

**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

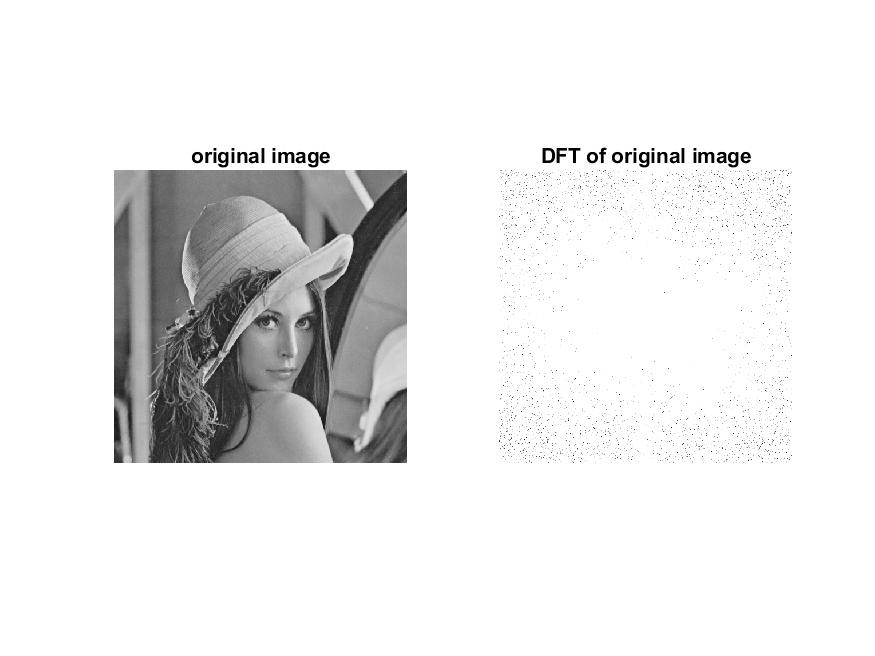
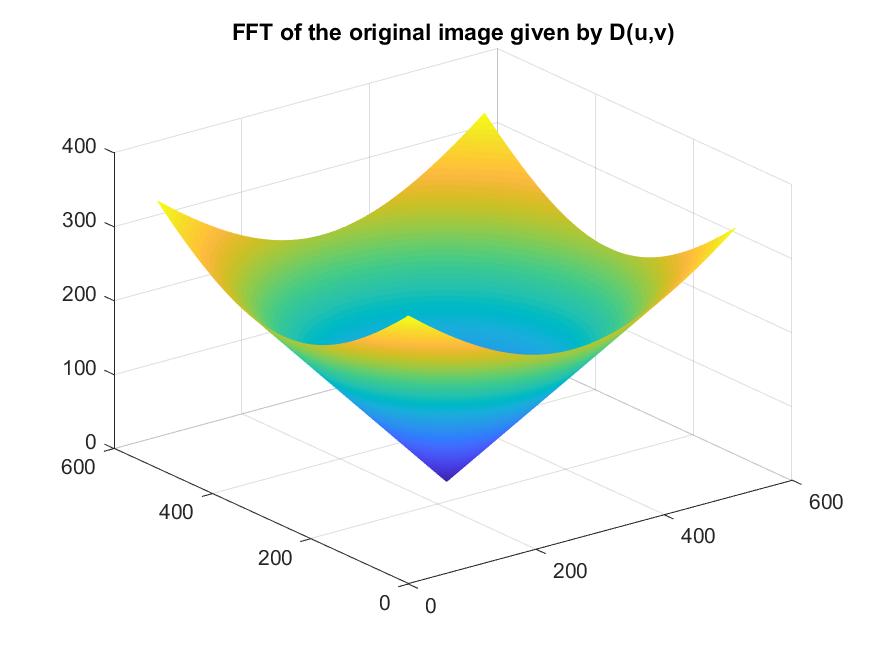
figure();

