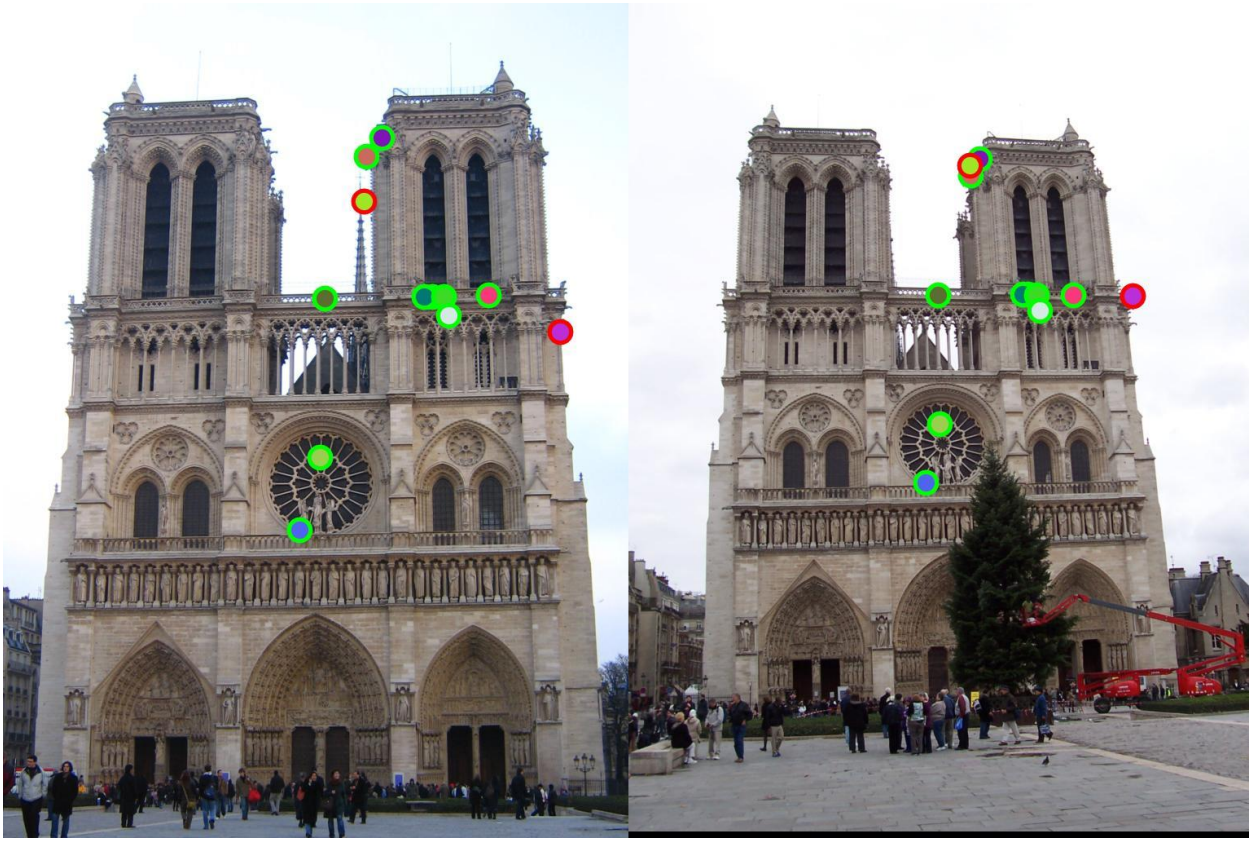
In this assignment we were tasked with implementing a SIFT matching algorithm, which would be able to find points of interest in an image, namely a corner, and then create a feature vector based on that. The algorithm would then be able to check features between two different feature vectors and determine whether or not there was any similarities at all. Through this, our algorithm would allow for a confidence detection between how likely it was that two images were of the same image.



1)

When picking a good feature detector it really depends on the dataset that is being given to the user. Based on the data, one would have to weigh the pros and cons between various different feature detectors. The trade off namely being, the difference between the invariance of the interest points, and the precise discrimination of interest points. The Harris detector for instance is great for finding precise x,y values of points of interest, however, since these points are so precise, they are not very robust to rotational or illumination changes. Using LoG detection methods however, is very robust to rotational and illumination, but is much more substantial to code as well as is not as precise.

2)

The eigenvalues of the moment matrix represent the ellipse that contains the distribution of gradients around a pixel. From lecture 4.1 slide 48, the eigenvalues of lambda can be represented as the axis lengths of the ellipse, and the eigenvectors, R of E are the orientations.

3) A good method for feature descriptor matching is the nearest neighbor ratio test. In general, it is the most optimal nearest neighbor algorithm. This is a very linear search and essentially finds the two nearest neighbors to the feature, and takes the distances from both of them. Then it finds the ratio of those two differences. This is computationally efficient depending on the data set characteristics.