THE UNIVERSITY OF TEXAS AT ARLINGTON, TEXAS
DEPARTMENT OF ELECTRICAL ENGINEERING

EE 5356

DIGITAL IMAGE PROCESSING

PROJECT # 9

by

SOUTRIK MAITI

1001569883

Presented to

Dr. K.R.RAO

April 5, 2019

High Pass Low Pass Filter

MATLAB Code:

```
clc;
clear all;
close all;

%% Reading the image

img = imread('lena512.bmp');
[m1,n1]=size(img);
figure();
subplot(1,2,1);
imshow(uint8(img));
title('original image');

%% Performing the DFT on the image read

img =im2double(img);
dft_img = fft2(img);
subplot(1,2,2);
imshow(log(1 + abs(fftshift(dft_img))));
title('DFT of original image');
saveas(gca,'origin_dft.jpg');

%% Making the 2D mesh and plotting for the FFT of the image

uu = 0:(m1-1);
vv = 0:(n1-1);
ii = find(uu > m1/2);
uu(ii) = uu(ii)-m1;
jj = find(vv > n1/2);
vv(jj)=vv(jj)-n1;
[kk,l]=meshgrid(vv,uu);
DD=sqrt((kk.^2+l.^2));
figure();
mesh(real(fftshift(DD)));
title('FFT of the original image given by D(u,v)');
saveas(gca,'mesh.jpg');



img_hpf  = HPF_img(30,DD);
img_g_lpf = g_LPF_img(16,DD);
img_g_hpf = g_HPF_img(16,DD);

img_b_lpf = b_LPF_img(50,3,DD);
img_b_hpf= b_HPF_img(50,3,DD);
```

```
%% Applying LPF to image and plotting results

img_lpf  = LPF_img(40,DD);
img_dft_LPF = img_lpf.*dft_img;
ifft_img_LPF = real(ifft2(img_dft_LPF));
figure();
subplot(1,2,1);
imshow(log(1+abs(fftshift(img_dft_LPF))),[]);
title('LPF frequency response of image');
subplot(1,2,2);
imshow(ifft_img_LPF,[]);
title('LPF filtered Image');
saveas(gca,'LPF_img.jpg');

%% Applying HPF to image and plotting results

img_dft_HPF = img_hpf.*dft_img;
ifft_img_HPF = real(ifft2(img_dft_HPF));
figure();
subplot(1,2,1);
imshow(log(1+abs(fftshift(img_dft_HPF))),[]);
title('HPF frequency response of image');
subplot(1,2,2);
imshow(ifft_img_HPF,[]);
title('HPF filtered image');
saveas(gca,'HPF_img.jpg');

%% Applying Gaussian LPF to image and plotting the results

img_dft_g_LPF = img_g_lpf.*dft_img;
ifft_img_g_LPF = real(ifft2(img_dft_g_LPF));
figure();
subplot(1,2,1);
imshow(log(1+abs(fftshift(img_dft_g_LPF))),[]);
title('GAUSSIAN LOW PASS freq response');
subplot(1,2,2);
imshow(ifft_img_g_LPF,[]);
title('GAUSSIAN LOW PASS filtered image');
saveas(gca,'g_LPF_img.jpg');


%% Applying Gaussian LPF to image and plotting the results

img_dft_g_HPF = img_g_hpf.*dft_img;
ifft_img_g_HPF = real(ifft2(img_dft_g_HPF));
figure();
subplot(1,2,1);
imshow(log(1+abs(fftshift(img_dft_g_HPF))),[]);
title('GAUSSIAN HIGH PASS freq response');
subplot(1,2,2);
imshow(ifft_img_g_HPF,[]);
```