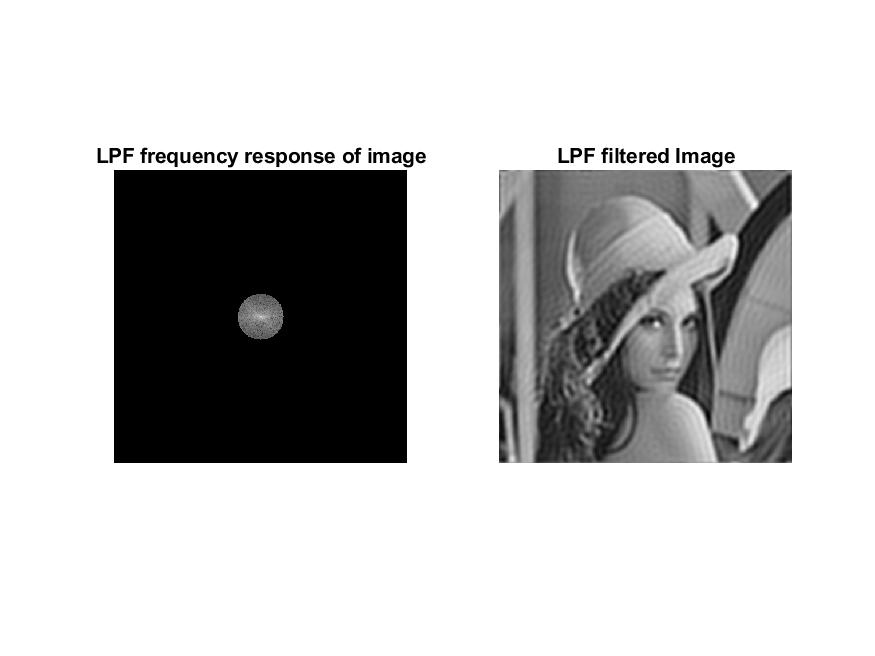
mesh(fftshift(b\_HPF)),

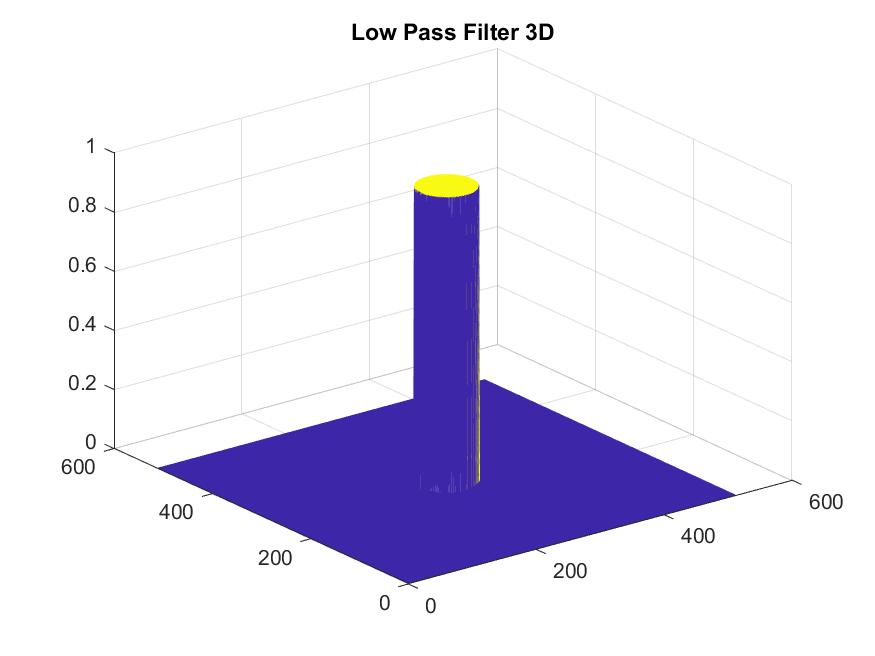
title('Butterworth High Pass Filter');

