LAB 9

AIM: Fourier Transform and frequency domain analysis in image processing.

1. Preliminary: Implement DFT in MATLAB for 1 D array and use your script to perform the following DFT calculations:

a. Input Sequence = [-2 2 -4 4]
Output

```
 0.0000 + 0.0000i   2.0000 + 2.0000i -12.0000 - 0.0000i   2.0000 - 2.0000i
```

b. Input Sequence = [-4 -2 0 2 4]
Output

```
 0.0000 + 0.0000i  -5.0000 + 6.8819i  -5.0000 + 1.6246i  -5.0000 - 1.6246i  -5.0000 - 6.8819i
```

```
clear all;
f = input('Enter the sequence ');
M = input('No. of sample ');

if(M > length(f))
    for i = 1 : M - length(f)
        f = [f 0];
    end
end

F = []
ff = 0
for u = 0 : M - 1
    for x = 0 : M - 1
        ff = ff + f(x+1) * exp((-j*2*pi*u*x)/M)
    end
    F = [F ff];
    ff = 0;
end
```

task1.m | task2.m

```
clear all;
f = input('Enter the sequence ');
M = input('No. of sample ');

if(M > length(f))
    for i = 1 : M - length(f)
        f = [f 0];
    end
end

F = []
ff = 0
for u = 0 : M - 1
    for x = 0 : M - 1
        ff = ff + f(x+1) * exp((-j*2*pi*u*x)/M)
    end
    F = [F ff];
    ff = 0;
```

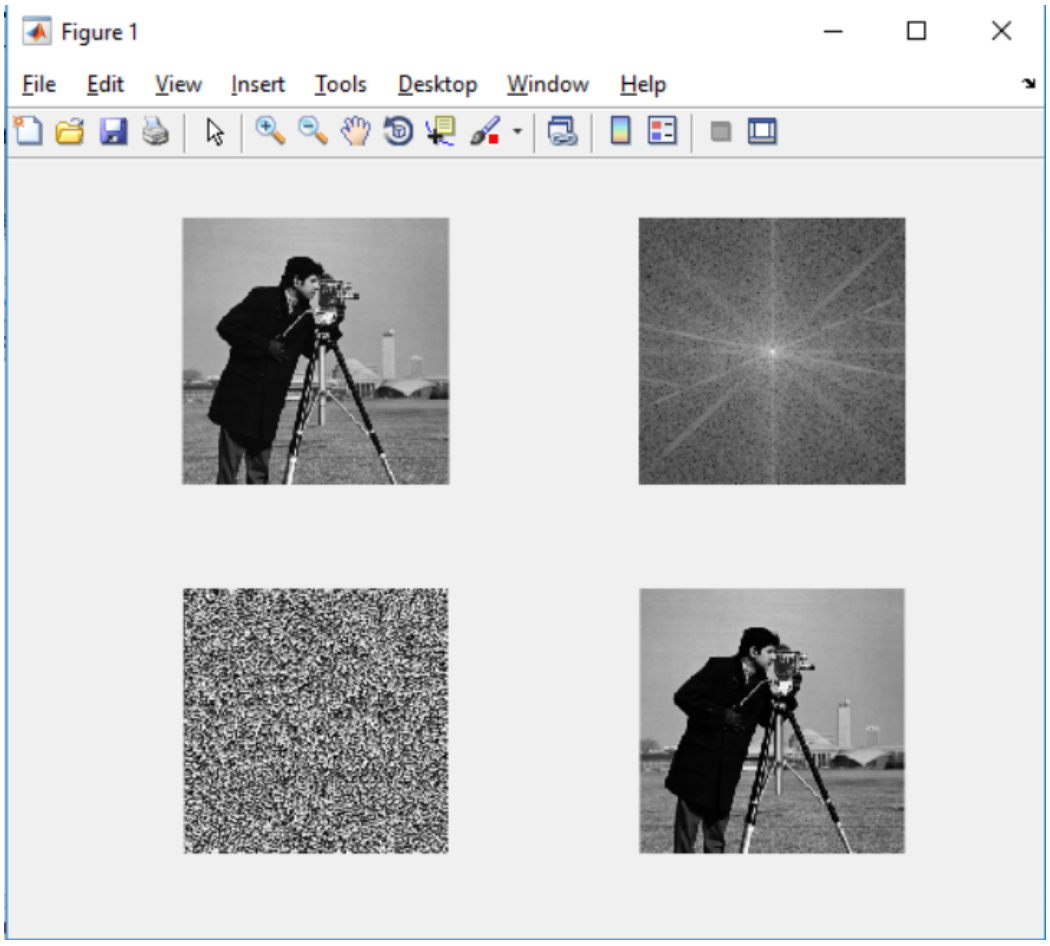
Command Window

```
ff =

   2.0000 + 2.0000i


ff =

   2.0000 - 2.0000i


F =

   0.0000 + 0.0000i   2.0000 + 2.0000i -12.0000 - 0.0000i   2.0000 - 2.0000i

>>
```

