CSE328 Fundamentals of Computer Graphics: Concepts, Theory, Algorithms, and Applications

Hong Qin

Department of Computer Science

State University of New York at Stony Brook (Stony Brook University)

Stony Brook, New York 11794--4400

Tel: (631)632-8450; Fax: (631)632-8334

qin@cs.sunysb.edu

http://www.cs.sunysb.edu/~qin

Methods and Techniques for Illumination and Shading (Lighting Models)

Light and Matter

- A very high-level introduction to some concepts and definitions underlying computer graphics and image synthesis.
  – Optics
  – Materials and surfaces
  – Radiometry and Photometry
  – Color
  – Energy transport

Optics

- The study of light has 3 sub-fields
  - *Physical optics*: study of the wave nature of light
  - *Geometric optics*: study of the particle nature of light
  - *Quantum optics*: study of the dual wave-particle nature of light and attempt to construct unified theories to support duality. Wave "packets" called photons
- Computer graphics most concerned with geometric optics (but need some of the others, too)

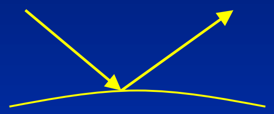
Reflection and Transmission

(Definitions from Glassner1995).

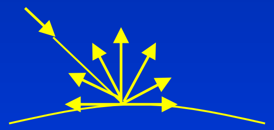
- *Reflection*: "process whereby light of a specific wavelength incident on a material is at least partly propagated outward by the material without change in wavelength."
- *Transmission* (or *refraction*): "process whereby light of a specific wavelength incident on the interface (boundary) between two materials passes (refracts) through the interface without change in wavelength."

Types of Reflection

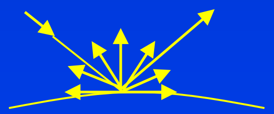
- *Specular* (a.k.a. *mirror* or *regular*) reflection causes light to propagate without scattering.



- *Diffuse* reflection sends light in all directions with equal energy.

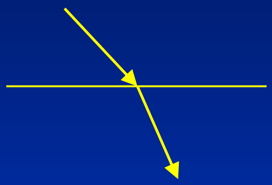


- *Mixed* reflection is a weighted combination of specular and diffuse.

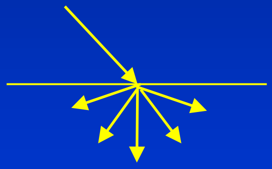


Types of Transmission

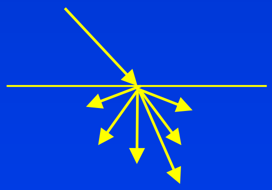
- *Specular* transmission causes light to propagate w/o scattering, as in clear glass.



- *Diffuse* transmission sends light in all directions with equal energy, as in frosted glass.



- *Mixed* transmission is a weighted combination of specular and diffuse transmission.



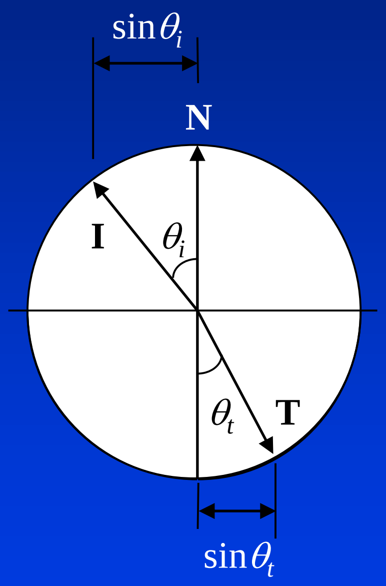
Snell's Law of Refraction

- Governs the geometry of refraction.

$$\eta_i(\lambda)\sin\theta_i = \eta_t(\lambda)\sin\theta_t$$

η_i = IOR of incident medium

η_t = IOR of medium into which the light is transmitted



- If the light is transmitted into a denser medium, it is refracted *toward* the normal of the interface.
- If the light is transmitted into a rarer medium, it is refracted *away* from the normal of the interface.

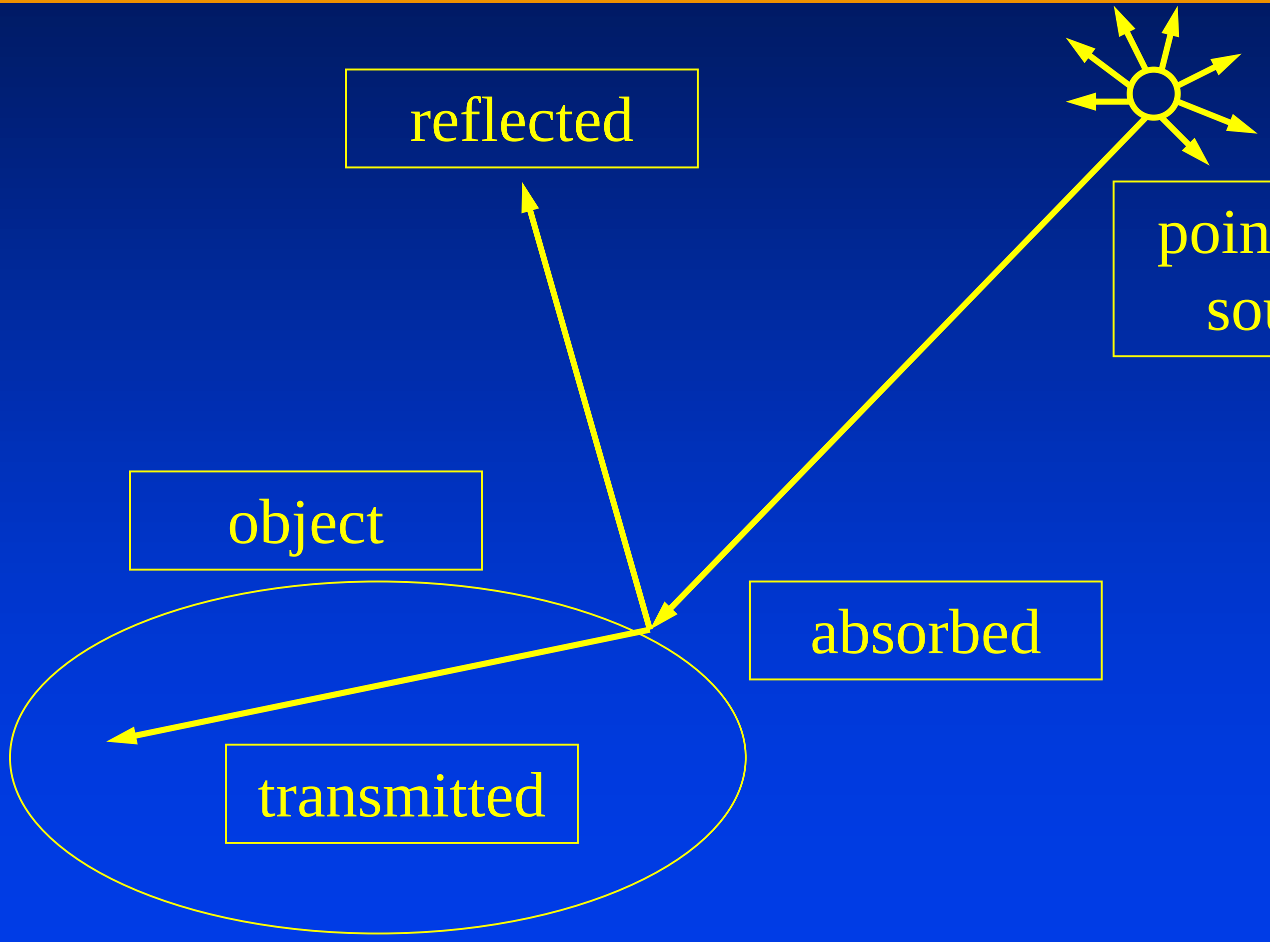
Surface Models

- Perfect mirrors and reflections don't exist.
- Perfect transmission requires a perfect vacuum.
- Real surfaces have some degree of roughness.
  - Even most basic simulation must account for specular and diffuse reflection / transmission.
  - More realism requires accounting for more factors.
    - Wavelength dependence: dispersion, diffraction, interference
    - Anisotropy: angular-dependence of reflectance.
    - Scattering: absorption and re-emission of photons.

Basic Surface Models

- Non-physically based, as used in OpenGL.
  - Materials have ambient, diffuse, and specular colors.
  - Ambient is a very coarse approx. Of light reflected from other surfaces. (Global illumination).
  - Diffuse usually just the “color” of the surface.
  - Specular determines highlight color.

Total Light Decomposition



Illumination and Shading

- *Illumination* and *shading* are two complementary aspects in graphics that add realism to rendered scenes
- Illumination refers to use of lights in virtual world
- Shading refers to effects that lights have on 3D objects in the scene
- Many kinds of *illumination models* and *shading models* are available in 3D graphics

Illumination (Light)

- Without lights a 3D scene is totally dark
- Seek to simulate effects of light
- Simplest type of light is *point light source*
- Light is infinitely far away
- Light rays are parallel
- Is this a good approximation of a light bulb? Flash light? The sun?

Shading Model

- A shading model checks lighting conditions and figures out what surface should look like based on lighting conditions and surface parameters:
  - Amount of light reflected (and which color(s))
  - Amount of light absorbed
  - Amount of light transmitted (passed through)
- Shading model tells us how much incoming light that strikes a surface is: (1) reflected to the eye, (2) absorbed by the object, and (3) transmitted through the object

Shading Model

- Typically in computer graphics, we are concerned with the reflected light – light which bounces off object and enters into our eyes
- Other effects like refraction and translucency require much more sophisticated shading models

Illumination

- Illumination models
  - Light and surfaces
  - Local illumination versus global illumination
  - Phong reflection model
    - Ambient reflection
    - Diffuse reflection
    - Specular reflection
    - Light attenuation
- Polygonal shading
  - Flat Shading
  - Gourand Shading
  - Phong Shading

Illumination Model

- Also called reflection model or lighting model.
- Describe the interaction between the light sources and the surfaces.
- Local illumination models versus global illumination models.
- Local models are ad-hoc, but is fast and easy.
- Global models are more accurate, but much more expensive.

