CS3388B: Lecture 13

March 16, 2023

13 Lighting

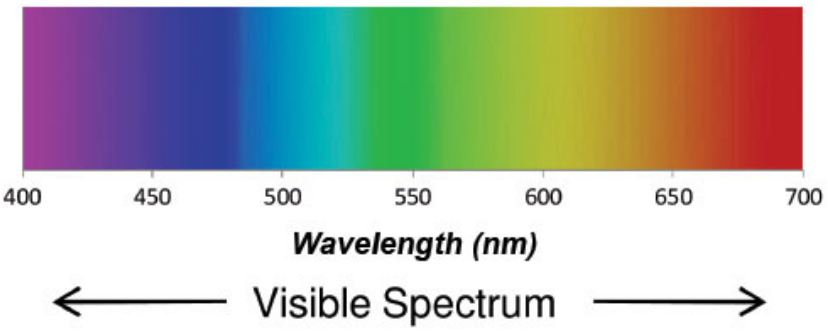
Light is a natural part of imagery. We need light to be able to see things. Thus, **lighting** is crucial in graphics for making believable computer graphics. Sometimes lighting can be faked with good textures and texture mapping. Most often, however, we need to do some computation. That means more linear algebra.

13.1 Lights, EM waves

In physics, electromagnetic waves and light are well-studied. From there we get terminology like *irradiance*, *radiance*, *reflection*, *refraction*, *albedo*, among others.

- **Irradiance** is a measure of the amount of light (radiant flux) received by a surface per unit area.
- **Radiance** is a measure of the amount of light emitted, reflected, and transmitted by a surface per unit area.
- **Reflection** is a change in direction of light caused by the boundary between two volumes such that the light returns to the originating volume.
- **Refraction** is a change in direction of light caused by the boundary between two volumes such that the light continues to move out of the originating volume.
- **Albedo** is a measure of the amount of light reflected vs the amount of light received by an object. In computer graphics, the term albedo is slightly abused. In graphics, Albedo refers to the "base color" of an object.

You may recall from grade school science that **white light** is the combination of every color (every wavelength) of light. Thanks Newton!