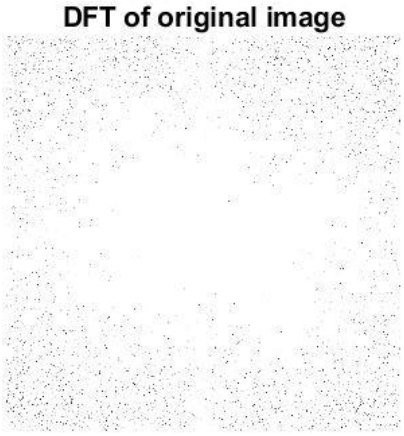
```
title('GAUSSIAN HIGHPASS filtered image');
saveas(gca,'g_HPF_img.jpg');


%% Applying Butterworth LPF to image and plotting the results

img_dft_b_LPF = img_b_lpf.*dft_img;
ifft_img_b_LPF = real(ifft2(img_dft_b_LPF));
figure();
subplot(1,2,1);
imshow(log(1+abs(fftshift(img_dft_b_LPF))),[]);
title('BUTTERWORTH LPF freq response');
subplot(1,2,2);
imshow(ifft_img_b_LPF,[]);
title('BUTTERWORTH LPF filtered image');
saveas(gca,'b_LPF_img.jpg');

%% Applying Butterworth HPF to image and plotting the results

img_dft_b_HPF = img_b_hpf.*dft_img;
ifft_img_b_HPF = real(ifft2(img_dft_b_HPF));
figure();
subplot(1,2,1);
imshow(log(1+abs(fftshift(img_dft_b_HPF))),[]);
title('BUTTERWORTH HPF freq response');
subplot(1,2,2);
imshow(ifft_img_b_HPF,[]);
title('BUTTERWORTH HPF filtered image');
saveas(gca,'b_HPF_img.jpg');



%% Algorithm for Low Pass Filter

function LPF = LPF_img(d_0,d_d)
mm=512;
nn=mm;
for uu = 1:mm
    for vv = 1:nn
        if(d_d(uu,vv) <= d_0)
            LPF(uu,vv) = 1;
        else
            LPF(uu,vv) = 0;
        end
    end
end
%% 3D plot of ideal LPF
figure();
mesh(fftshift(LPF)),
title('Low Pass Filter 3D');
saveas(gca,'LPF_3D.jpg');
end
```

