Homework 6

Wednesday, November 24, 2021 10:57 AM

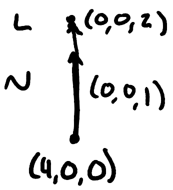
(a) Explain how you can make a light source stay at a fixed location or move with the camera.

You can make a light source stay at a fixed location by specifying its location AFTER the viewing transformation, and likewise you can make a light source move with the camera by specifying its location BEFORE viewing the transformation.

(b) Let $(0,0,2)$ be a position of a positional light source. Let $n=(0,0,1)$ be the normal at vertex $v=(4,0,0)$. Assume a white light source with ambient intensity $I_a=0.2$, diffuse intensity $I_d=0.3$, and specular intensity $I_s=0.4$. Assume constant and linear attenuation factors of 1, and a quadratic attenuation factor of 0. Given a material with ambient, diffuse, and specular coefficients, 0f 0.5, 0.6, and 0.7, respectively, compute the intensity at the vertex v due to ambient reflection.

$I_{ambient} = K_A I_a$ $\quad I_{ambient} = (0.5)(0.2) = 0.1$

(c) Using the information in the previous question compute the intensity at vertex v due to diffuse reflection.



$I_{diffuse} = K_d \cdot I_d \, (N \circ L)$

$I_{diffuse} = (0.6 \cdot 0.3)((0,0,1)\cdot(0,0,2)) = 0.36$

(d) Using the information in the previous question compute the intensity at vertex v due to specular reflection. Assume a shininess coefficient of 1 and that the position of the light source is given by $(8,0,2)$.

$I_{specular} = K_s \cdot I_s \, (R \cdot V)^{ns} = K_s \cdot I_s \, (N \cdot H)^{ns}$ $\quad H = \frac{L+V}{|L+V|}$

$I_{specular} = 0.7 \cdot 0.4 \left((0,0,1) \circ \frac{(8,0,2)+(4,0,0)}{|(8,0,2)+(4,0,0)|}\right)$

$= 0.7 \cdot 0.4 \left[(0,0,1) \circ \left(\frac{12}{\sqrt{148}}, 0, \frac{2}{\sqrt{148}}\right)\right] = 0.046$

(e) Let $t=(v_1,v_2,v_3)$ be a triangle where $v_1=(1,2,3)$, $v_2=(4,5,7)$, $v_3=(7,8,9)$. Compute a unit normal to this triangle assuming a counterclockwise winding order.

$N = (V_3 - V_2) \times (V_2 - V_1)$

$N = ((7,8,9)-(4,5,7)) \times ((4,5,7)-(1,2,3))$

$= (3,3,2) \times (3,3,4) = \begin{vmatrix} i & j & k \\ 3 & 3 & 2 \\ 3 & 3 & 4 \end{vmatrix} = (12-6)i - (12-6)j + 0$

$= (6,-6,0)$

(f) Let $v=(1,2,1)$ be a vertex with a normal $n=(0,1,0)$. Let a light source be located at $(2,1,1)$ and a camera located at $(0,1,1)$. Let the RGB Intensity of the light source be $I_d=(1,2,3)$, the diffuse reflection coefficient at the vertex be $k_d=0.9$, and the specular reflection coefficient be $k_s=0.2$. Compute the light intensity and the vertex due to diffuse and specular reflection (assume that there is no ambient light).

$I_{total} = I_{diffuse} + I_{specular}$

$I_{diffuse} = K_d I_d (N \circ L) = (0.9)(1,2,3)((0,1,0)\cdot(2,1,1)) = (0.9, 1.8, 2.7)$

$I_{specular} = K_s I_s (N \cdot H)^{ns} = (0.2)(1,2,3)\left[(0,1,0) \cdot \frac{(2,1,1)+(1,2,1)}{|(2,1,1)+(1,2,1)|}\right)$

$= (0.2, 0.4, 0.6) \cdot \frac{3}{}$

$I_{specular} = K_s I_s (N \cdot H) = \ldots \quad |(2,1,1)+(1,2,1)|$

$$= (0.2, 0.4, 0.6) \cdot \frac{3}{\sqrt{22}}$$

$$I_{total} = (0.9, 1.8, 2.7) + \frac{3}{\sqrt{22}}(0.2, 0.4, 0.6) = 0.8954$$

(g) Describe a method for approximating the dot product between a reflected ray and the direction of viewing with fewer computations (using the half way vector).

