A4 Writeup

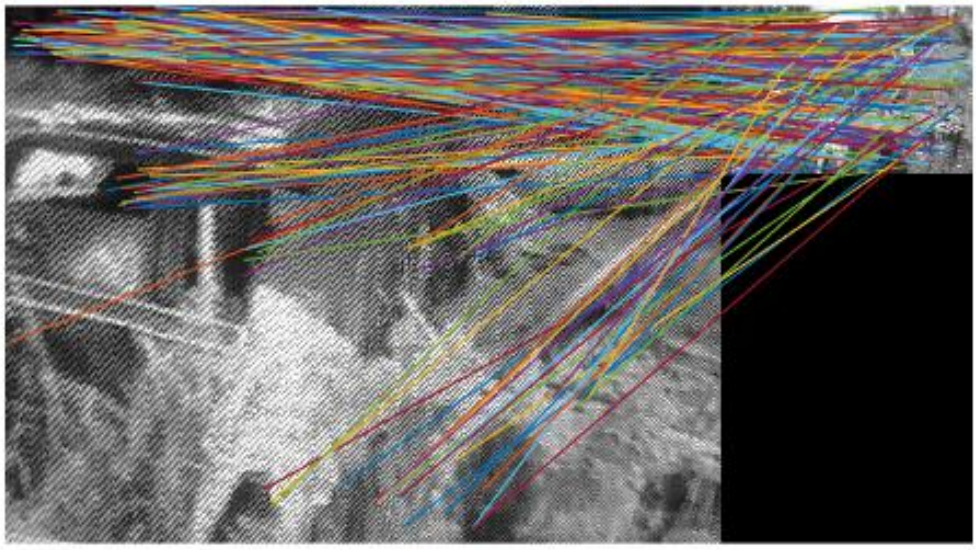
What to implement and discuss in the writeup

1. Raw descriptor matching

Run **rawDescriptorMatches.m**, which will provide the following output:

Input images paths: '/v/filer4b/v45q002/data/video_frames/5800/088.jpg'
'/v/filer4b/v45q002/data/video_frames/Reference/088.jpg'

Output image 1out.jpg:

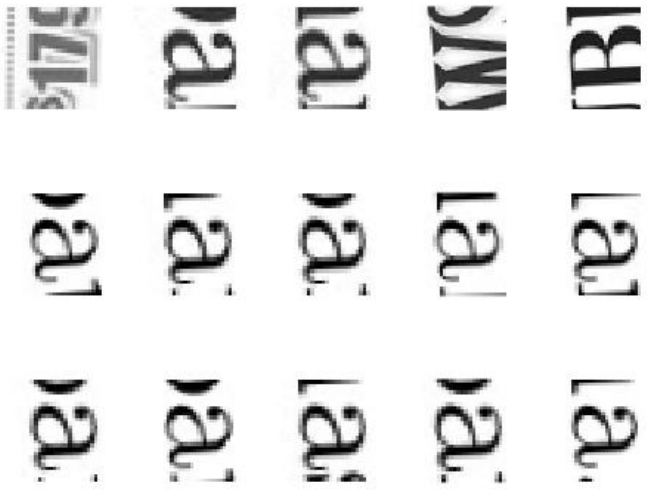


2. Visualizing the vocabulary

Output of **visualizeVocabulary.m**

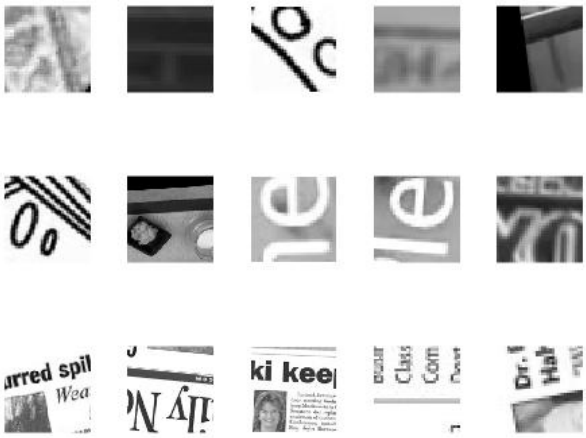
The following 2 vocabulary visualizing outputs are of different membership.
I grabbed all Reference images from Data file and sampled half the descriptors of all descriptors from all the Reference images. Then run K-mean with K value = 1000. I don't recommend test this script with K = 1000 since it will take a long time to finish. Feel free to test it with smaller K values like 500.

