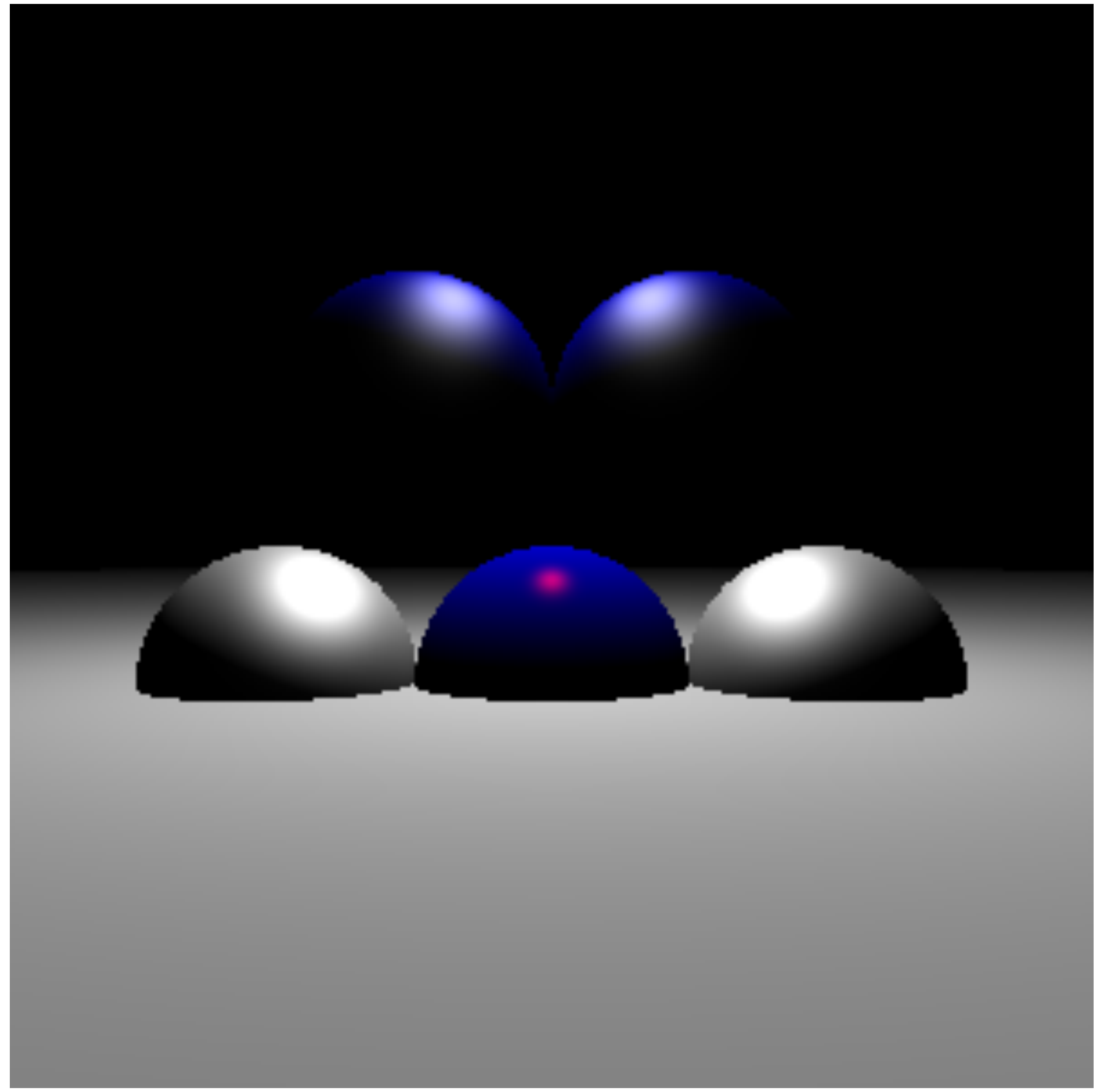


Computer Graphics

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

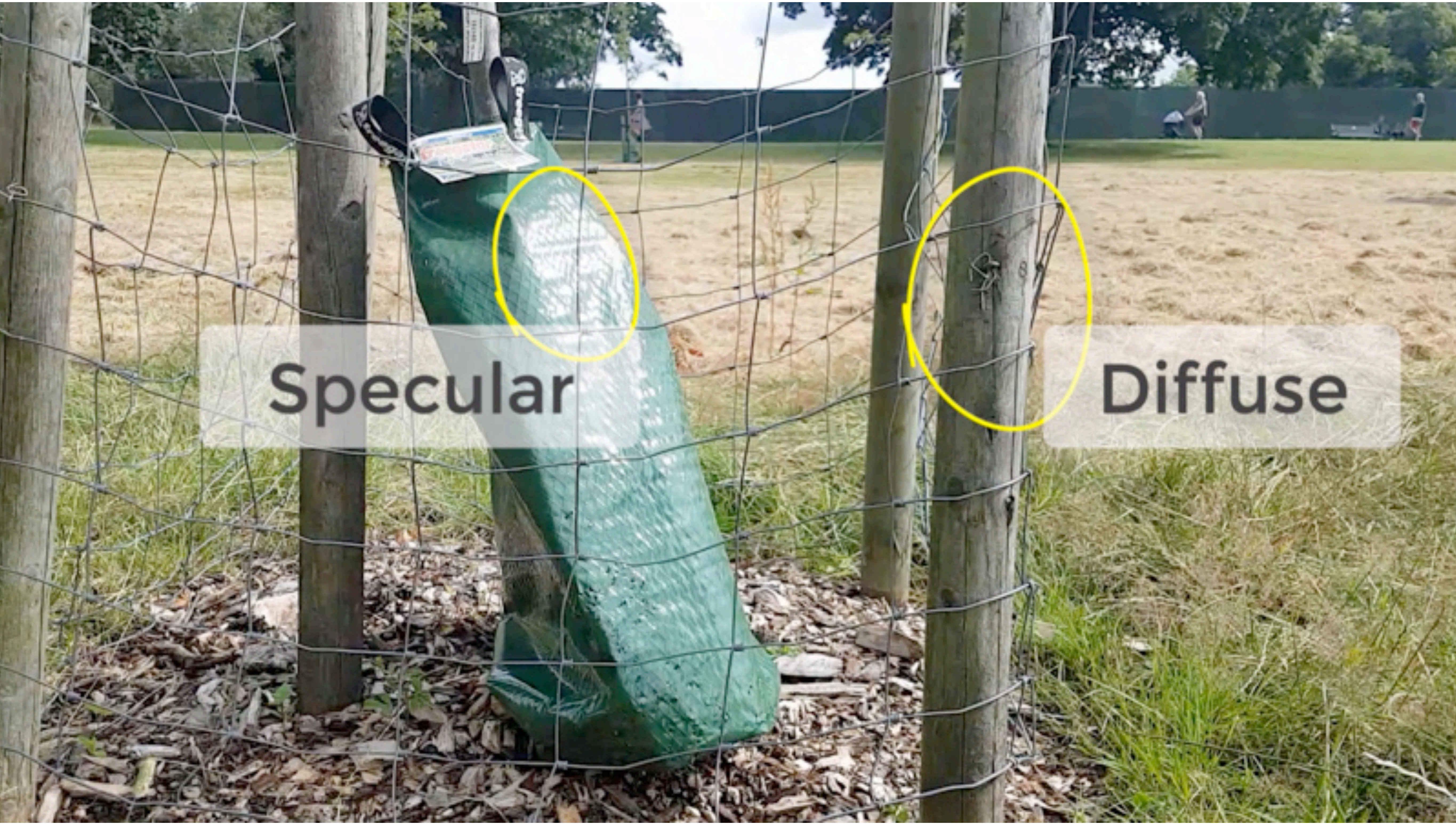
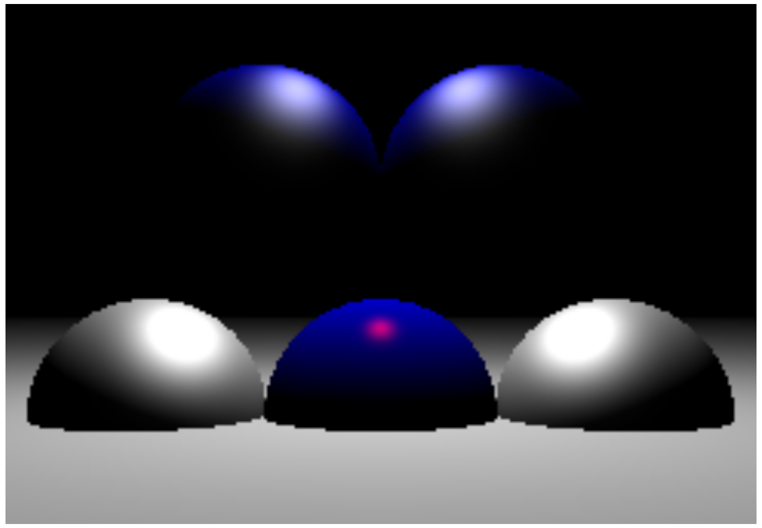


