Assignment 2: Filter

Junjia Liu[1]

[1]School of Mechanical Engineering, Shanghai Jiaotong University,
junjialiu@sjtu.edu.cn

November 27, 2018

Contents

1 Filter

In signal processing, a filter is a device or process that removes some unwanted components or features from a signal. Filtering is a class of signal processing, the defining feature of filters being the complete or partial suppression of some aspect of the signal. Most often, this means removing some frequencies or frequency bands. Filters are widely used in electronics and telecommunication, in radio, television, audio recording, radar, control systems, music synthesis, image processing, and computer graphics.

In my opinion, filter is just like a kind of weighting method.
There are many kinds of filters, such as:

- Low-pass filter – low frequencies are passed, high frequencies are attenuated.
- High-pass filter – high frequencies are passed, low frequencies are attenuated.
- Band-pass filter – only frequencies in a frequency band are passed.
- Band-stop filter or band-reject filter – only frequencies in a frequency band are attenuated.

2 Application

There are many fields which use filter to process information. In this article, we focus on the application of filter in Digital Image Processing(DIP).

Differ from the classical usage in Digital Signal Processing(DSP), which filter out some components from frequency domain, filters in DIP do operations directly on the grayscale value. Most linear spatial filters (such as Mean Filter) operate on the grayscale. This kind of linear spatial filters have a correspondence with the frequency domain filter. Essentially, the Mean Filter is a low-pass filter. However, for nonlinear filters (such as Maximum, Minimum, and Mean Filter), there is no such correspondence.

In general,linear spatial filtering of an image of size with a filter of size is given by the expression:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t) f(x+s, y+t)$$

2.1 Smoother filter

In the spatial domain, we refer to the smoothing filter, which has two forms: average filtering and weighted average filtering.

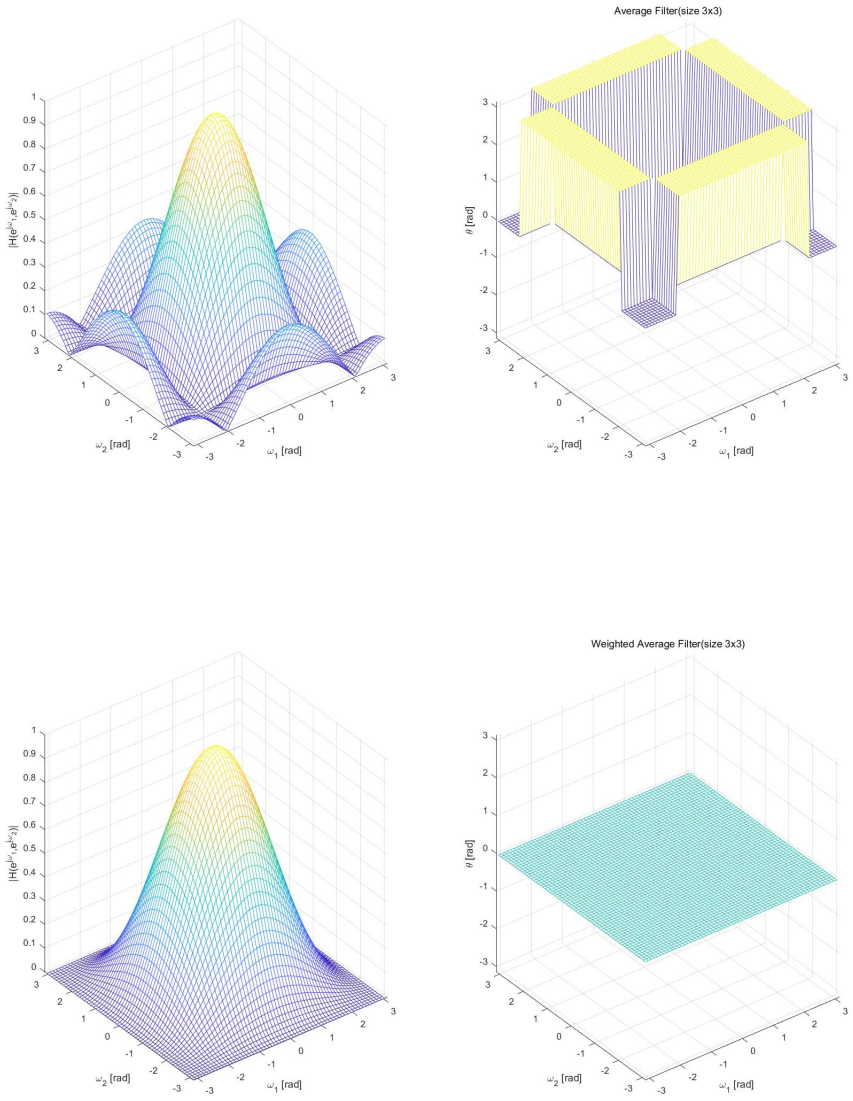
$(1/9)*$

1	1	1
1	1	1
1	1	1

$(1/16)*$

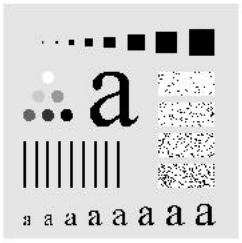
1	2	1
2	4	2
1	2	1

It is well understood here that the points in the range of the filter are averaged (or weighted averaged). This will smooth the image and help remove some noise. If we consider it in the frequency domain, this is actually a typical low-pass filter. It will filter out high frequency components, so that the image can be smoothed. The frequency response is as follows.

