CSI 4341, Computer Graphics

Lecture 11: Lighting and Shading

Date: 2012-10-02
Author(s): Slavka Jaromerska, Petr Praus
Lecturer: Fredrik Niemelä

1 Shading - Real world

Different types of light-material interactions cause that the object appears to have different colors or shading. Components that play role in these interactions:

- light sources
- material properties
- viewer location
- surface orientation

In real world the ligth is scattered back and forth between objects, partially absorbed here and there, all in an infinite and quite complex manner expressed by **Rendering equation**.

1.1 Rendering Equation

- describes infinite scattering and absorbtion
- cannot be solved in general
- global effect

1.2 Ray Tracing

- approach to compute perfectly reflecting surfaces
- not suitable for pipelining

1.3 Global effects

- like translucent surfaces, shadows, multiple light scattering

1.4 Local effects

- good for pipeline architecture, usually can provide sufficient approximation

2 Light Sources

2.1 Actual light source

- contains infinite number of points
- create soft shadows

2.2 Simple light sources

- **Point:** defined by position, color
- **Distant:** special case of Point source, placed in infinity, casts parallel arrays of light, defined by vector
- **Spotlight:** point source restricted in direction
- **Ambient:** same intensity everywhere, can model contribution of many sources and reflective surfaces

3 Surface types

3.1 Smooth

Ideal smooth surface is a perfect mirror, reflects all light concentrated in one direction.

3.2 Rough

Scatters light in all directions.

4 Phong Model

Simple, enables rapid computation. But cannot create mirrors. Three main light components are **diffuse**, **specular** and **ambient**. Works with four vectors: direction to light source $\mathbf{l}$, direction to viewer $\mathbf{v}$, surface normal $\mathbf{n}$ and perfect recflector $\mathbf{r}$.

4.1 Ideal Reflector

Ideal reflector has the same angle from $\mathbf{n}$ as $\mathbf{l}$ (**angle of incidence** or **angle of reflection**), we assume all three vectors are coplanar. Reflector can be computed $\mathbf{r} = 2(\mathbf{l} \cdot \mathbf{n})\mathbf{n} - \mathbf{l}$

4.2 Lambertian Surface

Lambertian surface is a perfect diffuse reflector. Amount of reflected light is proportional to the vertical component of light. In other words

$$reflectedLight \propto \cos\theta_i$$

where θ_i is the angle of incidence. A way to think about it is, the larger the angle, the wider the "beam" of light and the less dense the intensity.

$$\cos\theta_i = \mathbf{l} \cdot \mathbf{n}$$

as long as **n** and **l** are normalized.

4.3 Diffuse Term

If we allow only some portion of incoming light to be diffused (no more Lambertian surface), the **diffuse term** can be computed as

$$I_d = L_d k_d (\mathbf{l} \cdot \mathbf{n})$$

where L_d is incoming light intensity. For material, amounts of light diffused can be expressed separately for each color: k_{dr}, k_{dg}, k_{db}.

4.4 Specular Term

For shiny surfaces, the Phong Model constructs simplistic model: the amount of light the user sees depends on the angle ϕ between **r** and **v**. The shininess is specified by shininess coefficient α. α in range 100 and 200 approximates well metals, 5 to 10 plastic materials.

$$I_s = k_s L_s cos^\alpha \phi = k_s L_s (\mathbf{r} \cdot \mathbf{v})^\alpha$$

assuming that **r** and **v** are normalized. Again for material, amounts of light reflected can be expressed separately for each color: k_{sr}, k_{sg}, k_{sb}.

4.5 Ambient Term

$$I_a = k_a L_a$$

Where L_a is the incoming light intensity. Again for material, amounts of light used for ambient lighting can be expressed separately for each color: k_{ar}, k_{ag}, k_{ab}.

4.6 Distance term

For positional light sources, we may also want to account for the attenuation of light received due to its distance from the source. $(a + bd + cd^2)^{-1}$ where d is distance between **p** and $\mathbf{p_0}$. Constants a, b, c can be chosen to soften lighting. This term is added to diffuse and specular lighting equations. It's not correct in terms of physics but it makes it look nicer.

