CS512 Assignment 3 : Report

Parthkumar Patel
CWID : A20416508
Semester: Fall 2018
Department of Computer Science
Illinois Institute of Technology

October 16,2018

❖Problem statement

a. To write a program to load and display images and perform following operations:
   - To estimate gradients and apply own implementation of Harris corner detection.
   - To get better localization for each corner.
   - To compute feature vector for each corner point
   - To display rectangles over the original image centered around detected corners.

b. Using the feature vectors match the feature points in both the images and number the corresponding points.

c. Introduce trackbar/button to control variance of gaussian,neighbourhood size of correlation matrix, weight of trace in Harris corner detection and a threshold value.

❖Proposed solution

1. Harris corner detection:

$$R = \det(M) - k(\mathrm{trace}(M))^2$$

where

- $\det(M) = \lambda_1 \lambda_2$
- $\mathrm{trace}(M) = \lambda_1 + \lambda_2$
- λ_1 and λ_2 are the eigen values of M

If R > threshold , we mark it a corner.

2. Compute a feature vector for each corner were detected:

a. Find the key points and descriptors with SIFT

SIFT key points of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of key points that agree

on the object and its location, scale, and orientation in the new image are identified to filter out good matches.

b. Use BFmatcher to compare the point you detected and SIFT descriptor finded. And number corresponding points with identical numbers in the two images. Brute-Force matcher is simple. It takes the descriptor of one feature in first set and is matched with all other features in second set using some distance calculation. And the closest one is returned.

3. Better Localization:

Project gradients onto edge hypothesis and choose p with minimal projection.

The location of corner:

$$p = c^{-1}v = c^{-1}\sum \nabla I(p_i)\nabla I(p_i)^t p_i$$

$$c = \sum \nabla I(p_i)\nabla I(p_i)^t$$

c is nonsingular because : $\lambda \cdot \lambda_1 > \tau$

❖Implementation details

Problems faced and solutions:

- The very first was defining the termination criteria for cornerSubPix() which I used **criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.001)** where I used epsilon value as 0.001 and max-iterations 100.
- Also, deciding the initial values of k and threshold is an issue,

Instructions for the program:

Keyboard:

- Press 'q' to quit
- Press ‘H’ for help
- Press ‘h' to Estimate image gradients and apply Harris corner detection algorithm
- Press ‘b’ to Obtain a better localization of each corner
- Press ‘f' to Compute a feature vector for each corner were detected

- Write file name when you want to load image from computer instead off camera (Example: python3 filename.py image1.jpg image2.jpg)

❖Results and discussion

1. Load two picture or capture by webcam

Figure 1: Showing the image captured by camera or image from computer file

2. Include the help key describing its functionality

```
p@parth-Vostro-3558:~$ cd Desktop/CV/cs512-master/AS3/src
p@parth-Vostro-3558:~/Desktop/CV/cs512-master/AS3/src$ python AS3.py 1.jpg 2.jpg
input key to Process image(press 'H' for help, press 'q' to quit):
H
'h': Estimate image gradients and apply Harris corner detection algorithm.
'b': Obtain a better localization of each corner.
'f': Compute a feature vector for each corner were detected.

input key to Process image(press 'H' for help, press 'q' to quit):
```

Figure 2: Showing help menu

3. Take parameter to implement Harris:

```
Press input key to Process image
(press 'H' for help, press 'q' to quit):
h
the varience of Gussian scale(n):3
windowSize :2
the weight of the trace in the harris conner detector(user_input)[0, 0.5]:0.04
threshold:6000
processing...
Press input key to Process image
(press 'H' for help, press 'q' to quit):
```

