

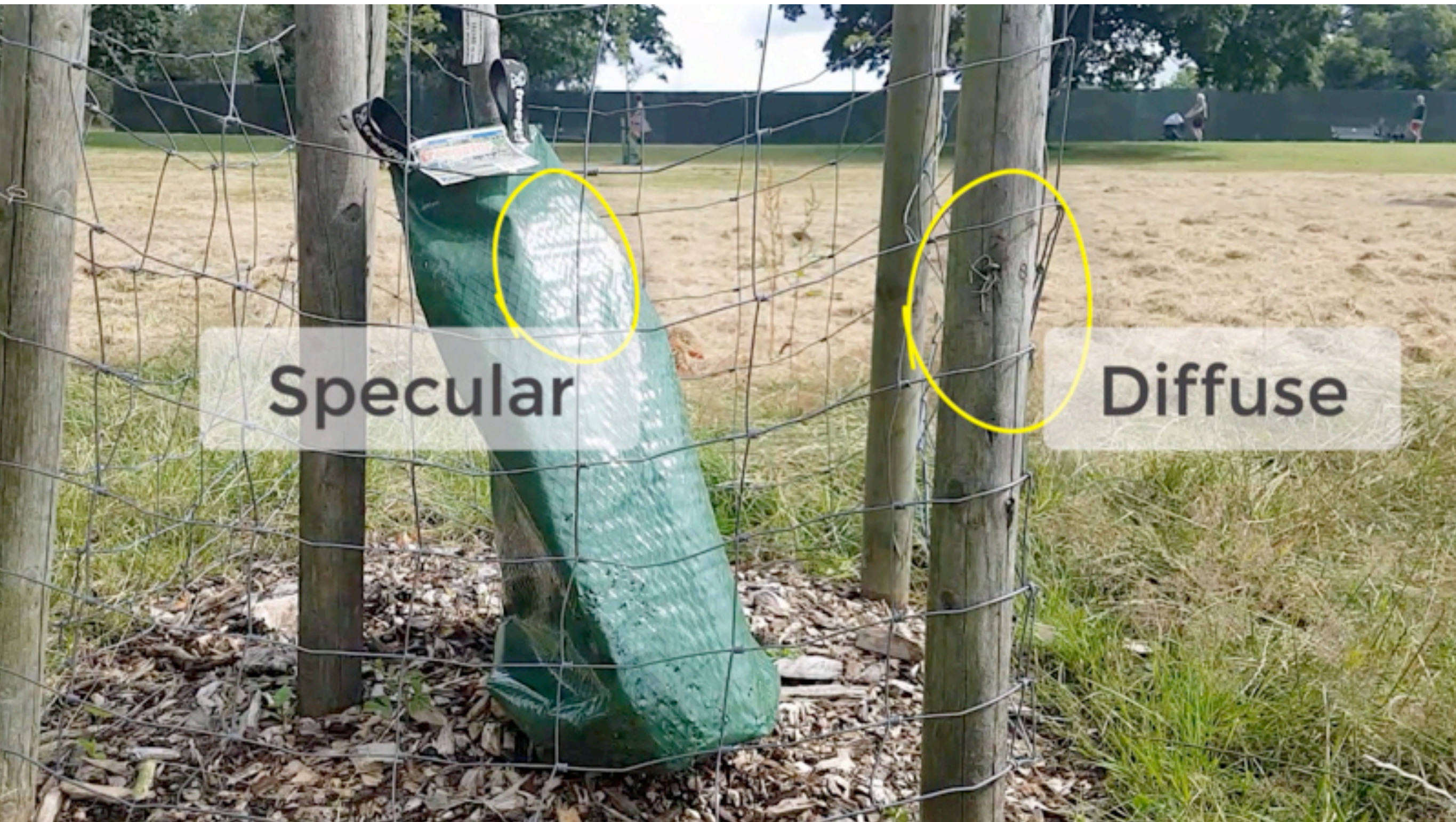
Computer Graphics

Lecture 10B
Specular Reflection

Goals

- Know what a **specular surface** looks like.
- Know how to implement the **Phong reflection model**.
- Know how to implement the **Blinn-Phong reflection model**.
- Know what to do with multiple lights.

Not all surfaces are Lambertian



Specular

Diffuse

Not all surfaces are Lambertian

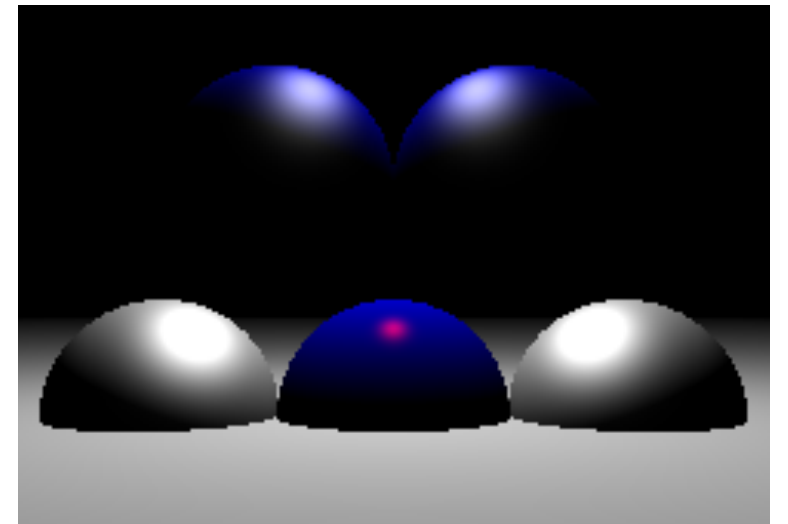


Image: <https://www.lovelifedrawing.com/artist-eye-training-highlights-reflections/>

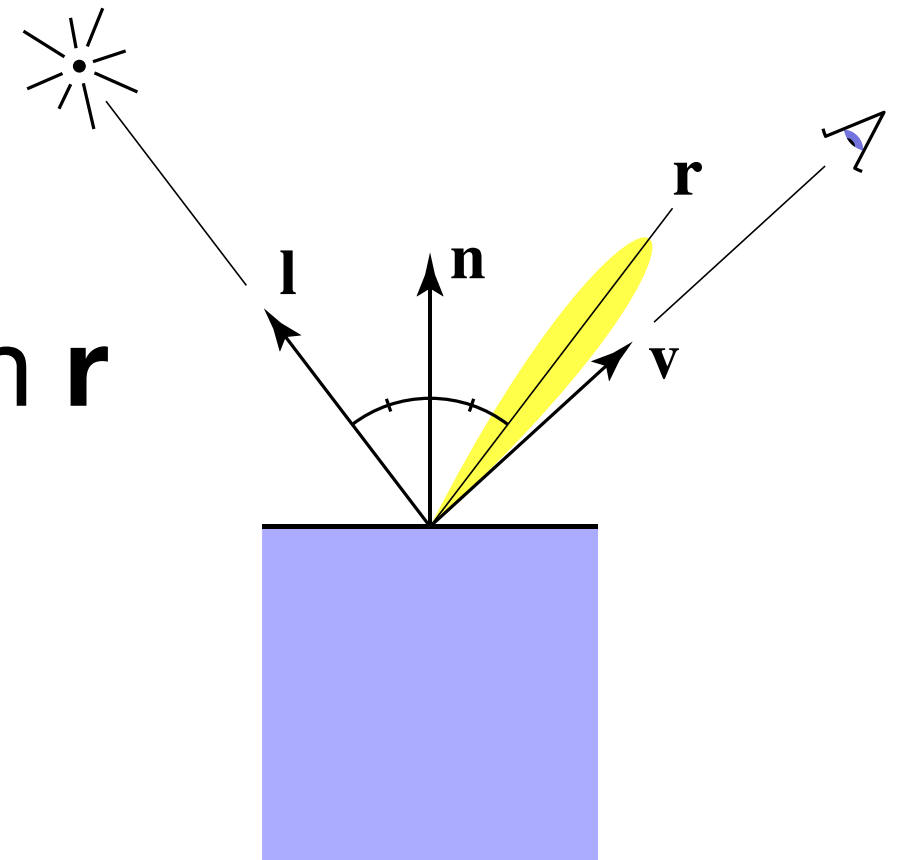
Key observation: the **specular highlight** is *not* pointing straight towards the light.

