Lab-09

-Jainil Trivedi (CE166)

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]

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

Code:

```
f = input('eNTER SEQUENCE');
M = input('Length of samples taken');

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((-2*pi*i*u*x)/M);
   end
   F = [F ff];
```

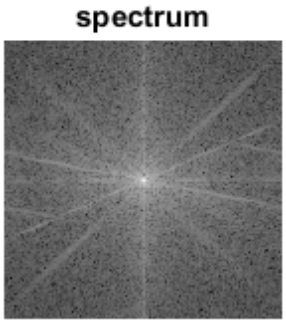
```
    ff=0;
end
%r F
```

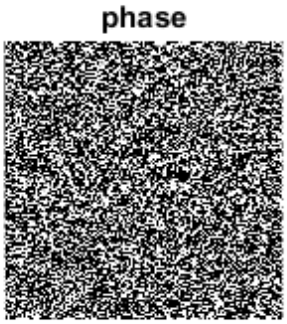
Output :

```
>> L9_1
eNTER SEQUENCE [-2 2 -4 4]
Length of samples taken 4
>> L9_1
eNTER SEQUENCE [-2 2 -4 4]
Length of samples taken 4

F =

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

>> L9_1
eNTER SEQUENCE [-4 -2 0 2 4]
Length of samples taken 5

F =

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

fx >>
```

```
% fft2 => fast fourier transform for 2d
img = imread('cameraman.tif');
img = double(img);
subplot(2,2,1);
imshow(img,[]);
title('Initial image');
% Fourier transform
ft_camera = fft2(img);

% find phase and amplitude
f_spec = fftshift(abs(ft_camera));   % shift and centralise
f_spec = log(1+f_spec);
subplot(2,2,2);
imshow(f_spec,[]);
title('spectrum');

f_phi = angle(ft_camera);%f_phi = atan2(img(Ft_camera),real(ft_camera);
subplot(2,2,3);
imshow(f_phi);
title('phase');

%  reconstruction  ifft2
original = abs((ft_camera)).*exp(1i*f_phi);

% IFT
IFT_ori = ifft2(original);
subplot(2,2,4);
imshow(IFT_ori,[]);
title('Reconstructed');
```

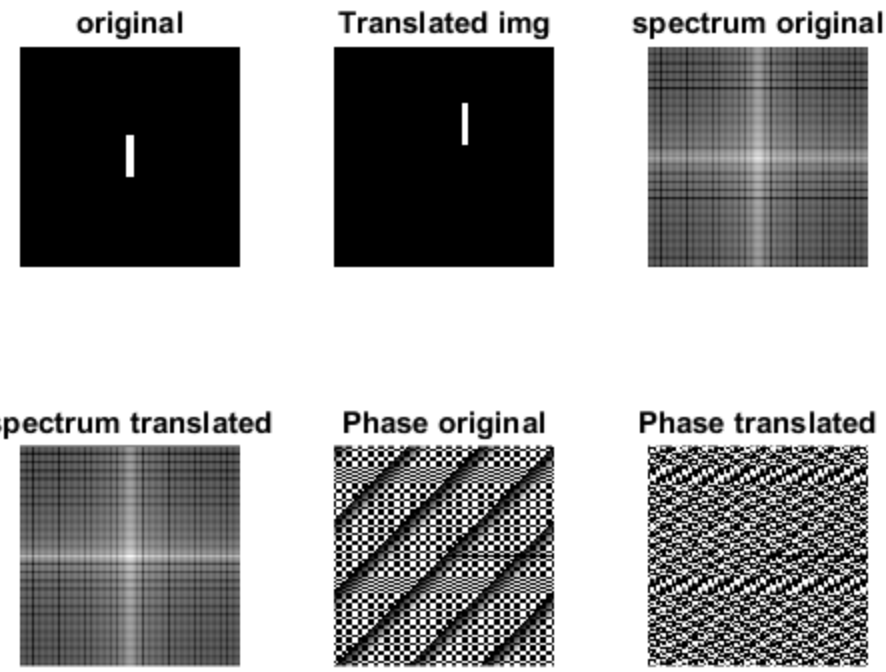
```
Warning: Displaying real part of complex input.
```





