EE 604, Digital Image Processing

Chapter 4: Filtering in the Frequency Domain

Dr. W. David Pan
Dept. of ECE
UAH

Topics

- Fourier Series
- Fourier Transform
- Discrete Fourier Transform
- Properties of 2-D DFT
- Basics of Filtering in the Frequency Domain
- Correspondence between Filtering in the Spatial and Frequency Domains
- Image Smoothing Using Frequency Domain Filtering
- Image Sharpening Using Frequency Domain Filtering
- Selective Filtering

Frequency and Filter

- Frequency
  - The number of times that a periodic function repeats the same sequence of values during a unit variation of the independent variable.
- Filter
  - A device or material for suppressing or minimizing waves or oscillations of certain frequencies.

Fourier Series

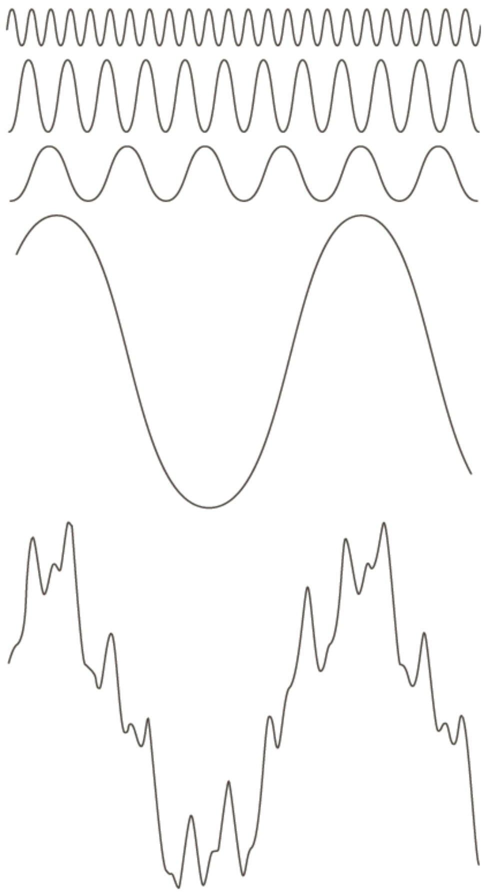


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{T}t}$$

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\, e^{-j\frac{2\pi n}{T}t}\, dt$$

for $n = 0, \pm 1, \pm 2, \ldots$

Fourier Transform

$$\Im\{f(t)\} = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi\mu t}\, dt$$

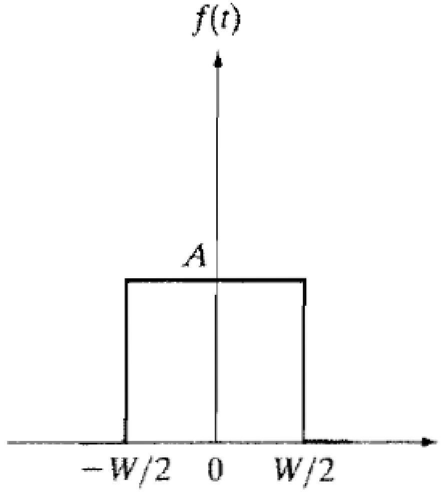
$$F(\mu) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi\mu t}\, dt$$

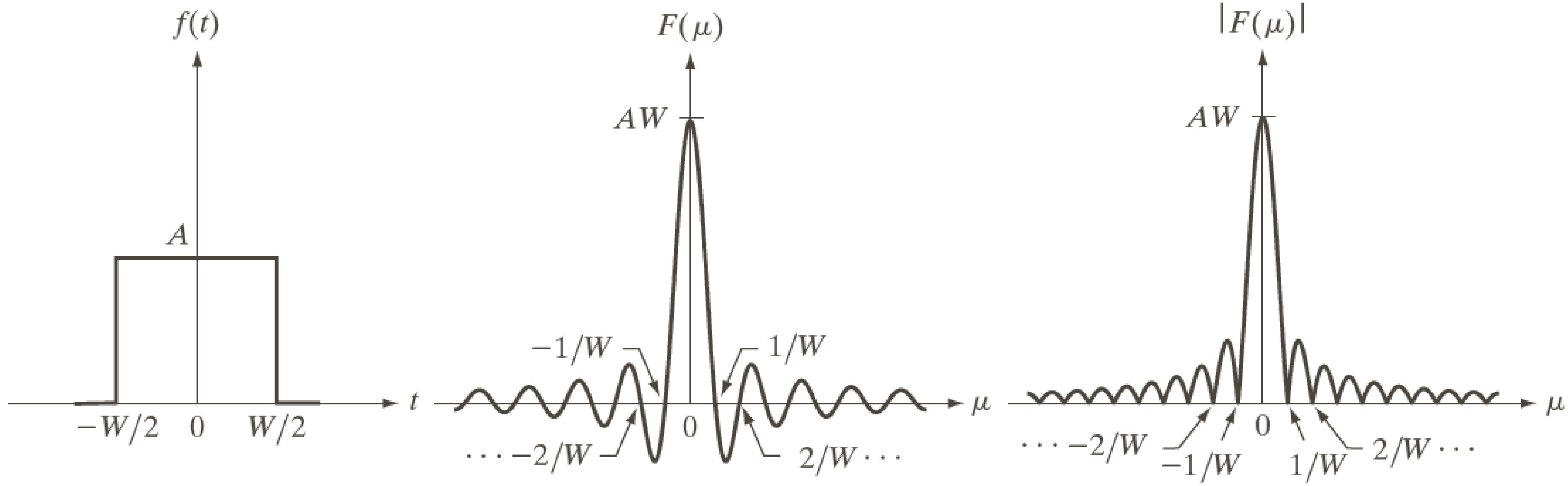
$$F(\mu) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi\mu t}\, dt = \int_{-W/2}^{W/2} A e^{-j2\pi\mu t}\, dt$$

$$= \frac{-A}{j2\pi\mu}\left[e^{-j2\pi\mu t}\right]_{-W/2}^{W/2} = \frac{-A}{j2\pi\mu}\left[e^{-j\pi\mu W} - e^{j\pi\mu W}\right]$$

$$= \frac{A}{j2\pi\mu}\left[e^{j\pi\mu W} - e^{-j\pi\mu W}\right]$$

$$= AW\frac{\sin(\pi\mu W)}{(\pi\mu W)}$$





a b c

FIGURE 4.4 (a) A simple function; (b) its Fourier transform; and (c) the spectrum. All functions extend to infinity in both directions.

Convolution

$$f(t) \star h(t) = \int_{-\infty}^{\infty} f(\tau) h(t - \tau) \, d\tau$$

$$\Im\{f(t) \star h(t)\} = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(\tau) h(t - \tau) \, d\tau \right] e^{-j2\pi\mu t} \, dt$$

$$= \int_{-\infty}^{\infty} f(\tau) \left[\int_{-\infty}^{\infty} h(t - \tau) e^{-j2\pi\mu t} \, dt \right] d\tau$$

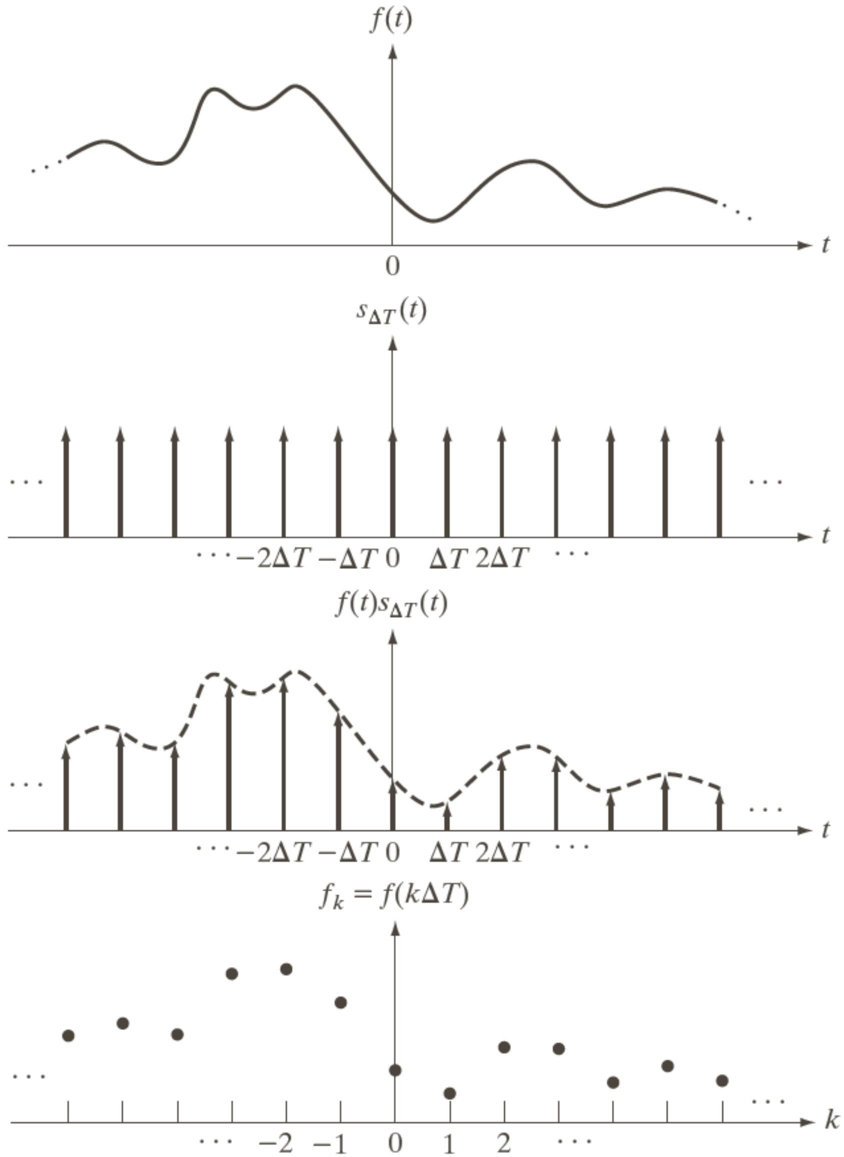
$$\Im\{f(t) \star h(t)\} = \int_{-\infty}^{\infty} f(\tau) \left[H(\mu) e^{-j2\pi\mu\tau} \right] d\tau$$

$$= H(\mu) \int_{-\infty}^{\infty} f(\tau) e^{-j2\pi\mu\tau} \, d\tau$$

$$= H(\mu) F(\mu)$$

$$f(t) \star h(t) \Leftrightarrow H(\mu) F(\mu)$$

Sampling



a
b
c
d

FIGURE 4.5
(a) A continuous function. (b) Train of impulses used to model the sampling process. (c) Sampled function formed as the product of (a) and (b). (d) Sample values obtained by integration and using the sifting property of the impulse. (The dashed line in (c) is shown for reference. It is not part of the data.)

