Shaders

Recall: Lighting Equation

- Multiplying the BRDF by an incoming irradiance gives the outgoing radiance

$$dL_{o\ due\ to\ i}(\omega_i, \omega_o) = BRDF(\omega_i, \omega_o) dE_i(\omega_i)$$

- For even more realistic lighting, we'll bounce light all around the scene
- It's tedious to convert between E and L, so use $dE = Ld\omega \cos\theta$ to obtain:

$$dL_{o\ due\ to\ i}(\omega_i, \omega_o) = BRDF(\omega_i, \omega_o) L_i d\omega_i \cos\theta_i$$

- Then,

$$L_o(\omega_o) = \int_{i \in in} BRDF(\omega_i, \omega_o) L_i \cos\theta_i \, d\omega_i$$

Recall: Area Lights

- Light power is emitted per unit area (not from a single point)
- The emitted light goes in various directions (measured with solid angles)

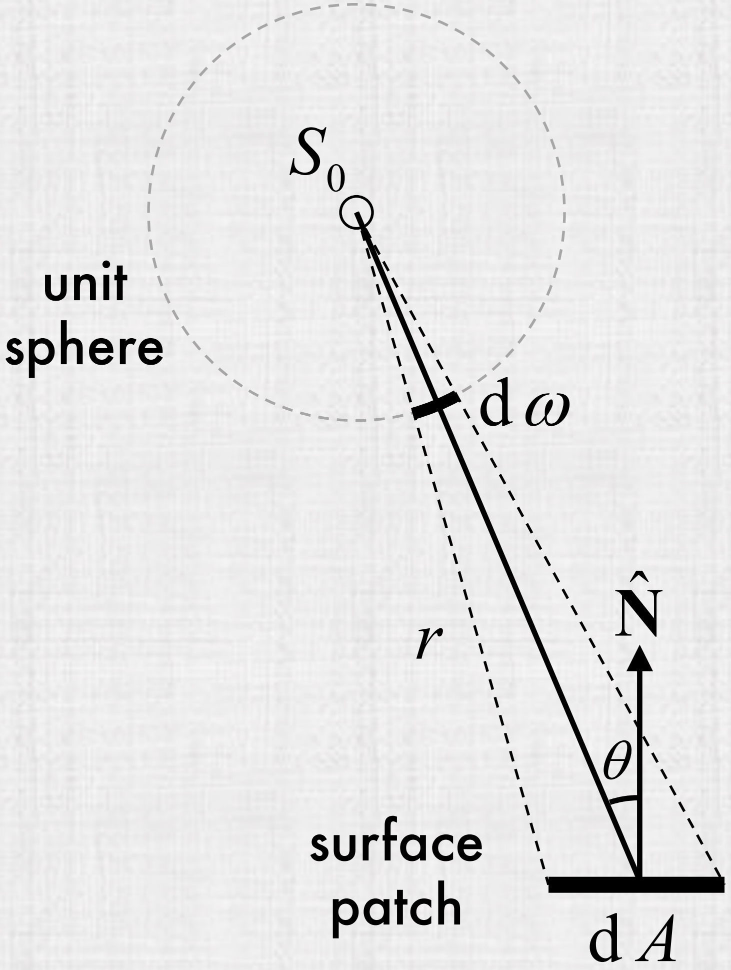
- Break an area light up into (infinitesimally) small area chunks
- Each area chunk emits light into each of the solid angle directions
  - i.e. radiant intensity per area chunk
- Each emitted direction also has a cosine term (similar to irradiance)

- Radiance – radiant intensity per area chunk

$$L = \frac{dI}{dA \cos\theta_{light}} \left(= \frac{d^2\Phi}{d\omega\, dA \cos\theta_{light}} = \frac{dE}{d\omega \cos\theta_{light}} \right)$$

Recall: Solid Angle vs. Cross-Sectional Area

- The (orthogonal) cross-sectional area is $dA\ cos\theta$
- So, $d\omega = \frac{dA_{sphere}}{r^2} = \frac{dA\ cos\theta}{r^2}$ (solid angle varies with tilting θ and distance r)



Point Lights

- Assume incoming light only comes from a single point light source (with direction ω_{light})
- Then the BRDF and the cosine terms are approximately constant:

$$L_o(\omega_o) = BRDF(\omega_{light}, \omega_o) \cos\theta_{light} \int_{i \in in} L_i d\omega_i$$

- Since $L = \frac{dI}{dA\cos\theta}$ and $d\omega = \frac{dA\,cos\theta}{r^2}$, the integral becomes $\int_{i \in in} \frac{dI}{r^2} = \frac{I}{r^2}$
- If objects are approximately equidistant from the light (e.g. the sun), then r is approximately constant and can be folded into I_{light} to get $\tilde{I}_{light}$:

$$L_o(\omega_o) = BRDF(\omega_{light}, \omega_o) \cos\theta_{light} \tilde{I}_{light}$$

- For each channel (R,G,B), sum over all the point lights:

$$L_o(\omega_o) = \sum_{j=1}^{\#lights} BRDF(\omega_j, \omega_o) \cos\theta_j \tilde{I}_j$$

Point Light Drawbacks

- All the lighting from other objects in the scene is turned off
- Thus, the scene is overall darker, there is no color bleeding, etc.

- Surfaces occluded from all point light sources are completely black

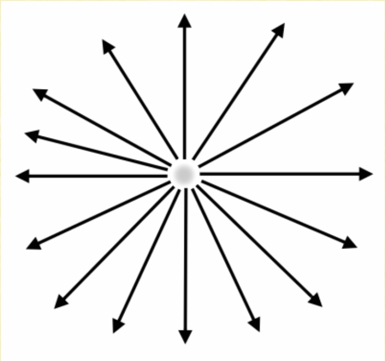
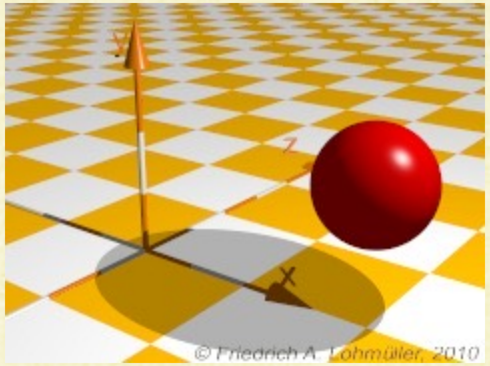
- Shadows have harsh boundaries

- Objects closer to a light source are not brighter than those father away (variance with radius has been removed)

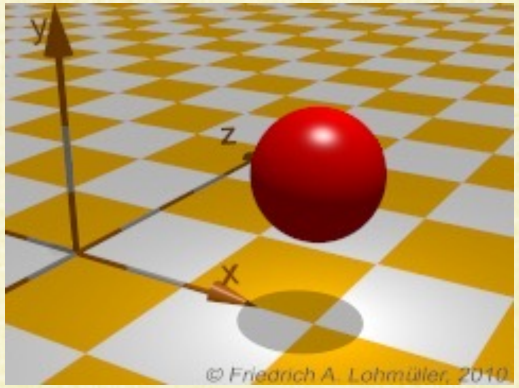
- Etc.

