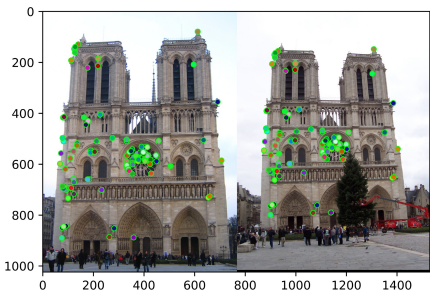
anirudh choudhary

Project 2: Local Feature Matching



Feature detection for Notre Dame Image

The assignment involves developing features detection and descriptor algorithm for local feature matching. The algorithm will be used for instance level matching i.e. matching multiple views of the same scene. The algorithm involves 3 steps:

1. Corner detection using Harris corner detector
2. Feature description using SIFT based features
3. Feature matching using nearest neighbourhood algorithm

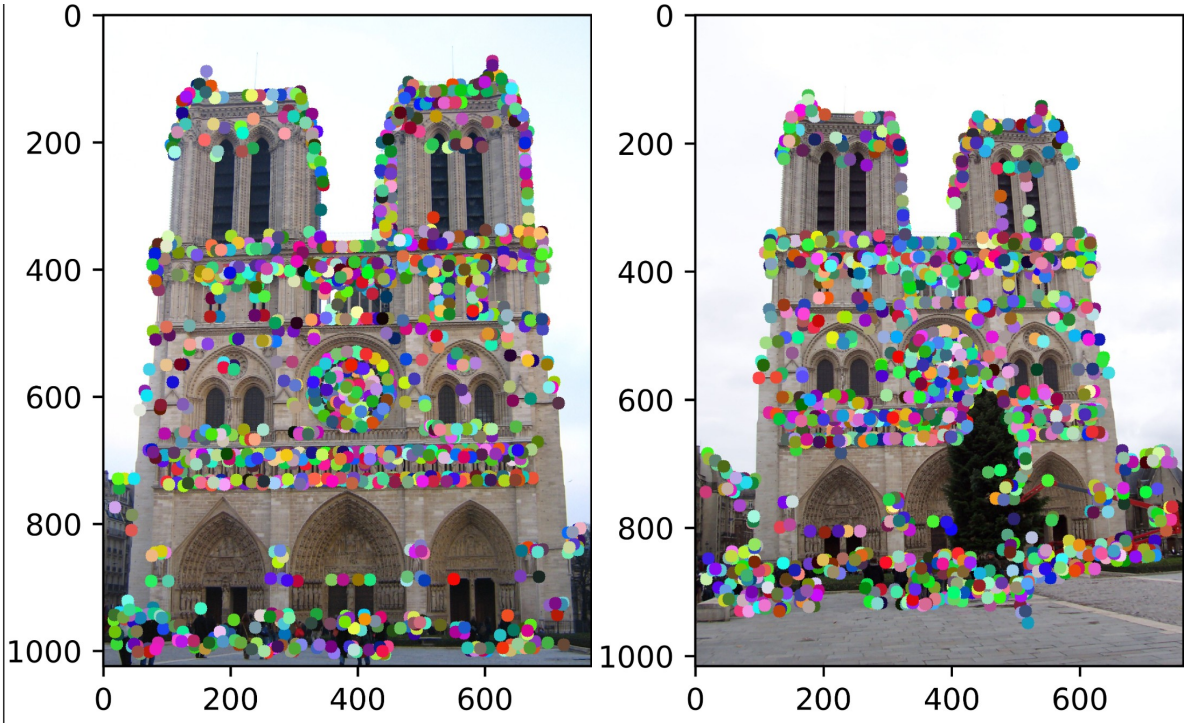
We analyse the accuracy of our algorithm over 3 scenes and implement certain enhancements (ANMS based nonmax suppression, GLOH features, KD Tree based matching) to improve the interest points matching accuracy.

Approach

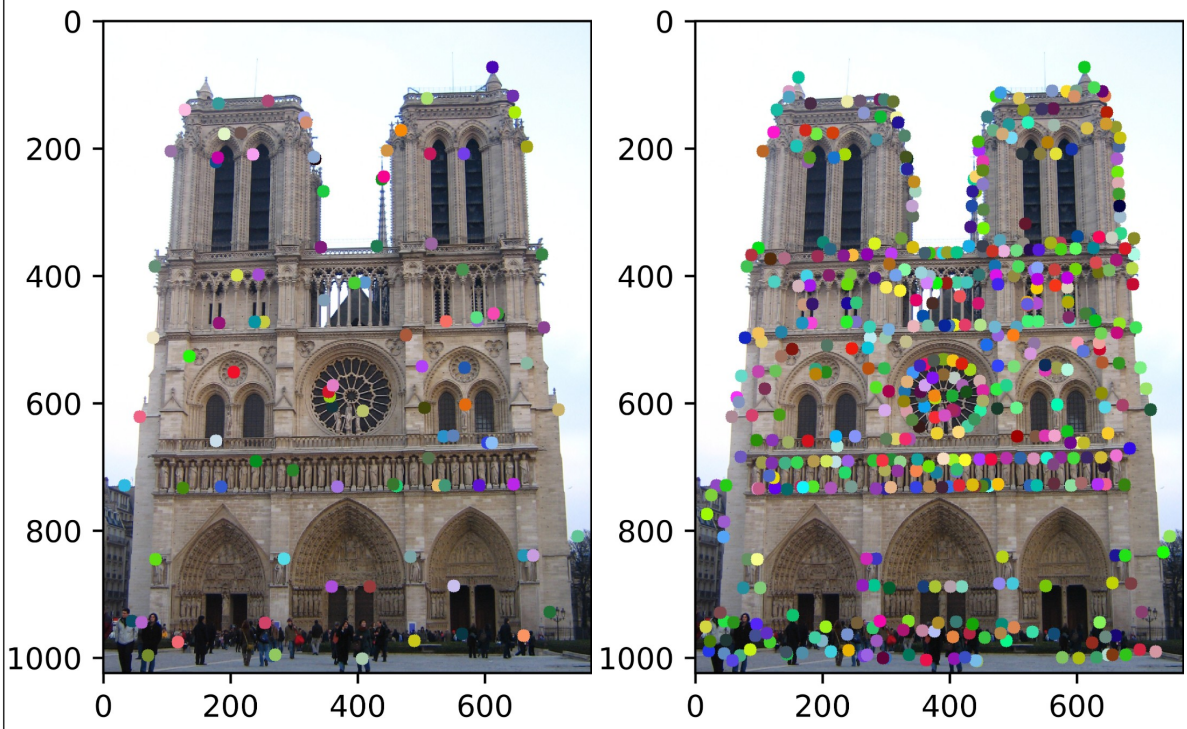
Harris Corner Detector

I blurred the image using a Gaussian function of sigma = 1 and size 4*sigma+1. This is to ensure that the output of edge detection using a Sobel filter is more robust to noise present in the image. After computing gradients Ix and Iy, I convolved their squares with Gaussian filter of sigma = 2 and size = 2*sigma+1 and computed the Cornerness score C for each pixel. I used threshold of 5 (C/mean(C) >= 5) for determining the corners and applied non-maximum suppression on each corner point taking 8 neighbour's value into account. This reduces the corner count significantly. However, not all corners are well separated and to rank the corners in terms of their distance from other corners, I apply Adaptive Non-Maximal Suppression using robustness constant c = 0.9. The corners are arranged in descending order of suppression radius. The first 100, 500 and 1000 corners after ANMS are better separated compared to similar versions of images obtained after window based non-maximal suppression. I use 1500 corners from the images for feature description and matching.