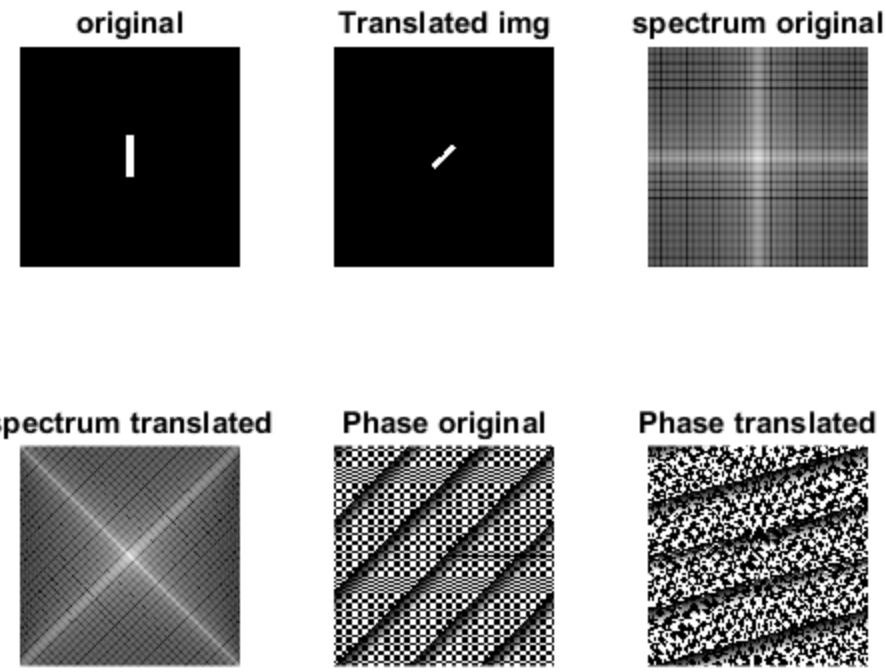
Published with MATLAB® R2015a

```
% Translate
img = double(imread('Rectangle.tif'));
subplot(2,3,1);
imshow(img);
title('original');

tran_img = (imtranslate(img,[100,-150],'fillvalues',0));
subplot(2,3,2);
imshow(tran_img);
title('Translated img');

f = fft2(tran_img);

f_phi = angle(f);
subplot(2,3,6);
imshow(f_phi);
title('Phase translated');

f = (fftshift(abs((f))));
subplot(2,3,4);
imshow(log(1+f),[]);
title('spectrum translated');



ff = fft2(img);
ff_phi = angle(ff);
subplot(2,3,5);
imshow(ff_phi);
title('Phase original');
ff = (fftshift(abs((ff))));
subplot(2,3,3);
imshow(log(1+ff),[]);
title('spectrum original');
```



Published with MATLAB® R2015a

```
% Rotate
img = double(imread('Rectangle.tif'));
subplot(2,3,1);
imshow(img);
title('original');

tran_img = (imrotate(img,-45,'bicubic'));
subplot(2,3,2);
imshow(tran_img);
title('Translated img');


f = fft2(tran_img);
f_phi = angle(f);
subplot(2,3,6);
imshow(f_phi);
title('Phase translated');

f = (fftshift(abs((f))));
subplot(2,3,4);
imshow(log(1+f),[]);
title('spectrum translated');


ff = fft2(img);
ff_phi = angle(ff);
subplot(2,3,5);
imshow(ff_phi);
title('Phase original');
ff = (fftshift(abs((ff))));
subplot(2,3,3);
imshow(log(1+ff),[]);
title('spectrum original');
```

