4

Chapter Four

Frequency Domain

The 2D Discrete Fourier Transform

4.1 Fourier Transform Properties

- Fourier Transform (FT) is performing many tasks which would be impossible to perform in any other ways. The advantages of FT in image processing field could be summarised as:

1. A powerful alternative method to linear spatial filtering
2. Fast and More efficient than large spatial filters
3. Process and isolate some particular frequencies
4. Perform LP and HP filters in very high precision
5. Perform Band Reject Filters for image restoration

- A periodic function may be written as the sum of sines and cosines of varying ***amplitudes*** and ***frequencies***

$f(\times) = \sin(\times) + 1/3 \sin(2\times) + 1/5 \sin(4\times) + 1/7 \sin(8\times)$

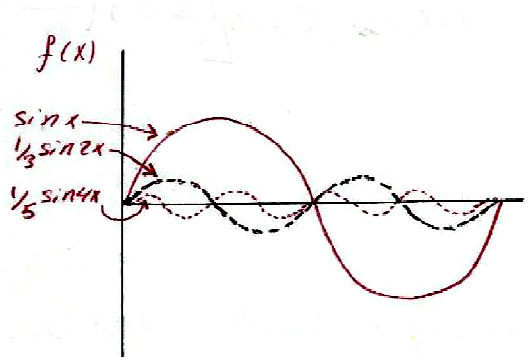


Fig 4.1 different amplitudes and frequencies for 1D periodic signals

and for a square wave

$f(\times) = \sin \times + 1/3 \sin 3\times + 1/5 \sin 5\times + 1/7 \sin 7\times$

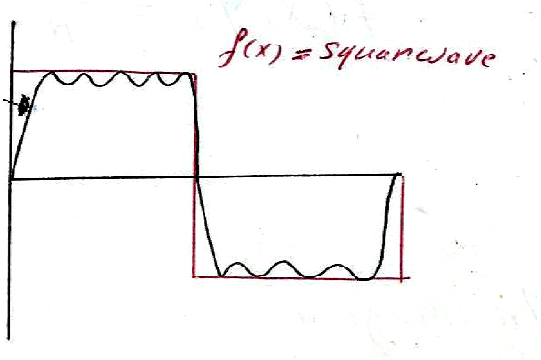


Fig 4.2 square wave components

The *one dimension* Discrete Fourier Transform is,

$$F(u) = a_0 + \sum_{X=1}^{M} f(x)\left(a_n \cos\frac{xu\pi}{M} + b_n \sin\frac{xu\pi}{M}\right)$$

or in exponential form,

$$F(u) = \sum_{x=0}^{M-1} f(x)\ e^{-2\,\pi\, j\left(\frac{xu}{M}\right)}$$

The inverse FT is,

$$f(x) = 1/M \sum_{u=0}^{M-1} \mathrm{F}(u)\ e^{2\,\pi\, j\left(\frac{xu}{M}\right)}$$

The *2D dimensions* Discrete Fourier Transform is,

$$F(u,v) = \sum_{N=0}^{N-1} \sum_{M=0}^{M-1} f(x,y)\ e^{-2\,\pi\, j\left(\frac{xu}{M}+\frac{yv}{N}\right)}$$

And, the *inverse* of 2D DFT would be,

$$f(x,y) = 1/MN \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} F(u,v)\ e^{2\,\pi\, j\left(\frac{xu}{M}+\frac{yv}{N}\right)}$$

- **Similarity**

The inverse and forward transforms are very similar with the exception of scale factor 1/MN and the negative sign.

So the term $e^{-/+2\,\pi\, j\left(\frac{xu}{M}+\frac{yv}{N}\right)}$ is look like a DFT spatial filter because:

The value of ***f*** and ***F*** are independent and could be calculated separately, so the value of *F(u,v)* is obtained by multiplying every value of *f(x,y)* by a fixed value, $e^{-2\,\pi\, j\left(\frac{xu}{M}+\frac{yv}{N}\right)}$ and adding up all the result and this is what a linear filter does where the convolution in spatial linear filter multiplies all elements under a mask with fixed values and adds them all up.

