Shading & Illumination Models

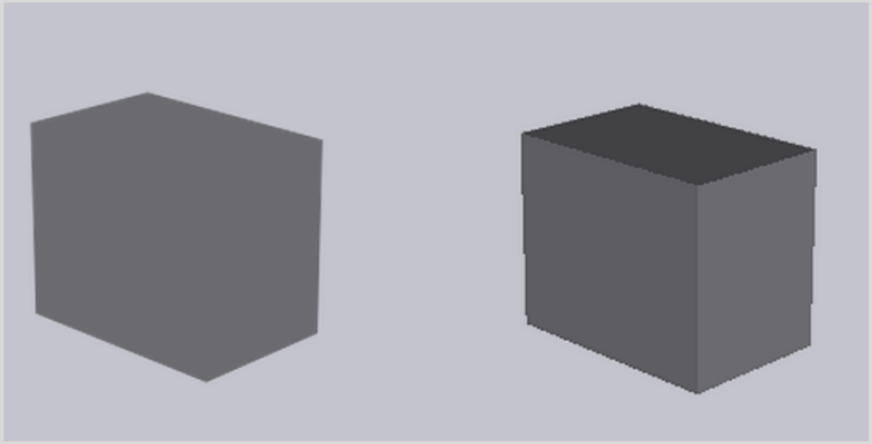
Lecturer: Carol O'Sullivan

Credits: Some slides from Rachel McDonnell

Shading and Illumination

- Shading: determining the colour of each pixel, includes illumination, transparency, texturing, and shadows
- Illumination: simulating light reflectance, absorption, and transmission

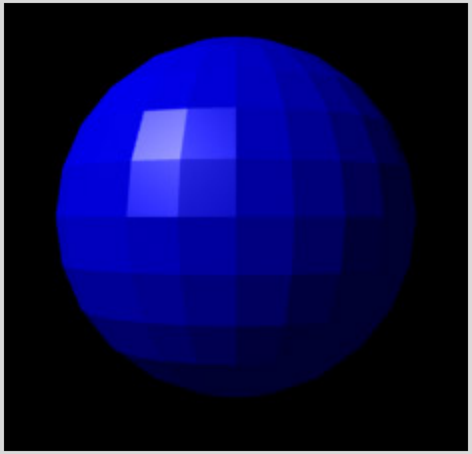
Shading



- Flat
  - Compute illumination once per polygon and apply it to whole polygon (per vertex with no interpolation)
- Smooth/Gouraud shading
  - Compute illumination at borders and interpolate (per vertex)
- More accurate/Phong shading
  - Compute illumination at every point of the polygon (per fragment)

Flat Shading

- Illumination model is applied only once per polygon
- Gives low-polygon models a faceted look.
- Works poorly if the model represents a curved surface.
- Smooth appearance implies large number of polygons.
- Adding more facets helps but… slows down the rendering.
- Advantageous in modeling boxy objects.
- Effectiveness is tempered by “Mach banding”.



Mach Banding

- Optical illusion named after physicist Ernst Mach
- Perceived intensity change at edges are exaggerated by receptors in our eyes, making the dark facet look darker and the light facet look lighter.

Mach Banding

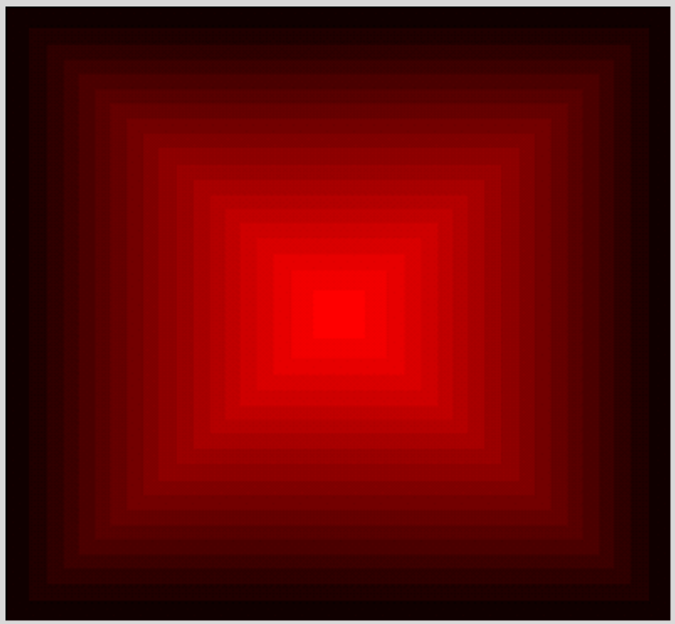
- Brightness perceived by the eye tends to overshoot at the boundaries of regions of constant intensity
- Abrupt changes in the shading of two adjacent polygons are perceived to be even greater

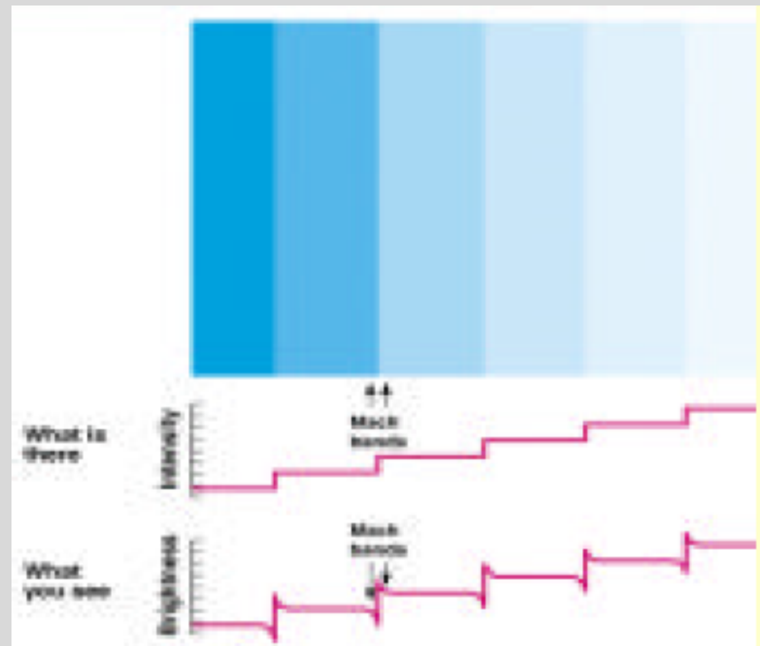
Mach Band Effect

- These "Mach Bands" are not physically there. Instead, they are illusions due to excitation and inhibition in our neural processing

The bright bands at 45 degrees (and 135 degrees) are illusory.

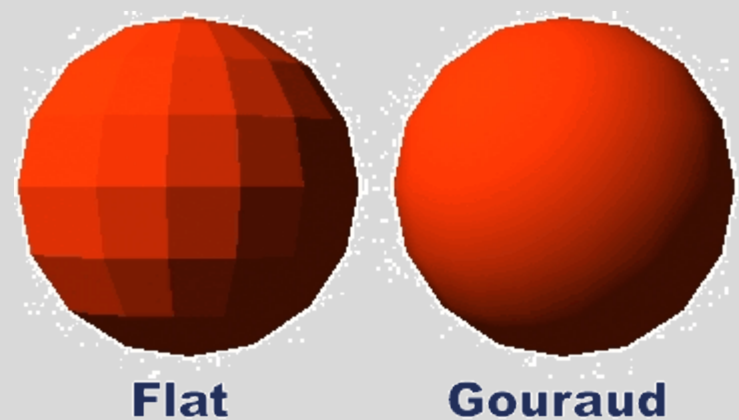
The intensity of each square is the same.





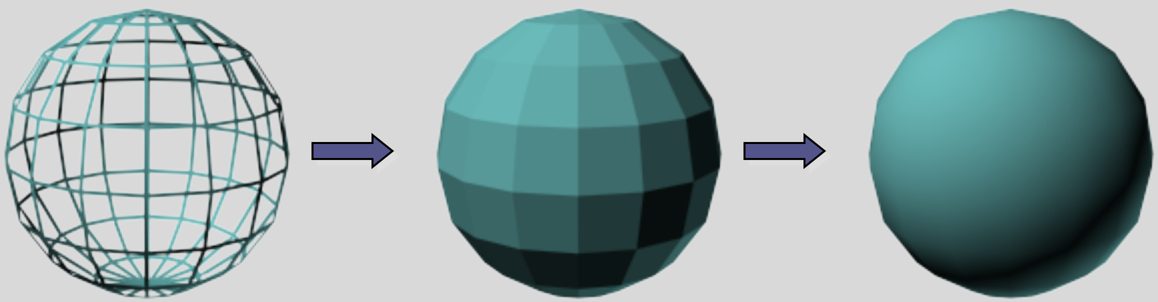
Smooth Shading

- Used to approximate curved surfaces with a collection of polygons.
- Calculate illumination based on approximation of curved surface.
- Does not change geometry, <u>silhouette</u> is still polygonal.



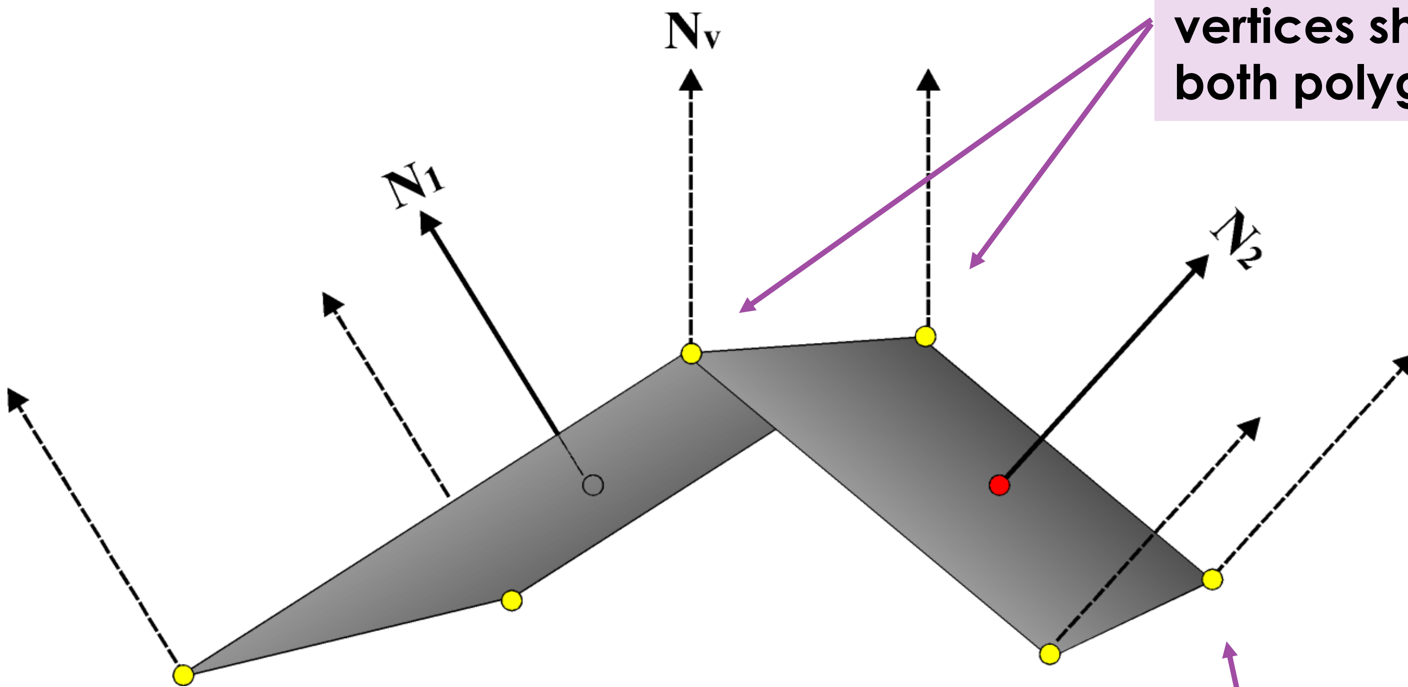
Surface Normal Interpolation

- Often we are approximating curved/smooth surfaces with polygons ⇒ we get *edge artifacts* at polygon boundaries:



- To combat this we determine average normals at the vertices of the polygons by averaging the normals of each polygon that shares a vertex and storing the result with that vertex.