Image: https://www.lovelifedrawing.com/artist-eye-training-highlights-reflections/

Not all surfaces are Lambertian

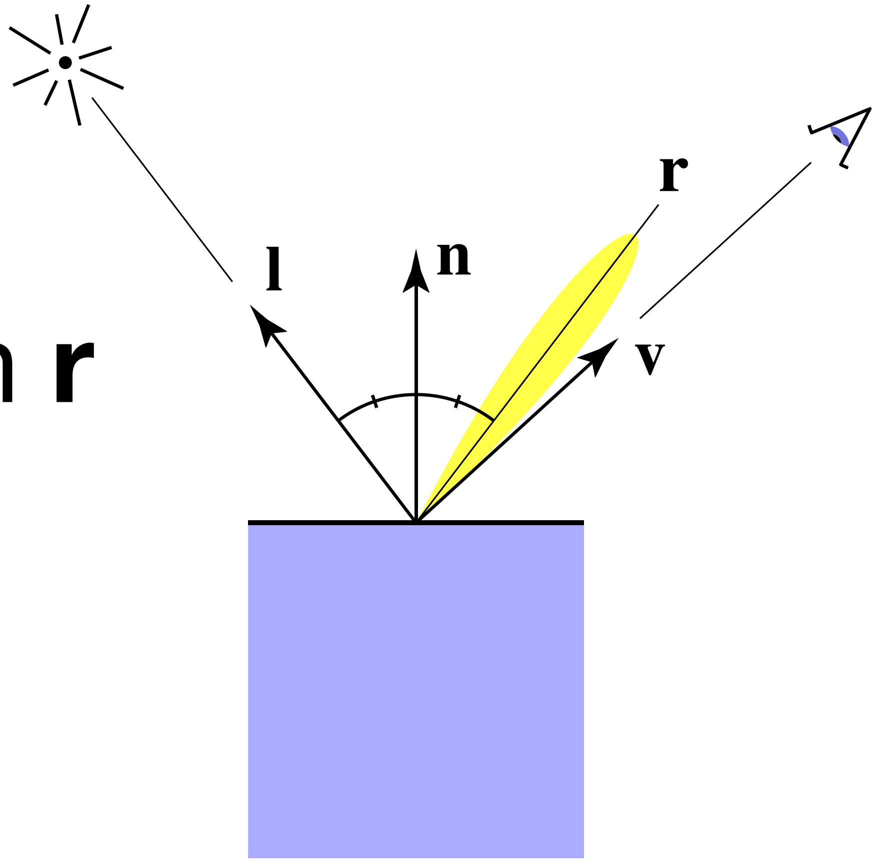
Key observation: the specular highlight is *not* pointing straight towards the light.

Specular Reflection

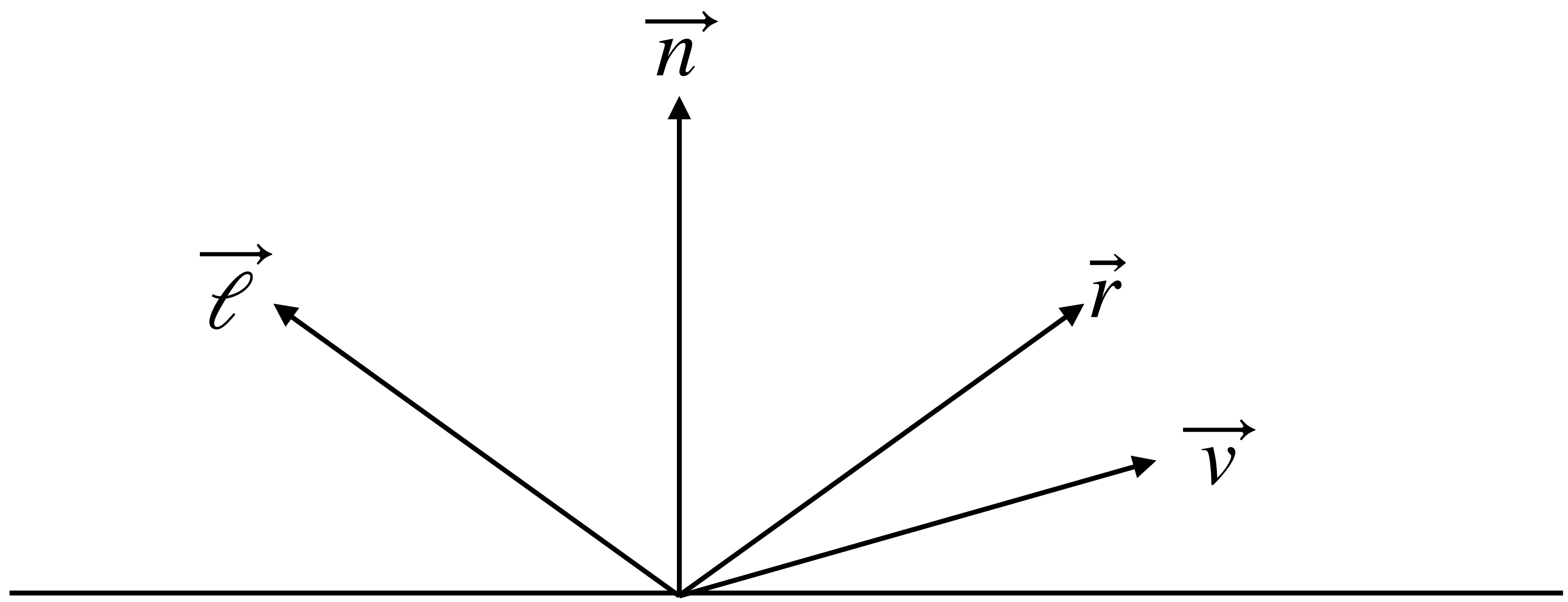


Specular surfaces appear brighter *near* "mirror" configuration

Phong reflection: specular reflection is a function of the angle between mirror direction **r** and view direction **v**.

