Computer Graphics
Assignment 3

Deangeli Gomes Neves
1651398

22/05/2014

Summary

This report briefly describes the development of an algorithm that draws objects in a scene and applies textures in these objects. The objects used were a car and a house, and all the elements in the scene have texture. The program creates a split window horizontally which the top region contains a top-down view and bottom region contains the vision of the driver of the car.

1. Introduction

In openGL, texture mapping is the process of using images (or texture) to influence in the color of pixels [1]. Basically, it is used a fixed pattern which maps the image coordinates in a surface. Textures can be one, two, three, or four dimensional, although mapping two-dimensional textures are more usual. One way to think about texture mapping is in terms of two concurrent mappings: the first from texture coordinates to object coordinates, and the second from parametric coordinates to object coordinates, as shown in figure 1. A third mapping takes us from object coordinates to screen coordinates [1].

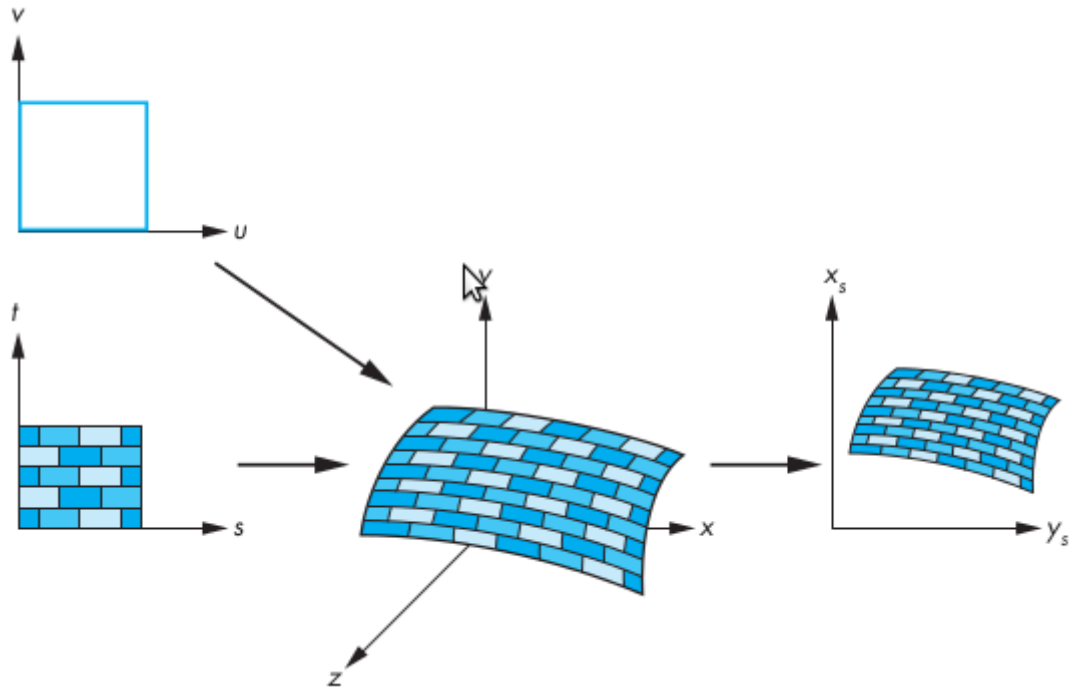

Figure 1. Texture maps for a parametric surface

2. Source code

At the beginning of the program are set some global variables that store several characteristics of the objects that will be designed in the scene, such as attributes and uniforms names; arrays which store the vertices information and texture coordinates; and model-view matrices for all objects. The textures were taken from the Internet, and then using the GIMP, the images were converted to c files. To facilitate debugging of the program, different functions were created in order to encapsulate repetitive tasks. In the initialization function, all these encapsulated functions are used for obtaining all objects' vertices, attributes / uniform / texture / VAO / VBO / program identifiers. There are two cameras in the program: the first creates a vision starting from the sky to the ground, and the second generate a driver view of car. To implement the split screen, the glViewport command was used. it was left a gap between the two sub windows to create a better visualization. To create the steering wheel, it was used (again) glViewport to create a region, in the bottom of the sub-window, where the texture of the steering wheel was mapped.

3. Results

Compiling and running the code, we can see the results show the elements of the scene in two points of view, as shown in the figure 2.

Figure 2. Split window horizontally. Top region: sky-view. Bottom region: driver viewer.

We can see that the objects' textures were correctly applied, therefore giving more realism to objects.

Reference list

[1] E. Angel and D Shreiner. Interactive Computer Graphics: A Top-Down Approach with shader-based OpenGL. Addison-Wesley, 6nd edition, pp. 366-375 2012