Determining Vertex Normals



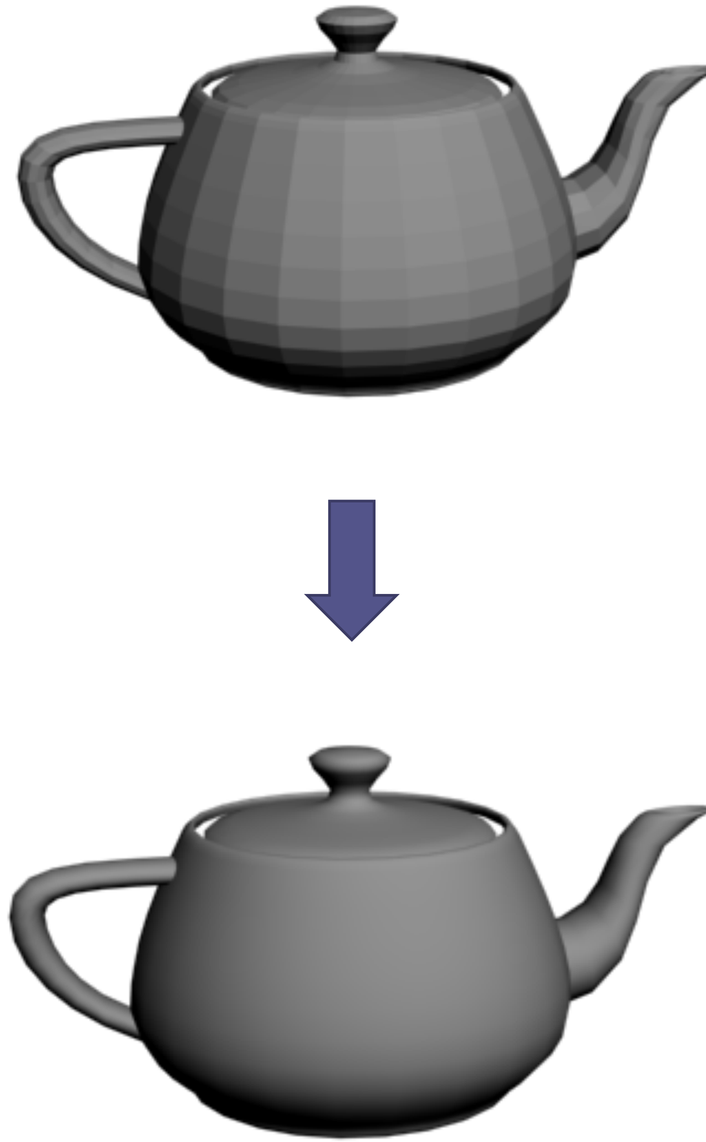
vertices shared between both polygons

vertices not shared

$$N_v = \frac{\frac{1}{2}\left(N_1 + N_2\right)}{\left|\frac{1}{2}\left(N_1 + N_2\right)\right|}$$

Gouraud Shading

- Gouraud shading is a method for linearly interpolating a colour or shade across a polygon.
- It was invented by Henri Gouraud in 1971.
- It is a very simple and effective method of adding a curved feel to a polygon that would otherwise appear flat.



Gouraud Shading

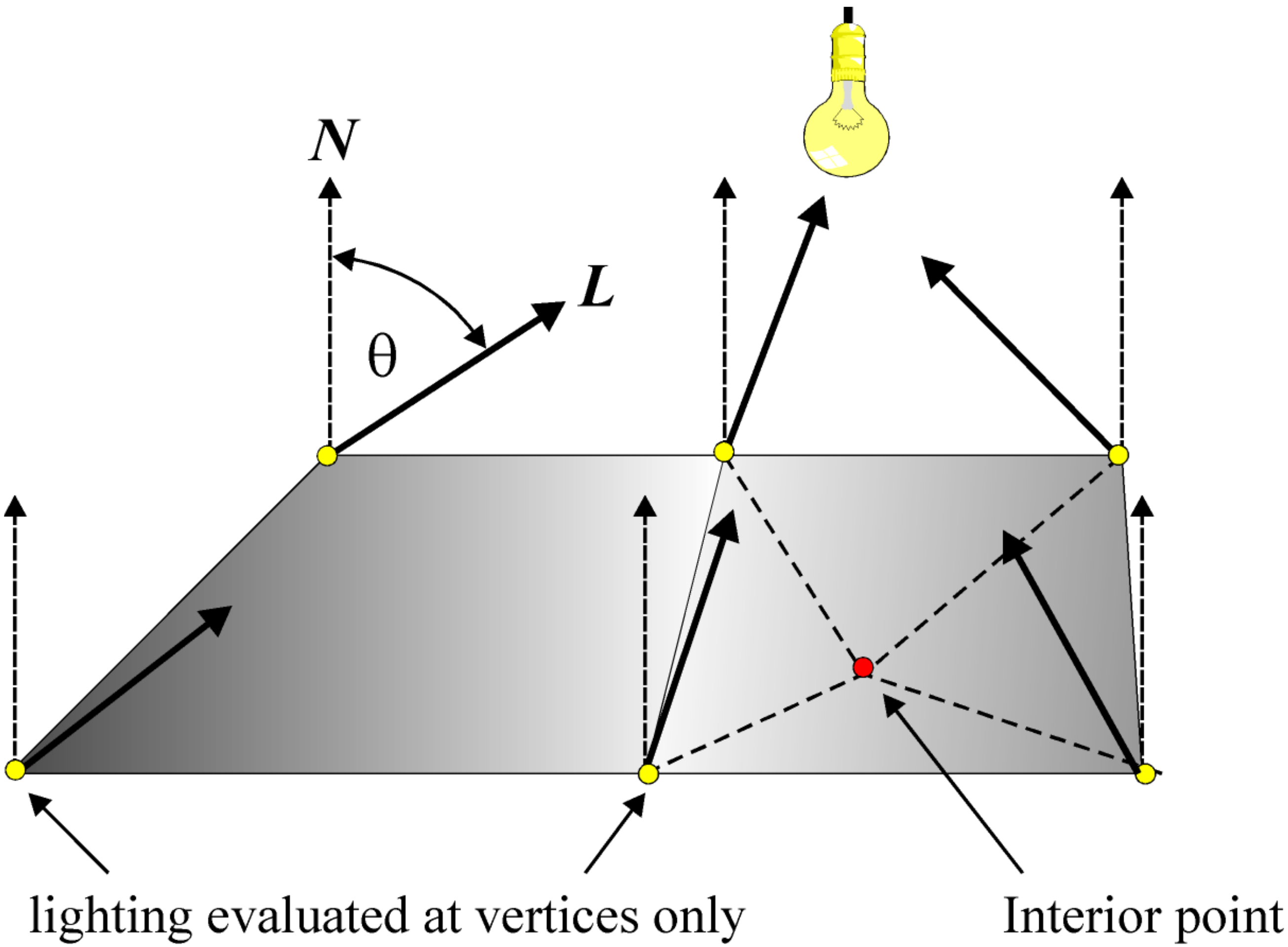
- For each interior point in the polygon being shaded we interpolate the intensity determined at the vertices.
- We do this in exactly the same way that we interpolated colour across the surface of a polygon.
- This is known as *Gouraud shading*.

$\Rightarrow$ we need to do lighting calculations at vertices only

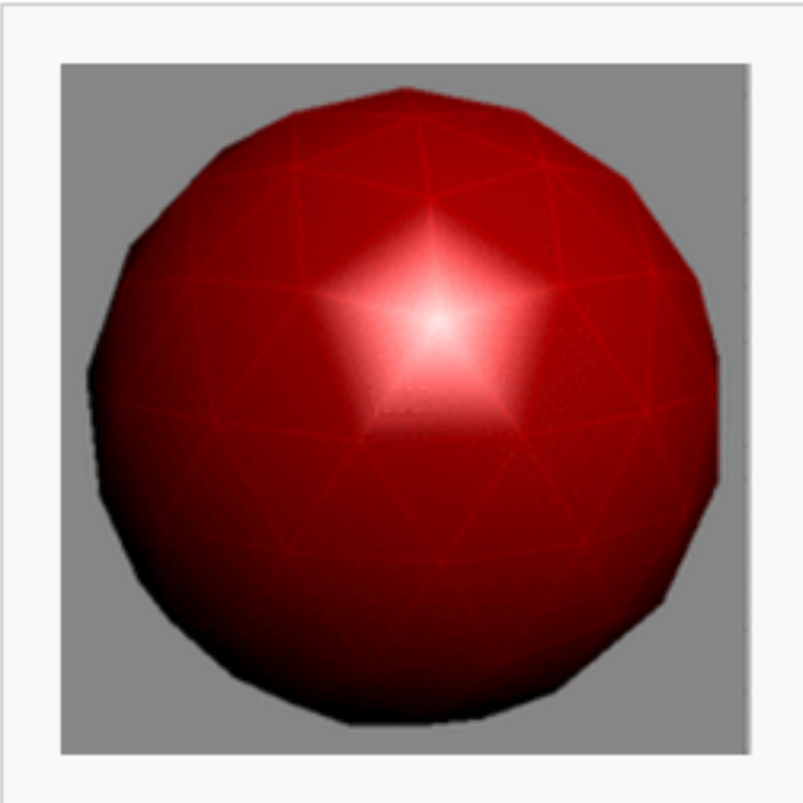
$\Rightarrow$ lighting is correct at vertices only

- This is OK if the polygons are small. As polygons increase in size the errors also increase leading to often unacceptable results.