Specular Reflection

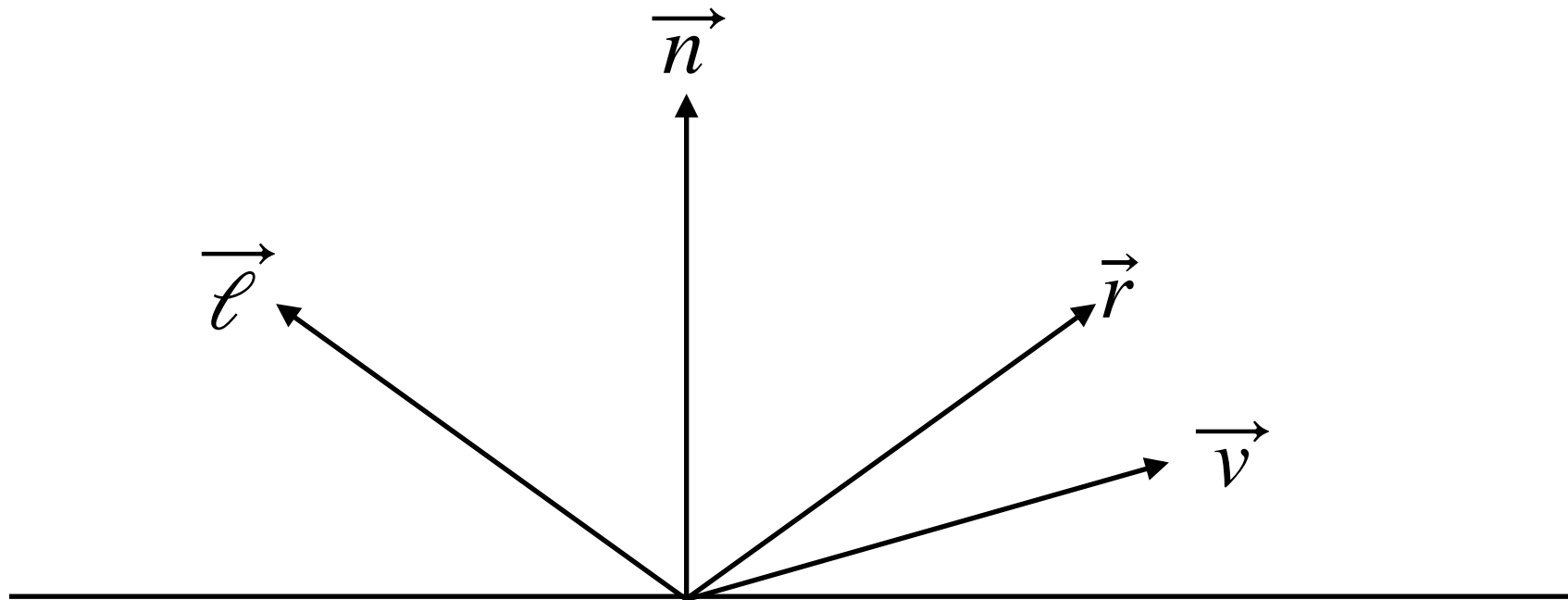
Specular surfaces appear brighter *near* "mirror" configuration



Phong reflection: specular reflection is a function of the angle between mirror direction \mathbf{r} and view direction \mathbf{v} .

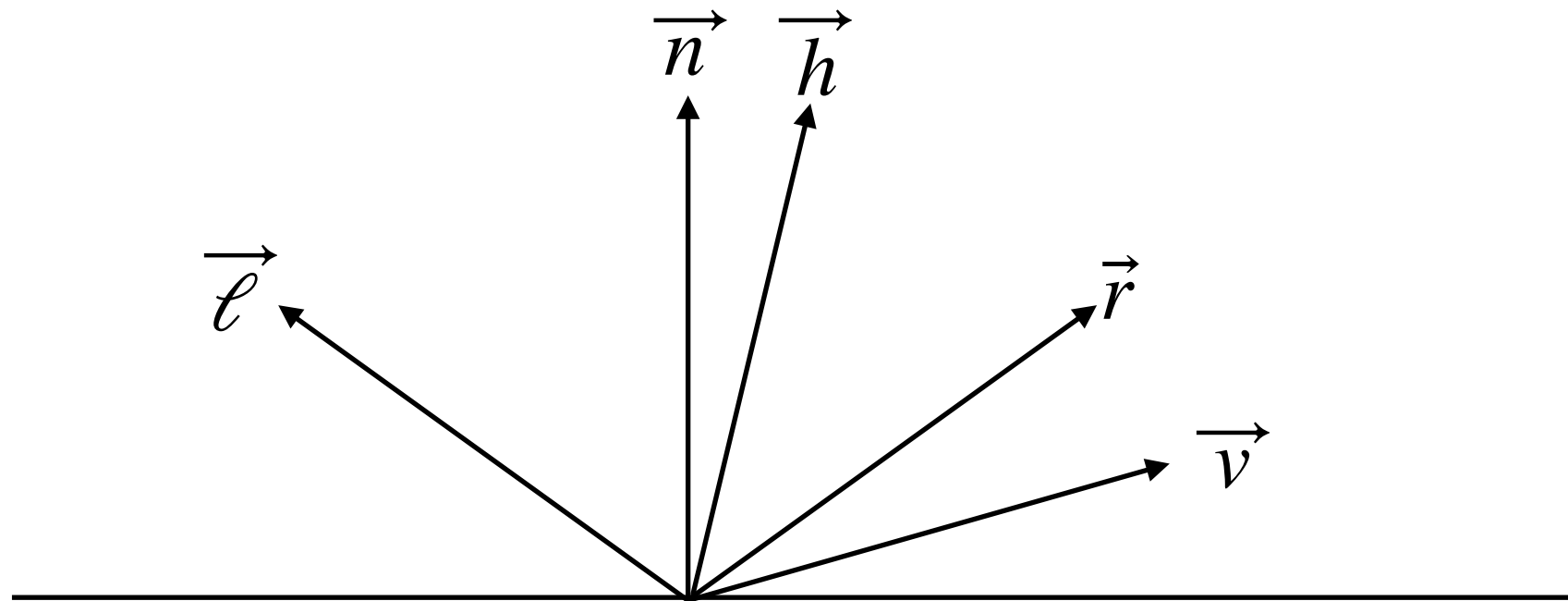


Phong Reflection



Not physically accurate, but perceptually "okay", and intuitive:

Blinn-Phong Reflection



A little less physically inaccurate, still perceptually "okay", and less intuitive:

Specular Reflection

- Blinn-Phong: specular reflection is a function of angle between (**half-way vector** between view and light) and (**the normal**).

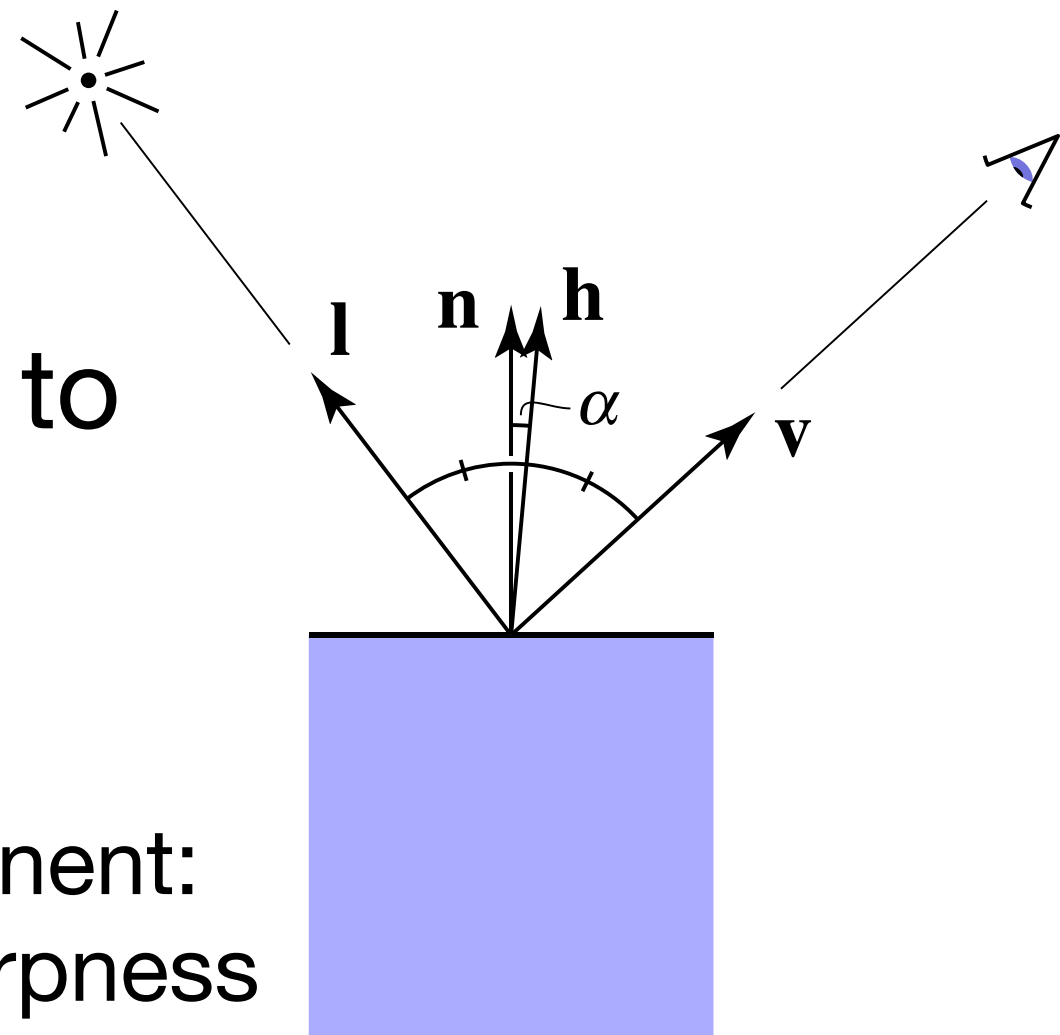
- $h = \text{bisector}(\mathbf{v}, \mathbf{l})$