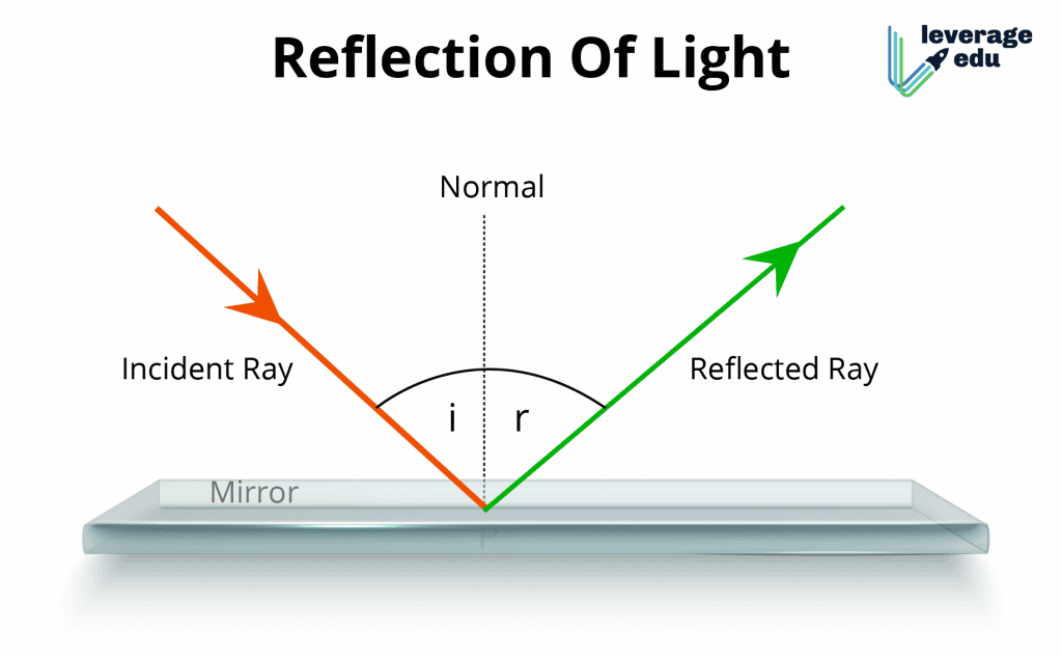


Then, the color we perceive of an object is based on the amount of light it **absorbs**. The color we see corresponds to the wavelengths of lights which are *not absorbed*. That is, the color of an object is the combination of wavelengths reflected. This is where albedo is used in graphics to mean "base color". It describes which colors are reflected from the surface and thus which colors are seen from the object.

13.2 Some Math, BRDF

Time for some linear algebra.

Given some normal vector and some incident ray, we can compute its reflection, of course. This kind of reflection is called **regular reflection** or **specular reflection**. They key is that the angle of reflection r is equal to angle of incidence i.



How do we get angles on vectors? Dot products.

$$\frac{\vec{i} \cdot \vec{n}}{|i|\ |n|} = \frac{\vec{r} \cdot \vec{n}}{|r|\ |n|}$$

Lucky for us, GLSL has a `reflect` function to compute this for us eventually.

Now, most surfaces are not mirrors. That is, the reflection is not perfect, as is shown in the previous image. Sometimes light is absorbed by the surface. Sometimes surfaces are shiny, sometimes they are matte. These all must be included in real life.

To create such **photorealistic** rendering, we need not just perfect reflection but so-called **physically based rendering**. This requires solving the **rendering equation**.

From wikipedia `https://en.wikipedia.org/wiki/Rendering_equation`:

$$L_{\mathrm{o}}(\mathbf{x}, \omega_{\mathrm{o}}, \lambda, t) = L_{\mathrm{e}}(\mathbf{x}, \omega_{\mathrm{o}}, \lambda, t) \; + \int_{\Omega} f_{\mathrm{r}}(\mathbf{x}, \omega_{\mathrm{i}}, \omega_{\mathrm{o}}, \lambda, t) L_{\mathrm{i}}(\mathbf{x}, \omega_{\mathrm{i}}, \lambda, t)(\omega_{\mathrm{i}} \cdot \mathbf{n}) \, \mathrm{d}\, \omega_{\mathrm{i}}$$

where

- $L_{\mathrm{o}}(\mathbf{x}, \omega_{\mathrm{o}}, \lambda, t)$ is the total spectral radiance of wavelength λ directed outward along direction ω_{o} at time t, from a particular position $\mathbf{x}$
- $\mathbf{x}$ is the location in space
- ω_{o} is the direction of the outgoing light
- λ is a particular wavelength of light
- t is time
- $L_{\mathrm{e}}(\mathbf{x}, \omega_{\mathrm{o}}, \lambda, t)$ is emitted spectral radiance
- $\int_{\Omega} \ldots \mathrm{d}\, \omega_{\mathrm{i}}$ is an integral over Ω
- Ω is the unit hemisphere centered around $\mathbf{n}$ containing all possible values for ω_{i} where $\omega_{\mathrm{i}} \cdot \mathbf{n} > 0$
- $f_{\mathrm{r}}(\mathbf{x}, \omega_{\mathrm{i}}, \omega_{\mathrm{o}}, \lambda, t)$ is the bidirectional reflectance distribution function, the proportion of light reflected from ω_{i} to ω_{o} at position $\mathbf{x}$, time t, and at wavelength λ
- ω_{i} is the negative direction of the incoming light
- $L_{\mathrm{i}}(\mathbf{x}, \omega_{\mathrm{i}}, \lambda, t)$ is spectral radiance of wavelength λ coming inward toward $\mathbf{x}$ from direction ω_{i} at time t
- $\mathbf{n}$ is the surface normal at $\mathbf{x}$
- $\omega_{\mathrm{i}} \cdot \mathbf{n}$ is the weakening factor of outward irradiance due to incident angle, as the light flux is smeared across a surface whose area is larger than the projected area perpendicular to the ray. This is often written as $\cos \theta_i$.