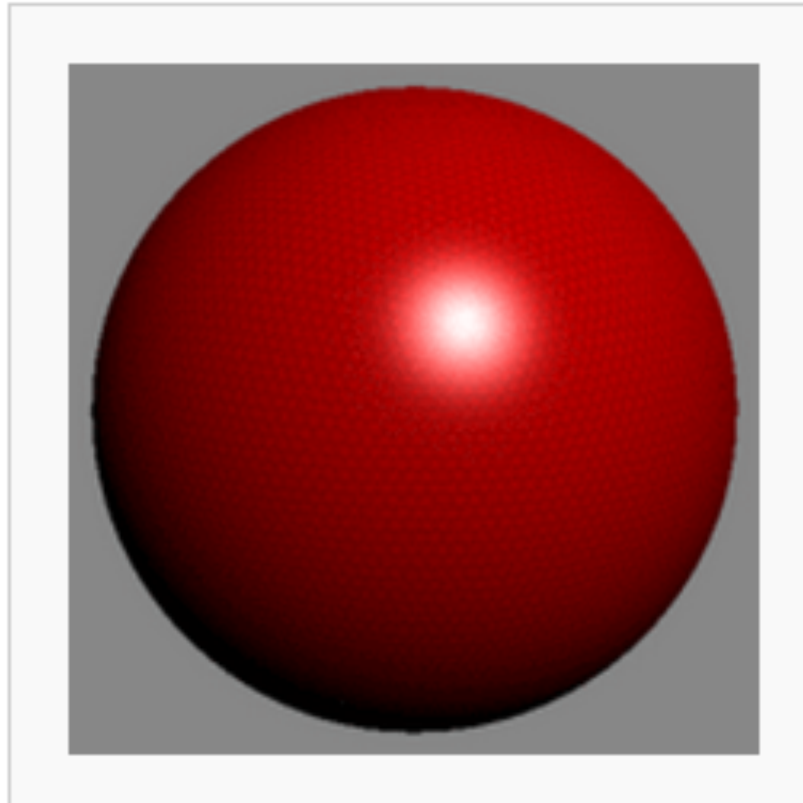
Gouraud Shading



Interpolation Errors



Poor behaviour of the specular highlight

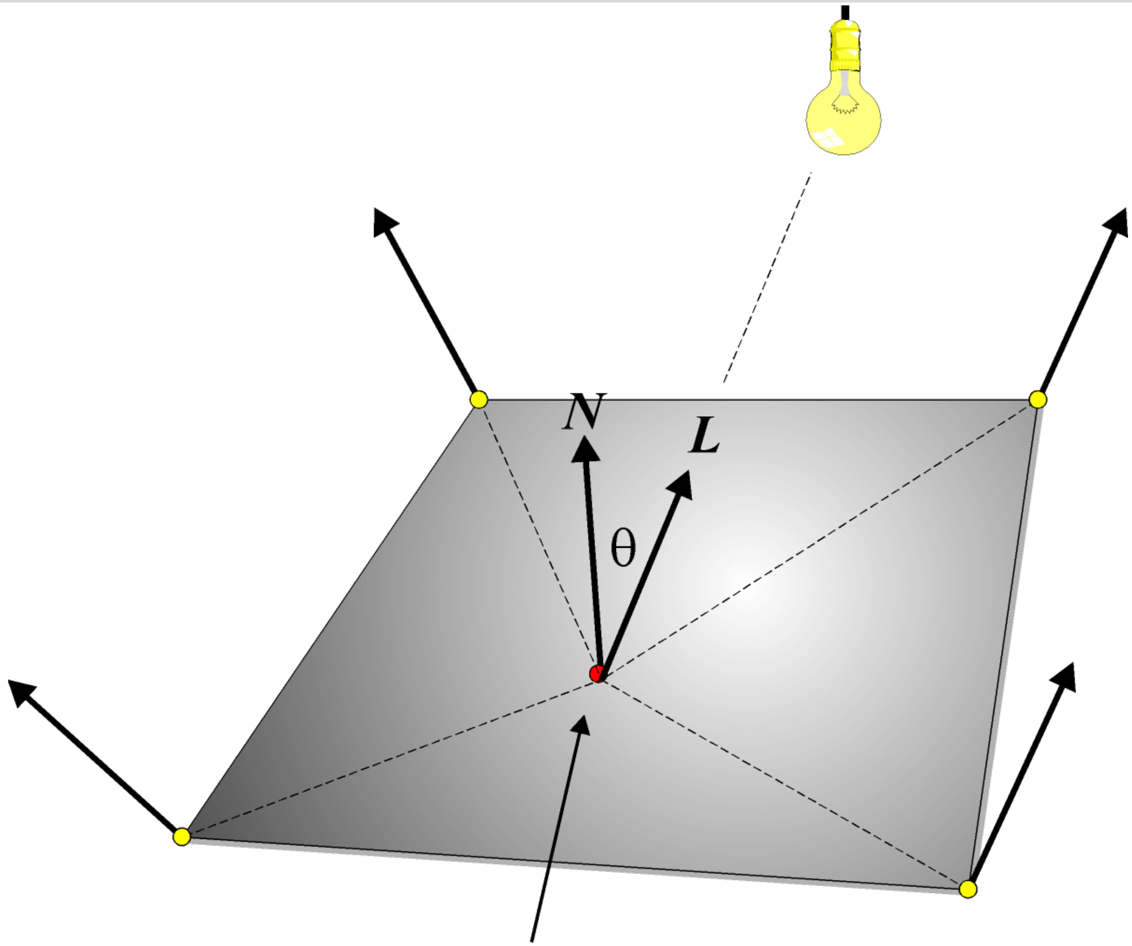


Improvement with very high polygon count

Phong Shading

- To improve upon Gouraud shading we can *interpolate the normals* across the surface and apply the lighting model at each point in the interior.
- This assumes we are working with polygonal models.
- Care must be taken to ensure that all interpolated normals are of unit length before employing the lighting model.
- This is known as *Phong shading* (as opposed to the *Phong illumination model*).

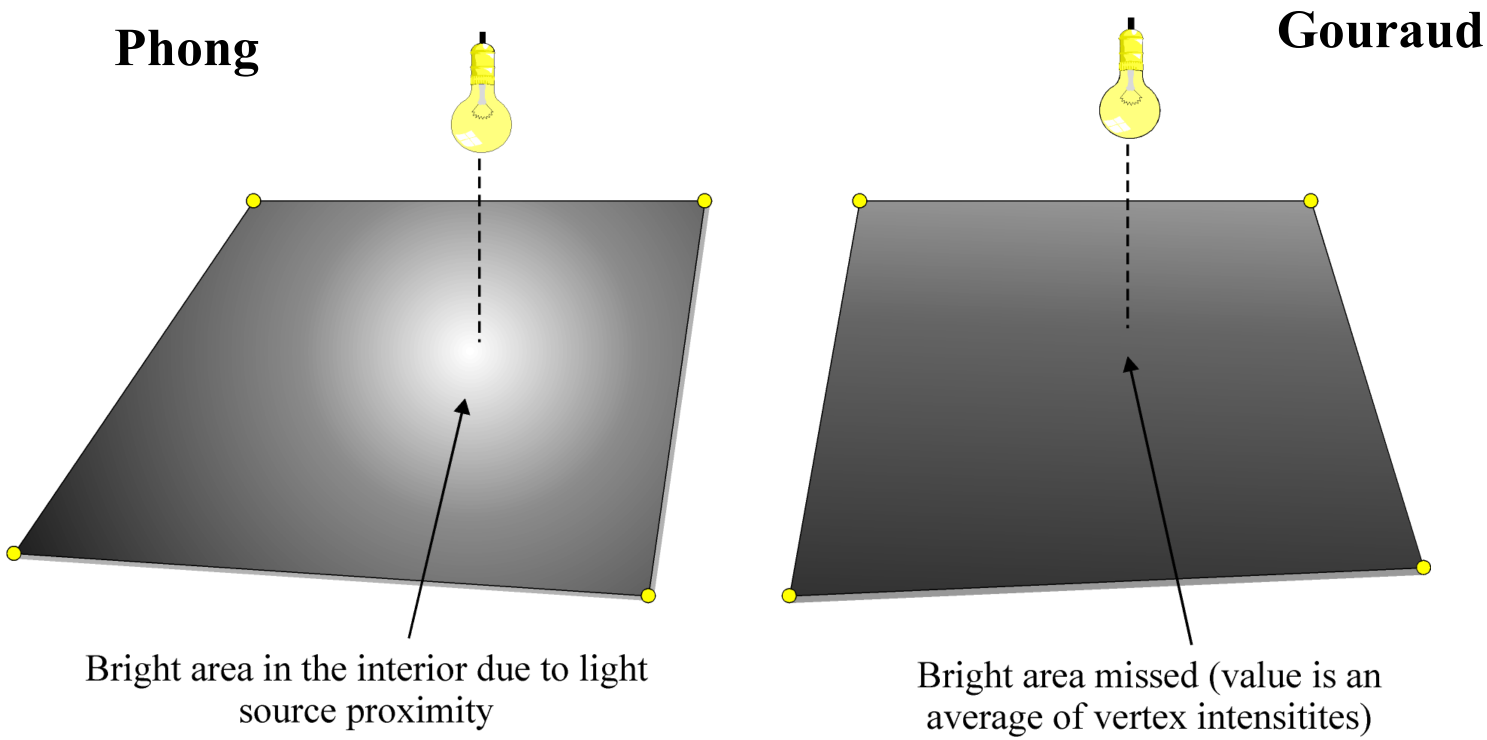
Phong Shading



Interpolated normal at the interior point

Phong Shading

- Phong shading is capable of reproducing *highlights* within the interior of a polygon that Gouraud shading will miss:

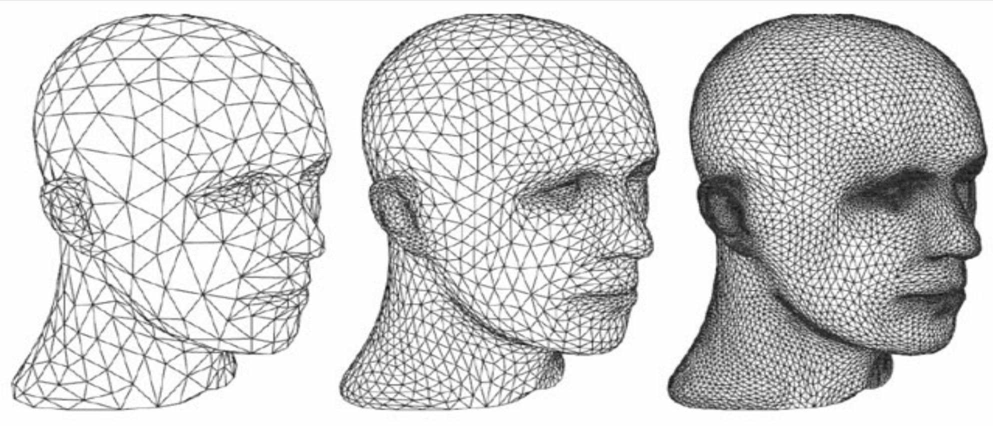


Phong vs Gouraud

- Phong shading:
  - Handles specular highlights much better
  - Does a better job in handling Mach bands
  - But more expensive than Gouraud shading

Which shading was used?

