Sharpening Filters

- The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as an natural effect of a particular method of image acquisition.
  – smoothing → integration
  – Sharpening → differentiation

Derivatives

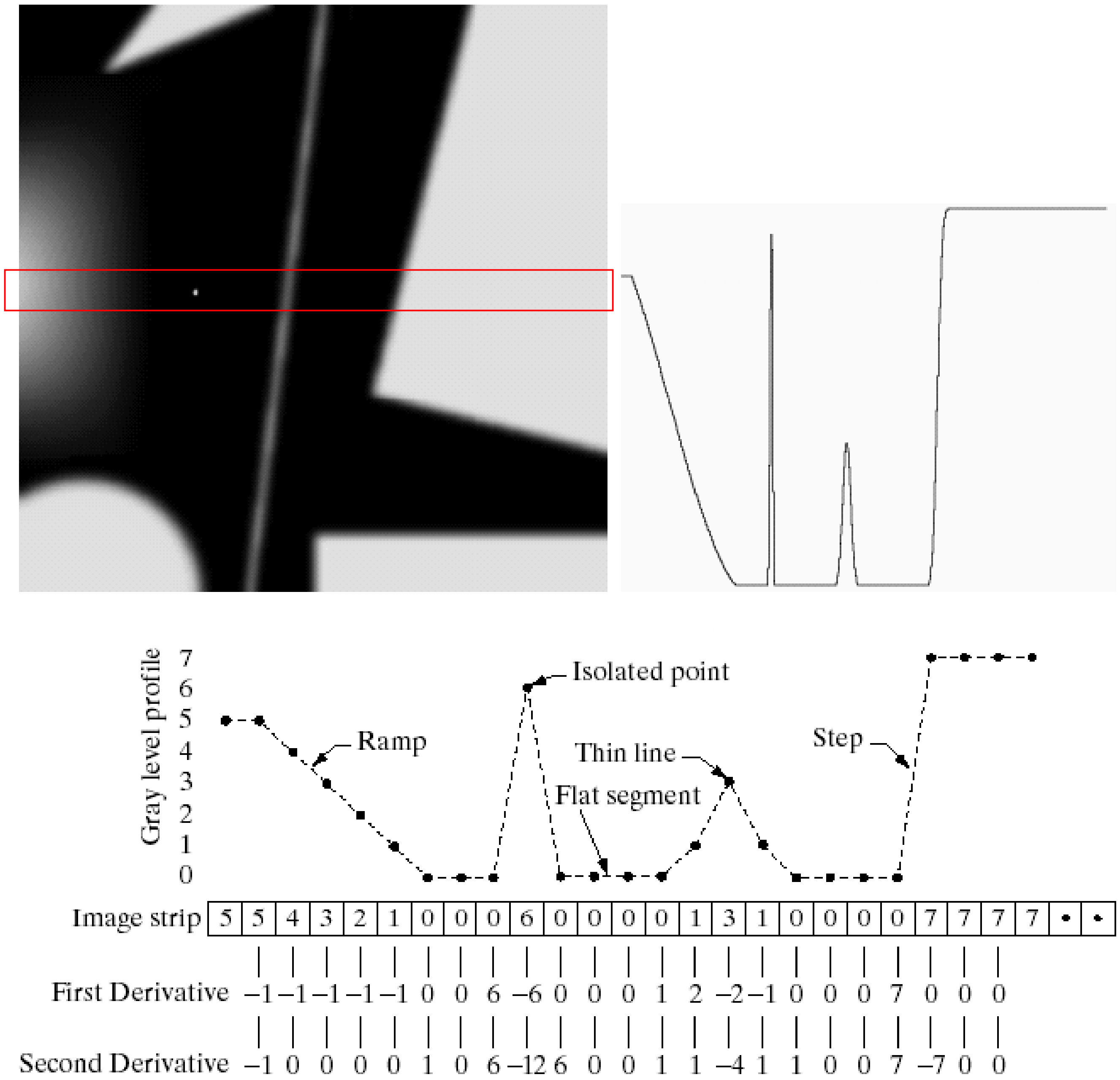
- Definition for a first derivative
  - Must be zero in flat segments
  - Must be nonzero at the onset of a gray-level step or ramp; and
  - Must be nonzero along ramps.
- A basic definition of the first-order derivative of a one-dimensional function f(x) is

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

Derivatives

- Definition for a second derivative
  - Must be zero in flat areas;
  - Must be nonzero at the onset and end of a gray-level step or ramp;
  - Must be zero along ramps of constant slope
- We define a second-order derivative as the difference

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



a b
c

FIGURE 3.38 (a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).

Observations

- The 1^{st}-order derivative is nonzero along the entire ramp, while the 2^{nd}-order derivative is nonzero only at the onset and end of the ramp.
- 1st make thick edge and 2nd make thin edge
- The response at and around the point is much stronger for the 2^{nd}- than for the 1^{st}-order derivative

The Gradient

- First Derivatives in image processing are implemented using the magnitude of the gradient.
- For a function *f*(x, y), the gradient of *f* at coordinates (x, y) is defined as the two-dimensional column *vector*

$$\nabla F = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude of this vector is given by

$$\nabla f = mag(\nabla F) = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

G_x

-1	1

G_y

1
-1

Robert's Method

$G_x = (z_9 - z_5)$ and $G_y = (z_8 - z_6)$

$$\nabla f = \sqrt{(z_9 - z_5)^2 + (z_8 - z_6)^2}$$

Roberts Cross-Gradient Operators

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

Mask of even size are awkward to apply so the smallest filter mask should be 3x3.

Sobel's Method

$$\nabla f \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

a
b c
d e

FIGURE 3.44 A 3 × 3 region of an image (the z's are gray-level values) and masks used to compute the gradient at point labeled z_5. All masks coefficients sum to zero, as expected of a derivative operator.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Roberts Cross-Gradient

−1	0
0	1

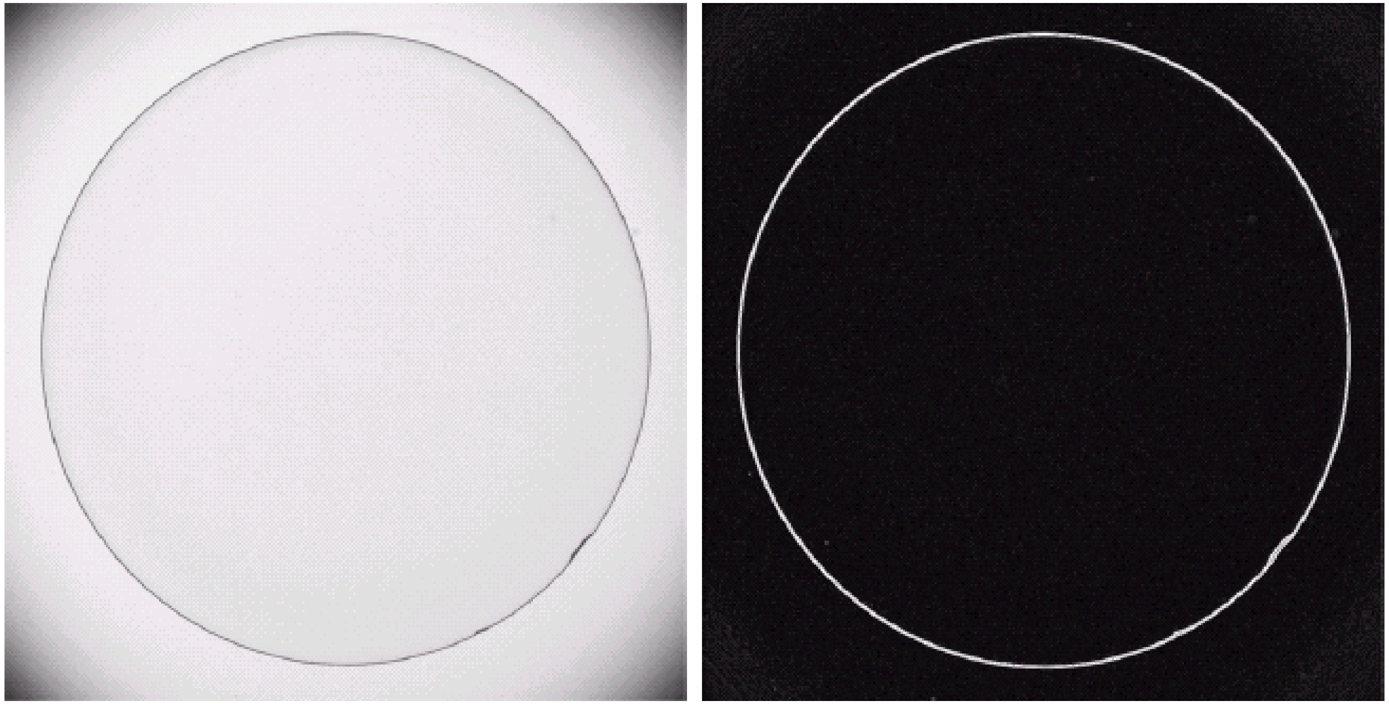
0	−1
1	0

Sobel Operators

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Example



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Laplacian

- 2^{nd} order derivative
- Isotropic filters: rotation invariant
- Linear operator
- Define as

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

Discrete version:

f(x-1,y)	*f(x,y)*	*f(x+1,y)*

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

f(x,y-1)
f(x,y)
f(x,y+1)

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

The digital implementation of the 2-Dimensional Laplacian is obtained by summing 2 components

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

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

0	1	0
1	-4	1
0	1	0

1	0	1
0	-4	0
1	0	1

The basic way in which we use the Laplacian for image enhancement is as follows:

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \text{If the center coefficient is negative} \\ f(x,y)+\nabla^2 f(x,y) & \text{If the center coefficient is positive} \end{cases}$$

Where f(x,y) is the original image

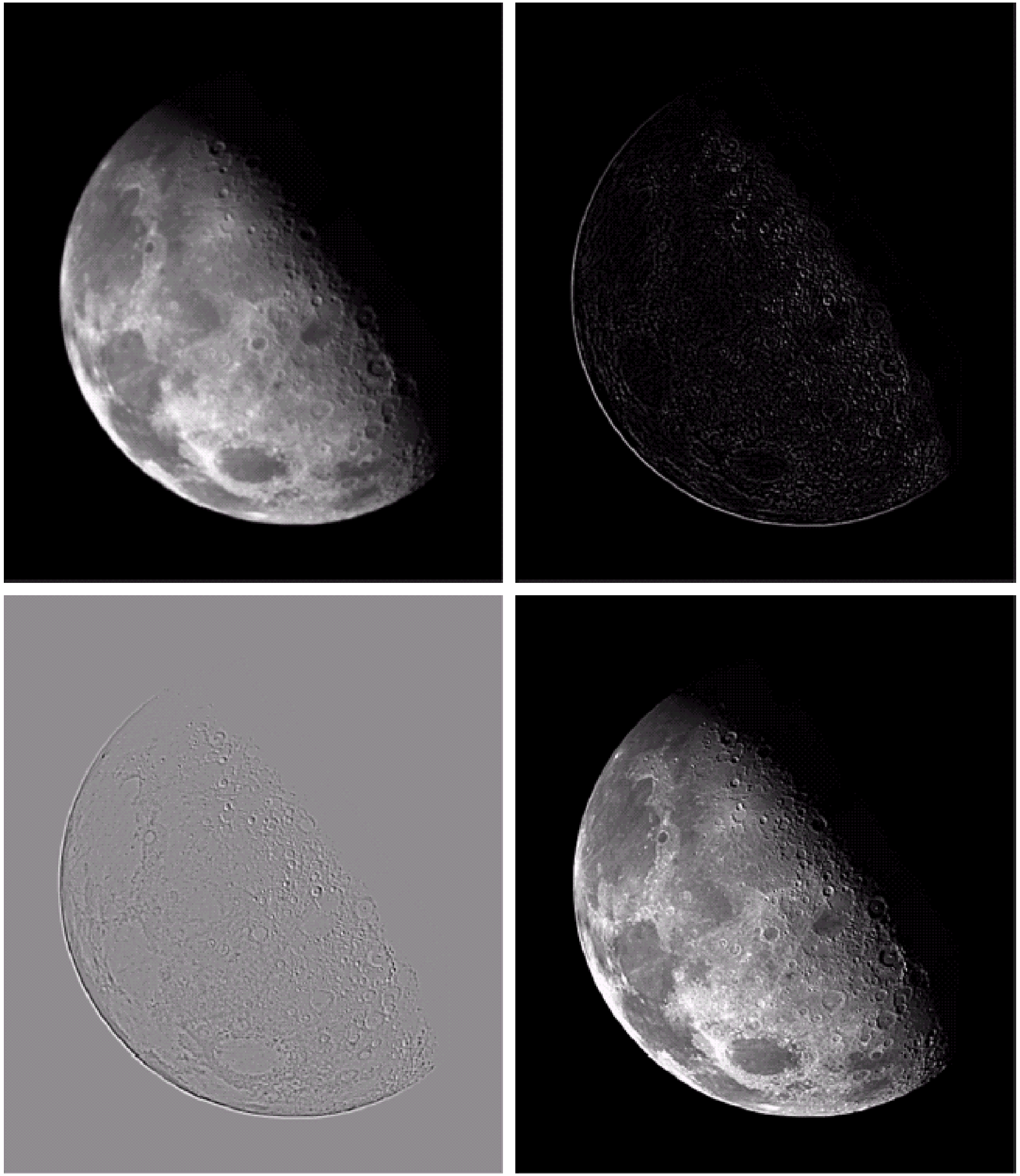
$\nabla^2 f(x,y)$ is Laplacian filtered image

g(x,y) is the sharpen image

Simplification：

$$g(x,y)=f(x,y)-f(x+1,y)-f(x-1,y)-f(x,y+1)-f(x,y-1)+4f(x,y)$$

$$g(x,y)=5f(x,y)-f(x+1,y)-f(x-1,y)-f(x,y+1)-f(x,y-1)$$

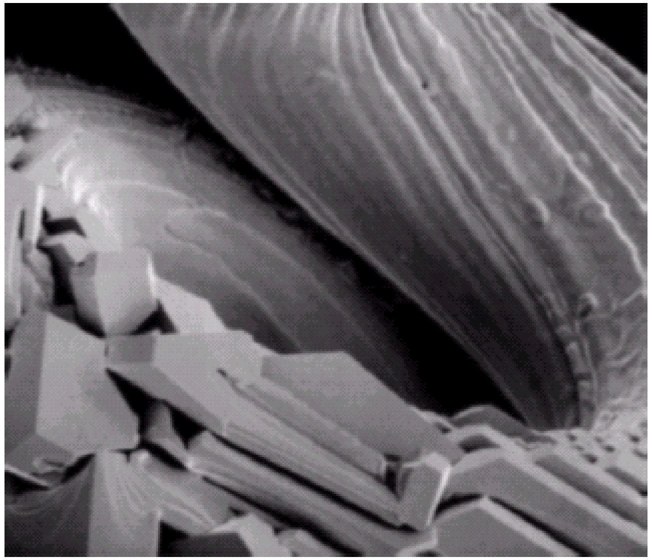


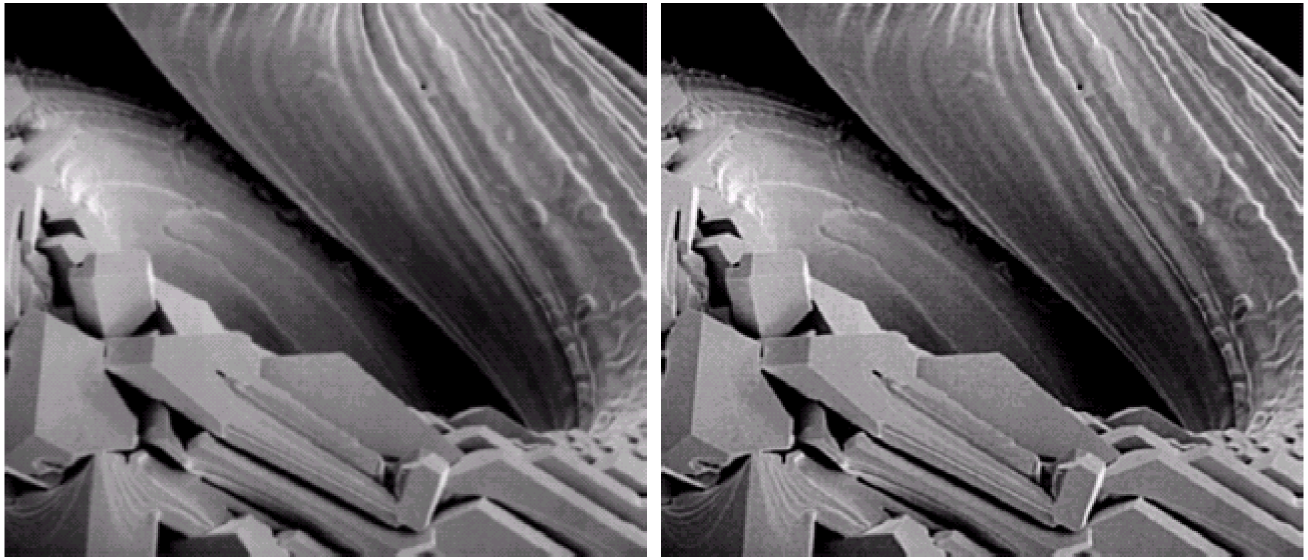
a b
c d

FIGURE 3.40 (a) Image of the North Pole of the moon. (b) Laplacian-filtered image. (c) Laplacian image scaled for display purposes. (d) Image enhanced by using Eq. (3.7-5). (Original image courtesy of NASA.)

