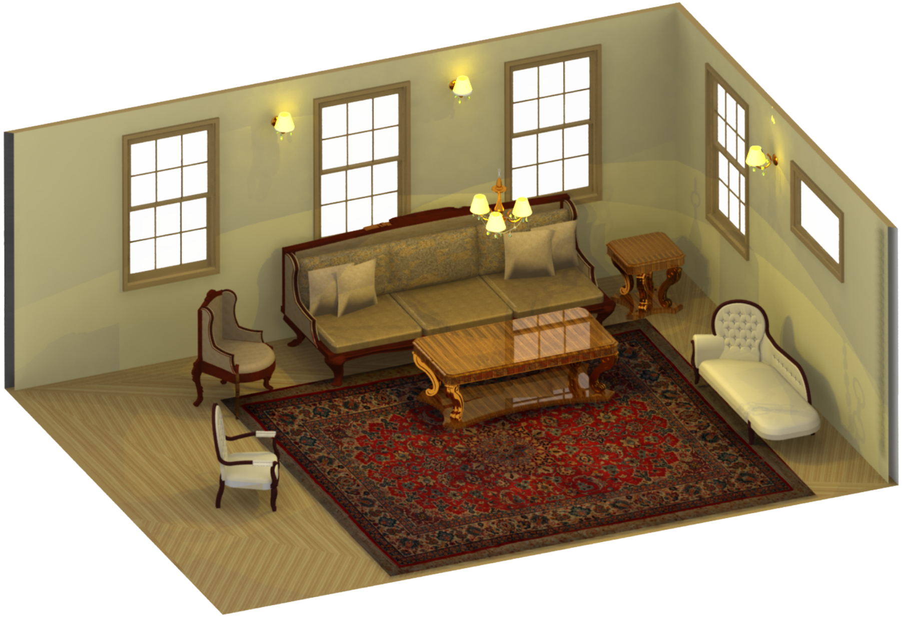
LIGHTING / “SHADING”

Models the interaction of light with surfaces. Generates a per (pixel/vertex) colour.

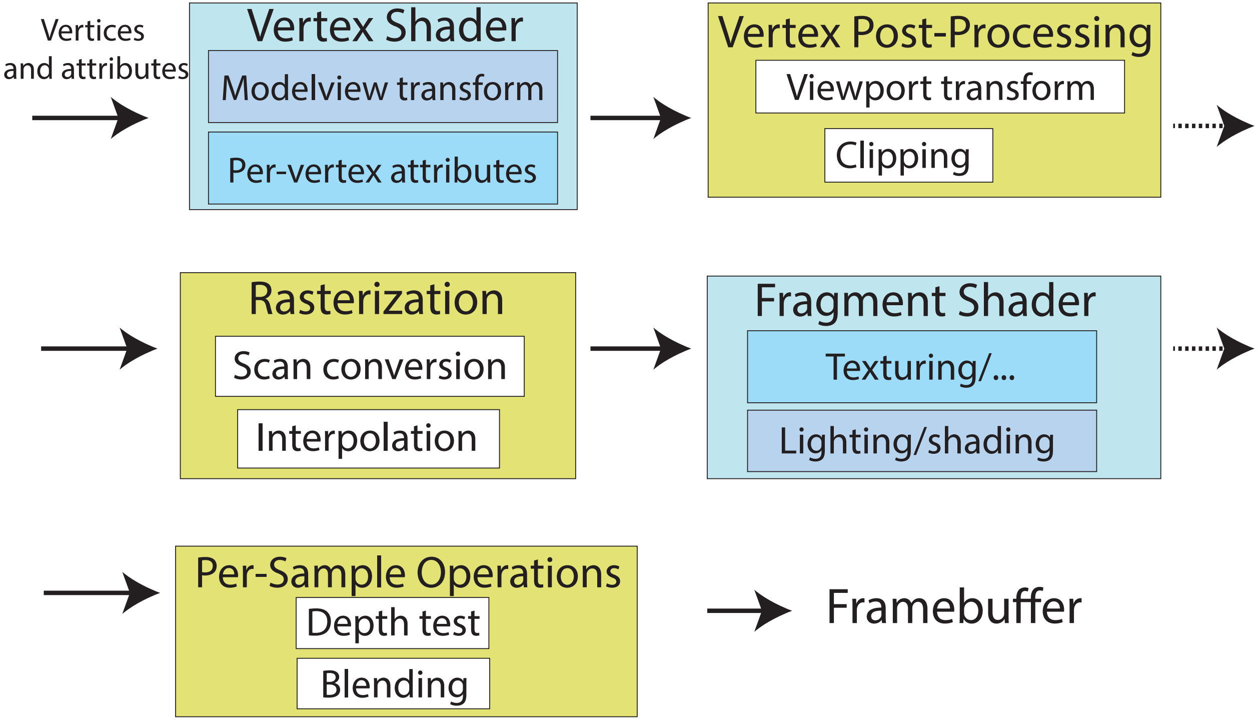


FACTORS:

- Light sources
  - Location, type & color
- Surface materials
  - How surfaces reflect light
- Transport of light
  - How light moves in a scene
- Viewer position

- How can we do this in the pipeline?

THE RENDERING PIPELINE



Vertices and attributes →

Vertex Shader
- Modelview transform
- Per-vertex attributes

→

Vertex Post-Processing
- Viewport transform
- Clipping

→

Rasterization
- Scan conversion
- Interpolation

→

Fragment Shader
- Texturing/...
- Lighting/shading

→

Per-Sample Operations
- Depth test
- Blending

→ Framebuffer

ILLUMINATION MODELS/ALGORITHMS

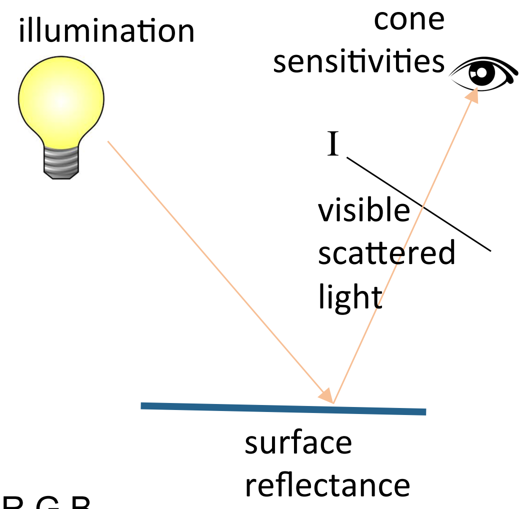
Local illumination – Fast

- assumes single-bounce path:
  light – surface – eye

Global illumination – Slow

- (More) Physically based computation of light transport throughout the scene

THE BIG PICTURE



- Light: energy in a range of wavelengths
  - White light – all wavelengths
  - Colored (e.g. red) – subset of wavelengths
- Surface "color" – reflected wavelength
  - White – reflects all lengths
  - Black – absorbs everything
  - Colored (e.g. red) absorbs all but the reflected color
- Multiple light sources add (energy sums)

- big simplification: just work with three "wavelengths": R,G,B which roughly correspond to the three types of cone-cells in the human eye.
  - white light (1,1,1) x yellow surf (0.8,0.8,0) = yellow visible (0.8,0.8,0)
  - yellow light (1,1,0) x cyan surf (0,1,1) = green visible(0,1,0)

$I_L \quad k_a, k_d \quad$ GLSL

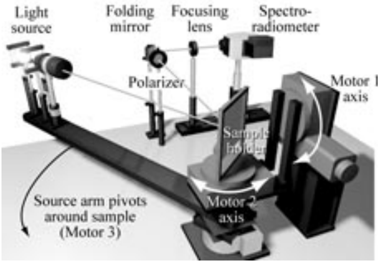
$$\begin{bmatrix}1\\1\\1\end{bmatrix} \cdot \begin{bmatrix}0.8\\0.8\\0\end{bmatrix} = I_L * k_a = \begin{bmatrix}0.8\\0.8\\0\end{bmatrix}$$

$I_L \quad k \quad I$

$$\begin{bmatrix}1\\1\\0\end{bmatrix}\begin{bmatrix}0\\1\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\end{bmatrix}$$

MATERIALS

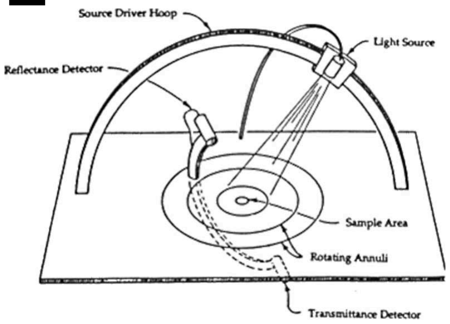
- Surface reflectance:
    - Illuminate surface point with a ray of light from different directions
    - How much light is reflected in each outgoing direction?