- **Separability**

The 2D DFT (filter elements) can be expressed as product of,

$$e^{-2\pi j\left(\frac{xu}{M}+\frac{yv}{N}\right)} = \underbrace{e^{-2\pi j\left(\frac{xu}{M}\right)}}_{\text{depends on x \& u only}} \cdot \underbrace{e^{-2\pi j\left(\frac{yv}{N}\right)}}_{\text{depends on y \& v only}}$$

so the 2D DFT can be calculated by using the separability property, we first compute the DFT *for all rows* and then complete the DFT *of all columns* of the result.

$$F(u/v) = \sum_{x,y=0}^{M-1,N-1} f(x/y)\ e^{-2\pi j\left(\frac{xu}{M}/\frac{yv}{N}\right)}$$

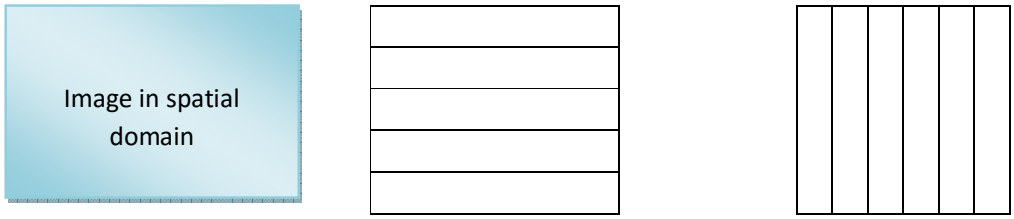


Fig . 4.3 Apply 2D DFT as a 1D DFT

- **Linearity**

$FT\ (f + g\ ) = FT(f\ ) + F(\ g\ )$
$FT(\ k f\ ) = k\ F(f\ )$

noise can be reduced or removed depending on this property,

$d = f + n$
$F(\ d\ ) = F\ (f\ ) + F\ (\ n\ )$

Where: d: degraded image, f: original image, n: noise

Some noise appears on the DFT in a way that makes it easy to remove

- **Digital Convolution**

Suppose we wish to convolve the large image M with a small mask S. This method of convolution is very slow, so we pad S with zeros to be with same size of image M and do the multiplication

$H = M * S$
$F(\ H\ ) = F\ (M)\ .\ F(\ {}'_S\ )$ ((convolution in time domain equals multiplication in frequency domain))
and the inverse would be, $F^{-1}(H) = \ F^{-1}(\ F(\ M\ )\ .\ F\ (\ {}'_S\ )\ )$

- **The DC coefficients**

The dc situation is satisfied when all frequency equal to zero. That means (*u=0, v=0*) in the transformation formula

$$F(0,0) = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x,y)\ e^{-2\pi j\left(\frac{x0}{M}+\frac{y0}{N}\right)}$$

So,

$$F(0,0) = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x,y)$$

That means the summation of pixels in the original image equal to the DC

H.W//

The convolution of $M_{512\times512}$ and $S_{32\times32}$ is required $(512)^2 \times (32)^2$ multiplication process to be completed. How many process of multiplications do the need for 2D DFT.

- **Conjugate Symmetry & Shifting**

  It is convenient to have the DC component in the center of the transformed matrix for the purpose of display. Also, the symmetry property of the Fourier transform could be obtained by substituting

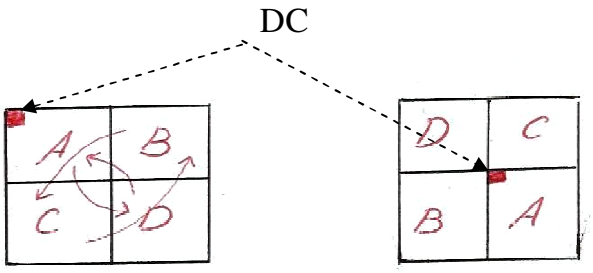
  (u = - u) and (v = - v) in the 2D DFT formula

The sign will be +ve

$$F(u,v) = \sum_{N=0}^{N-1} \sum_{M=0}^{M-1} f(x,y)\ e^{-2\pi j\left(\frac{xu}{M}+\frac{yv}{N}\right)}$$

$$F(u, v) = F\ (-u + pM, -v + qM)^*$$

There are mirror sub-images of conjugate spectrums between the top and bottom halves or left and right halves. The symmetry means that its information is given in just half of a transformation and the other half is redundant, see figure 4.4 a and b.



