

CS635: Assignment #2 – Lumigraph++

Out: February 17, 2009

Due: March 3, 2009

Objective

This objective of this assignment is to use explore a different approach to acquiring a scene – building a Lumigraph with some extensions. In particular, you will extend the Lumigraph to include the ability to vary the focal surface/aperture. Thus, you will be able to capture a scene and apparently change the focus/aperture of the virtual cameras. As explained in class, a Lumigraph (very similar to a Lightfield), creates novel views *without* having to reconstruct geometry and *without* having to establish correspondences.

The methods of the assignment are a merger, as well as simplification, of two papers: “Unstructured Lumigraph Renderer” (http://groups.csail.mit.edu/graphics/pubs/siggraph2001_ulr.pdf) and “Dynamically Re-parameterized Lightfields” (http://groups.csail.mit.edu/graphics/pubs/siggraph2000_drlf.pdf). You do not necessarily need to implement all the features of these papers, but something similar in spirit.

The assignment will also build upon parts of the previous assignments.

Detailed Description

Step 0 – Capture

To begin, capture a large set of images from in front of the object while keeping the object more or less centered in the image and spanning as much of the field-of-view as possible (but remember, you will also need to calibrate each image – e.g., compute the camera’s position using the calibration pad). Novel view rendering improves as you increase the number of images and increase the viewpoint density. I recommend you capture as many images as you can, but at least 20 images (i.e. place you camera on a book/table and slide the camera laterally with respect to the scene; this way you can capture images from 20 viewpoints densely packed along a line – conversely you can capture a grid of 5x4 images or more) . You can fully calibrate each image separately but better results will be obtained by using the same internal camera parameters for all views and only re-computing the external camera pose for each image. If you use the checkerboard to establish your camera pose, you will need to ‘click’ on points in order to establish a correspondence for pose estimation (points on the object/scene itself do not need to be corresponded).

Step 1a – Lumigraph Rendering

Using your calibrated images and a proxy of the environment, create views of the scene from a novel viewpoint. The proxy is an approximation of the scene. The simplest proxy is to represent the scene as a plane located in front of and parallel to the captured images by approximately the same distance the scene is from the captured viewpoints.

The unstructured lumigraph renderer uses several criteria to choose which pixels/images to use to construct a novel view. The simplest scheme is to choose the trio of images (or pair of images if you captured along a line) that surround the novel viewpoint (--this essentially becomes view-dependent texture mapping). From these images, you can select the rays that converge on the proxy in front of the camera and use a weighted and blended combination of the rays. Thus, as the viewpoint moves, the selection and combination of rays slowly and smoothly changes. For more details, refer to the aforementioned papers (and to comments in class).

Note that to implement the Lumigraph it is sufficient to use non-OpenGL programming and then only at the end use OpenGL to draw the final reconstructed image (e.g., `glDrawPixels`).

Step 1b – Calculating Weights

The viewpoints of the images you captured should lie approximately on the surface of a plane or a line (both or you choose). For this assignment, you can assume the novel viewpoints will be “near” the line or plane of captured viewpoints. For the case of captured viewpoints on a line, the needed linear interpolation is straightforward. For the case of captured viewpoints on a plane, you should triangulate the captured viewpoints (on the approximate capture plane), determine inside which triangle is the novel viewpoint and use that information to blend the rays from the three (or more if using a wide aperture) viewpoints surrounding the novel viewpoint. You may automatically triangulate the viewpoints, e.g. use Delaunay triangulation, or manually create a triangulation. If you project the observer’s viewpoint onto the triangular patches of the viewpoint surface, you can compute the barycentric coordinates of that observer viewpoint and obtain three weights (u,v,w) that tell you how much of each captured image to use.

Step 2 – Varying the focal surface

In order to simulate varying the focal surface, you simply need to “move the focal surface”. If you use a plane as a proxy, changing the focus corresponds to moving the plane forward or backward with respect to the captured viewpoint. Effectively, this will change what rays you blend together for each desired novel view ray.

Step 3 – Varying the aperture

To simulate varying the aperture, you can increase how many of the images surrounding the novel viewpoint are used and then combine the extracted rays using a weighted aperture filter (e.g., all rays contribute equally, rays “nearest” to the novel viewpoint contribute the most, etc.).

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!