Point Light Examples

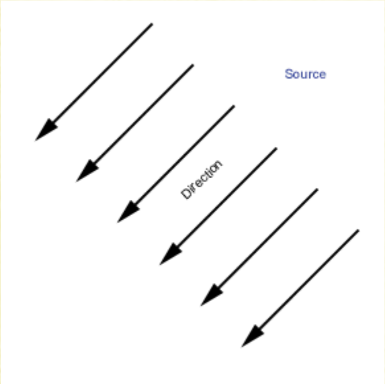
Point Light
• Light emitted from a single point in space, outwards in every direction

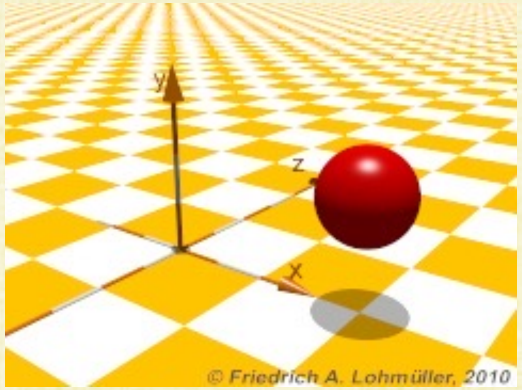



Spotlight
• Angular subset of a point light
• Prune directions a cutoff angle away from a central direction (use a dot product)

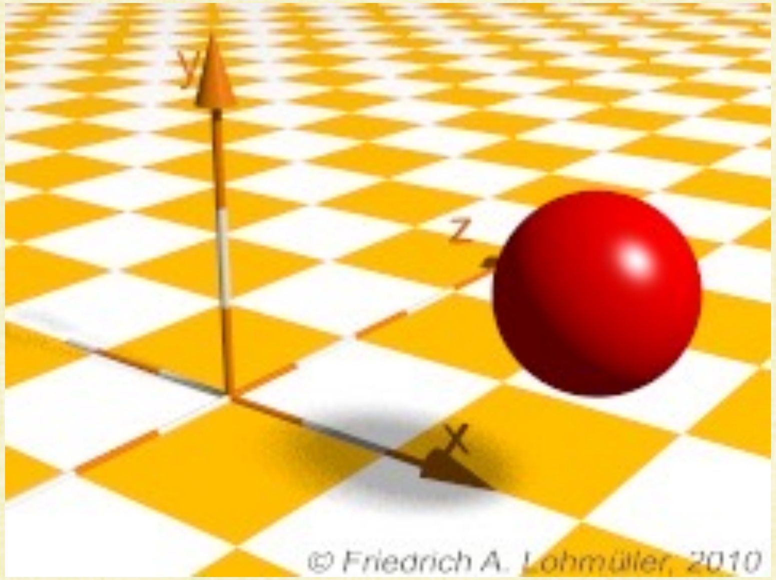
Directional Light
- Always use the the same incoming ray direction
- Models a far away point light (like the sun)

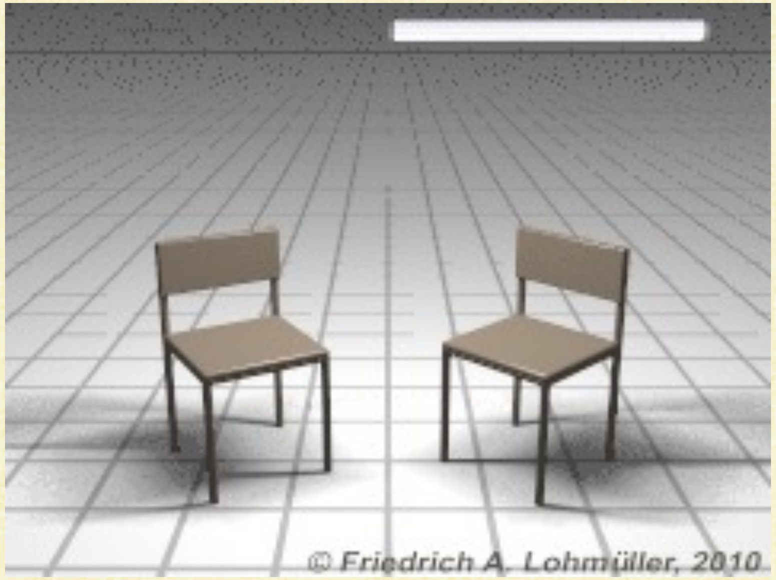


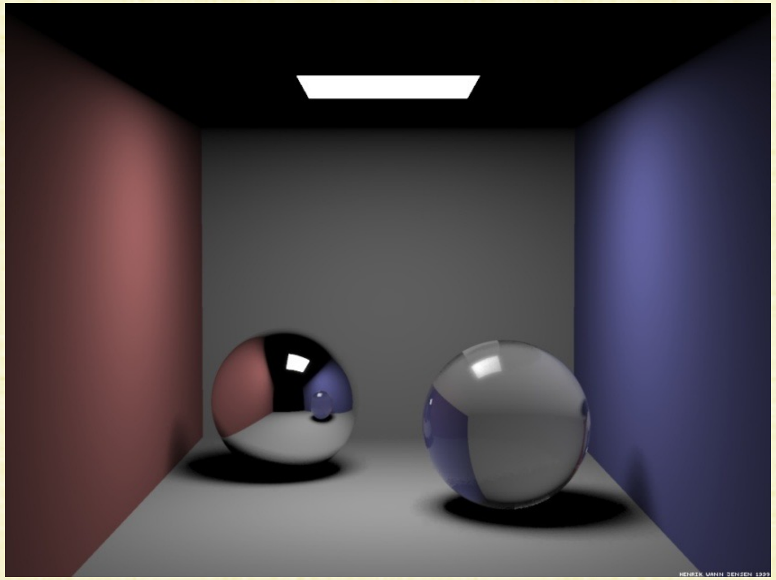


Area Lights (approximated by point lights)

- Light is emitted from a surface (objects behind the surface are not illuminated)
- Can approximate by distributing a (large) number of point lights across the surface
- The sum of the strengths of all the point lights should equal the strength of the area light
- Creates softer shadows!





