MIT CSAIL

6.869 Advances in Computer Vision

Fall 2013

Problem Set 7: Image Mosaicing

Posted: Thursday, Oct 24, 2013 **Due:** Thursday, Oct 31, 2013

You should submit a hard copy of your work in class, and upload your code (and all files needed to run it, images, etc) to stellar.

Your report should include images and plots showing your results, as well as pieces of your code that you find relevant.

Problem 1 *Photomerge*

Panoramic images are used to portray wide scenes that cannot be captured entirely within any single shot. In this problem you will develop your own automatic algorithm for creating panoramas that you can then show-off to your friends.

To create a panoramic image from two overlapping photos we need to map one image plane to the other. Since in general we do not know how to relate the position and orientation of the two camera views, we will use image features techniques discussed in class to recover the underlying mapping. First, we will identify key points in both images, and match between those points to find correspondences. From the correspondences we can compute a transformation that maps one set of points to the other. Once we have the transformation, we can render the images in a common coordinate system, and merge them to generate the final result.

For key points and point matching, we will use the SIFT descriptor, *the* most commonly used image descriptor in recent years in computational imaging. We will also use David Lowe's method for finding matches between the two sets of descriptors in each of the images.

For good quality panoramas, the transformation between the images needs to be as accurate as possible. Yet, image descriptors and feature matching are both rather noisy processes: the descriptors are subject to image noise and compression artifacts, and not all presumed correspondences are true correspondences due to descriptor error and ambiguities in the matching (See Figure 1 bottom left). Incorrect matches will insert error to our estimation and can adversely affect the result.

To make our algorithm robust, we will use the RANSAC algorithm discussed in class, a method for estimating a parametric model from noisy observations. You can refer to Lecture 14 slides or Szeliski's book for details on the algorithm.

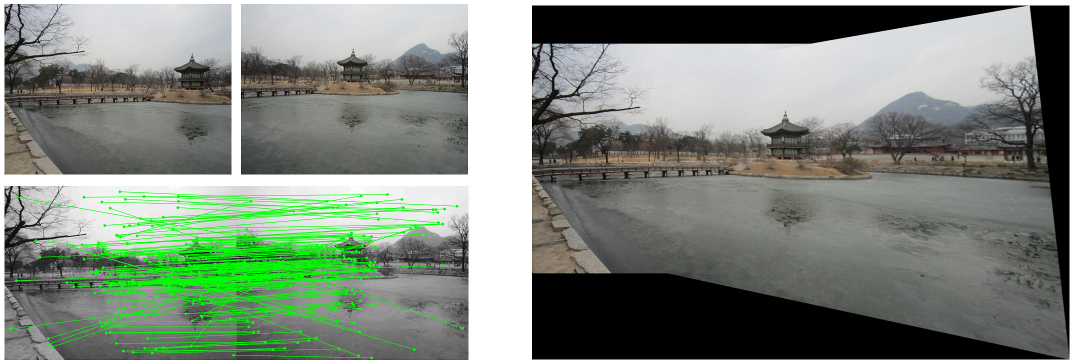

Figure 1: Panorama produced using our implementation. The image pair is shown in the top left, and below them are the detected point correspondences. On the right is the stitched panorama.

Download the code provided on the course website - this contains the images from Figure 1 and the *yosemite* sequence. You are encouraged to download two overlapping images of some famous place from Flickr (but there is no requirement). In addition, the code includes Andrea Vedaldi's implementation of SIFT (`http://www.vlfeat.org/~vedaldi/code/sift.html`).

(a) Compile this code in the *sift* folder by running `sift_compile`. Run it on the two images (see `sift_demo.m` for usage example) to find corresponding key points. Plot the detected features and the resulting correspondences between the two images.

(b) Implement the function `T = TransformRANSAC(x1,x2)` that takes as input two $M \times 2$ matrices `x1` and `x2` with the x and y coordinates of matching 2D points in the two images, and computes, using the RANSAC algorithm, the 3×3 homography `T` that maps `x1` to `x2`. In your report, write all the parameters of the algorithm, and the values that you found to produce the best results.

Hint: You can verify your homography computation using the function `cp2tform`. However, we do expect you to implement your own function to compute the homography as discussed in class.

(c) Write the function `im = MakePanorama(im1,im2,T)` that generates the panoramic image from the two images and the given transformation matrix `T`.

Hint: You can use the MATLAB functions `interp2` or `imtransform` to warp the images.

(d) Finally, write the function `im = Photomerge(im1,im2)` that produces a panorama from a given pair of overlapping images using your functions from the previous parts. Run the algorithm on the images provided or two that you have downloaded and add the resulting panorama to your report.

(e) **[Optional]** Extend the algorithm to handle n (> 2) images, and run it on either the *yosemite* sequence or one of your own photos, or photos you found on the web.