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1





a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Unsharp masking and high-boost filtering

- A process used for many years in the publishing industry to sharpen images consists of subtracting a blurred version of an image from the image itself. This process, called *unsharp masking*, is expressed as

$$f_s(x, y) = f(x, y) - \bar{f}(x, y)$$

Where $f_s(x, y)$ denotes the sharpened image obtained by unsharp masking, and $\bar{f}(x, y)$ is a blurred version of $f(x, y)$

Unsharp masking and high-boost filtering

- A slight further generalization of unsharp masking is called *highboost filtering*. A high-boost filtered image, f_{hb}, is defined at any point (x, y) as

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y) \quad where\ A \geq 1$$

$$f_{hb}(x,y) = (A-1)f(x,y) + f(x,y) - \bar{f}(x,y)$$

$$f_{hb}(x,y) = (A-1)f(x,y) + f_s(x,y)$$

This equation is applicable general and does not state explicitly how the sharp image is obtained

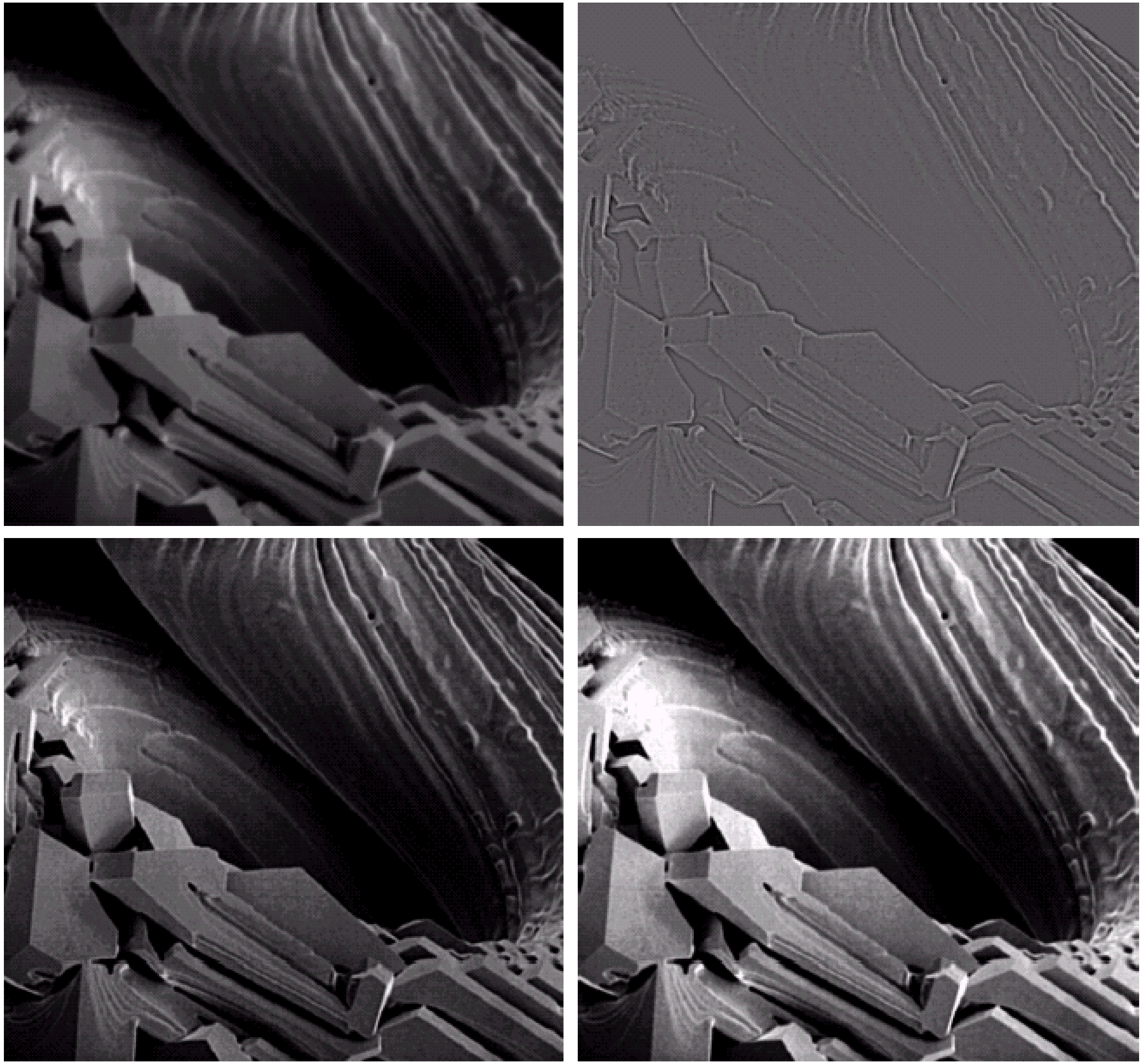
Unsharp masking and high-boost filtering

- If we elect to use the Laplacian, then we know that $f_{hb}(x, y)$ can be obtained

$$f_{hb} = \begin{cases} Af(x,y) - \nabla^2 f(x,y) & \textit{If the center coefficient is negative} \\ Af(x,y) + \nabla^2 f(x,y) & \textit{If the center coefficient is positive} \end{cases}$$

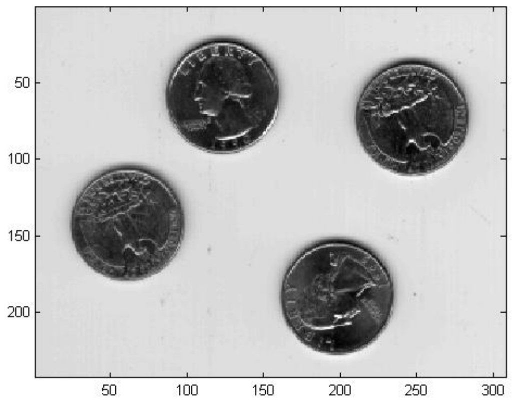
0	-1	0
-1	A+4	-1
0	-1	0

-1	-1	-1
-1	A+8	-1
-1	-1	-1

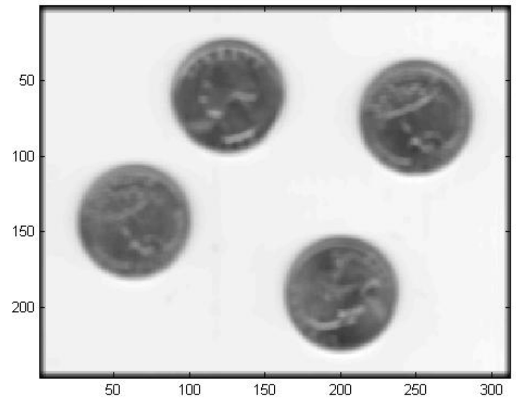


a	b
c	d

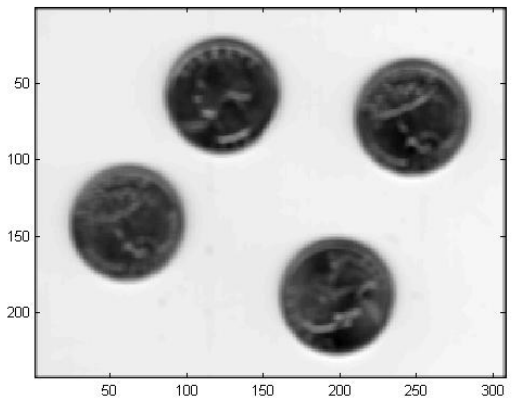
FIGURE 3.43
(a) Same as Fig. 3.41(c), but darker.
(a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$.
(c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.



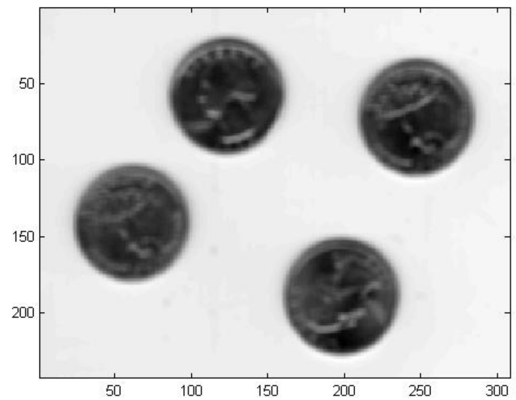
A (Original image)



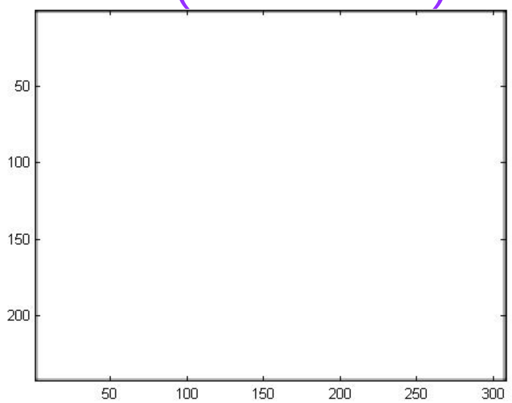
B (convolution)



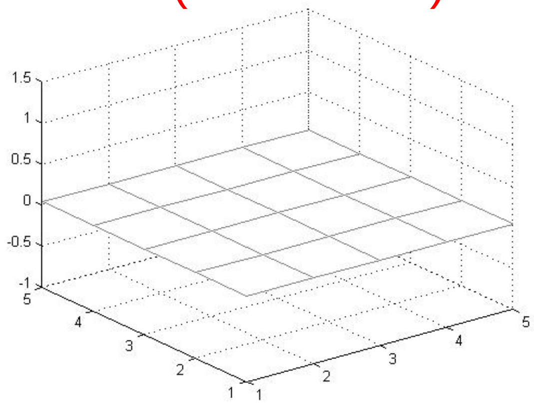
B (convolution)



B (convolution)

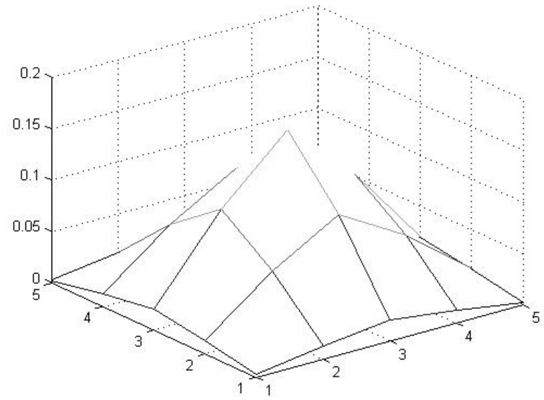


C1

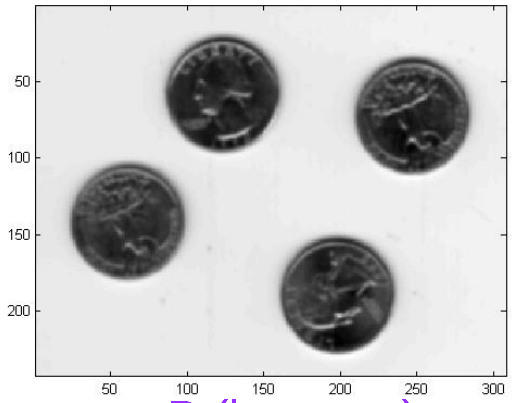


M

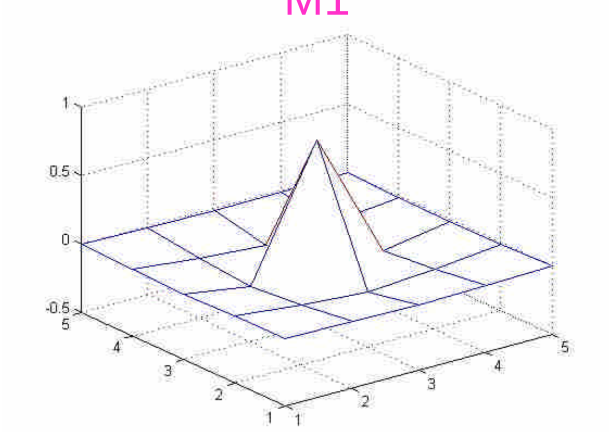
```
A=imread('eight.tif');
A=double(A);
imagesc(A)
colormap(gray)
M=ones(5)/25;
B=conv2(A,M);
figure
imagesc(B)
colormap(gray)
B=conv2(A,M,'same');
imagesc(B)
colormap(gray)
C=ones(size(B));
B=conv2(A,M,'same')./
conv2(C,M,'same');
imagesc(B)
colormap(gray)
C1=conv2(C,M,'same');
figure
colormap(gray)
imagesc(C1*255)
mesh(M)
```



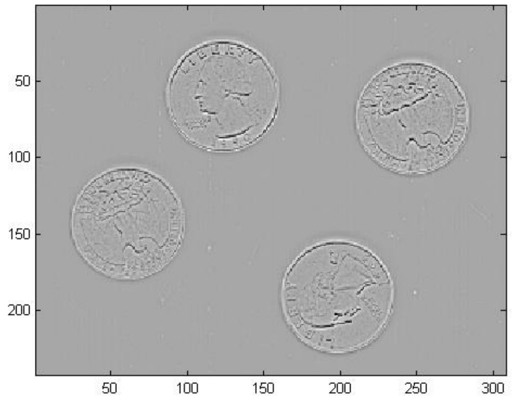
M1



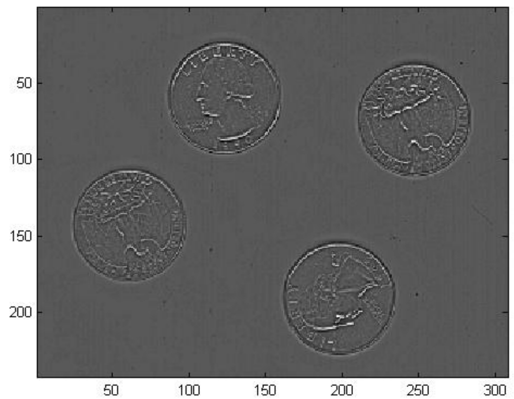
B (lowpass)



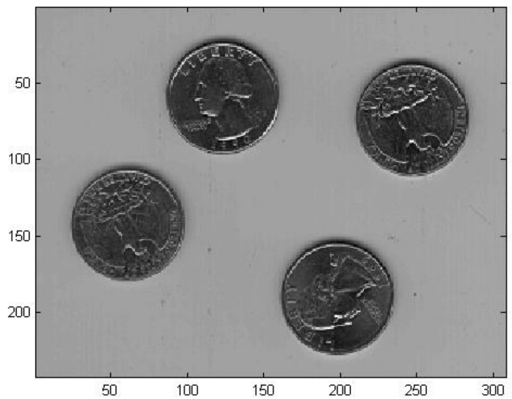
M2



B (highpass)



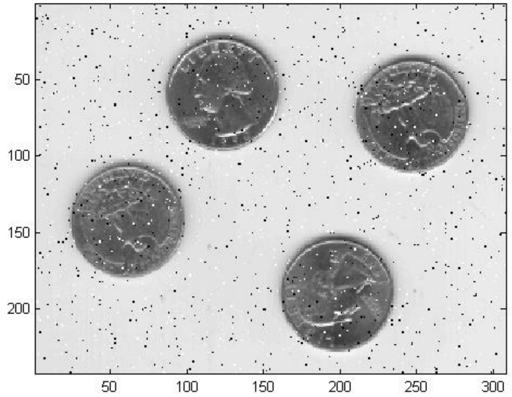
C (highpass=1-lowpass)



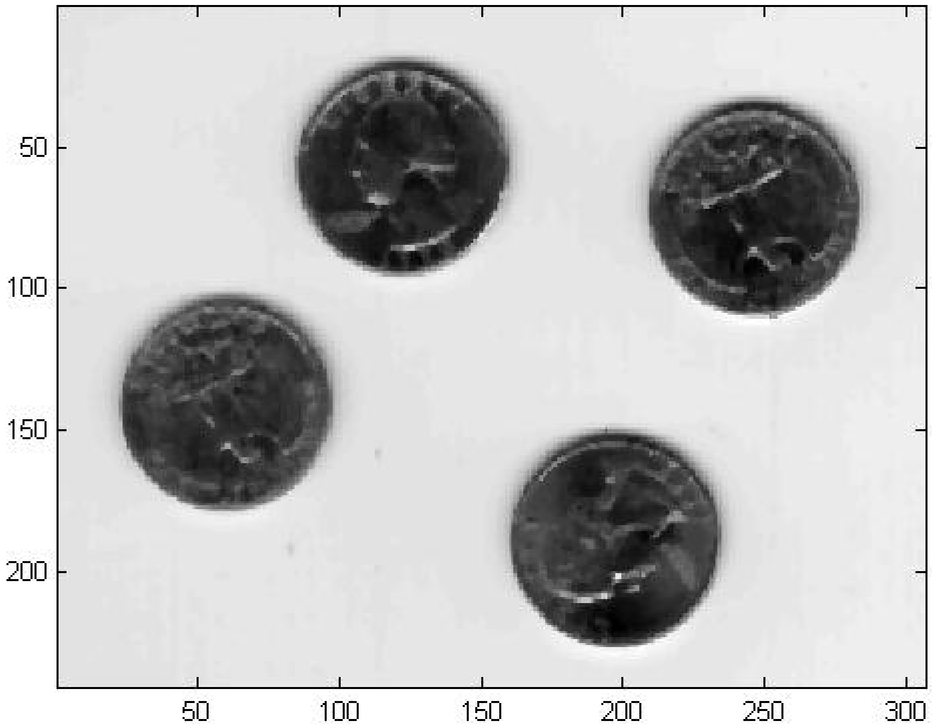
C+A (highboost)

```
f=[.05 .25 .4 .25 .05];
M1=f'*f;
mesh(M1)
B=conv2(A,M1,'same')./
conv2(C,M1,'same');
figure
imagesc(B)
colormap(gray)
D=zeros(5);
D(3,3)=1;
M2=D-M1;
mesh(M2)
B=conv2(A,M2,'same')./
conv2(C,M2,'same');
imagesc(B)
B=conv2(A,M1,'same')./
conv2(C,M1,'same');
C=A-B;
imagesc(C)
imagesc(C+A)
```