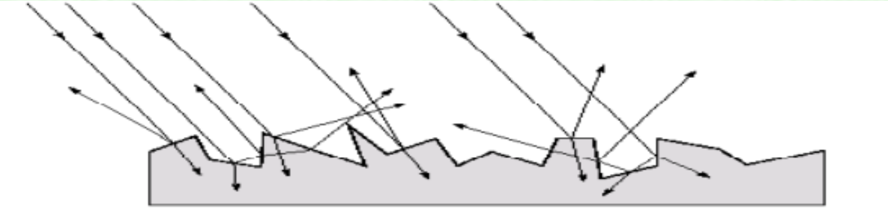


Volume Lights (approximated by point lights)

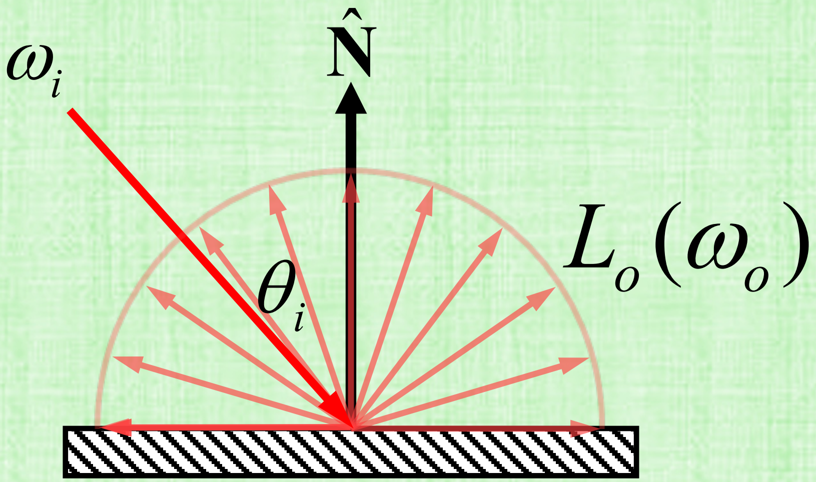
- Distribute a (large) number of point lights throughout the volume

Diffuse Materials

- Reflects light equally in all directions, independent of the incoming direction
- This can happen when a rough surface (with many tiny microfacets) reflects incoming light outwards in every possible direction:



- The BRDF doesn't depend on incoming/outgoing directions, and thus is simply a constant
- $BRDF(\omega_i, \omega_o) = k_d$ and $L_o = k_d \cos\theta_{light}\, \tilde{I}_{light}$



Diffuse Materials

- Shading depends on the position of the light source (because of the cosine term)
- Shading does not depend on the position of the viewer/camera
- Good approximation of diffuse/dull/matte surfaces (such as chalk)

- An object with (diffuse) color (k_d^R, k_d^G, k_d^B) hit by a light with color $(\tilde{I}^R, \tilde{I}^G, \tilde{I}^B)$ results in:

$$(L_o^R, L_o^G, L_o^B) = (k_d^R \tilde{I}^R, k_d^G \tilde{I}^G, k_d^B \tilde{I}^B)\max(0, -\hat{\omega}_{light} \cdot \hat{N})$$

$\cos\theta_{light}$

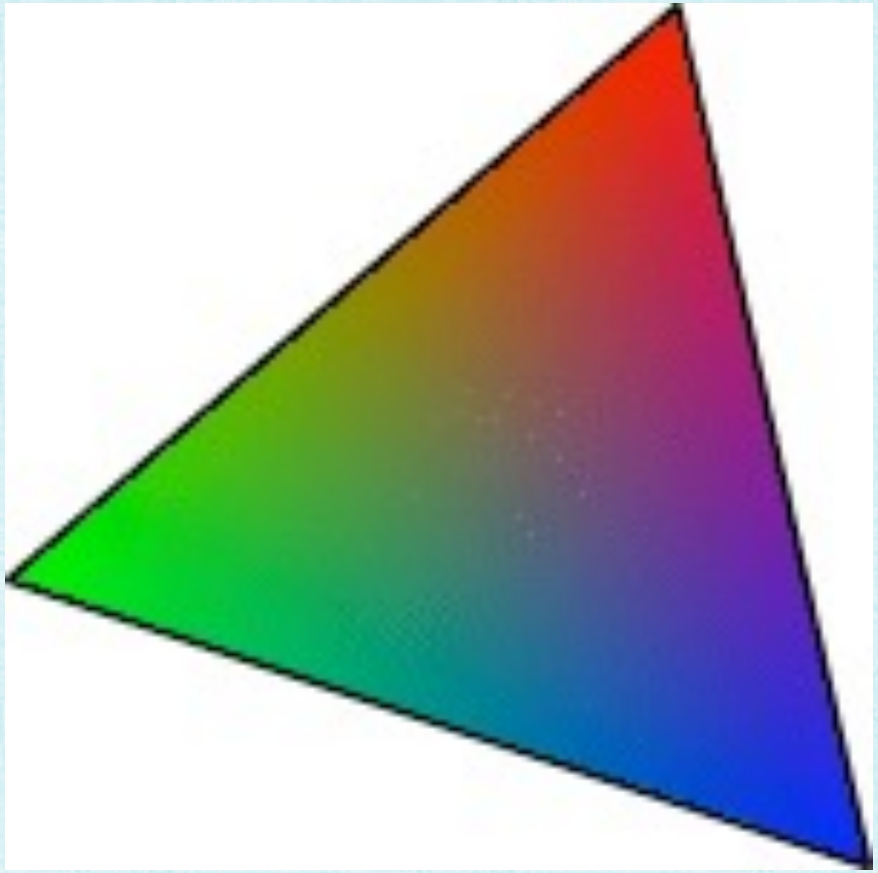
Ambient Lighting

- Useful for adding light in regions obscured from all the light sources
- Ignores the incident light direction (drops the cosine term)
- An ambient light (I_a^R, I_a^G, I_a^B) on an object with (ambient) color(k_a^R, k_a^G, k_a^B) results in:

$$(L_o^R, L_o^G, L_o^B) = (k_a^R I_a^R, k_a^G I_a^G, k_a^B I_a^B)$$

Vertex Colors

- k_a and k_d values are stored on the vertices p_0, p_1, p_2 of triangles
- Given a sub-triangle point p, compute barycentric weights: $p = \alpha_0 p_0 + \alpha_1 p_1 + \alpha_2 p_2$
- Then, compute $k = \alpha_0 k_0 + \alpha_1 k_1 + \alpha_2 k_2$ to interpolate all relevant k values (R, G, B for ambient/diffuse)

