

Short Answer Problems:

- 1) When performing interest point detection with LoG, we want positions whose filter responses reached the peak of Laplacian response. That is, we only want the characteristic scale. For example, for image A, suppose we set a threshold t_1 . If some scales' Laplacian responses reach t_1 , all positions correspond to those scales will be detected. Let image B be a similar image to A but with a different geometric view or scale. Now if we do the detection again, we have no idea how to pick threshold t_2 so that the same set of positions will be detected. In other words, setting a threshold to the Laplacian response of a certain scale cannot guarantee the repeatability of the detection. On the other hand, if we take the positions whose filter responses are local maxima, then the detector will stick to the same set of detected points (they always correspond to the maximum LoG response).
- 2) A single dimension of a SIFT descriptor is the count of a specific orientation of pixels in a patch. For example, in a certain patch, suppose there are n sub-patches and the SIFT descriptor has k dimensions. Then the whole SIFT descriptor will be a histogram with k bins (each bin represents an orientation; $2\pi/k$ interval) that counts all the n pixels.
- 3) The dimensionality is 4. We can use Generalized Hough Transform to do spatial verification on the matching SIFT descriptors. We could let each matched feature cast a vote for the model feature's parameters. Each SIFT descriptor has 3 parameters: position, scale and orientation (4 dimensions if represented in a vector form). After every matching descriptor has voted on the parameters, we verify the parameters with high votes.

NOTE: I deleted the "frames" and "sift" subdirectories in the working directory. Please include "frames" and "sift" in the same working directory when running the *.m files.

Programming: Video Search

NOTE: I deleted the frames and sift subdirectories in the working directory. Please include "frames" and "sift" in the same working directory when running the *.m files.

1) rawDescriptorMatches.m



p1_1_frame1.jpg

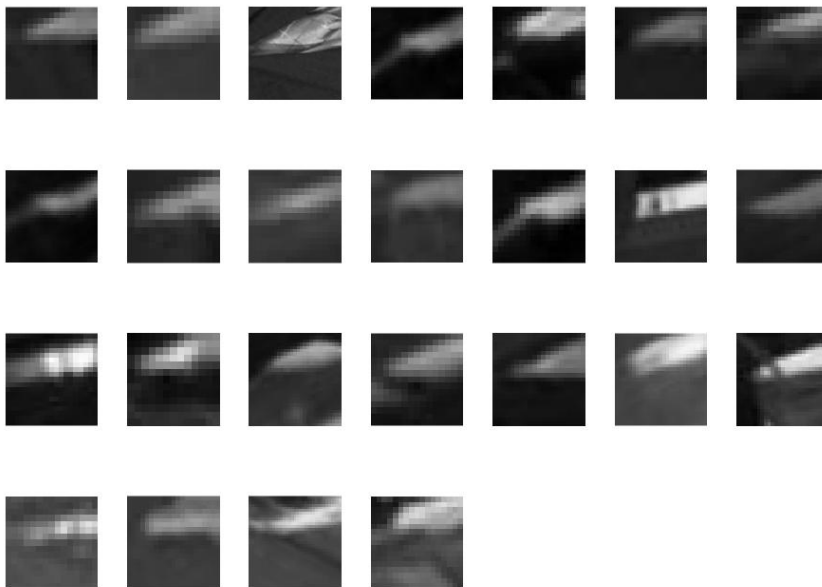


p1_1_frame2.jpg

2) visualizeVocabulary.m kMeans.mat (<5min)

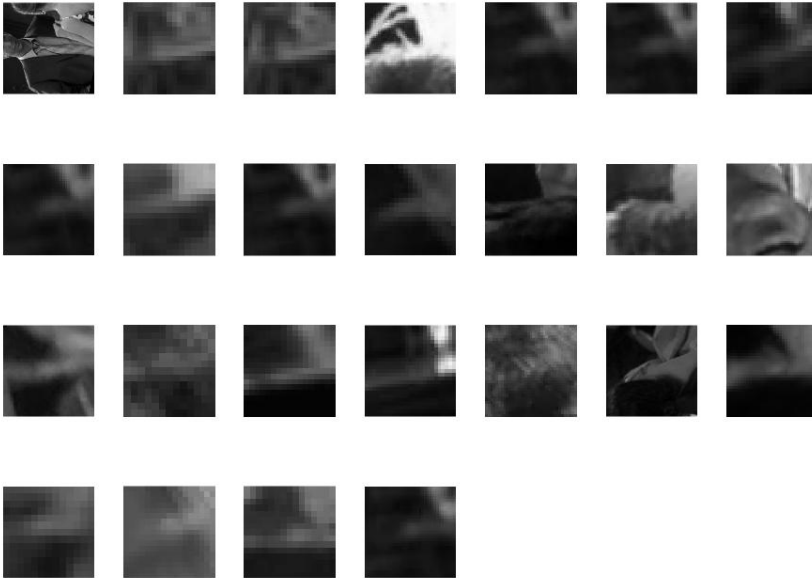
I picked a random sample of size 300 and $k = 1500$ for the kMeans clustering (kMeans.mat saved in the same directory). Below are the top 25 matching patches that I picked for the 2 example words. As shown, the patches are visually very similar in shape. Some have higher resolution (more pixels) / higher illuminations than others.

