REPORT

Vismaya Veeramanju Kalyan

A20423189

CS 512 FALL-18

Assignment 3

PROBLEM STATEMENT:

Corner detection:

1.Load and display two images containing similar content.

2.Change images into grayscale before processing.

3.In each image perform the following:

Estimate image gradients and apply the Harris corner detection algorithm.

Obtain a better localization of each corner.

Compute a feature vector for each point.

Display the corners by drawing empty rectangles over the original image centered at locations where corners are detected.

4.Number the corresponding feature points with identical number in two images.

The variance of the gaussian(scale), the neighbourhood size for computing the correlation matrix, the weight of the trace in the Harris corner detector and a threshold value must be allowed to control by the user.

PROPOSED SOLUTION:

1. **Harris Corner Detection:**

Compute the correlation matrix for the window.

Variance of the gaussian, window size, weight of the trace, threshold is taken as an input by the user.

Computing Correctness measure and detect corner where the correctness measure is above the threshold.

Formula: $C(c) = \det(c) - K\, tr^2(c)$

c – correlation matrix for the window

K - weight of the trace (usually between 0.04 to 0.15)

$\lambda_1 \lambda_2$ – Are the 2 eigen values of c

$Det(c) = \lambda_1 \lambda_2$

$Tr^2(c) = (\lambda_1 + \lambda_2)^2$

2. **Better Localization:**

To localize the corner points we first find if the window has a corner or not by using harris corner detection.

To determine if point p is the corner, connect each point x_i to point p and project the gradient at x_i onto $(x_i - p)$ the best p will minimize the sum of all the projections.

Formula: $P = C^{-1} \sum \nabla I(x_i) \nabla I(x_i)^T x_i$

C – correlation matrix

$\nabla I(x_i)$ – gradients of x_i

3. **Feature Vectors:**

Sift key points and descriptors are used to get the features for comparison.

The candidate is matched based on Euclidean distance.

BFmatcher is used to compare the points you detected and the once found by the SIFT descriptor.

Only the once that are numbered correspond to points that are identical in the two images.

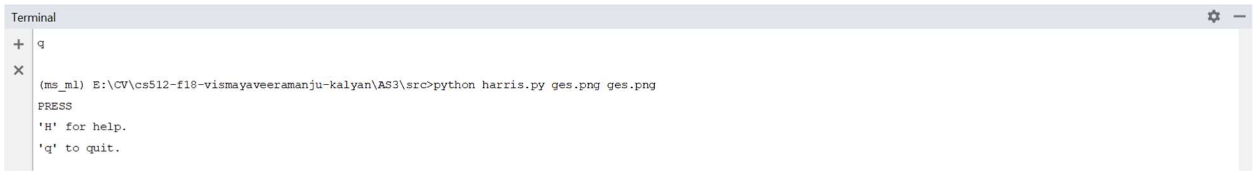
IMPLEMENTATION:

The program has 3 functions to carry out the 3 proposed solutions. The main program gives you the option to select which you want to apply.

Execution of the code:

1. The program can be run without any input in which case the two image is captured from the web camera or you can give any 2 images or a single image twice as the arguments in the command prompt.
   Python harris.py
   or

Python harris.py ges.png ges1.png

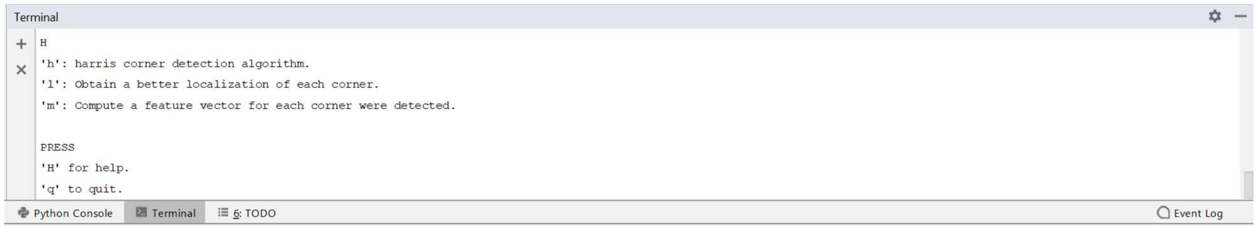


```
Terminal
q

(ms_ml) E:\CV\cs512-f18-vismayaveeramanju-kalyan\AS3\src>python harris.py ges.png ges.png
PRESS
'H' for help.
'q' to quit.
```