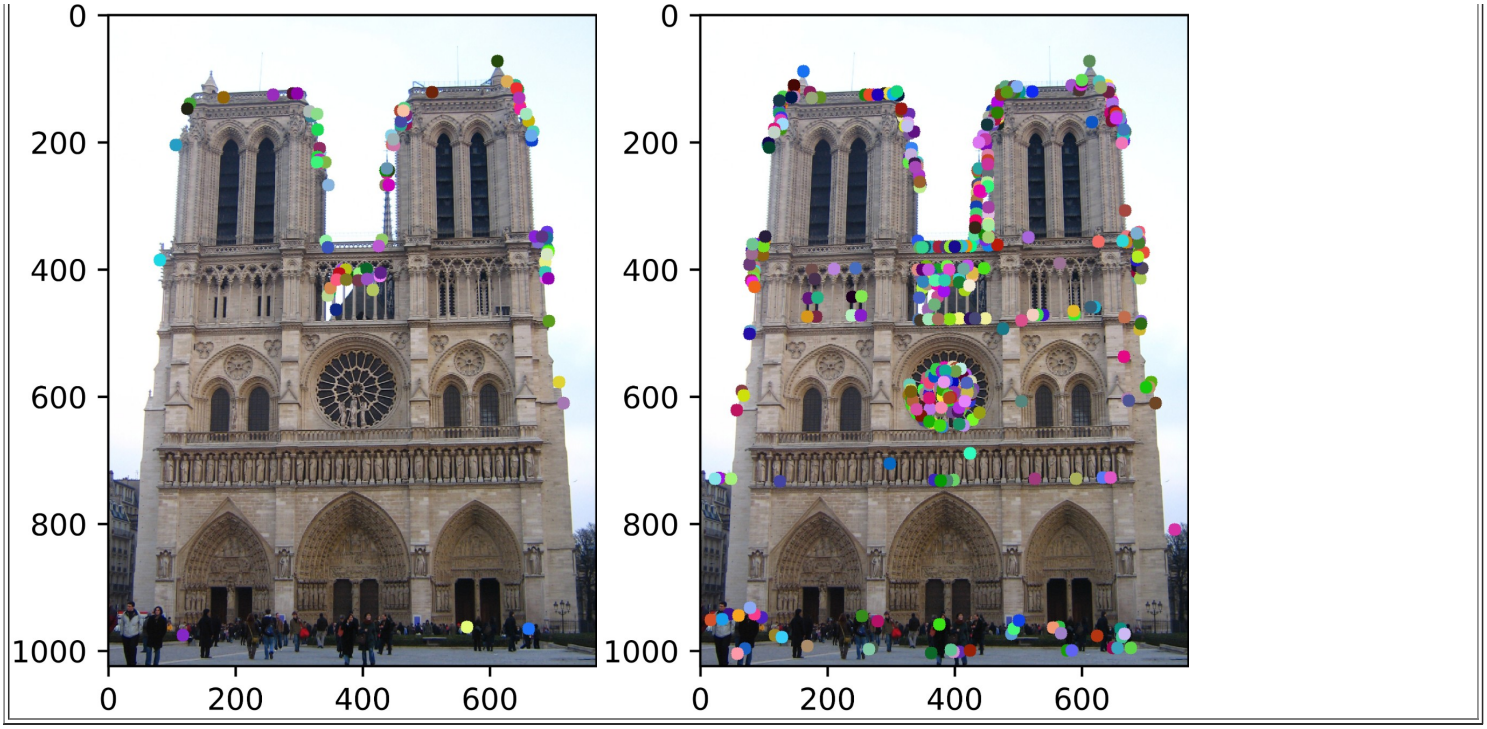
1500 corners used for feature matching (Notre Dame)



Corners detected using ANMS (100,500 corners)



Corners detected without using ANMS but local non-maximal suppression (100,500 corners)



Feature creation using SIFT-like features

Initially, I blurred the image with a gaussian filter and calculated the image gradients (angle and magnitude). Then, around each corner point, I considered a window of size feature_width and weigh the gradient magnitude with a gaussian filter. I paritition the window into 16 blocks and within each block, I divide the pixels into 8 orientation bins at a angle difference of 45 degrees and assign gradient magnitude to corresponding bin. I created a feature vector of size 4x4x8 and normalzied it. I also applied capping using a cut-off of 0.2 and normalized again. The capping seems to improve accuracy in Notre Dame and Mount Rushmore.

Feature creation using GLOH features (Extras)

Instead of dividing the window into 16 blocks, GLOH is based on log-polar binning. I divide the region around each corner into 17 regions (radii 6,11,15 and 8 orientations) and compute the magnitude weighted histogram (8 bins) for each region. The feature vector is of size 17*8. The flag GLOH within student_sift controls whether GLOH features are computed or not.

Feature Matching

Feature matching relies on confidence scores computed using nearest neighbours distance ratio(NNDR) method. I use the euclidean measure to compute the distances. NNDR is computed for feature 1 as:

```
NNDR = Distance to closest neighbour in feature 2/Distance to next closest neighbour in feature 2
```