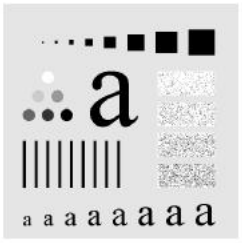


First, for the amplitude characteristics of the two filters, the pass band of the averaging filter is narrower than that of the weighted averaging filter, so the image processed using the average filter is more blurred than the the weighting filter. Note that the phase characteristics of the average filter are not a plane, and some of the value are π. The average filter is an even real function whose frequency response is a real function. However, some of its frequency response is negative, which results in the calculation of angle() of Matlab becomes π. In fact, it still has 0 phase characteristics.

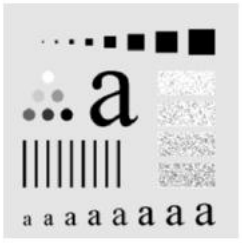
The result of using Smoother filter to process image is as follows.



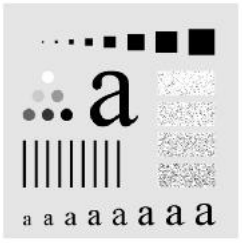
a).Original Image



b).Average Filter(size 3x3)



c).Average Filter(size 5x5)



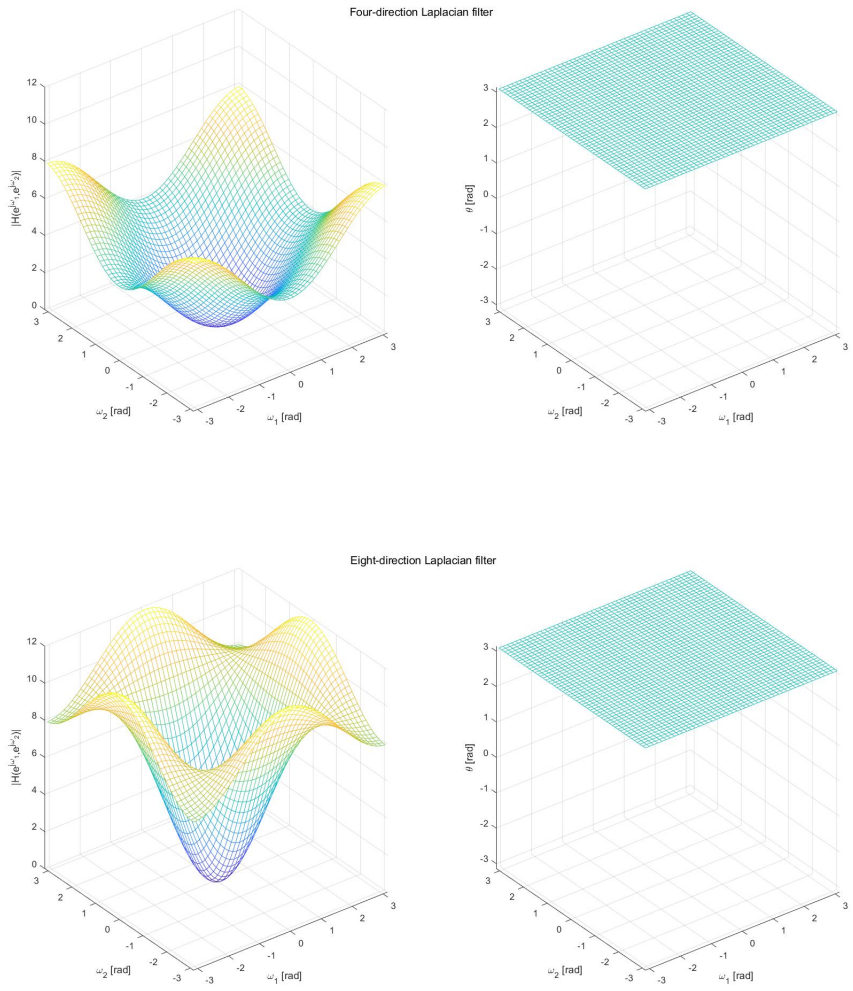
d).Weighted Average Filter(size 3x3)

```
%% ---------------------Smoother Filter---------------------
f = imread('test_pattern_blurring_orig.tif');
f = mat2gray(f,[0 255]);

w_1 = ones(3)/9;  %%%%%
g_1 = imfilter(f,w_1,'conv','symmetric','same');

w_2 = ones(5)/25;  %%%%%
g_2 = imfilter(f,w_2,'conv','symmetric','same');

w_3 = [1 2 1;
2 4 2;
1 2 1]/16;  %%%%%
g_3 = imfilter(f,w_3,'conv','symmetric','same');

