CSI 3202 Micro-Computer Graphics Lighting & Shading

Presenter: Girendra Persaud
University of Guyana

Outline

- What is Light?
- How to model light?
- Shading concept
- Light & Matter
- Approaches…
- Variables
- Material Properties
- Light Sources
- Phong Reflection Model
- Steps to add light to your scene
- Questions?
- Review Questions

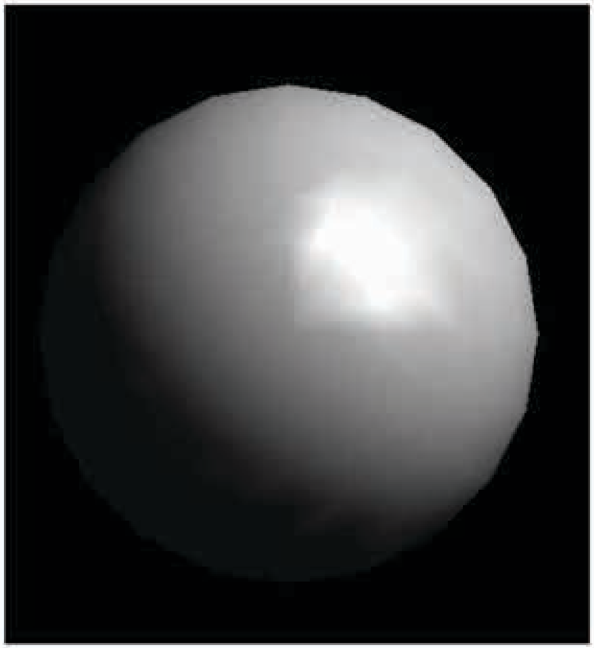
What is light?

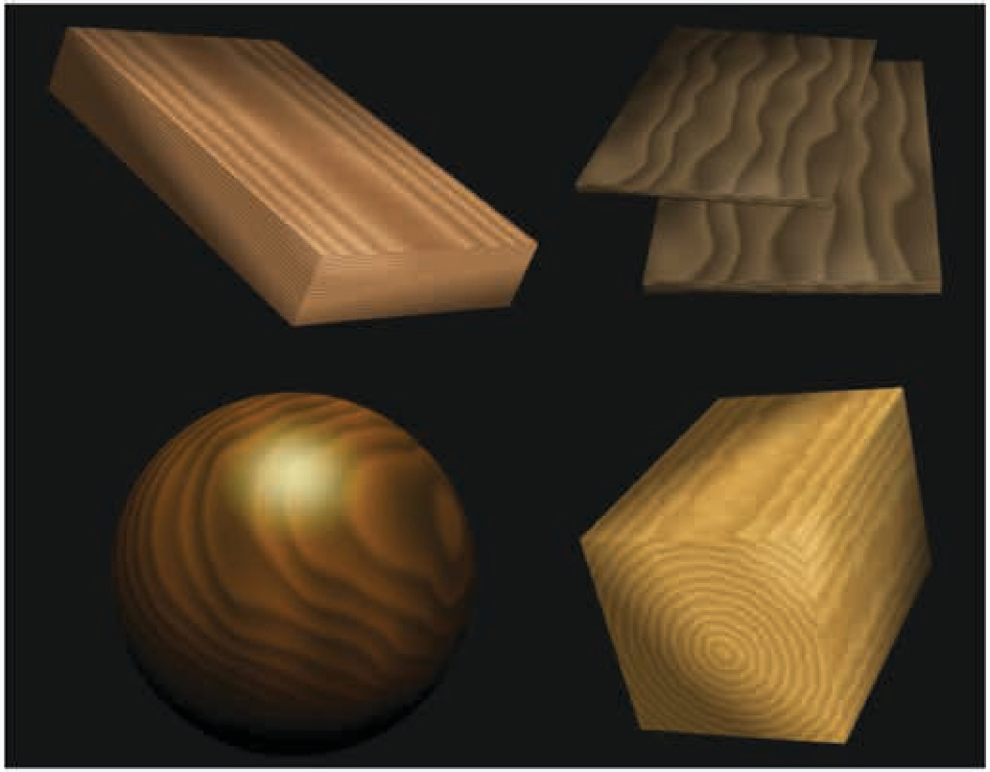
- Light is simply a name for a range of electromagnetic radiation that can be detected by the human eye*

*http://library.thinkquest.org/27356/p_index.htm

How can we represent this in CG?

