EE 4211 Computer Vision

Lecture 3B: Image enhancement (Frequency)

Semester A, 2020-2021

Spatial Domain vs. Frequency Domain

- Spatial Domain (image plane)
  - Techniques are based on direct manipulation of pixels in an image
- Frequency Domain
  - Techniques are based on modifying the spectral transform (in our course, we'll use Fourier transform) of an image
- There are some enhancement techniques based on various combinations of methods from these 2 domains

Lecture Outline

- Image Enhancement in Frequency Domain
- Filtering in Frequency Domain
  - Low-Pass Filtering
  - High-Pass Filtering
  - Laplacian Filtering
  - Homomorphic Filtering
  - Selective Filtering – Bandpass/Bandreject, Notch Filters

Image Enhancement in Frequency Domain

- Images after transformation to frequency domain can be modified with frequency filters

- Nature of periodicity & conjugate symmetry.

- So, origin of spectrum is always shifted for
  - Display purpose
  - Filtering purpose

Image Enhancement in Frequency Domain

- To filter an image in the frequency domain:
  - Compute F(u,v) the DFT of the image
  - Multiply F(u,v) by a filter function H(u,v)
  - Compute the inverse DFT of the result

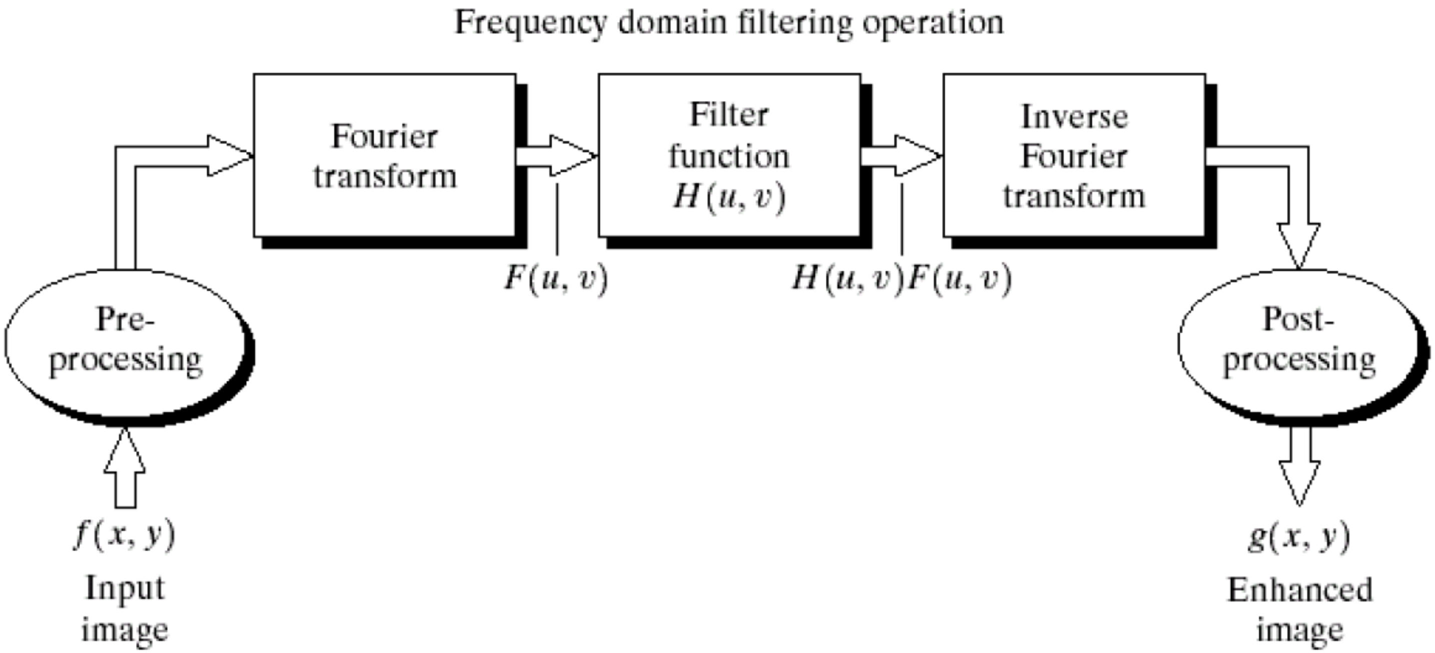


Image Enhancement in Frequency Domain

- Steps taken:
- Shift the origin of spectrum by multiplying image f(x,y) by $(-1)^{x+y}$ before performing transformation to frequency domain.

$$f(x,y)(-1)^{(x+y)} \qquad F\left(u-\frac{M}{2}, v-\frac{N}{2}\right)$$

Image Enhancement in Frequency Domain

- G(u,v), the frequency spectrum obtained after applying frequency filter H(u,v) to frequency-transform image F(u,v) by multiplication, is defined as:

$$G(u,v) = H(u,v) * F(u,v)$$

- The filtered image, g(x,y) can then be obtained by performing the inverse transform to G(u,v):

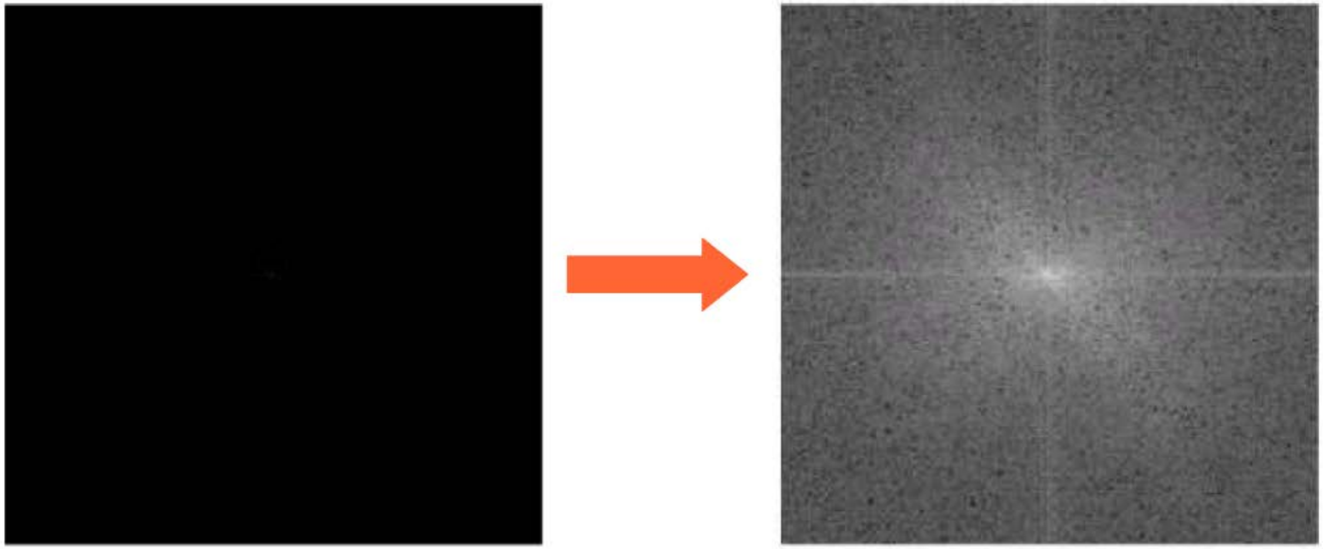
$$g(x,y) = G^{-1}(u,v)$$

- Any shifting to origin performed before filtering should also be reversed after filtering.

Image Enhancement in Frequency Domain

- Steps taken:
- Enhance the visual information of the transformed image G(u,v) using log transform:

$$D(u,v) = k \log\left[1 + |G(u,v)|\right]$$



Lecture Outline

- Image Enhancement in Frequency Domain
- Filtering in Frequency Domain
  - Low-Pass Filtering
  - High-Pass Filtering
  - Laplacian Filtering
  - Homomorphic Filtering
  - Selective Filtering – Bandpass/Bandreject, Notch Filters

Low-Pass Filter (in Freq. Domain)

- Lowpass filter – remove high-frequency information, or allow low-frequency information to PASS through
- Useful for removing noise in images
- Also have undesired effect of blurring an image

Ideal Low-Pass Filter (ILPF)

- An ideal low-pass filter contain only 1's and 0's – 1 for lower frequency and 0 for high frequency
- 2-D ideal lowpass filter (ILPF) is defined as:

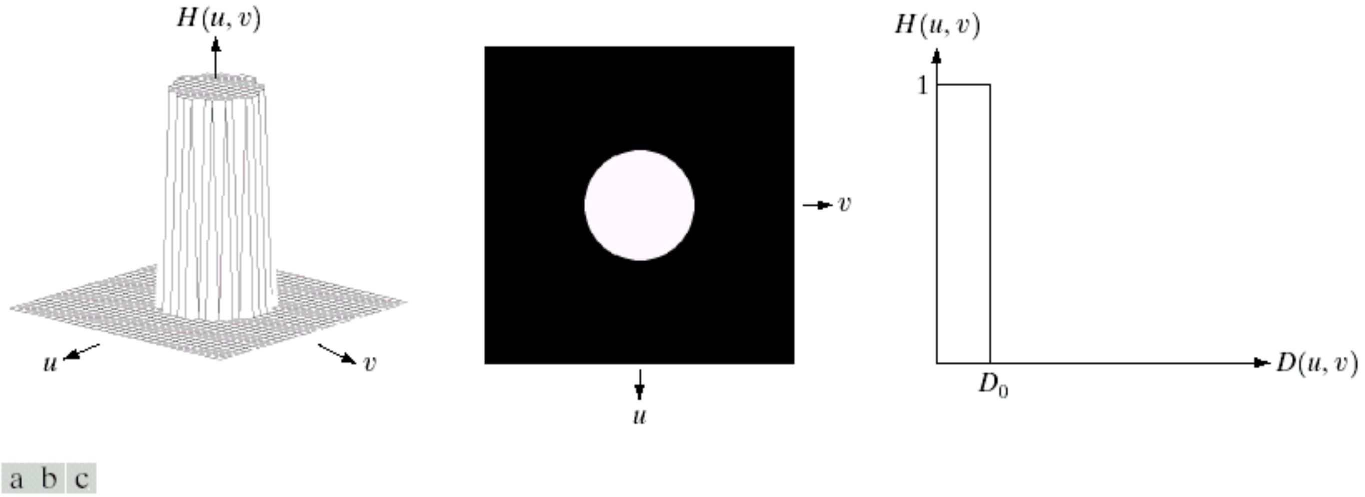
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