figure();
subplot(2,2,1);
imshow(f,[0 1]);
xlabel('a).Original Image');

subplot(2,2,2);
imshow(g_1,[0 1]);
xlabel('b).Average Filter(size 3x3)');

subplot(2,2,3);
imshow(g_2,[0 1]);
xlabel('c).Average Filter(size 5x5)');
```

```
28
29 subplot(2,2,4);
30 imshow(g_3,[0 1]);
31 xlabel('d).Weighted Average Filter(size 3x3)');
32
33 %% ----------Smoother filter Frequency----------
34 M = 64;
35 N = 64;
36 [H_1,w1,w2] = freqz2(w_1,M,N);
37 figure();
38 subplot(1,2,1);
39 mesh(w1(1:M)*pi,w2(1:N)*pi,abs(H_1(1:M,1:N)));
40 axis([-pi pi -pi pi 0 1]);
41 xlabel('\omega_1 [rad]');ylabel('\omega_2 [rad]');
42 zlabel('|H(e^{j\omega_1},e^{j\omega_2})|');
43
44
45 %figure();
46 subplot(1,2,2);
47 mesh(w1(1:M)*pi,w2(1:N)*pi,unwrap(angle(H_1(1:M,1:N))));
48 axis([-pi pi -pi pi -pi pi]);
49 xlabel('\omega_1 [rad]');ylabel('\omega_2 [rad]');
50 zlabel('\theta [rad]');
51
52 [H_2,w3,w4] = freqz2(w_3,M,N);
53 figure();
54 subplot(1,2,1);
55 mesh(w3(1:M)*pi,w4(1:N)*pi,abs(H_2(1:M,1:N)));
56 axis([-pi pi -pi pi 0 1]);
57 xlabel('\omega_1 [rad]');ylabel('\omega_2 [rad]');
58 zlabel('|H(e^{j\omega_1},e^{j\omega_2})|');
59
60
61 %figure();
62 subplot(1,2,2);
63 mesh(w3(1:M)*pi,w4(1:N)*pi,unwrap(angle(H_2(1:M,1:N))));
64 axis([-pi pi -pi pi -pi pi]);
65 xlabel('\omega_1 [rad]');ylabel('\omega_2 [rad]');
66 zlabel('\theta [rad]');
```

