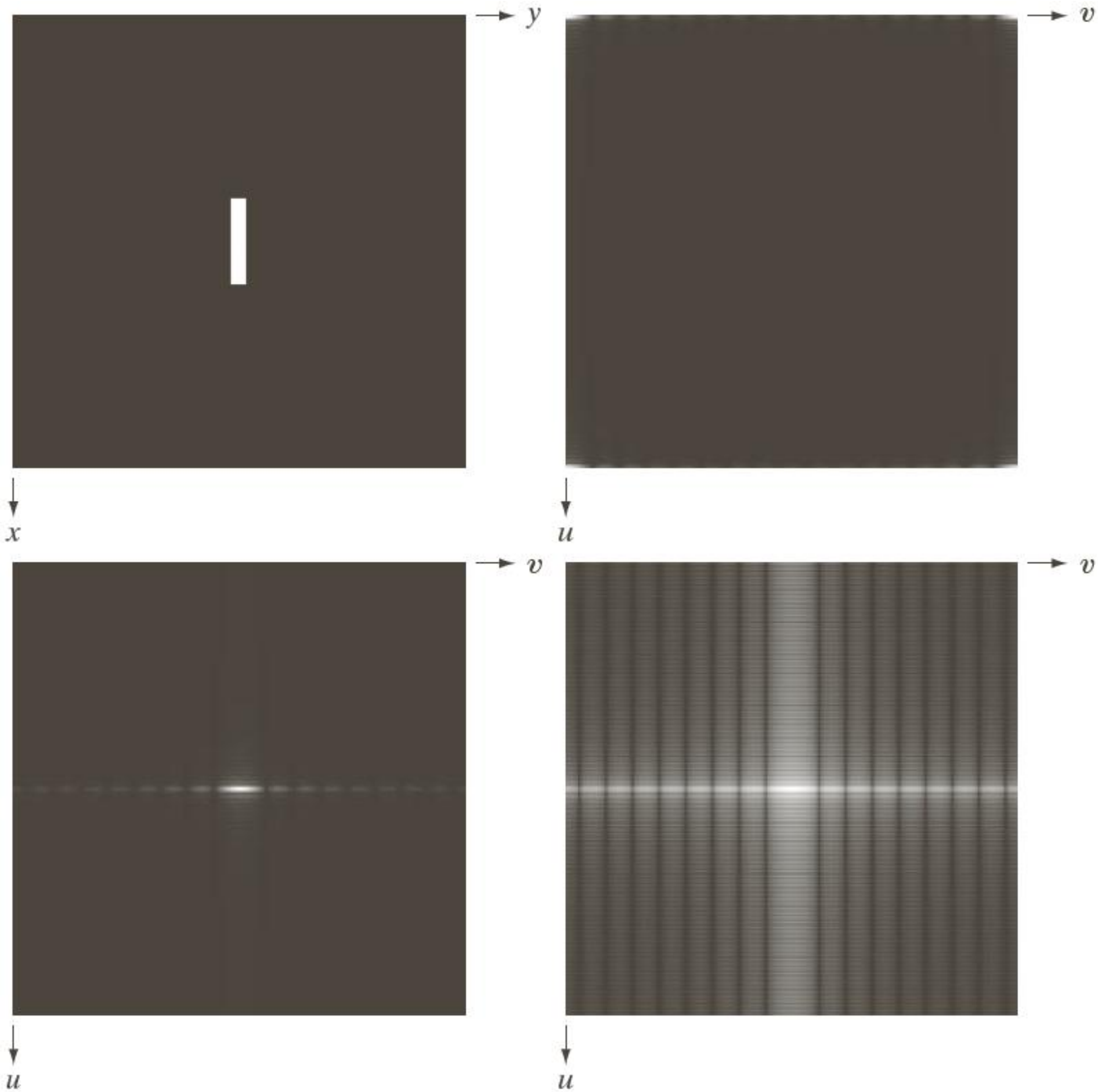
Digital Image Processing

Lecture # 13
Filtering in Frequency Domain

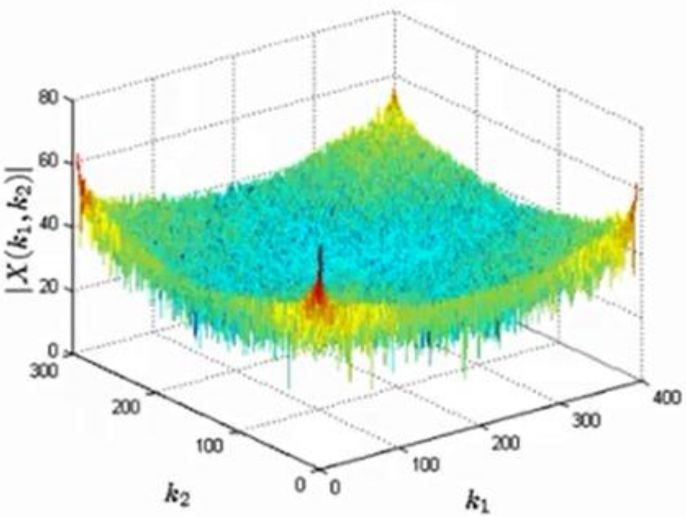
Frequencies in Images



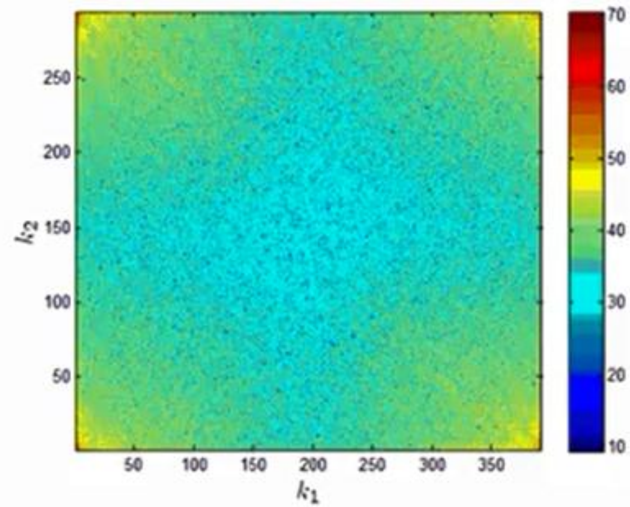
a b
c d

FIGURE 4.24
(a) Image. (b) Spectrum showing bright spots in the four corners. (c) Centered spectrum. (d) Result showing increased detail after a log transformation. The zero crossings of the spectrum are closer in the vertical direction because the rectangle in (a) is longer in that direction. The coordinate convention used throughout the book places the origin of the spatial and frequency domains at the top left.

DFT

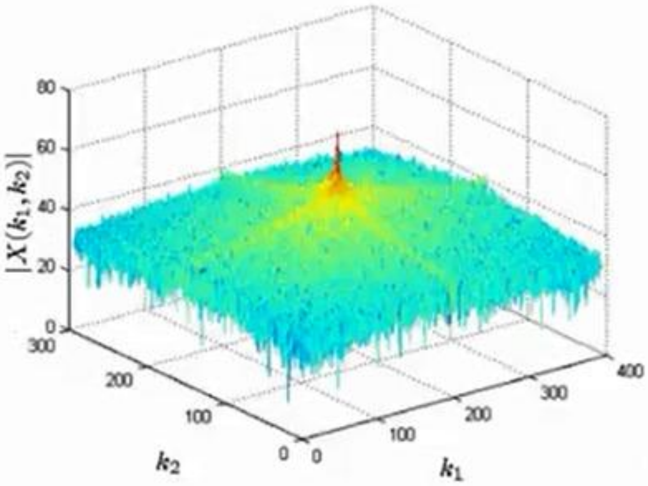


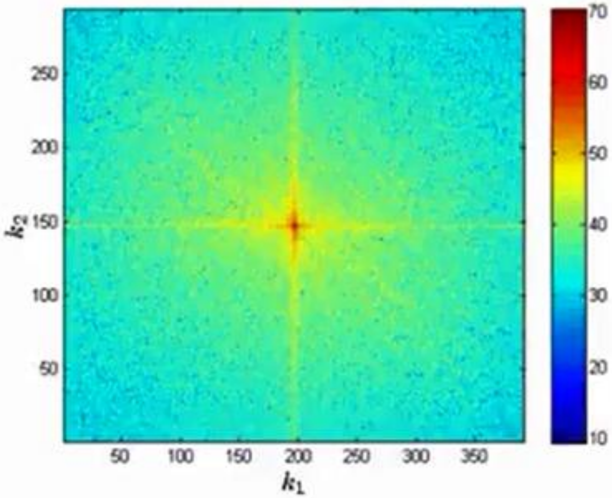
80
60
40
20
0
$|X(k_1, k_2)|$
300
200
100
0
k_2
0
100
200
300
400
k_1



250
200
150
100
50
k_2
50
100
150
200
250
300
350
k_1
70
60
50
40
30
20
10

Centered DFT

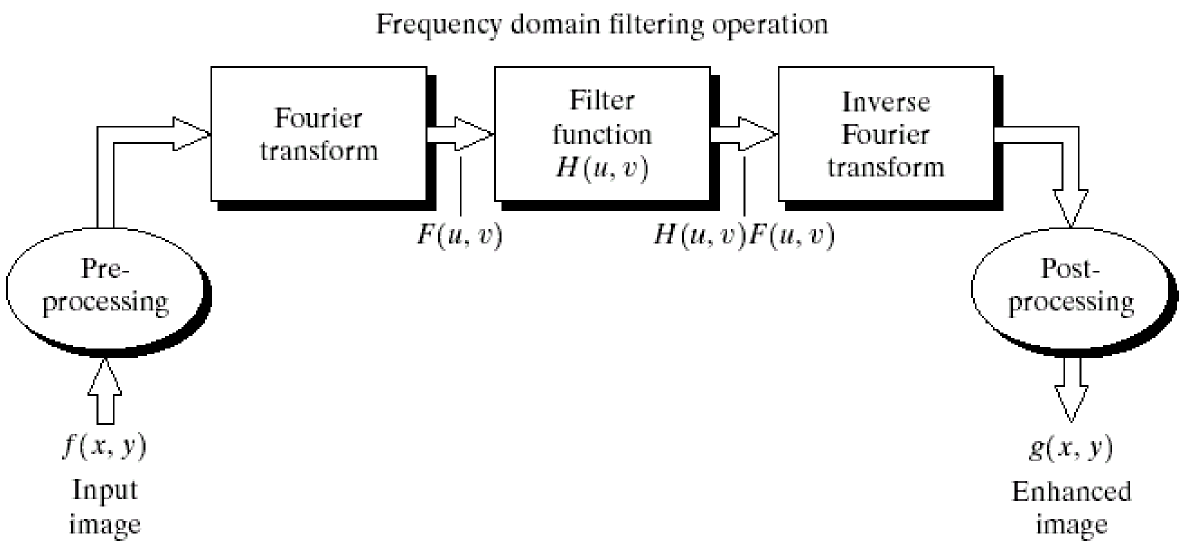




The DFT and Image Processing

To filter an image in the frequency domain:

1. Compute $F(u,v)$ the DFT of the image
2. Multiply $F(u,v)$ by a filter function $H(u,v)$
3. Compute the inverse DFT of the result



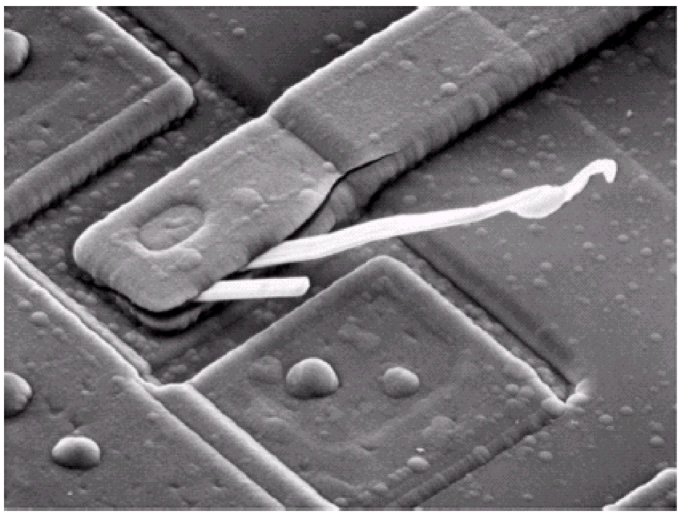
The Discrete Fourier Transform (DFT)

The *Discrete Fourier Transform* of $f(x, y)$, for x = 0, 1, 2…M-1 and y = 0,1,2…N-1, denoted by $F(u, v)$, is given by the equation:

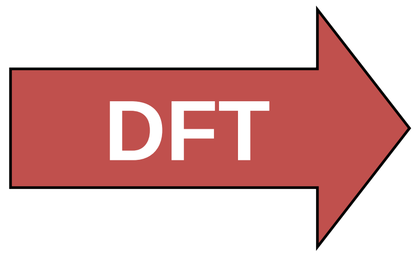
$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

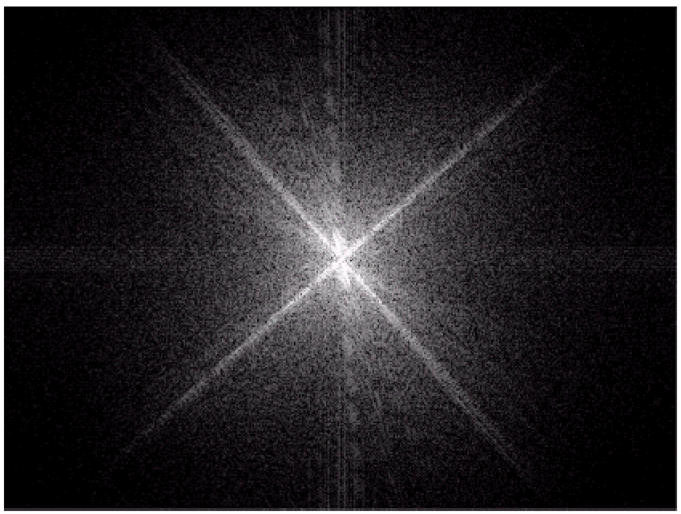
for u = 0, 1, 2…M-1 and v = 0, 1, 2…N-1.