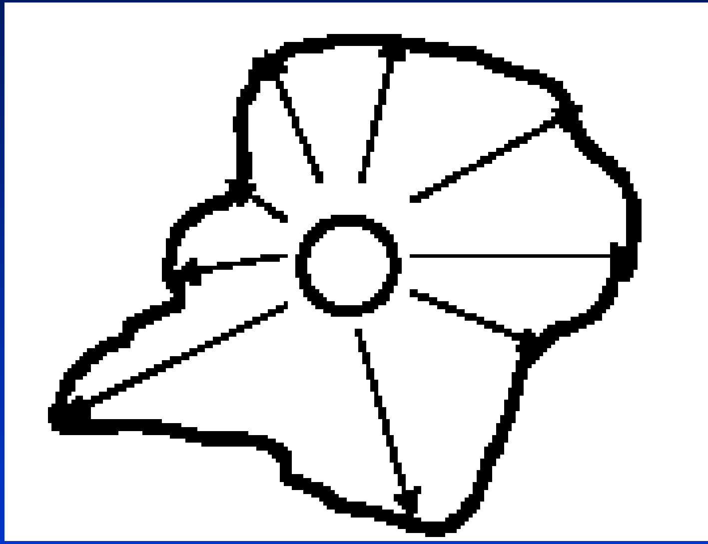
Light Sources

- Point sources
- Spotlights
- Distant light

Radiant Intensity

$$I(\omega) = \frac{d\Phi}{d\vec{\omega}}$$

$$\Phi = \int_{\Omega} I(\omega)\, d(\vec{\omega})$$



For an isotropic point source: $I(\omega) = \Phi/4\pi$

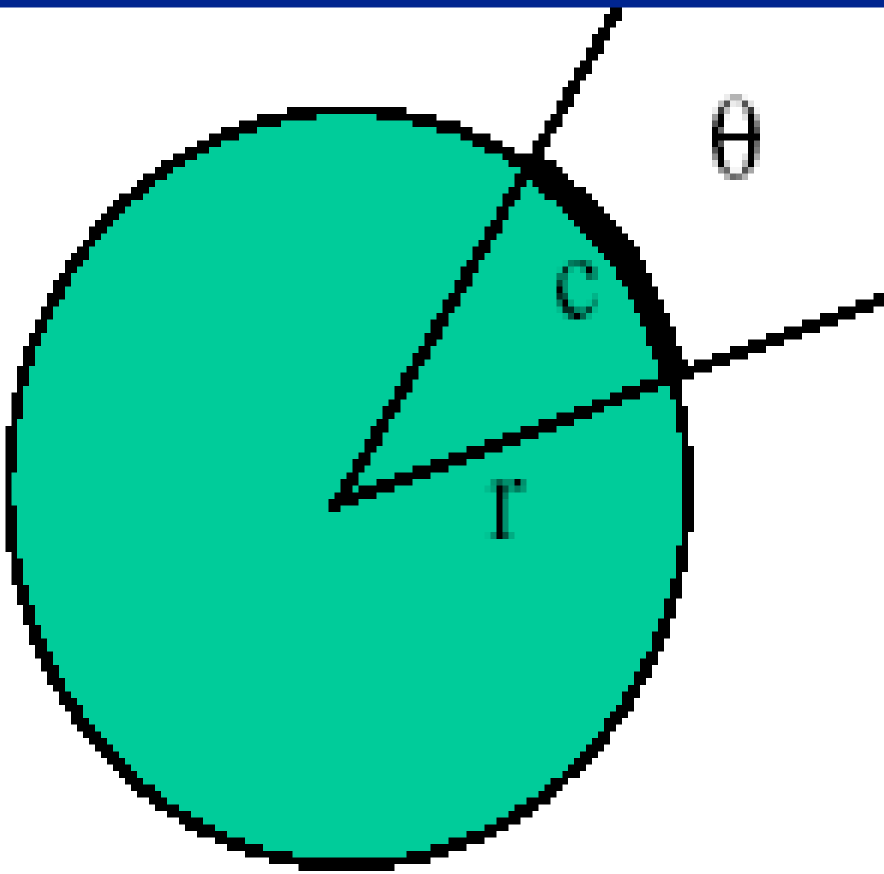
Radiant Intensity

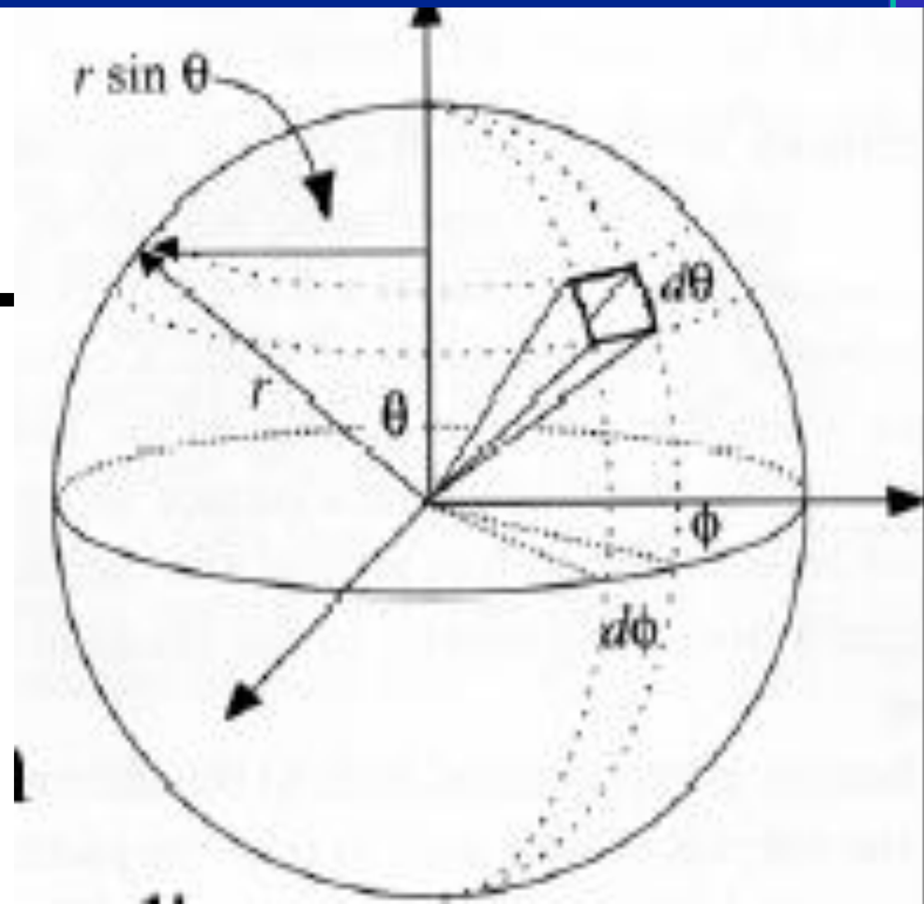
- **Radiant Intensity:** Radiant power per solid angle of a point source
- Units – watts per steradian
- "Intensity" is heavily overloaded
- What is a solid angle?

$$I(\omega) = \frac{d\Phi}{d\vec{\omega}}$$

Angles

- 2D Angle vs. 3D Solid Angle





Solid Angle

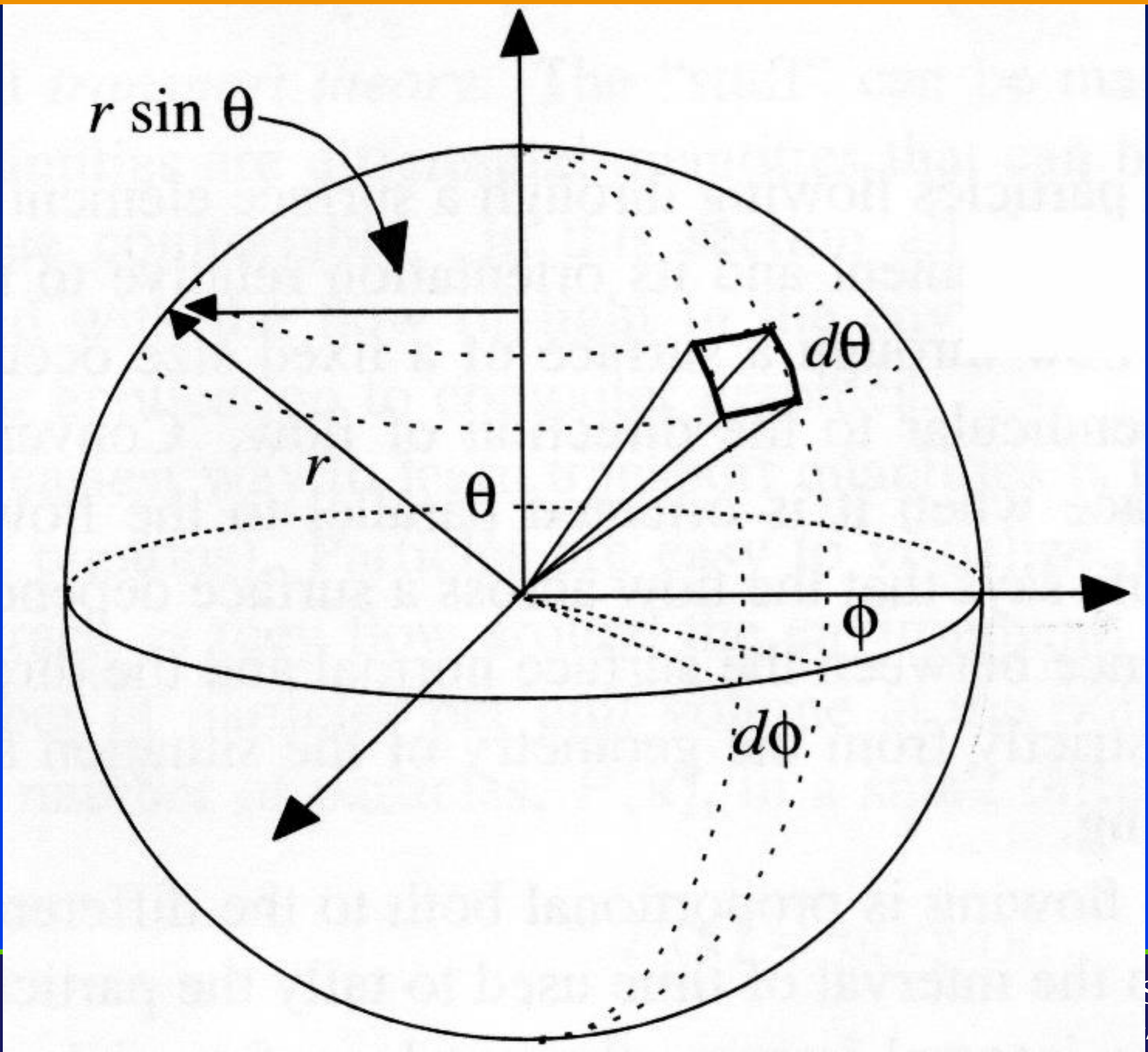
Definition: The solid angle (SA) subtended by an object from a point P is the area of projection of the object onto the unit sphere centered at P, the size of a differential patch, dA,

$$dA = (r\,d\theta)(r\sin\theta\,d\phi) = r^2 \sin\theta\,d\theta\,d\phi$$

The differential solid angle:

$$d\omega = dA/r^2 = \sin\theta\,d\theta\,d\phi$$

Solid Angle



Solid Angle

- Size of a patch, *dA*, is

$$dA = (r \sin\theta \, d\varphi)(r \, d\theta)$$

- Solid angle is

$$d\vec{\omega} = \frac{dA}{r^2}$$

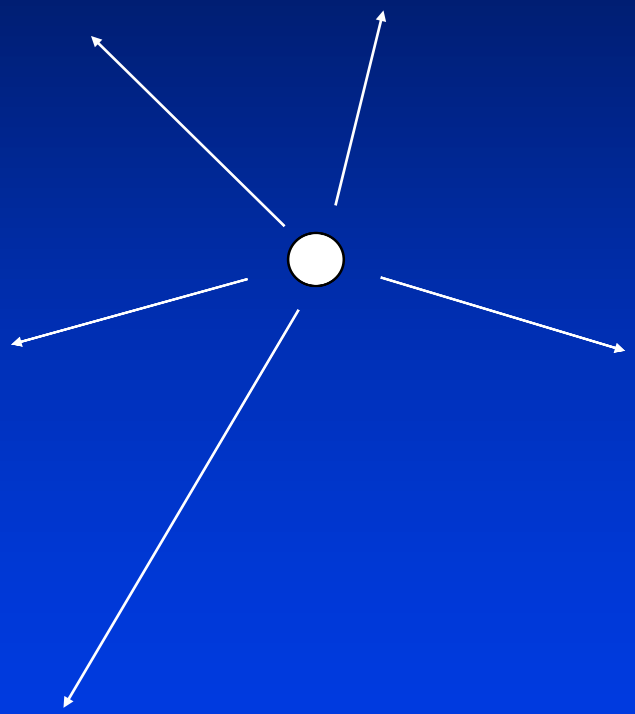
- Measured in sterradians (sr)

Radiometry for Surface Rendering

- Science of measuring light
- Analogous science called Photometry is based on human perception
- Investigate formally some methods for physically-based realistic rendering
- Present a practical method for producing highly realistic (and also physically correct) images (renderings) of 3D worlds

Radiometry for Surface Rendering

Point Source



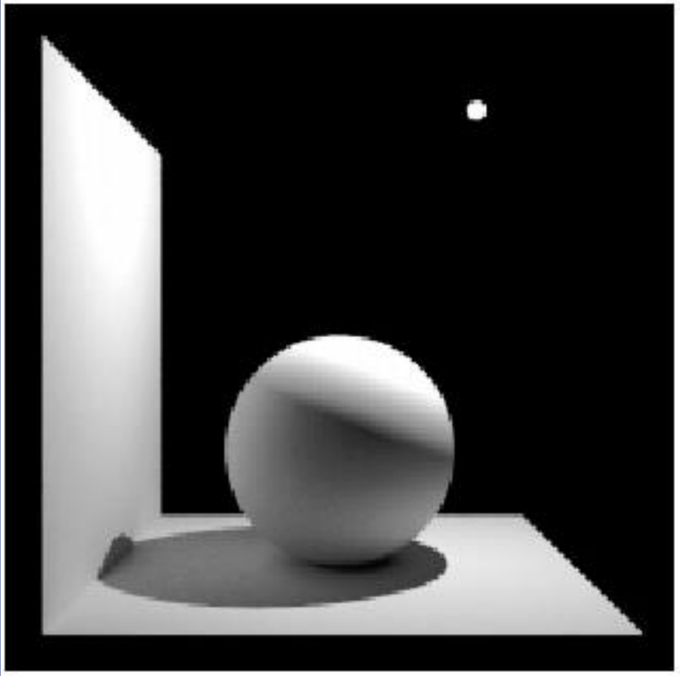
Isotropic Point Source

- Irradiates equally in all directions
- Even distribution of power over sphere

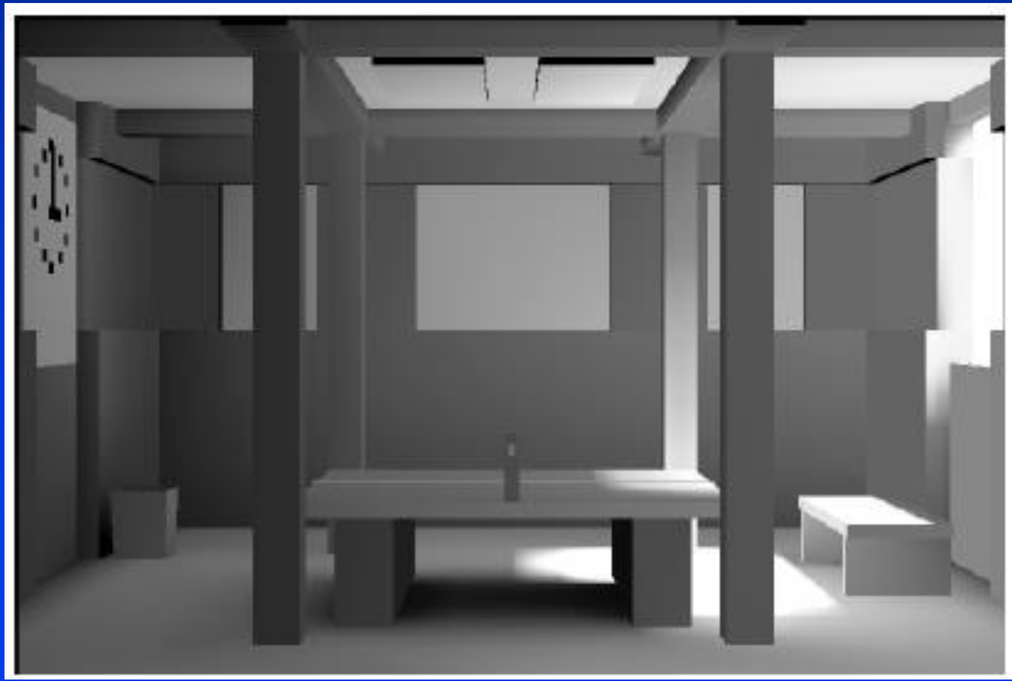
$$I = \frac{d\Phi}{d\vec{\omega}} = \frac{\Phi}{4\pi}$$

