EXPERIMENT NO. 3
PERFORM AND COMPARE FREQUENCY DOMAIN FILTERS

EXPERIMENT NO. 3

AIM: To implement ideal, butterworth, Gaussian low pass filters in frequency domain and compare their performances

OBJECTIVES:

1. To apply and compare performance averaging filters of various size and cutoff frequency
2. To understand to convert image from spatial domain to frequency domain.
3. To see the frequency spectrum of the image.
4. To understand the concept of frequency domain filtering.

EQUIPMENTS/SOFTWARE: SCILAB 6.0.0

THEORY: Using low pass filters -

$$\frac{1}{9}\times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16}\times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

The second mask, shown in Figure, yields a so-called weight average, thus giving more importance (weight) to some pixels at the expense of others.

Frequency domain filtering-

The reason for doing the filtering in the frequency domain is generally because it is computationally faster to perform two 2D Fourier transforms and filter multiplication in this domain than to perform convolution in the image (spatial) domain. Also convolution becomes more complex in spatial domain as filter size increases.

The transfer function of a Butterworth LPF of order ‘n’, and with cutoff frequency at a distance D_0 from its origin is defined as

$$H(u,v) = \frac{1}{1 + [\frac{D(u,v)}{D_0}]^{2n}}$$

The transfer function of a Gaussian LPF with cutoff frequency at a distance D_o from its origin is defined as

$$H(u,v) = e^{\left\{-D^2(u,v)\big/_{2Do^2}\right\}}$$

ALGORITHM:

Spatial domain filtering-

1. Read the image.
2. Define LPF masks
3. Run the mask on the image.
4. See result of the filtering

Frequency domain filtering-

1. Read the input image and its size.
2. Obtain the padding parameters P and Q. Typically, we select P=2M and Q=2N
3. Form a padded image, $f_p(x,y)$ of size PxQ by appending the necessary number of zeroes to f(x,y)
4. Multiply $f_p(x,y)$ by $(-1)^{x+y}$ to center its transform
5. Obtain the Fourier transform of the image
6. Generate a Butterworth and Gaussian filter function, H1 and H2, the same size as the image (PxQ)
7. Multiply the transformed image by the filter:
   G1=H1.*F; G2=H2.*F;
8. Obtain the real part of the inverse FFT of G.

FUNCTIONS USED (MATLAB / SCILAB):