2. The programs prompts:
   PRESS
   'H' for help.
   'q' to quit.

   On choosing H your showed with the functions you can apply-
   'h': Harris corner detection algorithm.
   'l': Obtain a better localization of each corner.
   'm': Compute a feature vector for each corner were detected.



```
Terminal
H
'h': harris corner detection algorithm.
'l': Obtain a better localization of each corner.
'm': Compute a feature vector for each corner were detected.

PRESS
'H' for help.
'q' to quit.
```

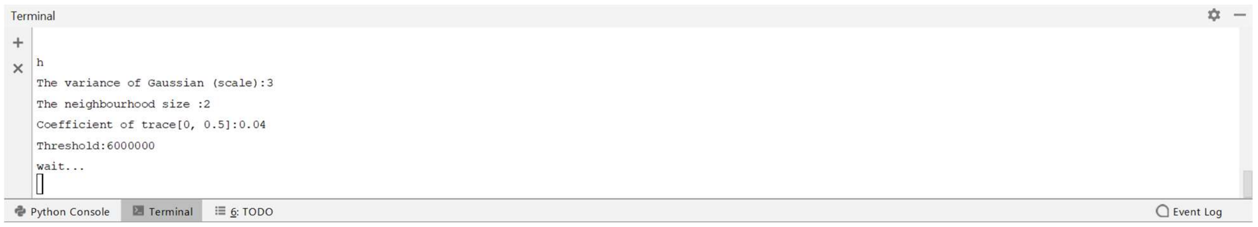
3. Harris corner detection algorithm provides you customizable parameters as stated in the problem statement.

   The variance of Gaussian (scale):3

   The neighbourhood size :3

   The trace in the harris conner detector(k)[0, 0.5]:0.04

   Threshold:6000000



```
Terminal

h
The variance of Gaussian (scale):3
The neighbourhood size :2
Coefficient of trace[0, 0.5]:0.04
Threshold:6000000
wait...
```

   Obtain a better localization of each corner and Compute a feature vector for each corner were detected do not require any parameters.

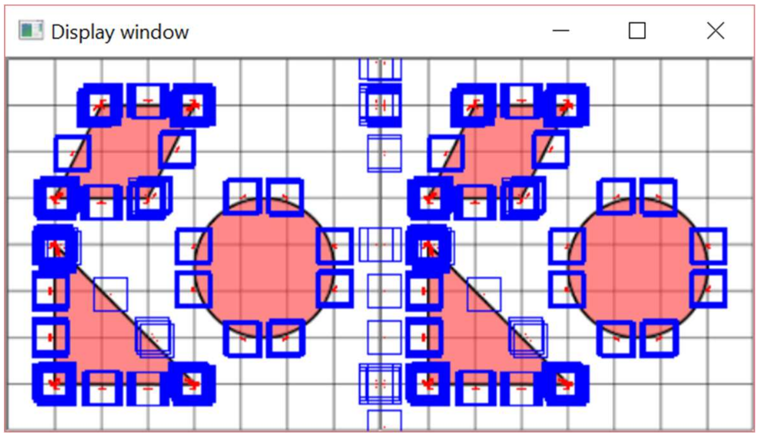
4. Press q to Quit.

Problems:

1. The parameters for Harris Corner detection depends on the image chosen.
2. The matched feature point in the feature vector for each corner is limited to a fixed constant in the program(10 in this case for better visualization) and is not under the control of the user.
3. The input is taken via the keyboard. Trackbar is not implemented and hence doesn't allow continuous change of values.
4. Same dimension image must be given for the feature matching.

