CMPE - 59H
Assignment 3
Interest Point Detectors

Abdullah Atakan Guney
2018700069

November 26, 2018

Contents

1 Harris Corner Detector

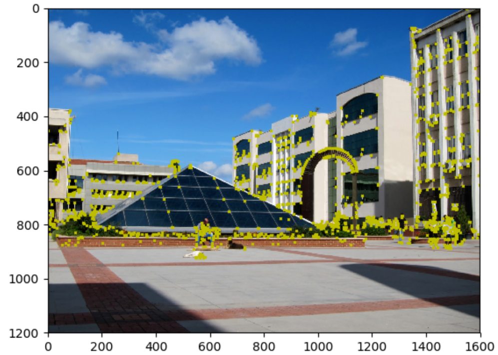
In this assignment, the aim is to compare different kinds of interest points detectors. In the first part, I have implemented ***Harris Corner Detector*** by using ”numpy”, ”opencv” libraries additional to python built-in methods. I have followed steps that have been described in this presentation. Pseudocode of my implementation is following:

Algorithm 1 Harris Corner Detection Alogrithm

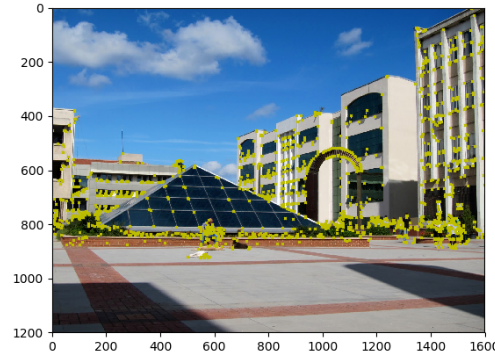
procedure MYHARRISCORNERDETECTOR
    $I_x \leftarrow$ Calculate Derivative with Sobel operator in x
    $I_y \leftarrow$ Calculate Derivative with Sobel operator in y
    $I_{x^2} \leftarrow I_x \odot I_x$
    $I_{y^2} \leftarrow I_y \odot I_y$
    $I_{xy} \leftarrow I_x \odot I_y$
    $S_{x^2} \leftarrow$ Convolve I_{x^2} with Gaussian kernel
    $S_{y^2} \leftarrow$ Convolve I_{y^2} with Gaussian kernel
    $S_{xy} \leftarrow$ Convolve I_{xy} with Gaussian kernel
    determinant of $H \leftarrow S_{x^2} \odot S_{y^2} - S_{xy} \odot S_{xy}$
    trace of $H \leftarrow S_{x^2} + S_{y^2}$
    $R \leftarrow$ determinantH $- k \times (traceH \odot traceH)$
    $R \leftarrow$ thresholding on R
    Compute non-max suppression on R
    return Local maxima of R

1.1 My Harris Corner Detector Results

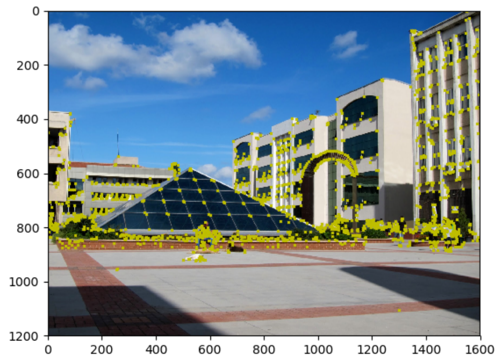
I have selected k as 0.04, ksize which is kernel size for Sobel operator as 3, block size which is kernel size for Gaussian kernel as 7. Here is the result of my Harris corner detector implementation on the picture of north campus. See the results with different block sizes



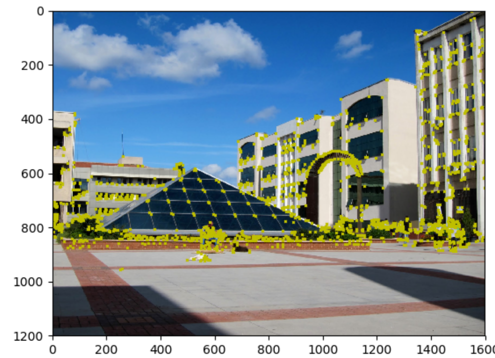
(a) Block size = 3



(b) Block size = 5



(c) Block size = 7



(d) Block size = 11

Figure 1: Harris with different block sizes

As we can easily observe that with block size increasing, the number of detected corner points is also increasing.

2 Data

There are 3 different data set I have used to compare 3 different kinds of corner detectors. These are following:

- My Harris Corner Detector
- SIFT (Scale-Invariant Feature Transform)
- SURF (Speeded-Up Robust Features)

2.1 JPEG quality different north campus

Here are versions with different JPEG quality of an image of north campus.