Point Light Source

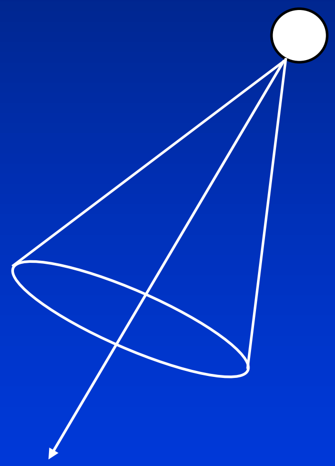
point source



area source(s)



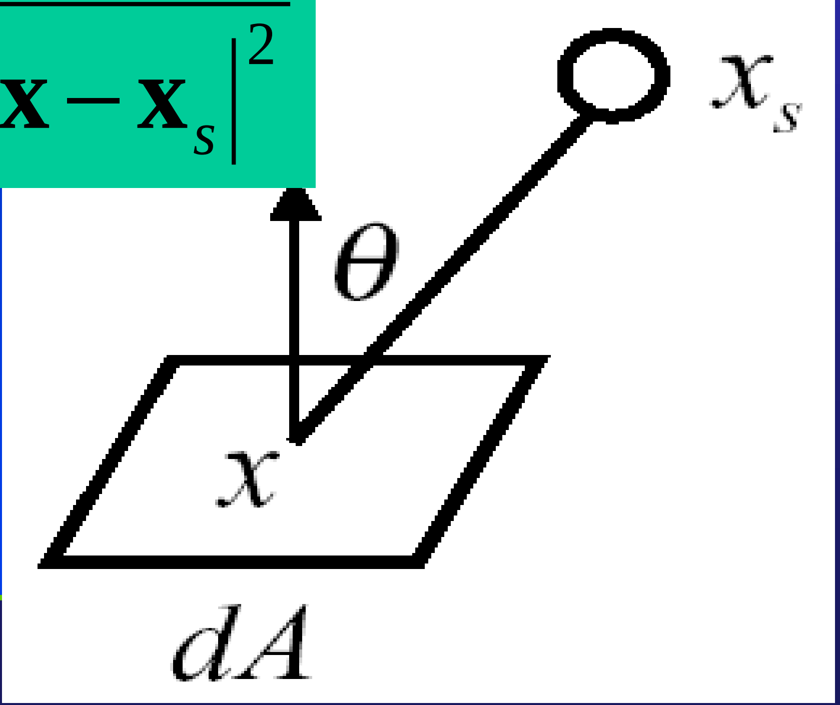
Spotlights



Irradiance due to a Point Light

Irradiance on a differential surface due to an isotropic point light source is

$$E = \frac{d\Phi}{dA} = I(\omega)\frac{d\omega}{dA} = \frac{\Phi}{4\pi}\frac{\cos\theta}{\left|\mathbf{x}-\mathbf{x}_s\right|^2}$$

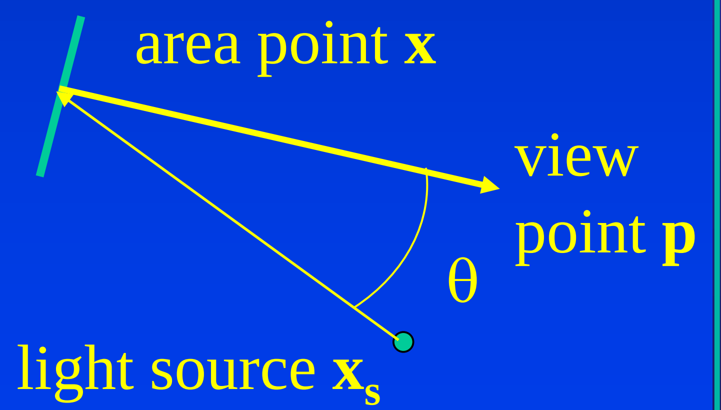


Irradiance on Differential Area

- What is the irradiance of a differential area, illuminated by a point source at $\mathbf{x}_s$, seen from a point $\mathbf{p}$?
- This is the "square law"

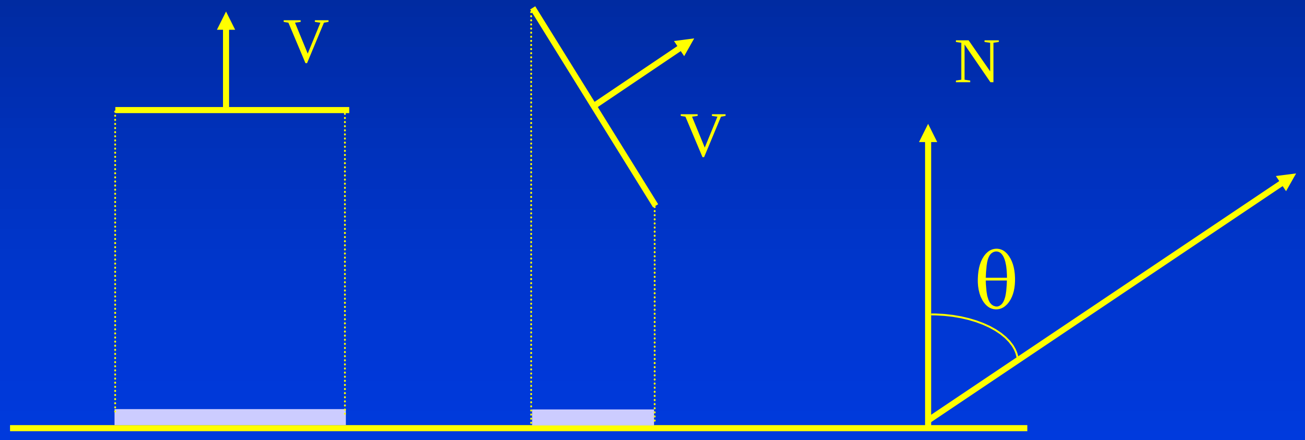
accounts for projected area

$$E = I\frac{d\omega}{dA} = \frac{\Phi}{4\pi}\frac{\boxed{\cos\theta}}{\left|\mathbf{x}-\mathbf{x}_s\right|^2}$$



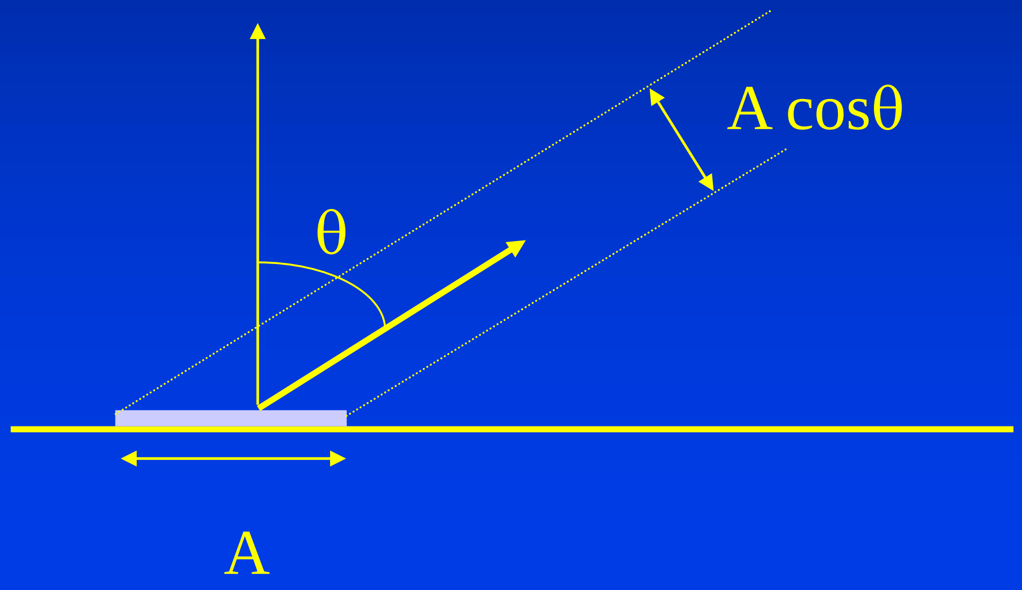
Projected Area

- $A_p = A\,(N \cdot V) = A \cos\theta$

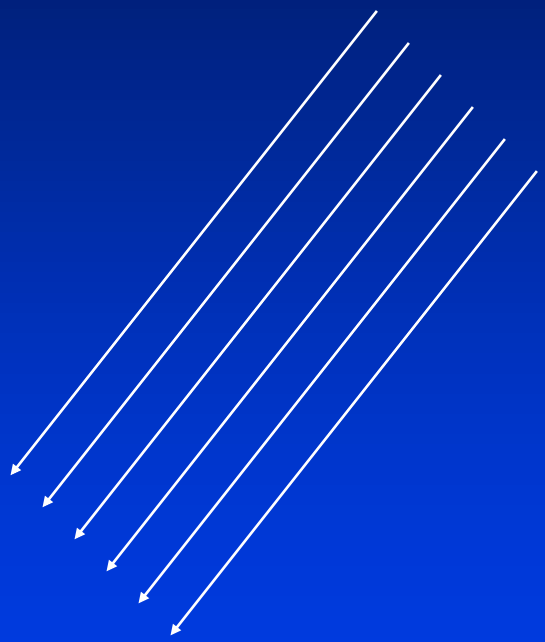


Why the Cosine Term?

- Foreshortening is by cosine of angle.
- Radiance gives energy by *effective* surface area, as seen from the view direction