I (Original image)

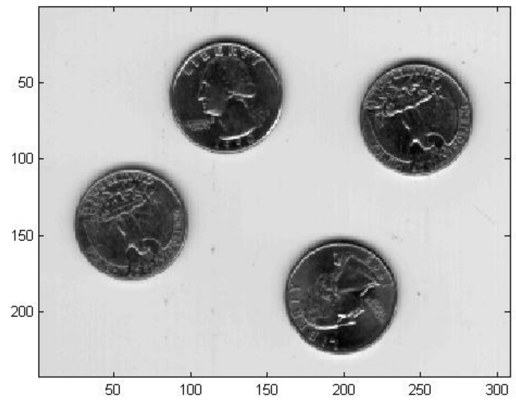


J (salt & pepper noise)

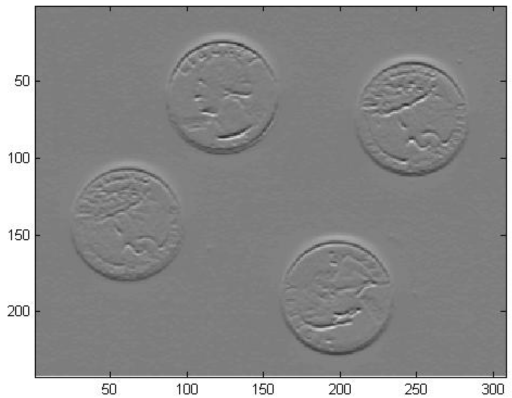


G (median filter)

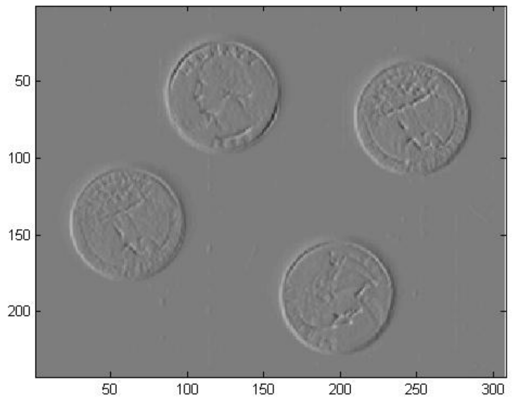
```
I = imread('eight.tif');
J = imnoise(I,'salt & pepper', 0.02);
imagesc(I), figure, imagesc(J)
J=double(J);
for i=1:240
for j=1:306
temp=J(i:i+2,j:j+2);
temp1=sort(temp(:));
G(i,j)=temp1(5);
end,end
figure
colormap(gray)
imagesc(G)
```



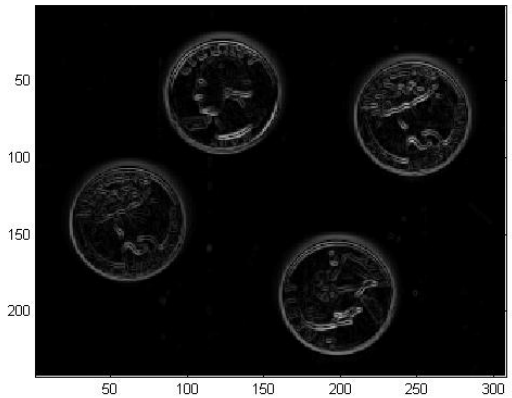
Original image



A1(Sobel1)



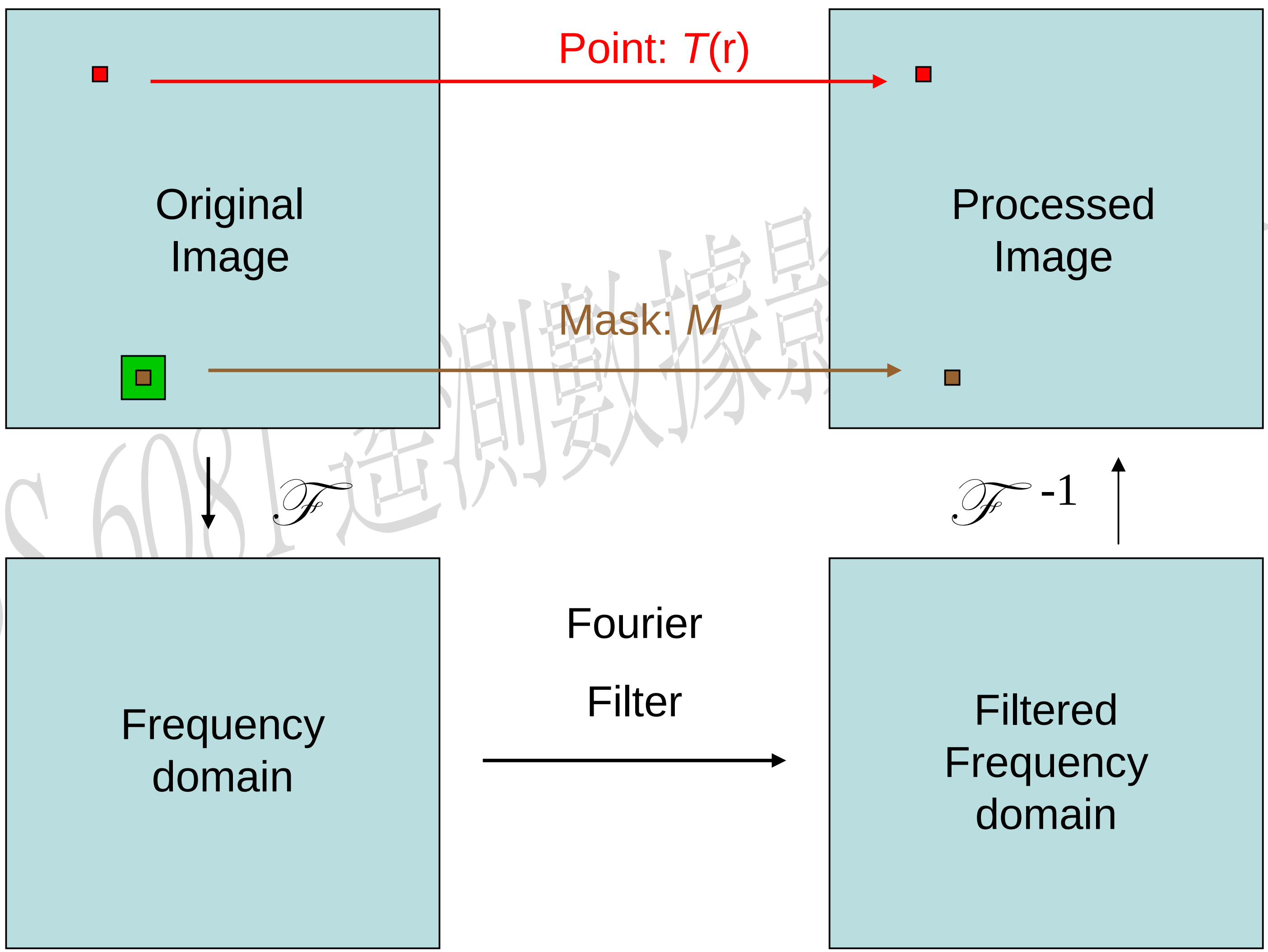
A2(Sobel2)



A1+A2

```
sobel1=[-1 -2 -1;0 0 0;1 2 1];
sobel2=sobel1‘;
A1=conv2(A,sobel1,'same');
A2=conv2(A,sobel2,'same');
figure
imagesc(A1)
colormap(gray)
figure
imagesc(A2)
colormap(gray)
imagesc(sqrt((A1).^2+(A2).^2))
```

Image Enhancement



Fourier Transform

- Fourier: a periodic function can be represented by the sum of sines/cosines of different frequencies, multiplied by a different coefficient (Fourier series)
- Non-periodic functions can also be represented as the integral of sines/cosines multiplied by weighing function (Fourier transform)

Fourier Transform

- f(x): continuous function of a real variable x
- Fourier transform of f(x):

$$\Im\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx =$$

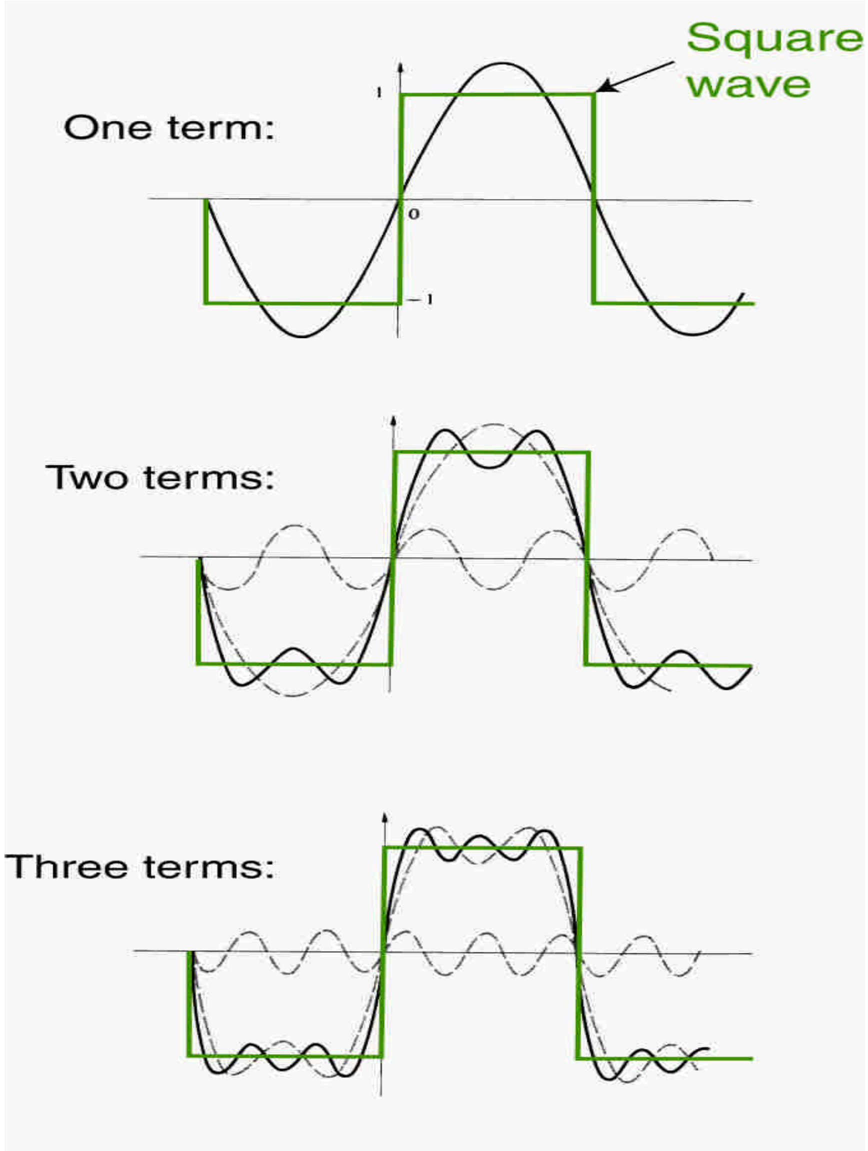
$$= \int_{-\infty}^{\infty} f(x)(\cos 2\pi ux - i \sin 2\pi ux) dx =$$

$$= \int_{-\infty}^{\infty} f(x) \underbrace{\cos 2\pi ux}_{\text{even}} dx - i \int_{-\infty}^{\infty} f(x) \underbrace{\sin 2\pi ux}_{\text{odd}} dx$$

$$\Im^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ux} du$$

The above two equations are the Fourier transform pair.

Fourier Transform



Approximating a square wave as the sum of sine waves.

2-D Fourier Transform

- Fourier transform for *f(x,y)* with two variables

$$\Im\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp(-i2\pi(ux+vy))dxdy$$

and the inverse Fourier transform

$$\Im^{-1}\{F(u,v)\} = f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\exp(i2\pi(ux+vy))dudv$$

Discrete Fourier Transform

- The discrete Fourier transform pair that applies to sampled functions is given by:

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)\exp(-i2\pi ux/M) \quad u=0,1,2,\ldots,M\text{-}1$$

and

$$f(x) = \sum_{u=0}^{M-1} F(u)\exp(i2\pi ux/M) \quad x=0,1,2,\ldots,M\text{-}1$$

2-D Discrete Fourier Transform

- In 2-D case, the DFT pair is:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp(-i2\pi(ux/M + vy/N))$$

u=0,1,2,…,M-1 and v=0,1,2,…,N-1

and:

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp(i2\pi(ux/M + vy/N))$$

x=0,1,2,…,M-1 and y=0,1,2,…,N-1

Polar Coordinate Representation of FT

- The Fourier transform of a real function is generally complex and we use polar coordinates:

$$F(u) = R(u) + iI(u)$$

$$\Downarrow \quad \text{Polar coordinate}$$

$$F(u) = |F(u)|e^{i\phi(u)}$$

Magnitude:

(Spectrum)

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2}$$

Phase:

$$\phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$$

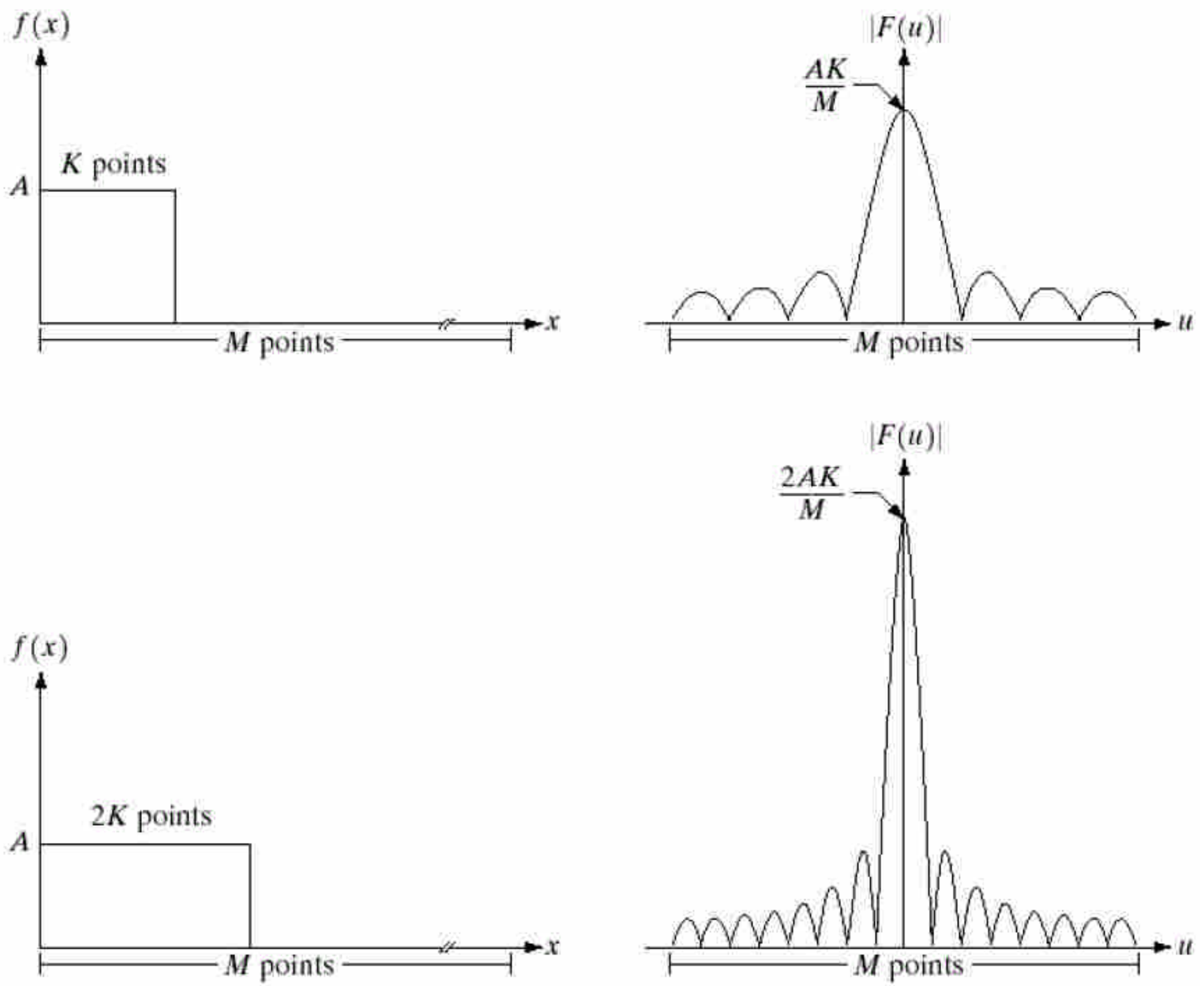
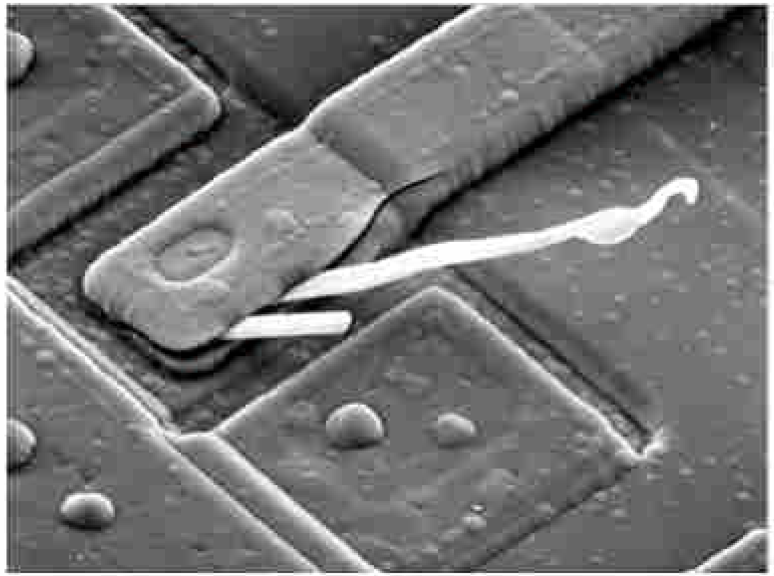
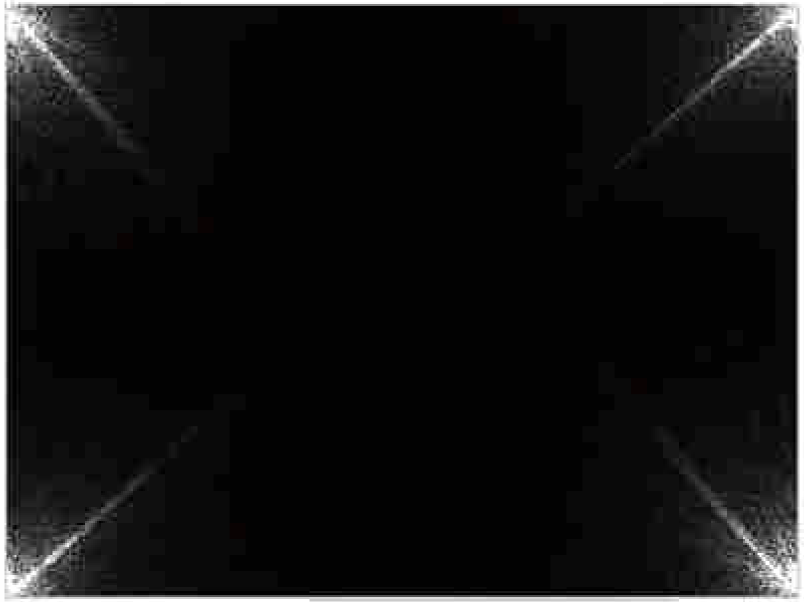


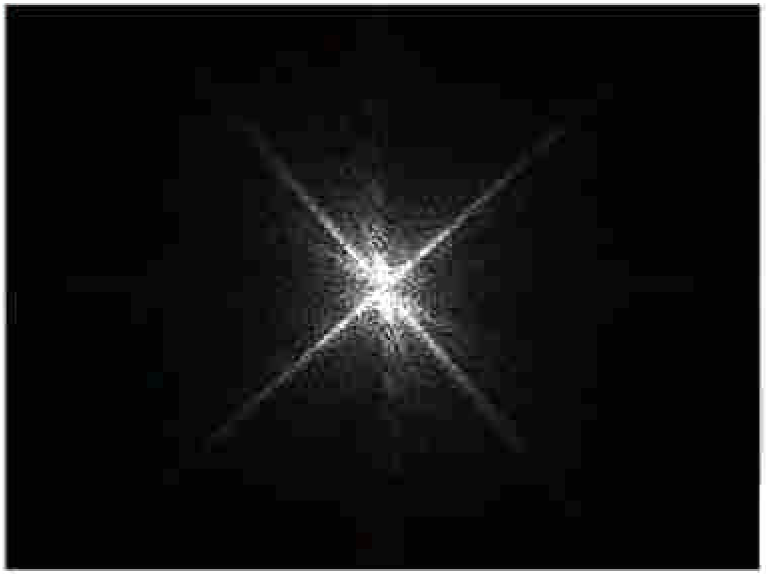
FIGURE 4.2 (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.



(a)



(b)



(c)

(a) A simple image, (b) Fourier spectrum without shifting, and (c) Fourier spectrum with shifting.

Fast Fourier Transform

- Number of complex multiplications and additions to implement Fourier Transform: M^2 (M complex multiplications and N-1 additions for each of the N values of u).
- The decomposition of FT makes the number of multiplications and additions proportional to M $\log_2 M$:
  - Fast Fourier Transform or FFT algorithm.

Steps of Filtering in the frequency domain

- Filtering in the frequency domain consists of the following steps:
  1. Multiply the input image by $(-1)^{x+y}$ to center the transform
  2. Compute the DFT : $F(u,v)$
  3. Multiply $F(u,v)$ by a filter function $H(u,v)$
  4. Compute the inverse DFT of the result in (3)
  5. Obtain the real part of the result in (4)
  6. Multiply the result in (5) by $(-1)^{x+y}$

Steps of Filtering in the frequency domain

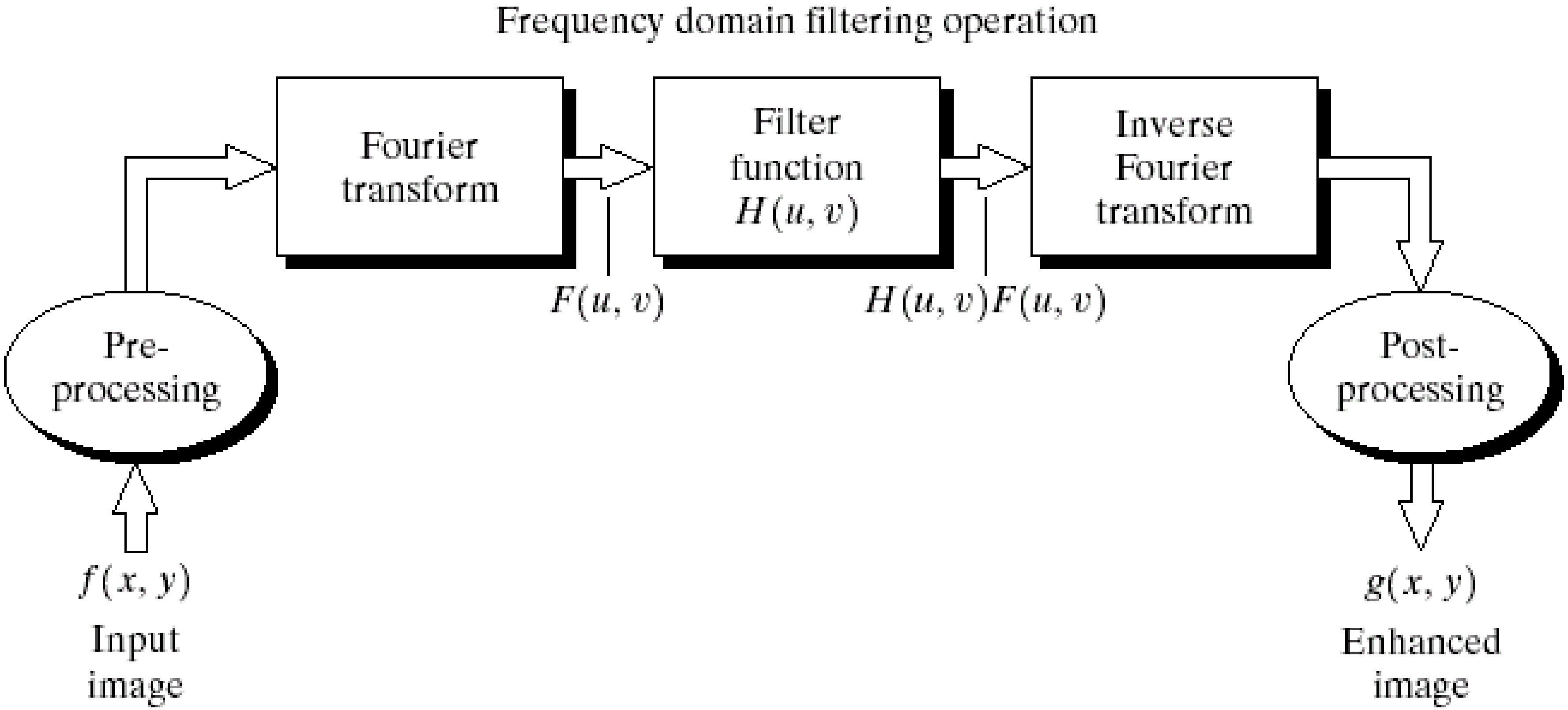


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Spatial & Frequency Domain

- The most fundamental relationship between the spatial and frequency domains is established by convolution theorem.
- Discrete 2D convolution

$$f(x,y) * h(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x-m, y-n)$$

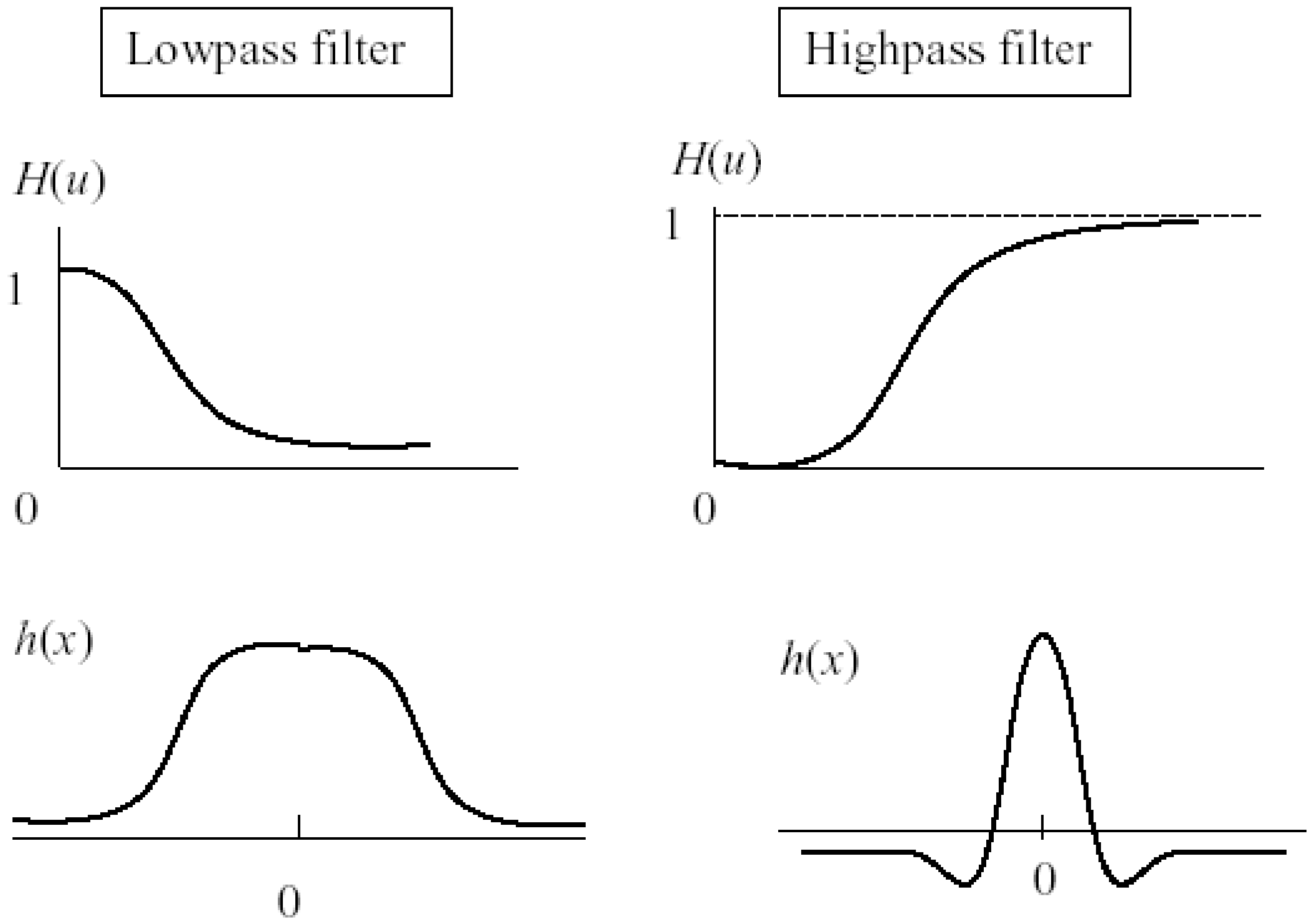
- Convolution theorem

$$f(x,y) * h(x,y) \quad \Leftrightarrow \quad F(u,v) H(u,v)$$

$$f(x,y) h(x,y) \quad \Leftrightarrow \quad F(u,v) * H(u,v)$$

$\rightarrow$ Fourier transform
$\leftarrow$ inverse Fourier transform

- Examples of some common filters (1-D case):



Basic filters

- Types of enhancement that can be done:
  - Lowpass filtering: reduce the high-frequency content -- blurring or smoothing
  - Highpass filtering: increase the magnitude of high-frequency components relative to low-frequency components -- sharpening.

Example

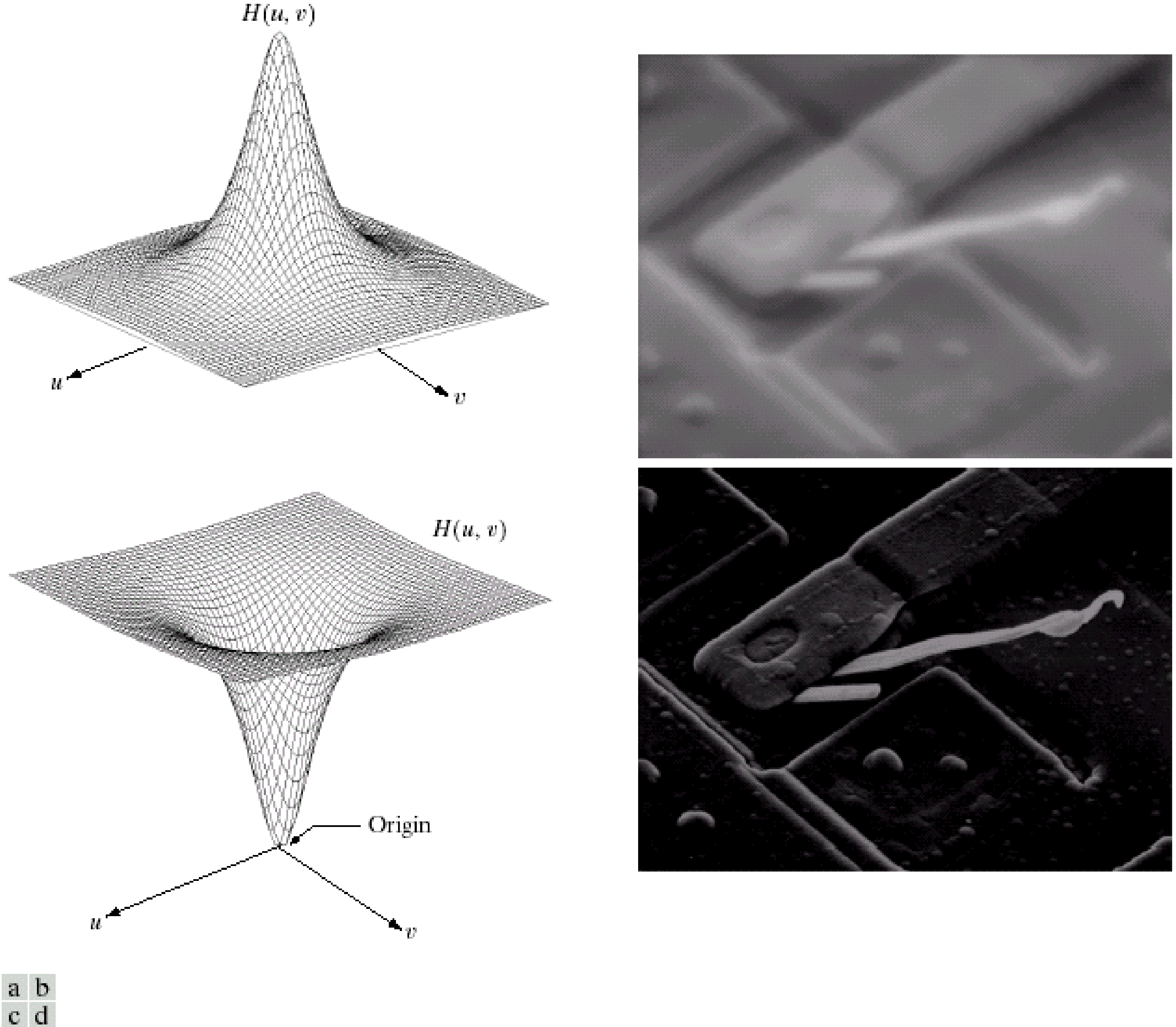


FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

Smoothing frequency-domain filters

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

- Ideal lowpass filters
- Butterworth lowpass filters
- Gaussian lowpass filters

Ideal lowpass filters

- A 2-D ideal low-pass filter:

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

where D_0 is a specified nonnegative quantity and D(u,v) is the distance from point (u,v) to the center of the frequency rectangle.