2.2 Sharpen filter

Using an average filter, the image can be smoothed, it is averaging the image over the filter range essentially. From the frequency domain, the average filter is a low-pass filter. However, sharpening is to emphasize the details of the image. Here we make a hypothesis, assuming that the detail part is the high frequency component of the image. From this point of view, in fact, the sharpening filter is the opposite of the average filter.The frequency response is as follows.

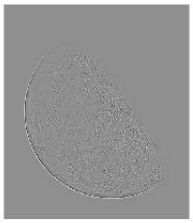


We can see that the eight-direction Laplacian filter has a strong emphasis on high-frequency components. The minimum value of the low frequency part is 0, which means that after the Laplacian filtering, only the high frequency part of the image is left (in the spatial domain, only the edge part is left). Therefore, if it is used for image sharpening, the result can be superimposed on the original image. It just like to increase the amplitude characteristic of the filter and ensure that the low frequency part is unchanged, just emphasizing the high frequency portion.

Four-direction Laplacian filter

a).Original Image

b).The Laplacian

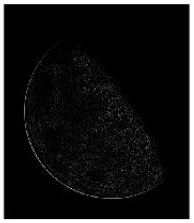


c).The Laplacian with scaling

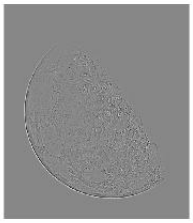
d).Result Image

Eight-direction Laplacian filter

a).Original Image



b).The Laplacian



c).The Laplacian with scaling

d).Result Image

```
f = imread('blurry_moon.tif');
f = mat2gray(f,[0 255]);

w_L = [0 1 0
1 -4 1
0 1 0];
g_L_whitout = imfilter(f,w_L,'conv','symmetric','same');
g_L = mat2gray(g_L_whitout);
g = f - g_L_whitout;
g = mat2gray(g ,[0 1]);

w_L8 = [1 1 1
1 -8 1
1 1 1];
g_L_whitout8 = imfilter(f,w_L8,'conv','symmetric','same');
g_L8 = mat2gray(g_L_whitout8);
g8 = f - g_L_whitout8;
g8 = mat2gray(g8 ,[0 1]);

%4
figure();
subplot(2,2,1);
imshow(f,[0 1]);
xlabel('a).Original Image');

subplot(2,2,2);
imshow(g_L_whitout,[0 1]);
xlabel('b).The Laplacian');

subplot(2,2,3);
imshow(g_L,[0 1]);
xlabel('c).The Laplacian with scaling');

subplot(2,2,4);
imshow(g,[0 1]);
xlabel('d).Result Image');

%8
figure();
subplot(2,2,1);
imshow(f,[0 1]);
xlabel('a).Original Image');

subplot(2,2,2);
imshow(g_L_whitout8,[0 1]);
xlabel('b).The Laplacian');

subplot(2,2,3);
```

```
imshow(g_L8,[0 1]);
xlabel('c).The Laplacian with scaling');

subplot(2,2,4);
imshow(g8,[0 1]);
xlabel('d).Result Image');
%% ------------------------------------------------------------
[M,N] = size(f);
[H,w1,w2] = freqz2(w_L,N,M);
figure();
subplot(1,2,1);
mesh(w1(1:10:N)*pi,w2(1:10:M)*pi,abs(H(1:10:M,1:10:N)));
axis([-pi pi -pi pi 0 12]);
xlabel('\omega_1 [rad]');ylabel('\omega_2 [rad]');
zlabel('|H(e^{j\omega_1},e^{j\omega_2})|');

subplot(1,2,2);
mesh(w1(1:10:N)*pi,w2(1:10:M)*pi,unwrap(angle(H(1:10:M,1:10:N))));
axis([-pi pi -pi pi -pi pi]);
xlabel('\omega_1 [rad]');ylabel('\omega_2 [rad]');
zlabel('\theta [rad]');

[H8,w3,w4] = freqz2(w_L8,N,M);
figure();
subplot(1,2,1);
mesh(w3(1:10:N)*pi,w4(1:10:M)*pi,abs(H8(1:10:M,1:10:N)));
axis([-pi pi -pi pi 0 12]);
xlabel('\omega_1 [rad]');ylabel('\omega_2 [rad]');
zlabel('|H(e^{j\omega_1},e^{j\omega_2})|');

subplot(1,2,2);
mesh(w3(1:10:N)*pi,w4(1:10:M)*pi,unwrap(angle(H8(1:10:M,1:10:N))));
axis([-pi pi -pi pi -pi pi]);
xlabel('\omega_1 [rad]');ylabel('\omega_2 [rad]');
zlabel('\theta [rad]');
```

3 Design a filter and analyze its characteristics

Here I will design three common low-pass filters: Ideal Lowpass Filters(ILPF), Butterworth Lowpass Filters(BLPF) and Gaussian Lowpass Filters(GLPF).

Ideal Lowpass Filters

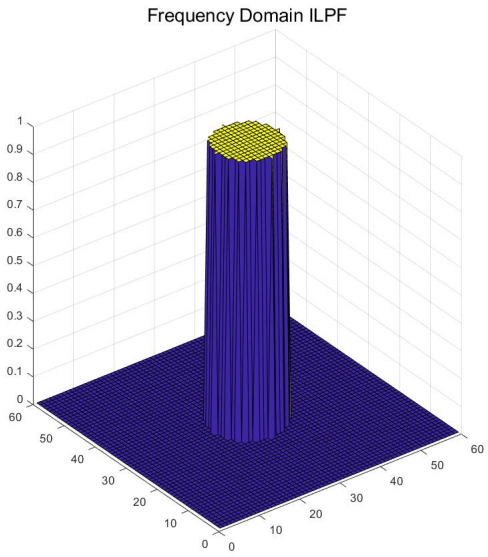
$$H(u,v) = \begin{cases} 1, D(u,v) \leq D_0 \\ 0, D(u,v) > D_0 \end{cases}$$