- ?

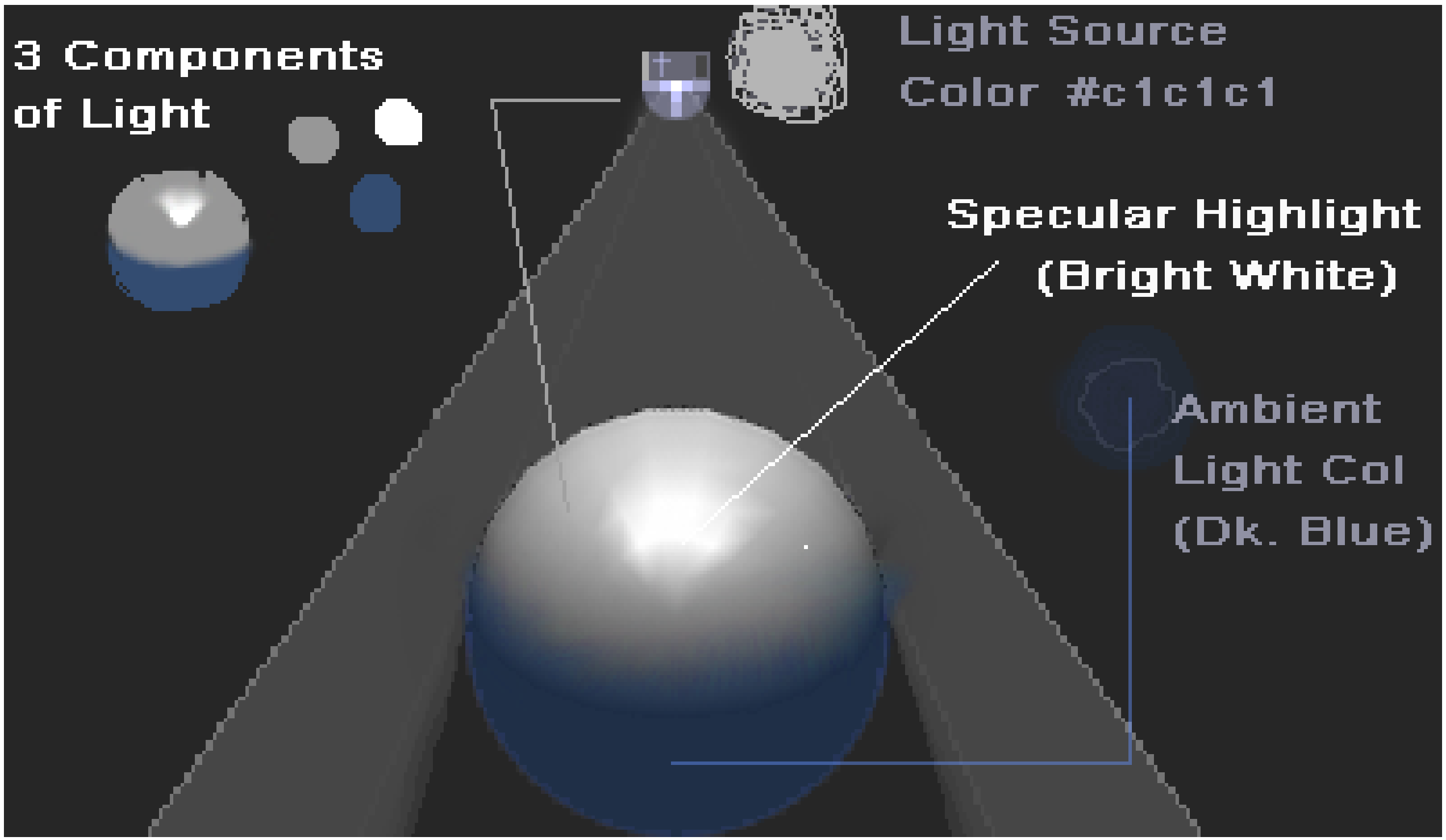




Shades

- If we look at a photograph of a lit sphere, we see not a uniformly colored circle but rather a circular shape with many gradations, or **shades,** of color…
- This gives 2D images a 3D look…

How can light be represented?



Lighting/Shading Approach

- Our aim is to add shading to a fast pipeline graphics architecture.

