CS663, Assignment 3 **Yash Sanghvi, Pradyot Prakash**
Instructor: Prof. Suyash Awate 13D070042, 130050008
Harris Corner Detection

Problem Statement

(30 points) Harris Corner Detection.
Input image: 2/data/boat.mat.
Assume the pixel dimensions to be equal along both axes, i.e., assume an aspect ratio of 1:1 for the axes. Shift and rescale the intensities in the image to lie within the range [0, 1]. Implement the Harris corner detector algorithm. The parameters underlying this algorithm are: two Gaussian smoothing levels involved in computing the structure tensor, the constant k in the corner-ness measure. Tune these three parameters to get the best results.

1. Write a function myHarrisCornerDetector.m to implement this.
2. Display the derivative images, corresponding to the derivatives along the X and Y axes.
3. Display the images (along with a colormap) of the two eigenvalues of the structure tensor, evaluated at each pixel.
4. Display the image (along with a colormap) of the Harris corner ness measure. Positive values in this image must correspond to a corner structure in the image.
5. Report all three parameter values used.

Code

Harris Corner Detection Code

```
function [outputImage, H, Ix, Iy, A,B,C] =
   myHarrisCornerDetection(inputImage, sigma1, sigma2, k)

        W = 5;

        h = fspecial('gaussian', [2, 2], sigma1);
        inputImage = imfilter(inputImage, h);
    FilterX = [-1 0 1;-2 0 2;-1 0 1];
    FilterY = FilterX';
    Ix = imfilter(inputImage,FilterX);
    Iy = imfilter(inputImage,FilterY);



        Ix2 = Ix .^ 2;
        Ixy = Ix .* Iy;
        Iy2 = Iy .^ 2;
```

```
        h = fspecial('gaussian', [W, W], sigma2);

        A = imfilter(Ix2, h);
        B = imfilter(Ixy, h);
        C = imfilter(Iy2, h);

        M1 = (A .* C) - B .^ 2;
        M2 = (A + C) .^ 2;
        H = M1 - (k * M2);

        % imregional max used for non-maximal supression and the product for the cornerness measure
        corners = imregionalmax(H) .* (H > 0.1);
        outputImage = corners;
end
```

Main Script

```
%% MyMainScript

tic;
%% Your code here
boat_struct = load('../data/boat');
boat = myLinearContrastStretching(boat_struct.imageOrig);
[corners, h, Ix, Iy, A, B, C] = myHarrisCornerDetector(boat, 0.66, 1, 0.01);
imshow(boat, []);
hold on;
[row, col] = find(corners);
plot(row, col, 'r*');
figure(2)
subplot(121)
imshow(Ix)
title('I_x: X Derivative of Image');

subplot(122)
imshow(Iy)
title('I_y: Y Derivative of Image');

[eig1, eig2] = myFindEigenValue(A,B,C);
figure(3)
subplot(121)
imshow(eig1)
title('EigenValue Images(I)');
subplot(122)
imshow(eig2)
title('EigenValue Images (II)');