Two Solutions



- Specular highlights that do not overlap any vertex will not be visible on meshes rendered with Gouraud shading
  - Re-tesselate the cylinder so that some vertices lie within the specular highlight (extra memory and computation costs.
  - Switch to per-pixel Phong shading

Shading model vs illumination model

- There is a difference between the shading model and the illumination model used in rendering scenes,
  - the illumination model captures how light sources interact with object surfaces, and
  - the shading model determines how to render the faces of each polygon in the scene.
- The shading model depends on the illumination model, for example
  - some shading models invoke an illumination model for every pixel (such as ray tracing),
  - others only use the illumination model for some pixels and then shade the remaining pixels by interpolation (such as Gouraud shading).

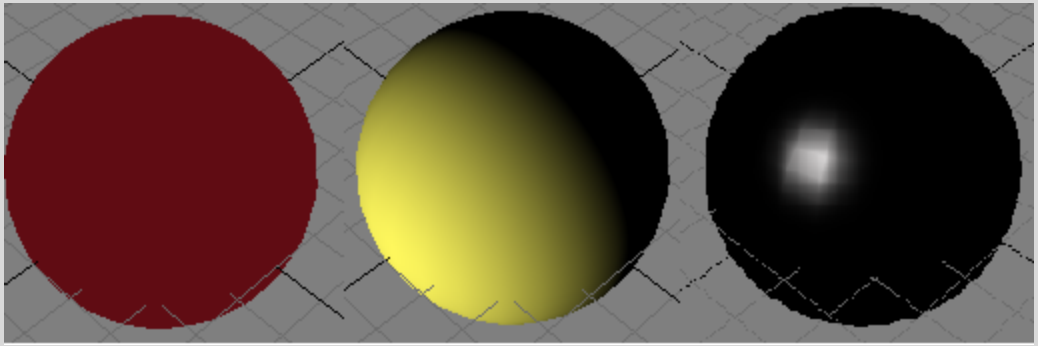
Shading model vs illumination model

- The illumination model is about determining how light sources interact with object surfaces

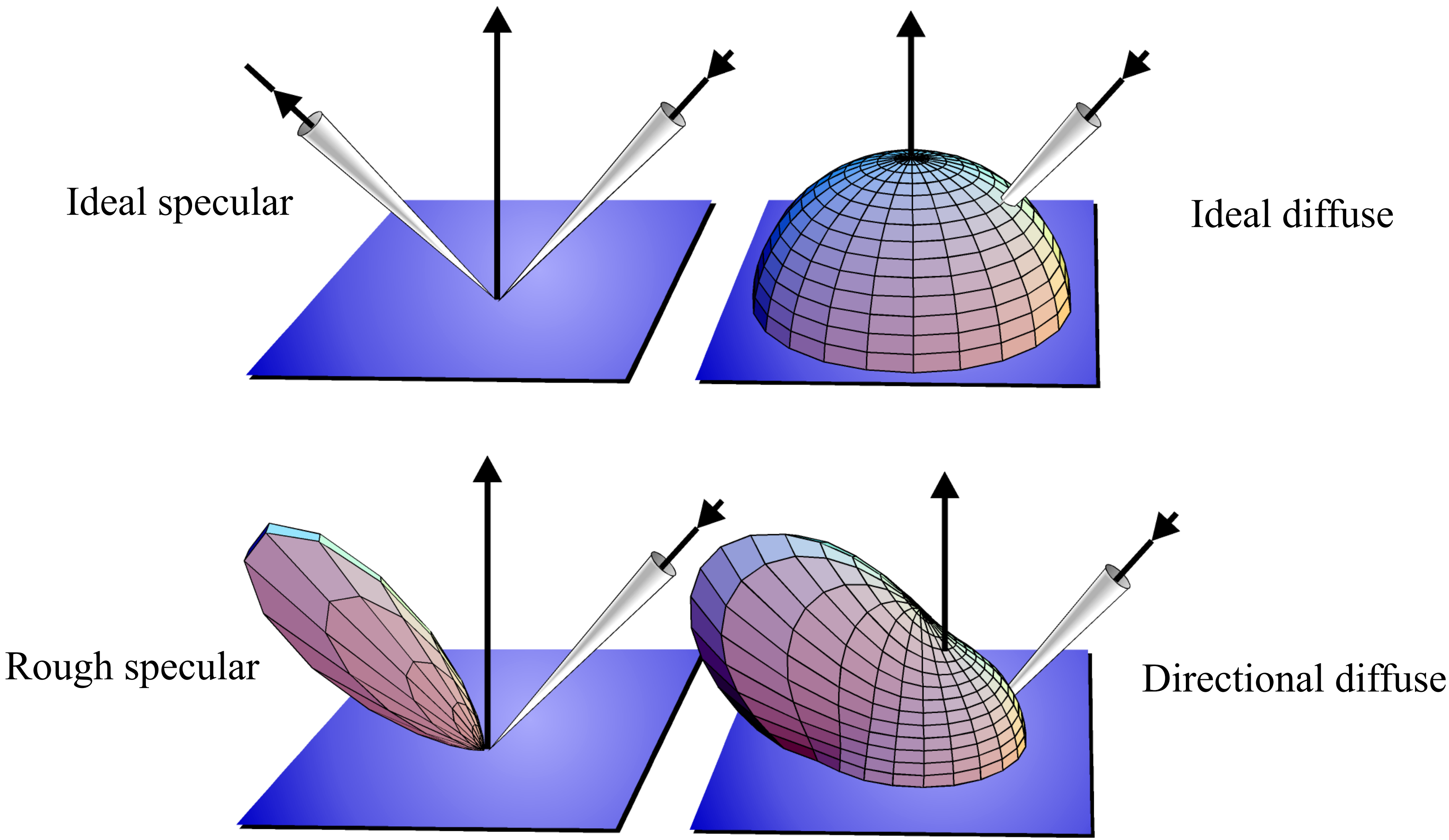
- whereas the shading model is about how to interpolate over the faces of polygons, given the illumination.

Illumination Models

- Lambertian *(diffuse)*
- Phong *(specular)*
- Ambient *(all other light)*



BRDF Approximations



Reflectance Equation

- The *reflectance equation* relates reflected radiance to incoming radiance that is scattered according to the surface's BRDF:

$$\underbrace{L_r(x,\omega_r)}_{\text{Reflected radiance}} = \int_\Omega \underbrace{f_r(x,\omega_i,\omega_r)}_{\text{BRDF}}\, \underbrace{L_i(x,\omega_i)}_{\text{Incident radiance}}\, \underbrace{\cos\theta_i}_{\text{cos of incident angle}}\, d\omega_i$$

Ω = domain of integration $\longrightarrow$ Hemisphere if surface is opaque

Radiance Equation

- The radiance equation includes a *self-emitted term* to account for light sources.
- This is the most important equation in rendering theory.
- We solve for $L_r(x, \omega_r)$ at each visible point in the scene.

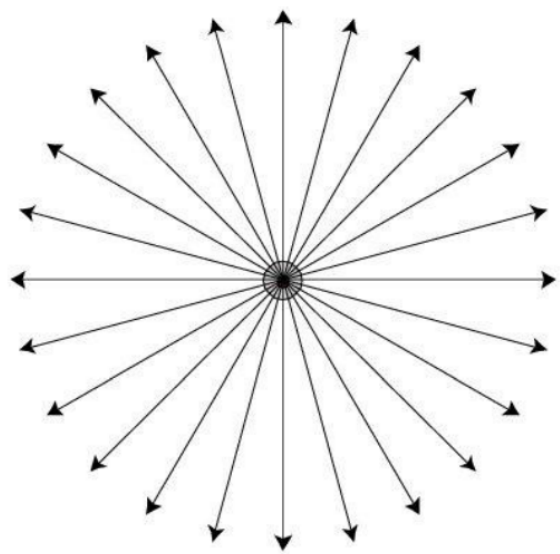
$$L_r(x,\omega_r) = \underbrace{L_e(x,\omega_r)}_{\substack{\text{Self emitted radiance} \\ \text{(non zero for light sources)}}} + \int_{\Omega} f_r(x,\omega_i,\omega_r) L_i(x,\omega_i) \cos\theta_i \, d\omega_i$$

Lighting Equation Approximations

- The Radiance equation: no general solution.
- We need to make certain assumptions to solve the lighting problem. This gives rise to different illumination models.
- Factors that influence lighting: intensity of light, visibility, BRDF
- Light-material interactions: specular reflection, diffuse reflection, transmission

Light Sources

- To simplify the solution to the radiance equation we normally employ ***isotropic point light sources*** defined by:
  - Position and Colour



Isotropic $\Rightarrow$ radiates energy equally in all directions

Light Sources

- Normally we wish to associate a radiance with a light source but the definition of radiance assumes an area over which energy is emitted/distributed
  - but *point sources have no area*

⇒ Use radiant intensity I instead (units Watts/sr).

