ECE 4984/5554: Computer Vision, Fall 2015

PS4

Christopher Mobley

Due: Monday, 16 NOV 15

1 Short answer problems [25 points]

1. When performing interest point detection with the Laplacian of Gaussian, how would the results differ if we were to (a) take any positions that are local maxima in scale-space, or (b) take any positions whose filter response exceeds a threshold? Specifically, what is the impact on repeatability or distinctiveness of the resulting interest points?

    a. In regards to repeatability, if you were to take any position that is a local maxima in scale-space you performance would exceed that of taking any position whose filter response exceeded that of a certain threshold due to the fact that if an interest point is a local maximum in one frame/scale it or a point close to it is likely to be one in the next frame/scale. Thus local maxima will result in more repeatable features. However, in regards to distinctiveness of these features taking any position whose filter response exceeds a certain threshold would exceed that by taking any position that is a local maxima in scale-space because since all interest point are above a certain specified threshold, they should be similar and therefore more distinctive and less likely to be some random noise or other point of local maxima but that is not of interest.

2. What is an "inlier" when using RANSAC to solve for the epipolar lines for stereo with uncalibrated views, and how do we compute those inliers?

    a. For stereo with uncalibrated views, not all epipolar lines will meet at the same point. So, RANSAC solves for these epipolar lines by calculating the epipolar line that minimizes the epipolar constraint of $x^TFx' = 0$, while retaining the most points. Thus, an inlier is a point that fails within some set minimized error of the calculated epipolar line.

3. Name and briefly explain two possible failure modes for dense stereo matching, where points are matched using local appearance and correlation search within a window

    a. Two possible failure modes for dense stereo matching, when point are matched using local appearance and correlation include occlusions, or a correspondence point in one image may be obscured by an object in another. Another possible failure mode includes images containing non-lambertian surfaces, which change in appearance depending on lighting conditions and thus makes correlating the two images difficult.

4. What exactly does the value recorded in a single dimension of a SIFT keypoint descriptor signify?

    a. A single dimension of a SIFT keypoint descriptor signifies one of eight gradient directions in a four by four subregion around the keypoint, which form a 128 dimensional vector.

5. If using SIFT with the Generalized Hough Transform to perform recognition of an object instance, what is the dimensionality of the Hough parameter space? Explain your answer.

    a. If using SIFT with the Generalized Hough Transform to perform recognition of an object instance, the Hough parameter space will be 4 dimensional. A 2-d position, orientation, and scale of a feature are all necessary to perform accurate object recognition.

2 Programming [75 points]

What to implement and discuss in the writeup
Write one script for each of the following (along with any helper functions you find useful), and in your pdf writeup report on the results, explain, and show images where appropriate.

1. **Raw descriptor matching [15 pts]**: Allow a user to select a region of interest (see provided *selectRegion.m(py)*) in one frame, and then match descriptors in that region to descriptors in the second image based on Euclidean distance in SIFT space. Display the selected region of interest in the first image (a polygon), and the matched features in the second image, something like the below example. Use the two images and associated

features in the provided file *twoFrameData.mat* (in the gzip file) to demonstrate. Note, no visual vocabulary should be used for this one. Name your script *rawDescriptorMatches.m(py)*