DFT & Images



Scanning electron microscope image of an integrated circuit magnified ~2500 times





Fourier spectrum of the image

The Inverse DFT

It is really important to note that the Fourier transform is completely **reversible**

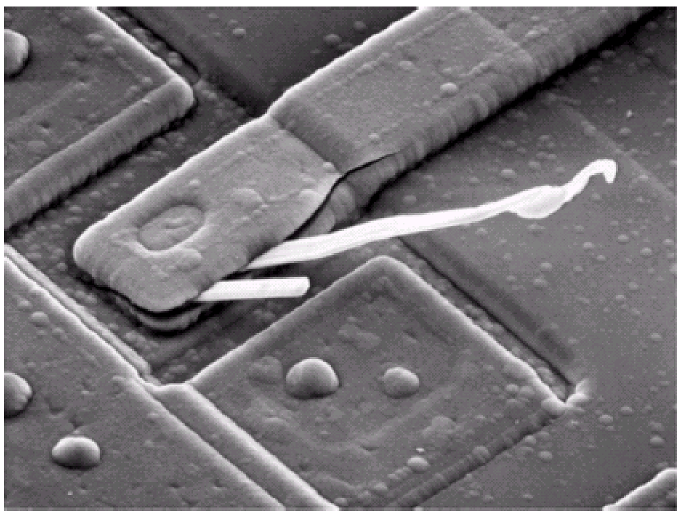
The inverse DFT is given by:

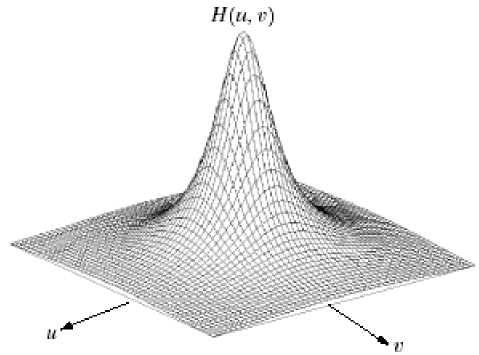
$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$$

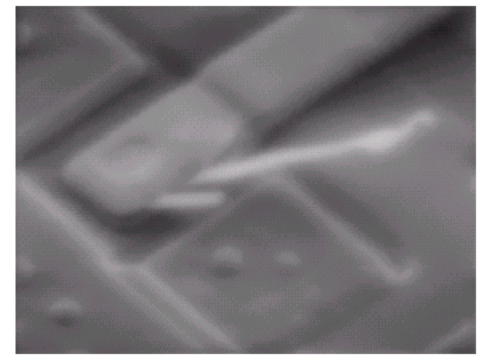
for x = 0, 1, 2…M-1 and y = 0, 1, 2…N-1

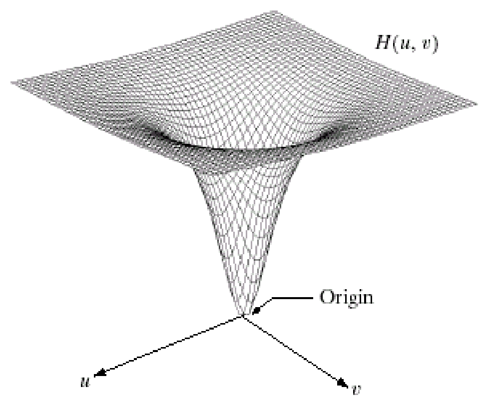
Some Basic Frequency Domain Filters

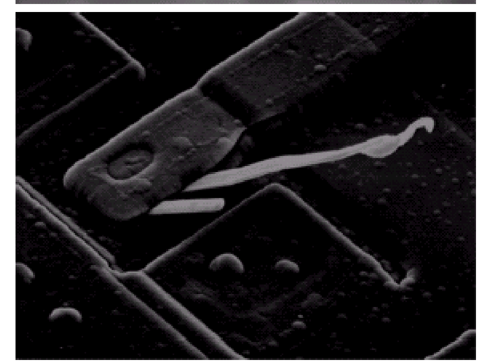
Low Pass Filter







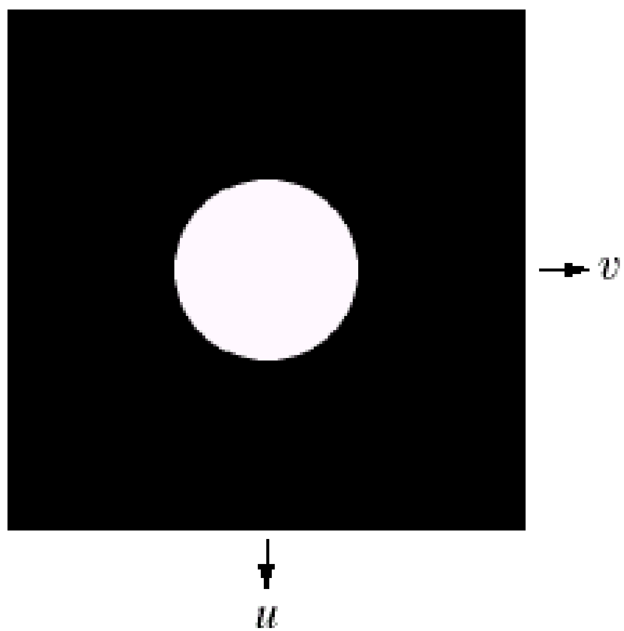


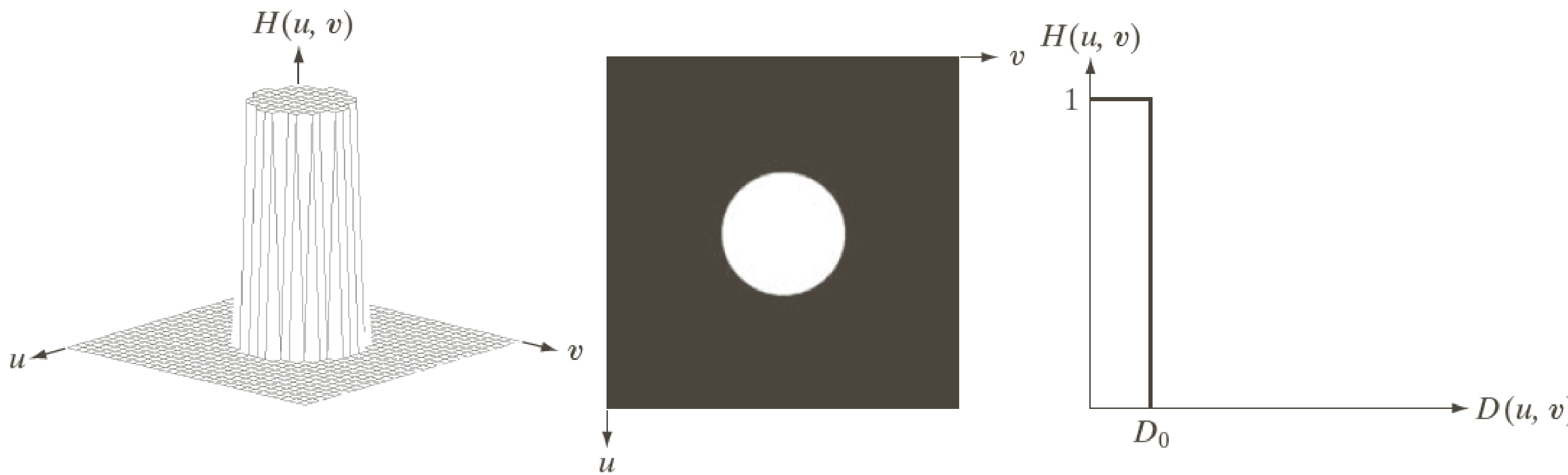


High Pass Filter

Ideal Low Pass Filter

Simply cut off all high frequency components that are a specified distance D_0 from the origin

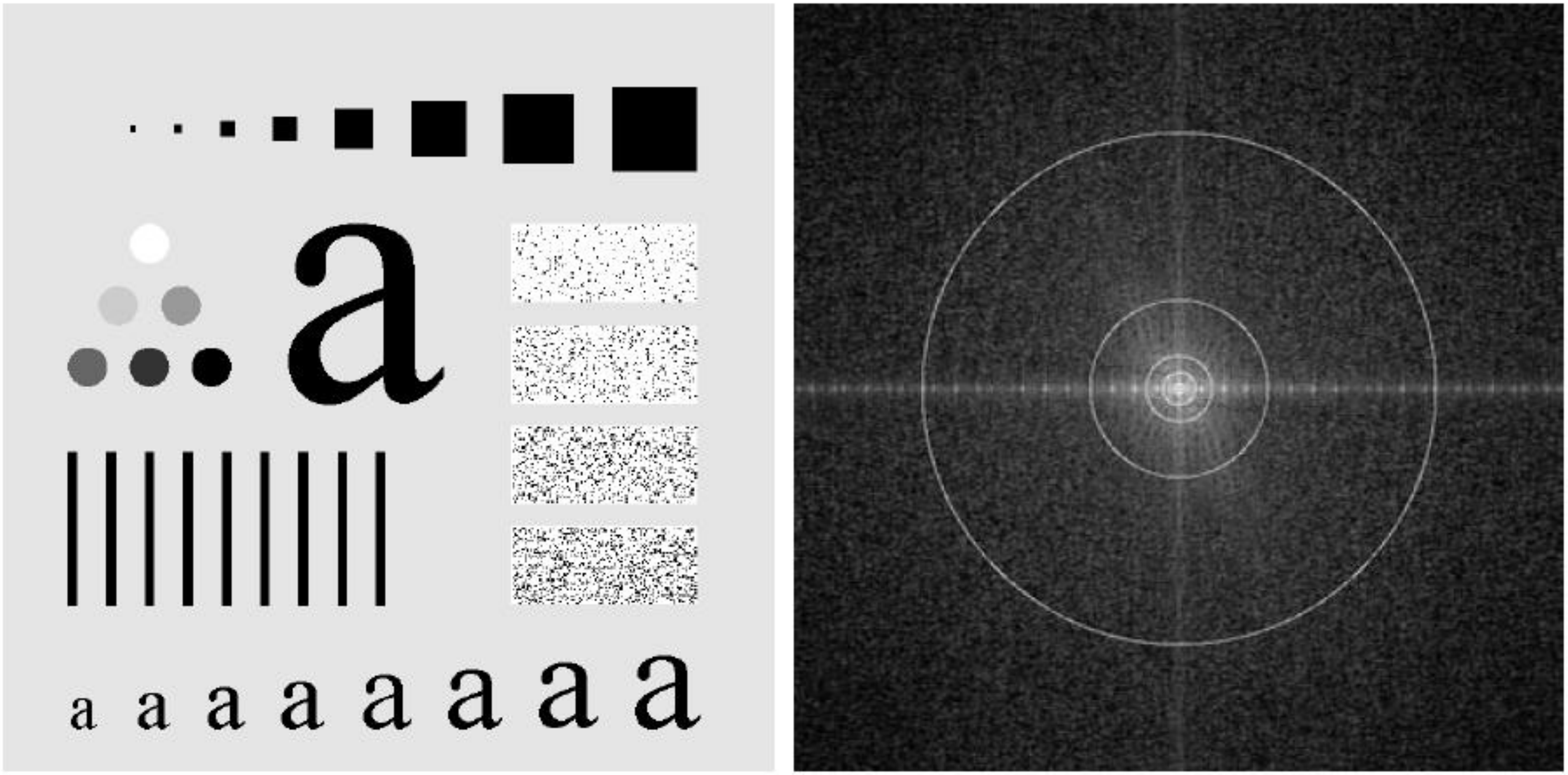




a b c

FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Ideal Low Pass Filter (cont…)



a b

FIGURE 4.41 (a) Test pattern of size 688 × 688 pixels, and (b) its Fourier spectrum. The spectrum is double the image size due to padding but is shown in half size so that it fits in the page. The superimposed circles have radii equal to 10, 30, 60, 160, and 460 with respect to the full-size spectrum image. These radii enclose 87.0, 93.1, 95.7, 97.8, and 99.2% of the padded image power, respectively.