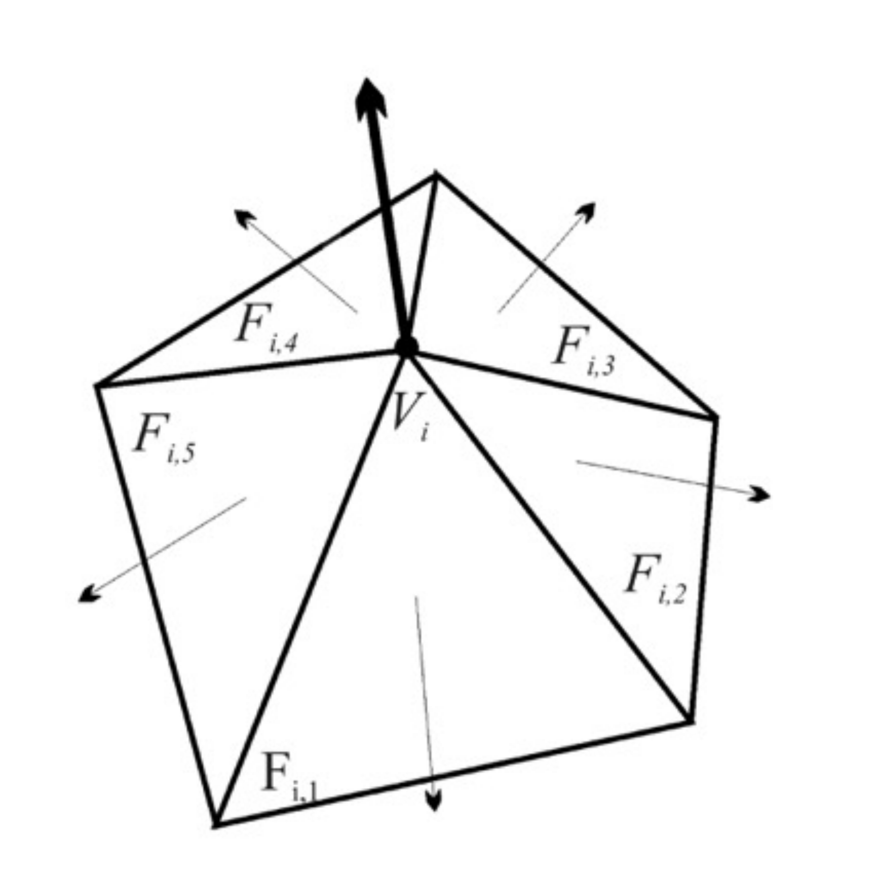


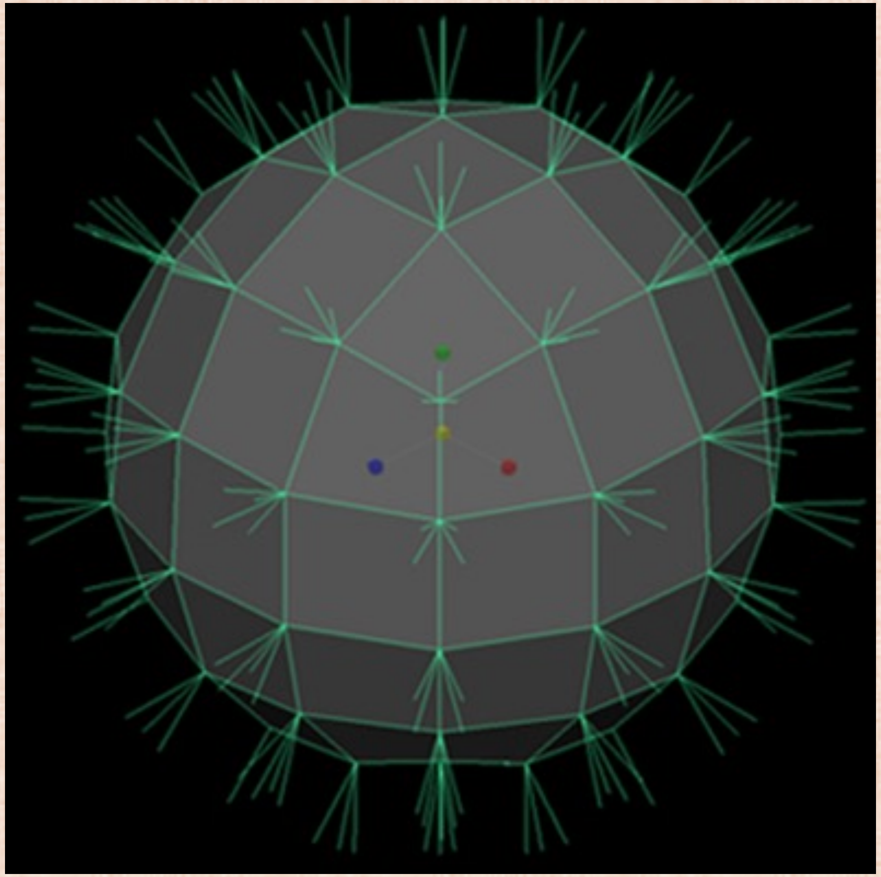
Flat Shading

- The change in normal direction from one triangle to another allows one to see individual triangles (as expected)
- This can be alleviated by using more and more triangles (but that's computationally expensive)

(Averaged) Vertex Normals

- Each vertex belongs to a number of triangles, each with their own normal
- Averaging those normals (weighted averaging, based on: area, angle, etc.) gives a unique normal for each vertex

