Text

Description automatically generated

**Department of Computer Science & Engineering**

Lab :04

Course title: Digital Image Processing

Course Code: CSE-438

*Student Information*

Name: Aahadul Islam Fardin

ID: 2019-1-60-224

Section: 02

Date: 01 April 2023

***(Show input and output side by side for all problems****)*

1. Apply Fourier transform to transform an image from spatial domain to frequency domain. Apply inverse Fourier transform to transform the image from frequency domain to spatial domain.

Code:

>> A = imread(‘MRI.png’);

>> imshow(A)

>> I1 = double(mat2gray(A));

>> I2 = fftshift(fft2(I1));

>> imshow(I2)

>> imshow(A), figure();

>> subplot(2,2,1);

>> imshow(A);

>> subplot(2,2,1);

>> figure();

>> subplot(2,2,1);

>> imshow(A);

>> title('original image');

>> subplot (2,2,2);

>> imshow(I1);

>> title('Done')

>> title('S to F transformation Image')

>> subplot(2,2,3);

>> imshow(I2);

>> title("FFT Image")

>> subplot(2,2,4);

>> imgFFT = fft2(double(A));

>> img2 = ifft2(imgFFT);

>> subplot(2,2,4);

>> imshow(imgFFT);

>> title('Inverse FFT image');

Output:

Qr code

Description automatically generated

1. Compress the image using Fourier Transform.

Code:

>> grayIm =rgb2gray(A);

[row col] = size(grayIm);

subplot(2, 2, 1);

imshow(grayIm);

title('original image')

X=fft2(grayIm); %2D fft

count\_pic=2;

for thresh=0.1\*[0.001 0.005 0.006]\*max(max(abs(X)))

ind=abs(X)>thresh;

count=row\*col-sum(sum(ind));

Alow=X.\*ind;

per=100-count/(row\*col)\*100;

Blow=uint8(ifft2(Alow));

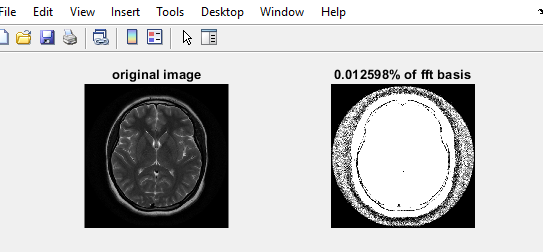
subplot(2,2,count\_pic);

imshow(Blow);

count\_pic=count\_pic+1;

title([num2str(per) '% of fft basis'])

Output:



1. Apply discrete cosine transform to transform an image from spatial domain to frequency domain. Apply inverse discrete cosine transform to transform the image from frequency domain to spatial domain.

Code:

image = imread('MRI.png');

image = im2gray(image);

compress = dct2(image);

imshow(log(abs(compress)),[])

compress(abs(compress) < 10) = 0;

K = idct2(compress);

K = rescale(K);

figure,montage({image,K});title('Original Image After DCT ');

inverseDct=abs(idct2(image))

figure,imshow(inverseDct);title('Inverse DCT Trasnform');

Output:

Graphical user interface

Description automatically generated

**Question 4**

Compress the image using Discrete Cosine Transform.

Code:

a=imread('MRI.png');

image = im2gray(a);

compress = dct2(image);

compress(abs(compress) < 10) = 0;

K = idct2(compress);

K = rescale(K);

subplot(221);imshow(a); title('Original image');

subplot(222);imshow(K); title('DCT Compress image');

Output:

Graphical user interface, text

Description automatically generated

**Question 5**

Compress the image using Haar Transform.