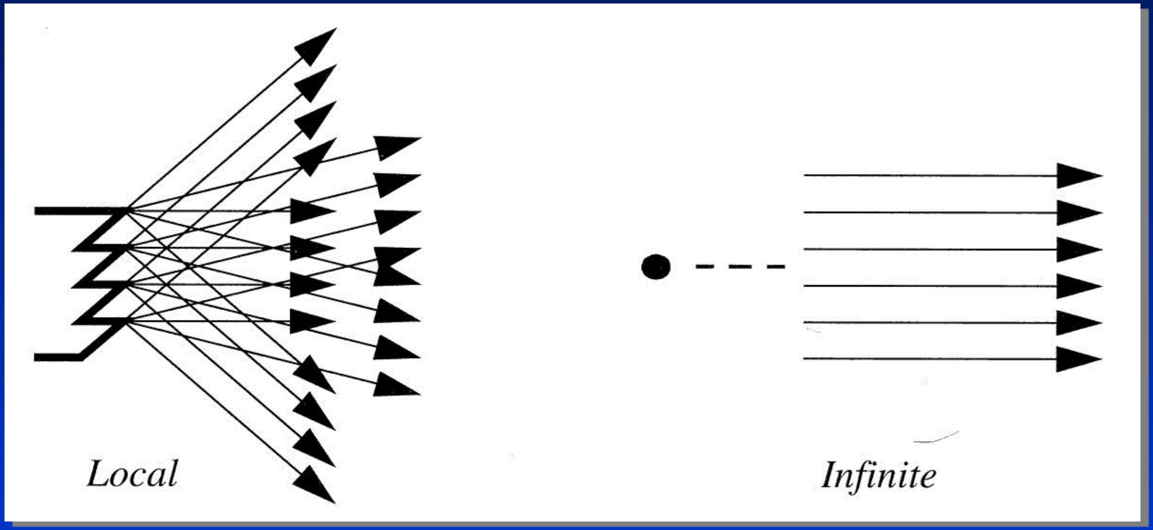


Distant Light



Local vs. Infinite Light Sources

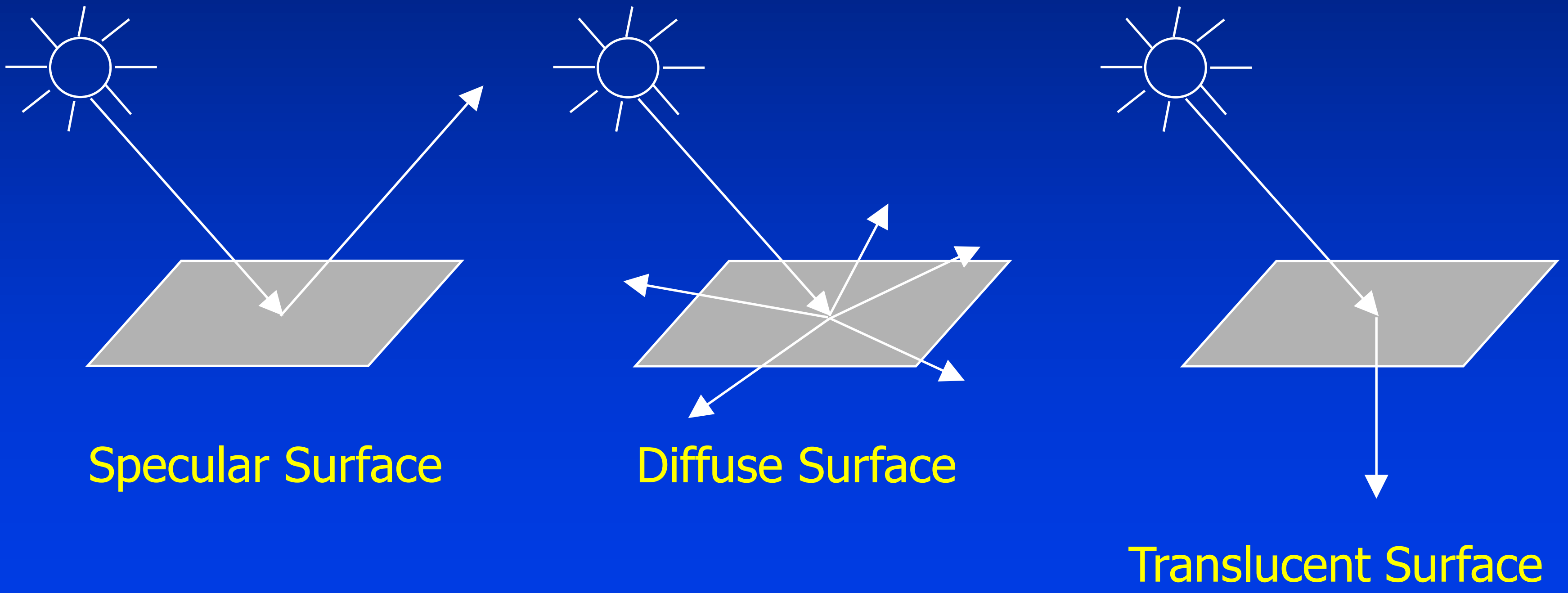
Local



Infinite

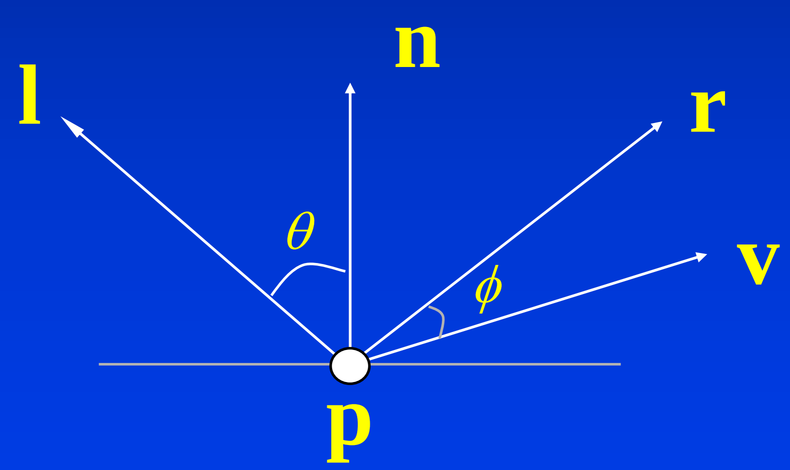
- Rays from a local light source emanate in different directions
- Rays from the infinite light source travel in the same direction

Surface Types



Phong Reflection Model

- An efficient approximation of physical reality.
- Supports three types of material-light interactions.
  - Ambient
  - Diffuse
  - Specular



Surface Properties – Ambient Lighting

- Light rays strike objects (or, *actors)* in the scene
- Illumination model determines how light and surface properties interact to generate a correct color image
- *Ambient lighting* is simplest illumination model
- It accounts for *indirect* light
- Models general level of brightness in the scene
- Accounts for light effects that are difficult to compute (secondary reflections, etc.)
- Constant for all surfaces and view directions

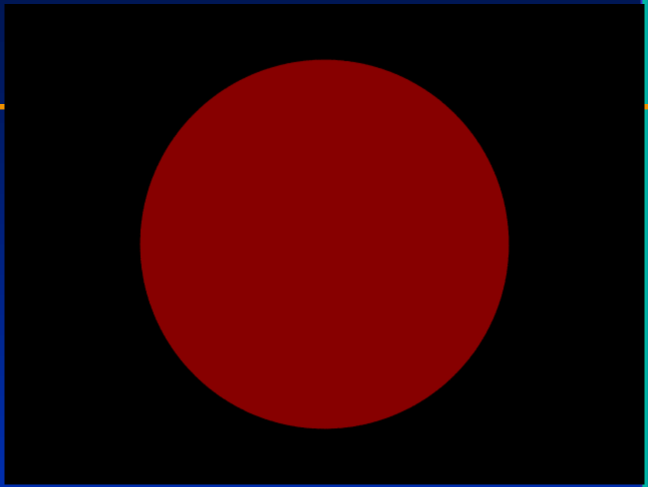
Ambient Reflection

- $I_a = k_a L_a$
- $0 \leq k_a \leq 1$

- *Background light: uniform illumination*
- *View direction independent*

Surface Properties – Ambient Lighting

- Imagine yourself in room with curtains drawn
- Some sunlight will still get in, but it will have bounced off many objects before entering room
- When an object reflects this kind of light, we call it *ambient reflection*



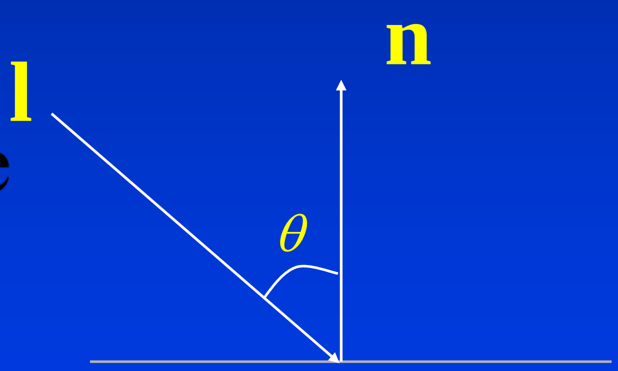
Ambient-lit sphere

Surface Properties – Diffuse Lighting

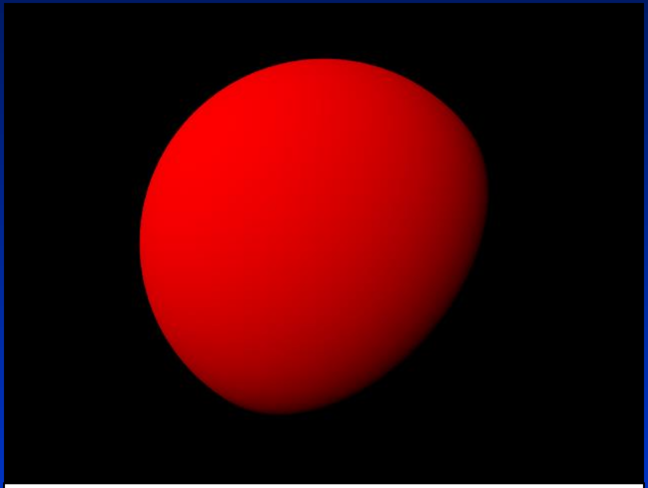
- Ambient lighting is a crude approximation of *secondary reflections*
- *Diffuse lighting* takes us one step closer to reality
- Direction of rays taken into consideration
- Unlike ambient reflection, diffuse reflection is dependent on location of light source relative to the object
- This is a type of *direct lighting*
- Models dullness, roughness of a surface
- Also called *Lambertian reflection*

Diffuse Reflection

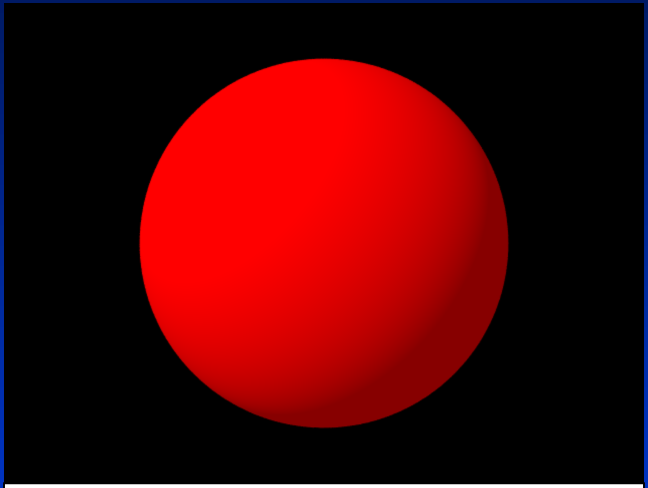
- $I_d = k_d L_d \cos\theta$
- $\cos\theta = \mathbf{l} \cdot \mathbf{n}$
- $I_s = k_d L_d (\mathbf{l} \cdot \mathbf{n})$
- Angle of incidence
- View independent



Surface Properties – Diffuse Lighting



Diffuse lighting



Ambient & diffuse

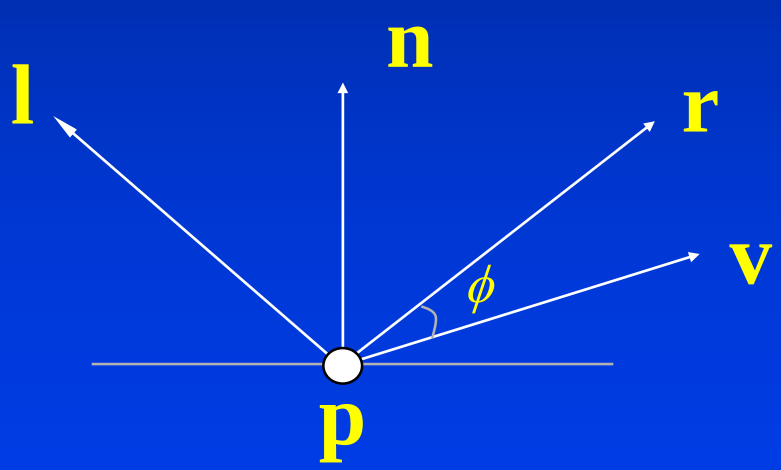
- Note difference between diffuse alone and diffuse with ambient lighting
- Suppose we moved light to around back of sphere – remind us: why would the sphere get darker?

Surface Properties – Specular Lighting

- Models reflections on shiny surfaces (polished metal, chrome, plastics, etc.)
- Specular reflection is *view-dependent* – specular *highlight* changes as camera's position changes
- Diffuse reflection is *view-independent* – reflection model is a function of light source direction and surface direction (normal)
- Specular reflection is a function of the light source direction, the surface direction, ***and the view direction***

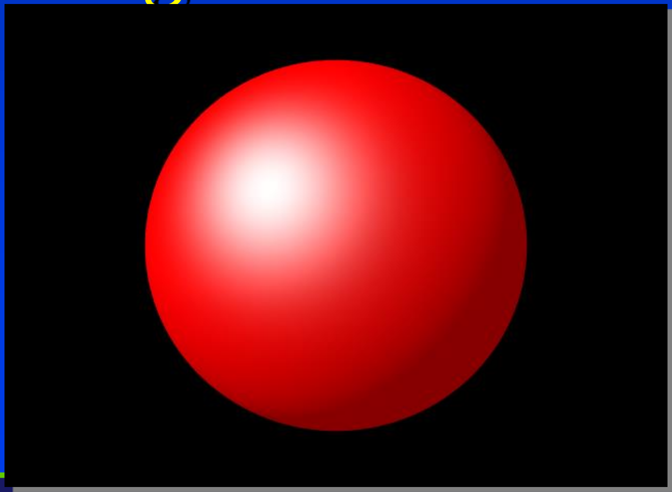
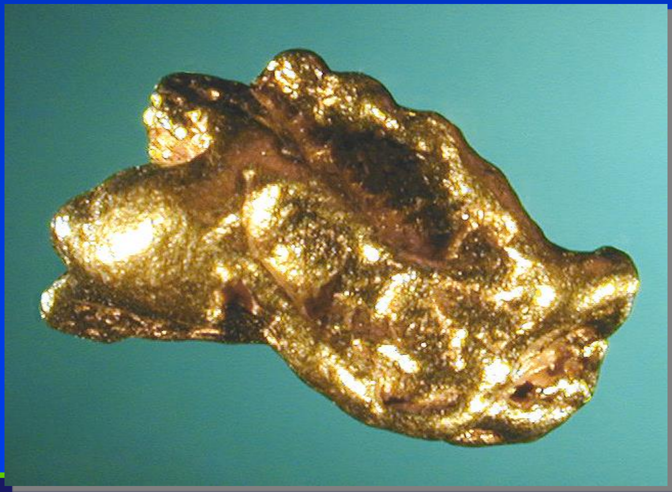
Specular Reflection

