# Question:

Determine the edge maps of noise free and noisy images using different gradient based edge detection techniques and compare their performances. In the first part, obtain the edge maps of noise free images using different Gradient based techniques i.e. Roberts, Prewitt and Sobel operators for different thresholds. In the second part, compare the performances of Gradient based edge detectors for noisy images of different SNRs (SNR=20, 10, 5dB). Write Matlab code and display your results in each case. Also, count the number of edges missed or false edges are found by comparing noiseless and noisy images in each individual case. Make use of the image illustrated in Figure 2.



## **Code:**

img = imread('Fig2.jpg');

imc = rgb2gray(img);

figure(1)

subplot(2,2,1)

imshow(imc);

J = double(imc)/255;

v = var(J(:));

SNR = [5 10 20];

v1 = v/(10^(SNR(1)/10));

v2 = v/(10^(SNR(2)/10));

v3 = v/(10^(SNR(3)/10));

title('Normal');

%edge detection of clear image

edgesc = edge(imc,'sobel');

subplot(2,2,2)

imshow(edgesc)

title('sobel');

edgerc = edge(imc,'Roberts');

subplot(2,2,3)

imshow(edgerc)

title('Roberts');

edgepc = edge(imc,'prewitt');

subplot(2,2,4)

imshow(edgepc)

title('Prewitt')

%

%add noise

%5dB

imn = imnoise(imc,'gaussian',0,v1);

figure(2)

subplot(2,2,1)

imshow(imn)

title('With 5dB noise');

edges = edge(imn,'sobel');

subplot(2,2,2)

imshow(edges)

title('sobel');

edger = edge(imn,'Roberts');

subplot(2,2,3)

imshow(edger)

title('Roberts');

edgep = edge(imn,'Prewitt');

subplot(2,2,4)

imshow(edgep)

title('Prewitt')

%10dB

imn1 = imnoise(imc,'gaussian',0,v2);

figure(3)

subplot(2,2,1)

imshow(imn1)

title('With 10dB noise');

edges1 = edge(imn1,'sobel');

subplot(2,2,2)

imshow(edges1)

title('sobel');

edger1 = edge(imn1,'Roberts');

subplot(2,2,3)

imshow(edger1)

title('Roberts');

edgep1 = edge(imn1,'Prewitt');

subplot(2,2,4)

imshow(edgep1)

title('Prewitt')

%20dB

imn2 = imnoise(imc,'gaussian',0,v3);

figure(4)

subplot(2,2,1)

imshow(imn2)

title('With 20dB noise');

edges2 = edge(imn2,'sobel');

subplot(2,2,2)

imshow(edges2)

title('sobel');

edger2 = edge(imn2,'Roberts');

subplot(2,2,3)

imshow(edger2)

title('Roberts');

edgep2 = edge(imn2,'Prewitt');

subplot(2,2,4)

imshow(edgep2)

title('Prewitt')

%Counting number of edges

%Clear

a = nnz(edgesc)

b = nnz(edgerc)

c = nnz(edgepc)

%snr case 1

a1 = nnz(edges)

b1 = nnz(edger)

c1 = nnz(edgep)

%snr case 2

a2 = nnz(edges1)

b2 = nnz(edger1)

c2 = nnz(edgep1)

%snr case 3

a3 = nnz(edges2)

b3 = nnz(edger2)

c3 = nnz(edgep2)

# Output:









# Number of Edges

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Sobel** | **Roberts** | **Prewitt** |
| Clear | 15744 | 6416 | 14865 |
| 5 dB SNR | 9317 | 1771 | 9092 |
| 10 dB SNR | 9361 | 1763 | 9176 |
| 20 dB SNR | 9228 | 1700 | 9044 |

# **Conclusion:**

We observe the following:

1. Number of edges counted in each of the pictures with decreases with the increasing SNR of the noisy images. This is because the type of noise we are using here is “Gaussian Noise” which generates white pixels only as the noise component, hence decreasing the number of effective edges.
2. When we repeated the same experiment with noise type “salt & pepper” we got an increase in edges since it also adds black pixels as the noise component.
3. Edge detection technique of “Sobel” gives the best result followed by “Prewitt” and the finally “Robert”.