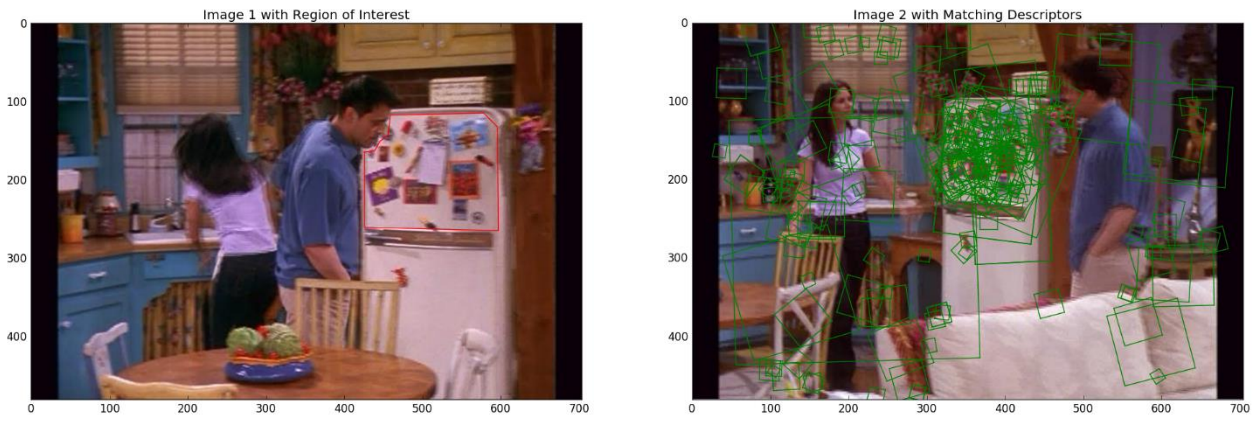
Image 1 with Region of Interest
Image 2 with Matching Descriptors
0
100
200
300
400
500
600
700

Figure 1. Raw Descriptor Matches

2. **Visualizing the vocabulary [20 pts]**: Build a visual vocabulary. Display example image patches associated with two of the visual words. Choose two words that are distinct to illustrate what the different words are capturing, and display enough patch examples so the word content is evident (e.g., say 25 patches per word displayed). See provided helper function *getPatchFromSIFTParameters.m(py)*. Explain what you see. Name your script *visualizeVocabulary.m(py)*

Figures 2 and 3 show the first 25 patches of label 3 and 1492 of 1500 labels assigned by kmeans on a sample set of 300 images with all descriptors available from each images. Figure 2, clearly shows a definable pattern in which a region of white in the center of the patch is surrounded by a region of darkness. While Figure 3, shows two definably different strip patterns. Which given a larger k value would have been subdivided into separate labels; however, since k was capped at 1500 they were clustered together due to similar pattern.

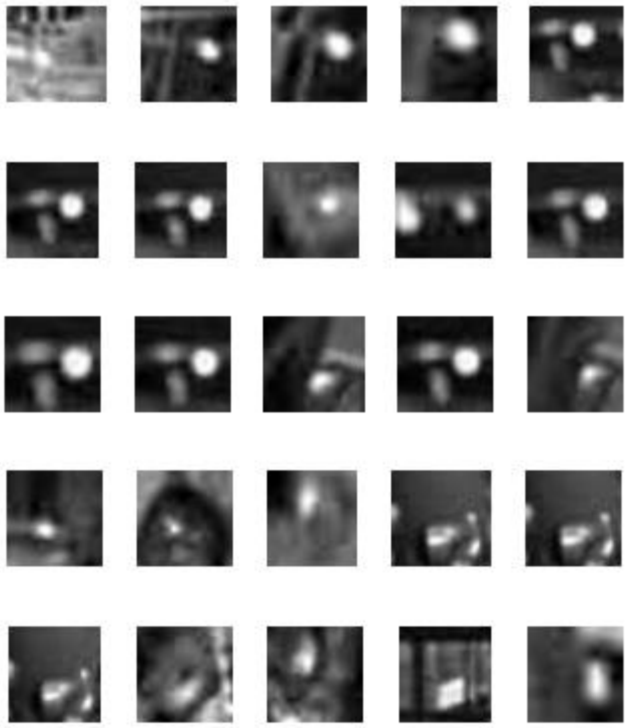
Figure 2. Patches labeled 1492 by kmeans with a value of 1500 on 300 images