1. imread
2. double

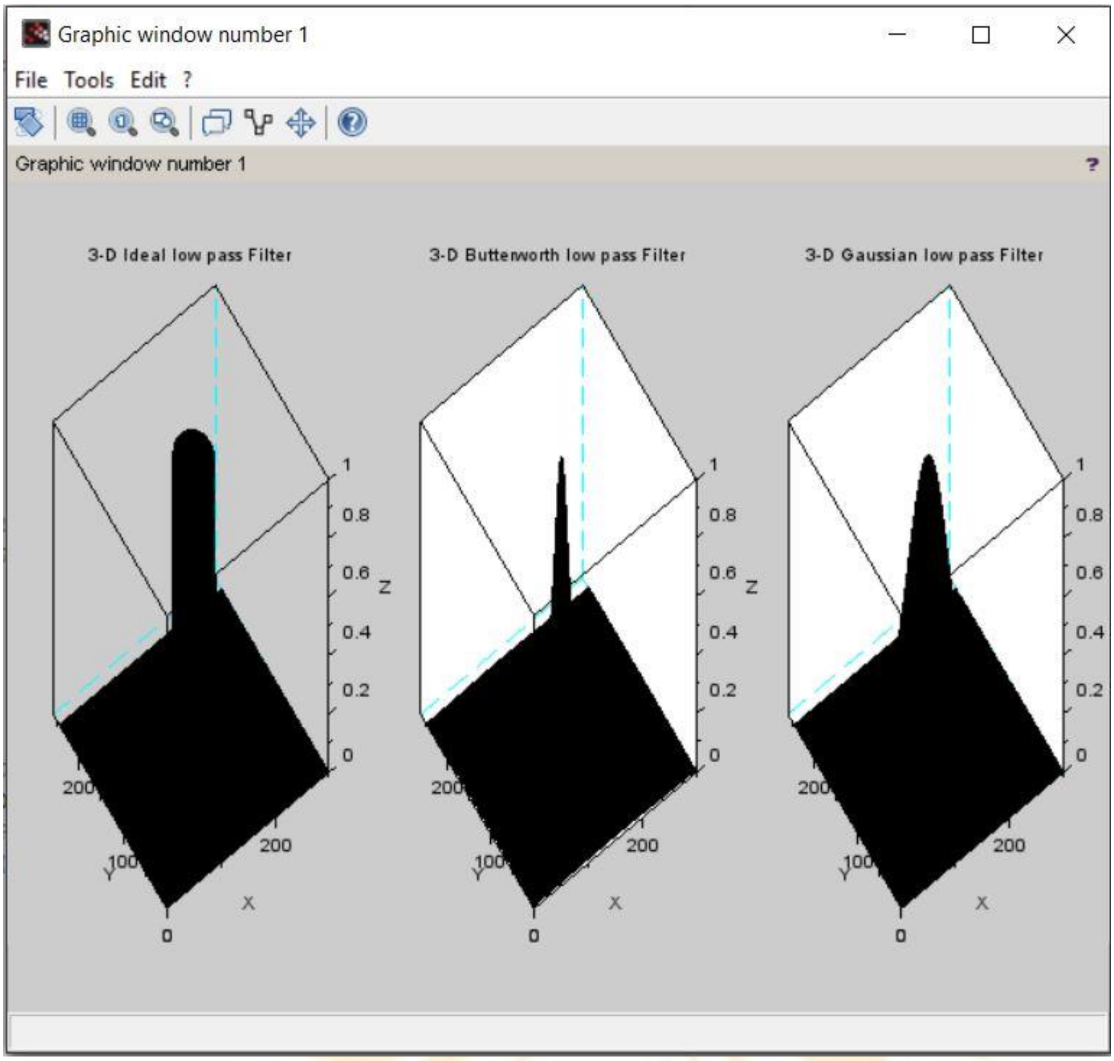
3. fft2
4. ifft2
5. real
6. imshow

PROGRAM:

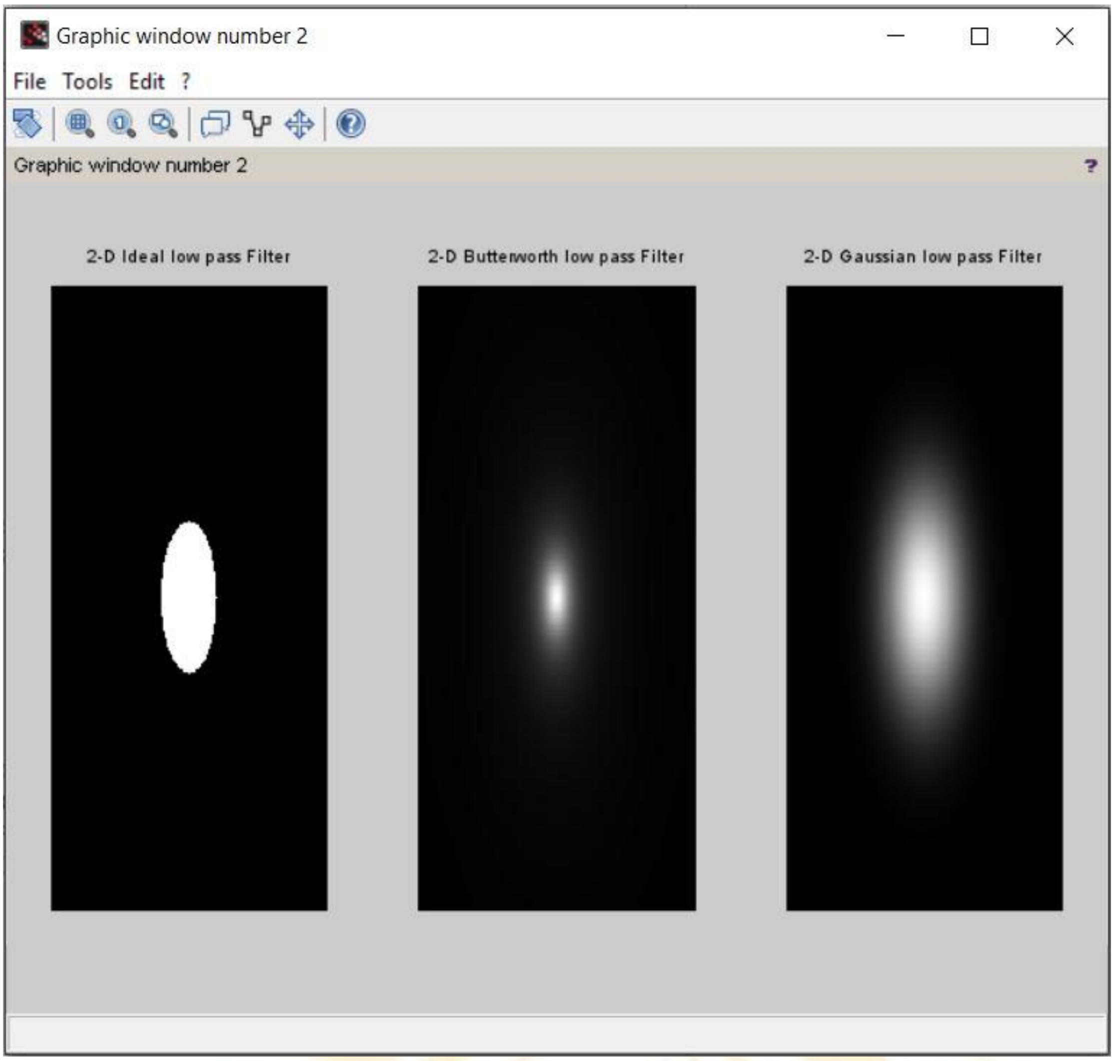
```
/// Frequency Domain Filters

///Dikshita Kambri 118A2044 A

clc;
clear all;
im=imread("C:\Users\hp\Documents\Image Processing-Scilab\Images\coins.png");
im=double(im);
[r c]=size(im);
I1=fft2(im);
Is=fftshift(I1);

// Defining filters
D0=30;
n=5;
Hi=zeros(r,c);
Hb=zeros(r,c);
Hg=zeros(r,c);
for i=1:r
for j=1:c
   D=sqrt(((r/2)-i)^2+((c/2)-j)^2);
   /// IDEAL LOW PASS FILTER
   if D<=D0 then
      Hi(i,j)=1;
   else
      Hi(i,j)=0;
   end

   /// BUTTERWORTH LOW PASS FILTER
   Hb(i,j)=(1/(1+(D/D0)^2*n));
   /// GAUSSIAN LOW PASS FILTER
   Hg(i,j)=exp(-D^2/(2*D0^2));
end
end

Gi=Is.*Hi;
Gb=Is.*Hb;
Gg=Is.*Hg;
gi=abs(ifft(Gi));
gb=abs(ifft(Gb));
gg=abs(ifft(Gg));
figure(1)
subplot(1,3,1)
mesh(Hi);
title('3-D Ideal low pass Filter');
subplot(1,3,2)
```

