Algorithm 7 Ray

Introduction to Computer Graphics, Fall 2015

Solution Key

1. The high-level view of our ray tracer is exactly the same as for intersect, except for a few additions. Below is the high-level pseudocode for intersect.

Algorithm 1 RAY-TRACE(Scene, Canvas)

for point ∈ Canvas do
    cast a ray to find the nearest object
    if ray intersects an object then
        for each light do
            cast a ray to the light and evaluate the lighting equation
        end for
    else
        Canvas[pt] = background color
    end if
end for

[1 point] What needs to be changed / added to make this a full implementation of the Phong lighting model?

[1 point] What needs to be changed / added to make this a full-fledged ray-tracer?

(Note: Just specify what changes need to be made – no pseudocode please.)

Solution:

(a) 1 point: Line 6 should be changed to add specular contributions, attenuation and texture mapping
(b) 1 point: Line 6 should be changed to add reflection. Shadows should also be mentioned somewhere.

2. [2 points] Given vector \( \vec{v} \) from the surface to a light and the surface normal \( \vec{n} \), find the equation for the vector \( \vec{r} \) which is the reflection of an incoming light ray about \( \vec{n} \). Write your equation in terms of vector operations. How do you compute the color contributed by the reflected ray? Give a brief description.

Solution:

\[ \vec{r} = 2\vec{n} (\vec{n} \cdot \vec{v}) - \vec{v} \]

To compute the color contributed by the reflected ray, trace the reflected ray (hint: use recursion). For the point with the smallest non-negative \( t \) value, compute the color using the lighting model. The contribution is that color multiplied by the global specular coefficient and the object’s reflection color.

3. [1 point] Is ray tracing a local or global illumination algorithm? Why?

Solution: Ray tracing is a global illumination model: it takes into account shadows, recursive specular (but not diffuse) reflection, and refraction.

4. [1 point] What are two cases where a surface will not be affected by a light source?

Solution:

(a) when there is an object intersecting the vector from the point of intersection on this object to the light source in question
(b) when the surface is not facing the light (the surface normal and the light vector form an angle of at least 90 degrees)
(c) when the surface does not reflect the color of the light (i.e. if the light is purely blue and the surface is purely green)

5. [2 points] Recall that we can think of texture mapping in two steps. First, mapping from the object to the unit square, and second, mapping from the unit square to the texture map. Let \( u \) and \( v \) be the \( x \) and \( y \) values in the unit square that a particular point on an object gets mapped to in the first step. Note that \( u \) and \( v \) are calculated differently depending on the object. From here, how do you find the coordinates \( (s,t) \) to look up in a texture map in terms of \( u, v, j, k, w \) and \( h \), where \( j \) and \( k \) are the number of repetitions in the \( x \) and \( y \) directions, respectively, \( w \) is the texture width, and \( h \) is the texture height? (You may assume that both the \( u,v \) and \( s,t \) coordinate systems are oriented with \( (0,0) \) in the same corner.)

Solution:

\[ s = \lfloor u * j \rfloor \cdot w \]
\[ t = \lfloor v * k \rfloor \cdot h \]
Here’s an alternative answer that’s faster and (possibly) easier to understand.

\[
\begin{align*}
  s &= (\text{int}(u * j * w)) \mod w \\
  t &= (\text{int}(v * k * h)) \mod h
\end{align*}
\]

6. [1 point] Given the ambient, diffuse, and specular components of a surface point’s color, how would you add in the color of its texture map at that point? (Hint: You’ll need both the color of the texture map and the blend value in the lighting equation.)

Solution:

\[
\text{final color} = a\text{ color} + (\text{blend}) \times (t\text{ color}) + (1 - \text{blend}) \times (d\text{ color}) + s\text{ color}
\]

7. [1 point] What is the purpose of the specular exponent in the Phong lighting model?

Solution: The exponent determines how quickly specular reflectance falls off. The larger the exponent, the shinier the surface.

8. [Extra credit] Compare and contrast radiosity and raytracing: what are the advantages and disadvantages to each? How might you combine the two - or what might you use instead?

Solution: Raytracing renders the scene from one specific viewpoint and only cares about light that could enter the eye. It is good for getting specular highlights and specular reflection, but it needs ambient hack to simulate diffuse reflection.

Radiosity divides the scene into patches that emit and/or reflect energy. The energy received from other patches for one patch is integrated to allow soft shadows and color bleeding, which are effects of diffuse reflection. Radiosity is also view-independent as the viewpoint doesn’t enter into the equation.

To combine raytracing and radiosity, first perform a view-independent radiosity pass to solve diffuse color for all patches. Then recursively raytrace the scene, sampling the specular-to-specular and diffuse-to-specular transfer among patches. To do this, construct reflection frustums about the points of reflection. These frustums use standard scan conversion and z-buffering to record the amount of diffuse light “seen” from the sample point. If a specular patch is seen by the frustum, then another frustum is cast recursively. This is one way of combining the two algorithms, but certainly not the only way.

Nowadays, it is more common to see stochastic methods used instead of combining radiosity and raytracing.