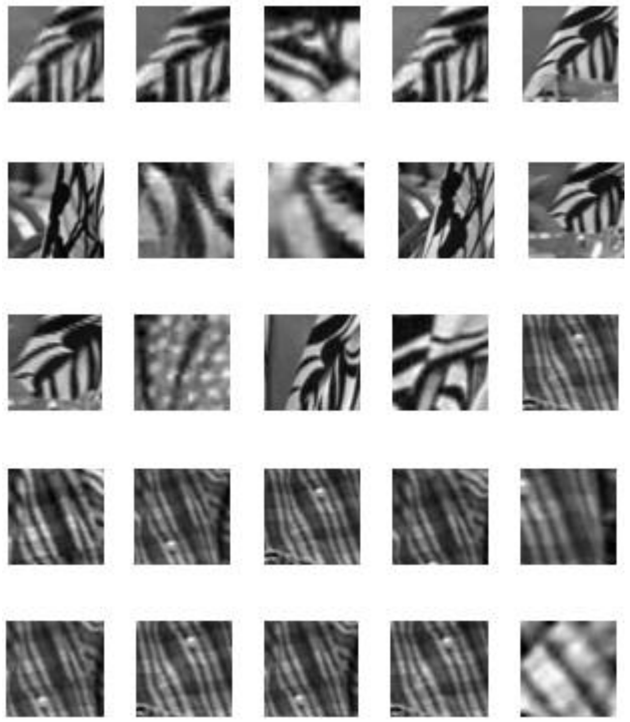
Figure 3. Patches labeled 3 by kmeans with a value of 1500 on 300 images

3. **Full frame queries [20 pts]**: After testing your code for bag-of-words visual search, choose 3 different frames from the entire video dataset to serve as queries. Display the M=5 most similar frames to each of these queries (in rank order) based on the normalized scalar product between their bag of words histograms. Explain the results. Name your script *fullFrameQueries.m(py)*

Three different full frame queries were performed. In each image, every descriptor was assigned a k label. A histogram was constructed for each full frame with respect to the number of each k labels that image was assigned. After which, the similarity between a query image's histogram and the rest of the images' histograms were computed and the top 5 matches, besides the images itself, were returned. Figures 4 through 6, show that a sample of 300 with only 1500 feature is quite accurate in describing a frame.