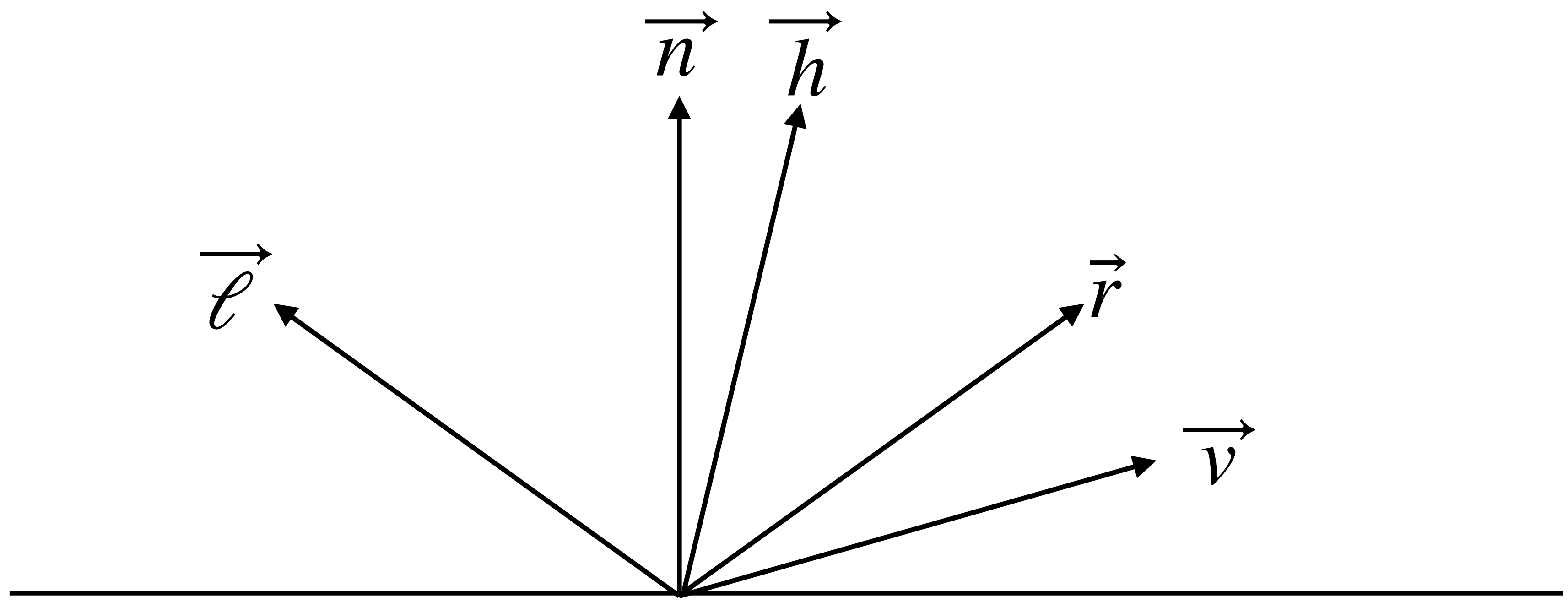


Phong Reflection



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

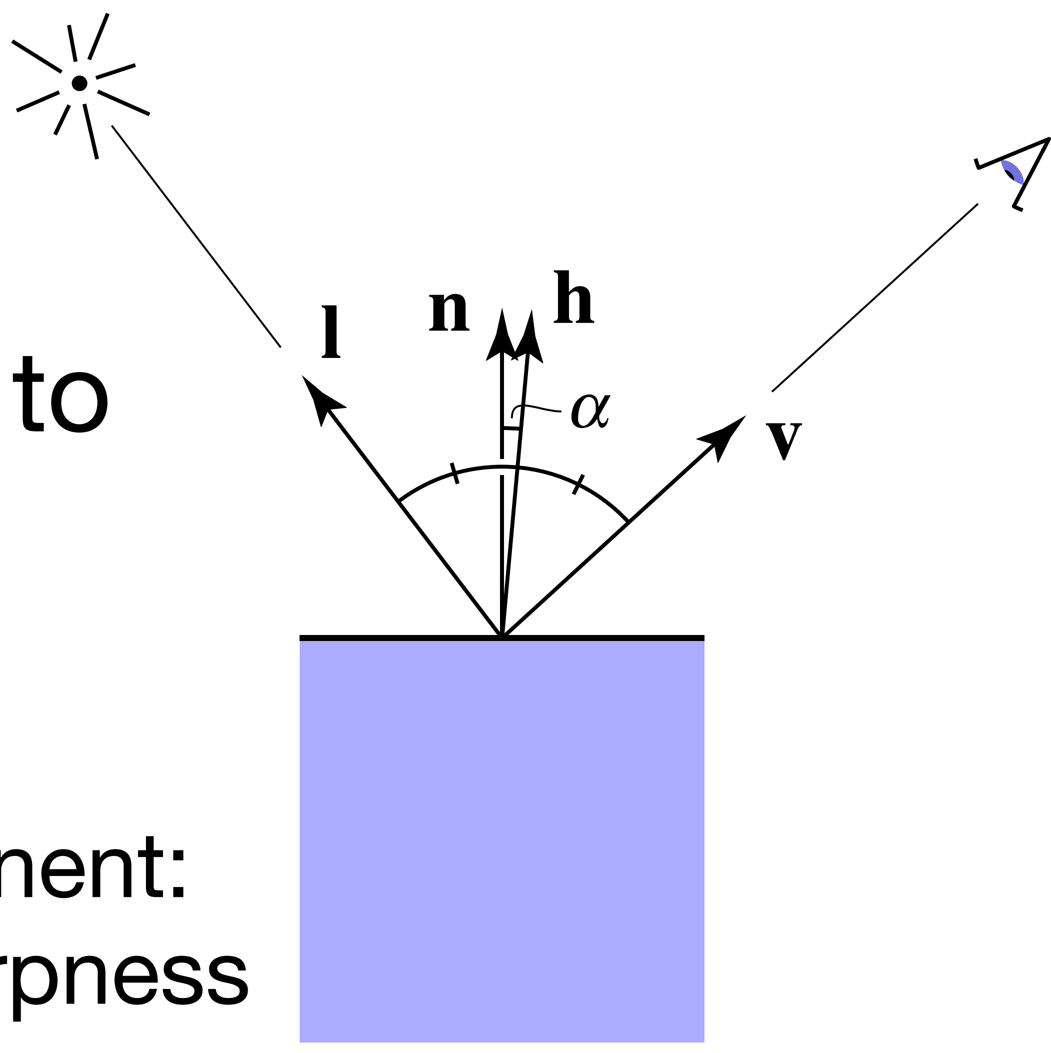
Blinn-Phong Reflection



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 = bisector(**v**, **l**)
- Reflected light proportional to