- $I_s = k_s L_s \cos^{\alpha}\phi$
- $\cos\phi = \mathbf{r} \cdot \mathbf{v}$
- $I_s = k_s L_s (\mathbf{r} \cdot \mathbf{v})^{\alpha}$

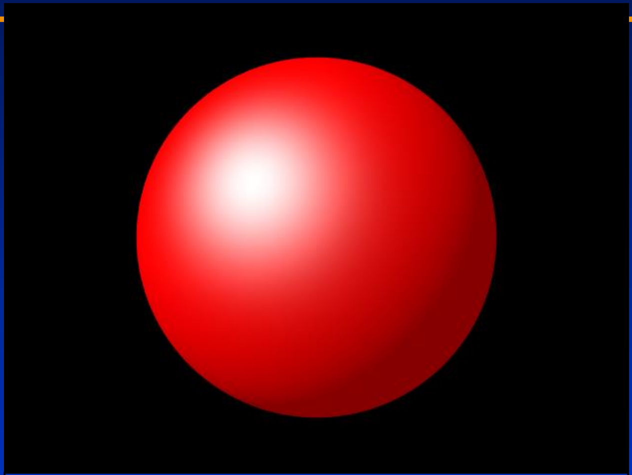


Surface Properties – Specular Lighting

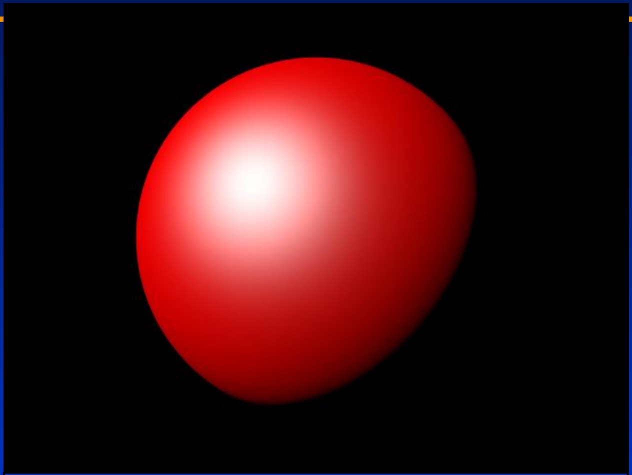
- Need angle light source makes with surface, and angle viewing ray makes with surface
- Example: chrome on your car shines in different ways depending on where you stand when looking at it

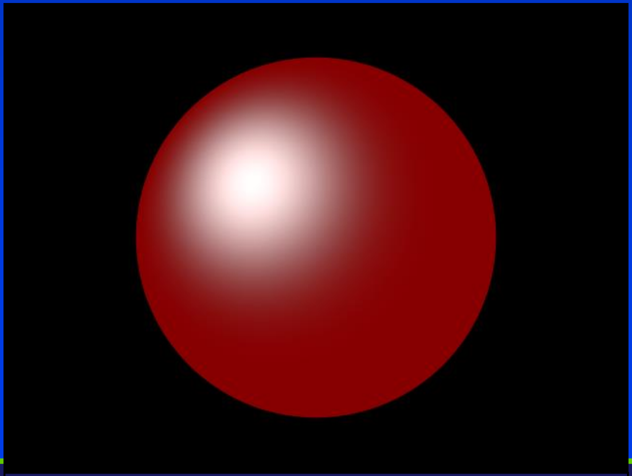
Surface Properties – Specular Lighting



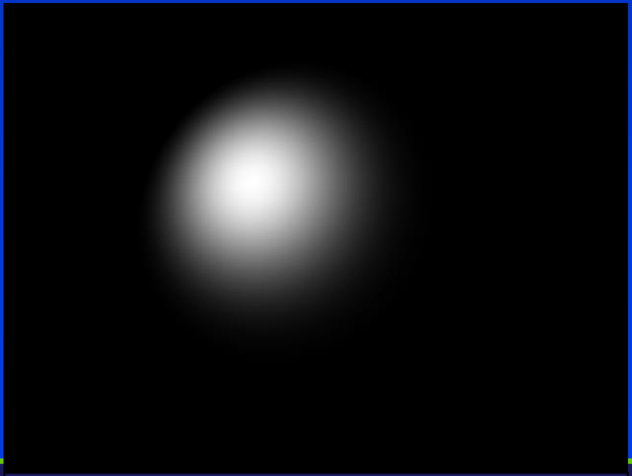
Specular & diffuse & ambient



Specular & diffuse

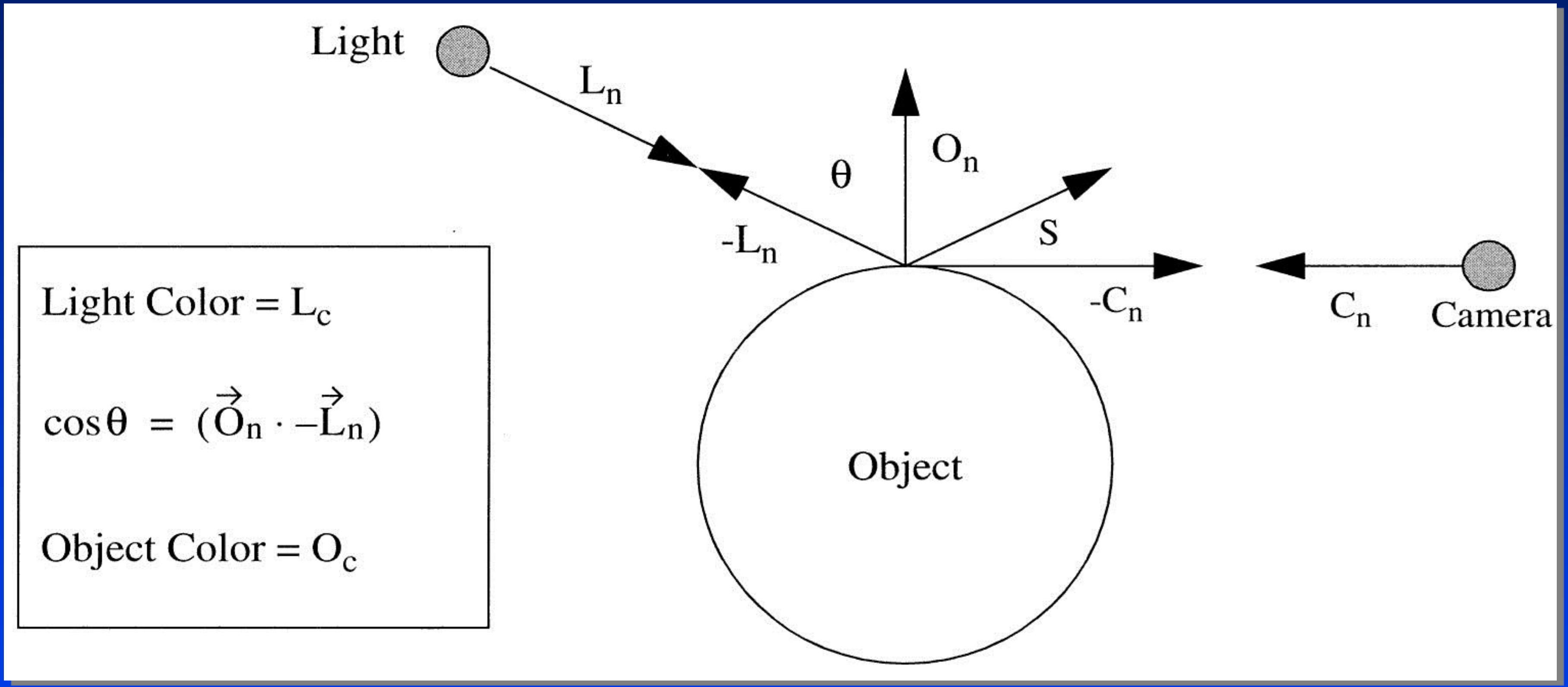


Specular & ambient



Specular only

Surface Properties – Specular Lighting



Surface Properties – Specular Lighting

$$R_c = L_c O_c [\vec{S} \cdot (-\vec{C}_n)]^{O_{sp}}$$

$$\vec{S} = 2[\vec{O}_n \cdot (-\vec{L}_n)]\vec{O}_n + \vec{L}_n$$

- S is the direction of specular reflection
- The angle S makes with O_n is the same angle $-L_n$ makes with O_n: θ
- C_n is the viewing direction
- O_{sp} is the *specular power* and indicates how shiny the object is

Surface Properties – Specular Lighting

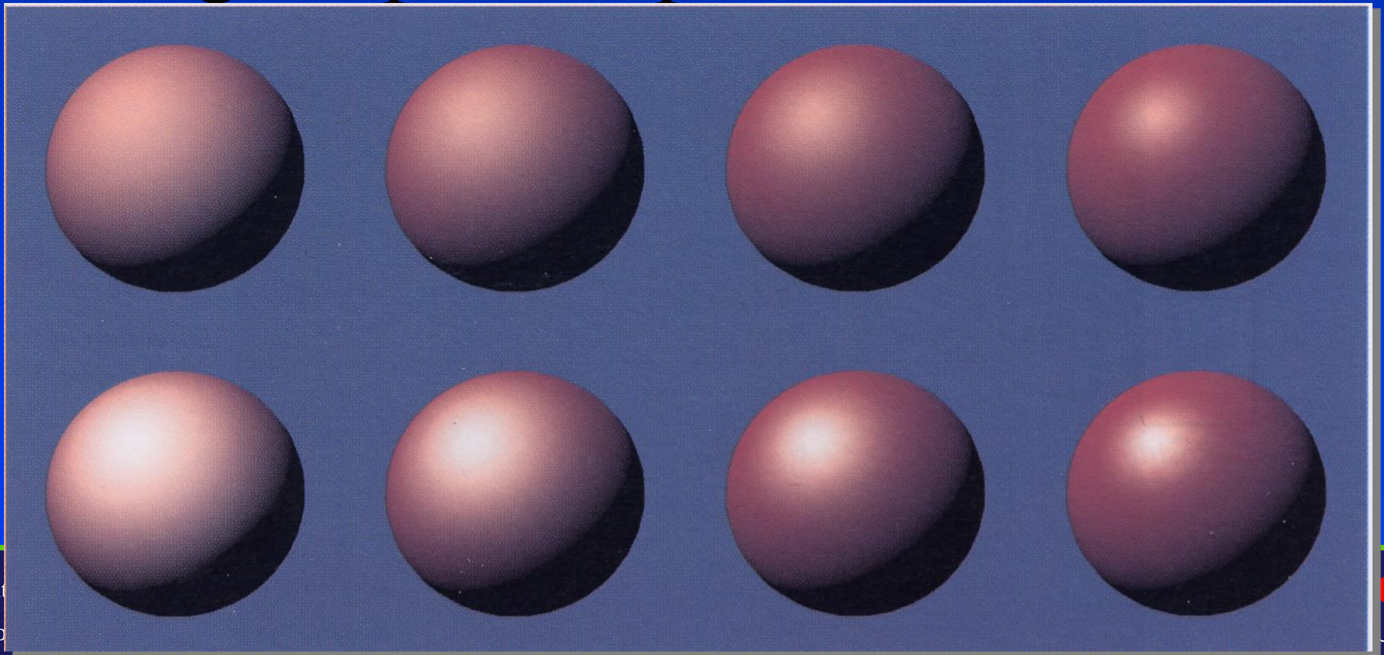
- Specular power indicates how quickly the specular reflection *diminishes* as direction of specular reflection deviates from view direction
- Specular power controls the size of specular highlight
- Inverse relationship:

$$R_c = L_c O_c [\vec{S} \cdot (-\vec{C}_n)]^{O_{sp}}$$

$$\vec{S} = 2[\vec{O}_n \cdot (-\vec{L}_n)]\vec{O}_n + \vec{L}_n$$

Surface Properties – Specular Lighting

- Top row: specular *intensity* = 0.5 (O_c, essentially)
- Bottom row: specular intensity = 1.0
- Left to right: specular power = 5, 10, 20, 40



Phong Model

- $I = I_a + I_d + I_s$

  $= k_s L_s (\mathbf{r} \cdot \mathbf{v})^{\alpha} + k_d L_d (\mathbf{l} \cdot \mathbf{n}) + k_a L_a$

- With light attenuation by distance

$$I = 1/(a+bd+cd^2)\,(k_d L_d (\mathbf{l} \cdot \mathbf{n}) + k_s L_s (\mathbf{r} \cdot \mathbf{v})^{\alpha}) + k_a L_a$$

Surface Properties – Total Illumination

- Ambient, diffuse and specular reflection are usually combined into a single equation:

$$R_c = O_{ai} O_{ac} L_c - O_{di} O_{dc} L_c (\vec{O}_n \cdot \vec{L}_n) + O_{si} O_{sc} L_c [\vec{S} \cdot (-\vec{C}_n)]^{O_{sp}}$$

- O_{ai}, O_{di} and O_{si} control the amounts of ambient, diffuse and specular lighting, with values in [0.0, 1.0] (these three values are called *reflection coefficients*)
- O_{ac}, O_{dc} and O_{sc} indicate the colors to be used with each type of lighting (specular color, O_{sc}, is usually white)

Surface Properties – Total Illumination

$$R_c = O_{ai}O_{ac}L_c - O_{di}O_{dc}L_c(\vec{O}_n \cdot \vec{L}_n) + O_{si}O_{sc}L_c[\vec{S} \cdot (-\vec{C}_n)]^{O_{sp}}$$

- What if $O_{sp} = 0$?
- What if O_{sp} = infinity?
- What if some vectors are not normalized?
- How would we disable ambient reflection?
- What if some dot product is negative? What does this indicate? How should it be handled by the illumination equation?