(1)VisualizingVocabulary1.jpg:



```
output of membership == 60 from K-mean result
k value = 1000
```

(2).VisualizingVocabulary2.jpg:



```
output of membership == 188 from K-mean result
k value = 1000
```

From the 2 outputs, I think some of the vocabularies in the same group are of pretty high similarity like the vocabularies in VisualizingVocabulary1.jpg. While there also exist some vocabulary groups like the one in VisualizingVocabulary2.jpg which the words have relatively low similarity to others. Since the randomness of K-mean, sampling results and K value will make a perfect words splitting almost impossible.

3. Full frame queries

Output of the returned images after running **output3.m. bagOfWordsQueries** function will be called in this script, the topK Reference images will be returned

(1) Query result of running with image path
`'/v/filer4b/v45q002/data/video_frames/iPhone/082.jpg'`

Output stored at **boW1.jpg**:

This is a successful output; the third matched reference image is the corresponding image wanted. Correct top3 match.

(2) Query result of running **bagOfWordsQueries** with image path
`'/v/filer4b/v45q002/data/book_covers/Droid/058.jpg'`

Output stored at **boW2.jpg**:

This is a bad output; there is no correct match in top5 match, and all the returned images has very small similarity to query image.

4. Quantitatively evaluate your results for retrieval

Output of **quantitativeEvaluation.m**
This script will iterate through all query directories and select 1/3 images of each directory as sample. Since it is samples, the accuracy can vary in some certain ranges.

fprintf result after running **quantitativeEvaluation.m**, this script takes several minutes.

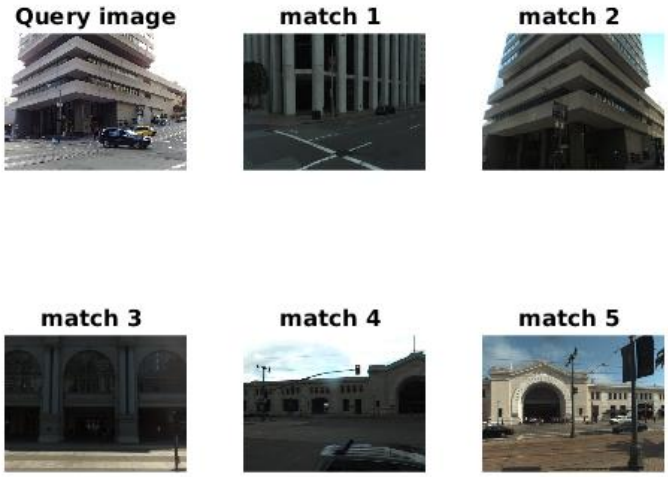
	video_frames	print	book_covers	landmarks
top1	5.30%	5.30%	0.00%	9.58%
top3	7.58%	8.33%	1.47%	20.96%
top5	10.61%	9.09%	2.21%	26.35%

5. Spatial verification

Output of the returned images after running **output5.m. bagOfWordsQueries** function will be called in this script, the topK Reference images will be returned

Query result of running with image path
`'/v/filer4b/v45q002/data/landmarks/Query/016.jpg'`

