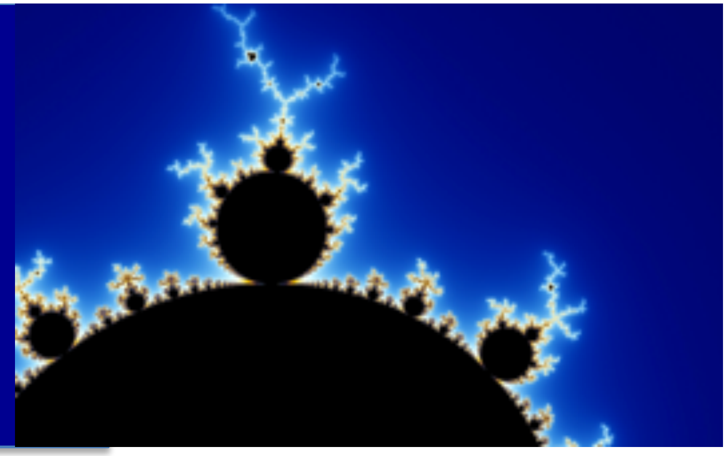
Computer Graphics



Lighting and Shading

Rendering Objects

- Now we want to make these objects look visually interesting, realistic, or both.
- We want to develop methods of **rendering** a picture of the objects of interest: *computing* how each pixel of a picture should look.

Rendering Objects (2)

- Much of rendering is based on different **shading models**, which describe how light from light sources interacts with objects in a scene.
  - It is impractical to simulate all of the physical principles of light scattering and reflection.
  - A number of approximate models have been invented that do a good job and produce various levels of realism.

What is light?

- Light is the most important idea behind visual representation of anything a human can visually perceive.
- What you see is not based on the objects that you are viewing but on the rays of light reflected from those objects.
- Your eyes don't directly see objects.
- There is no physical correlation between your eyes and those objects.

What is light?

- Light rays originate from an energy source (sun or lamp)
- Theoretically, a ray of light travels in a straight line.
- Your perception of an object comes from the light reflected or scattered off of an object that your eyes absorb.

What is light?

- Two rules:
  - Your eyes are mechanisms that perceive or absorb photons of light and not objects. You as a programmer, must simulate this functionality on the computer screen.
  - A ray of light travels in a straight line (not exactly true but we can think of it this way).

What is light?

- Albert Einstein in 1905 developed the theory of light:
  - He described the "photoelectric effect".
  - Described the activity of the ultraviolet light hitting a surface and emitting electrons off that surface.
  - This behavior was supported by an explanation that light was made up of a stream of energy packets called photons.

Color of light

- Light seen by human eye is a mixture of lights scattered and reflected against the surroundings of different material property.
- All physical matter is made up of atoms.
- Reflection of photons off of physical matter depends on
  - Kind of atoms
  - Amount of each kind
  - Arrangement of atoms in the object

Color of light

- Some photons are absorbed
- Some photons are reflected
- Color that the material reflects is observed as that material's color
- The more light the material reflects, the more shiny it will appear to the viewer.
- Each color is simply energy that can be represented by a wavelength.

Color of light

- Color is only a wavelength visible to the eye.
- A wavelength is measured by the distance between the peaks of the energy wave:

- Visible light is contained within the wavelengths ranging from 390 nanometers to 720 nanometers in length.

Color of light

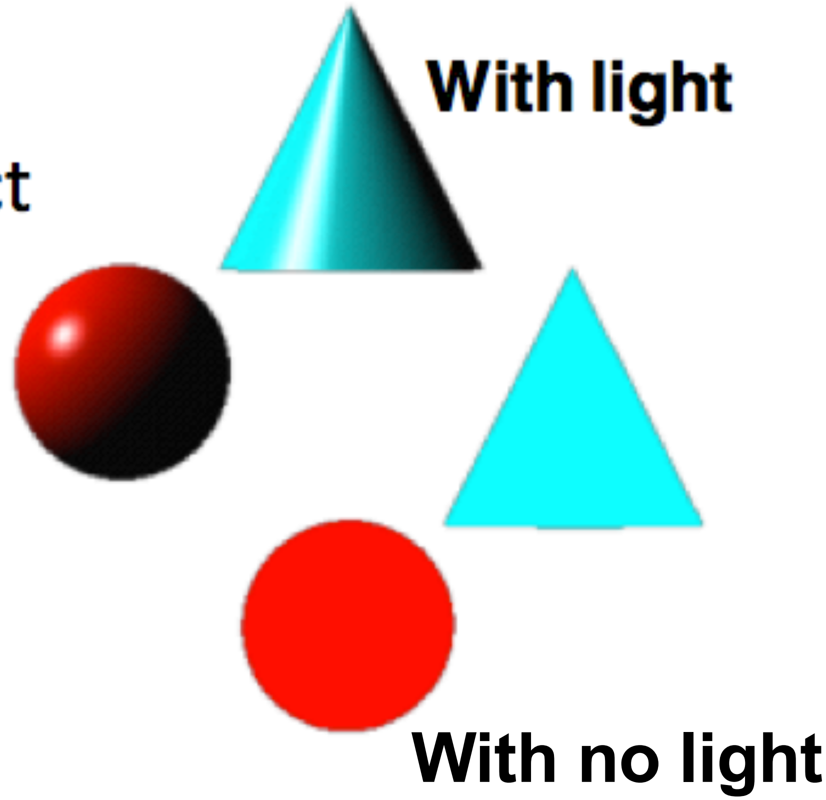
- 390 nanometers is the color violet
- 720 nanometers is the color red

- Everything in between is visible light and the range from 390 to 720 is called the color spectrum.

Lighting Principles

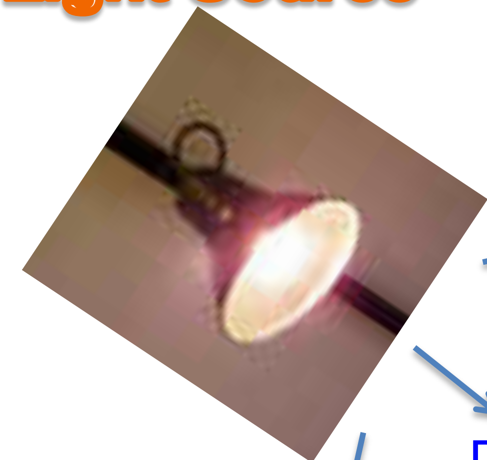
Lighting simulates how objects reflect light

- Lighting properties must be specified
- Light's color and position
- Material composition of object
- Global lighting parameters