```

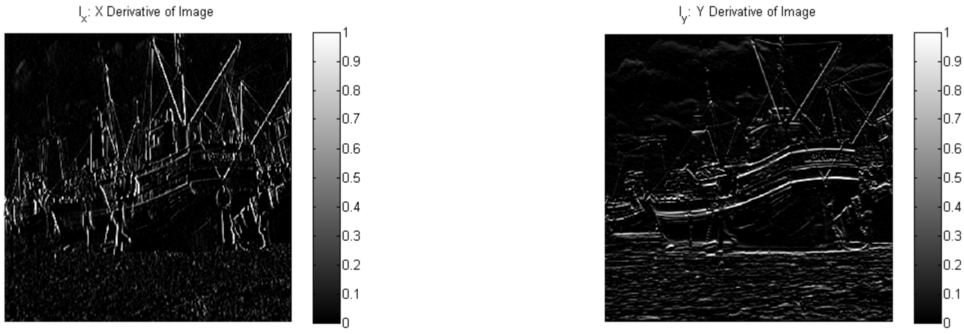
```
toc;
```

Implementation Details

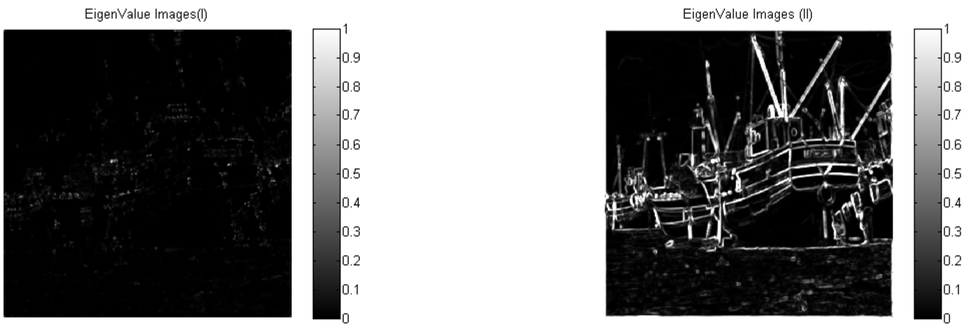
Firstly the image was smoothed with a Gaussian filter of size [W, W] and variance σ_{smooth}. Then the X and Y Derivatives of the image (I_x and I_y) were calculated by filtering the image with the Sobel operators for X and Y axis respectively.
For boundary conditions i.e. the points where mask lied outside the image, the rest of the image pixels were considered zero (Dirichlet boundary conditions). Then I_x^2, I_y^2, I_{xy} images were calculated using the derivatives and another Gaussian window of size [W1, W1] and variance $\sigma_{autocorrelation}$ was applied to these images. Dirichlet boundary conditions as explained above were used. Next for each pixel, cornerness was measured:
Cornerness = $\lambda_1 \lambda 2 - k(\lambda 1 + \lambda 2)^2$
Cornerness = Det(A) - k(Trace)^2
Cornerness = $(I_x^2 I_y^2 - I_{xy}^2) - k(I_x^2 + I_y^2)^2$
After thresholding the cornerness image with parameter ϵ, and non maximum suppression, we get the resulting corners as shown in the next section.

Result Images

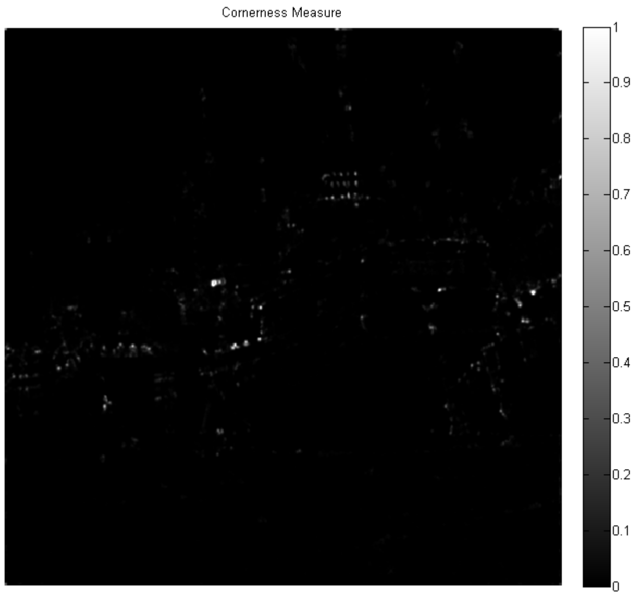
Final Corners in the image



Derivative of Image along X and Y Axis respectively



Eigenvalues of structure tensor



Cornerness Measure

Optimum Parameters

Gaussian Mask for Smoothing the image: Window Size: [5 5], Variance = 0.66
Gaussian Mask for Calculating I_x^2, I_{xy}, I_y^2: Window Size: [5 5], Variance = 1
k used for Cornerness Measure: 0.01
ϵ**i.e. Theshold Parameter**: 0.1