Warsaw University of Technology
Faculty of Mathematics
and Information Science

BACHELOR'S THESIS
COMPUTER SCIENCE

BACHELOR EXAM

Author:
Jakub Ciecierski

Supervisor:
Jezus

Warsaw, february 2016

....

Supervisor’s signature   Author’s signature

Contents

Chapter 1

Algorithm Design and Complexity

Chapter 2

Graph Theory

Chapter 3

Automata Theory and Languages

Chapter 4

Numerical Methods

Chapter 5

Programming

Chapter 6

Computer Graphics

Sources: [1], [2]

6.1. Color Models

6.2. Illumination Models

Illumination Models can be split into two groups:

- **Local.**
    - Objects are considered independently. Direct Reflections.
    - Tries to mimic reality with the small computational complexity
    - Example: **Phong Illumination**
- **Global.**
    - Light reflected from other objects
    - Closer to reality, requires big computational complexity
    - Example: **Ray Tracing, Radiosity.**

6.2.1. Phong Illumination

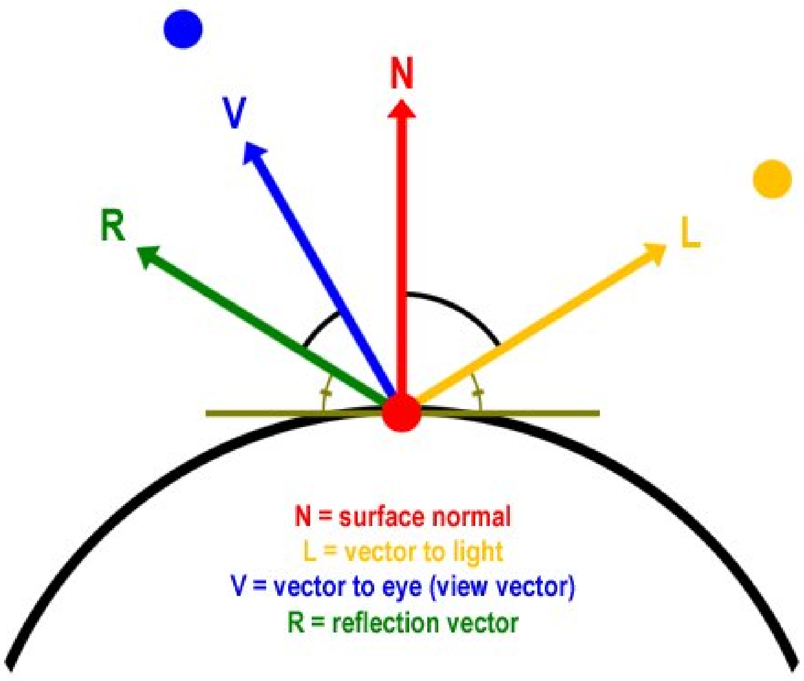

Figure 6.1

Phong illumination specifies the four vectors described in figure 6.1 and several colors:

- A_L, A_M - Ambient color of the light source and surface material
- D_L, D_M - Diffuse color of the light source and surface material
- S_L, S_M - Specular color of the light source and surface material

Because light is additive, the final intensity of the light reflected by a given surface is simply the sum of the ambient, diffuse and specular components:

$$I = A + D + S$$

6.2.2. Ambient Light

Ambient light is a constant amount of light that gets added to the scene no matter what. It's goal is to fake the contribution of indirect reflections, which can normally only be accounted for using global illumination solutions. The ambient term is used simply to keep shadows from turning pitch black, which would be very unrealistic.

Ambient light is a "hack" to simulate global illumination.