Steps in Adding Lighting in WebGL

Light Source



Ambient
Light comp

GLOBAL FACTORS

Diffuse
Light comp

Specular
Light comp

Ambient
Light reflecting

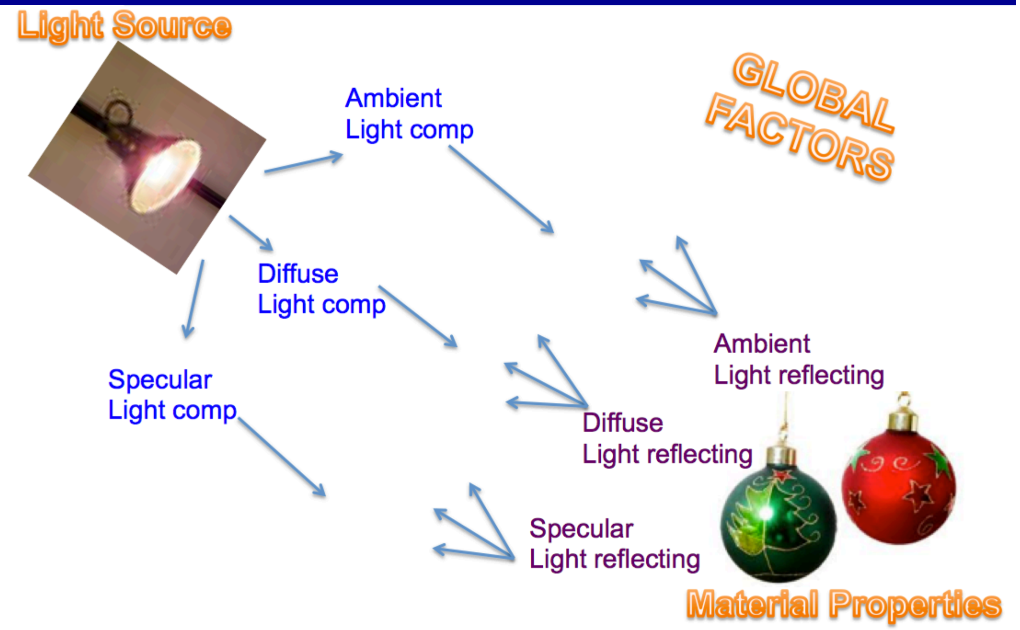
Diffuse
Light reflecting

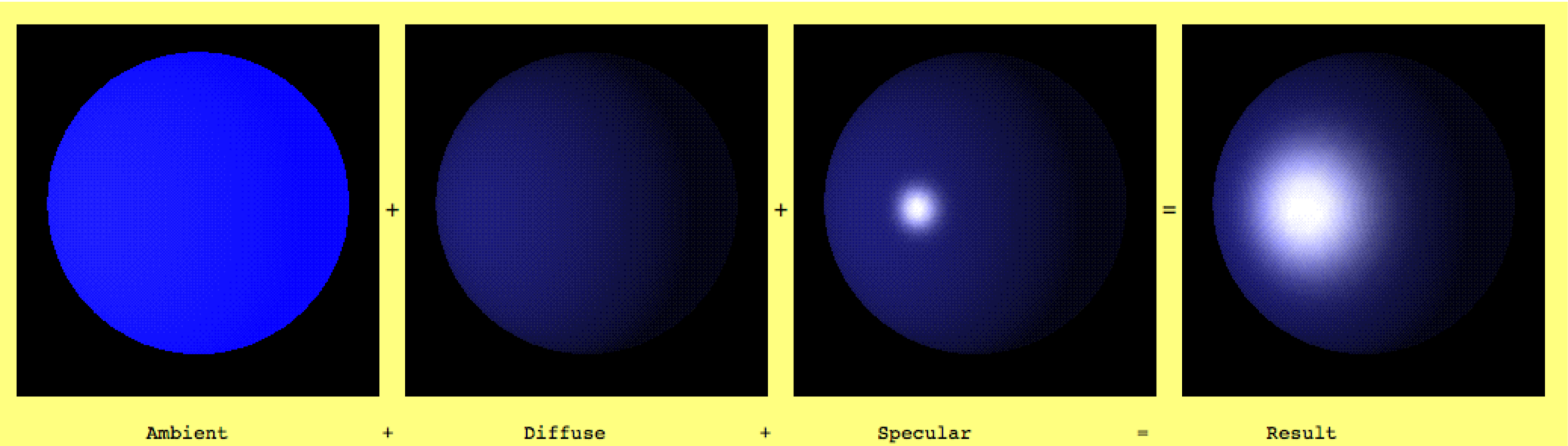
Specular
Light reflecting

Material Properties

How WebGL/OpenGL Simulates Lights

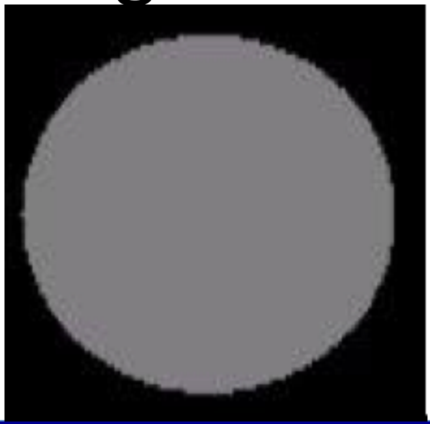
- Lighting contributors
  - Lighting properties
  - Surface material properties
  - Lighting model properties





Lighting Properties:

- Light comes from light sources that can be turned on/off
- ***Ambient*** lighting comes from light that is so scattered there is no way to tell its original location
- Ambient light is the average volume of light created from all light sources surrounding the lit area.
- Example: Backlighting in a room



3D sphere looks 2D

Lighting Properties:

- The ***diffuse*** component of lighting is light that comes from one direction.
- Diffuse light has a position in space and comes from a single direction.
- Once it hits a surface, it is scattered equally in all directions.
- Example: A flashlight

A diffuse red light cast onto a black object

Lighting Properties:

- To demonstrate how Ambient& Diffuse light works together:

Dark red ambient light only

Add diffuse light on the left side

Note sphere now looks 3D

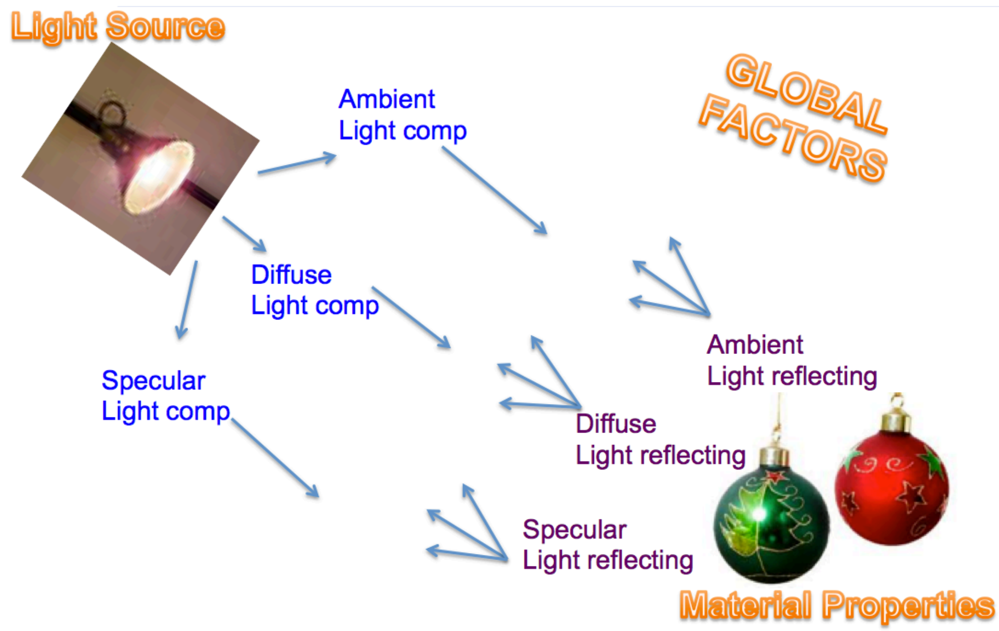
Lighting Properties:

- **Specular** light comes from a particular direction like a diffuse light.
- It bounces off the surface in a particular direction
- It relies on the angle between the viewer and the light source.
- Creates a highlighted area

RGB Values for Lights

- A light source is characterized by the amount of red, green, & blue light it emits.
- Examples: If R=G=B=1.0, the light is the brightest possible white.
- If R=G=B=.5, the color is still white, but only at half intensity, so it appears gray
- If R=G=1.0 and B=0.0, the light appears yellow.

Each light component is described by a light vector (R, G, B, A)

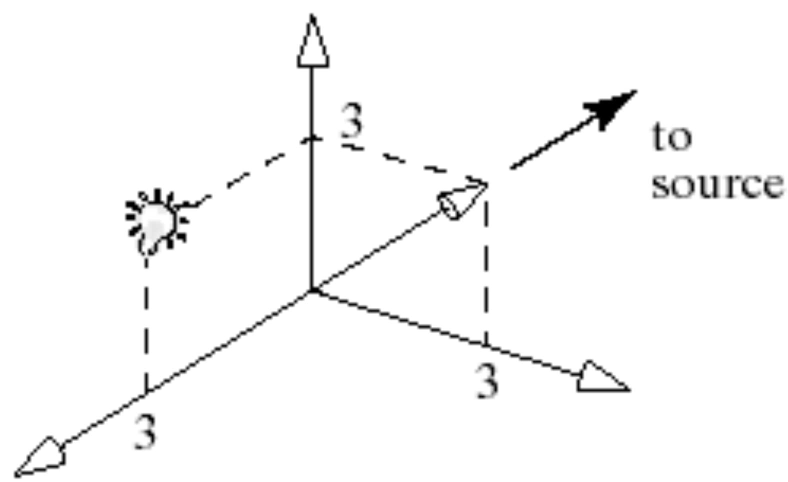


Types of Lights

- Two types of Lights
  - Local (Point) light source
    - Local light positioned at (x, y, z, 1)
  - Infinite (Directional) light sources
    - Infinite light directed along (x, y, z, 0)

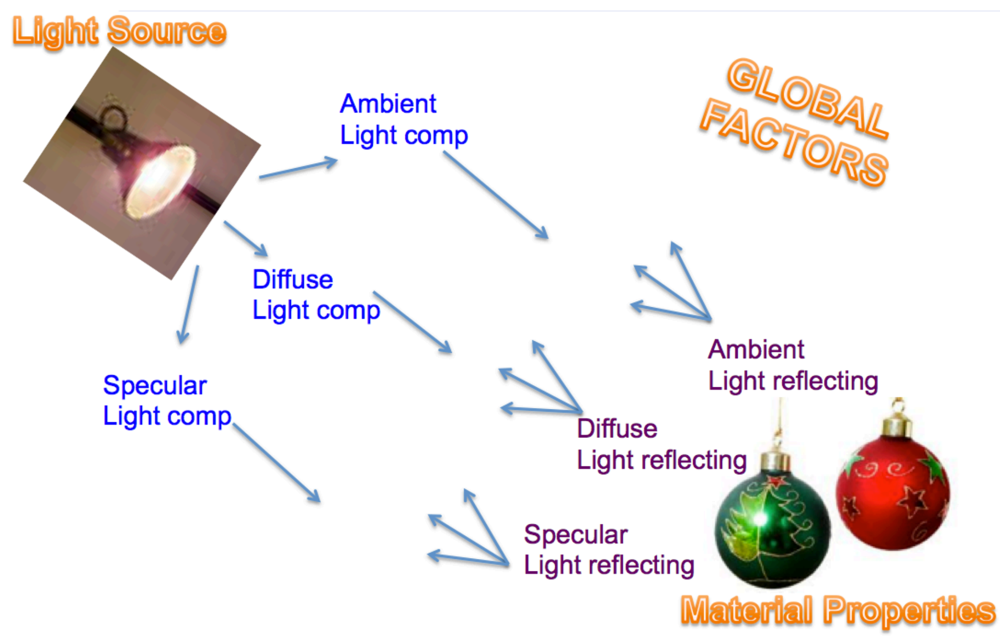
Point and Vector Light Locations

- The figure shows a local source at (0, 3, 3, 1) and a remote source "located" along vector (3, 3, 0, 0).
- Infinitely remote light sources are often called "**directional**". There are computational advantages to using directional light sources, since direction **s** in the calculations of diffuse and specular reflections is *constant* for all vertices in the scene.
- But directional light sources are not always the correct choice: some visual effects are properly achieved only when a light source is close to an object.