Results of ILPF

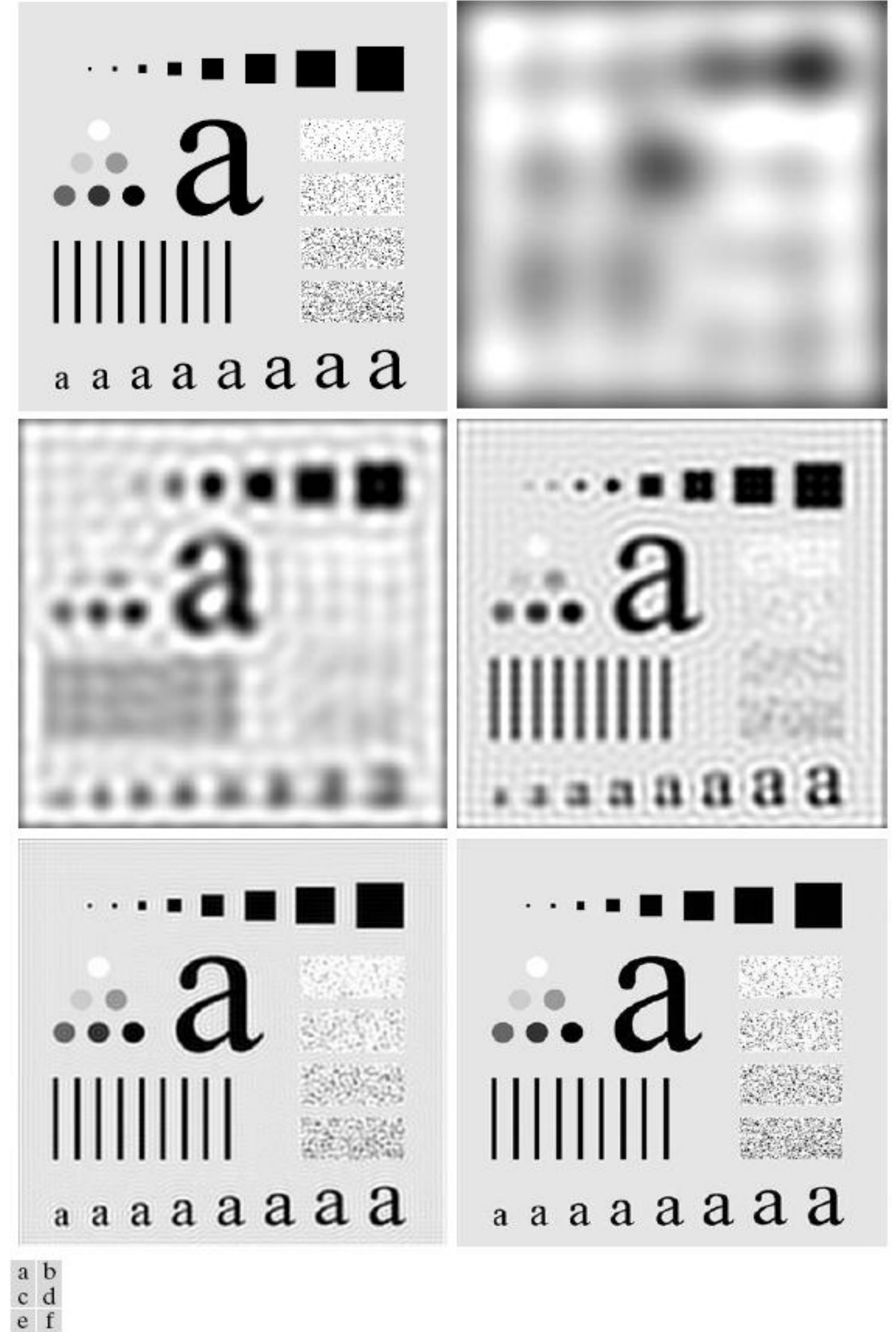
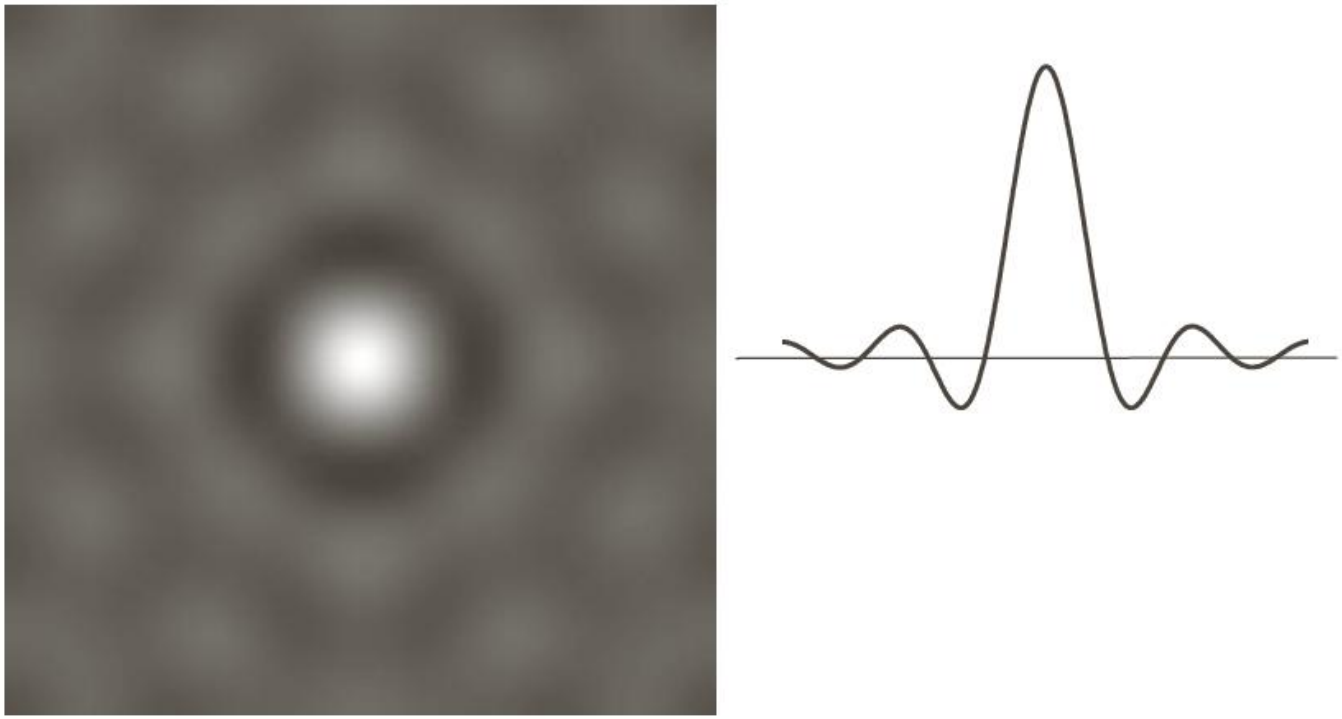


FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.

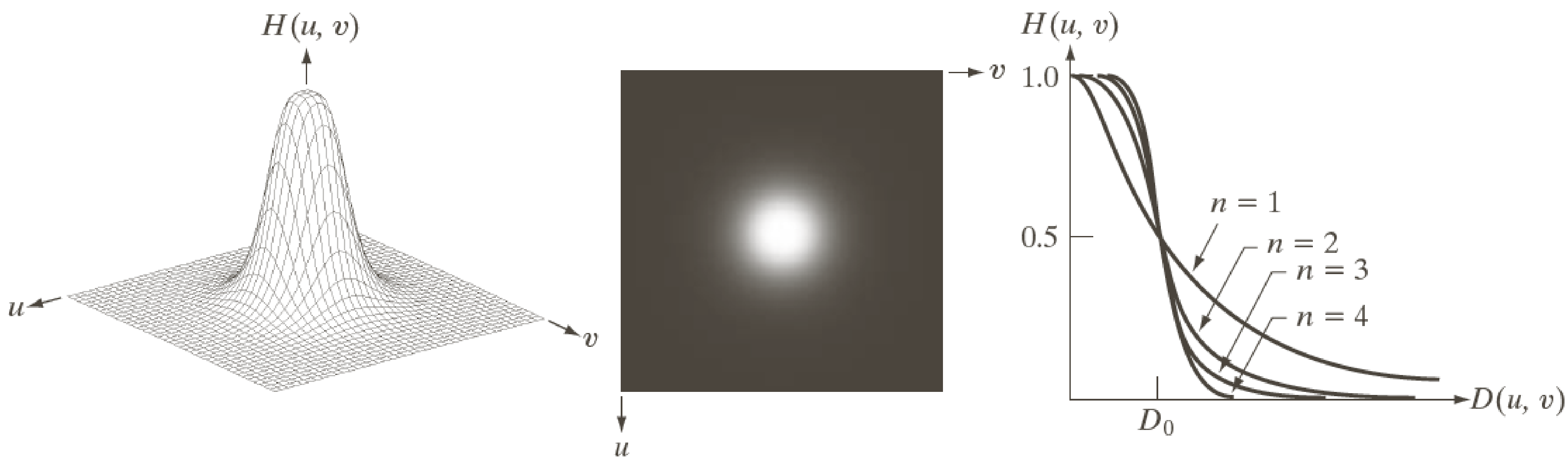
Spatial representation of ILPF



a b

FIGURE 4.43 (a) Representation in the spatial domain of an ILPF of radius 5 and size 1000×1000. (b) Intensity profile of a horizontal line passing through the center of the image.

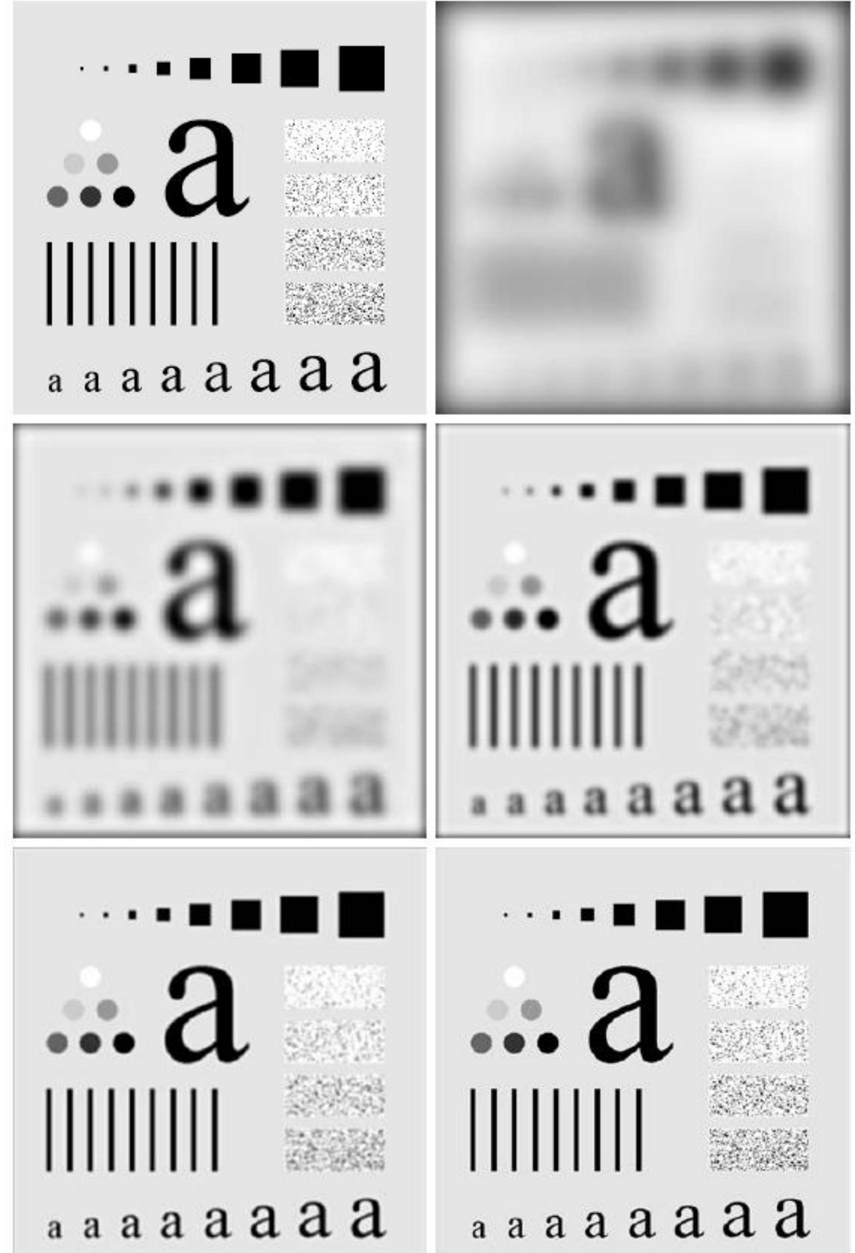
Butterworth LPF



a b c

FIGURE 4.44 (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.

Results of Butterworth LPF



a b
c d
e f

FIGURE 4.45 (a) Original image. (b)–(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Fig. 4.42.