- Reflected light proportional to

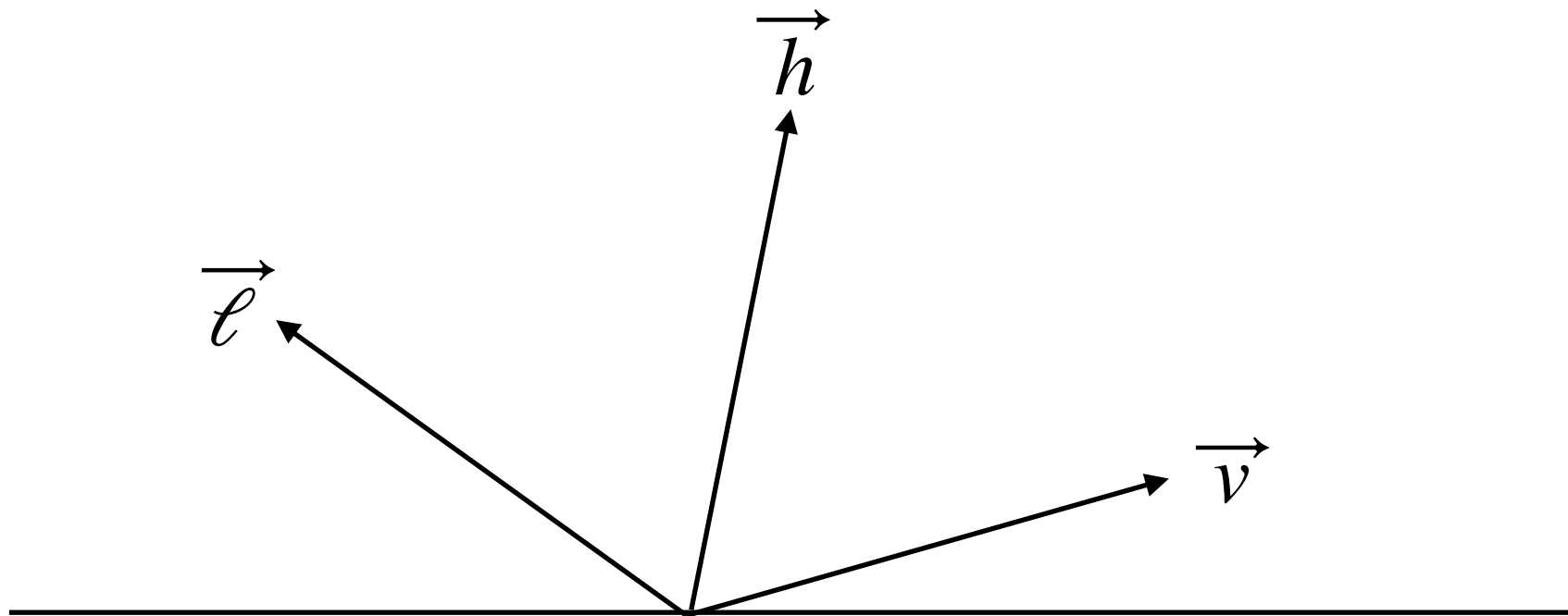
$$k_s \max(0, \vec{n} \cdot \vec{h})^p$$

