Homework Assignment 4, CS696, Applied Computer Vision

Hsuan Yu, Liu 823327369

**My algorithm and decision,**

Three functions:

1. get\_interest\_point():

In the beginning, filter an image with a gaussian filter. After that, I can use imgradientxy() to get x, y directional derivatives. Using these derivatives can calculate the variables for Harris corner detector. Matlab supplies element wise operations, so using derivatives to get scores for all pixels.

Harris score equation:

Ix2 = Gx.^2; Iy2 = Gy.^2; Ixy = Gx.\*Gy;

**harris** = Ix2 .\* Iy2 - Ixy.^2 – (0.04).\*((Ix2 + Iy2).^2);

Then, matlab function, colfilt() can help implement ‘Non Maximum Suppression’.

NMS Equation:

R\_NMS = colfilt(R\_corner,[3 3],'sliding',@max);

In my algorithm, firstly, I use the 0.01\*(Max value in harris) to be the threshold .

Finally, only return the pixels which are greater than threshold and in R\_NMS.

2. get\_features():

Matlab function, imgradient(), will return Gradient magnitudes(Gmag) and Gradient directions(Gdir). Use several for loop to create Histogram of Oriented Gradients. At the end, use normalize(mat,’norm’,1) to normalize the 128-dimension vectors to unit 1.

3. match\_features():

Matlab function, pdist2(), can calculate Euclidean distance between each feature to all the other features.

Distance Equation:

Dists = pdist2(features1, features2, 'euclidean');

Then, use ‘sort()’ to find nearest neighbor indexes.

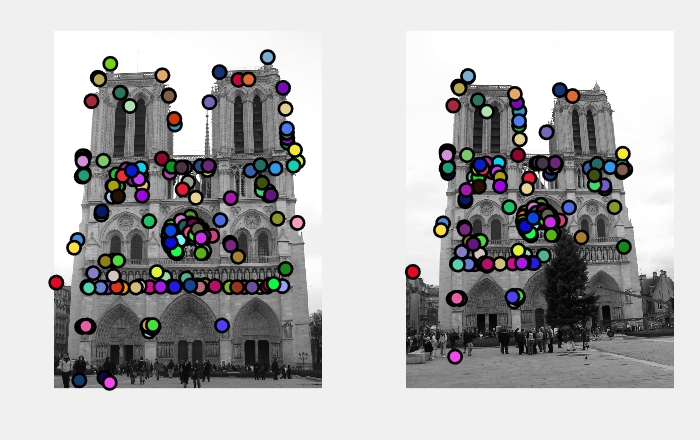
Nearest Neighbor equation:

[nn\_val,nn\_rank] = sort (Dists, 2);

Moreover, I can calculate NNDR by first and second values in nn\_val. Also, use 1/NNDR to be confidence. After that, I only focus on the ratios which are greater than 0.8, because a lower ratio mean that a feature’s similarity is lower. Ignore those features.