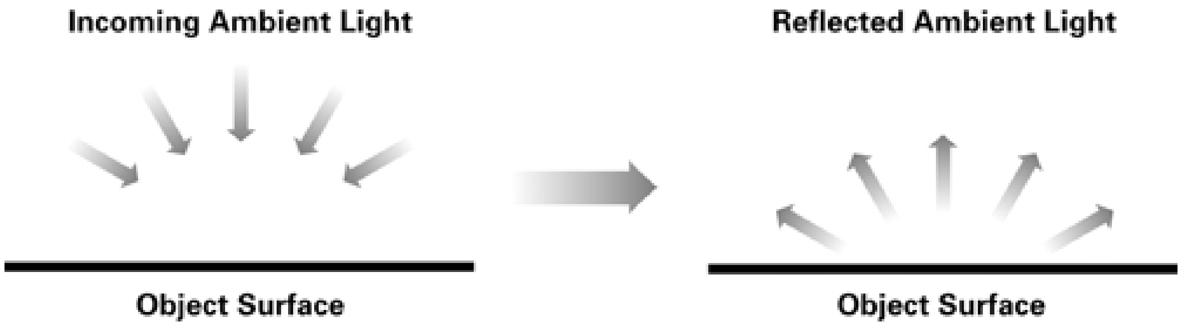

Figure 6.2

The ambient term is simply a combination of the ambient components of the light source and the surface material. By multiplying the light color and the material color, we account for the fact that any material only reflects certain frequencies of light. A green surface, for example, does not reflect red light. As we will see below, the diffuse and specular terms also combine the light and material colors in the same way.

$$A = A_L \cdot A_M$$

6.2.3. Diffuse Light

The diffuse term accounts for directed light reflected off a surface equally in all directions. In general, diffuse surfaces are rough on a microscopic scale, with small nooks and crannies that reflect light in many directions. When incoming rays of light hit these nooks and crannies, the light bounces off in all directions

Diffuse Object (Magnified)

Figure 6.3: Intuition behind the diffuse light. The magnified object has alot of nooks and crannies, thus the lights bounces off in all directions

The amount of light reflected is proportional to the angle of incidence of the light striking the surface. Surfaces with a dull finish, such as a dusty chalkboard, are said to be diffuse. The diffuse contribution at any particular point on a surface is the same, regardless of where the viewpoint is.

Diffuse light hits a surface and gets scattered equally into all directions

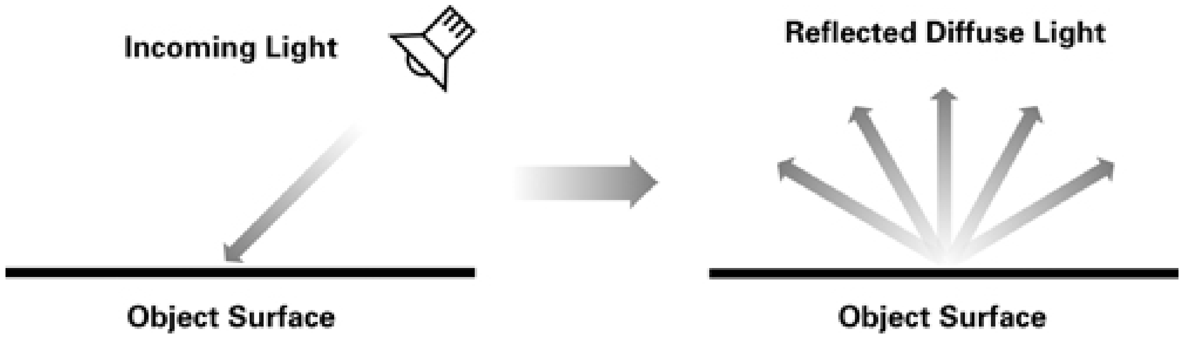

Figure 6.4

Diffuse lighting is based on the angle between the light vector and the surface normal. The surface normal is a given, and requires no further explanation. The light vector L is the normalized vector from the point being shaded to the light source. Given these two vectors, the diffuse term is:

$$D = D_L \cdot D_M \cdot max(L \bullet N, 0)$$

The dot product of L and N returns the cosine of the angle between those two vectors. When the two vectors are equal, the dot product is one. When they are perpendicular, the value is zero. If the two vectors point away from each other, the value becomes negative. When this happens, the value must be clamped to zero to prevent ending up with "negative light".

The clamped dot product is also multiplied (or modulated) with the diffuse colors of the light and the surface material.

6.2.4. Specular Light

The specular term represents light scattered from a surface predominantly around the mirror direction. The specular term is most prominent on very smooth and shiny surfaces, such as polished metals.

