Computer Graphics Labs

Abel J. P. Gomes

LAB. 9

Department of Computer Science and Engineering
University of Beira Interior
Portugal
2012

LAB. 8

TEXTURES

1. Learning goals
2. OpenGL Texture Mapping: an Introduction
3. Example: The Texturized Cube
4. Programming exercises

Lab. 8

TEXTURES

These laboratory units cover the basics of texture mapping in OpenGL. Usually, this involves 3 major steps:
- The reading of texture file (e.g., a TGA file) in memory,
- the uploading of the texture to the video memory, and
- the application of the texture onto geometry.

The first step is not strictly necessary because we can create the texture directly in memory.
In the following, we are using snippets of code to clarify ideas and concepts.

1. Learning Goals

At the end of this chapter **you should be able to**:

1. To map 1D and 2D textures onto different geometric objects.
2. To master more advanced texture techniques such as, for example, bump and lightmap texturing.

2. More about textures

Let us leave this to a forthcoming time.

3. Interesting links

Before proceeding any further, have a look at the following links in order to learn more about textures, in particular the first one:
www.gamedev.net/reference/articles/article947.asp
www.gamasutra.com/view/feature/3361/understanding_and_using_opengl_.php
http://glprogramming.com/red/chapter09.html
www.nullterminator.net/gltexture.html
www.opengl.org/wiki/Texture

4. Programming Exercises

1. **Texture mapping by using texture image.** Let us assume that we have some raw RGB image data in a memory buffer, and we want to apply it to a geometric object in OpenGL? As said above, before using OpenGL such a raw texture data, we need to upload it to the video memory. Then, it can be used every time we need while the OpenGL application runs. The usual procedure is as follows:
    a. Upload texture file into the main memory.

b. Transfer it from main memory to video/graphics memory.
c. Map it onto a polygon or surface.

Now, let us create a project to run the program within texturemapping.zip. Then, you must change the program to map the same texture onto the six faces of a cube.

2. **Automatic texture-coordinate generation.**

We can use texture mapping to make contours on geometric models or to simulate the reflections from an arbitrary environment on a shiny model. These effects can be achieved by letting OpenGL automatically generate the texture coordinates for us, rather than explicitly assigning them with **glTexCoord*()** as shown in the previous exercise. The automatic generation of texture coordinates is done using the command **glTexGen()**.

Now, let us create a project to run the program within pottexture.zip. Then, you must change the program to map the same texture onto a torus. For that purpose, use the **glutSolidTorus** primitive instead of the **glutSolidTeapot** primitive.

Afterwards, change the scene in order to move or rotate the torus to see how the texturing behaves.

Hint: to realize how this automatic texturing works have a look at:

http://glprogramming.com/red/chapter09.html

3. **Automatic texture-coordinate generation: again.**

Let us now to texturize the Earth map on a sphere as follows:

http://www.csci.csusb.edu/tongyu/courses/cs520/notes/texture.php

Then rotate the globe around its pole axis.