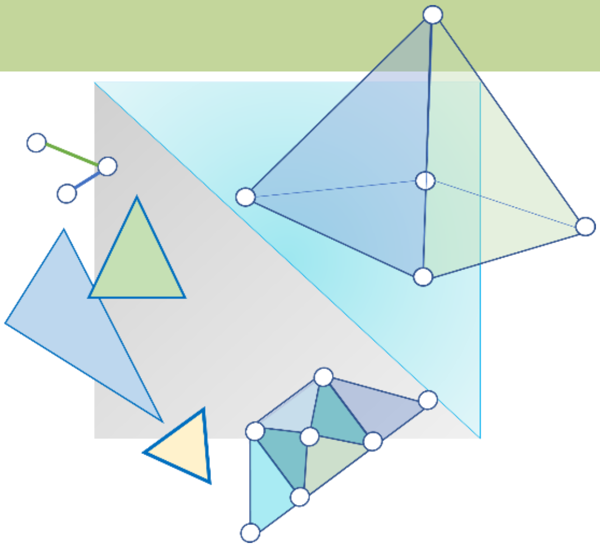
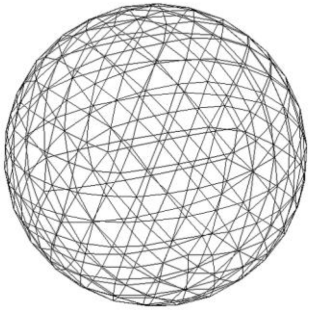
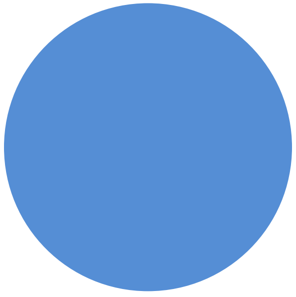
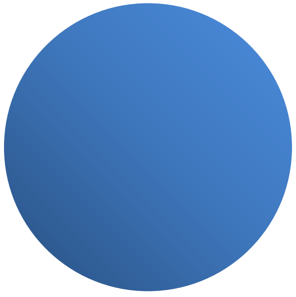
CITS3003 Graphics & Animation



Lecture 14&15: Shading







Objectives

- Learn that with appropriate shading objects appear as three-dimensional in the rendered image.
- Introduce the types of light-material interactions – global vs local.
- Understand some light sources and how to represent them.
- Understand different surface types.
- Introduce the Phong shading model that can be used with real time graphics hardware.

3D Perception

How do we perceive objects as 3D?

- Stereo vision is one reason, but we can still perceive 3D even we look with one eye.

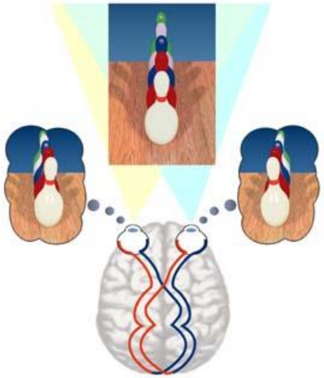
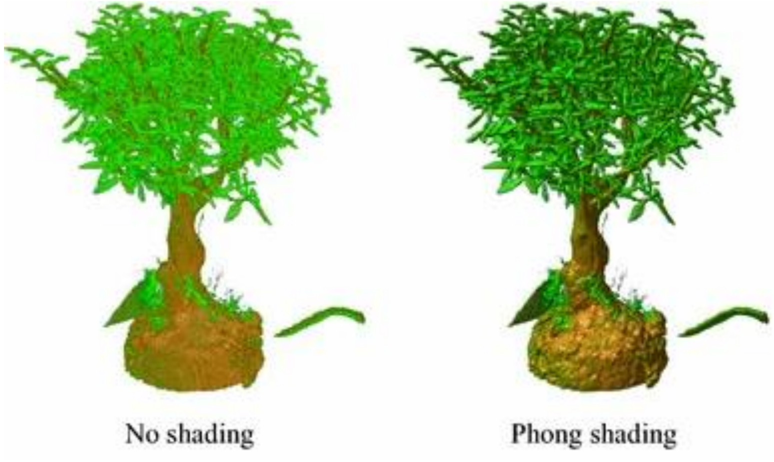


Image from here

- Hidden surface removal gives us depth perception.
- Shading is another way how we perceive 3D objects.



https://doi.org/10.1007/s00371-015-1151-6

Why we need shading?

- Suppose we build a model of a sphere using many polygons and color it with **glColor**. We get something like

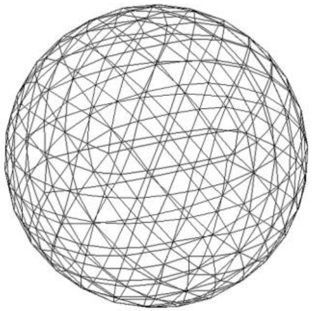
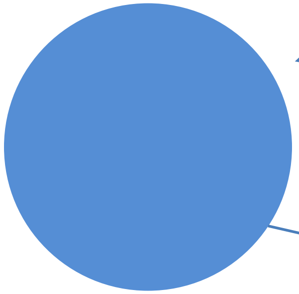
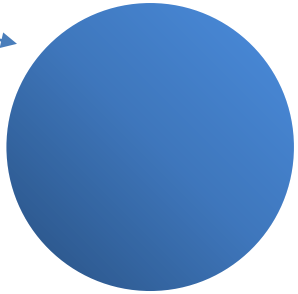


Image source



Sphere without shading

- But the image of a real sphere look like this



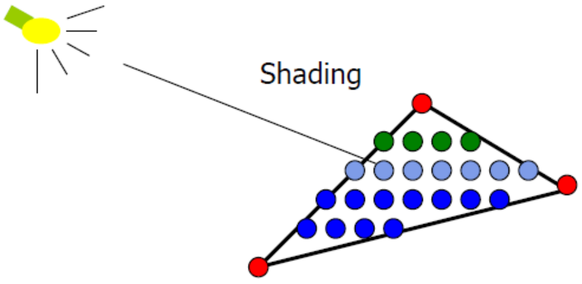
Sphere with shading

Shading:

- provides visual cues for humans.
- Provides information about:
  - Light sources
  - Material properties
  - Location of viewer
  - Surface orientation

What is Shading?