- *D(u,v)* is the distance from *(u,v)* to the origin of the frequency rectangle defined by :

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

where (M/2, N/2) is the center of the frequency rectangle with an $M \times N$ image.

Ideal lowpass filters

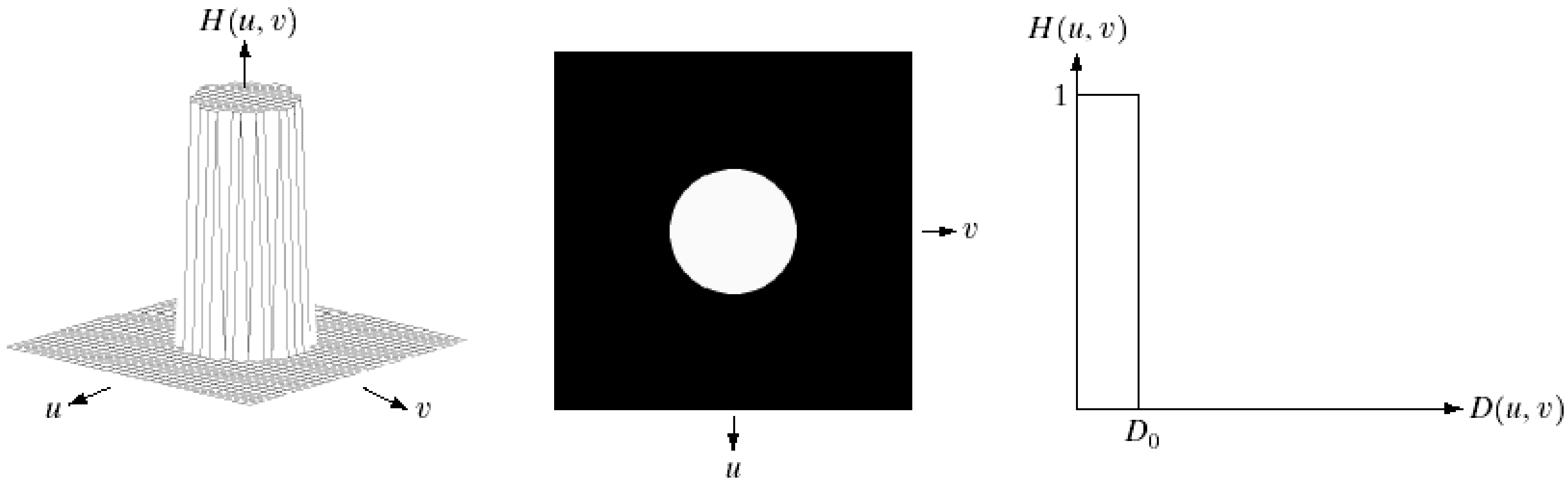
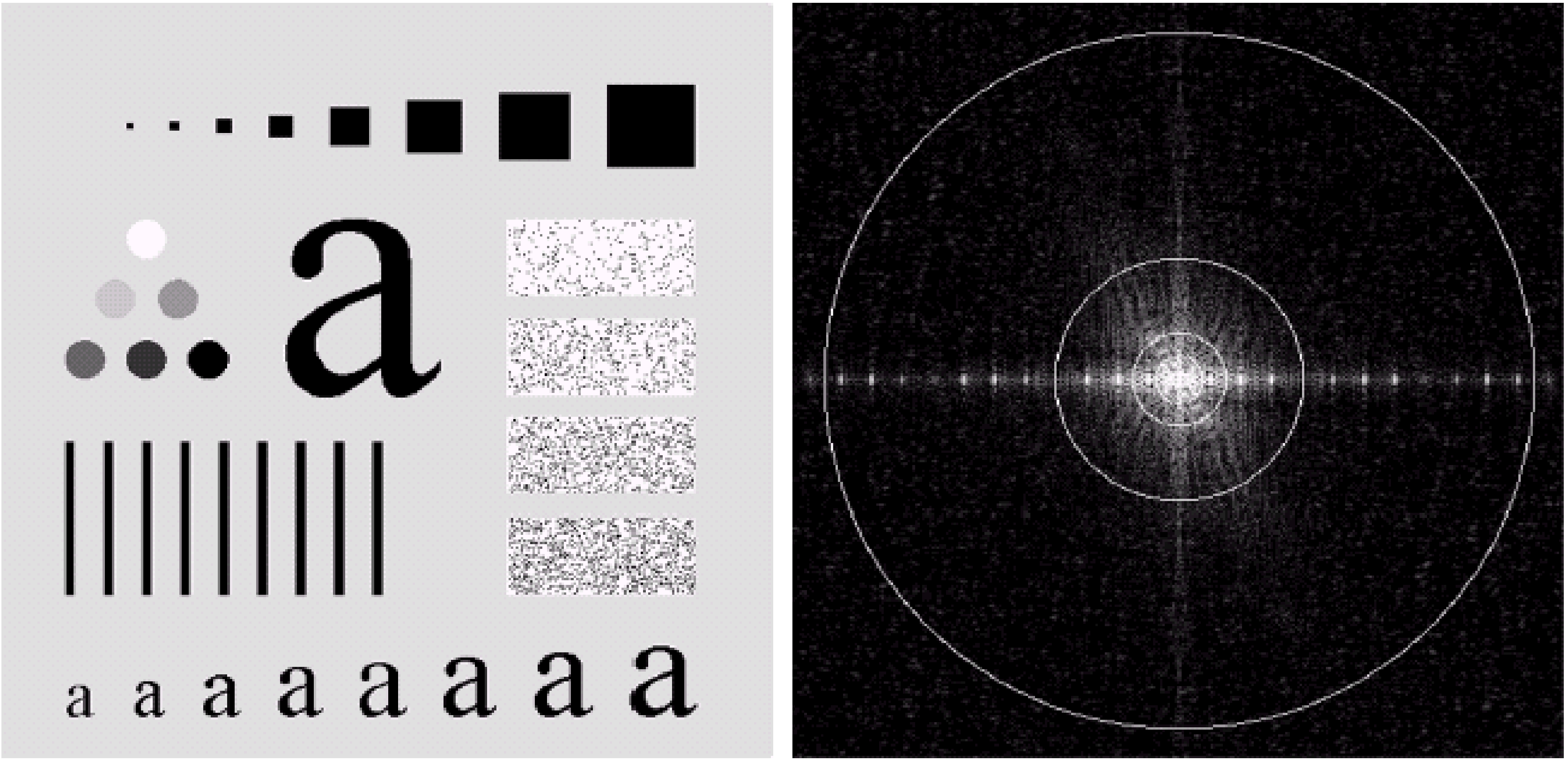
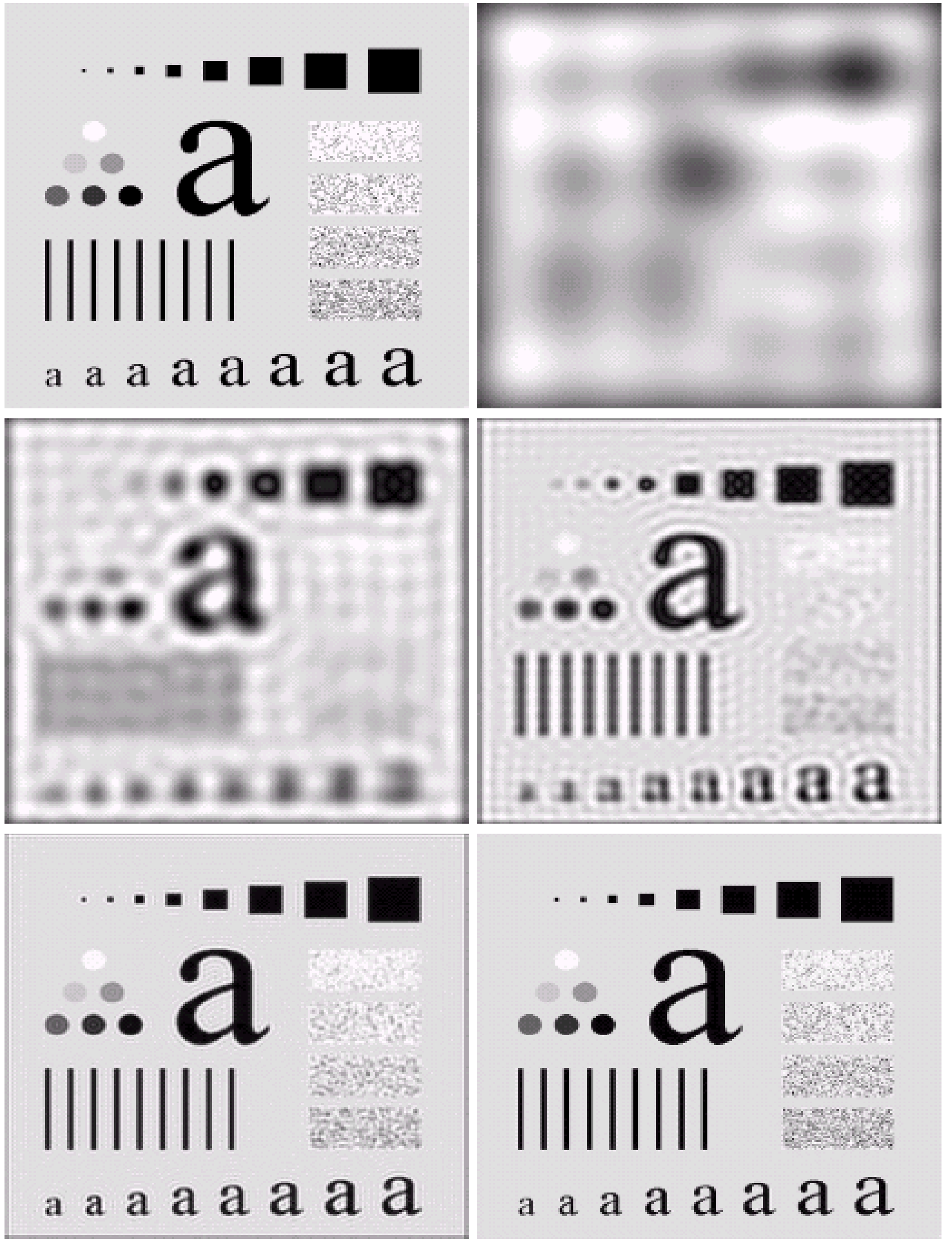


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.



a b

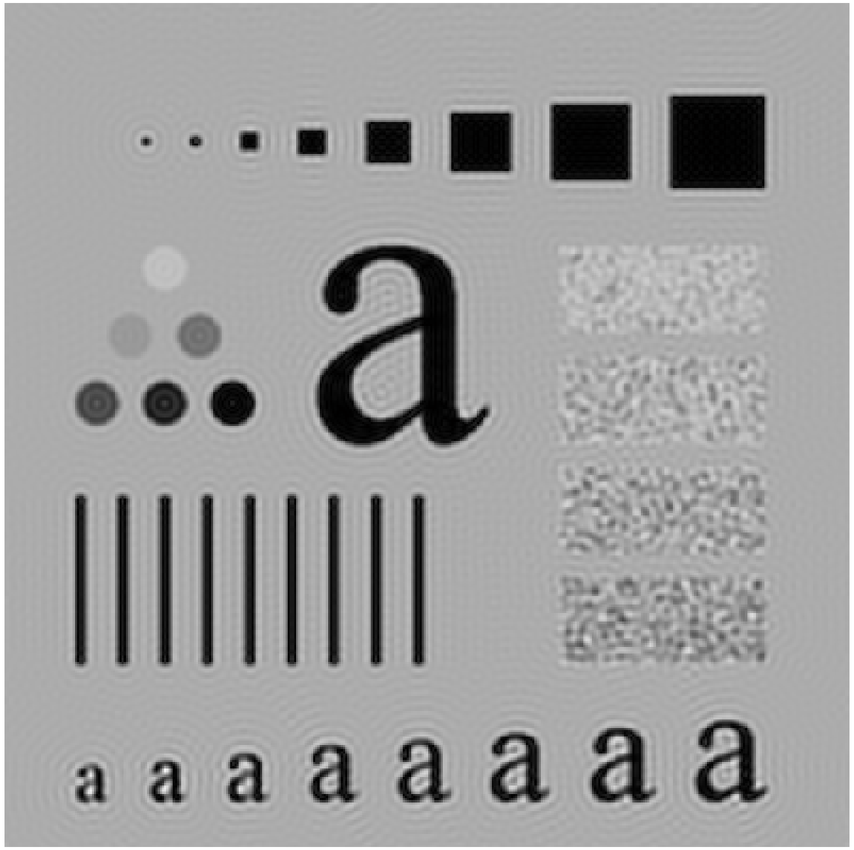
FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.



a	b
c	d
e	f

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

Ringing effect



Result of Ideal lowpass filtering with cutoff frequency set at radius value of 80.

Notice the severe ringing effect in the blurred images, which is a characteristic of ideal filters. It is due to the discontinuity in the filter transfer function.

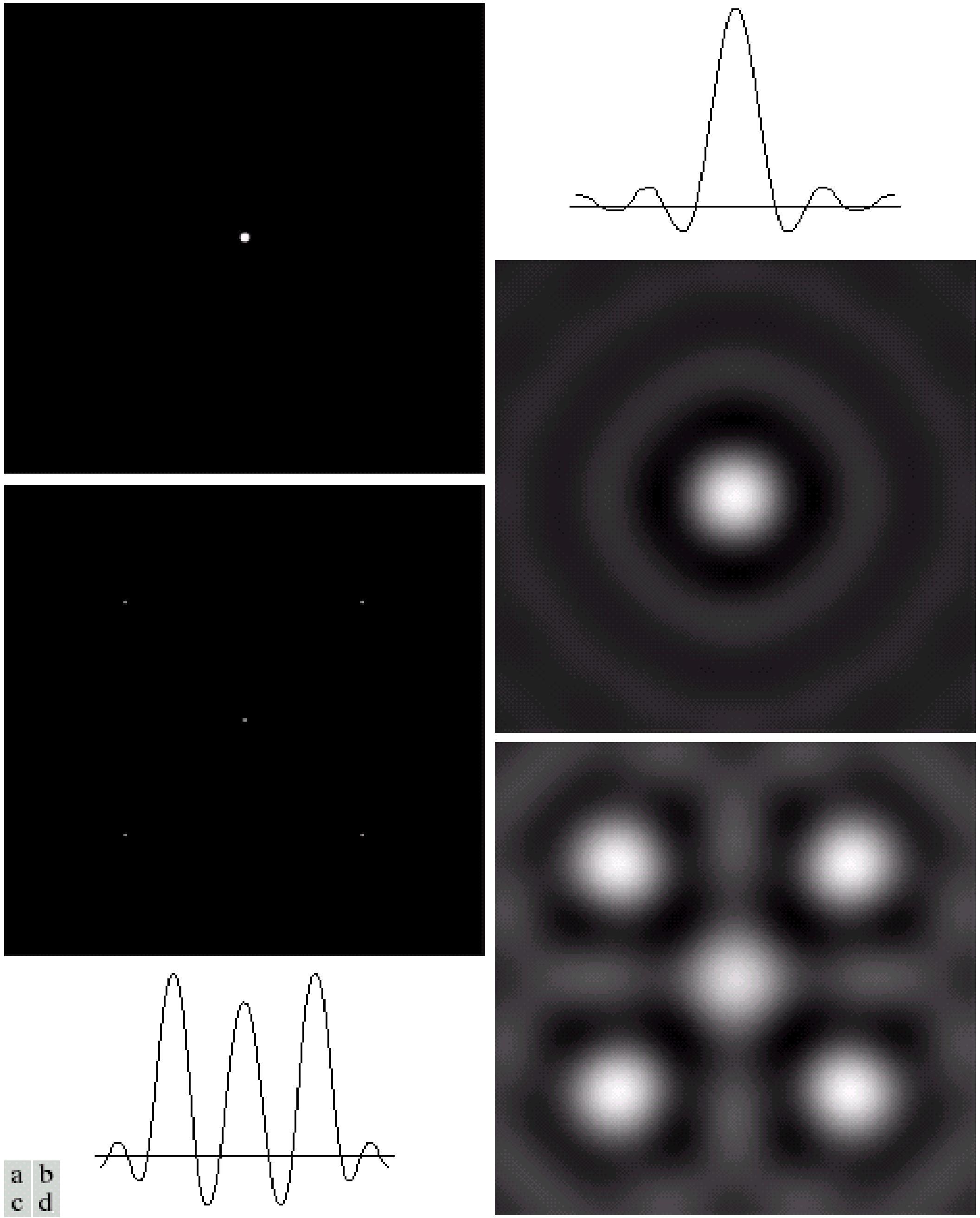


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

Butterworth Lowpass Filter

- This filter does not have a sharp discontinuity establishing a clear cutoff between passed and filtered frequencies.