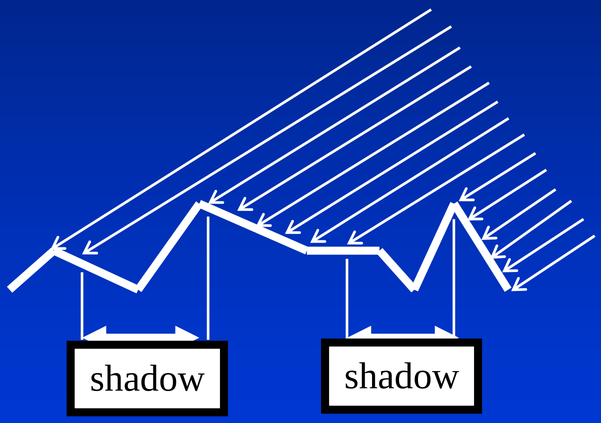
What’s Missing?

- What we’ve seen so far is just the basics of geometric optics.
  - Enough for classical ray tracing, Phong illumination model.
  - To get much better, we need more:
    - Better modeling of surface properties.
    - Wavelength dependence.
    - Radiometry / Photometry.
    - Energy transport.

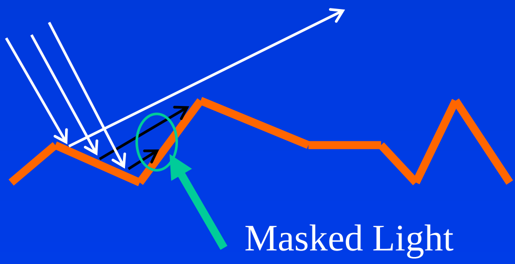
Surface Roughness

- At a microscopic scale, all real surfaces are rough:

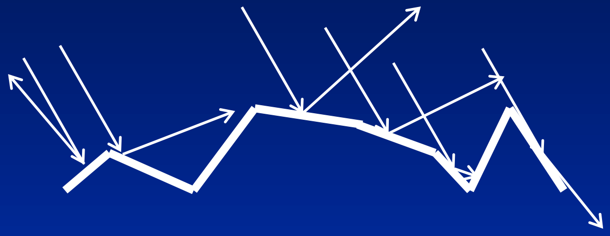
- Cast shadows on themselves:



- "Mask" reflected light:



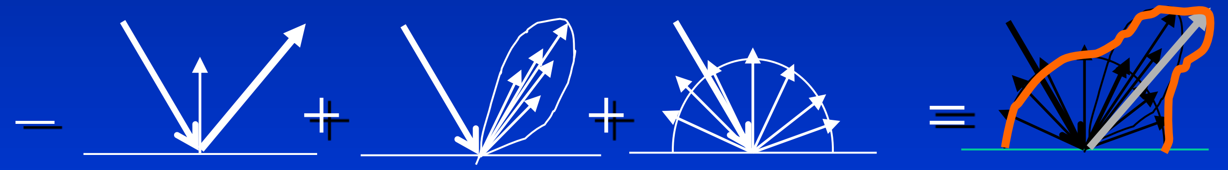
Surface Roughness



- Notice another effect of roughness:
  - Each "microfacet" is treated as a perfect mirror.
  - Incident light reflected in different directions by different facets.
  - End result is mixed reflectance.
    - Smoother surfaces are more specular or glossy.
    - Random distribution of facet normals results in diffuse reflectance.

Reflectance Distribution Model

- Most surfaces exhibit complex reflectances.
  – Vary with incident and reflected directions.
  – Model with combination:

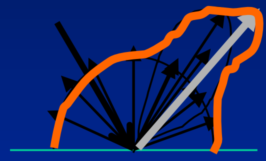


specular + glossy + diffuse = reflectance distribution

Anisotropy

- So far we've been considering *isotropic* materials.
  - Reflection and refraction invariant with respect to rotation of the surface about the surface normal vector.
  - For many materials, reflectance and transmission are dependent on this azimuth angle: *anisotropic* reflectance/transmission.
  - Examples?

BRDF



- Bidirectional Reflectance Distribution Function
- $\rho\,(x, \omega_i, \omega_o)$
  - – x is the position.
  - – $\omega_i = (\theta_i, \phi_i)$ represents the incoming direction. (elevation, azimuth)
  - – $\omega_o = (\theta_o, \phi_o)$ represents the outgoing direction (elevation, azimuth)

Light Transport

- To compute images, we need to simulate transport of light around a scene.
- *Transport theory*.
  - Analysis techniques for flow of moving particles in 3D.
    - Largely developed for neutrons in atomic reactors.
    - Can be applied to traffic flow, gas dynamics.
    - Most importantly, can be applied to light.
- Simulation techniques.
  - Ray tracing.
  - Radiosity.
  - Combinations and variations.

Local vs. Global Illumination

- Radiosity and ray tracing simulate *global* illumination.
  - Account for light transport between objects.
  - Not just between light sources and objects: *local* illumination.
- Don't need global illumination to use the concepts of geometric optics, surface modeling, and BRDF.
- Have been used to create diverse shading models.
  - Simplest and most common is Phong.
  - Next lecture: shading models.

Radiometry vs. Photometry

- *Photometry*: Science of human perception of light.
  - Perceptual analog of Radiometry.
  - All measurements relative to perception.

Color

- Color is determined by the wavelength of visible light.
- Still use geometric optics.
- But need to account for wavelength in reflectance (BRDF) and index of refraction.
- What natural phenomena can you think of that are wavelength dependent?

Reading

- Henri Gouraud, “Continuous Shading of Curved Surfaces”. IEEE Transactions on Computers; June 1971.
- Bui Tuong Phong, “Illumination for Computer Generated Pictures”. Communications of the ACM; June 1975.
- James F. Blinn, “Models of Light Reflection for Computer Synthesized Pictures.” Computer Graphics (SIGGRAPH 1977).

References

- Glassner, *Principles of Digital Image Synthesis*, Volume Two.
  - Highly detailed and low level.
- Cohen and Wallace, *Radiosity and Realistic Image Synthesis*.

Polygonal Shading

- Flat shading
- Gouraud shading
- Phong shading

Flat Shading

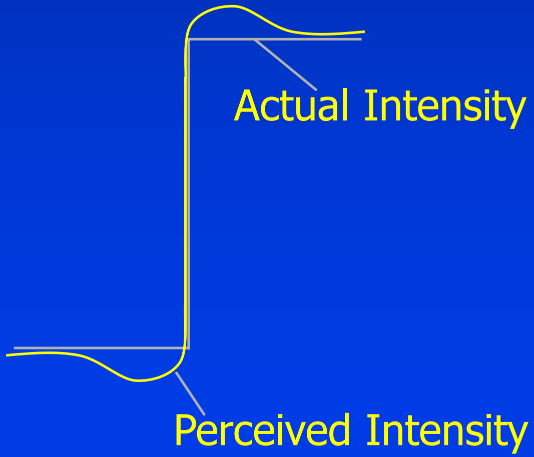
- glShadeModel(GL_FLAT)

- Constant intensity shading, i.e. the intensity is constant for each polygon.

- Very simple to implement, however, it may introduces intensity discontinuities by Mach band effect.

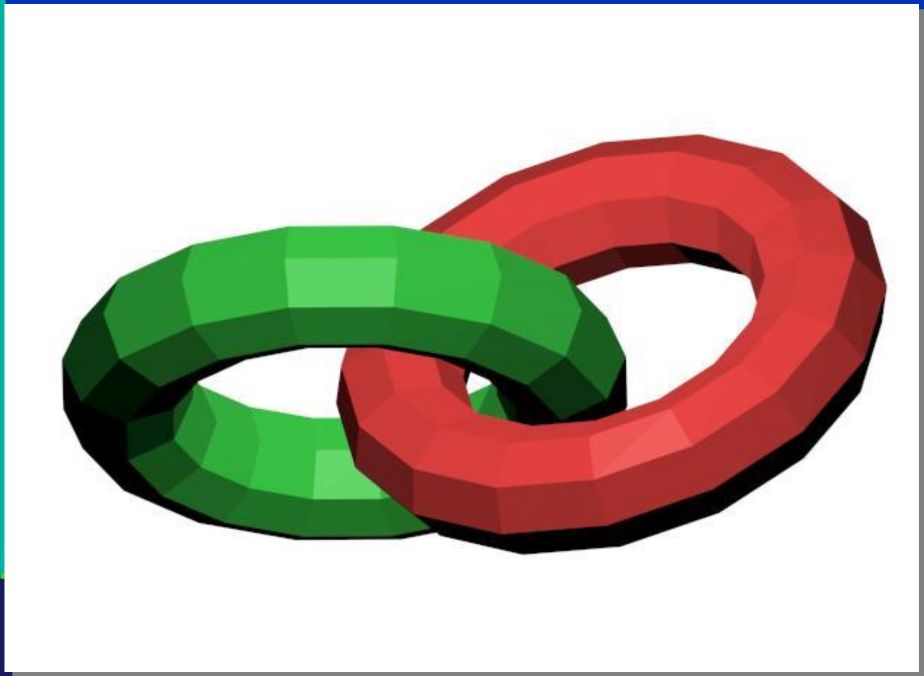
Mach Band Effect

- The human visual system is very sensitive to small differences in light intensity.
- Because of a property known as lateral inhibition.



Flat Surface Rendering

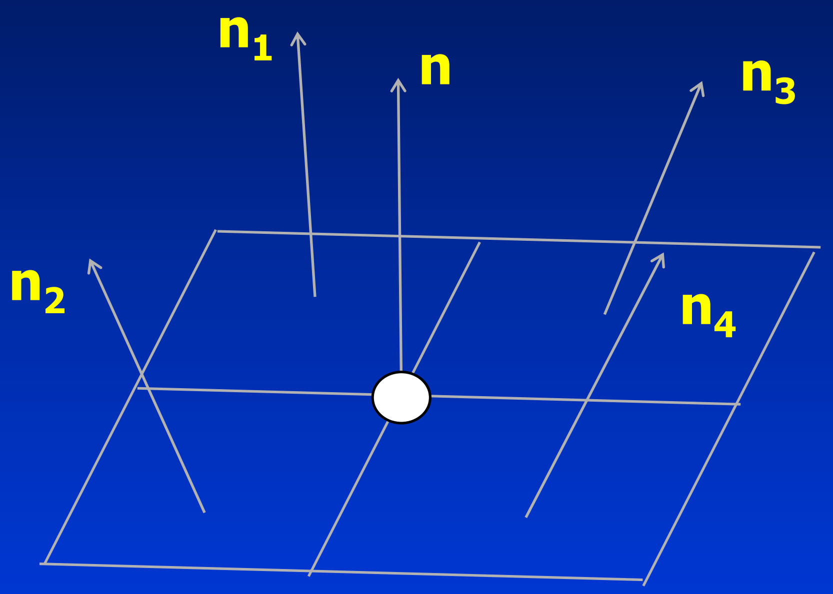
- Illumination equations applied to one normal vector of the polygon
- Result: all pixels for polygon have the same color



Gourand Shading

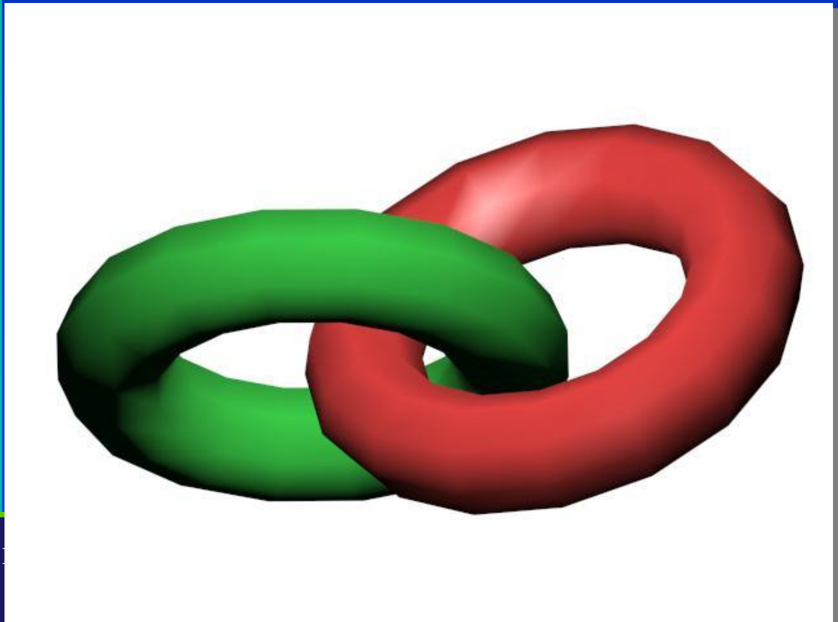
- glShadeModel(GL_SMOOTH)
- Interpolative intensity shading.
- Calculate intensity at each vertex of the polygon and interpolate the other intensity values.