How WebGL/OpenGL Simulates Lights

- Lighting contributors
  - Lighting properties
  - Surface material properties
  - Lighting model properties



Steps in Adding Lighting in WebGL

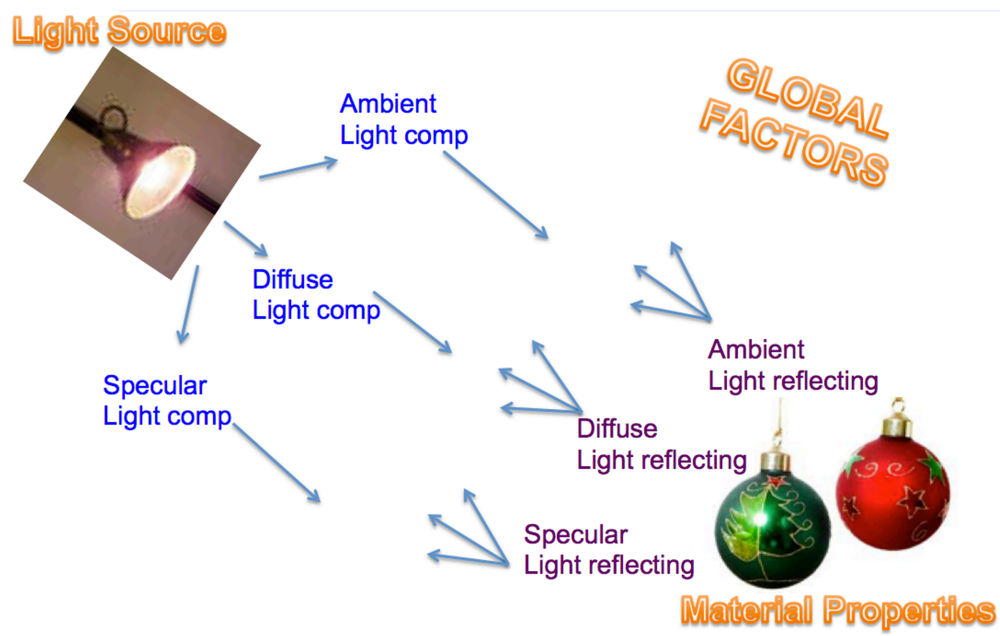
Establish vertex normal

1. Make sure the normal for each vertex has been defined
2. One normal for one vertex, or one normal for a group of vertices

<take a look at the example code>

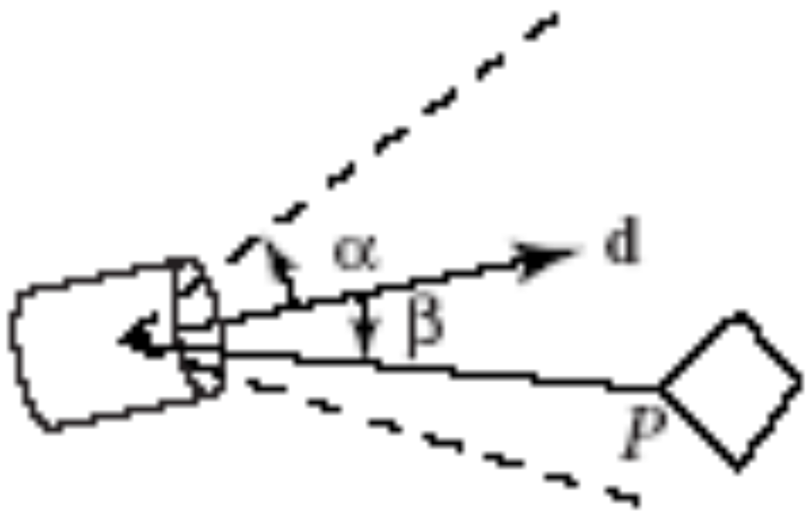
How WebGL/OpenGL Simulates Lights

- Lighting contributors
  - Lighting properties
  - Surface material properties
  - Lighting model properties (GLOBAL FACTORS)



Spotlights

- A spotlight emits light only in a cone of directions; there is no light outside the cone. Inside the cone, $I = I_s(\cos \beta)^{\varepsilon}$, where $\cos \beta$ uses the angle between **d** and a line from the source to *P*.




```
var d=[2, 2, 1, 0];
GL_SPOT_CUTOFF=45;
GL_SPOT_EXPONENT=4.0;
GL_SPOT_DIRECTION=d;
```

$\cos^{\phi}\beta$ (ϕ : exponent)

Spotlights in OpenGL (2)

- To create the spotlight, create a GLfloat array for **d**. Default values are **d** = {0, 0, 0, 0}, $\alpha = 180^o$, $\varepsilon = 0$: a point source.
- Then add the statements:

```
Glfloat d={2, 2, 2, 0};
glLightf (GL_LIGHT0, GL_SPOT_CUTOFF, 45.0); //45.0 is α in
    degrees)
glLightf (GL_LIGHT0, GL_SPOT_EXPONENT, 4.0); //4.0 is ε
glLightfv (GL_LIGHT0, GL_SPOT_DIRECTION, d);
```

Attenuation of Light with Distance

- You can specify how rapidly light diminishes with distance from a source.
- WebGL attenuates the strength of a positional light source by the following attenuation factor:

$$atten = \frac{1}{k_c + k_l D + k_q D^2}$$

where k_c, k_l, and k_q are coefficients and D is the distance between the light's position and the vertex in question.

Attenuation of Light with Distance (2)

$$atten = \frac{1}{k_c + k_l D + k_q D^2}$$

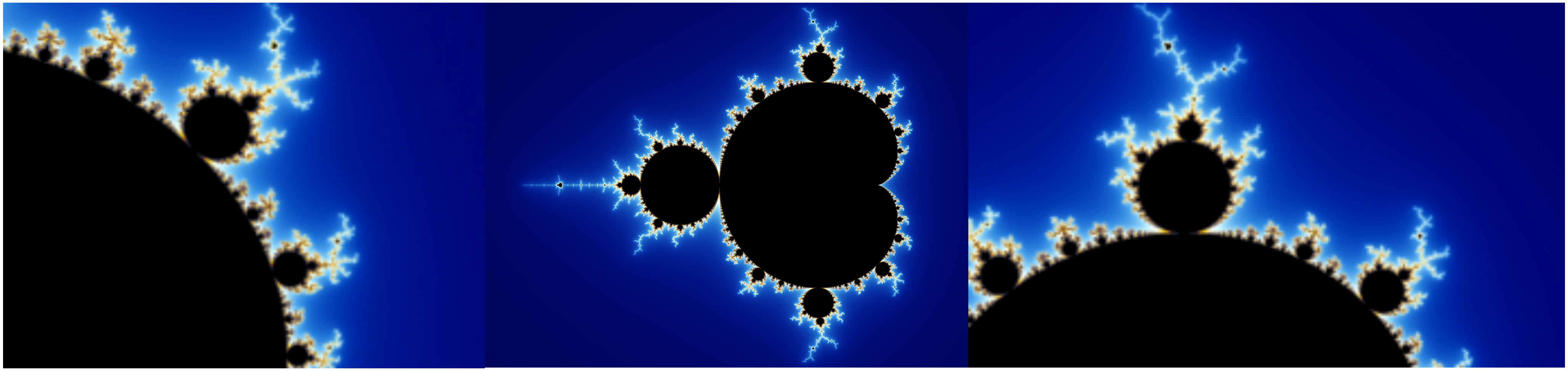
- These parameters are controlled by calling:

  GL_CONSTANT_ATTENUATION= 2.0;
  GL_LINEAR_ATTENUATION=2.0;
  GL_QUADRATIC_ATTENUATION=2.0;

The default values are $k_c = 1$, $k_l = 0$, and $k_q = 0$ (i.e., no attenuation).

