Project in Multidimensional Signal Processing

Shirdi Manjunath Adigarla (950608-4970)

Last Updated: 08 October 2021

# Image Restoration and Enhancement:

The main aim of this project is to restore and enhance (as much as possible) an image which has been distorted in an imaging system with imperfect lens and CCD. Restoration is objective while enhancement is subjective. Enhancement is to make image more appealing. Restoration recovers image which is corrupted by noise and not even be viewable unlike enhancing where we try to increase image contrast or details which doesn’t exist to make it more human viewable.

In this project to restore image we go in reverse order based on model of distortion of the image. First, we remove noise in image then remove blurring and finally wrapping distortion.

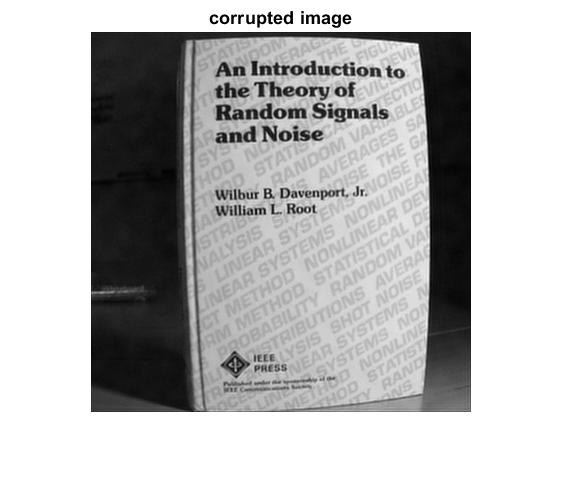


Figure 1: Corrupted image

## Noise Removal:

From Noise spectrum 4 we can observe that noise is more in lower frequencies and less in higher frequencies. Based on histogram and noise spectrum we can conclude that it is similar to gaussian shape, so it is gaussian noise. So we design a high pass filter (boosts pixels which are brighter than neighbour pixels based on differentiation.) to remove noise in lower frequencies. Usually noise tend to be in higher frequencies and image details are tending to be in lower frequencies. By using high pass filter to remove noise we also tend to increase more noise in higher frequencies. So, to remove it we use wiener filter which is best for removal of gaussian noise. Wiener filter adaptively removes noise. If variance is large it performs little smoothing and if it is low high smoothing. Wiener is best for additive white gaussian noise.