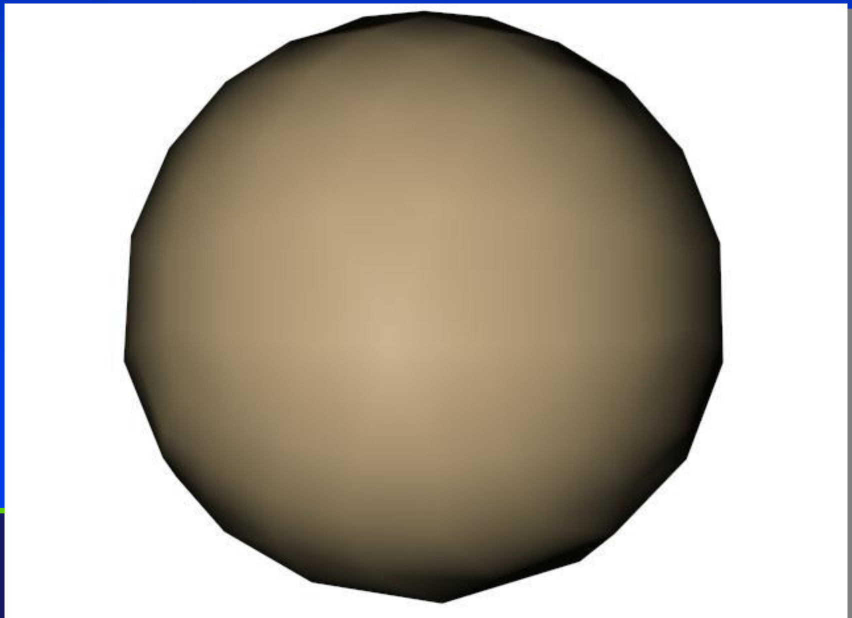
Normal vector calculation



Gouraud Surface Rendering

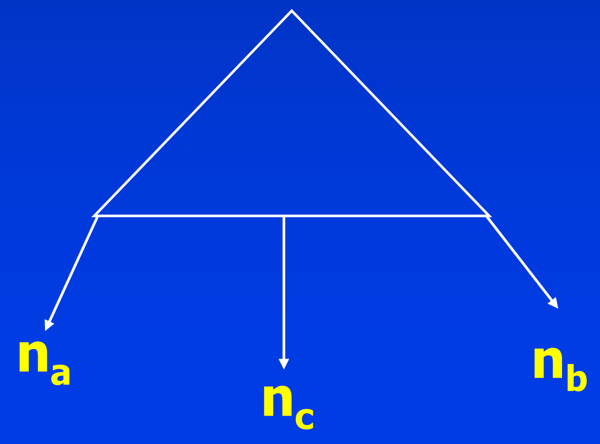
- Illumination equations are calculated at all vertices of polygon using vertex normals
- Edges and interior of polygon colored by *interpolating* or smoothly blending the ***colors*** computed at vertices
- Result: color varies across the polygon





Phong Shading

- Evaluate the intensity at each pixel.
- The normals are interpolated.
- Often done off-line.
- $\mathbf{n}_c = (1-\alpha)\mathbf{n}_a + \alpha\,\mathbf{n}_b$
- $\mathbf{n}_c = \mathbf{n}_c / |\mathbf{n}_c|$

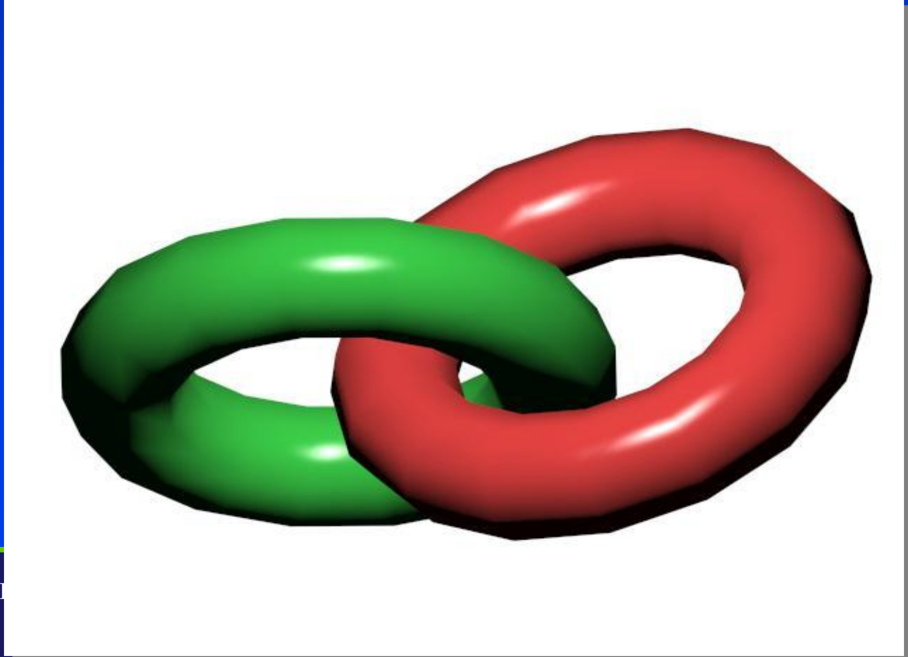


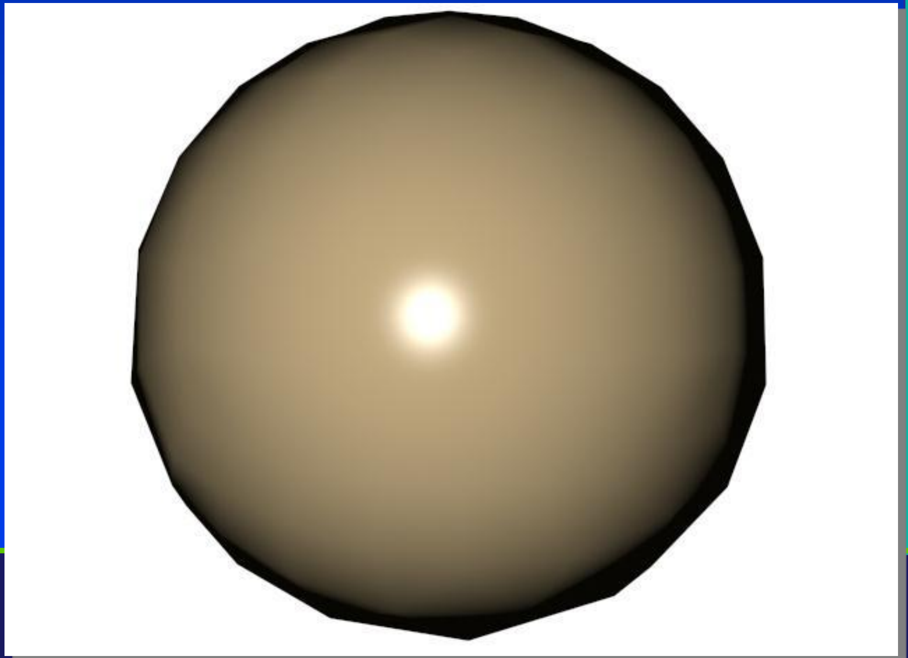
Phong Surface Rendering

- **Normals** are first interpolated across edges
- Then interpolated across the polygon interiors
- Illumination equations are computed *for each pixel*
- Result: color varies across the polygon, plus we can generate specular highlights
- What do you think of the efficiency of Phong shading?

Phong Surface Rendering

- Phong rendering just too expensive to use in real-time
- Software ray tracers use it, where speed is already slow

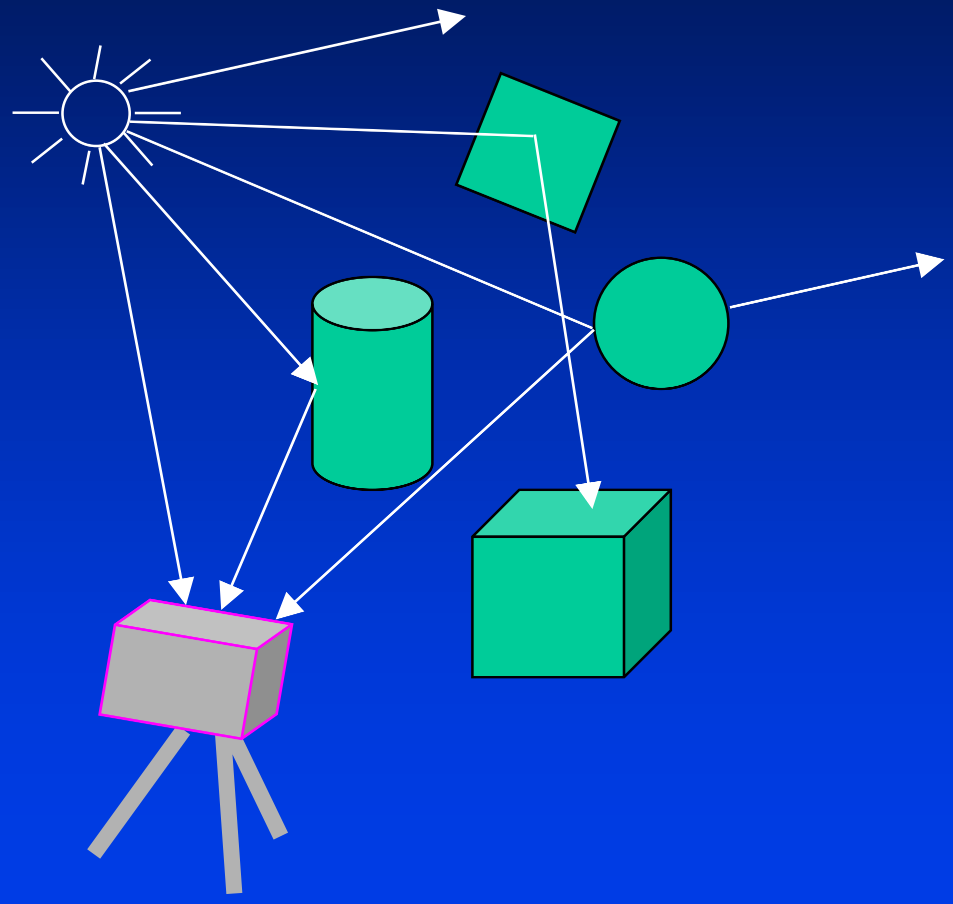




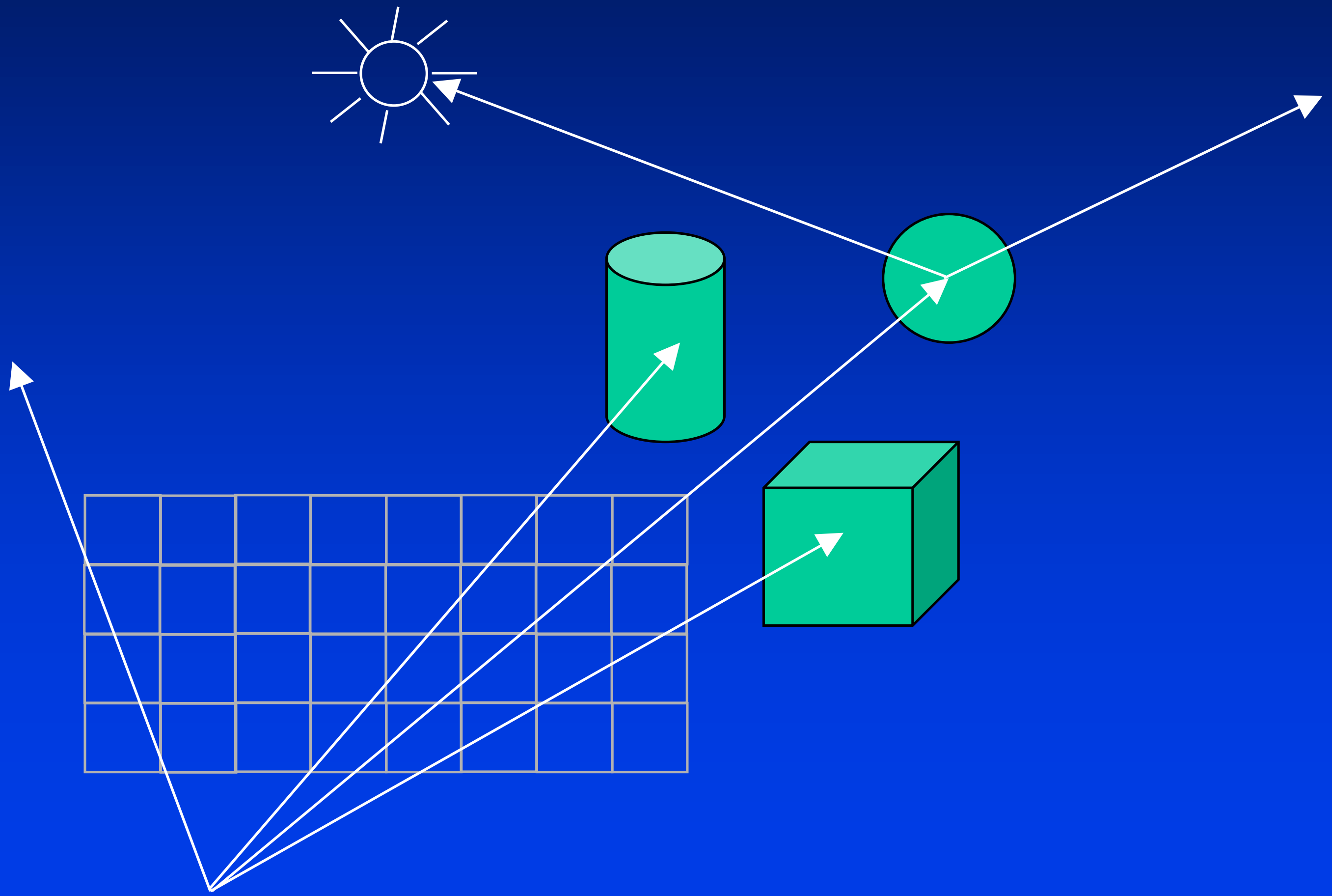
Global Illumination

- Shading is conducted by considering the interaction between all objects in the environment.
- More accurate rendering with more cost.
- Often done in off-line.
- Two main approaches:
  - Ray Tracing
  - Radiosity

