Sharpening Frequency Domain Filters

. highpass filtering sharpens the image by attenuating the Low frequencies and Leaving the high frequencies of the Fourier transform relatively unchanged.

The high pass filter transfer function H_{hp} is often represented by its relationship to the Lowpass filter $H_{\ell p}$

$$H_{hp}(u,v) = 1 - H_{Lp}(u,v)$$

(Note : See textBook for Visual representation (plots and Images) of Ideal, Butterworth and Gaussian high pass filters (IHPF, BHPF and GHPF respectively).

high pass filter generator matlab function:

```
function H = hpfilter( type, M, N, Do, n)
 % HPFILTER Computes frequency domain highpass filters
% H= HPFILTER (TYPE, M, N, Do, n) Creates the transfer
% function of a highpass filter, H, of the specified
% TYPE and Size (M-by-N).
% Valid Values for TYPE, Do and n are:
% 'Ideal'    Ideal highpass filter with Cutoff frequency
             Do. n need not be supplied. DO must be positive
% 'btw'      Butterworth high pass filter of order n,
%            and Cutoff Do. The default value of
%            n =1.0 . Do must be positive.
% 'gaussian' Gaussian highpass filter with cutoff
           ( Standard deviation) Do. n need not be supplied
             Do must be positive.
%  The transfer function Hhp of a high pass filter
% is 1- Hℓp, where Hℓp is the transfer function
% of the corresponding Low pass filter.
% We can use function ℓpfilter to generate
%  highpass filters
```

```
if  nargin == 4
    n = 1 ;   % Default value of n .
end
% Generate highpass filter.
    Hlp = lpfilter (type, M, N, Do, n) ;
    H = 1- Hlp ;
```

Examples :

1) Gaussian high-pass filter:

```
f = imread ('moon. tif');
PQ= paddedsize (size(f));
Do = 0.05* PQ(1) ;
H = hpfilter ('gaussian', PQ(1), PQ(2), Do);
g = dftfilt (f, H) ;
subplot (3, 1, 1) ;
imshow (f, [ ]);
Subplot (3, 1, 2) ;  imshow (fftshift (H) , [ ]);
Subplot (3, 1, 3) ;  imshow (g , [ ]);
```

2) filtering RGB images using Gaussian high-pass filter:

```
football = imread ('football. jpg');
football = football( :, :, 1) ; only red component
PQ= paddedsize ( size( football)) ;
Do = 0.05 * PQ(1) ;
H= hpfilter ('gaussian', PQ(1), PQ(2) ,Do) ;
F= fft2( double( football) , size(H, 1), size(H,2)) ;
HPF_ football = real (ifft2 (H.*F)) ;
HPF_ football = HPF- football (1: size (football,1), 1:size (football,2))
Fc = fftshift (F) ;
S2 = Log ( 1+ abs ( Fc)) ;
Subplot (3, 1, 1) ;  imshow (football, [ ]) ;
Subplot ( 3, 1, 2) ;  imshow( HPF-football , [ ]) ;
Subplot (3, 1, 3) ; imshow ( S2 , [ ]) ;
```

Wire frame and Surface plotting

3-D wire frame and surface plots are frequently used for Visualizing the transfer function of 2-D filters.

Matlab functions :

1) mesh function :

a- mesh (H) :

this function draws a wireframe for x = 1 : M and y = 1 : N, where [M, N) = size(H).

b- mesh(H(1:K:end, 1:K:end))

if M and N are large, we plot every Kth point (40 - 60) subdivisions along each axis provide a good balance between resolution and appearance.

2) Colormap function :

Colormap ([0 0 0]) : or : Colormap(gray); - - -

this function sets the wireframe to black.

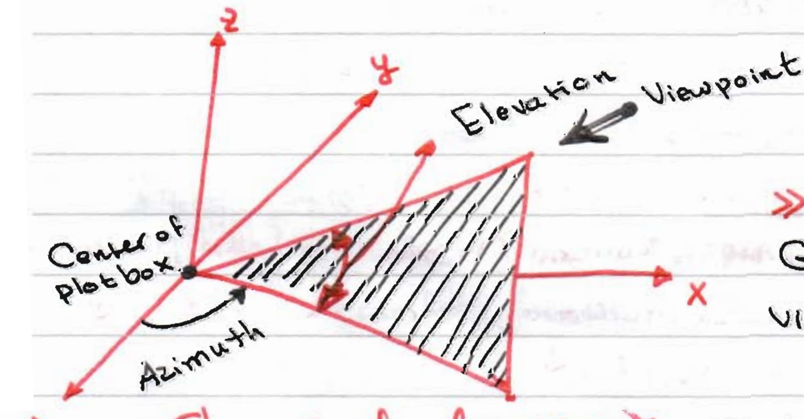
3) grid off %: turns off the grid.

4) axis off %: turns off the axis.

5) View function : % Controls the Viewing point (location

View (az , el) % of the observer)

az - azimuth angle, el - elevation angle (both in degrees) as show in Figure :



The default values:

az = -37.5°

el = 30 .

>> [az el] = View ;

Get the current Viewing geometry.

6) surf function : plots a function as a Surface instead of wireframe, We can use function Colormap, grid, axis and view with it, Syntax ⇒ Surf(H).

Examples:

1) wireframe plotting :

a) Gaussian Lowpass filter:

```
H = fftshift ( lpfilter ('gaussian', 500, 500, 50));
mesh( H( 1:10:500, 1:10: 500)) ;
Colormap ( [0 0 0] ) ;
axis off
grid off
View (-25, 30) ;
```

b) Butterworth lowpass filter:

```
H = fftshift (lpfilter ('btw', 500, 500, 50));
figure, mesh (H (1:10:500, 1: 10: 500)) ;
Colormap ( gray) ;
axis off
grid off
axis ([0 50 0 ·50 0 1])
figure , imshow (H) ;
```

2) Surface plotting :

a) Ideal High pass filter:

```
H = fftshift( hpfilter ('ideal', 500, 500, 50));
Surf ( H(1:10:500 , 1:10:500));
axis ( [0 50 0 50 0 1]);
Colormap ( gray) ;
grid off , axis off
```

b) to plot an analytic function of two Variables, We Use meshgrid function to generate the Coordinate Values and from these Values We generate the discrete matrix to use in mesh or surf.

to plot $f(x,y) = x \cdot e^{(-x^2 - y^2)}$ from -2 to 2 with increment 0.1

```
>> [Y, X] = meshgrid (-2: 0.1 : 2 , -2: 0.1:2);
>> Z = X.* exp (- X.^2 - Y.^2) ;
>> mesh(Z) ;
```