- where D_0 is a positive constant and D(u,v) is the distance from point (u,v) to the origin (center) of the frequency rectangle. It is denoted as

$$D(u,v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

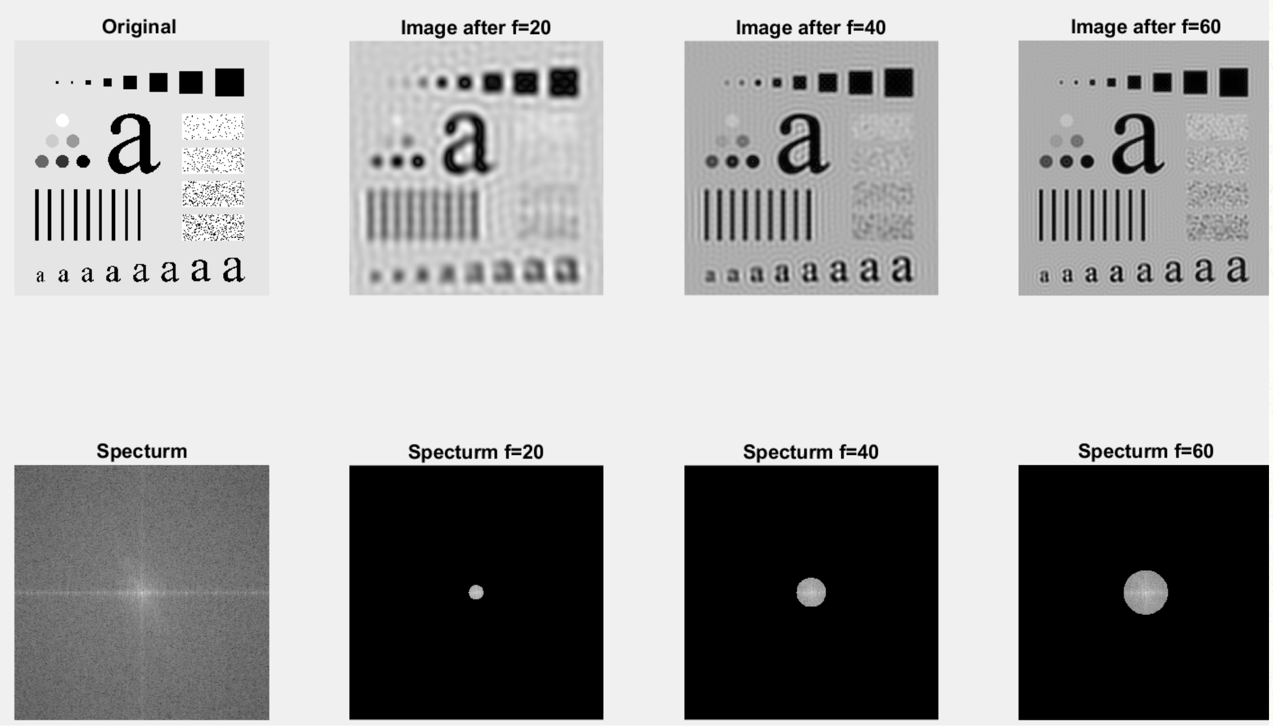
Ideal Low-Pass Filter (ILPF)

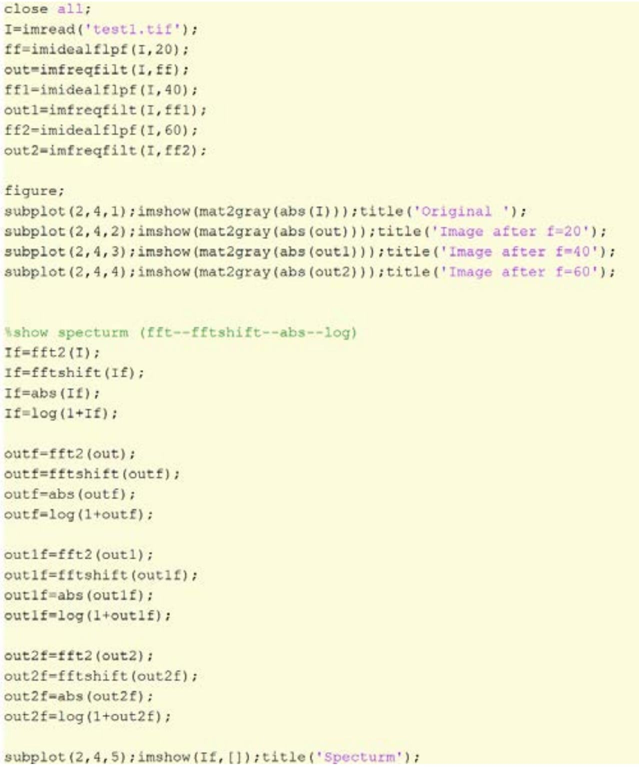
- Ideal Low pass filter (ILPF): all frequencies inside a circle of radius D_0 are passed with no attenuation



(a)Perspective plot of an ideal low pass filter transfer function
(b)Filter displayed as an image
(c)Filter radial cross section

Images Filtered by ILPFs

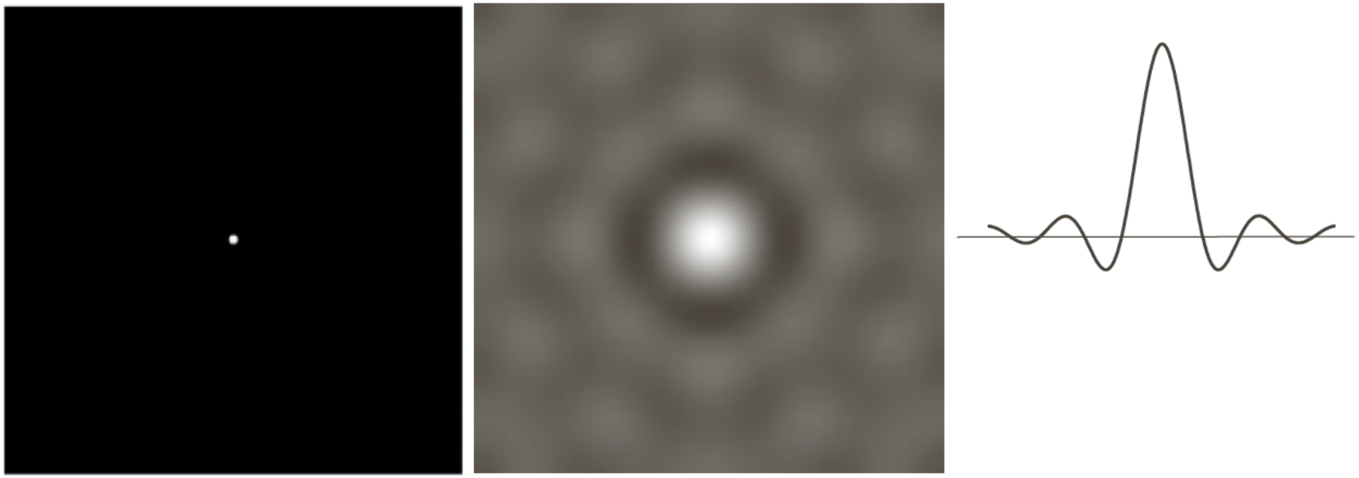




```
close all;
I=imread('test1.tif');
ff=imidealflpf(I,20);
out=imfreqfilt(I,ff);
ff1=imidealflpf(I,40);
out1=imfreqfilt(I,ff1);
ff2=imidealflpf(I,60);
out2=imfreqfilt(I,ff2);

figure;
subplot(2,4,1);imshow(mat2gray(abs(I)));title('Original ');
subplot(2,4,2);imshow(mat2gray(abs(out)));title('Image after f=20');
subplot(2,4,3);imshow(mat2gray(abs(out1)));title('Image after f=40');
subplot(2,4,4);imshow(mat2gray(abs(out2)));title('Image after f=60');


%show specturm (fft--fftshift--abs--log)
If=fft2(I);
If=fftshift(If);
If=abs(If);
If=log(1+If);

outf=fft2(out);
outf=fftshift(outf);
outf=abs(outf);
outf=log(1+outf);

out1f=fft2(out1);
out1f=fftshift(out1f);
out1f=abs(out1f);
out1f=log(1+out1f);

out2f=fft2(out2);
out2f=fftshift(out2f);
out2f=abs(out2f);
out2f=log(1+out2f);

subplot(2,4,5);imshow(If,[]);title('Specturm');
```

Ideal Low-Pass Filter (ILPF)

- An ideal filter has undesired artifacts in images
- Presence of ripples/waves whenever there are boundaries in the image – “ringing effect”



Butterworth Low-Pass Filter (BLPF)

- Transfer function of a Butterworth lowpass filter (BLPF) of order *n*, with D_0 cutoff frequency, is defined as

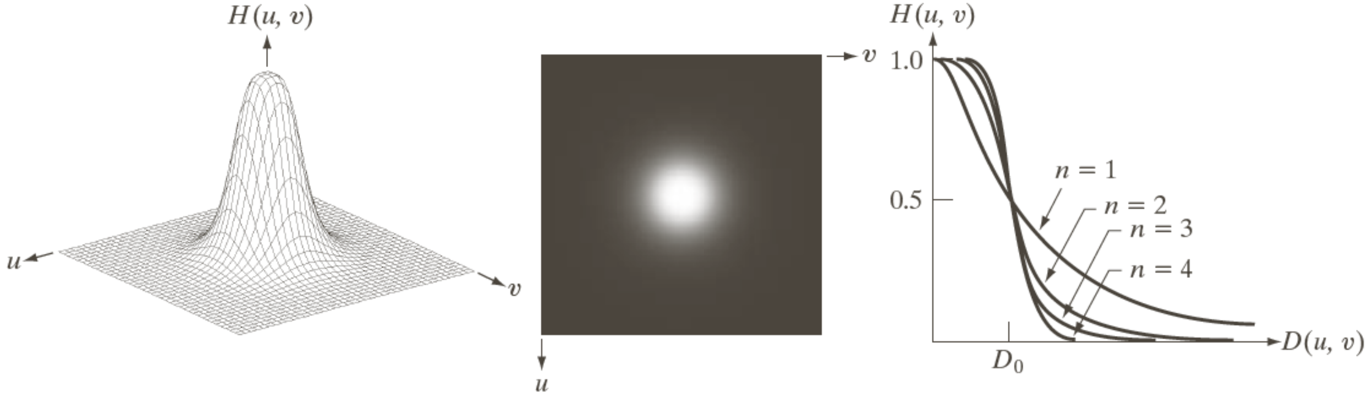
$$H(u,v) = \frac{1}{1 + [D(u,v)/D_o]^{2n}}$$

- where D(u,v) is the distance from origin of spectrum

- Butterworth lowpass filter – Can specify order of filter, which determines steepness of slope in the transition of the filter function
  - Higher order of filter – steeper slope – closer to ideal filter

Butterworth Low-Pass Filter (BLPF)

