LAB 8

Name:Kartikay Agrawal

Reg No:2148064

1. Write a program to demonstrate filtering in frequency domain (Both Low and High pass filtering using Ideal, Butterworth and Gaussian )

img=imread('cameraman.tif'); % reading the image

[M,N]=size(img); % Saving the the rows of X in M and columns in N

FFT=fftshift(fft2(img));

disp("Frequency Domain Filtering");

choice = input('\n1.Low-Pass Filtering\n2.High-Pass Filtering\n3.Exit\n\nEnter Choice: ');%menu driven options for users

switch choice

case 1

flag = 1;

while(flag==1)

clc;

disp('\*Low-Pass Filtering\*');

ch = input('\nType of Filter\n1.Ideal\n2.Butterworth\n3.Gaussian\n4.Back\n\nEnter Choice: ');%Choose type of filter

if(ch == 1)

f=input('Enter cut-off value(0-1) : '); % Giving cut-off value for input of

f1=f\*sqrt(M^2+N^2); % to put pixel in centre of an image

H1=zeros(M,N); % complete h1(filter) is zero initially (same dimension as image)

%Ideal Low Pass Filter

for i=1:M %iterates over the M rows

for j=1:N %iterates over the N columns for each row

if(sqrt(abs(i-M/2)^2+abs(j-N/2)^2) < f1) % calculates the euclidean distance and compares with f1

H1(i,j)=1; % assigns 1 if the distance is less than f if not it assigns zero

end

end

end %Ideal low pass filter H1 is created

G=H1.\*FFT; % Multiplying the Fourier transformed image with H1 (the filter)

G=ifftshift(G); % Inverse fourier transform

img2=uint8(real(ifft2(G))); % ideal low pass filtered image (output)

% displaying the images

figure

imshowpair(img,img2,"montage");title('Original image and Filtered image (ILP Filter)');xlabel("Cutoff - "),xlabel(f)

elseif(ch == 2)

f=input('Enter cut-off value(0-1) : '); % takes input for cut-off value

f1=f\*sqrt(M^2+N^2); % to put pixel in centre of an image

H2=zeros(M,N); % complete h1(filter) is zero initially (same dimension as image)

n = input('Input n value: ');

%Butter worth Low Pass Filter

for i=1:M %iterates over the M rows

for j=1:N %iterates over the N columns for each row

D = sqrt(abs(i-M/2)^2 + abs(j-N/2)^2);

H2(i,j)= 1/(1+(D/f)^(2\*n)); % assigns 1 if the distance is less than f if not it assigns zero

end

end %Butter worth low pass filter H1 is created

G=H2.\*FFT; % Multiplying the Fourier transformed image with H1 (the filter)

G=ifftshift(G); % Inverse fourier transform

img2=uint8(real(ifft2(G))); % butter worth low pass filtered image (output)

% displaying the images

figure

imshowpair(img,img2,"montage");title('Original image and Filtered image (BLP Filter)');xlabel("Cutoff - "),xlabel(f)

elseif(ch == 3)

f=input('Enter cut-off value(0-1) : '); % takes input for cut-off value

f1=f\*sqrt(M^2+N^2); % to put pixel in centre of an image

H3=zeros(M,N); % complete h1(filter) is zero initially (same dimension as image)

%Guassian Low Pass Filter

for i=1:M %iterates over the M rows

for j=1:N %iterates over the N columns for each row

D = sqrt(abs(i-M/2)^2 + abs(j-N/2)^2);

H3(i,j)= exp(-(D^2)/(2\*f^2)); % assigns 1 if the distance is less than f if not it assigns zero

end

end %Guassian low pass filter H1 is created

G=H3.\*FFT; % Multiplying the Fourier transformed image with H1 (the filter)

G=ifftshift(G); % Inverse fourier transform

img2=uint8(real(ifft2(G))); % butter worth low pass filtered image (output)

% displaying the images

figure

imshowpair(img,img2,"montage");title('Original image and Filtered image (BLP Filter)');xlabel("Cutoff - "),xlabel(f)

elseif(ch == 4)

break

else

disp('Invalid Choice');

end

end

case 2

flag = 1;

while(flag==1)

clc;

disp('High-Pass Filtering');

ch = input('\nType of Filter\n1.Ideal\n2.Butterworth\n3.Gaussian\n4.Back\n\nEnter Choice: ');

if(ch == 1)

f=input('Enter cut-off value(0-1) : '); % takes input for cut-off value

f1=f\*sqrt(M^2+N^2); % to put pixel in centre of an image

H1=zeros(M,N); % complete h1(filter) is zero initially (same dimension as image)