(a) With quality = 5

(b) With quality = 25

(c) With quality = 45

(d) With quality = 65

(e) With quality = 85

(f) With quality = 105

Figure 2: JPEG Qualities

As a note, I have used opencv's 'imwrite' function to get these images.

2.2 North Campus with different levels of Gaussian Noise

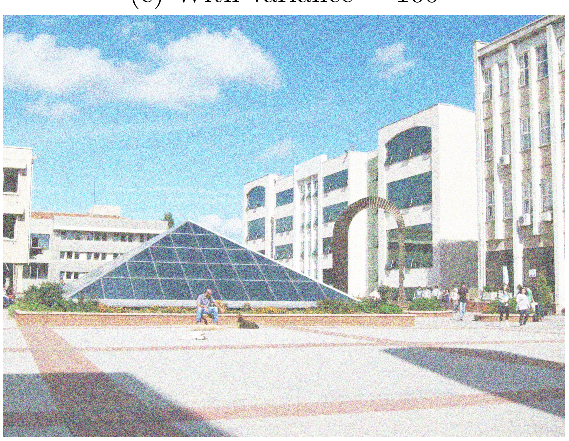
I have added Gaussian Noises with different variances to the image of north campus. Let us see the results.

(a) With variance = 5

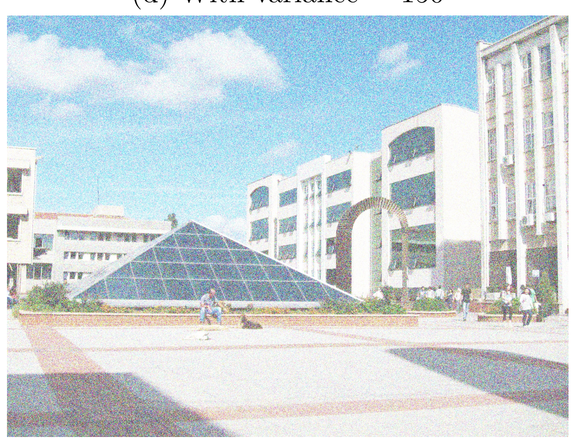
(b) With variance = 50

(c) With variance = 100

(d) With variance = 150



(e) With variance = 200



(f) With variance = 250

Figure 3: Different Levels of Gaussian Noise

2.3 Graffiti Image Set

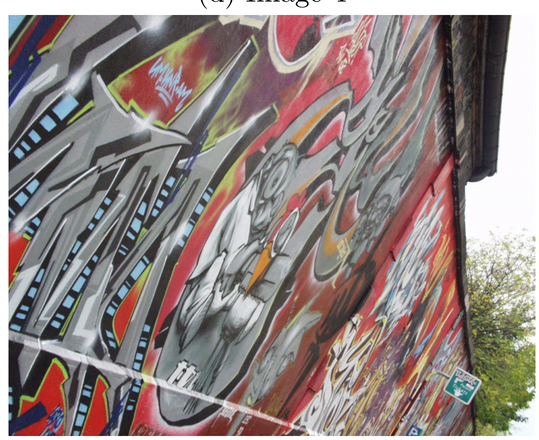
(a) Base Image

(b) Image 2

(c) Image 3

(d) Image 4

(e) Image 5



(f) Image 6

Figure 4: Graffiti Image Set

3 Measuring Repeatability

Repeatability is a measure that gives us the ability to compare different kinds of detectors. Here is the formula for it.

$$
\begin{aligned}
r_i(\epsilon) &= \frac{|R_i(\epsilon)|}{min(n_i, n_1)} \\
R_i(epsilon) &= \{(\tilde{x_1}, \tilde{x_i})\,|dist(H_{1i}\tilde{x_1}, \tilde{x_i}) < \epsilon, \text{ where } \tilde{x_1} \in \tilde{X_1}, \tilde{x_i} \in \tilde{X_i}\} \\
n_i &= \left|\tilde{X_i}\right| \\
n_1 &= \left|\tilde{X_1}\right| \\
\tilde{X_1} &= \{x_1 | H_{1i}x_1 \in \text{ Image i}\} \\
\tilde{X_i} &= \{x_i | H_{i1}x_i \in \text{ Image 1}\}
\end{aligned}
$$

Here is pseudocode of my implementation of this measure:

Algorithm 2 Repeatability Measure Algorithm

procedure MEASUREREPEATABILITY(keyPoints1, keyPoints2, H_{12}, image2size, $\epsilon = 1.5$)
  Convert keyPoints1 and keyPoints2 to homogeneous coordinates
  $keyPoints_{12} \leftarrow H_{12} \times keyPoints_1$
  $keyPoints_{21} \leftarrow H_{12}^{-1} \times keyPoints_2$
  $commonPart_1 \leftarrow keyPoinst_1[keyPoints_{12} \in \text{image2boundary}]$
  $commonPart_2 \leftarrow keyPoinst_2[keyPoints_{21} \in \text{image1boundary}]$
  $N_1 \leftarrow$ Size of $commonPart1$
  $N_2 \leftarrow$ Size of $commonPart2$
  $R \leftarrow \epsilon - neighborhood(H_{12} \times commonPart1, commonPart2)$
  return Size of $R/min(N1, N2)$

3.1 Results

Lets see the results for 3 different kinds of image sets.