specular coefficient:
determines strength of
specularity term

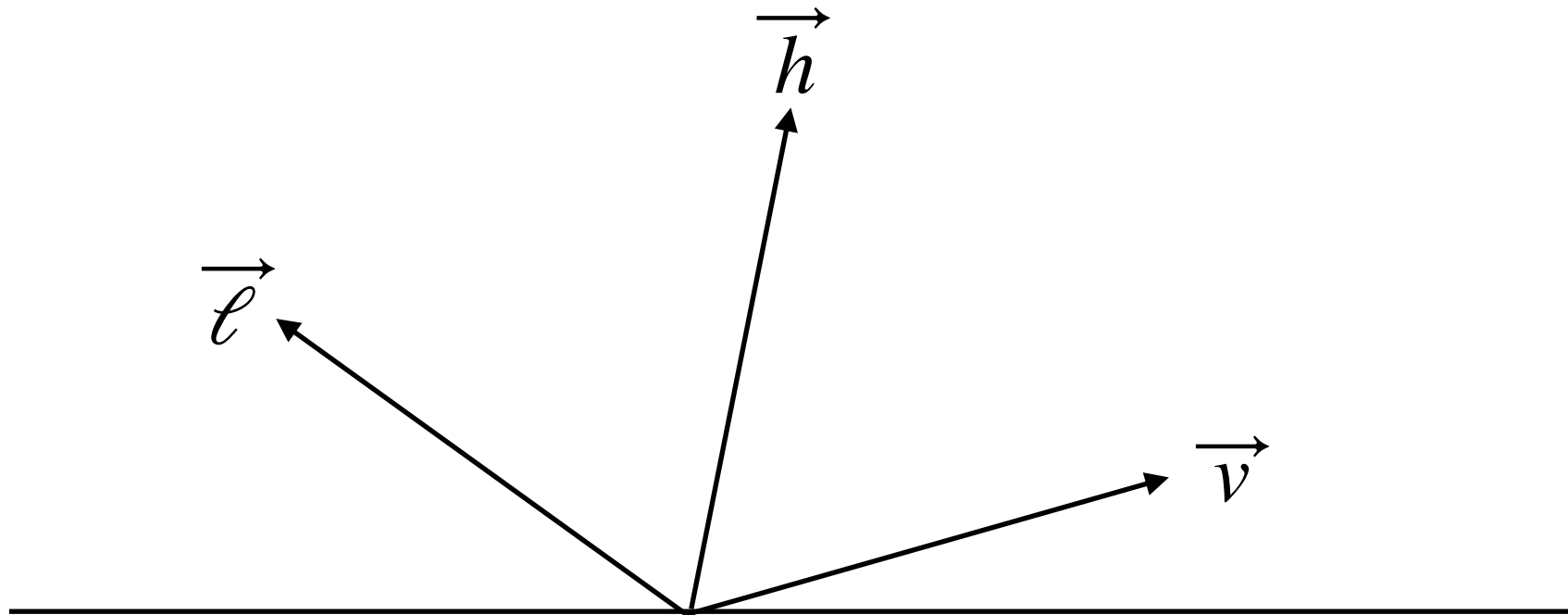
specular exponent:
determines sharpness



Computing \vec{h}



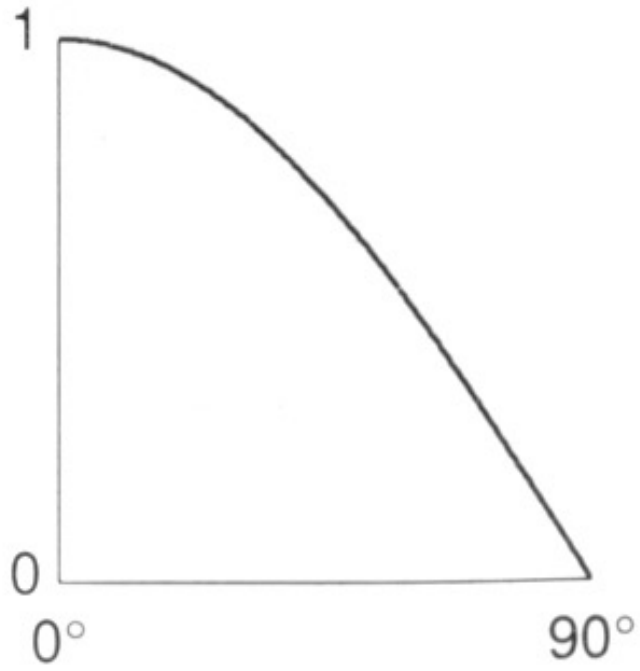
Computing \vec{h}



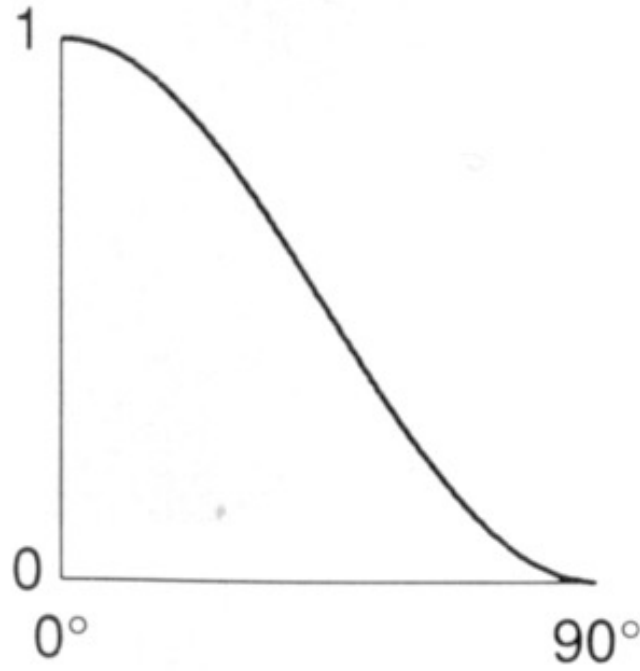
$$\text{bisector}(\vec{v}, \vec{l}) =$$

Effect of p

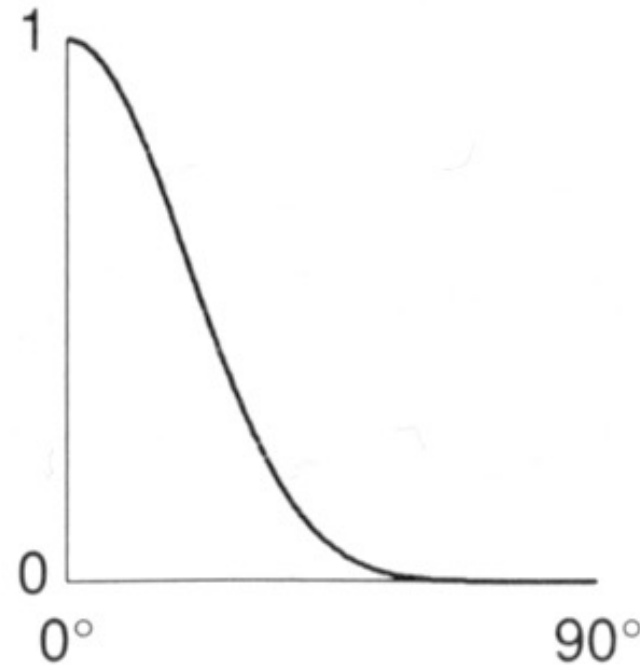
$\cos \alpha$



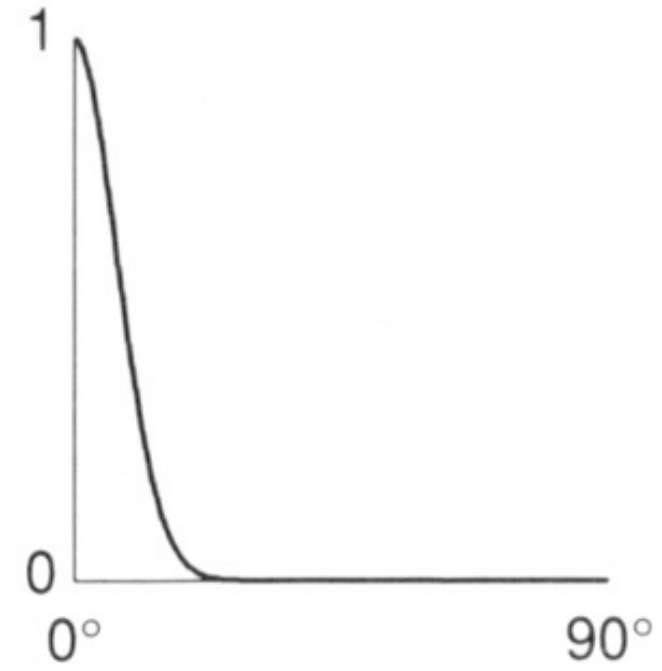
$\cos^2 \alpha$



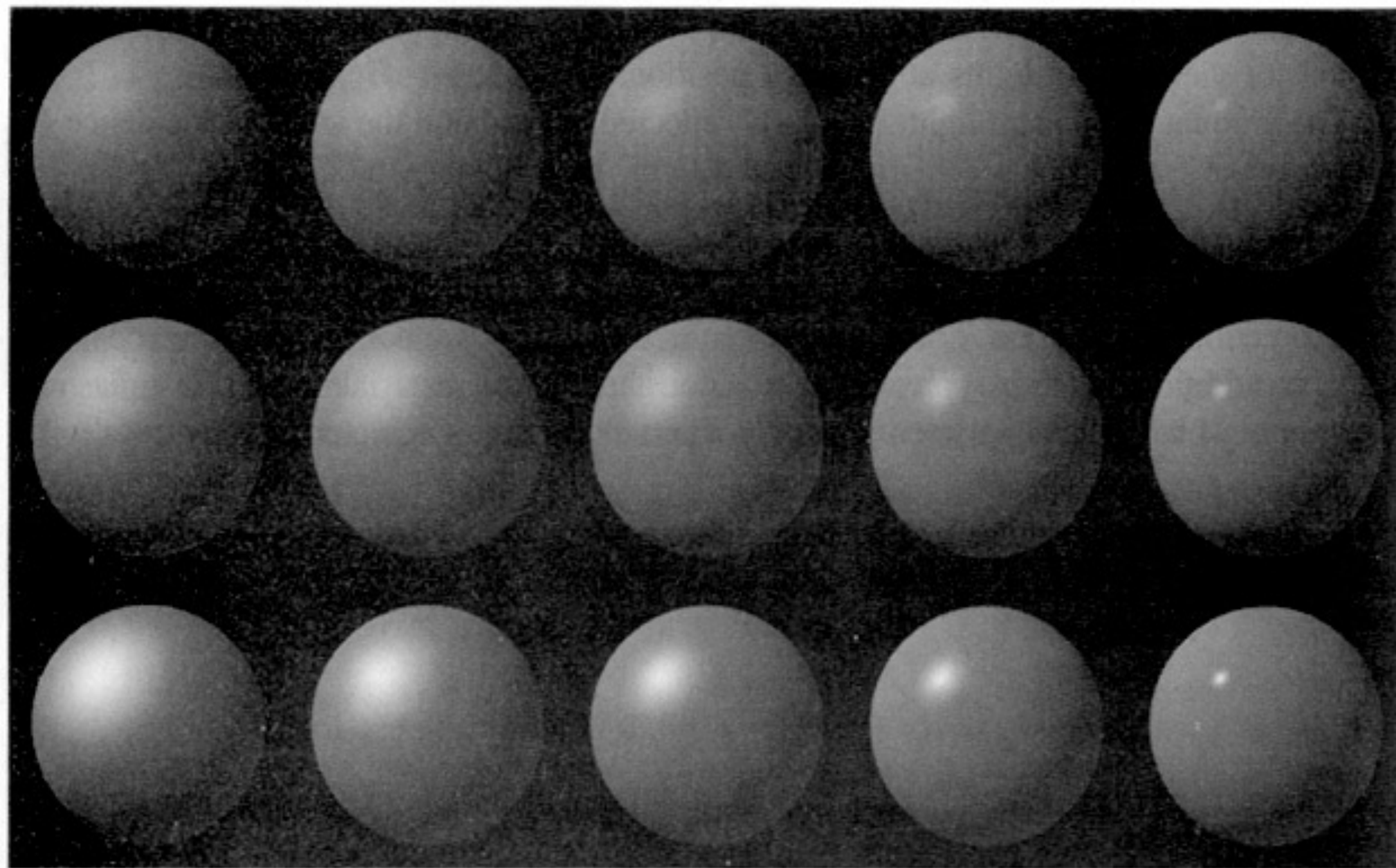
$\cos^8 \alpha$



$\cos^{64} \alpha$



k_s



p \longrightarrow

Putting it all Together: Blinn-Phong Reflection Model

Usually surfaces have both diffuse *and* specular components, so we'll combine the two:

$$\begin{aligned} L &= L_d + L_s \\ &= k_d I \max(0, \vec{n} \cdot \vec{l}) + k_s I \max(0, \vec{n} \cdot \vec{h})^p \end{aligned}$$