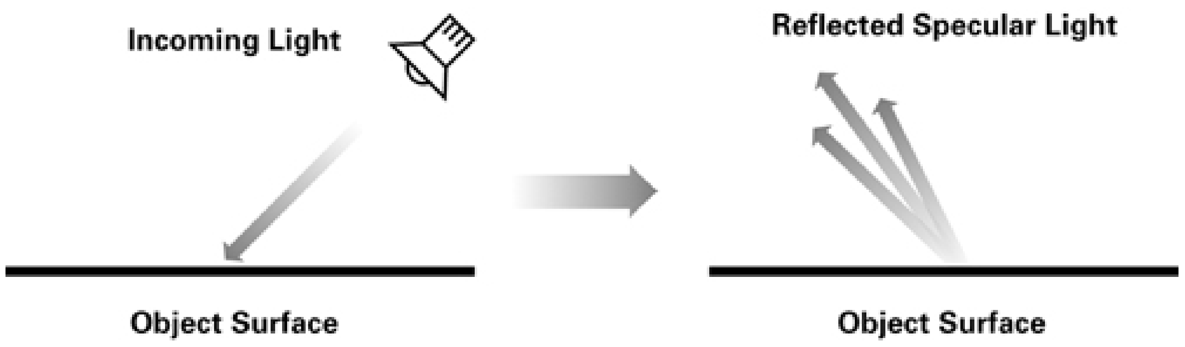

Figure 6.5

Specular term is dependent on the angle between the reflected light vector R and the view vector V. The view vector is the normalized vector from the point being shaded to the camera position. The reflection vector is calculated as follows:

$$R = 2(N \bullet L)N - L$$

The reflection vector represents the direction the incoming light would be reflected in if the surface were a perfect mirror. The angle between R and N is equal to that between L and N. Having calculated R and V, we can now compute the specular term:

$$S = S_L \cdot S_M \cdot max(R \bullet V, 0)^n$$

The dot product is there because we want to favor reflections in the direction of the camera to those that point away from the camera. The larger the angle between R and V, the lower the specular term will be. Notice how we also raise the dot product to a power of n, where n is called the specular exponent of the surface and represents its shininess. Higher values of n lead to smaller, sharper highlights, whereas lower values result in large and soft highlights

To summarize Specular Light:

- reflection off of shiny surfaces - you see a highlight
- shiny metal or plastic has high specular component
- chalk or carpet has very low specular component
- position of the viewer IS important in specular reflection

Remark: Blinn Specular uses half-angle vector which is a vector halfway between the light vector and the view vector

6.2.5. Summary

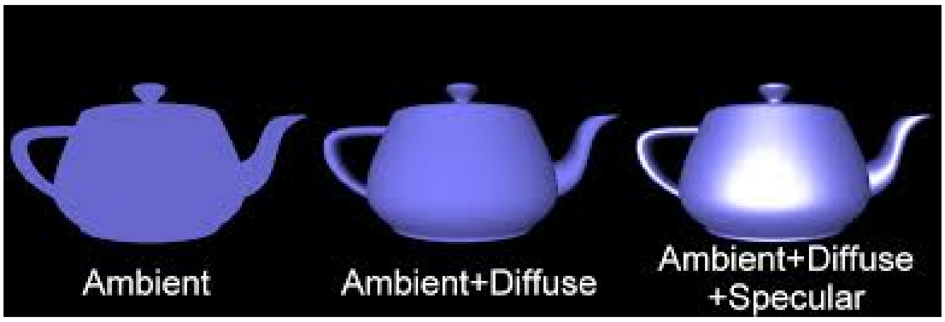

Figure 6.6

6.3. Shading

6.3.1. Flat Shading

- Each entire polygon is drawn with the same color
- Need to know one normal for the entire polygon
- Fast
- Lighting equation used once per polygon

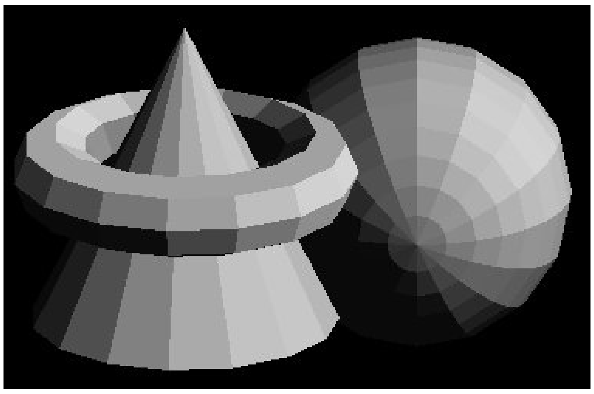

Figure 6.7

6.3.2. Goraud Shading

First, intensities on vertices are calculated. Then color at each pixel inside a polygon is a linear interpolation of intensities at vertices

