

Rotation, Zoom and Pan in OpenGL

Peter Salvi

June 29, 2008

1 Introduction

This is a memo that explains *one* way to do standard camera movement in a perspective OpenGL window. This method depends exclusively on the GLU functions `gluPerspective` and `gluLookAt`, which means that every movement is given as a change of the following data:

- The eye position in object space (`eye`)
- The central position (where the eye looks at) in object space (`center`)
- The (unit) direction of going *up* on the screen in object space (`up`)
- The clipping planes (`zNear` and `zFar`)

There are also a few parameters these movements depend on:

- The window geometry (`width` \times `height`)
- The field-of-view angle in the *y* direction (`FoVy`)

Although the `FoVy` is fixed in most applications (generally at a value between 45° and 60°), the window geometry can change in the course of the program. Since OpenGL needs to be informed about this, there should be a reshaping function that is called every time a change in the window geometry occurs. As this function is responsible for setting the correct perspective, it should also be called when initializing the OpenGL window.

Another important function is the display function that is called every time the program has to show the OpenGL window. This will take care of the modelview matrix transformations. Section 2 explains these functions and Section 3 and 4 introduces the computations required for the various movements. Calculation of the correct clipping planes is shown in Section 5, and an appendix contains the details of non-trivial computations.

All program snippets and computations are based on the program *SFView* I've written a few days ago.