Shading refers to the process of assigning colors to the pixels.

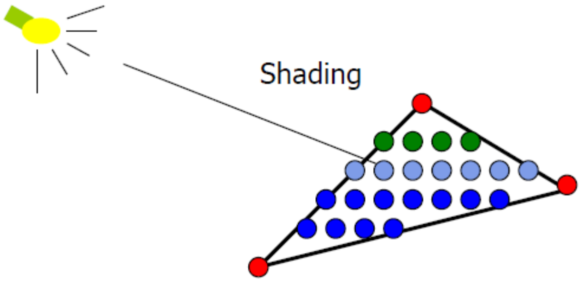


- ❑ Light-material interactions cause each point on the surface to have a different color or shade.
- ❑ Shading usually determined by lighting

Image credits :Emmanuel Agu, WPI, CS 543

What is Shading?

Shading refers to the process of assigning colors to the pixels.



- We need to understand interaction between light and objects, which results in a shaded image of the object.
  - Types of lights
  - Types of surfaces
- We will then study how shading is achieved in a graphics pipeline in a way that is
  - Efficient
  - Suits the pipeline architecture

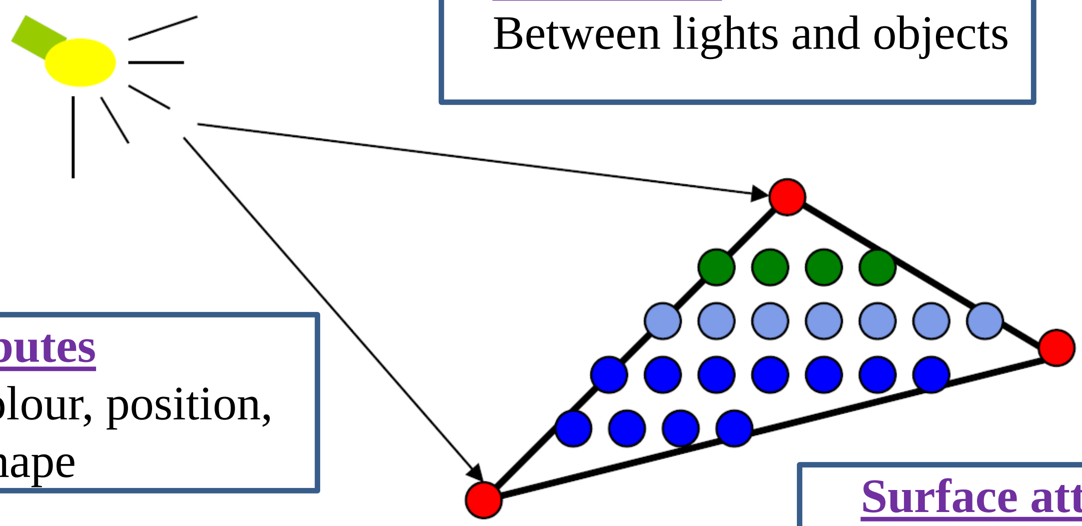
Image credits :Emmanuel Agu, WPI, CS 543

Illumination (or Lighting) Model

Illumination model is used to calculate the colour of an illuminated position on the surface of an object, also known as lighting/shading model.

- Equation for computing illumination
- An illumination model usually considers:

Interaction
Between lights and objects



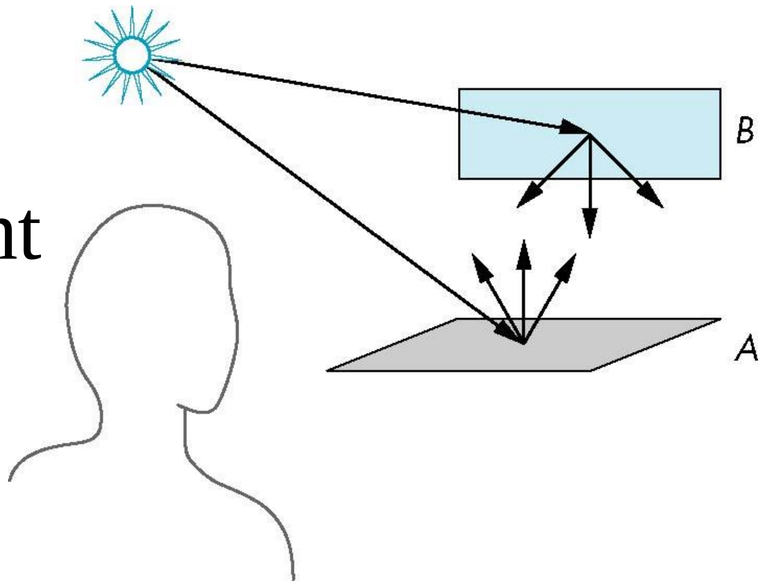
Light attributes
Intensity, colour, position, direction, shape