;

This is a **nasty** equation. You have to do an integral over a volume of a multiple vector-valued functions, for each wavelength of light, for each moment in time, for each output direction.

But, we can extract some useful and important pieces from this:

- L_o is the **output light** (the light we see from the surface);
- L_e is the **emitted light** coming out of the surface;
- L_i is the **incident light** coming on to the surface; and
- f_r is the **Bidirectional Reflectance Distribution Function** (BRDF)

BRDF is a function $f_r(\omega_i, \omega_o, \lambda, t)$ which describes how light is reflected at an opaque surface. Where ω_i is the incoming light direction, ω_o is the outgoing light direction, and λ is the wavelength of light. The BRDF is different for every surface and must be measured experimentally. However, we can find a good approximation with certain **lighting models**.

13.3 Lighting Models

Lighting models, also known as **shading models**, are a mathematical simplification and description of how light, color, reflection, refraction, etc., work.

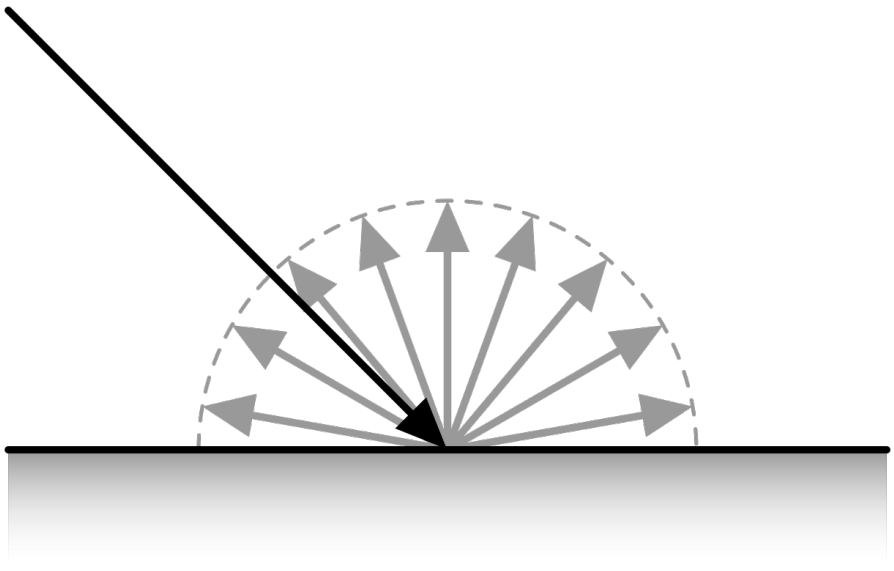
Ambient Reflectance.

Even when there is no specific light source, like the sun or a light bulb, there is still somehow some **ambient light**. This can be thought of as all the light which is *just there*. It is a constant intensity of light coming from no particular direction. It is the sum of all light which is mutually reflected off everything.

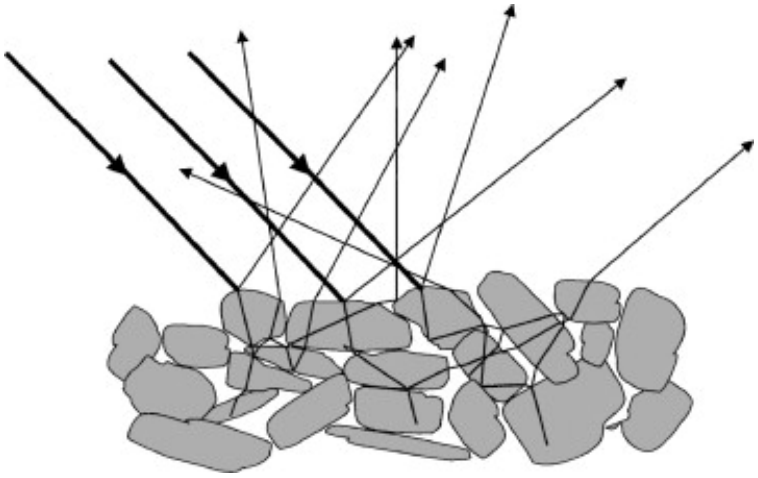
Ambient light does not cast shadows. Where there's a shadow, there's a light source's direction.