frameNum:300 word1: 500



p2_word1.jpg

frameNum:300 word2: 1200



p2_word2.jpg

3) fullFrameQueries.m (<5min)

The matching results of the 3 query frame appear to be successful. The human figures in the matching frames all stay in similar positions as the query frame. In particular, I included the query frame in the matching frame list to show that the query frame itself is the “most similar frame” to itself (by normalized dot product comparison).

Query frame1:

Query frame1: 2247



Top 6 matching frames (1st match is the query frame itself):

matching frame: 2247, rank: 1; matching frame: 96, rank: 2; matching frame: 97, rank: 3;



matching frame: 2252, rank: 4; matching frame: 2248, rank: 5; matching frame: 102, rank: 6;



Query frame2:

Query frame2: 1288



Top 6 matching frames (1st match is the query frame itself):

matching frame: 1288, rank: 1;matching frame: 1291, rank: 2;matching frame: 1289, rank: 3;

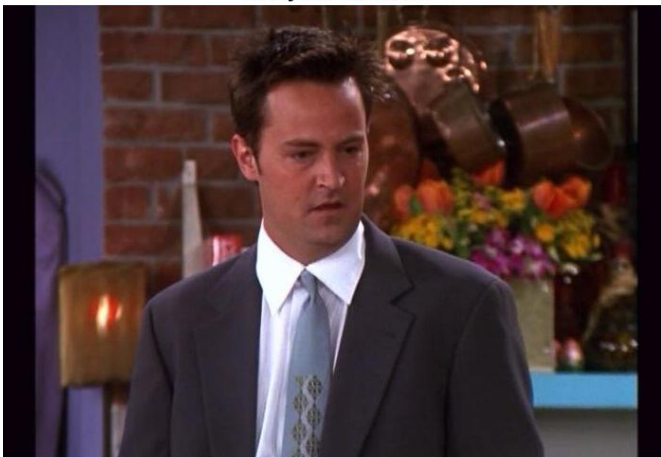


matching frame: 1290, rank: 4;matching frame: 1287, rank: 5;matching frame: 1292, rank: 6;



Query frame3:

Query frame3: 5407



Top 6 matching frames (1st match is the query frame itself):

matching frame: 5407, rank: 1; matching frame: 5406, rank: 2; matching frame: 5404, rank: 3;

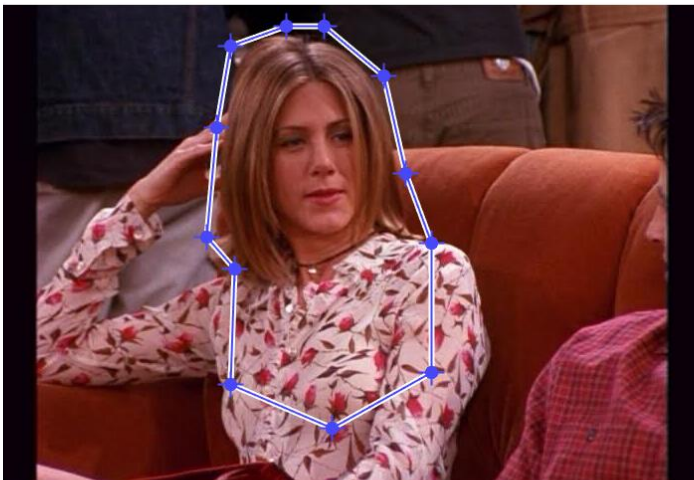


matching frame: 5408, rank: 4; matching frame: 5405, rank: 5; matching frame: 5403, rank: 6;



- 4) regionQueries.m (<5min, need to uncomment the line 3-10 by turns to run)
The first 2 set of matching frames all had the same object in the same background. In the 3rd set of matching frames, the same object is in different backgrounds. The 4th set of matching frames have failures. This could be caused by, (1) k's value is too low k=100 due to too few descriptors in the query region; this results inaccurate clustering of words; the formed bag of words is bad at distinguishing different contents (2) There isn't enough candidate frames in the whole series to search for. The first season is most possible.