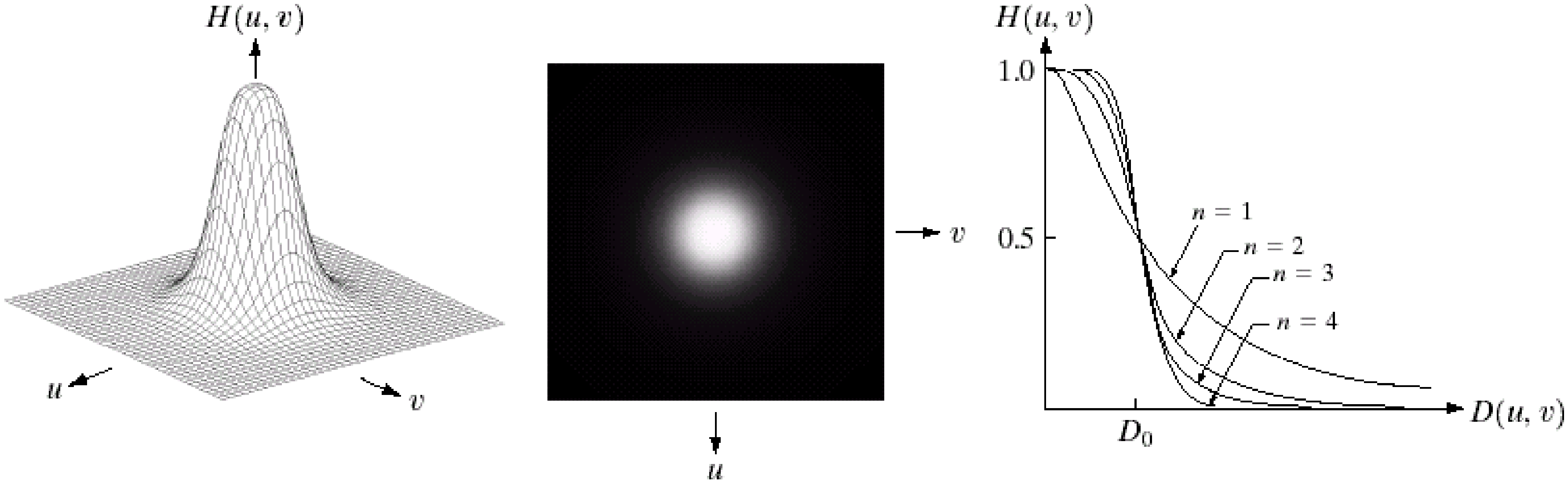
$$H(u,v)=\frac{1}{1+[D(u,v)/D_0]^{2n}}$$

where D_0 is the cutoff frequency , and D(u,v) is

$$D(u,v)=[(u-M/2)^2+(v-N/2)^2]^{1/2}$$

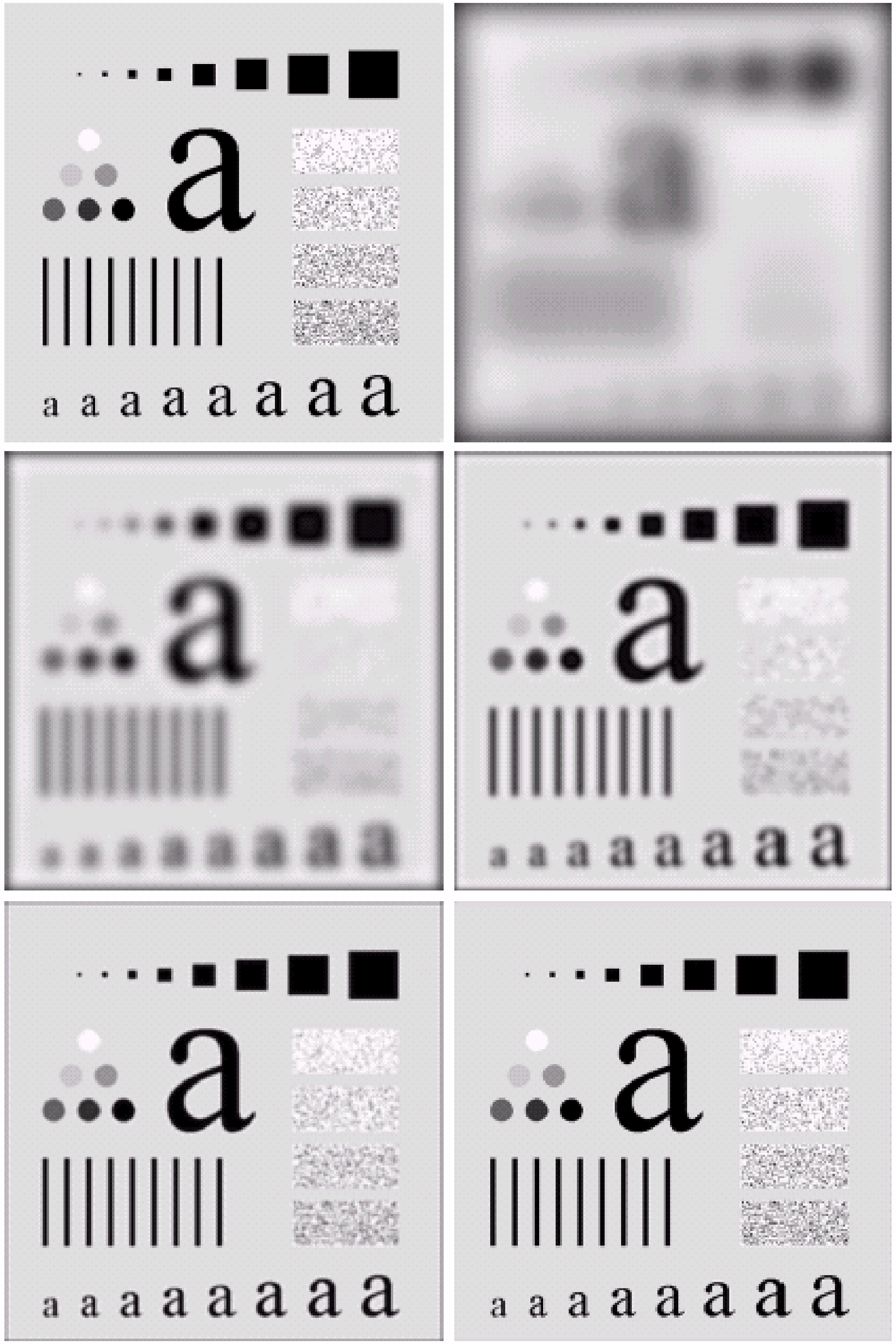
- This is more appropriate for image smoothing than the ideal LPF, since this does not introduce ringing.

Butterworth Lowpass Filter



a b c

FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.



a	b
c	d
e	f

FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

Gaussian Lowpass Filter

- Gaussian lowpass filter is defined by

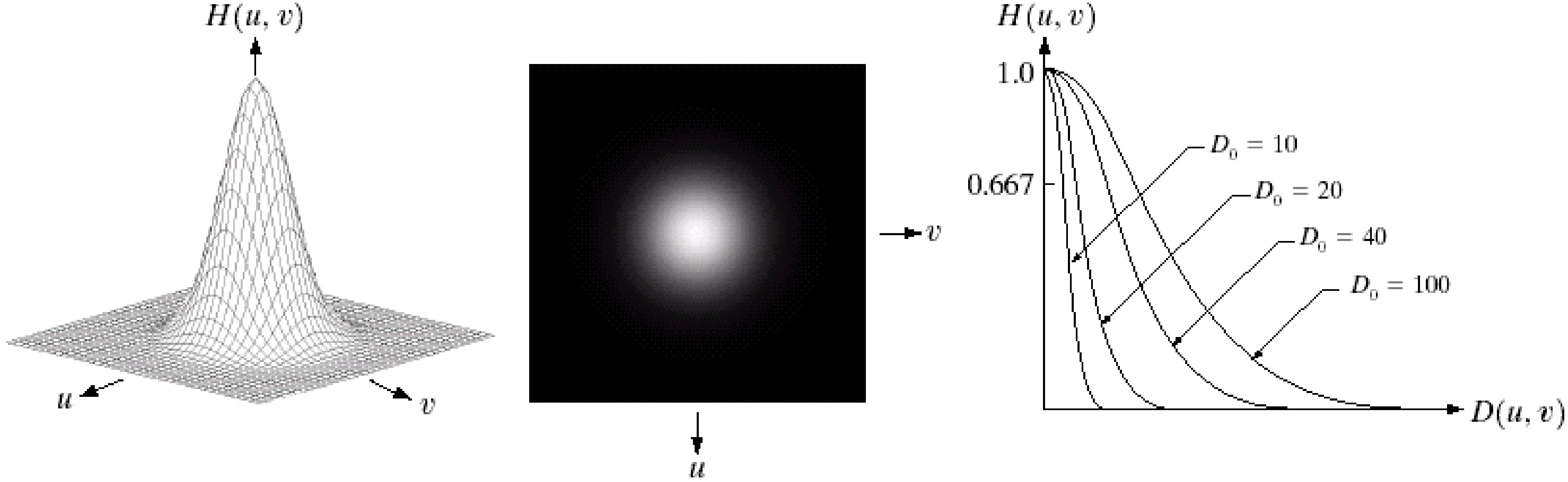
$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$$

D(u,v) is the distance from the origin of the Fourier transform.
by letting $\sigma = D_0$, we have

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

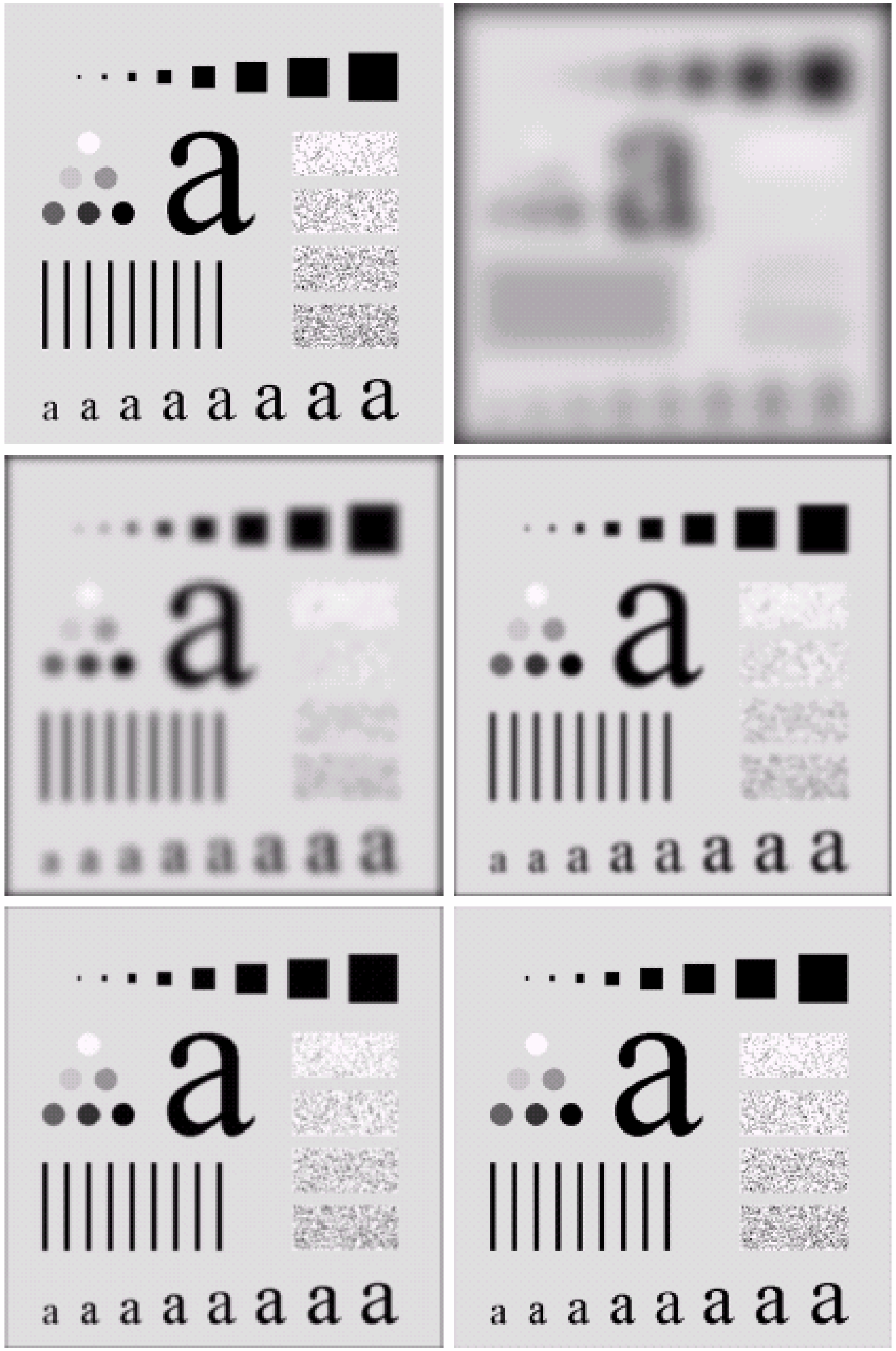
where D_0 is the cutoff frequency.

Gaussian Lowpass Filter



a b c

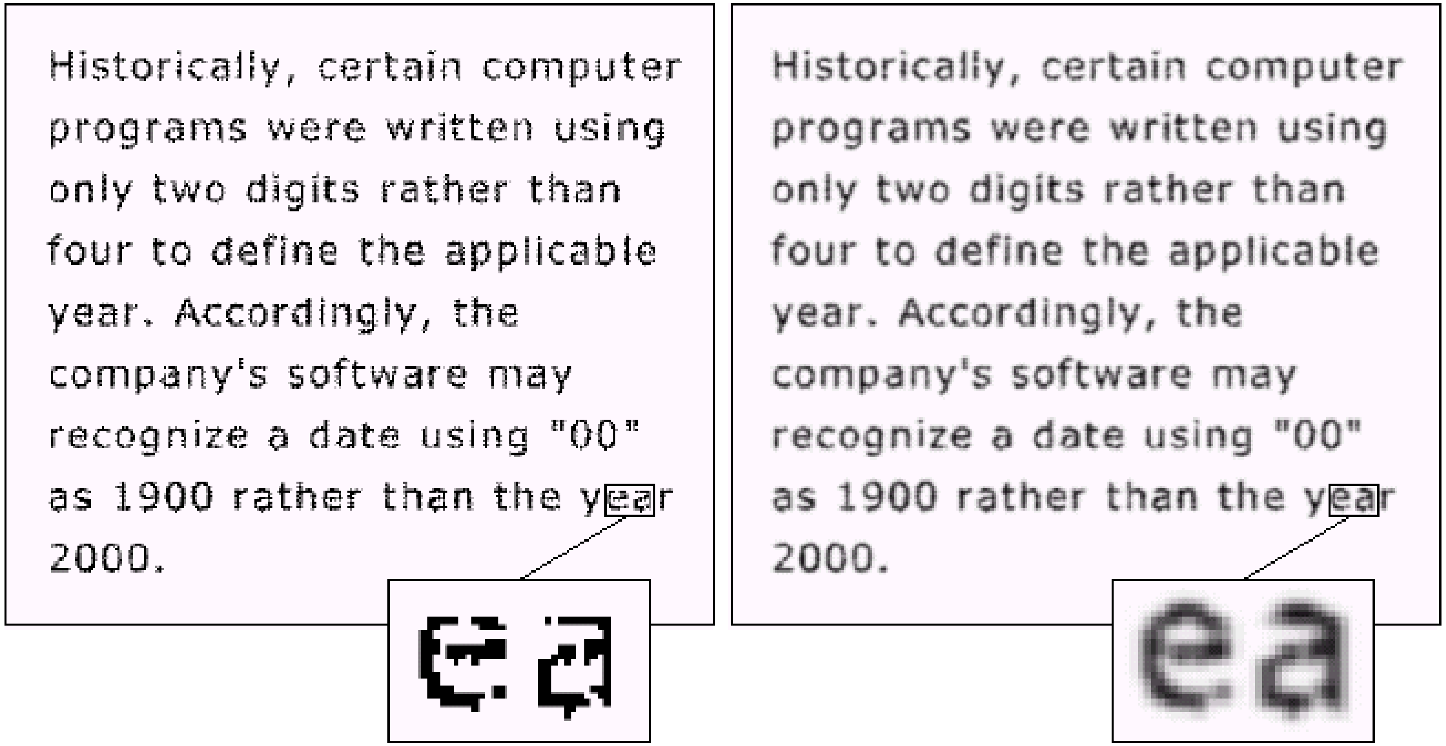
FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.



a	b
c	d
e	f

FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

Image Enhancement in the Frequency Domain



a b

FIGURE 4.19
(a) Sample text of poor resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).

