PROGRAMMING ASSIGNMENT 3

Due: April 21, 2015 (12:30pm)

Automatic Panorama Stitching ¹

Figure 1: A panorama stitching example using a total of 57 images (taken from [1]).

In this assignment, you will implement a simple image stitching algorithm to automatically generate panoramic images. Your program will take four or more images as input and create a panoramic image by computing homographies, applying warping, and blending images with overlapping regions seamlessly. Specifically, you will implement the following steps:

1. Take pictures,
2. Automatically find correspondences between images,
3. Recover homographies using RANSAC,
4. Warp and blend the images into a mosaic.

A starter pack is provided to you in the course homepage. The details about each step are given below:

Step 1. Taking pictures.
Take at least four photos from the same point of view but with different view directions, and with overlapping fields of view. Be sure that consecutive images overlap at least 30%.

Step 2. Automatically finding correspondences between two images.
In this step, you will implement the process described in [2] to extract SIFT feature points from the input images and find the correspondences between them to determine the overlapping regions. The steps you will follow are summarized as follows but for a full understanding of each step, please read the corresponding section in the paper.

- Detect corner points in each image (use the provided \texttt{harris.m})
- Extract features on interest points as follows:

¹Adapted from the assignment developed by Alexia Efros at Berkeley University.
1. Blur the image with a Gaussian filter, and

2. For each interest point, take the intensity values of the patch surrounding that point as features.

   • Implement feature matching as follows:
     1. Specify interest points for each image
     2. Extract features on interest points for each image will be stitched by using your function.
     3. Calculate Euclidean distance between interest points of two images.
     4. Sort the distances and use D. Lowe’s ratio test given in the starter code.
     5. Take only points that have ratio under the threshold you will specify in range 0-1.

NOTE: If you would like to implement any one of the extensions listed below, you can also use VLFeat implementation of SIFT in this step.

Step 3. Recovering homographies using RANSAC.

The next step is to recover a projective transformation $H$, a $3 \times 3$ matrix with 8 degrees of freedom. One way to recover the homography is via a set of $(p, p')$ pairs of corresponding points between two images, using the relation $p' = Hp$. Hence, you need at least four pairs to form 8 linear equations to solve for the 8 unknowns of $H$. Unfortunately, not all the corresponding points found in Step 3 will be correct matches, there will some false positives. To eliminate the influences of outliers, you will estimate the homography matrix by using RANSAC algorithm as described in the lecture slides. Here, you can use the provided Ransac code (ransac.m).

Step 4. Warping and blending the images into a mosaic.

Using the estimated homography, warp each image into a reference image to compose a mosaic image from the warped images. For that purpose, you can use vgg_warp_H.m function provided in the starter pack. In order to obtain a high-quality result, you need to implement the multi-band blending strategy described in [1], where the idea is to blend low frequencies over a large spatial range, and high frequencies over a short range. Please refer to [1] for details.

Extra Credit:

• Cylindrical mapping. Instead of projecting your mosaic onto a plane, try reprojecting your panorama onto a virtual cylinder. This is particularly useful when displaying really wide mosaics. See cylindrical.pdf for the formulation cylindrical projection.

• Mini planets. Use the stereographic projection to produce funny panoramas as seen in http://www.miniplanets.co.uk/. Please refer to http://en.wikipedia.org/wiki/Stereographicprojection.

• Video mosaics. Combine frames of a video sequency taken from a stationary point to generate a background panorama. You can look at the assignment by Derek Hoiem offered in his Computational Photography class.

Grading

The assignment will be graded out of 100 points:

• 0 (no submission), 20 (an attempt at a solution), 40 (a partially correct solution), 60 (a mostly correct solution), 80 (a correct solution), 100 (a particularly creative or insightful solution)

Note: Preparing good report is important as well as your solutions!
What to Hand In

You are required to submit all your report along with a short webpage in HTML. For that purpose, prepare a folder containing

- HTML/README.txt (text file containing details about your project)
- HTML/code/ (directory containing all your code)
- HTML/ (directory containing all your documents, including your images)
- HTML/data/ (including your data images)
- HTML/result/ (including your result images)
- HTML/index.html (html report)

Archive this folder as pset3.zip and email to my email address (aykut@cs.hacettepe.edu.tr).

Each student must individually do the coding and prepare detailed HTML report which contains a brief overview of the problems, details of your implementation and results with your observations. If your implementation failed to give a satisfactory results, provide a brief explanation of the reason(s).

Late Policy

You may use up to five extension days (in total) over the course of the semester for the three PSets. Any additional unapproved late submission will be weighted by 0.5.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specic problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else’s work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References