Putting it all Together: Blinn-Phong Reflection Model

Usually surfaces have both diffuse *and* specular components, so we'll combine the two:

light reflected

diffuse reflection specular reflection

$$L = L_d + L_s$$
$$= k_d I \max(0, \vec{n} \cdot \vec{l}) + k_s I \max(0, \vec{n} \cdot \vec{h})^p$$

Putting it all Together: Blinn-Phong Reflection Model

Usually surfaces have both diffuse *and* specular components, so we'll combine the two:

light reflected

diffuse reflection

specular reflection

$$L = L_d + L_s$$
$$= k_d I \max(0, \vec{n} \cdot \vec{l}) + k_s I \max(0, \vec{n} \cdot \vec{h})^p$$

diffuse coefficient
(surface brightness
and color)

light intensity

normal

light direction

The diagram illustrates the Blinn-Phong reflection model equation. At the top, 'light reflected' is labeled with an arrow pointing to the variable L . Below it, the equation $L = L_d + L_s$ is shown, with 'diffuse reflection' and 'specular reflection' labeled above L_d and L_s respectively, and arrows pointing to them. The second equation is $= k_d I \max(0, \vec{n} \cdot \vec{l}) + k_s I \max(0, \vec{n} \cdot \vec{h})^p$. Labels with arrows point to various parts of this equation: 'diffuse coefficient (surface brightness and color)' points to k_d ; 'light intensity' points to I ; 'normal' points to \vec{n} ; 'light direction' points to \vec{l} ; and another 'light direction' label points to \vec{h} .

Putting it all Together: Blinn-Phong Reflection Model

Usually surfaces have both diffuse *and* specular components, so we'll combine the two:

light reflected

diffuse reflection specular reflection

$$L = L_d + L_s$$

specular exponent
(sharpness of specularity)

$$= k_d I \max(0, \vec{n} \cdot \vec{l}) + k_s I \max(0, \vec{n} \cdot \vec{h})^p$$

diffuse coefficient
(surface brightness and color)

light intensity

normal

light direction

specular coefficient
(strength [and color] of specularity)

half-vector
between \mathbf{l} and \mathbf{v}

The diagram illustrates the Blinn-Phong reflection model equation. At the top, it shows the total light reflected L as the sum of diffuse reflection L_d and specular reflection L_s . Below this, the equation is expanded: $L = k_d I \max(0, \vec{n} \cdot \vec{l}) + k_s I \max(0, \vec{n} \cdot \vec{h})^p$. Annotations with arrows point to various parts of the equation: k_d is the diffuse coefficient (surface brightness and color); I is the light intensity; \vec{n} is the normal; \vec{l} is the light direction; k_s is the specular coefficient (strength and color of specularity); \vec{h} is the half-vector between the light direction \vec{l} and the view direction \vec{v} ; and p is the specular exponent (sharpness of specularity).

Putting it all Together: Blinn-Phong Reflection Model

Usually surfaces have both diffuse *and* specular components, so we'll combine the two:

light reflected

diffuse reflection specular reflection

$$L = L_d + L_s$$

specular exponent
(sharpness of specularity)

$$= k_d I \max(0, \vec{n} \cdot \vec{l}) + k_s I \max(0, \vec{n} \cdot \vec{h})^p$$

diffuse coefficient
(surface brightness and color)

light intensity

normal

light direction

specular coefficient
(strength [and color] of specularity)

half-vector between \mathbf{l} and \mathbf{v}

In code: `function shade_light(light, hitrec, ...)`

What if there are multiple lights?

Light is additive - add them together:

$$L = \sum_{i=1}^{\# \text{ lights}} k_d I \max(0, \vec{n} \cdot \vec{l}_i) + k_s I \max(0, \vec{n} \cdot \vec{h}_i)^p$$

In code:

What if there are multiple lights?

Light is additive - add them together:

$$L = \sum_{i=1}^{\# \text{ lights}} k_d I \max(0, \vec{n} \cdot \vec{l}_i) + k_s I \max(0, \vec{n} \cdot \vec{h}_i)^p$$

In code:

```
function determine_color(hitrec, ray, scene, ...):
```

What if there are multiple lights?

Light is additive - add them together:

$$L = \sum_{i=1}^{\# \text{ lights}} k_d I \max(0, \vec{n} \cdot \vec{l}_i) + k_s I \max(0, \vec{n} \cdot \vec{h}_i)^p$$

In code:

```
function determine_color(hitrec, ray, scene, ...):  
    color = black
```


What if there are multiple lights?

Light is additive - add them together:

$$L = \sum_{i=1}^{\# \text{ lights}} k_d I \max(0, \vec{n} \cdot \vec{l}_i) + k_s I \max(0, \vec{n} \cdot \vec{h}_i)^p$$

In code:

```
function determine_color(hitrec, ray, scene, ...):  
    color = black  
    for light in scene.lights:
```

What if there are multiple lights?

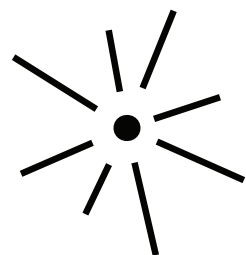
Light is additive - add them together:

$$L = \sum_{i=1}^{\# \text{ lights}} k_d I \max(0, \vec{n} \cdot \vec{l}_i) + k_s I \max(0, \vec{n} \cdot \vec{h}_i)^p$$

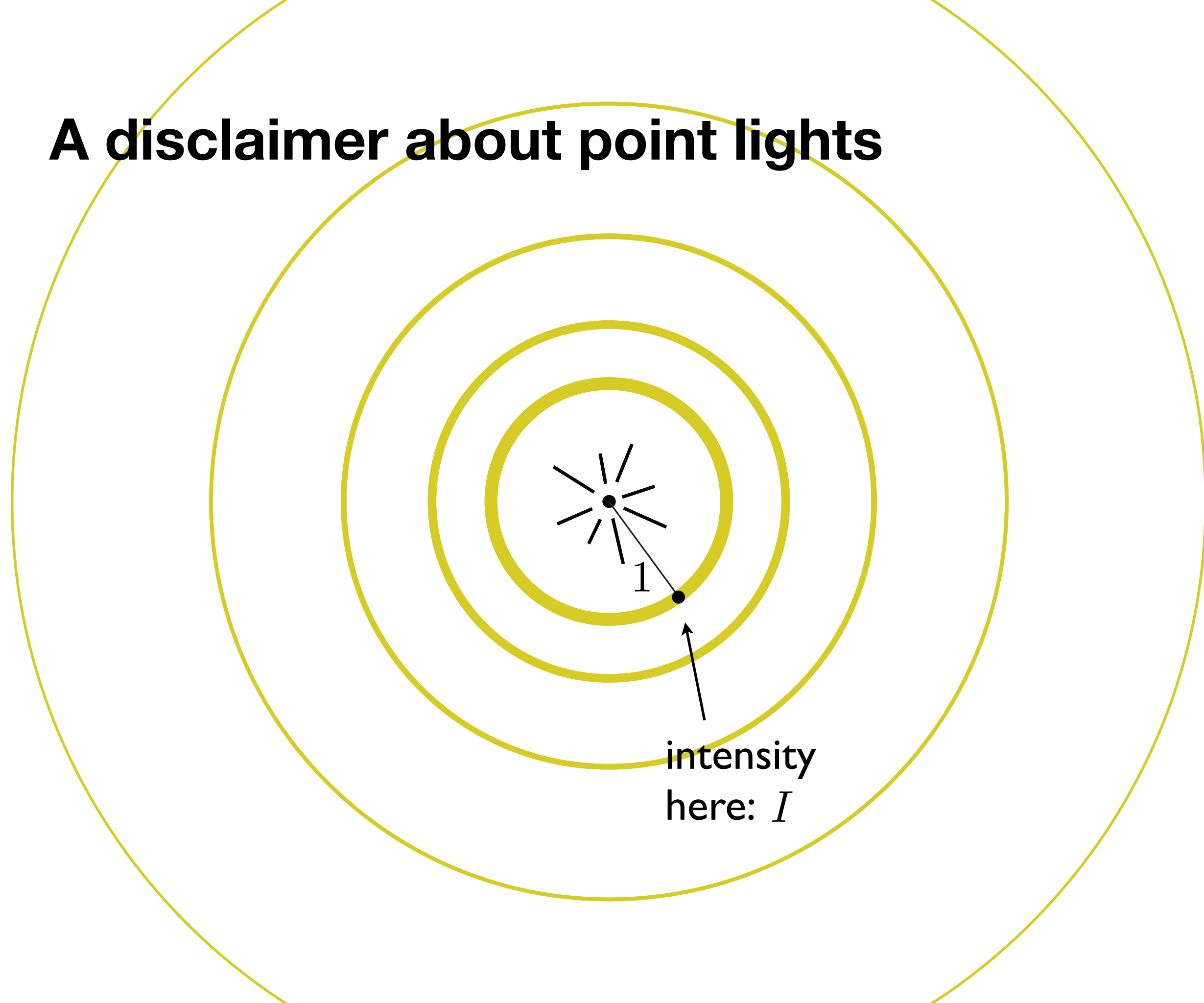
In code:

```
function determine_color(hitrec, ray, scene, ...):  
    color = black  
    for light in scene.lights:  
        color += shade_light(light, hitrec, ...)
```