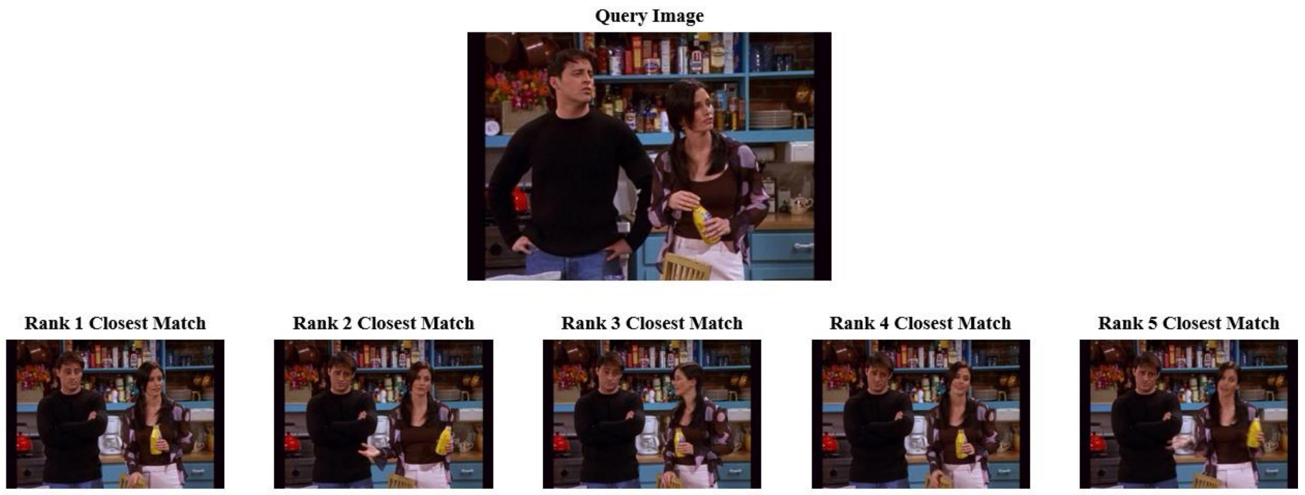

Figure 4. Closest matching frames for full frame query

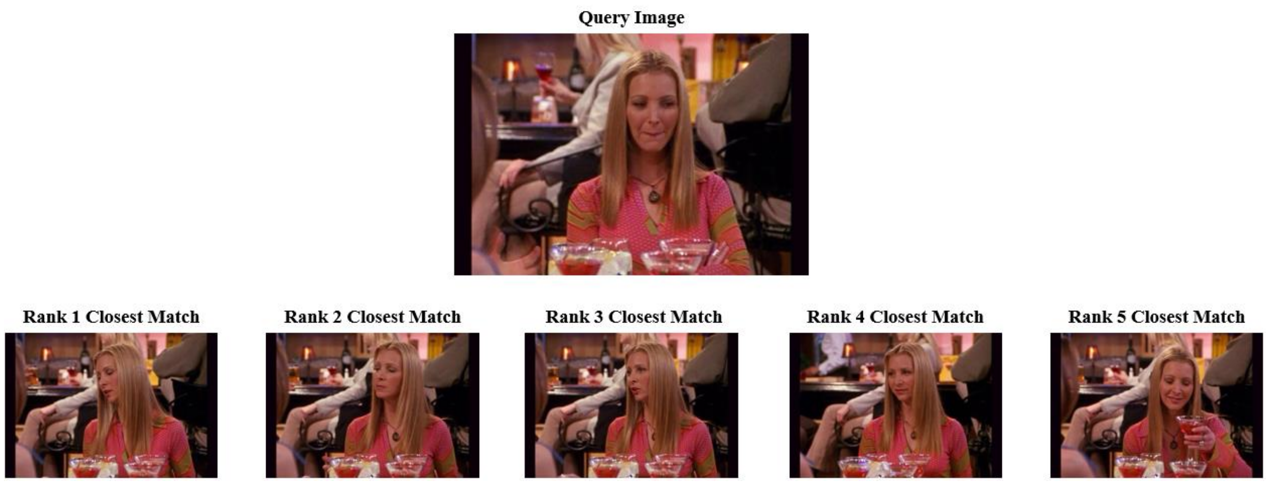

Figure 5. Closest matching frames for full frame query

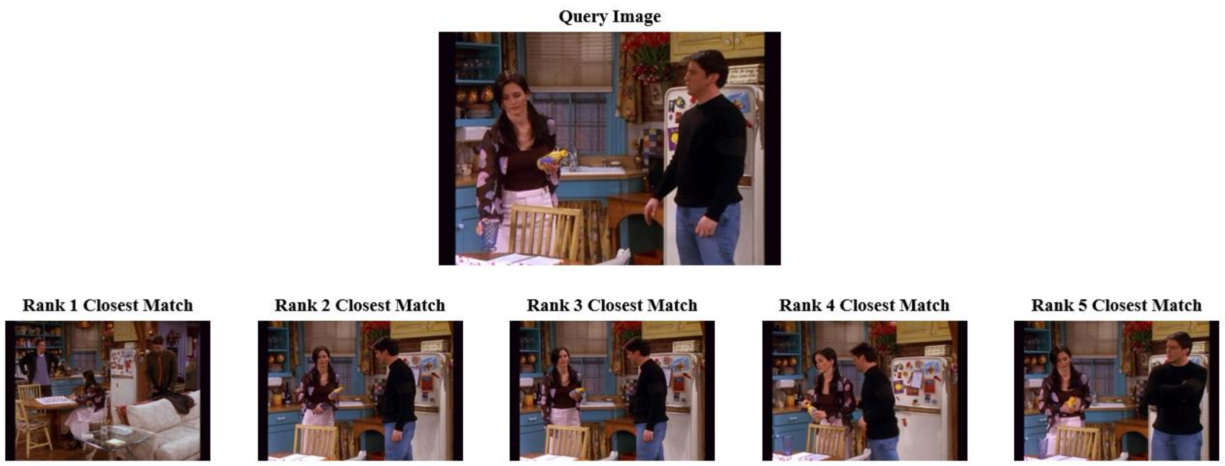

Figure 6. Closest matching frames for full frame query

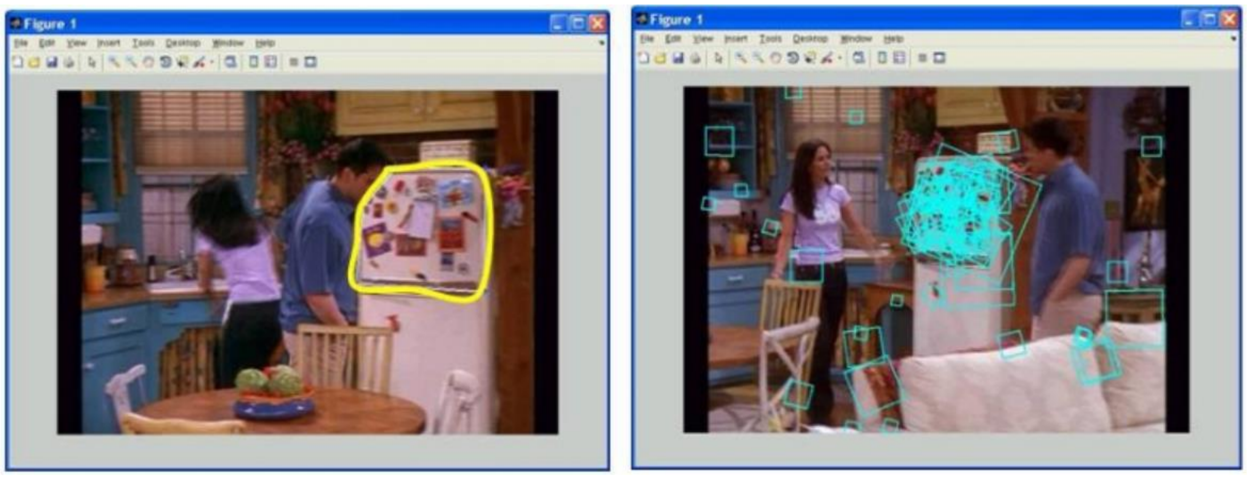
Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help

Figure 7. Raw descriptor matches

4. **Region queries [20 pts]:** Select your favorite query regions from within 4 frames (which may be different than those used above) to demonstrate the retrieved frames when only a portion of the SIFT descriptors are used to form a bag of words. Try to include example(s) where the same object is found in the most similar M frames but amidst different objects or backgrounds, and also include a failure case. Explain the results, including possible reasons for the failure cases. Name your script *regionQueries.m(py)*

Three different region based frame queries were performed. In each queries, a k label was assigned to each feature in a user selected region of an image. A histogram was constructed for each selected region with respect to the number of each k labels the descriptors in that region were most closely related to. After which, the similarity between the query region and the rest of the image set were computed and the top 5 matches, besides including the original images from which the region was extracted were returned. Figures 8 through 10, show that a sample of 300 with only 1500 feature is quite accurate in describing large regions or regions with a high level of descriptors; however, it fails in small regions or regions with very low number of descriptors. Figure 10 shows a complete failure in which a region with a tie was selected and no images with a tie was returned. In order to make the process more robust one could increase the number of k, use a different classifier, or implement a stop list for common word and apply a tf-idf weighting to the bag of words.