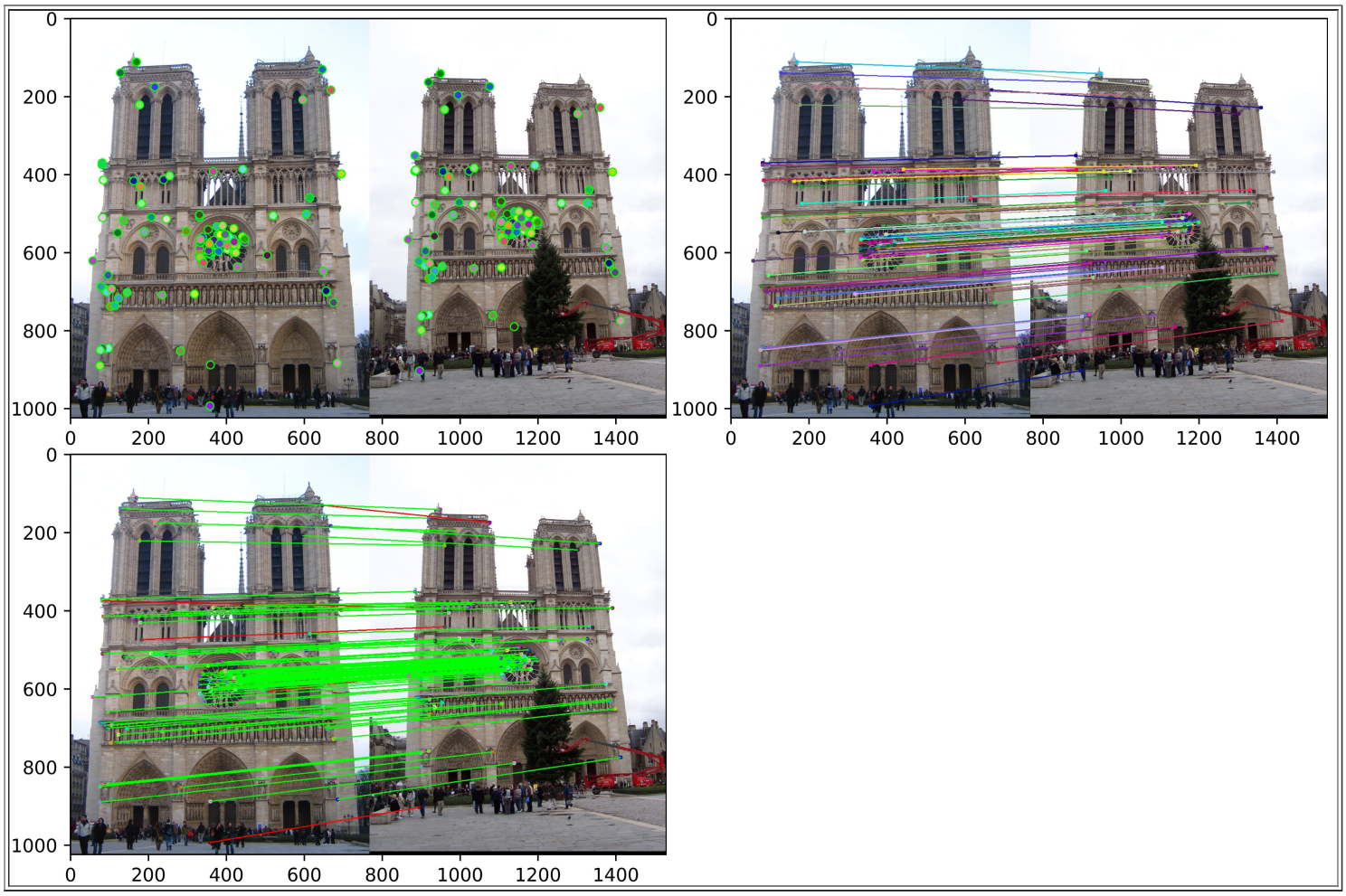
Then, all points with NNDR below 0.85 are considered as matches in image 1. Also, after NNDR, some points in image 1 might be matched to same point in image 2. Among these points, I picked the point in image 1 which has the least distance to the point in image 2.

EXTRAS - I also implemented a KD-tree based feature matching algorithm for improving the runtime of the matrix based approach used above. The KD-tree approach improves the runtime significantly and the accuracy doesn't vary much.

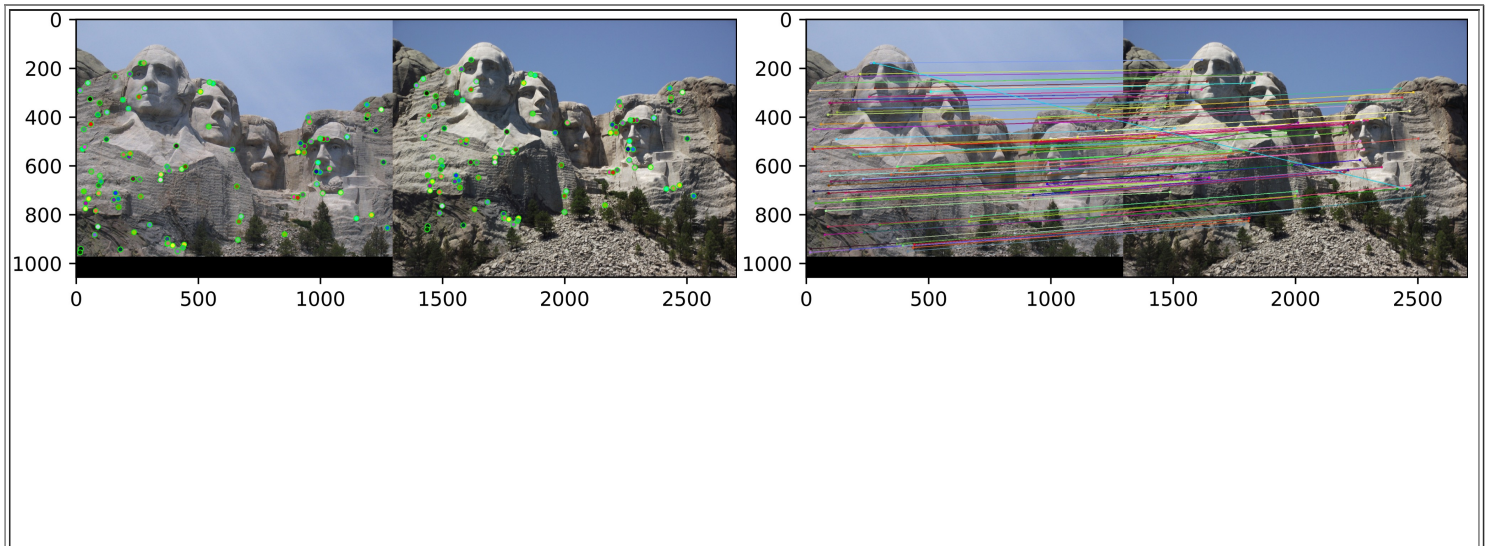
Evaluation

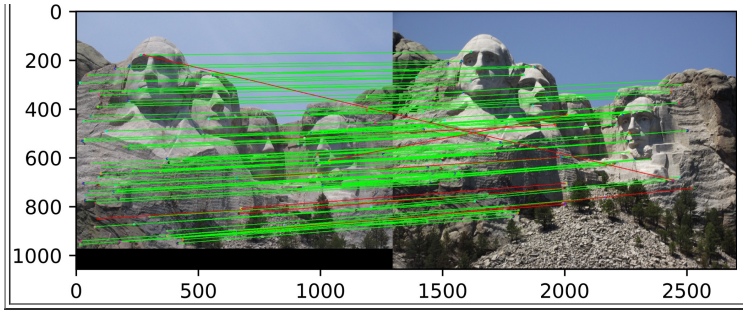
1. Accuracy using Harris Corner Detector (gaussian sigma=2), SIFT(16x16) & NNDR

Notre Dame (Accuracy - 95%)