WebGL lighting parameters

```
/* LightParameter */
#define GL_AMBIENT                  0x1200
#define GL_DIFFUSE                  0x1201
#define GL_SPECULAR                 0x1202
#define GL_POSITION                 0x1203
#define GL_SPOT_DIRECTION           0x1204
#define GL_SPOT_EXPONENT            0x1205
#define GL_SPOT_CUTOFF              0x1206
#define GL_CONSTANT_ATTENUATION     0x1207
#define GL_LINEAR_ATTENUATION       0x1208
#define GL_QUADRATIC_ATTENUATION    0x1209
```



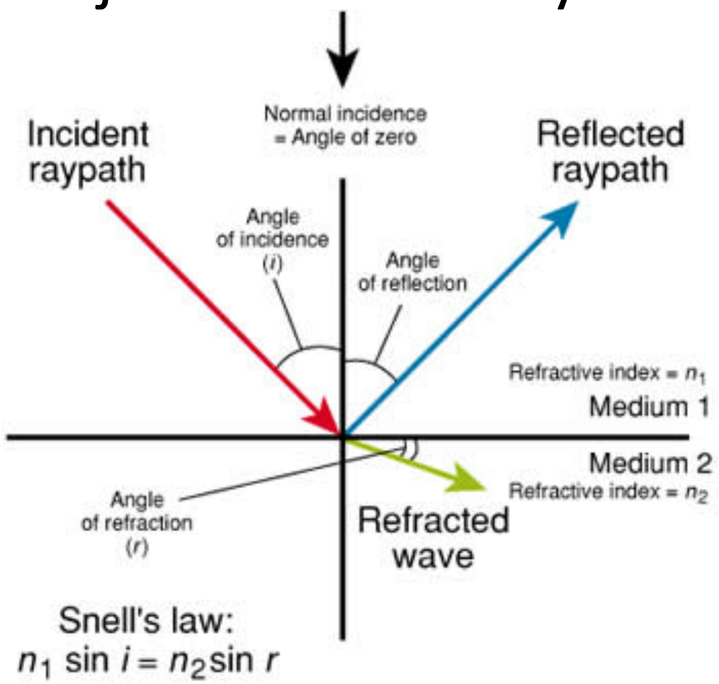
Shading Models

Shading Models: Introduction

- Assume to start with the light with no color, only brightness: R = G = B
- Assume we also have a point source of light (sun or lamp) and a general ambient light

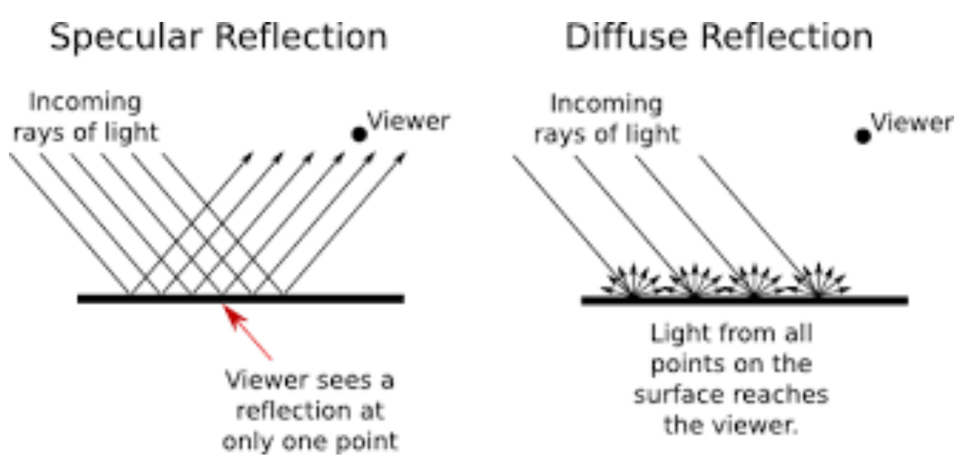
Shading Models: Introduction (2)

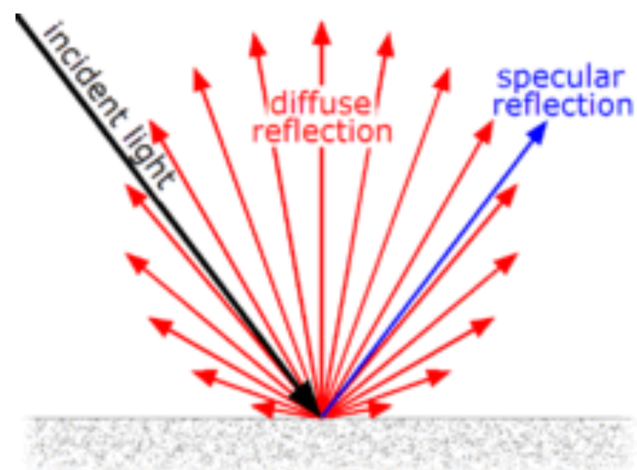
- When light hits an object, some light is absorbed (and turns into heat), some is reflected, and some may penetrate the interior (e.g., of a clear glass object).
- If all the light is absorbed, the object appears black and is called a blackbody.
- If all the light is transmitted, the object is visible only through refraction.



Shading Models: Introduction (3)

- When light is reflected from an object, some of the reflected light reaches our eyes, and we see the object.
  - **Diffuse** reflection: some of the light slightly penetrates the surface and is re-radiated uniformly in all directions.
  - **Specular** reflection: more mirror-like. Light is reflected directly from the object's outer surface, giving rise to highlights of approximately the same color as the source. The surface looks shiny.





Shading Models: Introduction (4)

- In the simplest model, specular reflected light has the same color as the incident light. This tends to make the material look like plastic.
- In a more complex model, the color of the specular light varies over the highlight, providing a better approximation to the shininess of metal surfaces.

Shading Models: Introduction (5)

- Most surfaces produce some combination of diffuse and specular reflection, depending on surface characteristics such as roughness and type of material.
- The total light reflected from the surface in a certain direction is the sum of the diffuse component and the specular component.
  - For each surface point of interest, we compute the size of each component that reaches the eye.
- Computation can be per vertex (in vertex shader), or per fragment (in fragment shader)

Lighting and Material Notations

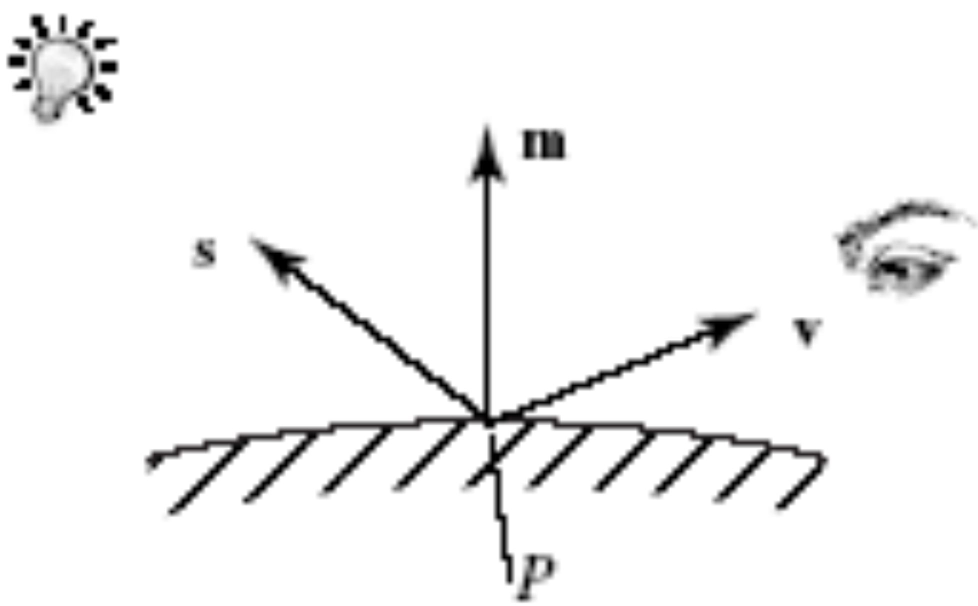
- We say the light sources have three types of color: ambient $I_a = (I_{ar}, I_{ag}, I_{ab})$, diffuse $I_d = (I_{dr}, I_{dg}, I_{db})$, and specular $I_s = (I_{spr}, I_{spg}, I_{spb})$.
- Similarly, each object material has three properties: ambient $\rho_a = (\rho_{ar}, \rho_{ag}, \rho_{ab})$, diffuse $\rho_d = (\rho_{dr}, \rho_{dg}, \rho_{db})$, and specular $\rho_s = (\rho_{spr}, \rho_{spg}, \rho_{spb})$
  - Each object's color (in white light) is specified by 9 coefficients :
    - ambient reflection coefficients: ρ_{ar}, ρ_{ag}, and $\rho_{ab;}$
    - diffuse reflection coefficients: ρ_{dr}, ρ_{dg}, and ρ_{db}
    - specular reflection coefficients: ρ_{sr}, ρ_{sg}, and ρ_{sb}.

