

Exp-5: Frequency Domain filtering of image.

a. Lowpass (smoothing) filter in frequency domain

```
% Matlab Program to demonstrate the "Low pass Filtering of an image
using
% 2D-DFT"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Algorithm for filtering in the frequency Domain
% Step1: Given an input image f(x,y) of size M x N, obtain the padding
% parameters P and Q. Typically, we select P = 2M and Q = 2N
% Step2: Form a padded image fp(x,y) of size P X Q by appending the
% necessary number of zeros to f(x,y).
% Step3: Multiply fp(x,y) by (-1)^(x+y)
% Step4: Compute the DFT, F(u,v) of the image from Step 3
% Step5: Generate a Real, Symmetric Filter Function H(u,v) of size P X Q
% with center at coordinates (P/2,Q/2),
% Step 6: Form the product G(u,v) = H(u,v)F(u,v) using array
multiplication
% Obtain the processed image
% Step 7: gp(x,y) = {real{inverse DFT[G(u,v)]}}(-1)^(x+y)
% Step 8: Obtain the final processed result g(x,y) by extracting the M X N
region
% from the top, left quadrant of gp(x,y)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Algorithm Implementation
% Step1: Given an input image f(x,y) of size M x N, obtain the padding
% parameters P and Q. Typically, we select P = 2M and Q = 2N

clc;
clear all;
close all;

Im = imread('cameraman.tif');
%a=imnoise(Im,'salt & pepper', 0.02);    % Adding High Frequency Noise
%a=imnoise(Im,'speckle', 0.08);
a=imnoise(Im,'gaussian', 0, 0.04);
```

```

% Converting the image class into "double"
b = im2double(a);
% reading the image size
[m,n] = size(b);
% creating a null array of size 2m X 2n
c = zeros(2*m,2*n);
% reading the size of the null array
[p,q] = size(c);

% Step 2
% appending the original image with the null array to create a padding
% image hence it is Step 2
for i = 1:p
    for j = 1:q
        if i <= m && j <= n
            c(i,j) = b(i,j);
        else
            c(i,j) = 0;
        end
    end
end
imshow(b);title('input noisy image');
figure;
imshow(c);title('padded image');

% Step 3
% creating a null array of size p X q
d = zeros(p,q);
% Multiplying the padded image with  $(-1)^{(x+y)}$ 
for i = 1:p
    for j = 1:q
        d(i,j) = c(i,j).*(-1).^(i + j);
    end
end
figure;
imshow(d);title('pre processed image for calculating DFT');

% Step 4
% Computing the 2D DFT using "fft2" matlab command
e = fft2(d);
figure;imshow(e);title('2D DFT of the pre processed image');

```

```

%%%%%%%%%%%%%%
% Step 5
% Generating the Real, Symmetric Filter Function
% Here we will implement a "Low Pass Filter" using "freqspace" matlab
% command
[x,y] = freqspace(p,'meshgrid');
z = zeros(p,q);
for i = 1:p
    for j = 1:q
        z(i,j) = sqrt(x(i,j).^2 + y(i,j).^2);
    end
end
% Choosing the Cut off Frequency and hence defining the low pass filter
% mask
H = zeros(p,q);
for i = 1:p
    for j = 1:q
        if z(i,j) <= 0.4 % here 0.4 is the cut-off frequency of the LPF
            H(i,j) = 1;
        else
            H(i,j) = 0;
        end
    end
end
figure;imshow(H);title('Low Pass Filter Mask');

%%%%%%%%%%%%%%
% Step 6:Form the product  $G(u,v) = H(u,v)F(u,v)$  using array
multiplication
% Obtain the processed image
% from the previous program lines we know that,
% e : the 2D DFT output of pre processed image
% H : the mask for Low Pass Filter
% let out is the variable
h1 = e.*H;
figure;
imshow(h1);title('Low passed output');

% Step 7:  $gp(x,y) = \{\text{real}\{\text{inverse DFT}[G(u,v)]\}(-1)^{(x+y)}$ 
% calculation of inverse 2D DFT of the "out"
h2 = ifft2(h1);

```

```

figure;
imshow(h2);title('output image after inverse 2D DFT');
% post process operation
h3 = zeros(p,q);
for i = 1:p
    for j = 1:q
        h3(i,j) = h2(i,j).*((-1).^(i+j));
    end
end
figure;
imshow(h3);title('Post Processed image');

% Step 8: Obtain the final processed result g(x,y) by extracting the M X N
region
% from the top, left quadrant of gp(x,y)
% let the smoothed image or low pass filtered image is "out"
out = zeros(m,n);
for i = 1:m
    for j = 1:n
        out(i,j) = h3(i,j);
    end
end
figure;
imshow([b out]);title('input image          output image');

```