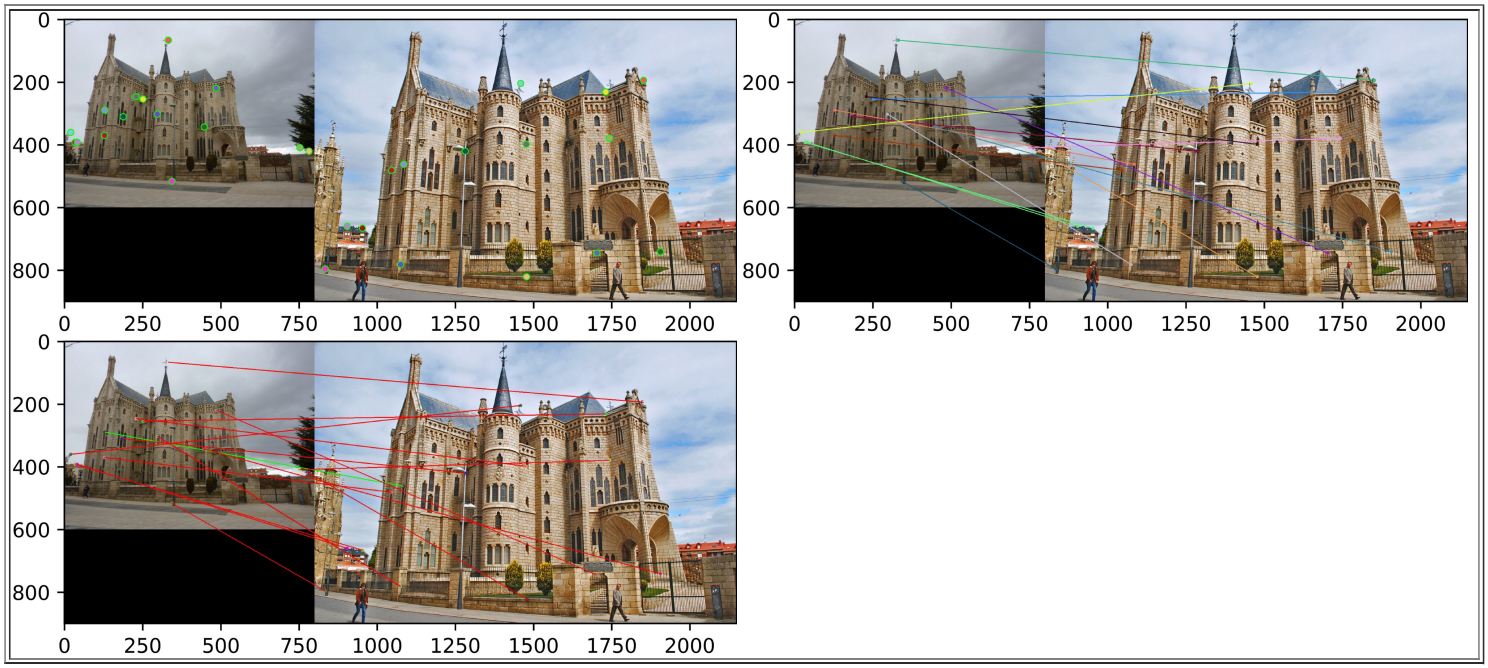


Mount Rushmore (Accuracy - 91%)



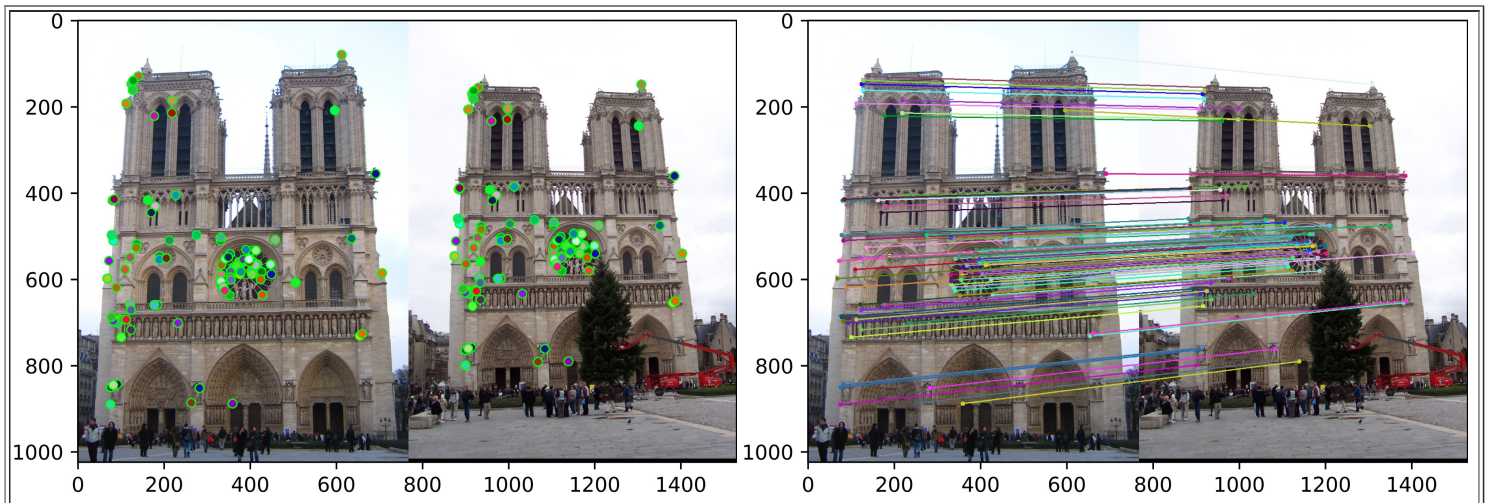


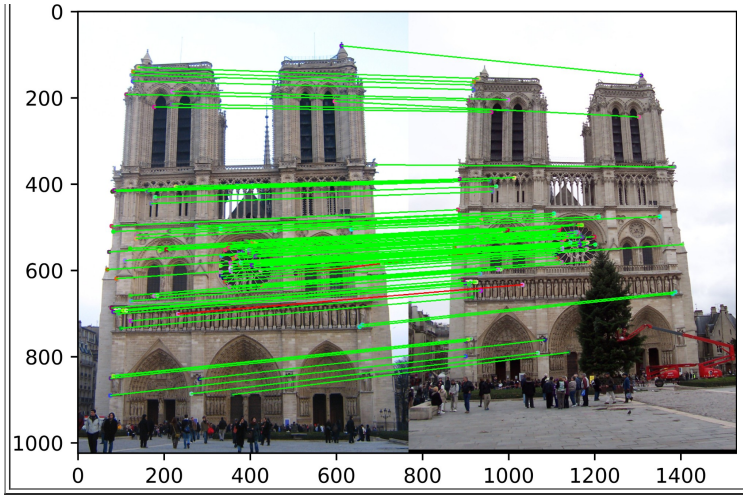
Episcopal Gaudi (Accuracy - 1%)



