بسمه‌تعالي

Digital Image Processing

Image Enhancement in the Frequency Domain (Chapter 5)

Fourier Series

Any function that periodically repeats itself can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficients. This sum is called a Fourier series.

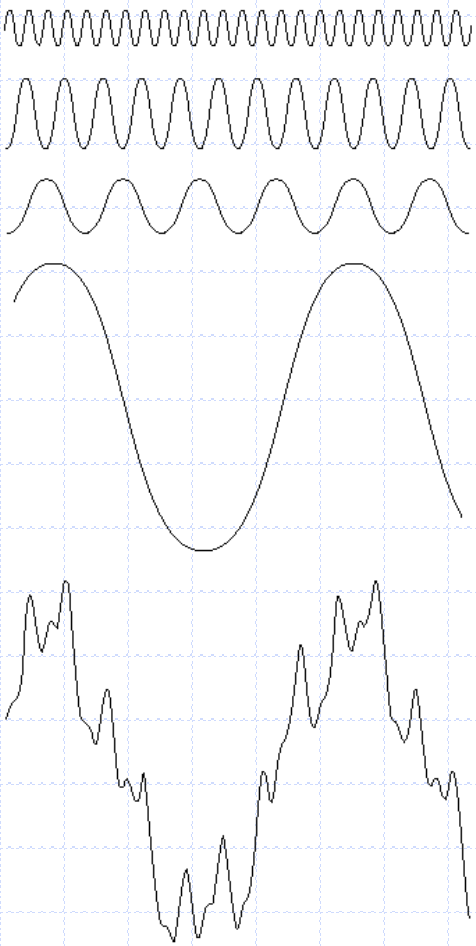


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Fourier Series

$$g(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos n w_0 t + b_n \sin n w_0 t$$

$$a_0 = \frac{1}{T_0} \int_{t_0}^{t_0+T_0} g(t)\,dt$$

$$a_n = \frac{2}{T_0} \int_{t_0}^{t_0+T_0} g(t) \cos n w_0 t \; dt$$

$$b_n = \frac{2}{T_0} \int_{t_0}^{t_0+T_0} g(t) \sin n w_0 t \; dt$$

Fourier Transform

A function that is not periodic but the area under its curve is finite can be expressed as the integral of sines and/or cosines multiplied by a weighing function. The formulation in this case is Fourier transform.

Continuous One-Dimensional Fourier Transform and Its Inverse

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

Where $j = \sqrt{-1}$

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

- (u) is the frequency variable.
- $F(u)$ is composed of an infinite sum of sine and cosine terms and…
- Each value of u determines the frequency of its corresponding sine-cosine pair.

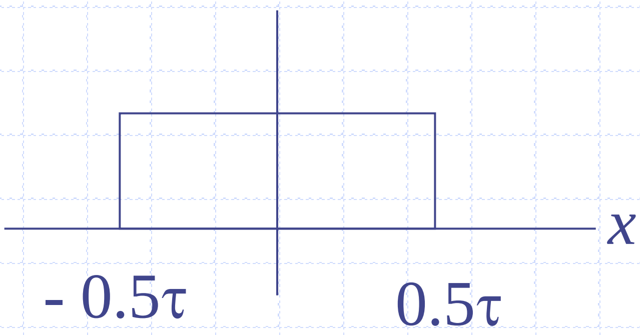
Continuous One-Dimensional Fourier Transform and Its Inverse

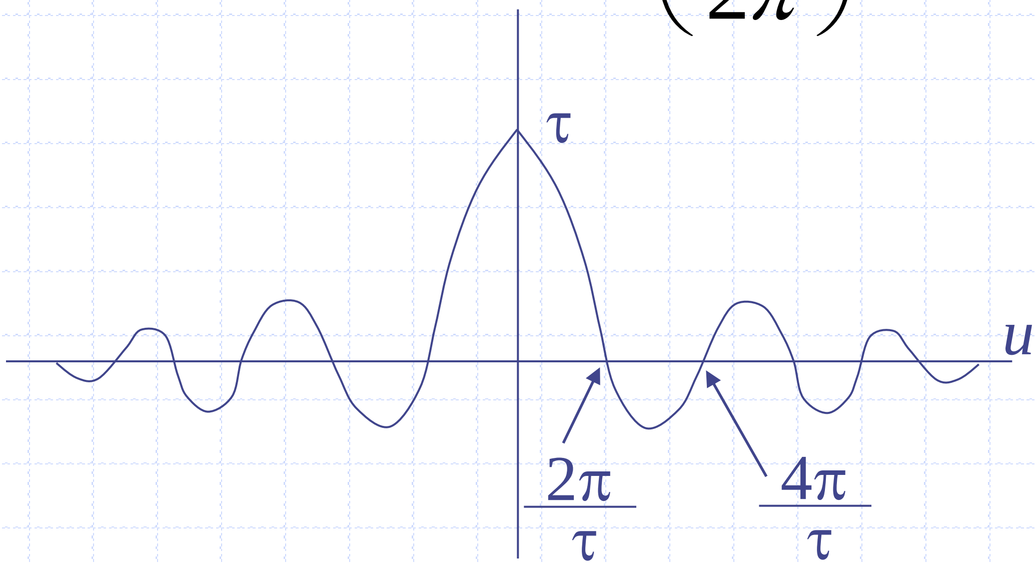
Example

Find the Fourier transform of a gate function Π(t) defined by

$$\Pi(x) = \begin{cases} 1 & |x| < \dfrac{1}{2}\tau \\ 0 & |x| < \dfrac{1}{2}\tau \end{cases}$$

