Assignment 4 - Shading

Mikkel Willén

January 4, 2023

Contents

1 Introduction

This assignment is about understanding and implementing phong shading of triangles. In this report, we will go through what phong shading is, and the three terms in phong shading: ambient, diffuse and specular reflections.

2 Theory

Phong shading is a simple way of modeling how light reflects on a surface. Phong shading is made up of three terms, ambient, diffuse and specular reflections. Ambient reflection are reflections not directly on the surface, but reflected on multiple surfaces. Ambient reflections are also known as inter reflections. The model we use for phong reflection is a simple model and does not take into account, that light can come from many directions. We have a direct lightning that we model. We can write this as

$$R_{\text{ambient}} = l_a k_a \quad : \quad k_a \in [0, 1]$$

Here l_a is the lighting intensity and k_a is the reflection coefficient between 0 and 1.

Diffuse reflections are when light is reflected equally in all directions, when it hits a mat surface. This is also known as labertian reflection. The figure below shows a light beam with energy E hitting a surface dA.

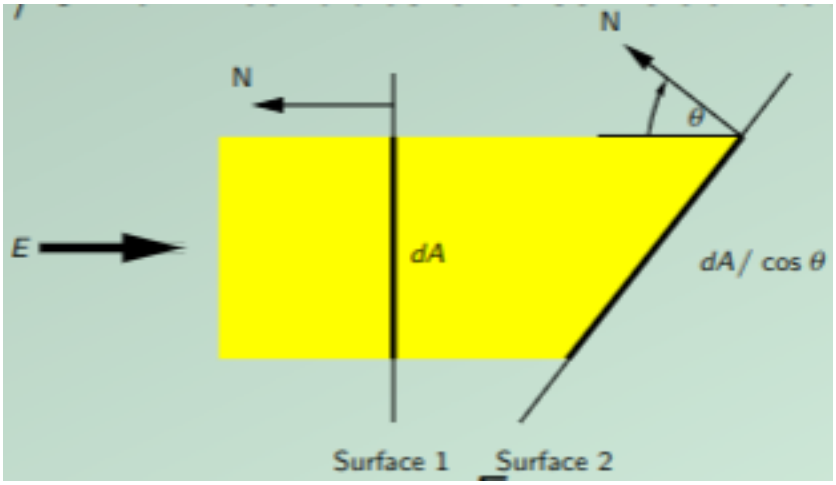

Figure 1: Visualization of diffuse reflection

dA is a very small area, and is perpendicular on the light beam. We call this surface "surface 1". We also have another surface "surface 2", which is surface 1 turned, so they have a difference of angle θ. The area on surface 2 is bigger, which means that the energi is more intens on surface 1. The intensity on the two surfaces can be written as

$$\begin{aligned} \text{surface 1}: \quad & l = \frac{E}{dA} \\ \text{surface 2}: \quad & l_\theta = \frac{E\cos\theta}{dA} = l\cos\theta \end{aligned}$$

When we look at the surface from they eye, we can figure out how much light is reflected from the surface and into the eye. We have an angle Φ, which is the angle between normale vector and the view vector. Lamberts law, if we have some energi being reflected from a surface, we have an amount of energy being emitted perpendicular on the surface equal to the normals directions. We call this E_N. If we look in the direction of V and want to know how much light is being emitted, we can use Lamberts law. Lamberts law says, that the light being emitted in the V direction is equal to the light being emitted in the N direction multiplied with cosΦ. We write this as

$$\begin{aligned} \text{brightness}: \quad & E_V = E_N\cos\Phi \\ \text{projection area}: \quad & dA_V = dA\cos\Phi \end{aligned}$$

$$\text{intensity}: \quad I_V = \frac{E_V}{dA_V} = \frac{E_N \cos\Phi}{dA \cos\Phi} = \frac{E_N}{dA} = I_n$$

R_{diffuse} can now be written as

$$R_{\text{diffuse}} = I_p k_d \cos\theta = I_p k_d (N \cdot L)$$

The figure below shows how the different variables are illustrated.