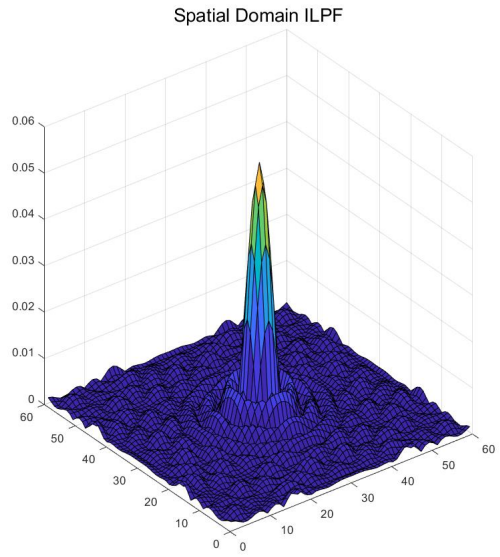
Butterworth Lowpass Filters

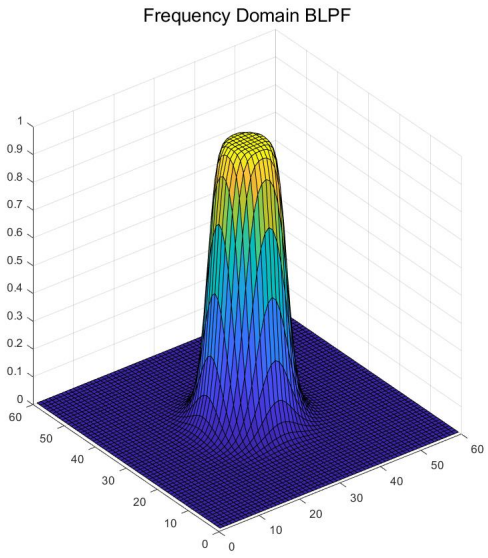
$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^2}$$

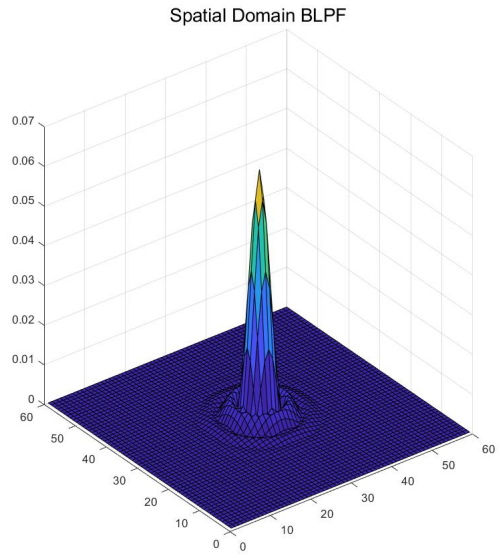
Gaussian Lowpass Filters

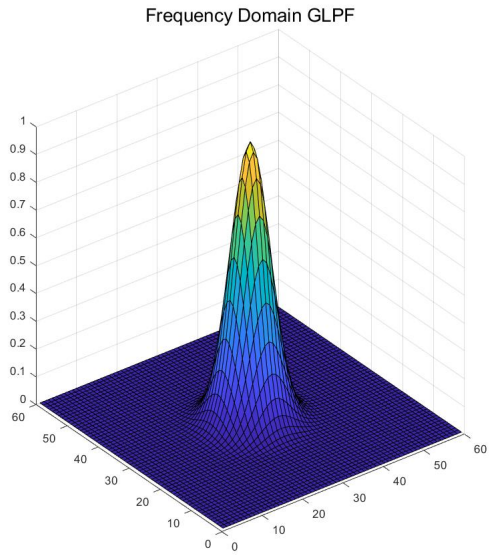
$$H(u,v) = \exp\left(-\frac{D(u,v)^2}{2D_0^2}\right)$$

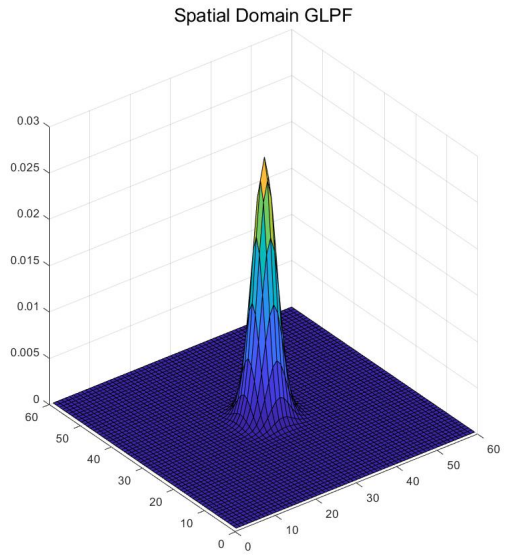










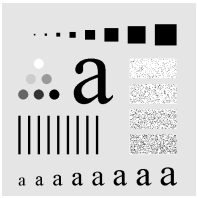



```
d0=8;
M=60;N=60;
c1=floor(M/2);
c2=floor(N/2);
h1=zeros(M,N);          %Ideal
h2=zeros(M,N);          %ButterWorth
h3=zeros(M,N);          %Gaussian
D0=4;
n=4;
for i=1:M
for j=1:N
d=sqrt((i-c1)^2+(j-c2)^2);
if d<=d0
h1(i,j)=1;
else
h1(i,j)=0;
end
h2(i,j)=1/(1+(d/d0)^(2*n));
h3(i,j)=exp(-d^2/(2*D0^2));
end
end
draw2(h1,' ILPF');
draw2(h2,' BLPF');
draw2(h3,' GLPF');

function draw2(h,name)
figure;
subplot(1,2,1);
surf(h);title(strcat('Frequency Domain',name));
fx=abs(ifft2(h));
```