$$F(u) = \tau \operatorname{sinc}\left(\frac{u\tau}{2\pi}\right)$$



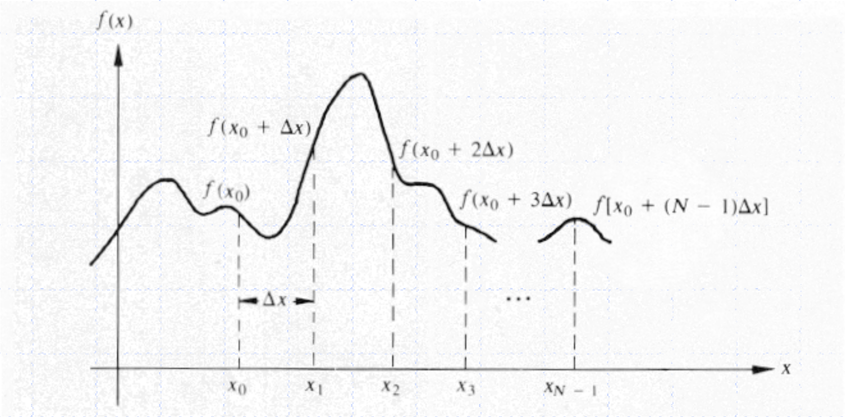


Discrete One-Dimensional Fourier Transform and Its Inverse

- **A continuous function f(x) is discretized into a sequence:**

$$\{f(x_0), f(x_0+\Delta x), f(x_0+2\Delta x), \ldots, f(x_0+[N-1]\Delta x)\}$$

by taking N or M samples Δx units apart.



Discrete One-Dimensional Fourier Transform and Its Inverse

- **Where x assumes the discrete values (0,1,2,3,…,M-1) then**

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

- The sequence {f(0),f(1),f(2),…f(M-1)} denotes any M uniformly spaced samples from a corresponding continuous function.

Discrete One-Dimensional Fourier Transform and Its Inverse

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi u \frac{x}{M}}$$

u =[0,1,2, …, M-1]

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) \left[\cos 2\pi u \frac{x}{M} - j \sin 2\pi u \frac{x}{M} \right]$$

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi \frac{u}{M} x}$$

x =[0,1,2, …, M-1]

Discrete One-Dimensional Fourier Transform and Its Inverse

- The values u = 0, 1, 2, …, M-1 correspond to samples of the continuous transform at values 0, Δu, 2Δu, …, (M-1)Δu.

i.e. *F(u)* represents *F(uΔu)*, where:

$$\Delta u = \frac{1}{M\Delta x}$$

Discrete One-Dimensional Fourier Transform and Its Inverse

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

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

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

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

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

Discrete One-Dimensional Fourier Transform and Its Inverse

- **The square of the spectrum**

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

is referred to as the Power Spectrum of f(x) (spectral density).

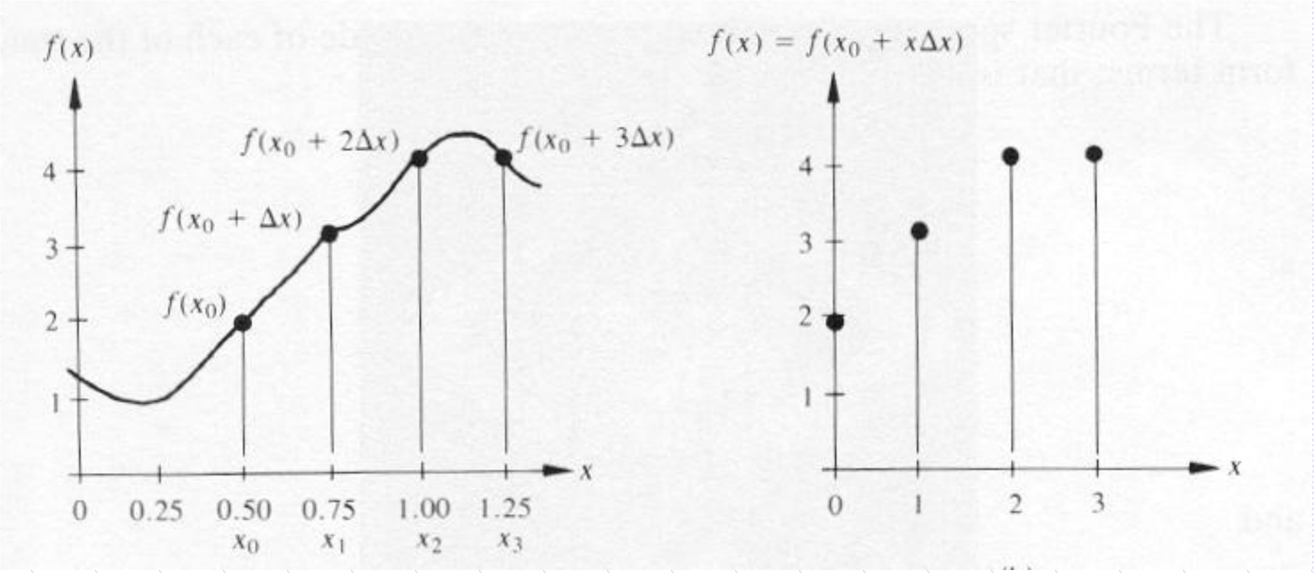
Discrete 2-Dimensional Fourier Transform