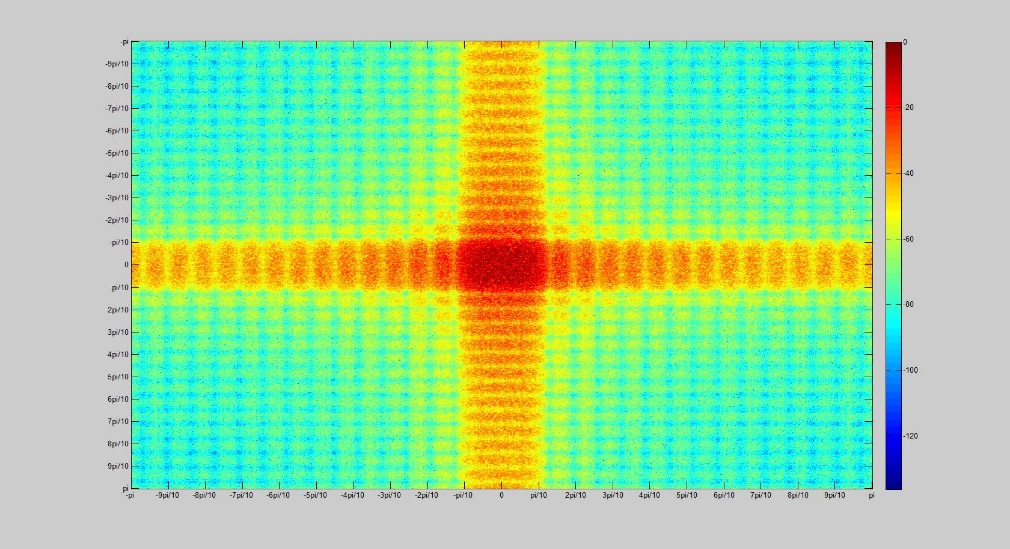


Figure 1: Noise Spectrum 4

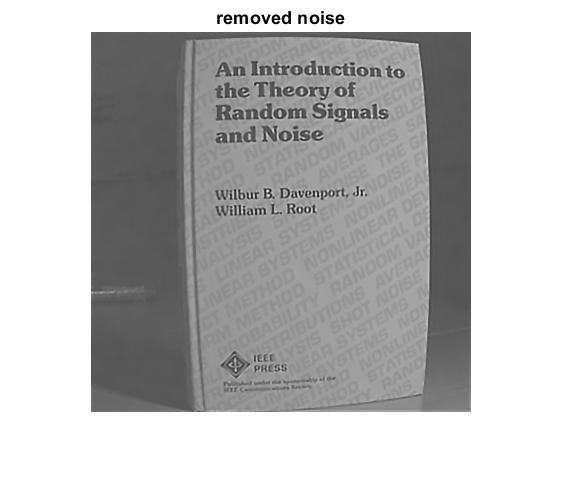


Figure 2: Noise removal image

## Deblurring:

Noise is simply additive in nature while in deblurring we have degrading function which is convolved with image to get corrupted image. The image is corrupted with point spread function in spatial domain with convolution. Based on convolution property it is multiplication in frequency domain. We can use inverse property in frequency domain to get original image based on point spread function. Blurring is usually caused by low pass filter performing more smoothing. Based on blurring Spectrum 4 which has radius around 0.7 we recreate the blurring spectrum and perform inverse operation to corrupted image to get original image.

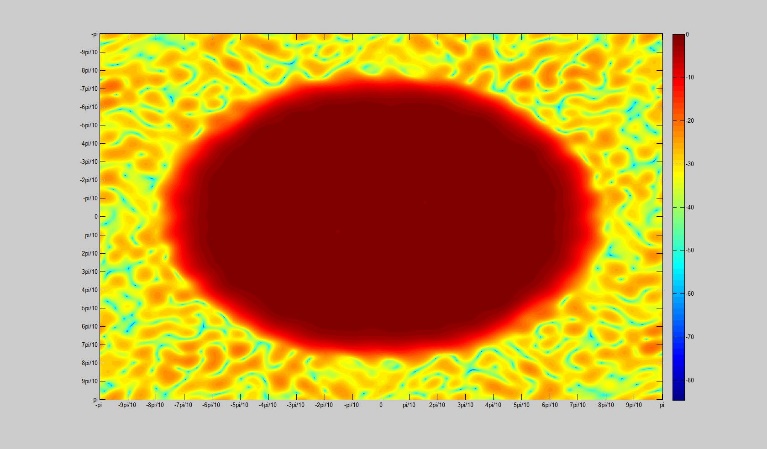


Figure 2: Blurring Spectrum 4

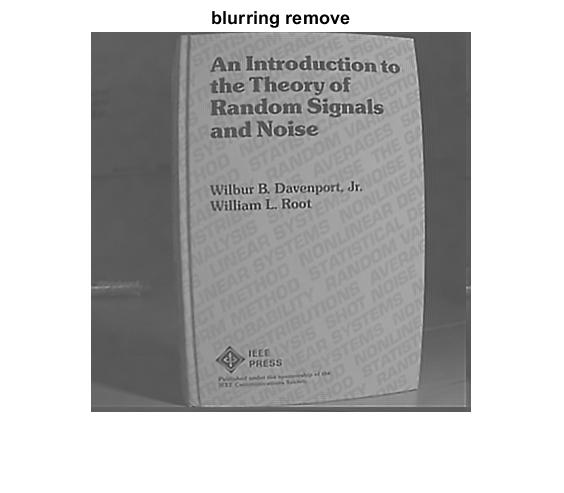
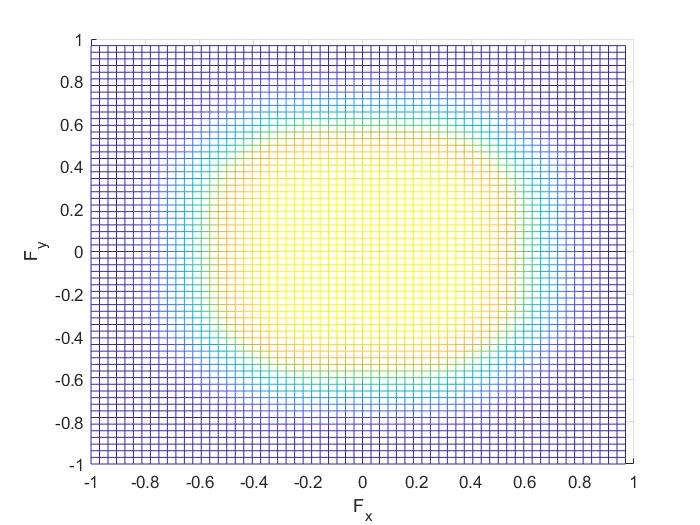


Figure 2: Blurring Spectrum 4

## Unwrapping:

Finally, we remove wrapping distortion of image using polar coordinates and transforming given image to get our final image without distortion.

