**EXPERIMENT NO. 4**

**PERFORM AND COMPARE OUTPUTS OF FREQUENCY DOMAIN FILTERING**

**EXPERIMENT NO. 4**

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

**OBJECTIVES:**

1. To apply and compare performance of averaging filters of various types, sizes and cutoff frequencies.
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

**THEORY: -**

**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 ideal LPF is given by,



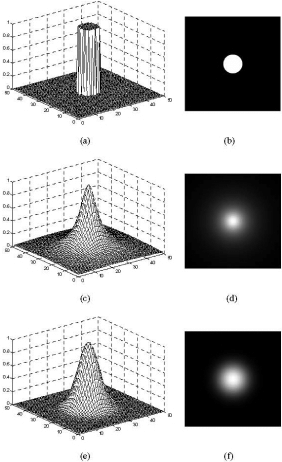
Where,



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

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

Following figure indicates 3-D and 2-D view of these filters sequentially.



**ALGORITHM:**

**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, fp(x,y) of size P X Q by appending the necessary number of zeroes to f(x,y)
4. Multiply fp(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; G3=H3.\*F
8. Obtain the real part of the inverse FFT of G.

**FUNCTIONS USED (MATLAB / SCILAB):**

1. imread
2. double
3. fft2
4. fftshift
5. ifft
6. real
7. imshow

**CODE AND OUTPUT:**

*/// Frequency Domain Filters*

clc;

clear all;

close;

im = imread('C:\Users\admin\Downloads\doogy.jpg');

imshow(im);

title('Original image')

im =rgb2gray(im);

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(abs(Gi)));

title('FT of ILPF Filtered Output');

subplot(1,3,2)

imshow(uint8(abs(Gb)));

title('FT of BLPF Filtered Output');

subplot(1,3,3)

imshow(uint8(abs(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');

****

**A picture containing diagram

Description automatically generated**

**A picture containing icon

Description automatically generated**

**Text

Description automatically generated**

**A picture containing text, white, standing, different

Description automatically generated**

**CONCLUSION:**

**Implementation of ideal, Butterworth, Gaussian low pass filters in frequency domain was performed successfully in scilab.**

**Gaussian filter gives us the best output compared to other filters. In Ideal filter there are ripples in the images. Butterworth smooths the images but the some data is also lost with it. Gaussian filters gives us a clear image with smoothing effect.**