$$\frac{1}{\Delta T} > 2\mu_{\max}$$

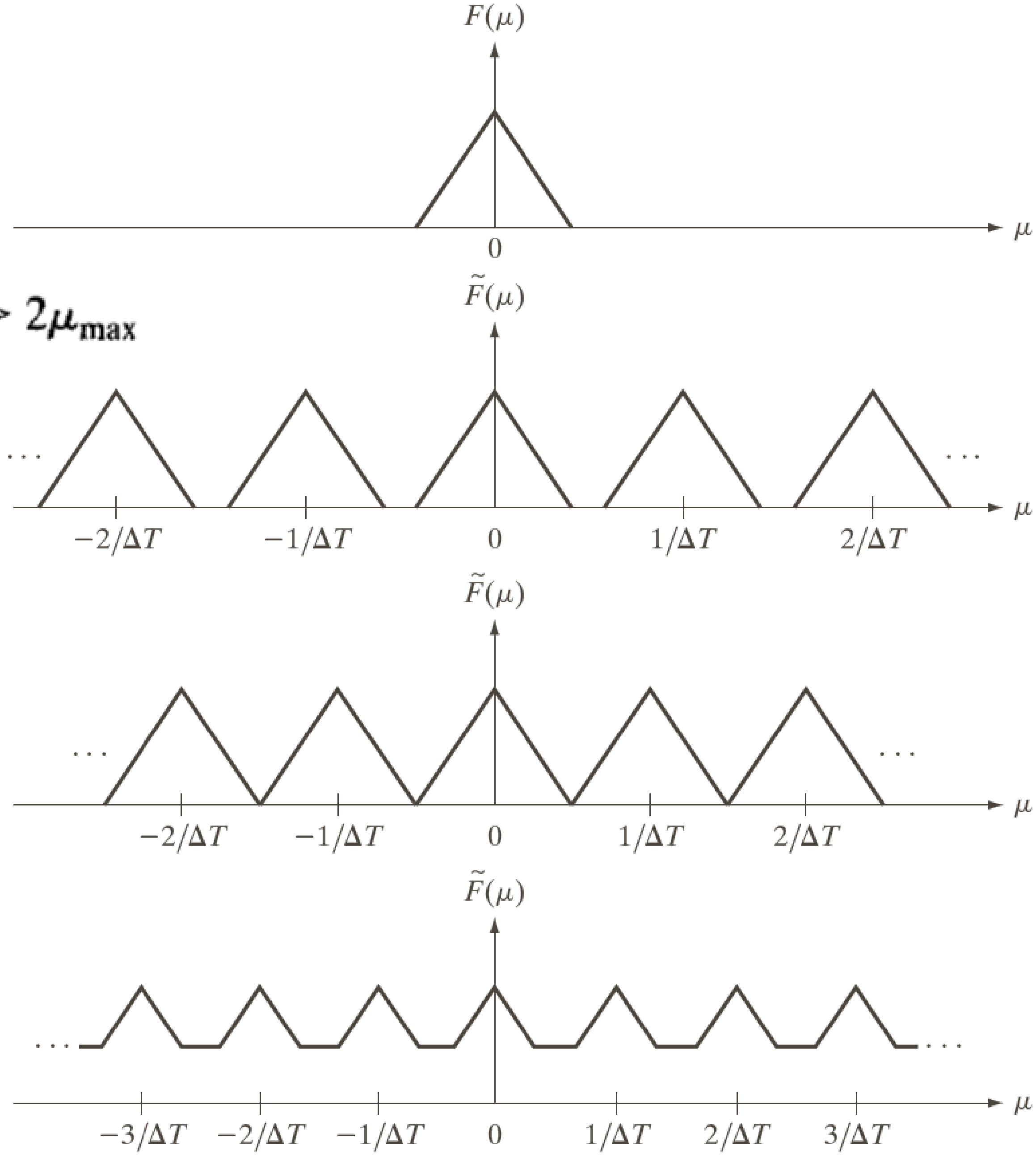
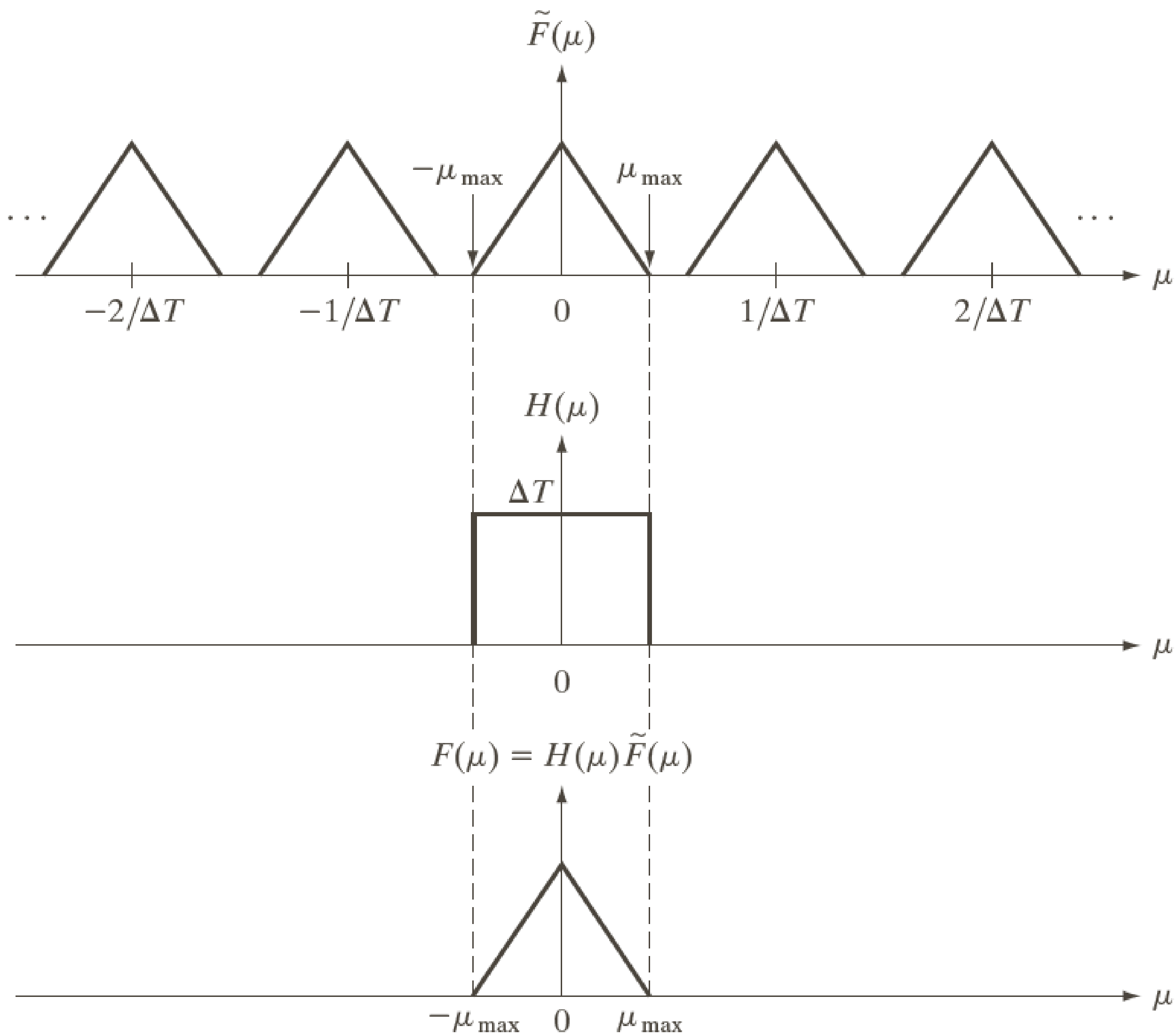


FIGURE 4.6 (a) Fourier transform of a band-limited function. (b)–(d) Transforms of the corresponding sampled function under the conditions of over-sampling, critically-sampling, and under-sampling, respectively.



a
b
c

FIGURE 4.8 Extracting one period of the transform of a band-limited function using an ideal lowpass filter.

Aliasing

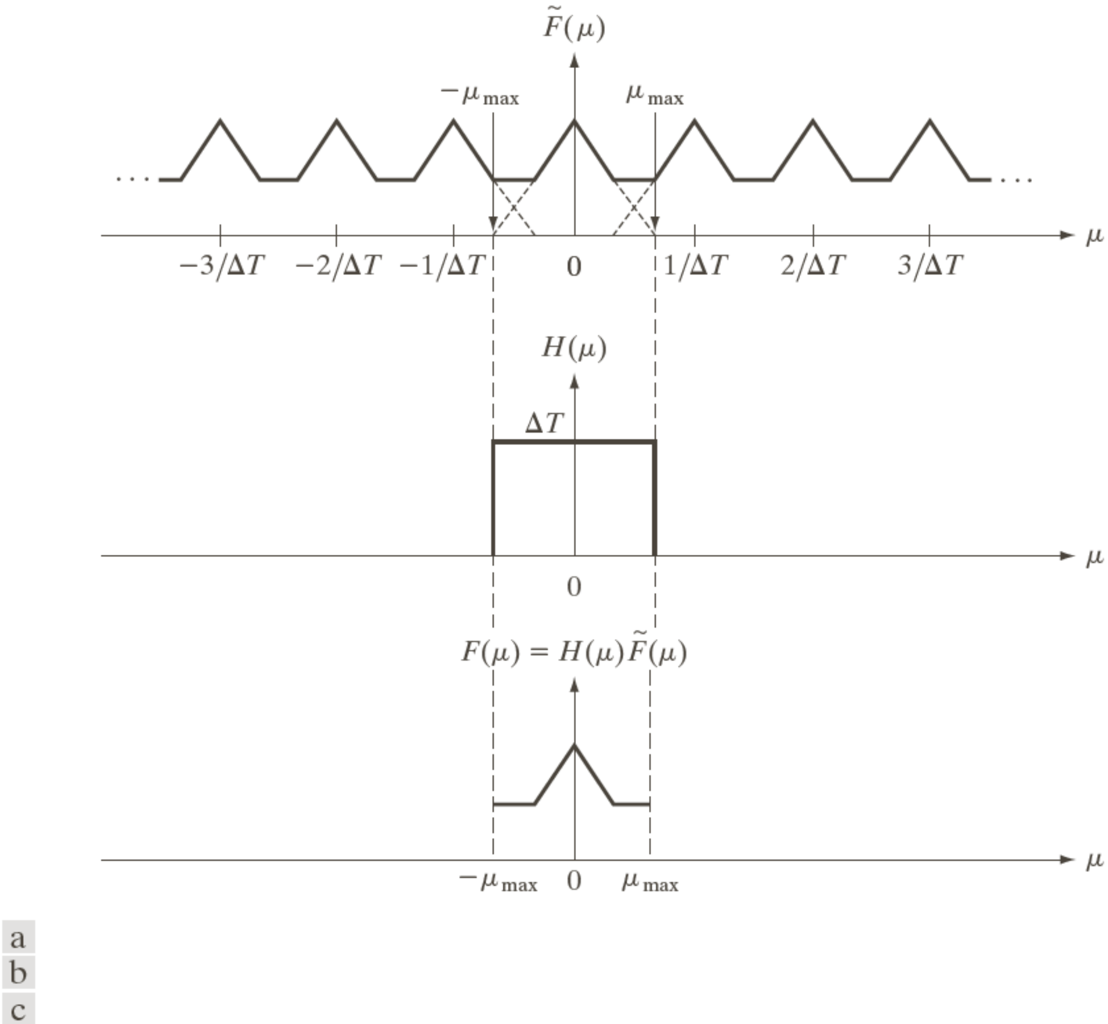


FIGURE 4.9 (a) Fourier transform of an under-sampled, band-limited function. (Interference from adjacent periods is shown dashed in this figure). (b) The same ideal lowpass filter used in Fig. 4.8(b). (c) The product of (a) and (b). The interference from adjacent periods results in aliasing that prevents perfect recovery of $F(\mu)$ and, therefore, of the original, band-limited continuous function. Compare with Fig. 4.8.

Discrete Fourier Transform

$$F_m = \sum_{n=0}^{M-1} f_n e^{-j2\pi mn/M} \qquad m = 0, 1, 2, \ldots, M-1$$

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \qquad u = 0, 1, 2, \ldots, M-1$$

$$F(u) = F(u + kM)$$

$$f_n = \frac{1}{M} \sum_{m=0}^{M-1} F_m e^{j2\pi mn/M} \qquad n = 0, 1, 2, \ldots, M-1$$

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \qquad x = 0, 1, 2, \ldots, M-1$$

$$f(x) = f(x + kM)$$

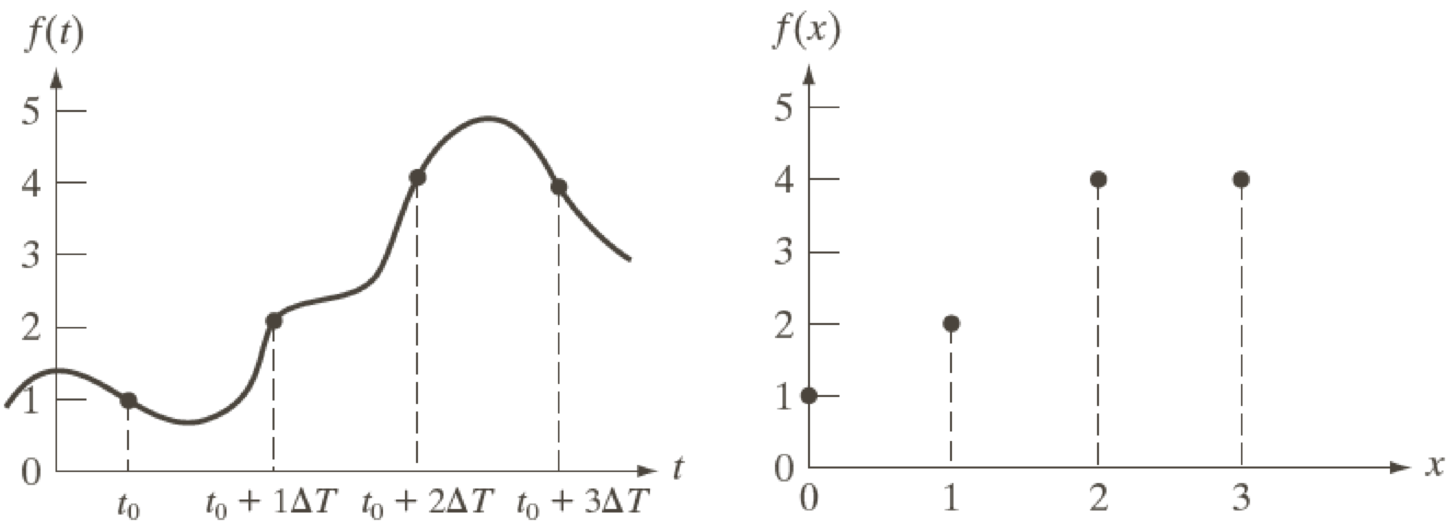
$$F(0) = \sum_{x=0}^{3} f(x) = \left[f(0) + f(1) + f(2) + f(3)\right] = 1 + 2 + 4 + 4 = 11$$

$$F(1) = \sum_{x=0}^{3} f(x) e^{-j2\pi(1)x/4}$$

$$= 1e^{0} + 2e^{-j\pi/2} + 4e^{-j\pi} + 4e^{-j3\pi/2} = -3 + 2j$$

$$f(0) = \frac{1}{4}\sum_{u=0}^{3} F(u) e^{j2\pi u(0)}$$

$$= \frac{1}{4}\sum_{u=0}^{3} F(u) = \frac{1}{4}[11 - 3 + 2j - 1 - 3 - 2j] = \frac{1}{4}[4] = 1$$



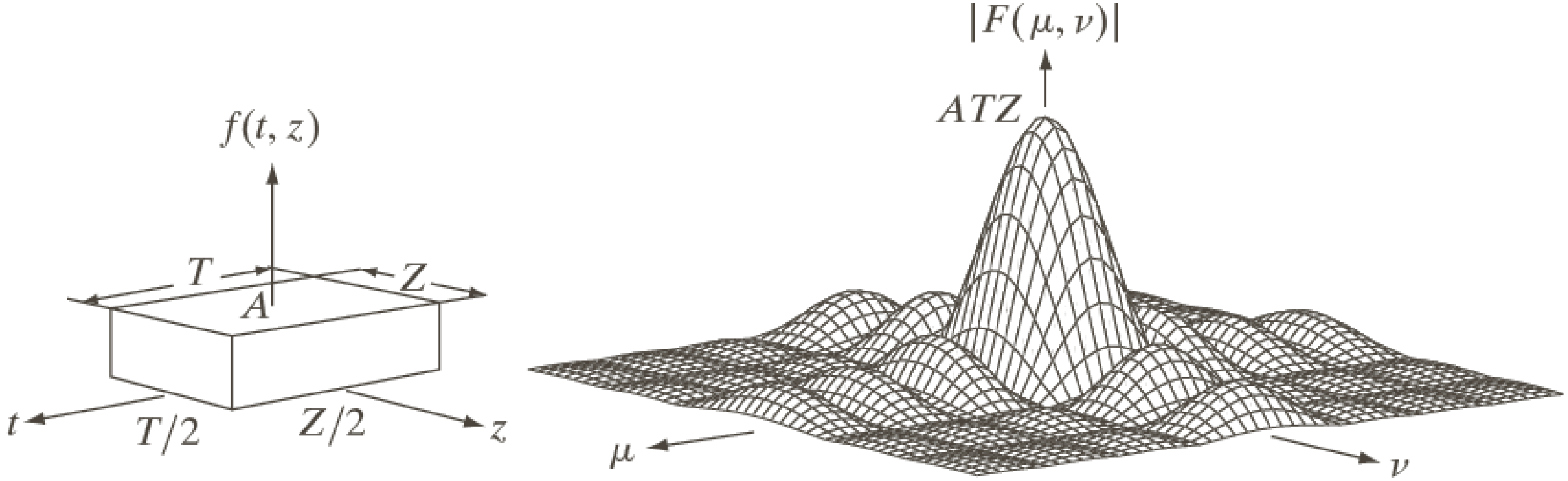
a b

FIGURE 4.11 (a) A function, and (b) samples in the x-domain. In (a), t is a continuous variable; in (b), x represents integer values.