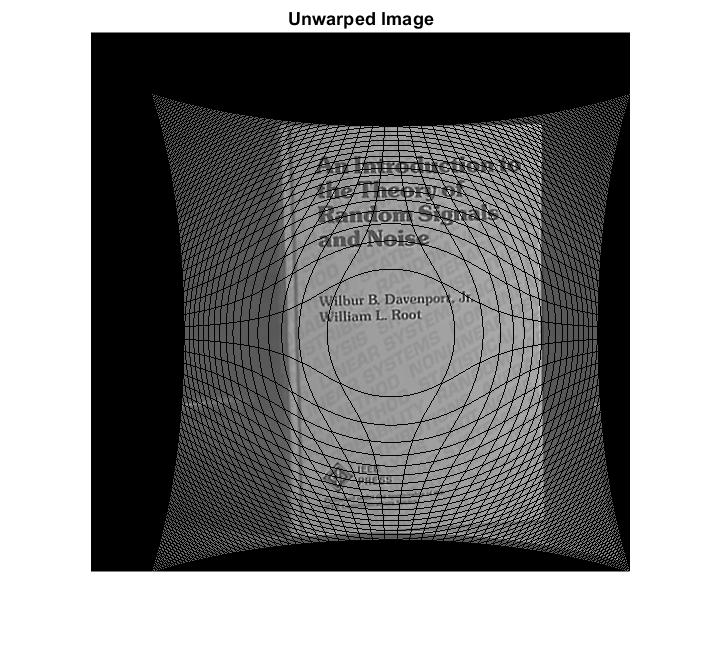


Figure 2: Unwrapped image

# Appendices:

## MATLAB Code

clc;

clear all;

close all;

load('Image4.mat'); % loading data of original image

I=IMAGE;

I=mat2gray(I);

imshow(I) % original image

title('corrupted image')

%% histogram

imhist(I)

%% noise remove

h1 = [0 -1 0; -1 5 -1; 0 -1 0]; % High Pass Filter\_1

I1 = mat2gray(filter2(h1,I));% Filtering the Image through High pass filter\_1

I1=wiener2(I1);

figure

a=imshow(mat2gray(I1));

title('removed noise')

% h=fspecial('gaussian',[7,7]);

% I1=imfilter(I,h);

% a=imshow(mat2gray(I1))

%% Deblurring

N=11;

[z1,z2]=freqspace(64); %two-dimensional frequency vectors f1 and f2 for an 64-by-64 matrix.

[c,d]=meshgrid(z1,z2); % representing on a grid

H=zeros(size(c)); % generating a zeros matrix of the size w1

r=sqrt(c.^2+d.^2);

d=find(r<0.7); %finding the values which satisfy the condition

H(d)=ones(size(d)); % generating a ones matrix of the size d

h1=fwind1(H,hamming(N),hamming(N));

figure(4)

freqz2(h1)

B=inverseFilter(I1,h1,1);

DB2=mat2gray(B);

figure(5)

imshow(DB2)

title('blurring remove')

%% Unwarping

W=zeros(480,480);

R0=294;

for i=1:380

for j=1:380

x\_ = j - 190.5;

y\_ = 190.5 - i;

%taking absolute values of x and y

x = abs(x\_);

y = abs(y\_);

% determining polar coordinates

t = abs(atan(y/x));

r\_ = sqrt(x^2+y^2);

r = abs(R0.\*(asin((r\_/R0)))); %given condition

%Converting Polar Coordinates (r,theta) to Cartesian Coordinates (x,y)

x = r.\*cos(t) ;

y = r.\*sin(t) ;

if x\_ < 0

x = -x;

end

if y\_ < 0

y = -y;

end

%towards positive infinity

i\_= (ceil(300- y));

j\_ = (ceil(x +300));

W(i\_,j\_) = DB2(i,j);

end

end

figure

imshow(mat2gray(W)); %unwarping image

title('Unwarped Image')

% W1=medfilt2(W);

% figure

% imshow(W1);

% title('unwarping first level')

% W2=medfilt2(W1);

% figure;

% imshow(W2);

% title('unwarping second level')