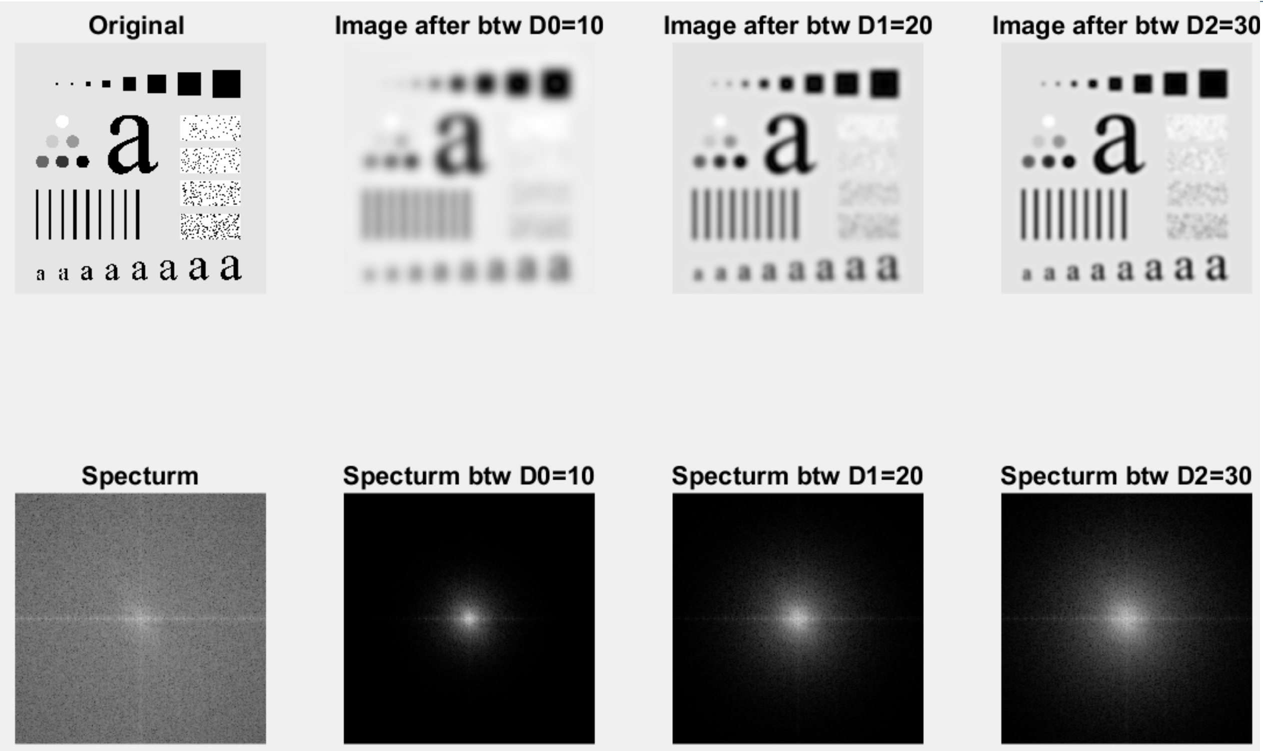
- Advantages
  - Reduces “ringing” while keeping clear cutoff
  - Tradeoff between amount of ringing and sharpness of cutoff

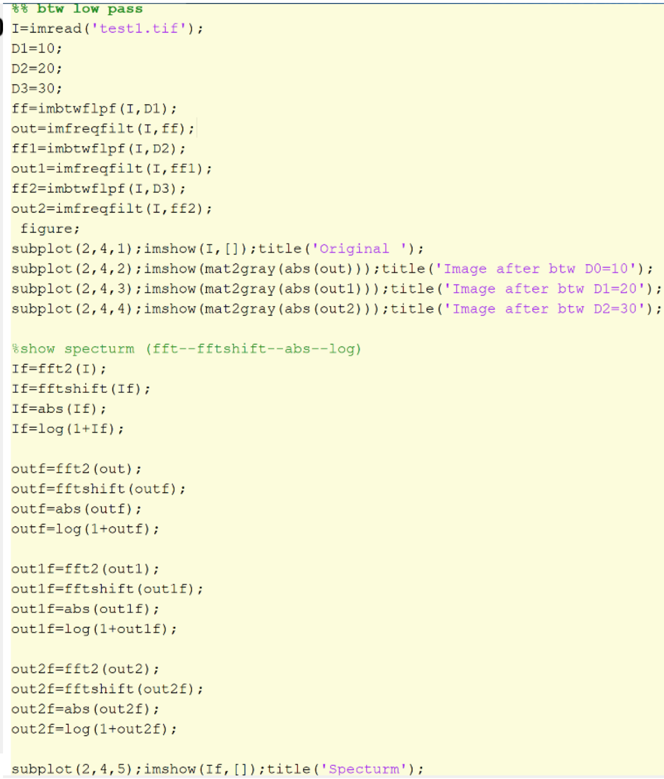


a b c

(a)Perspective plot of a Butterworth low pass filter transfer function
(b)Filter displayed as an image
(c)Filter radial cross section of orders 1 through 4

Butterworth Low-Pass Filter (BLPF)

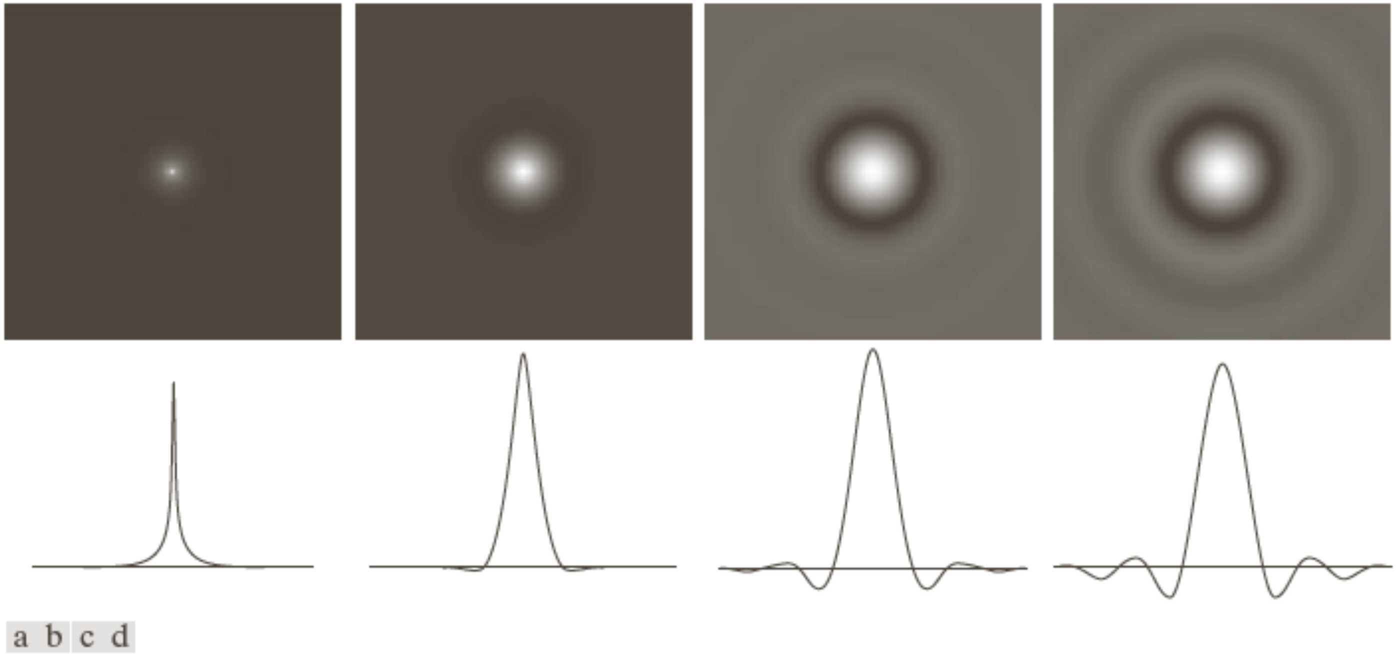




```
%% btw low pass
I=imread('test1.tif');
D1=10;
D2=20;
D3=30;
ff=imbtwflpf(I,D1);
out=imfreqfilt(I,ff);
ff1=imbtwflpf(I,D2);
out1=imfreqfilt(I,ff1);
ff2=imbtwflpf(I,D3);
out2=imfreqfilt(I,ff2);
 figure;
subplot(2,4,1);imshow(I,[]);title('Original ');
subplot(2,4,2);imshow(mat2gray(abs(out)));title('Image after btw D0=10');
subplot(2,4,3);imshow(mat2gray(abs(out1)));title('Image after btw D1=20');
subplot(2,4,4);imshow(mat2gray(abs(out2)));title('Image after btw D2=30');

%show specturm (fft--fftshift--abs--log)
If=fft2(I);
If=fftshift(If);
If=abs(If);
If=log(1+If);

outf=fft2(out);
outf=fftshift(outf);
outf=abs(outf);
outf=log(1+outf);

out1f=fft2(out1);
out1f=fftshift(out1f);
out1f=abs(out1f);
out1f=log(1+out1f);

out2f=fft2(out2);
out2f=fftshift(out2f);
out2f=abs(out2f);
out2f=log(1+out2f);

subplot(2,4,5);imshow(If,[]);title('Specturm');
```

Butterworth Low-Pass Filter (BLPF)

- Ringing properties increase if we increase the BLPF filter order, *n*



(a)-(d) Spatial representation of BLPFS of order 1, 2, 5, 20 and corresponding intensity profiles

Gaussian Low-Pass Filter (GLPF)

- Gaussian lowpass filter in 2-D is defined as

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$$

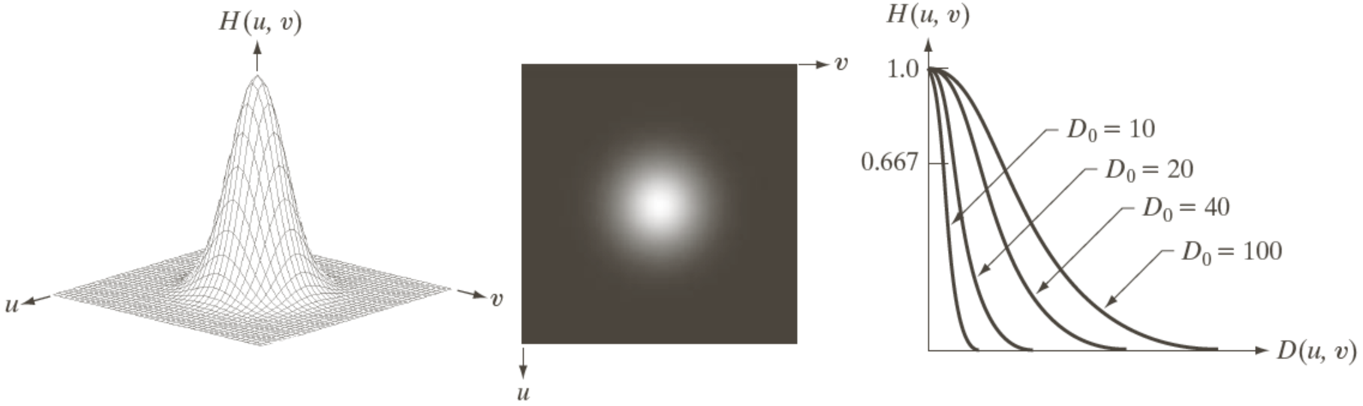
- where D(u,v) is the distance from the origin of spectrum, σ is the measure of spread of the Gaussian curve