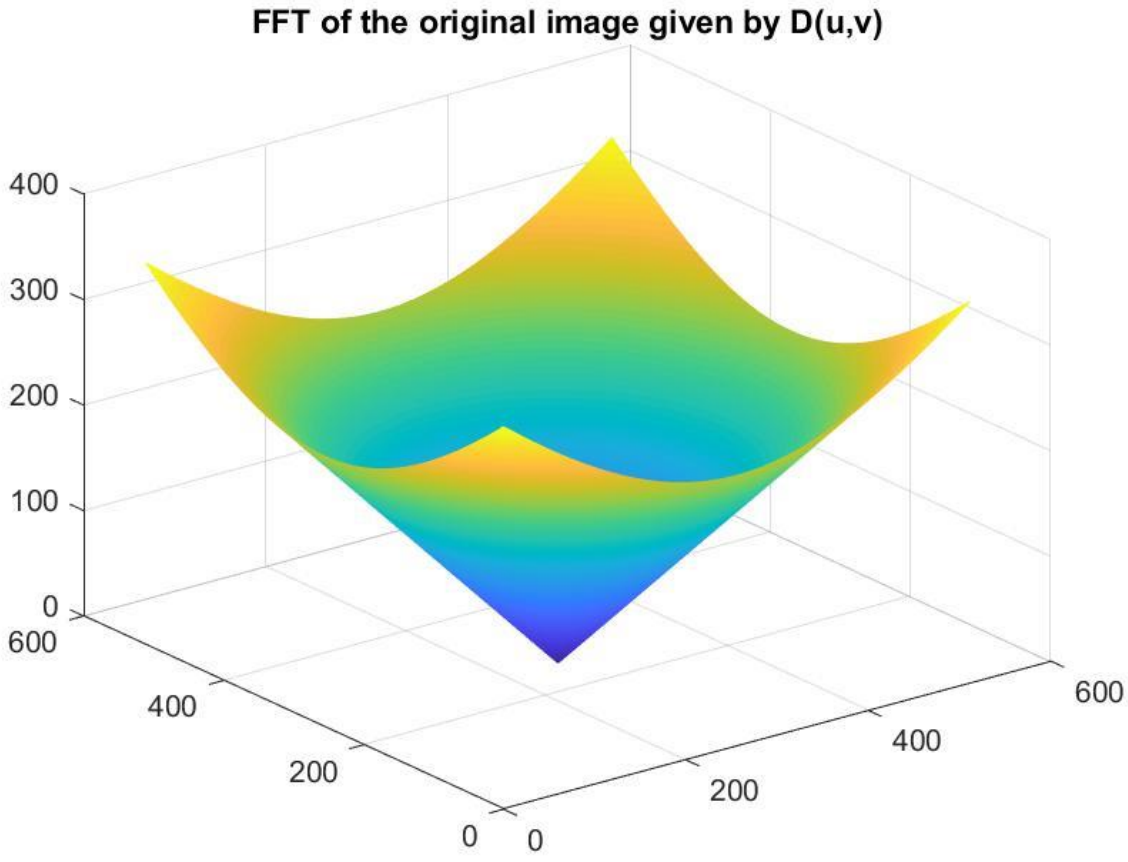
```
%% Algorithm for High Pass Filter

function HPF = HPF_img(d_0,d_d)
mm=512;nn=mm;
for uu = 1:mm
    for vv = 1:nn
        if(d_d(uu,vv)<=d_0)
            HPF(uu,vv) = 0;
        else
            HPF(uu,vv) = 1;
        end
    end
end
%% 3D plot of ideal HPF

figure();
mesh(fftshift(HPF)),
title('High Pass Filter 3D');
saveas(gca,'HPF_3D.jpg');
end

%% Algorithm for Gaussian Low Pass Filter

function g_LPF = g_LPF_img(sgma,d_d)
mm=512;nn=mm;
for uu = 1:mm
    for vv = 1:nn
        g_LPF(uu,vv) = exp(-1*(d_d(uu,vv)^2)/(2*sgma^2));
    end
end
%% 3D plot of Gaussian Low Pass Filter

figure();
mesh(fftshift(g_LPF)),
title('Gaussian Low Pass Filter 3D');
saveas(gca,'g_LPF_3D.jpg');
end


%% Algorithm for Gaussian High Pass Filter

function g_HPF = g_HPF_img(sgma,d_d)
mm=512;nn=mm;
for uu = 1:mm
    for vv = 1:nn
        g_HPF(uu,vv) = 1-exp(-1*(d_d(uu,vv)^2)/(2*sgma^2));
    end
end
%% 3D plot of Gaussian High Pass Filter

figure();
```

```
mesh(fftshift(g_HPF)),
title('Gaussian HPF');
saveas(gca,'g_HPF_3D.jpg');
end

%% Algorithm for Butterworth Low Pass Filter

function b_LPF = b_LPF_img(d_0,o,d_d)
mm=512;nn=mm;
for uu=1:mm
    for vv=1:nn
        b_LPF(uu,vv) = 1/(1+(d_d(uu,vv)/d_0)^(2*o));
    end
end
%% 3D plot of Butterworth Low Pass Filter
figure();
mesh(fftshift(b_LPF)),
title('Butterworth Low Pass Filter');
saveas(gca,'b_LPF_3D.jpg');
end


%% Algorithm for Butterworth High Pass Filter

function b_HPF = b_HPF_img(d_0,o,d_d)
mm=512;nn=mm;
for uu = 1:mm
    for vv = 1:nn
        b_HPF(uu,vv) = 1/(1+(d_0/d_d(uu,vv))^(2*o));
    end
end
%% 3D plot of Butterworth High Pass Filter

figure();
mesh(fftshift(b_HPF)),
title('Butterworth High Pass Filter');
saveas(gca,'b_HPF_3D.jpg');
end
```