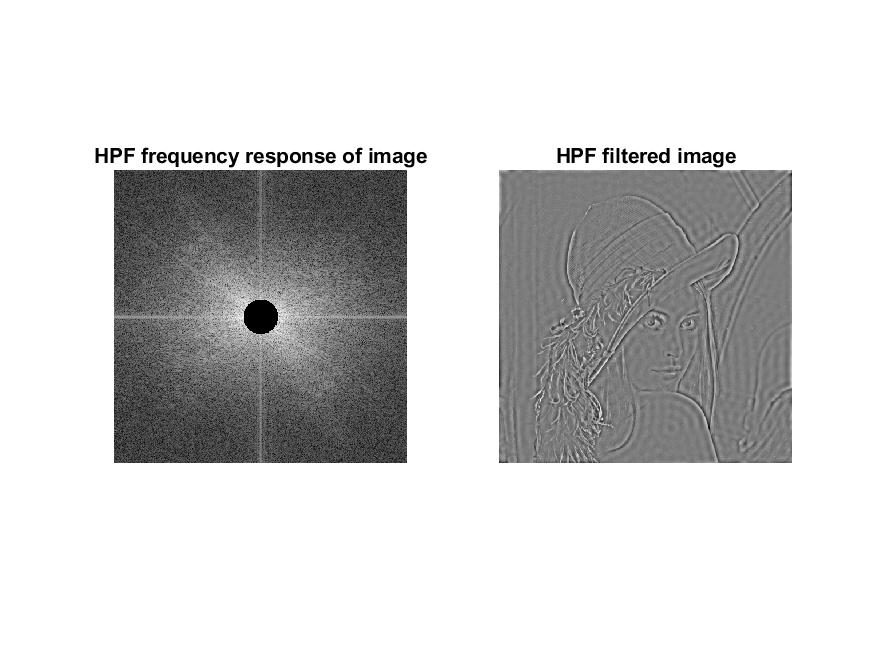
saveas(gca,'b\_HPF\_3D.jpg');