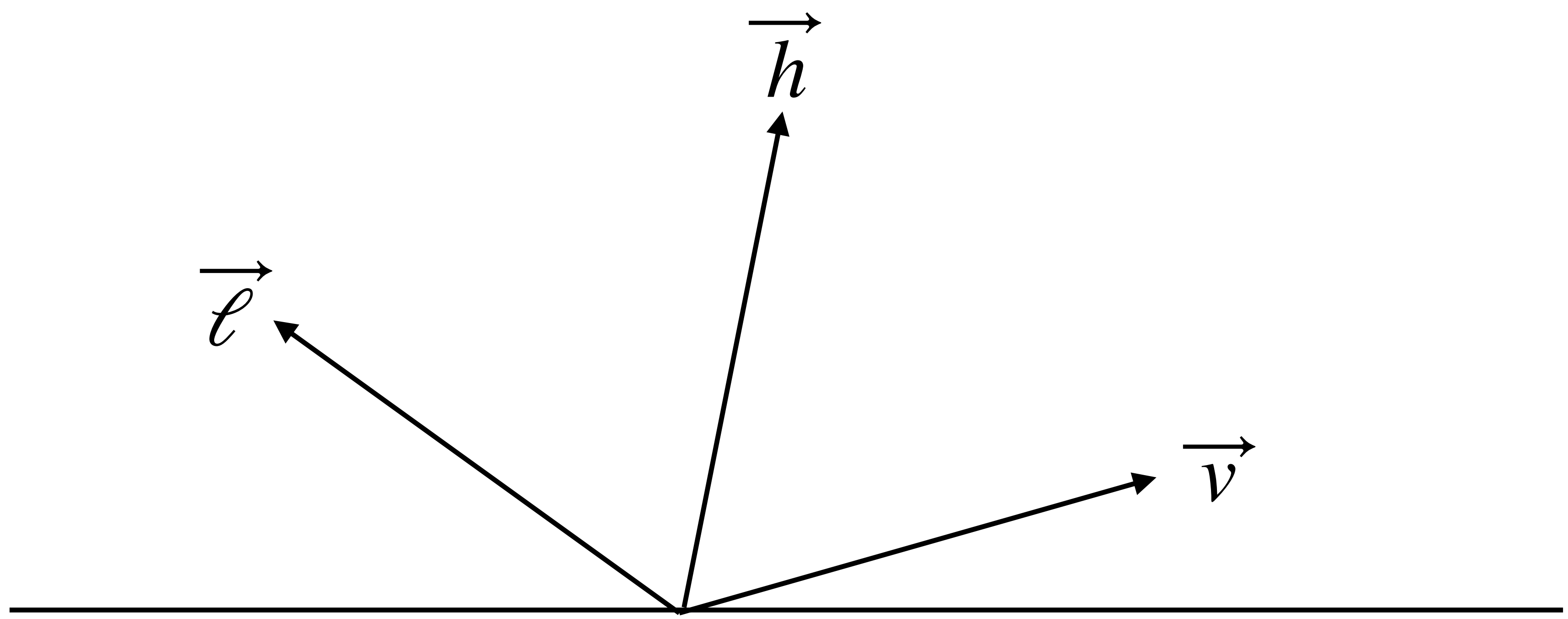


$$k_s \max(0, \vec{n} \cdot \vec{h})^p$$

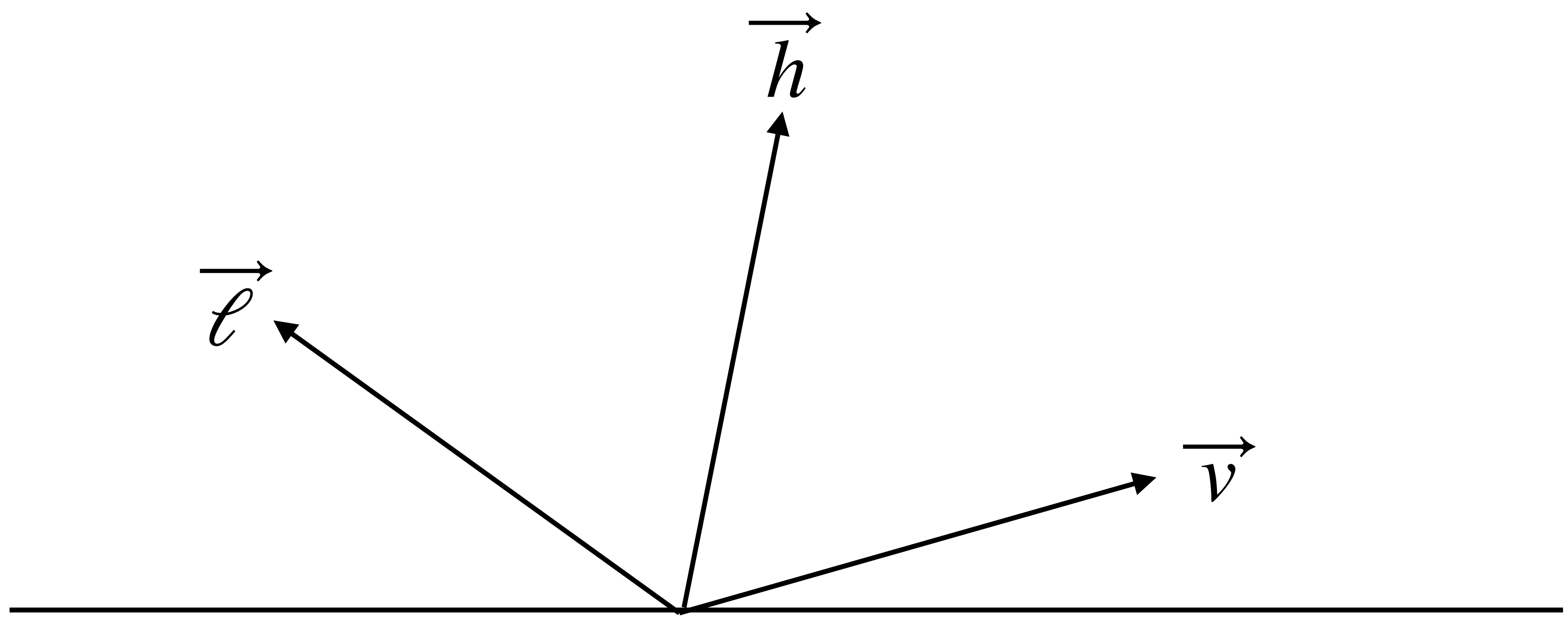
specular coefficient: determines strength of specularity term