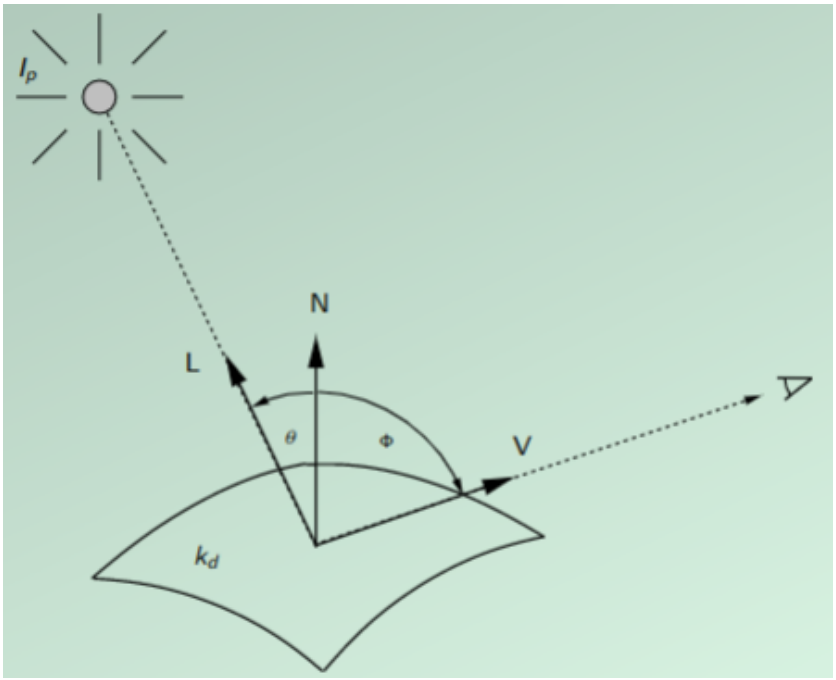

Figure 2: Lamberts law seen from the eye

If we want the vectors to be unit vectors then $\cos\theta$ is equal to the dot product of N and L.

Specular reflections are reflection when light hits shinning surfaces, such as a mirror. Another example is an apple, that has an almost white point and the rest of the apple is diffuse reflections. The almost white light will then move, if we move our head. This is because the light is reflected differently on the surface. If we had a perfect mirror, light would only be reflected in one dirrection R which would be the mirror L with the normal vector N. Since we do not have a perfect mirror, the light is reflected over a bigger

area, which is getting smaller the further we move away, with an angle α from R. I_{specular} can be written as

$$I_{\text{specular}} = I_p k_s \cos^n \alpha = I_P k_s (R \cdot V)^n$$

Again the vectors are unit vectors and we can use the dot product instead. Since R is mirrored of the L vector, it means that thier angles are identical. We know the vectors L and N, so the first thing we do is to project the L vector on the N vector. Then we find $N\cos\theta$, which is a vector on the normal vector.

$$\begin{aligned} S &= N\cos\theta - L \\ R &= N\cos\theta + S = N\cos\theta + N\cos\theta - L \\ &= 2N\cos\theta - L \\ &= 2N(N \cdot L) - L \end{aligned}$$

Below is a figure of how to find the R vector.

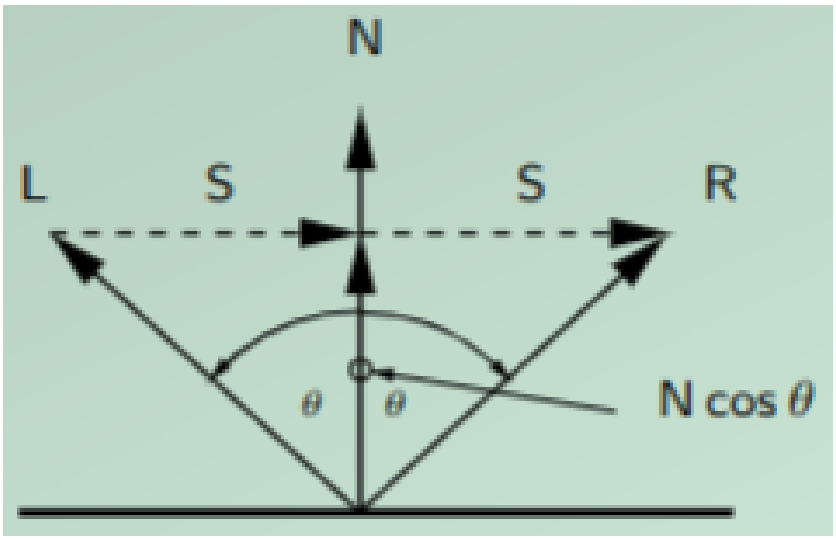

Figure 3: Finding the R vector

Now that we the three terms that makes up phong relfections, we can add them together, and we get

$$I_{\text{phong}} = I_{\text{ambient}} + I_{\text{diffuse}} + I_{\text{specular}}$$

$$= I_a k_a + I_p k_d (N \cdot L) + I_p k_s (R \cdot V)^n$$

In some of the terms, we have to compute the dotproduct. If this dotproduct is negativ, it would mean that the light would be negativ. We will now, given a point P with the samme vector N, transform P with a general homogenous transformation matrix M.