**Notre Dame result:**

Interest Point Threshold: 0.01\*(Score Max) K=0.06 NNDR: <0.8

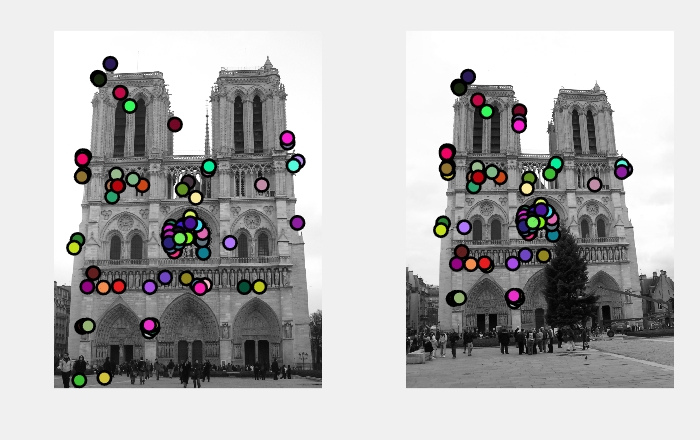


Evaluate: match features= 253

195 total good matches, 58 total bad matches 🡺 Accuracy: 77.07%

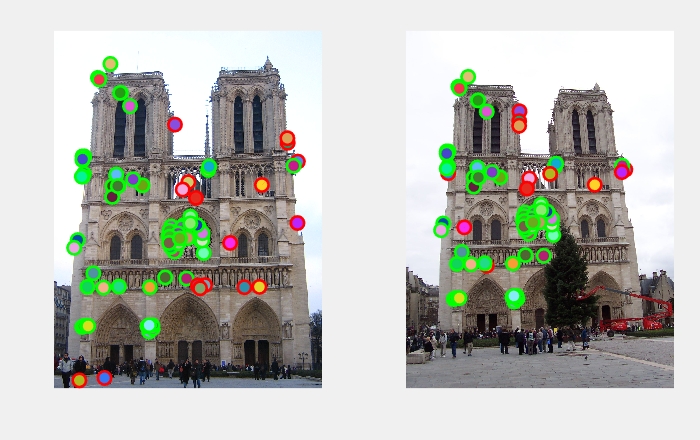


Interest Point Threshold: 0.01\*(Score Max) K=0.06 NNDR: <0.7

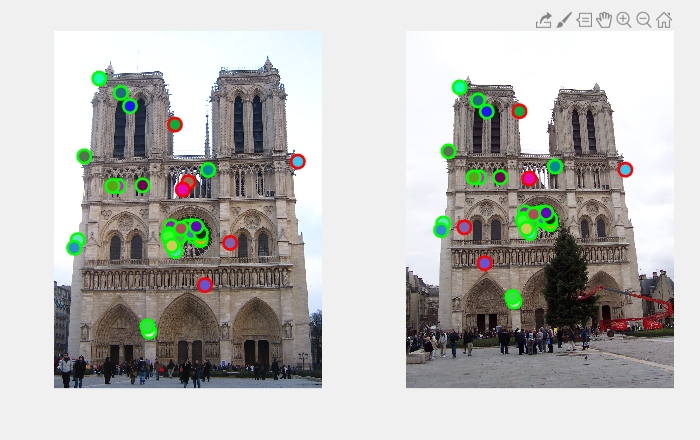


Evaluate: match features= 117

100 total good matches, 17 total bad matches 🡺 Accuracy: 85.4%

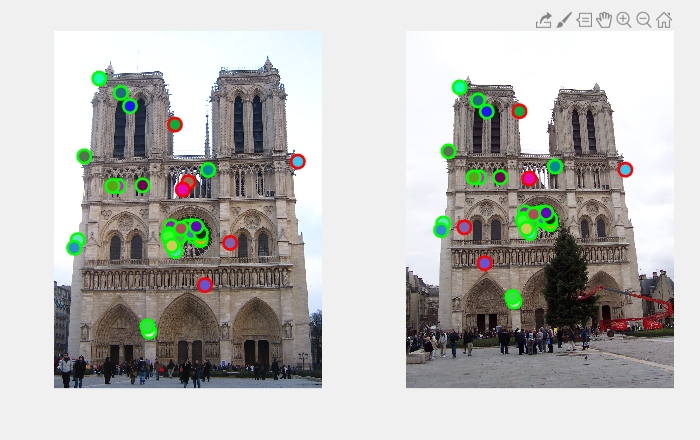


Interest Point Threshold: 0.01\*(Score Max) K=0.06 NNDR: <0.6

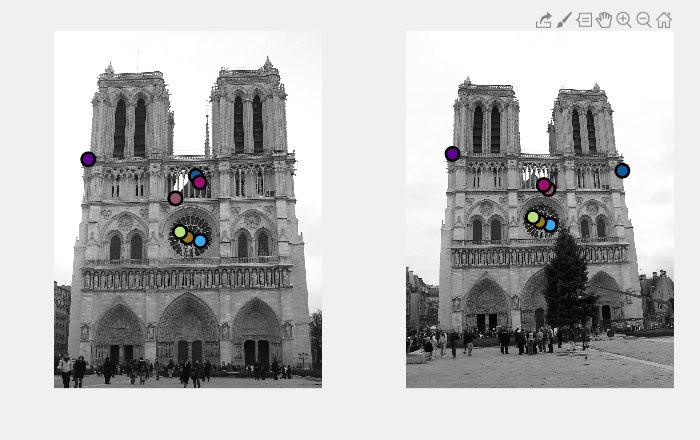


Evaluate: match features= 39

33 total good matches, 6 total bad matches🡺 Accuracy: 84.6%

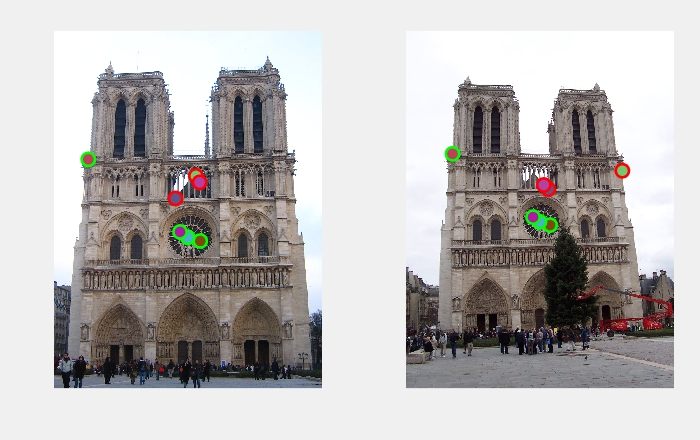


Interest Point Threshold: 0.1\*(Score Max) K=0.06 NNDR: <0.7



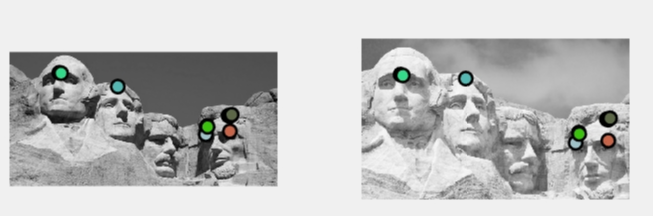
Evaluate: match features= 7