⇒ A point source radiating energy *in all directions equally* has a radiant intensity of:

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

Total energy

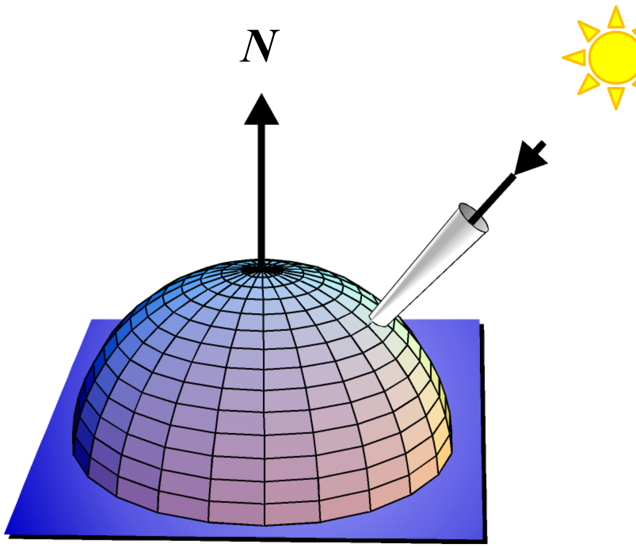
Solid angle
e.g., for the sun this would be a sphere

$4\pi r^2$

= sphere surface area

Diffuse Light

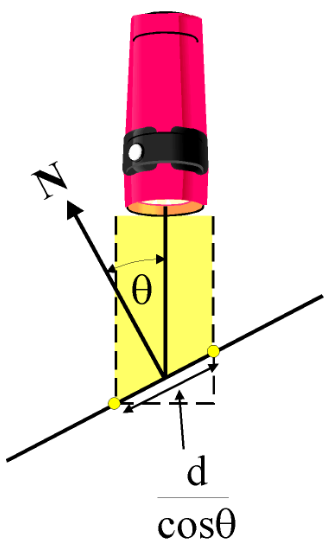
- The illumination that a surface receives from a light source and reflects equally in all directions
- This type of reflection is called Lambertian Reflection
- The brightness of the surface is independent of the observer position (since the light is reflected in all direction equally)

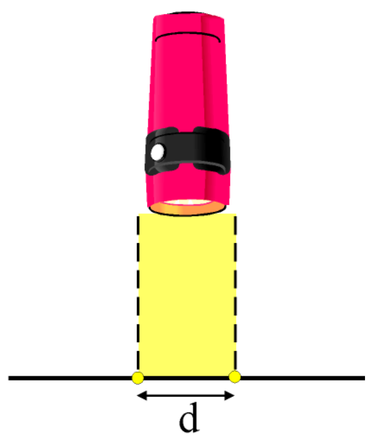


The Cosine Rule

- A surface which is oriented perpendicular to a light source will receive more energy per unit area (and thus appear brighter) than a surface oriented at an angle to the light source.
- The irradiance *E* is proportional to $\dfrac{1}{area}$
- As the area increases, the irradiance decreases therefore:

θ	$Cos(\theta)$
0	1.00
10	0.98
30	0.87
50	0.64
70	0.34
90	0.00
110	-0.34
130	-0.64



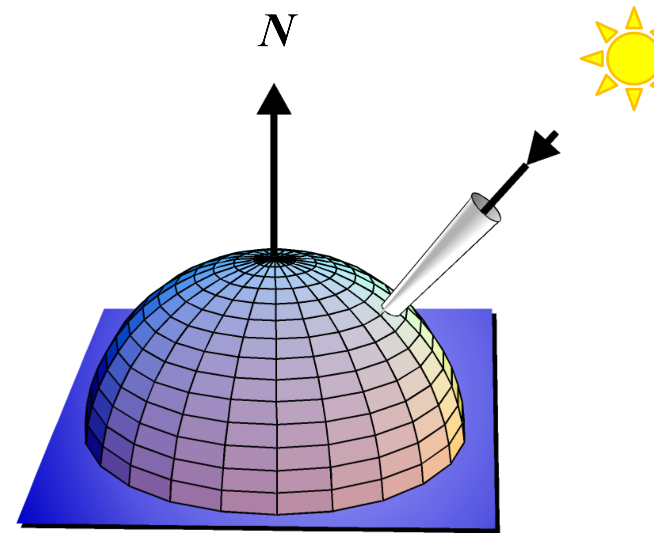


$$E = \frac{\cos\theta\,\Phi}{4\pi r^2}$$

As θ increases, the irradiance and thus the brightness of a surface decreases by cos θ

Lambertian Illumination Model

- We can use the cosine rule to implement shading of *Lambertian* or *diffuse* surfaces.
- Diffuse surfaces reflect light in all directions equally:
  - BRDF is a constant with respect to reflected direction
  - surface may be characterized by a *reflectance* ρ_d rather than a BRDF
  - the reflectance ρ_d gives the ratio of the total reflected power Φ_r to the total incident power Φ_i



$$f_r(x) = \frac{\rho_d(x)}{\pi}$$

← With unit reflectivity

$$\rho_d(x) = \frac{\Phi_i}{\Phi_r}$$

Reflectivity

- The reflectivity varies from zero for a completely absorbing ("black") surface, to one for a completely reflecting ("white") surface.
- There are no Lambertian surfaces in nature, but matte paper is good approximation except at grazing angles and near 90 degrees), where the surface begins to look "shiny."

Lambertian Illumination Model

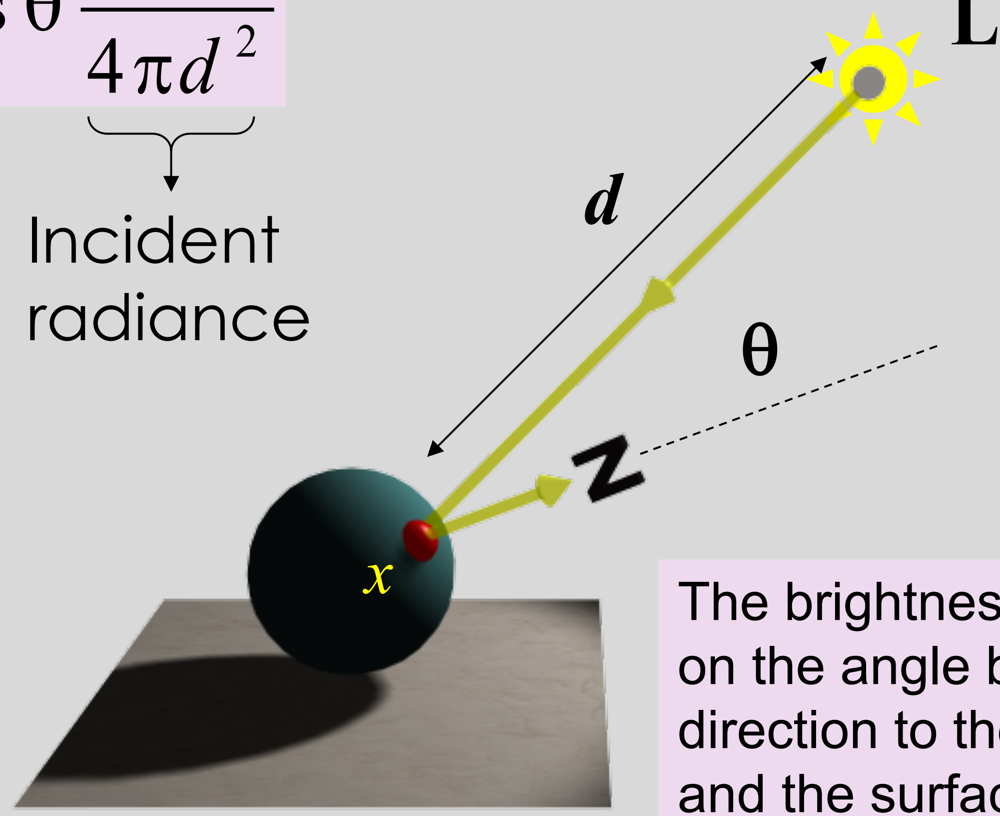
- To shade a diffuse surface we need to know:
  - *normal* to the surface at the point to be shaded
  - diffuse *reflectance* of the surface
  - *positions* and *powers* of the light source in the scene
- We will assume *isotropic point sources*

Isotropic $\Rightarrow$ radiates energy equally in all directions

Lambertian Illumination Model

- contribution from a single source is given by:

$$\underbrace{L_{r,d}(x,\cdot)}_{\text{Reflected radiance}} = \underbrace{\frac{\rho_d}{\pi}}_{\text{BRDF}} \cos\theta \underbrace{\frac{\Phi_S}{4\pi d^2}}_{\text{Incident radiance}}$$



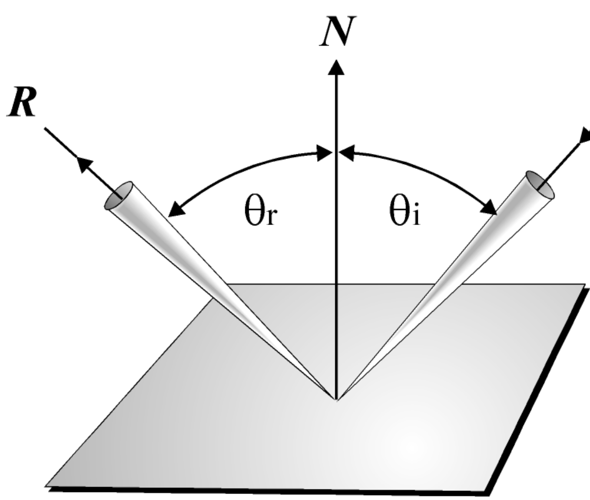
The brightness depends only on the angle between the direction to the light source and the surface normal

Lambertian surfaces

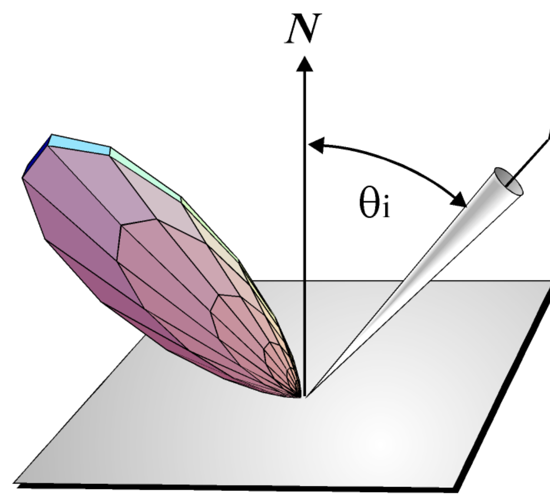
- Below are several examples of a spherical diffuse reflector with varying lighting angles.