2-D DFT

$$F(\mu, \nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t, z)e^{-j2\pi(\mu t + \nu z)}\,dt\,dz$$

$$f(t, z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu, \nu)e^{j2\pi(\mu t + \nu z)}\,d\mu\,d\nu$$



a b

FIGURE 4.13 (a) A 2-D function, and (b) a section of its spectrum (not to scale). The block is longer along the t-axis, so the spectrum is more "contracted" along the μ-axis. Compare with Fig. 4.4.

Sampling and Aliasing

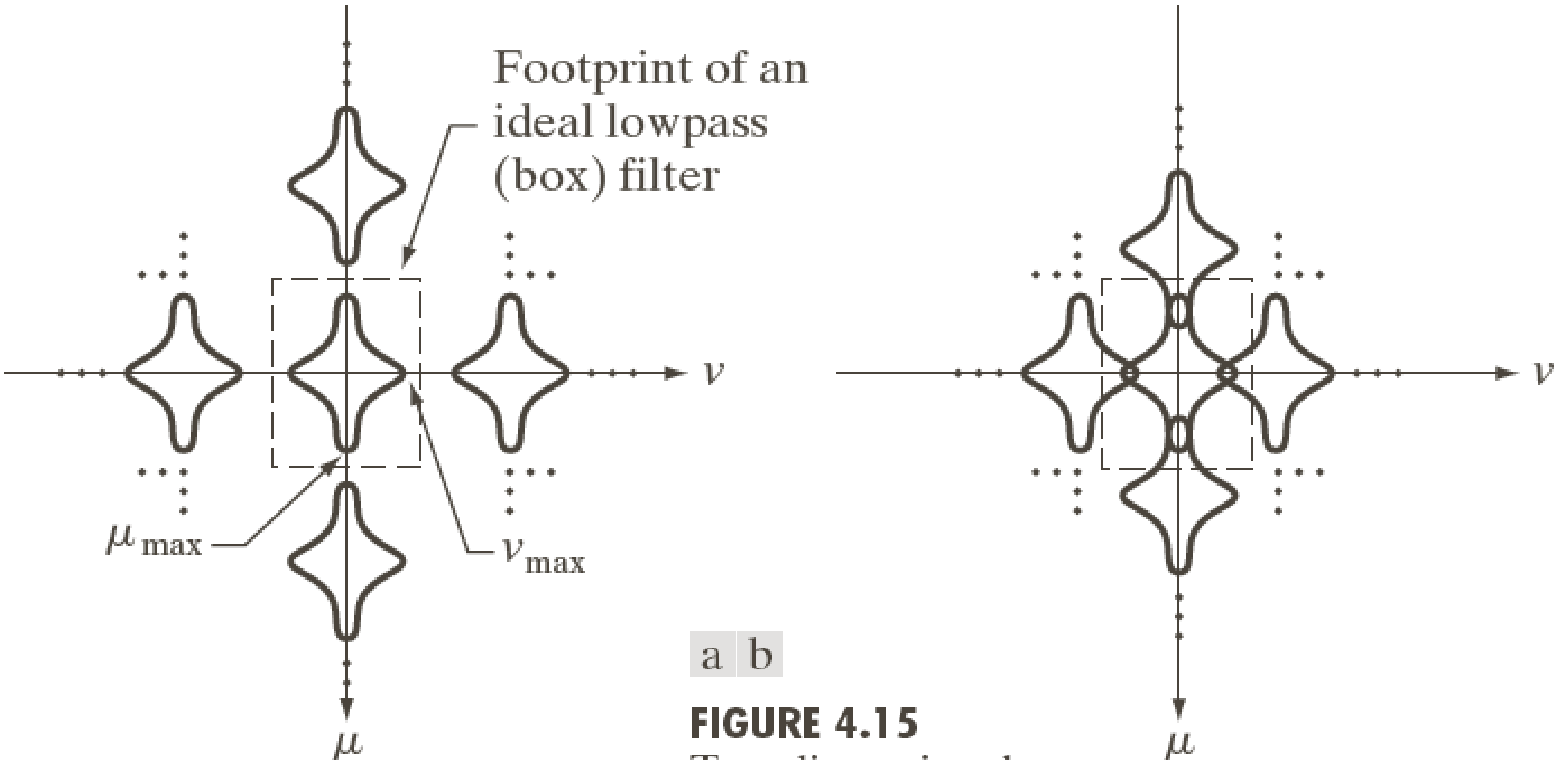


FIGURE 4.15 Two-dimensional Fourier transforms of (a) an over-sampled, and (b) under-sampled band-limited function.

2-D DFT and its Properties

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

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

Translation:

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

$$f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi\left(\frac{x_0u}{M}+\frac{y_0v}{N}\right)}$$

Rotation:

$$x = rcos\theta \quad y = rsin\theta; \quad u = \omega cos\phi \quad v = \omega sin\phi$$

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

Periodicity

$$F(u,v) = F(u + k_1M, v) = F(u, v + k_2N) = F(u + k_1M, v + k_2N)$$

$$f(x,y) = f(x + k_1M, y) = f(x, y + k_2N) = f(x + k_1M, y + k_2N)$$

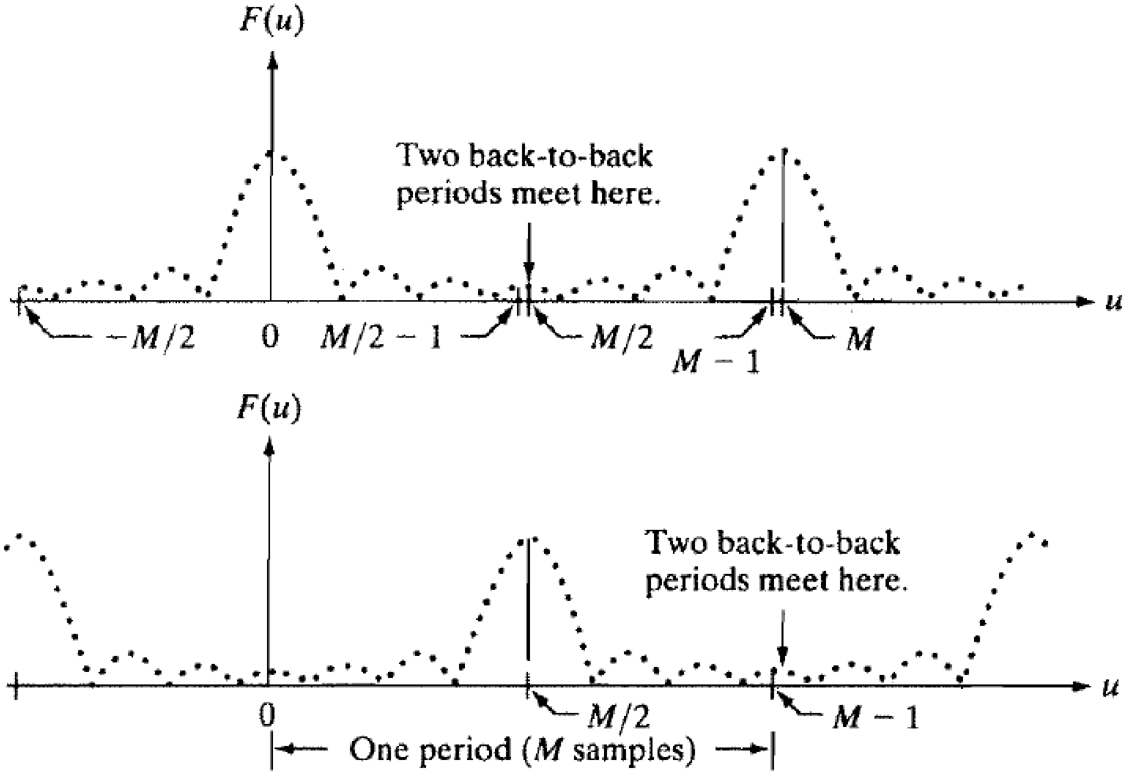
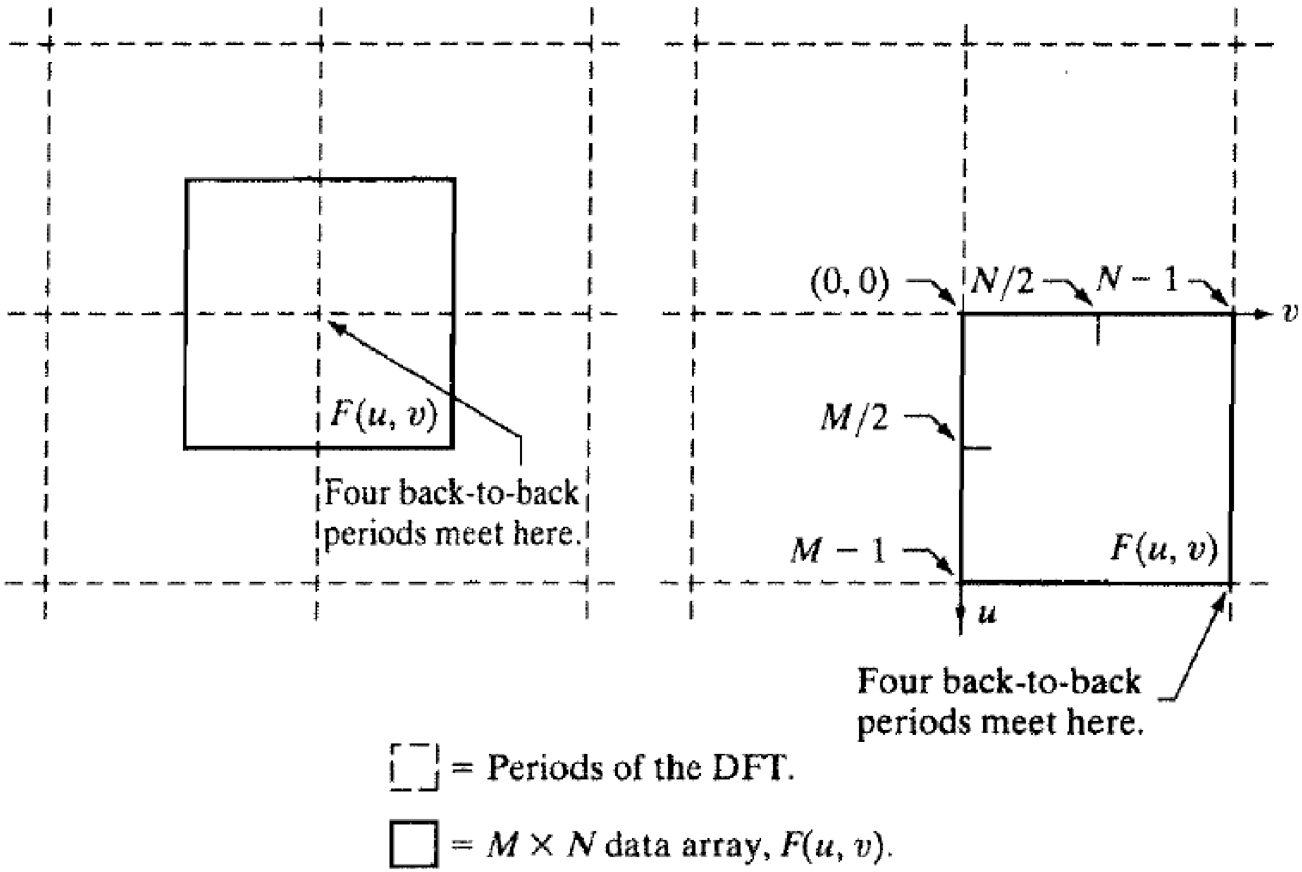


FIGURE 4.23 Centering the Fourier transform. (a) A 1-D DFT showing an infinite number of periods. (b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$.

$$f(x)e^{j2\pi(u_0x/M)} \Leftrightarrow F(u - u_0) \qquad f(x)(-1)^x \Leftrightarrow F(u - M/2)$$



(c) A 2-D DFT showing an infinite number of periods. The solid area is the $M \times N$ data array, $F(u, v)$, obtained with Eq. (4.5-15). This array consists of four quarter periods.