Code:

image\_name = 'MRI.png';

delta = 0.0001;

disp(delta)

if (delta>1 || delta<0)

error('harr\_wt: Delta must be a value between 0 and 1');

end

H1=[0.5 0 0 0 0.5 0 0 0;0.5 0 0 0 -0.5 0 0 0;0 0.5 0 0 0 0.5 0 0 ;0 0.5 0 0 0 -0.5 0 0 ;0 0 0.5 0 0 0 0.5 0;0 0 0.5 0 0 0 -0.5 0;0 0 0 0.5 0 0 0 0.5;0 0 0 0.5 0 0 0 -0.5;];

H2=[0.5 0 0.5 0 0 0 0 0;0.5 0 -0.5 0 0 0 0 0;0 0.5 0 0.5 0 0 0 0;0 0.5 0 -0.5 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1;];

H3=[0.5 0.5 0 0 0 0 0 0;0.5 -0.5 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 0 0 1 0 0 0 0;0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1;];

H1o = (H1.\*(2^0.5));

H2o = (H2.\*(2^0.5));

H3o = (H3.\*(2^0.5));

Ho=normc(H1o\*H2o\*H3o);

H = H1\*H2\*H3;

x=double(imread(image\_name));

len=length(size(x));

if len~=2

error('harr\_wt: Input image must be a grey image, use "haar\_wt\_rgb" function to compress RGB Images');

end

yo = zeros(size(x));

y = zeros(size(x));

[r,c]=size(x);

for i=0:8:r-8

for j=0:8:c-8

p=i+1;

q=j+1;

yo(p:p+7,q:q+7)=(Ho')\*x(p:p+7,q:q+7)\*Ho;

y(p:p+7,q:q+7)=(H')\*x(p:p+7,q:q+7)\*H;

end

end

figure;

imshow(x/255);

n1=nnz(y);

zo=yo;

m=max(max(yo));

yo=yo/m;

yo(abs(yo)<delta)=0;

yo=yo\*m;

z=y;

y=y/m;

y(abs(y)<delta)=0;

y=y\*m;

n2=nnz(y);

for i=0:8:r-8

for j=0:8:c-8

p=i+1;

q=j+1;

zo(p:p+7,q:q+7)=Ho\*yo(p:p+7,q:q+7)\*Ho';

z(p:p+7,q:q+7)=inv(H')\*y(p:p+7,q:q+7)\*inv(H);

end

end

figure;

subplot(121);

imshow(x/255);

title("original image");

subplot(122)

imshow(z/255);

title("compressed image");

imwrite(x/255,'orginal.tif');

imwrite(z/255,'compressed.tif');

compression\_ratio = n2/n1;

Output:

Graphical user interface

Description automatically generated

**Question 6**

**Apply three types of highpass filtering in the frequency domain on Figure 1;**

**A. Ideal highpass filter (IHPF)**

**B. Butterworth highpass filter (BHPF)**

**C. Gaussian highpass filter (GHPF)**

A)

input\_image = imread('Brain MRI.png');

[M, N] = size(input\_image);

FT\_img = fft2(double(input\_image));

D0 = 30;

u = 0:(M-1);

idx = find(u>M/2);

u(idx) = u(idx)-M;

v = 0:(N-1);

idy = find(v>N/2);

v(idy) = v(idy)-N;

[V, U] = meshgrid(v, u);

D = sqrt(U.^2+V.^2);

H = double(D <= D0);