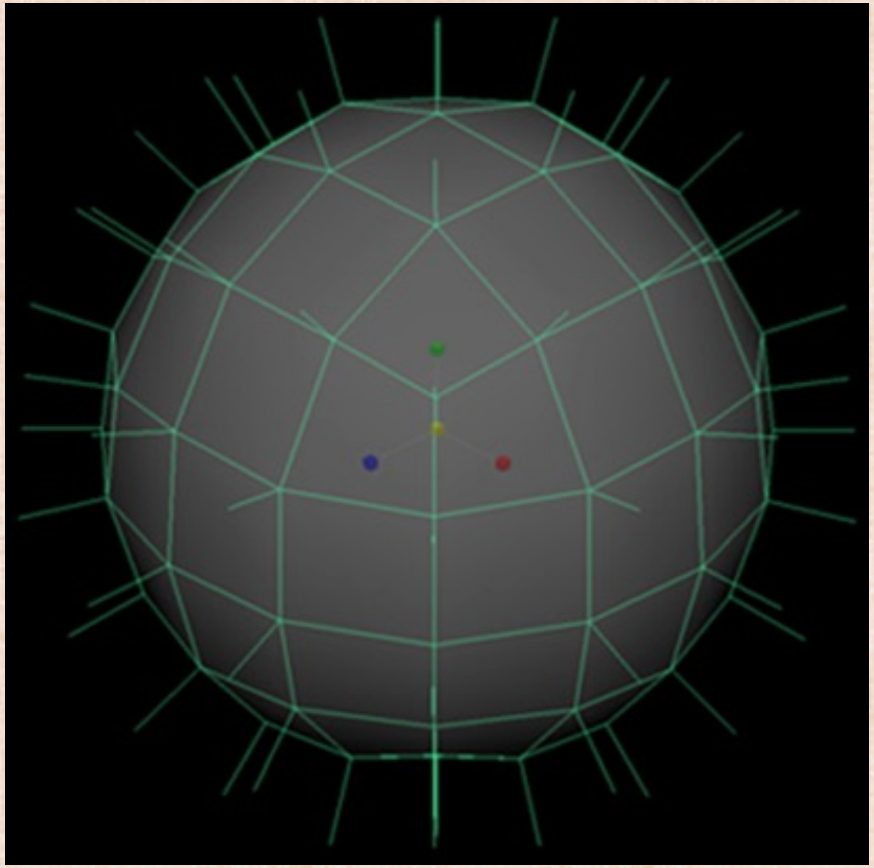




Smooth Shading

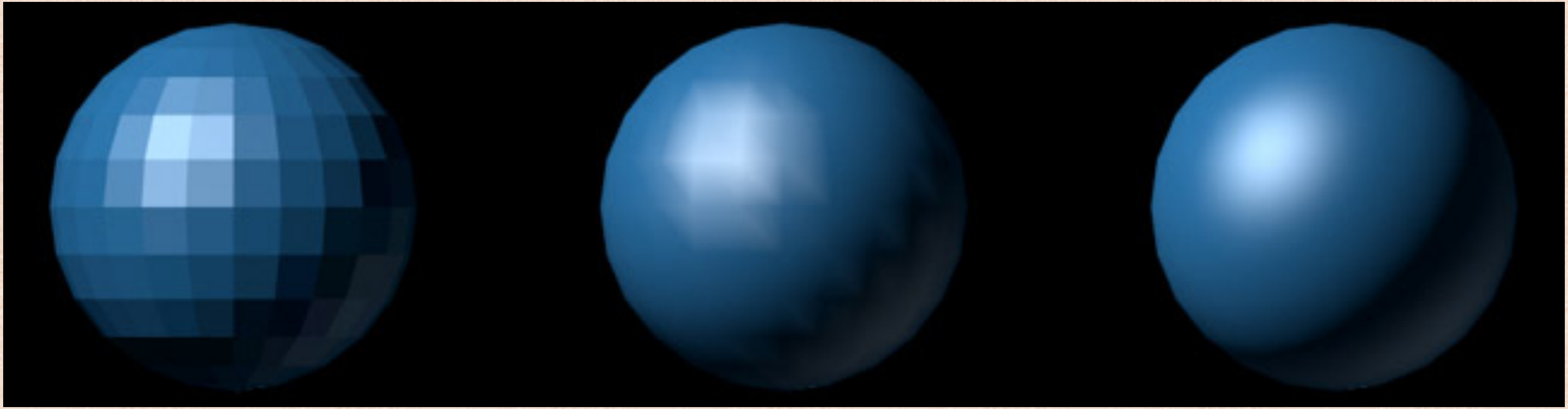
- Use barycentric weights to interpolate (averaged) vertex normals to the interior of the triangle:

$$\hat{N}_p = \frac{\alpha_0 \hat{N}_0 + \alpha_1 \hat{N}_1 + \alpha_2 \hat{N}_2}{\left\| \alpha_0 \hat{N}_0 + \alpha_1 \hat{N}_1 + \alpha_2 \hat{N}_2 \right\|_2}$$



Flat vs. Gouraud vs. Phong (Shading)

- <u>Flat</u>: use the actual normal, i.e. the real geometry (you can see the triangles)
- <u>Gouraud</u>: use (averaged) vertex normals; but, evaluate the BRDF at each vertex and interpolate the resulting colors to the triangle interior
- <u>Phong</u>: use (averaged) vertex normals, and interpolate those normals to the triangle interior (smooth shading)

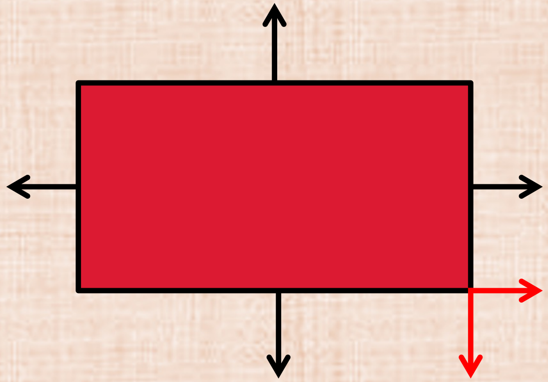


flat Gouraud Phong

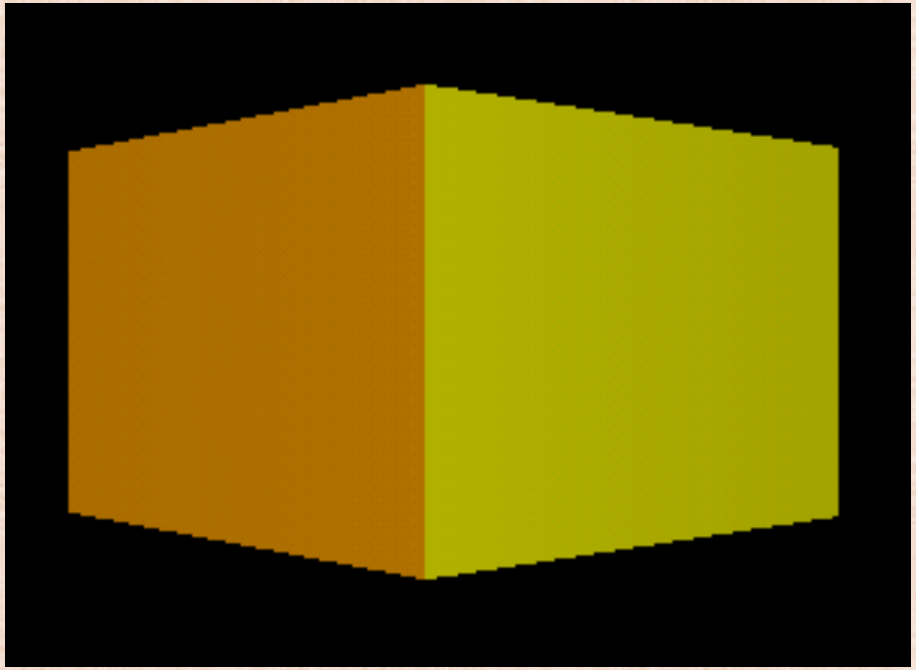
*Don't mix up Phong shading with the Phong reflection model

