CS 6476
PS 4

1.

(1) If we perform interest point detection using Laplacian or Gaussian and (a) take any positions that are local maxima in scale-space, then we may end up with many interest points detected but less distinctive ones, because each interest point can be a good match to many other interest points. If we were to (b) take any position whose filter response exceeds a threshold, then we can make the interest points less repeatable, because the local maxima will be different and threshold will be different, and so the same image content may not be detected in different images. But similarly, we can't find the real interest point depending on the threshold. So the interest points would be less distinctive as well.

(2) Inliers refer to a model having a particular set of parameters but outliers does not fit into that model. An inlier is a line that can be calculated by minimizing the epipolar constraint. However, with uncalibrated view, epipolar lines may not meet at the same point. Hence, we can differentiate the outliers by looking at the distance from the line to the corresponding points.

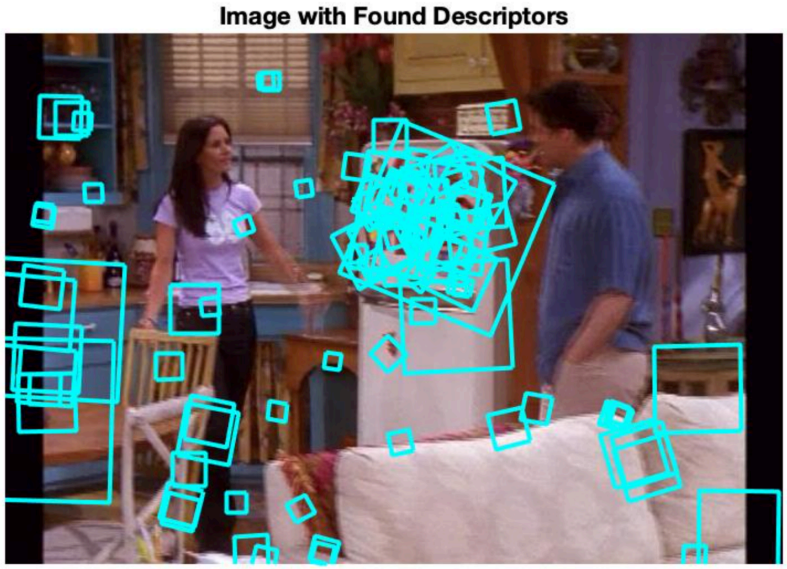
(3) First we can consider the material types of the objects in the image. If the objects are like mirrors, then it may be difficult to correlate the windows. Second, we can think of textures. It might be difficult to match and correlate the windows if the lines are at areas without texture, e.g, blue sky.

(4) Each value recorded in a single dimension of a SIFT keypoint descriptor specifies a magnitude of gradient directions, which is one of 8 gradient directions of a 4x4 neighboring blocks. (8x4x4 = 128 dimensional descriptors)

(5) It would be 4 dimensional. They are: x-position, y-position, scale and orientation. For each descriptor space, the elements will vote for keypoints with these parameters.

2.

(1) I set a relatively small threshold for this problem, filtering out only the closest features to the selected region in image. And as we can see the patches are mostly concentrated at the selected region. If we were to choose larger thresholds, there will be more patches.



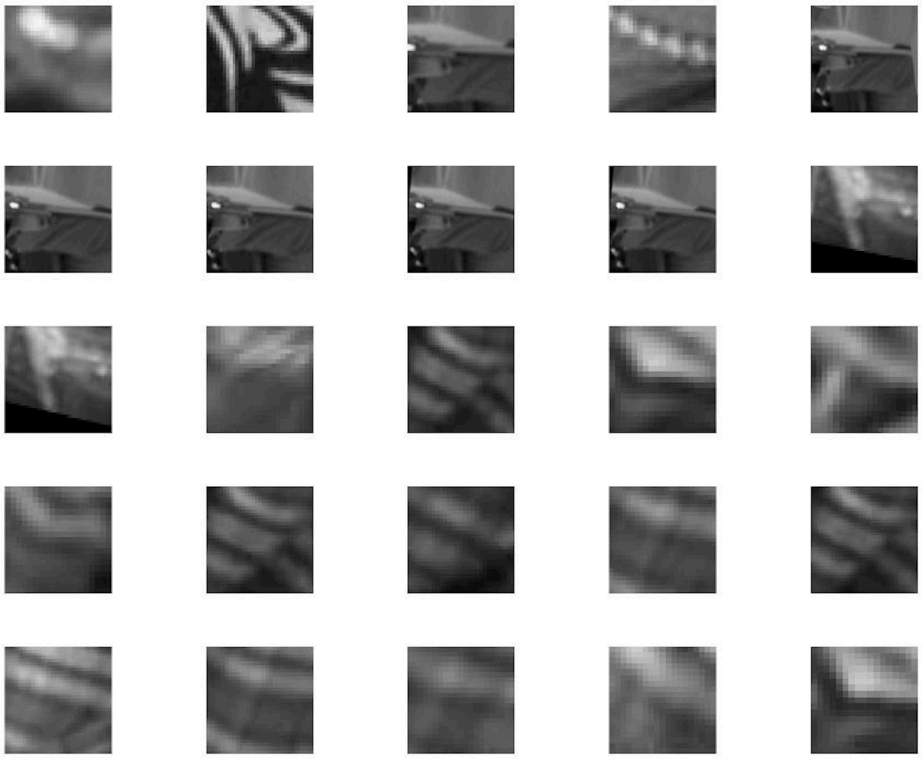
(2)

