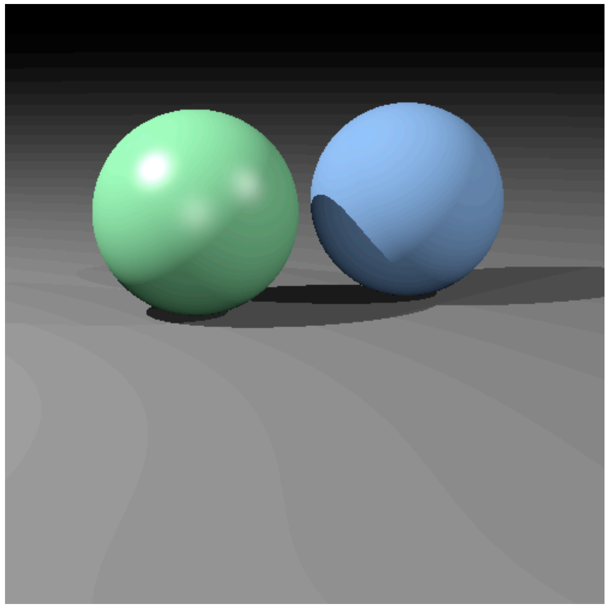
[Foley et al.]

Specular shading



[Foley et al.]

Diffuse + Phong shading



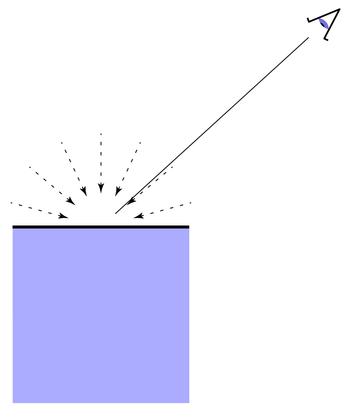
Multiple lights

- Just loop over lights, add contributions
- Important to fill in black shadows

- Ambient shading
  - black shadows are not really right
  - one solution: dim light at camera
  - alternative: add a constant "ambient" color to the shading…

Ambient shading

- Shading that does not depend on anything
  - add constant color to account for disregarded illumination and fill in black shadows



$$L_a = k_a \, I_a$$

ambient coefficient

reflected ambient light

Putting it together

- Usually include ambient, diffuse, Phong in one model

$$\begin{aligned} L &= L_a + L_d + L_s \\ &= k_a \, I_a + k_d \, I \max(0, \mathbf{n} \cdot \mathbf{l}) + k_s \, I \max(0, \mathbf{n} \cdot \mathbf{h})^n \end{aligned}$$

- The final result is the sum over many lights

$$L = L_a + \sum_{i=1}^{N} \left[(L_d)_i + (L_s)_i \right]$$

$$L = k_a \, I_a + \sum_{i=1}^{N} \left[k_d \, I_i \max(0, \mathbf{n} \cdot \mathbf{l}_i) + k_s \, I_i \max(0, \mathbf{n} \cdot \mathbf{h}_i)^n \right]$$