Spatial representation of butterworth LPF

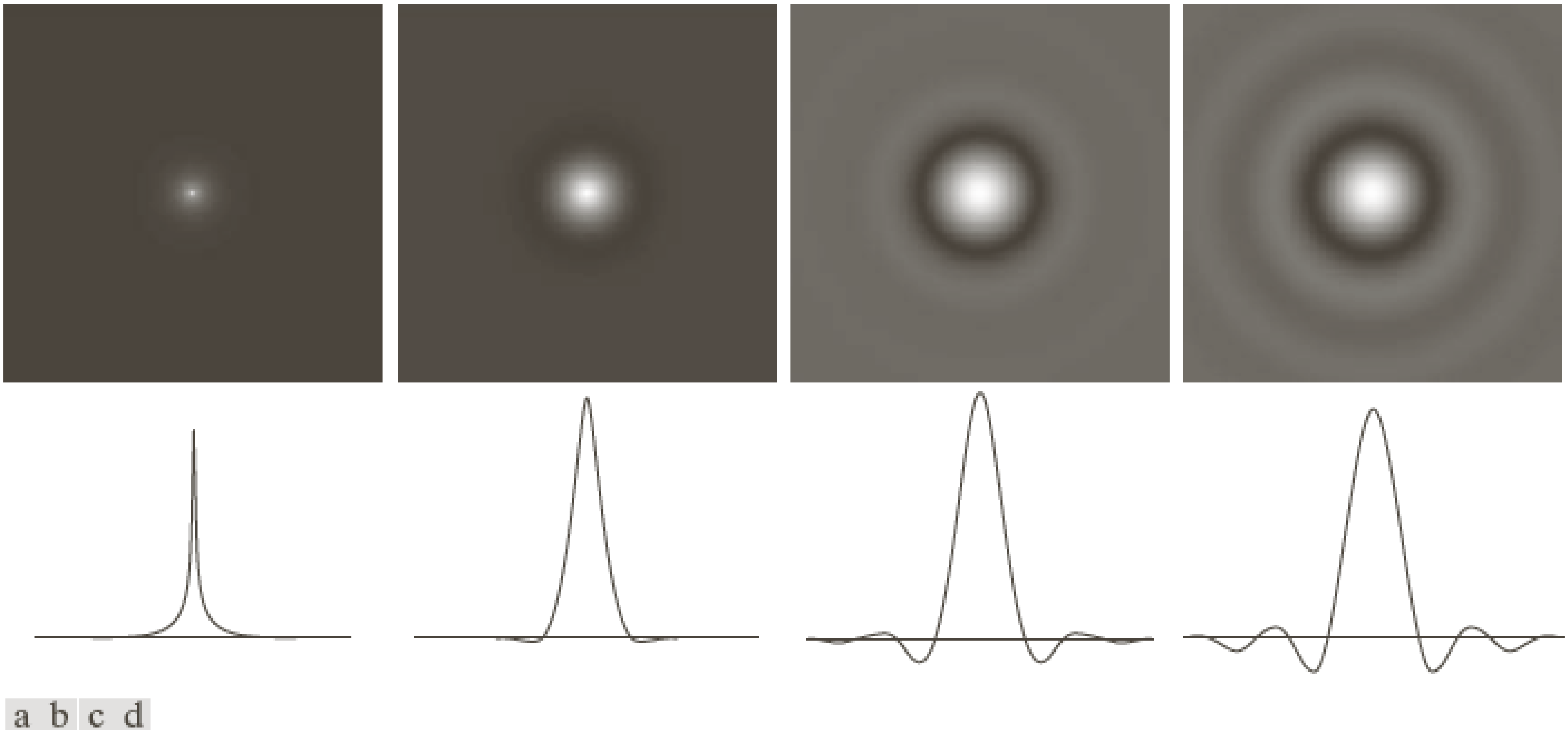
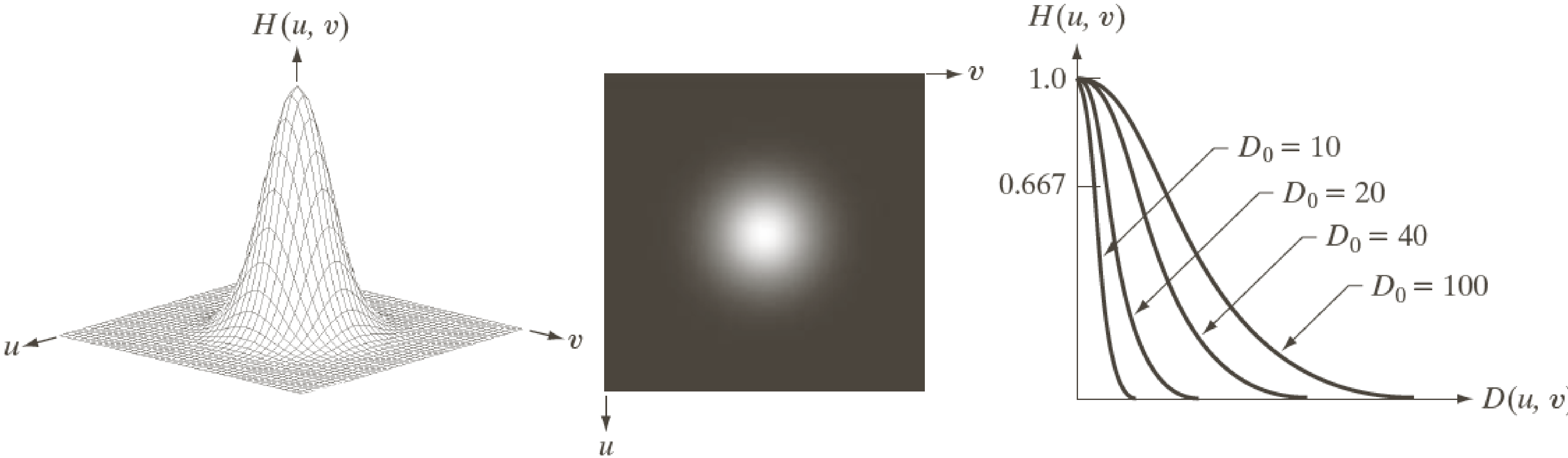


FIGURE 4.46 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding intensity profiles through the center of the filters (the size in all cases is 1000 × 1000 and the cutoff frequency is 5). Observe how ringing increases as a function of filter order.

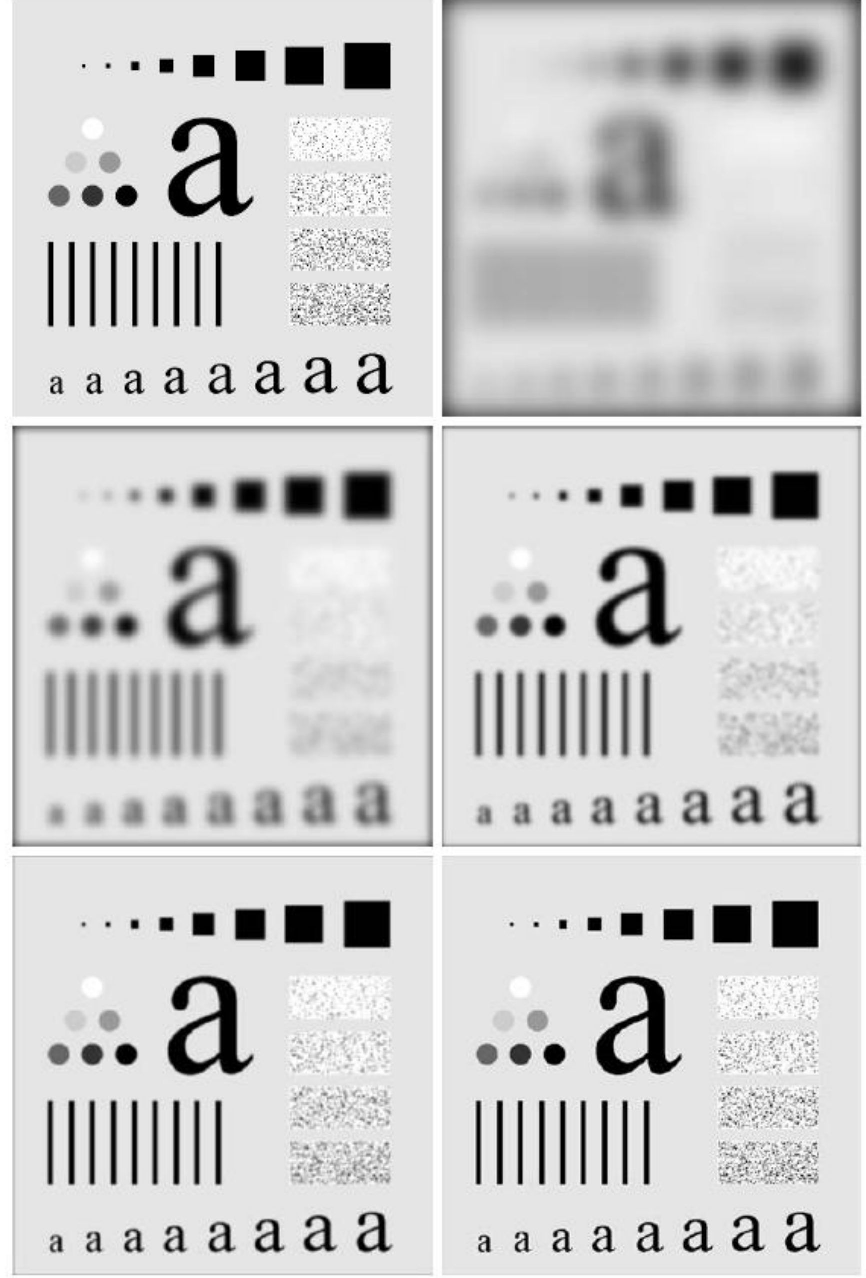
Gaussian LPF



a b c

FIGURE 4.47 (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.

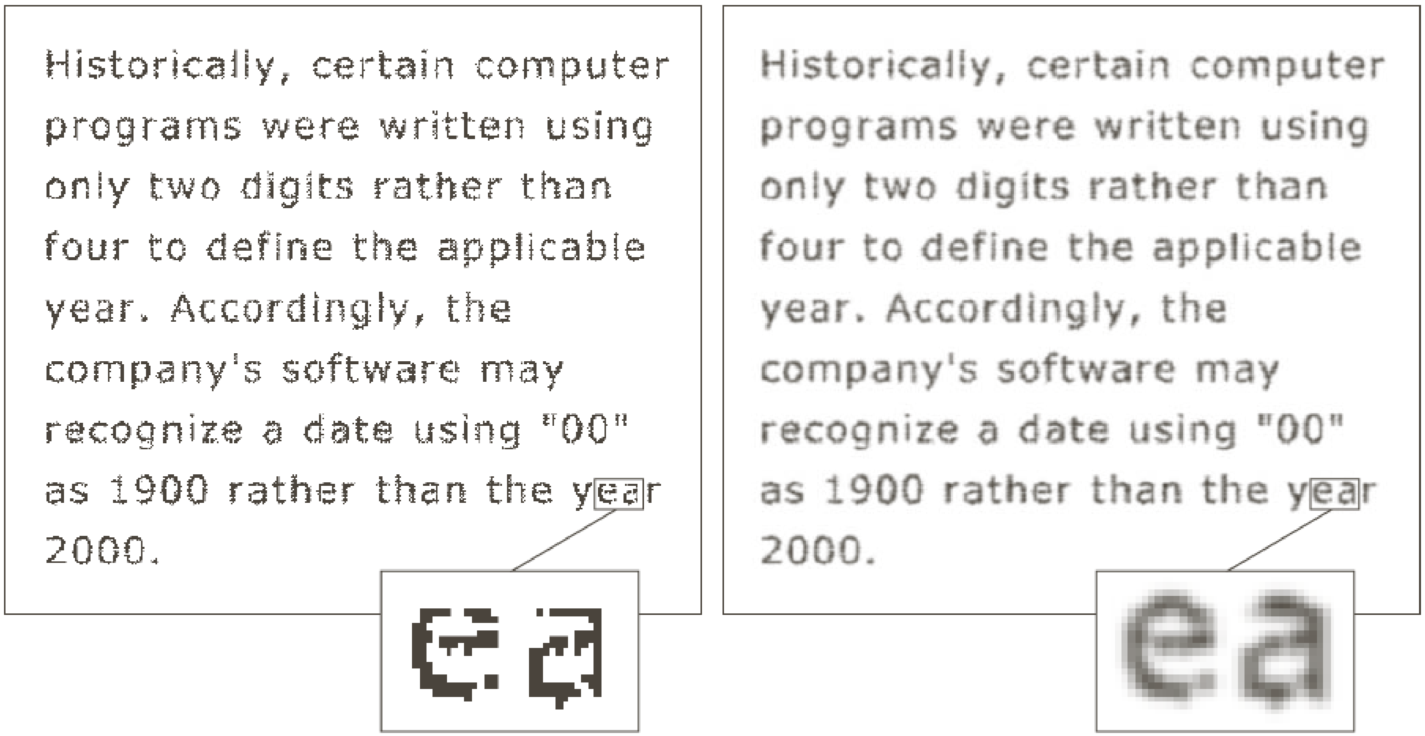
Results of Gaussian LPF



a	b
c	d
e	f

FIGURE 4.48 (a) Original image. (b)–(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Figs. 4.42 and 4.45.