(d) A Shifted DFT obtained by multiplying $f(x, y)$ by $(-1)^{x+y}$ before computing $F(u, v)$. The data now contains one complete, centered period, as in (b).

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

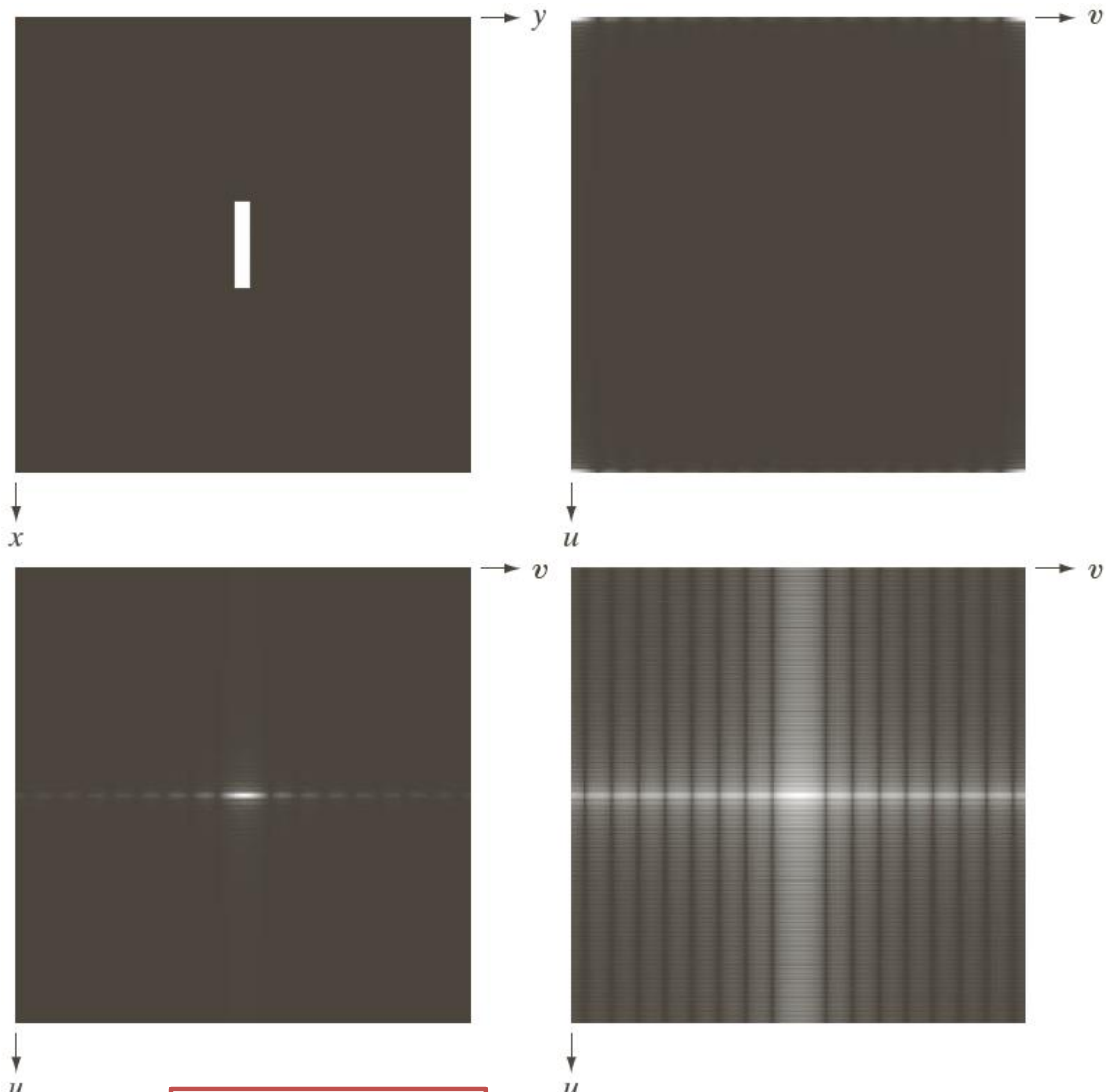
$$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$$

fftshift and ifftshift

- Moving the zero-frequency component to the center of the array.
- It is useful for visualizing a Fourier transform with the zero-frequency component in the middle of the spectrum.

```
>> A = [1 2; 3 4]
>> fftshift (A)
ans =
    4    3
    2    1
>> ifftshift (fftshift(A))

>> F = fft2(I);
     % Centered transform by swapping quadrants of F
>> Fc = fftshift (F);
```



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.

See Matlab code

Symmetry Properties

Any real or complex function can be expressed as the sum of an even and odd part:

$$w(x, y) = w_e(x, y) + w_o(x, y)$$

$$w_e(x, y) \triangleq \frac{w(x, y) + w(-x, -y)}{2}$$

$$w_o(x, y) \triangleq \frac{w(x, y) - w(-x, -y)}{2} \Rightarrow w_o(0,0) = 0$$

$$w_e(x, y) = w_e(-x, -y)$$

$$w_o(x, y) = -w_o(-x, -y)$$

$$w_e(x, y) = w_e(M - x, N - y) \qquad w_o(x, y) = -w_o(M - x, N - y)$$

M and N are the number of rows and columns of a 2-D array

Even and Odd Sequences

$$f = \{f(0) \quad f(1) \quad f(2) \quad f(3)\} = \{2 \quad 1 \quad 1 \quad 1\}$$

$$f(0) = f(4), \quad f(2) = f(2), \quad f(1) = f(3), \quad f(3) = f(1)$$

$$\{a \quad b \quad c \quad b\}$$

$$g = \{g(0) \quad g(1) \quad g(2) \quad g(3)\} = \{0 \quad -1 \quad 0 \quad 1\}$$

$$g(x) = -g(4 - x)$$

Odd Image:

0	0	0	0	0	0
0	0	0	0	0	0
0	0	−1	0	1	0
0	0	−2	0	2	0
0	0	−1	0	1	0
0	0	0	0	0	0

	Spatial Domain†		Frequency Domain†
1)	$f(x, y)$ real	$\Leftrightarrow$	$F^*(u, v) = F(-u, -v)$
2)	$f(x, y)$ imaginary	$\Leftrightarrow$	$F^*(-u, -v) = -F(u, v)$
3)	$f(x, y)$ real	$\Leftrightarrow$	$R(u, v)$ even; $I(u, v)$ odd
4)	$f(x, y)$ imaginary	$\Leftrightarrow$	$R(u, v)$ odd; $I(u, v)$ even
5)	$f(-x, -y)$ real	$\Leftrightarrow$	$F^*(u, v)$ complex
6)	$f(-x, -y)$ complex	$\Leftrightarrow$	$F(-u, -v)$ complex
7)	$f^*(x, y)$ complex	$\Leftrightarrow$	$F^*(-u - v)$ complex
8)	$f(x, y)$ real and even	$\Leftrightarrow$	$F(u, v)$ real and even
9)	$f(x, y)$ real and odd	$\Leftrightarrow$	$F(u, v)$ imaginary and odd
10)	$f(x, y)$ imaginary and even	$\Leftrightarrow$	$F(u, v)$ imaginary and even
11)	$f(x, y)$ imaginary and odd	$\Leftrightarrow$	$F(u, v)$ real and odd
12)	$f(x, y)$ complex and even	$\Leftrightarrow$	$F(u, v)$ complex and even
13)	$f(x, y)$ complex and odd	$\Leftrightarrow$	$F(u, v)$ complex and odd

†Recall that x, y, u, and v are *discrete* (integer) variables, with x and u in the range $[0, M - 1]$, and y, and v in the range $[0, N - 1]$. To say that a complex function is *even* means that its real *and* imaginary parts are even, and similarly for an odd complex function.

TABLE 4.1 Some symmetry properties of the 2-D DFT and its inverse. $R(u, v)$ and $I(u, v)$ are the real and imaginary parts of $F(u, v)$, respectively. The term *complex* indicates that a function has nonzero real and imaginary parts.

1-D Examples of Symmetry

Property	$f(x)$		$F(u)$
3	$\{1 \;\; 2 \;\; 3 \;\; 4\}$	$\Leftrightarrow$	$\{(10)\,(-2+2j)\,(-2)\,(-2-2j)\}$
4	$j\{1 \;\; 2 \;\; 3 \;\; 4\}$	$\Leftrightarrow$	$\{(2.5j)\,(.5-.5j)\,(-.5j)\,(-.5-.5j)\}$
8	$\{2 \;\; 1 \;\; 1 \;\; 1\}$	$\Leftrightarrow$	$\{(5)\,(1)\,(1)\,(1)\}$
9	$\{0 \;\; -1 \;\; 0 \;\; 1\}$	$\Leftrightarrow$	$\{(0)\,(2j)\,(0)\,(-2j)\}$
10	$j\{2 \;\; 1 \;\; 1 \;\; 1\}$	$\Leftrightarrow$	$\{(5j)\,(j)\,(j)\,(j)\}$
11	$j\{0 \;\; -1 \;\; 0 \;\; 1\}$	$\Leftrightarrow$	$\{(0)\,(-2)\,(0)\,(2)\}$
12	$\{(4+4j)\,(3+2j)\,(0+2j)\,(3+2j)\}$	$\Leftrightarrow$	$\{(10+10j)\,(4+2j)\,(-2+2j)\,(4+2j)\}$
13	$\{(0+0j)\,(1+1j)\,(0+0j)\,(-1-j)\}$	$\Leftrightarrow$	$\{(0+0j)\,(2-2j)\,(0+0j)\,(-2+2j)\}$

2D Circular Convolution

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

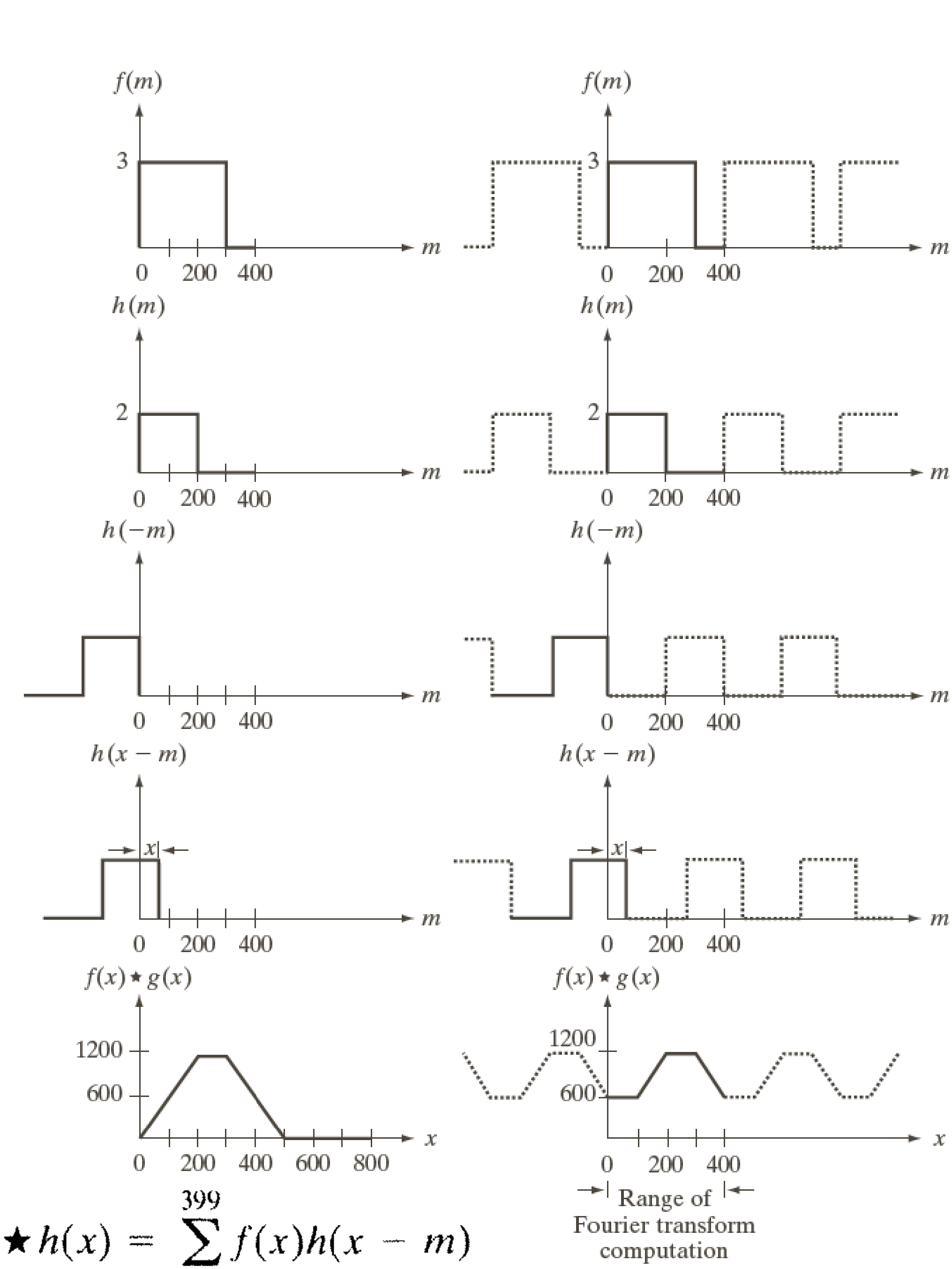
for $x = 0, 1, 2, \ldots, M-1$, and $x = 0, 1, 2, \ldots, N-1$

The 2-D convolution theorem is given by

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

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

- Because we are dealing with discrete quantities, computation of the Fourier transforms is carried out with a DFT algorithm.
- If we elect to compute the spatial convolution using the IDFT of the product of the two transforms, then the periodicity issues must be taken into account.