RESULTS:

Harris corner detection:



Multiple rectangles can be seen on the same corner as the window draws a rectangle each time a corner is detected in the window. Harris corner detection is used for corner detection but not for corner localizations.

The above image is taken with the

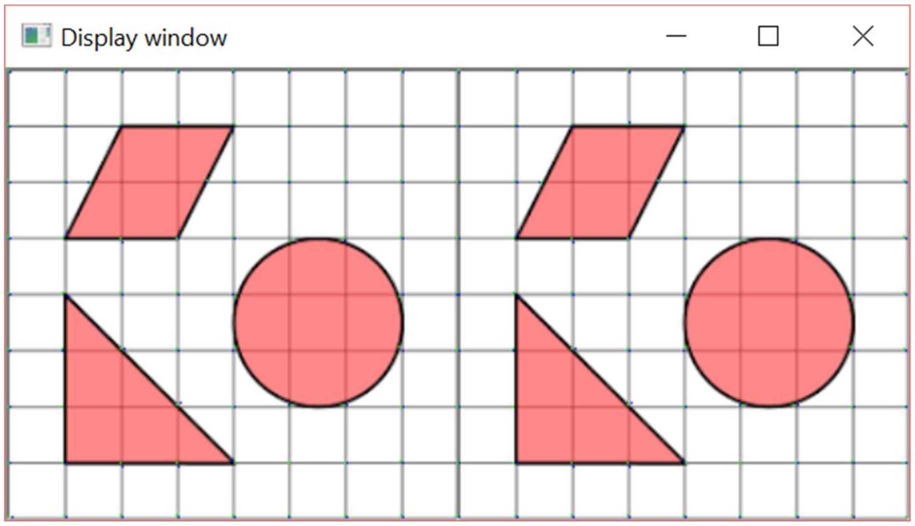
The variance of Gaussian (scale):3

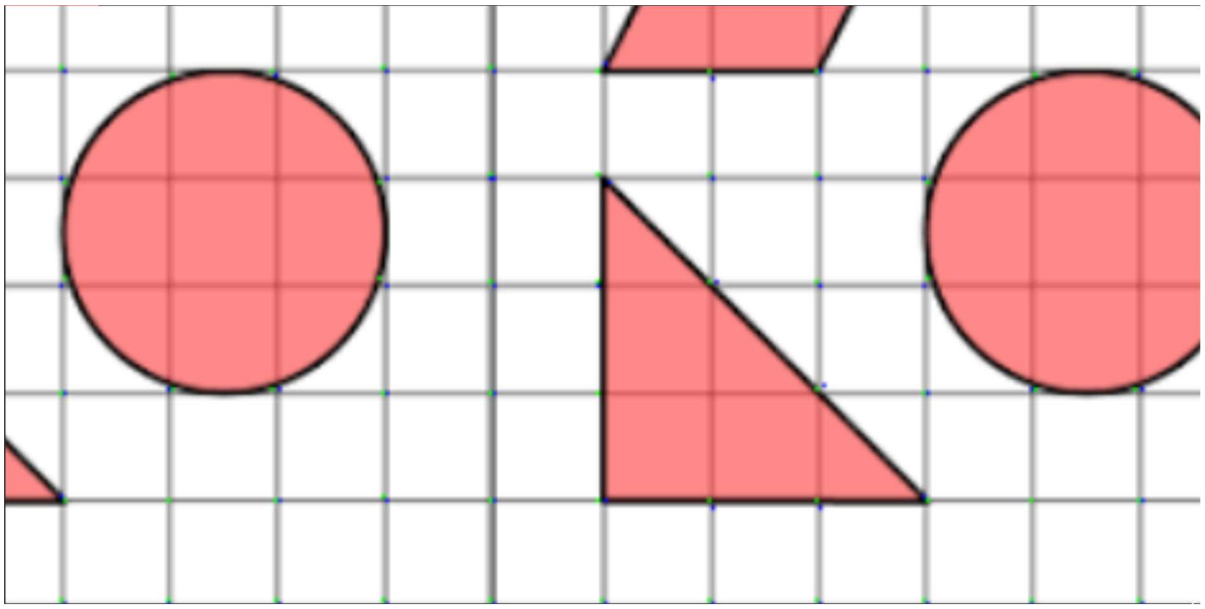
The neighbourhood size :2

The trace in the harris conner detector(k)[0, 0.5]:0.04

Threshold :6000000

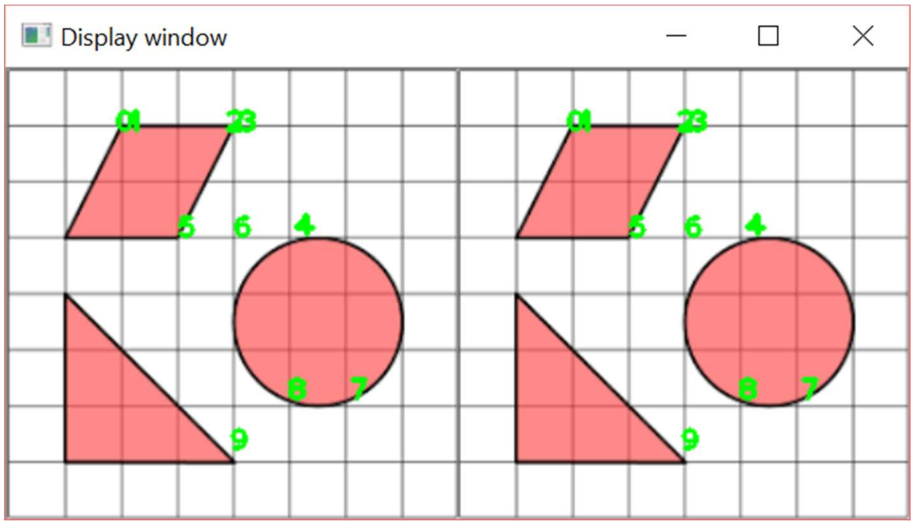
Better Corner Localization:



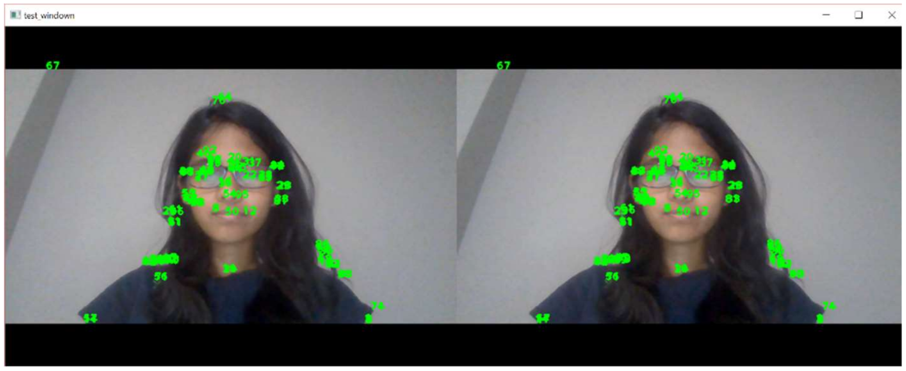


The Green dots can be seen on the exact locations of the corners.

Matched Features:



Only 10 matched features are showed in the image below for better visualization and understanding. We can change the value to match n number of points.



100 matched points using image from the web camera.

REFFERENCE:

https://docs.opencv.org

https://docs.opencv.org/3.4/dc/d0d/tutorial_py_features_harris.html

https://docs.opencv.org/2.4/doc/tutorials/features2d/trackingmotion/corner_subpixeles/corner_subpixeles.html

http://aishack.in/tutorials/harris-corner-detector/

https://docs.opencv.org/3.4/dc/dc3/tutorial_py_matcher.html