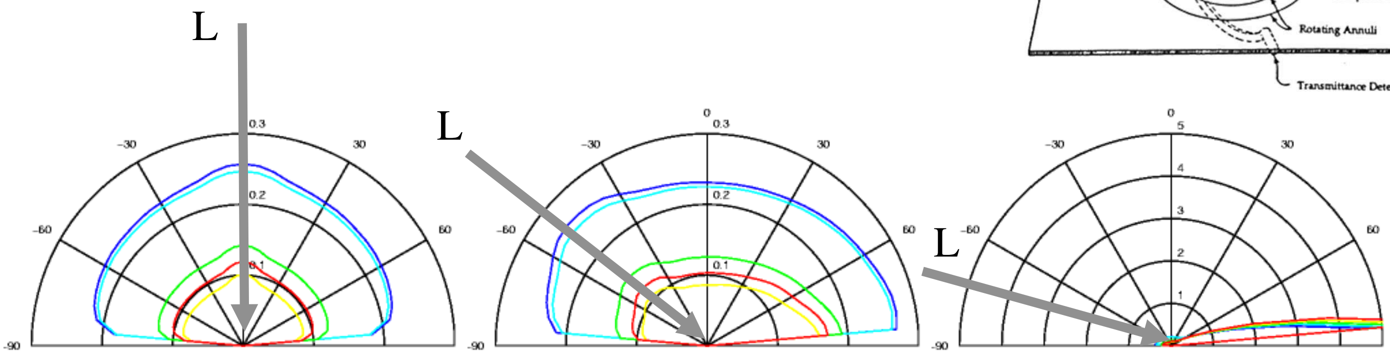


BRDF

BIDIRECTIONAL REFLECTANCE DISTRIBUTION MODEL

- How much light is reflected in every outgoing direction for each possible incoming direction of light?





BRDF of spray-painted latex blue paint, Pratt & Lambert, Vapex Interior Wall Base 1, Color #1243, Cal.III.
http://www.graphics.cornell.edu/online/measurements/reflectance/housepaints/

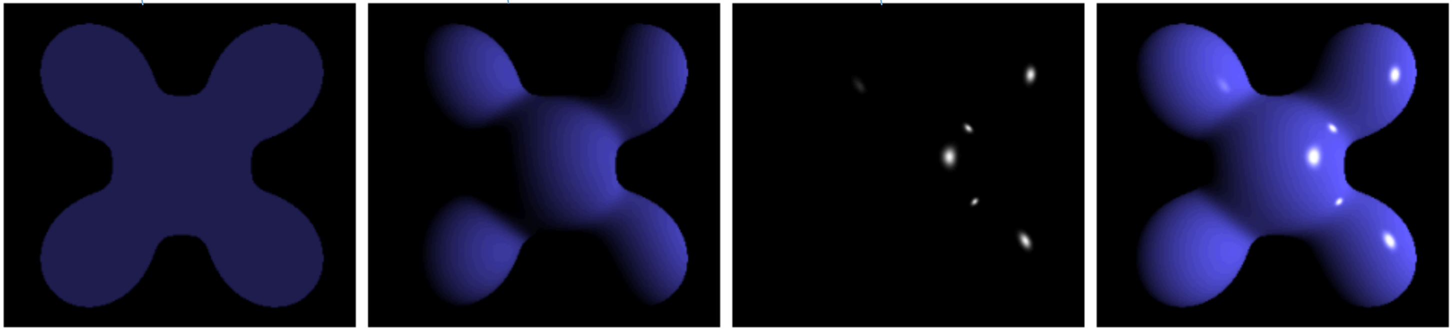
PHONG MODEL

Widely used;
you will implement this!