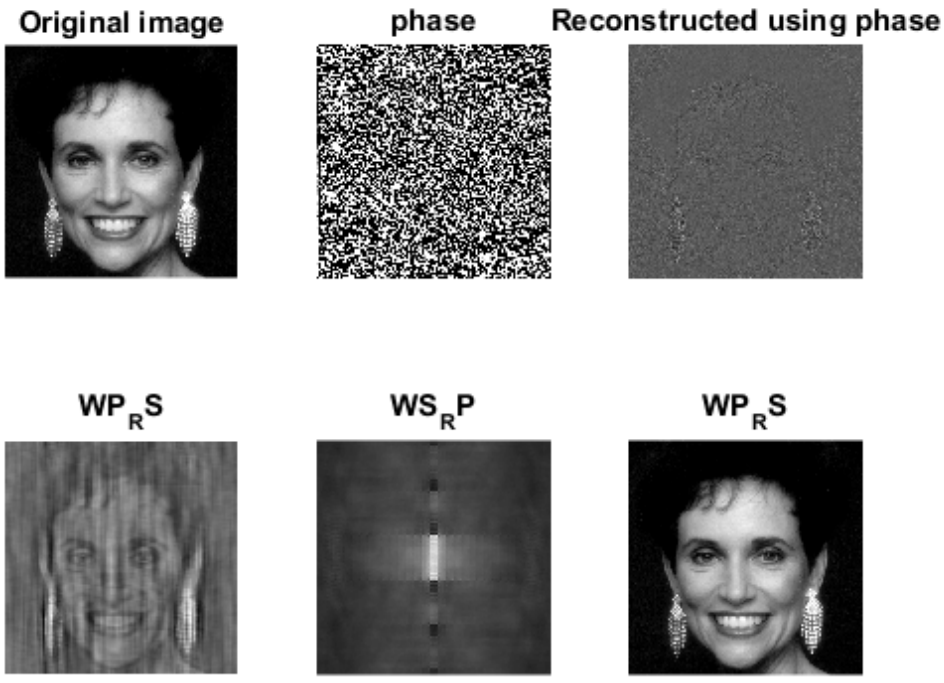


Published with MATLAB® R2015a

```
% women
img = imread('woman.tif');
img = double(img);
subplot(2,3,1);
imshow(img,[]);
title('Original image');
% Fourier transform
ft_woman = fft2(img);

% find phase and amplitude
f_spec = fftshift(abs(ft_woman));
f_spec = log(1+f_spec);
% subplot(2,3,3);
% imshow(f_spec,[]);
% title('spectrum');

f_phi = angle(ft_woman);
subplot(2,3,2);
imshow(f_phi);
title('phase');


% const using phase
original = exp(1i*f_phi);

% IFT
IFT_ori = ifft2(original);
subplot(2,3,3);
imshow(IFT_ori,[]);
title('Reconstructed using phase');



%  RECTANGLE
rect = double(imread('Rectangle.tif'));
rect = imresize(rect,0.5);
f_rect = fft2(rect);
rect_phi = angle(f_rect);
rect_spec = log(1+(fftshift(abs((f_rect)))));

% resize
% rect_phi1 = imresize( rect_phi,0.5);
% rect_spec1 = imresize(rect_spec,0.5);

% getting origginal image
original = abs(ft_woman).*exp(i*rect_phi);

% IFT
IFT_ori = ifft2(original);
subplot(2,3,5);
imshow(IFT_ori,[]);
title('WS_RP');

original = abs(f_rect).*exp(i*f_phi);
```

```
% IFT
IFT_ori = ifft2(original);
subplot(2,3,4);
imshow(IFT_ori,[]);
title('WP_RS');

original = abs(f_rect).*exp(i*f_phi);

% IFT
IFT_ori = ifft2(original);
subplot(2,3,4);
imshow(IFT_ori,[]);
title('WP_RS');

original = abs(ft_woman).*exp(i*f_phi);

% IFT
IFT_ori = ifft2(original);
subplot(2,3,6);
imshow(IFT_ori,[]);
title('WP_RS');
```

```
Warning: Displaying real part of complex input.
Warning: Displaying real part of complex input.
Warning: Displaying real part of complex input.
Warning: Displaying real part of complex input.
Warning: Displaying real part of complex input.
```



Published with MATLAB® R2015a