Assignment 2: Local Feature Matching

Aim:

The aim of this assignment is to create an algorithm for local feature matching. This is based on techniques described in Szeliski chapter 4.1. For this, we match multiple views of the same physical scene.

Setup for the assignment:

For the first attempt, I tried to run it on my local machine using IntelliJ Idea. Wasn't able to install the required libraries. Hence I switched to jupyter notebook, was able to install the libraries required.

The solution I took is inherited from a github repository.

Install anaconda package. Anaconda package comes with Jupyter Notebook.

Install numpy, opencv and other such required libraries.

Run the anaconda prompt, initialize jupyter, and open the project.

Part 1: Harris Corner Detection for keypoint identification.

First we apply Gaussian filter, for a better processing, and then we calculate gradients.

$$\mathbf{A} = \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix},$$

Here, Ix and Iy are image gradients in two different directions.

The autocorrelation matrix is the gradient matrix here.

$$R = \det(\mathbf{A}) - \alpha\ \mathrm{trace}(\mathbf{A})^2 = I_x^2 I_y^2 - (I_xI_y)^2 - \alpha(I_x + I_y)^2.$$

The Response R is calculated.

In the original code, the value of alpha was 0.06, which was changed to 0.3.

Identification of the local maxima:

For better identification of feature points, image dilation is used to identify local maxima. When the local maxima are calculated, there is a reduction in the number of key points.

Adaptive Non-Maximal Suppression (ANMS):

High contrast areas often give out a lot of key points. Since we have limited the number of key points based on their priorities (magnitudes), it would be advantageous to have key points all over the image, rather than them being focused on the high contrast areas.

For this, the nearest neighbor with a high corner value was identified. The distance to this point was termed as radius, and then they were sorted by the radius values. Out of them, the top 1500 points(for this code snippet) were chosen.

Feature Description:

Once the points have been identified, they have to be described.

A feature descriptor is generated, by generating a 16 x 16 pixel patch. The Scale Invariant Feature Transform (SIFT) like descriptor is generated, at each keypoint. The gradients Ix and Iy are polarised and determined in a 16 x 16 pixel neighborhood, with the key point as its center.

A histogram is built by adding the magnitudes corresponding to each of the 8 directions. The result would be a 16 x 8, which would be converted to a 128-element feature vector, corresponding to each keypoint.

Feature Matching:

A Nearest Neighbour search algorithm is implemented to compare the feature points in the two images. The Nearest Neighbour Distance Ratio (NNDR) is calculated.

$$\mathbf{NNDR} = \frac{d_1}{d_2},$$

The values of d1 and d2 are Euclidean distances into their first and second neighbors.

The closer NNDR is to 1 (can be <0 or >0), the less confidence in the match. The higher value suggests that the points are further away from each other, and hence a chance of matching based on their descriptors is better. i.e the closer the values of the feature descriptors, the better the match.

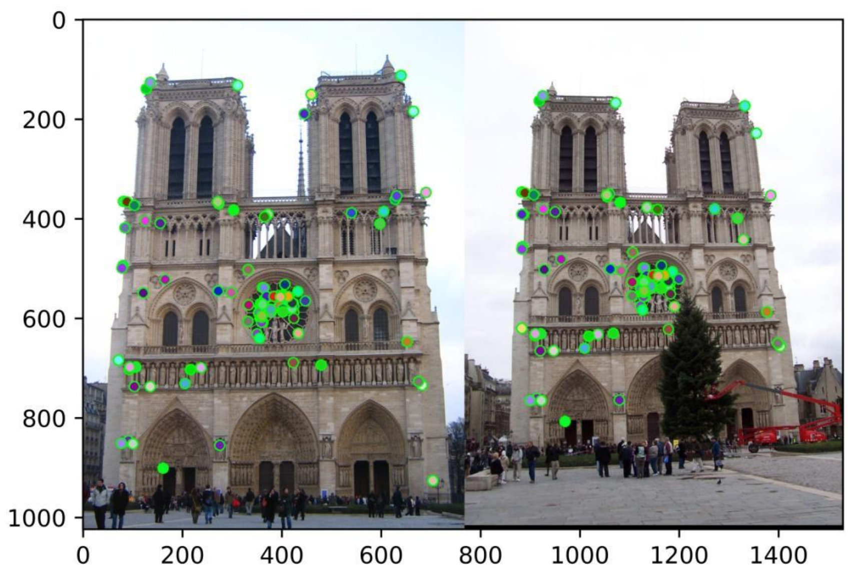
Results:

For the first image pair: Notre Dame

The value of alpha used is 0.03 for the pictures.