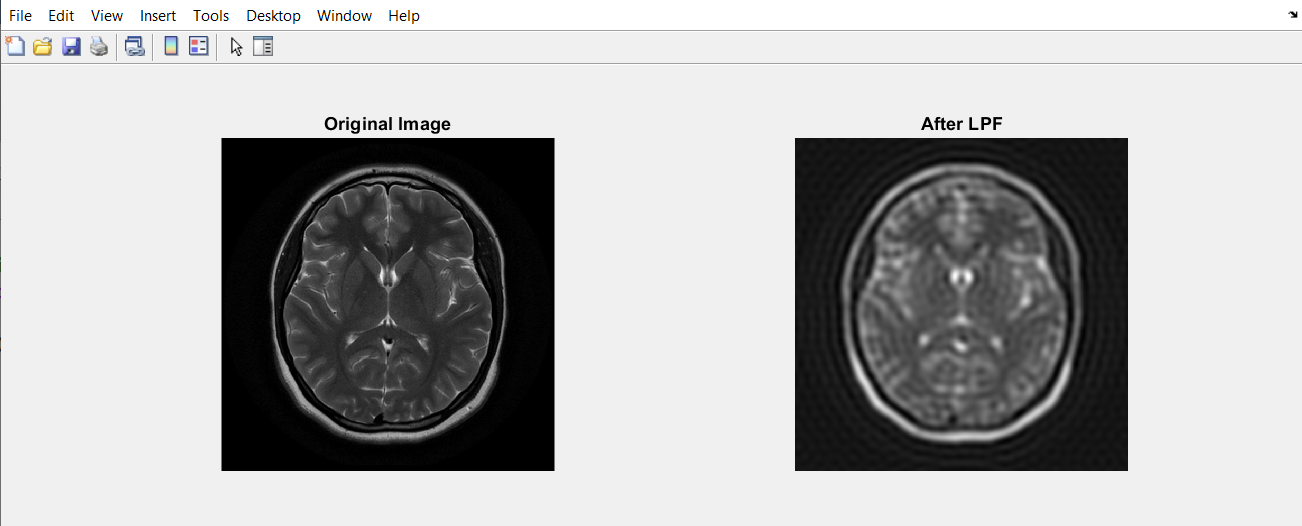
G = H.\*FT\_img;

output\_image = real(ifft2(double(G)));

subplot(2, 2, 1), imshow(input\_image), title('Original Image');

subplot(2, 2, 2), imshow(output\_image, [ ]), title('After LPF');

**Output:**



**B)**

input\_image = imread('Brain MRI.png');

[M, N] = size(input\_image);

FT\_img = fft2(double(input\_image));

n = 2;

D0 = 20;

u = 0:(M-1);

v = 0:(N-1);

idx = find(u > M/2);

u(idx) = u(idx) - M;

idy = find(v > N/2);

v(idy) = v(idy) - N;

[V, U] = meshgrid(v, u);

D = sqrt(U.^2 + V.^2);

H = 1./(1 + (D./D0).^(2\*n));

G = H.\*FT\_img;

output\_image = real(ifft2(double(G)));

subplot(2, 2, 1), imshow(input\_image), title('Original Image');

subplot(2, 2, 2), imshow(output\_image, [ ]), title('After BLF');

**Output:**

Graphical user interface, application

Description automatically generated

**C)**

img = imread('Brain MRI.png');

[m, n] = size(img);

fc = 10;

p = round(m/2);

q = round(n/2);

H = zeros(m,n);

for i = 1:m

for j = 1:p

d = (i-p).^2 + (j-q).^2;

H(i, j) = exp(-d/2/fc/fc);

end

end

A\_f = fftshift(fft2(img));

B = A\_f.\*H;

C = abs(ifft2(B));

glpf = uint8(C);

subplot(2, 2, 1), imshow(img), title('Original Image');

subplot(2, 2, 2), imshow(glpf), title('After GLPF');

**Output:**

Graphical user interface

Description automatically generated

**Question 7**

**Apply three types of lowpass filtering in the frequency domain on Figure 1;**

**A. Ideal lowpass filter (ILPF)**

**B. Butterworth lowpass filter (BLPF)**

**C. Gaussian lowpass filter (GLPF)**

**A)**

input\_image = imread('Brain MRI.png');

[M, N] = size(input\_image);

FT\_img = fft2(double(input\_image));

D0 = 10;

u = 0:(M-1);

idx = find(u>M/2);

u(idx) = u(idx)-M;

v = 0:(N-1);

idy = find(v>N/2);

v(idy) = v(idy)-N;

[V, U] = meshgrid(v, u);

D = sqrt(U.^2+V.^2);

H = double(D > D0);