Phong Illumination Model

- *Specular* surfaces exhibit a high degree of *coherence* in their reflectance, i.e. the reflected radiance depends very heavily on the outgoing direction.
  - An *ideal specular* surface is *optically smooth* (smooth even at resolutions comparable to the wavelength of light).
  - Most specular surfaces (rough specular) reflect energy in a tight distribution (or *lobe*) centered on the *optical reflection direction*:



Ideal Specular



Rough Specular

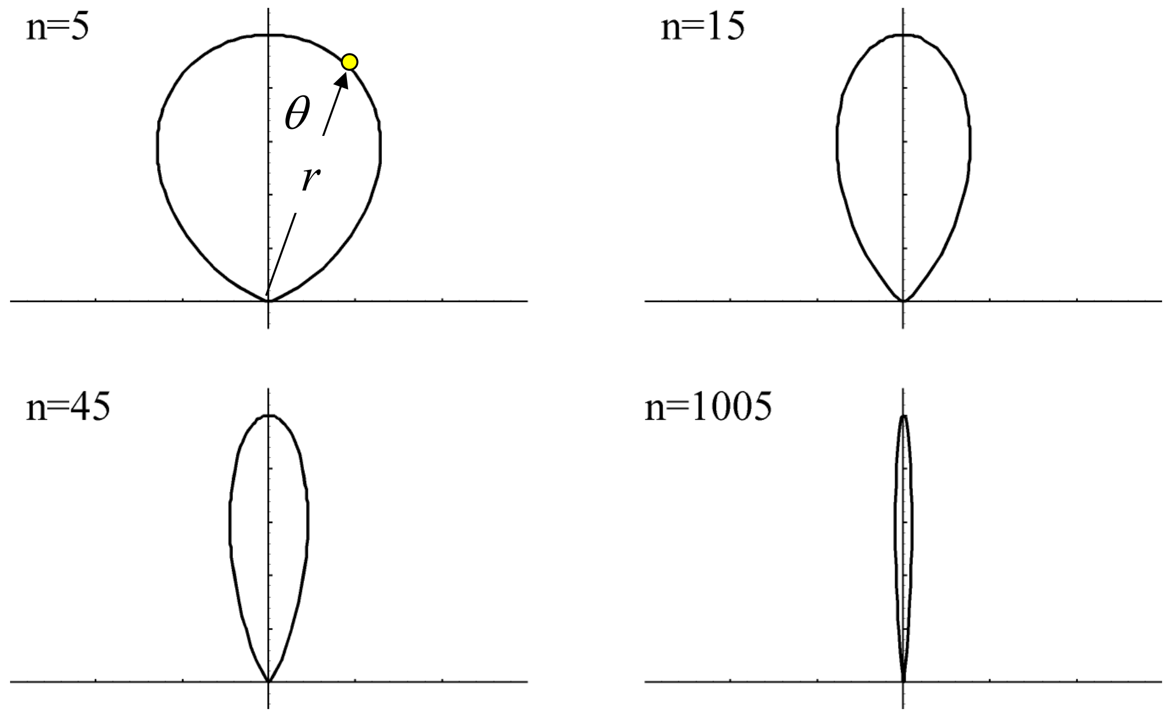
Phong Illumination Model

- To simulate reflection we should examine surfaces in the reflected direction to determine incoming flux
  $\Rightarrow$*global illumination*
- A local illumination approximation considers only reflections of light sources.
- The Phong model is an *empirical* (based on observation) local model of shiny surfaces (tend to appear like plastic).
- It is assumed that the BRDF of such surfaces may be approximated by a *spherical cosine function* raised to a power (known as the *Phong exponent*).

The Cos^n Function

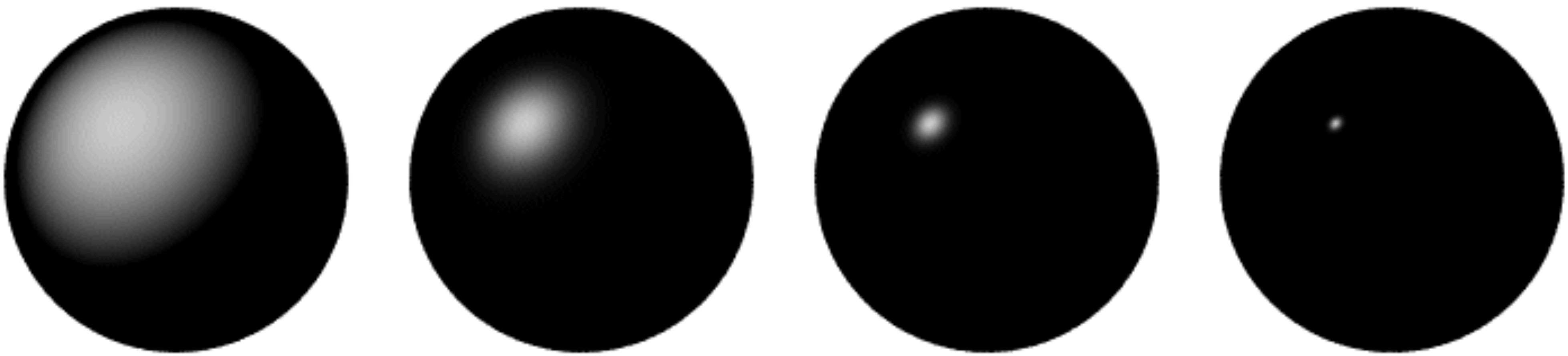
The cosine function (defined on the sphere) gives us a lobe shape which approximates the distribution of energy about a reflected directio controlled by the shininess parameter ***n*** known as the *Phong exponent*

$$r = \cos^n \theta$$



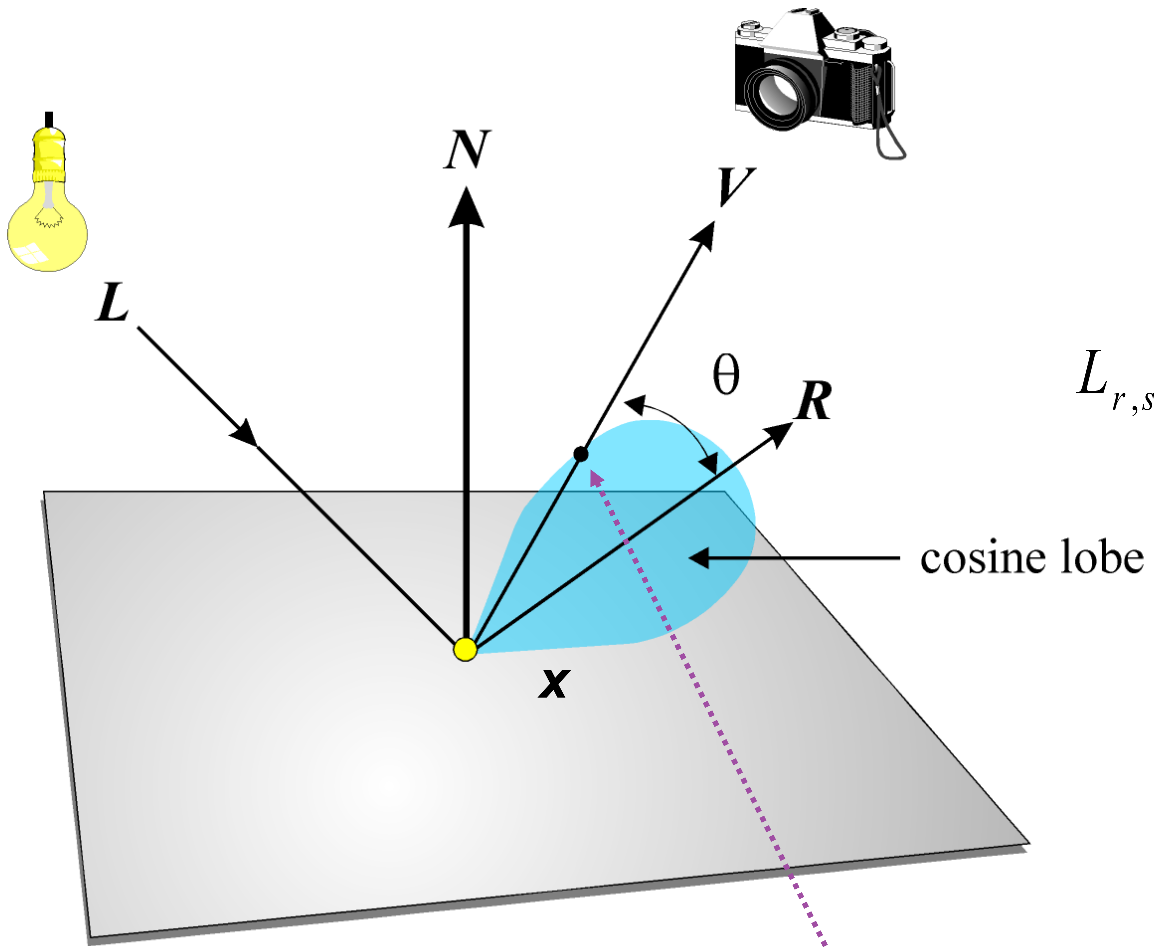
In the limit ($n \to \infty$) the function becomes a single spike (i.e. ideal specular).

The Cos^n Function



Low n --→ High n

The Phong Model



specular reflectivity

light source irradiance

$$L_{r,s}(x,V) = \boxed{\frac{n+2}{2\pi}}\, \rho_s\, \boxed{\cos^n\theta}\, \frac{\Phi_S}{4\pi d^2}$$

Normalization term

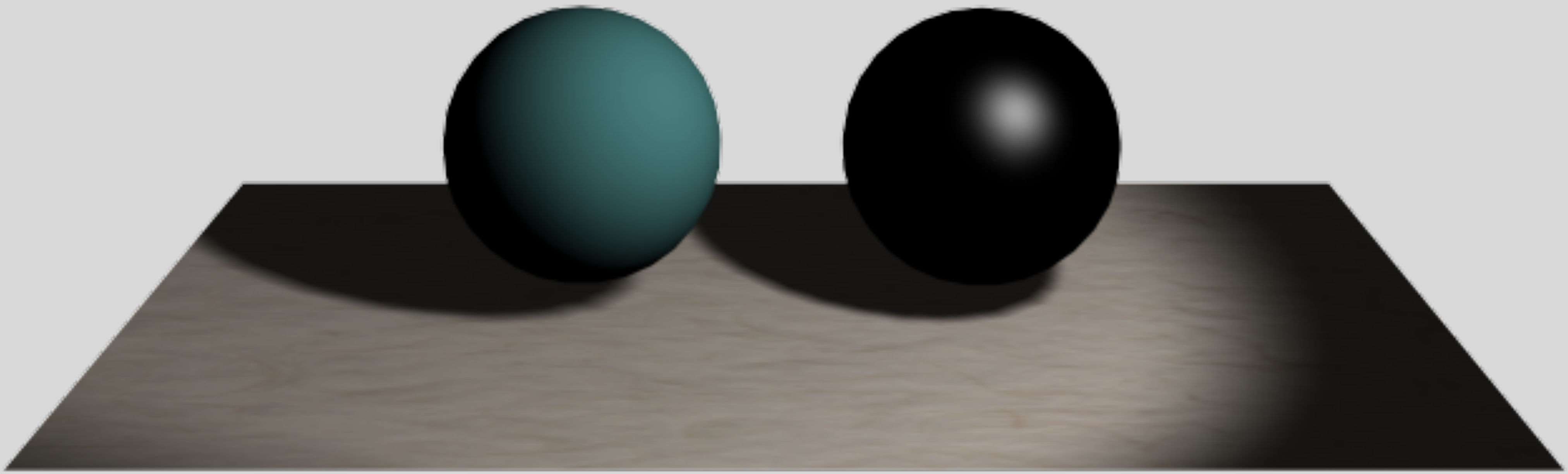
cosine lobe

Radiance of reflected light given by cosine function

Pure Lambertian vs. Phong

Lambertian Surface

Phong Illuminated Specular Surface



Lambertian vs. Phong

- Lambertian surfaces exhibit surface reflection independent of orientation and distance from viewer but not light source, leading to matte appearance.
- Specular surfaces exhibit surface reflection, dependent on orientation and distance of both viewer and light source, leading to glossy appearance with highlights.