(a) DC and shifting property

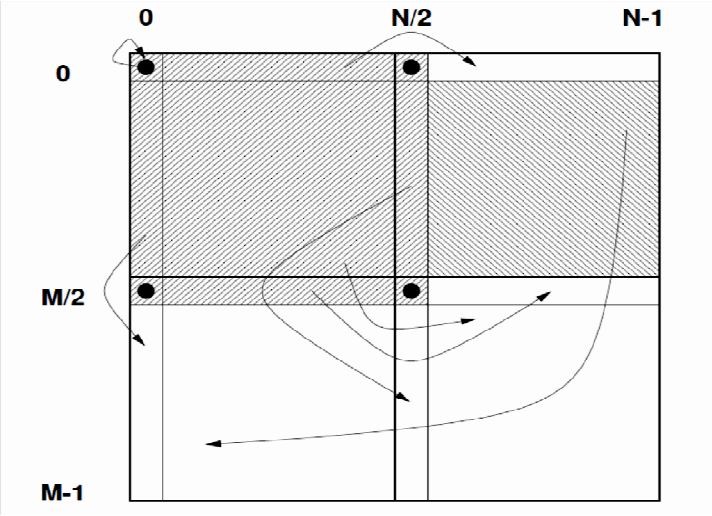


Fig. 4.4 (a and b) The shifting and conjugate symmetry property

- **MATLAB built-in function for 2D DFT**
  - *fft2 (image matrix)*
  - *ifft2 (image matrix)*
  - *fftshift (image matrix)*
  - *fftshift (fft2 (image matrix))*

- **Ex-1**

>> A = ones (8)
>> Af = fft2 (A)

Ans = 64 0 0 0 0 0 0 0
0
:
0 0 0 0 0 0 0 0

An image is considered as a two dimensional function f(x,y) and can be expressed as sum of ***corrugation*** functions having the general form,
$Z = a \sin(bx + cy)$. In this example, no corrugations are required to form a constant.

- **Ex-2**:

```
>> B = [ 100 200; 100 200];   % a matrix B in this example consisting a single corrugation
>>  B = repmat (B, 4,4)
>>  BF = fft2(B)
```

ans =

100	200	100	200	100	200	100	200
100	200	100	200	100	200	100	200
:	:	:	:	:	:	:	:
100	200	100	200	100	200	100	200

9600	0	0	-3200	0	0	0	0
0	0	0	0	0	0	0	0
:	:	:	:	:	:	:	:
0	0	0	0	0	0	0	0

Note: The DC = 9600 = 150*64, the mirroring of values about the dc coefficient is a sequence of the symmetry of the DFT which is required to form an edge.

There is one way to display the image spectrums in frequency domain. We stretch out the spectrum values by taking the logarithm of $|F(u,v)|$ using the MATLAB command

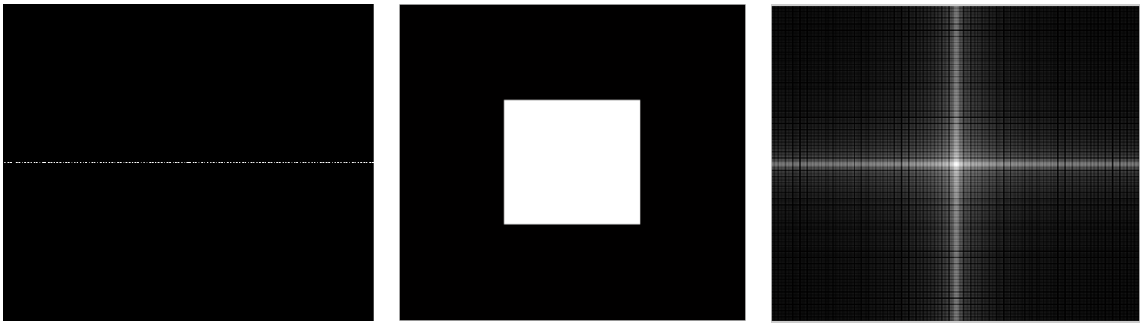
>> imshow (mat2gray (log (1+ abs (FD image matrix))))