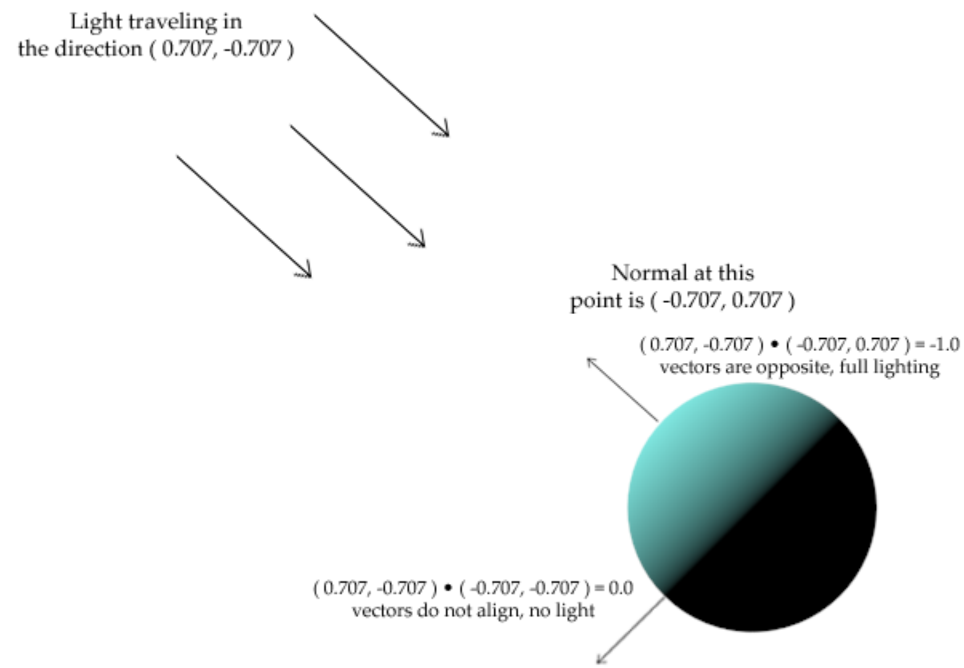
Reflected Light Model

- Finding Reflected Light: a model
  - Model is not completely physically correct, but it provides fast and relatively good results on the screen.
  - Intensity of a light is related to its brightness. We will use I_s for intensity, where s is R or G or B.

Calculating Reflected Light

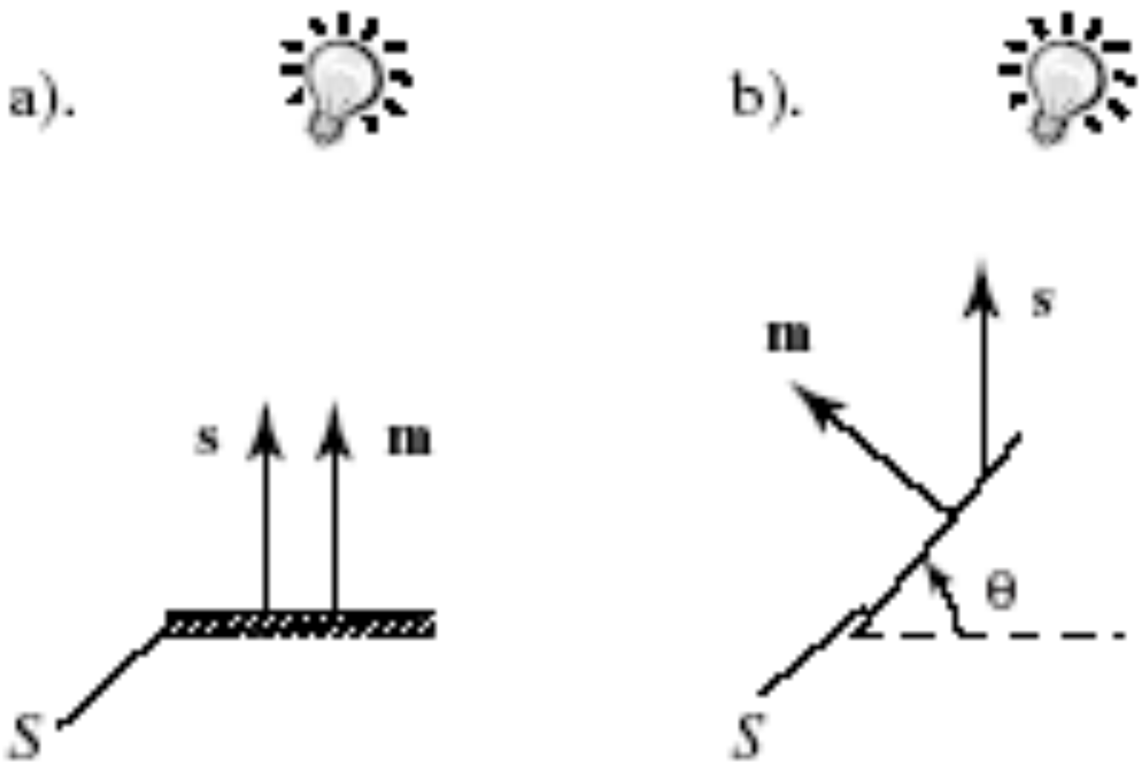
- To compute reflected light at point P, we need 3 vectors: normal **m** to the surface at P and vectors **s** from P to the source and **v** from P to the eye. We use world coordinates.





Calculating Diffuse Light (2)

- The intensity depends on the projection of the face perpendicular to **s** (Lambert's law): $I_d \cos\theta$



Calculating Diffuse Light

- Diffuse scattering is assumed to be independent of the direction from the point, *P*, to the location of the viewer's eye.
- Because the scattering is uniform in all directions, the orientation of the face *F* relative to the eye is not significant, and I_d is independent of the angle between **m** and **v** (unless $\mathbf{v} \cdot \mathbf{m} < 0$, making $I_d = 0$.)
- The amount of light that illuminates the face *does* depend on the orientation of the face relative to the point source, s and m: the amount of light is proportional to the area of the face that it sees: the area *subtended* by a face.

Calculating Diffuse Light (3)

- For θ near 0^o, brightness varies only slightly with angle, because the cosine changes slowly there. As θ approaches 90^o, the brightness falls rapidly to 0.
- We know $\cos\theta = (\mathbf{s}\cdot\mathbf{m})/(|\mathbf{s}||\mathbf{m}|)$.
- $I_d = I_d \rho_d (\mathbf{s}\cdot\mathbf{m})/(|\mathbf{s}||\mathbf{m}|)$.
  - I_d is the intensity of the source.
  - ρ_d is the diffuse reflection coefficient and depends on the material the object is made of.
- $\mathbf{s}\cdot\mathbf{m} < 0$ implies $I_d = 0$.
- Diffusing light model (Lambert model):

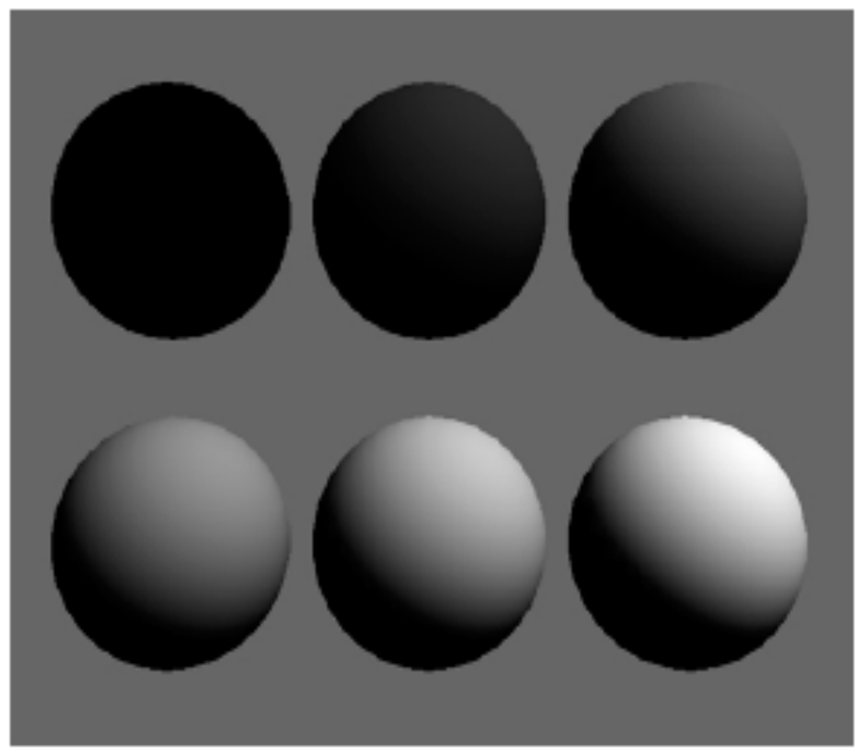
$$I_d = I_d \rho_d \max[\cos\theta, 0],$$

or $$I_d = I_d \rho_d \max[(\mathbf{s}\cdot\mathbf{m})/(|\mathbf{s}||\mathbf{m}|), 0]$$

Example: Spheres Illuminated with Diffuse Light.

- Spheres with reflection coefficients from 0 to 1 by 0.2.
- The source intensity is 1.0. Background intensity is 0.4.
- The sphere is totally black when $\rho_d = 0.0$, and the shadow in its bottom half (where $(\mathbf{s} \cdot \mathbf{m})/(|\mathbf{s}||\mathbf{m}|) < 0$) is also black.