Image Enhancement in the Frequency Domain

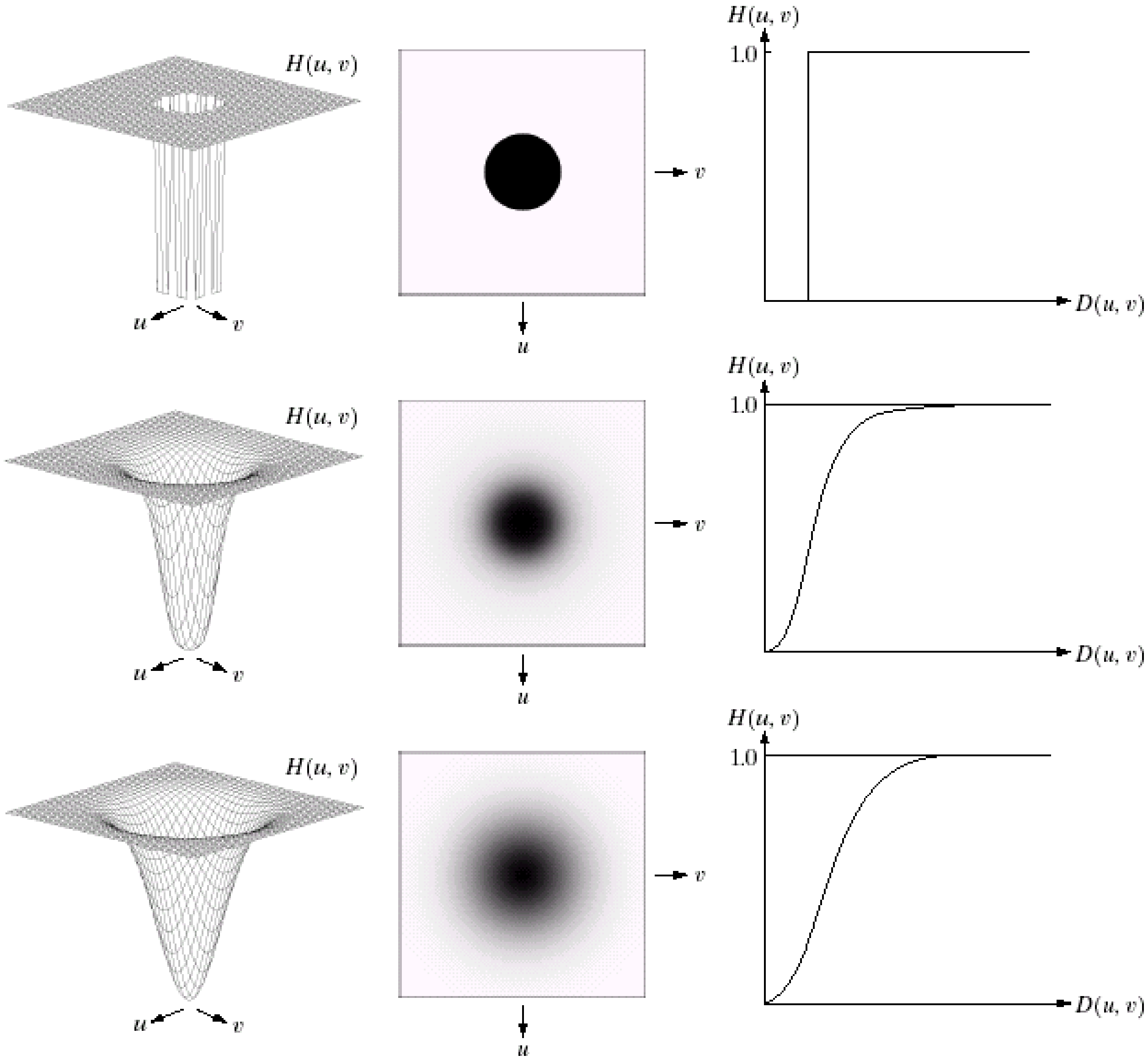
FIGURE 4.20 (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

Sharpening frequency-domain filters

- Ideal highpass filters
- Butterworth highpss filters
- Gaussian highpass filters
- The laplacian in the frequency domain

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

H_{hp}=*highpass filter*
H_{lp}=*lowpass filter*



a b c
d e f
g h i

FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

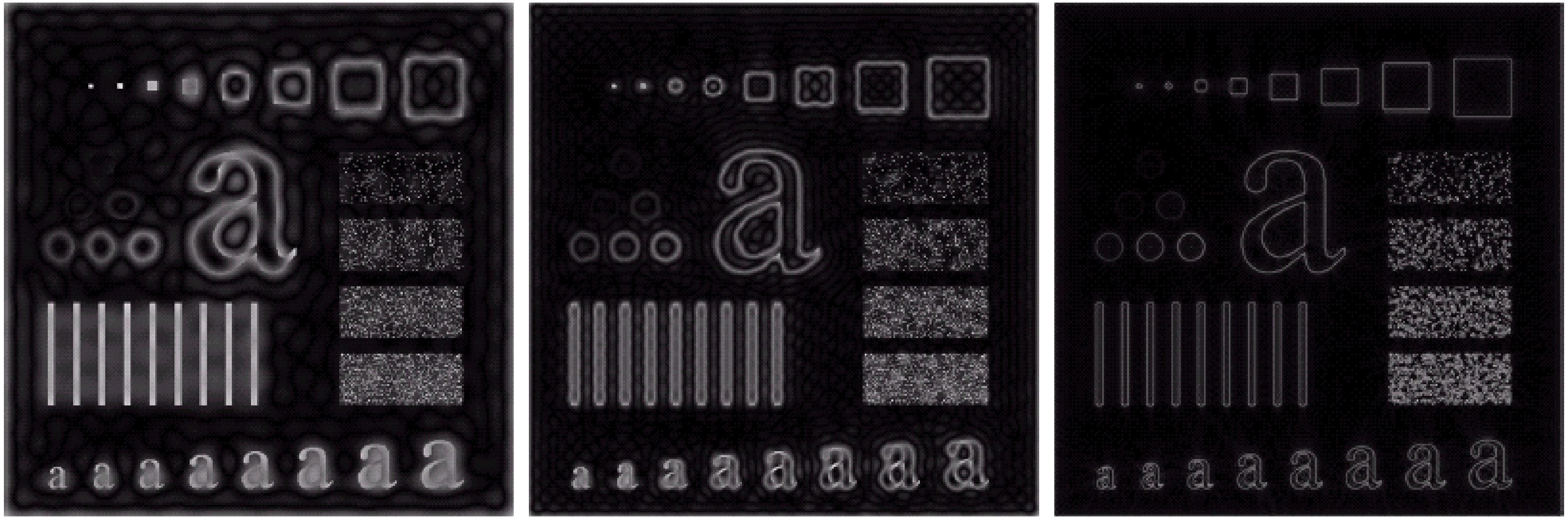
Ideal highpass filters

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

where $D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$

This filter is the opposite of the ideal lowpass filter.

Ideal highpass filters



a b c

FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

Butterworth highpss filters

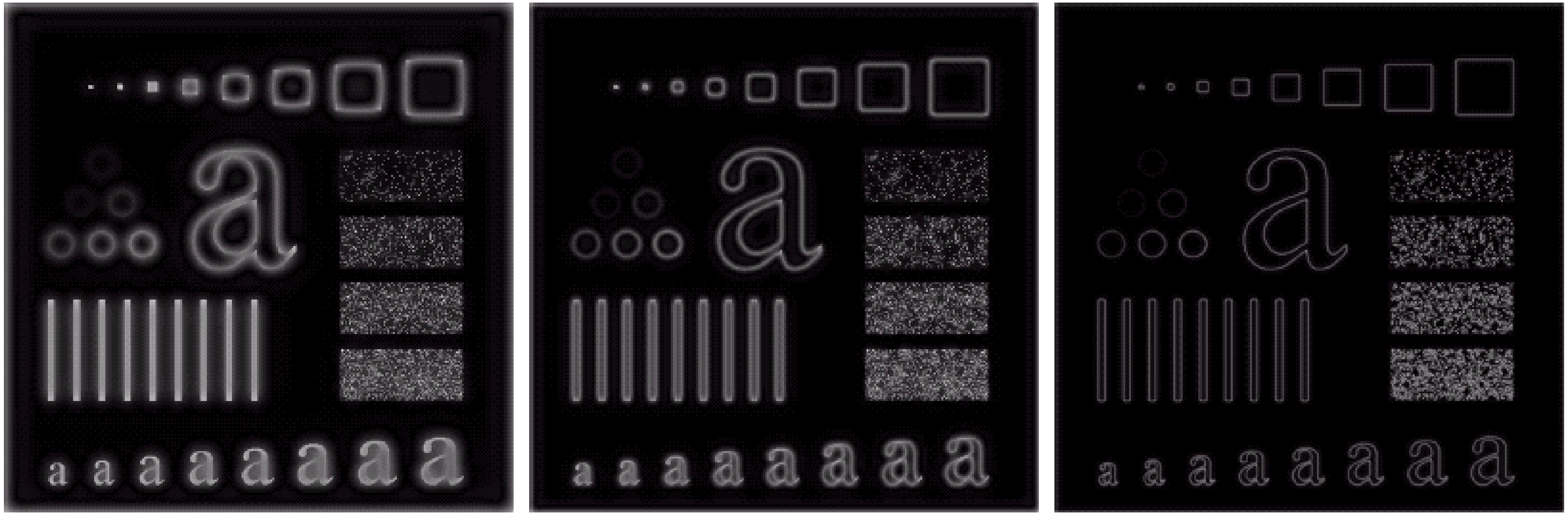
- The transfer function of a Butterworth highpass filter (BHPF) of order n is defined by

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

where D_0 is the cutoff frequency , and D(u,v) is

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

Butterworth highpss filters



a b c

FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

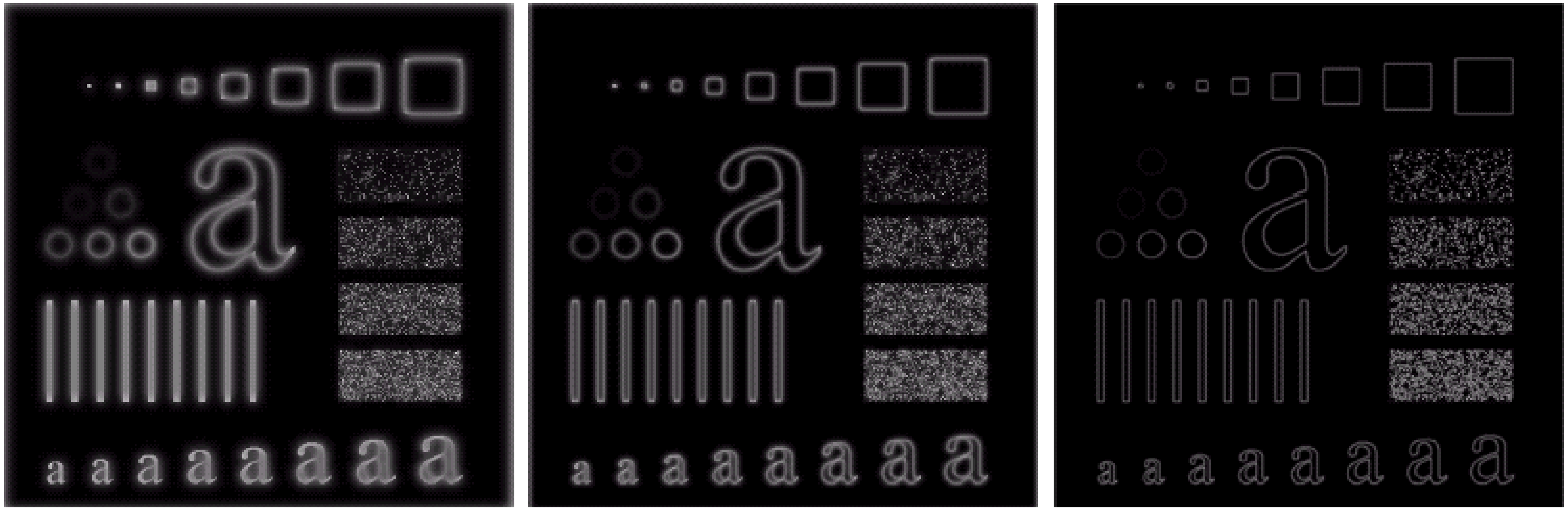
Gaussian Highpass Filter

- Gaussian highpass filter is defined by

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

D(u,v) is the distance from the origin of the Fourier transform and D_0 is the cutoff frequency.

Gaussian Highpass Filter



a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

Laplacian in the Frequency Domain

- It can be shown that:

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

- From the above expression, we get

$$\Im\left[\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}\right] = -(u^2 + v^2)F(u,v)$$

$$\Im\left[\nabla^2 f(x,y)\right] = -(u^2 + v^2)F(u,v)$$

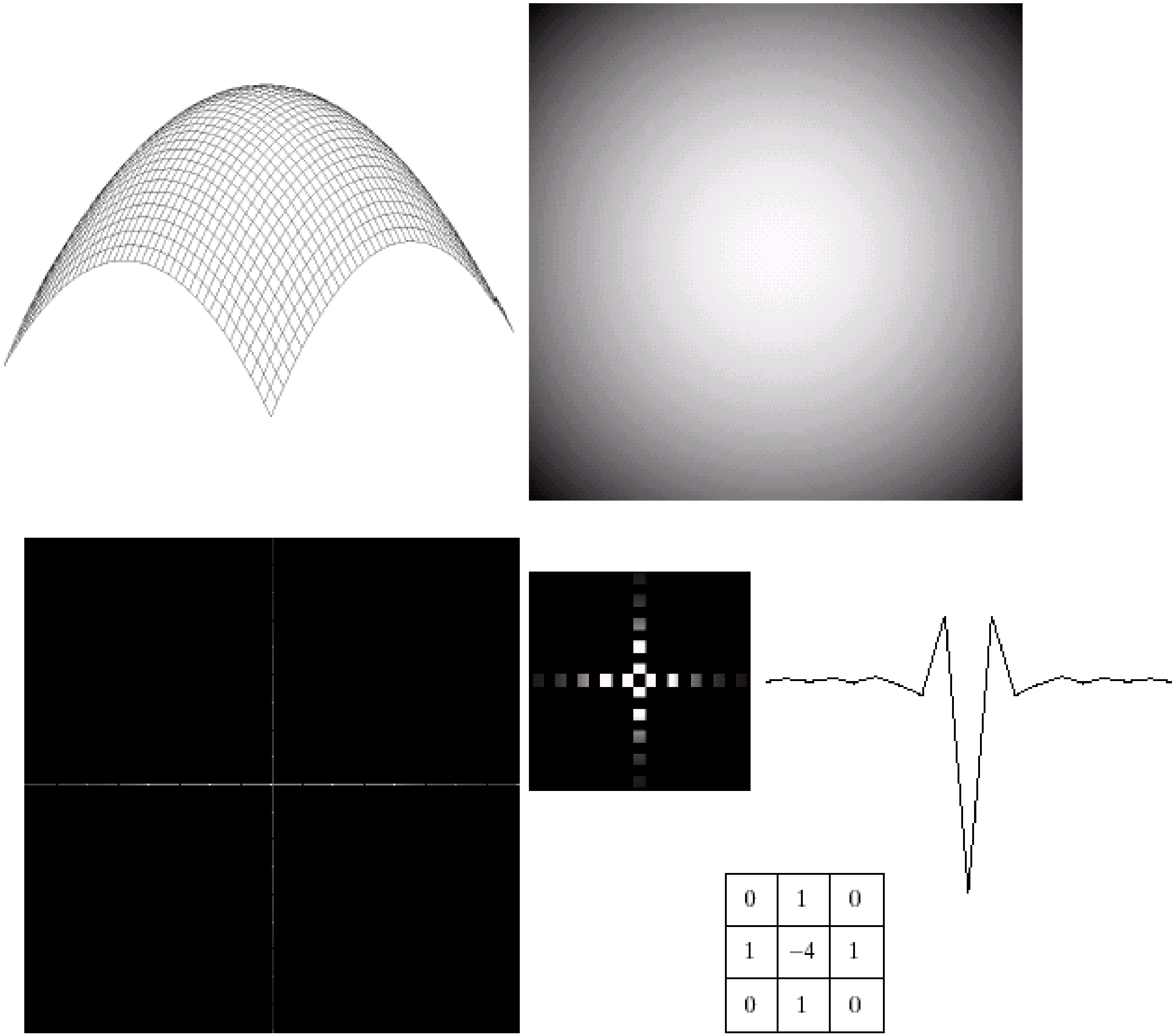
The Laplacian can be implemented in the frequency domain by using the filter