The half way vector is a way of approximating the dot product between a reflected ray and the direction of viewing, it can be used to replace many computations that may be inefficient overall.
As long as L, V, R are coplanar, instead of solving for cos(phi) we can instead solve for the normal vector dotted with the half way vector which can be calculated with the light vector and the viewing vector as L + V / |L+V|

(h) Explain how the efficiency of intensity computations can be increased when assuming a distant viewer and/or a distant light source.

Intensity computations can be increased when assuming a distant viewer and/or distant light source because only 1 normal needs to be calculated, which greatly reduces computations. However, it is less accurate.

(i) Explain why we do not consider the ambient contribution of refracted light when computing the combined reflected+refracted light intensity at a location.

We do not consider the ambient contribution of refracted light when computing the combined reflected + refracted light intensity at a location because it is already computed within the reflections.

(j) Given a normal vector n, a vector l pointing towards the light source, a vector v pointing towards the viewer, and a vector t indicating the direction of a refracted ray, write a formula for computing intensity due to specular and diffuse refraction (i.e. light transmitted through the surface). Assume that the intensity of the light source is I and that the specular and diffuse refraction coefficients are both 1.

$$I_{refracted} = I_{specular_T} + I_{diffuse_T}$$

$$I_{diffuse_T} = K_d I_d (-N \cdot L)$$

$$I_{specular_T} = K_s I_s (T \cdot V)^{n_s}$$

$$I_{refracted} = K_d I_d (-N \cdot L) + K_s I_s (T \cdot V)^{n_s}$$

plug in givens

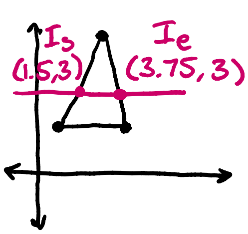
$$= (1)(I)(-N \cdot L) + (1)(I)(T \cdot V)^{ns}$$

$$= I\left[(-N \cdot L) + (T \cdot V)^{n_s}\right]$$

(k) Explain the main difference between Gouraud and Phong shading. Explain which of these models will produce more accurate results and which will be more efficient.

The main difference between Gouraud and Phong shading is that Gouraud shading uses bilinear interpolation, while Phong shading does light computations on every single pixel. Phong shading will have a higher computation cost, and can produce more accurate results with less polygons, while Gouraud shading will be faster, but less accurate.

(l) Let $v_1 = (1, 2), v_2 = (4, 2), v_3 = (3, 6)$be the vertices of a 2d triangle after projection. Let the RGB light intensities in these vertices be $I_1 = (1, 1, 1), I_2 = (2, 2, 2), I_3 = (3, 3, 3)$. Compute the interpolated intensity at location $p = (3, 3)$ when using the Gouraud shading algorithm.



$$I(x_u, y_u) = \frac{x_u - x_s}{x_e - x_s} I_e + \frac{x_e - x_u}{x_e - x_s} I_s$$

$$I_s = \frac{y_3 - y_s}{y_3 - y_1} I_1 + \frac{y_s - y_1}{y_3 - y_1} I_3$$

$$I_e = \frac{y_3 - y_e}{y_3 - y_2} I_2 + \frac{y_e - y_2}{y_3 - y_2} I_3$$

1. find I_s and I_e

$I_s = \frac{6-3}{\ldots}(1,1,1) + \frac{3-2}{\ldots}(2,2,2) = \ldots (3,3,3) = (1.5, 1.5, 1.5)$

1. find I_s and I_e

$$I_s = \frac{6-3}{6-2}(1,1,1) + \frac{3-2}{6-2}(3,3,3) = \frac{3}{4}(1,1,1) + \frac{1}{4}(3,3,3) = (1.5, 1.5, 1.5)$$

$$I_e = \frac{6-3}{6-2}(2,2,2) + \frac{3-2}{6-2}(3,3,3) = \frac{3}{4}(2,2,2) + \frac{1}{4}(3,3,3) = (2.25, 2.25, 2.25)$$

2. use I_s and I_e to find $I(3,3)$

$$I(3,3) = \frac{3-1.5}{3.75-1.5}(2.25, 2.25, 2.25) + \frac{3.75-3}{3.75-1.5}(1.5, 1.5, 1.5) = \boxed{(2,2,2)}$$

(m) Explain the difference between using vertex normals and face normals. Explain how vertex normals can be computed from face normals.