2. The MATLAB routines for computing the 2-D DFT and the inverse 2-D DFT are the routines fft2 and ifft2.
Using the image file cameraman.tif. Read in the image and perform DFT computations such that you get the following results.

```
f = imread('cameraman.tif');
subplot(2,2,1);
imshow(f);
f = double(f);

%FA
F = fft2(f);
F_spec = abs(F);
```

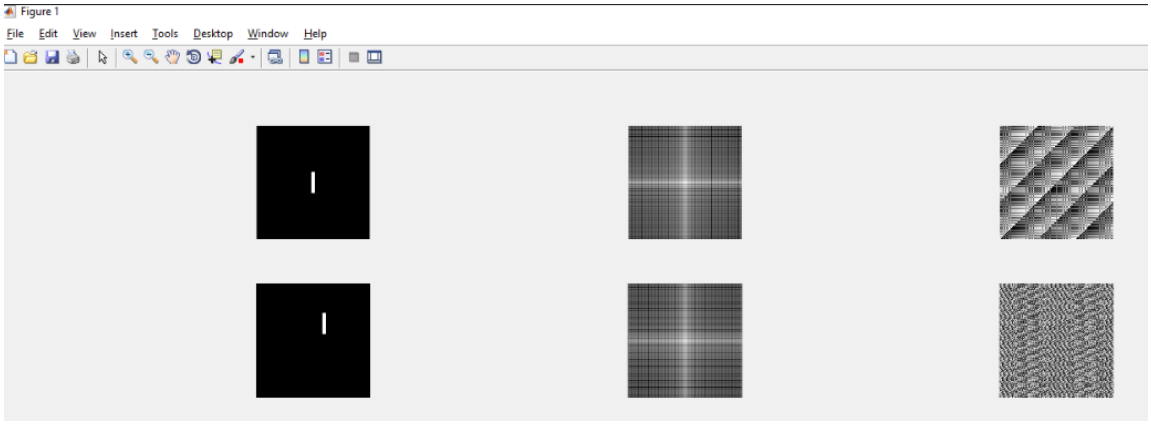
```
F_center_spec = fftshift(F_spec);
F_phase = angle(F);

subplot(2,2,2)
imshow(log(1+F_center_spec),[])

subplot(2,2,3)
imshow(F_phase,[])

%Reconstruction
recon = F_spec.*exp(i*F_phase)

subplot(2,2,4)
imshow(ifft2(recon),[])
```



3. Use the image Rectangle.tif, validate

a. If we translate the image using imtranslate ( ) by (x0, y0) = (100.0, -150.0), then magnitude spectrum of the original and translated image remains the same. However phase angle changes. You should get approximately the following result

```
clear all;
f = imread('Rectangle.tif');
subplot(4,3,1);
imshow(f);
f = double(f);

%FOA on original image
FO = fft2(f);
FO_spec = abs(FO);
FO_center_spec = fftshift(FO_spec);
FO_phase = angle(FO);

subplot(4,3,2)
imshow(log(1+FO_center_spec),[])

subplot(4,3,3)
imshow(FO_phase,[])

%FOA on translated image
f_t = imtranslate(f,[100.0 -150.0],'fillvalue',0);
subplot(4,3,4);
imshow(f_t);
f_t = double(f_t);

FOT = fft2(f_t);
FOT_spec = abs(FOT);
FOT_center_spec = fftshift(FOT_spec);
FOT_phase = angle(FOT);

subplot(4,3,5)
imshow(log(1+FOT_center_spec),[])

subplot(4,3,6)
imshow(FOT_phase,[])
```



4. Importance of DFT phase

Using the image woman.tif and Rectangle.tif, perform 2D DFT and IDFT computations to get the following results

a. Phase angle of woman.

b. Woman reconstructed using only the phase angle.

c. Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle

d. Reconstruction using the phase of the rectangle and the spectrum of the woman.

e. Reconstruction using woman spectrum and phase. It should match the original image.

```
clear all;
%women
wf = imread('woman.tif');
subplot(2,3,1);
imshow(wf);
wf = double(wf);

% phase & spectrum of women
WFO = fft2(wf);
WFO_spec = abs(WFO);
WFO_phase = angle(WFO);

subplot(2,3,2)
imshow(WFO_phase,[])

%rectangle
rf = imread('Rectangle.tif');
rf = imresize(rf,[512 512])
rf = double(rf);

% phase & spectrum of rectangle
```

```
RFO = fft2(rf);
RFO_spec = abs(RFO);
RFO_phase = angle(RFO);


%Reconstruction using women phase
recon = exp(i*WFO_phase);
subplot(2,3,3)
imshow(ifft2(recon),[])

%Reconstruction using women phase and rect spectrum
recon_2 = RFO_spec.*exp(i*WFO_phase);
subplot(2,3,4)
imshow(ifft2(recon_2),[])

%Reconstruction using women spectrum and rect phase
recon_3 = WFO_spec.*exp(i*RFO_phase);
subplot(2,3,5)
imshow(ifft2(recon_3),[])

%Reconstruction using women spectrum and phase
recon_4 = WFO_spec.*exp(i*WFO_phase);
subplot(2,3,6)
imshow(ifft2(recon_4),[])
```