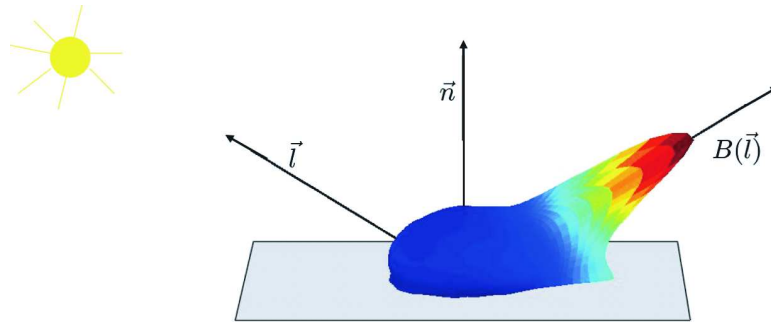


# Question Set 11

CS 420

## Chapter 14

1. Given  $\vec{l}$  and  $\vec{n}$ , derive the equation of the bounce vector.



2. What happens to light intensity as the angle between the bounce vector and the viewing position increases?
3. Describe the visual effect of using true surface normals for the geometry of, say, a sphere.
4. What is required to smoothly shade a sphere?
5. Compare and contrast performing shading calculations in the vertex shader versus performing shading calculations in the fragment shader.
6. Explain how diffuse intensity is calculated.

7. Consider the following fragment shader, keeping in mind that `uLight`, `uLight2`, and `vPosition` are expressed in eye coordinates.

```
#extension GL_EXT_gpu_shader4 : enable
uniform vec3 uLight, uLight2, uColor;
uniform sampler2D uTexUnit;

varying vec3 vNormal;
varying vec3 vPosition;
varying vec2 vTexCoord;

void main() {
    vec3 tolight = normalize(uLight - vPosition);
    vec3 tolight2 = normalize(uLight2 - vPosition);
    vec3 normal = normalize(vNormal);

    float diffuse = max(0.0, dot(normal, tolight));
    diffuse += max(0.0, dot(normal, tolight2));
    vec3 intensity = uColor * diffuse;

    gl_FragColor = vec4(intensity, 1.0);
}
```

- (a) Explain the `tolight` calculation.
- (b) Why is the `max()` function used?
- (c) What, exactly, is computed by `dot(normal, tolight)`?
8. How would you implement an ambient light component in the fragment shader?

9. Consider the following fragment shader:

```
#extension GL_EXT_gpu_shader4 : enable
uniform vec3 uLight, uColor;

varying vec3 vNormal;
varying vec3 vPosition;

void main() {
    vec3 viewDir = normalize(-vPosition);
    vec3 lightDir = normalize(uLight - vPosition);
    vec3 h = normalize(viewDir + lightDir);
    vec3 normal = normalize(vNormal);

    float diffuse = max(0.0, dot(normal, lightDir));
    float specular = pow(max(0.0, dot(h, normal)), 64.0);

    vec3 intensity = vec3(0.1, 0.1, 0.1) + uColor * diffuse
        + vec3(0.6, 0.6, 0.6) * specular;

    gl_FragColor = vec4(intensity, 1.0);
}
```

- (a) What color is the ambient, diffuse, and specular light?
- (b) Explain the computation of `viewDir`.
- (c) This fragment shader uses the halfway vector. Modify it to use the bounce vector.