HOMEWORK ASSIGNMENT #3

DUE: 5PM, October 4, 2018

CSCI 677: Advanced Computer Vision, Prof. Nevatia

Fall Semester, 2018

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:
https://docs.opencv.org/3.4/dc/dc3/tutorial_py_matcher.html

Basic sequence is to compute SIFT features, match using a simple matcher and refine the matches as well as compute the homography transformation. Finally, we need to assess if enough matches are found to consider an object to have been detected.

You are asked to use "brute force" matcher (BFmatcher). We recommend the use of bf.knnMatch which returns k of the best matches by filtering out weak matches based on some ratio between the best and second best matches. A snippet of codes look like:

```
say k=5,
bf = cv2.BFMatcher()
matches = [m[0] for m in bf.knnMatch(query, train, k=5)]
```

You can refer https://docs.opencv.org/3.4/dc/dc3/tutorial_py_matcher.html for more details and try different ks to get the best matches.

Documentation of the matchers can be found in
https://docs.opencv.org/3.4/d3/da1/classcv_1_1BFMatcher.html

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 (*both the locations 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. (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. src_1 and src_2 contain objects to be detected; dst_1, dst_2 and dst_3 are target images. You need to detect src_1 and src_2 in dst_1 and dst_2, and dst_3 respectively. Namely, you should show the matching results before and after RANSAC for the following 6 image pairs:
(src_1, dst_1), (src_1, dst_2), (src_1, dst_3), (src_2, dst_1), (src_2, dst_2), (src_2, dst_3)

What to Submit?

You should submit a SINGLE PDF file which includes:

1. A brief description of the programs you write (paste your code in the PDF). Each module should be **well commented**.

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

3. An qualitative 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? Note that the main goal is to locate the objects in the given images. We could transform the entire object image and overlay on the target image, using the computed homography, but this is not asked for. It will suffice to make your judgment based on results of feature matching (after homography computation).