- **Ex-3**

```
>> a  =  [ zeros (256,128) ,  ones (256,128) ];      % a simple single edge image
>> af = fftshift( fft2 ( a ) );
>> imshow ( mat2gray ( log ( 1+ abs( af ) ) ) );
```

- **Ex-4**

```
>> c  =  zeros (256,256 )                          % a single box image
>> c(78:178, 78:178)  = 1;
>> cf = fftshift( fft2 ( c ) );
>>  imshow (c);
>> imshow ( mat2gray ( log ( 1+ abs( cf) ) ) );
```

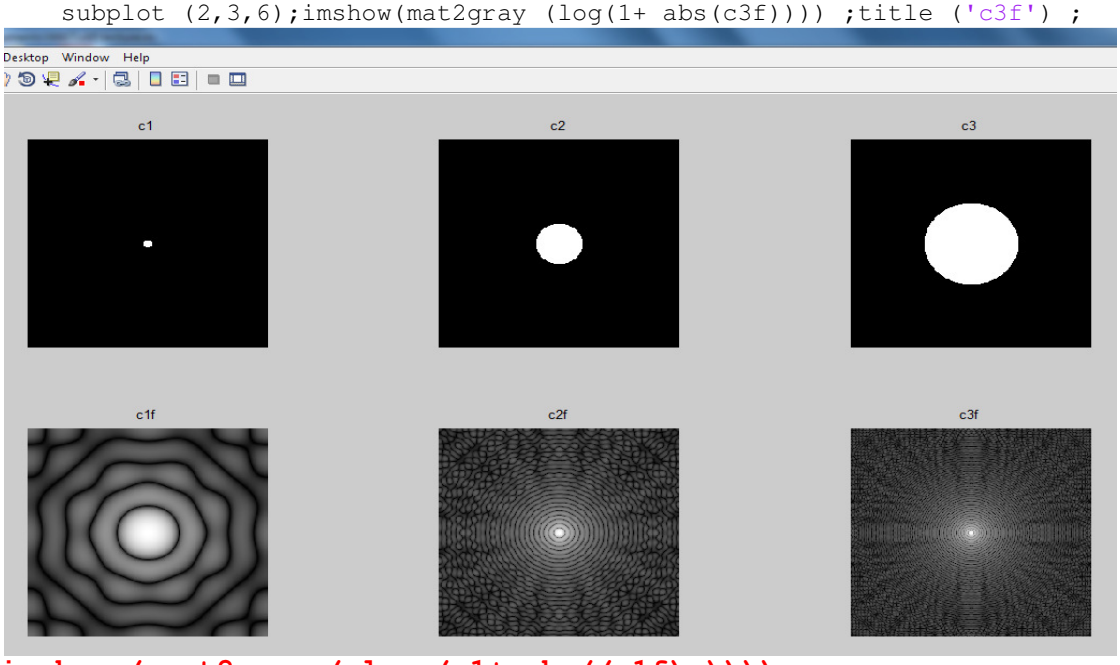


af — c — cf

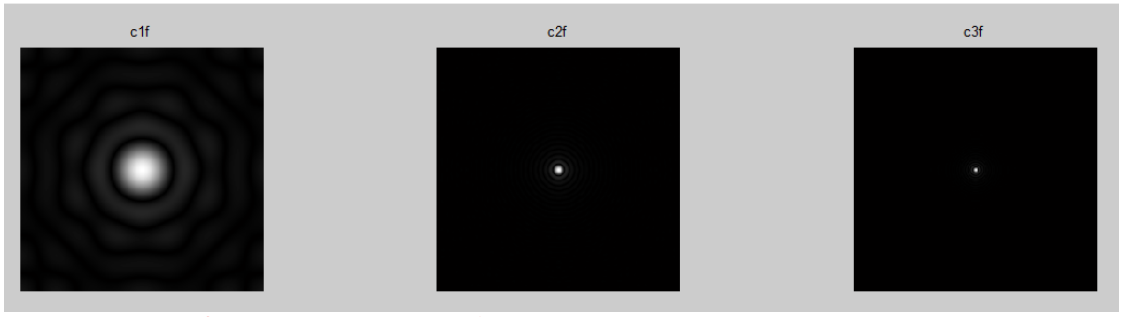
Fig 4.5 frequency domain — spatial domain — frequency domain