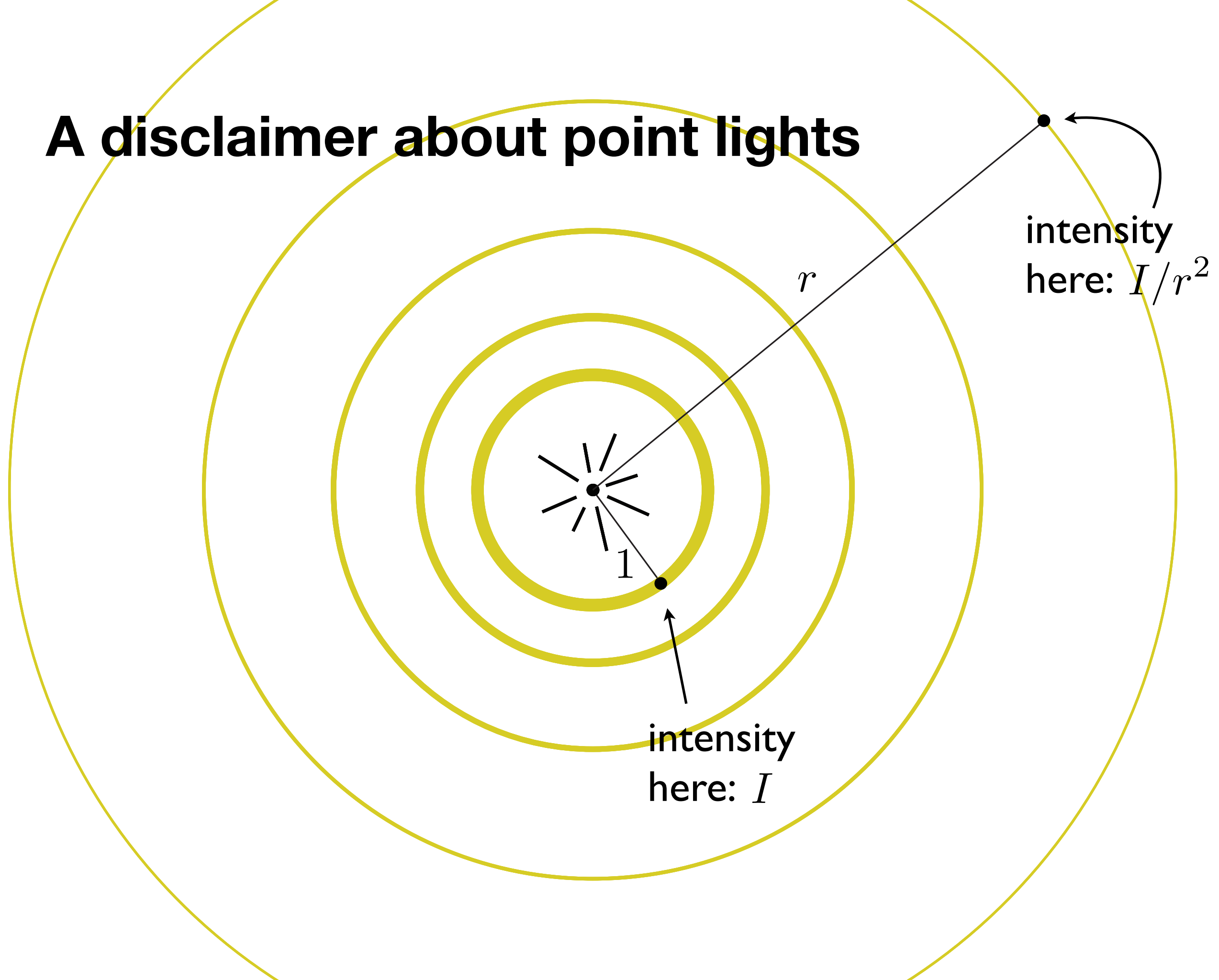
A disclaimer about point lights



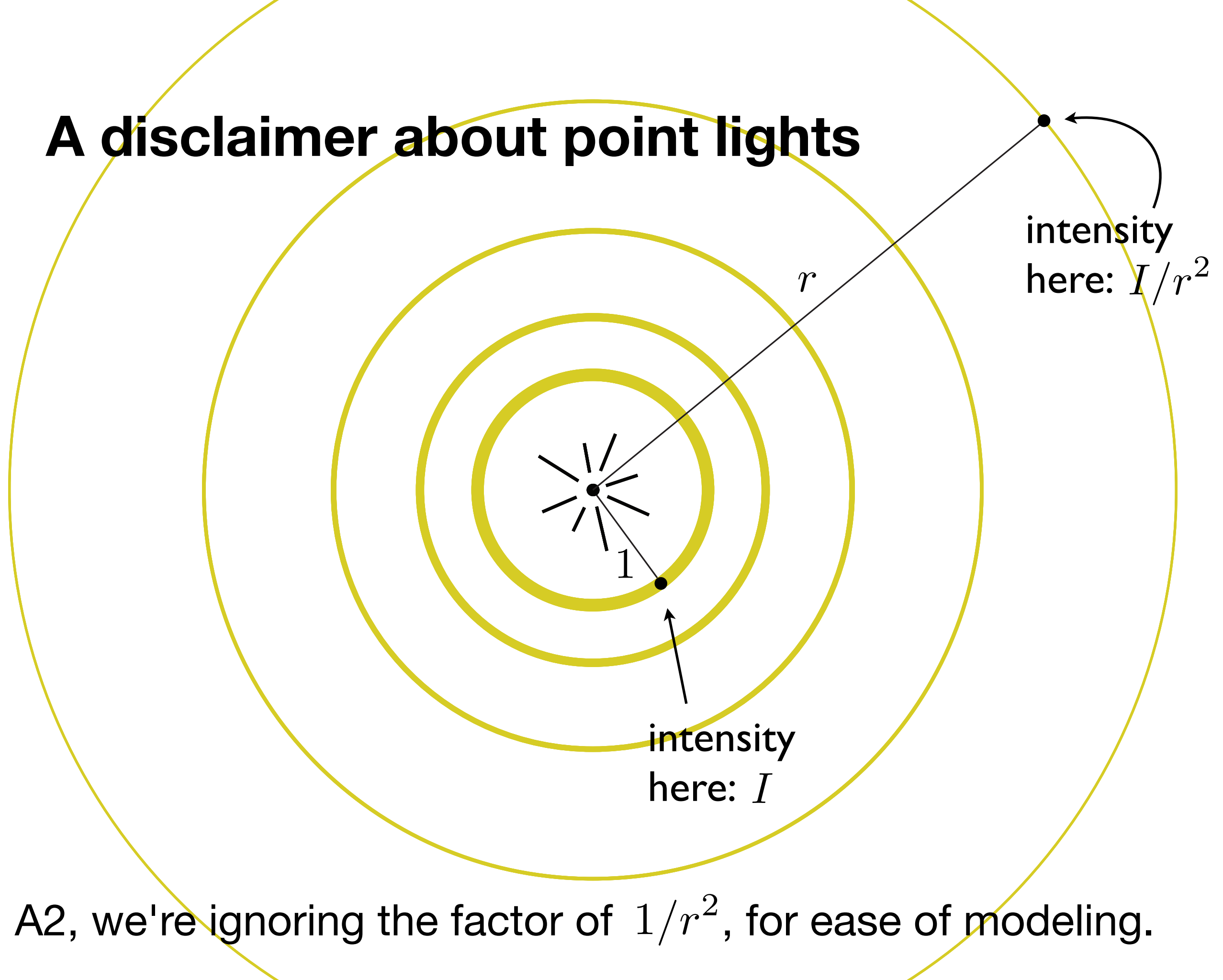
A disclaimer about point lights



A disclaimer about point lights

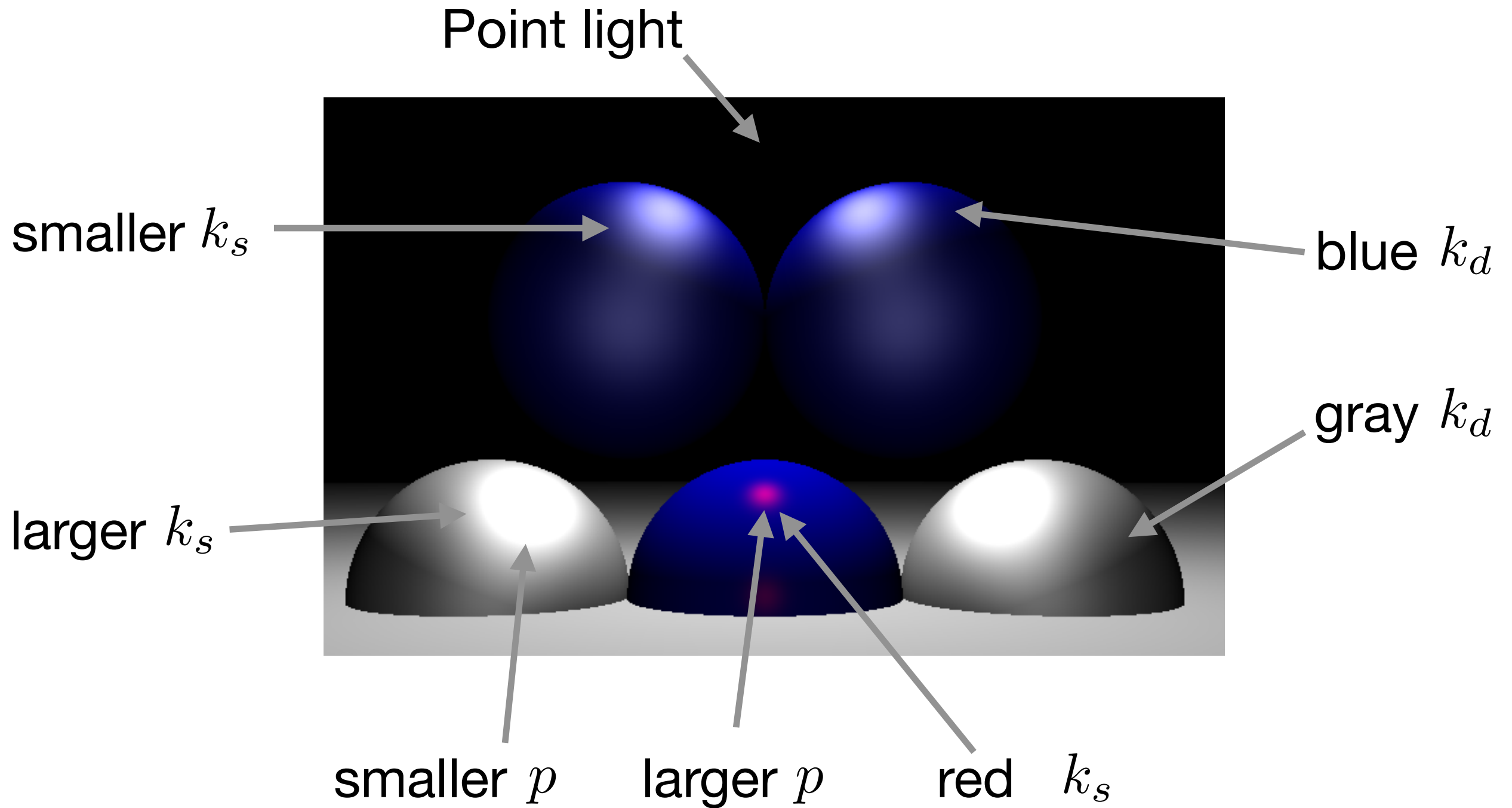


