

Homework 2

Due Monday, March 5, 2007 at 11:59pm (written portion in class on Monday)

The purpose of this homework is to play around a bit with Cg and to work out some of the sampling issues regarding textures.

1. Marble Teapot (40 pts)

In this exercise you will implement Ken Perlin's marble shader in real-time using Cg. Download the code provided and use the CgFragmentProgram template to start your work. To understand how to do a marble shader, read the description of the shader in Ken's 1985 SIGGRAPH paper "An Image Synthesizer." You are to use the same algorithm using the texture coordinates provided. The result should look something similar to that of the marble vase in his paper.

2. Fun with Fourier transforms (15 pts)

In this exercise we will prove some of the properties of Fourier transforms, which are important to discussion of texturing as we discussed in class. These are pretty simple proofs, but they are good to understand.

Assume the following formulas for the Fourier transform and inverse transform:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t} dt$$

And that convolution is:

$$h(t) = f * g = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

Now prove the following:

1) Shift property:

$$f(t-a) \xrightarrow{FT} F(\omega)e^{-j\omega a}$$

2) Scaling property:

$$f(at) \xrightarrow{FT} \frac{1}{|a|} F\left(\frac{\omega}{a}\right)$$

3) Convolution Theorem:

$$f(t)*g(t) \xrightarrow{FT} F(\omega)G(\omega)$$

3. Explanations (5 pts)

Explain how the proof of 2.2 helps explain the (intuitive) reason that a higher resolution texture will contain more high-res detail than a low resolution texture for a given texture. Assume that the texture will be filtered for antialiasing with a box filter.

4. Texture caching (20 pts)

Suppose you have a large texture that you are texture mapping onto an object, and that the texture is too big to fit into the texture cache. Which do you think will run faster when the texture is minimized: running with standard texture mapping enabled, or enabling mipmapping and then using the mipmap to texture the quad? Why?