I_d=1.0,
I_a=0.4
P_d=0, 0.2, 0.4,
0.6, 0.8, 1.0



Calculating Diffuse Light (4)

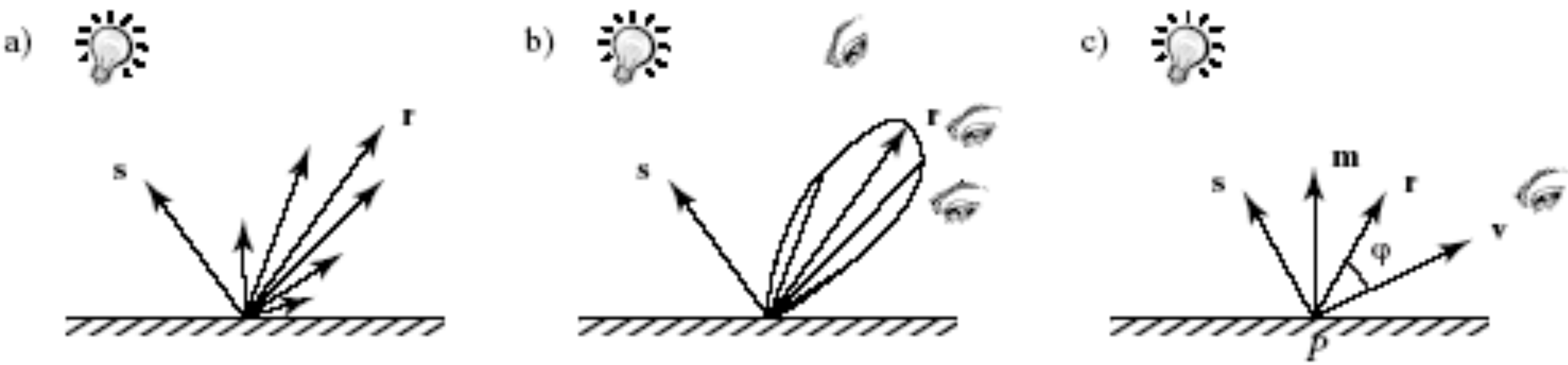
- The light intensity falling on facet *S* from the point source is known to decrease as the inverse square of the distance between *S* and the source. We could try to incorporate this in our model.
- Experiments show that using this law yields pictures with exaggerated depth effects.
- Also, we sometimes model light sources as if they lie "at infinity". Using an inverse square law in such a case would quench the light entirely.
- So we ignore distance effects in calculating light intensity.

Calculating the Specular Component

- Real objects do not scatter light uniformly in all directions; a specular component is added to the shading model.
- Specular reflection causes highlights, which can add significantly to realism of a picture when objects are shiny.
- A simple model for specular light was developed by Phong. It is easy to apply.
  - The highlights generated by the Phong model give an object a plastic-like or glass-like appearance.

Calculating the Specular Component (2)

- Most of the light reflects at equal angles from the (smooth and/or shiny) surface, along direction **r,** the reflected direction.
- So, specular light is mainly determined by the angle between the viewing direction **v** and the reflected light direction **r**



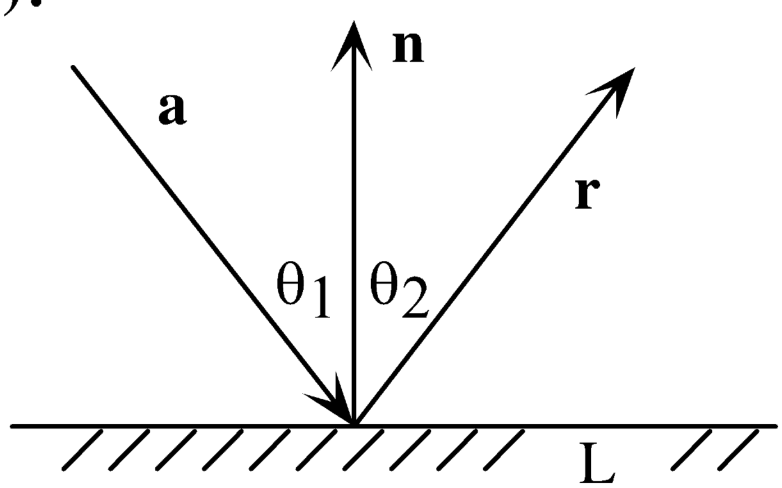
Reflections

- When light reflects from a mirror, the angle of reflection must equal the angle of incidence:

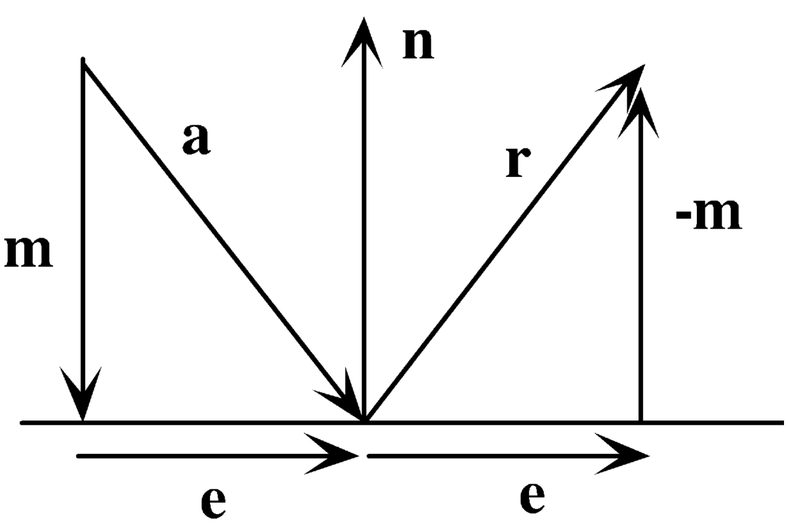
  $\theta_1 = \theta_2$.

- Vectors and projections allow us to compute the new direction **r**, in either two-dimensions or three dimensions.

a).



b).



Reflection – dot product

- The illustration shows

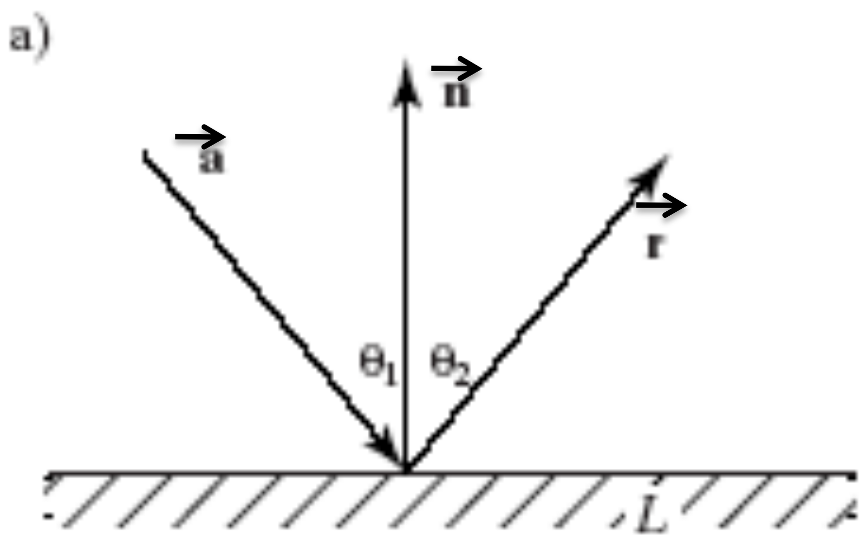
  $\vec{\mathbf{e}} = \vec{\mathbf{a}} - \vec{\mathbf{m}}$

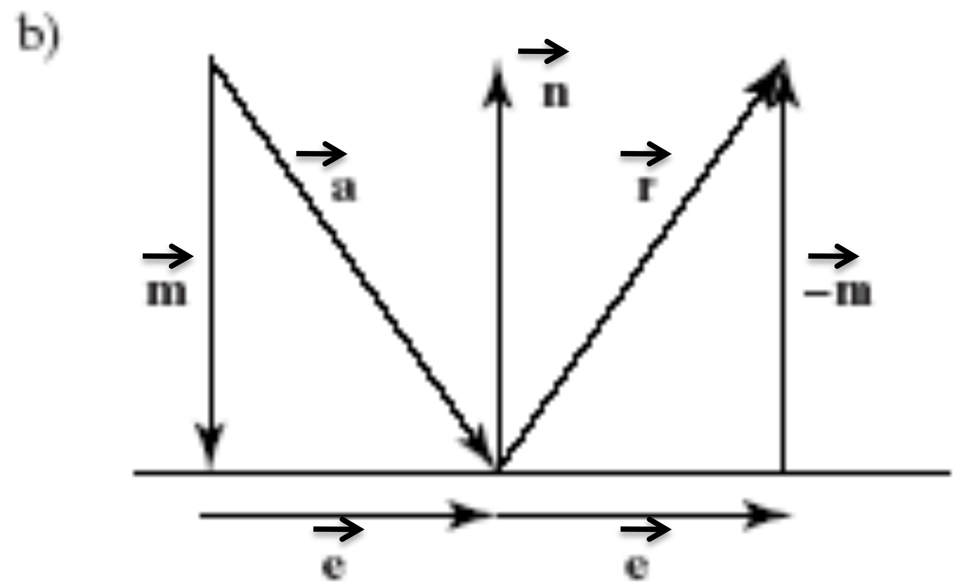
  $\vec{\mathbf{r}} = \vec{\mathbf{e}} - \vec{\mathbf{m}} = \vec{\mathbf{a}} - 2\vec{\mathbf{m}}$