Surface attributes
colour, reflectivity, transparency, etc.,

Image credits :Emmanuel Agu, WPI, CS 543

Scattering

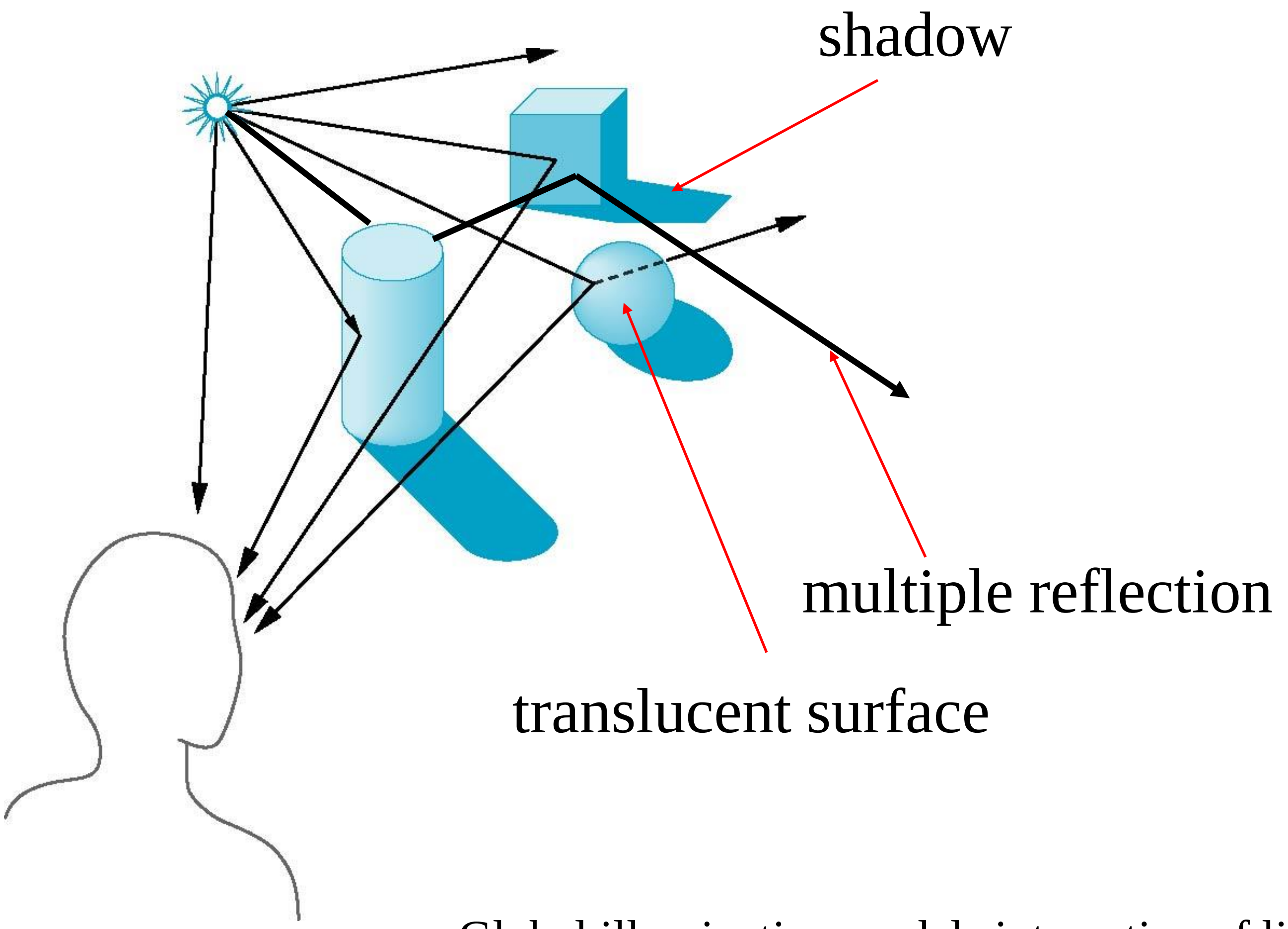
- Light strikes object A
  - Some scattered.
  - Some absorbed.
- Some of scattered light strikes object B
  - Some scattered.
  - Some absorbed.
- Some of this scattered light strikes A and so on.



Rendering Equation

- The infinite scattering and absorption of light can be described by the *rendering equation*
  - Which cannot be solved in general
  - Ray tracing solves special case for perfectly reflecting surfaces
- Rendering equation is a Global illumination model
  - Reflection
  - Shadows
  - Multiple scatterings from object to object

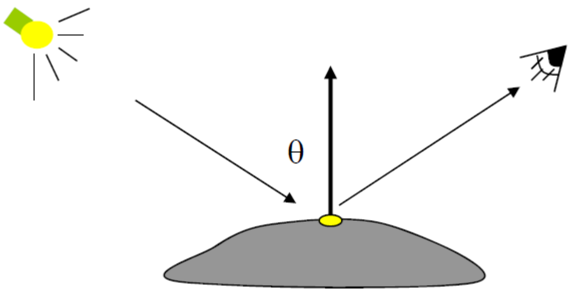
Global Illumination Model



Global illumination models interaction of light from all surfaces in scene (track multiple bounces)

Local vs Global Illumination Model

- Local Illumination model does not track inter-reflections, transmission.