- By letting σ = D_0, where D_0 is the cutoff frequency, we get

$$H(u,v) = e^{-D^2(u,v)/2D_0^{\ 2}}$$

Gaussian Low-Pass Filter (GLPF)

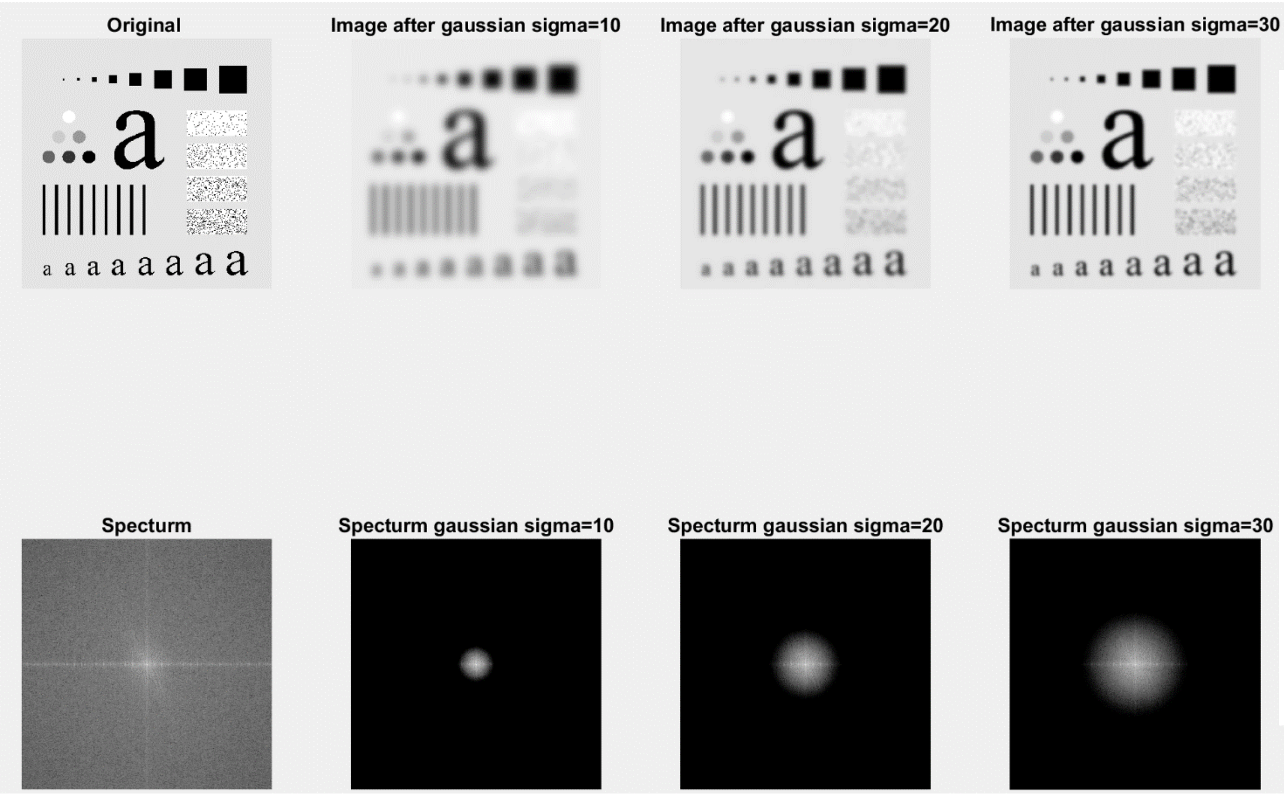
- A Gaussian in the spatial domain also has the form of a Gaussian in the frequency domain
- No ringing, but allows high frequencies to pass

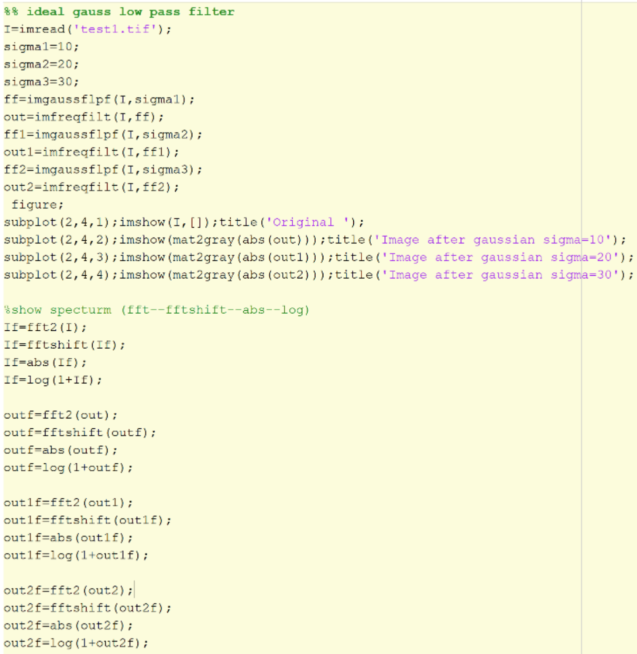


a b c

(a)Perspective plot of a GLPF transfer function
(b)Filter displayed as an image
(c)Filter radial cross sections for various values of D_0

Images Filtered by GLPFs


Original
Image after gaussian sigma=10
Image after gaussian sigma=20
Image after gaussian sigma=30
Specturm
Specturm gaussian sigma=10
Specturm gaussian sigma=20
Specturm gaussian sigma=30



```
%% ideal gauss low pass filter
I=imread('test1.tif');
sigma1=10;
sigma2=20;
sigma3=30;
ff=imgaussflpf(I,sigma1);
out=imfreqfilt(I,ff);
ff1=imgaussflpf(I,sigma2);
out1=imfreqfilt(I,ff1);
ff2=imgaussflpf(I,sigma3);
out2=imfreqfilt(I,ff2);
 figure;
subplot(2,4,1);imshow(I,[]);title('Original ');
subplot(2,4,2);imshow(mat2gray(abs(out)));title('Image after gaussian sigma=10');
subplot(2,4,3);imshow(mat2gray(abs(out1)));title('Image after gaussian sigma=20');
subplot(2,4,4);imshow(mat2gray(abs(out2)));title('Image after gaussian sigma=30');

%show specturm (fft--fftshift--abs--log)
If=fft2(I);
If=fftshift(If);
If=abs(If);
If=log(1+If);

outf=fft2(out);
outf=fftshift(outf);
outf=abs(outf);
outf=log(1+outf);

out1f=fft2(out1);
out1f=fftshift(out1f);
out1f=abs(out1f);
out1f=log(1+out1f);

out2f=fft2(out2);
out2f=fftshift(out2f);
out2f=abs(out2f);
out2f=log(1+out2f);
```

Lecture Outline

- Image Enhancement in Frequency Domain
- Filtering in Frequency Domain
  - Low-Pass Filtering
  - High-Pass Filtering
  - Laplacian Filtering
  - Homomorphic Filtering
  - Selective Filtering – Bandpass/Bandreject, Notch Filters

High-Pass Filter (in Freq. Domain)

- Highpass filter – remove low-frequency information, or allow HIGH-frequency information to PASS through
- Useful for sharpening, edge enhancement
- Transfer function of the highpass filters can be obtained using the relation

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

- where $H_{lp}(u,v)$ is the transfer function of the corresponding lowpass filter

High-Pass Filters

- Ideal highpass filter

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

- Butterworth highpass filter of order *n* and with cutoff frequency at a distance D_0 from the origin:

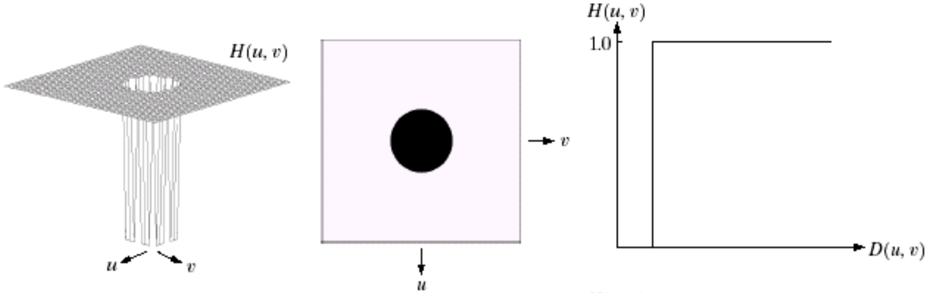
$$H(u,v) = 1 - \frac{1}{1 + [D_o / D(u,v)]^{2n}}$$

- Gaussian highpass filter with cutoff frequency at a distance D_0 from the origin

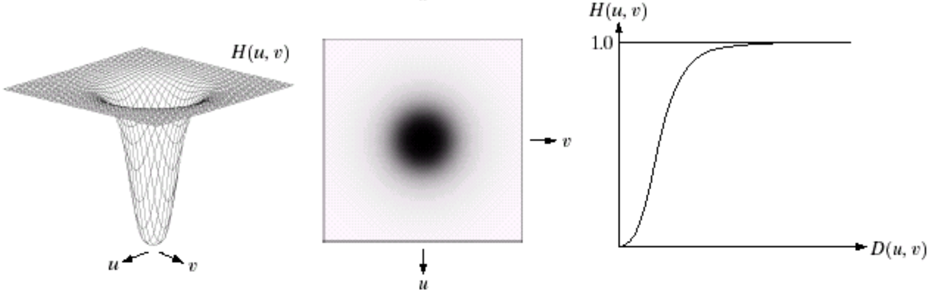
$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^{\,2}}$$