A disclaimer about point lights



In A2, we're ignoring the factor of $1/r^2$, for ease of modeling.

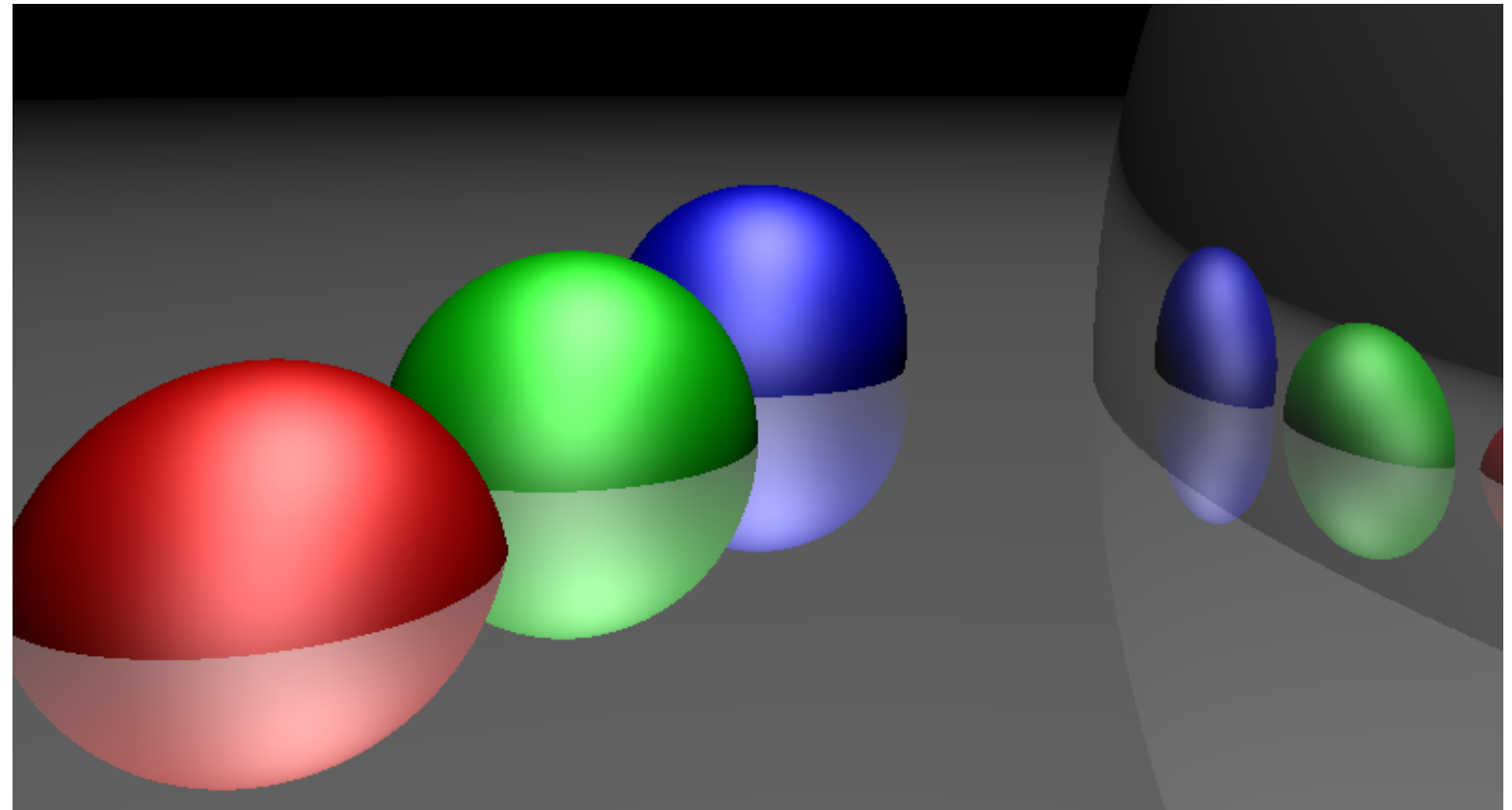
Our images so far:



Partially-Mirrored Surfaces

Notice the floor is gray but also mirror-reflective.