p4_region1.jpg: (k=550)



p4_region1_matches.jpg:

matching frame: 1019, rank: 1;



matching frame: 1020, rank: 2;



matching frame: 1018, rank: 3;



matching frame: 1017, rank: 4;



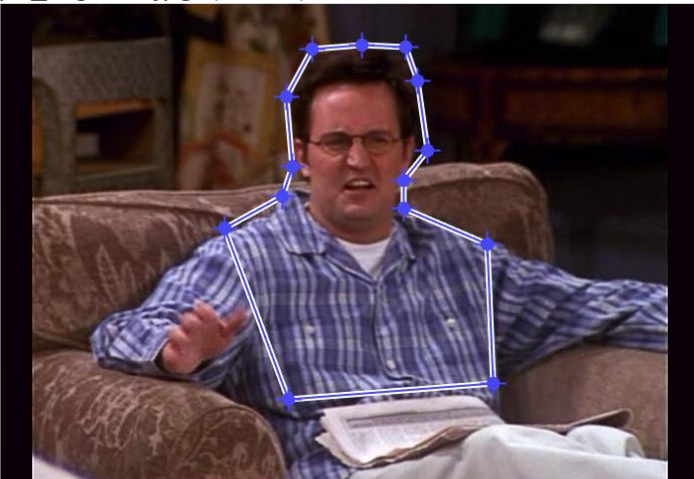
matching frame: 1016, rank: 5;



matching frame: 1015, rank: 6;



p4_region2.jpg: (k=500)



p4_region2_matches.jpg:

matching frame: 2234, rank: 1; matching frame: 83, rank: 2; matching frame: 2236, rank: 3;



matching frame: 86, rank: 4; matching frame: 85, rank: 5; matching frame: 2235, rank: 6;



p4_region3.jpg: (k=200)



p4_region3_matches.jpg:

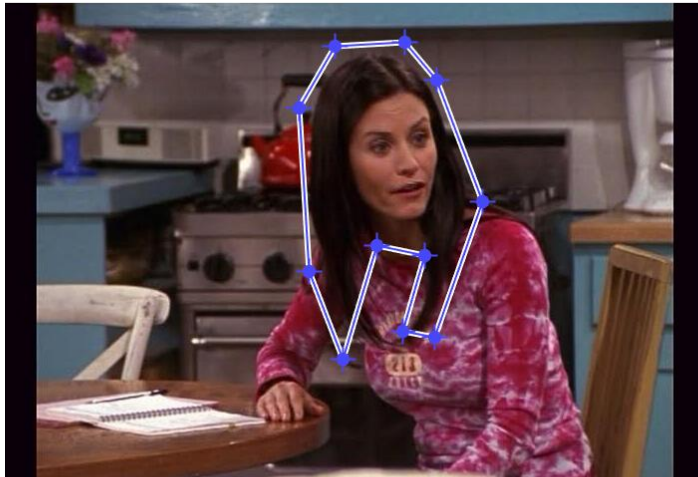
matching frame: 1613, rank: 1;matching frame: 1612, rank: 2;matching frame: 1625, rank: 3;



matching frame: 1675, rank: 4;matching frame: 1674, rank: 5;matching frame: 1673, rank: 6;



p4_region4_failure.jpg: (k=100)



p4_region4_failure_matches.jpg:

matching frame: 2222, rank: 1; matching frame: 2223, rank: 2; matching frame: 2225, rank: 3;



matching frame: 6467, rank: 4; matching frame: 2385, rank: 5; matching frame: 5915, rank: 6;



Optional

5) extra1.m (~5min)

The original $k=1500$. The stop list set the new $k_{\text{new}} = 1000$, with 500 common words deleted from the bag of words list. Then I added the tf-idf weighting to the word histograms. From the detection results, there's no obvious improvement in with the 2 processes. But I assumed, with less emphasis on the frequent words, the detection will be more sensitive to moving object than background objects. (background words appear more frequently in the database)

Below is the comparison of the results: (this example cannot really show a difference)

extra_frame.jpg:

Query frame1: 4125



extra_no_process.jpg

matching frame: 4125, rank: 1;matching frame: 4126, rank: 2;matching frame: 4127, rank: 3;



matching frame: 4124, rank: 4;matching frame: 4128, rank: 5;matching frame: 4041, rank: 6;



extra_stop_tf.jpg:

matching frame: 4125, rank: 1;matching frame: 4126, rank: 2;matching frame: 4124, rank: 3;



matching frame: 4127, rank: 4;matching frame: 4128, rank: 5;matching frame: 3466, rank: 6;