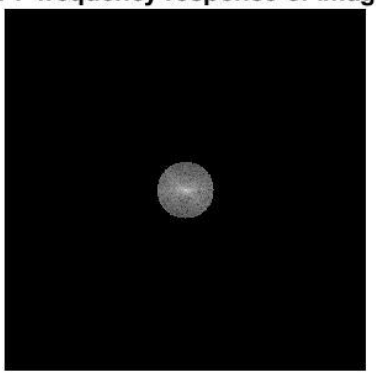
Results:


original image


DFT of original image

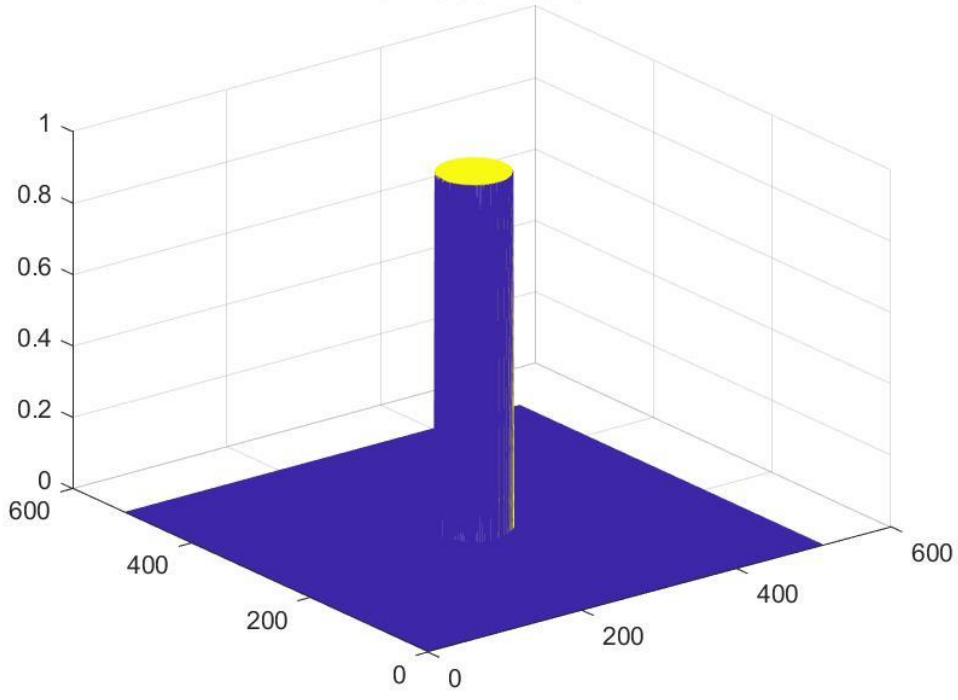

FFT of the original image given by D(u,v)
400
300
200
100
0
600
400
200
0
0
200
400
600