See Matlab code

$$f(x) \star h(x) = \sum_{m=0}^{399} f(x)h(x - m)$$

FIGURE 4.28 Left column: convolution of two discrete functions obtained using the approach discussed in Section 3.4.2. The result in (e) is correct. Right column: Convolution of the same functions, but taking into account the periodicity implied by the DFT. Note in (j) how data from adjacent periods produce wraparound error, yielding an incorrect convolution result. To obtain the correct result, function padding must be used.

Zero Padding in 2D Image Filtering

$$f_p(x, y) = \begin{cases} f(x, y) & 0 \le x \le A-1 \quad \text{and} \quad 0 \le y \le B-1 \\ 0 & A \le x \le P \quad \text{or} \quad B \le y \le Q \end{cases}$$

$$h_p(x, y) = \begin{cases} h(x, y) & 0 \le x \le C-1 \quad \text{and} \quad 0 \le y \le D-1 \\ 0 & C \le x \le P \quad \text{or} \quad D \le y \le Q \end{cases}$$

$$P \ge A + C - 1 \qquad\qquad Q \ge B + D - 1$$

- The resulting padded images are of size $P \times Q$.
- As a rule of thumb, DFT algorithms tend to execute faster with arrays of even size, so it is good practice to select P and Q as the smallest even integers that satisfy the preceding equations.
- If the two arrays are of the same size, then P and Q are selected as twice the array size.

See Matlab code “paddedsize.m”

See Matlab code

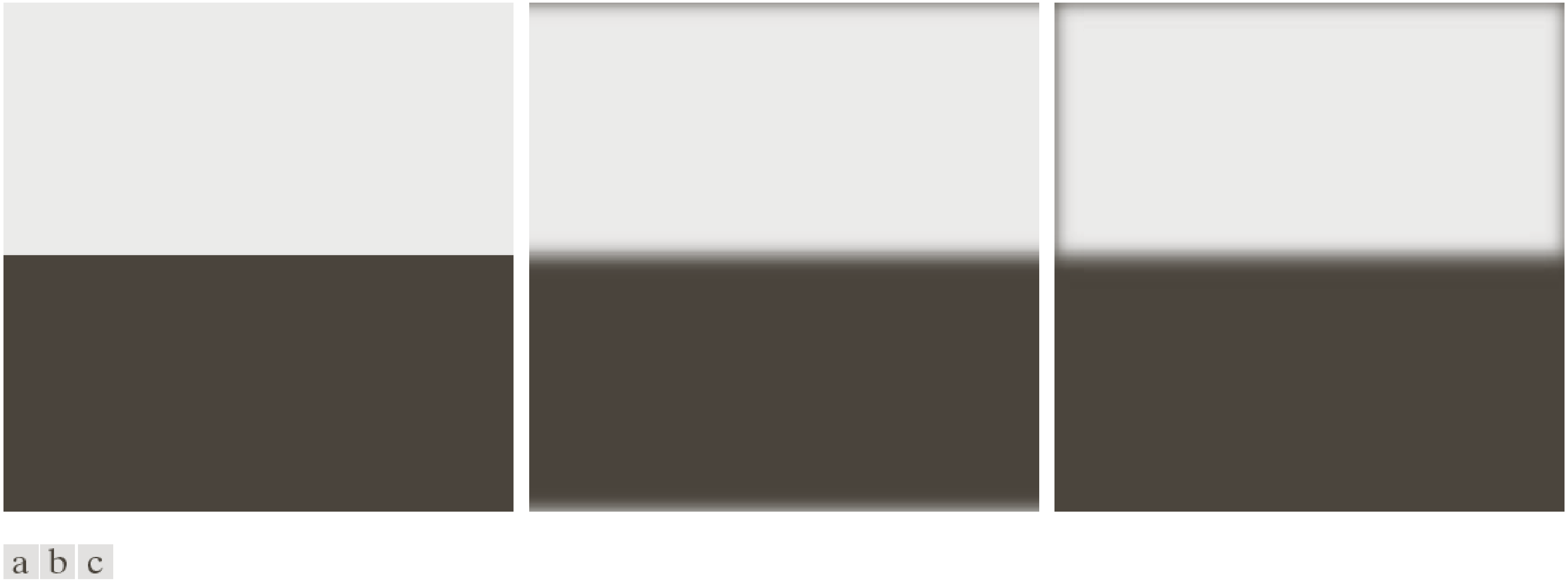
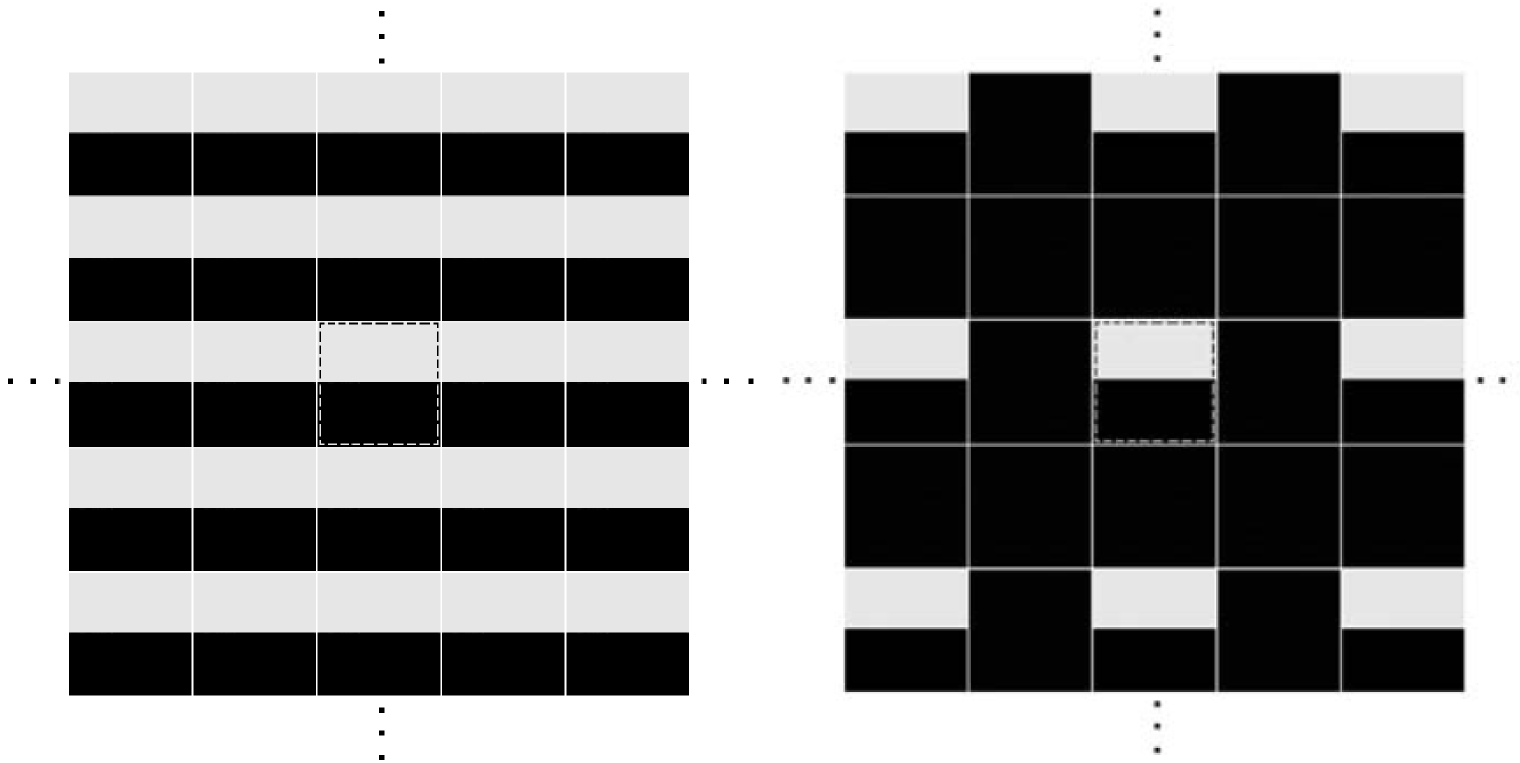


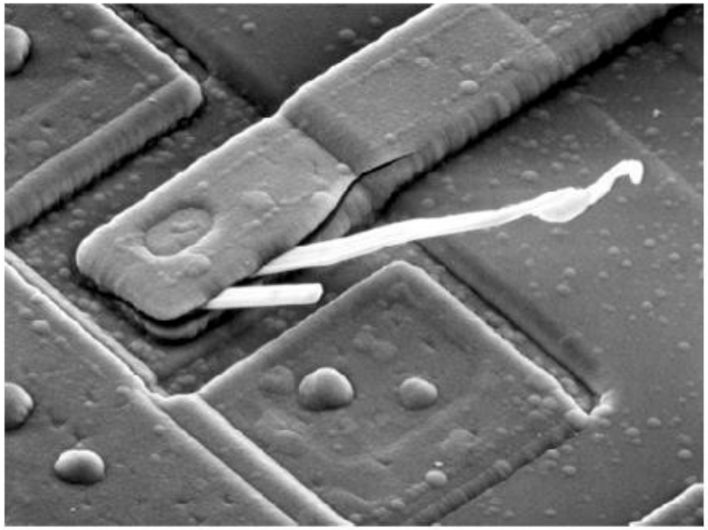
FIGURE 4.32 (a) A simple image. (b) Result of blurring with a Gaussian lowpass filter without padding. (c) Result of lowpass filtering with padding. Compare the light area of the vertical edges in (b) and (c).



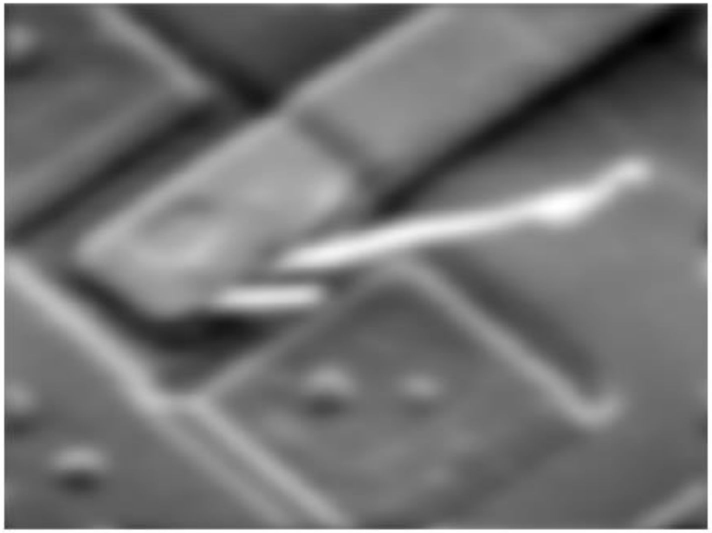
a b

FIGURE 4.33 2-D image periodicity inherent in using the DFT. (a) Periodicity without image padding. (b) Periodicity after padding with 0s (black). The dashed areas in the center correspond to the image in Fig. 4.32(a). (The thin white lines in both images are superimposed for clarity; they are not part of the data.)