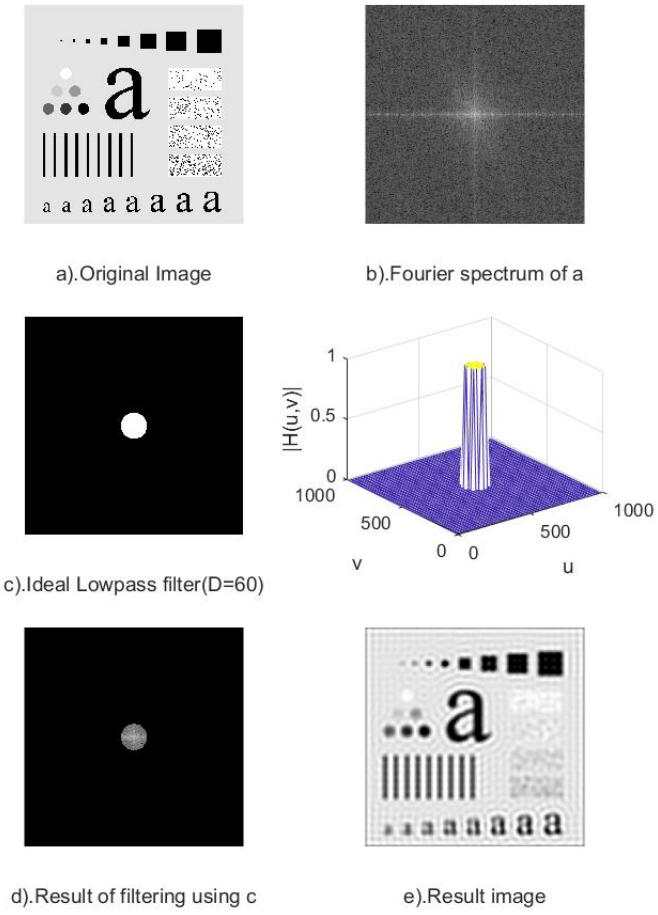
```
31 fx=fftshift(fx);
32 subplot(1,2,2);
33 surf(fx);title(strcat('Spatial Domain',name));
34 end
```

4 Process an image using filters and analyze the results

I will still use this classical image to figure out the difference between these three kinds of low-pass filters.

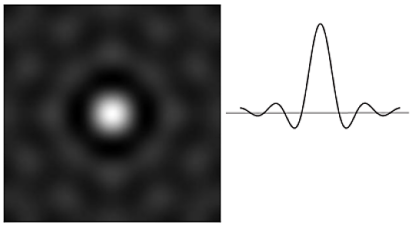


4.1 Ideal Lowpass Filters



It filters out high frequency components, which blurs the image. Since the excessive characteristics of the ideal low-pass filter are too severe, ringing occurs.