Lambertian Reflectance.

The Lambertian model is the simplest reflection. It assumes the BRDF is contant. That is, the amount of light from a light source that is reflected from a particular point on the surface is the same in every direction. It models a perfectly **matte** surface. This matte appearance is called **diffuse reflection**.

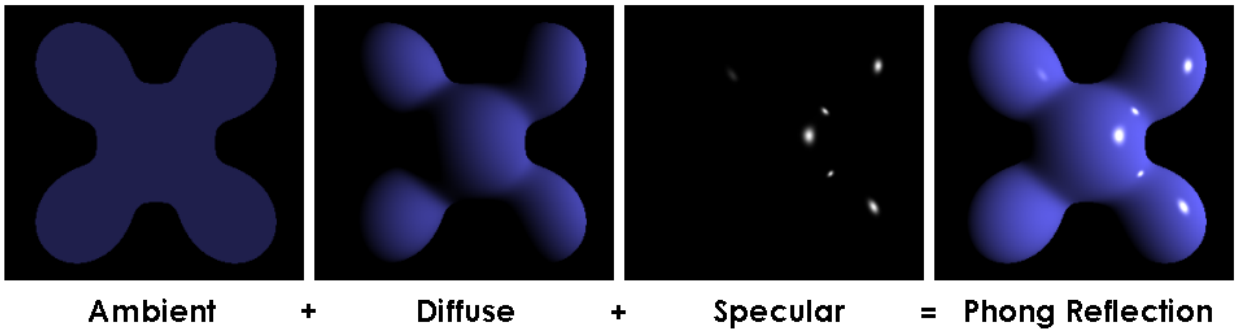


In reality, diffuse reflection is much more common than a perfect mirror reflection. Diffuse reflection if physically caused by light partially entering a material, bouncing around the individual particles that make up that material, and then being reflected and scattered in random directions.



Phong Reflection Model.

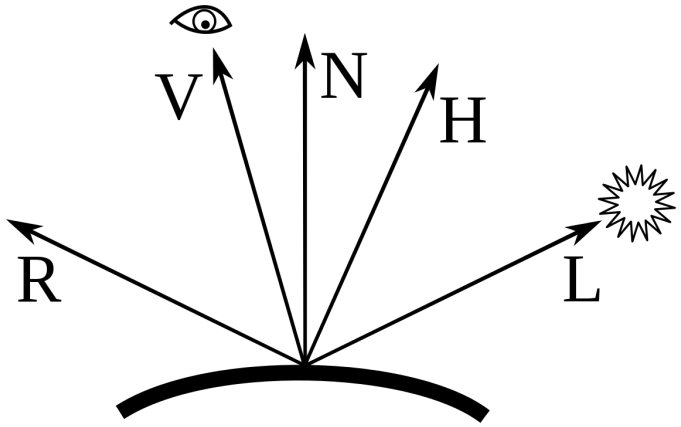
The Phong reflection model (a.k.a Phong lighting model) combines all three previous kinds of lighting: ambient, diffuse (Lambertian), and specular.



Time to do some actual math with these models. That means normal vectors, light source vectors, reflection vectors, and viewing vectors.

In the Phong model, the visible color of a surface is calculated independently for each point on that surface. In graphics, this would be the **fragments**. For each fragment in a rasterized surface, we compute that fragment's color as the combination of ambient reflection, diffuse reflection, and specular reflection.

$$color_{out} = ambient + diffuse + specular$$
$$L_o = L_a + L_d + L_s$$



- The **ambient component** of Phong reflection is uniform everywhere on the surface. Thus, this is constant number for every point on (fragment of) a surface.