High-Pass Filters

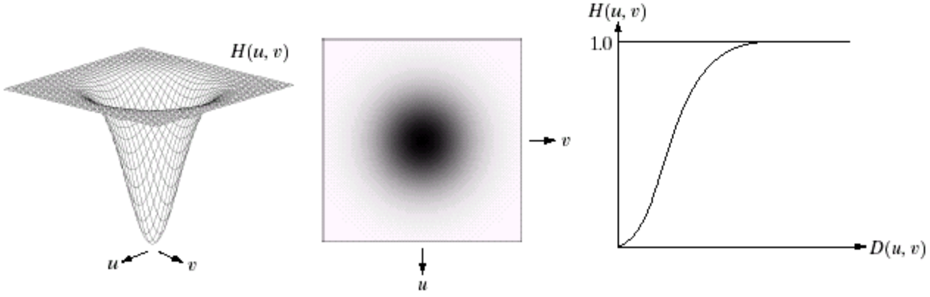
Ideal highpass filter



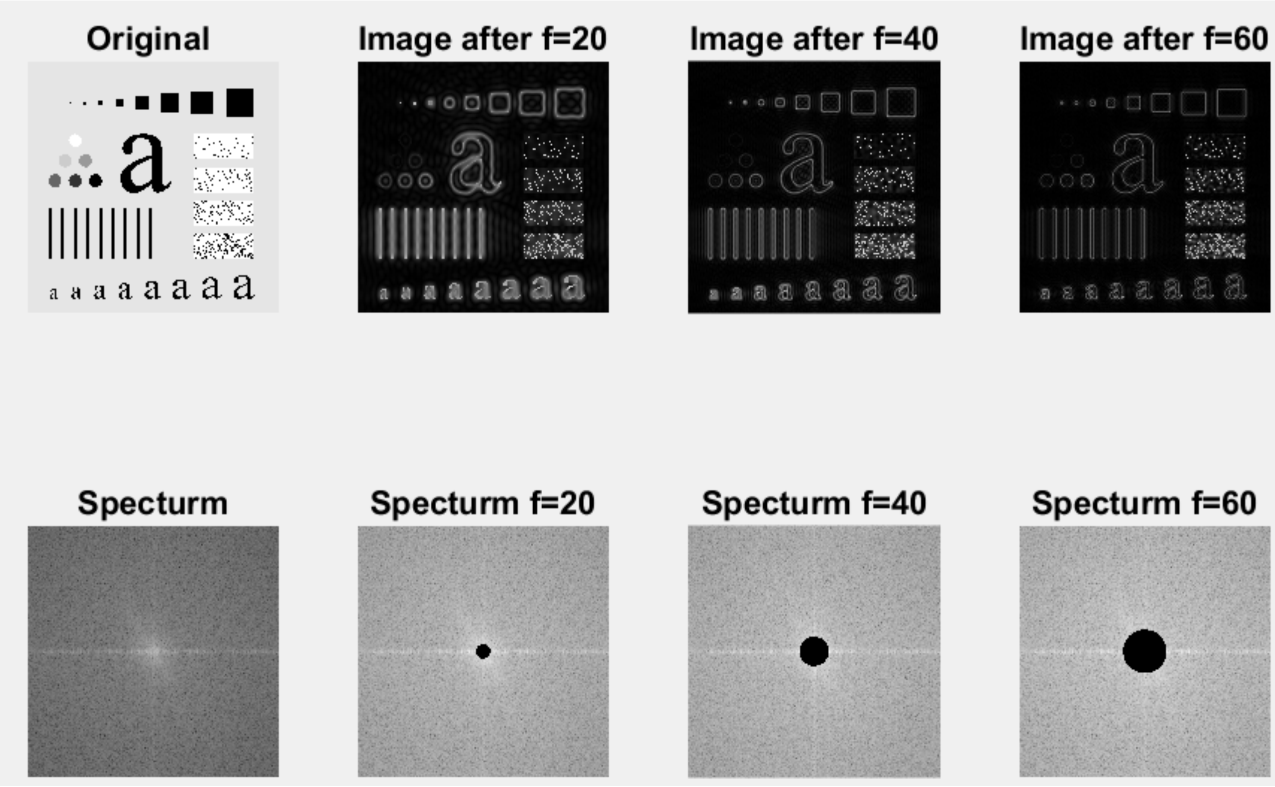
Butterworth highpass filter

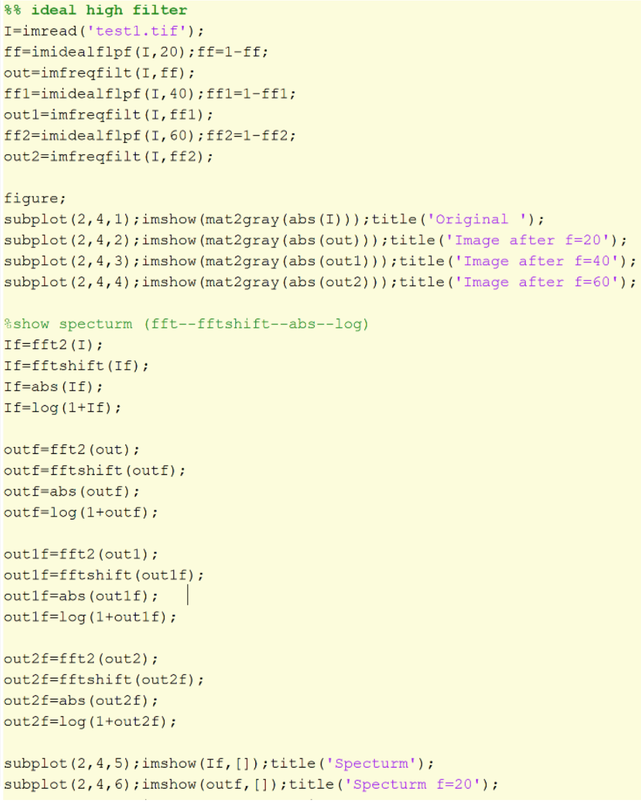


Gaussian highpass filter



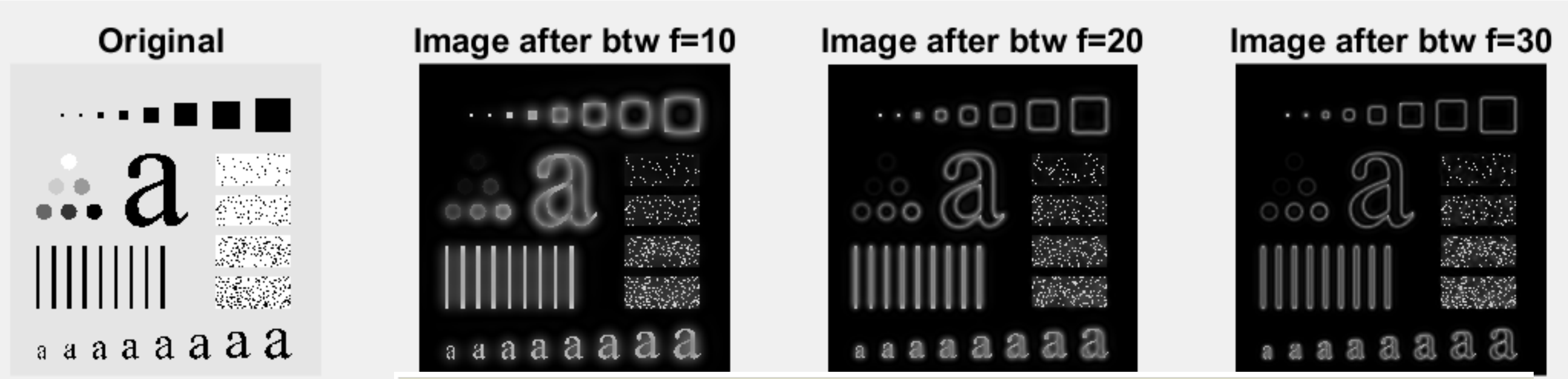
Filtering Results by IHPF

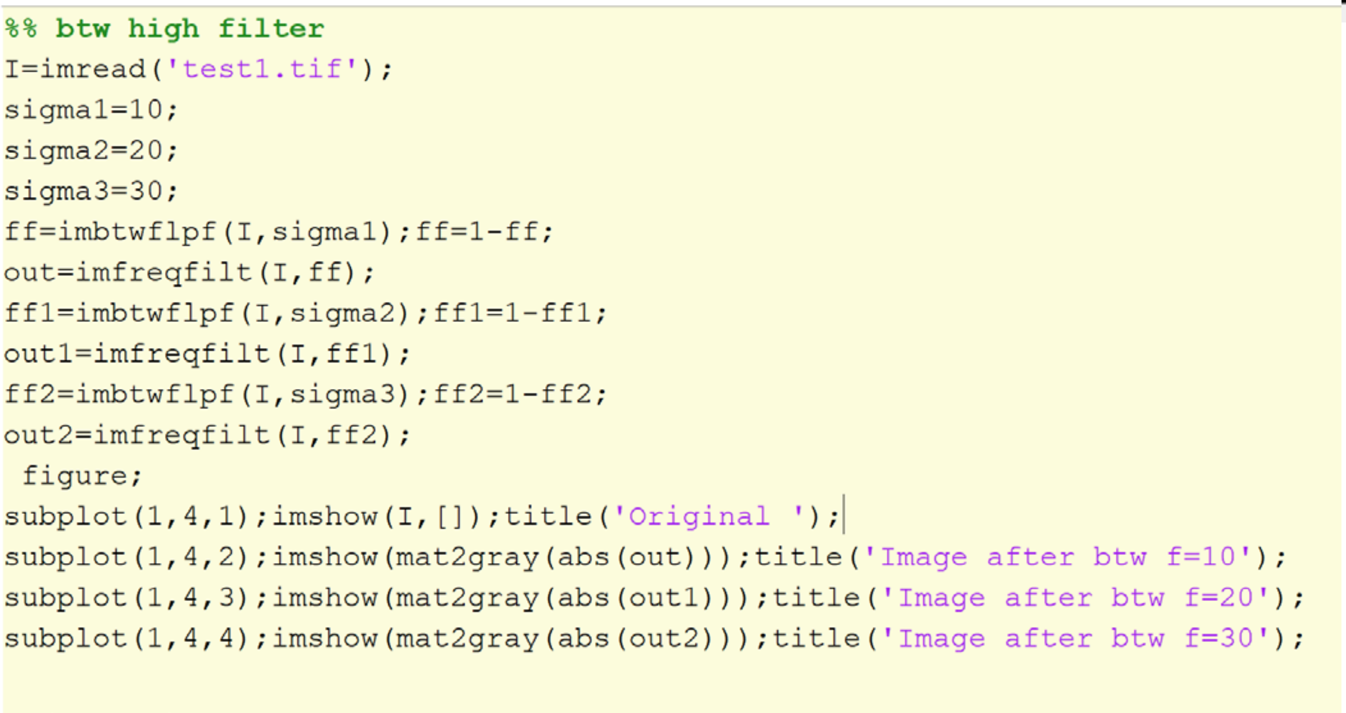




```
%% ideal high filter
I=imread('test1.tif');
ff=imideallpf(I,20);ff=1-ff;
out=imfreqfilt(I,ff);
ff1=imideallpf(I,40);ff1=1-ff1;
out1=imfreqfilt(I,ff1);
ff2=imideallpf(I,60);ff2=1-ff2;
out2=imfreqfilt(I,ff2);

figure;
subplot(2,4,1);imshow(mat2gray(abs(I)));title('Original ');
subplot(2,4,2);imshow(mat2gray(abs(out)));title('Image after f=20');
subplot(2,4,3);imshow(mat2gray(abs(out1)));title('Image after f=40');
subplot(2,4,4);imshow(mat2gray(abs(out2)));title('Image after f=60');

%show specturm (fft--fftshift--abs--log)
If=fft2(I);
If=fftshift(If);
If=abs(If);
If=log(1+If);

outf=fft2(out);
outf=fftshift(outf);
outf=abs(outf);
outf=log(1+outf);

out1f=fft2(out1);
out1f=fftshift(out1f);
out1f=abs(out1f);
out1f=log(1+out1f);

out2f=fft2(out2);
out2f=fftshift(out2f);
out2f=abs(out2f);
out2f=log(1+out2f);

subplot(2,4,5);imshow(If,[]);title('Specturm');
subplot(2,4,6);imshow(outf,[]);title('Specturm f=20');
```

