

CS635: Assignment #1 – 3D Reconstruction

Out: February 3, 2009

Due: February 17, 2009, 8:59am

Objective

The objective of this assignment is for you to reconstruct a significant fragment of a 3D object of your choosing. This assignment will build off the previous assignment and using your calibrated camera, you will take 10 or more pictures of an object, reconstruct its geometry, and use texture-mapping to color the object. Remember: precision is very important -- small errors can lead to big calibration errors.

This assignment requires some mental planning work, some lab work, and some programming. The whole assignment is not that much work and mostly requires you to understand the process. I do recommend **STARTING EARLY!** (i.e. -- you will not be able to complete the assignment if you start the day before its due --).

Detailed Description

Step 0 – Image Capture and External Pose Estimation

Using your internally calibrated camera, take at least 10 pictures spanning an angular viewing area in front of your object. Since your camera is now internally calibrated, you only need to determine the camera's external pose. You should choose a few well-chosen and known 3D points across all images and use them to perform pose estimation: 5 or 6 points are enough. You may implement a closed-form (linear) pose estimator or you can use an iterative optimization. The optimization can be of a general nonlinear form or a linear form (with some assumptions of locality). Also, if you take your pictures using a smoothly changing viewing position, you can use the pose of the previous picture to initialize the pose of the new picture.

You should also plan on taking your pictures wisely – they should span a wide field-of-view of the object you choose. The object you choose should also be adequate for the reconstruction, both in terms of its colors (for point selection) and in terms of its aspect-graph (for preventing too many disocclusions).

Step 1 – Reconstruction

To perform the reconstruction of the object, correspond points over at least 2 images (use the mouse) and reconstruct them. You might have points corresponded over 2 images up to all 10 or more images. Regardless, you can calculate their 3D position using a single formulation for linear reconstruction as explained in class. The more points you correspond, the more 3D data you have. You should have at least 10 points reconstructed between any triple of captured images. But this number is rather arbitrary, it should be “as many as you can get”. Thus use it as a minimum guideline, not as a goal. In general, the front half of the object you choose should be “well reconstructed”.

Step 2 – Triangulation and Texture-Mapping

To demonstrate the object, please implement an OpenGL/GLUI/GLUT based browser of your reconstructed object. The virtual viewpoint should be somewhere in front of the object and the closest captured viewpoint surrounding the viewpoint should be used to perform a view-dependent reconstruction and texture mapping of the object. The corresponded (and reconstructed) points of the object should be triangulated and the triangles used for texture mapping. The texture of the viewpoint closest (in viewing direction) to the object should be used to texture-map it.

It is probably easiest if all your points are visible in all images, however this is not necessary.

It is probably easiest if you take pictures from two semi-circular paths in front of the object with one semi-circle vertically higher than the other. This forms a nice pattern of a horizontal row of triangles to use for view-dependent texture-mapping. Again, this is not necessary but recommended.

Important: Your program should include a virtual-trackball like interface to manipulate the reconstructed object **and** to change the virtual viewpoint.

Step 3 – Extra Credit

- a. Implement a linear-pose estimator, such as the one from “Generalised Linear Pose Estimation”, A. Ess, A. Neubeck, L. van Gool, BMVC 2007.
- b. Implement “weighted texture map blending” as opposed to snapping to the use of the closest texture; in other words do view-dependent texture mapping, as in “Efficient View-Dependent Image-based Rendering with Projective Texture-Mapping”, P. Debevec, G. Borshukov, Y. Lu, SIGGRAPH, 1997.
- c. Allow different number of points from different triples of captured viewpoints and perform a geo-morph and blend when switching triples of captured viewpoints. Thus, in addition to the texture changing smoothly, the geometry can also change smoothly.

Tools

To help with this assignment, we will provide you with a *2D triangulation library* and with a basic program for *rendering 3D models* which can either use or replace with your own tools. This software is just to help get you started with a 3D synthetic rendering framework (including the virtual trackball). The programs are on the website.

Grading/Demonstration

Your demonstration will consist of you showing me your program in short demo session to be arranged (in my office). On or before the due date, please provide me with a CD or a zip file with a single directory called “<your-name>-asgn0” containing:

- Windows PC Executable
- Data files (i.e., images)
- Other necessary files, DLLs, etc...

During your demo session, I will use the provided CD/zip-file to grade your program. Your grade will be influenced by how well your particular camera/object is reconstructed, by the presentation and usability of your program, and by how well you complete the assignment requirements.

In this assignment, you may collaborate only to help with the mechanics of the assignment (e.g., using the pad, your camera, etc). *Everybody must take their own pictures!* Practically speaking, this means that nobody should have the same pictures or calibration results.

If you have questions, please come see me ASAP – do not wait until the last moment.

Have fun and good luck!