  $\vec{\mathbf{m}} = [(\vec{\mathbf{a}}\cdot\vec{\mathbf{n}})/|\vec{\mathbf{n}}|^2]\vec{\mathbf{n}} = (\vec{a}\cdot\hat{n})\cdot\ \hat{n}$

$$\vec{r} = \vec{a} - 2\left(\frac{\vec{a}\cdot\vec{n}}{|\vec{n}|^2}\right)\vec{n}$$

$$= \vec{a} - 2(\vec{a}\cdot\hat{n})\cdot\hat{n}$$



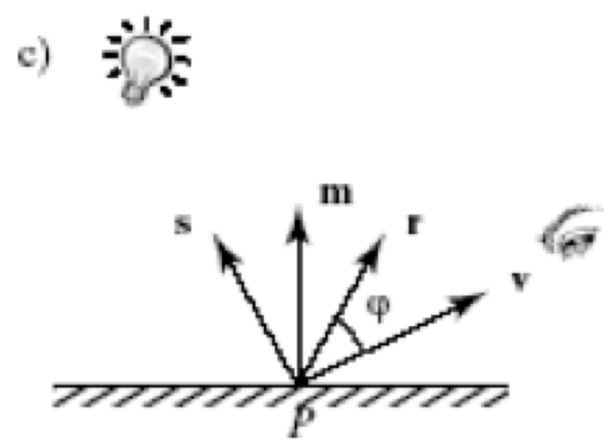


Calculating the Specular Component (2)

- An exact calculation of the reflected light r:

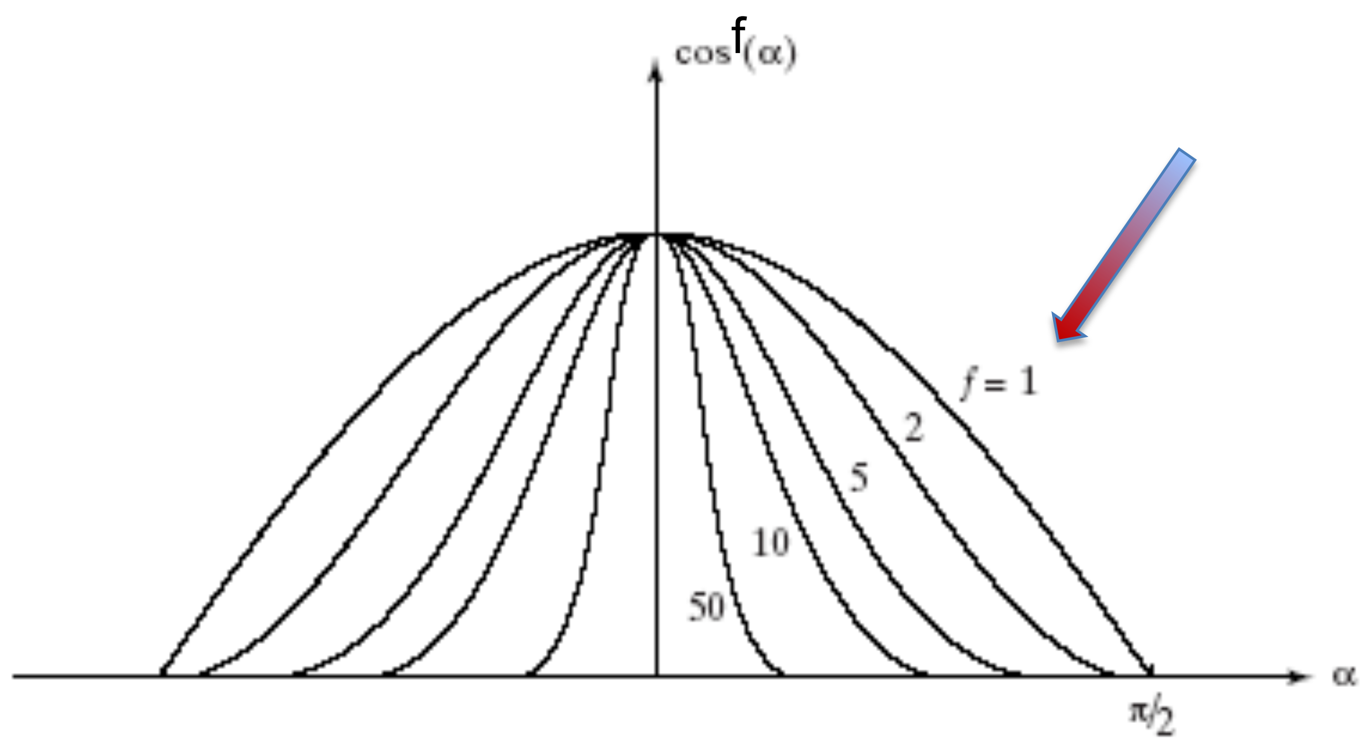
  $\mathbf{r} = \mathbf{a} - 2\,\mathbf{n}\,(\mathbf{a}\cdot\mathbf{n})/(|\mathbf{n}|^2)$ (mirror reflection direction).

- For surfaces that are not mirrors, the amount of reflected light decreases as the angle ϕ between **r** and **v** (vector from reflection point to eye) increases.
- For a simplified model, we say the intensity decreases as $\cos^f \phi$, where **f** is chosen experimentally between 1 and 100.



Calculating the Specular Component (3)

- The effect of f value: large f values give concentrated highlights; smaller ones give larger dimmer highlights.



Calculating the Specular Component (4)

- $\cos\phi = \mathbf{r}\cdot\mathbf{v}/(|\mathbf{r}||\mathbf{v}|)$
- $I_{sp} = I_s\, \rho_s\, (\mathbf{r}\cdot\mathbf{v}/(|\mathbf{r}||\mathbf{v}|))^f$.
  - ρ_s is the specular reflection coefficient, which depends on the material.
- If $\mathbf{r}\cdot\mathbf{v} < 0$, there is no reflected specular light.
- Specular light model (Phone model):

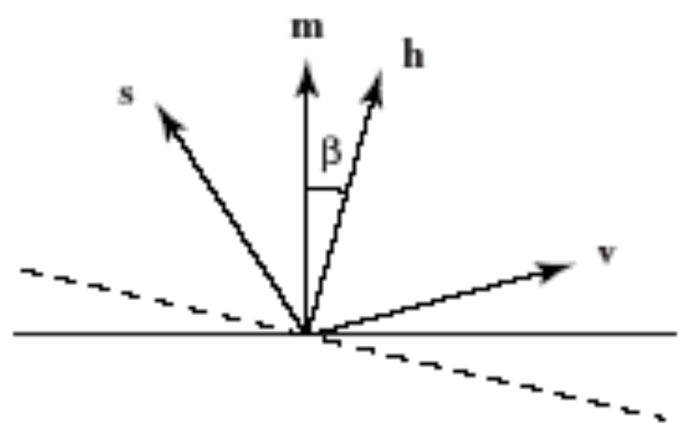
$$I_{sp} = I_s\, \rho_s \max[\cos^f \phi, 0], \text{ or}$$

$$I_{sp} = I_s\, \rho_s \max[(\mathbf{r}\cdot\mathbf{v}/(|\mathbf{r}||\mathbf{v}|))^f, 0]$$

Speeding up Calculations for Specular Light

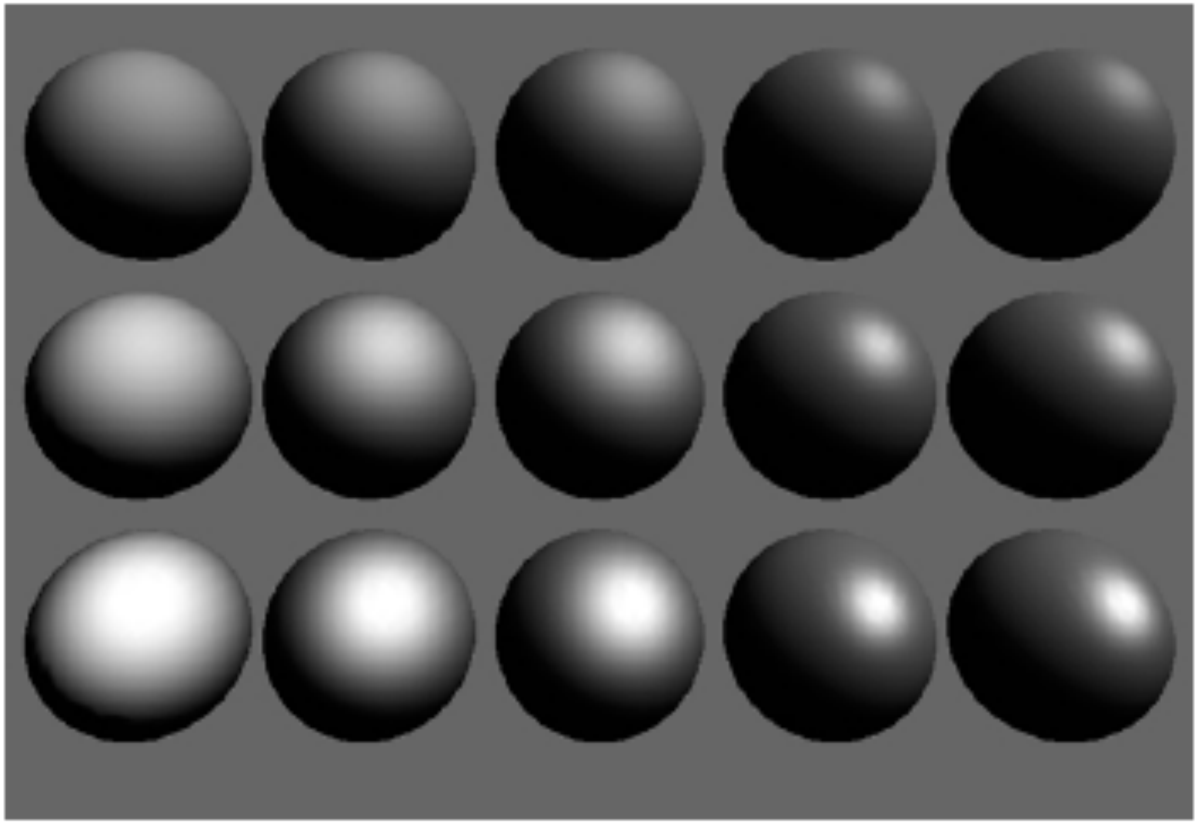
- Find the halfway vector **h** = **s** + **v**.
- Then the angle β between **h** and **m** approximately measures the falloff of intensity:

$I_{sp} = I_s \rho_s \max[(\mathbf{h}\cdot\mathbf{m}/(|\mathbf{h}||\mathbf{m}|))^f, 0]$



Calculating the Specular Component (5)

- From bottom to top, ρ_s = 0.75, 0.5, 0.25. From left to right, f = 3, 6, 9, 25, 128.
- ρ_a= 0.1
- ρ_d= 0.4



ρ_s = 0.25

ρ_s = 0.5

ρ_s = 0.75

Ambient Light

- Our desire for a simple reflection model leaves us with far from perfect renderings of a scene.
  - E.g., shadows appear to be unrealistically deep and harsh.
- To soften these shadows, we can add a third light component called *ambient light*.

Ambient Light (2)

- The scenes we observe around us always seem to be bathed in some soft non-directional light.
- This light arrives by multiple reflections from various objects in the surroundings and from light sources that populate the environment, such as light coming through a window, fluorescent lamps, and the like.
- We assume a uniform background glow called **ambient light** exists in the environment.
- This ambient light source is not situated at any particular place, and it spreads in all directions uniformly.

Calculating Ambient Light

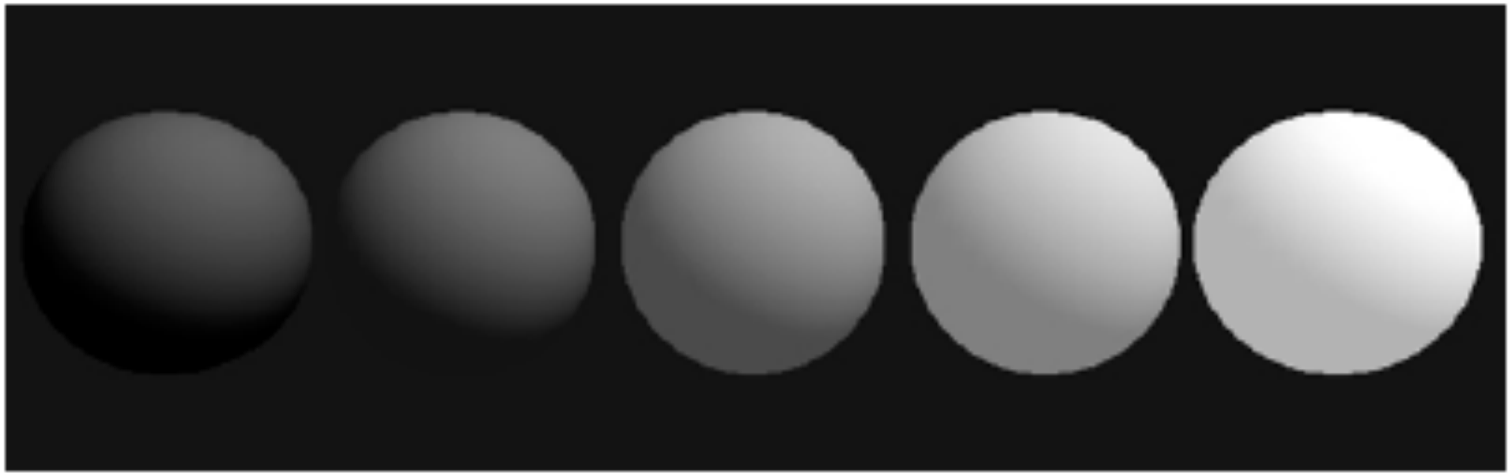
- The source is assigned an intensity, I_a.
- Each face in the model is assigned a value for its **ambient reflection coefficient**, ρ_a (often this is the same as the diffuse reflection coefficient, ρ_d), and the term $I_a \rho_a$ is simply added to whatever diffuse and specular light is reaching the eye from each point *P* on that face.
- Ambient light model:

$$\boxed{I_a = I_a \rho_a}$$

- I_a and ρ_a are usually arrived at experimentally, by trying various values and seeing what looks best.

Adding Ambient Light to Diffuse Reflection

- The diffuse and ambient sources have intensity 1.0, and $\rho_d = 0.4$. $\rho_a = 0, 0.1, 0.3, 0.5, 0.7$ (left to right).
- Modest ambient light softens shadows; too much ambient light washes out shadows.



Combining Light Contributions and Adding Color

- Model combining ambient, diffuse, and specular light:

$$\boxed{I_a\, \rho_a + I_s\, \rho_d\, \text{lambert} + I_{sp}\, \rho_s \times \text{phong}^f}$$

where Lambert = max[($\mathbf{s}\cdot\mathbf{m}$)/(|$\mathbf{s}$||$\mathbf{m}$|), 0] and Phong = max[($\mathbf{h}\cdot\mathbf{m}$/(|$\mathbf{h}$||$\mathbf{m}$|), 0]

- To add color, we use 3 separate total intensities like that above, one each for Red, Green, and Blue, which combine to give any desired color of light.

Combining Light Contributions and Adding Color (2)

- Generally use white light
  - Specular, diffuse and specular components
- Each object's color (in white light) is specified by 9 coefficients :
  - ambient reflection coefficients: ρ_{ar}, ρ_{ag}, and $\rho_{ab;}$
  - diffuse reflection coefficients: ρ_{dr}, ρ_{dg}, and ρ_{db}
  - specular reflection coefficients: ρ_{sr}, ρ_{sg}, and ρ_{sb}
- Specify object material :
  - Diffuse light properties determines the color of the object
  - Specular light properties having the same color as diffuse light.
  - Use light ambient light material with the same color as diffuse light properties

Example

- If the color of a sphere is 30% red, 45% green, and 25% blue, it makes sense to set its ambient and diffuse reflection coefficients to (0.3*K*, 0.45*K*, 0.25*K*) respectively, where *K* is some scaling value that determines the overall fraction of incident light that is reflected from the sphere.
- Now if it is bathed in white light having equal amounts of red, green, and blue ($I_{sr} = I_{sg} = I_{sb} = I$) the individual diffuse components have intensities $I_r = 0.3\,K\,I$, $I_g = 0.45\,K\,I$, $I_b = 0.25\,K\,I$, so as expected we see a color that is 30% red, 45% green, and 25% blue.

Example (2)

- Suppose a sphere has ambient and diffuse reflection coefficients (0.8, 0.2, 0.1), so it appears mostly red when bathed in white light.
- If we illuminate it with a greenish light I_s = (0.15, 0.7, 0.15).
  - The color of the object is then: (0.12, 0.14, 0.015), which is a fairly even mix of red and green, and would appear yellowish.
  - 0.12 = 0.8 x 0.15, 0.14 = 0.2 x 0.7, 0.015 = 0.1 x 0.15

Example (3)

- Because specular light is mirror-like, the color of the specular component is often set the same as that of the light source.
  - E.g., the specular highlight seen on a glossy red apple when illuminated by a yellow light is yellow rather than red.
- To create specular highlights for a plastic-like surface, set the specular coefficients $\rho_{sr} = \rho_{sg} = \rho_{sb} = \rho_s$, so that they are 'gray' in nature and do not alter the color of the incident light.
- The designer might choose $\rho_s = 0.5$ for a slightly shiny plastic surface, or $\rho_s = 0.9$ for a highly shiny surface.

Practice Question (1)

- Given the light specification as $I_d = I_s = I_a = (1, 1, 1)$, and the object material property specification as $\rho_d = (0.5, 0, 0.5)$, $\rho_s = (0.5, 0.5, 0.5)$, $\rho_a = (0.5, 0, 0.5)$, what is the color of the object, and what will the color of the object and the highlight of the object be displayed as?

Practice Question (2)

- Given the light specification as $I_d = I_a = (0.8, 0.8, 0.8)$ and $I_s = (0.5, 0.2, 0)$; the object material property specification as $\rho_d = (0, 0, 0.5)$, $\rho_s = (0.2, 0.5, 0)$, $\rho_a = (0, 0, 0.5)$, what will the color of the object and the highlight of the object be displayed as?

Practice Question (3)

- Given the light specification as $I_d = I_a = (0.8, 0, 0)$ and $I_s = (0.5, 0.5, 0.5)$; the object material property specification as $\rho_d = (1, 0, 0.5)$, $\rho_s = (0.9, 0.9, 0.9)$, $\rho_a = (.5, 0.5, 0.5)$, what will the color of the object and the highlight of the object be displayed as?