- The calculations depend only
  - on the material properties assigned to the surface,
  - the local geometry of the surface,
  - and the locations and properties of the light sources

Light & Matter

- A surface can either emit or reflect Light
- The color of a point on an object is determined by multiple interactions among light sources…
- The recursive reflection results to subtle shading, – the *rendering equation*

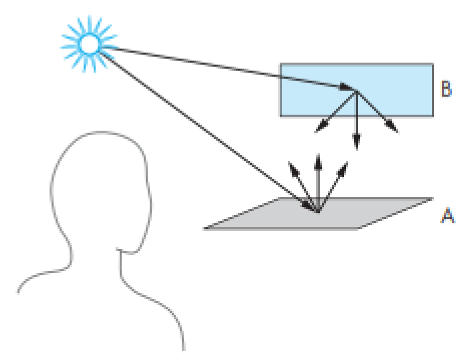
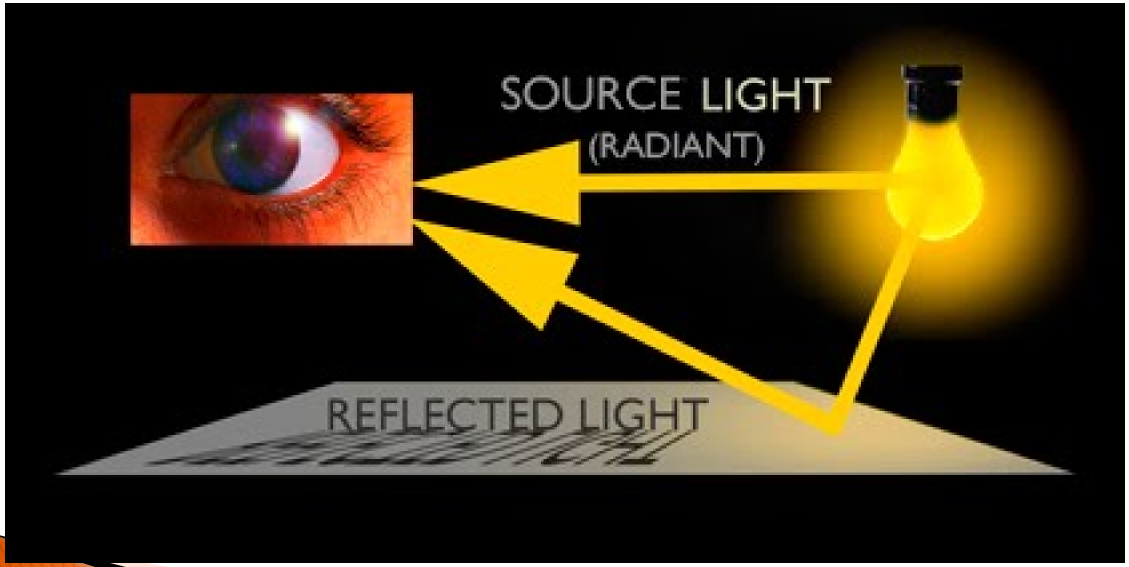


FIGURE 5.1 Reflecting surfaces.

Approximate Approaches

- Ray Tracing
- Radiosity (global illumination algorithm)
- Phong reflection model (also called Phong illumination or Phong lighting)