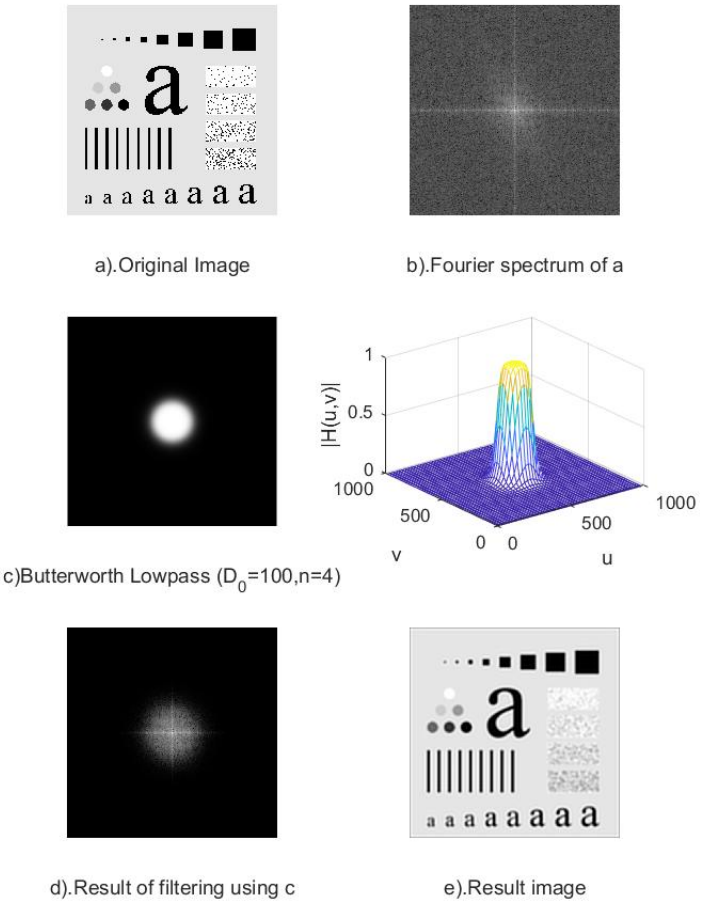
The blurring and ringing properties of ILPF can be explained using the convolution theorem. Because a cross section of the ILPF in the frequency domain looks like a box filter, it is not unexpected that a cross section of the corresponding spatial filter has the shape of a sinc function. Filtering in the spatial domain is done by convolving $h(x, y)$ with the image. Imagine each pixel in the image being a discrete impulse whose strength is proportional to the intensity of the image at that location.Convolving a sinc with an impulse copies the sinc at the location of the impulse.The center lobe of the sinc is the principal cause of blurring, while the outer, smaller lobes are mainly responsible for ringing.



```
%% ----------Ideal Lowpass Filters (Fre. Domain)----------
f = imread('test_pattern_blurring_orig.tif');
f = mat2gray(f,[0 255]);

[M,N] = size(f);
P = 2*M;
Q = 2*N;
fc = zeros(M,N);

for x = 1:1:M
for y = 1:1:N
fc(x,y) = f(x,y) * (-1)^(x+y);
end
end

F = fft2(fc,P,Q);

H_1 = zeros(P,Q);

for x = (-P/2):1:(P/2)-1
for y = (-Q/2):1:(Q/2)-1
D = (x^2 + y^2)^(0.5);
if(D <= 60)  H_1(x+(P/2)+1,y+(Q/2)+1) = 1; end
end
end

G_1 = H_1 .* F;
g_1 = real(ifft2(G_1));
g_1 = g_1(1:1:M,1:1:N);
```

```
30
31 for x = 1:1:M
32 for y = 1:1:N
33 g_1(x,y) = g_1(x,y) * (-1)^(x+y);
34 end
35 end
36
37 %% ------show-------
38 figure();
39 subplot(3,2,1);
40 imshow(f,[0 1]);
41 xlabel('a).Original Image');
42
43 subplot(3,2,2);
44 imshow(log(1 + abs(F)),[ ]);
45 xlabel('b).Fourier spectrum of a');
46
47 subplot(3,2,3);
48 imshow(H_1,[0 1]);
49 xlabel('c).Ideal Lowpass filter(D=60)');
50
51 subplot(3,2,4);
52 h = mesh(1:20:P,1:20:Q,H_1(1:20:P,1:20:Q));
53 axis([0 P 0 Q 0 1]);
54 xlabel('u');ylabel('v');
55 zlabel('|H(u,v)|');
56
57 subplot(3,2,5);
58 imshow(log(1 + abs(G_1)),[ ]);
59 xlabel('d).Result of filtering using c');
60
61 subplot(3,2,6);
62 imshow(g_1,[0 1]);
63 xlabel('e).Result image');
```

