HOMEWORK ASSIGNMENT #3
DUE: Thursday, October 5, 2017, 5 P.M.
CSCI 677: Advanced Computer Vision, Prof. Nevatia
Fall Semester, 2017

This is a programming assignment. The goal is to use SIFT features to locate desired objects in images that may contain other objects, occlusion and general clutter. It is assumed that the SIFT points on objects will arise primarily from a planar surface and that they can be matched in the other view by a homography transformation. The needed functions are all available in OpenCV and you are free to use them as you wish. An outline of the procedure can be found at:

http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_feature_homography/py_feature_homography.html

Basic sequence is to computer SIFT features, match using a simple matcher and refine the matches as well as computes the homography transformation. You are asked to not use the FLANN matcher referenced in the tutorial (we did not cover FLANN in class) but to use a simple, “brute force” matcher (BFmatcher) instead. The function also gives you a choice of robust fitting methods, use RANSAC for this assignment.

As this assignment uses built-in functions, you are asked to display some intermediate results to understand and show the internal workings of the program.

1. SIFT features: show the detected features overlaid on the images (just the location and directions and not the 128 dimensional vectors). Also give the number of features that are found in each image.

2. Show, graphically, the top 20 scoring matches found by the matcher before RANSAC operation is applied. Provide statistics of how many matches are found (per image pair)

3. Show the top 10 (or more) matches that are found after homography has been computed; also provide the total numbers consistent with the computed homography.

4. Output the computed homography matrix.

5. The built in homography finder also applies a non-linear optimization step at the end; you can ignore the details of this step or disable it if you wish.

Image Data: Assignment folder contains a HW3_Data folder with five images. Images 1 and 2 contain objects to be detected; Images 3,4 and 5 are target images.

What to Submit?

You should submit the following.

1. A brief description of the programs you write (include the source listing). Each module should be **well commented**.

2. Show the results of intermediate steps as listed in the descriptions above.

3. An analysis of your test results: how well does the method work? Does it work equally well on the different examples? If not, why might the performance be better in one case then the other?