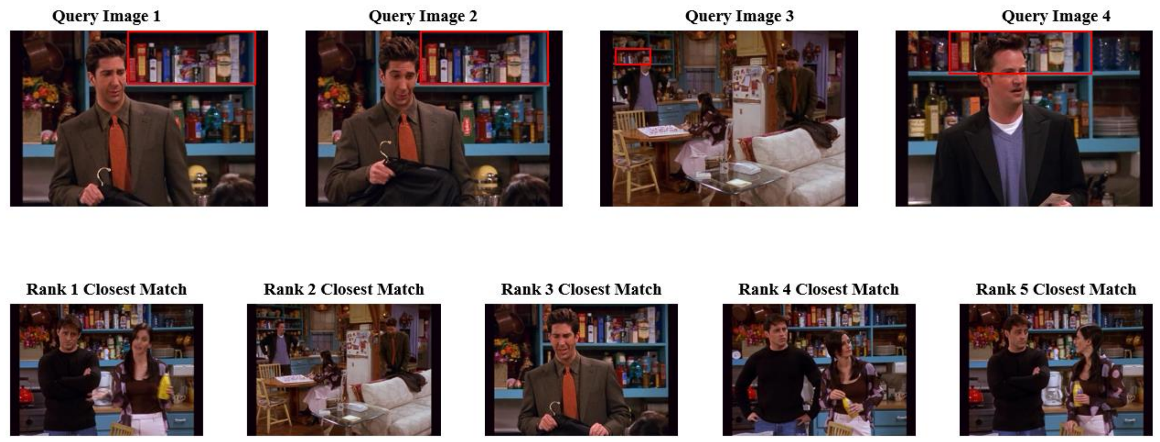

Figure 8. Closest match with shelf query region

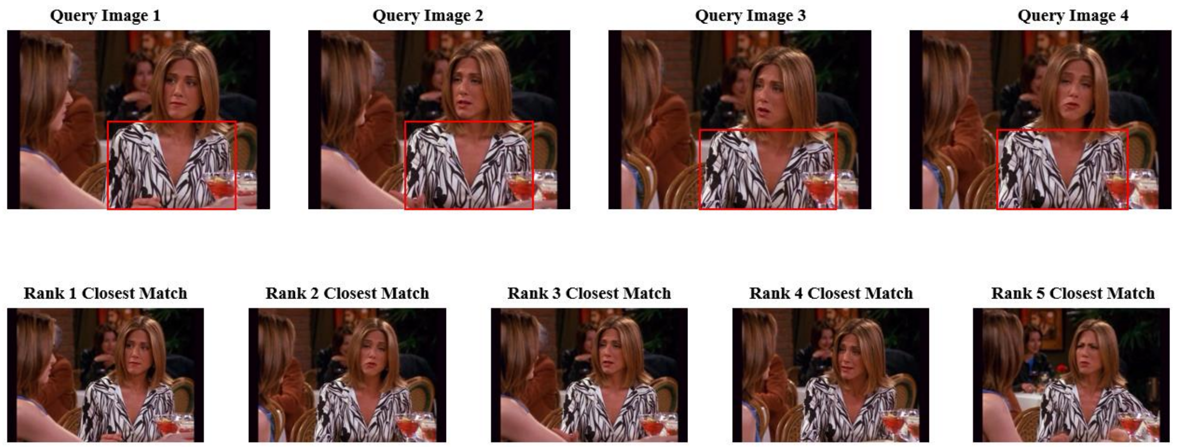

Figure 9. Closest match with blouse query region

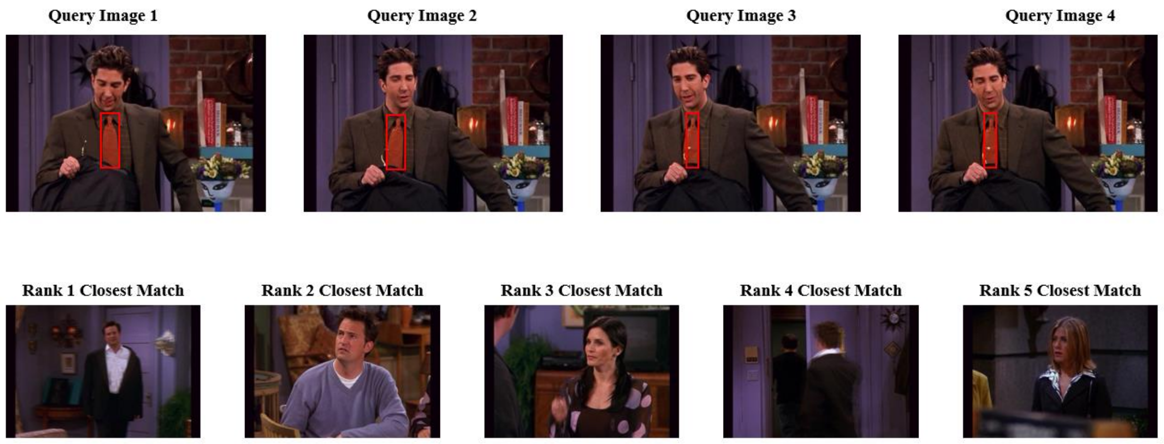

Figure 10. Closest match with tie query region

Note: I discussed this assignment with Rich Fedora, Murat Ambarkutuk, Orson Lin, Evan Smith, and Yi Tien