Applications of LPFs



a b

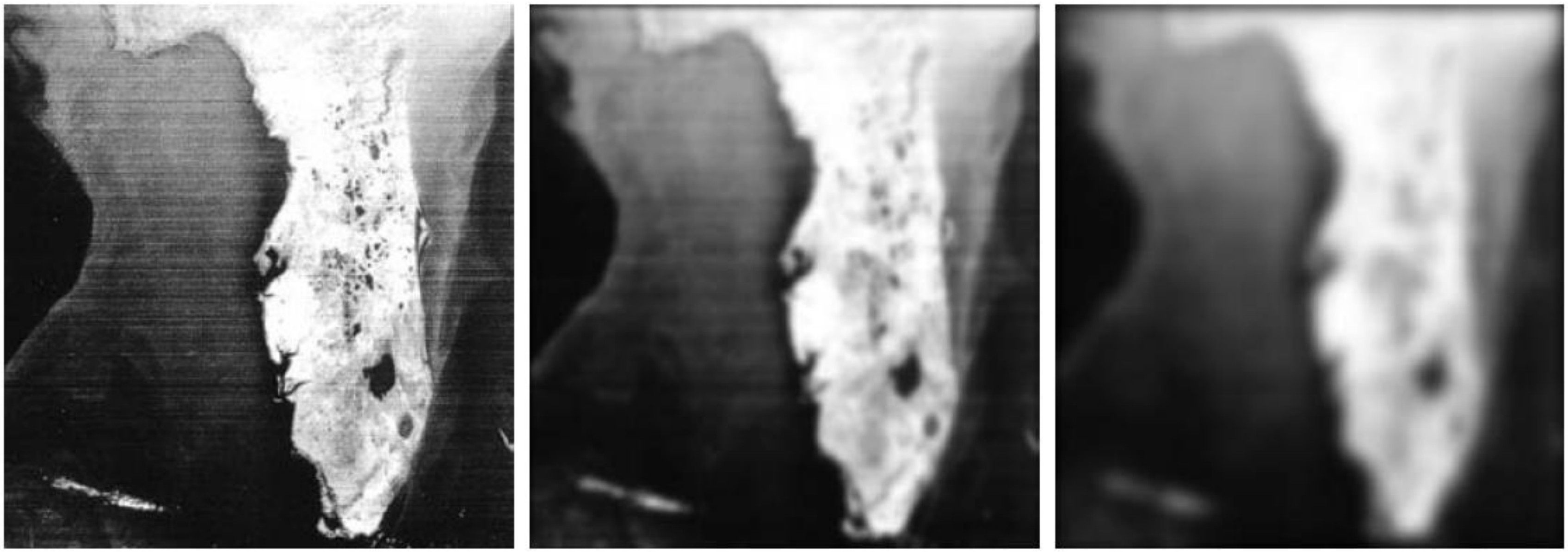
FIGURE 4.49
(a) Sample text of low resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).

Applications of LPFs

a b c

FIGURE 4.50 (a) Original image (784 × 732 pixels). (b) Result of filtering using a GLPF with $D_0 = 100$. (c) Result of filtering using a GLPF with $D_0 = 80$. Note the reduction in fine skin lines in the magnified sections in (b) and (c).

Applications of LPFs



a b c

FIGURE 4.51 (a) Image showing prominent horizontal scan lines. (b) Result of filtering using a GLPF with $D_0 = 50$. (c) Result of using a GLPF with $D_0 = 20$. (Original image courtesy of NOAA.)

LPF Summary

TABLE 4.4
Lowpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D(u,v)/D_0]^{2n}}$	$H(u,v) = e^{-D^2(u,v)/2D_0^2}$

Highpass Filter (HPFs)

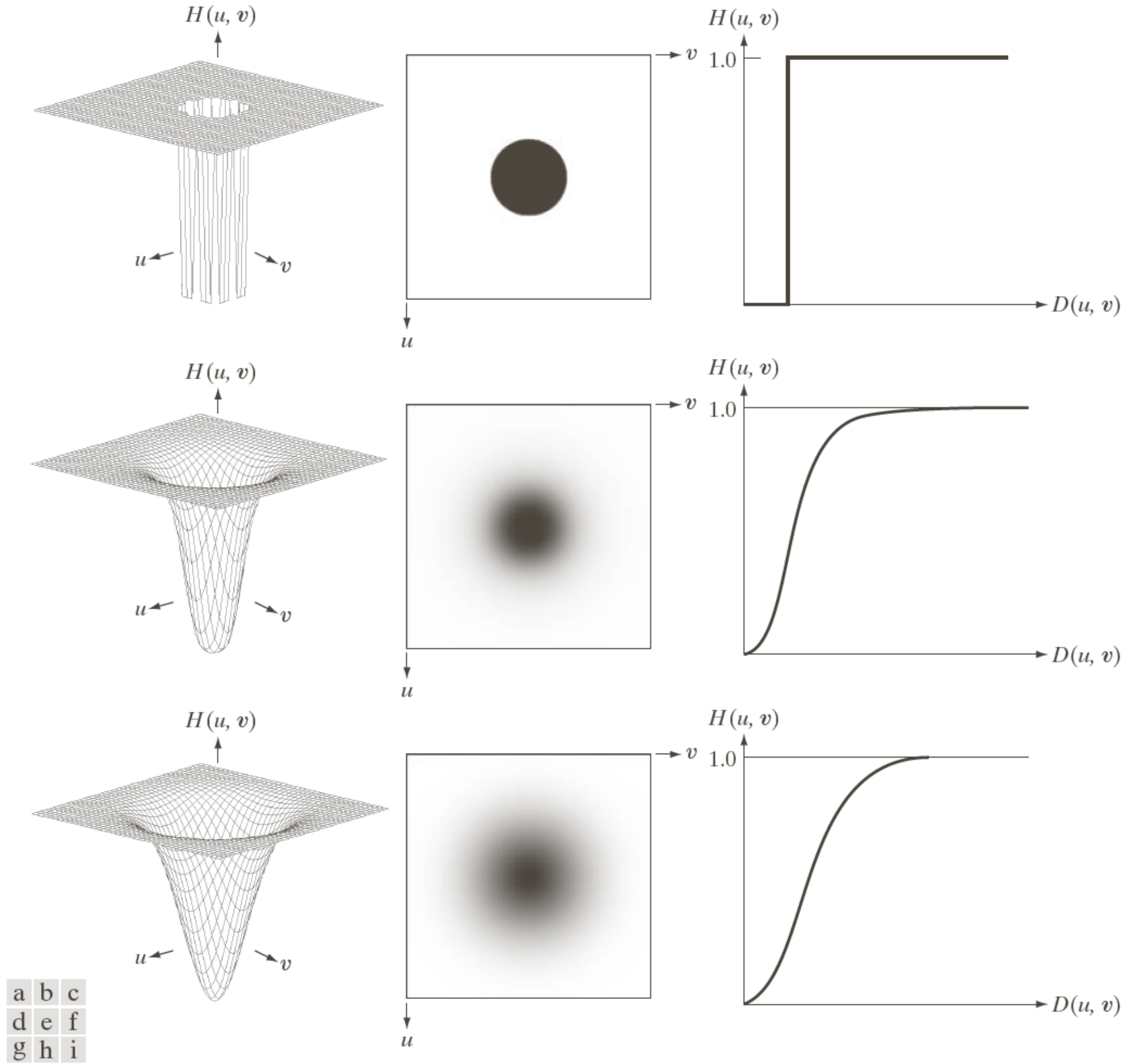


FIGURE 4.52 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Spatial representation of IHPF

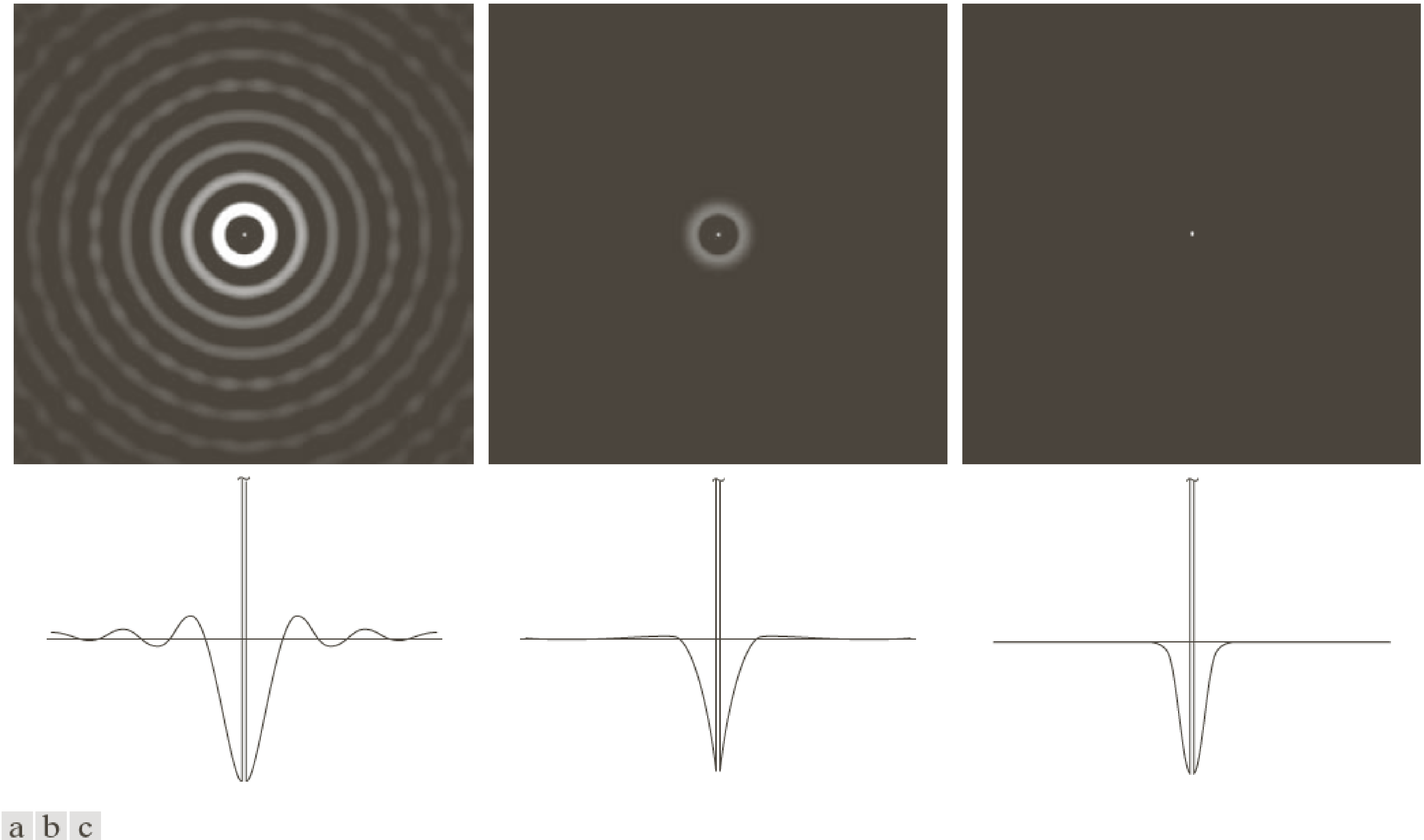


FIGURE 4.53 Spatial representation of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding intensity profiles through their centers.

Results of IHPF

a b c

FIGURE 4.54 Results of highpass filtering the image in Fig. 4.41(a) using an IHPF with $D_0 = 30, 60$, and 160.

Results of BHPF

a b c

FIGURE 4.55 Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). These results are much smoother than those obtained with an IHPF.

Results of GHPF

a b c

FIGURE 4.56 Results of highpass filtering the image in Fig. 4.41(a) using a GHPF with $D_0 = 30, 60$, and 160, corresponding to the circles in Fig. 4.41(b). Compare with Figs. 4.54 and 4.55.