specular exponent: determines sharpness

Computing $\overrightarrow{h}$

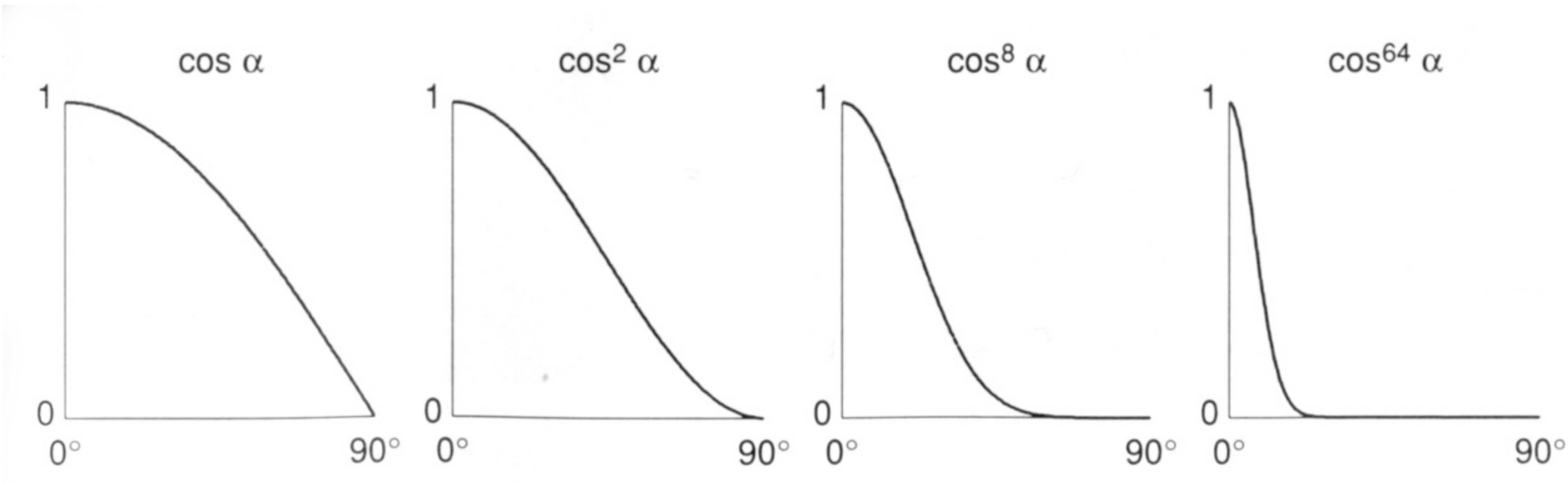


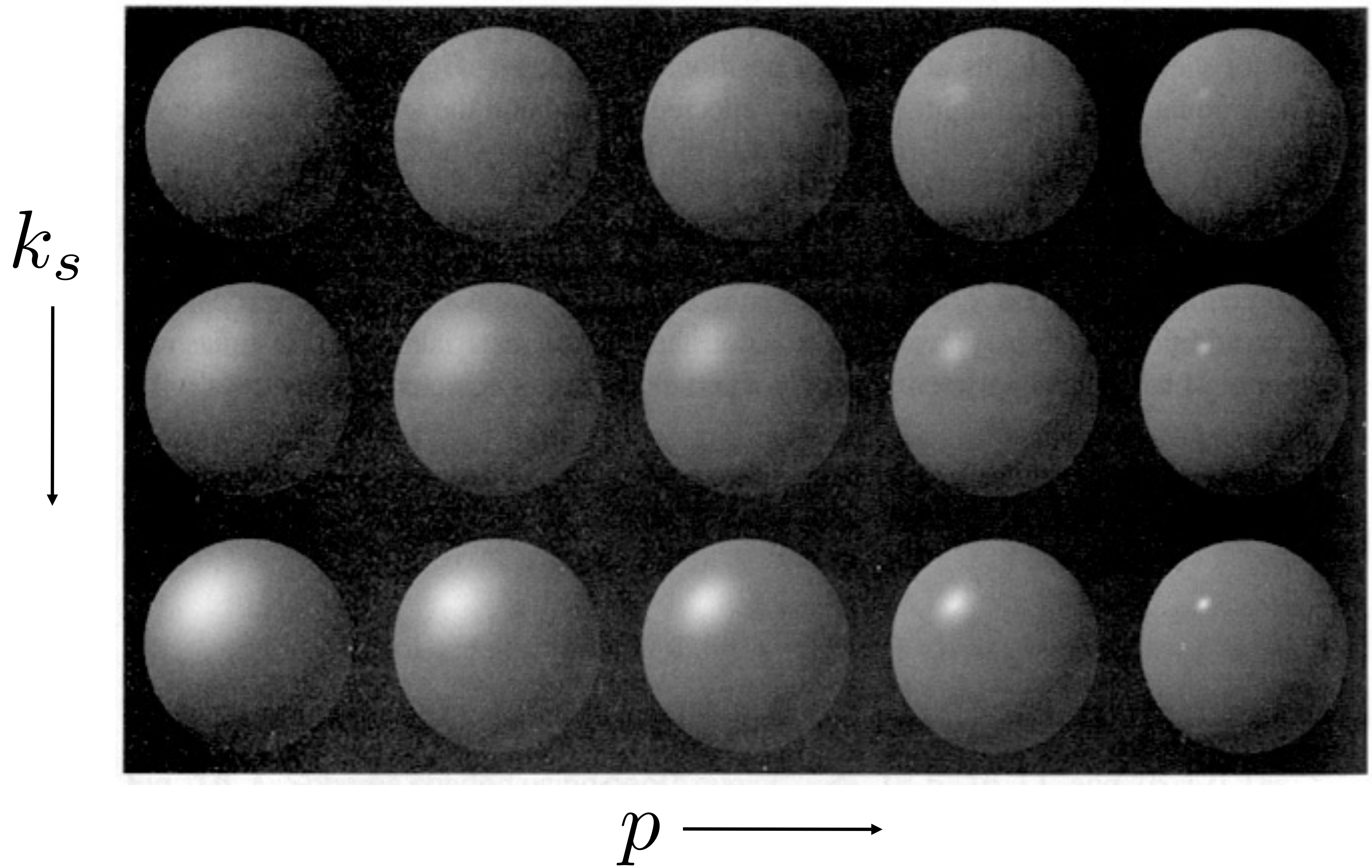
Computing $\vec{h}$



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

Effect of *p*




k_s
p

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

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

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

$$L = L_d + L_s$$

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

diffuse coefficient (surface brightness and color) — light intensity — normal — light direction

Putting it all Together: Blinn-Phong Reflection Model

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

$$L = L_d + L_s$$

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

light reflected

diffuse reflection

specular reflection

specular exponent (*sharpness* of specularity)

diffuse coefficient (surface brightness and color)

light intensity

normal

light direction

specular coefficient (*strength* [and color] of specularity)

half-vector between **l** and **v**

Putting it all Together: Blinn-Phong Reflection Model

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

$$L = L_d + L_s$$

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

- L: light reflected
- L_d: diffuse reflection
- L_s: specular reflection
- k_d: diffuse coefficient (surface brightness and color)
- I: light intensity
- $\vec{n}$: normal
- $\vec{l}$: light direction
- k_s: specular coefficient (*strength* [and color] of specularity)
- $\vec{h}$: half-vector between **l** and **v**
- p: specular exponent (*sharpness* of specularity)

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?

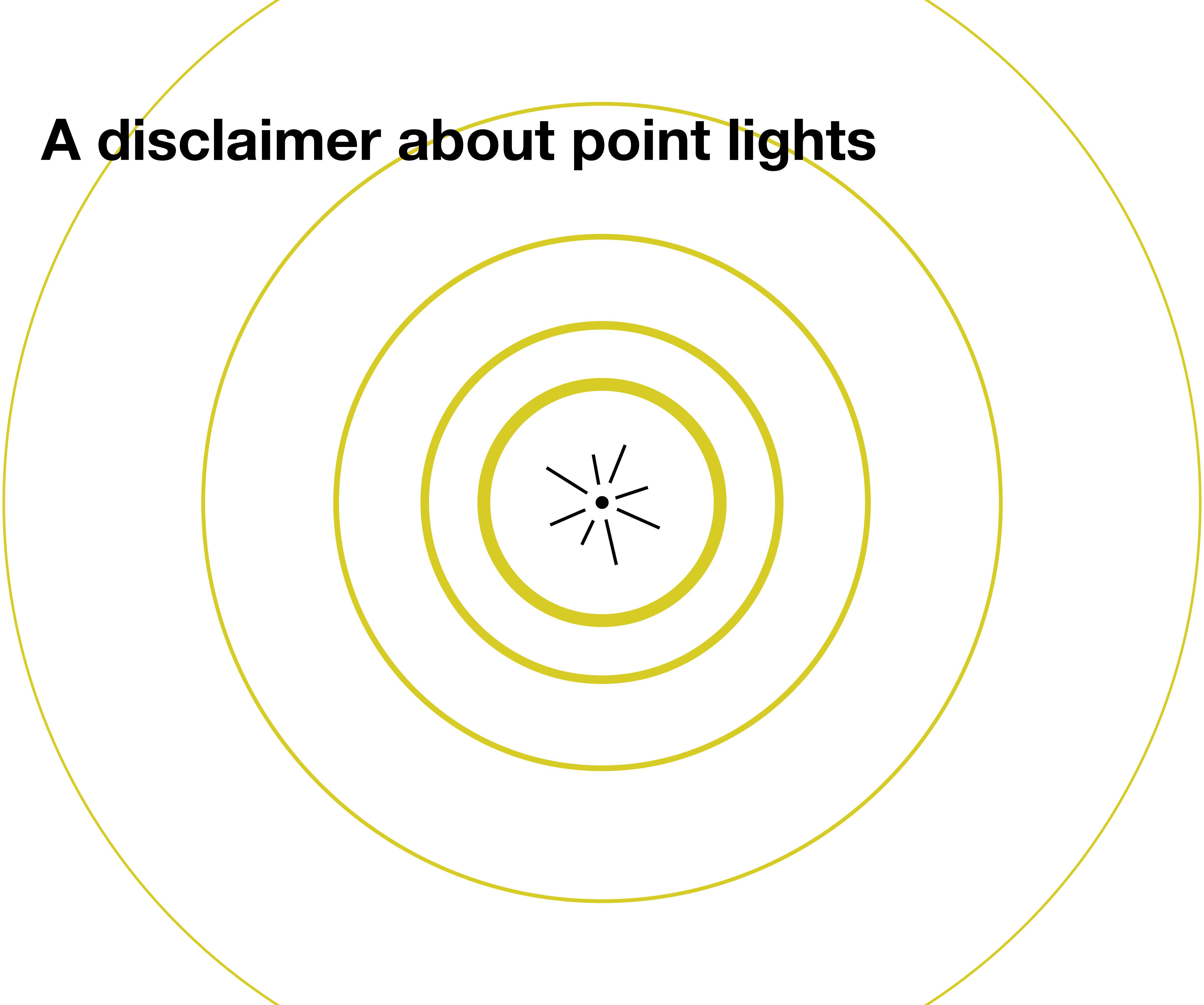
Light is additive - add them together:

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

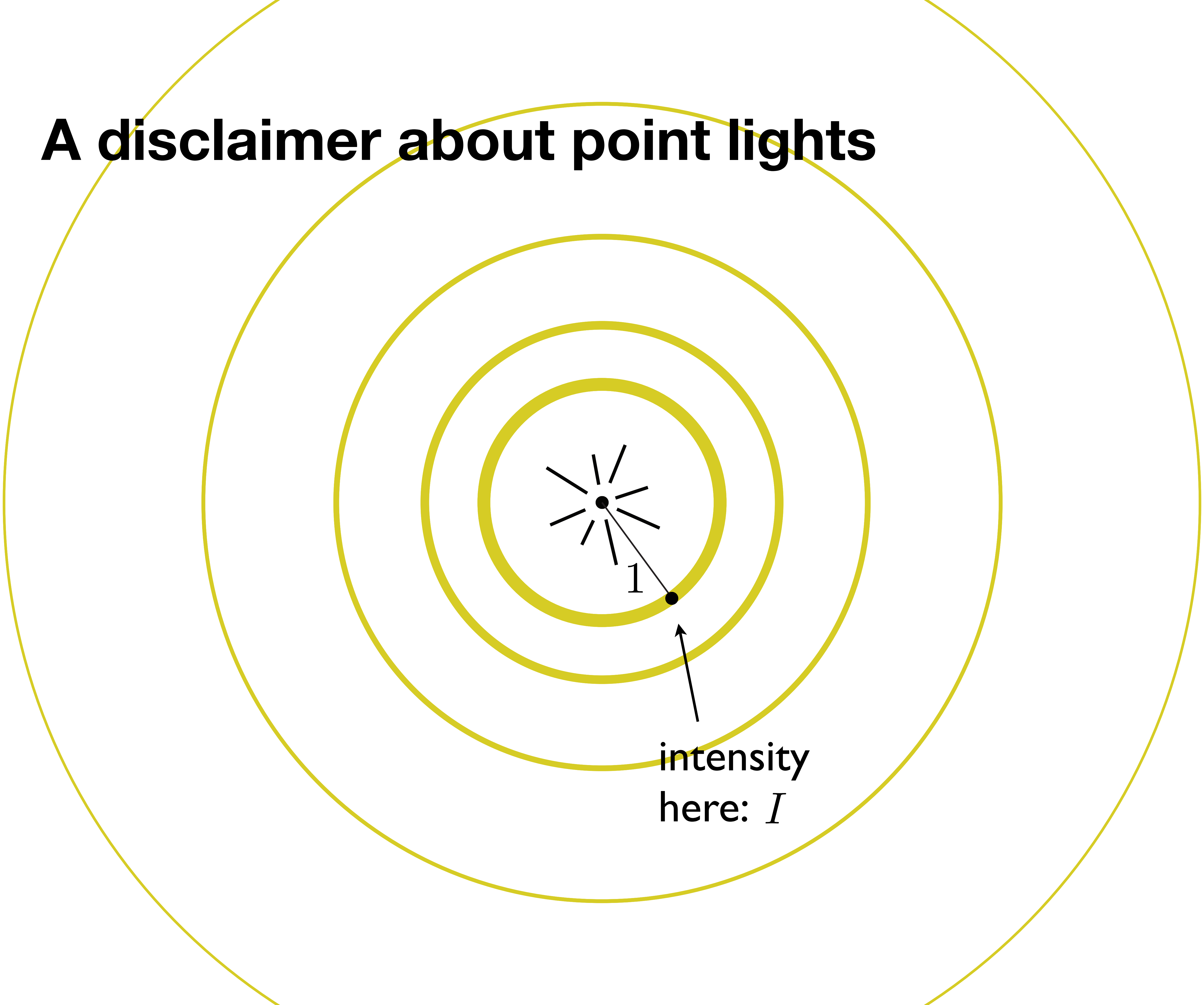
In code:

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

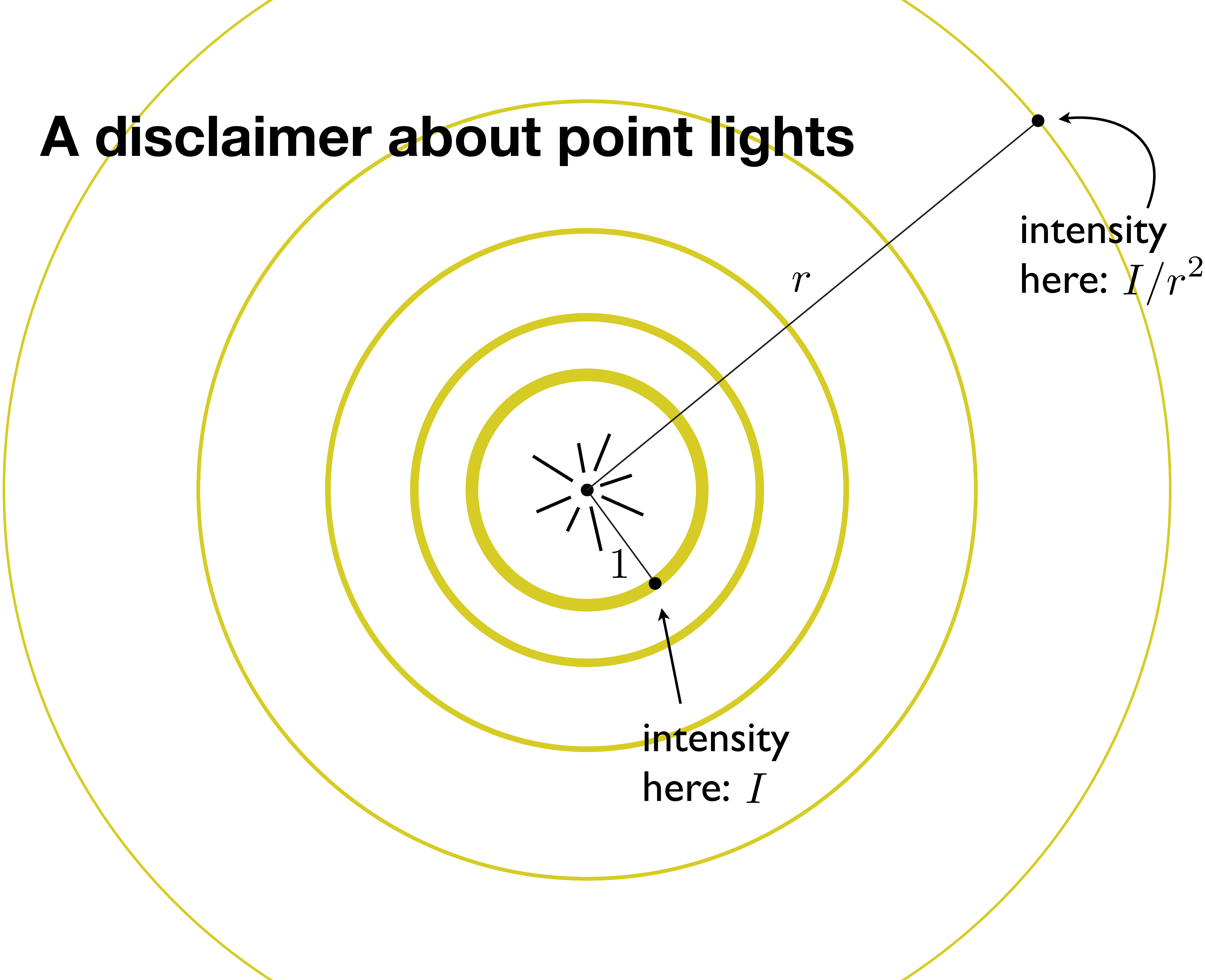
A disclaimer about point lights



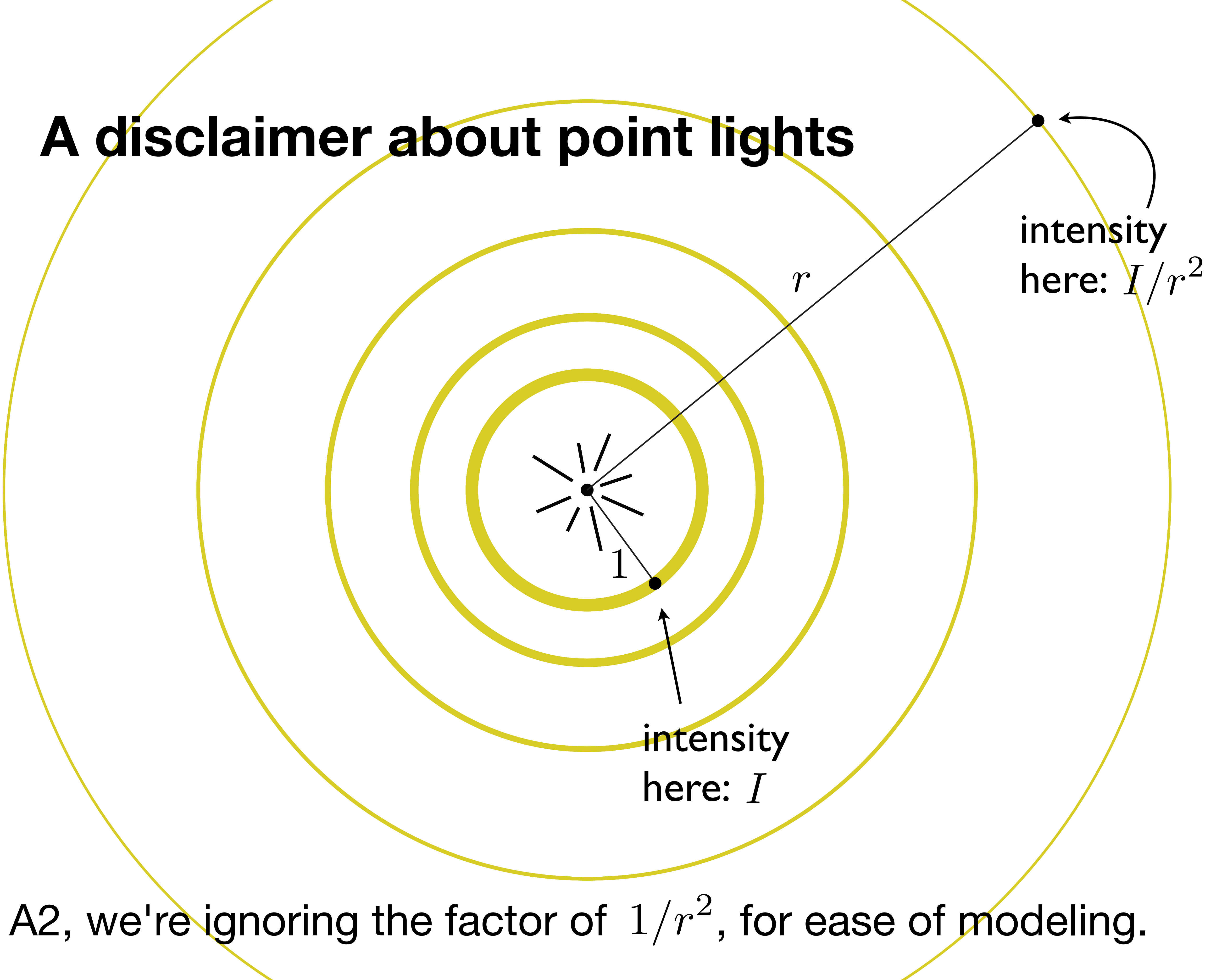
A disclaimer about point lights