- The **diffuse component** of Phong reflection is *independent of the viewer's position*. For each point on the surface, its diffuse reflection is constant in every direction. But, that reflection is not constant for every point on the surface.

$$L_d = I \cdot \max(0, \vec{n} \cdot \vec{l})$$

  where I is the incoming light intensity, $\vec{n}$ is a normalized vector for the surface normal at that point, and $\vec{l}$ is a normalized vector for the direction to the light source *from* the point on the surface.

  We take max because the dot product may be negative. In such a case, it is because the angle between the light and the normal is more than 90°(the light is *behind the surface*).

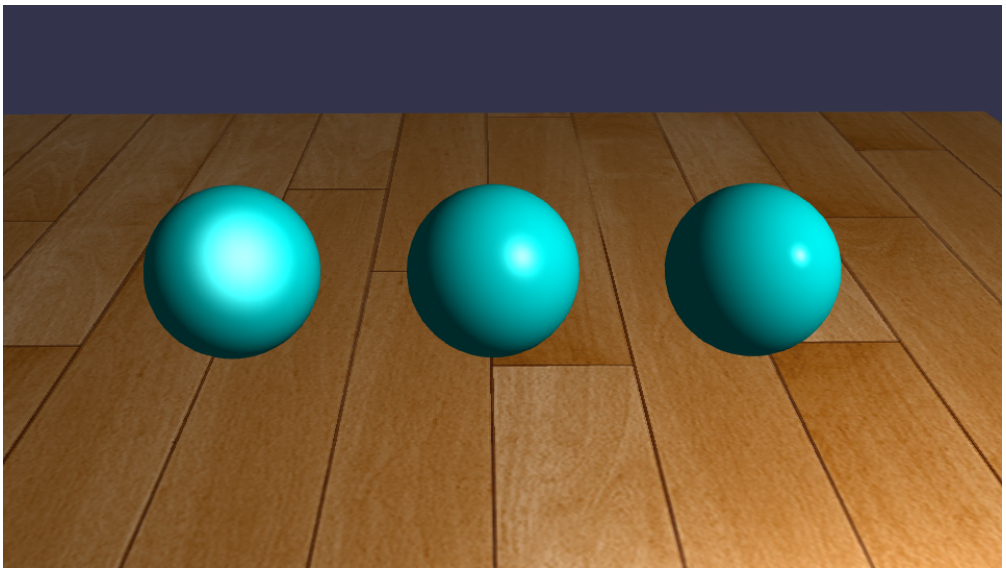
- The **specular component** of Phong reflection depends on the viewer's position. In specular/mirrored reflection, the output light is in direction $\vec{r}$ which is the reflection of incoming light (-$\vec{l}$) across the surface normal $\vec{n}$.

  If the viewer happens to be in line with the reflected ray, the they see that ray. However, we want to measure not just *perfect reflection* but **glossy reflection**. Such a surface has some so-called *specular highlights* but not necessarily a mirrored finish. Much like where we used the angle between the light source's direction and the normal to determine the intensity of the diffuse component, we will use the angle between the viewer's direction and the reflected ray to determine the intensity of the specular component.

$$L_s = I \cdot \max(0, \vec{v} \cdot \vec{r})^{\alpha}$$

  where I is the incoming light intensity, $\vec{r}$ is a normalized vector in the direction of the reflected light ray, and $\vec{v}$ is a normalized vector for the direction to the viewer *from* the

point on the surface. Finally, α is the **shininess** parameter that controls how "shiny" the surface appears. The larger the value, the more shiny. Typical values are between 0 and 100.



