

CS635: Capturing, Modeling, and Rendering 3D Scenes

Instructor: Daniel G. Aliaga

Classroom: LWSN 3151A

Time: TTh @ 9:00-10:15am

Course Overview

The objective of this course/seminar is to understand the fundamental problems and challenges encountered when capturing, modeling, and rendering 3D structures and objects. The course covers several subjects within computer graphics, computer vision, and computer science so as to provide to the student a full understanding of the capture/model/render pipeline. From this understanding and cross-fertilization of ideas, it is expected that students will in the future be able to develop new and improved approaches.

Topics for the semester will be tentatively extracted from the following topics and based on latest research results:

Basics

OpenGL, GLUI, GLUT, OpenCV

Cameras

Pinhole camera model
Catadioptric camera model
Camera calibration: basics

Numerical Optimization

A brief primer

Passive Features (Epipolar Geometry and Correspondence Searching)

Basics of Epipolar Geometry
1D Correspondence Searching
Finding and defining features

Active Features (Lasers and Structured Light)

Structured Light Methods

Geometry-based Reconstruction

Structure from Motion
Pose Free Acquisition
Multi-view Stereo Reconstruction
Space-time Stereo
Iterative-Closest Point (ICP)

Image-based Reconstruction

Space Carving and Voxel Coloring
Lumigraphs and Lightfields
Light Transport, Inverse Light Transport, and Helmholtz Reciprocity
Photometric Stereo

Advanced Topics:

Shape and Motion under Varying Illumination: Unifying Structure from Motion, Photometric Stereo, & Multi-view Stereo
Photogeometric Structured Light
Exploiting Symmetry for Reconstruction (with a psychological component)
Dynamic models:
One-shot structured light
Photogeometric structured light for moving objects
Appearance Editing and Radiometric Compensation

The course is divided into two parts.

- The first part describes, during the lectures, the aforementioned research methods to be presented by reviewing the latest works in the field. The students will also present informal presentations and summaries about the referenced work. A schedule is to be determined once the semester starts.
- The second part of the course consists of a guided research programming project that progressively implements one of the above methods and/or a derivative to be determined based on student expertise. The intent is to give the students a hands-on experience; depending on the particular case, necessary hardware can be provided, and a publication outlet for the semester project would be an ideal goal.

Grading

The course grade is determined by the performance in a short series of small programming assignments at the beginning, class participation (in the form of an informal presentation of a topic to be selected) and a final project. Each assignment will be evaluated during an interactive session with the instructor. The grade depends on a combination of meeting the requirements, the presentation,

and the sophistication of the solution. There will be no final exam but rather a public demo day at the end of the semester with all projects.

3. Administrative Issues

Deadlines

All assignments must be handed-in by the specified due date/time. An assignment late by up to one day receives a 50% penalty (e.g., if maximum score is 10, it will be a maximum of 5), by up to two days a 75% penalty and after that a 100% penalty. The final project consists of 2 formal presentations (initial background research and final project presentation) and one informal presentation (intermediate progress report). These dates will be established once students and projects are settled. The final project presentations and demonstrations must be on time; otherwise a grade of 0 is given.

Code and Collaboration

All assignments, presentations, and projects must be done individually unless otherwise indicated by instructor. In research, it is highly encouraged to “build upon the shoulders” of others, however due credit must be given to the sources. Unreported copying or plagiarism will give you a failing grade in the course and you will be subject to standard departmental and University policies. For the programming assignments, code obtained from the Internet, books, or other sources may *not* be used. For the final project, previously-written code is permissible pending instructor approval.

Tentative Schedule:

Assignment #0 – Camera Calibration (Jan 22nd – Feb 3rd)

Given image data, proceed to “calibrate” a camera. You may use your own camera or images provided by the instructor. The resulting calibrated camera should be used to reconstruct points on a simple observed object (correspondence can be established, for example, manually via mouse clicking for instance).

Assignment #1 – Example Real-world 3D Reconstruction (Feb 3rd – Feb 17th)

Using the previous assignment, reconstruct a more complete 3D object (triangles and texture mapping). Then render the object within an OpenGL program where you can intuitively control the viewpoint and/or object.

Assignment #2 – Lumigraph (Feb 17th – Mar 3rd)

Further extend the previous assignment to implement a Lumigraph with varying focal-plane. Capture a large set of images of a scene and support varying the apparent “focus” of the reconstructed scene.

Final Project (Mar 3rd – April 30th)

Perform background research, develop, and present a project. Students may select from the following project-topic suggestions. Details and actual project descriptions will be determined during the course. Each student will do a formal “background research presentation”, an informal “mid project presentation”, and a formal “final project presentation”.

- Geometric or Image-based Reconstruction: propose a robust method to do some form of 3D model capture
- Lightfield/Lumigraph rendering: improve upon a particular aspect of this approach
- Point-based rendering: capture or render a model using points
- Your own project...