Applications of HPFs

a b c

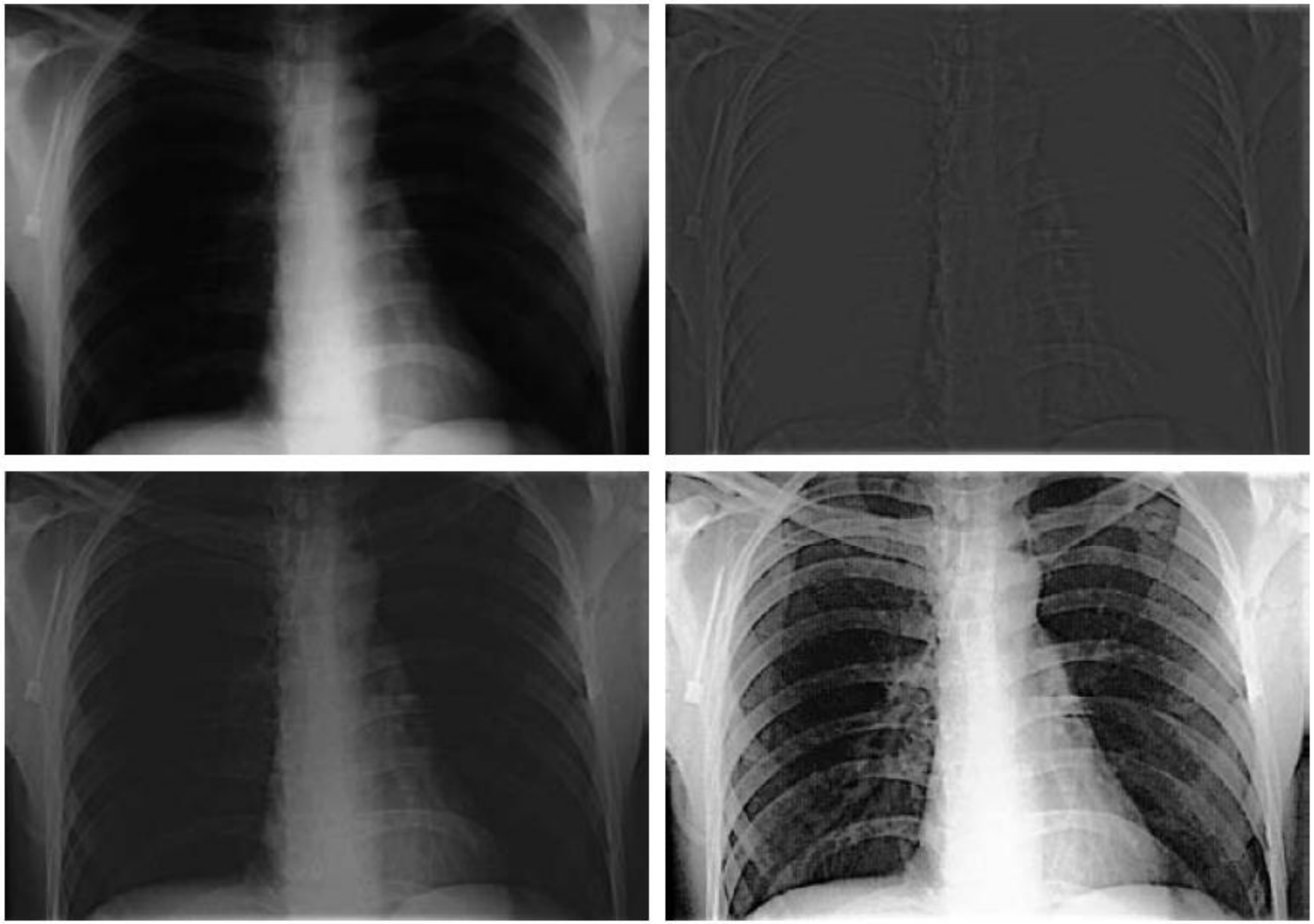
FIGURE 4.57 (a) Thumb print. (b) Result of highpass filtering (a). (c) Result of thresholding (b). (Original image courtesy of the U.S. National Institute of Standards and Technology.)

Applications of HPFs

a b

FIGURE 4.58 (a) Original, blurry image. (b) Image enhanced using the Laplacian in the frequency domain. Compare with Fig. 3.38(e).

Applications of HPFs



a b
c d

FIGURE 4.59 (a) A chest X-ray image. (b) Result of highpass filtering with a Gaussian filter. (c) Result of high-frequency-emphasis filtering using the same filter. (d) Result of performing histogram equalization on (c). (Original image courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)

Summary of HPFs

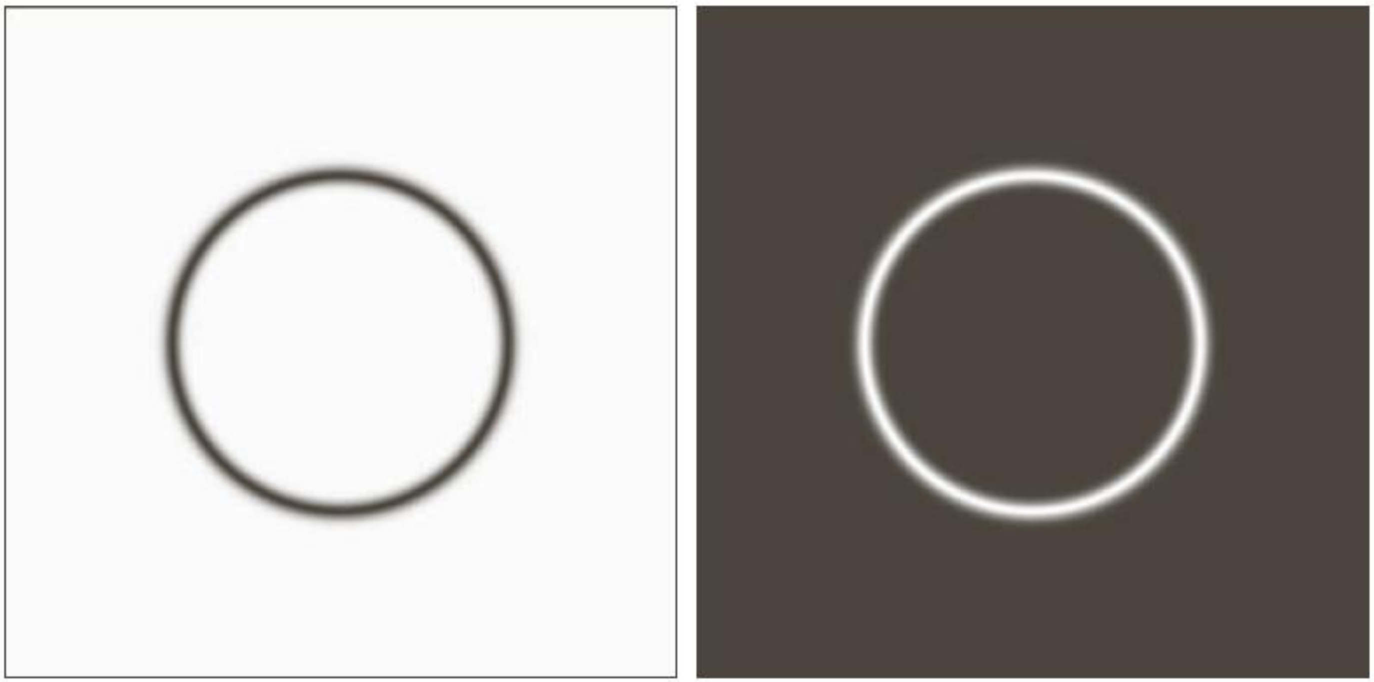
TABLE 4.5
Highpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

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

Bandreject Filters

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Bandreject and bandpass Filters



a b

FIGURE 4.63
(a) Bandreject Gaussian filter. (b) Corresponding bandpass filter. The thin black border in (a) was added for clarity; it is not part of the data.

Summary of Bandreject Filters

TABLE 4.6
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
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