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

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

Laplacian in the Frequency Domain

- An enhanced image can be obtained by subtracting (not addition, because the transfer function already has a negative sign) Laplacian from the original image.



a b
c d e
f

FIGURE 4.27 (a) 3-D plot of Laplacian in the frequency domain. (b) Image representation of (a). (c) Laplacian in the spatial domain obtained from the inverse DFT of (b). (d) Zoomed section of the origin of (c). (e) Gray-level profile through the center of (d). (f) Laplacian mask used in Section 3.7.

a b
c d

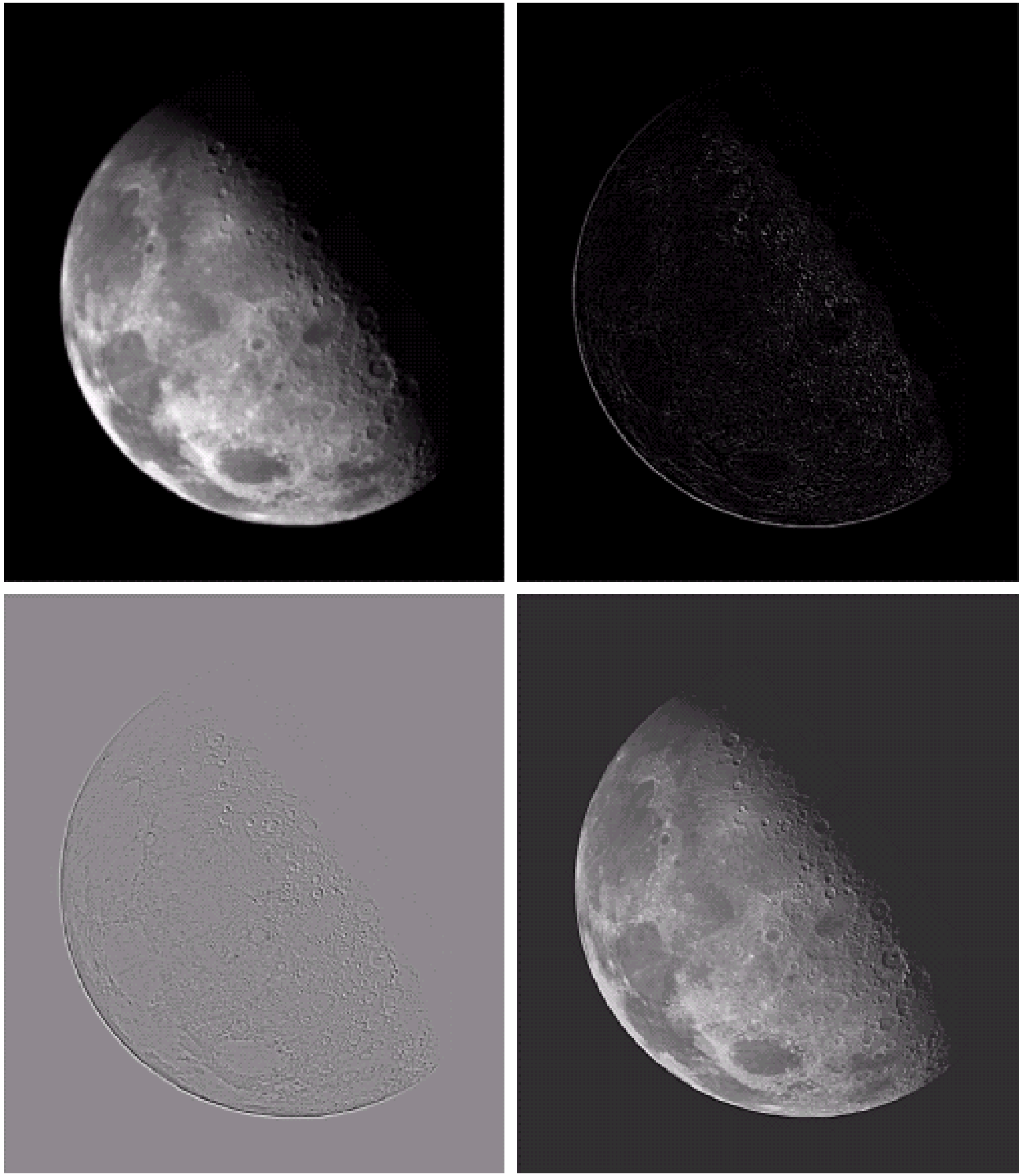


FIGURE 4.28 (a) Image of the North Pole of the moon. (b) Laplacian filtered image. (c) Laplacian image scaled. (d) Image enhanced by using Eq. (4.4-12). (Original image courtesy of NASA.)

Unsharp Masking & High-Boost Filtering

- The relationship between Unsharp Masking and High-Boost Filtering

$$f_{hb}(x,y) = (A-1)f(x,y) + f(x,y) - f_{lp}(x,y)$$

$$\Rightarrow f_{hb}(x,y) = (A-1)f(x,y) + f_{hp}(x,y) \qquad (4.4\text{-}17)$$

Frequency domain

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

$$\Rightarrow H_{hb}(u,v) = (A-1) + H_{hp}(u,v)$$

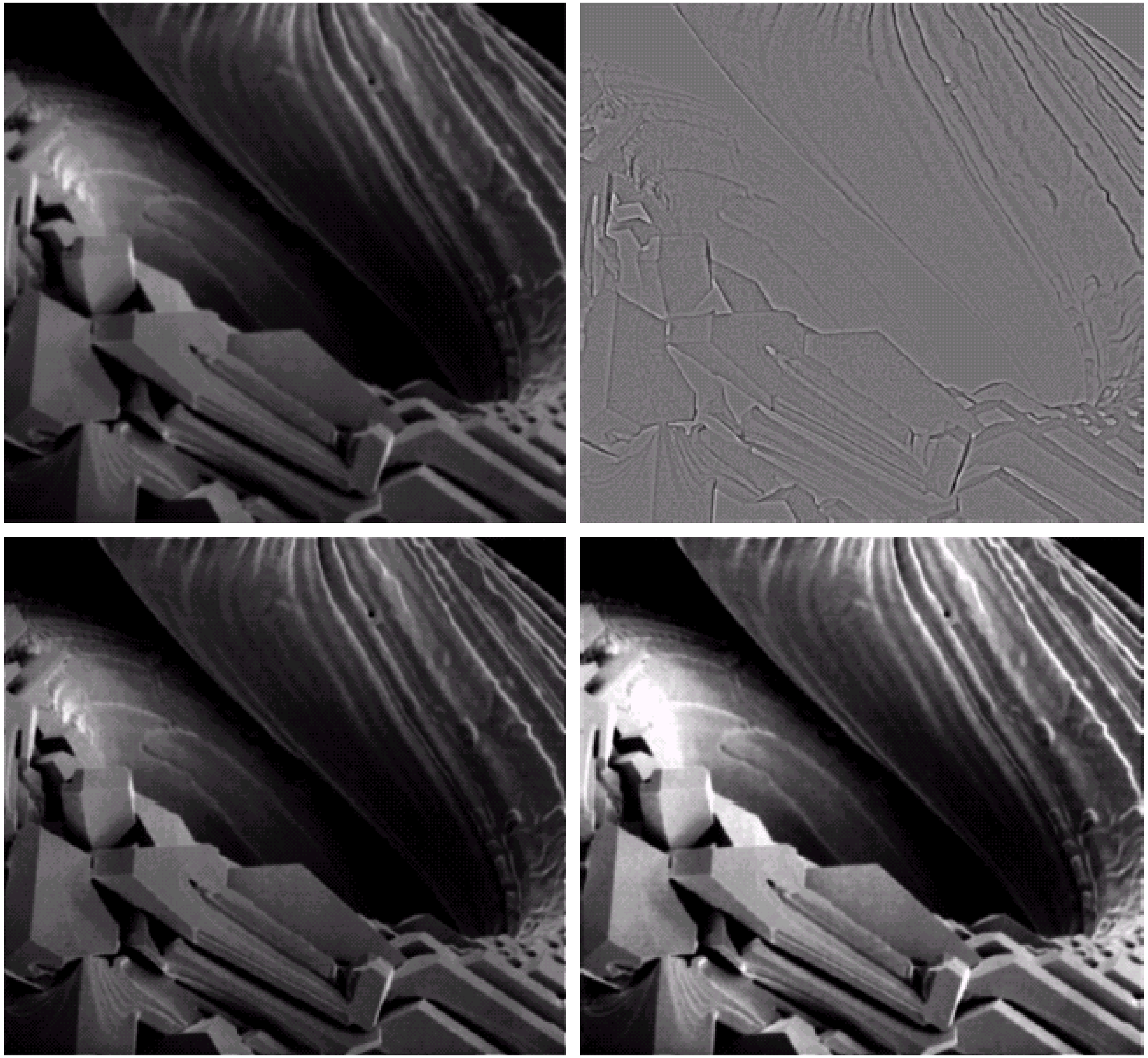


FIGURE 4.29 Same as Fig. 3.43, but using frequency domain filtering. (a) Input image. (b) Laplacian of (a). (c) Image obtained using Eq. (4.4-17) with $A = 2$. (d) Same as (c), but with $A = 2.7$. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Frequency Emphasis Filtering

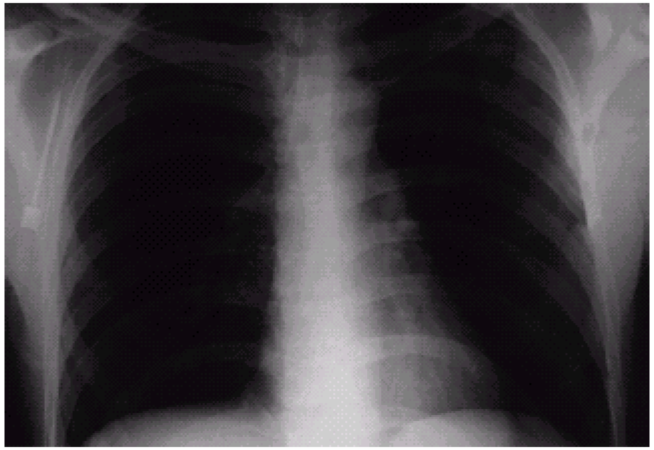
- Similar to high-boost filtering
- Sometimes, it is advantageous to accentuate the contribution to enhancement by the high-frequency components of the image
- To do this in the frequency domain, we simply multiply any high-pass filter by a constant and add an offset so that the zero frequency term is not eliminated by the filter.
- High-Frequency Emphasis

$$H_{hfe}(u,v) = a + bH_{hp}(u,v)$$

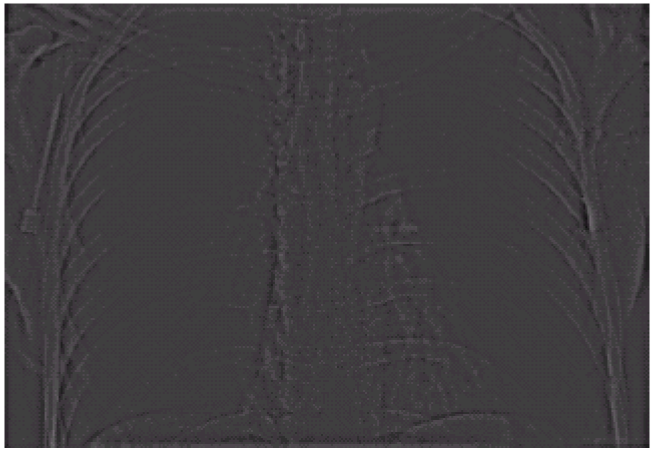
where a>0 and b>a . Typical values: a = (0.25 to 0.5) and b = (1.5 to 2)

Example

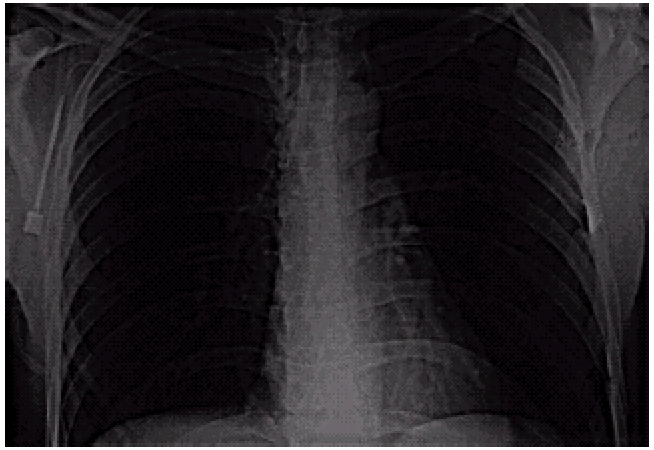
Typically, x-ray images are slightly blurred, because x-rays cannot be focused in the same manner that lenses are focused. In this case, the image is generally dark.



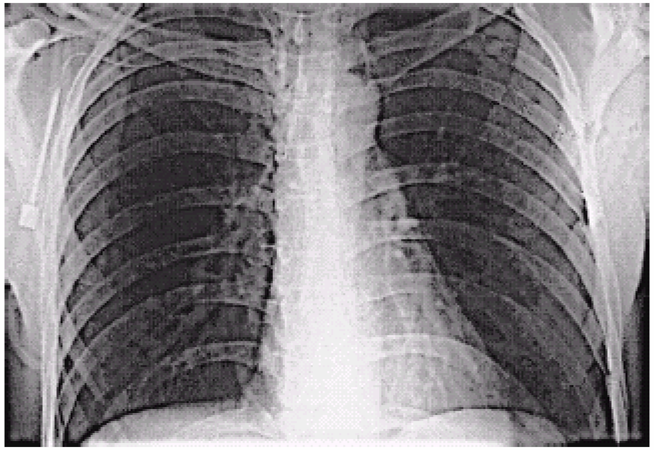
Let us use a Butterworth high-pass filter of order 2 and a value of Do equal to 5% of the image vertical dimension. It shows the principal edges in the image.



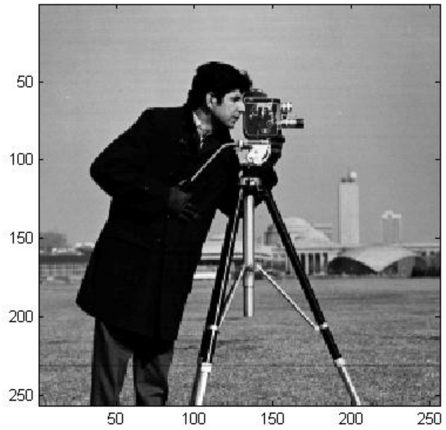
Let us take the frequency response of Butterworth filter of order 2 and modify it slightly to make a high-frequency-emphasis filter (a=0.5,b=2.0). Result is better, but still the image is dark. Probably because, the gray-level tonalities due to low-frequency components was not lost.



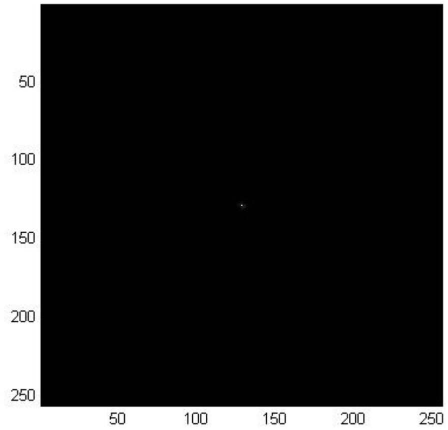
An image characterized by gray levels in a narrow range of the gray scale is an ideal candidate for histogram equalization. The final enhanced image is a little noisy, but this is typical of X-ray images when their gray scale is expanded.



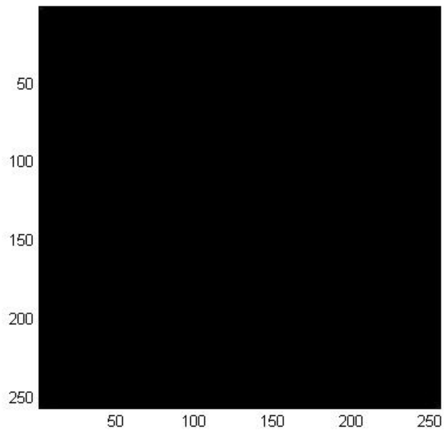
Matlab



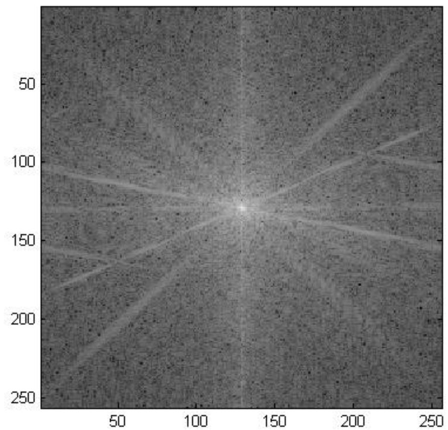
A



C



B



log(abs(C)+1)

```
A=imread('cameraman.tif');
A=double(A);
B=fft2(A);
figure
imagesc(A)
colormap(gray)
figure
imagesc(abs(B))
colormap(gray)
C=fftshift(B);
imagesc(abs(C))
imagesc(log(abs(C)+1))
```