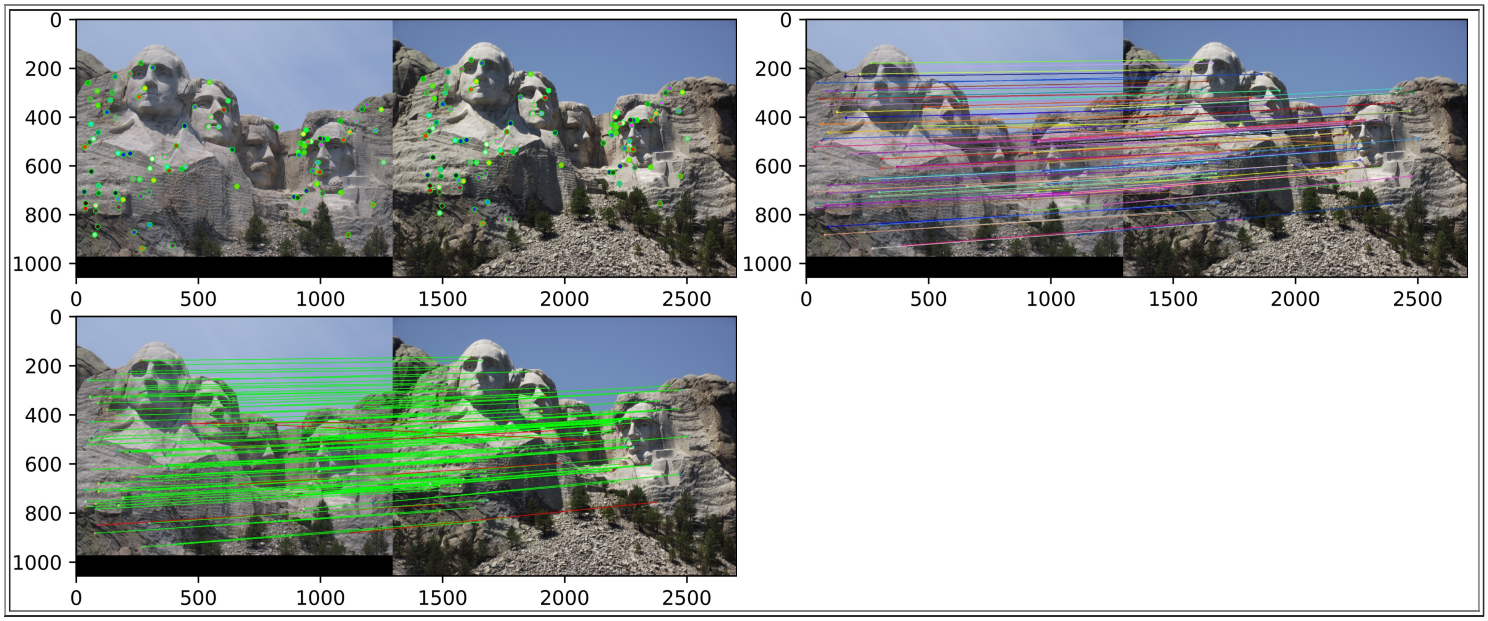
2. Accuracy using Harris Corner Detector(gaussian sigma=2), SIFT(32x32) & NNDR

Notre Dame (Accuracy - 98%) - The abberations due to people are removed in this case

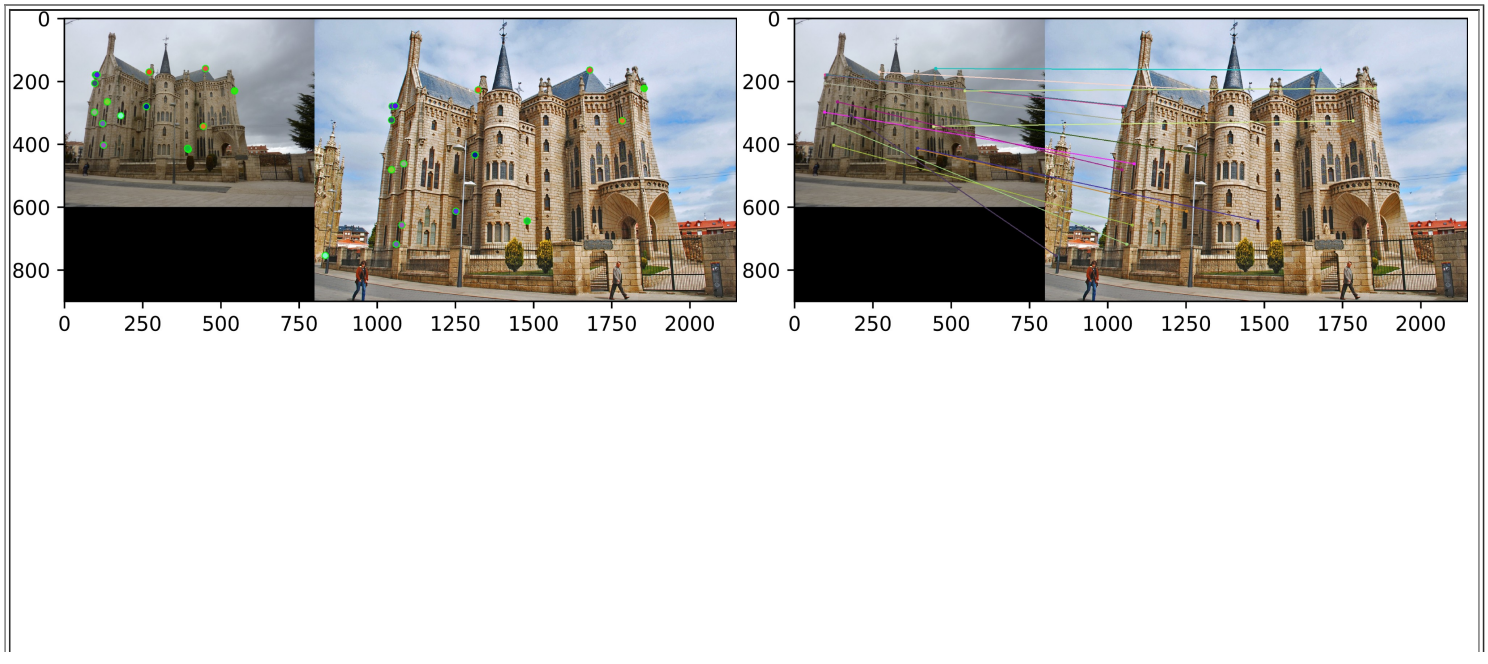


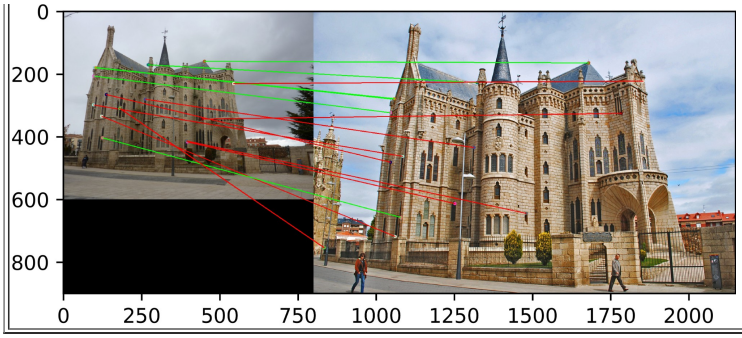


Mount Rushmore (Accuracy - 95%)



