

## Texture Mapping for Solar System Simulation(OpenGL Implementation)

Eri Prasetyo W.  
Gunadarma University  
Jl. Margonda Raya 100 Depok  
Indonesia  
eri@staff.gunadarma.ac.id

Bismar Junatas  
Bachelor Master Degree Program  
Gunadarma University  
Indonesia  
lepen\_72@student.gunadarma.ac.id

Vega Valentine  
Bachelor Master Degree Program  
Gunadarma University  
Indonesia  
slaved\_jepun@student.gunadarma.ac.id

### Abstract

*Texture mapping is one of the techniques in computer graphics for image processing applications. It has been used to add realism to computer graphics, images were rendered became more interesting. This technique made a simple image looks closer to the original object in real life. In this paper, we explain about how to apply texture mapping technique using OpenGL library and also applying texture mapping in solar system simulation using Microsoft Visual Basic language supported with OpenGL library. Open GL was made by Saadat Ali Shah, using OpenGL library which is known as an open source software that compatible to any programming applications.*

**Keywords:** *Texture Mapping, Computer Graphics, Solar System, Simulation, Open GL.*

## 1 Introduction

### 1.1 Texture Mapping

Texture mapping is a method for adding detail, surface textures, or colors to a computer-generated graphics or 3D models. This technique basically patching a texture image onto an object, usually applied to a surface of a shape or polygon [3].

Generally, mapping a texture requires these steps [4]:

1. Create a Texture Object and Specify a Texture for That Object

A texture is usually thought of as being two-dimensional, like most images, but it can also be one-

dimensional or three-dimensional. The data describing a texture may consist of one, two, three, or four elements per texel and may represent an R, G, B or A quadruple, a modulation constant, or a depth component.

2. Indicate How the Texture Is to Be Applied to Each Pixel

When mapping an image onto an object, the color of the object at each pixel is modified by a corresponding color from the image. There are several functions to compute the final RGBA value from the fragment(object) color and the texture image data.

One of those functions called Replace Mode. It uses the texture color as the final color, in which the texture is painted on top of the fragment.

Another method is using the texture to modulate or scale, the fragment's color. This technique is useful for combining the effects of lighting with texturing. Finally, a constant color can be blended with that on the fragment, based on the texture value.

3. Enable Texture Mapping

You need to enable texturing before drawing your scene.

4. Draw the Scene, Supplying Both Texture and Geometric Coordinates

We need to indicate how the texture should be aligned relative to the fragments it is being applied before the texture glued on. That is, we need to specify both texture coordinates and geometric coordinates as you specify the objects in your scene.

## 1.2 Solar System Simulation

Solar system consists of sun, planets, and moons. Solar system simulation gave us some information about the orbit of planets and moons, and of course the sun [2]. To run this simulation, an adequate knowledge dealing with the solar system modelling is required. It is related with coordinate determination, object movements, and object formation. Solar system modelling deals with:

### 1. Mapping

In computer graphics, mapping method has a lot of functions. Basically, mapping is used to make a model look closer to the real image by adding some texture. Texture mapping is just one kind of the basic mapping method (further about texture mapping has been discussed in section 1). Other kinds of mapping methods are Bump mapping, Normal mapping, Parallax mapping or Offset Mapping, and Displacement mapping.

### 2. Bresenham's Algorithm

Bresenham's line algorithm is an algorithm that determines which points in an n-dimensional raster should be plotted in order to form a close approximation to a straight line between two given points. Bresenham's algorithm was one of the earliest algorithms which was developed in the field of computer graphics.

### 3. Transformation Matrix

Transformation matrix is a matrix changing by an algebra function. In this paper, transformation matrix was used to rotating, translating and scaling.

Transformation matrix was used in computer graphics oftentimes and furthermore was used in modelling animation.

## 2 Application Design in OpenGL

### 2.1 Texture Mapping with OpenGL

OpenGL (Open Graphics Library) is a specification standard which defines an API platform language for making an application resulting 2D and 3D computer graphics. Its interface consists of more than 250 different functions which allows it to draw complex 3D scenes, from the simplest and easy scene to a more complicated one.

OpenGL is developed by Silicon Graphics Inc. (SGI) in 1992. It is a popular application in video game industry which is a competitor of Direct3D application developed by Microsoft Windows. OpenGL development has raised to a CAD, Virtual Reality, knowledge visualization, informatics visualization, flight simulation, and video game development.

Solar system simulation needed certain techniques. One of the techniques is texture mapping which needed to adding realism to the simulation. OpenGL currently has the statements allowing texture mapping functions, such as:

#### 1. glBindTexture()

glBindTexture statement form the geometric object texture.

#### 2. GL\_TEXTURE\_1D, GL\_TEXTURE\_2D, GL\_TEXTURE\_3D, and GL\_TEXTURE\_CUBE\_MAP

GL\_TEXTURE works as the texture parameter destination from the geometric object. It is also a symbolic constant which usually follow the statement glEnable() or glDisable(). These two statements is to activate or deactivate texturing in geometric objects.

#### 3. gluSphere QObj, x, y, z

This statement is to make a spherical object as a ball with x, y, and z parameters so that the object become 3D. gluSphere draws a sphere of the given radius centered around the origin. The sphere is subdivided around the z axis into slices and along the z axis into stacks (similar to lines of longitude and latitude).

#### 4. glTexParameteri

This statement determines the parameters used in the texture. Used as the integer texture with two parameters, magnification and minification.