- Colors are interpolated across the polygon
- Need to know a normal for each vertex of the polygon
- Slower than flat shading
- Lighting equation used at each vertex

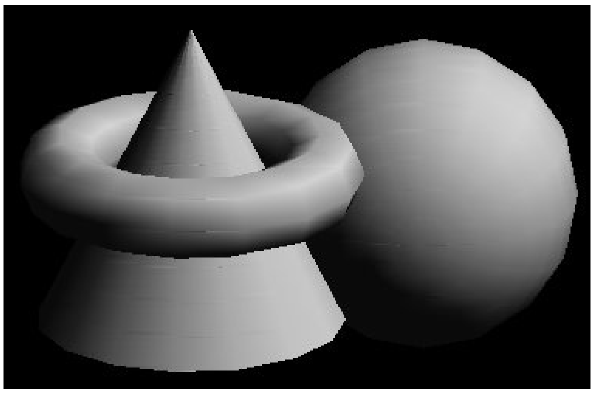

Figure 6.8

6.3.3. Phong Shading

- Normals are interpolated across the polygon
- Need to know a normal for each vertex of the polygon
- Better at dealing with highlights than Goraud shading
- Slower than Goraud shading
- Lighting equation used at each pixel

Where Goraud shading uses normals at the vertices and then interpolates the resulting colours across the polygon, Phong shading goes further and interpolates tha normals. Linear interpolation of the normal values at each vertex are used to generate normal values for the pixels on the edges. Linear interpolation across each scan line is used to then generate normals at each pixel across the scan line.

6.4. Raster Algorithms

A line drawing algorithm is a graphical algorithm for approximating a line segment on discrete graphical media. On discrete media, such as pixel-based displays and printers, line drawing requires such an approximation (in nontrivial cases). Basic algorithms rasterize lines in one color. A better representation with multiple color gradations requires an advanced process, anti-aliasing.

6.4.1. Line Drawing

Naive line drawing uses floating point operations which is slowing down the computations

Bresenham

Midpoint line algorithm

We choose E or NE by calculating on which side of the line the mid point lies

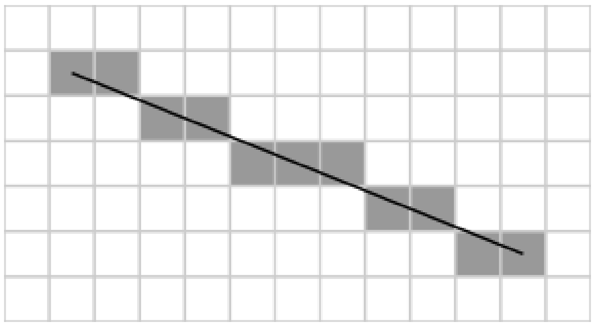

Figure 6.9

6.4.2. Aliasing and Anti-Aliasing

Bresenham yields a aliased line: appearing stepped or jagged.

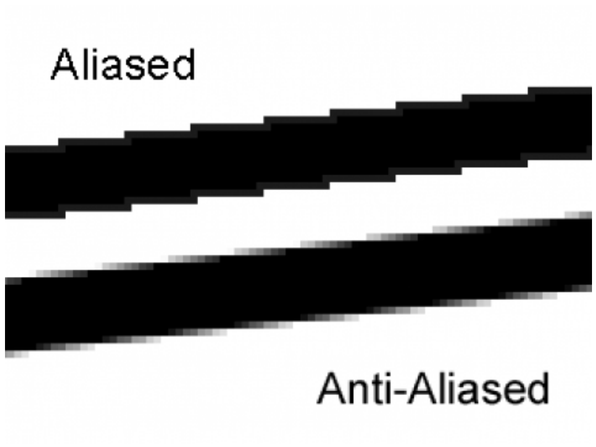

Figure 6.10

6.5. Spline Functions

6.6. Filling the Area

6.7. Visible Surface Determination

6.8. Image Filtering

Chapter 7

Software Engineering

Bibliography

[1] $http://www.gameprogrammer.net/delphi3dArchive/phongfordummies.html$

[2] $http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter05.html$