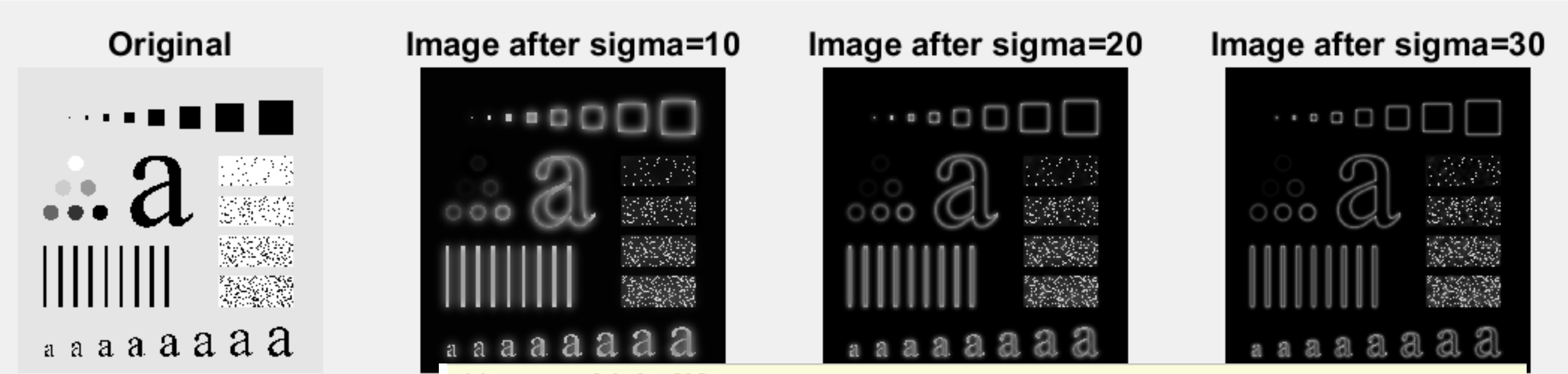
Filtering Results by BHPF





```
%% btw high filter
I=imread('test1.tif');
sigma1=10;
sigma2=20;
sigma3=30;
ff=imbtwflpf(I,sigma1);ff=1-ff;
out=imfreqfilt(I,ff);
ff1=imbtwflpf(I,sigma2);ff1=1-ff1;
out1=imfreqfilt(I,ff1);
ff2=imbtwflpf(I,sigma3);ff2=1-ff2;
out2=imfreqfilt(I,ff2);
 figure;
subplot(1,4,1);imshow(I,[]);title('Original  ');
subplot(1,4,2);imshow(mat2gray(abs(out)));title('Image after btw f=10');
subplot(1,4,3);imshow(mat2gray(abs(out1)));title('Image after btw f=20');
subplot(1,4,4);imshow(mat2gray(abs(out2)));title('Image after btw f=30');
```

Filtering Results by GHPF

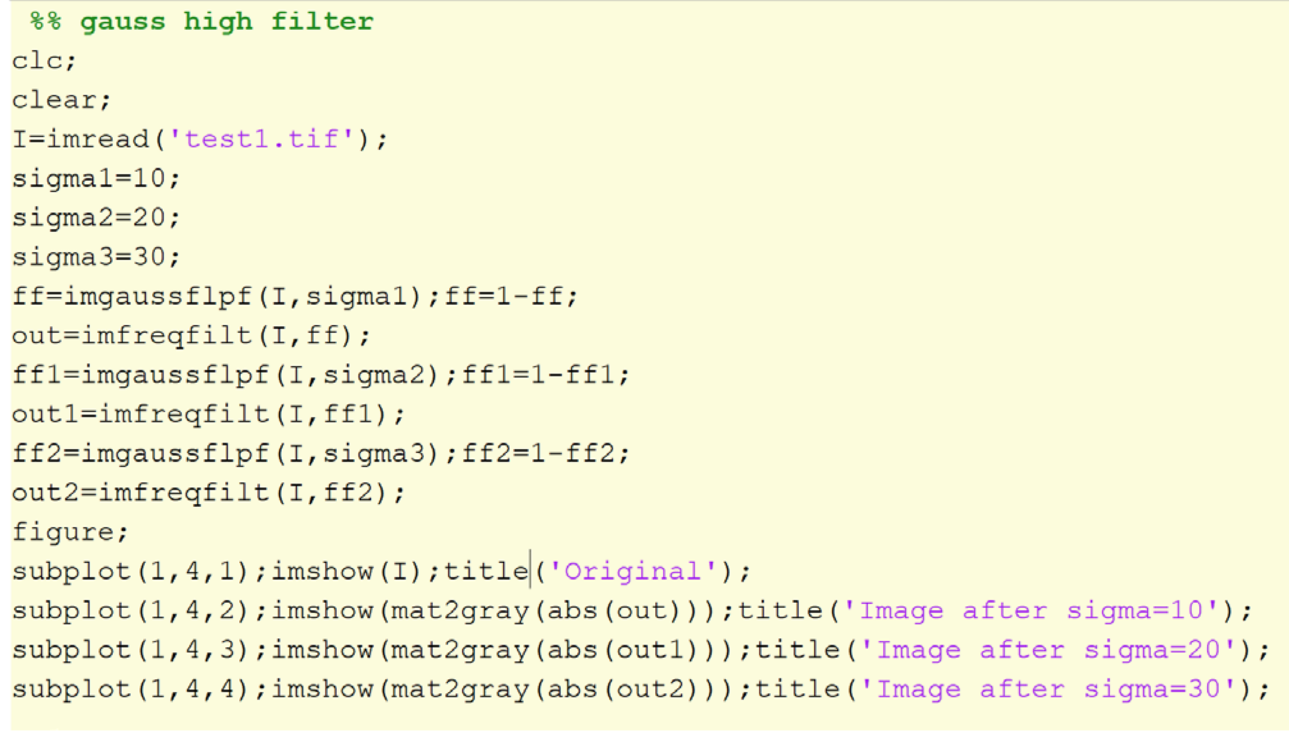


Original

Image after sigma=10

Image after sigma=20

Image after sigma=30



```
%% gauss high filter
clc;
clear;
I=imread('test1.tif');
sigma1=10;
sigma2=20;
sigma3=30;
ff=imgaussflpf(I,sigma1);ff=1-ff;
out=imfreqfilt(I,ff);
ff1=imgaussflpf(I,sigma2);ff1=1-ff1;
out1=imfreqfilt(I,ff1);
ff2=imgaussflpf(I,sigma3);ff2=1-ff2;
out2=imfreqfilt(I,ff2);
figure;
subplot(1,4,1);imshow(I);title('Original');
subplot(1,4,2);imshow(mat2gray(abs(out)));title('Image after sigma=10');
subplot(1,4,3);imshow(mat2gray(abs(out1)));title('Image after sigma=20');
subplot(1,4,4);imshow(mat2gray(abs(out2)));title('Image after sigma=30');
```

Lecture Outline

- Image Enhancement in Frequency Domain
- Filtering in Frequency Domain
  - Low-Pass Filtering
  - High-Pass Filtering
  - Laplacian Filtering
  - Homomorphic Filtering
  - Selective Filtering – Bandpass/Bandreject, Notch Filters

The Laplacian in Spatial Domain

- The Laplacian is defined as follows:

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}$$

- where the partial 2nd order derivative in the *x* direction is defined as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

- and in the *y* direction as follows:

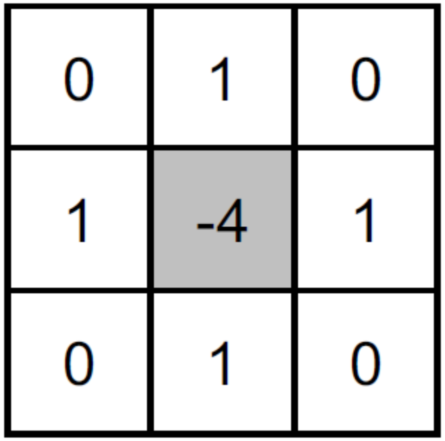
$$\frac{\partial^2 f}{\partial^2 y} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

The Laplacian in Spatial Domain

- So, the Laplacian can be given as follows:

- $\nabla^2 f = [\, f(x+1, y) + f(x-1, y)$
$+ f(x, y+1) + f(x, y-1) - 4f(x, y)$

- We can easily build a filter based on this



0	1	0
1	-4	1
0	1	0

Laplacian Filter in Frequency Domain

$$\mathcal{F}\left[\frac{d^n f(x)}{dx^n}\right] = (ju)^n F(u)$$

- The Fourier transform of Laplacian equation is

$$\begin{aligned}\mathcal{F}\left[\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}\right] &= (ju)^2 F(u,v) + (jv)^2 F(u,v) \\ &= -(u^2+v^2) F(u,v)\end{aligned}$$

$$\mathcal{F}[\nabla^2 f(x,y)] = -(u^2+v^2)F(u,v)$$

- This means that the Laplacian filter in frequency domain can be express as

$$H(u,v) = -(u^2+v^2)$$

Laplacian Filter in Frequency Domain

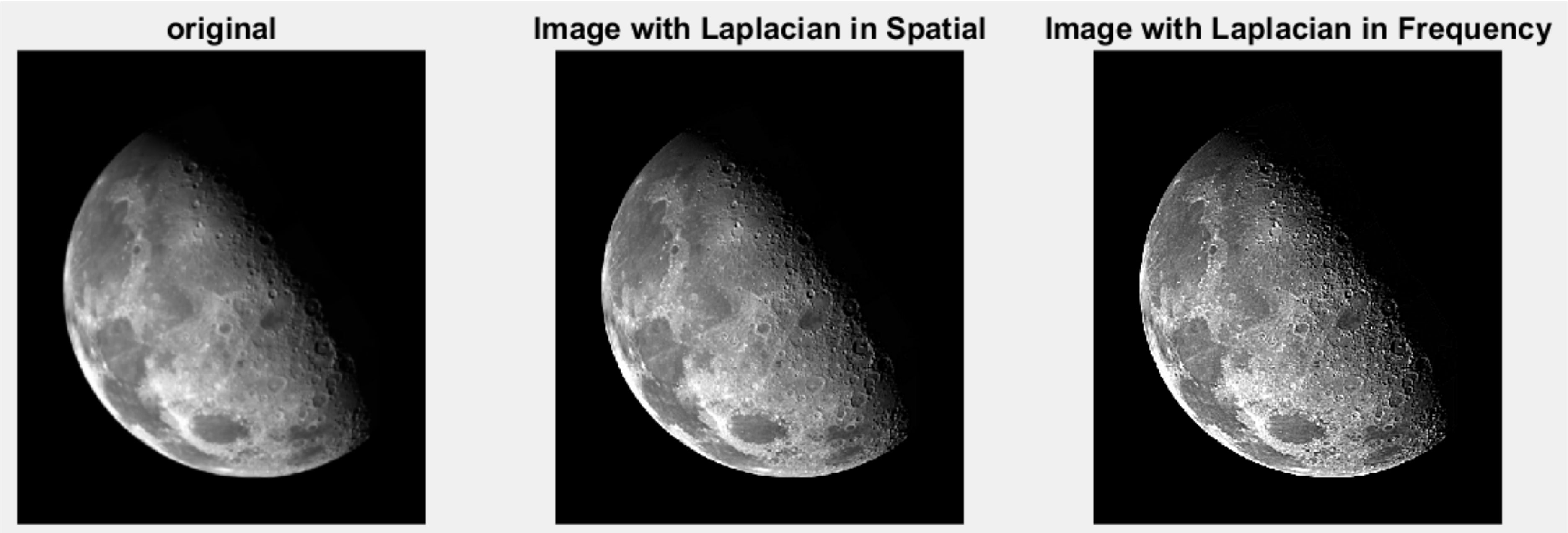
- As the filter origin was shifted to the center of image, we may shift the Laplacian filter in frequency domain by M/2 and N/2 respectively

$$H(u,v) = -\left[(u - M/2)^2 + (v - N/2)^2\right]$$

- Dual relationship in the familiar Fourier transform-pair notation:

$$\nabla^2 f(x,y) \Leftrightarrow -\left[(u - M/2)^2 + (v - N/2)^2\right]F(u,v)$$