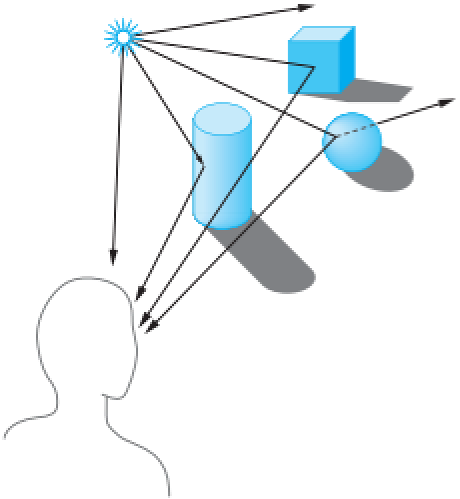
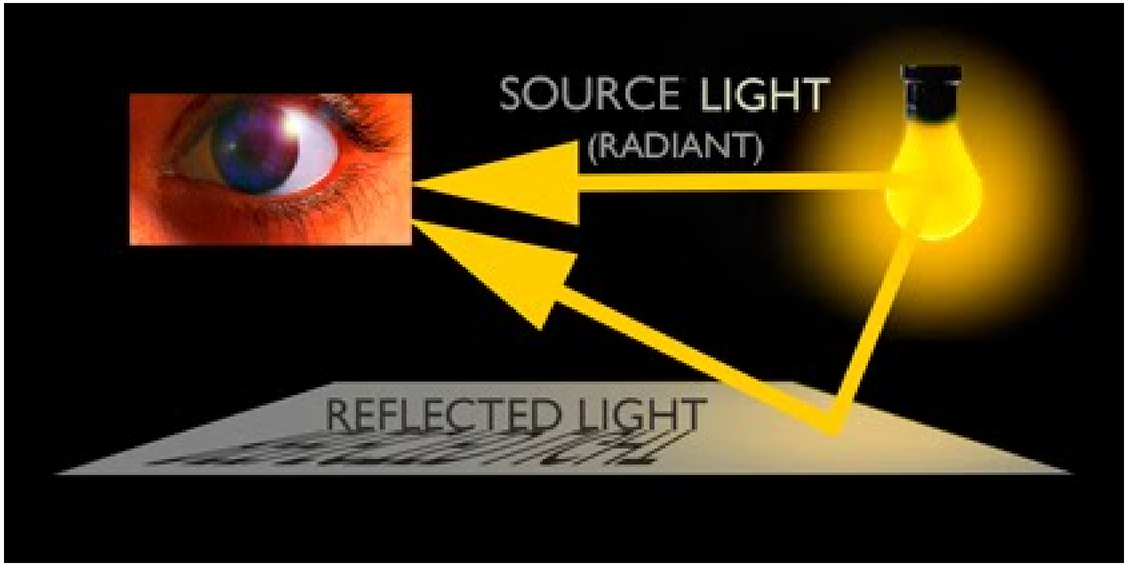


FIGURE 5.2 Light and surfaces.

Problem Variables

- The Material Properties
- The Light Source (Properties, Location)
- The Viewer (COP)



Material Properties

- Specular Surfaces
  - Mirror, shiny objects, smooth reflective surfaces
- Diffuse Surfaces
  - Matte, flat paint, terrain
- Translucent Surfaces
  - Glass, water, tint

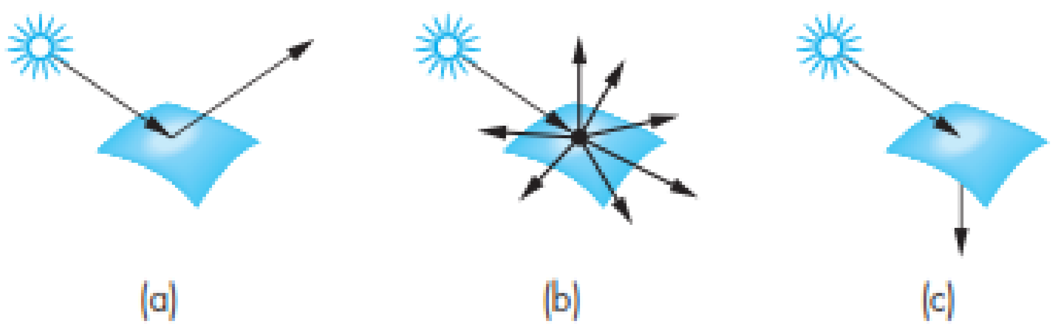


FIGURE 5.4 Light–material interactions. (a) Specular surface. (b) Diffuse surface. (c) Translucent surface.

Light Sources

- Source
  - Self-emission
  - Reflection
- P (x,y,z)
- Energy emitted at each wavelength λ
- Direction of emission (θ , ϕ)
- illumination function $I(x, y, z , \theta , \phi, \lambda)$

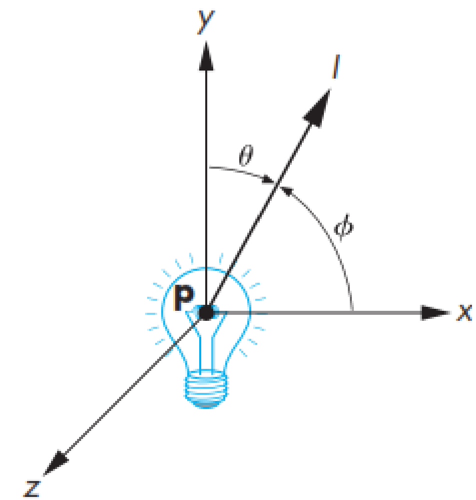


FIGURE 5.5 Light source.