Local illumination model only considers single interactions between light sources and surfaces

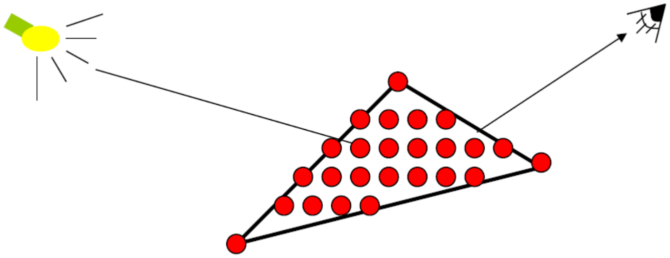
- Correct shading requires a global calculation involving all objects and light sources
  - This is incompatible with pipeline architecture which shades each polygon independently (i.e., local rendering)
- In computer graphics, especially real time graphics, we are happy if things “look right”
  - There exist many techniques for approximating global effects

Image credits :Emmanuel Agu, WPI, CS 543

Light-Material Interaction

- Light that strikes an object is partially absorbed and partially scattered (reflected)

- The amount reflected light determines the colour and brightness of the object
  - A surface appears red under white light because the red component of the light is reflected, and the rest is absorbed

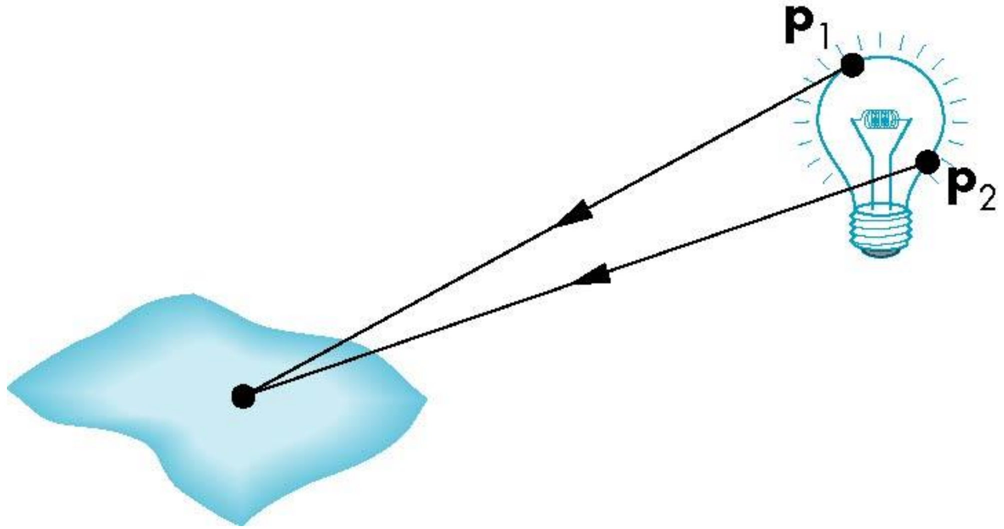


The reflected light is scattered in a manner that depends on the smoothness and orientation of the surface

Image credits :Emmanuel Agu, WPI, CS 543

Light Sources

- General light sources are difficult to work with because we must integrate light coming from all points on the source



Finite-size Light Source & Shadows

- Real light sources have a finite size
- Parts that do not see the light source at all will be the darkest
  (inner shadow circle)
- Parts that see the full light sources will be the brightest
  (outside the two circles)
- Parts that see the light source partially will have brightness values in between the two extremes depending on how much of the light source is visible
  (outer circle excluding the inner circle)

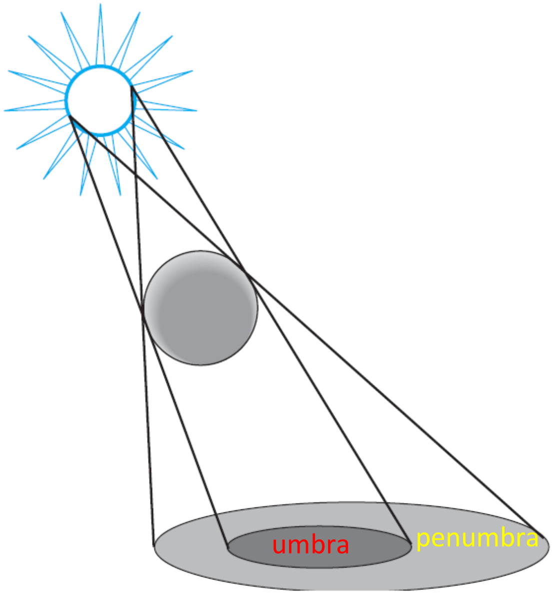
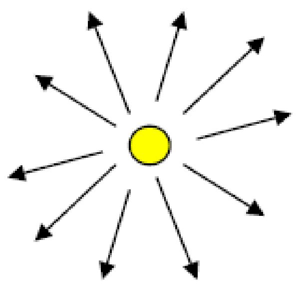


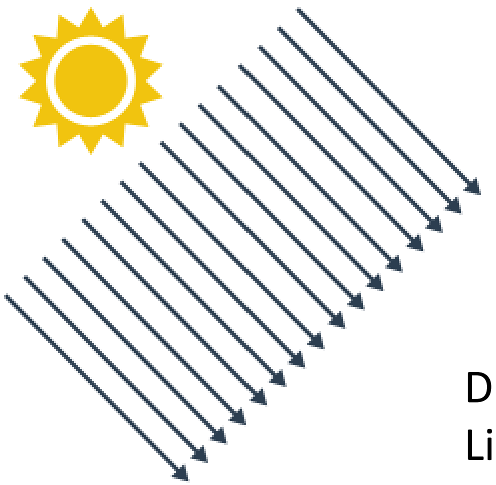
FIGURE 5.8 Shadows created by finite-size light source.