Edges and Corners

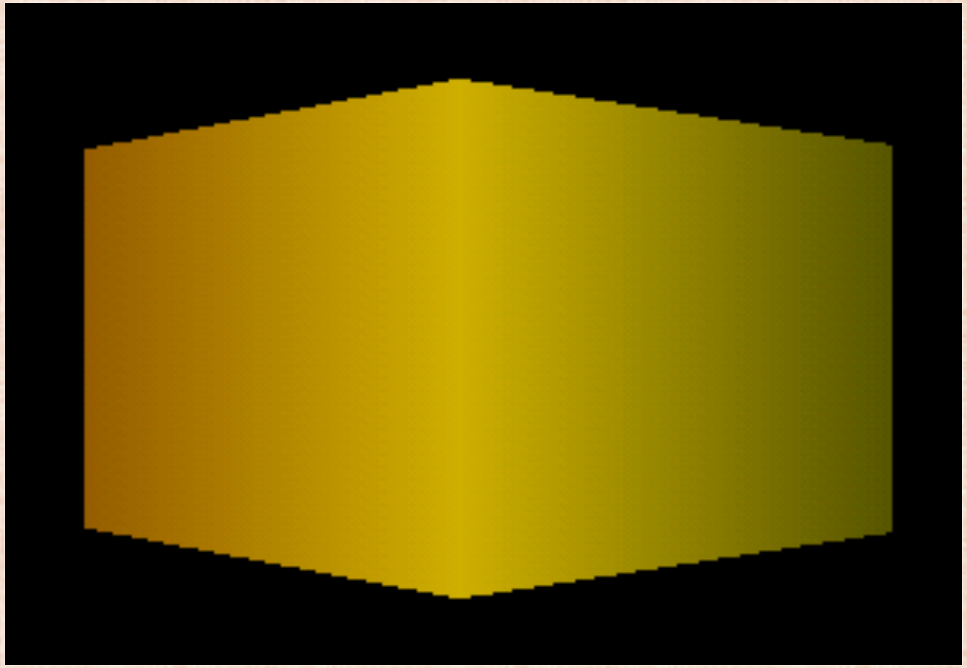
- Normals are poorly defined and difficult to compute at edges/corners
- Averaging vertex normals creates unrealistic-looking edges/corners
- Different types of shading can be used on different parts of the same object (in fact, the same triangle may require both flat and smooth shading)



What should the normal be at the corner?



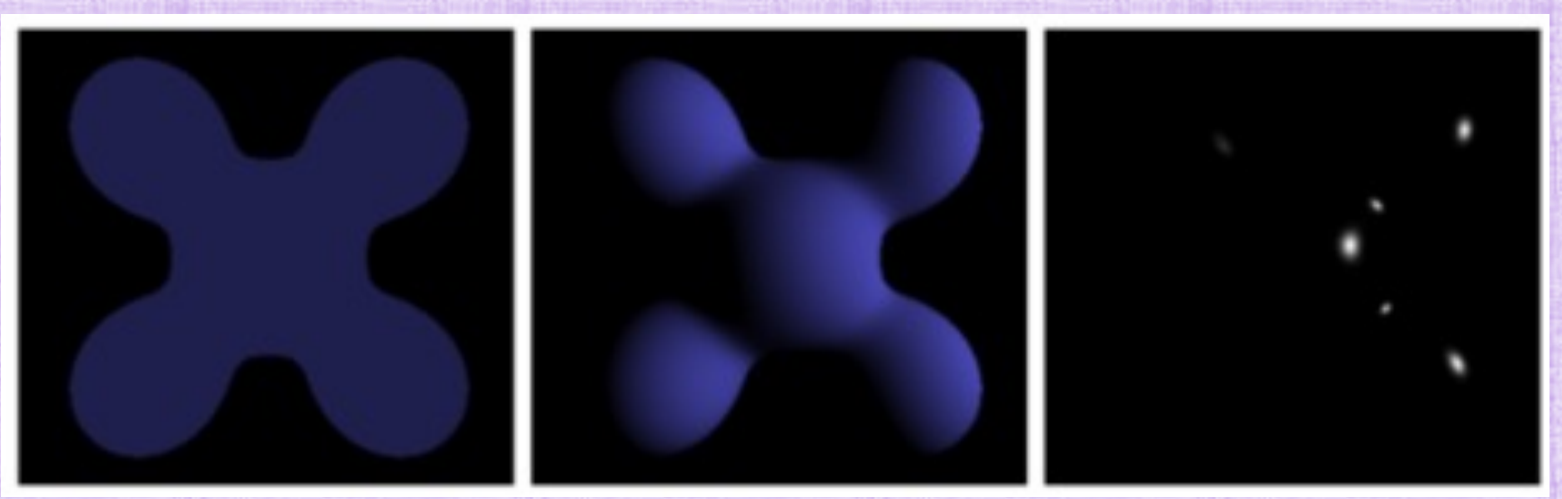
flat



smooth

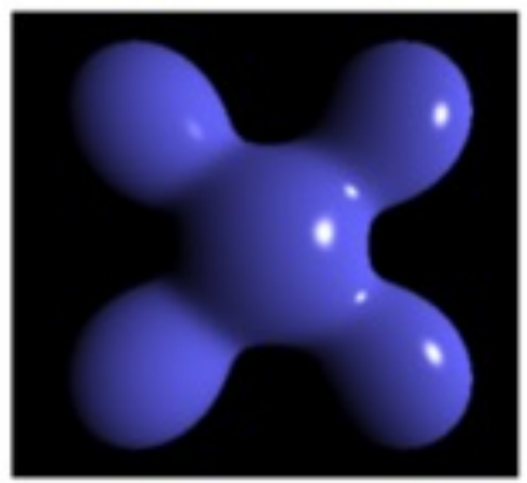
Phong Reflection Model

- Ambient, Diffuse, and Specular lighting (specular approximates glossy surfaces)