Noise Removal using Frequency Filters

Noise Removal using Bandreject Filters

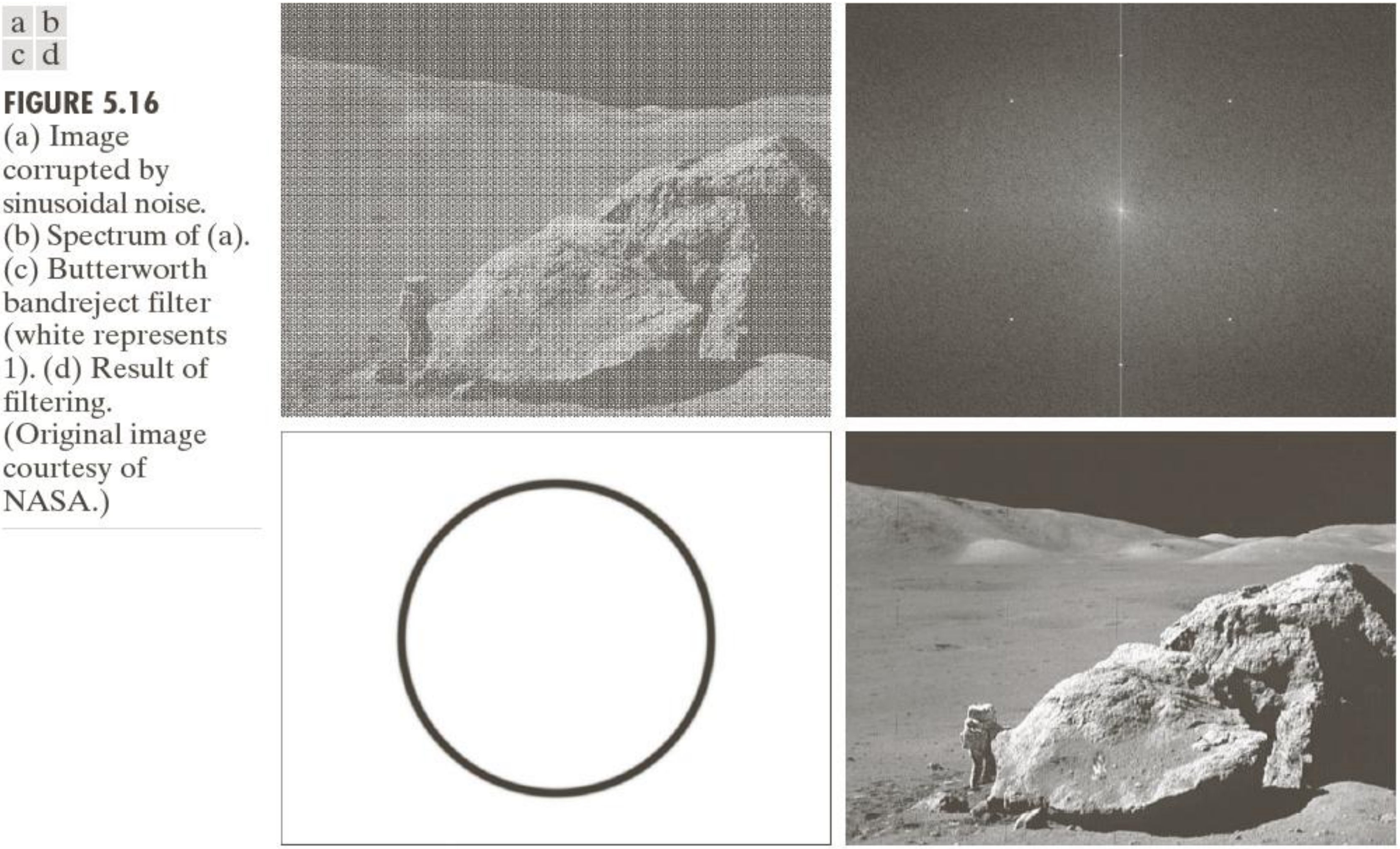


FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Noise Pattern

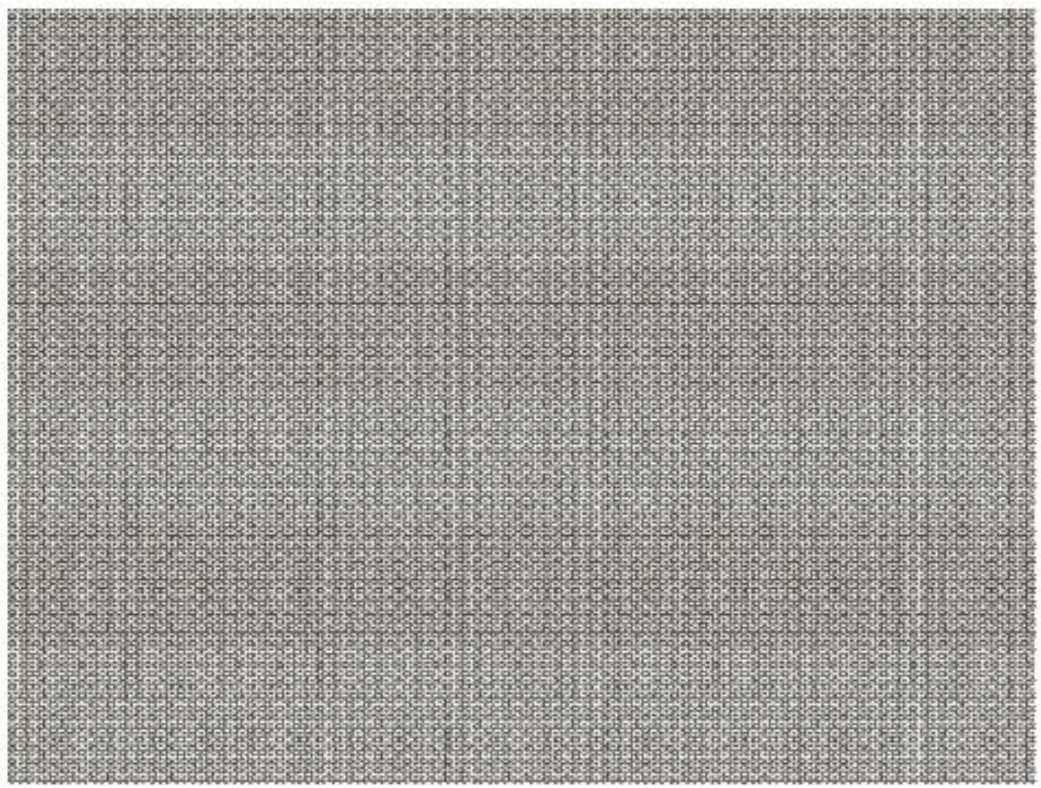
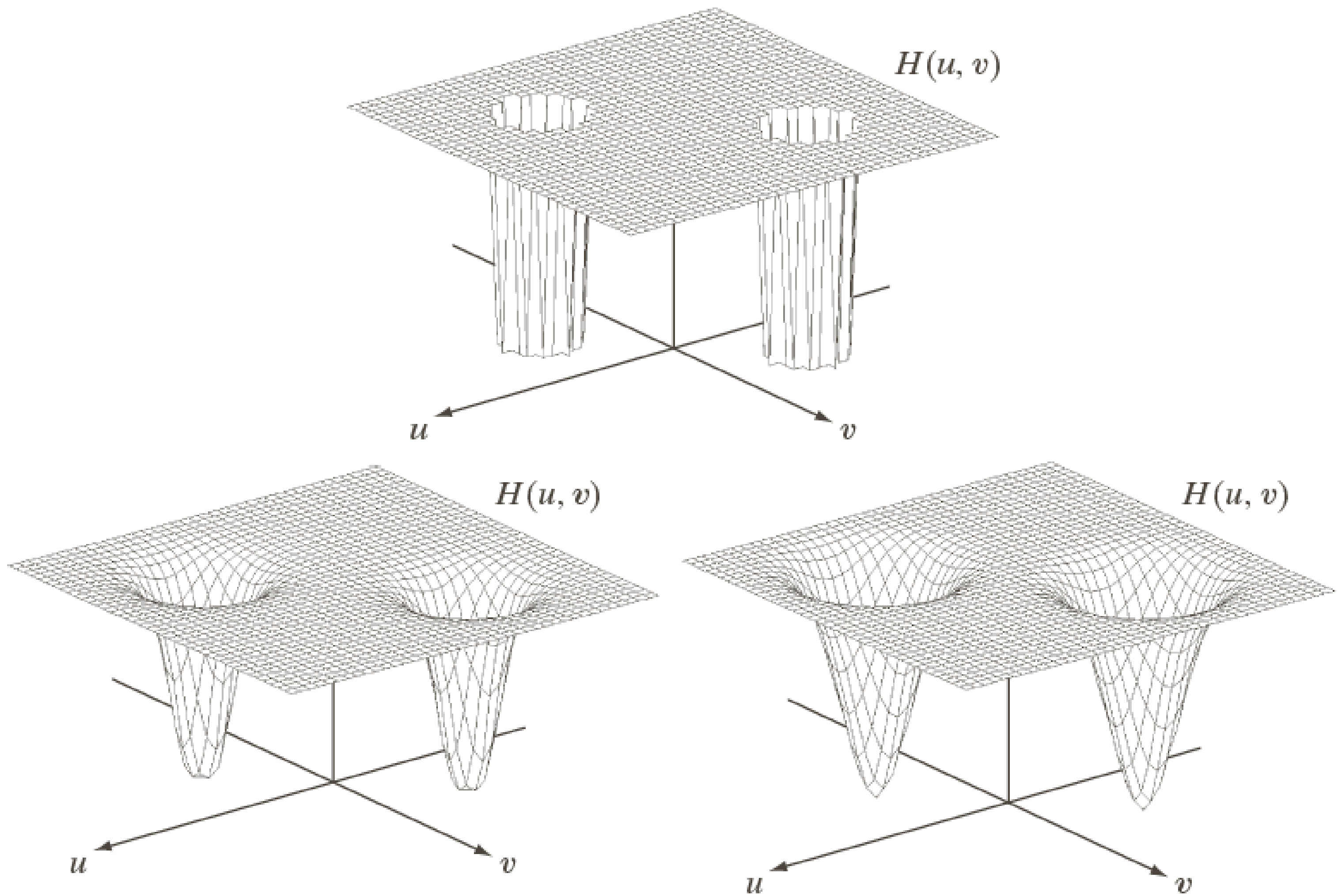


FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.

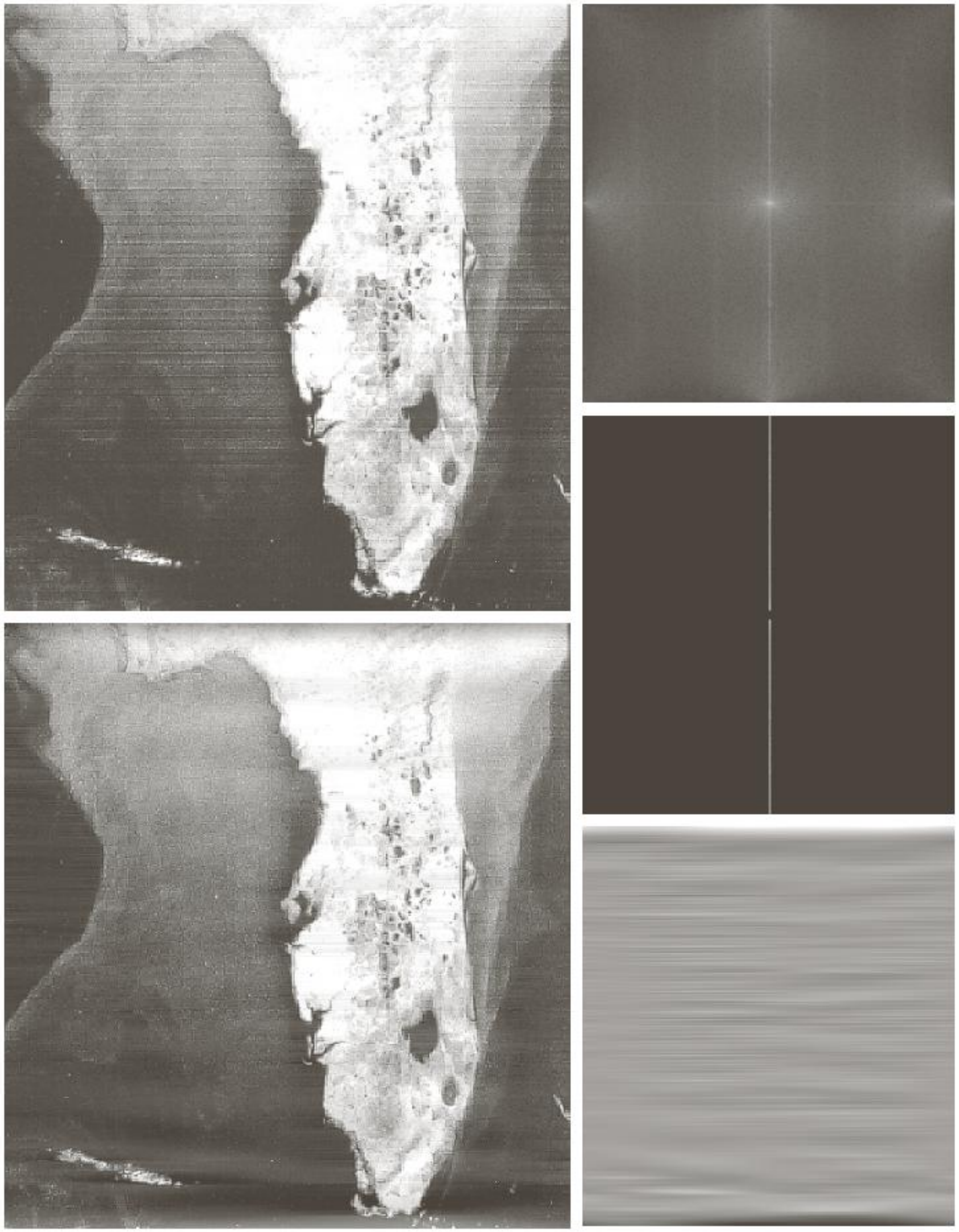
Notch Filters

a
b c

FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.



Application of Notch Filter



a b c e d

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico showing horizontal scan lines. (b) Spectrum. (c) Notch pass filter superimposed on (b). (d) Spatial noise pattern. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

Noise Removal using Frequency Filtering

Noise Removal using Frequency Filtering

$$y(n_1,n_2) = x(n_1,n_2) + \cos(0.1\pi n_2)$$

Noise Removal using Frequency Filtering

Spectral Texture Analysis

Spectral techniques: Fourier transform

- Suitable to detect directionality of periodic and almost periodic 2-D patterns in an image
- Periodic texture patterns are easily detectable by concentration of high energy burst in the spectrum
- Features of Fourier spectrum for texture representation are:
  - Prominent peaks in the spectrum give the principal direction of texture patterns
  - The location of peaks give the frequency and thus the scale of repetition of a pattern
- Eliminating any periodic components via filtering leaves non-periodic image elements which can be described by statistical techniques

Spectral techniques: Fourier transform

- Simplified by expressing the spectrum in polar coordinates to yield a function $S(r, \theta)$ where S is the spectrum function and r and θ are the polar coordinates.

  For each direction θ, $S(r, \theta)$ = a 1-D function $S_\theta(r)$

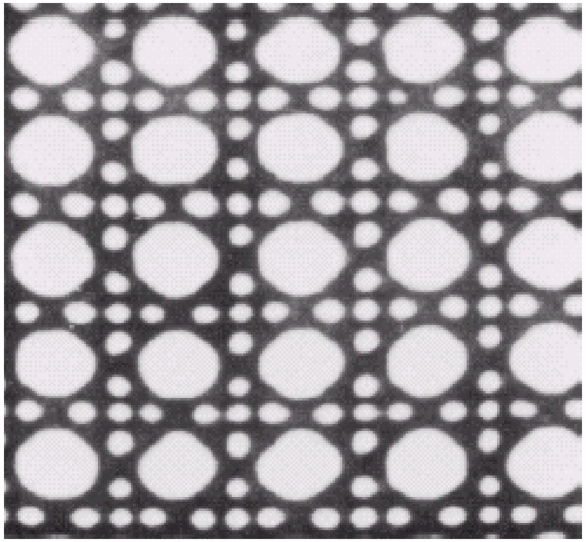
  For each frequency r, $S(r, \theta)$ = a 1-D function $S_r(\theta)$
- Analyzing $S_\theta(r)$ for a fixed θ, gives the distance from the origin and thus the scale of repetition of a texture pattern.
- Analyzing $S_r(\theta)$ for a fixed r, gives the direction and thus the orientation of the periodic texture pattern.
- To measure this analysis, we define two quantities

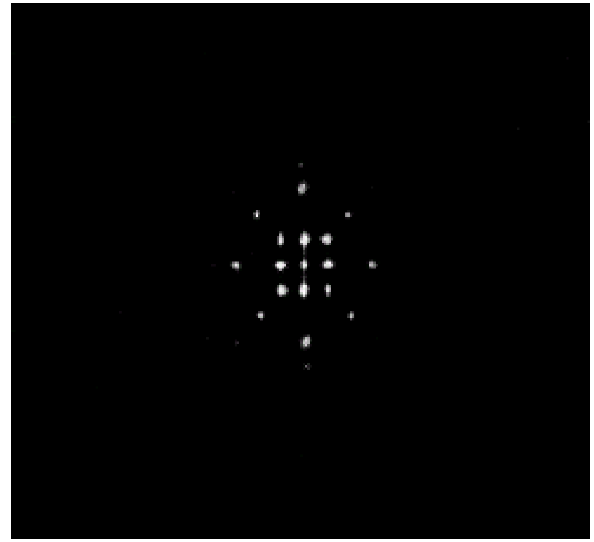
$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r), \qquad S(\theta) = \sum_{r=1}^{R_0} S_r(\theta).$$

These quantities measure the spectral response and give the dominant directions and scales of periodic texture patterns.