Named after Phong, who developed the model for the specular term.

$$\overset{\text{Hack}}{\boxed{I_a k_a}} + \sum_L I_L k_d \overset{\text{physics based.}}{\boxed{(N \cdot L)}} + I_L k_s \overset{\text{Hack}}{\boxed{(R \cdot V)^n}}$$

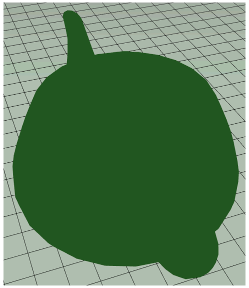
ambient + diffuse + specular = total



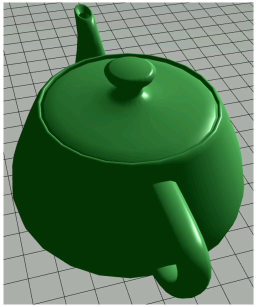
AMBIENT TERM

- simple hack that provides a default minimal illumination

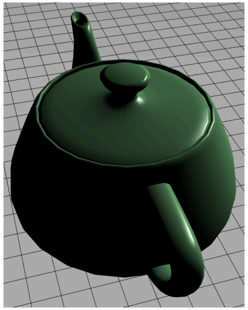
$$\underline{I_a k_a} + \sum_L I_L k_d (N \cdot L) + I_L k_s (R \cdot V)^n$$



ambient term



with ambient light



without ambient light

scene property
ambient light color

surface property:
ambient color

$$\begin{array}{cccc} I & & I_a & k_a \\ \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix} & = & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix} \end{array}$$

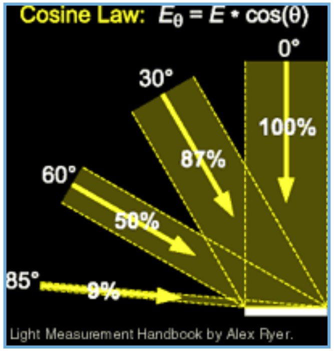
apply individually to RGB components

see: http://www.realtimerendering.com/teapot/

DIFFUSE TERM

“Lambertian Surface”

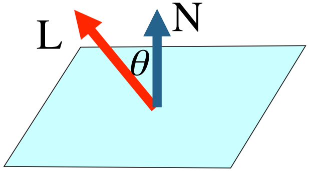
$$I_a k_a + \sum_L \underline{I_L k_d (N \cdot L)} + I_L k_s (R \cdot V)^n$$



Intuition: Brightness depends on the angle of the surface with respect to the incoming light direction. Sharper angles mean that the surface receives less light.

$$N \cdot L = \cos(\theta)$$

only consider angles from 0-90,
i.e., if N.L negative, then set to zero

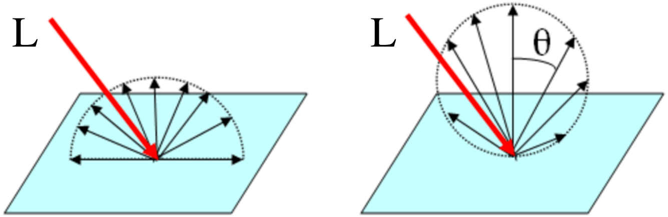


N,L should be normalized !

DIFFUSE TERM

The apparent brightness (the reflected radiance)
is ***independent*** of the viewing direction.

Two correct interpretations:



reflected radiance reflected intensity

[http://www.oceanopticsbook.info/view/surfaces/lambertian_brdfs]

SPECULAR (PHONG) TERM

$$I_a k_a + \sum_L I_L k_d (N \cdot L) + \underline{I_L k_s \boxed{(R \cdot V)}^n}$$

reflected ray

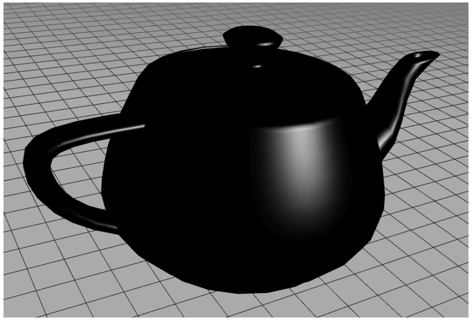
viewing direction

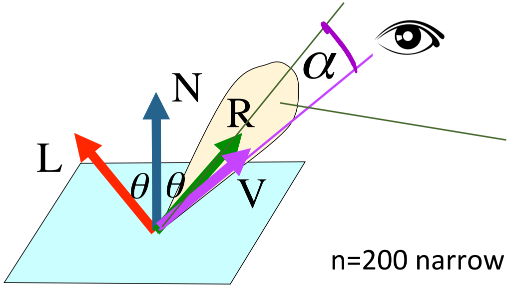
"shininess"

$$R \cdot V = \cos(\alpha)$$

$\cos(\alpha)$

Many surfaces are not perfect mirrors, but still reflect much light in the general direction of the reflected ray. As we get further away from the reflected ray, we see less light being reflected in those directions.