Ambient Diffuse Specular

⇒



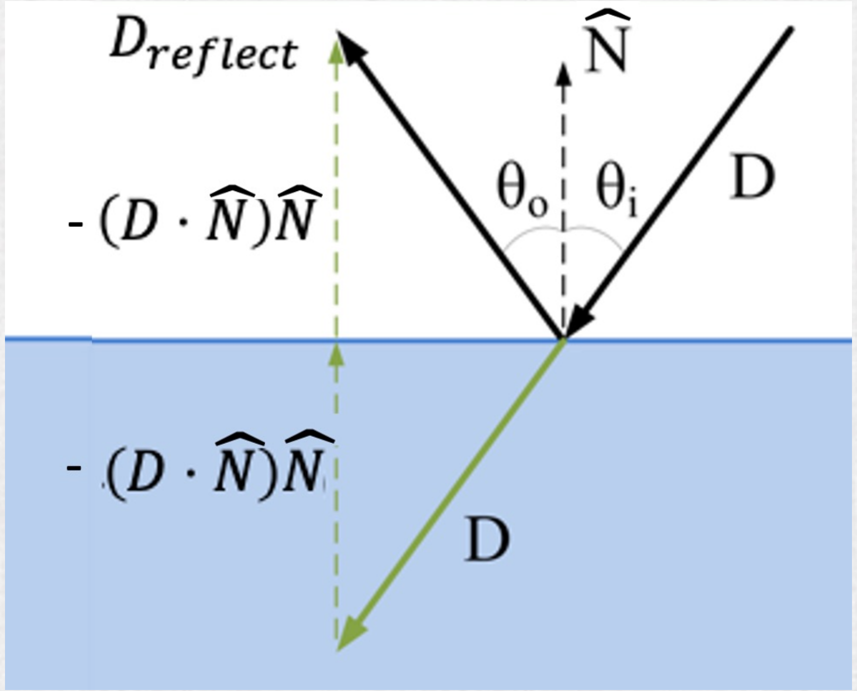
Final Image

$$L_o(\omega_o) = \sum_{j=1}^{\#lights} \underbrace{k_a I_a^j}_{\text{Ambient}} + \underbrace{k_d \tilde{I}_d^j \max\left(0, -\hat{\omega}_{light} \cdot \hat{N}\right)}_{\text{Diffuse}} + \underbrace{k_s \tilde{I}_s^j \max\left(0, \hat{\omega}_o \cdot \hat{D}_{reflect}\right)^s}_{\text{Specular}}$$

* Don't mix up Phong shading with the Phong reflection model

Recall: Reflected Ray

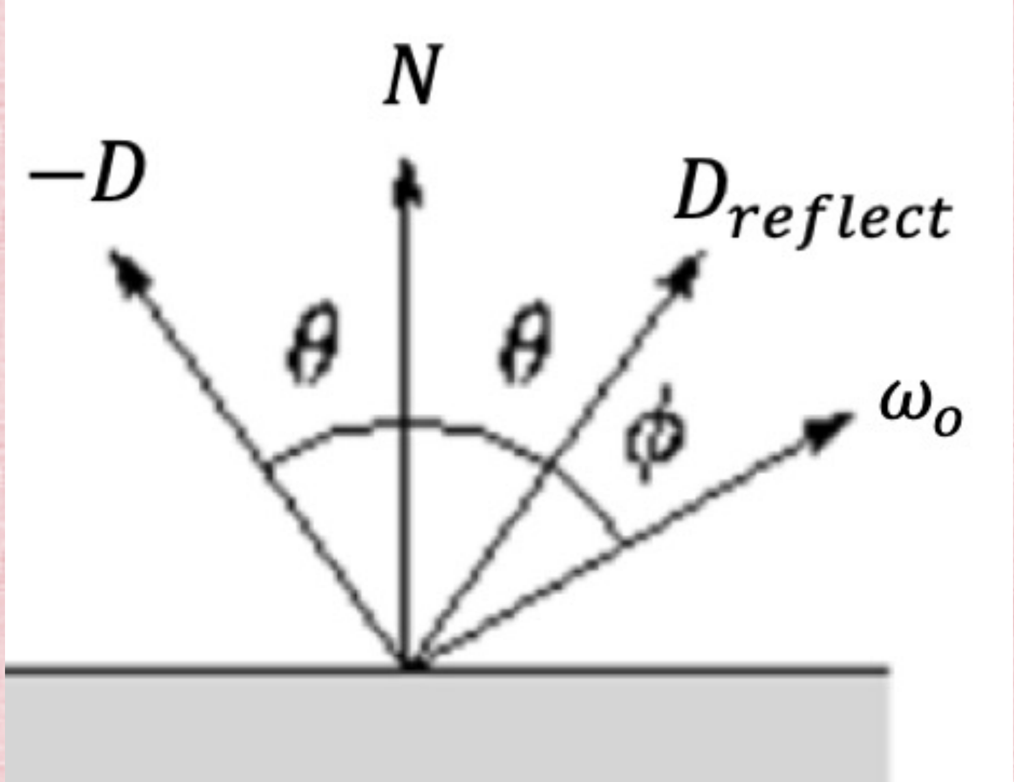
- Given an incoming ray $R(t) = A + Dt$, and (outward) unit normal $\widehat{N}$, the angle of incidence is defined via $D \cdot \widehat{N} = -\|D\|_2 \cos\theta_i$
- Mirror reflection: incoming/outgoing rays make the same angle with $\widehat{N}$, i.e. $\theta_o = \theta_i$
  - Note: all the rays and the normal are all coplanar
- Reflected ray direction: $D_{reflect} = D - 2(D \cdot \widehat{N})\widehat{N}$
- Reflected ray: $R_{reflect}(t) = R(t_{int}) + D_{reflect}t$



Specular Highlights

- For a glossy (but not completely mirror-like) surface, microscopic spatial variation (of normals) smooths reflections into a lobe
- Intensity falls off as the viewing direction differs from the mirror reflection direction:

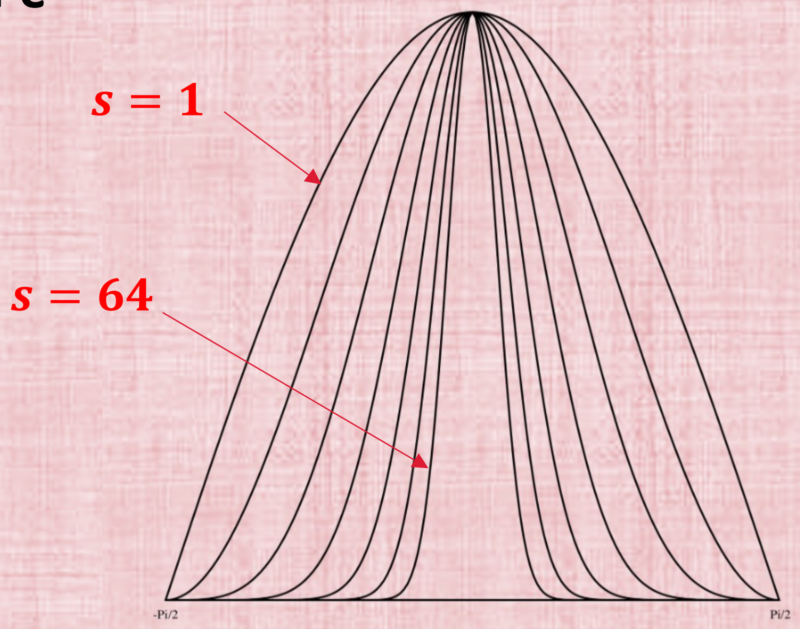
$$L_o(\omega_o) = k_s\, \tilde{I}_{\text{light}} \max\left(0, \hat{\omega}_o \cdot \hat{D}_{reflect}\right)^s$$