%Ideal Low Pass Filter

for i=1:M %iterates over the M rows

for j=1:N %iterates over the N columns for each row

if(sqrt(abs(i-M/2)^2+abs(j-N/2)^2) > f1) % calculates the euclidean distance and compares with f1

H1(i,j)=1; % assigns 1 if the distance is less than f if not it assigns zero

end

end

end %Ideal high pass filter H1 is created

G=H1.\*FFT; % Multiplying the Fourier transformed image with H1 (the filter)

G=ifftshift(G); % Inverse fourier transform

img2=uint8(real(ifft2(G))); % ideal low pass filtered image (output)

% displaying the images

figure

imshowpair(img,img2,"montage");title('Original image and Filtered image (IHP Filter)');xlabel("Cutoff - "),xlabel(f)

elseif(ch == 2)

f=input('Enter cut-off value(0-1) : '); % takes input for cut-off value

f1=f\*sqrt(M^2+N^2); % to put pixel in centre of an image

H2=zeros(M,N); % complete h1(filter) is zero initially (same dimension as image)

n = input('Input n value: ');

%Butter worth High Pass Filter

for i=1:M %iterates over the M rows

for j=1:N %iterates over the N columns for each row

D = sqrt(abs(i-M/2)^2 + abs(j-N/2)^2);

H2(i,j)= 1/(1+(f/D)^(2\*n)); % assigns 1 if the distance is less than f if not it assigns zero

end

end %Butter worth high pass filter H1 is created

G=H2.\*FFT; % Multiplying the Fourier transformed image with H1 (the filter)

G=ifftshift(G); % Inverse fourier transform

img2=uint8(real(ifft2(G))); % butter worth high pass filtered image (output)

% displaying the images

figure

imshowpair(img,img2,"montage");title('Original image and Filtered image (BHP Filter)');xlabel("Cutoff - "),xlabel(f)

elseif(ch == 3)

f=input('Enter cut-off value(0-1) : '); % takes input for cut-off value

f1=f\*sqrt(M^2+N^2); % to put pixel in centre of an image

H3=zeros(M,N); % complete h1(filter) is zero initially (same dimension as image)

%Guassian Low Pass Filter

for i=1:M %iterates over the M rows

for j=1:N %iterates over the N columns for each row

D = sqrt(abs(i-M/2)^2 + abs(j-N/2)^2);

H3(i,j)= 1 - exp(-(D^2)/(2\*f^2)); % assigns 1 if the distance is less than f if not it assigns zero

end

end %Guassian high pass filter H1 is created

G=H3.\*FFT; % Multiplying the Fourier transformed image with H1 (the filter)

G=ifftshift(G); % Inverse fourier transform

img2=uint8(real(ifft2(G))); % Guassian high pass filtered image (output)

% displaying the images

figure

imshowpair(img,img2,"montage");title('Original image and Filtered image (GHP Filter)');xlabel("Cutoff - "),xlabel(f)

elseif(ch == 4)

break

else

disp('Invalid Choice');