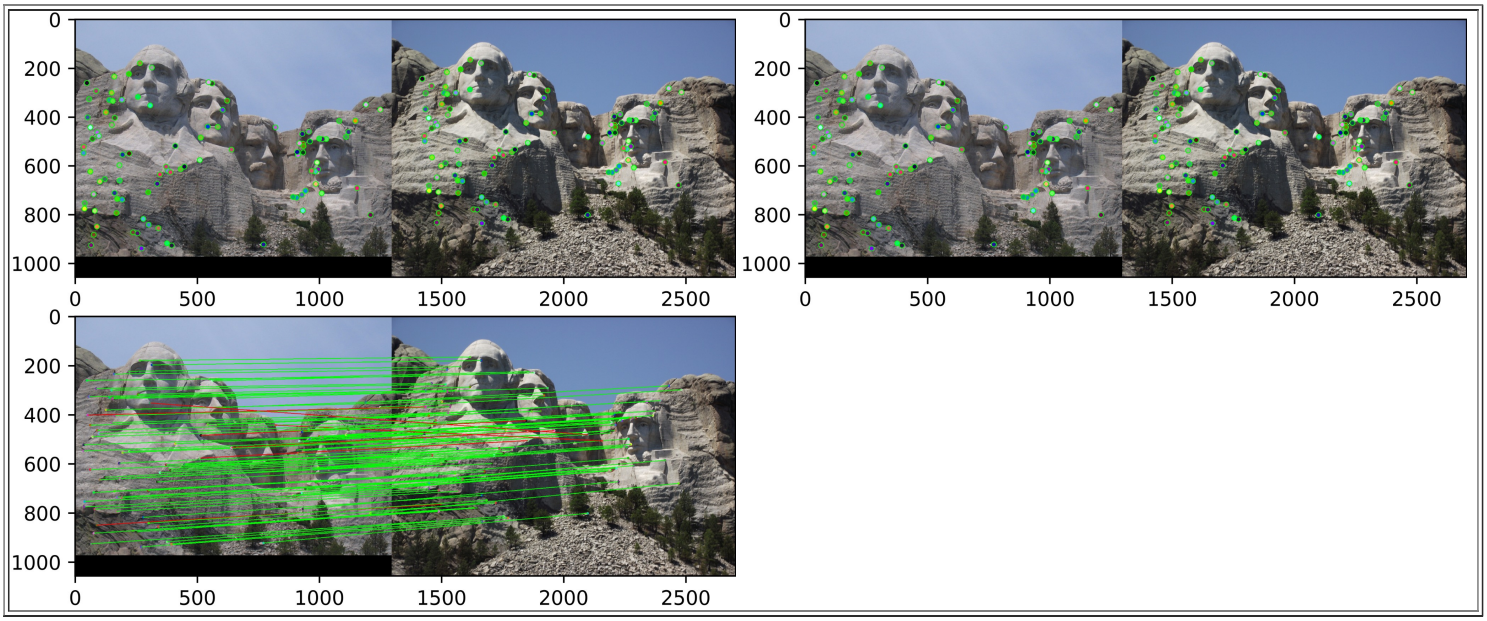
Episcopal Gaudi (Accuracy - 6%)



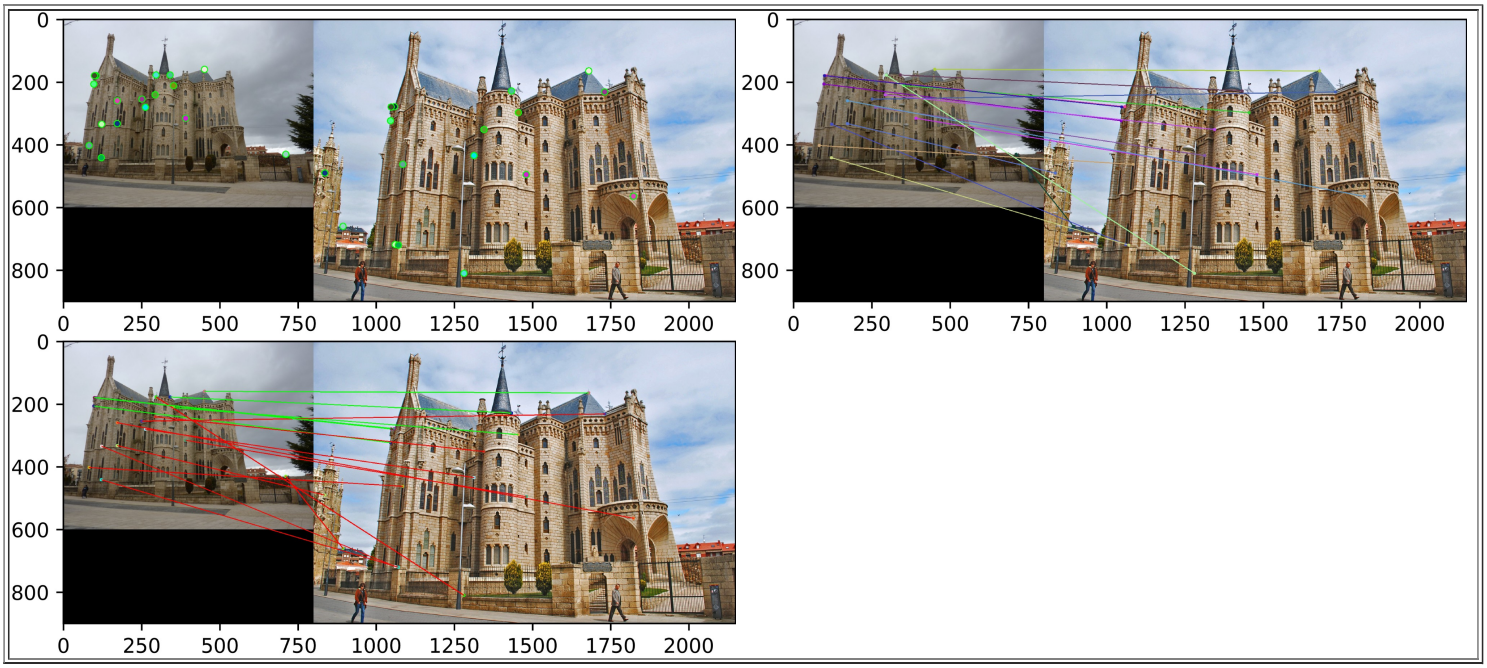


3. Accuracy using Harris Corner Detector(gaussian sigma=2), GLOH & NNDR

Mount Rushmore (Accuracy - 93%) - GLOH leads to 2% higher accuracy than the baseline SIFT with feature_width=16

