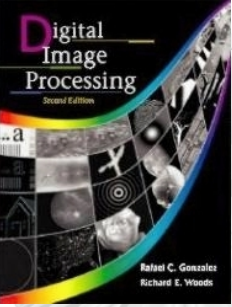
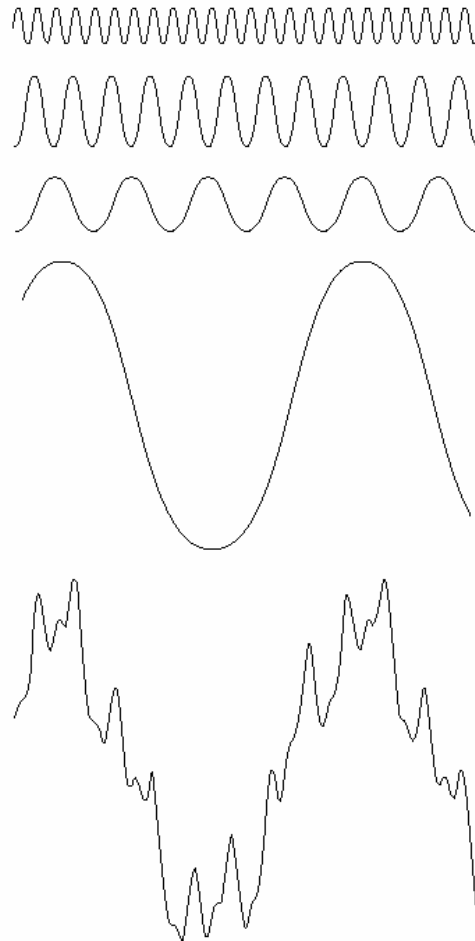


## Chapter 4 Image Enhancement in the Frequency Domain

- Fourier Series
- Fourier Transform
- Discrete Fourier Transform
- Fourier Transform for Image Enhancement
- Implementation



## Chapter 4.1 Background



**FIGURE 4.1** The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.



## 4.2 Fourier Transform in the Frequency Domain

- Fourier transform  $F(u)$  of  $f(x)$  is defined as

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux} dx$$

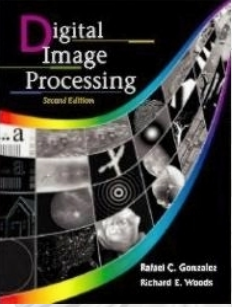
- The inverse Fourier Transform is

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux} du$$

- DFT for Discrete function  $f(x)$ ,  $x=0, 1, \dots, M-1$

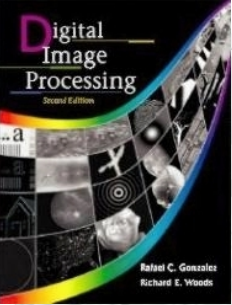
$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M} \text{ for } u=0, 1, \dots, M-1$$

- Inverse DFT  $f(x) = \sum_{u=0}^{M-1} F(u)e^{-2\pi ux/M}$

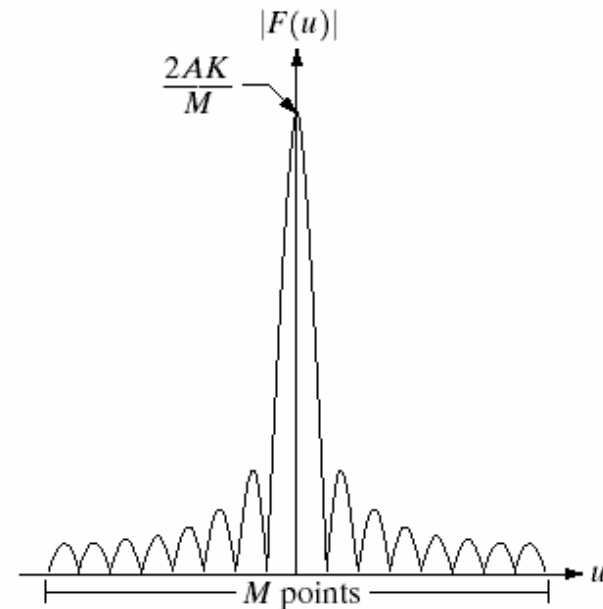
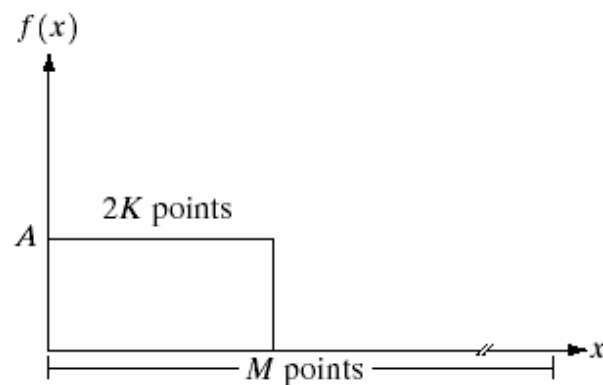
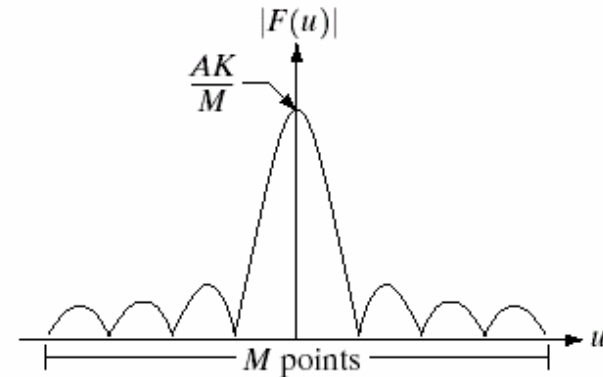
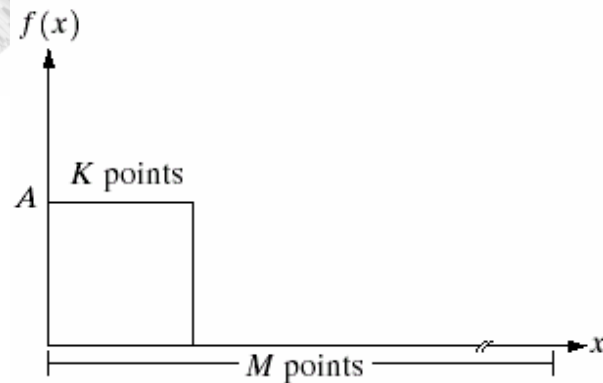


## 4.2 Fourier Transform in the Frequency Domain

- **Example** (1-D DFT, Fig.4.2)
- $f(x)$  sampled  $K$  points **or**  $f(x)$  sampled  $2K$  points
- More **samples** in **time domain (higher resolution)**  $\rightarrow$  **Lower resolution** in **frequency domain.**
- *Scaling property of DFT*
$$f(ax, by) \Leftrightarrow F(u/a, v/b)/|ab|$$
- *Sampling* :  $f(x) = f(x_0 + \Delta x)$ ,  $\Delta x$  is the time resolution
- $f(x) \rightarrow F(u) = F(u + \Delta u)$ ,  $\Delta u$  is the freq. resolution.
- $\Delta u = 1/(M\Delta x)$



## 4.2 Fourier Transform in the Frequency Domain



a	b
c	d

**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.



## 4.2 Fourier Transform in the Frequency Domain

- For discrete case (**DFT: Discrete Fourier Transform**)

- $f(x), x=0, \dots, M-1$

- $f(x_0), f(x_0 + \Delta x), \dots, f(x_0 + (M-1)\Delta x)$

$$f(x) = f(x_0 + x\Delta x)$$

- $F(u), u=0, \dots, M-1$

- $F(u_0), F(u_0 + \Delta u), \dots, F(u_0 + (M-1)\Delta u)$

$$F(u) = F(u_0 + u\Delta u)$$

$$\Delta u = 1/M\Delta x$$



## 4.2.2 The Two-dimensional Discrete Fourier Transform(DFT)

- 2D-DFT of  $f(x, y)$  of size  $M \times N$

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

- Inverse 2-D DFT

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

- Magnitude and Phase of  $F(u, v) = R(u, v) + jI(u, v)$

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

$$\phi(u, v) = \tan^{-1}(I(u, v)/R(u, v))$$

- Power Spectrum

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$$



## 4.2.2 The Two-dimensional Discrete Fourier Transform(DFT)

- Modulation in space domain

$$f(x, y)(-1)^{xy}$$

- $F(u, v)$  will be shifted to  $(M/2, N/2)$

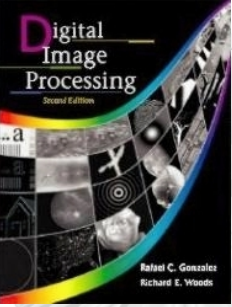
$$F(u-M/2, v-N/2)$$

- The center of  $(u, v)$ ,  $u=1, \dots, M$ ,  $v=1, \dots, N$

$$u=(M/2)+1, v=(N/2)+1$$

- Average of  $f(x, y)$   $F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$





## 4.2.2 The Two-dimensional Discrete Fourier Transform(DFT)

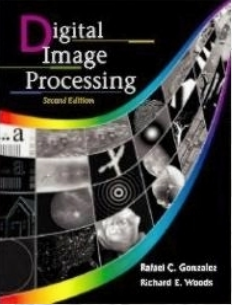
- For real  $f(x,y)$

$$F(u, v) = F^*(-u, -v)$$

$$|F(u, v)| = |F(-u, -v)|$$

- Samples in the space domain and frequency domain  $\Delta u = 1/M\Delta x$

$$\Delta v = 1/N\Delta y$$



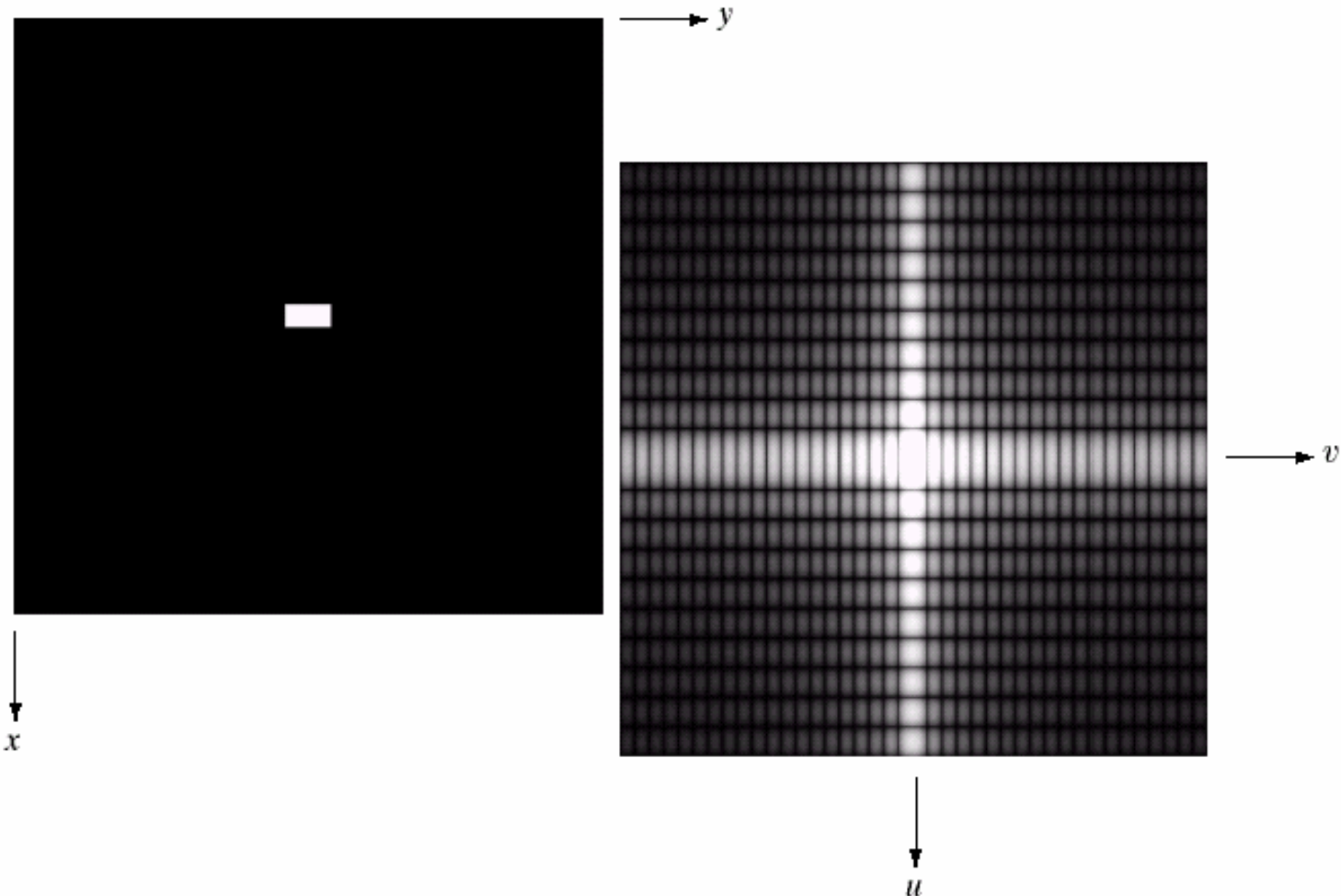
## 4.2.2 The Two-dimensional Discrete Fourier Transform(DFT)

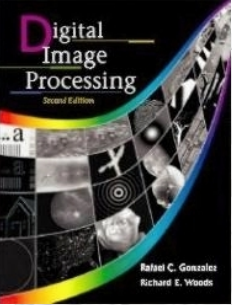
a b

**FIGURE 4.3**

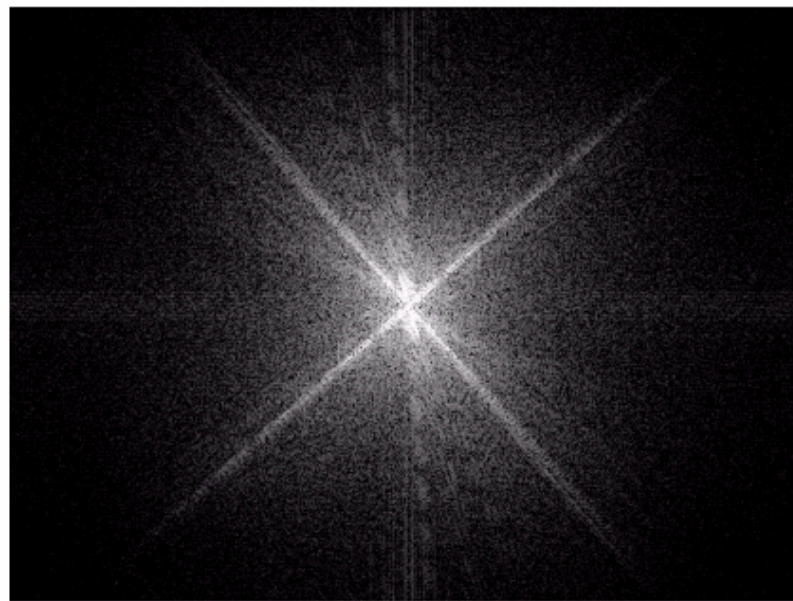
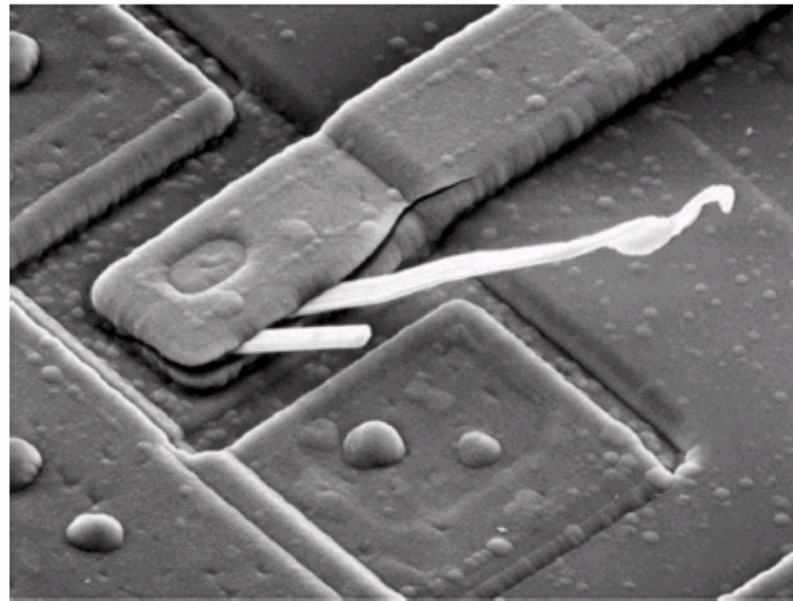
(a) Image of a  $20 \times 40$  white rectangle on a black background of size  $512 \times 512$  pixels.

(b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.





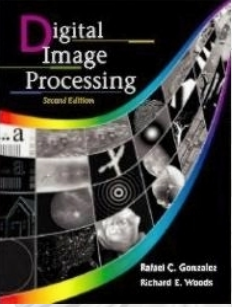
## 4.2.3 Filtering in the Frequency Domain



a  
b

**FIGURE 4.4**

(a) SEM image of a damaged integrated circuit.  
(b) Fourier spectrum of (a).  
(Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

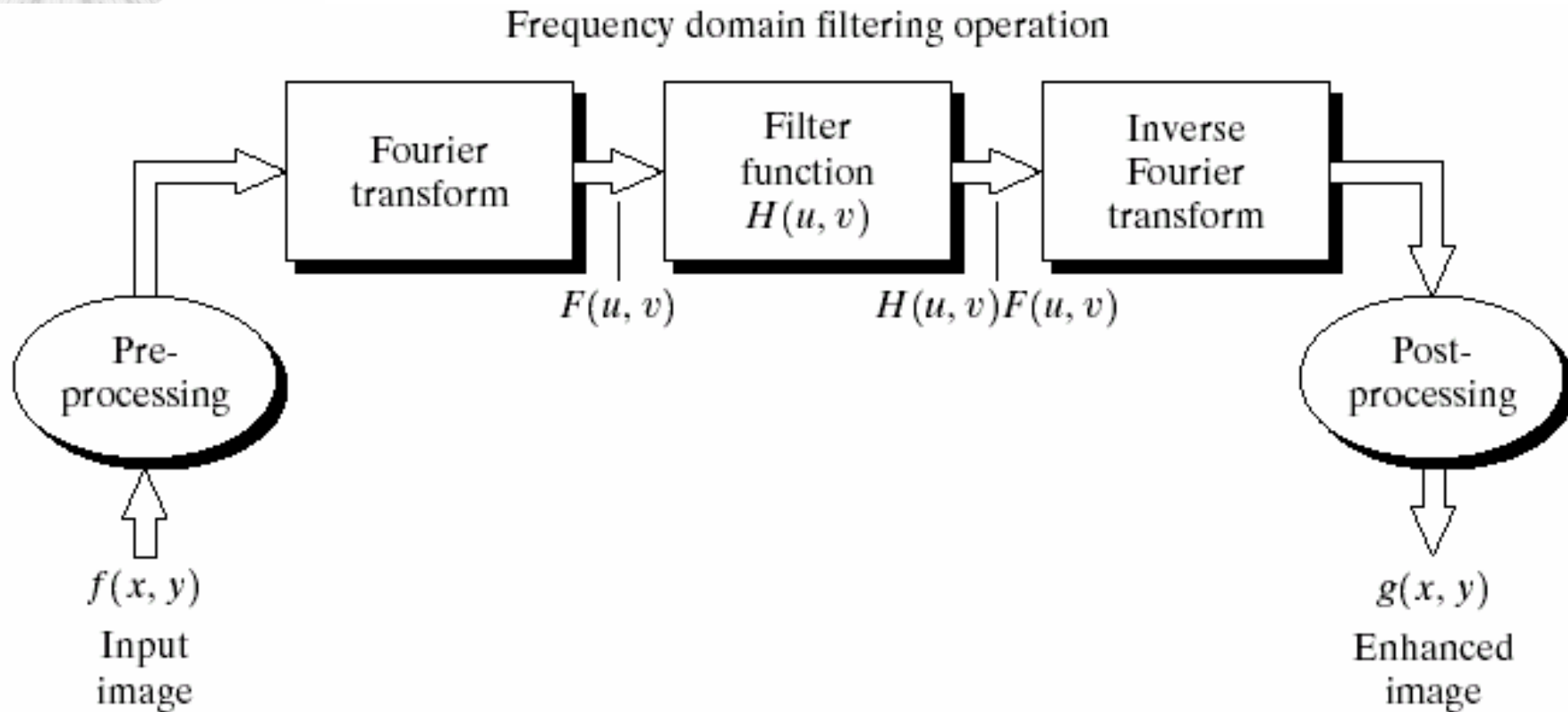


## 4.2.3 Filtering in the Frequency Domain

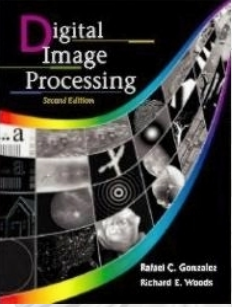
***Filtering in Frequency domain steps:***

- 1) Multiply the input image by  $(-1)^{x+y}$
- 2) Compute DFT of the input image and get  $F(u, v)$
- 3) Multiply  $F(u, v)$  by a filter function  $H(u, v)$ 
$$G(u, v) = F(u, v)H(u, v)$$
- 4) Computer the inverse DFT of  $G(u, v)$ , *i.e.*,  $F^{-1}\{G(u, v)\}$
- 5) Obtain the real part of the  $g(x, y)$
- 6) Multiply  $g(x, y)$  with  $(-1)^{x+y}$

## 4.2.3 Filtering in the Frequency Domain



**FIGURE 4.5** Basic steps for filtering in the frequency domain.

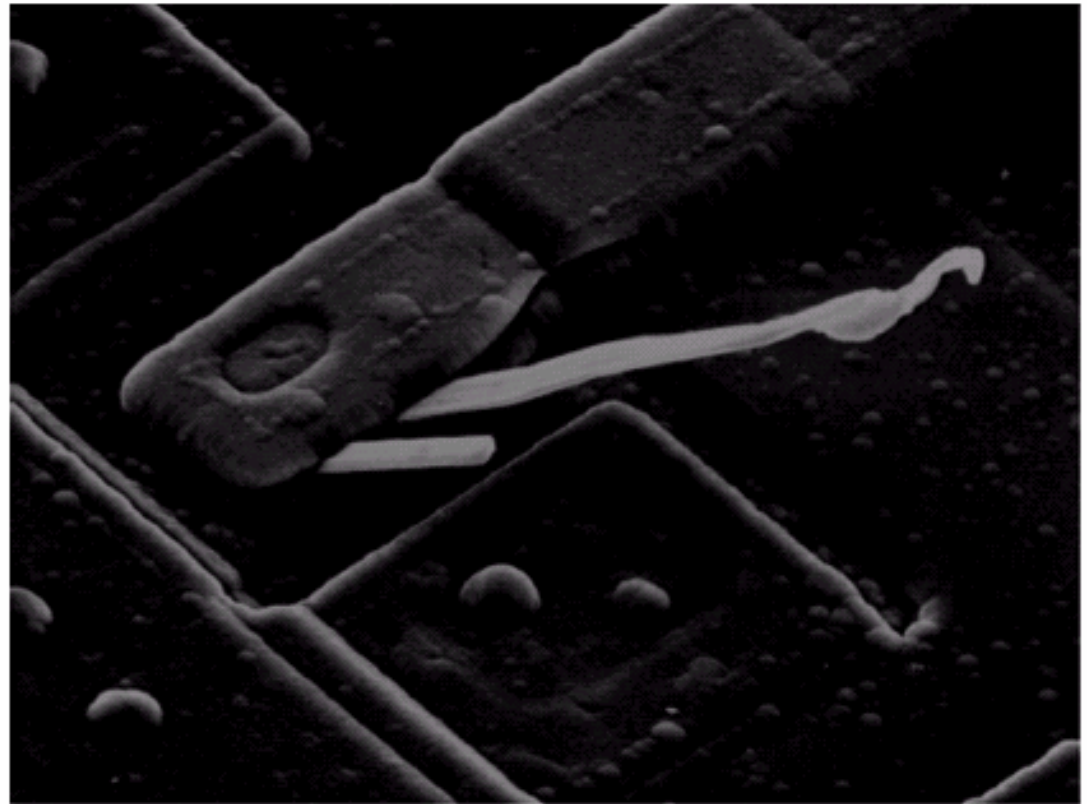
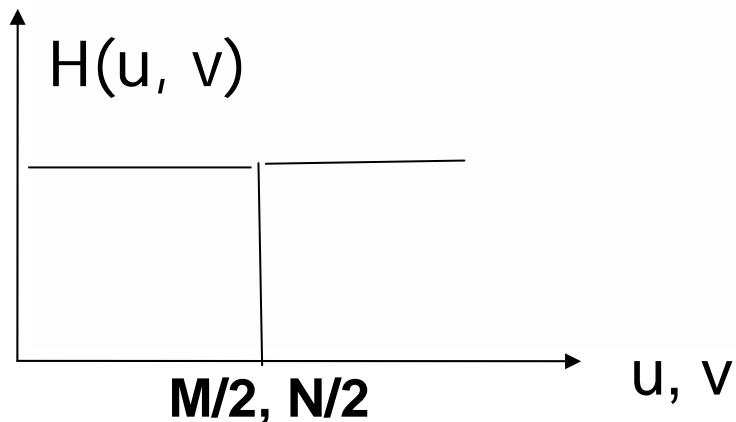


## 4.2.3 Filtering in the Frequency Domain

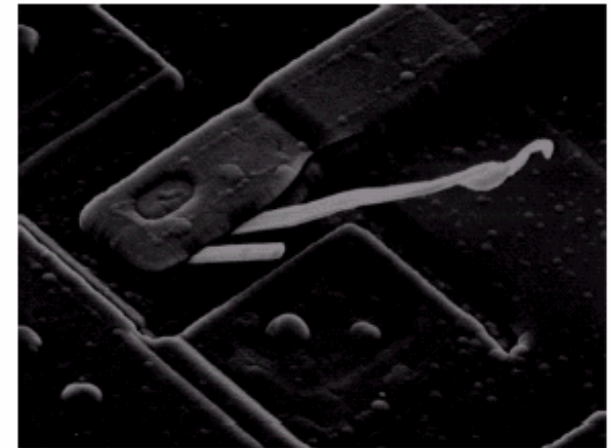
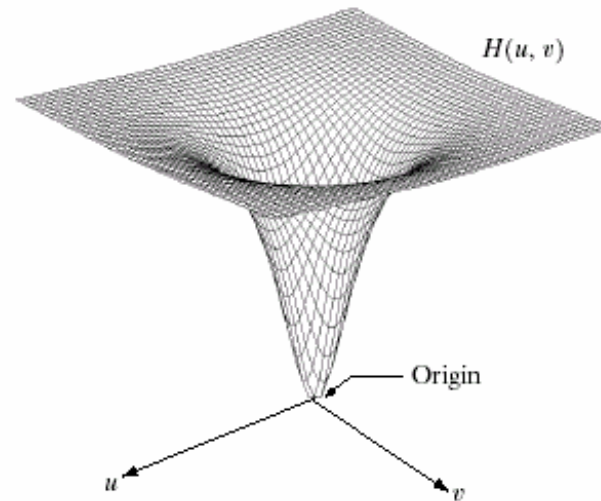
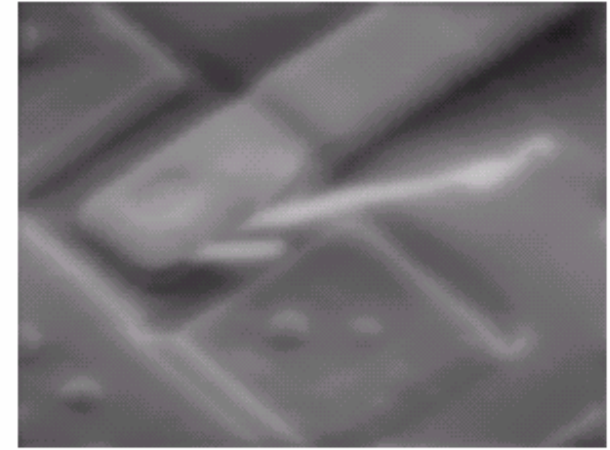
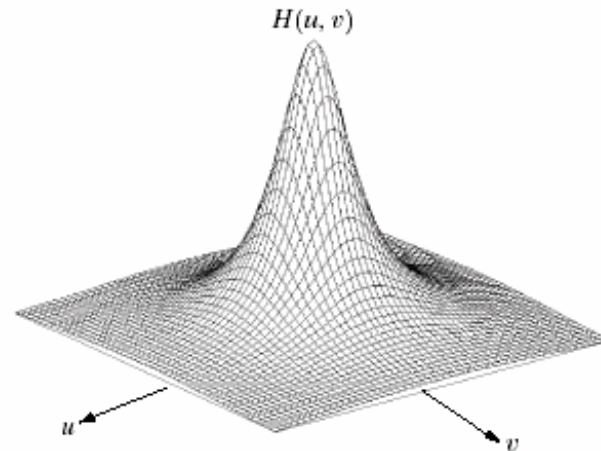
Notch filter:  $H(u, v) = 0$  if  $(u, v) = (M/2, N/2)$ ,  
 $H(u, v) = 1$  otherwise

**FIGURE 4.6**

Result of filtering the image in Fig. 4.4(a) with a notch filter that set to 0 the  $F(0, 0)$  term in the Fourier transform.

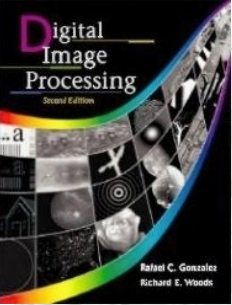


## 4.2.3 Filtering in the Frequency Domain



a b  
c d

**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).

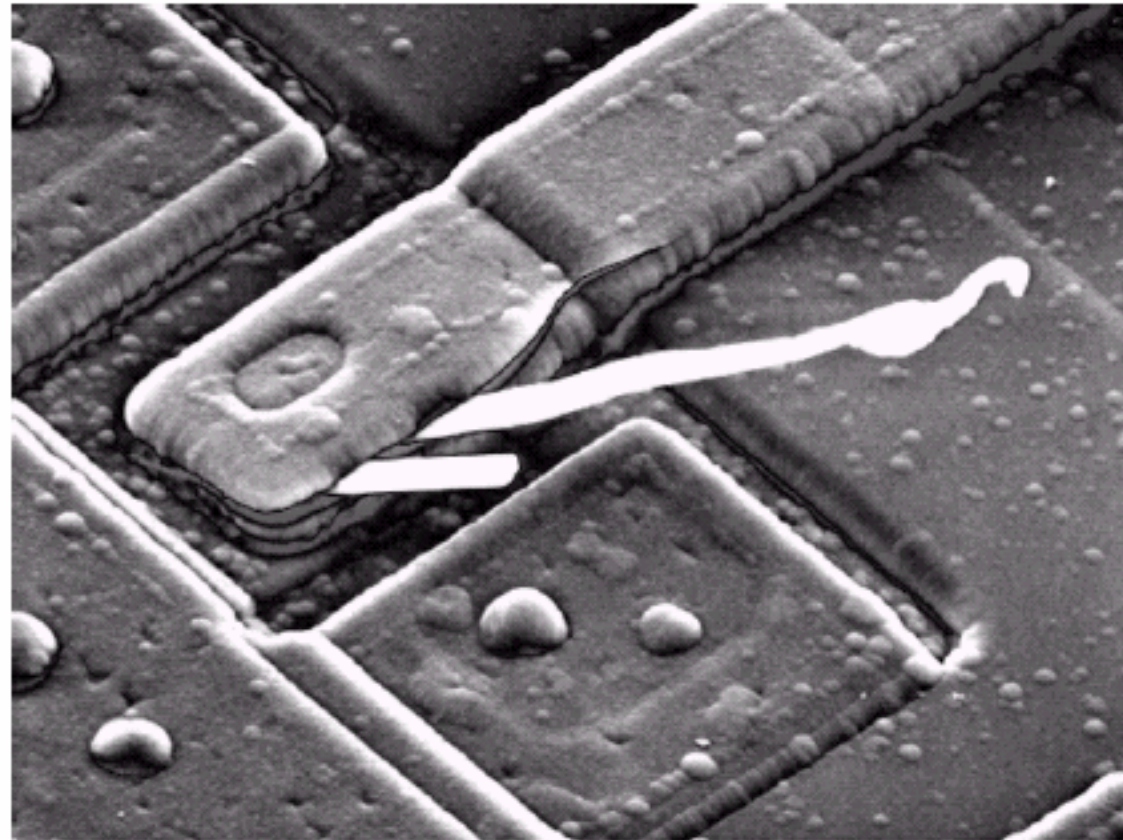


## 4.2.3 Filtering in the Frequency Domain

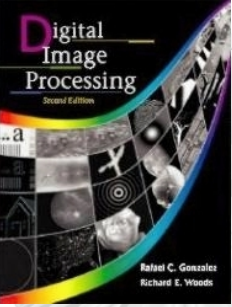
### FIGURE 4.8

Result of highpass filtering the image in Fig. 4.4(a) with the filter in Fig. 4.7(c), modified by adding a constant of one-half the filter height to the filter function. Compare with Fig. 4.4(a).

---





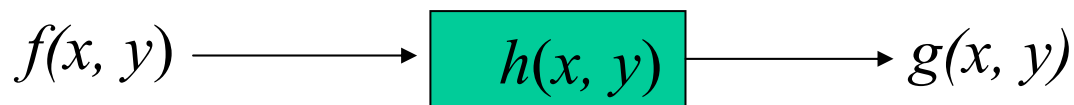


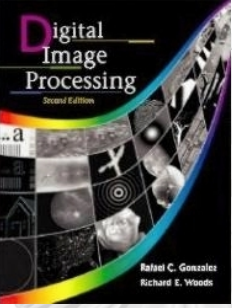
## 4.2.4 Filtering in spatial and frequency domains

- Convolution in **time domain**

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

- $f(x, y) * h(x, y) \Leftrightarrow F(u, v)H(u, v)$   
i.e.,  $f(x, y) * h(x, y) = \mathcal{F}^{-1} \{F(u, v)H(u, v)\}$
- Convolution in **freq domain**
- $F(u, v) * H(u, v) \Leftrightarrow f(x, y)h(x, y)$   
i.e.,  $\mathcal{F}\{f(x, y)h(x, y)\} = F(u, v) * H(u, v)$

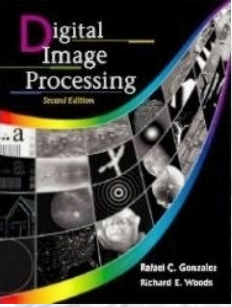




## 4.2.4 Filtering in spatial and frequency domains

- An impulse of strength  $A$ , located at coordinates  $(x_0, y_0)$ , is denoted as  $A\delta(x-x_0, y-y_0)$
- An ***unit impulse*** is defined as  $\delta(x, y)$ . *i.e.*,  $\delta(x, y)=1$  only when  $x=0, y=0$ ,  $\delta(x, y)=0$  otherwise.
- For function  $s(x, y)$  (a) sampled at  $(0, 0)$  is denoted as  $\sum_x \sum_y s(x, y)\delta(x, y) = s(0, 0)$ , (b) sampled at  $(x_0, y_0)$ , is  $\sum_x \sum_y s(x, y)\delta(x - x_0, y - y_0) = s(x_0, y_0)$ ,
- The Fourier transform of  $\delta(x, y)$  is

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \delta(x, y) e^{-j2\pi(ux/M + vy/N)} = \frac{1}{MN}$$



## 4.2.4 Filtering in spatial and frequency domains

- If we let  $f(x, y) = \delta(x, y)$  then

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

- $\delta(x, y) * h(x, y) \Leftrightarrow F \{ \delta(x, y) \} H(u, v)$

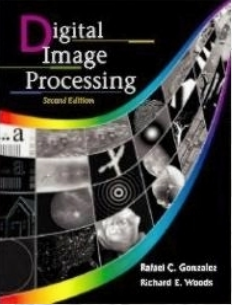
$$h(x, y) \Leftrightarrow H(u, v)$$

- Gaussian filters – **lowpass filtering**

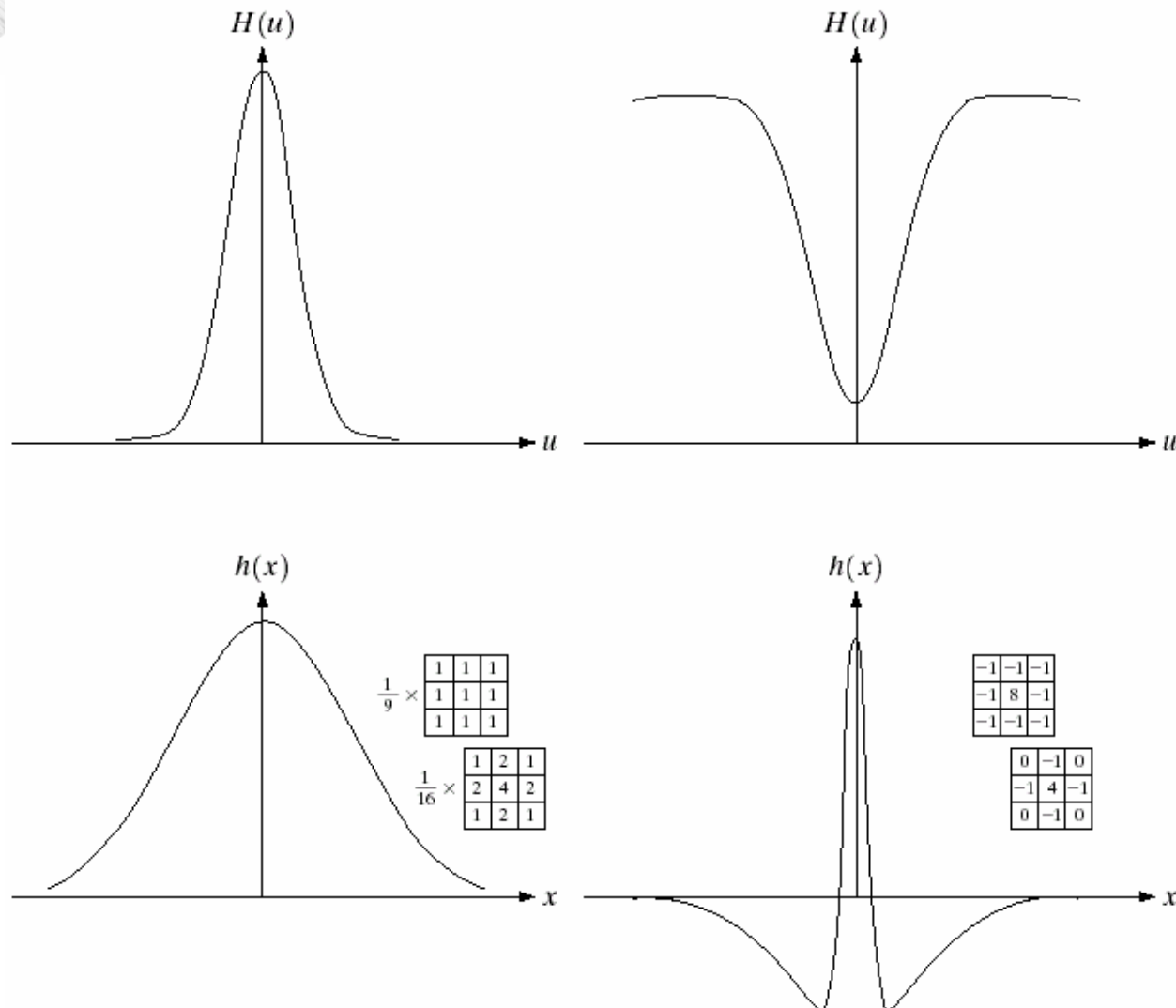
$$H(u) = A e^{-u^2 / 2\sigma^2} \quad h(x) = \sqrt{2\pi}\sigma A e^{-2\pi u^2 \sigma^2 x^2}$$

- Difference of two Gaussian filters – **highpass filtering**

$$H(u) = A e^{-u^2 / 2\sigma_1^2} - B e^{-u^2 / 2\sigma_2^2}$$



## 4.2.4 Filtering in spatial and frequency domains



a	b
c	d

**FIGURE 4.9**  
 (a) Gaussian frequency domain lowpass filter.  
 (b) Gaussian frequency domain highpass filter.  
 (c) Corresponding lowpass spatial filter.  
 (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.



## 4.3 Smoothing Frequency-Domain Filters

- Freq-Domain Filtering:

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

- Filter  $H(u, v)$
- Ideal filter
- Butterworth filter
- Gaussian Filter



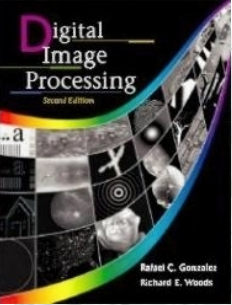
## 4.3.1 Ideal Low pass filter

- $H(u, v)$  with **Sharp cut-off** at **cut-off frequency**  $D_0$ ,  
i.e.,  
$$H(u, v) = 1 \text{ if } D(u, v) \leq D_0$$
$$= 0 \text{ if } D(u, v) > D_0$$
- For image of size  $M \times N$ , the center is at  $(u, v) = (M/2, N/2)$ . The distance from any point to the center is

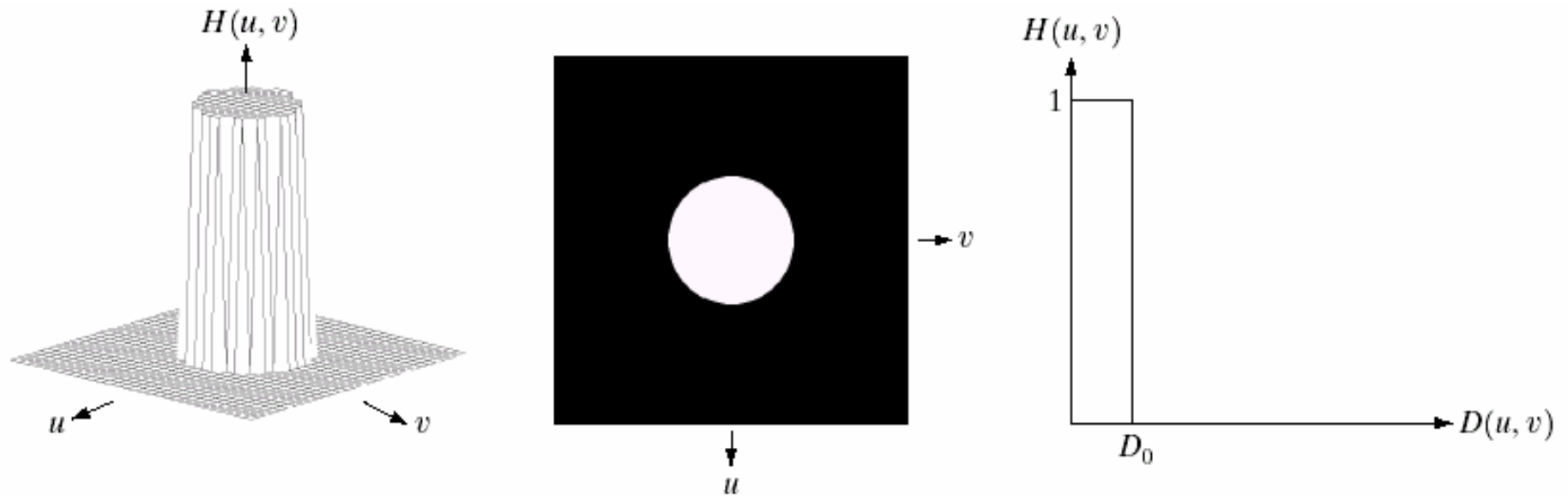
$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

- **Cut off frequency** is  $D_0$
- Total power:

$$P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u, v)$$



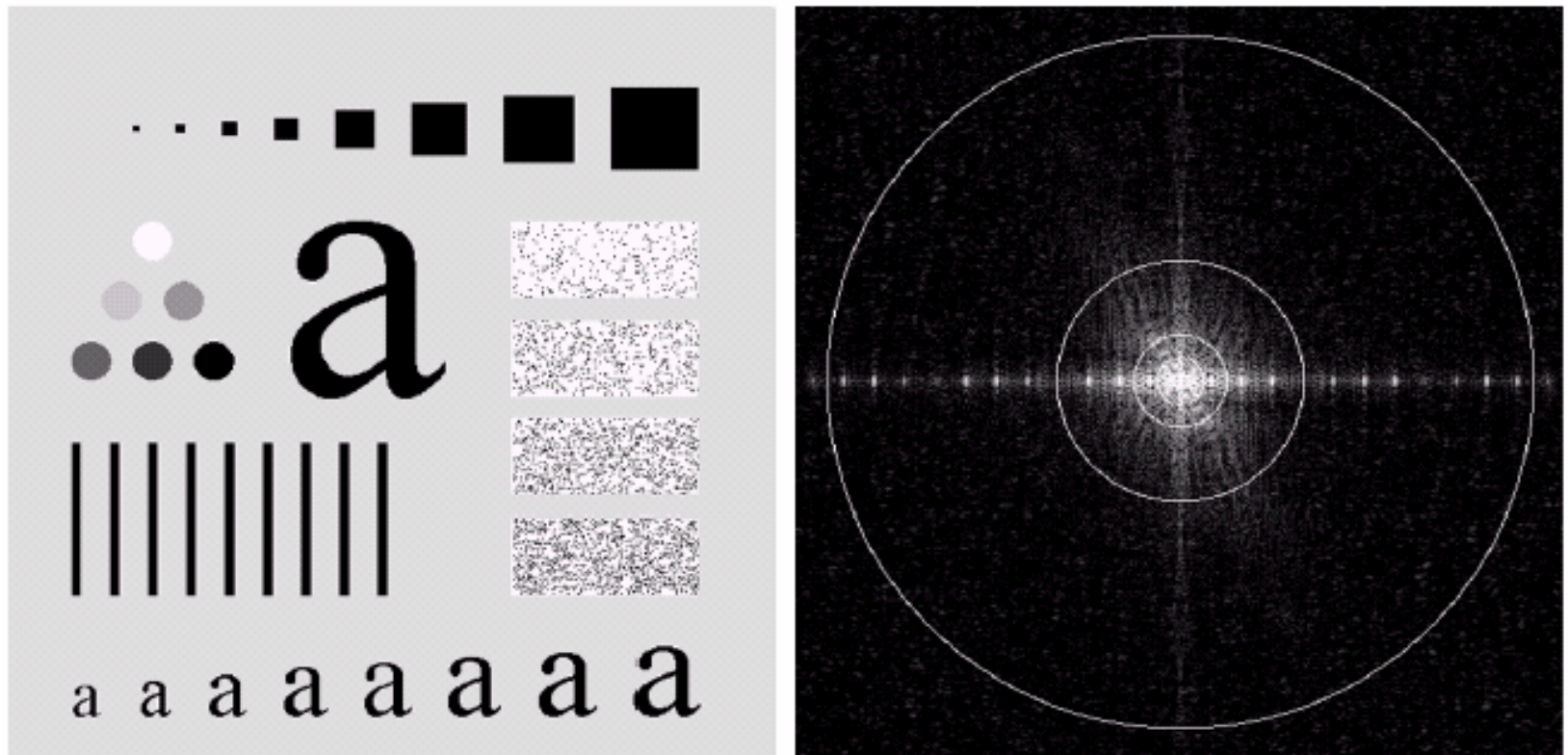
## 4.3.1 Ideal Low pass filter



a b c

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

## 4.3.1 Ideal Low pass filter

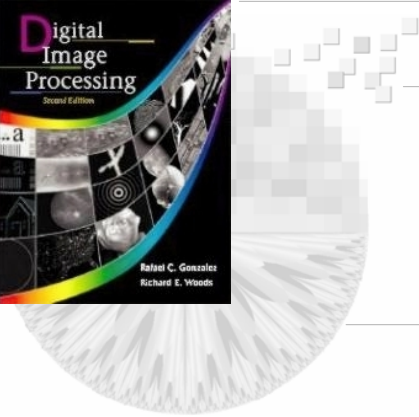


a b

**FIGURE 4.11** (a) An image of size  $500 \times 500$  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.

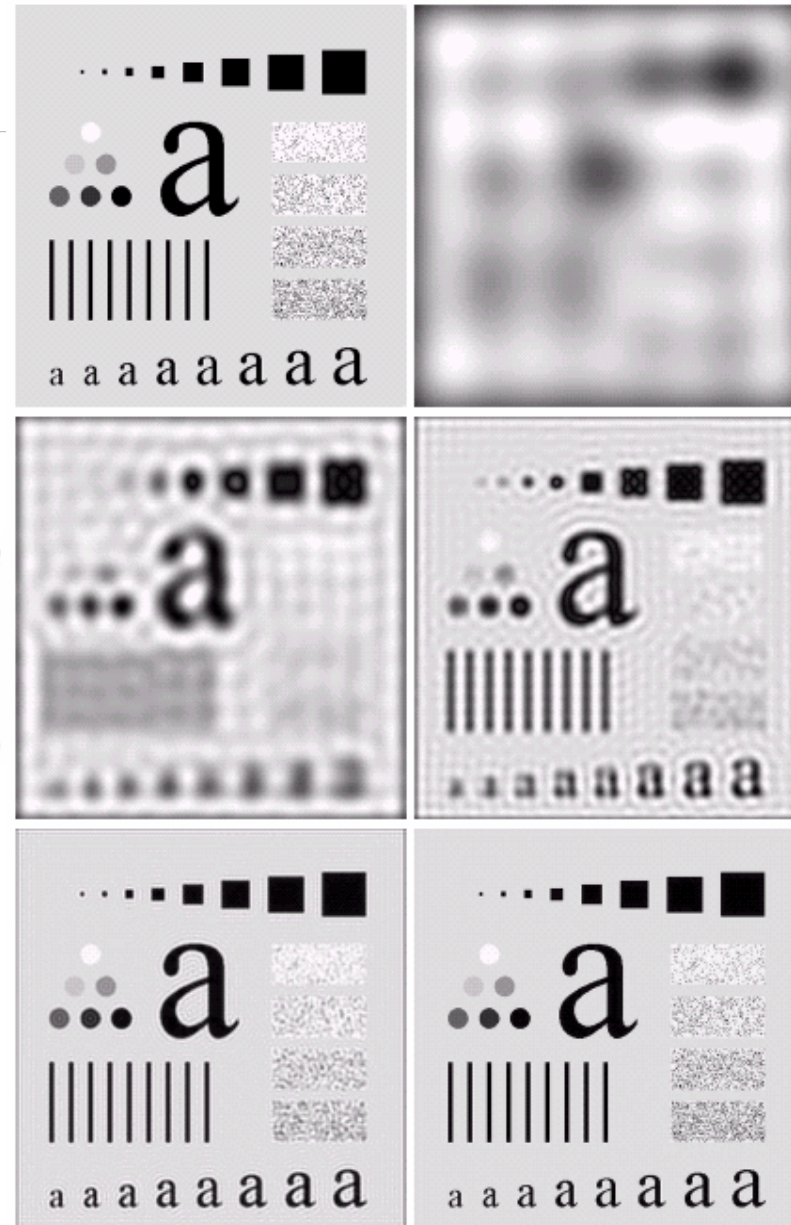


## 4.3.1 Ideal Low pass filter

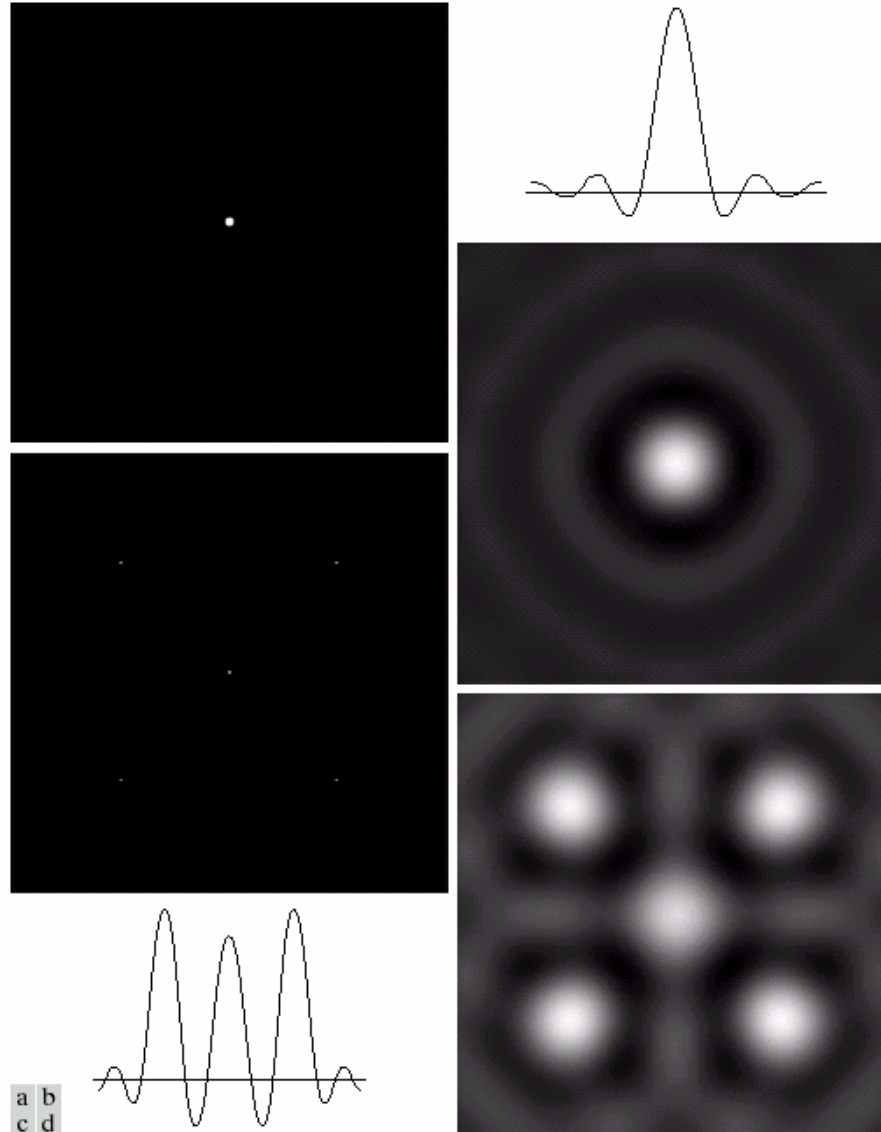


a b  
c d  
e f

**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.



## 4.3.1 Ideal Low pass filter



a b  
c d

**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.



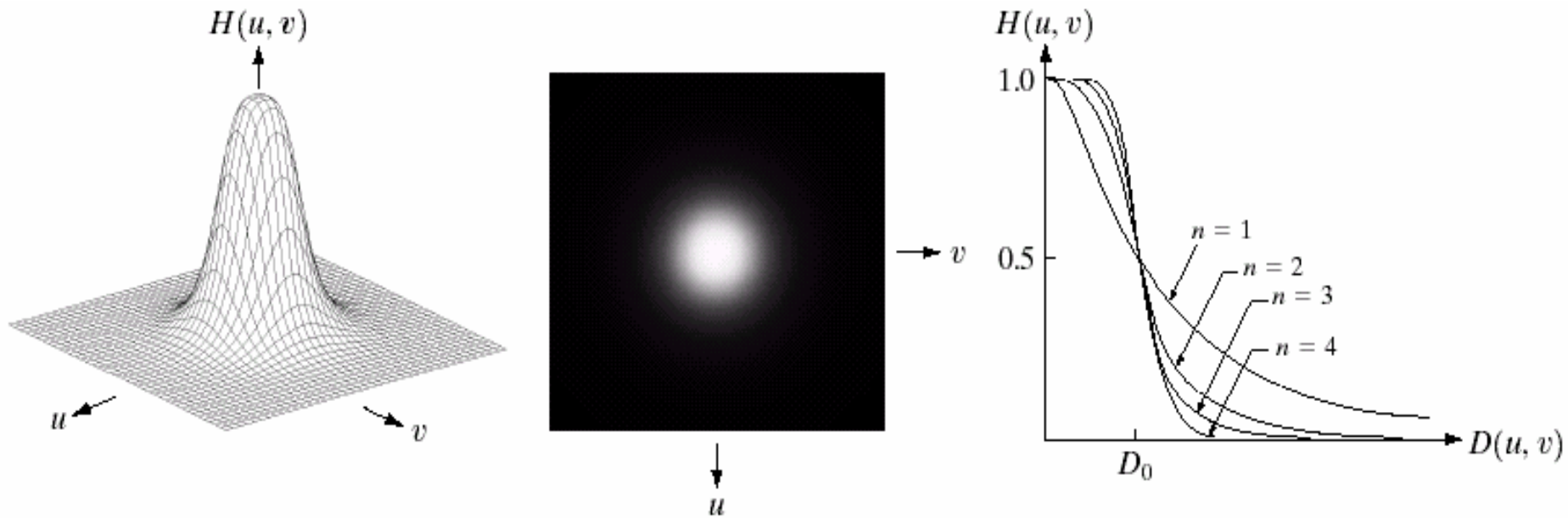
## 4.3.2 Butterworth Lowpass Filter (BLPF)

- Butterworth filter has no sharp cutoff

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

- At cutoff frequency  $D_0$ :  $H(u, v) = 0.5$

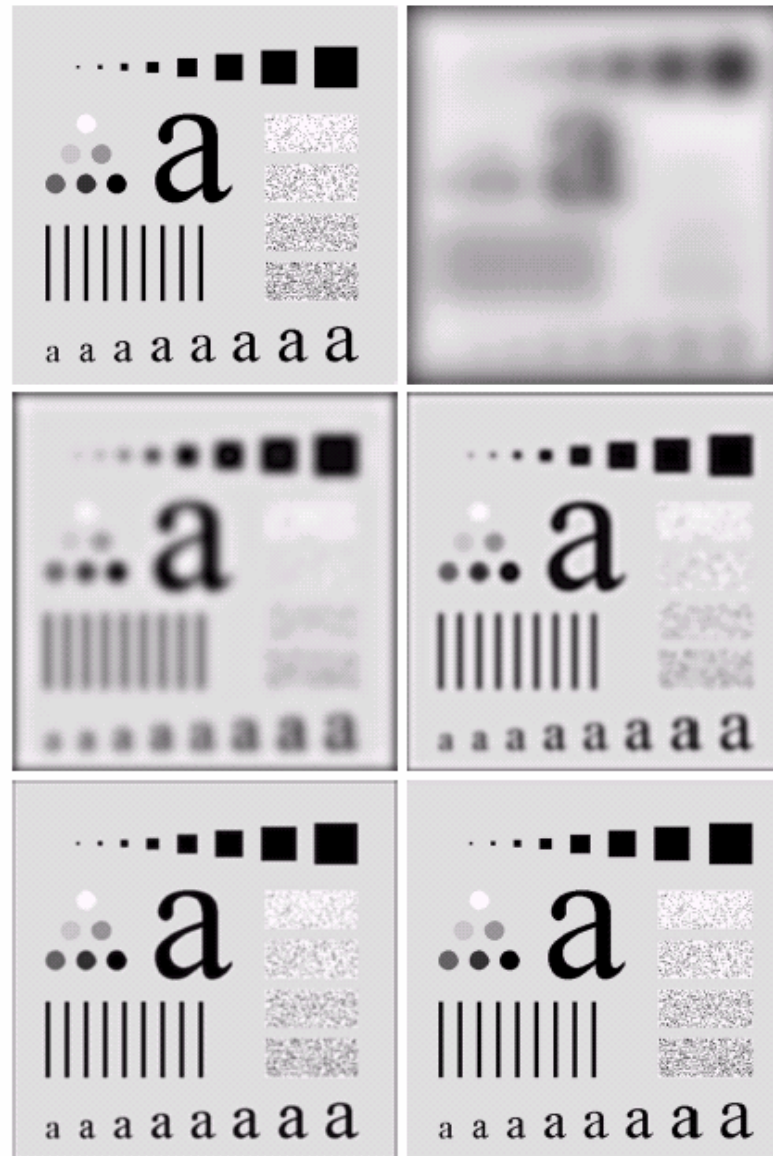
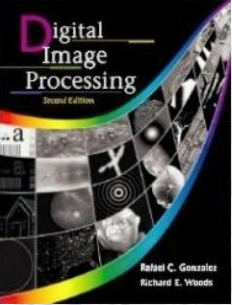
## 4.3.2 Butterworth Lowpass Filter



a b c

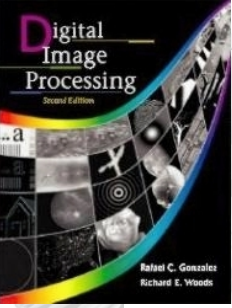
**FIGURE 4.14** (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

## 4.3.2 Butterworth Lowpass Filter

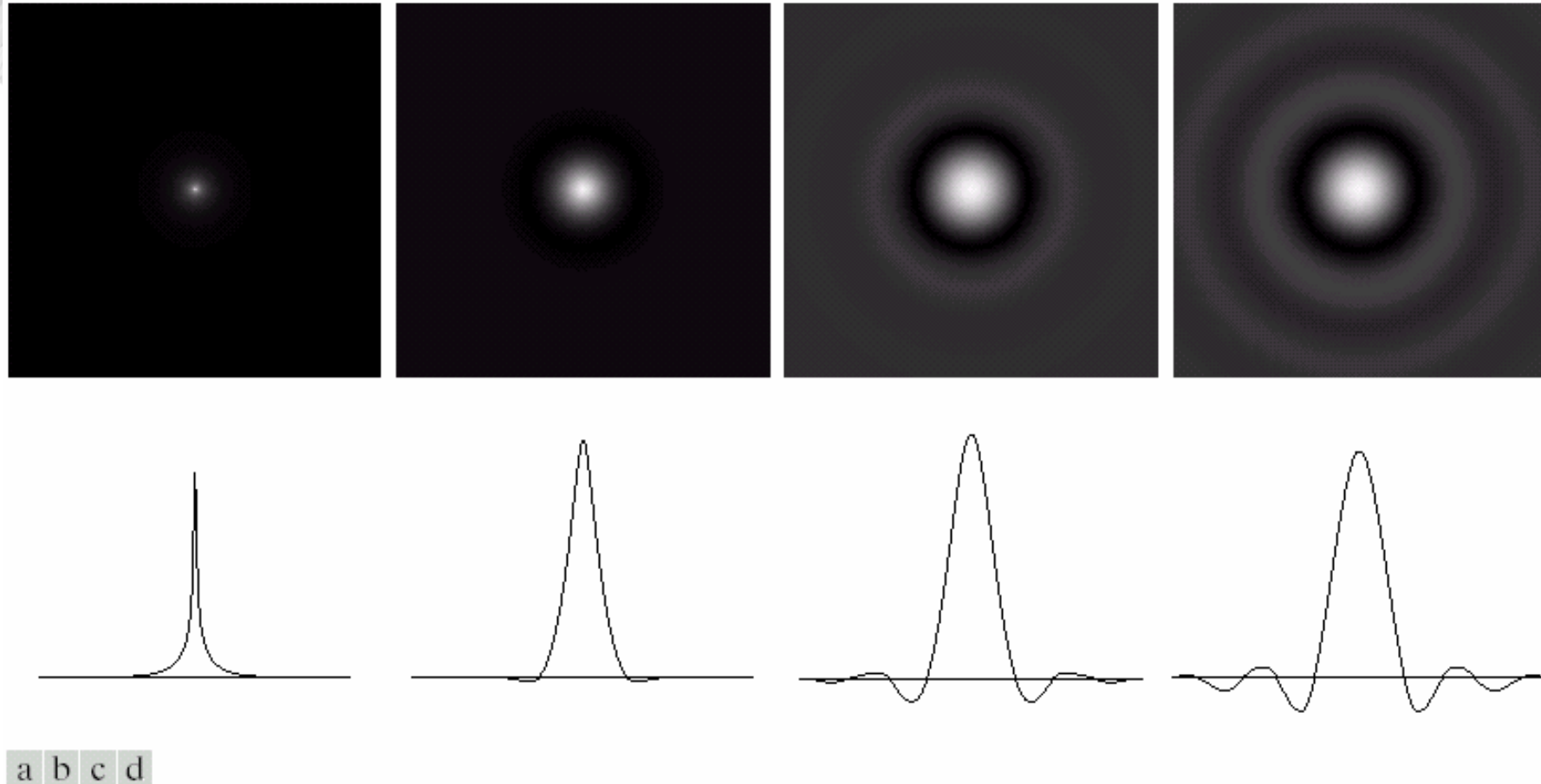


a  
b  
c  
d  
e  
f

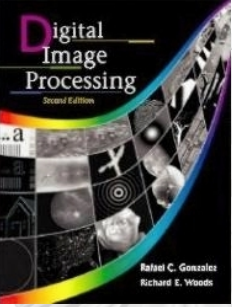
**FIGURE 4.15** (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.



## 4.3.2 Butterworth Lowpass Filter



**FIGURE 4.16** (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.



### 4.3.3 Gaussian Lowpass Filter (GLPF)

- Gaussian low-pass filter (GLPF)

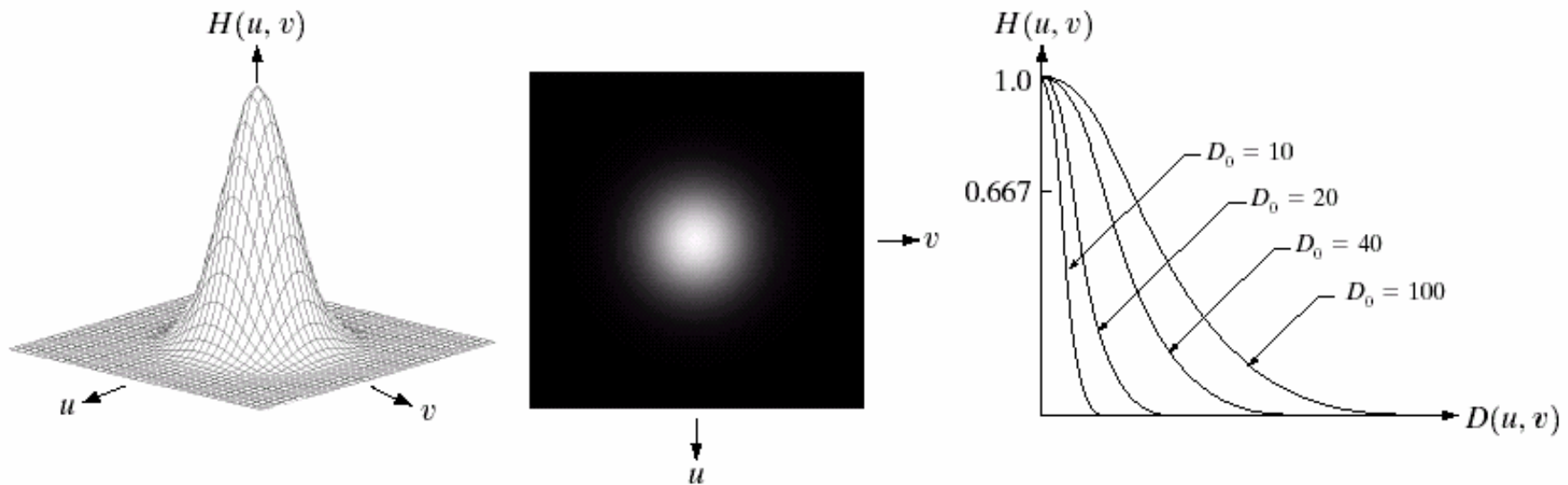
$$H(u, v) = e^{-D^2(u, v)/2\sigma^2}$$

- Let  $\sigma = D_0$

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

- When  $D(u, v) = D_0$ ,  $H(u, v) = 0.667$

## 4.3.3 Gaussian Lowpass Filter

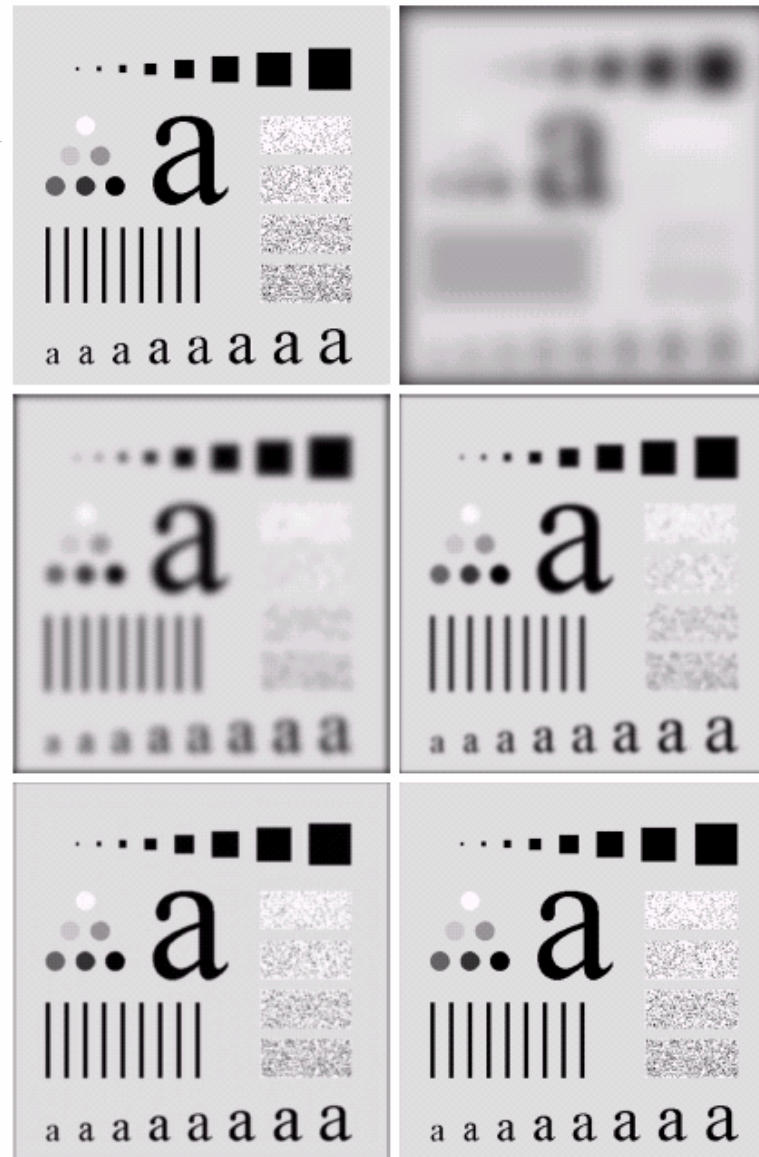
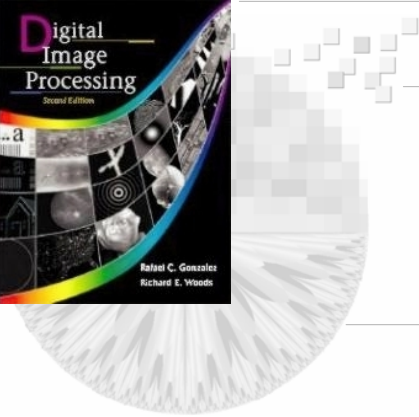


a b c

**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  $D_0$ .

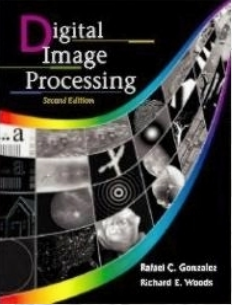


## 4.3.3 Gaussian Lowpass Filter



**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.

a b  
c d  
e f



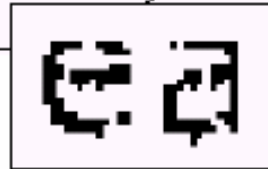
## 4.3.4 Other Lowpass filtering examples

a b

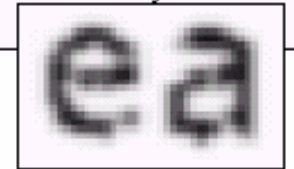
**FIGURE 4.19**

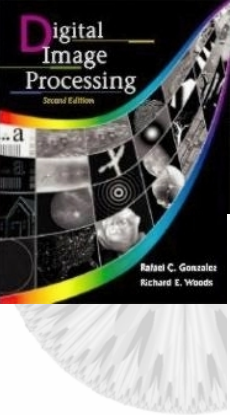
(a) Sample text of poor resolution (note broken characters in magnified view).  
 (b) Result of filtering with a GLPF (broken character segments were joined).

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.





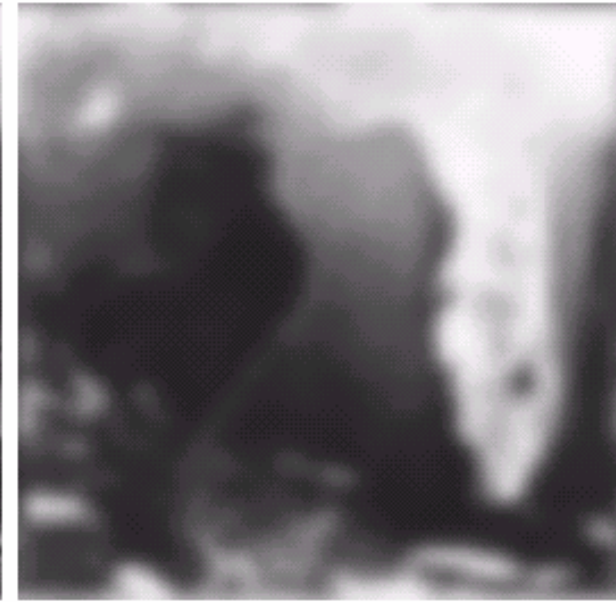
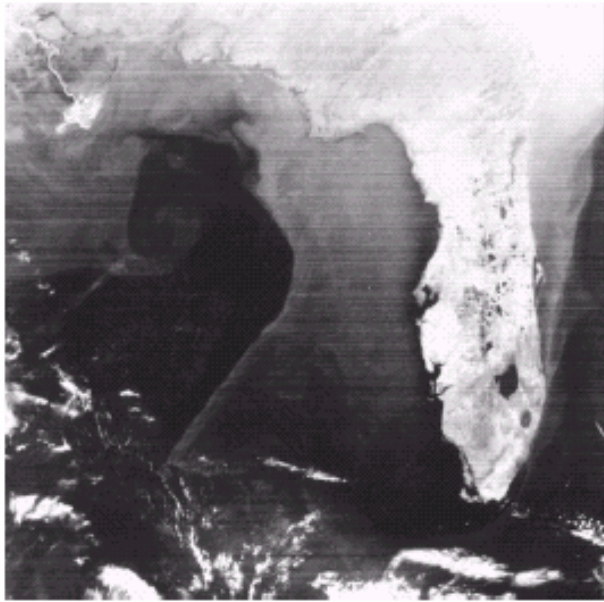
## 4.3.4 Other Lowpass filtering examples



a b c

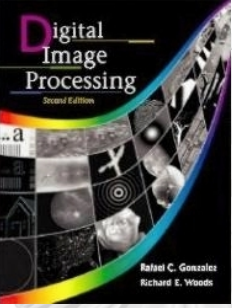
**FIGURE 4.20** (a) Original image ( $1028 \times 732$  pixels). (b) Result of filtering with a GLPF with  $D_0 = 100$ . (c) Result of filtering with a GLPF with  $D_0 = 80$ . Note reduction in skin fine lines in the magnified sections of (b) and (c).

## 4.3.4 Other Lowpass filtering examples



a b c

**FIGURE 4.21** (a) Image showing prominent scan lines. (b) Result of using a GLPF with  $D_0 = 30$ . (c) Result of using a GLPF with  $D_0 = 10$ . (Original image courtesy of NOAA.)



## 4.4 Sharpening Frequency-Domain Filter

- Highpass filtering:

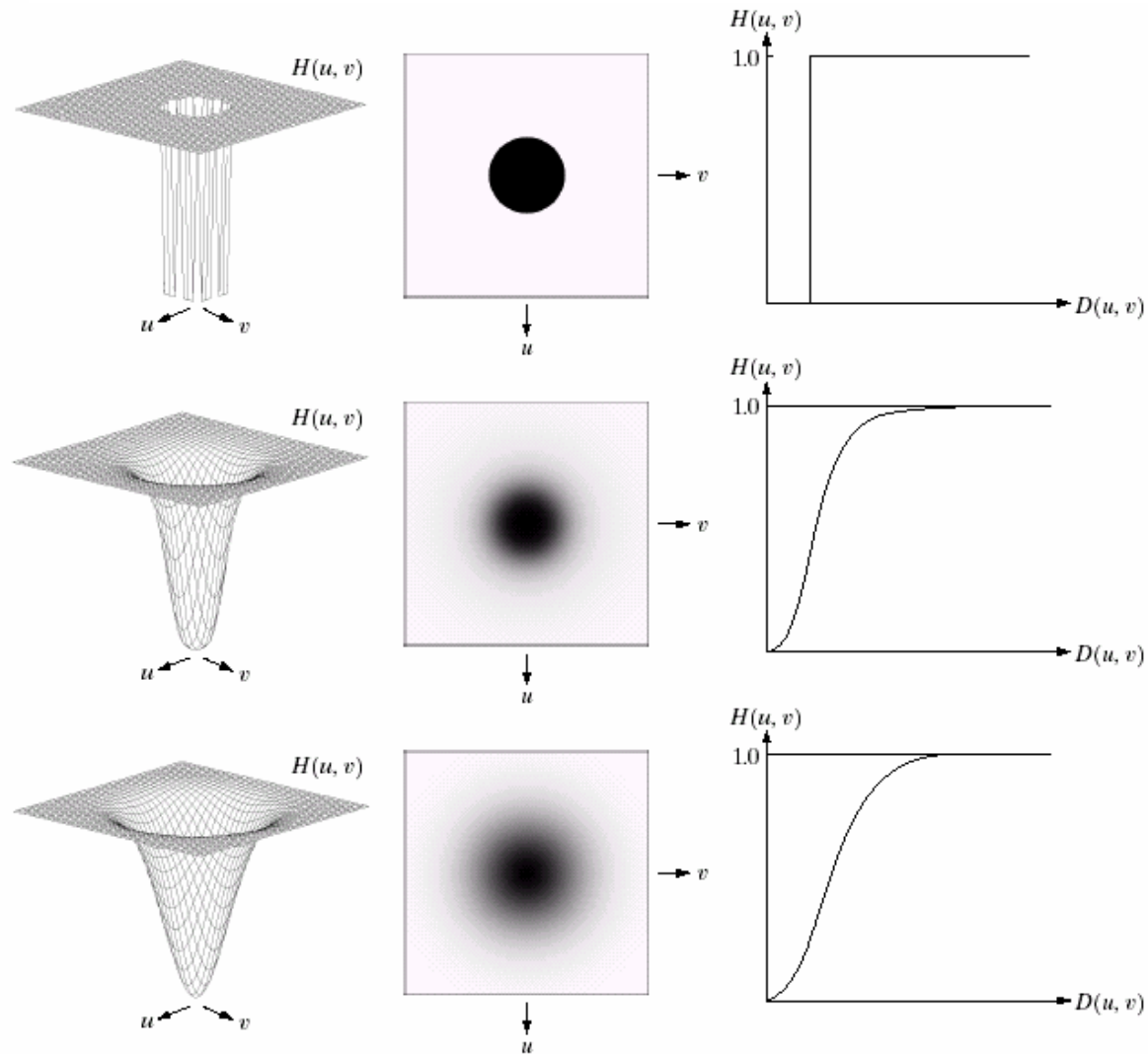
$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

- Given a lowpass filter  $H_{lp}(u, v)$ , find the *spatial representation* of the highpass filter

(1) Compute the inverse DFT of  $H_{lp}(u, v)$

(2) Multiply the real part of the result with  $(-1)^{x+y}$

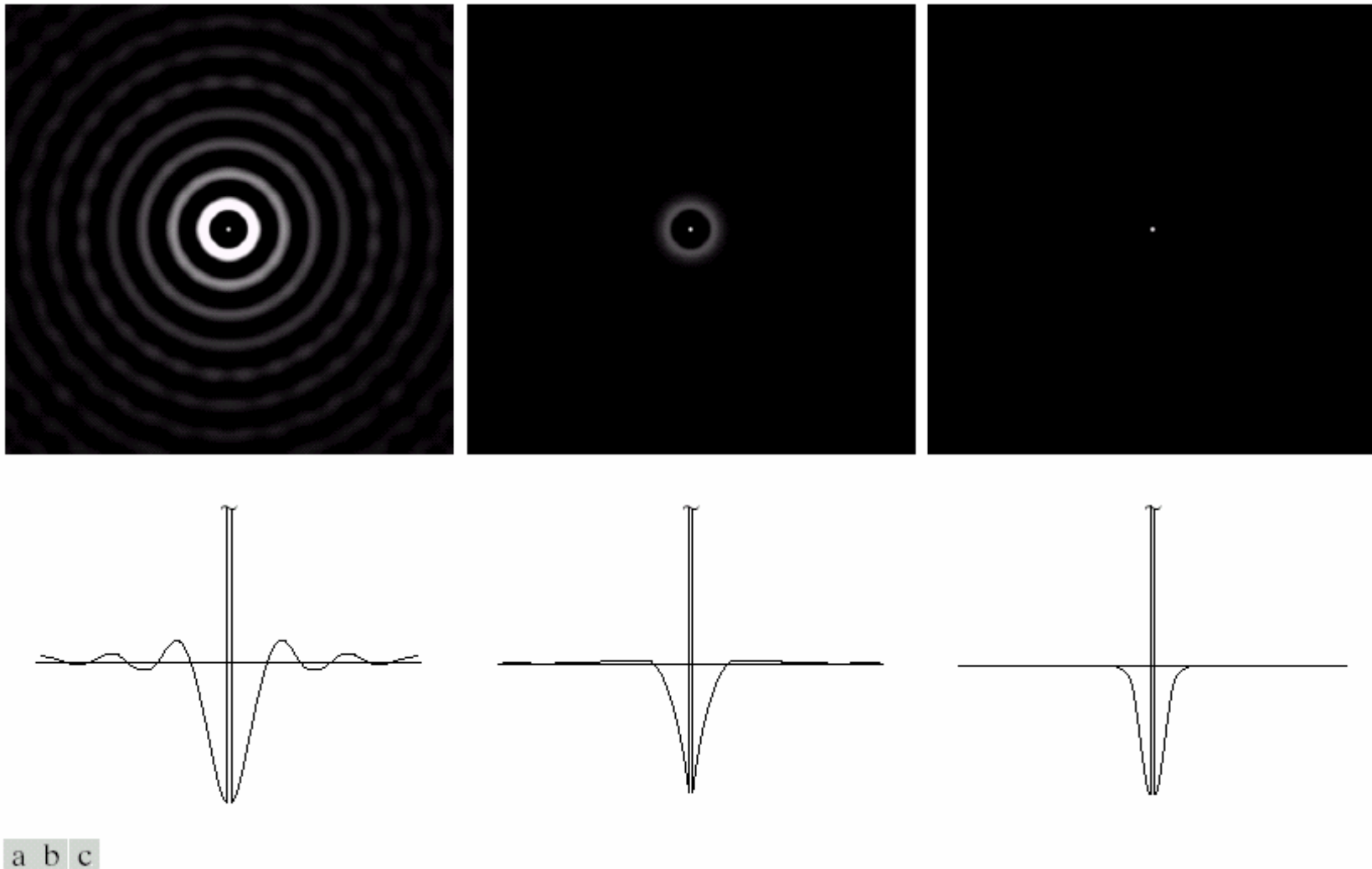
## 4.4 Sharpening Frequency-Domain Filter



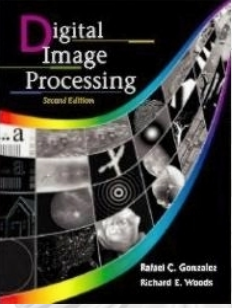
a	b	c
d	e	f
g	h	i

**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.

## 4.4 Sharpening Frequency-Domain Filter



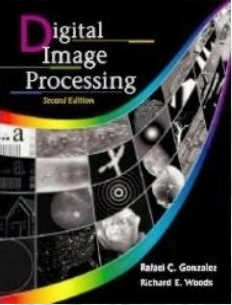
**FIGURE 4.23** Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.



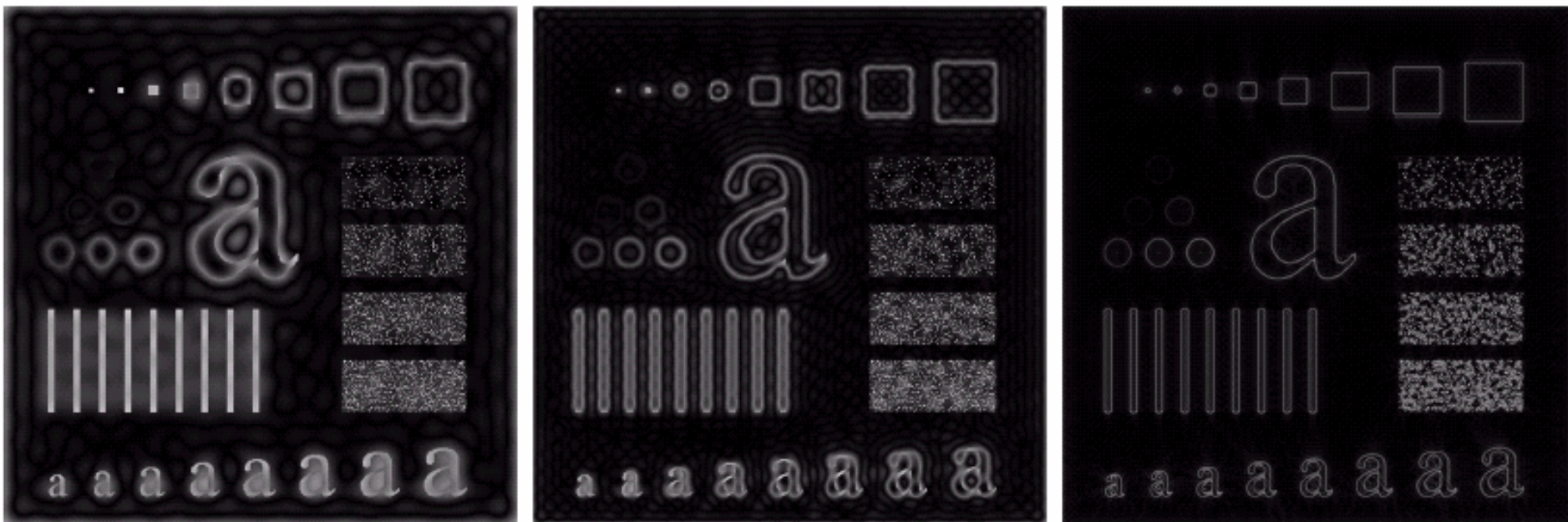
## 4.4.1 Ideal Highpass Filter (IHPF)

- $H(u, v) = 0$  if  $D(u, v) \leq D_0$   
 $= 1$  if  $D(u, v) > D_0$
- The center is at  $(u, v) = (M/2, N/2)$   
$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$
- Cut off frequency is  $D_0$





## 4.4.1 Ideal Highpass Filter (IHPF)



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).

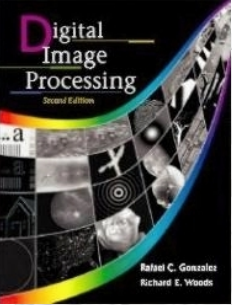


## 4.4.2 Butterworth Highpass Filter (BHPF)

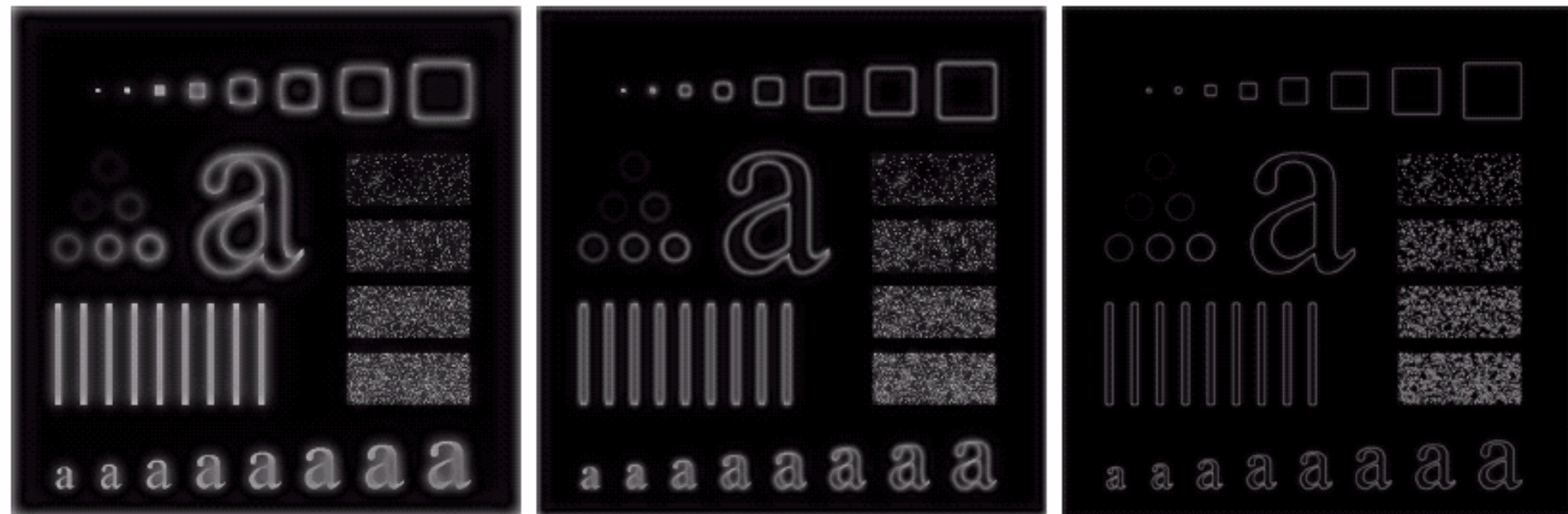
- Butterworth filter has no sharp cutoff

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

- At cutoff frequency  $D_0$ :  $H(u, v) = 0.5$



## 4.4.2 Butterworth Highpass Filter (BHPF)



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 **IHPF**



## 4.4.3 Gaussian Highpass Filter (GHPF)

- Gaussian Highpass filter (GHPF)

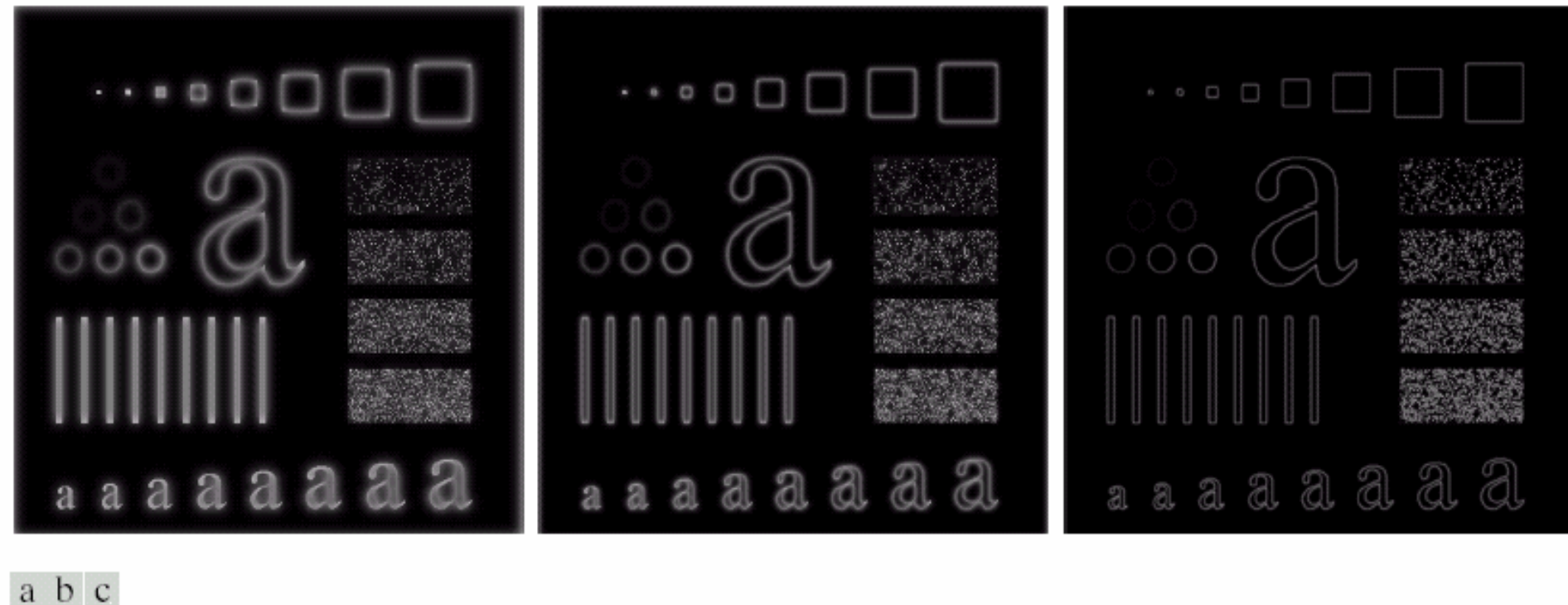
$$H(u, v) = 1 - e^{-D^2(u, v) / 2\sigma^2}$$

- Let  $\sigma = D_0$

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

- When  $D(u, v) = D_0$ ,  $H(u, v) = 0.667$

## 4.4.3 Gaussian Highpass Filter



**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.



## 4.4.4 Laplacian in the Frequency Domain

- Fourier property:

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

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

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

- $H(u, v) = -(u^2 + v^2)$
- If  $F(u, v)$  has been centered by  $f(x, y)(-1)^{x+y}$  then  $H(u, v) = -[(u - M/2)^2 + (v - N/2)^2]$



## 4.4.4 Laplacian in the Frequency Domain

- Laplacian filtering

$$\nabla^2 f(x, y) = \mathcal{F}^{-1} \{ -[(u - M/2)^2 + (v - N/2)^2] F(u, v) \}$$

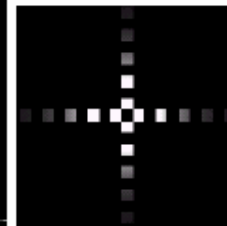
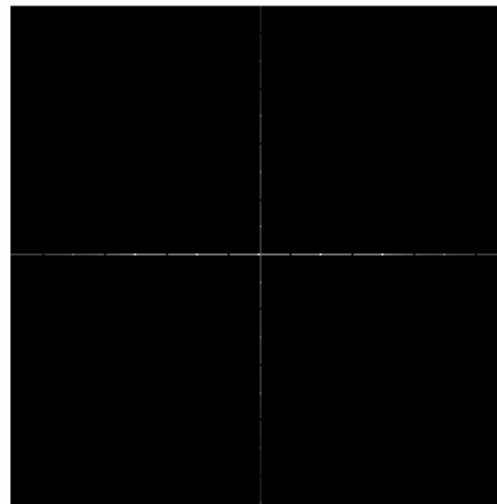
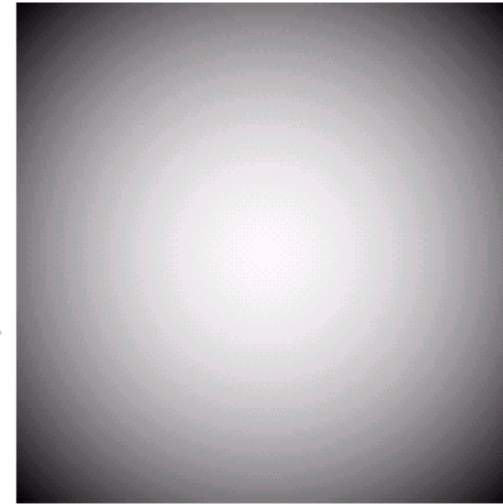
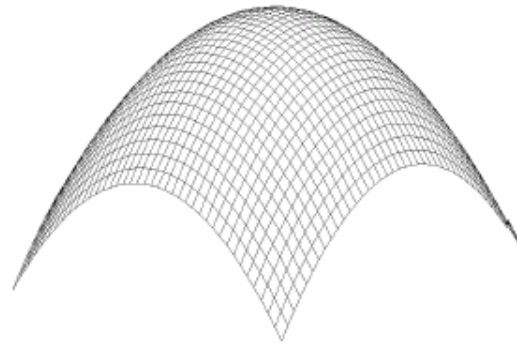
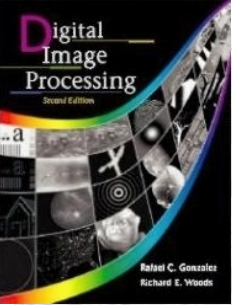
- Enhanced image  $g(x, y) = f(x, y) - \nabla^2 f(x, y)$

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

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

- $g(x, y) = \mathcal{F}^{-1} \{ [1 - ((u - M/2)^2 + (v - N/2)^2)] F(u, v) \}$

## 4.4.4 Laplacian in the Frequency Domain



0	1	0
1	-4	1
0	1	0

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.





## 4.4.4 Laplacian in the Frequency Domain

a	b
c	d

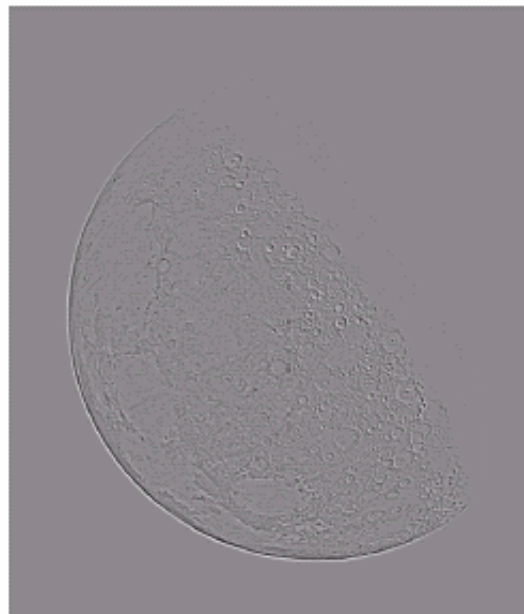
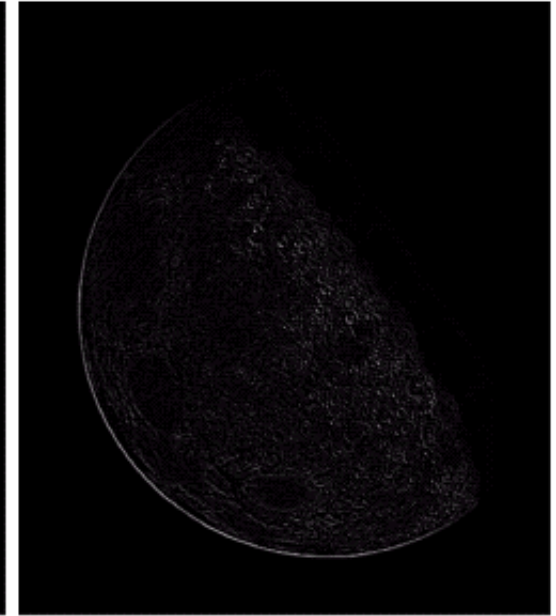
**FIGURE 4.28**

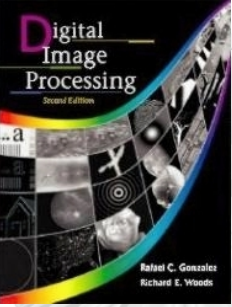
(a) Image of the North Pole of the moon.

(b) Laplacian filtered image.

(c) Laplacian image scaled.

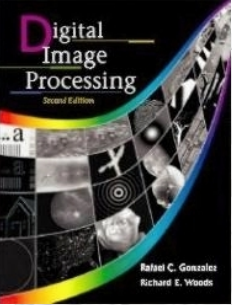
(d) Image enhanced by using Eq. (4.4-12). (Original image courtesy of NASA.)





## 4.4.5 Unsharp masking High-boost filtering and High frequency emphasis filtering

- Highpass filtered image:  $f_{hp}(x, y) = f(x, y) - f_{lp}(x, y)$
- High-boost image:  $f_{hb}(x, y) = Af(x, y) - f_{lp}(x, y)$   
 or  $f_{hb}(x, y) = (A-1)f(x, y) - f_{hp}(x, y)$
- In Freq. Domain:  $F_{hp}(u, v) = F(u, v) - F_{lp}(u, v)$   
 The highpass filter:  $H_{hp}(u, v) = 1 - H_{lp}(u, v)$   
 The high-boost filter:  $H_{hb}(u, v) = (A-1) + H_{hp}(u, v), A \geq 1$
- High frequency emphasis filter:  
 $H_{hfe}(u, v) = a + bH_{hp}(u, v)$  where  $a \geq 0, b > a$   
 $a = 0.25 \sim 0.5, b = 1.5 \sim 2.0$



## 4.4.5 Unsharp masking High-boost filtering and High frequency emphasis filtering

a b  
c d

**FIGURE 4.29**

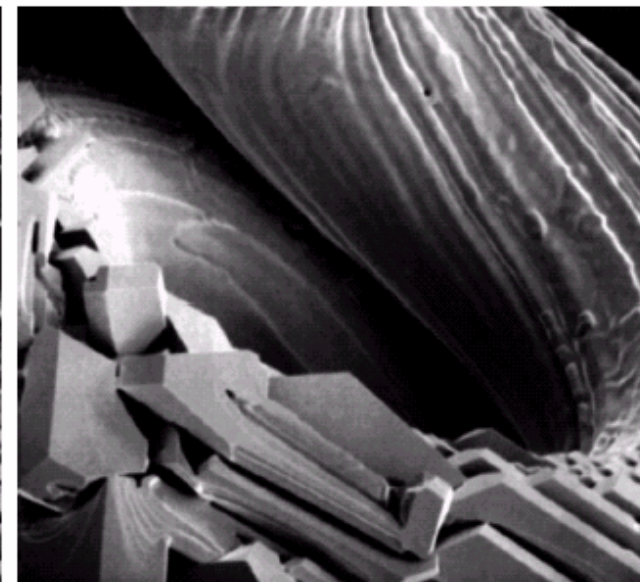
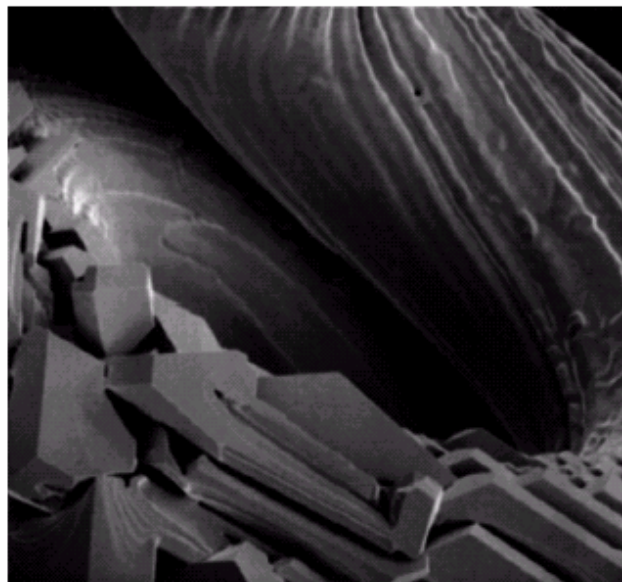
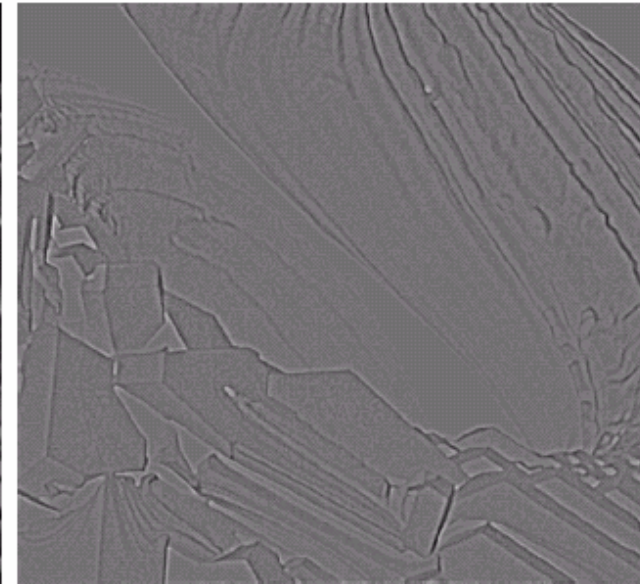
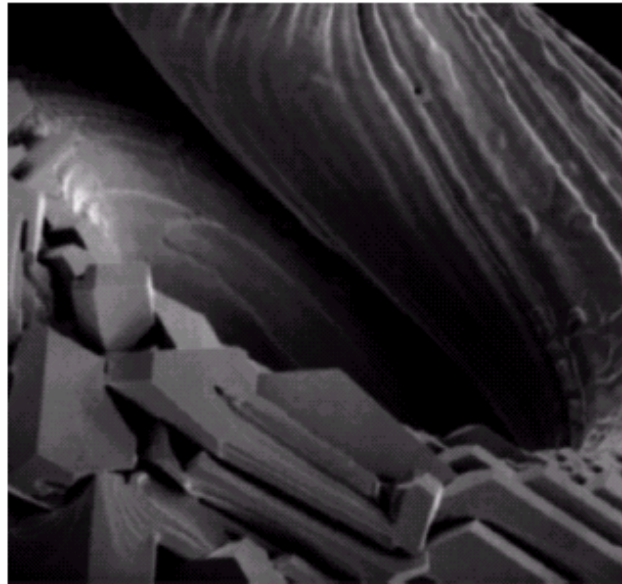
Same as Fig. 3.43, but using frequency domain filtering. (a) Input image.

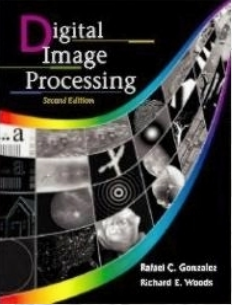
(b) Laplacian of (a).

(c) Image obtained using Eq. (4.4-17) with  $A = 2$ .

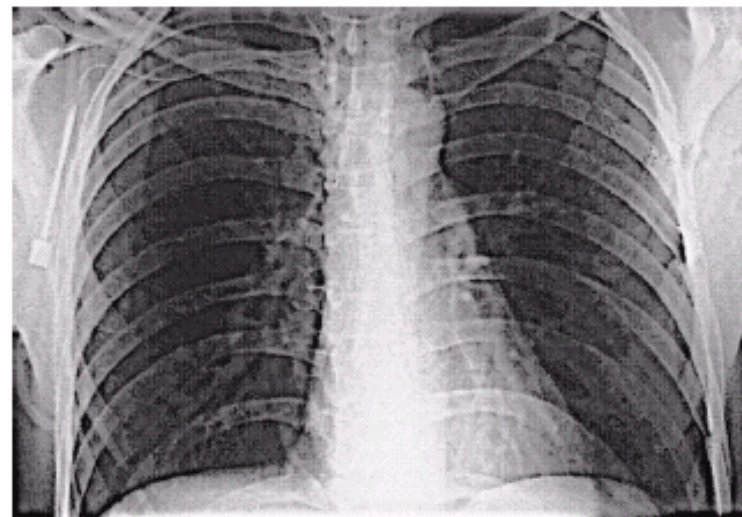
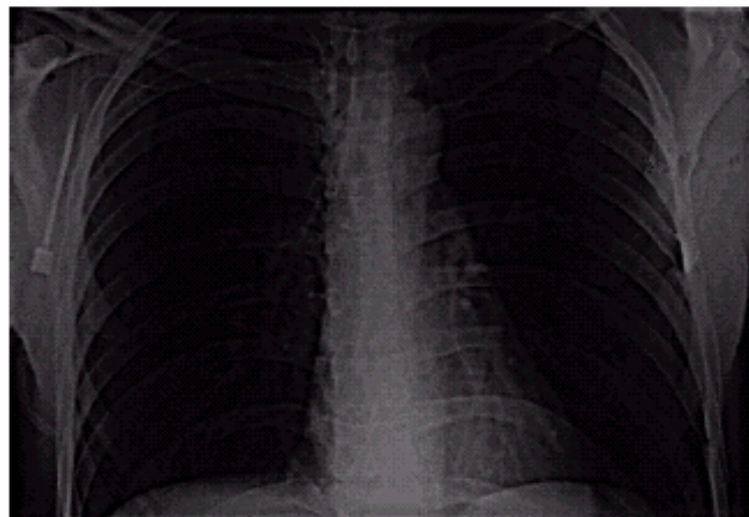
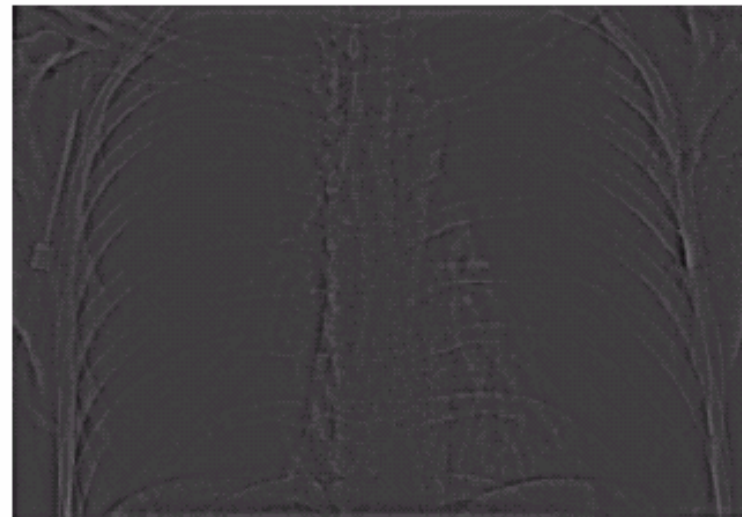
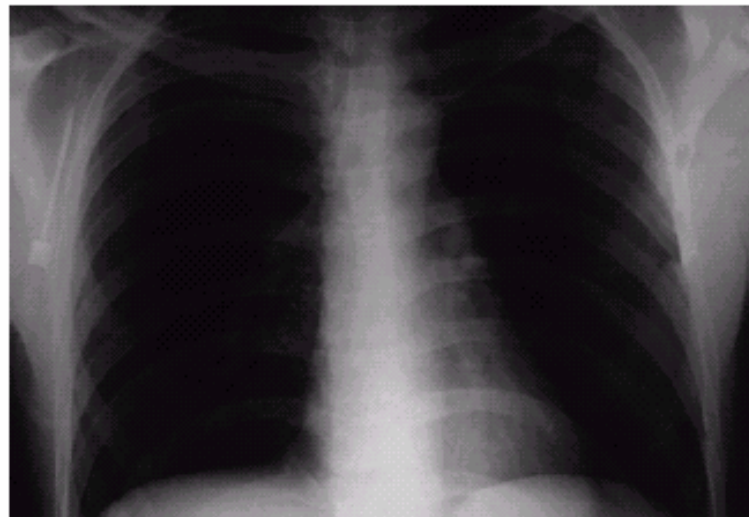
(d) Same as (c), but with  $A = 2.7$ .

(Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)





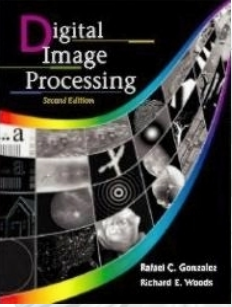
## 4.4.5 Unsharp masking High-boost filtering and High frequency emphasis filtering



a	b
c	d

**FIGURE 4.30**

(a) A chest X-ray image. (b) Result of Butterworth highpass filtering. (c) Result of high-frequency emphasis filtering. (d) Result of performing histogram equalization on (c). (Original image courtesy Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)



## 4.5 Homomorphic Filtering

- Improve image by simultaneous ***gray-level range compression*** and ***contrast enhancement***

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

where  $i(x, y)$  = illumination,  $r(x, y)$  = reflectance

- $i(x, y)$  and  $r(x, y)$  are not separable
- Define  $z(x, y) = \ln \{f(x, y)\} = \ln \{i(x, y)\} + \ln \{r(x, y)\}$
- $\mathcal{F}\{z(x, y)\} = \mathcal{F}\{\ln f(x, y)\} = \mathcal{F}\{\ln i(x, y)\} + \mathcal{F}\{\ln r(x, y)\}$

$$\text{or } Z(u, v) = F_i(u, v) + F_r(u, v)$$

- Applying filter  $H(u, v)$  :

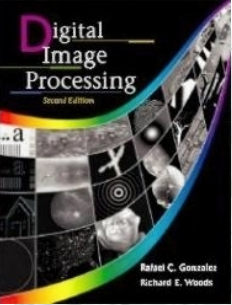
$$S(u, v) = H(u, v)Z(u, v)$$

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

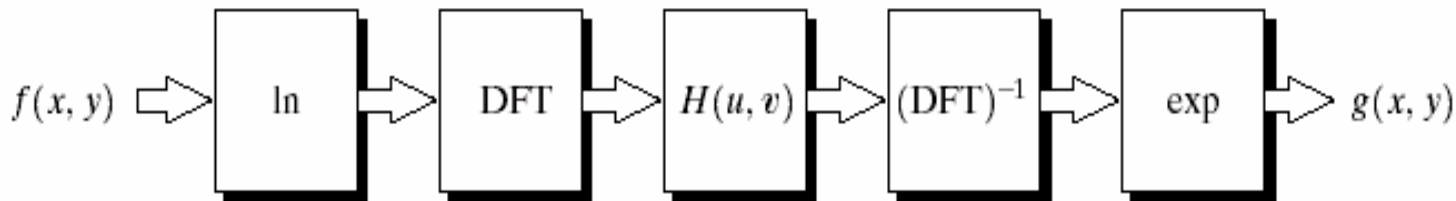


## 4.5 Homomorphic Filtering

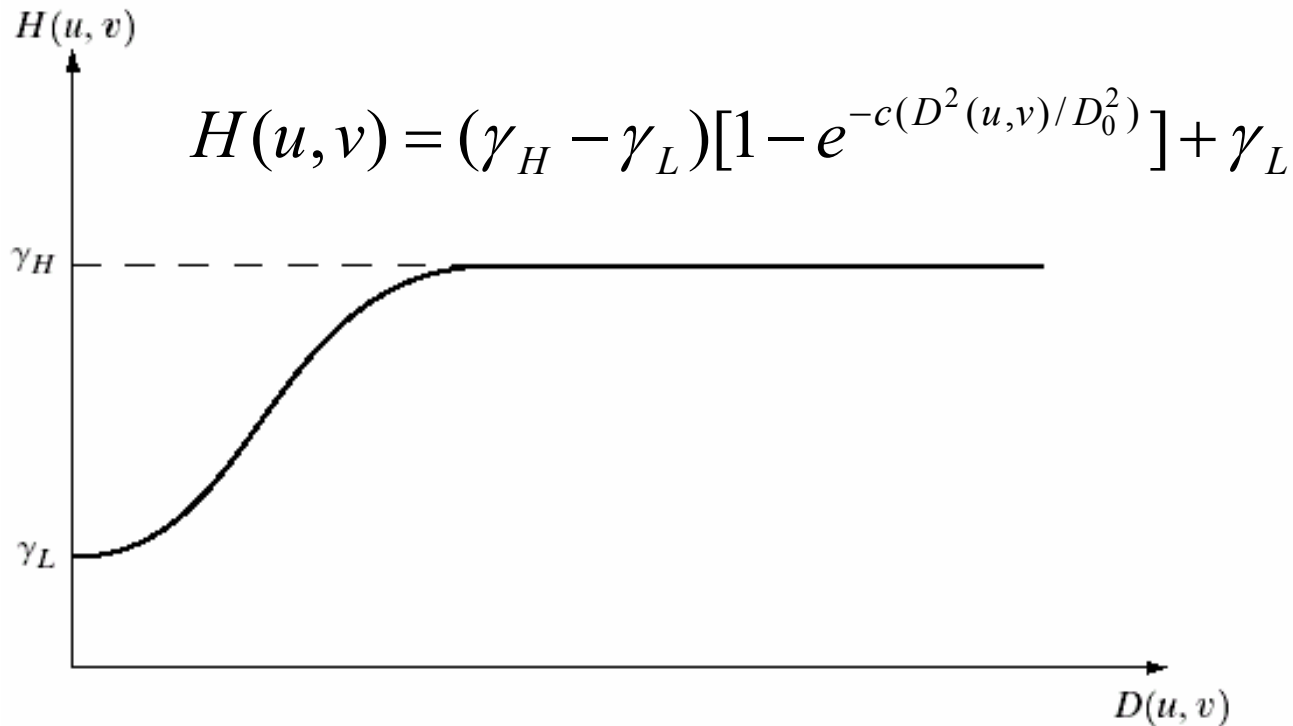
- $$S(x, y) = \mathcal{F}^{-1}\{S(u, v)\}$$
$$= \mathcal{F}^{-1}\{H(u, v)F_i(u, v)\} + \mathcal{F}^{-1}\{H(u, v)F_r(u, v)\}$$
$$= i'(x, y) + r'(x, y)$$
- $$g(x, y) = e^{s(x, y)} = e^{i'(x, y)} \cdot e^{r'(x, y)} = i_0(x, y)r_0(x, y)$$
- The illumination component is characterized by slow spatial variation
- The reflectance component tends to vary abruptly especially at the junction of dissimilar objects.
- **Homomorphic filter**  $H(u, v)$  affects the low and high frequency components in a different ways.



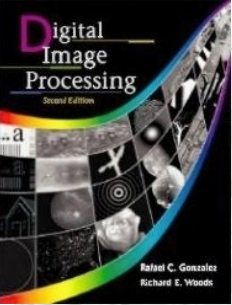
## 4.5 Homomorphic Filtering



**FIGURE 4.31**  
Homomorphic filtering approach for image enhancement.



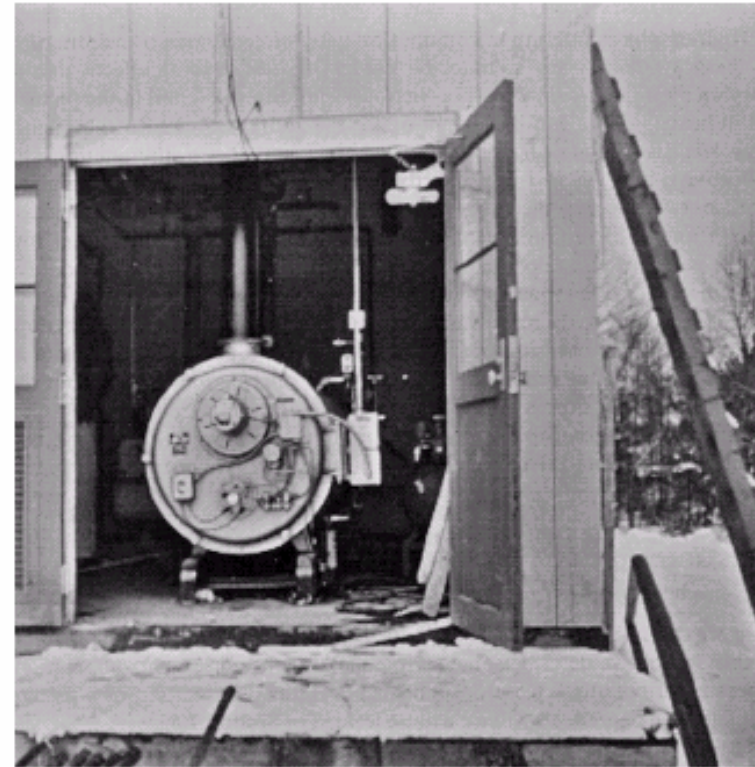
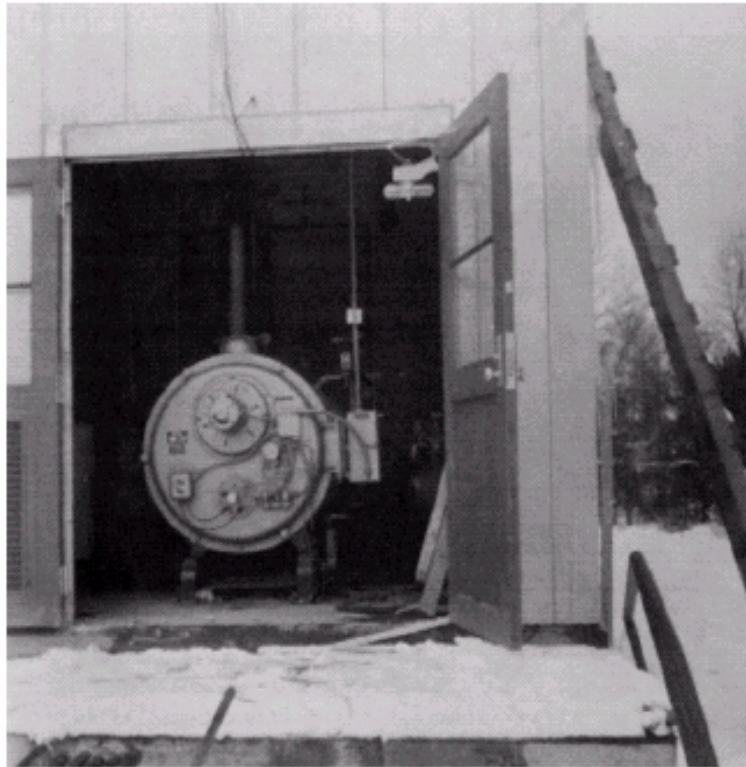
**FIGURE 4.32**  
Cross section of a circularly symmetric filter function.  $D(u, v)$  is the distance from the origin of the centered transform.



## 4.5 Homomorphic Filtering

a b

**FIGURE 4.33**  
(a) Original image. (b) Image processed by homomorphic filtering (note details inside shelter). (Stockham.)







## 4.6 Implementation

- Translation

$$f(x, y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$$

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0/M+vy_0/N)}$$

- Distributivity

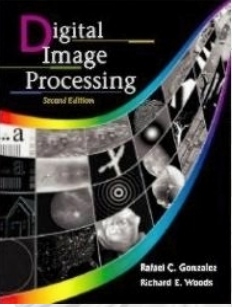
$$F[f_1(x, y) + f_2(x, y)] = F[f_1(x, y)] + F[f_2(x, y)]$$

- Scaling

$$af(x, y) \Leftrightarrow aF(u, v) \text{ and } f(ax, by) \Leftrightarrow F(u/a, v/b)/|ab|$$

- Rotation (in polar coordinate,  $f(x, y) \rightarrow f(r, \theta)$ ,  $F(u, v) \rightarrow F(\omega, \varphi)$ ).

$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$$



## 4.6 Implementation

- Periodicity and conjugate symmetry

$$F(u, v) = F(u+M, v) = F(u, v+N) = F(u+M, v+N).$$

$$f(x, y) = f(x+M, y) = f(x, y+N) = f(x+M, y+N).$$

- *DFT is conjugate symmetry, i.e.*

$$F(u, v) = F^*(-u, -v)$$

$$F(u, v) = |F(-u, -v)|$$

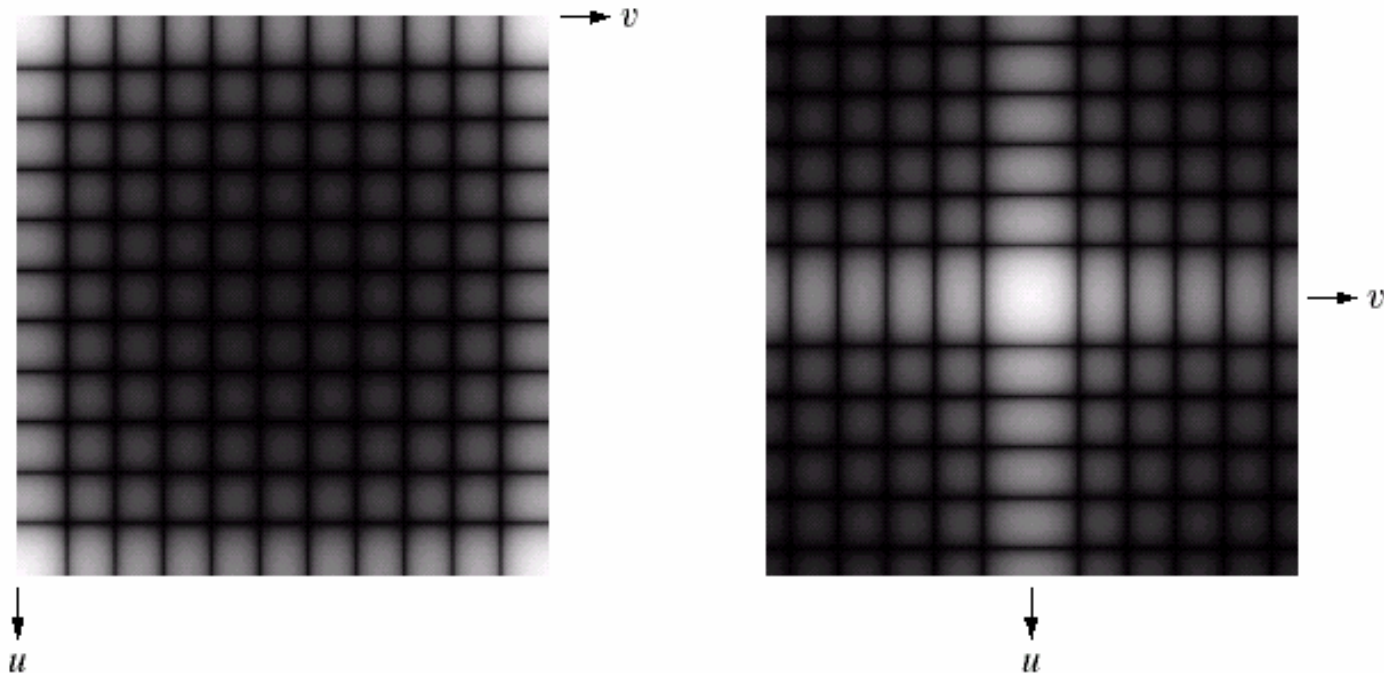
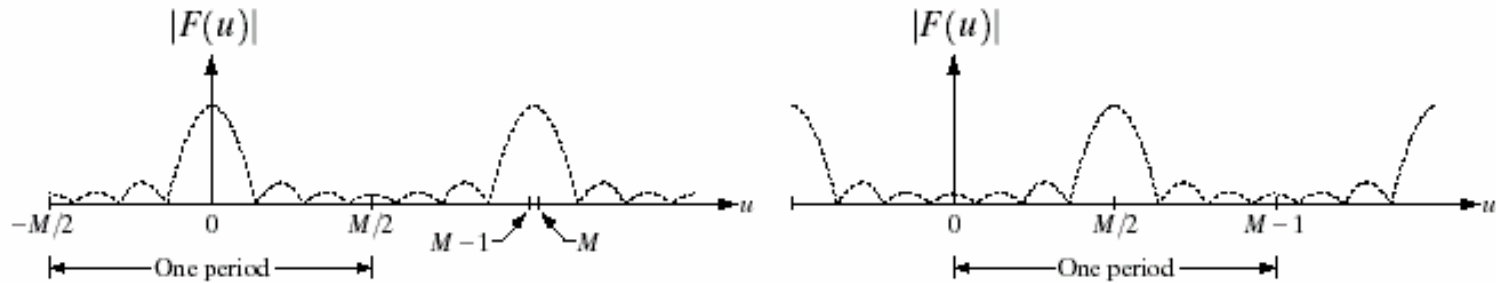
- The values of  $F(u, v)$ ,  $u, v = (M/2)+1 \sim M-1$  are reflection of the values of  $F(u, v)$ ,  $u, v = (M/2)-1 \sim 0$

## 4.6 Implementation

a b  
c d

**FIGURE 4.34**

(a) Fourier spectrum showing back-to-back half periods in the interval  $[0, M - 1]$ .  
 (b) Shifted spectrum showing a full period in the same interval.  
 (c) Fourier spectrum of an image, showing the same back-to-back properties as (a), but in two dimensions.  
 (d) Centered Fourier spectrum.

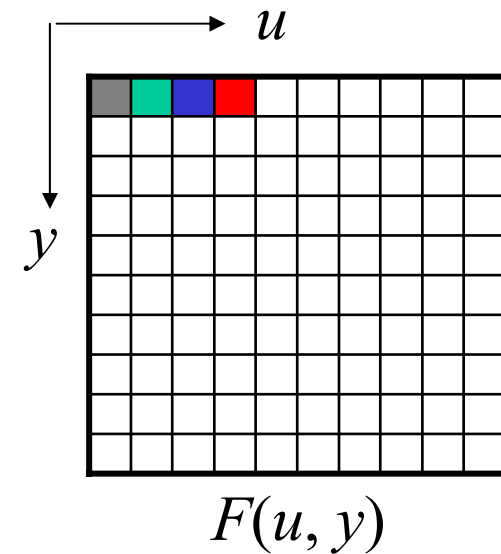
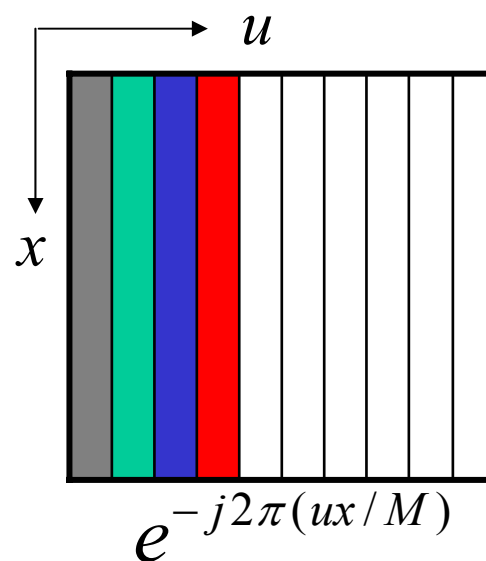
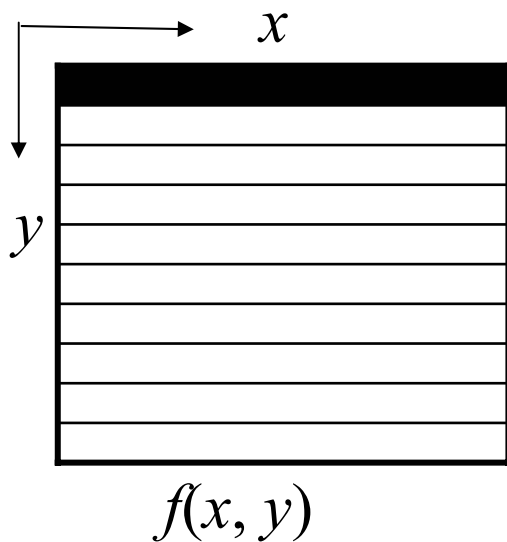


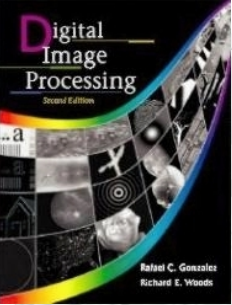
## 4.6 Implementation

- Separability

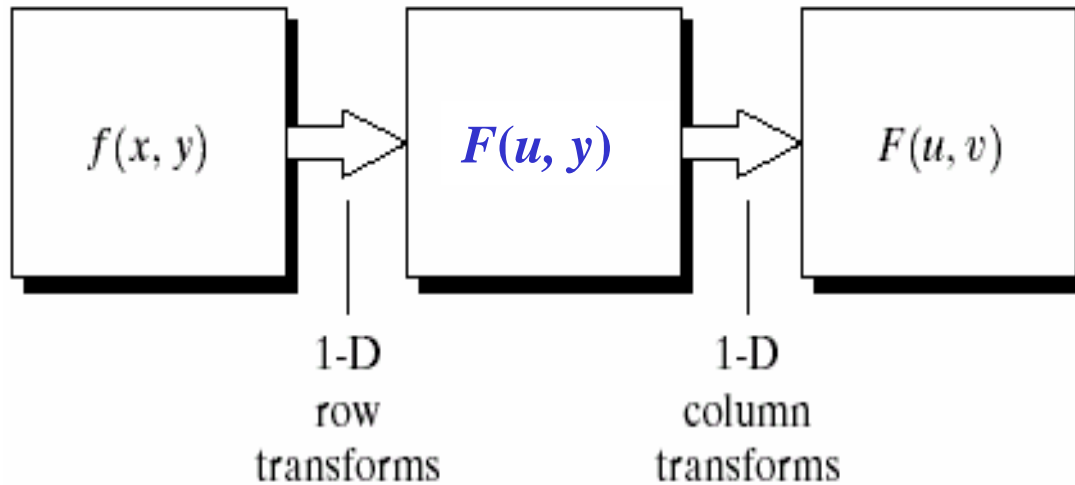
$$F(u, v) = \frac{1}{N} \sum_{y=0}^{N-1} e^{-j2\pi(vy/N)} \frac{1}{M} \sum_{x=0}^{M-1} f(x, y) e^{-j2\pi(ux/M)}$$

$$= \frac{1}{MN} \sum_{y=0}^{N-1} F(u, y) e^{-j2\pi(vy/N)}$$





## 4.6 Implimentation



**FIGURE 4.35**  
Computation of the 2-D Fourier transform as a series of 1-D transforms.



## 4.6 Implementation

- **Computing Inverse DFT using forward DFT**

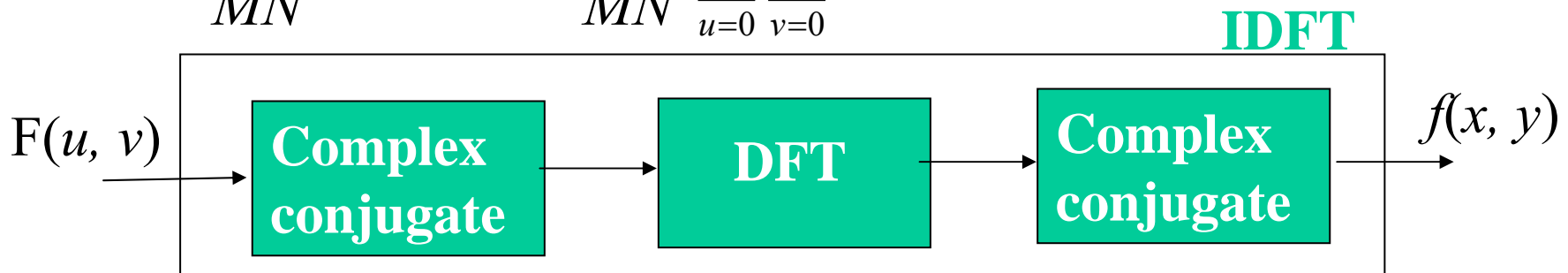
- The DFT and Inverse DFT are

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

- Take complex conjugate on both side and divide MN as

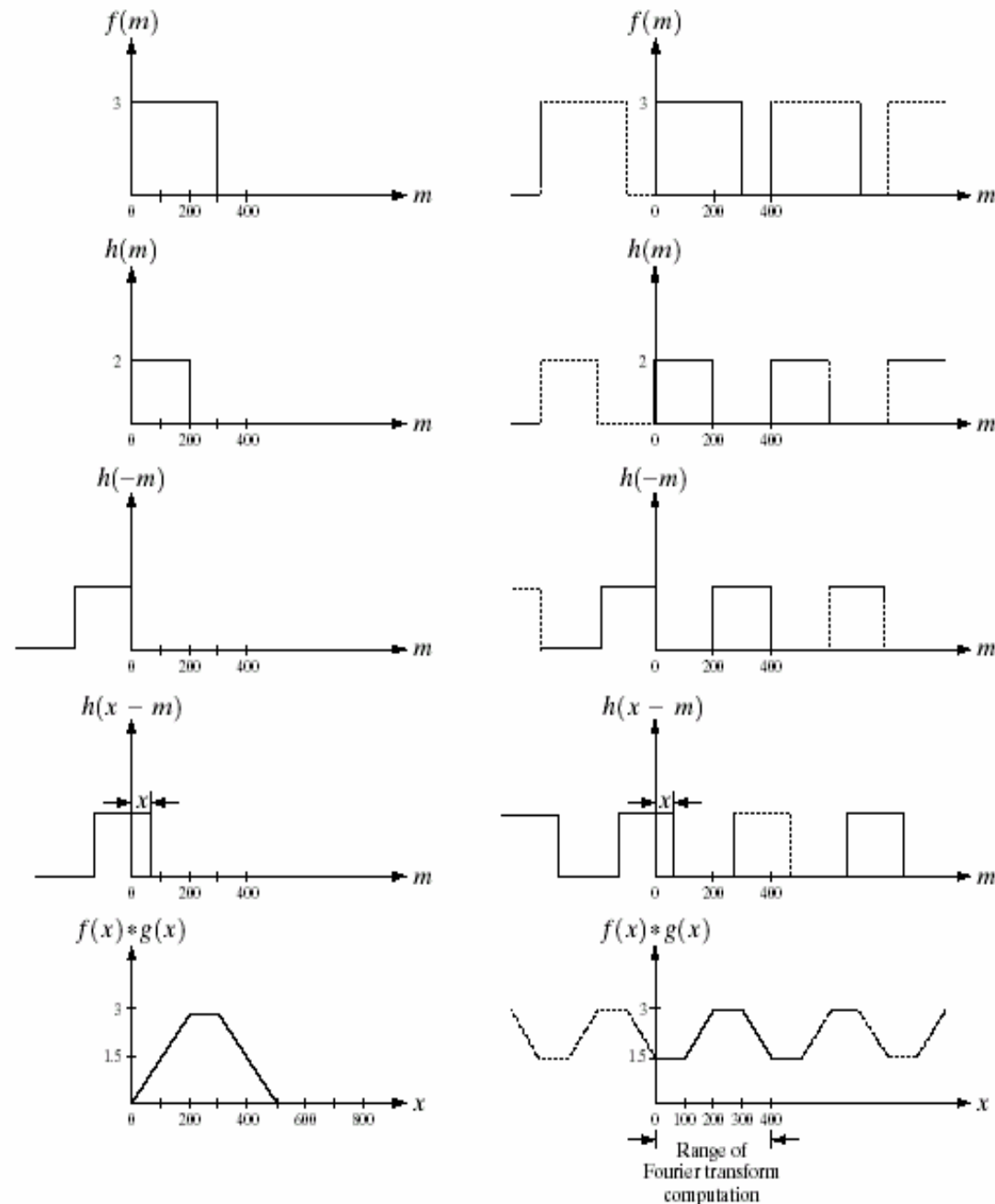
$$\frac{1}{MN} f^*(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi(ux/M + vy/N)}$$





a	f
b	g
c	h
d	i
e	j

**FIGURE 4.36** Left: convolution of two discrete functions. Right: convolution of the same functions, taking into account the implied periodicity of the DFT. Note in (j) how data from adjacent periods corrupt the result of convolution.



## 4.6 Implmentation

The periodic property of DFT

$$f(x) * h(x) = \frac{1}{M} \sum_{m=0}^{M-1} f(m)h(x - m)$$

Range of Fourier transform computation



a  
b  
c  
d  
e

**FIGURE 4.37** Result of performing convolution with extended functions. Compare Figs. 4.37(e) and 4.36(e).

## 4.6 Implimentation

$$f_e(x) = f(x), \quad 0 \leq x \leq A-1$$

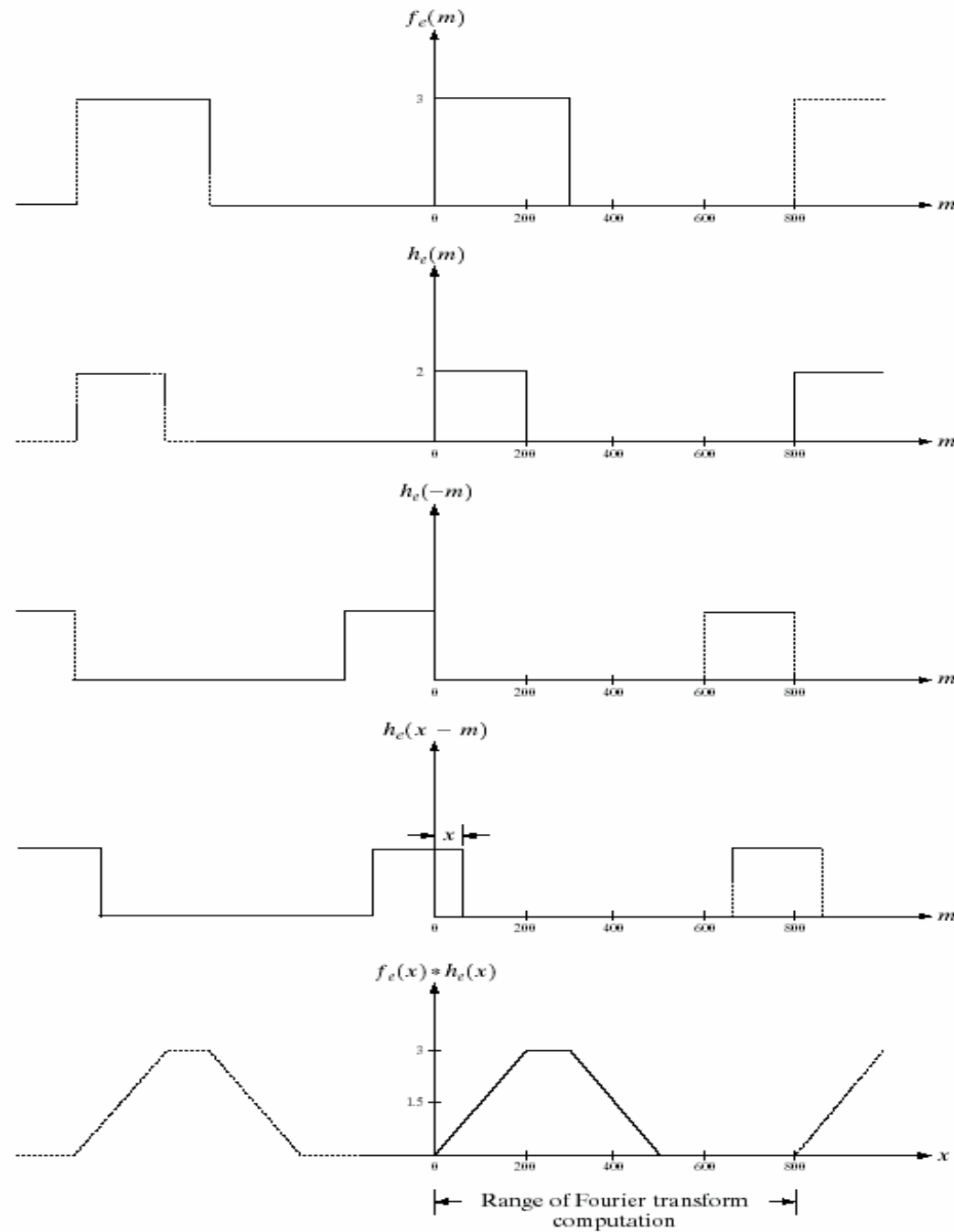
$$f_e(x) = 0, \quad A \leq x \leq P$$

$$h_e(x) = h(x), \quad 0 \leq x \leq B-1$$

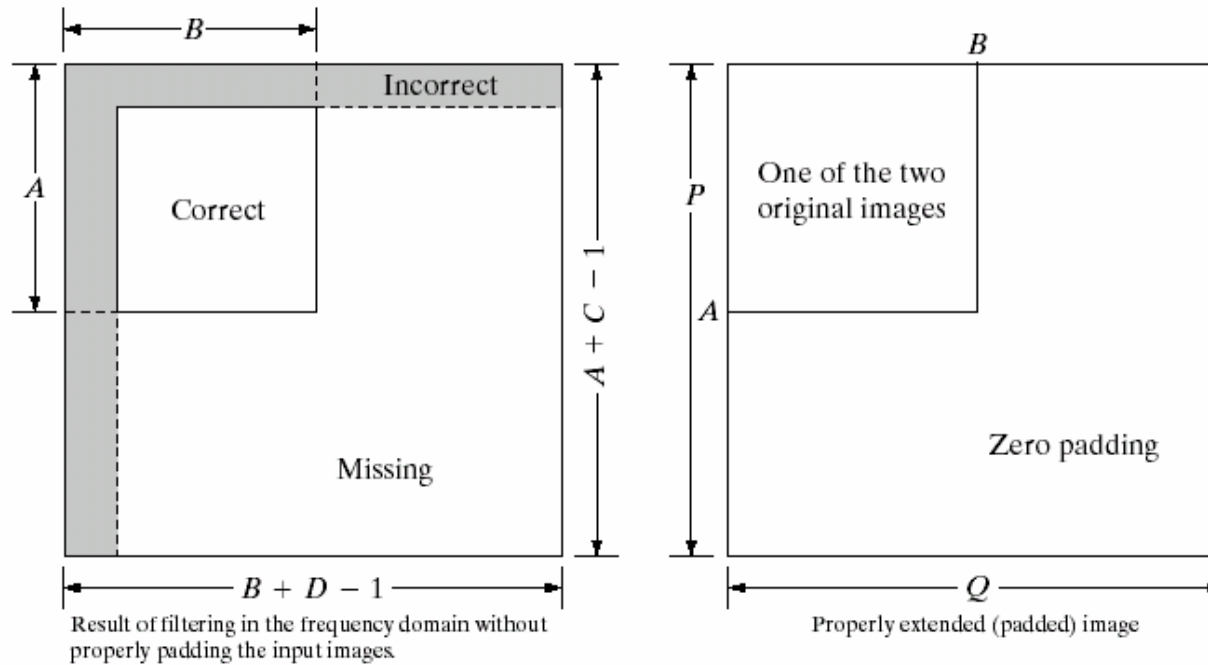
$$h_e(x) = 0, \quad B \leq x \leq P$$

$$f(x) * h(x) \rightarrow f_e(x) * h_e(x)$$

$$\text{length } P = A + B - 1$$



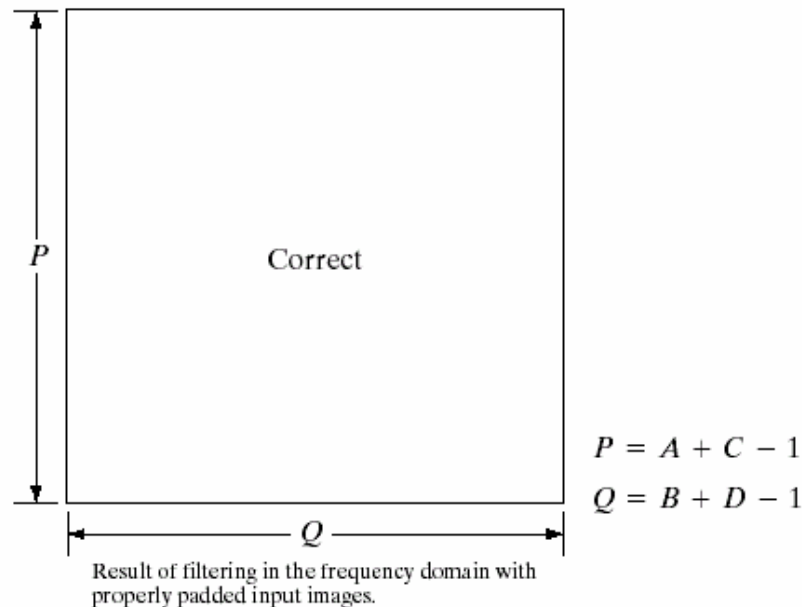


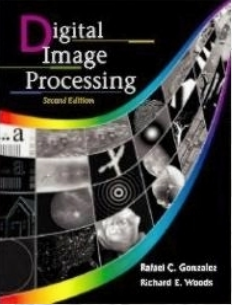


a b  
c

**FIGURE 4.38** Illustration of the need for function padding. (a) Result of performing 2-D convolution without padding. (b) Proper function padding. (c) Correct convolution result.

## 4.6 Implementation



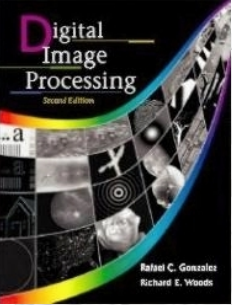


## 4.6 Implementation



**FIGURE 4.39** Padded lowpass filter is the spatial domain (only the real part is shown).

**FIGURE 4.40** Result of filtering with padding. The image is usually cropped to its original size since there is little valuable information past the image boundaries.



## 4.6.4 The Convolution and Correlation Theorems

- **Convolution**

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

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

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

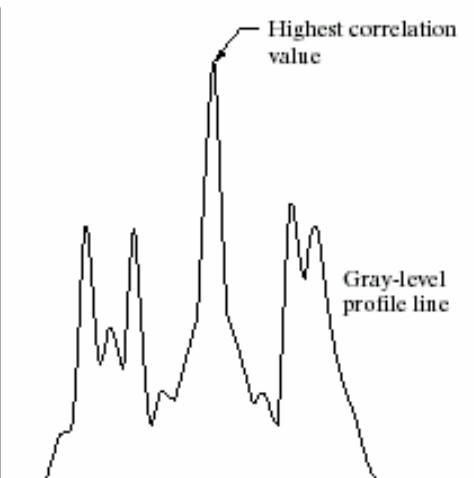
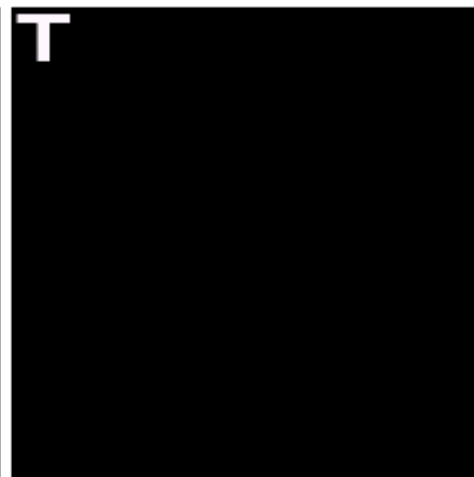
- **Correlation**

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

- $f(x, y) \circledast h(x, y) \Leftrightarrow F^*(u, v) H(u, v)$

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

- **Autocorrelation:**  $f(x, y) f(x, y) \Leftrightarrow |F(u, v)|^2$



a	b
c	d
e	f

**FIGURE 4.41**  
 (a) Image.  
 (b) Template.  
 (c) and  
 (d) Padded  
 images.  
 (e) Correlation  
 function displayed  
 as an image.  
 (f) Horizontal  
 profile line  
 through the  
 highest value in  
 (e), showing the  
 point at which the  
 best match took  
 place.

## 4.6.4 The Correlation

## 4.6 Implementation

**TABLE 4.1**  
Summary of some important properties of the 2-D Fourier transform.

Property	Expression(s)
Fourier transform	$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$
Inverse Fourier transform	$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$
Polar representation	$F(u, v) =  F(u, v)  e^{-j\phi(u, v)}$
Spectrum	$ F(u, v)  = [R^2(u, v) + I^2(u, v)]^{1/2}, \quad R = \text{Real}(F) \text{ and } I = \text{Imag}(F)$
Phase angle	$\phi(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right]$
Power spectrum	$P(u, v) =  F(u, v) ^2$
Average value	$\bar{f}(x, y) = F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$
Translation	$f(x, y) e^{j2\pi(u_0 x/M + v_0 y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v) e^{-j2\pi(ux_0/M + vy_0/N)}$ <p>When <math>x_0 = u_0 = M/2</math> and <math>y_0 = v_0 = N/2</math>, then</p> $f(x, y) (-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v) (-1)^{u+v}$

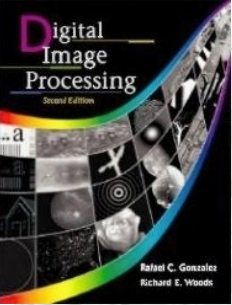


## 4.6 Implementation

Conjugate symmetry	$F(u, v) = F^*(-u, -v)$ $ F(u, v)  =  F(-u, -v) $	
Differentiation	$\frac{\partial^n f(x, y)}{\partial x^n} \Leftrightarrow (ju)^n F(u, v)$ $(-jx)^n f(x, y) \Leftrightarrow \frac{\partial^n F(u, v)}{\partial u^n}$	<p><b>TABLE 4.1</b> (continued)</p>
Laplacian	$\nabla^2 f(x, y) \Leftrightarrow -(u^2 + v^2)F(u, v)$	
Distributivity	$\mathfrak{F}[f_1(x, y) + f_2(x, y)] = \mathfrak{F}[f_1(x, y)] + \mathfrak{F}[f_2(x, y)]$ $\mathfrak{F}[f_1(x, y) \cdot f_2(x, y)] \neq \mathfrak{F}[f_1(x, y)] \cdot \mathfrak{F}[f_2(x, y)]$	
Scaling	$af(x, y) \Leftrightarrow aF(u, v), f(ax, by) \Leftrightarrow \frac{1}{ ab } F(u/a, v/b)$	
Rotation	$x = r \cos \theta \quad y = r \sin \theta \quad u = \omega \cos \varphi \quad v = \omega \sin \varphi$ $f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$	
Periodicity	$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ $f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$	
Separability	<p>See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.</p>	

## 4.6 Implementation

Property	Expression(s)
Computation of the inverse Fourier transform using a forward transform algorithm	$\frac{1}{MN} f^*(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi(ux/M + vy/N)}$ <p style="text-align: right;"><b>TABLE 4.1</b> (continued)</p> <p>This equation indicates that inputting the function <math>F^*(u, v)</math> into an algorithm designed to compute the forward transform (right side of the preceding equation) yields <math>f^*(x, y)/MN</math>. Taking the complex conjugate and multiplying this result by <math>MN</math> gives the desired inverse.</p>
Convolution <sup>†</sup>	$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$
Correlation <sup>†</sup>	$f(x, y) \circ h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n)$
Convolution theorem <sup>†</sup>	$f(x, y) * h(x, y) \Leftrightarrow F(u, v) H(u, v);$ $f(x, y) h(x, y) \Leftrightarrow F(u, v) * H(u, v)$
Correlation theorem <sup>†</sup>	$f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v) H(u, v);$ $f^*(x, y) h(x, y) \Leftrightarrow F(u, v) \circ H(u, v)$



## 4.6 Implimentation

**TABLE 4.1**  
(continued)

Some useful FT pairs:

*Impulse*  $\delta(x, y) \Leftrightarrow 1$

*Gaussian*  $A\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2(x^2+y^2)} \Leftrightarrow Ae^{-(u^2+v^2)/2\sigma^2}$

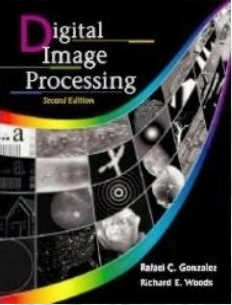
*Rectangle*  $\text{rect}[a, b] \Leftrightarrow ab \frac{\sin(\pi ua)}{(\pi ua)} \frac{\sin(\pi vb)}{(\pi vb)} e^{-j\pi(ua+vb)}$

*Cosine*  $\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$   
 $\frac{1}{2} [\delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0)]$

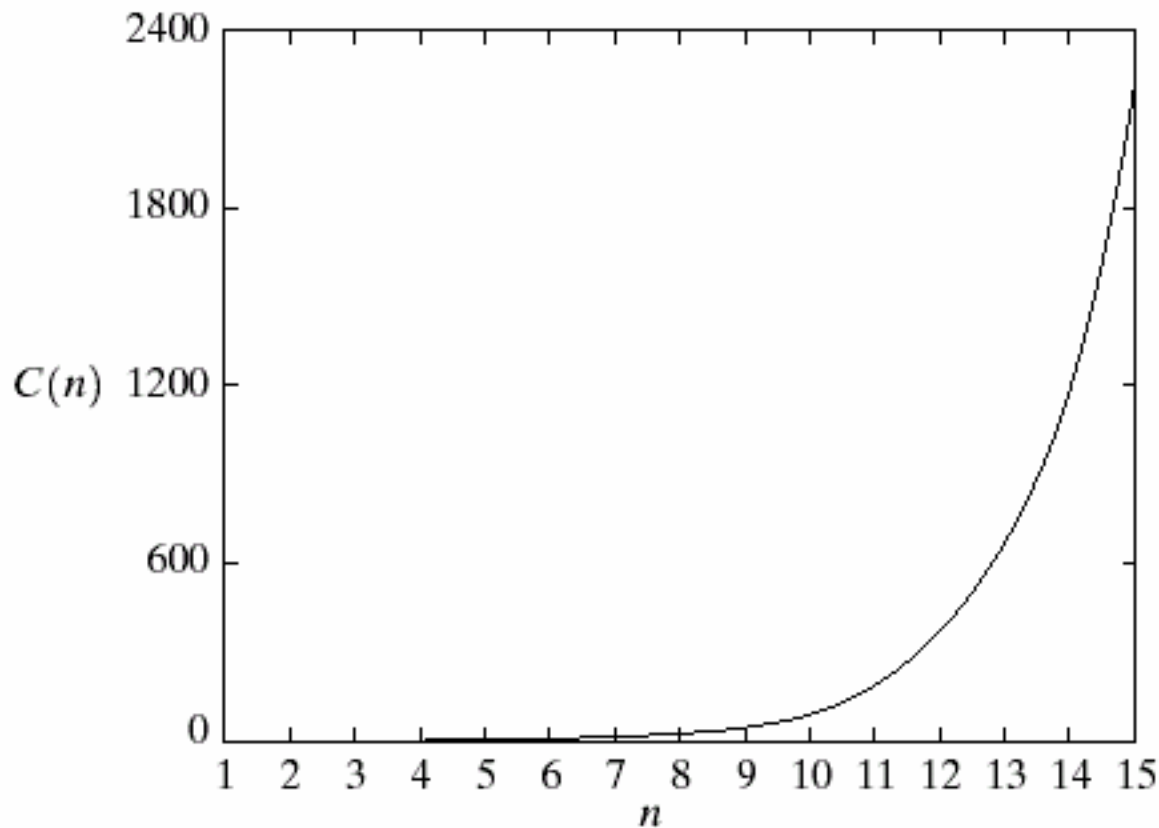
*Sine*  $\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$   
 $j \frac{1}{2} [\delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0)]$

† Assumes that functions have been extended by zero padding.





## 4.6 Implementation – Fast Fourier Transform



**FIGURE 4.42** Computational advantage of the FFT over a direct implementation of the 1-D DFT. Note that the advantage increases rapidly as a function of  $n$ .