end

end

case 3

exit

end

**Outputs:**

1)Low-Pass Filtering

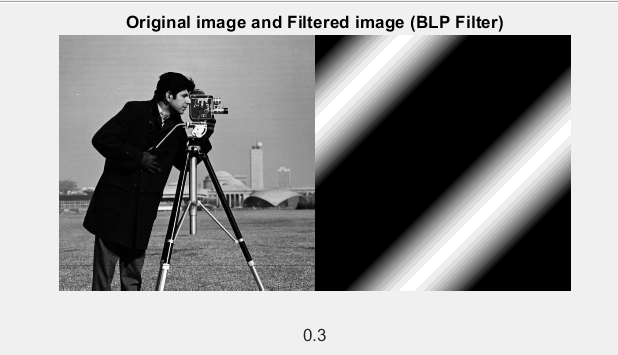
Ideal





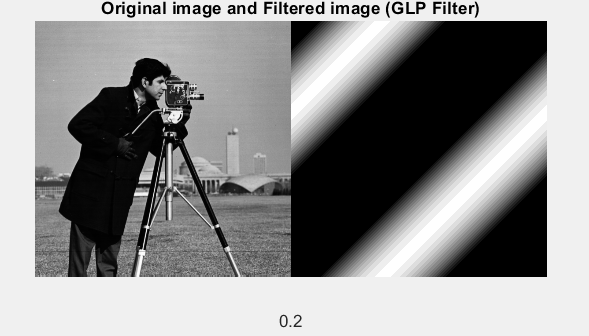


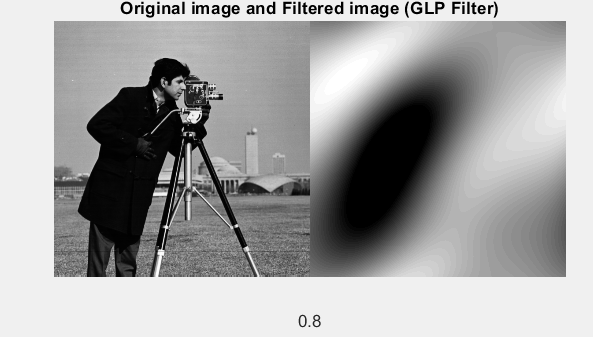
Butterworth

 n=4,cutoff=0.3

 n=2,cutoff=0.7

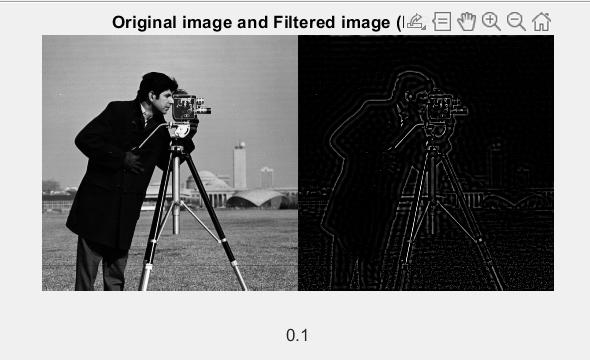
Gaussian

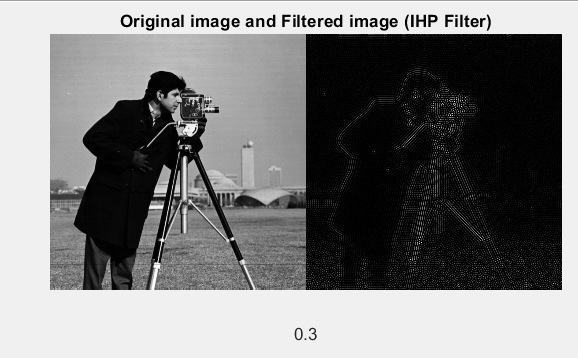




2)High-Pass Filtering

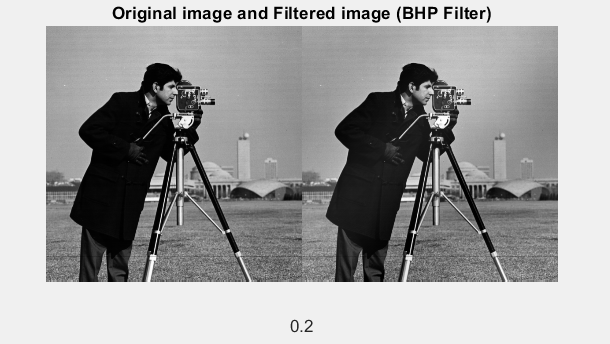
Ideal





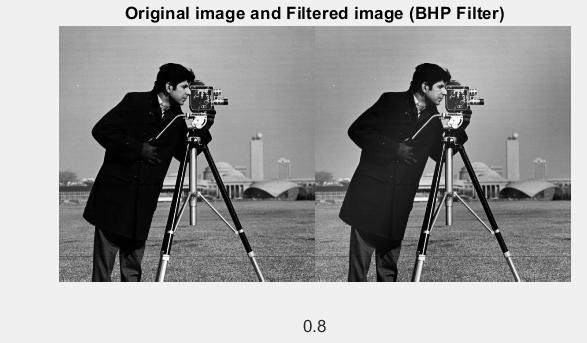


Butterworth

 n=4

 n=6

 n=10

n=3

Gaussian