- **Ex-5** This example explains how to create a circle as it used as a frequency domain filter.

```
clc; close all; clear all

[ x, y] = meshgrid (-128 : 127 , -128 : 127);
      z = sqrt ( x.^ 2 + y.^2);
     c1 = (z < 5 );
     c2 = (z < 25);
     c3 = (z < 50);

 subplot  (2,3,1); imshow(mat2gray (c1)) ;title ('c1')  ;
 subplot  (2,3,2); imshow(mat2gray (c2)) ;title ('c2')  ;
 subplot  (2,3,3); imshow(mat2gray (c3)) ;title ('c3')  ;

 c1f = fftshift ( fft2 ( c1 ));
 c2f = fftshift ( fft2 ( c2 ));
 c3f = fftshift ( fft2 ( c3 ));

 subplot (2,3,4);imshow(mat2gray (log(1+ abs(c1f)))) ;title ('c1f') ;
 subplot (2,3,5);imshow(mat2gray (log(1+ abs(c2f)))) ;title ('c2f') ;
 subplot (2,3,6);imshow(mat2gray (log(1+ abs(c3f)))) ;title ('c3f') ;
```



imshow (mat2gray (log (1+ abs((c1f)))))



imshow(mat2gray (abs ((c1f))))

Fig. 4.6 The circle (LPF) and its transformation in the frequency domain

The *meshgrid(m , n) of [x,y]* will create two overlaid matrices (x and y) each one with two dimensions *(n , m)* , for example:

>> [x,y] = meshgrid(1:3,1:5) *Ans* *x =* *1 2 3* *1 2 3* *1 2 3* *1 2 3* *1 2 3* *y =* *1 1 1* *2 2 2* *3 3 3* *4 4 4* *5 5 5*	*>> [x,y] = meshgrid(2:6,1:3)* *ans* *x =* *2 3 4 5 6* *2 3 4 5 6* *2 3 4 5 6* *y =* *1 1 1 1 1* *2 2 2 2 2* *3 3 3 3 3*

4.2 Filtering in the Frequency Domain

One of the reasons for the use of the Fourier transform in image processing is due to convolution theorem. A spatial convolution can be performed element wise multiplication of the Fourier transform by a suitable "filter matrix"

4.2.1 Ideal Low Pass Filtering

After shifting the DC coefficients of any transformed image toward the centre, all low frequency components will be moved to the centre as well. We can perform LP filtering by multiplying the transform by a matrix in such a way that centre values are maintained and values away from the centre are either removed or minimized. One way to do this is to multiply by an ideal *low-pass* matrix, which is a binary matrix, say ***m*** defined by

$$\textit{LP filter} \rightarrow m(x,y) = \begin{cases} 1 \text{ if } (x,y) \text{ is closer to the centre than some value, } C_{cut\ off} \\ 0 \text{ if } (x,y) \text{ is further from the centre than } C_{cut\ off} \end{cases}$$

The inverse FT of the element-wise product of any image says, ***I*** and ***m*** is,