Simple Light Sources

We generally use simpler light sources



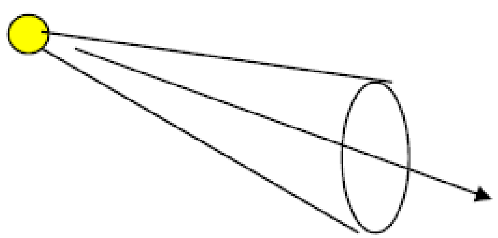
Point light



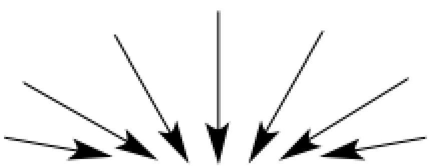
Directional Light

Image source

A directional source (or distant source) uniformly lights a scene



Spot light



Ambient light

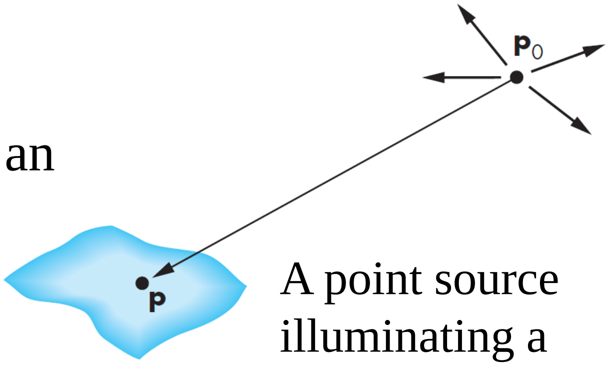
Image source

Light intensity can be **independent** or **dependent** of the distance between object and the light source

Simple Light Sources

- A **point source**
  - Is the simplest light source.
  - Is modelled with a position $\mathbf{p}_0 = (x, y, z, 1)^{\mathrm{T}}$ and an illumination intensity $\mathbf{I} = (I_{\mathrm{r}}, I_{\mathrm{g}}, I_{\mathrm{b}})$.
  - Illuminates equally in all directions.

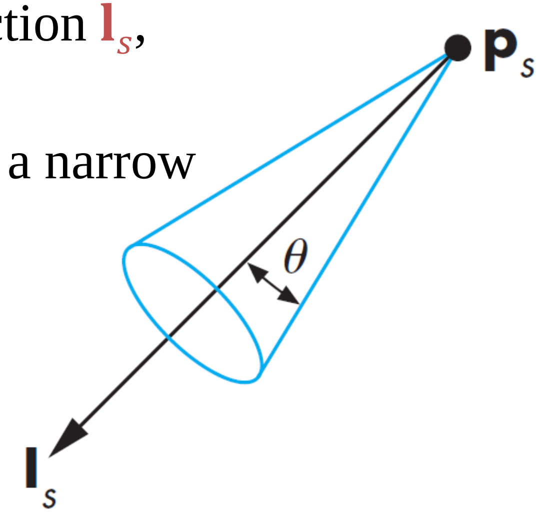
  A distance term can be incorporated to attenuate **I**.



A point source illuminating a surface.

- A **spotlight**
  - Is modelled with a cone whose apex is at $\mathbf{p}_s$, a direction $\mathbf{l}_s$, the cone angle θ, and an intensity $\mathbf{I} = (I_{\mathrm{r}}, I_{\mathrm{g}}, I_{\mathrm{b}})$.
  - Is a point source whose emitted light is restricted to a narrow range of angles.
  - If $\theta = 180°$, the spotlight becomes a point source.

  A distance term can be incorporated to attenuate **I**.



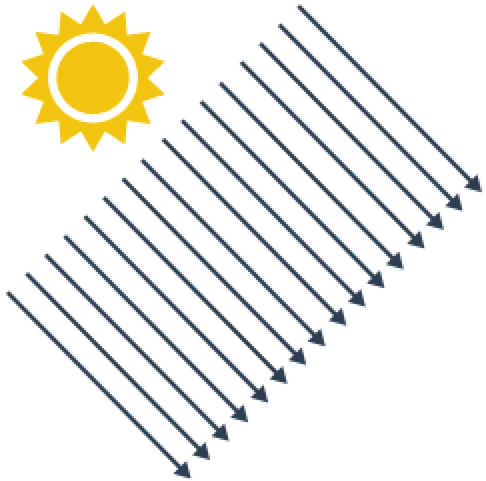
Light Attenuation with Distance

- For point light sources, light intensity decreases with the square of the distance i.e. multiply the intensity by $\frac{1}{d^2}$ (where d is the distance from light source)

- Point sources are easy to model but they result in harsh (unrealistic) rendering with very sharp drop in brightness and crisp shadows.

- Hence, the light attenuation is generally modelled by a quadratic term $\frac{1}{a+bd+cd^2}$ (where a, b, c are user defined parameters)

Simple Light Sources (cont.)



- A **directional light source**
  - Is also known as a **distance light source**.
  - Illuminates objects with parallel rays of light.
  - Is modelled with a direction $\mathbf{d} = (x, y, z, 0)^{\mathrm{T}}$ and a and intensity $\mathbf{I} = (I_{\mathrm{r}}, I_{\mathrm{g}}, I_{\mathrm{b}})$.

