CS268: MACHINE PERCEPTION

Homework 3: Turning your camera phone into a 3D scanner.

Due November 16th, 2015

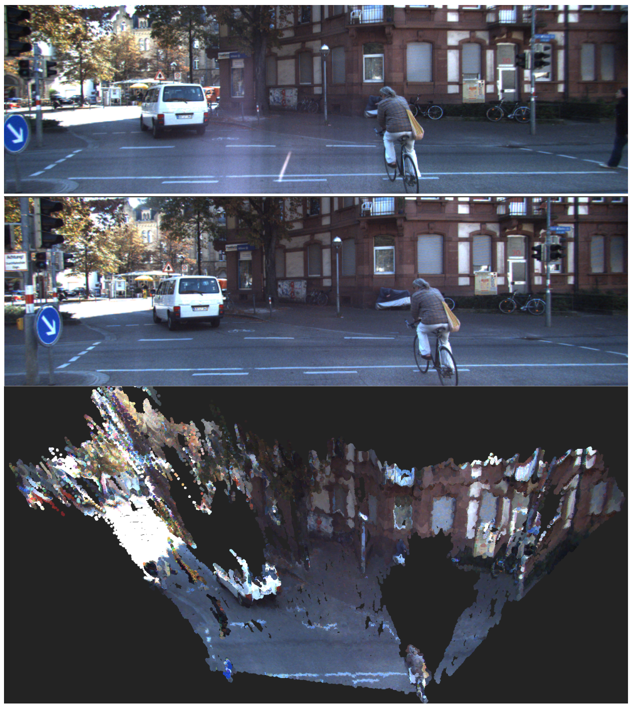

Figure 1: *Right image (top) and left image (middle) yield the resulting textured 3D reconstruction (bottom)*

In this homework, you will build a rudimentary stereo pipeline to reconstruct a coarse geometric model of a scene from two images, similar to Fig. 1. The goal is to use some of the basic concepts you have learned in this course to build simple 3D models using your camera phone.

1 Assigment

Part 1: Camera calibration

Using a publically available toolbox or tutorial, examples below, calibrate your camera to obtain the intrinsic camera parameter matrix K and lens distortion parameters.

- Matlab: `http://www.vision.caltech.edu/bouguetj/calib_doc/`
- OpenCV: `http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html`

Please include K and the parameters of the lens distortion model in your report.

Part 2: Estimate the epipolar geometry

Estimate the epipolar geometry between the image and plot the epipolar lines on the image. Then use the epipolar geometry to rectify the images so that the epipolar lines coincide with scanlines of the image plane. For reference, please refer to section 11.5.1 in the class textbook. Rectifying the images will greatly simplify the dense matching problem in part 3.

Please include images showing epipolar lines before and after image rectification in the report and the code you wrote to perform these two steps in your submission.

Part 3: Dense stereo matching

In the previous homeworks, you produced mosaics using a small number of correspondances between two frames. Here, we need a much larger number of matched points in order to produce as dense of a point cloud as possible. Once the images are rectified in the previous step, finding a corresponding pixel in the second image for each pixel in the first is simplified from a 2D search problem to searching across a 1D scanline. Using this knowledge, find dense correspondances between your two images and compute a dense point cloud. (Chapter 11.5.2 of the text serves as a good guide for this step.)

Please include an image of the dense depth image in your report and the code you wrote to perform this step in your submission.

You can test this step independently from the other steps with the data we have provided on the class website from the KITTI Dataset `http://www.cvlibs.net/datasets/kitti/raw_data.php?type=campus`. Choose any example and run your algorithm on the *rectified* images and visually validate the depth map you obtain.

Visualize your resulting 3D point cloud

Familiarize yourself with a 3D visualiztion tool (e.g. Matlab, Python, Meshlab) and use it to visualize your resulting 3D point cloud. To improve the aesthetics of your visualization, you can use color information from the images to texture your point cloud (you can simply use intensity information from the images to color the points you visualize).

Please include an image of the colored visualization in your report.

2 Note

Keep in mind that each step can be completed independently from the others. We strongly recommend that you work on each step independendently so that you can verify that each one is working before attempting to put the whole pipeline together. We also *highly* recommend shrinking your images to a size smaller than VGA (480x640) while testing some parts of your algorithm (be sure to make appropriate changes to your intrinsic calibration matrix).

3 Grading and Submission

If you put all the steps together and successfully create a 3D point cloud from two images from your camera phone, congratulations you have now built a 3D scanner from your phone at no extra monetary cost! Even without putting all of the steps together, because they can be tested independently, you can still receive most of the credit for the assignment by thoroughly demonstrating your understanding of each of the parts.

A day or more before the deadline, a submission link will be posted on the website for you to upload a zip of all of your code, image files, and an associated report. Collaboration is allowed and generally encouraged, however each individual must submit their own report. If you have any questions regarding this assignment, please direct them to Brian Taylor at btay@cs.ucla.edu. If you do not have access to a camera, try to work with other students to obtain data. If that is not possible, please contact any of the TAs and we will make sure you have access to some data. The goal is for you to build a cool piece of technology using an everyday item that most of us have in our pockets, so good luck and have fun!