Light Sources

- Color
  - Light source can emit in varying frequencies (different colors)
  - For simplicity we only consider the three primaries at varying intensities
  - Three-component intensity

$$\mathbf{I} = \begin{bmatrix} I_{\mathrm{r}} \\ I_{\mathrm{g}} \\ I_{\mathrm{b}} \end{bmatrix},$$

Light Sources

- Ambient Light
  - Equally scattered from all direction on the surface
  - Source direction negligible
  - Little or no shadow
  - Uniform lighting
  - Illumination intensity, I_a where

$$\mathbf{I}_{\mathrm{a}} = \begin{bmatrix} I_{\mathrm{ar}} \\ I_{\mathrm{ag}} \\ I_{\mathrm{ab}} \end{bmatrix}$$

Light Sources

- Point Sources
  - Emit equally in all directions

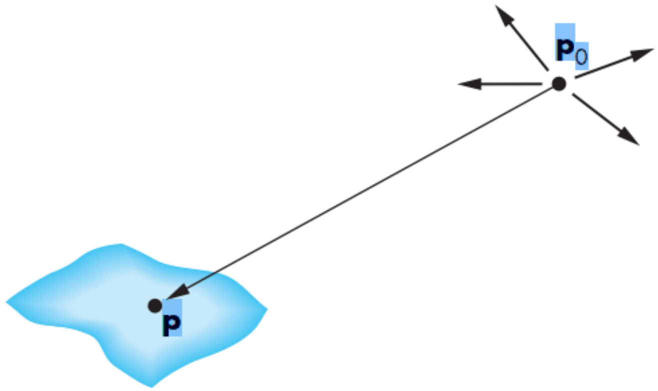


FIGURE 5.7 Point source illuminating a surface.

$$\mathbf{I}(\mathbf{p}_0) = \begin{bmatrix} I_r(\mathbf{p}_0) \\ I_g(\mathbf{p}_0) \\ I_b(\mathbf{p}_0) \end{bmatrix}$$

- The intensity of illumination received from a point source is proportional to the inverse square of the distance between the source and surface

Point Sources

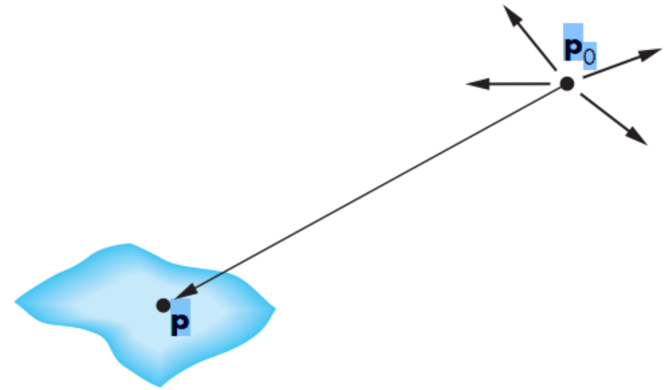


FIGURE 5.7 Point source illuminating a surface.

- Hence, intensity at P

$$\mathbf{i}(\mathbf{p}, \mathbf{p}_0) = \frac{1}{|\mathbf{p} - \mathbf{p}_0|^2}\mathbf{I}(\mathbf{p}_0).$$

- Easy to use but does not resemble reality
- High contrast images…
- Can add ambient light to eliminate ill effects

Light Sources

- Spotlights
  - Narrow range of angle through which light is emitted
  - If the angle = 180, spotlight = point source
  - Intensity is a function of the angle at ϕ
  - Light is brighter in the center
  - Intensity at a point on the surface is $\cos^e \phi$, *where e determines how rapidly light intensity drops off*
  - *If both s, l are unit vectors then $\cos \phi = s.l$ (dot product)*

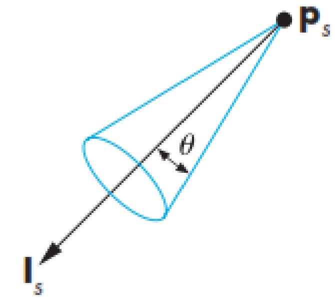


FIGURE 5.9 Spotlight.