4.2 Butterworth Lowpass Filters

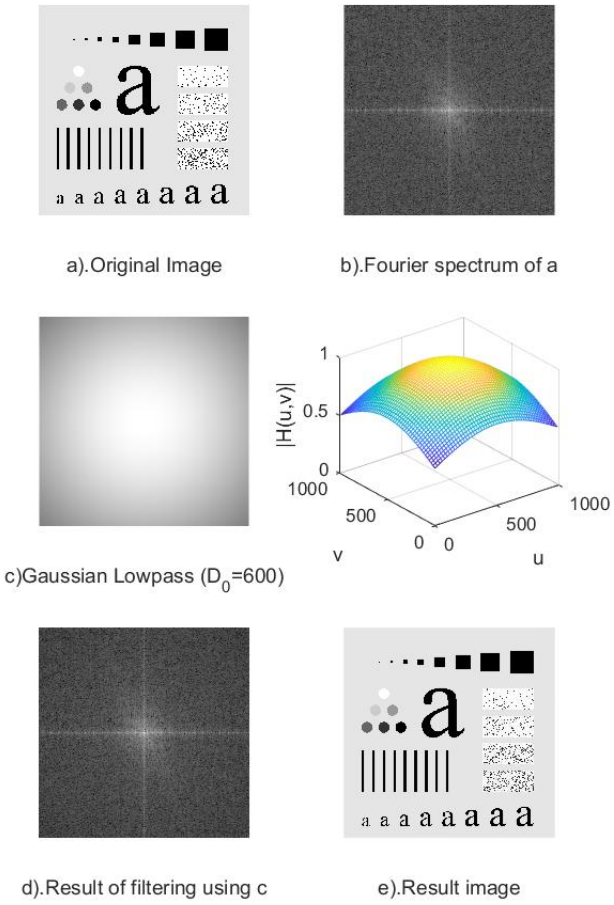


D_0 represents the radius of the pass band, and n represents the times of Butterworth filters. As the number of times increases, the ringing phenomenon becomes more and more obvious.

```
%% ----------Butterworth Lowpass Filters (Fre. Domain)----------
f = imread('test_pattern_blurring_orig.tif');
f = mat2gray(f,[0 255]);

[M,N] = size(f);
P = 2*M;
Q = 2*N;
fc = zeros(M,N);

for x = 1:1:M
for y = 1:1:N
fc(x,y) = f(x,y) * (-1)^(x+y);
end
end

F = fft2(fc,P,Q);
```

```
17
18 H_1 = zeros(P,Q);
19 H_2 = zeros(P,Q);
20
21 for x = (-P/2):1:(P/2)-1
22 for y = (-Q/2):1:(Q/2)-1
23 D = (x^2 + y^2)^(0.5);
24 D_0 = 100;
25 H_1(x+(P/2)+1,y+(Q/2)+1) = 1/(1+(D/D_0)^8);
26 end
27 end
28
29 G_1 = H_1 .* F;
30 g_1 = real(ifft2(G_1));
31 g_1 = g_1(1:1:M,1:1:N);
32
33 for x = 1:1:M
34 for y = 1:1:N
35 g_1(x,y) = g_1(x,y) * (-1)^(x+y);
36 end
37 end
38
39 %% ------show-------
40 figure();
41 subplot(3,2,1);
42 imshow(f,[0 1]);
43 xlabel('a).Original Image');
44
45 subplot(3,2,2);
46 imshow(log(1 + abs(F)),[ ]);
47 xlabel('b).Fourier spectrum of a');
48
49 subplot(3,2,3);
50 imshow(H_1,[0 1]);
51 xlabel('c)Butterworth Lowpass (D_{0}=100,n=4)');
52
53 subplot(3,2,4);
54 h = mesh(1:20:P,1:20:Q,H_1(1:20:P,1:20:Q));
55 %set(h,'EdgeColor','k');
56 axis([0 P 0 Q 0 1]);
57 xlabel('u');ylabel('v');
58 zlabel('|H(u,v)|');
59
60 subplot(3,2,5);
61 imshow(log(1 + abs(G_1)),[ ]);
62 xlabel('d).Result of filtering using c');
63
64 subplot(3,2,6);
65 imshow(g_1,[0 1]);
66 xlabel('e).Result image');
```

4.3 Gaussian Lowpass Filters



D_0 represents the radius of the pass band. The transition of the Gaussian filter are very smooth, so there is no ringing.

```
%% ----------Butterworth Lowpass Filters (Fre. Domain)----------
f = imread('test_pattern_blurring_orig.tif');
f = mat2gray(f,[0 255]);

[M,N] = size(f);
P = 2*M;
Q = 2*N;
fc = zeros(M,N);

for x = 1:1:M
for y = 1:1:N
fc(x,y) = f(x,y) * (-1)^(x+y);
end
end

F = fft2(fc,P,Q);
```

```
17
18 H_1 = zeros(P,Q);
19 H_2 = zeros(P,Q);
20
21 for x = (-P/2):1:(P/2)-1
22 for y = (-Q/2):1:(Q/2)-1
23 D = (x^2 + y^2)^(0.5);
24 D_0 = 100;
25 H_1(x+(P/2)+1,y+(Q/2)+1) = 1/(1+(D/D_0)^8);
26 end
27 end
28
29 G_1 = H_1 .* F;
30 g_1 = real(ifft2(G_1));
31 g_1 = g_1(1:1:M,1:1:N);
32
33 for x = 1:1:M
34 for y = 1:1:N
35 g_1(x,y) = g_1(x,y) * (-1)^(x+y);
36 end
37 end
38
39 %% ------show-------
40 figure();
41 subplot(3,2,1);
42 imshow(f,[0 1]);
43 xlabel('a).Original Image');
44
45 subplot(3,2,2);
46 imshow(log(1 + abs(F)),[ ]);
47 xlabel('b).Fourier spectrum of a');
48
49 subplot(3,2,3);
50 imshow(H_1,[0 1]);
51 xlabel('c)Butterworth Lowpass (D_{0}=100,n=4)');
52
53 subplot(3,2,4);
54 h = mesh(1:20:P,1:20:Q,H_1(1:20:P,1:20:Q));
55 %set(h,'EdgeColor','k');
56 axis([0 P 0 Q 0 1]);
57 xlabel('u');ylabel('v');
58 zlabel('|H(u,v)|');
59
60 subplot(3,2,5);
61 imshow(log(1 + abs(G_1)),[ ]);
62 xlabel('d).Result of filtering using c');
63
64 subplot(3,2,6);
65 imshow(g_1,[0 1]);
66 xlabel('e).Result image');
```