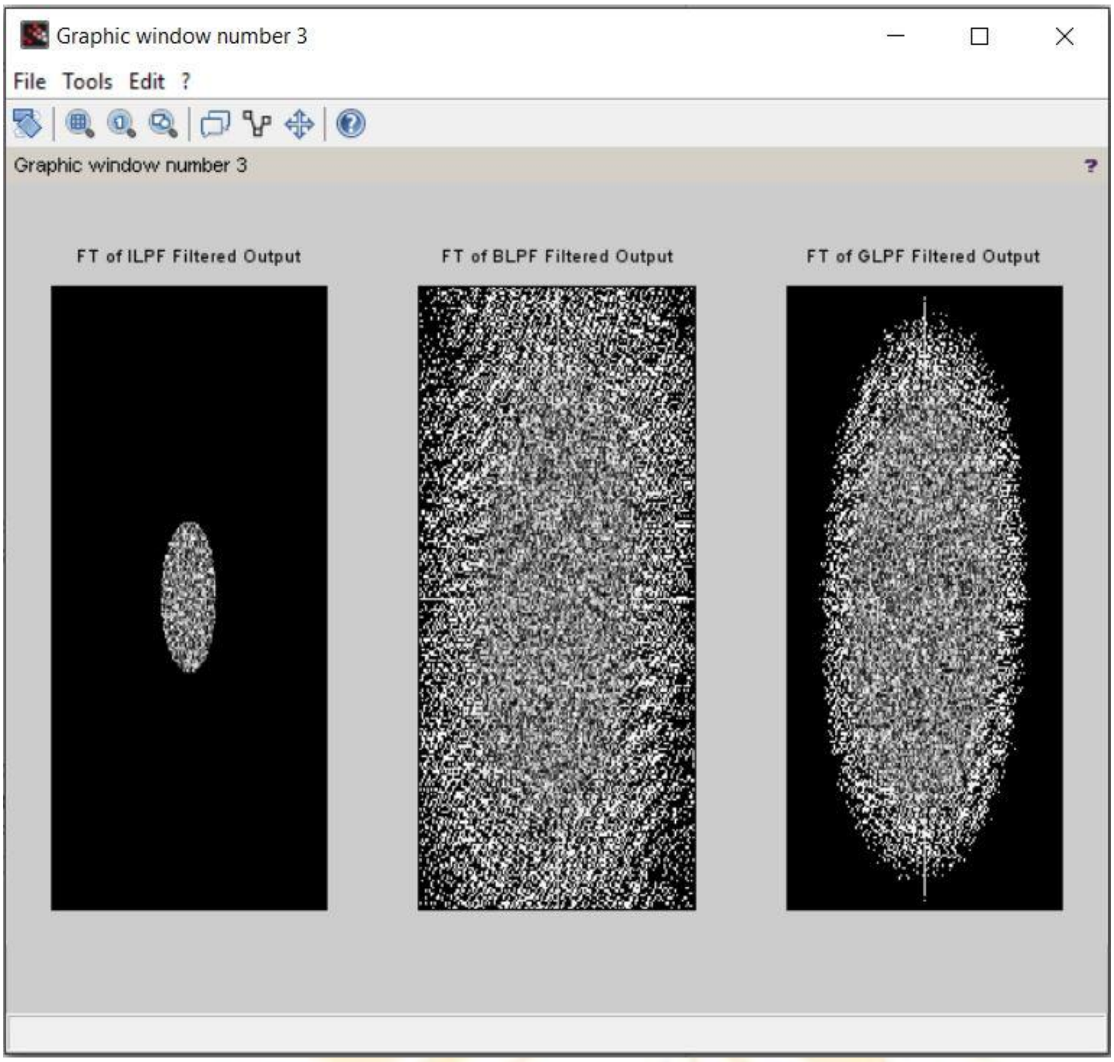
```
mesh(Hb);
title('3-D Butterworth low pass Filter');
subplot(1,3,3)
mesh(Hg);
title('3-D Gaussian low pass Filter');

figure(2)
subplot(1,3,1)
imshow(uint8(255*Hi));
title('2-D Ideal low pass Filter');
subplot(1,3,2)
imshow(uint8(255*Hb));
title('2-D Butterworth low pass Filter');
subplot(1,3,3)
imshow(uint8(255*Hg));
title('2-D Gaussian low pass Filter');

figure(3)
subplot(1,3,1)
imshow(uint8(Gi));
title('FT of ILPF Filtered Output');
subplot(1,3,2)
imshow(uint8(Gb));
title('FT of BLPF Filtered Output');
subplot(1,3,3)
imshow(uint8(Gg));
title('FT of GLPF Filtered Output');

figure(4)
subplot(1,3,1)
imshow(uint8(gi));
title('ILPF Filtered image');
subplot(1,3,2)
imshow(uint8(gb));
title('BLPF Filtered image');
subplot(1,3,3)
imshow(uint8(gg));
title('GLPF Filtered image');
```