LPF frequency response of image

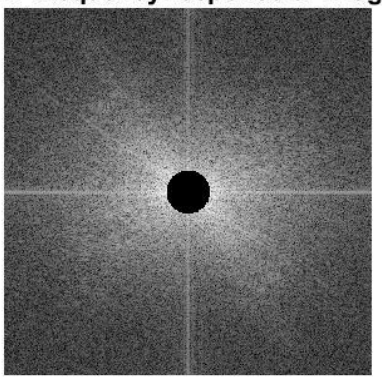


LPF filtered Image

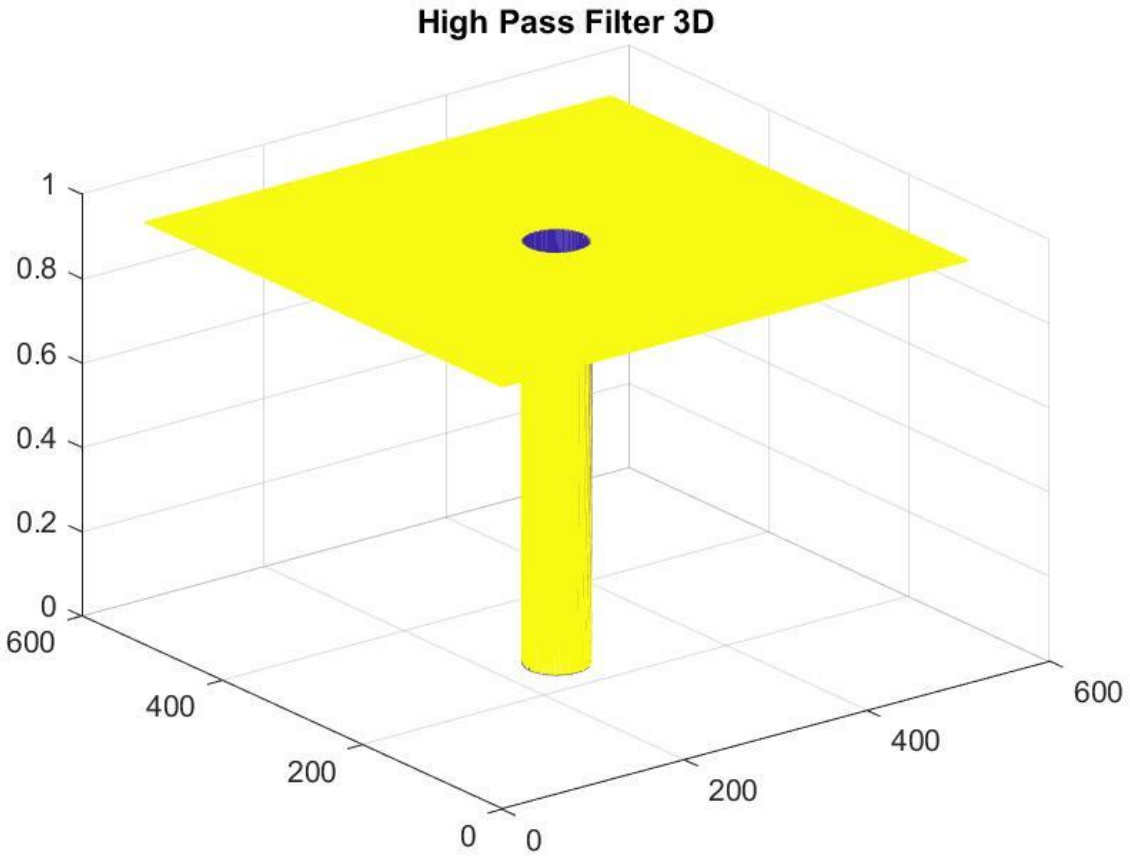
Low Pass Filter 3D



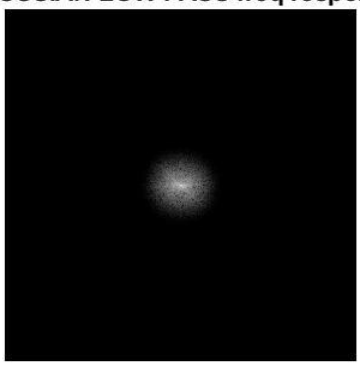
HPF frequency response of image



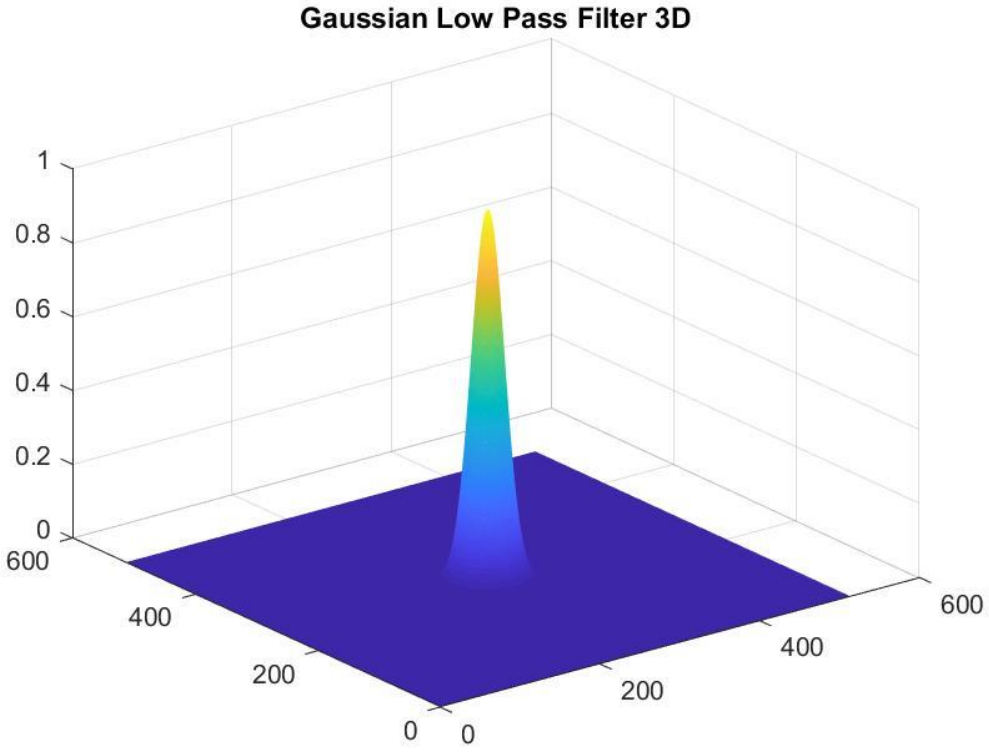
HPF filtered image



