CIS581, Computer Vision
Project 3, Automatic 2D image mosaic
Due November 21, 11:59 PM

Overview

This project is to be done individually.

This project focuses on image feature detection, feature matching and image mosaic techniques. The goal of this project is to create an image mosaic or stitching, which is a collection of small images which are aligned properly to create one larger image. We will follow technique outlined in the following papers, which are available on the course website:

“Multi-image Matching using Multi-scale image patches”, Brown, M.; Szeliski, R.; Winder, S. CVPR 2005

”Shape Matching and Object Recognition Using Shape Contexts”,S. Belongie, J. Malik, and J. Puzicha. PAMI 2002: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/

Function Specifications

This project requires you to write a number of functions. The specifications for each have been included on the Project page of the course website under the ”Project-3 Function Specifications” link. Please make sure your code follows those exactly.

Submission

Zip your files into a folder names ”3_<penn_user_name>” and email it to CIS581F11@gmail.com. This should include the .m files for the code, the source images for your mosaic, and the final result.

Task 1: CAPTURE IMAGES

For this project, you need to capture multiple images of a scene, which you will use to create image mosaic. In general, you should limit your camera motion to purely translational, or purely rotational (around the camera center).

Bonus points will be given for interesting selection of images.

Task 2: AUTOMATIC CORRESPONDENCES

You need to implement the following steps.

1. Detecting corner features in an image. You could follow method outlined in lecture note on Interest Point Detector, or your own image corner detector(for example derived from edge detector). You can probably find free “harris” corner detectors on-line, and you are allowed to use them. Software links are provided on the course website.

2. Implement Adaptive Non-Maximal Suppression. The goal is to create uniformly distributed points. See section 3 of the reference paper, as well as lecture notes.

3. Extracting a Feature Descriptor for each feature point. You should use the subsampled image around each point feature. Don’t worry about rotation-invariance just extract axis-aligned 8x8 patches. Note that it’s extremely important to sample these patches from the larger 40x40 window to have a nice big blurred descriptor. Don’t forget to bias/gain-normalize the descriptors.

   As a bonus, you can implement the Geometric Blur features.

4. Matching these feature descriptors between two images. Remember to use the trick to compare the best and second-best match (take their ratio).

TASK 3: RANSAC and IMAGE MOSAIC

Not all the matches computed in Task 2 will be correct. One way to remove incorrect matches is by implement the additional step of RANSAC (see lecture notes).

1. Use a robust method (RANSAC) to compute a homography. Use 4-point RANSAC as described in class to compute a robust homography estimate.

   Recall the RANSAC steps are:

   (a) Select four feature pairs (at random), p_i, p_i^1

   (b) Compute homography H (exact). Code is available for this on the Project page.

   (c) Compute inliers where $SSD(p_i^1, Hp_i) < thresh$

   (d) repeat step a-c

   (e) Keep largest set of inliers

   (f) Re-compute least-squares H estimate on all of the inliers

2. Produce a mosaic by overlaying the pairwise aligned images to create the final mosaic image. As a bonus, you can implement smooth image blending of the final mosaic.