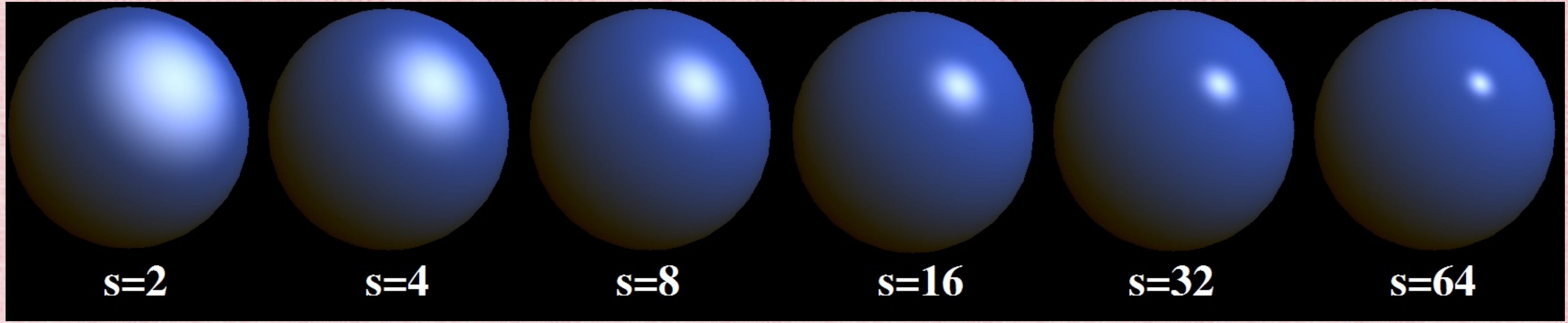


Shininess Coefficient

- A shininess coefficient s determines the size of the lobe
- A larger s gives a smaller highlight (converging to mirror reflection as $s \to \infty$)

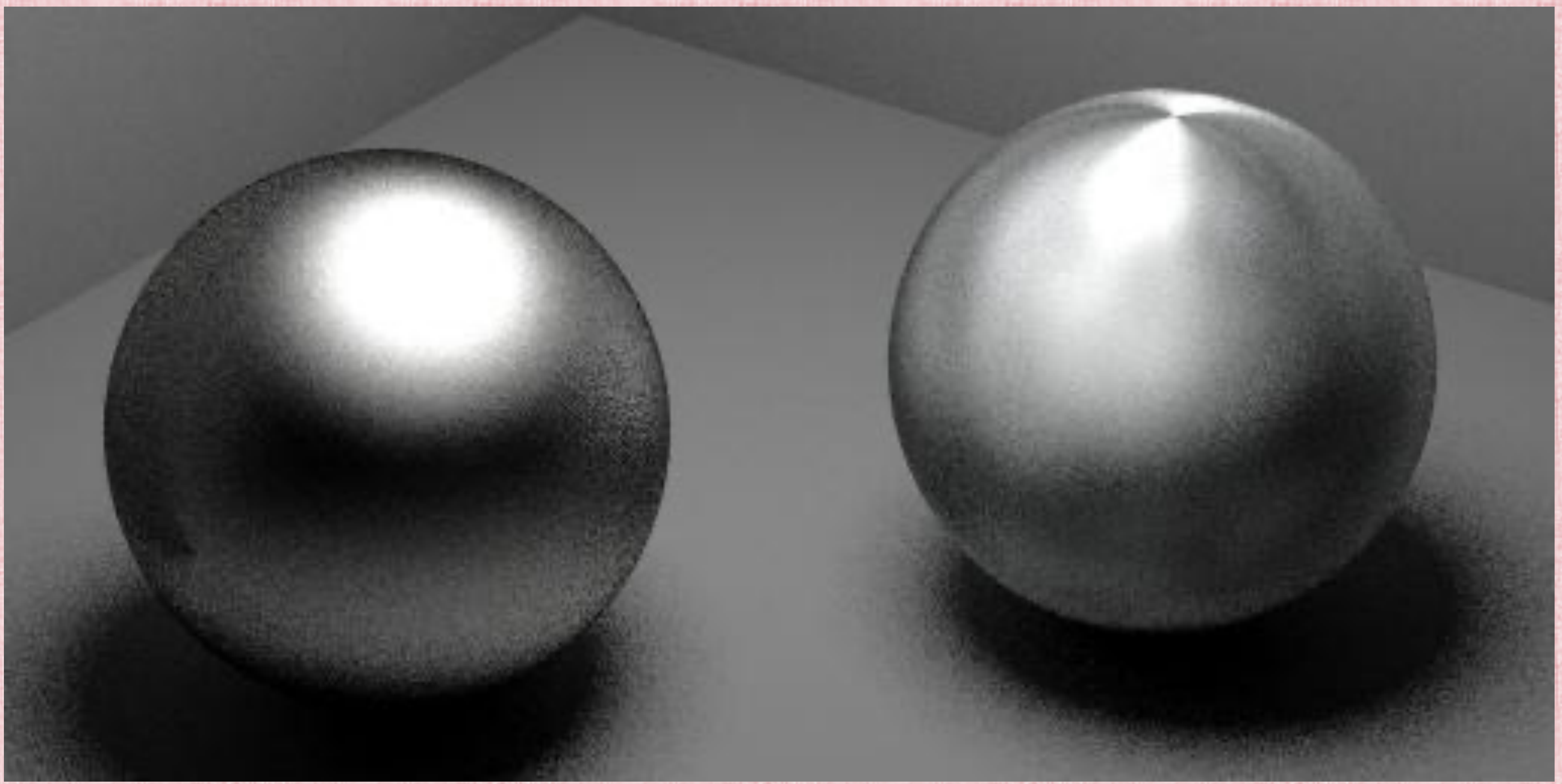
$$L_o(\omega_o) = k_s\, \tilde{I}_{\text{light}} \max\left(0, \hat{\omega}_o \cdot \hat{D}_{reflect}\right)^s$$





Anisotropic Specular Highlights

- There are various other (impressive) approximations to specular highlights as well



isotropic anisotropic