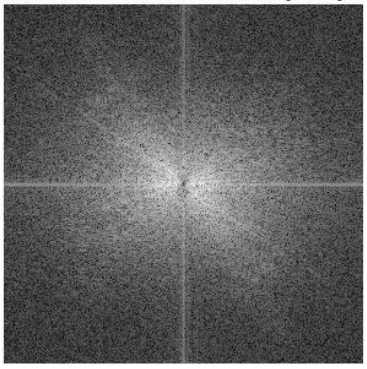
GAUSSIAN LOW PASS freq response



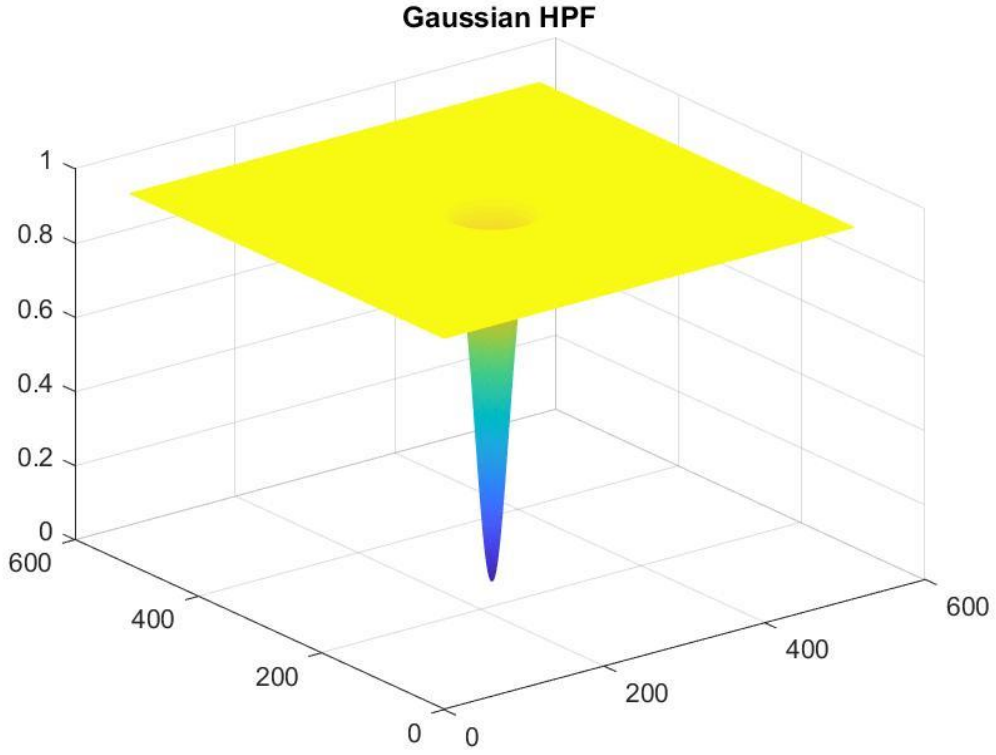
GAUSSIAN LOW PASS filtered image



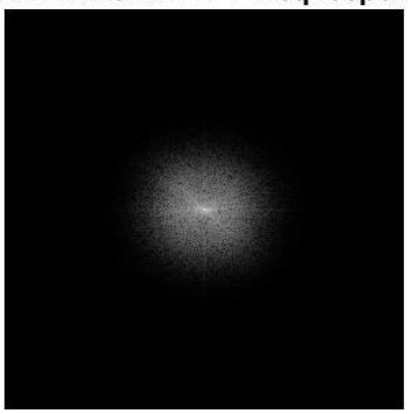
GAUSSIAN HIGH PASS freq response



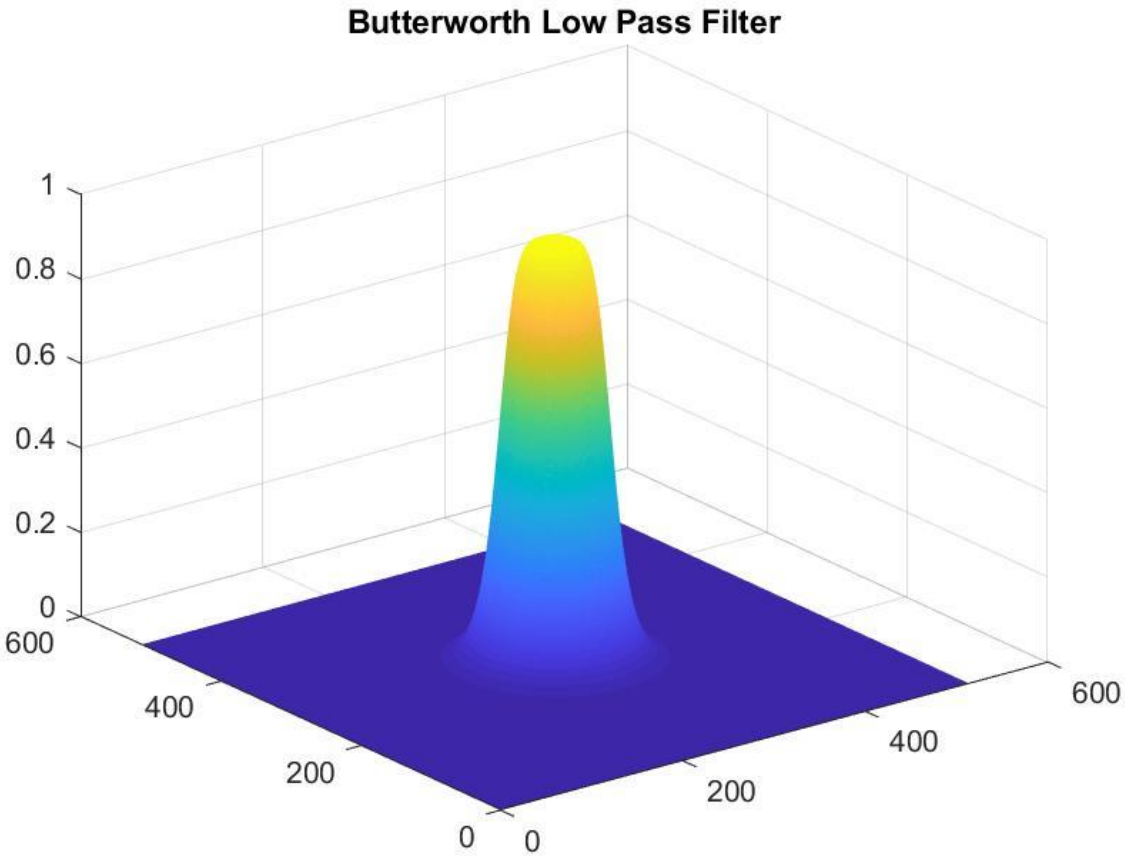
GAUSSIAN HIGHPASS filtered image



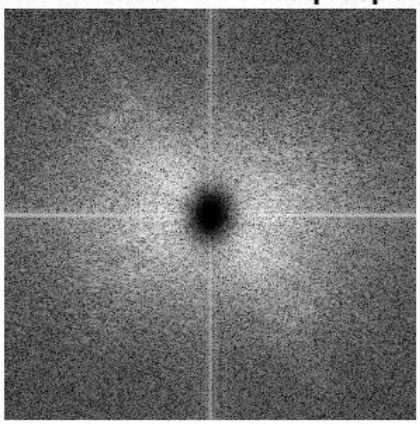