When using vertex normals, generally more computations will occour, and we will be computing in reference to edges while face normals describe the direction a face is pointing. Vertex normals can be computed from face normals by normalization.

(n) Explain the differences between global and local illumination models. Describe which models provide with grater realism and which are more efficient.

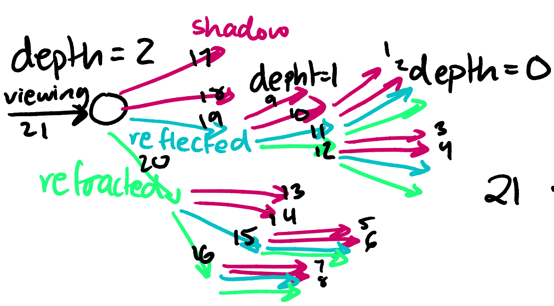
Global illumination takes into account all the lights and surfaces within a scene, to try and estimate how the light will interact, which emulates how light bounces in real life creating a realistic look. Local illumination models only consider the light intensity at a single point, meaning it is much faster but can look funny at times.

(o) Explain the fundamental difference between ray tracing and radiosity. Which method is more suitable for real time rendering with the existing OpenGL pipeline architecture.

The fundamental difference between ray tracing and radiosity is that ray tracing starts from the camera, and follows light back to the sources, while Radiosity simulates light more similarly to real life, where it starts at the light sources and diffuses across a scene. Radiosity is more suitable for real time rendering with the existing OpenGL pipeline architecture.

(p) Explain the difference between reflected rays, refracted rays, and shadow rays. Given an image with 10 pixels, and a scene with two light sources, what is the total number of rays that need to be traced (including reflected, refracted, and shadow rays) assuming that a ray tracing depth of 2, that all surfaces allow for both reflection and refraction, and that reflected/refracted rays always hit another surface.

Reflected Rays are rays that bounce off of a surface and are visible by the light source, refracted rays are rays that move through surfaces and are visible by the light source and shadow rays are rays that are not visible by the light source.



at depth=0, only shadow rays are traced

21 rays traced per pixel x 10 pixels = 210 rays traced.

(q) Explain how can we use the reflected ray direction R_v in computing specular reflection due to a light source pointed by a vector L.

We can use the reflected ray direction Rv in computing the specular reflection due to a light source pointed by a vector L by using the following equation:

$$I_{\text{local reflected}} = k_a I_a + k_d I_d (N \cdot L) + k_s I_s (V \cdot R)^{n_s}$$

$$= k_a I_a + K_d I_d (N \cdot L) + k_s I_s (L \cdot R_v)^{n_s}$$

we prefer $(L \cdot R_v)$ over $(V \cdot R)$ because we have to compute the secondary ray R_v anyways to trace it

(r) Given a ray through the pixel (x_i, y_i, z_i) in camera coordinates, explain how to convert this ray to world coordinates. Assume that the camera is located at point c and has a coordinate system defined by vectors u_1, u_2, u_3.

$$M_{wc \leftarrow vc} = M^{-1}_{vc \leftarrow wc} = T(CoP)R^T$$

$$= \begin{bmatrix} | & | & | & | \\ x_v & y_v & z_v & cop \\ | & | & | & | \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where,

$$x_v = \frac{VP \times z_v}{|VP \times z_v|} \quad y_v = z_v \cdot x_v \quad z_v = \frac{RP - CoP}{|RP - CoP|}$$

(s) Given a ray with position $p_0$and orientation u (a unit vector), and given a sphere with position $p_c$and radius r, write the quadratic equation that has to be solved to determine the ray parameter at the point of intersection between the ray and the sphere.

