CIE 552 mini-project 2

Team name: Kimochi yamate

Ahmed Mahmoud Abd-Elmoniem 201800683

Hazem Muhammad Tarek 201800283

Local feature matching

Introduction

This project focuses on feature matching between two images. It consists of three main steps:

• Interest points detection: using Harris Corner Detector algorithm.

• Local feature description: Representing the features and regions around the location in the image as a vector called feature vector using Scale-Invariant feature transform (SIFT) technique.

• Feature matching: Matching features by calculating the distance between features and find similarities by implementing the Nearest Neighbor Distance Ratio algorithm.

Interest points detection algorithm

- Blur the image using the gaussian filter.
- Calculate the gradient of the image using the following equation: $\nabla I_0(x_{i)} = (\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y})x_i$.

- Calculate the following parameters:

$$S_{xx} = \sum_{x,y} g(I_x^2)$$
$$S_{yy} = \sum_{x,y} g(I_y^2)$$
$$S_{xy} = \sum_{x,y} g(I_{xy})$$

- Calculate the C matrix using the following equation:

$C = S_{xx}\, \mathrm{o}\, S_{yy} - s_{xy}^2 - alpha * (S_{xx} + S_{yy})^2$

- Select interest points based on a certain threshold.

SIFT implementation algorithm

- Represent the points in a 128-point vector.
- Blur the image using the gaussian filter.
- Calculate the gradient of the image using the following equation: $\nabla I_0(x_{i)} = (\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y})x_i$.
- Calculate the magnitude and angle gradients using the following equations:

$$magnitude\ gradient = \sqrt{I_x^2 + I_y^2}$$

$$Direction\ gradient = \arctan(I_x, I_y)$$

- Apply a gaussian filter on a 16x16 size window.
- Divide the 16x16 slots into 16 4x4 slots.
- On the edge of the 4x4 table, calculate the histogram with bins=8 for the 128-point vector at each interest point.
- Reshape the 16x8 matrix into a 128-point vector.
- Normalize the vector.
- Clamp all vector values > 0.2 to 0.2.

Feature matching

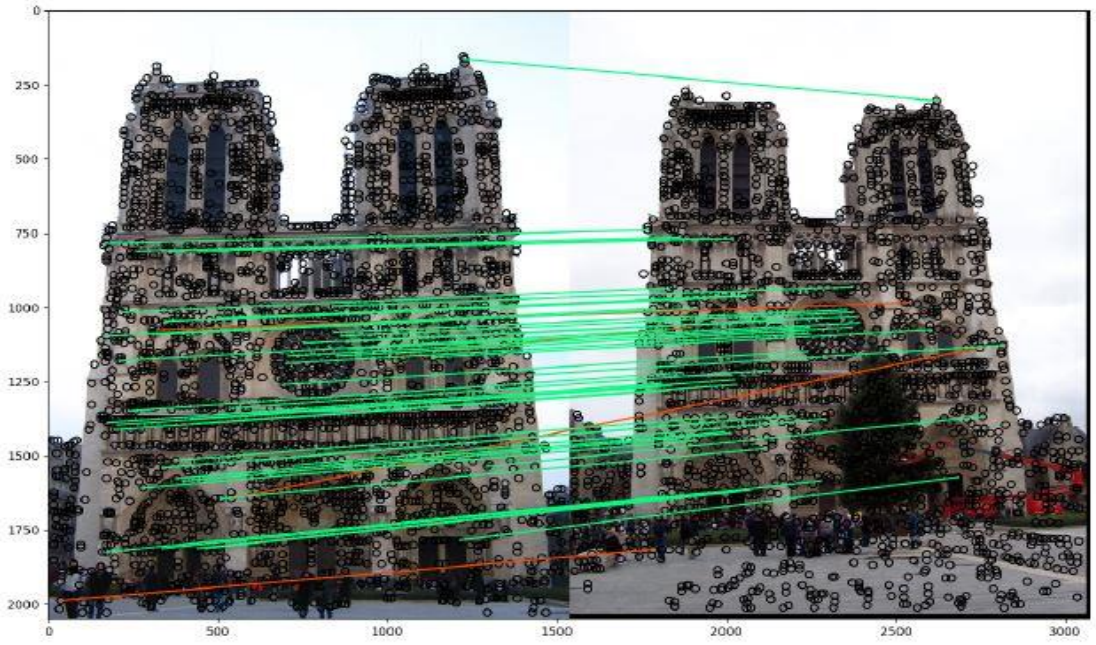
- Compare the points using the nearest neighbor distance ratio.
- Calculate the Euclidean distances using the following equation:

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

- If the 0.8 distance is equal the 2nd distance, then the value vector with minimum distance is calculated as a match between two keys points.