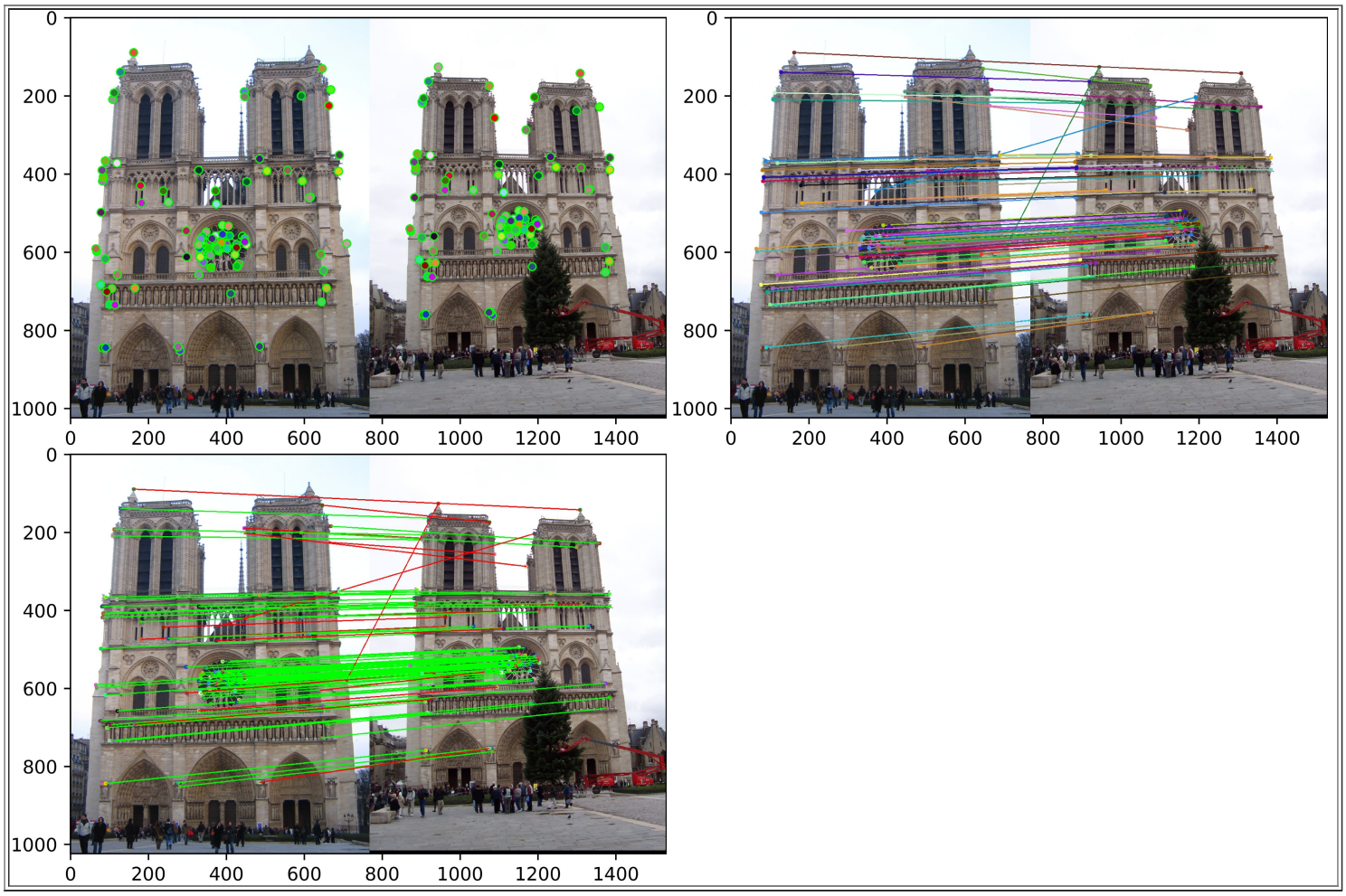


Episcopal Gaudi (Accuracy - 6%)