Figure 3 : Entering parameters for Harris corner detection

4. Getting the result from Harris corner detection

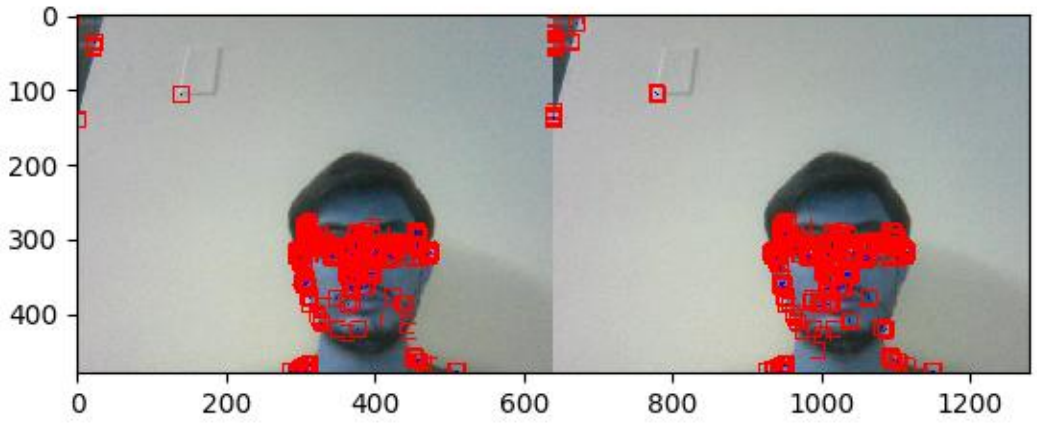

Figure 4 : Output of Harris corner detection on captured image from camera

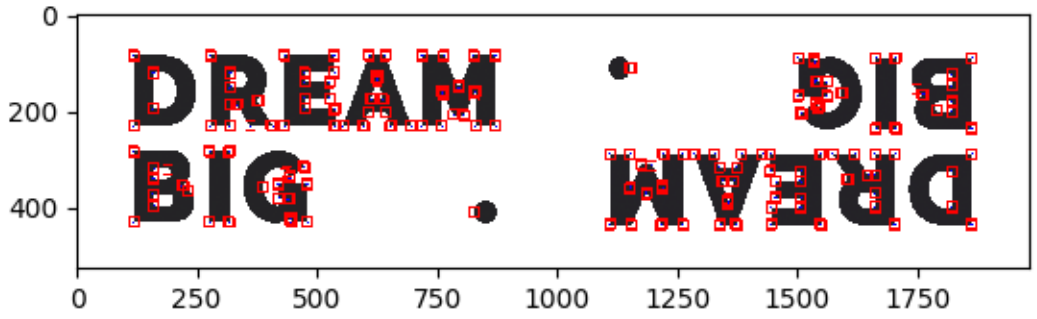

Figure 5 : Output of Harris corner detection on input file from computer

5. Get better location

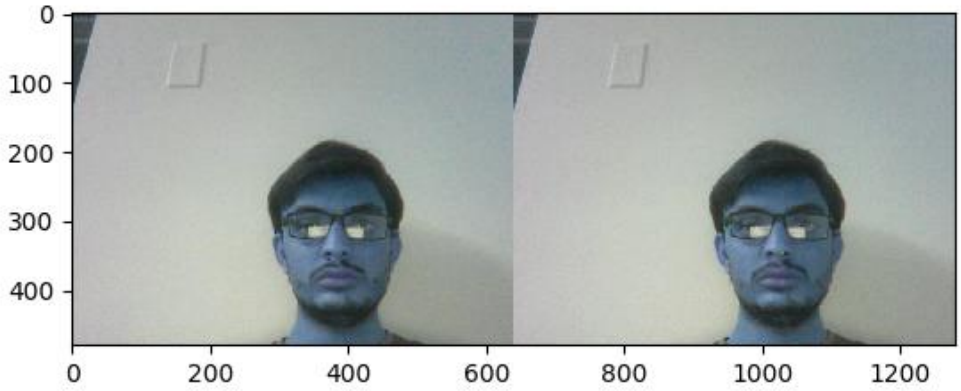

Figure 6 : Output of better localization on captured image from camera

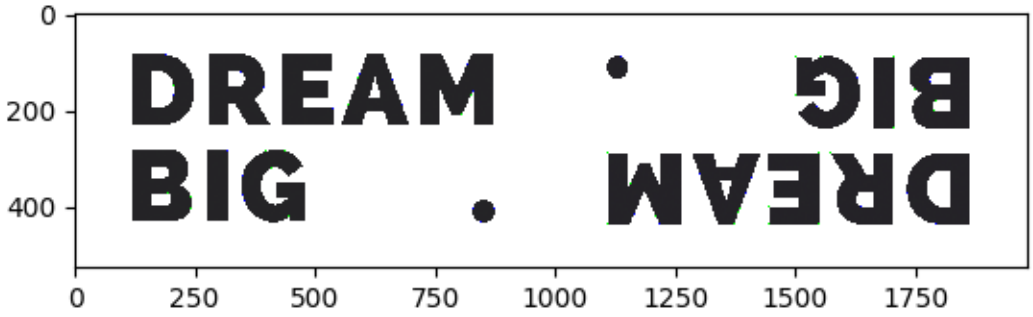

Figure 7: Output of better localization on input image from computer

6. Using the feature vectors, you computed match the feature points. Number corresponding points with identical numbers in the two images. I record first 50 points in each image.

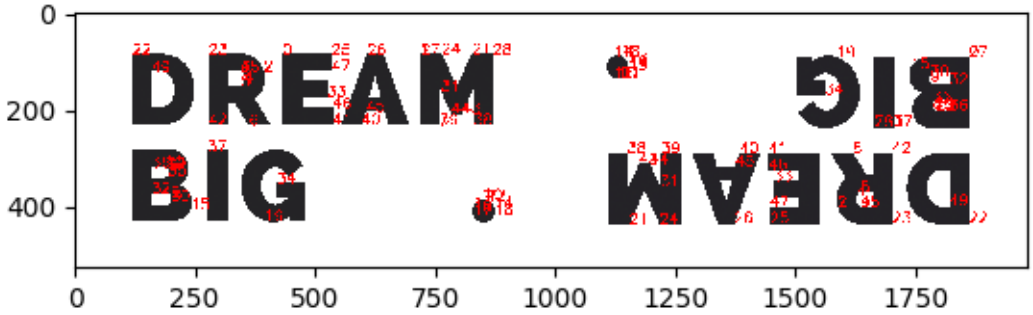

Figure 8 : Output of feature vector on input image from computer

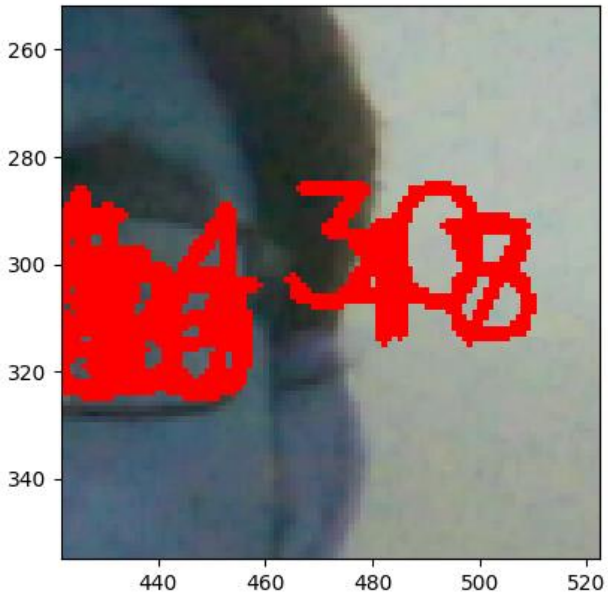

Figure 9 : zoomed image of output of captured image from camera