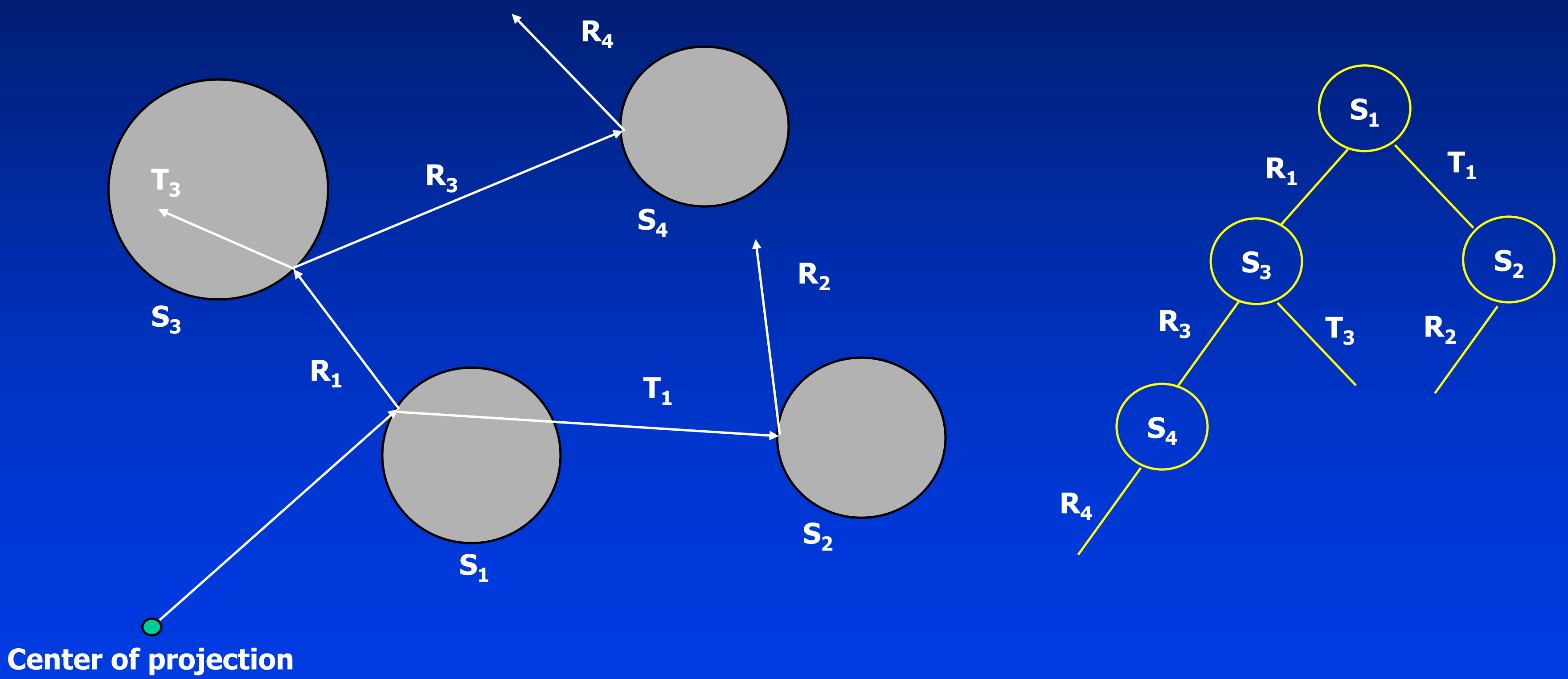
Ray Tracing



Ray Tracing



Ray Tracing Tree

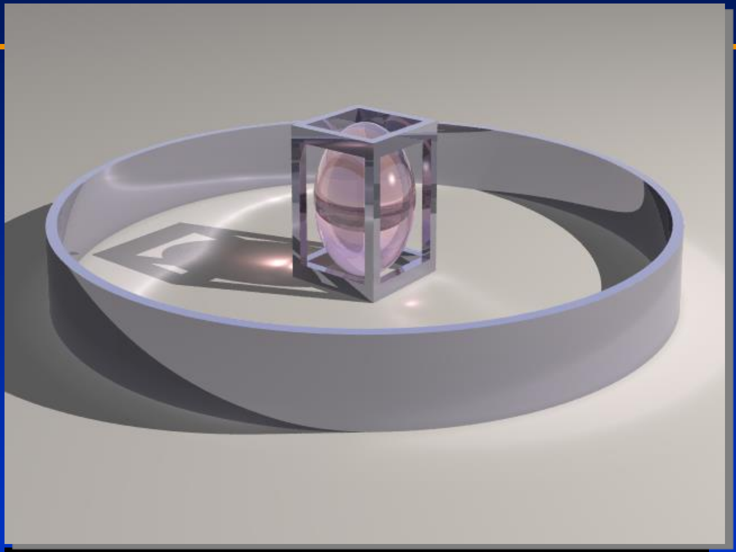


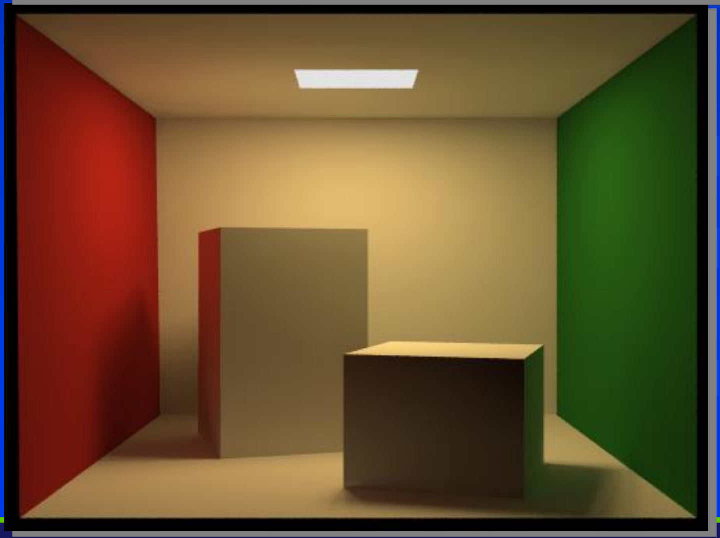
Ray Tracing

- The intensity assigned to a pixel is then determined by accumulating the intensity contributions, starting at the bottom (terminal nodes) of its ray-tracing tree.
- Surface intensity from each node in the tree is attenuated by the distance from the "parent" surface (next node up the tree) and added to the intensity of the parent surface. Pixel intensity is then the sum of the attenuated intensities at the root node of the ray tree.

Other Shading and Illumination Effects

- Area lights
- Shadows
- Refraction
- Reflection
- Caustics
- Color bleeding
- Radiosity
- How do we generate these effects?



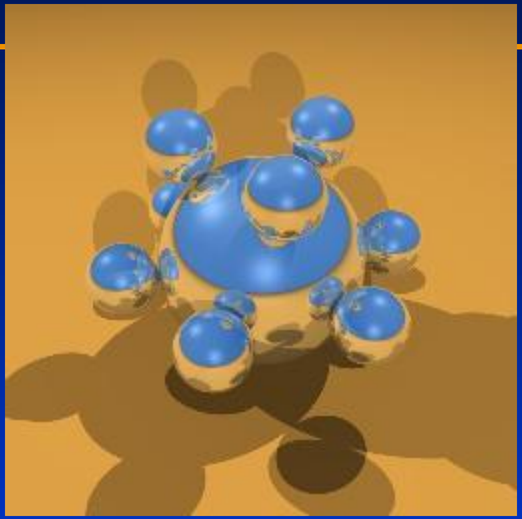


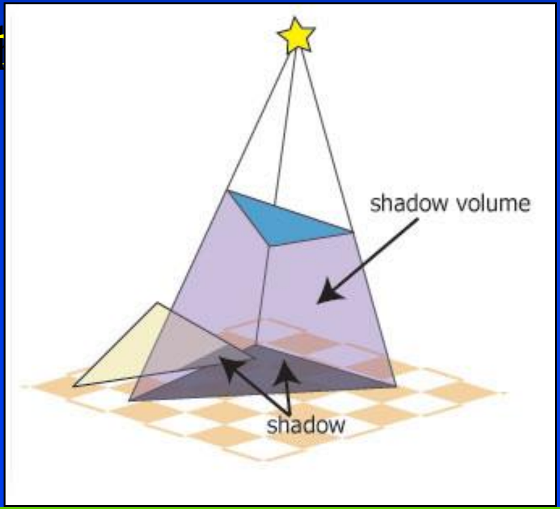
Global Illumination

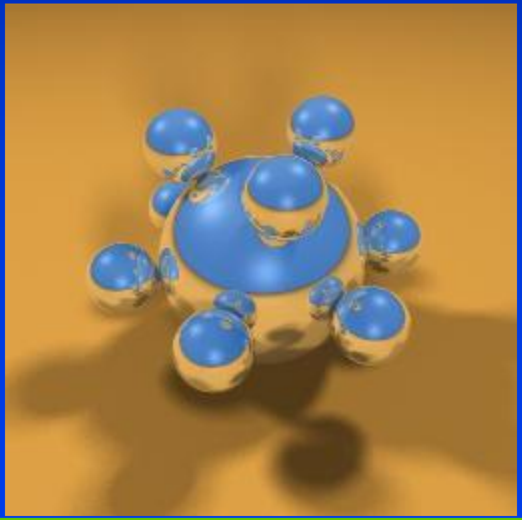
- These effects require *global illumination*, which is capable of generating all those *photorealistic* images you see in movies and special effects
- Most require the use of ray-tracing and radiosity, an $O(n^2)$ illumination technique
- Want to try it yourself? Go to www.povray.org and try out the free POV-Ray ray-tracing program

Shadows

- Hard and soft shadows
- Hard shadows: caused by very distant
  light sources, like the sun
- Soft shadows: caused by close light
  sources, usually area light sources, like
  light bulbs
- Several techniques for generating shadows





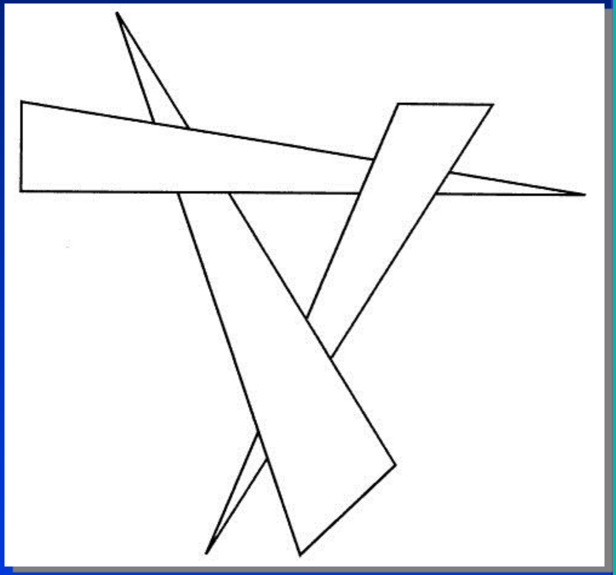


Hidden Surface Removal

- We looked earlier at ray-casting
- We trace rays from the camera, through the images and into the scene
- We see whatever objects the rays strike
- Usually we don't use ray-casting and instead use the object-order approach we've been talking
- A complex scene could contain thousands or even millions of triangles that will overlap
- How do we know in which order to draw the triangles?

Hidden Surface Removal: Painter’s Algorithm

- One solution is called the *painter’s algorithm*
- Sort the triangles
- Back-to-front or front-to-back?
- One major problem:



- Can cut into smaller triangles, but the way we cut the triangles is *view-dependent*
  What does that mean?

Hidden Surface Removal: Z-Buffer Algorithm

- An easier and very efficient solution is the z-buffer algorithm
- We store a 2D array the same dimensions as the image
- Before we draw a pixel for a triangle, we compare its z value to what is stored in the z-buffer
- If the new pixel would be in front of the z-buffer's algorithm, we replace the current pixel with the new one
- Otherwise, we do not change the pixel
- How should we initialize the z-buffer?