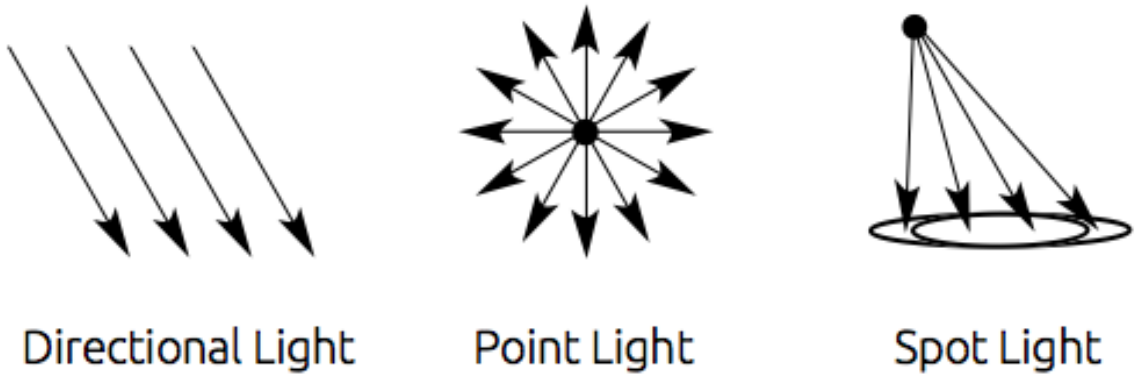
Alpha values of 2, 16, 64 (left to right).

13.4 Light Sources

In computer graphics, there are three main categories of light sources.

- **Directional light** is light that comes from a certain direction, but has no particular position. This can be thought of as "the sun". Light from the sun comes down at a particular angle, but its actual source if far away off at (nearly) infinity.
- **Point lights** is light that exists at a particular position in the world. It emits lights spherically, in all directions, originating at that point.
- **Spot light** is a light that is at a particular position, but only emits a *cone* of light from that position.

In most scenes, they have a single directional light. But, there are no limits to the number of lights you can have. However, there are practical limits. For each fragment, for each light source, you have to do a lighting calculation. This can get very expensive.



13.5 Lighting Colors and Materials

In order to create many different visual effects, we often change various parameters with respect to light sources and the surface being rendered.

Each light source can specify a **light color** that it emits and an **intensity** representing the "strength" or "power" of the light source. Going back to Newton, light may be white, and thus have all colors, or only some colors. Moreover, some graphics libraries allow you to specify different ambient, specular, and diffuse colors for a single light source.

Materials represent, well, the material that a surface is made of. Depending on your graphics library, custom shaders, etc., materials can have numerous and varied parameters. The most basic material properties are:

- k_a: the ambient reflection constant; the fraction of ambient light that is reflected
- k_d: the diffuse reflection constant; the fraction of ambient light that is reflected
- k_s: the specular reflection constant; the fraction of specular light that is reflected

Going back again to electromagnetic waves, recall that different materials appear to be different colors due to their absorption of some colors and reflectance of others. Therefore, it is often not enough to specify a material's, for example, diffuse reflectance with just a single number. Really, we need a reflection constant **for each wavelength of light**.

But that's crazy-talk when it comes to computational efficiency. Rather, it makes much more sense to use the color channels we already have: red, green, and blue. Then, a material can have a **ambient color**, **diffuse color**, and **specular color**.

A Phong shader in GLSL:

```
#version 400

// Interpolated values from the vertex shaders
in vec3 Normal;
in vec3 EyeDirection;
in vec3 LightDirection;

out vec4 color;

void main(){

    //material colors
    vec4 diffuse = vec4(0.0, 1.0, 1.0, 1.0); //blue-green color
    vec4 ambient = vec4(0.2,0.2,0.2,1.0);
    vec4 specular = vec4(0.7, 0.7, 0.7, 1.0);

    vec3 n = normalize( Normal );
    vec3 l = normalize( LightDirection );
    float cosTheta = clamp( dot( n,l ), 0,1 );
    //ensure dot product is between 0 and 1

    vec3 E = normalize(EyeDirection );
    vec3 R = reflect(-l,n);
    float cosAlpha = clamp( dot( E,R ), 0,1 );

    float alpha = 64;
    color = ambient + diffuse*cosTheta + specular*pow(cosAlpha,alpha);
}
```