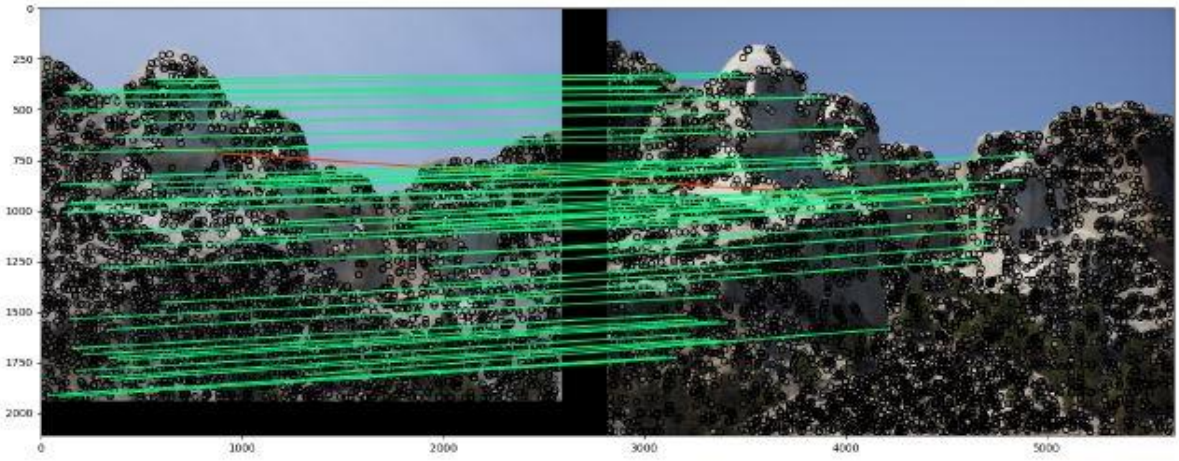
Results

Notre Dame



```
Getting interest points...
alpha: 0.06, threshold: 0.01, stride: 2, sigma: 0.1, sigma0: 0.1,
    min_distance: 3
alpha: 0.06, threshold: 0.01, stride: 2, sigma: 0.1, sigma0: 0.1,
    min_distance: 3
Done!
Getting features...
sigma_gradient_image: 0.1, sigma_16x16: 0.4, threshold: 0.2
sigma_gradient_image: 0.1, sigma_16x16: 0.4, threshold: 0.2
Done!
Matching features...
threshold: 0.8
Done!
Matches: 134
Accuracy on 50 most confident: 92%
Accuracy on 100 most confident: 83%
Accuracy on all matches: 82%
```

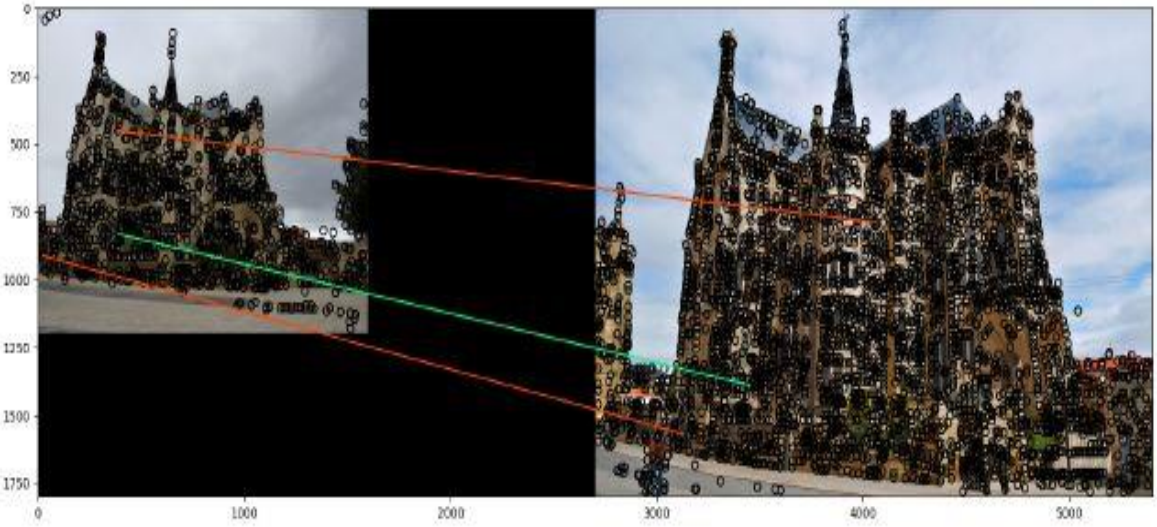
Mount Rushmore




```
Getting interest points...
alpha: 0.06, threshold: 0.01, stride: 2, sigma: 0.1, sigma0: 0.1,
    min_distance: 3
alpha: 0.06, threshold: 0.01, stride: 2, sigma: 0.1, sigma0: 0.1,
    min_distance: 3
Done!
Getting features...
sigma_gradient_image: 0.1, sigma_16x16: 0.4, threshold: 0.2
sigma_gradient_image: 0.1, sigma_16x16: 0.4, threshold: 0.2
Done!
```

```
Matching features...
threshold: 0.8
Done!
Matches: 168
Accuracy on 50 most confident: 98%
Accuracy on 100 most confident: 99%
Accuracy on all matches: 96%
```

Episcopal Gaudi



```
alpha: 0.06, threshold: 0.01, stride: 2, sigma: 0.1, sigma0: 0.1,
    min_distance: 3
Done!
Getting features...
sigma_gradient_image: 0.1, sigma_16x16: 0.4, threshold: 0.2
sigma_gradient_image: 0.1, sigma_16x16: 0.4, threshold: 0.2
Done!
Matching features...
threshold: 0.8
Done!
Matches: 3
Accuracy on all matches: 33%
```