BUTTERWORTH LPF freq response



BUTTERWORTH LPF filtered image

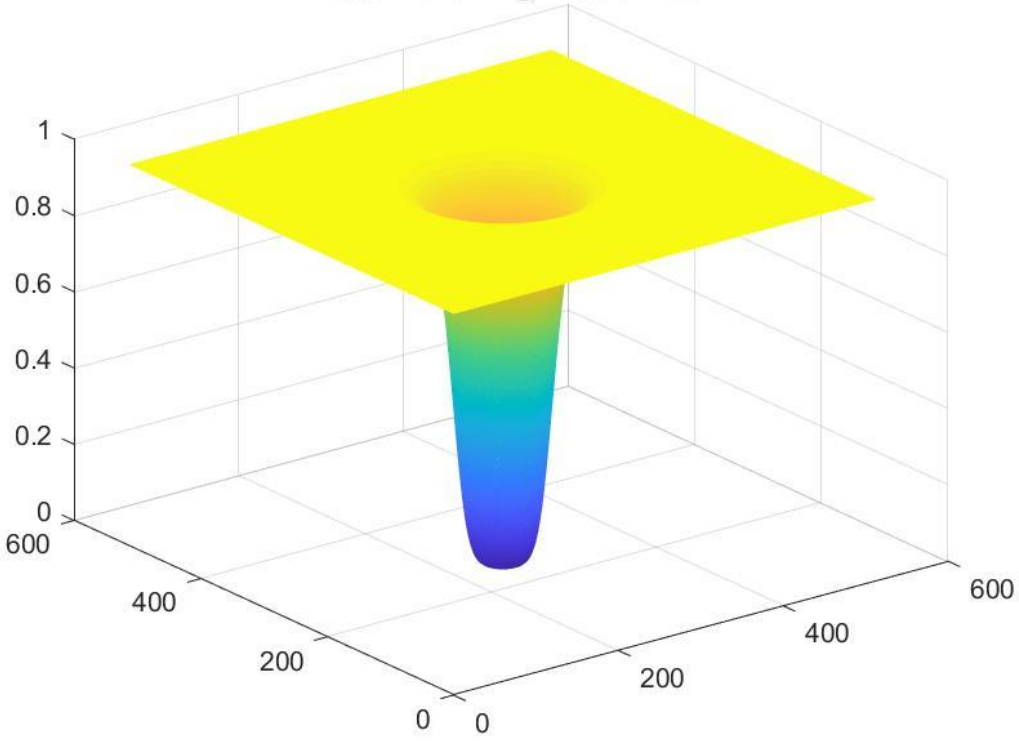


BUTTERWORTH HPF freq response



BUTTERWORTH HPF filtered image

Butterworth High Pass Filter



Conclusion:

- *DFT is performed on the image loaded into the workspace*
- *The algorithms of the various filters have been written in the form of functions*
- *Inverse DFT is performed after the various filters are applied to the image in the frequency domain*
- *High Pass filter makes the edges more prominent*
- *Low Pass filter doesn't make any drastic changes*
- *Therefore LPF are much more efficient than HPF*