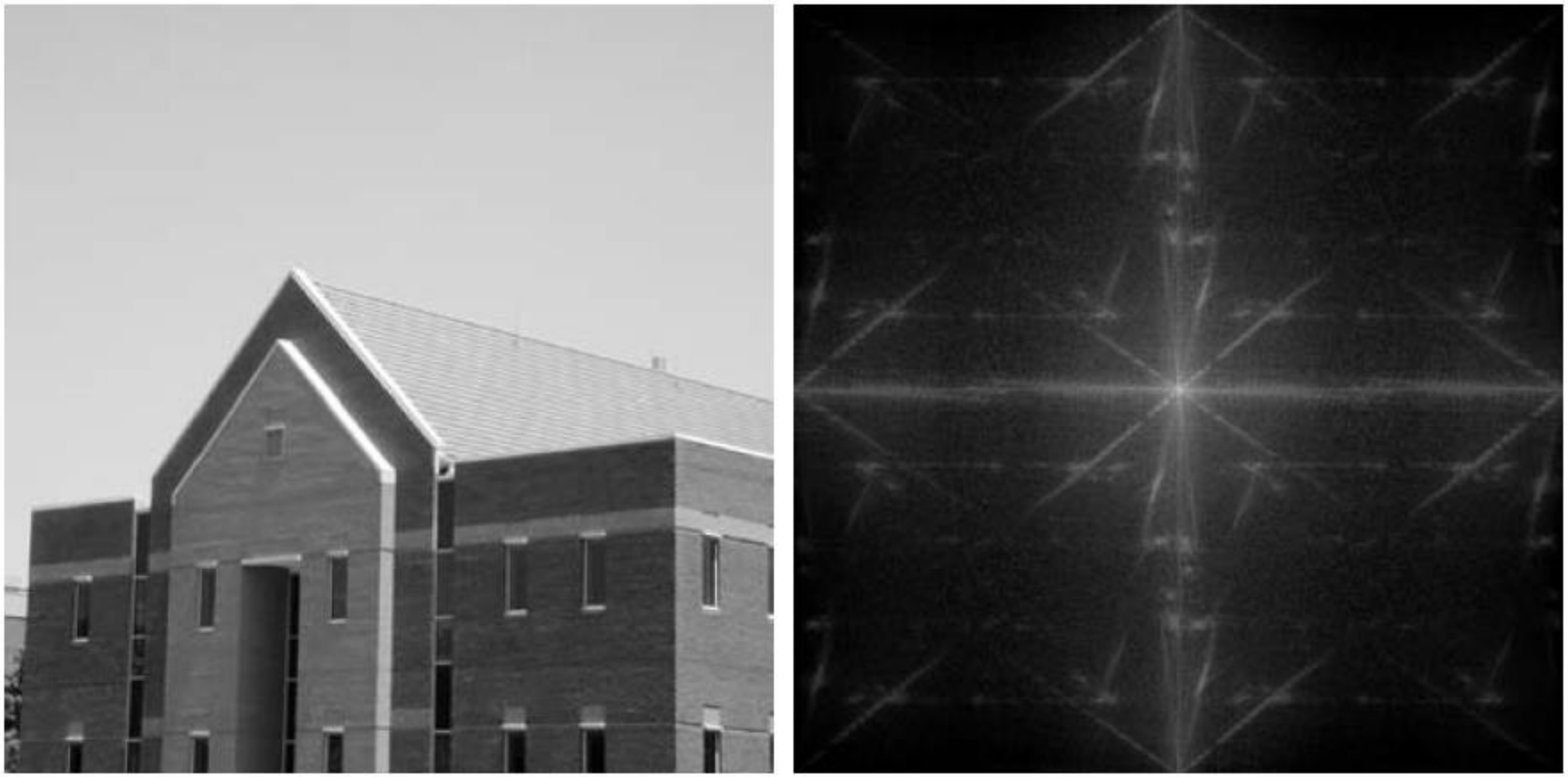
Filtering in Frequency Domain



See Matlab code "4_36.m"
for Gaussian Filtering



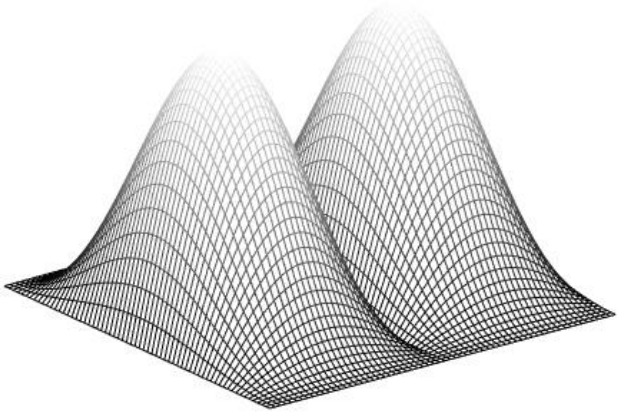
Filtering in Spatial and Frequency Domains

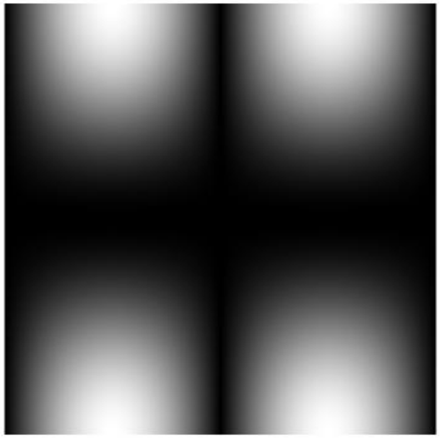


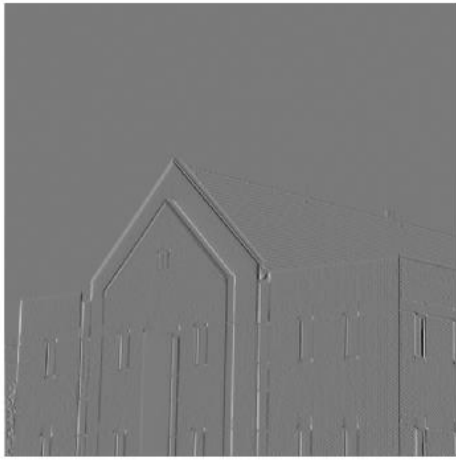
a b

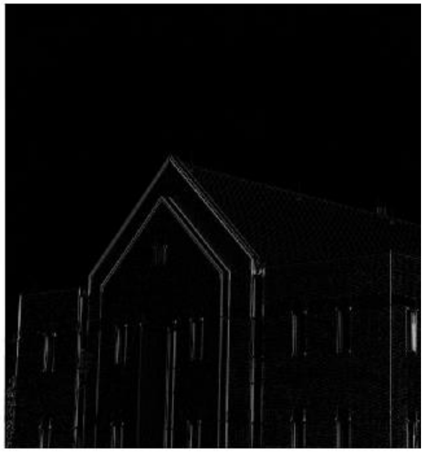
FIGURE 4.38
(a) Image of a building, and (b) its spectrum.

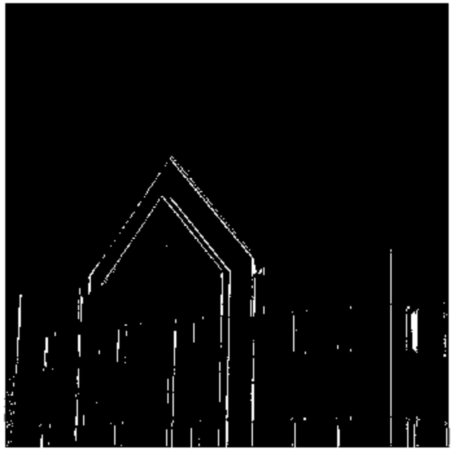
See Matlab code
for edge detection using Sobel mask











Generating Filters Directly in Frequency Domain

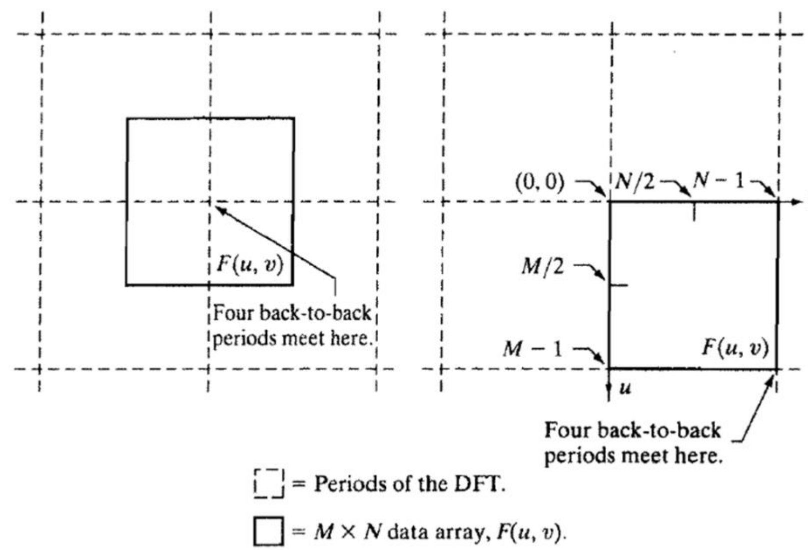
- Implement filter functions directly in the frequency domain.
- Focus on circularly symmetric filters that are specified as various functions of distance from the origin of the transform
- Begin with smoothing (low-pass) filters
- Then discuss sharpening (high-pass) filters

Meshgrid Arrays for Implementing Filters in the Frequency Domain

- FFT computations in Matlab assume that the origin of the transform is at top, left of the frequency rectangle, our distance computations are with respect to that point.
- The data can be rearranged for visualization purposes by using *fftshift*.
- The M-function, *dftuv*, provides the necessary meshgrid array in distance computations for *fft2* or *ifft2*, so no rearrangement of the data is required.

```
>> [U, V] = dftuv(5, 5);
% distance 0 at top left, larger distance
% at the center of the frequency rectangle
>> D = U.^2 + V.^2

>> fftshift(D)
% The array is symmetric about the center
```



Lowpass Frequency Domain Filters

- Ideal lowpass filter

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

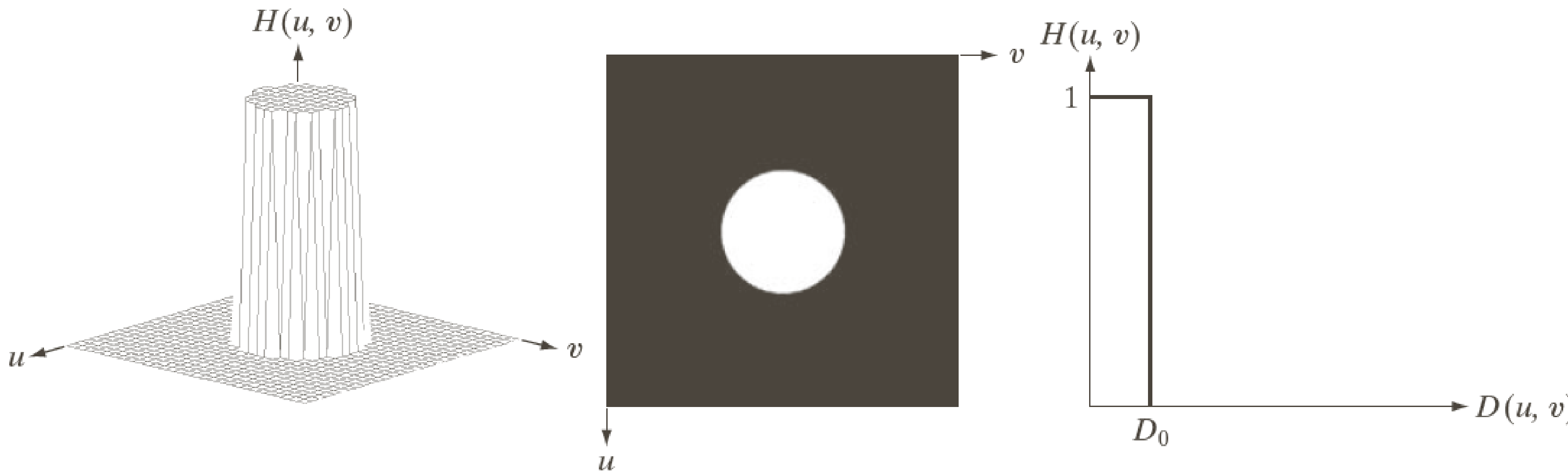
- Butterworth lowpass filter

$$H(u,v) = \frac{1}{1+\left[\frac{D(u,v)}{D_0}\right]^{2n}}$$, where D_0 is the cutoff freq.

- Gaussian lowpass filter

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

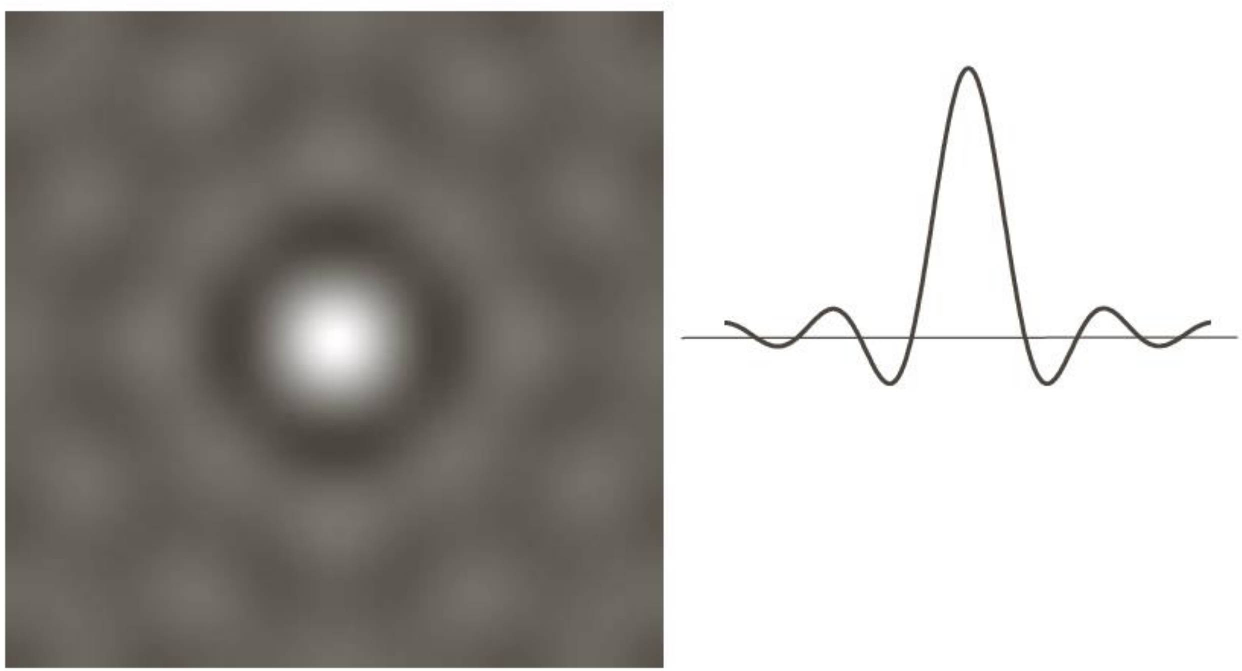
ILPF Transfer Function



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.