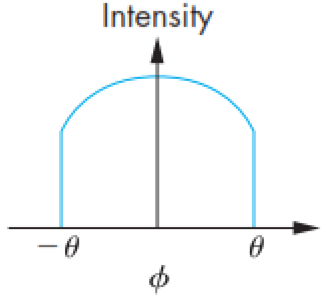


FIGURE 5.10 Attenuation of a spotlight.

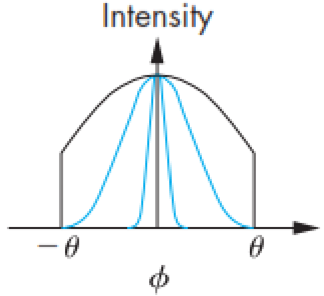


FIGURE 5.11 Spotlight exponent.

Light Sources

- Distant Light Sources
  - shading calculations require the direction from the point on the surface to the light source position
  - If the light source is far away the direction vector (of the light source) remains constant
  - All points visible to the light will have the same intensity

$$\mathbf{P}_0 = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

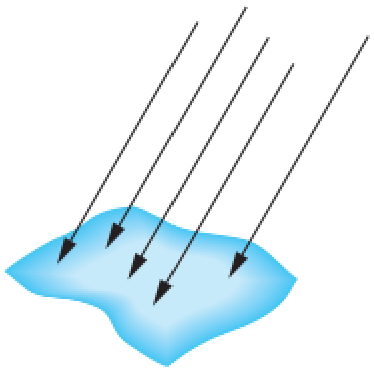
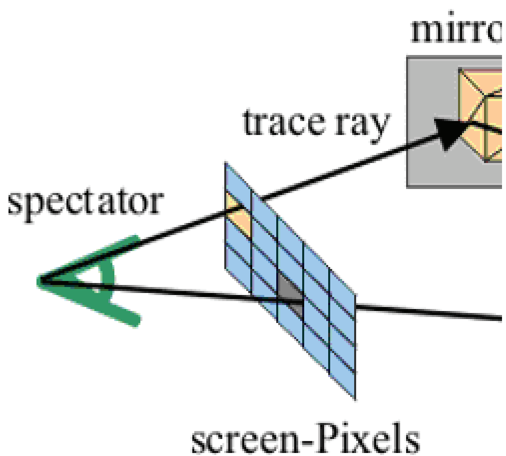


FIGURE 5.12 Parallel light source.

Phong Reflection Model

- Similar to ray tracing but it only considers a single interaction



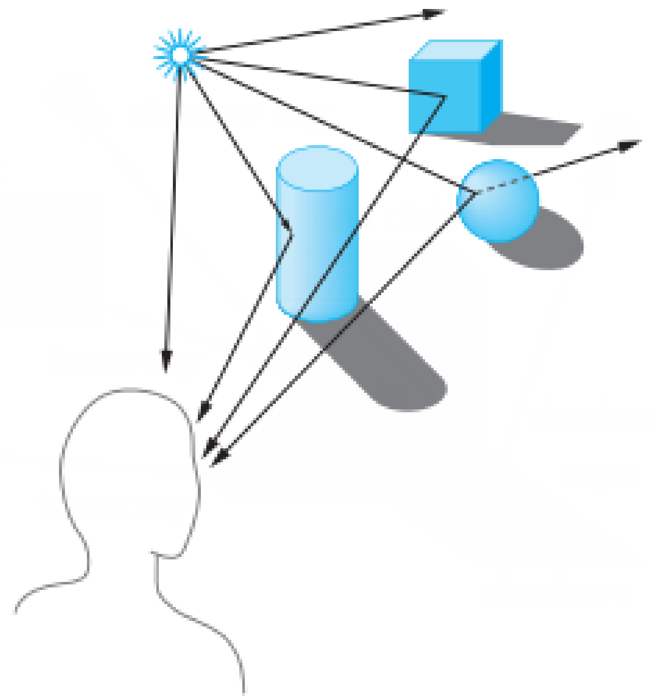
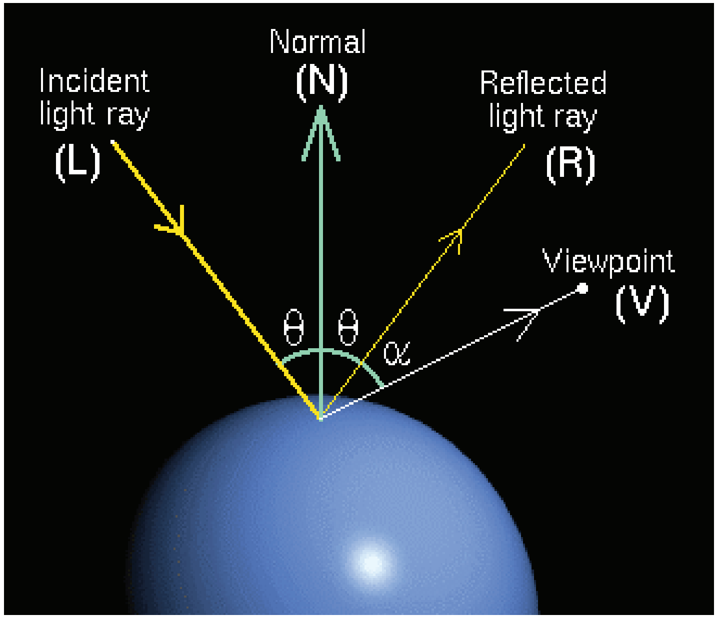