A disclaimer about point lights

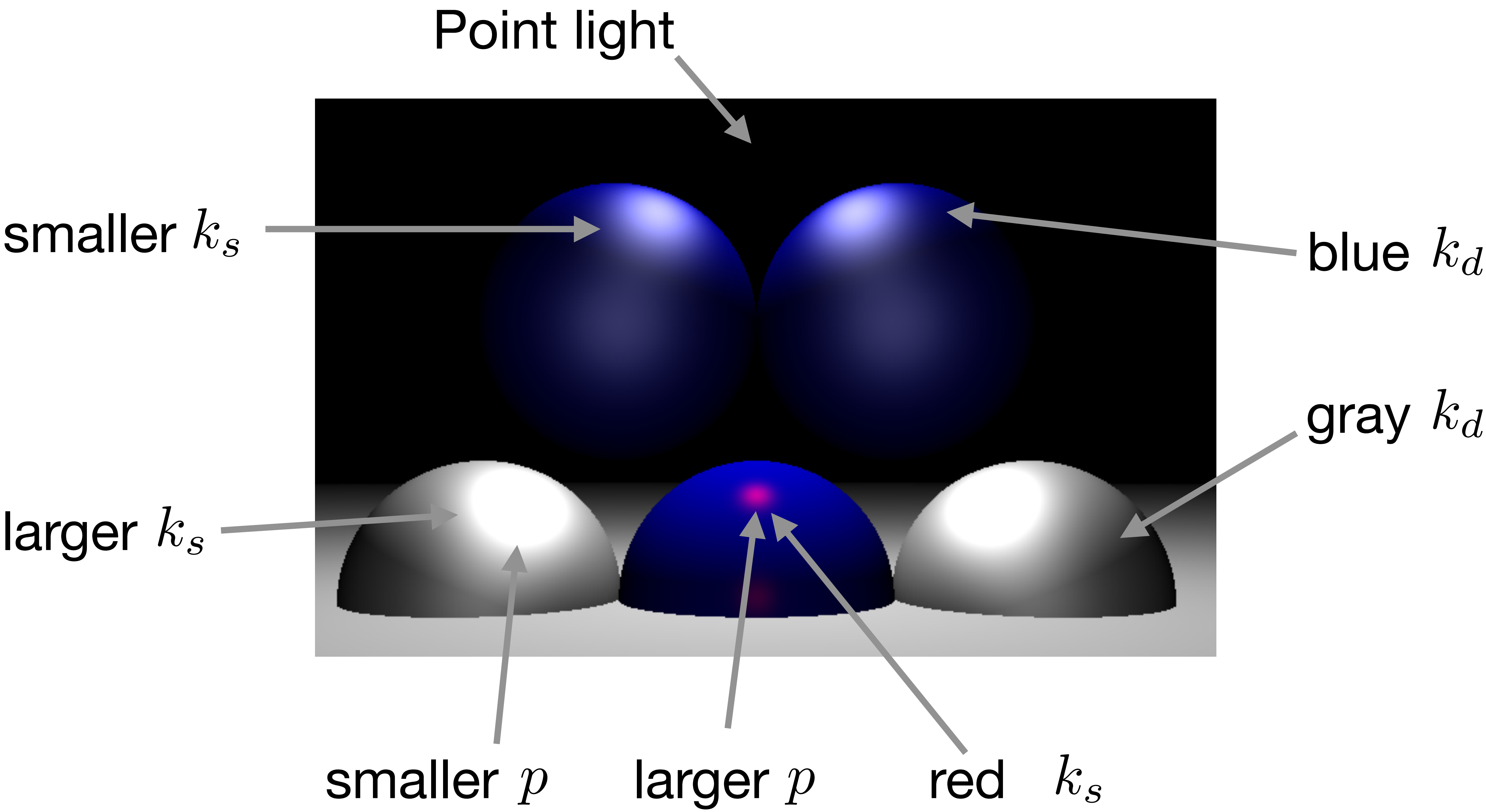


A disclaimer about point lights



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

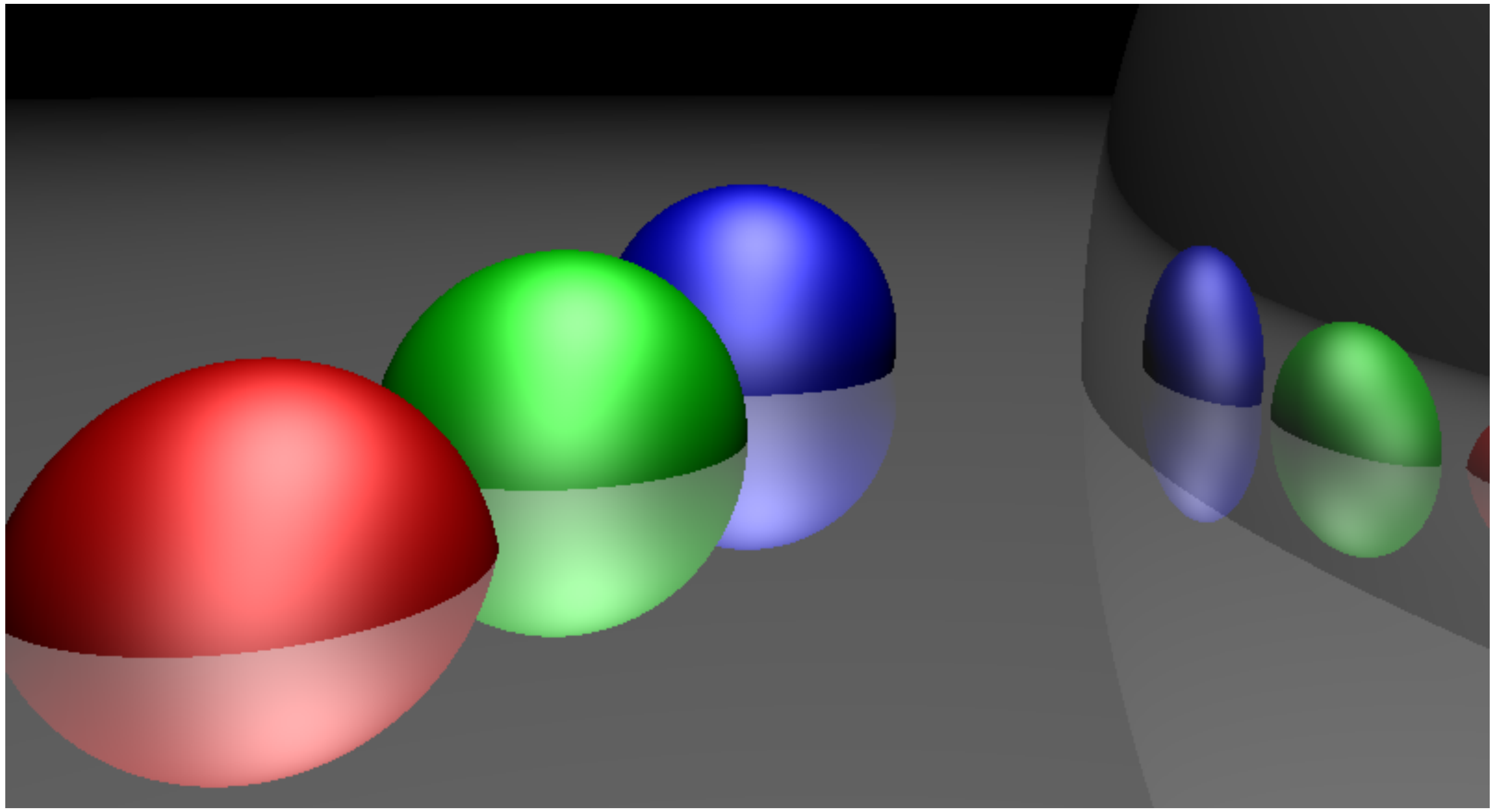
Our images so far:



Partially-Mirrored Surfaces

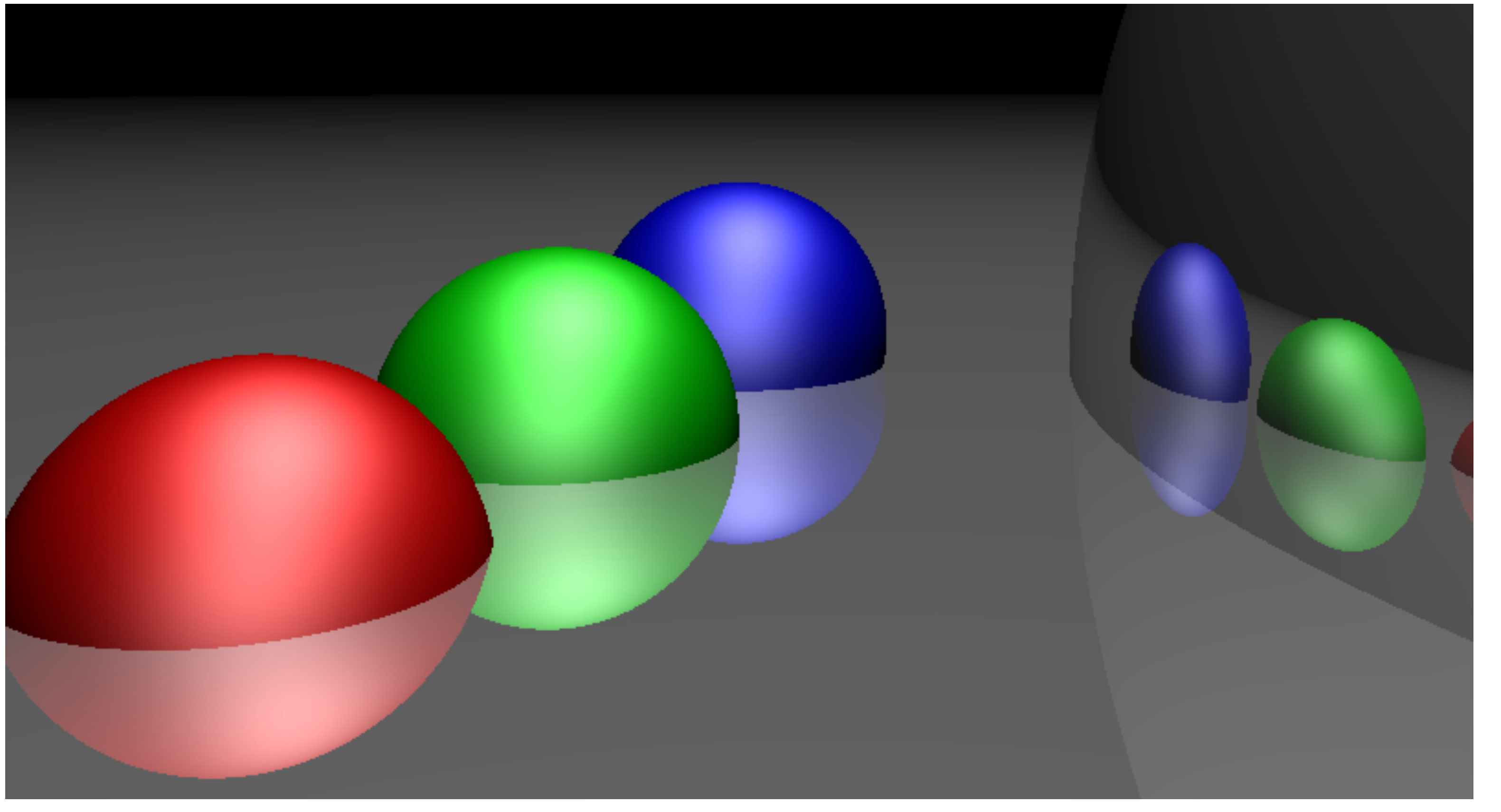
Notice the floor is gray but also mirror-reflective.

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.

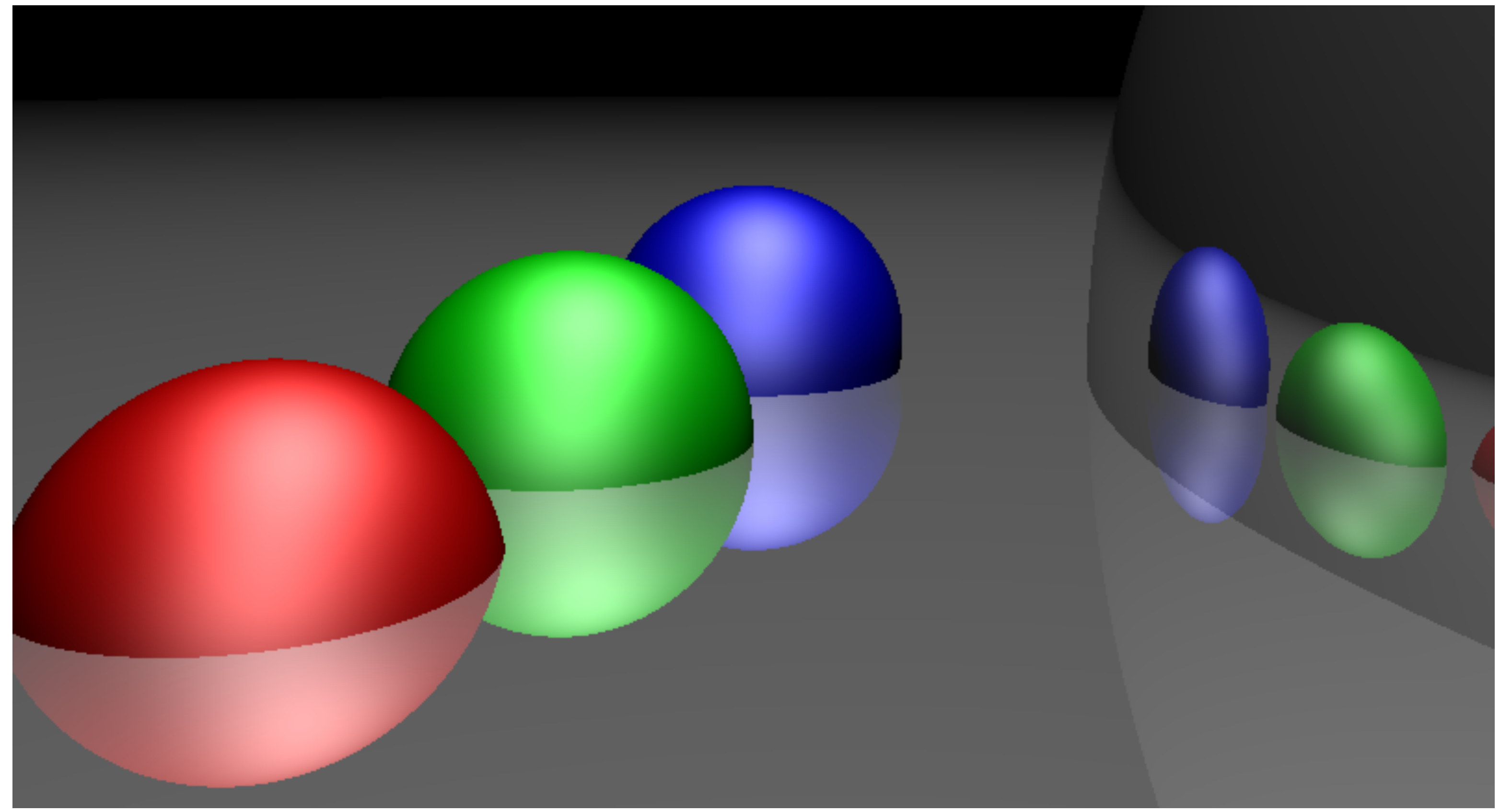


Materials can store a **mirror coefficient**: fraction of light that is reflected in a mirror-like fashion

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

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

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

k_m: mirror coefficient

L_r: mirror-reflected light

$L_d + L_s$: "local" color (Blinn-Phong)