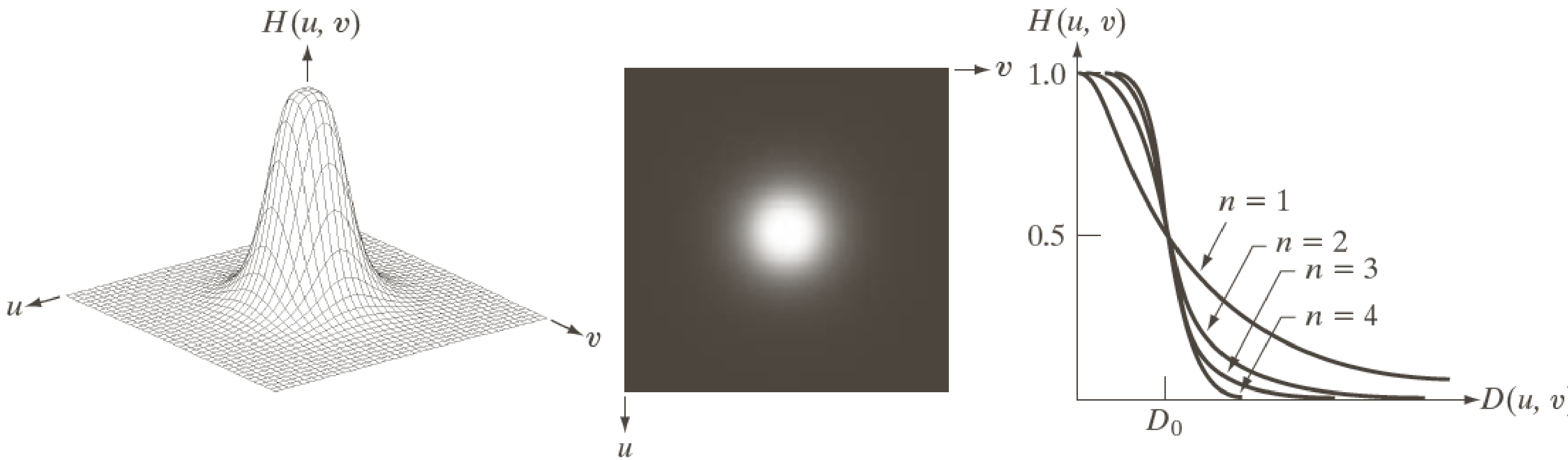
Ideal Lowpass Filter in Spatial Domain



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 Transfer Function



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.

BLPF in Spatial Domain

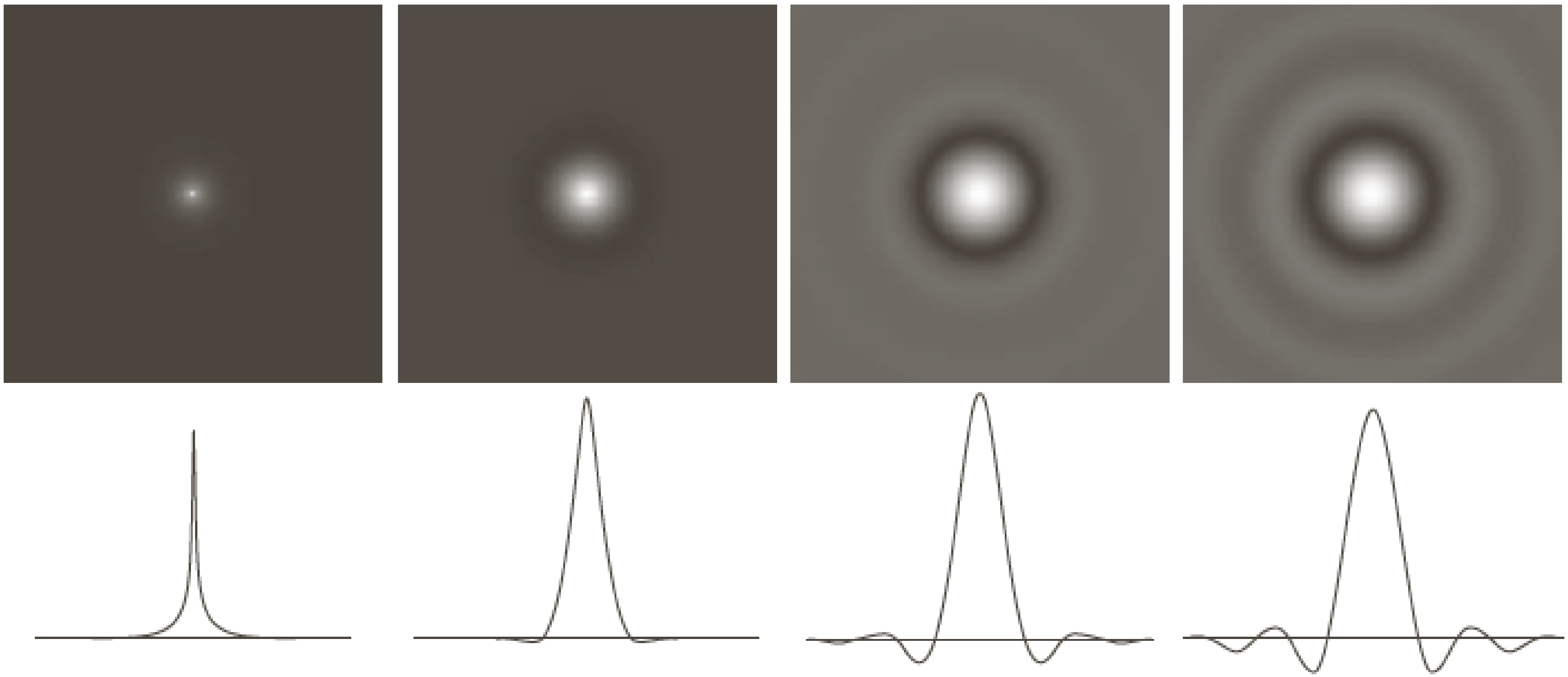


FIGURE 4.46 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding intensity profiles through the center of the filters (the size in all cases is 1000×1000 and the cutoff frequency is 5). Observe how ringing increases as a function of filter order.

Gaussian Filter Transfer Function

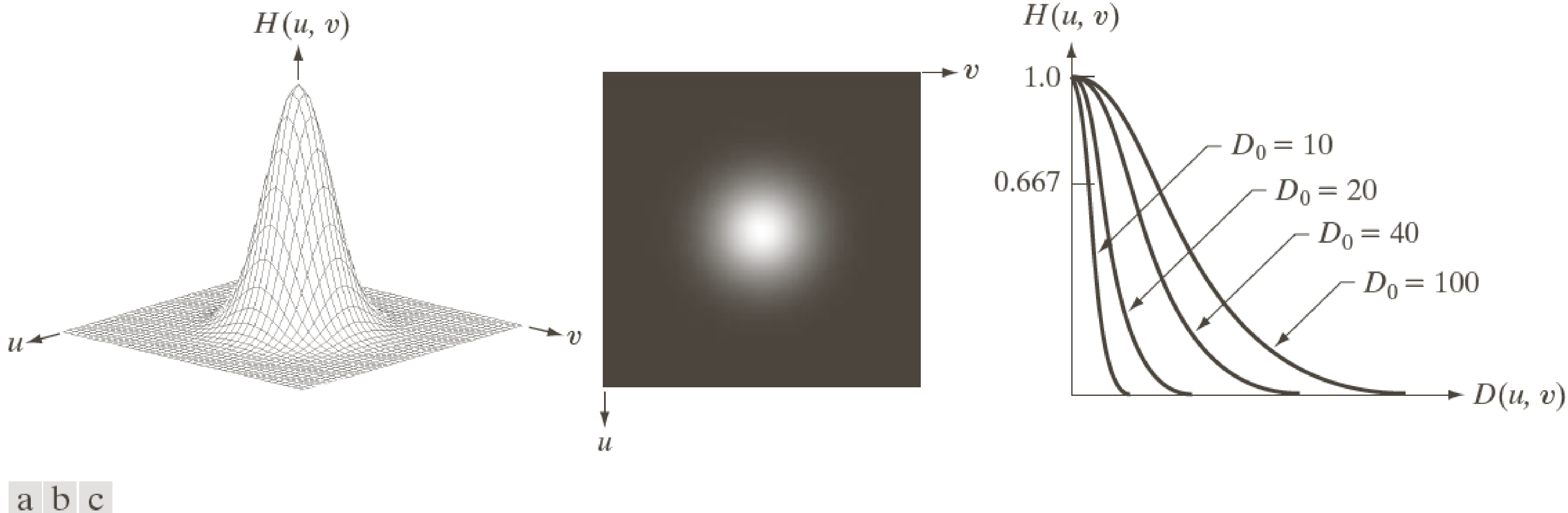
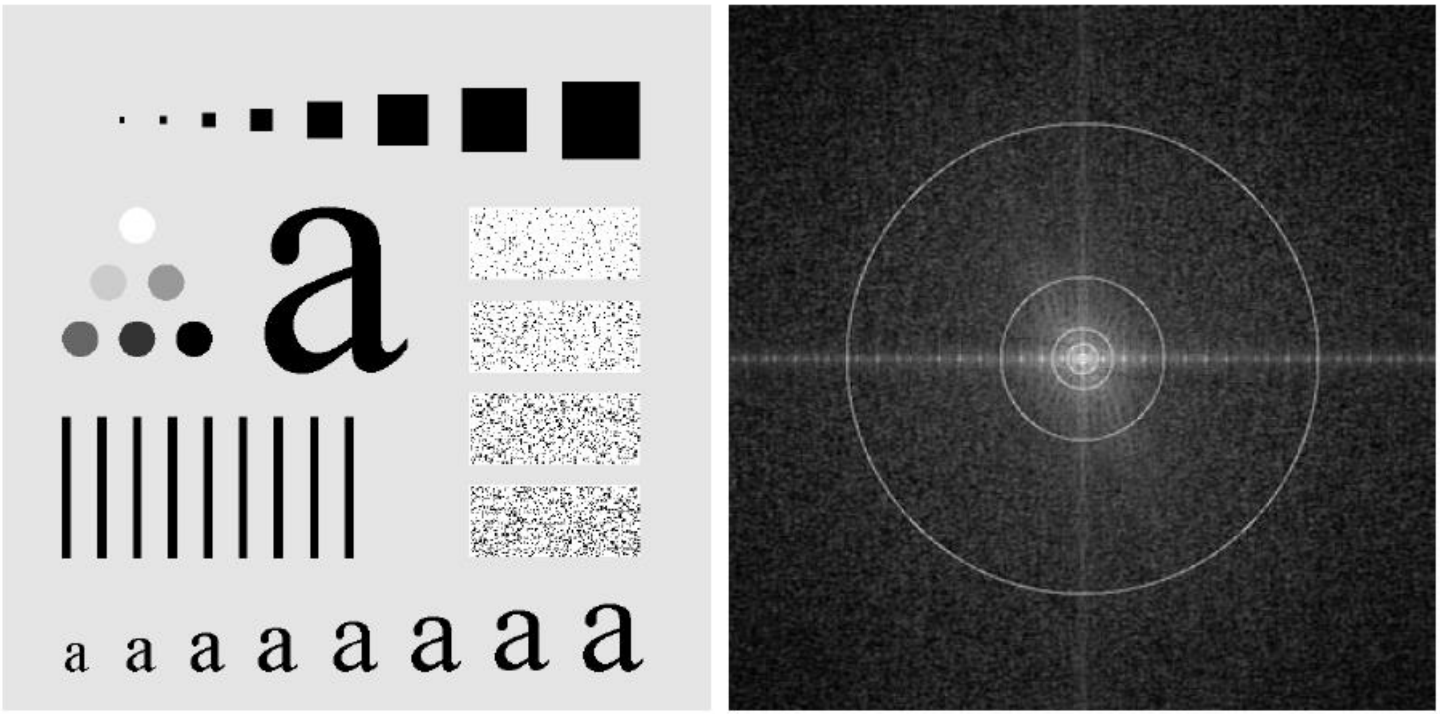


FIGURE 4.47 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

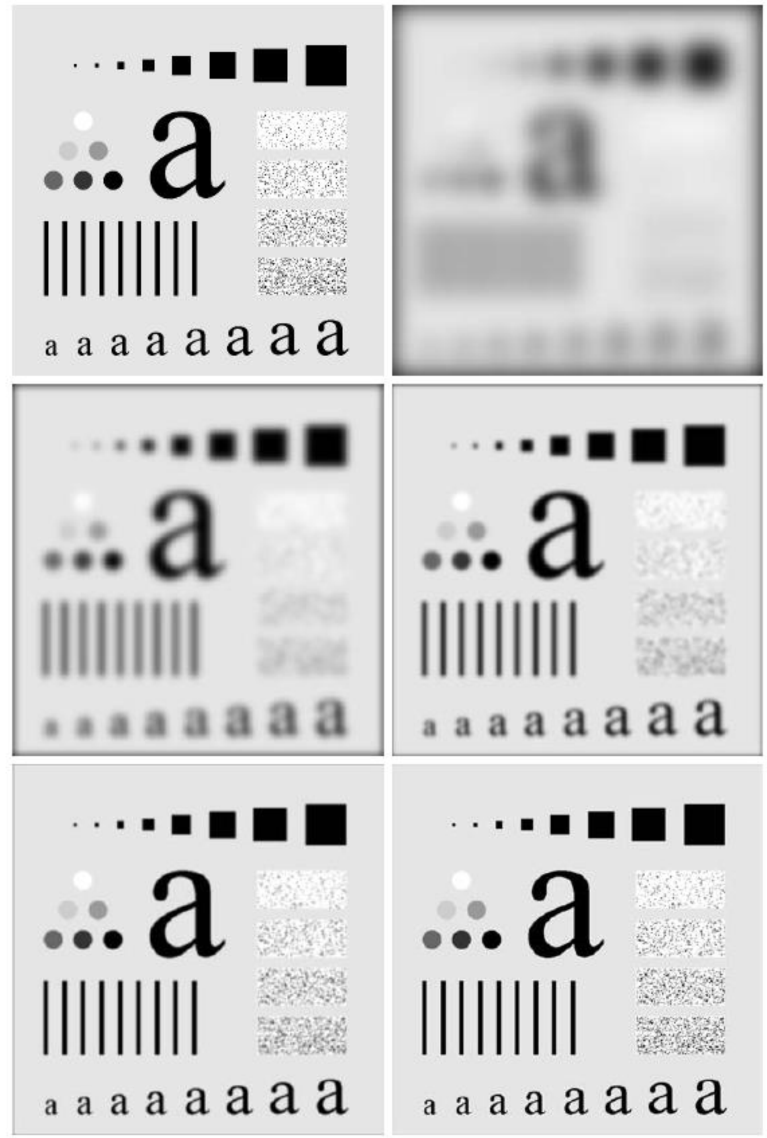
Example: Gaussian Filter



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.