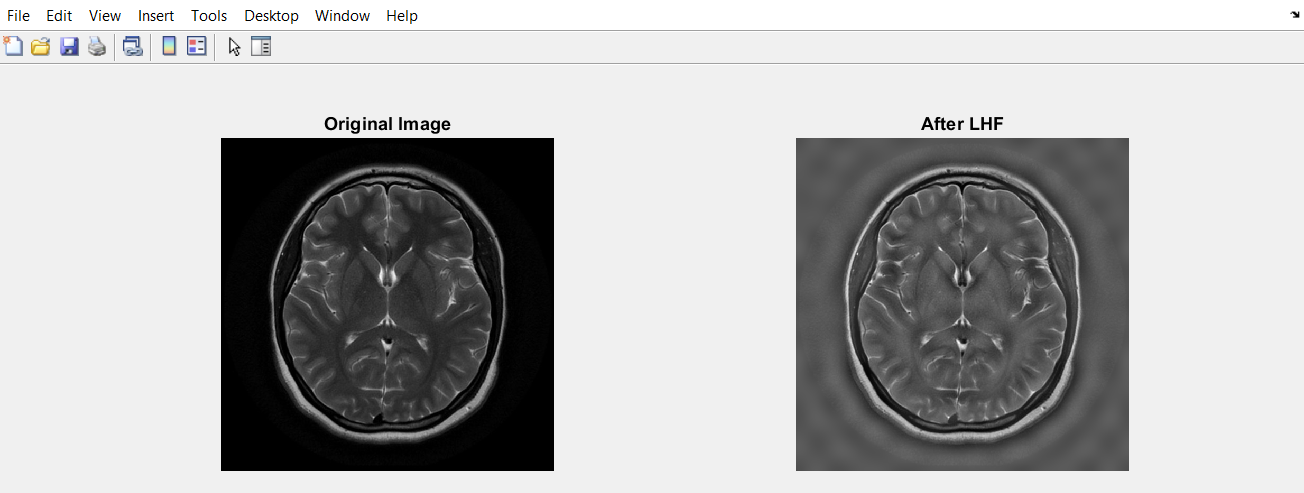
G = H.\*FT\_img;

output\_image = real(ifft2(double(G)));

subplot(2, 2, 1), imshow(input\_image), title('Original Image');

subplot(2, 2, 2), imshow(output\_image, [ ]), title('After LHF');

**Output:**



**B)**

input\_image = imread('Brain MRI.png');

[M, N] = size(input\_image);

FT\_img = fft2(double(input\_image));

n = 2;

D0 = 10;

u = 0:(M-1);

v = 0:(N-1);

idx = find(u > M/2);

u(idx) = u(idx) - M;

idy = find(v > N/2);

v(idy) = v(idy) - N;

[V, U] = meshgrid(v, u);

D = sqrt(U.^2 + V.^2);

H = 1./(1 + (D0./D).^(2\*n));

G = H.\*FT\_img;

output\_image = real(ifft2(double(G)));

subplot(2, 2, 1), imshow(input\_image), title('Original Image');

subplot(2, 2, 2), imshow(output\_image, [ ]), title('After BHF');

**Output:**

Graphical user interface

Description automatically generated

**C)**

img = imread('Brain MRI.png');

[m, n] = size(img);

fc = 10;

p = round(m/2);

q = round(n/2);

H = zeros(m,n);

for i = 1:m

for j = 1:p

d = (i-p).^2 + (j-q).^2;

H(i, j) = exp(-d/2/fc/fc);

end

end

H = 1-H;

A\_f = fftshift(fft2(img));

B = A\_f.\*H;

C = abs(ifft2(B));

ghpf = uint8(C);

subplot(2, 2, 1), imshow(img), title('Original Image');

subplot(2, 2, 2), imshow(ghpf), title('After GHPF');

**Output:**

Graphical user interface

Description automatically generated

**Question 8**

**Accurate diagnosis is the most vital part of the healthcare sector. For a precise diagnosis, the test results must display the features as clearly as possible. According to your observations while solving the previous problems, which filtering technique would you choose if you were given the task of performing image enhancement on the given image? Explain and justify your reasoning.**

**Answer:**

We should use the low pass and the high pass filters. Because The low pass filter is used as anti-aliasing filter while the high pass filter is used in audio amplifier for coupling or removing distortions due to low-frequency signal such as noise.