RI3004A 3D Graphics Rendering

Discussion 4 (Answers)

For **Lecture 7**

Please attempt the following questions before you go to your discussion class. Some of the questions may be quite open-ended and some may be even ambiguous. In those cases, you are encouraged to make your own (reasonable) assumptions.

(1) Given the Phong Illumination Equation (PIE)

$$I_{Phong} = I_a k_a + f_{att} I_p k_d (N \cdot L) + f_{att} I_p k_s (R \cdot V)^n .$$

(i) Propose values for the coefficients k_a, k_d, k_s, and n to model the surface material of a yellow billiard ball. **(ii)** Propose values for the coefficients k_a, k_d, k_s, and n to model the surface material of a gold-plated ball.

(i) k_a = [1, 1, 0], k_d = [1, 1, 0], k_s = [1, 1, 1], n = 100.

(ii) k_a = [1, 1, 0], k_d = [1, 1, 0], k_s = [1, 1, 0], n = 100.

For a non-metallic surface, the color of the specular highlight remains the same as the color of the light source, but for a metallic surface, the color of the specular highlight changes with the "color" of the surface. Note that the PIE is still a very inaccurate model for the reflectance of metallic surfaces.

(2) What happens to the PIE if the surface point is back-facing the light (i.e. the angle between N and L is greater than 90 degrees)? How can you modify the PIE to deal with this situation?

The value of the diffuse component of PIE will become negative. This is physically impossible. We can modify the PIE to

$$I_{Phong} = I_a k_a + f_{att} I_p k_d \max(N \cdot L, 0) + f_{att} I_p k_s \max(R \cdot V, 0)^n$$

(3) Given a sphere that is approximated by a set of triangles on its surface. Assume the sphere center is at the origin. For a vertex v on the surface, what should be its normal vector?

The normal vector at v should be a vector that points from the center of the sphere towards v. Therefore, since the sphere center is at the origin, the normal vector at v can be computed as $v / |v|$.

(4) **(i)** Show the OpenGL code to put a point light source at the location (0, 1, 0) in the *World* coordinate frame. **(ii)** Show the OpenGL code to put a point light source at the location (0, 1, 0) in the *Camera* coordinate frame. For the second case, describe how the position / motion of the light source would appear to be.

(i)

```
GLfloat light0_pos[] = {0.0, 1.0, 0,0, 1.0};
...
glMatrixMode( GL_MODELVIEW );
glLoadIdentity();
gluLookAt(...);
glLightfv(GL_LIGHT0, GL_POSITION, light0_pos);
...
```

(ii)

```
GLfloat light0_pos[] = {0.0, 1.0, 0,0, 1.0};
...
glMatrixMode( GL_MODELVIEW );
glLoadIdentity();
glLightfv(GL_LIGHT0, GL_POSITION, light0_pos);
gluLookAt(...);
...
```

In the case that the light source is positioned directly in the Camera coordinate frame, the light source will appear to move together with the viewpoint (e.g. a miner's headlamp).

(5) We have learned about the Phong model $I_{Phong} = I_a\, k_a + I_p\, (\, k_d\, (N \cdot L) + k_s\, (R \cdot V)^n\,)$. Consider a variant of the Phong model, called the *Blinn-Phong* model, whose computation is $I_{Blinn} = I_a\, k_a + I_p\, (\, k_d\, (N \cdot L) + k_s\, (H \cdot N)^n\,)$, where H is called the *Half vector* and it is the vector right in the middle of L and V.

(i) Write an expression to compute H. Note that H must be a unit vector.

(ii) What is the advantage of using $H \cdot N$ over using $R \cdot V$?

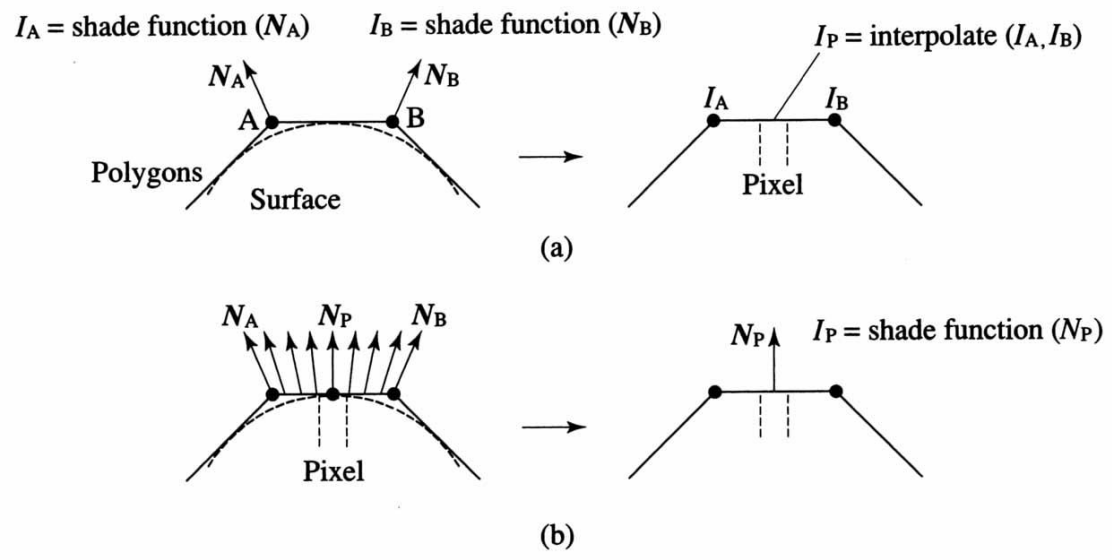
(iii) What is the difference in appearance between a surface rendered using Phong model and the same surface rendered using Blinn-Phong model?

(i) H = normalize($L + V$)

(ii) H is easier to compute (or derive) than R. Also, when the light source and the viewpoint is infinitely far away, the vector H is the same for all surface points.

(iii) The specular highlight from the Blinn-Phong model will look relatively larger.

(6) Explain Gouraud Shading and Phong Shading. At which stage of the rendering pipeline must the lighting computation be done for each? Which is more expensive to compute?



For Gouraud Shading, the lighting computation is done for each vertex only, therefore it can be done in the vertex processing stage. As for Phong Shading, the lighting should be done at each fragment, therefore it should be done after the fragments are produced, and in practice, it is often done in the fragment processing stage.

(7) **(i)** What is the main drawback of Gouraud Shading when compared to Phong Shading? **(ii)** Suggest one way to mitigate the above problem when *still* using Gouraud Shading.

(i) If a specular highlight falls entirely within a polygon, it will be missed when rendered using Gouraud Shading. It will likely not be missed when using Phong Shading.

(ii) We can subdivide (or tessellate) the polygon into many smaller polygons and smoothly interpolate the vertex normals at the vertices of the smaller polygons.

——— End of Document ———