Image Enhancement in the frequency Domain

A frequency content refers to the rate at which the gray level change in the image, where the rapidly changing in brightness values correspond to high frequency, while the slowly change in brightness values correspond to low frequency.

Discrete Fourier Transform

MATLAB has three functions to compute the DFT:

1. fft -for one dimension (useful for audio)
2. fft2 -for two dimensions (useful for images)
3. fftn -for n dimensions

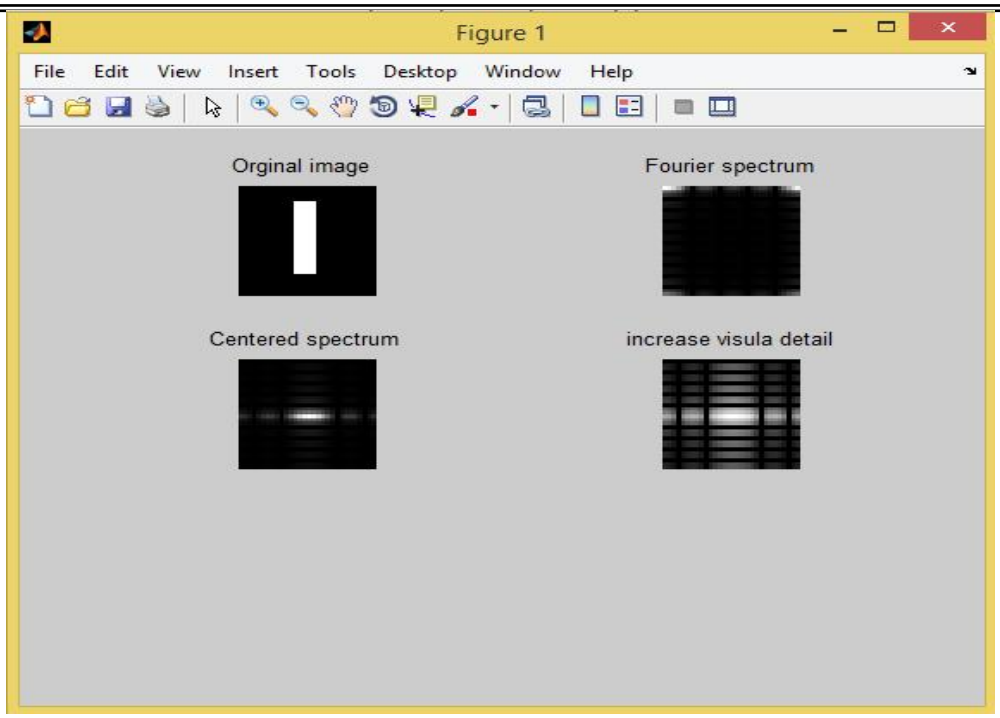
MATLAB has three related functions that compute the inverse DFT:

0. ifft
1. ifft2
2. ifftn

How to Display a Fourier Spectrum using MATLAB

The following program is meant to describe the various steps behind displaying the Fourier Spectrum.

```
clc
f=zeros(30,30);%Create a black 30x30 image
f(5:24,13:17)=1; %With a white rectangle in it.
subplot(3,2,1), imshow(f,[],'InitialMagnification','fit');title('Orginal image ');
%Calculate the DFT.
F=fft2(f);
F2=abs(F); %There are real and imaginary parts to F, Use the abs function to compute the magnitude of the combined components.
subplot(3,2,2), imshow(F2,[],'InitialMagnification','fit');title('Fourier spectrum');
Fc=fftshift(F);
F2=abs(Fc);
 subplot(3,2,3), imshow(F2,[],'InitialMagnification','fit');title('Centered spectrum');
F2=log(1+F2);
 subplot(3,2,4), imshow(F2,[],'InitialMagnification','fit');title('increase visula detail');
```



Filtering

Creates a variety of two dimensional filters by using *fspecial* function:

```
h = fspecial(type, parameters)
```

where:

h is two-dimensional correlation kernel.

type is one of the specified special filter types.

parameters are particular to the type of filter chosen

Filtering using *imfilter* function

```
B = imfilter(A, H)
```

Where

B is output image

A is input image

H is the filter

```
Y = filter2 (h,X)
```

filters the data in X with the two-dimensional filter in the matrix h.

The following table is meant to show you three filters, created by fspecial, and the results on an image of a cameraman:

MATLAB Code	Resulting Image
%original picture cam=imread('cameraman.tif'); figure, imshow(cam)	

% blurring
h=fspecial('average',[7 7]);
cam2 =imfilter(cam,h);
figure,imshow(cam2)
	

%sharpening
h=fspecial('unsharp');
cam_sharp=imfilter(cam,h);
figure, imshow(cam_sharp)
	

%edge-detection
h=fspecial('sobel');
cam_sobel=imfilter(cam,h);
figure, imshow(cam_sobel)
	

Example: Applying the gaussian Filter in the Frequency Domain

```
%Create the Frequency Filtered Image
clc;
clear;
f=imread('rice.png');
subplot(1,3,1);imshow(f);title('original image')
F=fft2(f);
h=fspecial('gaussian');
g=imfilter(f,h);
subplot(1,3,2);imshow(g);title('spatial')
[r c]=size(f);
H=fft2(h,r,c);
FF=F.*H;
Fi=ifft2(FF);
subplot(1,3,3);imshow(Fi,[]);title('frequincy')
```