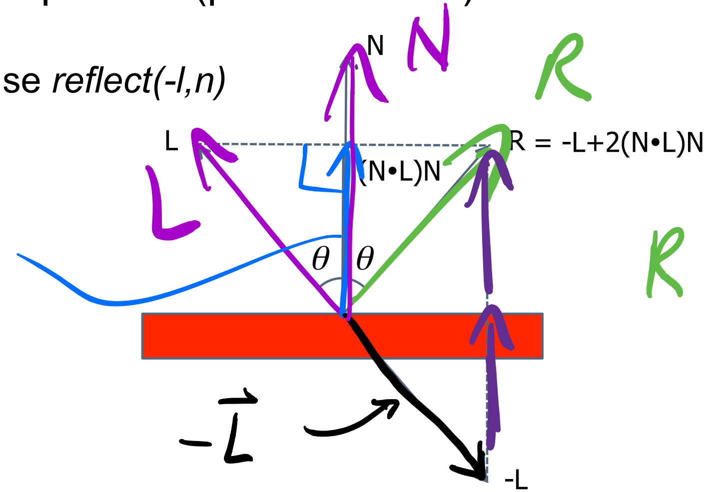
n=200 narrow lobe, highly specular

n=10 wide lobe

$R = \text{reflect}(I, N)$

SPECULAR TERM: COMPUTATIONS

- Geometry of specular (perfect mirror) reflection
  - Snell's law
  - In GLSL: use *reflect(-l,n)*

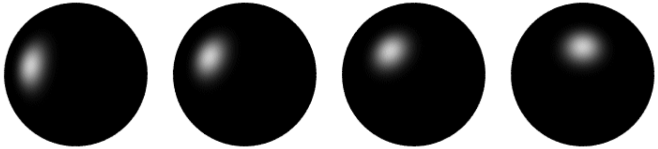


$$\underbrace{(\vec{N}\cdot\vec{L})}_{\text{Scalar}}\vec{N}$$

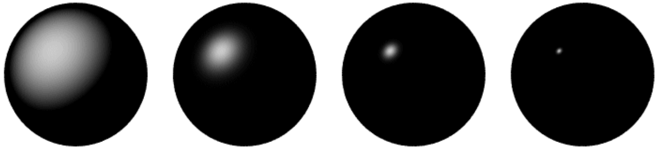
$$R = -\vec{L} + 2(N\cdot L)\vec{N}$$

PHONG EXAMPLES

varying light position



varying n_s



The Phong specular term is an emperical model, i.e., not physics-based.

It generally produces a “plastic” appearance.

(skip)

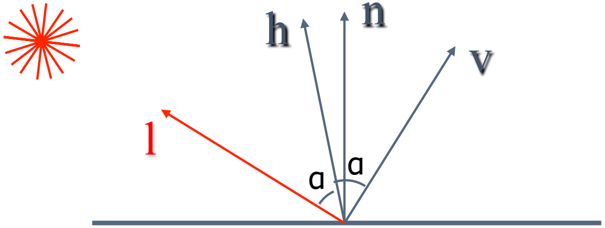
ALTERNATIVE: BLINN-PHONG MODEL

(specular term)

(Jim Blinn, 1977)

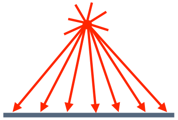
- Variation with better physical interpretation
  - **h**: halfway vector; r: roughness

$$I_{specular} = k_s \cdot (\mathbf{h} \cdot \mathbf{n})^{1/r} \cdot I_{light}; \text{with } \mathbf{h} = (\mathbf{l} + \mathbf{v})/2$$

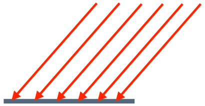


LIGHT SOURCE TYPES

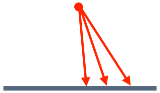
- Point Light
  - light originates at a point
  - defined by **location** only



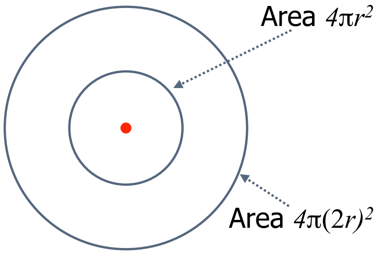
- Directional Light (point light at infinity)
  - light rays are parallel
  - Rays hit a planar surface at identical angles
  - defined by **direction** only