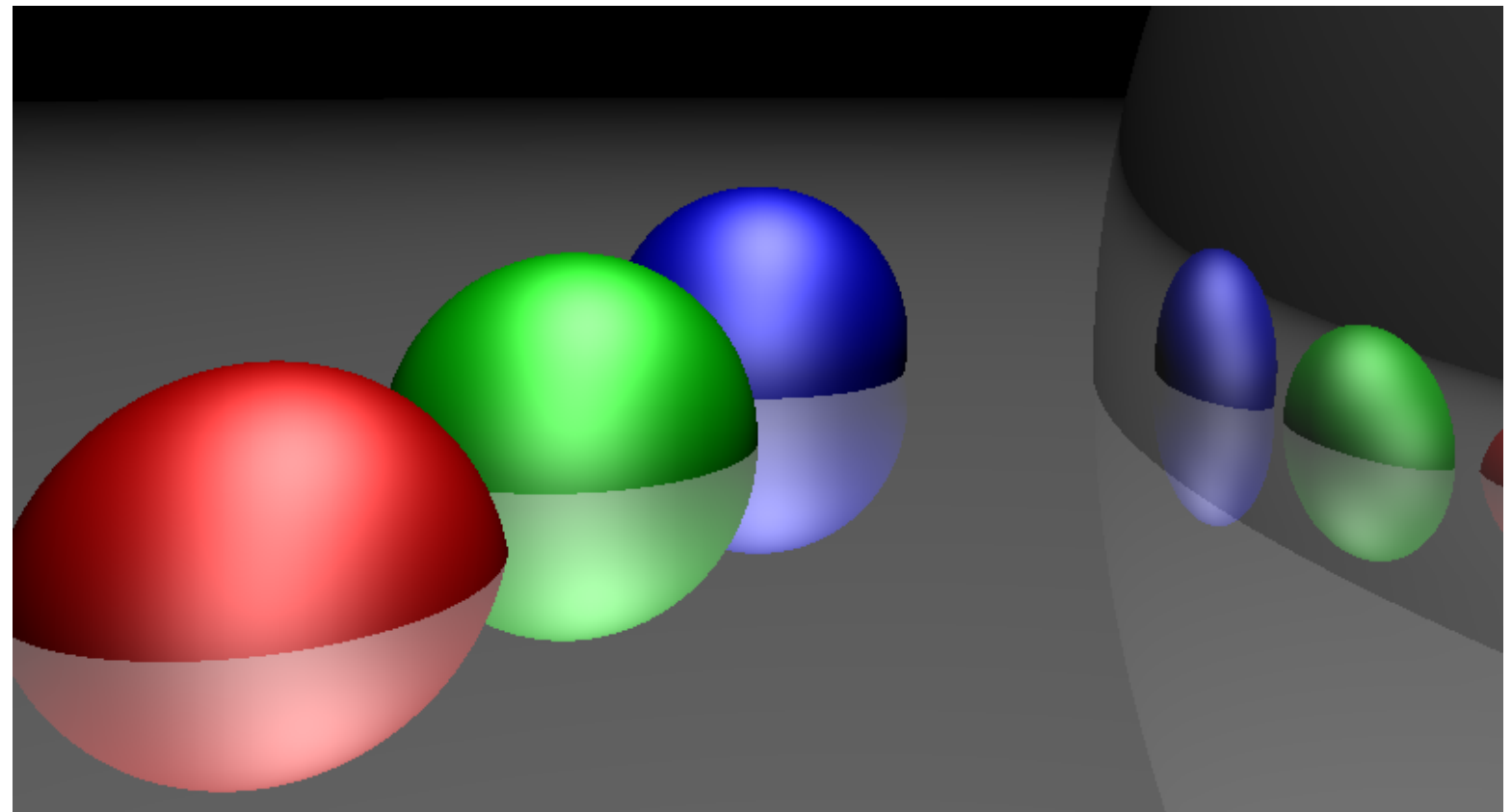
Materials can store a **mirror coefficient**: fraction of light that is reflected in a mirror-like fashion



Partially-Mirrored Surfaces

Notice the floor is gray but also mirror-reflective.

Materials can store a **mirror coefficient:** fraction of light that is reflected in a mirror-like fashion

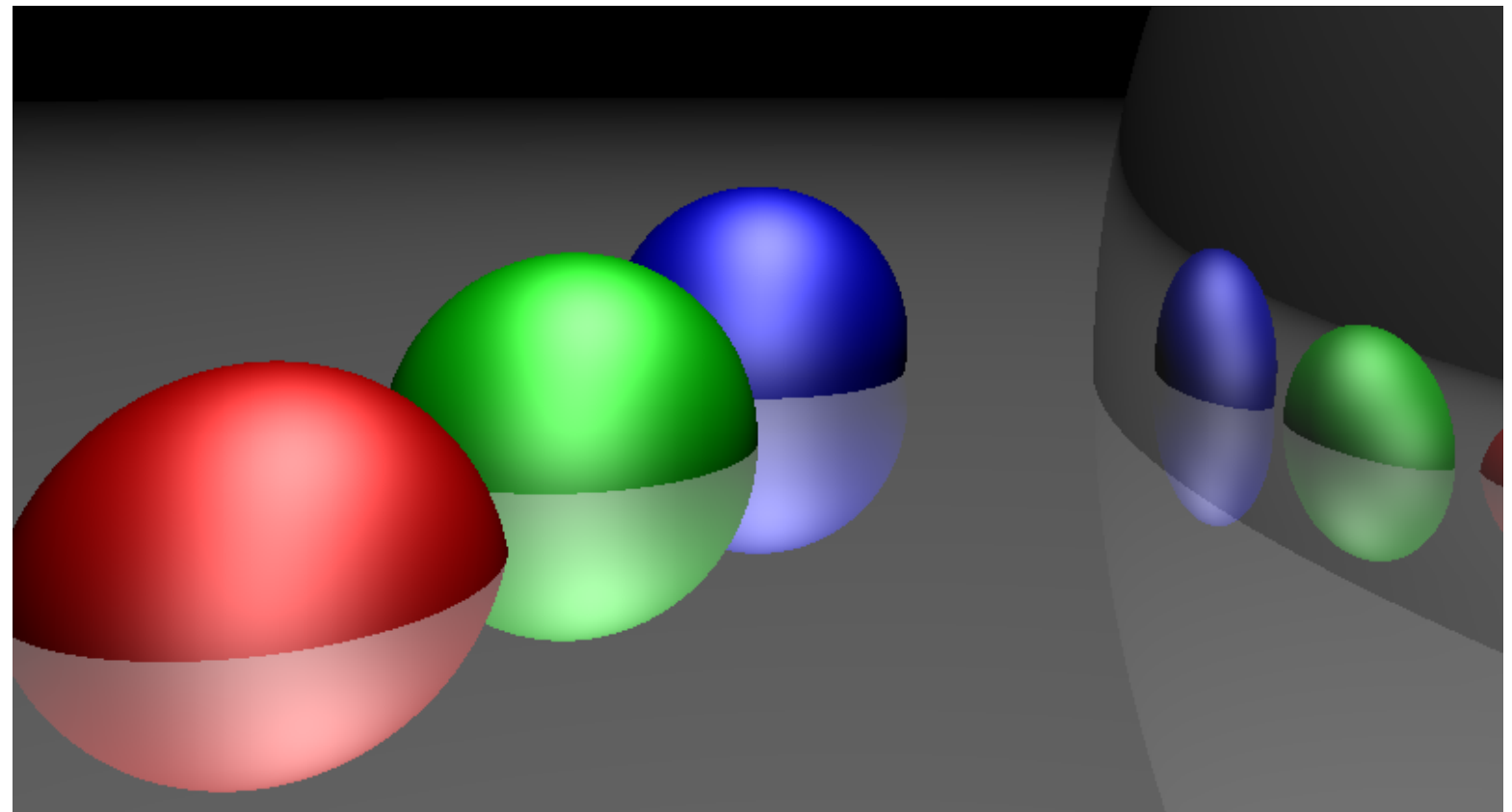


$$L = k_m L_r + (1 - k_m)(L_d + L_s)$$

Partially-Mirrored Surfaces

Notice the floor is gray but also mirror-reflective.

Materials can store a **mirror coefficient**: fraction of light that is reflected in a mirror-like fashion



$$L = k_m L_r + (1 - k_m)(L_d + L_s)$$

mirror coefficient

mirror-reflected light

"local" color (Blinn-Phong)