Programming Project 2 – 3D Reconstruction from Two Views

CAP 5415, Principles and Algorithms of Computer Vision, Fall, 2014
Department of Computer Science, Florida State University

Points: 100
Due: Monday, November 10, 2014
Maximum Team Size: 2
Submission: You can submit a hardcopy of your report with source code attached or email your report along with source code as compressed tar archive to liux@cs.fsu.edu (with subject line starting with "Programming Project #2 Submission for CAP5415"). Only one copy from each team is required.

Purpose: To know how to implement the normalized eight-point algorithm for estimating the fundamental matrix, and how to generate 3D models from images.

Background: In computer graphics, 3D models are required for rendering; in computer vision, 3D models provide additional information that can be used to resolve ambiguities inherent in 2D images. While there exist many different ways of generating 3D models, stereopsis is the simplest in that it does not require substantial devices other than a camera. In addition, the principles in stereopsis can be used in other situations (such as Kinect) and the accuracy of the reconstructed surfaces is sufficient for certain applications (e.g., reconstructing Rome using photos on the web). Another aspect of this assignment is to be able to generate good quality textured models from images in VRML (Virtual Reality Modeling Language) format or another format of your choice.

Assignment: 1) Implement a 3D reconstruction algorithm based on a set of correspondences, 2) provide an interface/means to define/refine corresponding points, 3) provide an interface/means to establish/mark triangles/quadrilaterals, 4) generate a texture image for each triangle/quadrilateral by rectifying the given images, and 5) save the results into a VRML model (see the Additional Information Section for a Matlab program) or any other format that supports texture-mapped surfaces (in Matlab, you can use warp, which supports on quadrilateral surfaces).

After you are done with your implementation, apply your program(s) to the Kinect dataset (Credit: Nathan Crock), which is available from http://www.cs.fsu.edu/~liux/courses/cap5415-2014/class-only/kinect.zip (see the Additional Information Section on the format of the dataset). You can pick any two pairs of images of the dataset and for each pair you need to reconstruct at least the visible region of the outlined desk top (in red below), including the books on the desk. For the occluded parts, you can model using prior knowledge (e.g., the desk top is rectangular).

Submission:

- **Report** – You need to turn in a report, summarizing the algorithms you have used/developed, showing significant intermediate results to demonstrate the correctness of your algorithms and programs and documenting your findings.

- **Source code** – You need to attach all the source programs you developed for this programming assignment. Note that if you have used other's programs, you need to give proper references and credits, and you will be graded based on the additional work you have done.

- **Results** – Include in the report a snapshot of each 3D model you have generated and then specify the URLs where the models are available on the web. Note that the results you submitted must be obtained from your own implementation. If you choose a format other than VRML for 3D models, please also describe how they can be rendered and I may request a demonstration if needed.

Grading

- **Report** – 20 points.
  - Description, analysis and justification – 15 points.
  - Performance analysis (e.g., computation time and required manual editing) – 5 points.

- **Correct implementation** – 50 points.
  - Program organization and supporting functions – 15 points.
  - Eight-point algorithm – 20 points.
  - Effectiveness of correspondence interface/algorithm – 5 points.
  - Effectiveness of surface specification interface/algorithm – 5 points
  - Texture mapping – 5 points.

- **Results** – 30 points.
  - Estimated extrinsic camera parameters – 10 points
  - Estimated three dimensional points **–** 10 points.
  - Textured models for the chosen desk top – 10 points.

- **Evaluation using Kinect –** 10 points.

- **Multiple view reconstruction of the scene –** 10 points.

Extra Credit: Please state clearly in your report if you have implemented the following extra credit options.

Evaluation of the reconstruction accuracy. Here you need to evaluate your three dimensional point estimation using that given by Kinect. While the depth estimation (actually the modified disparity values are provided for each image, in order to perform evaluation, you need to register the IR camera that is used by Kinect for depth estimation and the RBG camera (that you use for 3D point estimation) and some of the needed parameters are not known and you may need to estimate some of the parameters. Please refer to the slides (http://www.cs.fsu.edu/~liux/courses/cap5415-2014/notes/vision-06-kinect.ppt) for more information on Kinect.

Multiple view reconstruction of the scene. Here you need to register the estimated three dimensional points from at least five images and create a model of the visual regions of the scene (including at least the floor, two desks, and one sofa).

Additional Information

The dataset includes 20 views with the RGB images named from rgb00.jpg to rgb19.jpg and the depth data (raw disparity from Kinect) are saved as 480 x 640 matrices (first row, second row, and so on, where each row consists of 640 disparity values). A Matlab function is provided at http://www.cs.fsu.edu/~liux/courses/cap5415-2014/assignments/lab2/read_depth_matrix.m to read the disparity data. Note that 2047 means that the value at that pixel location is invalid. The images and disparity files are also available at http://www.cs.fsu.edu/~liux/courses/cap5415-2014/assignments/lab2/ .

A Matlab function for creating a VRML model is given here http://www.cs.fsu.edu/~liux/courses/cap5415-2014/assignments/lab2/make_vrml_model.m . To view the model, a plug-in is needed and one option is Cortona3D (available from http://www.cortona3d.com/Products/Cortona-3D-Viewer.aspx).

In order to do calibrated eight-point reconstruction, intrinsic camera parameters need to be known; for this assignment, you can use the following camera model: $\begin{bmatrix} -525 & 0 & 320 \\ 0 & -525 & 240 \\ 0 & 0 & 1 \end{bmatrix}$. Optionally, you can use the disparity data to calibrate the camera or calibrate a Kinect unit that is available to you using programs available from http://www.ros.org/wiki/kinect_calibration/ .