FIGURE 5.2 Light and surfaces.

Phong Reflection Model



- Uses 4 vectors
- It supports
  - ambient,
  - diffuse and
  - specular light material-interactions

Phong Reflection Model

- Each source can have separate ambient, diffuse, specular components for eact primaries (RGB)
- At any point on a surface we can compute the illumination matrix for the ith light source

$$\mathbf{L}_i = \begin{bmatrix} L_{i\mathrm{ra}} & L_{i\mathrm{ga}} & L_{i\mathrm{ba}} \\ L_{i\mathrm{rd}} & L_{i\mathrm{gd}} & L_{i\mathrm{bd}} \\ L_{i\mathrm{rs}} & L_{i\mathrm{gs}} & L_{i\mathrm{bs}} \end{bmatrix}$$

- vec3 light_i_ambient, light_i_diffuse, light_i_specular;

Phong Reflection Model

- We assume that we can compute how much of each of the incident lights is reflected at the point of interest
- For example, for the red diffuse term from source *i*, L_{ird}, we can compute a reflection term R, and the latter's contribution to the intensity at **p is** $R_{ird} L_{ird}$

$$\mathbf{R}_i = \begin{bmatrix} R_{ira} & R_{iga} & R_{iba} \\ R_{ird} & R_{igd} & R_{ibd} \\ R_{irs} & R_{igs} & R_{ibs} \end{bmatrix}$$

Phong Reflection Model

- We can then compute the contribution for each color source by adding the ambient, diffuse, and specular components.
- For example, the red intensity that we see at **p from** source *i* is

$$
\begin{aligned}
I_{ir} &= R_{ira}L_{ira} + R_{ird}L_{ird} + R_{irs}L_{irs} \\
&= I_{ira} + I_{ird} + I_{irs}.
\end{aligned}
$$

Review

- CG API represents light in three components together with varying intensities
    - red
    - green
    - blue
- The color of light sources is characterized by the amount of red, green, and blue light they emit
- Distance play a part to determine (calculate) the intensity
- Material of surfaces is characterized by the percentage of the incoming red, green, and blue components that is reflected in various directions

Material Properties

- It is important to specify your material properties in your scene because once color is enabled everything in the scene will be painted by the color of the light

- In reality, a red ball in white light does not turn white

How to specify Material Properties?

- Example:

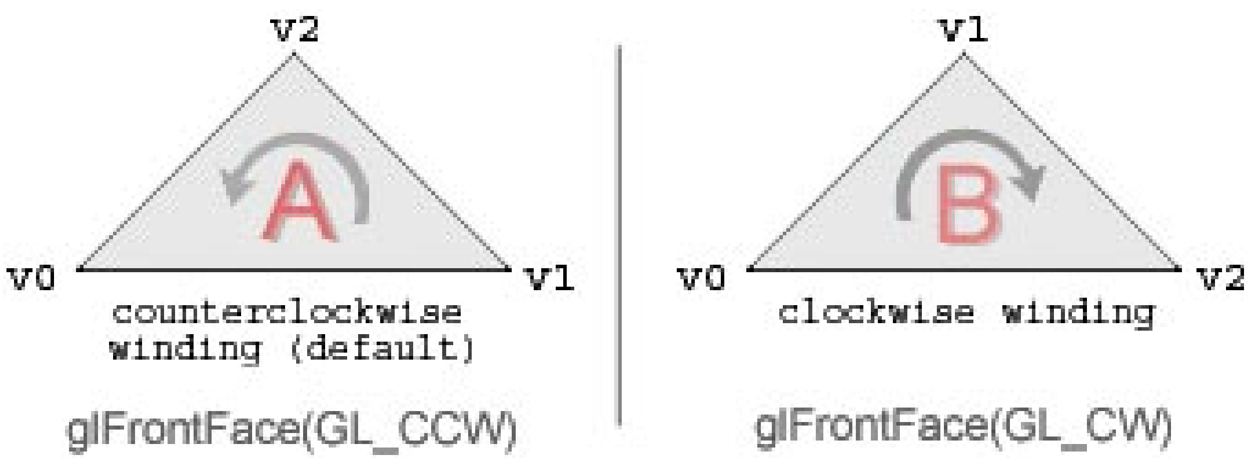
  - float mcolor[] = { 1.0f, 0.0f, 0.0f, 1.0f };

  - glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mcolor);

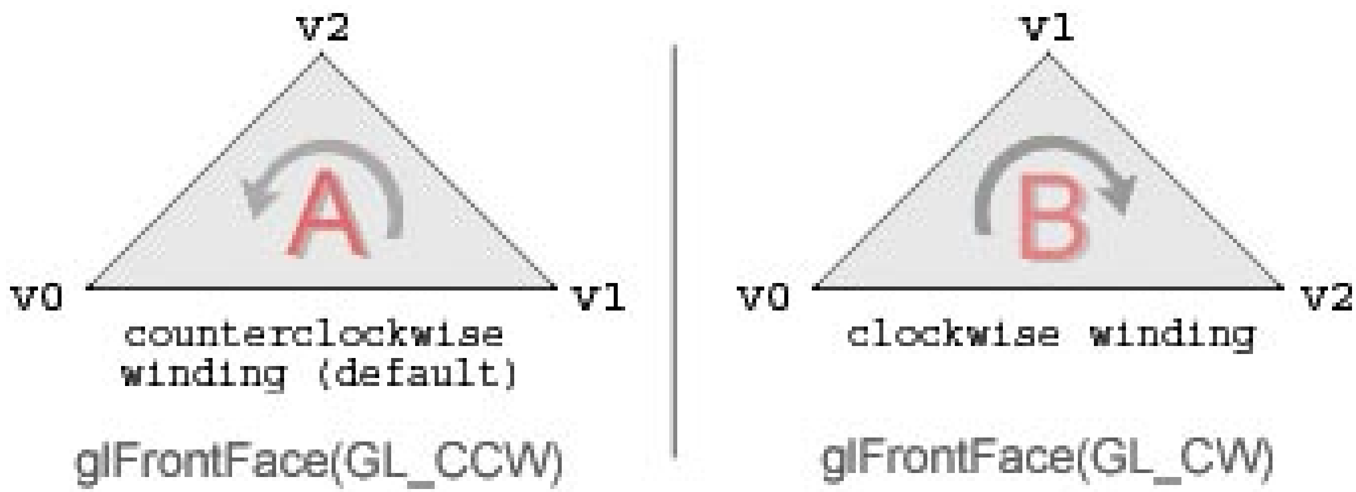
- Where: GL_FRONT indicates which face of the polygon should reflect the light specified by mcolor

The face of the polygon?

- Polygon has two sides
  - GL_BACK and GL_FRONT
  - Vertices specified in CW and CCW directions
  - Default is CCW, command to specify with way to draw, glFrontFace(GL_CCW);



The face of the polygon?



- After a call to glFrontFace(GL_CW); polygons drawn with vertices in CW that will be the front
- You can specify lighting to both sides of the polygon by GL_FRONT_AND_BACK

Steps for adding light…

- Define normal vectors for each vertex of all the objects. These normals determine the orientation of the object relative to the light sources

```
Example: glBegin(GL_QUADS);
glNormal3f(1,1,1);
glVertex3f(1,1,1);
```

- Create, select, and position one or more light sources.
- Create and select a lighting model, which defines the level of global ambient light and the effective location of the viewpoint (for the purposes of lighting calculations)
- Define material properties for the objects in the scene

Questions?

Resources

- Interactive computer graphics: a top down approach with OpenGL / Edward Angel. ISBN: 0-201-38597-X
- http://glprogramming.com/red/chapter05.html
- http://www.swiftless.com/tutorials/opengl/material_lighting.html