I randomly selected 30 descriptors per frame. Then, I used the provided KmeansML function to cluster the descriptors matrix into k=1500 clusters. So there are 1500 centers/visual words in our vocabulary. For visualizing word patches I randomly selected 3 indices [404, 666, 1268] in the vocabulary. I found sift patches belonging to that index and displayed 25 first patches.

As seen below, some of the patches are clustered well – we can clearly tell they are patterns on clothing. However, some are pretty blurry, or have other patterns that do not belong to the rest of the patches, but nonetheless, they do have the minimal distance to the centers. For extremely similar patterns, like word 1268, the algorithm worked really well.

Word 400

Word 668

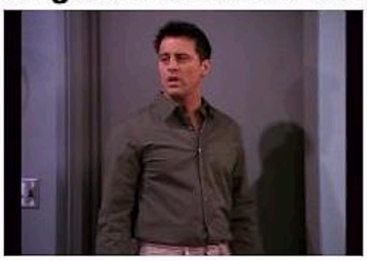


Word 1268

(3) Explanation:
I loaded kmeans results from last question (to save the time to calculate it again) and then computed (normalized) bag of words histograms for each frame, which took some time. After that, I compared the similarity between reference frame and all other frames by using the scalar product.

This method works well in most cases; it is able to extract the most similar frames to the original one. Sometimes with 2 frames that have similar features but not in the same scene. By the id of the frames we can see that the similar frames are the ones immediately before or after the original frame, which makes sense because scenes should be continuous in a video.

Original Frame No. 668



Similar Frame: No. 669

Similar Frame: No. 667

Similar Frame: No. 674

Similar Frame: No. 670

Similar Frame: No. 673

Original Frame No. 1000

Similar Frame: No. 1001

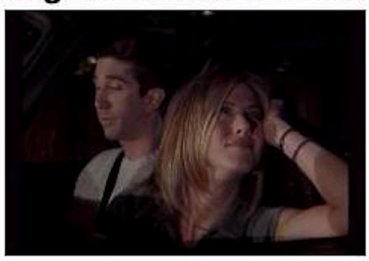
Similar Frame: No. 998

Similar Frame: No. 993

Similar Frame: No. 1007

Similar Frame: No. 995

Original Frame No. 4377



Similar Frame: No. 4376

Similar Frame: No. 4374

Similar Frame: No. 4375

Similar Frame: No. 4359

Similar Frame: No. 4360

(4) In the first 3 cases the selected regions are Rachel's shirt, Ross' suit and the white board. In failure case the selected region is the black and white mug in Phoebe's hand. I used descriptors in the regions that I selected to make the bag of words, instead of using all descriptors. However, it is still comparing a region to other image frames, which is why the result might be off. As we can see, the algorithm can pick up the features of an object and find similar images. In case 3 it's able to locate the board even different person is standing in front of it. However, in the failure case, we selected a very small region, which might be a limitation, and the algorithm picked up Rachel's black and white dress, and the lights that have similar features of the mug.

Case 1

Original904

Similar1066

Similar903

Similar988

Similar899

Similar986

Case 2

Original1945

Similar1961

Similar4195

Similar1964

Similar1916

Similar1942

Case 3

Original1784

Similar1801

Similar1799

Similar1800

Similar2249

Similar865

Failure case

Original4123

Similar530

Similar687

Similar686

Similar4183

Similar4182

3 (1) The difference between this problem and the previous one is that we calculated bag of words with tf-idf. The results with image frame 904 is improved, we see that not only a similar frame 903 is selected, an partially covered shirt (899), shirt from different viewpoints(900, 988) and a much smaller shirt (892) are also selected correctly. This method is better than just using selected regions.

Original 904

Similar 903

Similar 899

Similar 900

Similar 892

Similar 988