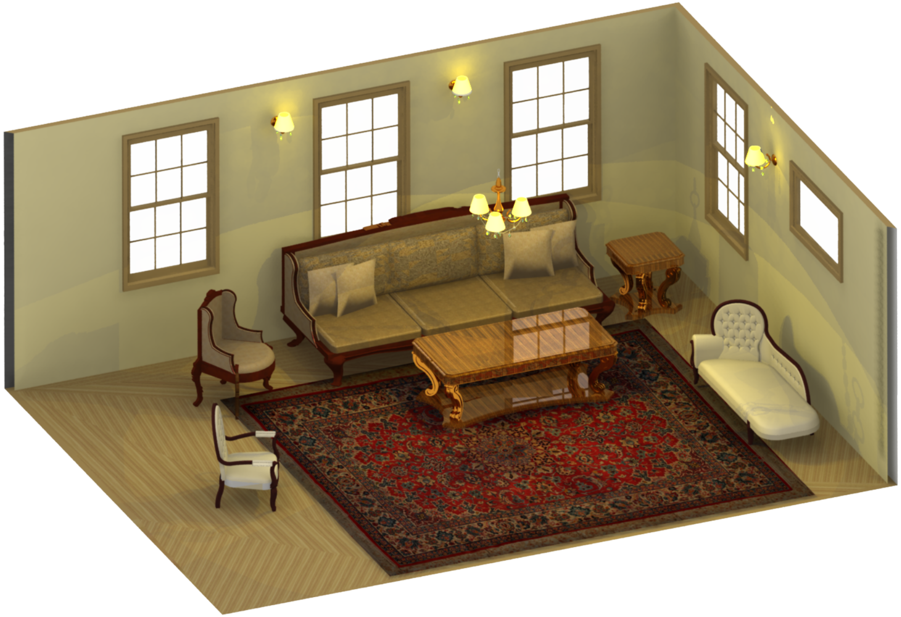
- Spot Light
  - point light with limited angles
  - defined by **location, direction, and angle range**



- Light source fall-off: intensity drops with distance squared



WHICH LIGHTS/MATERIALS ARE USED HERE?



LIGHT AND MATERIAL SPECIFICATION

- Light source: amount of RGB light emitted
  - value = intensity per channel

  e.g., (1.0,0.5,0.5)
  - every light source emits ambient, diffuse, and specular light
- Materials: amount of RGB light reflected
  - value represents percentage reflected
    e.g., (0.0,1.0,0.5)
- Interaction: multiply components
  - Red light (1,0,0) x green surface (0,1,0) = black (0,0,0)

$$\begin{bmatrix}1\\0\\0\end{bmatrix} \times \begin{bmatrix}0\\1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

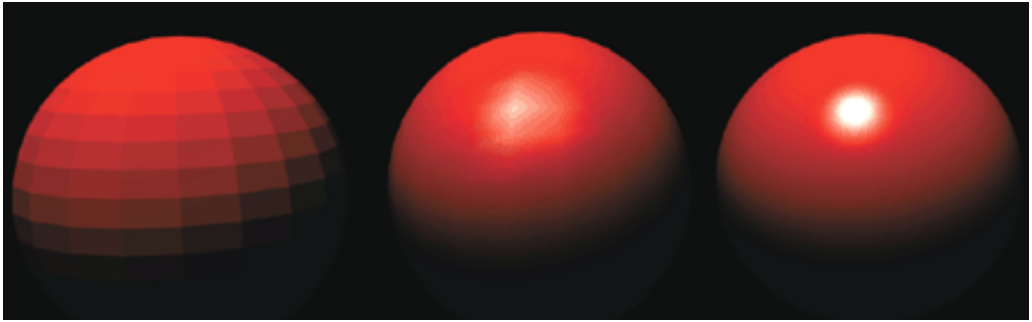
NOTES ON SHADING

- Typically compute lighting model in VCS. Therefore need:
  - VCS Vertex Coordinates, VCS Normals, VCS Light Positions
- How often to compute the lighting model?

per polygon

per vertex, then interpolate colors

per pixel, using interpolated normals



flat shading

“Gouraud shading”