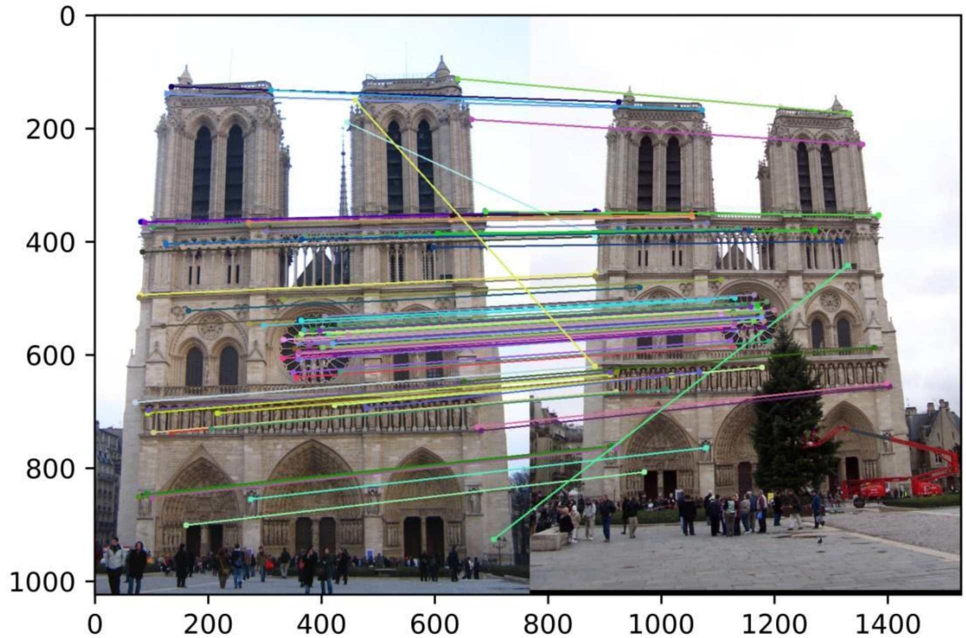
Accuracy turned out to be 93% for the pair.

Key points:

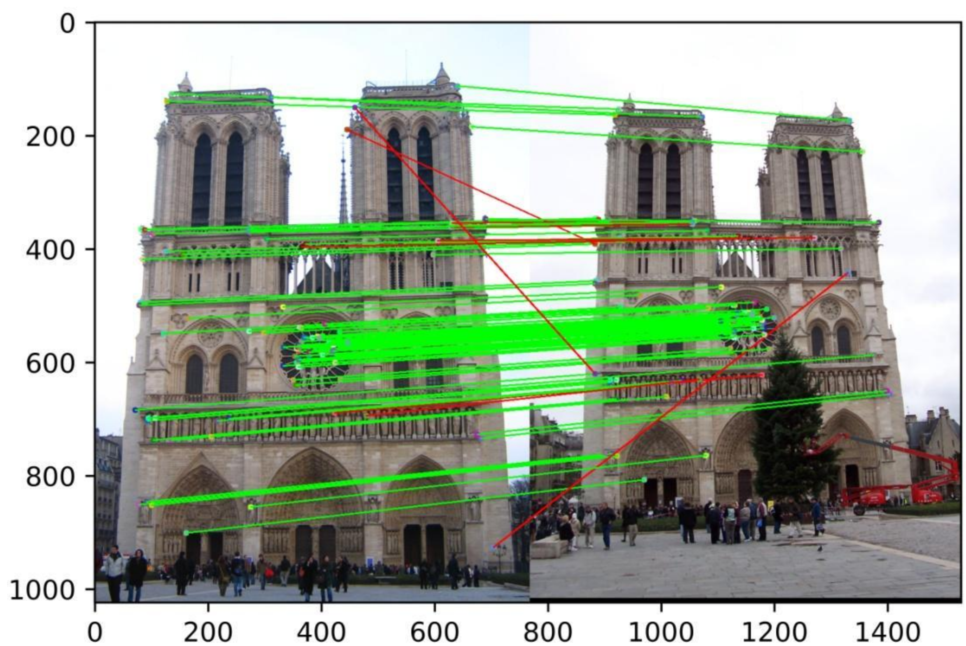


Match features:

200465976



Matching Evaluation:

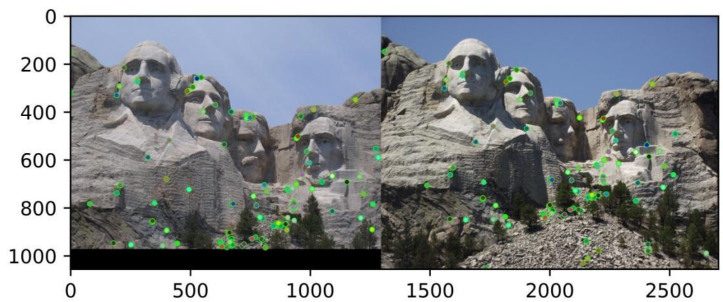


For the second image pair:

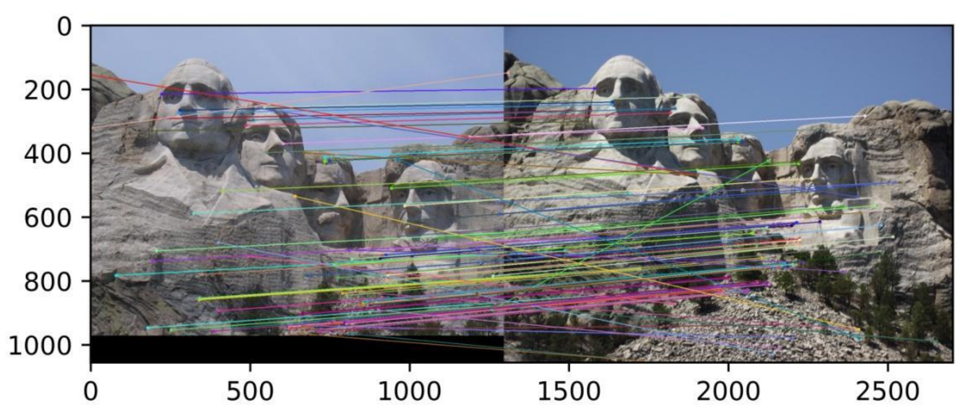
Value of alpha: 0.01

Accuracy: 80%

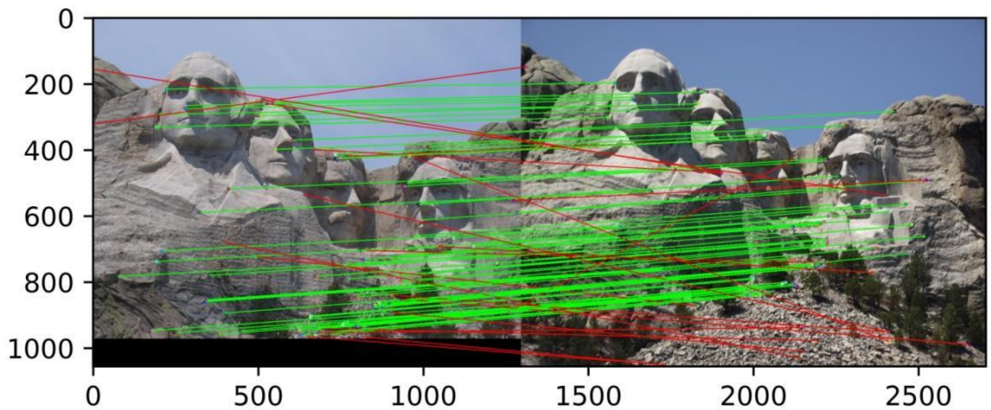
Key points:



Matching features:



Matching Evaluation:

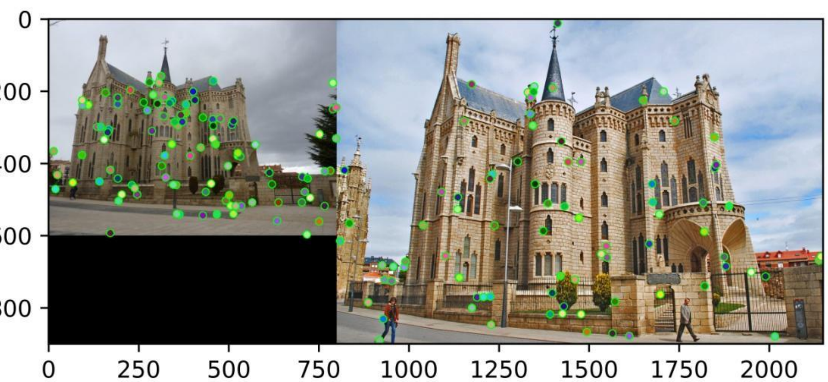


For the third image pair: Episcopal Gaudi

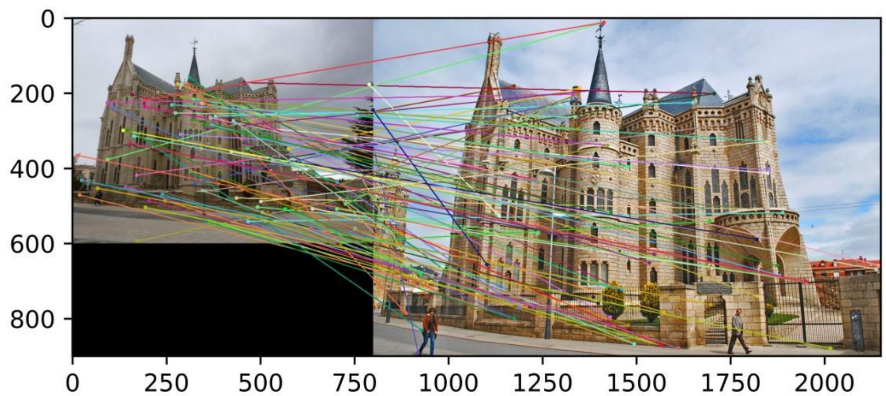
Value of alpha: 0.2

Accuracy: 6%

Key points:



Match features:



Matching Evaluation: