

# **cs512 Assignment 2 : Program Report**

Pushpendra Singh Chauhan

Department of Computer Science

Illinois Institute of Technology

October 20, 2020

## **Abstract**

This is a report for programming question in cs512 Assignment 2. It will contain the description of the problem I am trying to solve, the algorithms I employed to solve the problem, implementation details like program design issues and the results obtained. The results obtained will be analyzed for correctness. The performance of the algorithm will be evaluated and discussed.

## **1 Problem Statement:**

- Write a program to estimate image gradients and detect corners in an image, obtain better localization of each corner, compute feature vector of each corner point and display the corners by drawing (in color) empty rectangles over the original (grayscale) image centered at locations where corners were detected.
- The image to be processed by the program should be either read from a file or captured directly from a camera. If a filename is specified, the image should be read from it. Otherwise the program should attempt to capture an image from a camera. When capturing an image from the camera, continue to capture and process images continuously.
- The read image should be read as a 3 channel color image.

- The program should work for any size image.

## 2 Proposed Solution:

- Converting the 3 channel color image (RGB) to gray scale using openCV function.
- Used sobel filter for finding image gradients.
- Used Harris Corner Detection algorithm for detecting corners in the image.
- **Harris Corner Detection Algorithm:**
  - (a) Scan image top – bottom, left – right. At each pixel select a neighborhood.
  - (b) Build correlation matrix:  $C = \sum_i g_i g_i^T$ , where  $g_i$  = gradient vector at pixel  $i$  and  $g_i^T$  = transpose of gradient vector at pixel  $i$ .
  - (c) Compute the cornerness measure:  $C(c) = \text{determinant}(c) - k * (\text{trace}(c))^2$ , where  $c$  = correlation matrix,  $C$  = cornerness measure,  $k$  = weight of trace.  $K$  is a user parameter and belongs to  $[0, 0.5]$ .
  - (d) Detect corners where cornerness measure is higher than a threshold. Threshold is also is a user specified parameter.
- **Corner Localization:**
  - (a) Given that there is a corner in a window, find its location. To determine if 'P' is the corner, connect each point ' $X_i$ ' to 'P' and project the gradient at ' $X_i$ ' onto ' $(X_i - P)$ '.
  - (b) The 'best' P will minimize the sum of all the projections. Equation:  $E(P) = \sum_i [(X_i - P)^T (\nabla I(X_i) \nabla I(X_i)^T) (X_i - P)]$ . Here, we have to find P which minimizes  $E(P)$  i.e.  $P^* = \text{argmin}_P E(P)$ . Here, ' $(X_i - P)$ ' = line from P to  $X_i$ ,  $\nabla I(X_i)$  = gradient of image at  $X_i$ ,  $\nabla I(X_i)^T$  = transpose of gradient of image at  $X_i$  and P = point at the corner in the window.
- For feature point characterization, first found the feature vectors of the corner points in both the images. Then, find the difference of feature vector

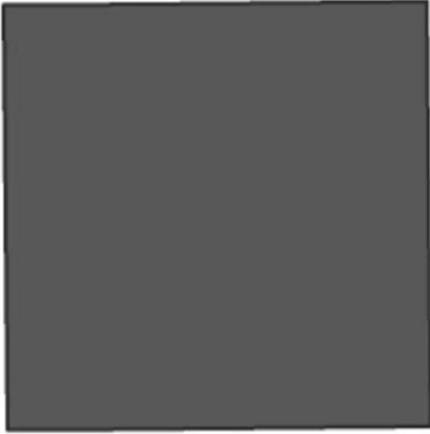
of one corner point in first image with feature vectors of all other corner points of the second image and find the corner point in the second image whose difference is smallest. Label corresponding corner points in both the images with same number.