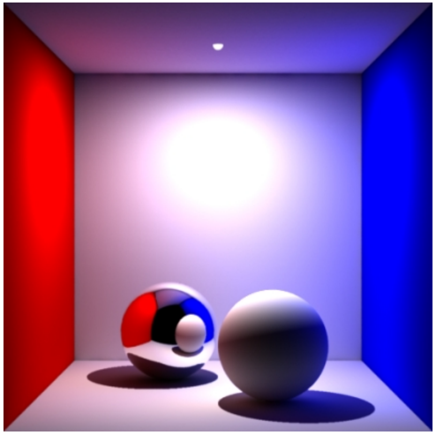
per-vertex lighting (in vertex shader) then interpolate

per-pixel shading

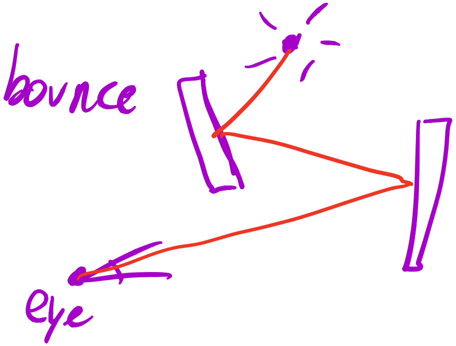
(in fragment shader)

Image © Intergraph Computer Systems

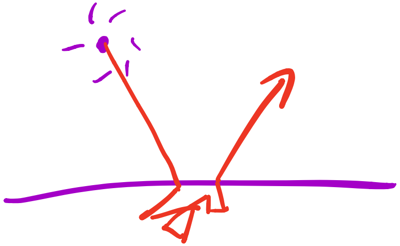
GLOBAL ILLUMINATION



modeling light that takes >1 bounce between the light and the eye:



SUBSURFACE SCATTERING



QUESTIONS

(e) $\begin{bmatrix}0\\1\\0\end{bmatrix} \times \begin{bmatrix}1\\1\\0\end{bmatrix} = \begin{bmatrix}0\\1\\0\end{bmatrix}$ green

- What do we see when a yellow surface is lit with:
  (a) white light?
  (b) blue light?
  (c) red light?
  (d) yellow light?
  (e) green light?

$\begin{bmatrix}\text{red}\\\text{green}\\\text{blue}\end{bmatrix}$

(a) $\begin{bmatrix}1\\1\\1\end{bmatrix} \times \begin{bmatrix}1\\1\\0\end{bmatrix} = \begin{bmatrix}1\\1\\0\end{bmatrix}$ yellow

(b) $\begin{bmatrix}0\\0\\1\end{bmatrix} \times \begin{bmatrix}1\\1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}$ black

(c) $\begin{bmatrix}1\\0\\0\end{bmatrix} \times \begin{bmatrix}1\\1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\end{bmatrix}$ red.

(d) $\begin{bmatrix}1\\1\\0\end{bmatrix} \times \begin{bmatrix}1\\1\\0\end{bmatrix} = \begin{bmatrix}1\\1\\0\end{bmatrix}$ yellow

- Do specular highlights have the colour of the light or the surface?

Specular highlights have the colour of the light, not the surface.
(some metals are a partial exception to this)

QUESTIONS

- Give ~~the~~ approximate Phong equation parameters that are needed to render a shiny yellow material with a single white light.

$$I = I_a k_a + I_L k_d (N \cdot L) + I_L k_s (R \cdot V)^n$$

$$I = \begin{bmatrix}1\\1\\1\end{bmatrix}\begin{bmatrix}0.2\\0.2\\0\end{bmatrix} + \begin{bmatrix}1\\1\\1\end{bmatrix}\begin{bmatrix}0.6\\0.6\\0\end{bmatrix}(N \cdot L) + \begin{bmatrix}1\\1\\1\end{bmatrix}\begin{bmatrix}0.7\\0.7\\0.7\end{bmatrix}(R \cdot V)^{100}$$

- What is the colour of the darkest pixel this model can produce?

$$I_a k_a = \begin{bmatrix}0.2\\0.2\\0\end{bmatrix}$$

- What is the colour of the brightest pixel this model can produce?

$$I = \begin{bmatrix}0.2\\0.2\\0\end{bmatrix} + \begin{bmatrix}0.6\\0.6\\0\end{bmatrix} + \begin{bmatrix}0.7\\0.7\\0.7\end{bmatrix} = \begin{bmatrix}1.5\\1.5\\0.7\end{bmatrix} \xRightarrow{\text{clamp}} \begin{bmatrix}1\\1\\0.7\end{bmatrix}$$