CS663 Assignment 3 Question 1

Shreya Laddha, Archishman Biswas, Shreyan Jabade, Rishabh Arya

October 15, 2020

This document describes the findings regarding question 1 of assignment 3. We have added the relevant comments and output images.

1 Overview

In this question, we have implemented the Harris corner detection algorithm in order to figure out the corners in the boat image.

For the algorithm, three parameters are manually tuned to get a good resultant corner detection. The parameters are Gaussian standard deviation(sig_blur) for input image blurring, Gaussian standard deviation(sig_w) for averaging required in the structure tensor calculation, and finally the Harris corner-ness measure coefficient(K).
Thus the overall code is implemented as a MATLAB function myHarrisCornerDetector(in_image, sig_blur, sig_w, K).

2 Code Implementation

First, we normalize the input image to lie in the range [0,1]. This image is then smoothed using Gaussian blur(sig_blur) to remove noise that might come up as "false Corners" in the output image. Then the partial derivative along X and along Y is taken using imfilter() and appropriate masks. The resultant images are stored in I_x and I_y respectively. This images are displayed

Next, using the above I_x and I_y, we calculate I_x^2, Iy^2 and I_xI_y by point-wise multiplication of he two images. Each of the above three images are then blurred using the sig_w parameter. Using the blurred I_x^2, Iy^2 and I_xI_y, we can define the structure tensor(A) at every pixel location (i,j). Using the eig() function, the eigen value of A(i,j) is obtained and are stored in eigens1, eigens2. We then display these obtained images for the two eigen values.

For every pixel point (i,j), we use the structure tensor A(i,j) to calculate the corner-ness measure(C1), After non-maximum suppressing and thresh-holding of C1, we finally get "corners" image which is 1 at location where a corner is "present", and otherwise it is zero. The function instertMarker() is used to mark the corners in the final image and it is then displayed.

3 Results and Comments

The tuned parameter for the results shown in the diagram below are sig_blur = 0.2, sig_w = 2.5, K = 0.06.

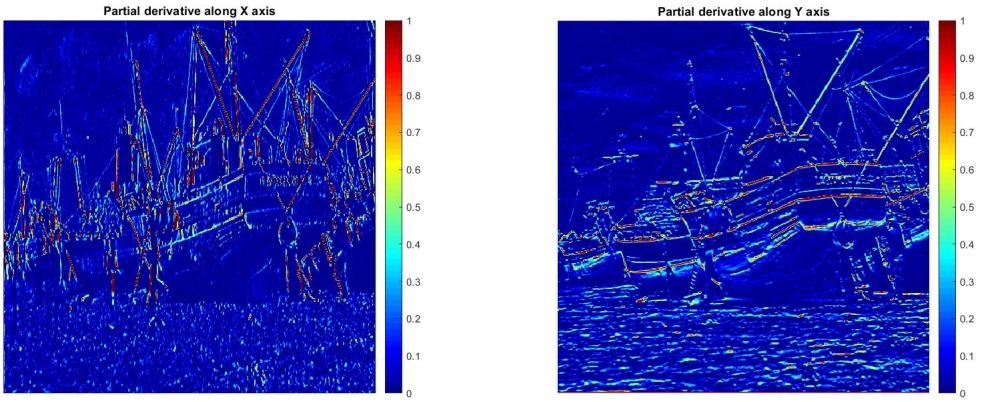

Figure 1: Partial derivatives along the X and Y direction

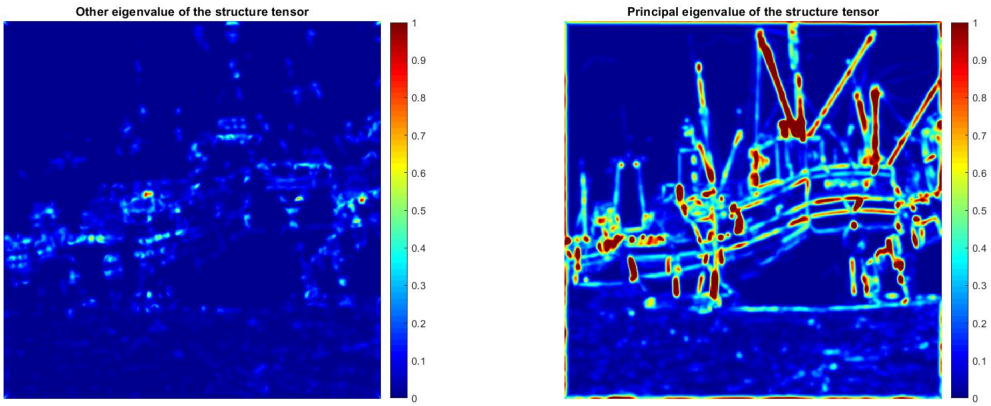

Figure 2: Principle and other eigenvalue image

It can be seen in the above images that the principal eigen value image has high value at both edges and corners, whereas the corners are detected mainly when both the eigen value images have a high value as shown in the below corner-ness measure image.

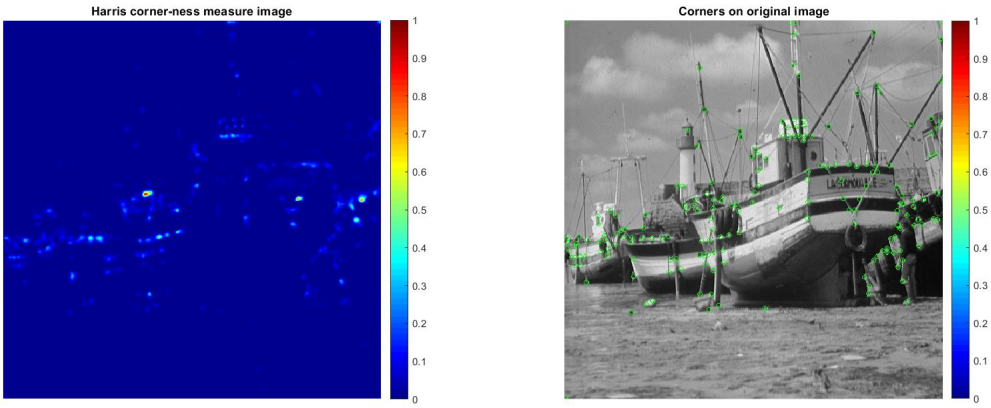

Figure 3: Corner-ness measure and final output image

The corners are denoted using the green circles in the image. Thus we can see that almost all the corner features in the image are detected.

Note that as we increase the sig_blur, the sharp corner features are removed from the original image, thus detected corners will be less. If we increase the sig_w, the spread in eigen values will increase, leading to more corners. For good results, we should have K in [0,0.25], though the preferable range is [0.04,0.06].