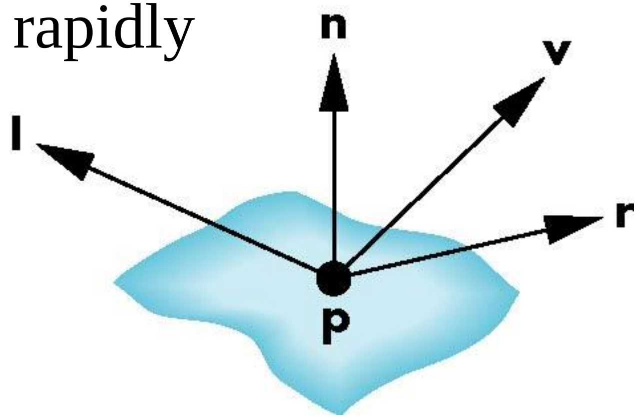
- **Ambient light**
  - Models the same amount of light everywhere in the scene. The ambient illumination $\mathbf{I}_{\mathrm{a}} = (I_{\mathrm{r}}, I_{\mathrm{g}}, I_{\mathrm{b}})$ is therefore characterized by an intensity that is identical at every point in the scene.
  - Can also be used to model the integration of multiple sources on a reflecting surface in the scene.

The Phong Reflection Model

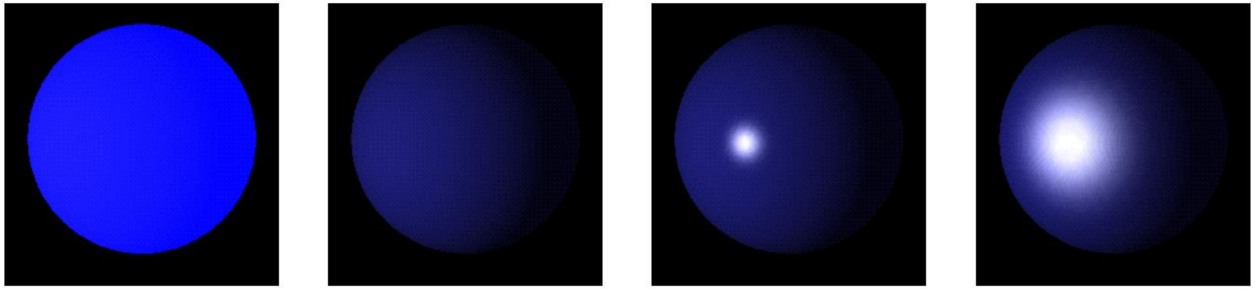
The Phong Model:

Used to model local illumination of points on a surface

- Is a simple model that can be computed rapidly
- Has three terms
  - Diffuse term
  - Specular term
  - Ambient term
- Compute each component separately



Vertex/point illumination = ambient + diffuse + specular

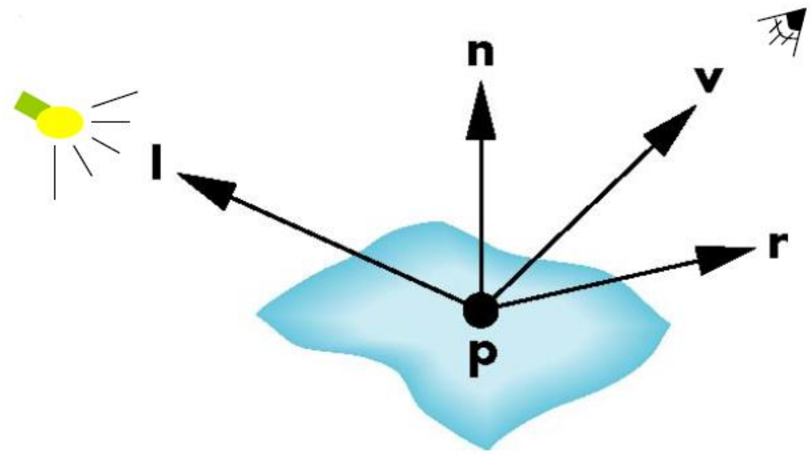


ambient diffuse specular combined

Image source

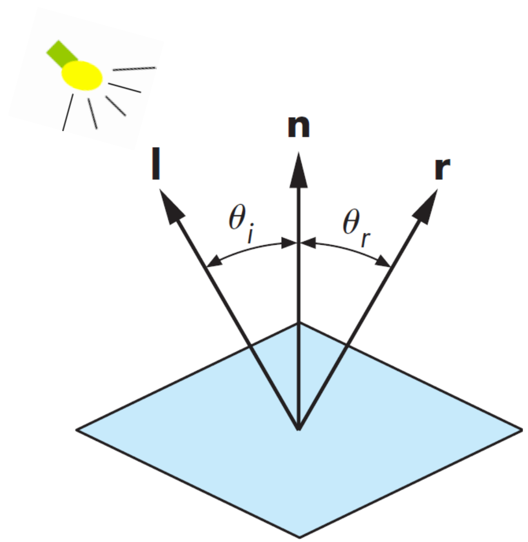
The Phong Reflection Model

- Uses the four vectors below to calculate illumination for an arbitrary point **p** on a surface:



  - Vector **l** (to light source)
  - Vector **v** (to viewer or camera)
  - Vector **n** (Normal vector at p)
  - Vector **r** (Perfect reflector of **l** with respect to **n**)

Angle of reflection = angle of incidence

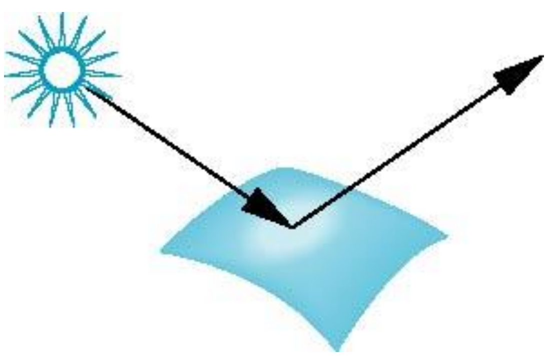


- Normal is determined by surface orientation
- The three vectors **(l, n, r)** must be coplanar

