CG Programming II (VGP 352)

Agenda:

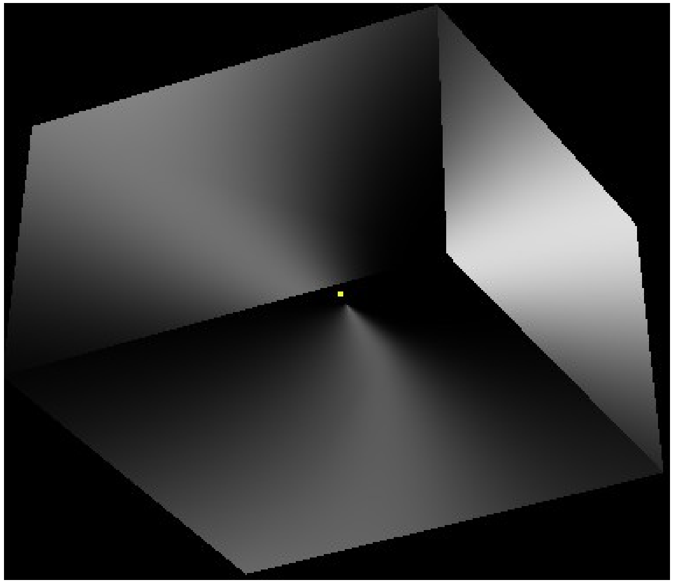
- Discuss last week's assignment.
  - Common problems with assignment #2.
- More BRDFs
  - Why is the Cook-Torrance model poor for metals?
  - How can the model be improved?

Common Problems with Assignment #2

- Problem #1: normalizing the light position *before* passing it to the vertex shader.
  - For the simple per-pixel Phong shader, this *is* correct. That shader does no processing on the light in the vertex shader. It just passes the interpolated value directly to the fragment shader.
  - The bumpmap shader has to transform the light position to surface space *before* passing it to the fragment shader. In this case, the position must be passed to the vertex shader.

Common Problems (cont.)

- Problem #2: Using the wrong eye direction.
  - `e = gl_ModelViewMatrix * gl_Vertex` gives the vector *from* the eye *to* the vertex. This is what the GLSL `reflect` function expects.
  - If you're calculating h by hand, you want the vector *to* the eye *from* the vertex.



Common Problems (cont.)

- Problem #3: Not updating the light uniform each frame.
- Problem #4: Not enabling the texture with the bumpmap bound.

Why does C-T fail for metals?

- The Cook-Torrance model relies heavily on Fresnel attenuation of specular reflection and on uniform diffuse reflection.
  - Metals exhibit very little Fresnel attenuation.
  - Metals exhibit very little uniform diffuse reflection.
- In this sense, Phong or Blinn lighting is *much* better for metals...but we *can* do better.

How Metals Reflect Light

- Two main components to metallic reflection:
  - A mostly pure specular component a la Phong or Blinn.
  - A *directional* diffuse component.
    - Note: In this case, specular means that the color of the light is reflected and diffuse means that the color of the material is reflected.
  - **None** of the models that we have studied have a directional diffuse component...therefore we need a new model!

Enter the Lafortune Model

- Remember the classic Phong lighting model:

$$f(l, e) = \rho_s C_s \cos^n \alpha$$

  - Alpha is the angle between the eye-vector and the ideal mirror reflection vector.

$$\cos\alpha = l_m \cdot e$$

  - l_m can be calculated a couple ways:

$$l_m = 2 \times (n \cdot l) n - l$$

$$l_m = l^T \left(2\mathrm{nn}^T - I \right) = l^T M$$

Lafortune (cont).

- Using some math that I won't cover, we can generate a matrix *M* that is a diagonal matrix.
  - We can also assume $M_{1,1} = M_{2,2}$.
- If we could come up with a series of the matrices, for each color component, to modify the specular lobes, we could approximate directional diffuse reflection *and* specular reflection.

$$K = \Sigma_i \left[l^T M_i e \right]^{n_i}$$

$$K = \Sigma_i \left[M_{i\,1,1} l_x e_x + M_{i\,2,2} l_y e_y + M_{i\,3,3} l_z e_z \right]^{n_i}$$

Lafortune (cont.)

- If we approximate specular, directional diffuse, and uniform diffuse each with a single term, that means (2 coefficients + 1 exponent) * 3 terms * 3 colors = 27 values to pass into the shader.
- With that data, the math is pretty straight forward.

For next time...

- We'll look at some Lafortune data and how it applies to some models.
- The current BRDFs ignore surface anisotropy. We'll look at some other BRDFs, including a simple hair / fur shader, that take this into consideration.

Questions?

Legal Statement

- This work represents the view of the authors and does not necessarily represent the view of IBM or the Art Institute of Portland.
- OpenGL is a trademark of Silicon Graphics, Inc. in the United States, other countries, or both.
- Khronos and OpenGL ES are trademarks of the Khronos Group.
- Other company, product, and service names may be trademarks or service marks of others.