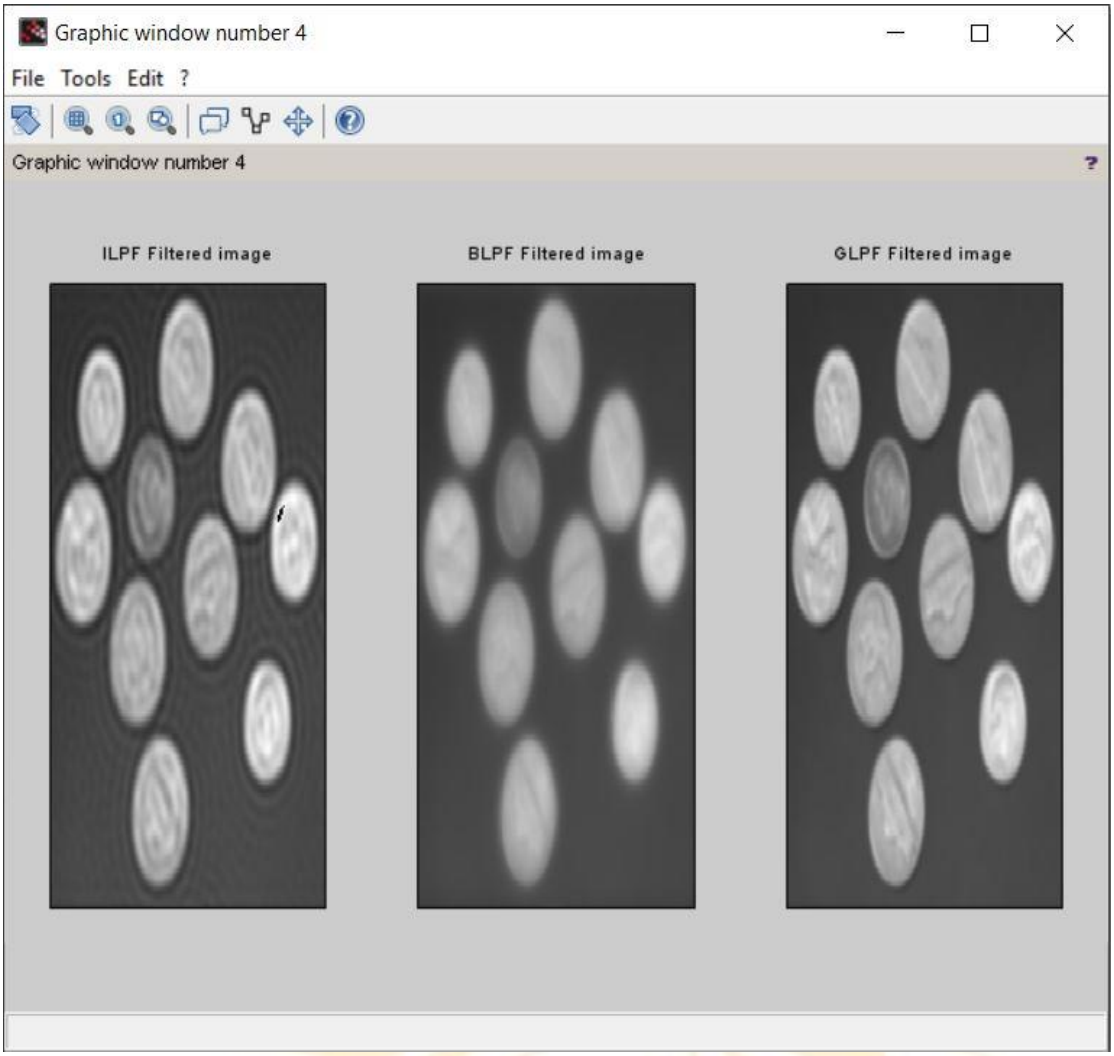
OUTPUT:



Graphic window number 1
File Tools Edit ?
Graphic window number 1
3-D Ideal low pass Filter
3-D Butterworth low pass Filter
3-D Gaussian low pass Filter
Z
X
Y
0
200
100
1
0.8
0.6
0.4
0.2


Graphic window number 2
File Tools Edit ?
Graphic window number 2
2-D Ideal low pass Filter
2-D Butterworth low pass Filter
2-D Gaussian low pass Filter


Graphic window number 3
File Tools Edit ?
Graphic window number 3
FT of ILPF Filtered Output
FT of BLPF Filtered Output
FT of GLPF Filtered Output



CONCLUSION:

We converted the image from spatial domain to frequency domain and observed the frequency spectrum of image.