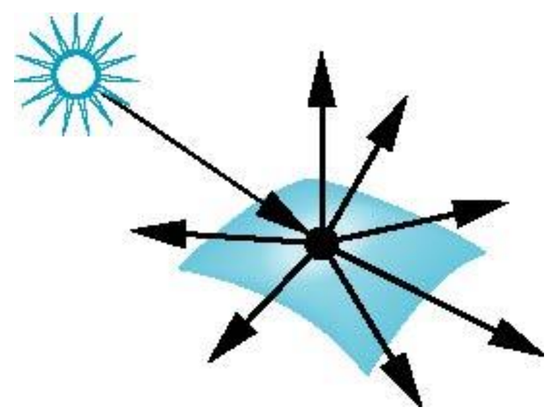
Surface Types

- For **smooth surfaces**, the reflected light concentrates more in the direction that is a perfect mirror reflection of the incident light.

- For **rough surfaces**, the reflected light scatters in all directions



smooth surface



rough surface

Lambertian Surface

Many objects in the world have a surface appearance loosely described as “matte,” indicating that the object is not at all shiny, such as:

- Paper
- Wood
- Dry unpolished stones

Light scattered equally in all directions

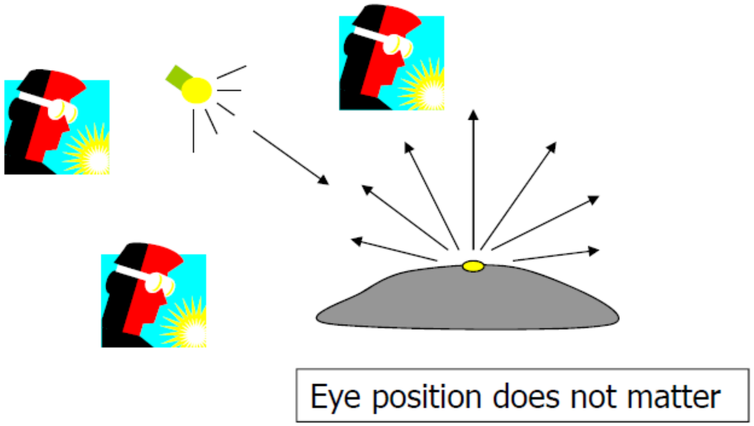


Image credits :Emmanuel Agu, WPI, CS 543

Lambertian Reflection (Diffuse term)

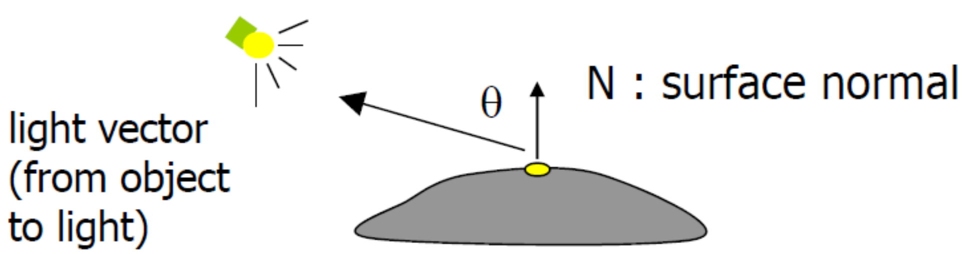
Lambertian object obeys *Lambert's cosine law.*

Amount of light reflected is proportional to the vertical component of incoming light

- i.e., reflected light ~$\cos\theta_{\mathrm{i}}$
- where $\cos\theta_{\mathrm{i}} = \mathbf{l}\cdot\mathbf{n}$ when vectors **l** and **n** are normalized to unit vectors.
- There are also three coefficients, $k_{\mathrm{r}}, k_{\mathrm{g}}, k_{\mathrm{b}}$ that show how much of each color component is reflected (these coefficients are specific to the surface, i.e., different types of surfaces have different values).

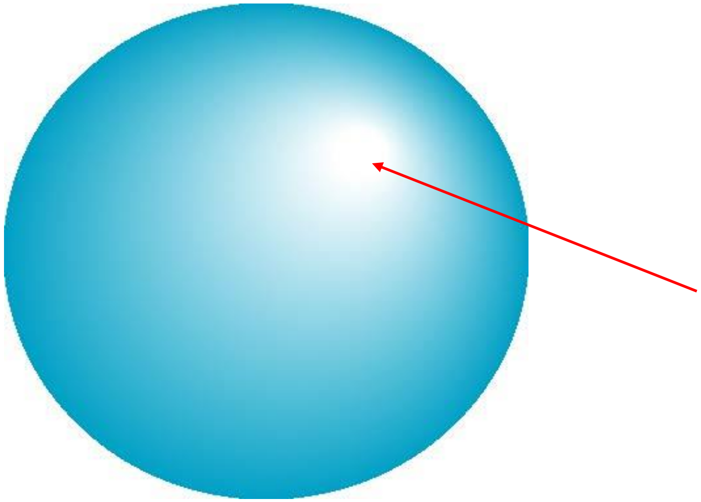
$$I_{diffuse} = k_d L_{\mathrm{d}} \cos\theta$$

L_{d} is the incoming intensity



Specular Surfaces

- Whereas a diffuse surface is rough, a specular surface is smooth.
- Smooth surfaces show specular highlights due to incoming light being reflected in directions concentrated close to the direction of a perfect reflection.