4.7 Light source

For diffuse, specular and ambient component and each for each color, we can express the source intensities with 9 variables: $L_{dr}, L_{dg}, L_{db}, L_{sr}, L_{sg}, L_{sb}, L_{ar}, L_{ag}, L_{ab}$.

4.8 Adding sources

We can then compute the contribution for each color by all sources source by adding the ambient, diffuse, and specular components computed in previous sections together: $I_{dr}, I_{dg}, I_{db}, I_{sr}, I_{sg}, I_{sb}, I_{ar}, I_{ag}, I_{ab}$.

For example, for red term, it would be:

$$I_r = \sum_i (I_{iar} + I_{idr} + I_{isr}) + I_{ar}$$

I_{ar} is the red component of the global ambient light.

Note, that we might be using a slightly different notation than in textbook. All L's represent light components of different light sources. All I's represent light intensities transformed by materials. Thus in the above equation, the L's of each light source we previously "transformed" into I's by material (by equation in next section) and now are being added together since we have multiple sources.

4.9 Material properties

We are specifying ambient, diffuse, and specular reflectivity coefficients (k_a, k_d, k_s) for each primary color.

Phong model allows us to compute the intensity of the reflected light that the viewer sees. It can be expressed as

$$I = k_d L_d \mathbf{l} \cdot \mathbf{n} + k_s L_s (\mathbf{v} \cdot \mathbf{r})^\alpha + k_a L_a$$

Or in a more developed fashion including the attenuation by the distance term:

$$I = \frac{1}{a + bd + cd^2} (k_d L_d \max(\mathbf{l} \cdot \mathbf{n}, 0) + k_s L_s \max((\mathbf{r} \cdot \mathbf{v})^\alpha, 0)) + k_a L_a$$

4.10 Modified Phong model (Blinn-Phong Model)

With the Phong model, we have to recalculate $\mathbf{r} \cdot \mathbf{v}$ at every point on the surface. If we take vector $\mathbf{h}$ halfway between $\mathbf{v}$ and $\mathbf{r}$

$$\mathbf{h} = \frac{\mathbf{l} + \mathbf{v}}{|\mathbf{l} + \mathbf{v}|}$$

and $\mathbf{v}$ lies in the same plane as $\mathbf{l}$, $\mathbf{n}$ and $\mathbf{r}$ (which usually does not, but it still works pretty well), the we can replace $\mathbf{r} \cdot \mathbf{v}$ with $\mathbf{n} \cdot \mathbf{h}$ and avoid calculation of $\mathbf{r}$. However, the new angle is half the size of the previous and we have to appropriately fix the shininess coefficient.

5 Implementation in OpenGL

- Normally, we have to specify normal for each vertex.
- If we have normalized vectors, we can use dot product to compute cosines.
- GLSL vector normalization using $n = normalize(v)$

5.1 Finding normal to the plane

Given noncollinear points p_0, p_1, p_2, we need to find the normal of the plane defined by these points. We can find the normal like this:

$$\mathbf{n} = (\mathbf{p_2} - \mathbf{p_0}) \times (\mathbf{p_1} - \mathbf{p_0})$$

5.2 Light sources

- Diffuse, ambient, specular.
- Specified in homogeneous coordinates, 0 in w coordinate for parallel light source (distant), 1 for point light source.
- Attenuation at the edges for spotlight is proportional to $\cos^\alpha \phi$.
- We can use global ambient term instead of ambient term for each light source. Easier calculation, same result.

5.3 Polygon shading

- If we use normals of the polygons that approximate the shape (e.g. sphere), it'll be very ugly.
- In simple cases we can solve analytically (like sphere), we can get precise normals. For more complex shapes, we can't get normals everywhere.
- Gouraud shading solves this by defining the normal at a vertex to be the normalized average of the normals of the polygons that share the vertex.