### 3 Implementation Details:

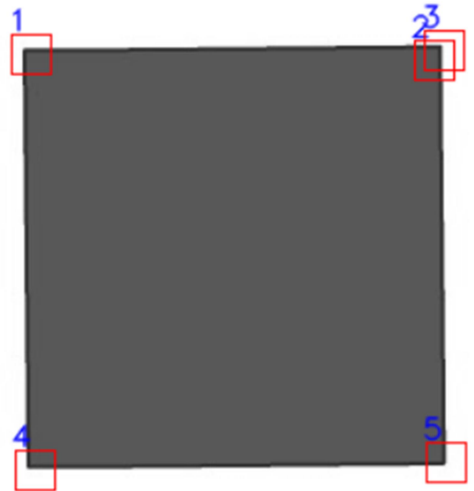
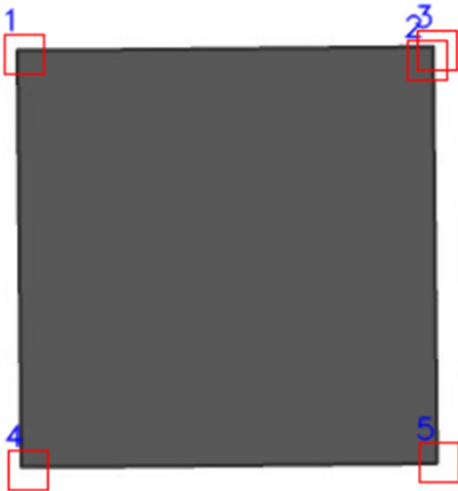
- The main issue I faced in program design is that to control the neighborhood size for computing the correlation matrix, to control the weight of the trace in the Harris corner detector and to control the threshold value which will be used for cornerness measure. I resolved them all using 'createTrackbar' function of openCV.
- Special Keys on the keyboard used to modify the displayed image are as follows:
- 'c' : Perform corner detection on both the images, perform corner localization, feature point characterization and also display the corners by drawing (in color) empty rectangles over the original (grayscale) image centered at locations where corners were detected.
- 'h' : display a short description of the program, its command line arguments and the keys it supports.
- 'exit' : exit the program
- Command Line instructions for using the program: (1). python main.py 'image\_filename1.jpg' 'image\_filename2.jpg' (2). python main.py
- Folder structure: AS2 (main folder) →
  - (1). src (sub-folder) – It contains source code files ('main.py').
  - (2). data (sub-folder) – It contains all test files (images).
  - (3). doc (sub-folder) – It contains two pdf files ('review questions answers.pdf' and 'program report.pdf').

## 4 Results and discussion:

- Original (grayscale) image before applying Harris corner detection algorithm:



- Original (grayscale) image after applying Harris corner detection algorithm:



- The above result is obtained with the following user specified parameter:  
threshold = 10 (This will value will be multiplied by 1000000 inside the code), neighbor (neighborhood size for computing the correlation matrix) = 5 (Window size will become 11X11) and weight\_k (weight of the trace) = 246 (It will be divided by 1000 inside the code).

## 5 References:

- <https://www.geeksforgeeks.org/python-opencv-cv2-puttext-method/>
- [https://docs.opencv.org/master/d7/dfc/group\\_highgui.html#ga122632e9e91b9ec06943472c55d9cda8](https://docs.opencv.org/master/d7/dfc/group_highgui.html#ga122632e9e91b9ec06943472c55d9cda8)
- <https://answers.opencv.org/question/91613/trackbar-with-floating-values/>
- <https://www.geeksforgeeks.org/python-opencv-cv2-rectangle-method/>
- <https://answers.opencv.org/question/175912/how-to-display-multiple-images-in-one-window/>
- [https://en.wikipedia.org/wiki/Corner\\_detection](https://en.wikipedia.org/wiki/Corner_detection)
- [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_feature2d/py\\_features\\_harris/py\\_features\\_harris.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html)
- <http://www.cs.iit.edu/~agam/cs512/share/fdetect2.pdf>