4 total good matches, 3 total bad matches 🡺 Accuracy: 57.14%

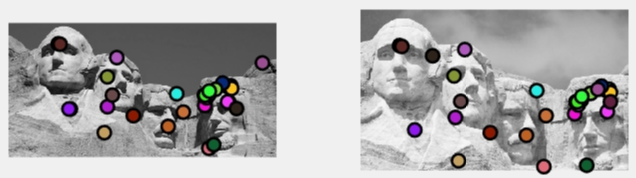


**My images result:**

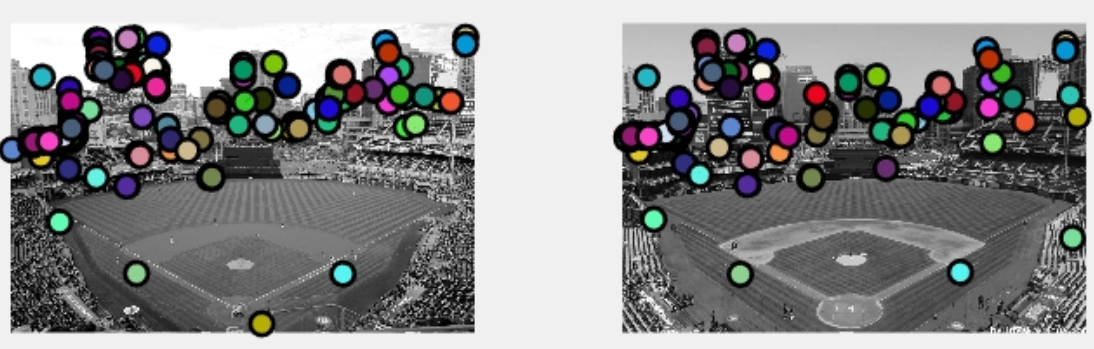
Interest Point Threshold: 0.01\*(Score Max) K=0.06 NNDR: <0.7



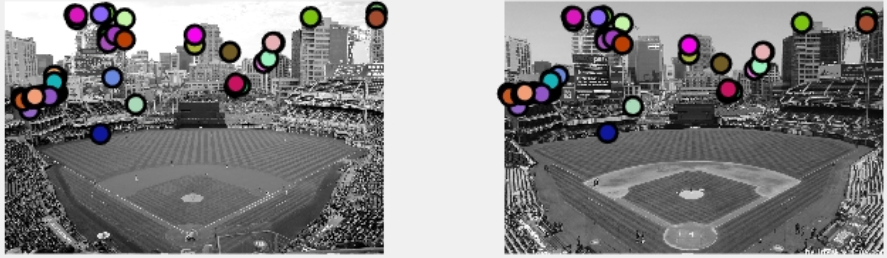
Interest Point Threshold: 0.01\*(Score Max) K=0.06 NNDR: <0.8



Interest Point Threshold: 0.01\*(Score Max) K=0.06 NNDR: <0.8



Interest Point Threshold: 0.01\*(Score Max) K=0.06 NNDR: <0.7



**Conclusion:**

Increasing Harris score threshold can get more interesting points. This can improve the accuracy, so I will keep the interesting around one thousand.

Another parameter, thresholds of NNDR. Greater threshold gets less features. Lower threshold gets more features. However, it’s not the more features, the better. In my report, I used 0.6, 0.7, 0.8 to get features. Finally, the accuracy can approach to 85%.