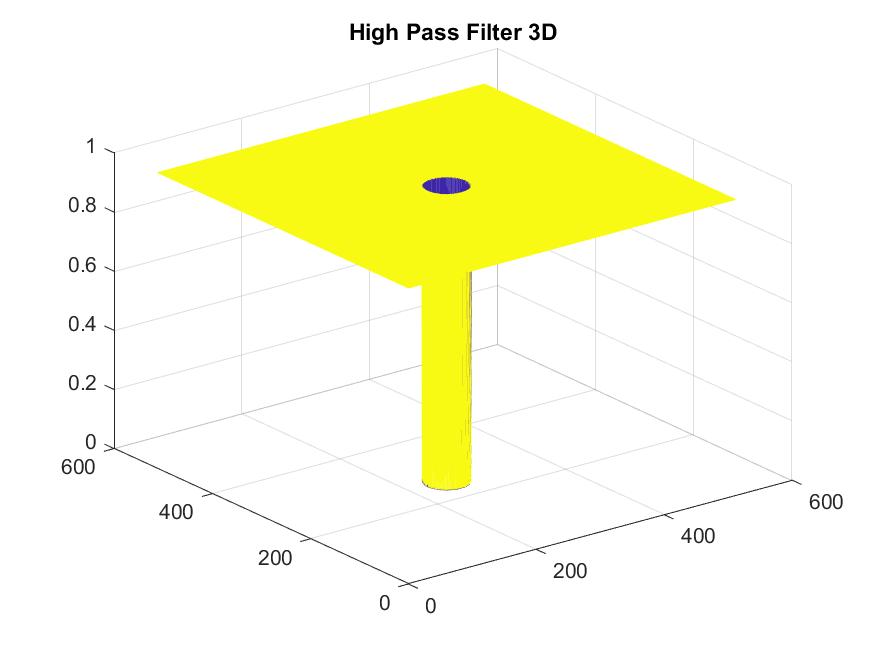
end

***Results:***

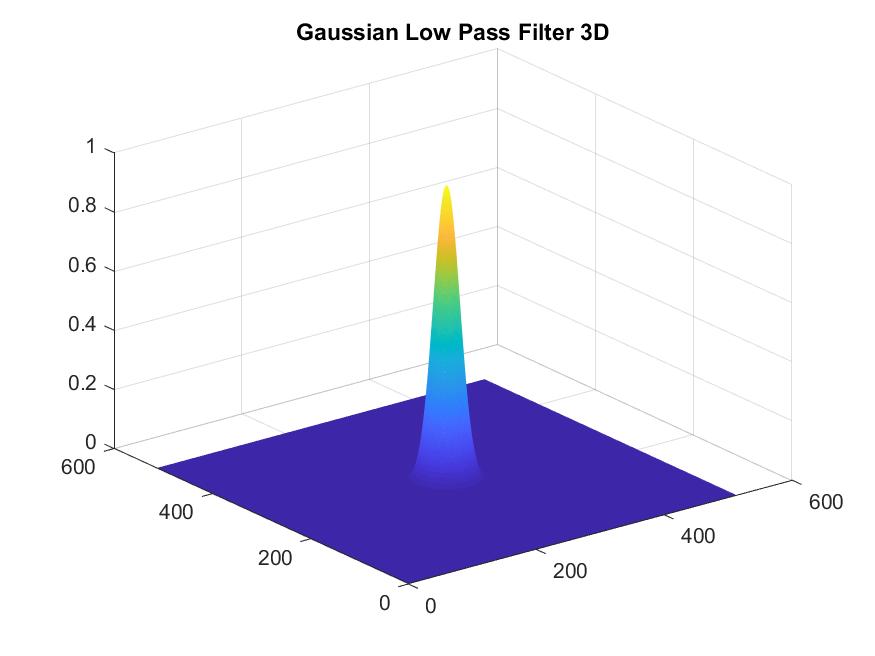


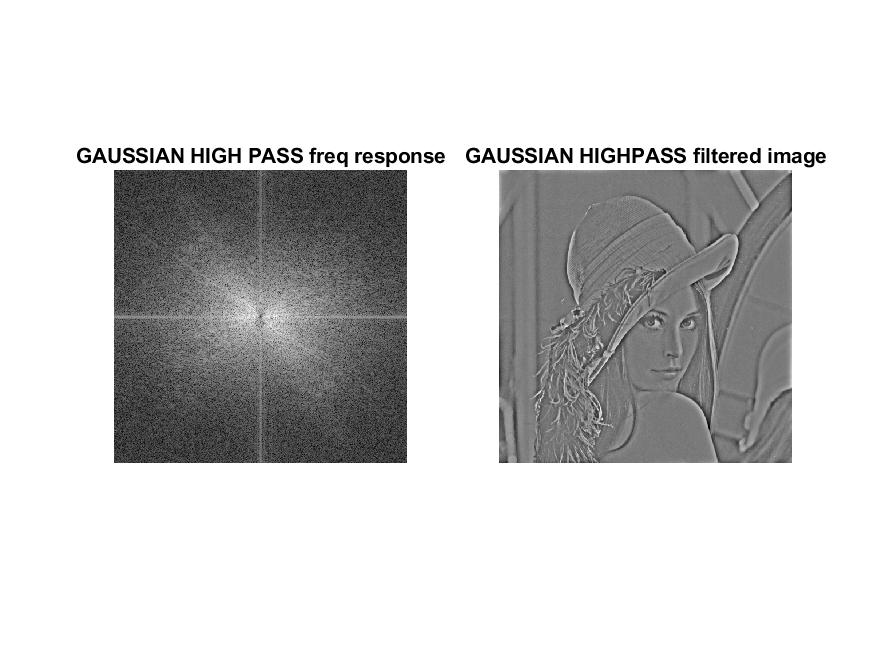