See Matlab code “4_48.m”
for Gaussian Filtering



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.

M-function for LPF Transfer Function Generation

- H = lpfilter (type, M, N, D0, n)
- Centered transferred function

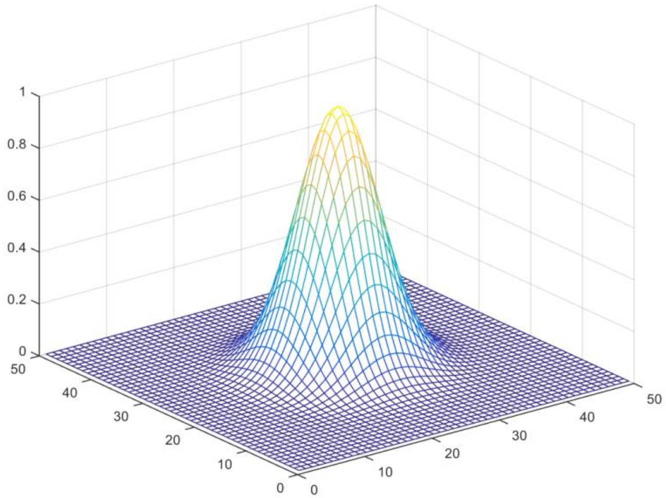
  H = lpfilter_center (type, M, N, D0, n)

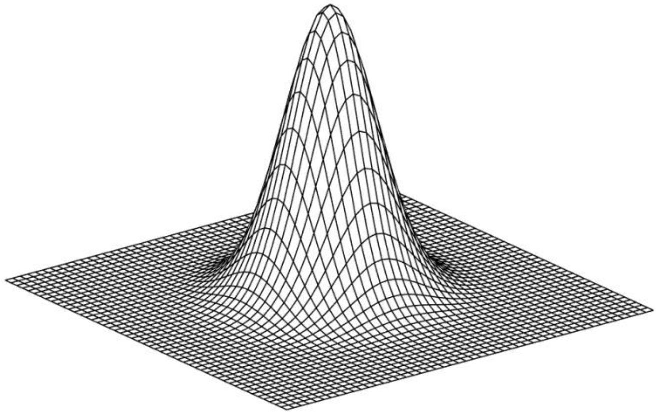
See Matlab code "4_36.m"
for Gaussian Filtering

- lpfilter can be used for generating highpass filters.

Visualization of Filter Transfer Functions

- Wireframe





- Surface plot
- See Matlab code

Highpass Filters

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

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

[H] = hpfilter (type,M,N,D0,n)

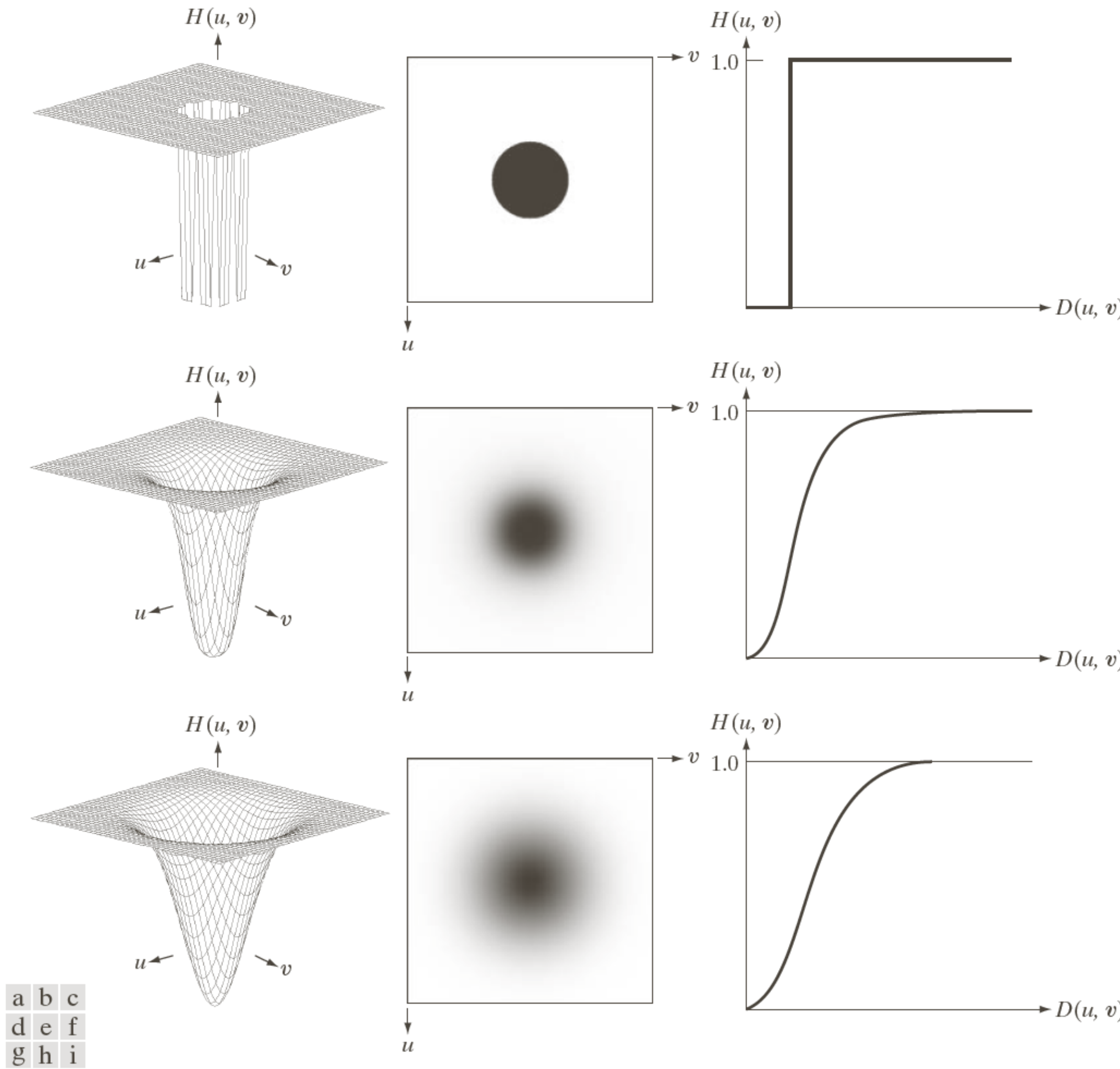
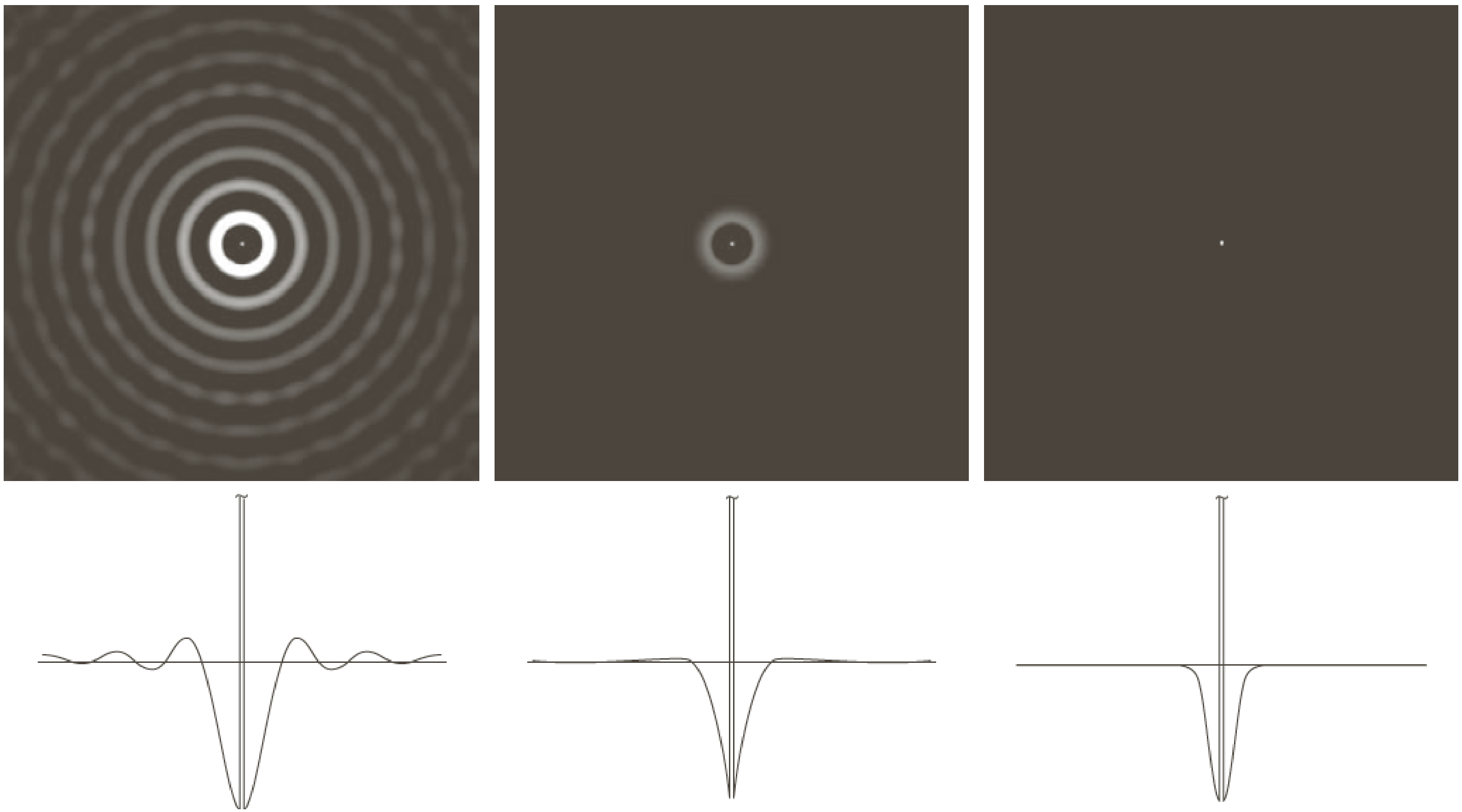


FIGURE 4.52 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Spatial Domain Representation



a b c

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.

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.

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.

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.

HPF followed by Thresholding

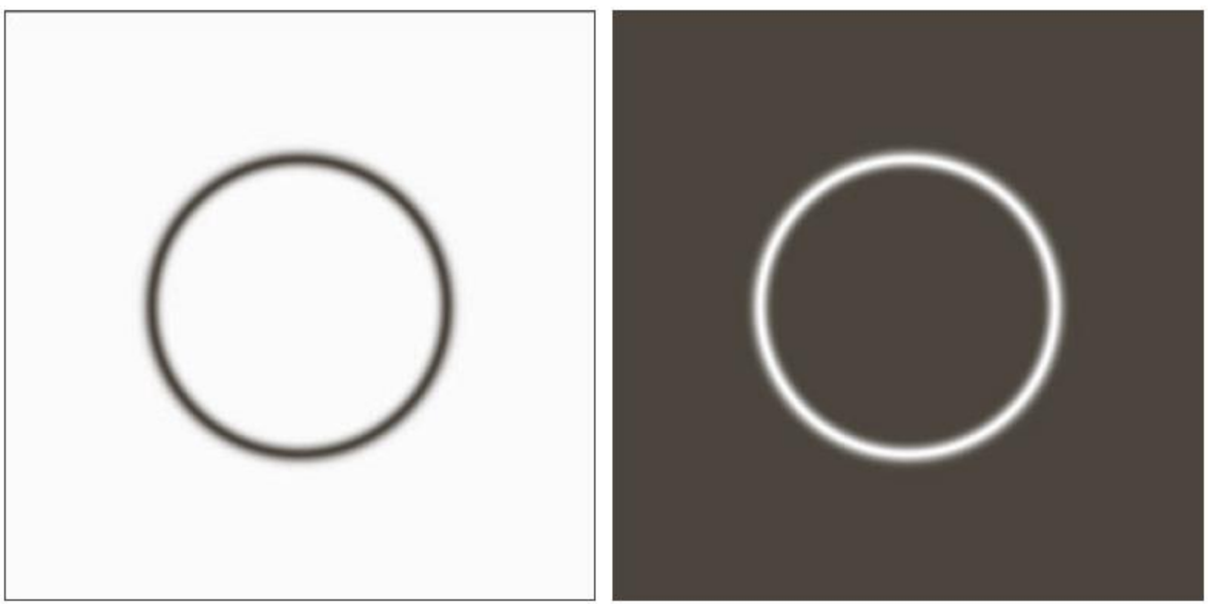
a b c

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.)

Selective Filter

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



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.