$F^{-1}(I . m)$

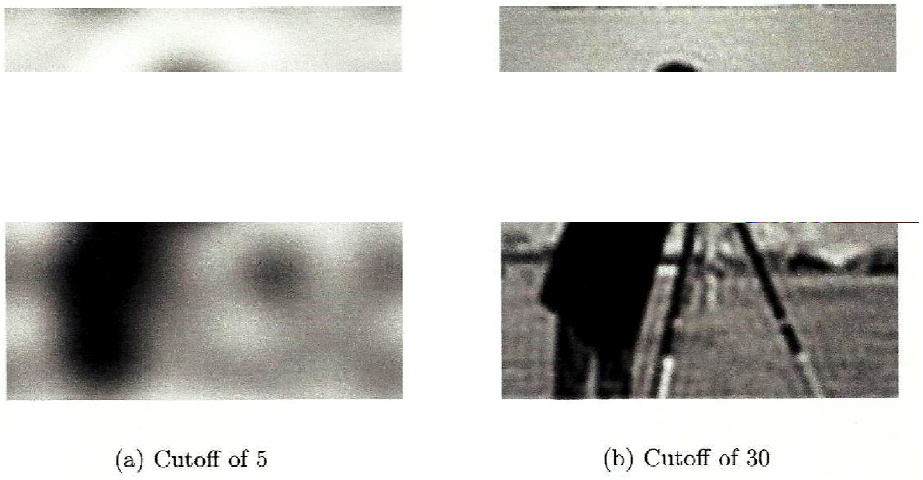
The smallest circle radius, the sharpest $C_{cut\ off}$ frequency we obtained and the inversed image would be more smeared. See figure 4.7

4.2.2 Ideal High Pass Filtering

Just as we can perform LP filters by keeping the 2D DFT centre values and eliminating the others, so HP filters can be performed by the opposite: eliminating centre values and keep the others. This can be done with a minor modification of the preceding method of LP filtering. See figure 4.8 and 4.9

- **Ex-6**

```
clc; close all; clear all
        [ x, y] = meshgrid (-128 : 127 , -128 : 127);
        z = sqrt ( x.^ 2 + y.^2);
        c1 = (z < 5);  c2 = (z < 25); c3 = (z < 50);

 subplot  (2,3,1); imshow(mat2gray (c1)) ;title ('c1')  ;
 subplot  (2,3,2); imshow(mat2gray (c2)) ;title ('c2')  ;
 subplot  (2,3,3); imshow(mat2gray (c3)) ;title ('c3')  ;

 c1f = fftshift ( fft2 ( c1 ));
 c2f = fftshift ( fft2 ( c2 ));
 c3f = fftshift ( fft2 ( c3 ));

 subplot (2,3,4);imshow(mat2gray (log(1+ abs(c1f)))) ;title ('c1f') ;
 subplot (2,3,5);imshow(mat2gray (log(1+ abs(c2f)))) ;title ('c2f')   ;
 subplot (2,3,6);imshow(mat2gray (log(1+ abs(c3f)))) ;title ('c3f') ;
```



(a) Cutoff of 5 (b) Cutoff of 30

Fig 4.7 Frequency domain LPF of cameraman image

As long as ideal filtering is required a dot product to be completed, it is very important to ***match the size of filter with the size of transformed image***. The necessary MATLAB steps for size matching are:

```
I=  imread ('kids.tif');
[r,c] = size (I);
[ x, y] = meshgrid (1:c , 1:r);
z = sqrt ( (x-c/2).^ 2 + (y-r/2).^2);
c = ( z > 15 )
```