$$P = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \rightarrow \hat{P} = MP = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{w} \end{bmatrix}$$

The point P lies in the plane Π, where

$$\Pi^T P = 0$$

If the point P is transformed with the matrix M, we also need to transform the plane Π. We do this with an unknown matrix Q

$$\hat{\Pi} = QH$$

We can now find Q since we know that $\hat{P}$ lies in the plane $\hat{\Pi}$, and get got

$$\begin{aligned} \hat{\Pi}^T \hat{P} &= 0 \\ (Q\Pi)^T (MP) &= 0 \\ \Pi^T Q M P &= 0 \end{aligned}$$

We know that

$$\Pi^T P = 0$$

so we get

$$Q^T M = I$$

which we can rewrite

$$Q = (M^{-1})^T$$

Now $\hat{P}$ lies in $\hat{\Pi}$, and if we have to transform the normal N we need to transform it with $(M^{-1})^T$ aswell.

3 Implementation

Implementing phong shading, we looked at two files. vertextransform.vert and phong.frag. In vertextransform-vert we got the CTM matrix, which is the current transformation matrix from the last assignment. We use this variable to get new positions, which we do with

```
gl_Position = CTM * vec4(Vertex, 1.0f);
```

In the phong.frag file, we make the phong reflections. phong.frag has the variables AmbientLightColor, LightPosition, LightColor, EyePosition, AmbientColor, DiffuseColor, SPecularColor and Shininess. These variables are used to find the ambient, diffuse and specular reflections. The start by computing the ambient reflection with

```
vec3 ambientColor = AmbientLightColor * AmbientColor;
```

and then adding it to a variable color, which in the end, should be the final color. We then compute the vectors N, L, V and R and then normalizing them. Now we can compute diffuse and specular reflection with the formulars from the theory section, though we start with a check, checking that the dotproducts are positiv. If they are negativ, we do not add them to the final color. In the end we set

```
FragColor = vec4(color, 1.0f);
```

so it gets drawn.

4 Testing

In the figure below, we show each of the terms ambient, diffuse and specular individually.

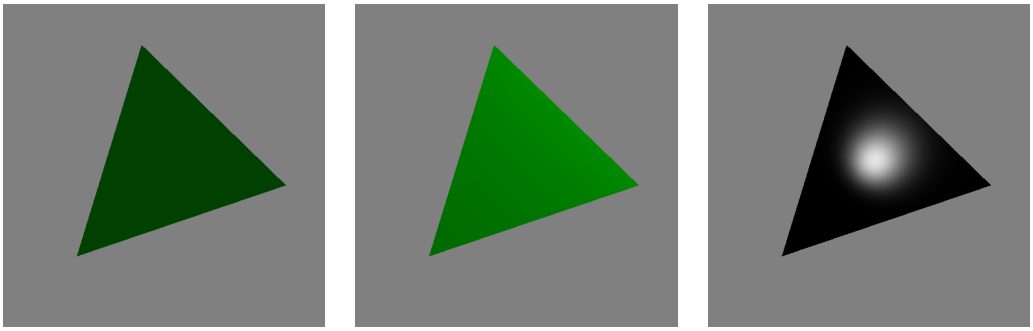

Figure 4: Ambient, diffuse and specular reflections

Below is a figure with the final shading of the triangle, when we add up the three terms.

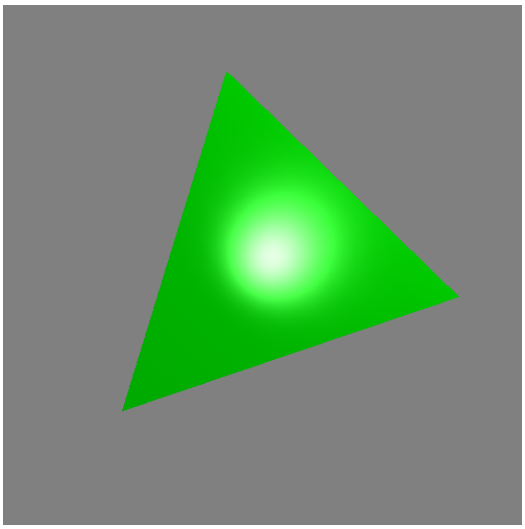

Figure 5: Phong shading of a triangle

5 Conclusion

We have gone through the three terms of phongs reflection model and how they together combine to make phong shading. Then we have implemented it in code. We made some tests to make sure our program works, and that

all three of the terms work correctly. After the tests, we can see, that the program works as it should.

References