Output stored at **Spcial2.jpg**:
The output from boW top5, from **output3.m:**

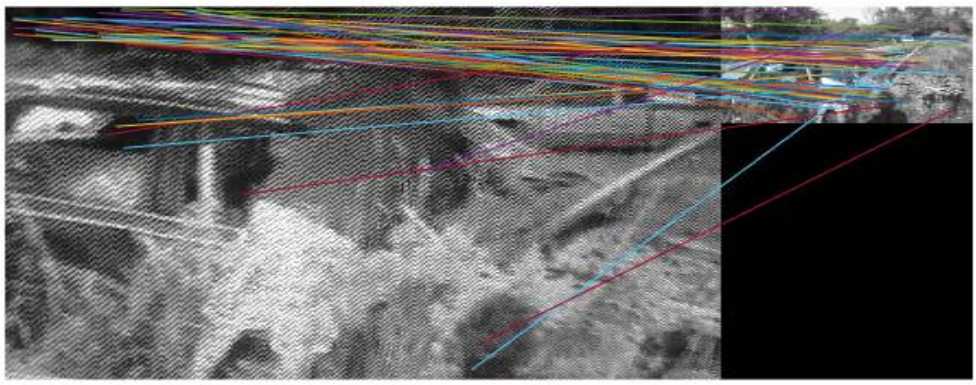


Output stored at **Spcial1.jpg**:
The output from Spatial verification, top5, from **output5.m:**

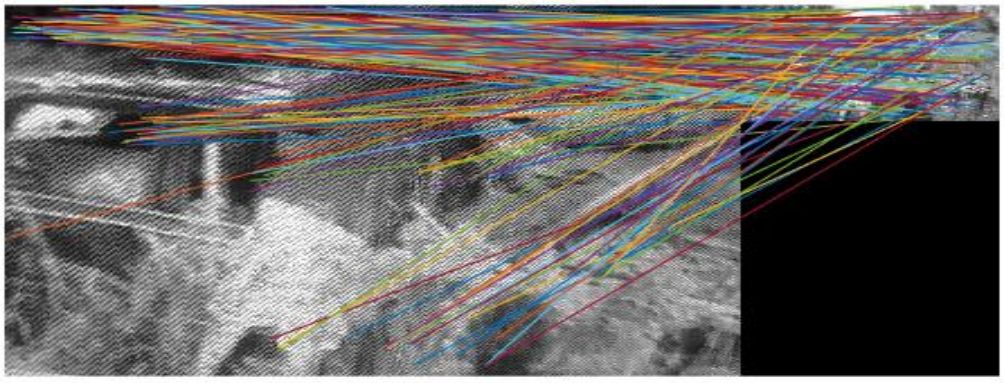
OPTIONAL : Extra credit

2. Add a "ratio test" functionality to part 1 above that discards ambiguous neighbor matches (see lecture 13). Display the results for a matched pair of images where the ratio test appears valuable

run extra1.m, get following result:



Question 1 output:



Compared with the output in question 1, we can tell some messy descriptor matches were gone. But since the descriptors were randomly chosen, there are still some descriptors have better matches not chosen during sampling.

3. Quantitatively evaluate the results of adding spatial verification to the bag of words shortlist. Compare the performance before and after, again broken down per image type. fprintf result after running **spatialVerificationQuantitativelyEvaluate.m**, this script takes many hours to finish.

Output of **spatialVerificationQuantitativelyEvaluate.m**
This script will iterate through all query directories and select 1/3 images of each directory as query image. And check whether the real corresponding image is in topK, K = 1,3,5.

	video_frames	print	book_covers	landmarks
top1	3.79%	0.76%	0.74%	13.77%
top3	9.85%	4.55%	0.74%	26.95%
top5	10.61%	6.06%	1.47%	31.74%

Compared with the result in question 4:

	video_frames	print	book_covers	landmarks
top1	5.30%	5.30%	0.00%	9.58%
top3	7.58%	8.33%	1.47%	20.96%
top5	10.61%	9.09%	2.21%	26.35%

I think there is no huge difference between them. Since the RANSAC chooses points randomly, the iterating time of 100 in my setting is also not confident to conclude that we find the best count. Also, the thresh distance value we set can affect the result much. Making the accuracy outputs no much higher than the one before special verification.