### 2.2 Solar System Simulation with OpenGL

- First, loading a picture into the program
- Second, after the image loaded, displaying the image using the code below:

1. glColor3f 1, 0.6, 0 is a OpenGL statement to display colors in an object, on the listing above, this statement is to give colors to the sun, 1 for red, 0.6 for green, and 0 for blue.
2. glRotatef Rt, 0, 1, 0 Rt, 0, 1, 0 to rotate the sun. Rt in the statement above means rotation done in Rt0 value in a clockwise direction. The second, third, and fourth parameters define the rotation fuse, which is the x, y, and z axis. If the third parameter value is 1, it means the rotation done in y axis.
3. gluSphere QObj, 3, 20, 20 used to display sun as a sphere which has the value 3 for diameter, has 20 slices and 20 stacks, and also call the QObj function which is used to give texture in gluQuadric mode with true value and normal with smooth value.

4. In the listing above, there are no statement to control the migration of the sun because the sun doesnt do any revolution. Unlike the sun, planets are doing revolution of the sun in a certain orbits. In order to enable the planets move along its orbit in this simulation, there are the statement `glTranslatef 10!, 0!, 0!..` This statement is to make a translation of an object with 10 position for x axis, and 0 for both y and z axis.
- Third, display the sun flare if user chooses sun in the combobox using the code below:
    1. `glEnable GL BLEND` is used to unite two objects. In this program, `GL BLEND` used to unite 2D texture with the texture owned by sun and planets. To deactivate `GL BLEND` mode, use `glDisable GL BLEND`.
    2. `glColor4fv SunClr(0)` used to display `SunClr(0)` color consist of four components. Each component reached by calling the `SunClr` function.
    3. `glPushMatrix` used to temporarily store all statements between `glPushMatrix` and `glPopMatrix` into a stack. `glPopMatrix` used to pop statements in the stack.
    4. `glRotatef -25, 1, 0, 0` dan `glRotatef Rt, 0, 0, 1` used to rotate the sun. `-25` and `Rt` means rotation done in a clockwise direction in 25o value. The second, third, and fourth parameters denote the rotation fuse which is x, y, and z axis.

### 3 Solar System Simulation in MS Visual Basic (OpenGL Library)

In this section we will explain about Solar System simulation program which made with Visual Basic 6.0 (using OpenGL library). OpenGL library has been explained in the second section of this paper.

This program consist of three modules, two forms, and a class module. First we will examine the module which the program have. This program has three modules,i.e. :

#### 1. Hook Module

This module works to configure program view if the windows size of this program has changed. This Module comprises three parts, first is declaration of the variables which are used, second is hook and unhook function, and last is windows proc. function. Hook Function contains statement which works to begin subclassing which call by the main program. Unhook function is the opposite of hook function,which is this function to stop the module. Windows proc. function

is a function which call by hook function at this module. This function defines the size of solar system information program windows.

#### 2. OGL Module

This Module works when the program loads picture at time the program runs. Besides, this module manages all of the property it has.

#### 3. XP Module

This Module works to give XP Style to the exe program. XP module function is not giving big contribution to the main program.

Hereafter we will introduce all forms available on this program. This program has two forms, i.e. :

#### 1. Main Form

This form performs the solar system simulation.The Form also manage the speed of planet revolution to the sun.

#### 2. Form About

This form has explanation and information about the programmer. The form only appears when user call that form with command button "about" from the main form.

The program will appear as shown below:

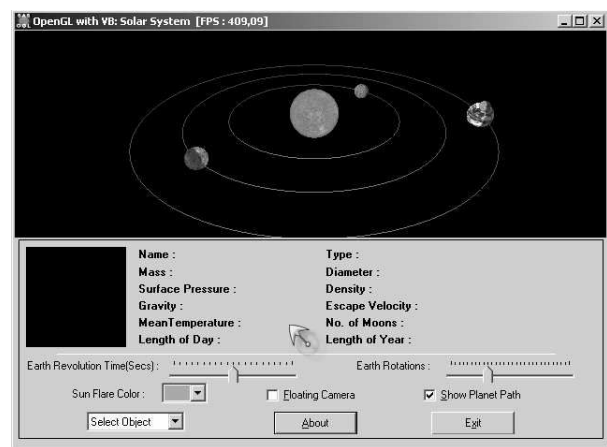


Figure 1. Solar System Simulation Program

This program has a feature to change the Sun flare color, it is the Sun Flare Color combo box.

This application also has VB\_timer class module. VB\_timer module giving FPS and Rt3 value which used in floating camera. Floating Camera check box change the camera view by changing Rt3 value, so when sub render\_glview1 called to display sun and planets, Rt3 value being used is the new Rt3 value, so the camera view is changed.

The program also giving choices whether the user wants to see orbit lines or not. The choices determined by Show Planet Path check box.

## 4 Conclusion

Solar system simulation was made by computer graphics, using some texture mapping techniques to have more realistic simulation.

In this paper, we explain a program, made by Saadat Ali Shah [1], using OpenGL library which is known as an open source software that compatible to any programming applications. OpenGL is one of the most common library used in computer graphics. This application was made by integrating OpenGL library onto Microsoft Visual Basic. By using functions provided in OpenGL and Microsoft Visual Basic, texture mapping can be applied to the simulation and form a realistic and natural simulation.

This application hopefully can help us to understand solar system dan of course about computer graphics theory(in this case texture mapping).

In the future, we can enhance this application. Maybe by making the window size more flexible because we still have problem with its static size. Besides, this program still have a bug. If we click on the sun, planets, or other parts of the simulation, the program will terminate automatically. Hopefully on the future, we can fix this problems to make an optimum application of solar system simulation.

## References

- [1] <http://www.programmersheaven.com>.
- [2] Adrifina and Wisnu Maulana. Solar System Simulation Using OpenGL. Technical report, Gunadarma University, 2007.
- [3] Paul & Mark Segal Haeberli. Texture Mapping as a Fundamental Drawing Primitive. 1993.
- [4] Shreiner. Texture Mapping. 2005.