Ambient Illumination

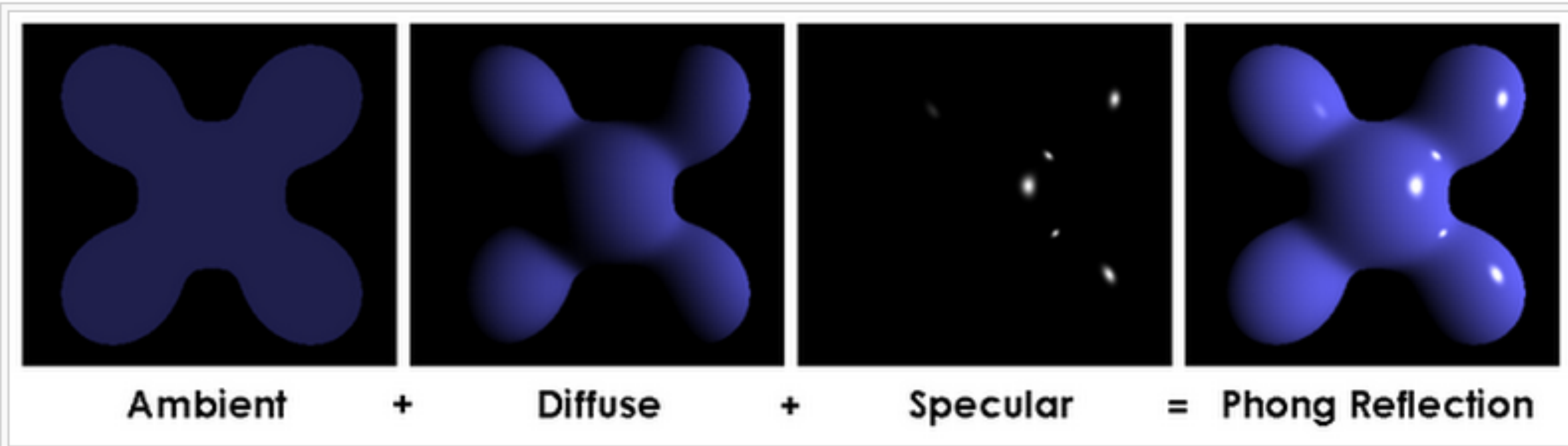
- Local illumination models account for light scattered from the light sources only.
- Light may be scattered from all surfaces in the scene
  - we are missing some light; in fact we are missing a lot of light, typically over 50%.
- *Ambient term* = a coarse approximation to this missing flux
- Ambient term defined by the ambient coefficient ρ_a:

$$I_a = \rho_a I_s$$

- This is a <u>constant</u> everywhere in the scene.
- The ambient term is sometimes estimated from the total powers and geometries of the light sources.

Putting it all together...

- The complete Phong Illumination Model includes
  - Lambertian model for diffuse reflection
  - Cosine lobe for specular reflection
  - Ambient term to approximate all other light



Putting it all together...

- An object must therefore have material data associated with it to define how diffuse, specular (and shiny) or ambient it is:

$$\text{Surface Data} = \begin{cases} \rho_{\mathrm{a}} = \text{ambient reflectance} \\ \rho_{\mathrm{d}} = \text{diffuse reflectance} \\ \rho_{\mathrm{s}} = \text{specular reflectance} \\ n = \text{phong exponent} \end{cases}$$

Point Light Sources

- The irradiance E of a surface due to a point source obeys the inverse square law:

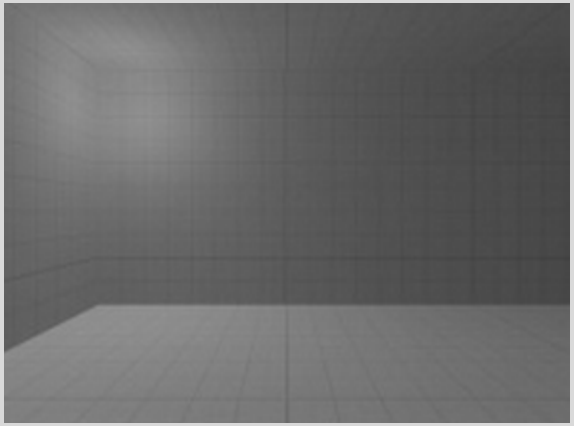
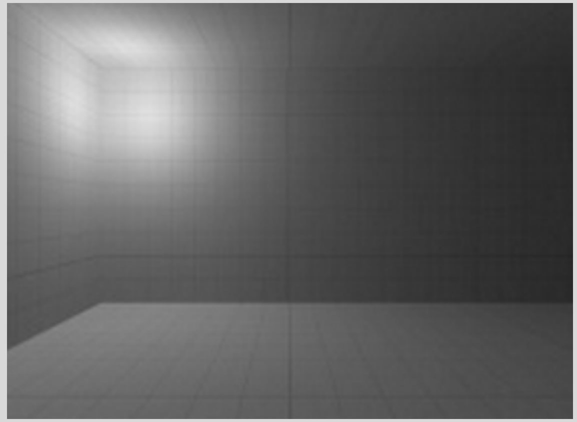
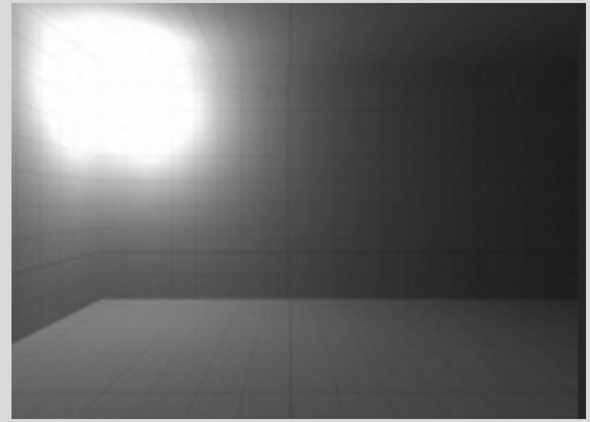
$$E = \frac{\Phi_S}{4\pi d^2} = \frac{I_S}{d^2}$$

- However, this often makes it difficult to control the lighting in the scene, so we employ a less accurate but more flexible model of irradiance:

$$E = \frac{I_S}{a + bd + cd^2}$$

  - *a* = constant attenuation factor
  - *b* = linear attenuation factor
  - *c* = quadratic attenuation factor

Point Light Sources

- Constant Linear Quadratic

- The 100% constant attenuation will result in a light that has no attenuation at all.
- Linear light will diminish as it travels from its source.
- Quadratic: the further light travels from its source the more it will be diminished – sharp drop in light

Phong Illumination Model

$$L_r(x,V) = L_{r,a}(x) + L_{r,d}(x) + L_{r,s}(x,V)$$ = (ambient + diffuse + specular)

Phong Illumination Model

$$L_r(x,V) = L_{r,a}(x) + L_{r,d}(x) + L_{r,s}(x,V) \quad = (\text{ambient} + \text{diffuse} + \text{specular})$$

$$= E\rho_a + E\rho_d\,(N \cdot L) + E\rho_s (V \cdot R)^n$$

$cos\Theta$ $cos^n\Theta$

Note:

-ambient term not affected by light L or viewing angle V

-diffuse term affected by light but not by viewing angle

-specular term affected by viewing angle but not light

Phong Illumination Model

$$L_r(x,V) = L_{r,a}(x) + L_{r,d}(x) + L_{r,s}(x,V) \quad = (\text{ambient} + \text{diffuse} + \text{specular})$$

$$= E\rho_a + E\rho_d\,(N \cdot L) + E\rho_s (V \cdot R)^n$$

$$= \frac{\Phi_s}{4\pi}\frac{1}{a + bd + cd^2}\left[\rho_a + \rho_d\,(N \cdot L) + \rho_s (V \cdot R)^n\right]$$

Phong Illumination Model

$$L_r(x,V) = L_{r,a}(x) + L_{r,d}(x) + L_{r,s}(x,V)$$ = (ambient + diffuse + specular)

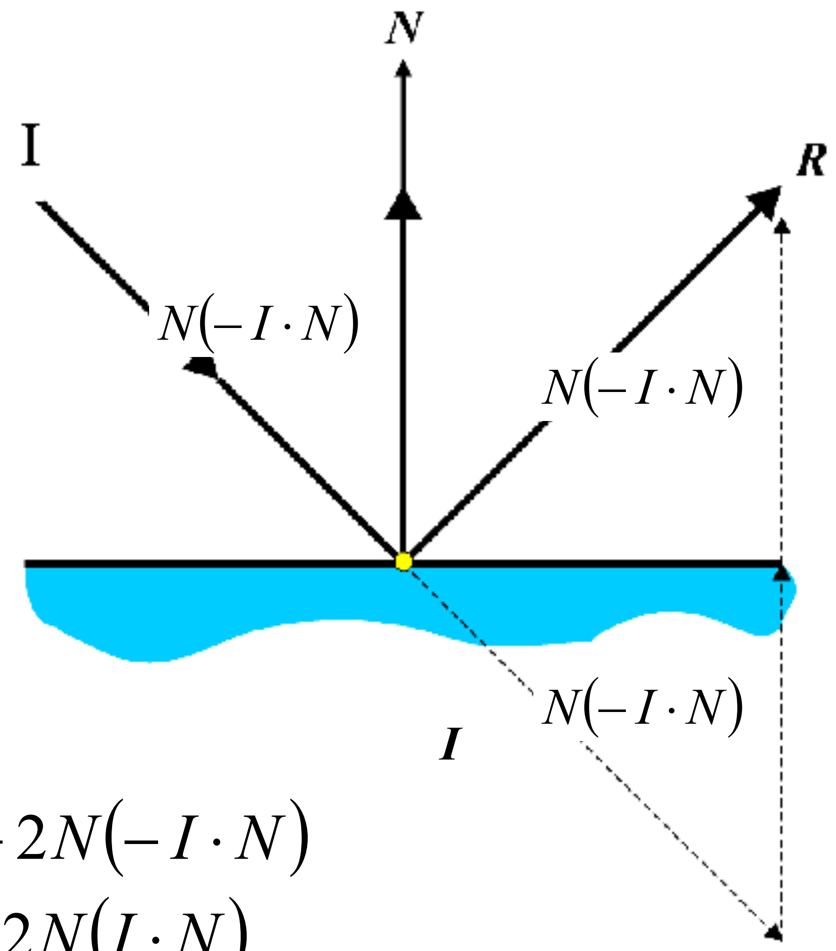
$$= E\rho_a + E\rho_d\,(N \cdot L) + E\rho_s (V \cdot R)^n$$

$$= \frac{\Phi_s}{4\pi}\frac{1}{a+bd+cd^2}\left[\rho_a + \rho_d\,(N \cdot L) + \rho_s (V \cdot R)^n\right]$$