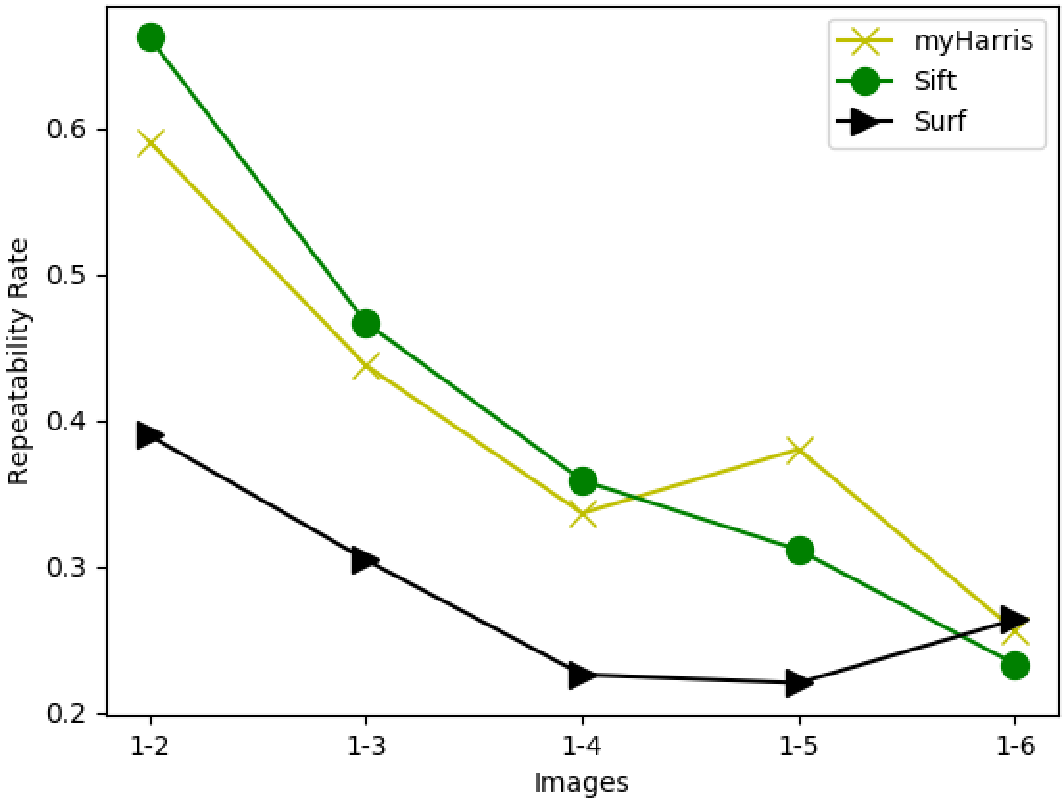


Figure 5: Results for Graffiti Set

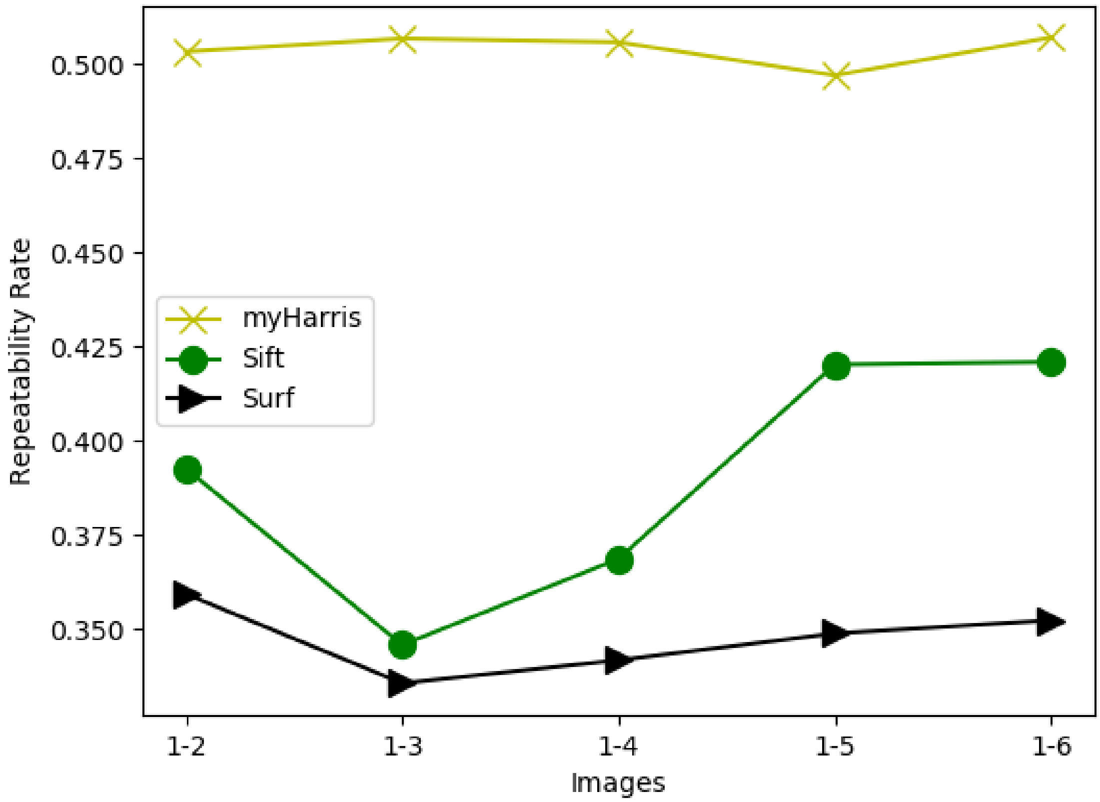


Figure 6: Results for North Campus with Different JPEG quality Set

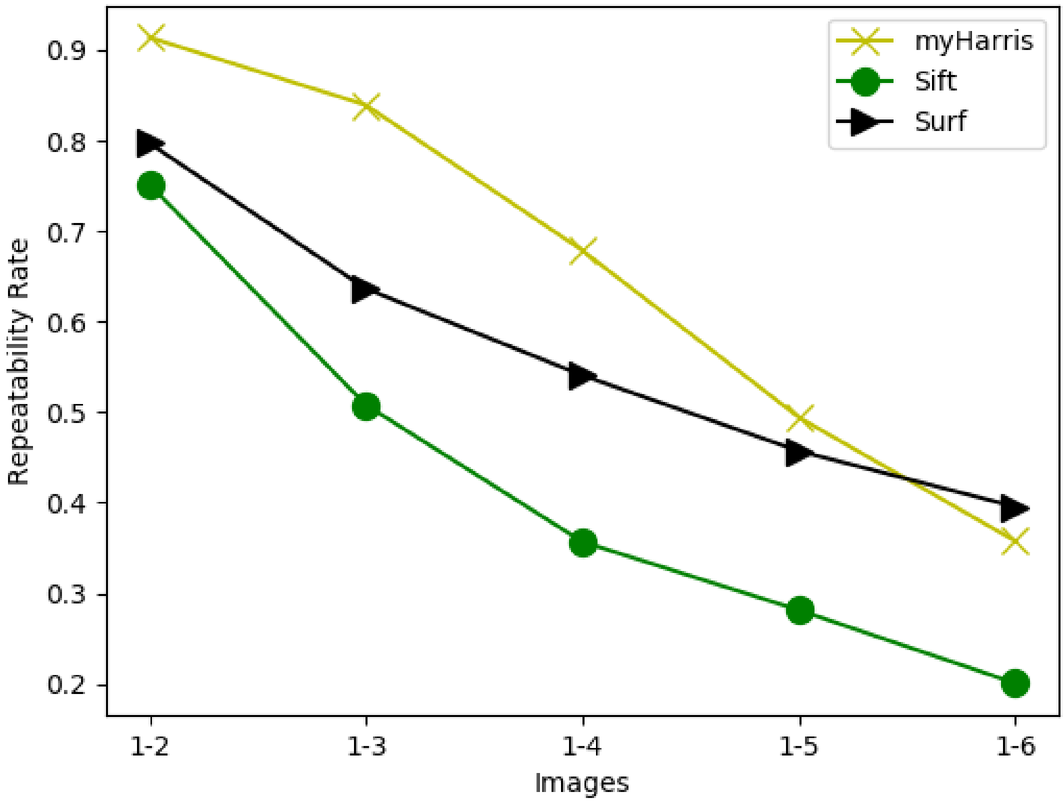


Figure 7: Results for North Campus with Different Levels of Gaussian Noise Set

3.2 Conclusion

As I have observed from results, quality of the image is not important for detectors except SIFT, Gaussian noise affects detectors obviously. From the graffiti set, it shows that with the rotations, based on rotation angle and direction, it differs that which detector is better. If we rotate the image with angle α, SIFT seems to have high repeatability, on the other hand if we move camera in to another place Harris seems more reliable. For the quality, Harris results better than others. And for noise case, Harris is still better most of the time but with increasing noise level, SURF gets better than Harris.