Spectral techniques: Fourier transform

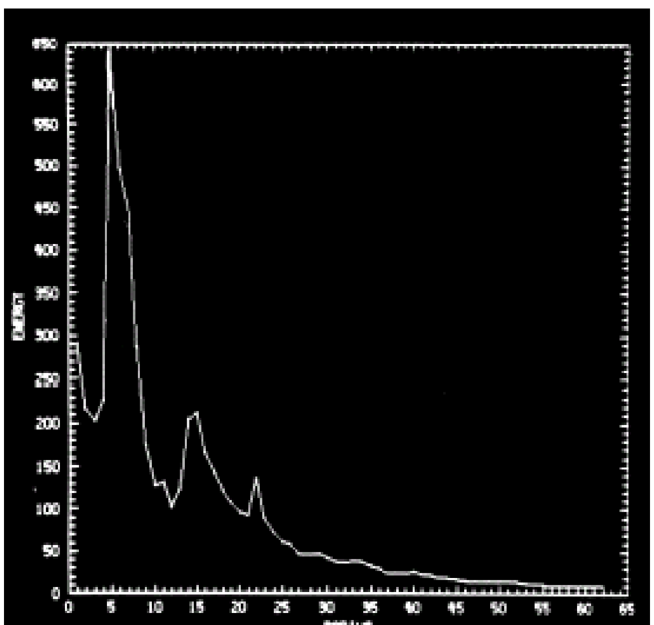
Image showing periodic texture

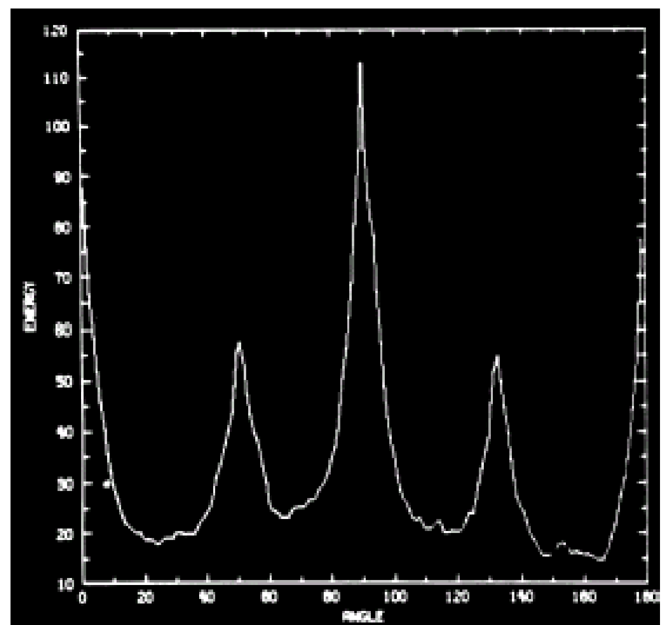




Spectrum

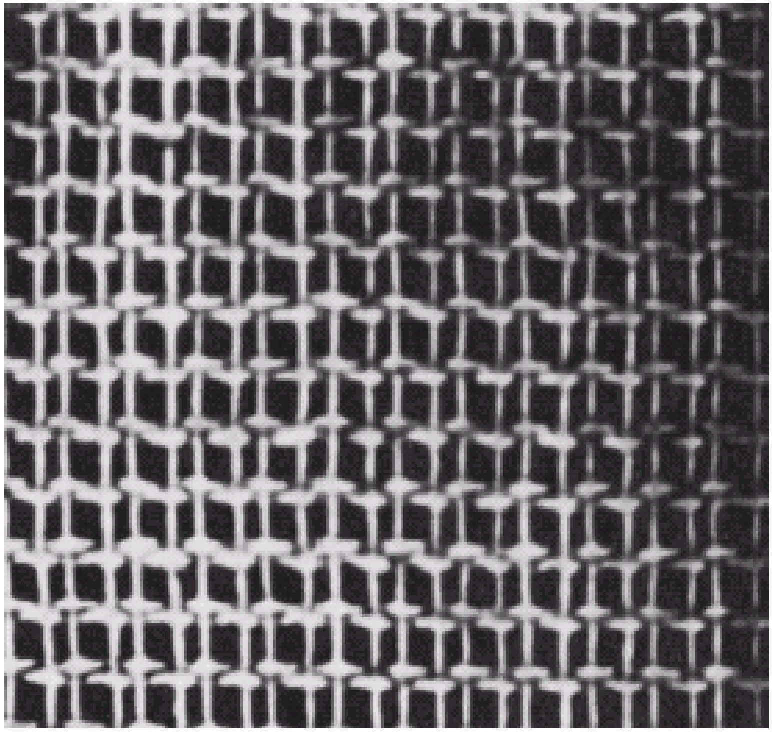
Plot of *$S(r)$*



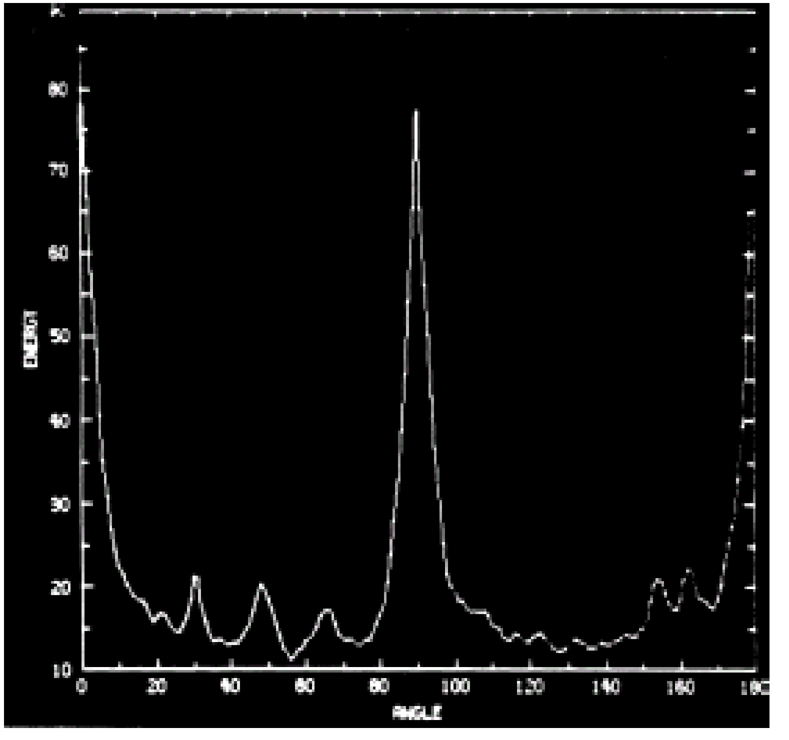


Plot of *$S(\theta)$*

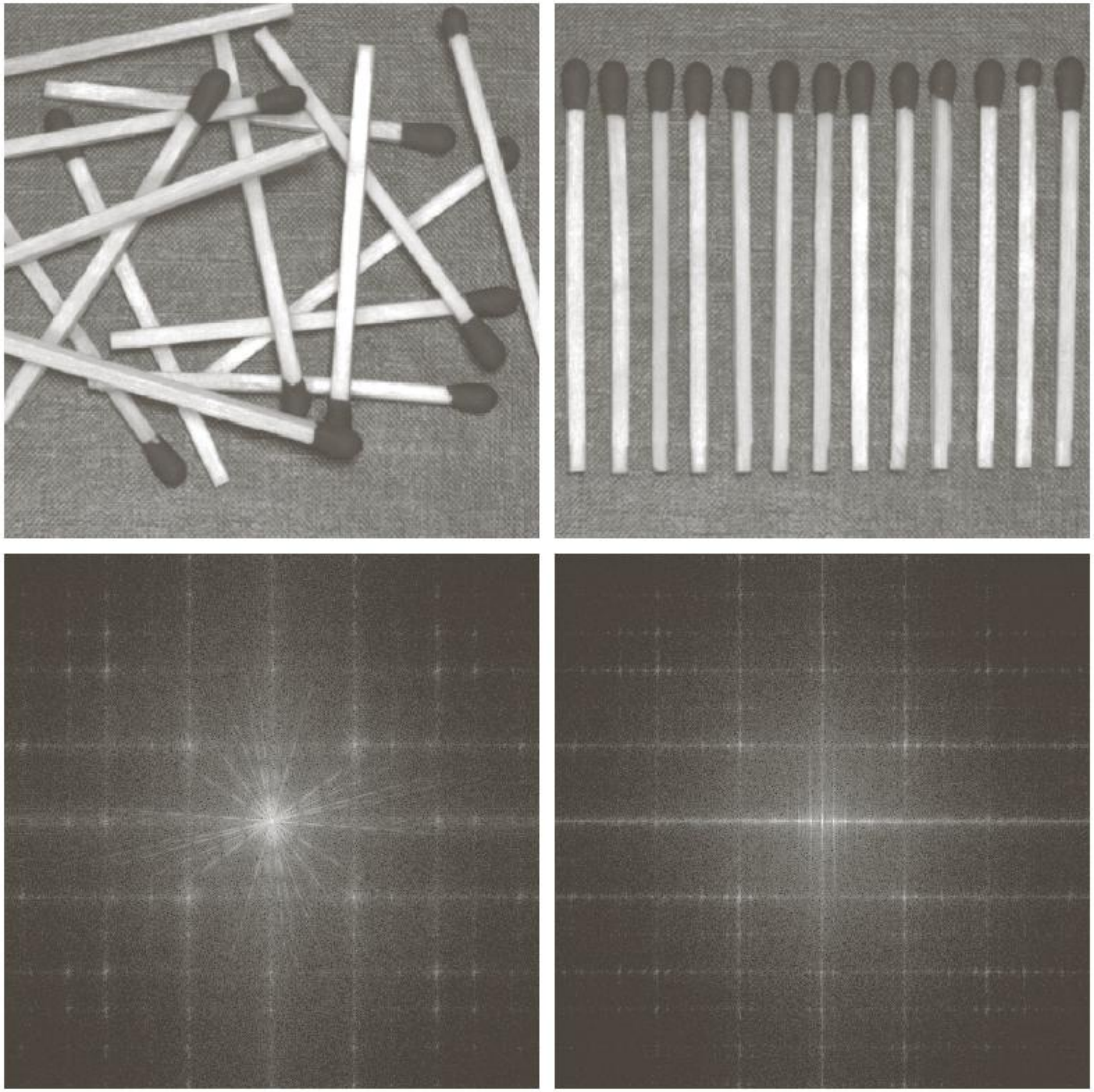
Spectral techniques: Fourier transform (example)



Another image showing periodic texture



Plot of $S(\theta)$



a b
c d

FIGURE 11.35 (a) and (b) Images of random and ordered objects. (c) and (d) Corresponding Fourier spectra. All images are of size 600×600 pixels.

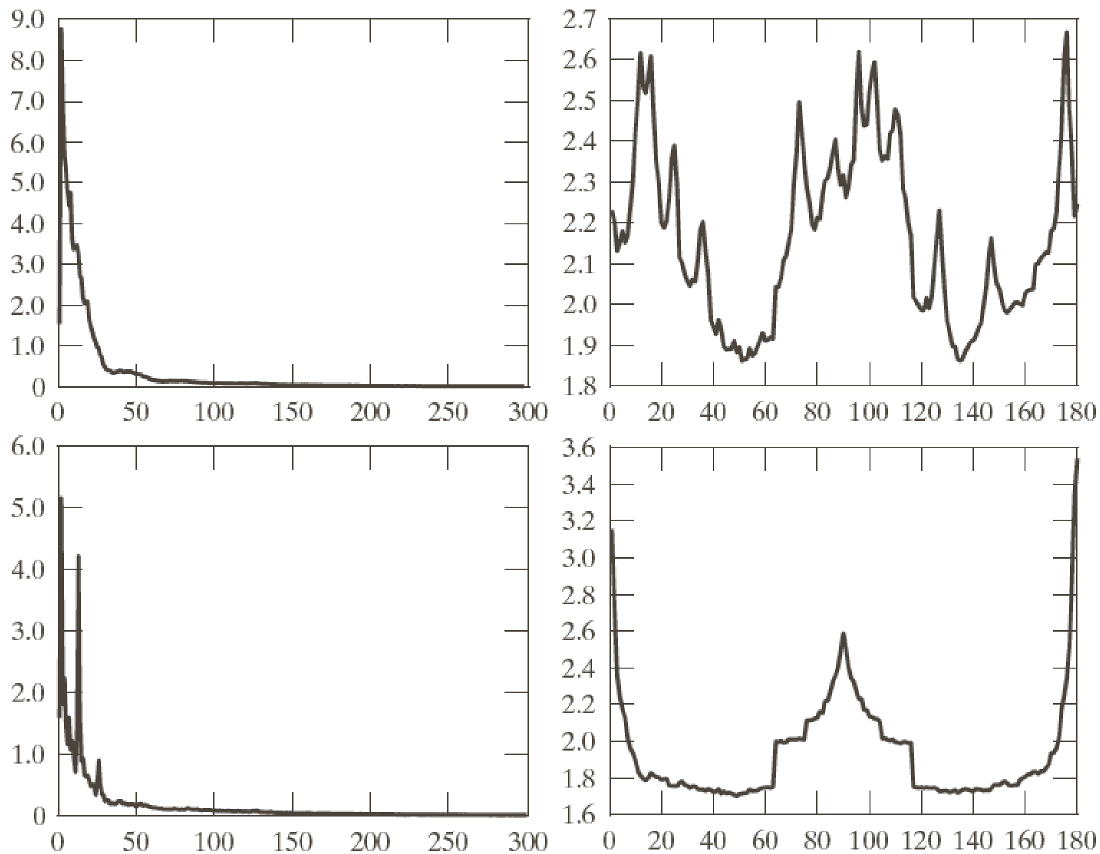


FIGURE 11.36 Plots of (a) $S(r)$ and (b) $S(\theta)$ for Fig. 11.35(a). (c) and (d) are plots of $S(r)$ and $S(\theta)$ for Fig. 11.35(b). All vertical axes are $\times 10^5$.

Readings from Book (3rd Edn.)

- Frequency Filters (Chapter-4)
- Noise Removal using Frequency Filters (Chapter-5)
- Spectral Texture (Chapter-11)

Acknowledgements

Material in these slides has been taken from, the following resources

- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Peters, Richard Alan, II, Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008
- Brian Mac Namee, Digitial Image Processing, School of Computing, Dublin Institute of Technology
- Computer Vision for Computer Graphics, Mark Borg