Example: Laplacian Filtering



original

Image with Laplacian in Spatial

Image with Laplacian in Frequency

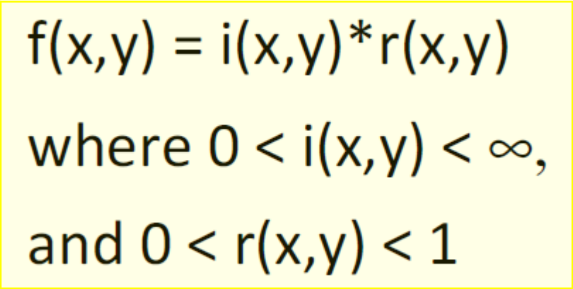
Lecture Outline

- Image Enhancement in Frequency Domain
- Filtering in Frequency Domain
    - Low-Pass Filtering
    - High-Pass Filtering
    - Laplacian Filtering
    - Homomorphic Filtering
    - Selective Filtering – Bandpass/Bandreject, Notch Filters

Homomorphic Filtering

- Illumination-reflection Model:
  - Let $f(x,y)$ be non-zero and finite image that $0 < f(x,y) < \infty$, $f(x,y)$ may be characterized by components:
  - Source illumination ($i(x,y)$) incident on the scene being viewed
  - Amount of illumination reflection ($r(x,y)$) by the objects in the scene

- The two functions combine as a product to form



$f(x,y) = i(x,y)*r(x,y)$

where $0 < i(x,y) < \infty$,

and $0 < r(x,y) < 1$

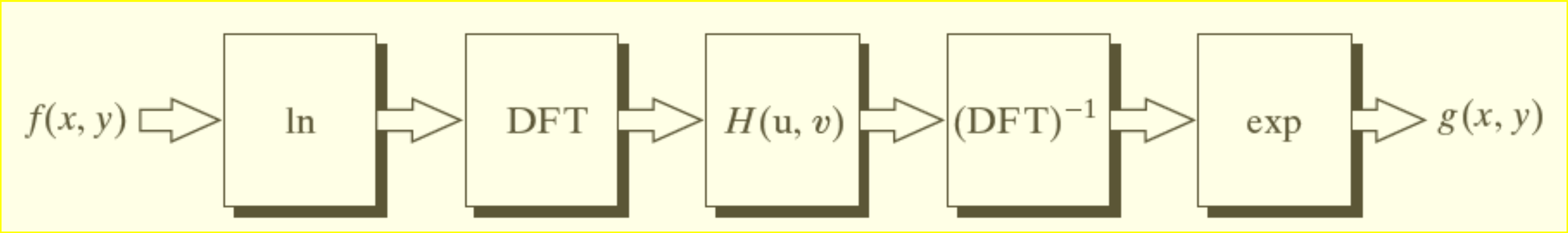
Homomorphic Filtering

- Illumination and Reflection have different characteristics:
  - Illumination components tend to be slow in spatial variation (low frequency components)
  - Reflection of various objects tends to vary abruptly (high frequency components)

- Better control can be achieved if the two components are separated by log function and filters are applied separately to each of the respective components.

Homomorphic Filtering

- The whole process can be summarized as follows



$$z(x, y) = \ln(f(x, y)) = \ln(i(x, y)) + \ln(r(x, y))$$

$$Z(u, v) = \Im\{z(x, y)\} = \Im\{\ln(i(x, y))\} + \Im\{\ln(r(x, y))\}$$

$$= F_i(u, v) + F_r(u, v)$$

$$S(u, v) = H(u,v)Z(u, v) = H(u, v)F_i(u, v) + H(u, v)F_r(u, v)$$

$$s(x, y) = \Im^{-1}\{H(u, v)F_i(u, v)\} + \Im^{-1}\{H(u, v)F_r(u, v)\} = i'(x, y) + r'(x, y)$$

$$g(x, y) = e^{s(x, y)} = e^{i'(x, y)}e^{r'(x, y)} = i_0(x, y)r_0(x, y)$$

Homomorphic Filtering

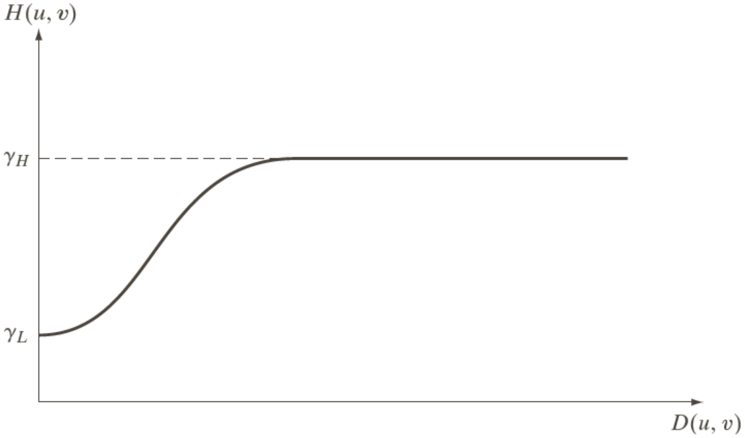
- Filter H(u,v) can be designed such that it tends to decrease the contribution made by low frequencies (illumination, $\gamma_L < 1$) and amplify the contribution made by high frequencies (reflectance, $\gamma_H > 1$)
- The result is simultaneous dynamic range compression and contrast enhancement

Homomorphic Filtering

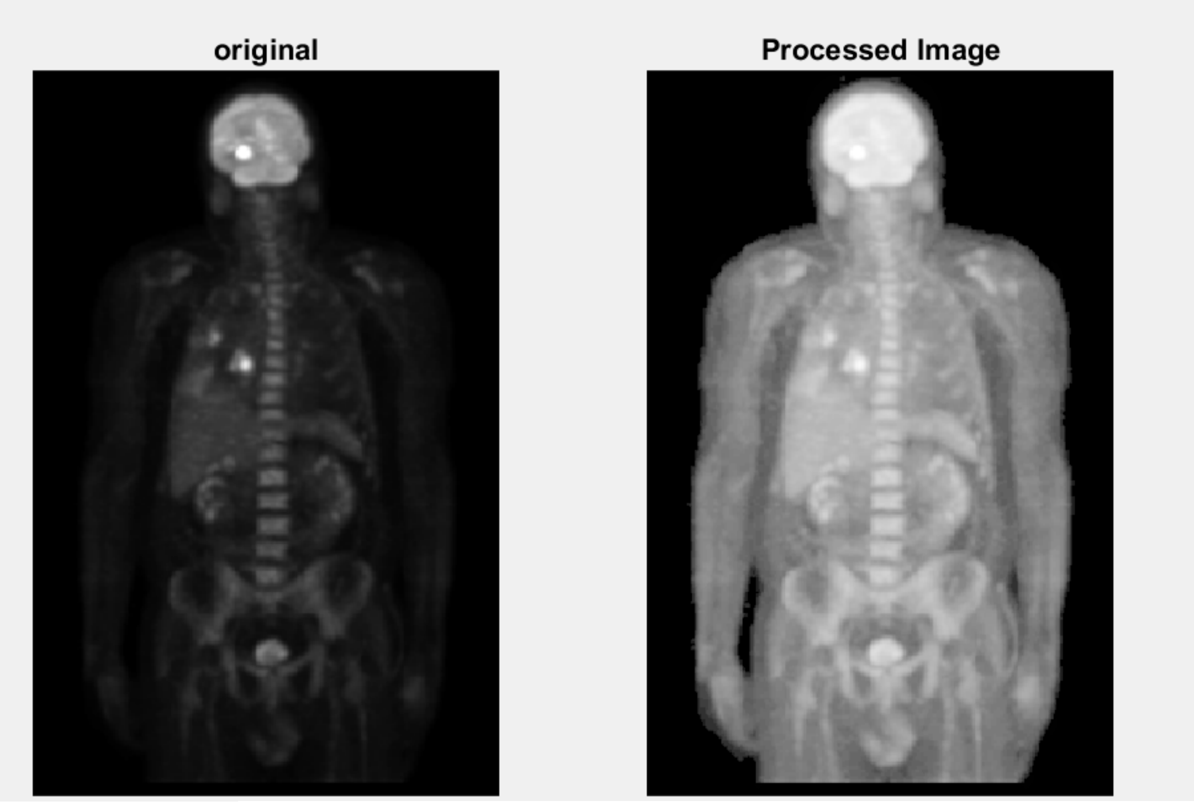
- The curve shape can be approximated using the basic form of any of the ideal highpass filters. Example: using a slightly-modified form of GHPF gives:

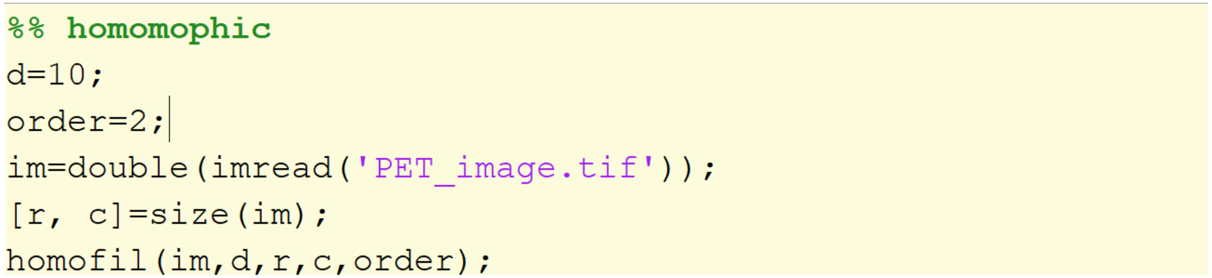
$$H(u,v) = (\gamma_H - \gamma_L)[1 - e^{-cD^2(u,v)/D_0^2}] + \gamma_L$$