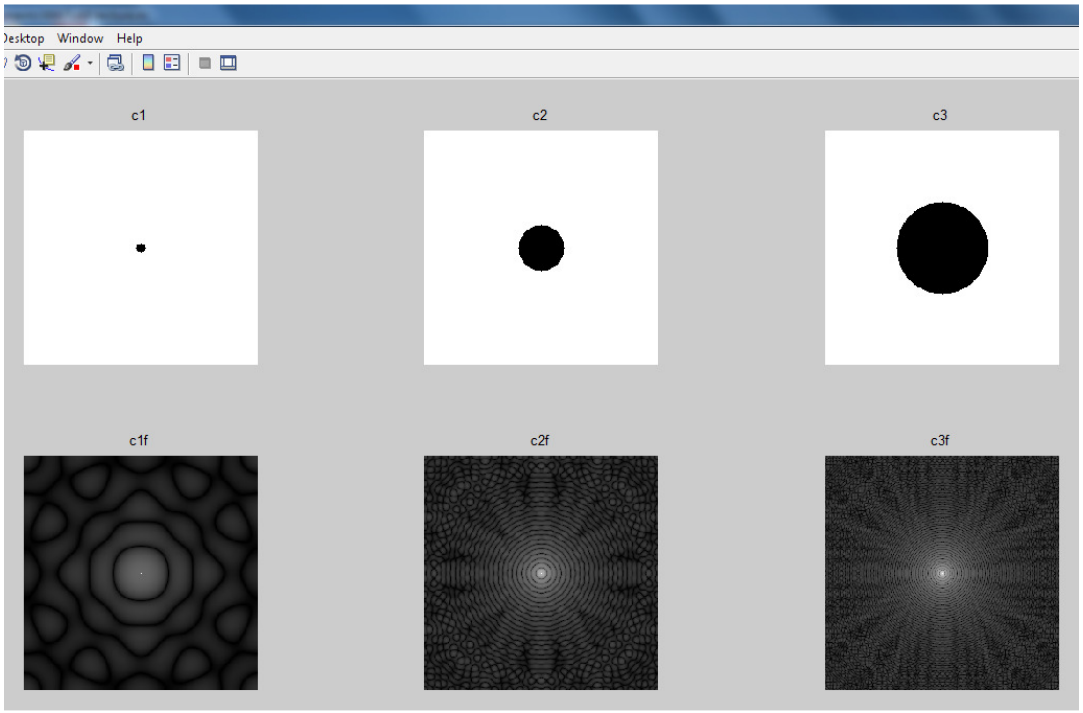


Fig. 4.8 Ideal HP filters with their transformations: c1>5, c2>25, c3>50

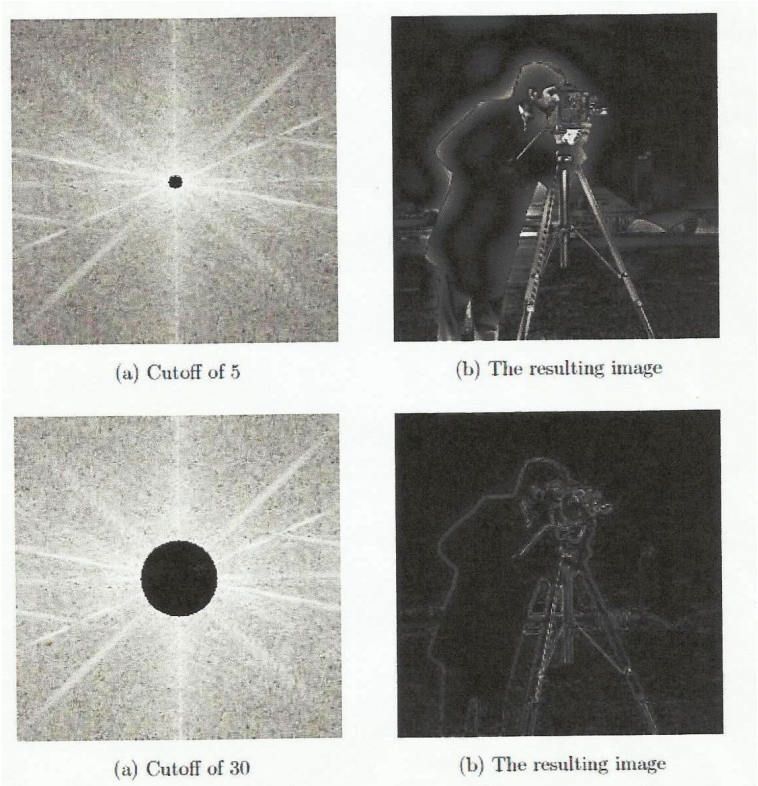


Fig. 4.9 ideal HP filtering for cameraman with cut off frequencies 5 and 30

Tutorial

The image matrix $A_{3\times3}=\begin{bmatrix}5 & 0 & 7\\ 1 & -1 & 0\\ 2 & 3 & 4\end{bmatrix}$ is required to be transformed to the frequency domain by applying the 2D DFT, calculate:

a. The DC values
b. The transformed image AF
c. The shifted transform image AFS
d. The mirror components
e. Restore the original $A_{3\times3}$ matrix from AF
f. Restore the original $A_{3\times3}$ matrix from AFS
g. Make the DC value of ASF = 0 and recalculate the original $A_{3\times3}$.

Is there any difference in the $A_{3\times3}$ values in e, f, and g?