$$|P_o + S_u - P_c|^2 - r^2 = 0$$

$$(P_{ox} + S_{ux} - P_{cx})^2 + (P_{oy} + S_{uy} - P_{cy})^2 - r^2 = 0$$

where $P_o = p_o + su$

(t) Let t be a triangle defined by vertices $v_1 = (1, 2, 3), v_2 = (4, 2, 4), v_3 = (3, 8, 2)$. Let r be a ray with position$p_0 = (3, 3, 0)$ and direction $u = (0, 0, 1)$. Write the system of equations that has to be solved to find the intersection of the ray with the triangle using the barycentric coordinates method. After computing the barycentric coordinates t the point of intersection explain how to verify that the ray indeed intersects the triangle.

$$\begin{bmatrix} \beta \\ \gamma \\ \ \end{bmatrix} = \begin{bmatrix} | & | & | \\ (V_2 - V_1) & (V_3 - V_1) & -u \\ | & | & | \end{bmatrix}^{-1} \begin{bmatrix} P_o - V_1 \end{bmatrix}$$

$(V_2 - V_1) = (4,2,4) - (1,2,3) = (3,0,1)$
$(V_3 - V_1) = (3,8,2) - (1,2,3) = (2,6,-1)$
$-u = (0,0,-1)$
$P_o - V_1 = (3,3,0) - (1,2,3) = (2,1,-3)$

$$\begin{bmatrix} \beta \\ \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} 3 & 2 & 0 \\ 0 & 6 & 0 \\ 1 & -1 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix} = \begin{bmatrix} 5/9 \\ 1/6 \\ 61/81 \end{bmatrix}$$

we can verify this intersection by checking $\beta \geq 0$, $\gamma \geq 0$, and $1 - \beta - \gamma \geq 0$ is true, which in this case it is. So we can find the point by using the following:

$$P_{intersect} = \alpha V_1 + \beta V_2 + \gamma V_3 = (1 - \beta - \gamma) V_1 + \beta V_2 + \gamma V_3$$
$$= (1 - (5/9) - (1/6))(1,2,3) + (5/9)(4,2,4) + (1/6)(3,8,2)$$
$$= (3, 3, \tfrac{61}{18})$$

(u) Explain how distribution ray tracing can be used to perform anti-aliasing without increasing the computational cost.

Distribution ray tracing can be used to preform anti-aliasing without increasing computational cost by simply sending a random ray through a subpixel inside each pixel.

(v) Explain the meaning of the form factor $F_{i,j}$and the radiosity of a patch $B_i$in the radiosity method. Write the radiosity equation and explain it. Explain the fundamental problem we face when trying to compute the radiosity of a surface patch.

The form factor F_{ij} is the amount of light transferred from a patch i to a patch j. The radiosity of a patch, Bi in the radiosity method refers to the amount of light leaving a surface patch i.

$$B_i = E_i + S_i B_i^{in}$$

radiosity at patch

emitted light

radiosity coefficient

radient energy recieved from all other surfaces

The fundamental problem we face when trying to compute the radiosity of a surface patch is that it is pretty inefficient, and we must look to iterative solutions to accurately and efficiently compute the radiosity of a surface patch.

(w) Write in matrix form the system of equations that has to be solved to find the radiosity in each surface patch. Explain how many equations and how many unknowns are in the system.

$$B = (I - M)^{-1} E$$

there are n eqvations and n unknowns

$$\begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix} = (I_{n\times n} - M_{n\times n})^{-1} \begin{bmatrix} E_1 \\ \vdots \\ E_n \end{bmatrix}$$

(x) Explain the need for an iterative solution of the system. Explain the difference between the Jacobi iteration and the Gauss-Seidel iteration.

A direct solution using the equation for radiosity is inefficient, as it uses complicated matrix algebra. An iterative solution can help reduce these computation costs while still being accurate. Jacobi iteration does not use the updates of B[i] until the next iteration, meaning it stays very true to the original formula, but it is less efficient than the Guass-Seidel iteration method where updates of B[i] are used immediately resulting in faster convergence.