- where constant *c* controls the sharpness of the slope of the filter function as it transitions between γ_L and $\gamma_{H.}$



Example: Homomorphic Filtering





```
%% homomophic
d=10;
order=2;
im=double(imread('PET_image.tif'));
[r, c]=size(im);
homofil(im,d,r,c,order);
```

Lecture Outline

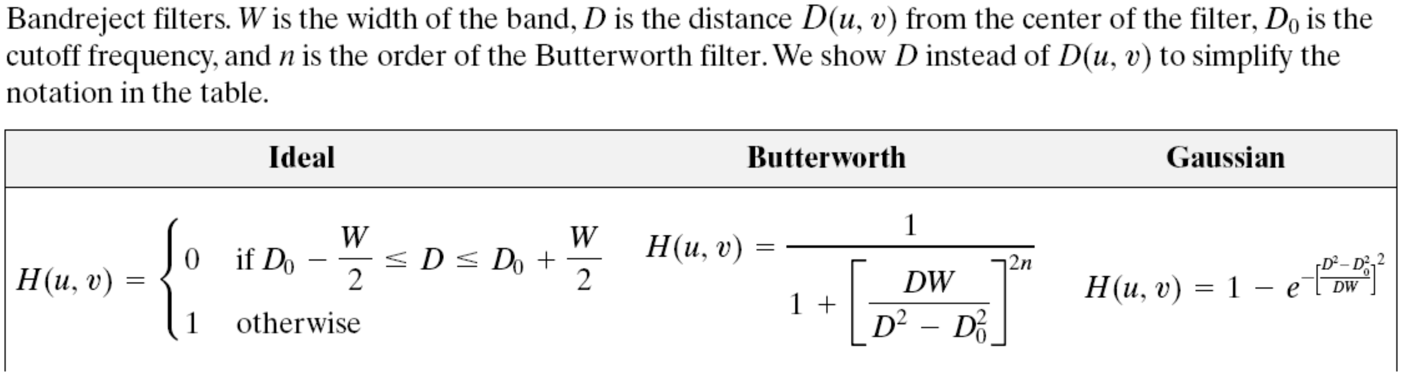
- Image Enhancement in Frequency Domain
- Filtering in Frequency Domain
    - Low-Pass Filtering
    - High-Pass Filtering
    - Laplacian Filtering
    - Homomorphic Filtering
    - Selective Filtering – Bandpass/Bandreject, Notch Filters

Selective Filtering

- The previous filters all operate over the entire frequency rectangle. However, some applications only require processing of specific bands of frequencies or small regions of the frequency rectangle
  - Bandreject / Bandpass Filters: process specific bands
  - Notch Filters: process small regions of the frequency rectangle

Selective Filtering:

- Based on previous types of filters (Ideal, Butterworth, Gaussian), Bandreject filters can be constructed by adding an addition parameter – width of the band, *W*



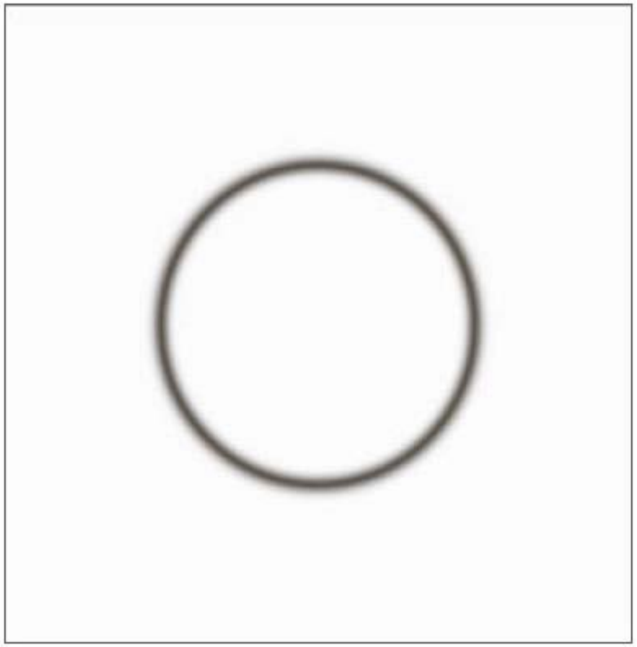
Bandreject filters. W is the width of the band, D is the distance $D(u, v)$ from the center of the filter, D_0 is the cutoff frequency, and n is the order of the Butterworth filter. We show D instead of $D(u, v)$ to simplify the notation in the table.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

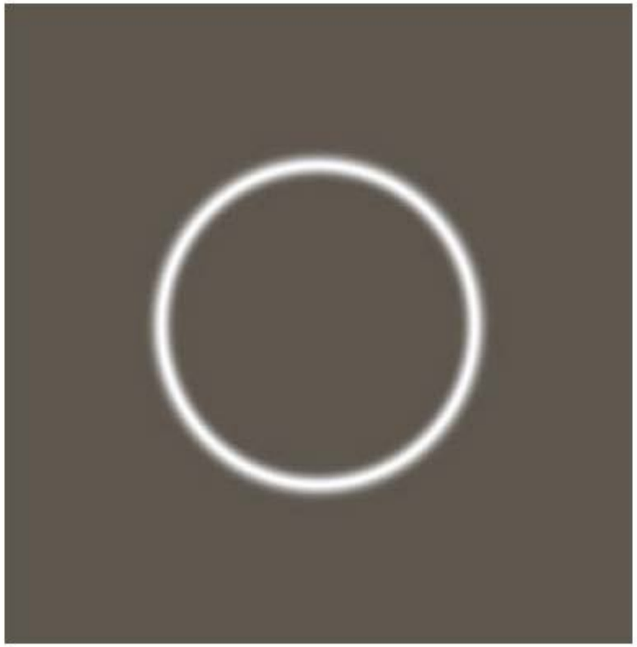
- Equivalent Bandpass filters can be obtained from a bandreject filter by inversing its effect

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

Selective Filtering



Bandreject Gaussian filter



Corresponding bandpass filter

Notch Filters

- **A filter that rejects (or passes) specific frequencies**
- Example: periodic noise corresponds to spikes or lines in the Fourier domain
- Can design a filter with zeros at those frequencies, this will remove the noise
- Examples:
  - Image mosaics
  - Scan line noise
  - Halftoning noise (moire patterns)

Steps in Notch Filtering

Look at spectrum $|F(u,v)|$ of noisy image $f(x,y)$, find frequencies corresponding to the noise

Create a mask image $M(u,v)$ with notches (zeros) at those places, 1's elsewhere

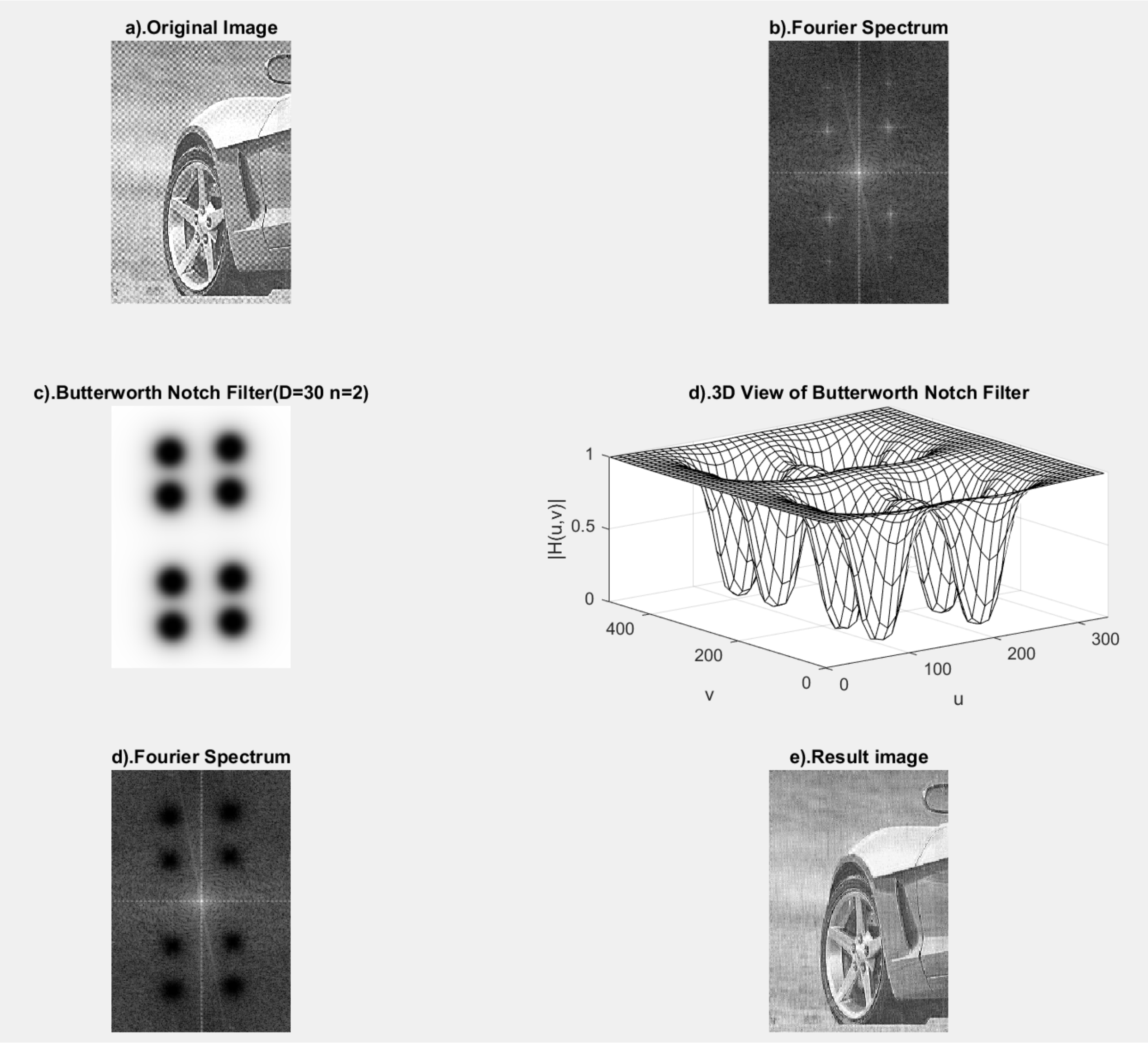
Multiply mask with original image transform; this zeros out noise frequencies

$$\boxed{G(u,v) = M(u,v)\, F(u,v)}$$

Take inverse Fourier transform to get restored image

$$\boxed{g(x,y) = \mathcal{F}^{-1}(G(u,v))}$$

Example Notch Filters



a).Original Image

b).Fourier Spectrum

c).Butterworth Notch Filter(D=30 n=2)

d).3D View of Butterworth Notch Filter

d).Fourier Spectrum

e).Result image

Summary

- An “ideal lowpass filter” passes all frequencies with magnitudes below a specific level, and attenuates all frequencies above that level.
- An “ideal highpass filter” does the opposite.
- A “notch” filter rejects (or passes) frequencies at a specific poin (the notch).