Specular highlight
-shows colour of the light source (rather than of the object)

Modeling Specular Reflections

- Phong proposed using a term $\cos^{\alpha}\phi$ that drops off as the angle between the viewer and the ideal reflection increased

Each of these terms has a red, green, and blue component. This formula repeats 3 times, one for each component.

~ means "proportional to".

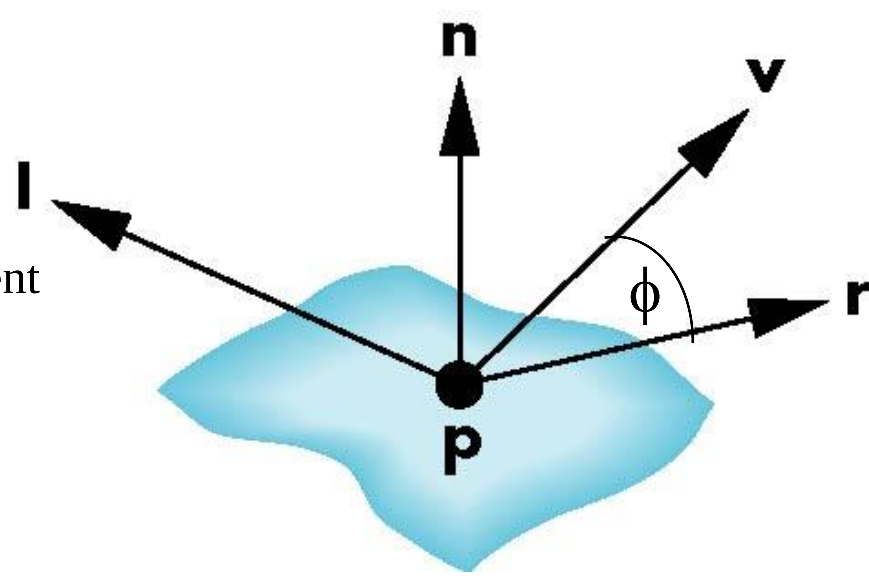
$$I_{\text{specular}} \sim k_s L_s \cos^{\alpha}\phi$$

reflected Intensity

coefficient of surface/ absorption co-effiecient

incoming intensity

shininess coefficient of material



Small ϕ = more specular

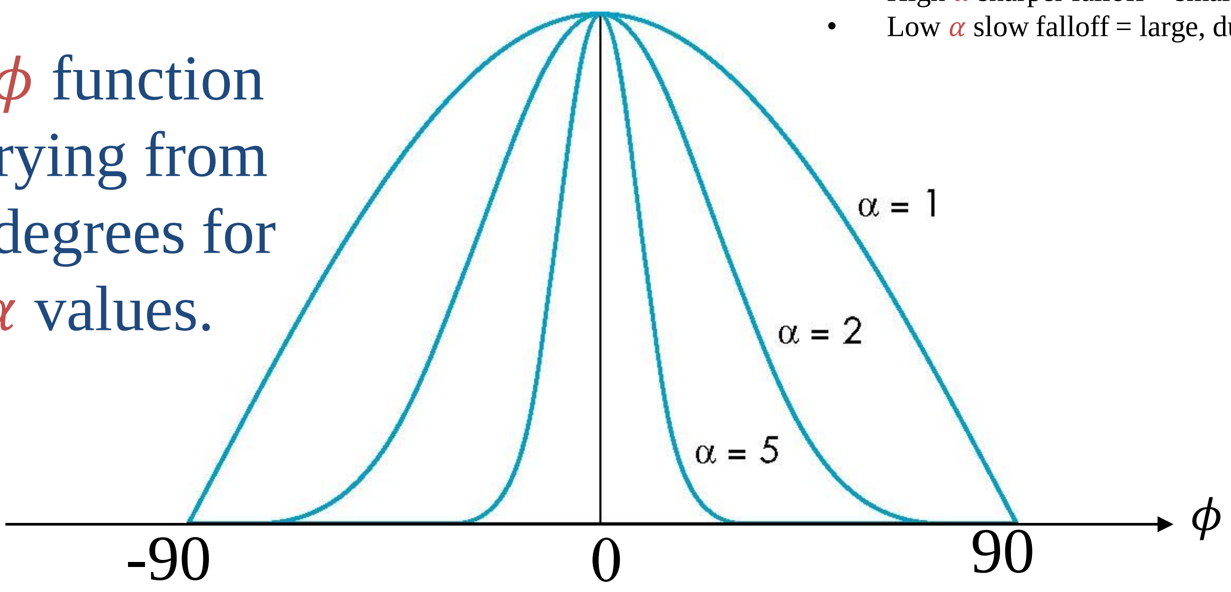
The Shininess Coefficient α

- Values between 100 and 200 correspond to **metals**.
- Values between 5 and 10 give surfaces that look like **plastic**.

α controls falloff sharpness

- High α sharper falloff = small, bright highlight
- Low α slow falloff = large, dull highlight

The $\cos^{\alpha} \phi$ function with ϕ varying from -90 to 90 degrees for different α values.



Ambient Reflection

- Ambient light is the result of multiple interactions between (large) light sources and the objects in the environment
  - Does not come from a specific direction
  - "Base" lighting

- Amount and color of ambient reflection depends on both the incident ambient light(s) and the material properties of the object
- $I_{ambient} = k_a L_a$

reflection coefficient of surface

intensity of incoming ambient light

Further Reading

"Interactive Computer Graphics – A Top-Down Approach with Shader-Based OpenGL" by Edward Angel and Dave Shreiner, 6th Ed, 2012

- Sec. 5.1. Light and Matter
- Sec. 5.2. Light Sources
- Sec. 5.3. The Phong Reflection Model