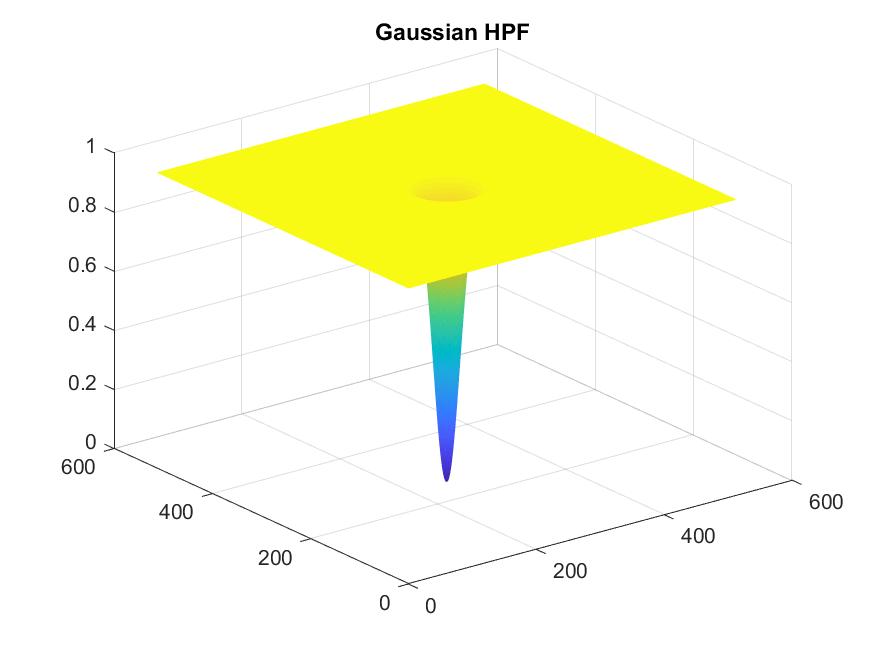


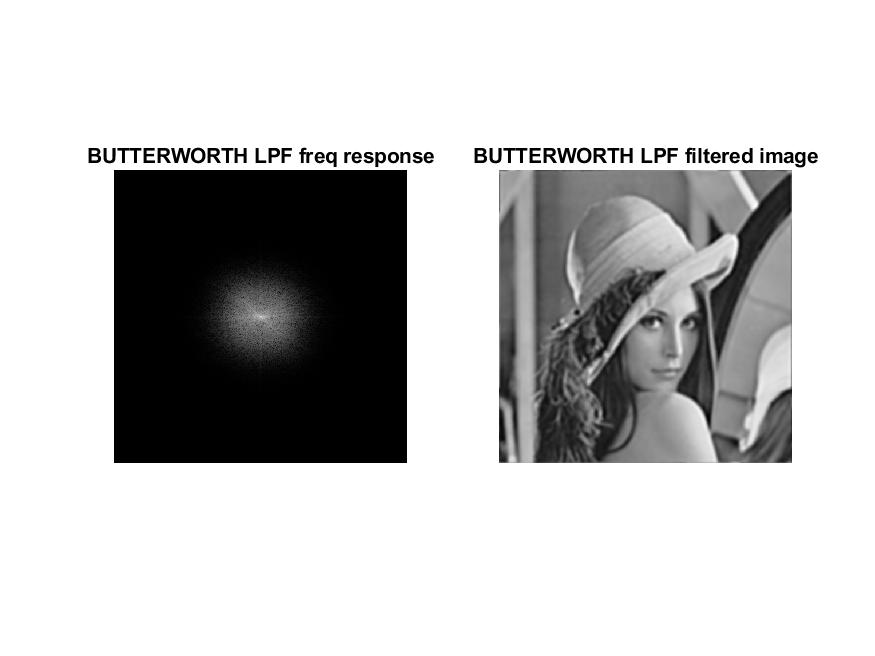
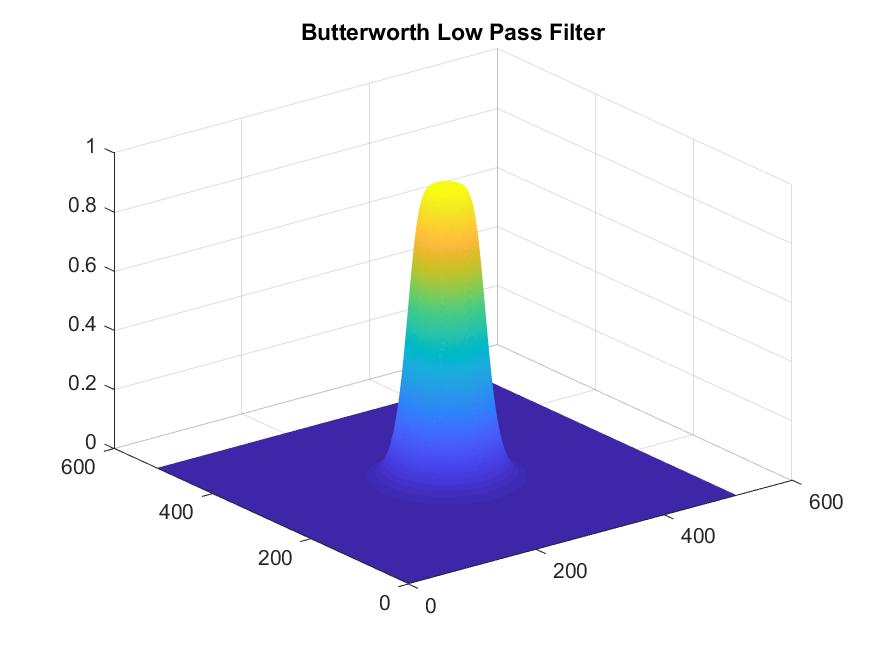
******

