

CS635: Assignment #0 – Synthetic-World 3D Reconstruction

Out: January 12, 2007, 10:21am

Due: January 26, 2007, 9:29am

Objective

This objective of this assignment is to implement a synthetic simulator to perform basic 3D point reconstruction. Using an OpenGL program, you will have both the projected scene points and the actual scene points – thus verification of your program working correctly is trivial. The next assignment will replace the synthetic (perfect) data with real-world data -- thus organize your program well!

To give you a head start, and to ensure homogeneity, you will be provided with a simple OpenGL/GLUT/GLUI program that reads a 3D model from a program, renders it to the screen, and supports 3D navigation. You will also be given sample 3D models. This program can be compiled using MSVC 6.0 (or later). If you prefer not to use this program, please ask the instructor – in all cases a similar style program must be provided for this assignment.

The assignment consists of modifying the program so that at will you can save the camera parameters and model parameters and then “reconstruct” the scene. The camera parameters include the focal length, near z, field-of-view, and 3D position and 3D orientation of the camera. The model parameters include the 3D positions of the observed scene points (e.g., the vertices of the given 3D model) and their 2D projections.

Detailed Description

Step 0 – Capture

To begin, modify the GUI so that with a single button (“Capture”), you save the current camera and model parameters to a text file. This consists of a “view” of the scene. You may also want to save the current framebuffer image. You must support obtaining two or more views of the scene.

Step 1 – Reconstruction

Next, add another button to the GUI (“Reconstruct”) so that when pressed it loads all saved “views” of the scene and estimates the 3D scene points. As described in class, you can use the camera parameters and the screen-space projections of scene points observed in at least two images to recover their 3D position.

Step 2 – Visualization

Finally, add two checkboxes to the GUI (“View True Points” and “View Reconstructed Points”) that enable rendering the true scene points (e.g., the vertices of the model rendered as large and colored OpenGL points) and the reconstructed scene points (in a different color as the true scene points but ideally should exactly overlay the true scene points).

Bells and Whistles: add extra visualizations and options to view the saved framebuffer images (this requires reading/writing images to disk and to/from the framebuffer).

Grading and Deliverable

Your grade will be influenced by how well your particular object is reconstructed, by the presentation and usability of your program, and by how well you complete the assignment requirements.

I will schedule a brief meeting with each of you and you can demo your assignment to me. A zipped file containing source, compiled executable, and auxiliary files must be emailed to me before the due date.

Software:

1. MSVC/.NET is installed on all department computers.
2. OpenGL is installed on all department computers.
3. You may need to install GLUT on your computer or in your project directory. Information on GLUT can be found at <http://www.xmission.com/~nate/glut.html>
4. You may need to install GLUI on your computer or in your project directory. Information on GLUI can be found at <http://www.cs.unc.edu/~rademach/glui>.
5. The provided rendering program and example 3D models can be found on the course webpage, <http://www.cs.purdue.edu/homes/aliaga/cs635-07/basicmodeler.zip> and <http://www.cs.purdue.edu/homes/aliaga/cs635-07/basicmodeler-models.zip>. In its current form, it is a .NET 2003 solution. To recompile in another .NET version, convert it automatically or simply create a new project with all .cpp and .h files.

Primer on basicmodeler:

- To run the program, simply start the executable and then load a “.sim” file. You can also specify the model file on the command line. The GUI is simple and self-explanatory. To obtain a summary of mouse and keyboard commands, press “h” in the main graphics windows. The .zip file includes a model called “cube.sim” and a current version of GLUI/GLUT for compiling.
- The file format .sim is simple and self-explanatory – you can easily create your own test models.
- In the source code, you only need to look at main.cpp. The modifications for this assignment consist in changes to the GUI calls and refresh calls in main.cpp. Enjoy!