For multiple light sources:

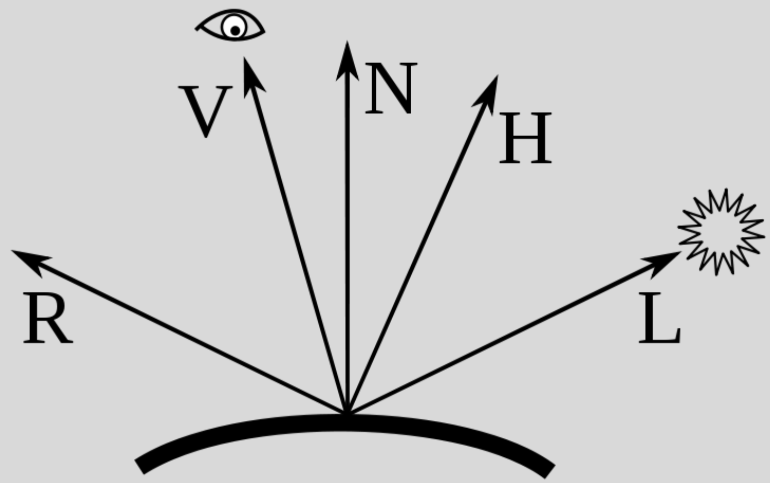
$$L_r(x,V) = \sum_{i=1}^{N}\left[\frac{\Phi_i}{4\pi}\frac{1}{a+bd_i+cd_i^2}\left[\rho_a + \rho_d\,(N \cdot L_i) + \rho_s (V \cdot R_i)^n\right]\right]$$

Determining the Reflected Vector



$$
\begin{aligned}
R &= I + 2N(-I \cdot N) \\
&= I - 2N(I \cdot N)
\end{aligned}
$$

Blinn-Phong



- Problem with computing the perfect reflection direction, R:
  - Normal N is different for every point on the surface, so must recompute *R* for every polygon – this is slow
- Blinn proposed an alternative form:
  - use the *half vector H* in place of *R*:

$$H = \frac{V + L}{|V + L|}$$

- The specular term is

$$(N \cdot H)^n$$

Half vector H is a vector with a direction half-way between the eye vector V and the light vector L

Blinn-Phong vs. Phong

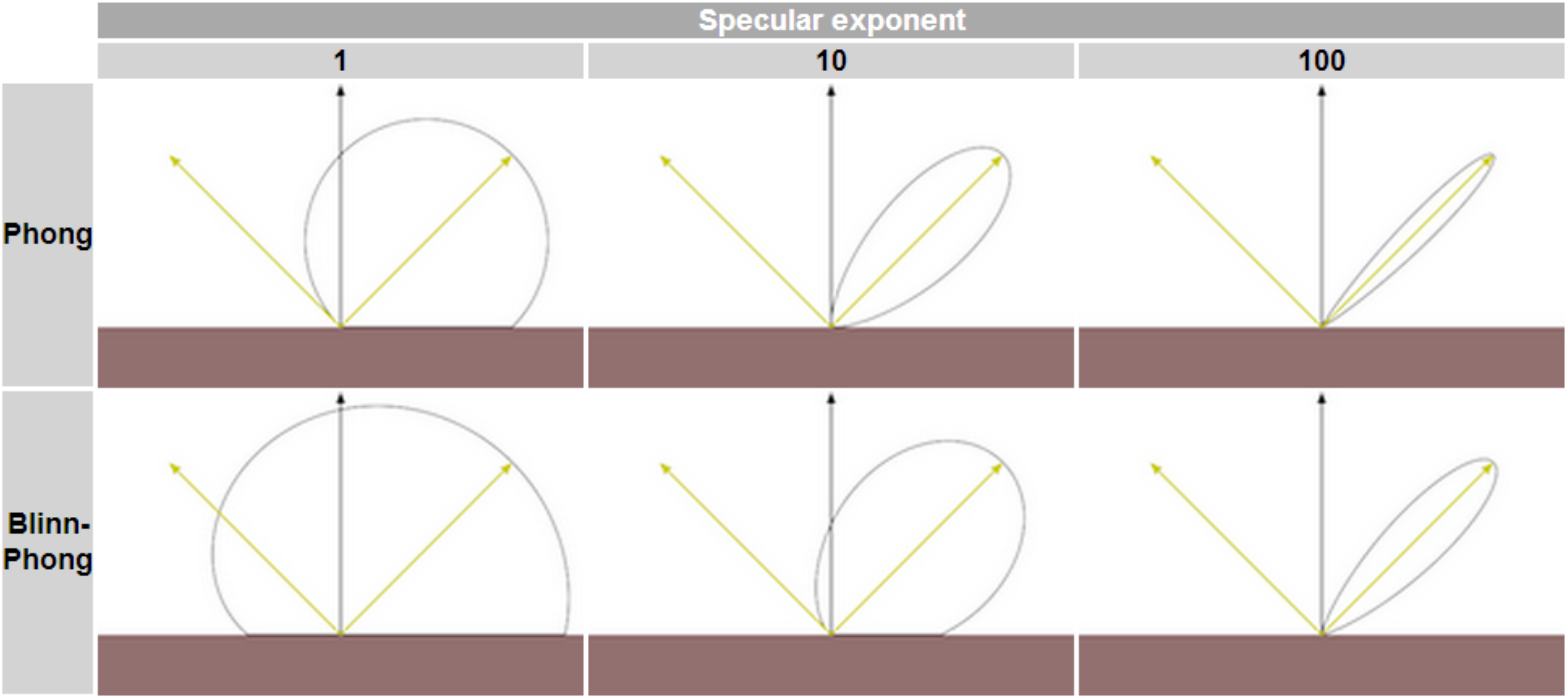
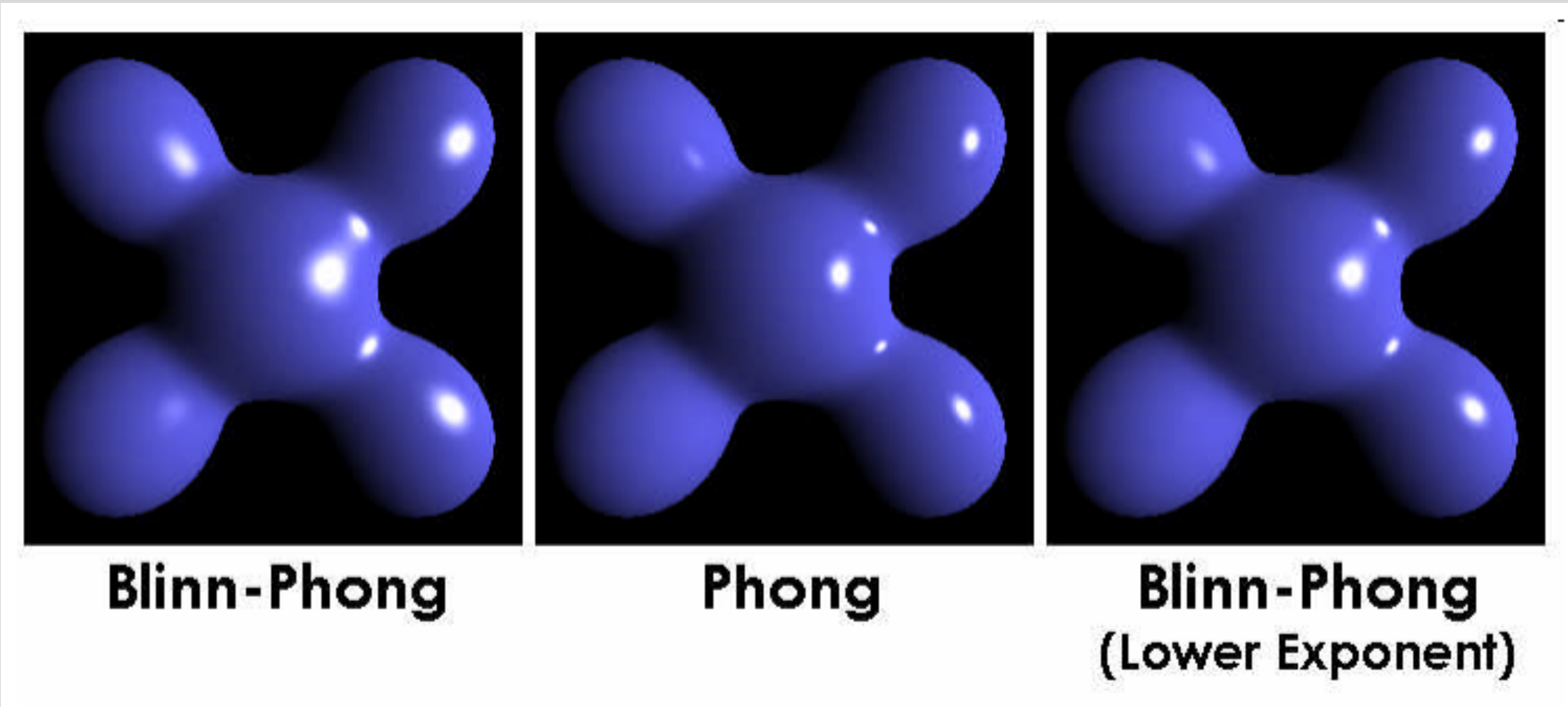


Figure 5: Phong vs Blinn-Phong With Varying Shininess Values.
Both the Phong and Blinn-Phong reflectance functions cause a highlight to appear around the direction of reflection. Blinn-Phong is cheaper to calculate, but appears more spread out at the same shininess.

Approximating Phong

- Change the exponent so that the Blinn-Phong model matches Phong



Incorporating Colour

- In fact, many of the systems that implement the Phong model differ only in the manner in which colour is handled.
- Typically we handle diffuse and specular illumination separately:
  - **diffusely** reflected light results from the reflection via multiple scattering events in the micro-scale geometry ⇒ *reflected light is coloured by selective absorption by the surface* i.e. a green surface absorbs all wavelengths except green
  - **specularly** reflected light interacts once with the surface and is thus *not coloured by the surface* i.e. the reflection of a light source takes on the colour of the source

Incorporating Colour

- Therefore the Phong model becomes:

$$L_r(x,V) = L_{r,a}(x) + L_{r,d}(x) + L_{r,s}(x,V)$$

$$= EC_{ambient}\rho_a + EC_{surface}\,\rho_d\,(N \cdot L) + EC_{light}\,\rho_s(V \cdot R)^n$$

- For flexibility, the ambient, diffuse and specular reflections are scaled independently by colour vectors:

$$L_r(x,V) = EC_{amb}\rho_a + EC_{diff}\,\rho_d\,(N \cdot L) + EC_{spec}C_{light}\,\rho_s(V \cdot R)^n$$

- The model is applied using colour vectors yielding the final colour vector to assign to a pixel.
- Note: only the specular term is scaled by the light's colour.