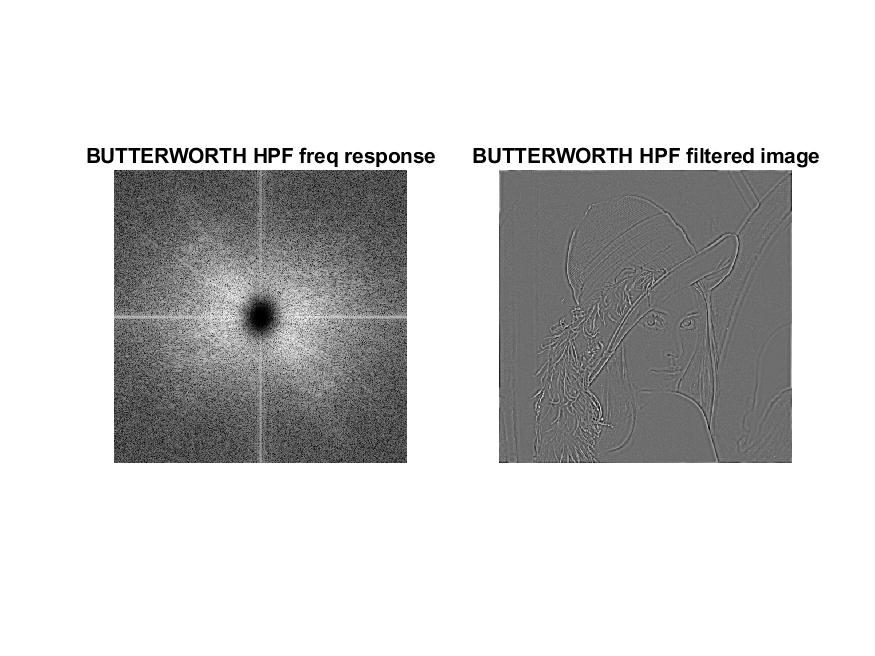
******

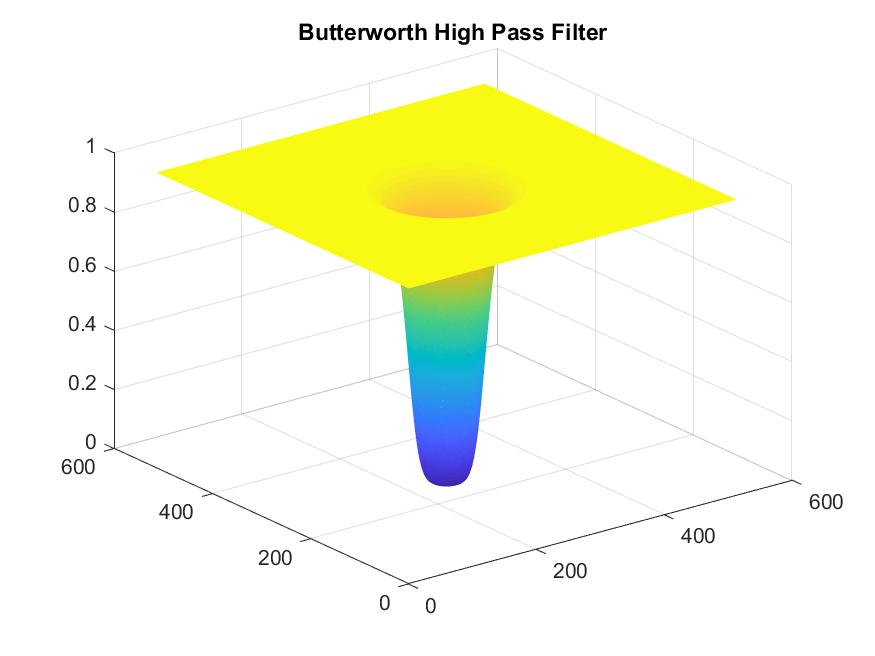
******

******

******

******

******

******

***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*