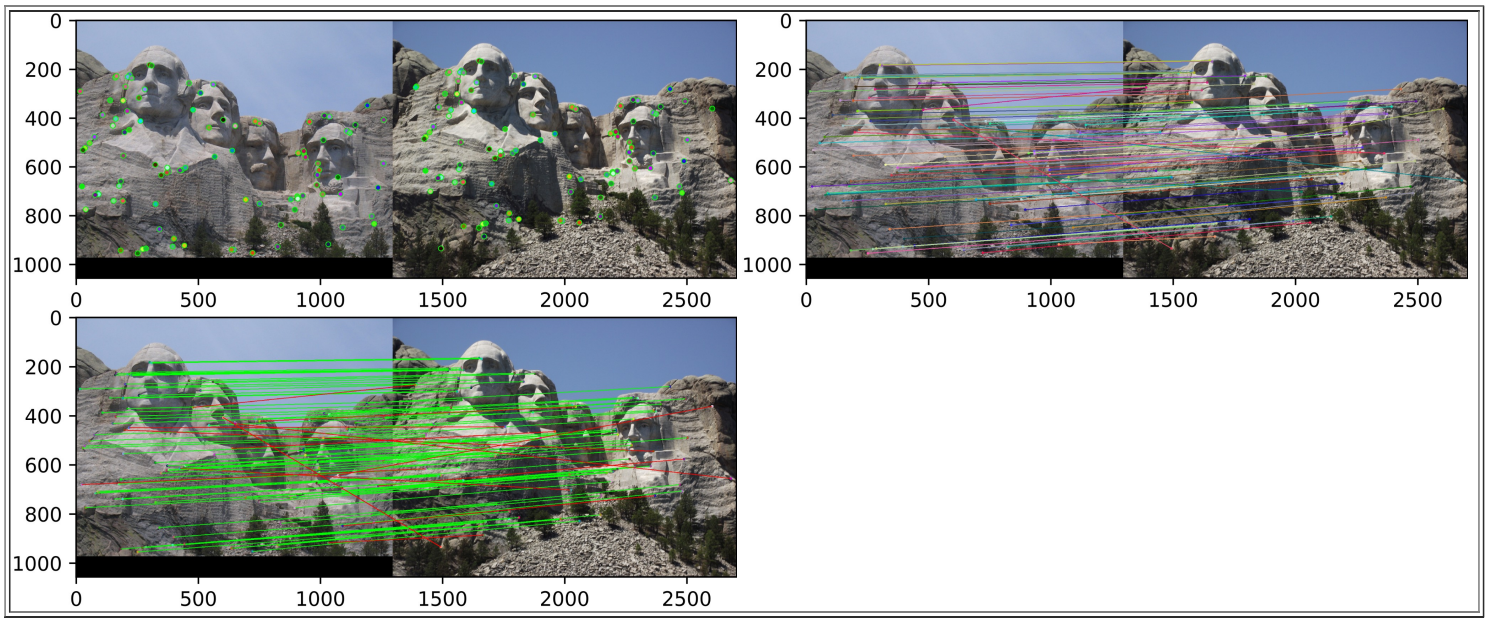


4. Accuracy using Harris Corner Detector (gaussian sigma=2), SIFT(16x16) & NNDR

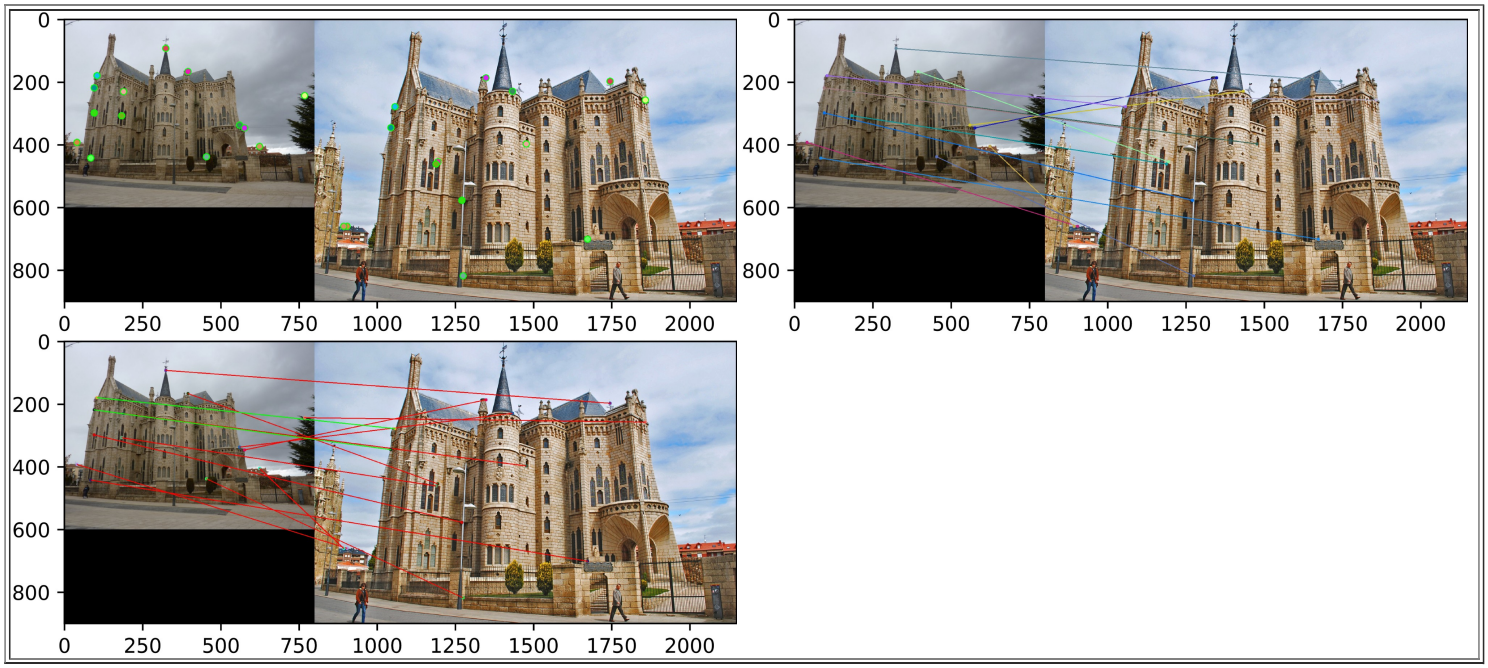
Notre Dame (Accuracy - 83%) - In this case, only 1062 corners are found for image 1 while 981 corners for image 2



Mount Rushmore (Accuracy - 87%) 1500 corners are found during the Harris corner detector step



Episcopal Gaudi (Accuracy - 2%) 632 corners are found in first image



5. Efficiency in Feature Matching

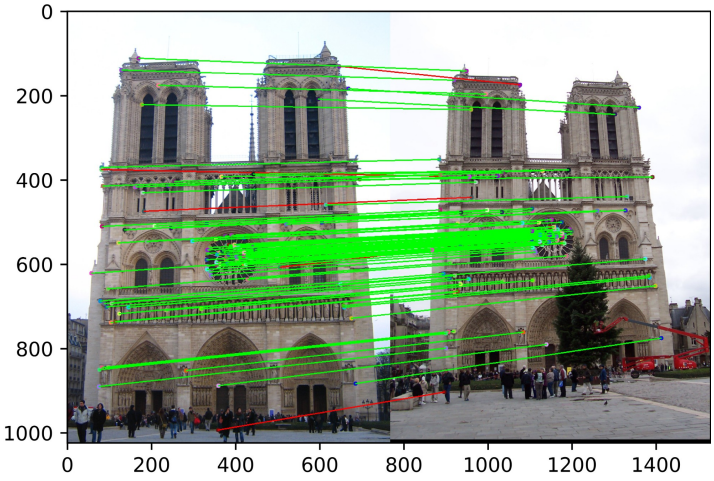
I have implemented KD-Tree using scipy's cKDTree function. The execution is about 10 times faster than matrix approach in computing NNDR. The result below is for 1500 features.

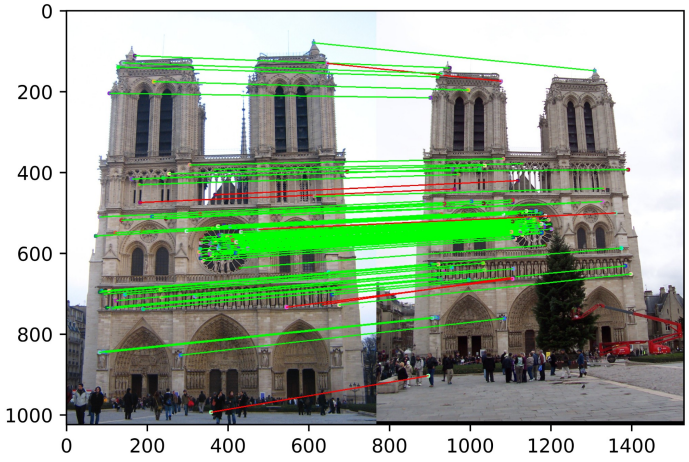
```
Matrix Execution Time:3.0867443084716797
168 matches from 1500 corners

KD Tree Execution Time:0.34607505798339844
168 matches from 1500 corners
```

Conclusion

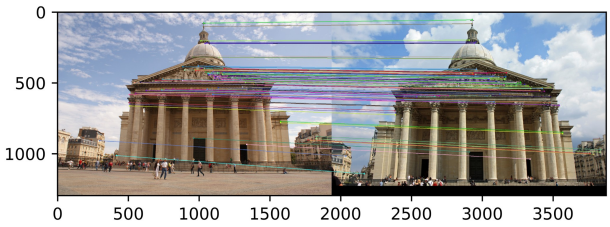
We have been able to attain good accuracy on Notre Dame and Mount Rushmore scenes by varying the SIFT descriptor size as well as considering GLOH descriptor. We found Gaussian with sigma 2 to be better than Gaussian with sigma 3 (the integral gaussian for calculating cornerness) in terms of accuracy. Gaudi scene gives very low accuracy primarily due to the difference in scale of the images. We can further improve the accuracy by using Harris Laplace operator and calculating orientation at each corner using a window of 1.5 times the Gaussian used for cornerness measure computation and computing the dominant gradient in that window using histogram binning. Also, we have found that clipping generally helps in improving accuracy (+2% for Notre Dame using 16x16 SIFT descriptor).
Some more results are show below:

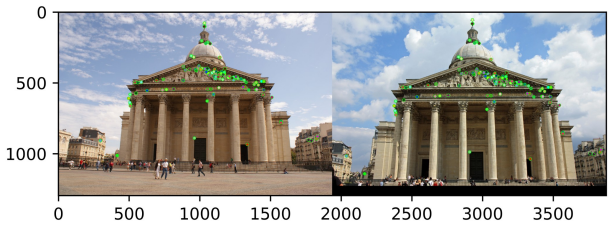
Notre Dame with feature vector clipping (left) and without featire vector clipping (right)





Pantheon (335 matches, most of the matches seem to be correct)





Statue of Liberty (96 matches, approximately 15 are incorrect)