Anh Tran

Assignment 4

How-to paper for Cage Shader with cgfx for Maya

CS 4300

**Introduction**

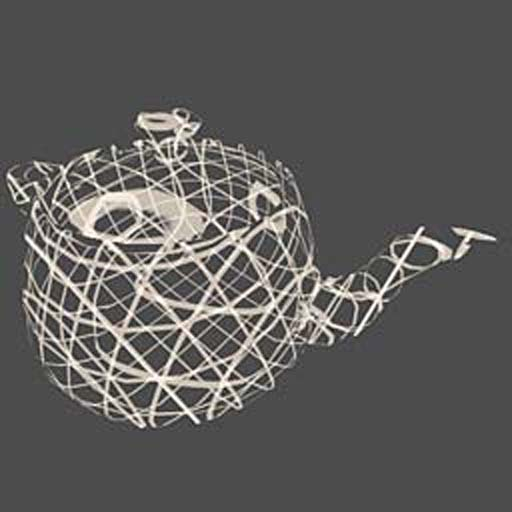
Shading is one of the most important processes in 3-D modeling. It assigns shades, lightings, color to the shape using different techniques, such as vertex, pixel shading. One of the most used tools for this process are NDIVIA library that contains a collection of different shaders and Maya software for constructing the image. This paper is going to analyze the Cage shader from this paper and describe the process of creating this shader.

**Methods and Instructions**

This shader is created by procedural texturing that has been pre-calculated by HLSL. The first step is defining two matrixes that are the WorldViewProjection and the World itself. Next you define four tweakable variables: gWireColor, gEmptyColor, gBalance and gScale. Initialize gWireColor to rgba(1, 0.8, 0, 1), gEmpty to black, gBalance (the balance between light and dark parts of the stripe) to 0.1 and gScale (size of a pattern ranging from 0 to 20) to 5.1. Next we define data structures for the shader. Appdata is a structure that get the input of an application and vertexOutput which is a data passed from the vertex shader to the pixel shader. Then we define functions that calculate the vector and pixeld shaders. In mainVS you will return the vertexOutput in which the position is equal to the multiplication the given WorldViewProjector and the vector of 4 numbers Po, and the texcoord0 is equal to the multiplication of matrix Po, WorldViewProjector and a scale. The pixel shader is calculated in strokeTexPS that returns a new modified color in a float4 format. At the end of the program, in Main, we list out all the features, such as VertexProgram, DepthMask, BlendFunc and others, and initialize them. The technique has two passes that renders the image twice and returns the combined result. The difference between two passes is that the first one is focusing on the back facet culling while the second one is focusing on the front.

**Results**

Applying this shader a 3D shape we will get an illusion of write ribbons outlining the figure.



**Discussions**

This shader doesn’t change the existing image. It creates a new one based on the given image. In my opinion, this shader is very artsy and unique. Therefore, it can be used if you want to have an unusual image of a given shape. This shader looks good on simple shapes. For more complex shapes, applying this shader you might get a messy image result.

**References**

NVIDIA (2008), Shader Library. Retrieved from <http://developer.download.nvidia.com/shaderlibrary/webpages/shader_library.html>