- **Fourier spectrum:** $$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$$

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

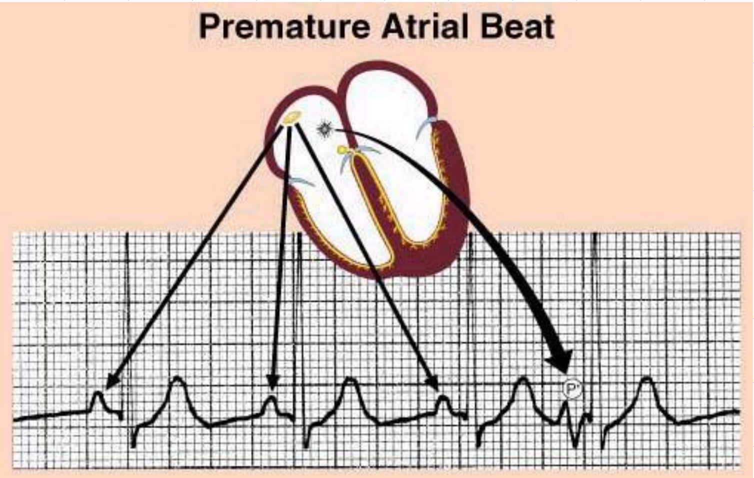
- **Power spectrum:** $$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

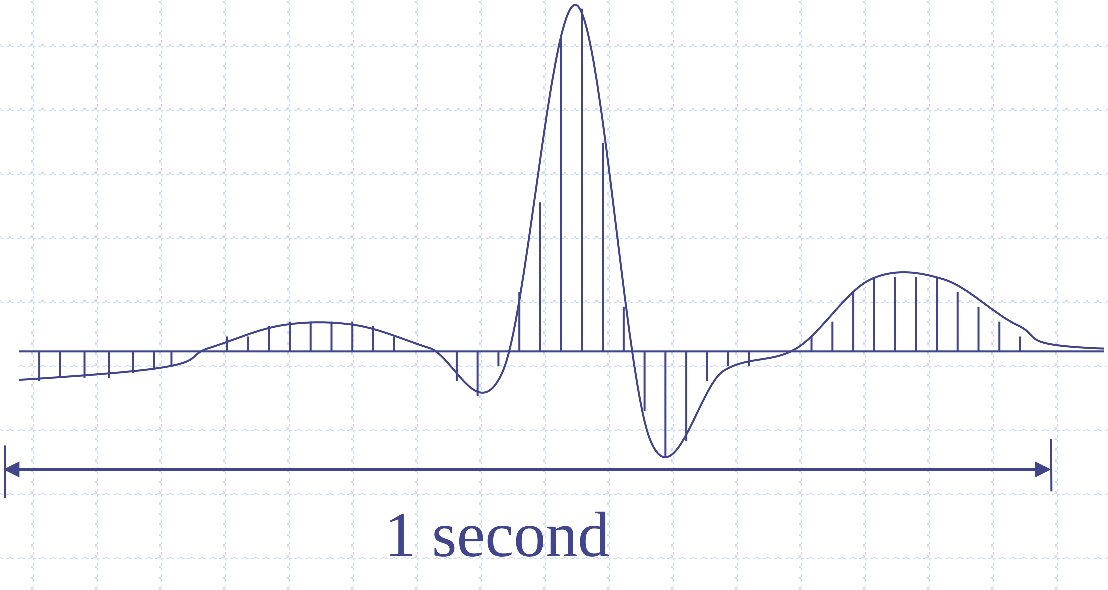
Discrete One-Dimensional Fourier Transform and Its Inverse



Time and Frequency Resolution and Sampling

F_{max} = 100 Hz
What is the sampling rate (Nyquist)?
What is the time resolution?
What is the frequency resolution?
What if we take samples for two seconds with the Nyquist sampling rate?





Discrete Two-Dimensional Fourier Transform and Its Inverse

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

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

$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$$

Fourier Spectrum

Discrete Two-Dimensional Fourier Transform and Its Inverse

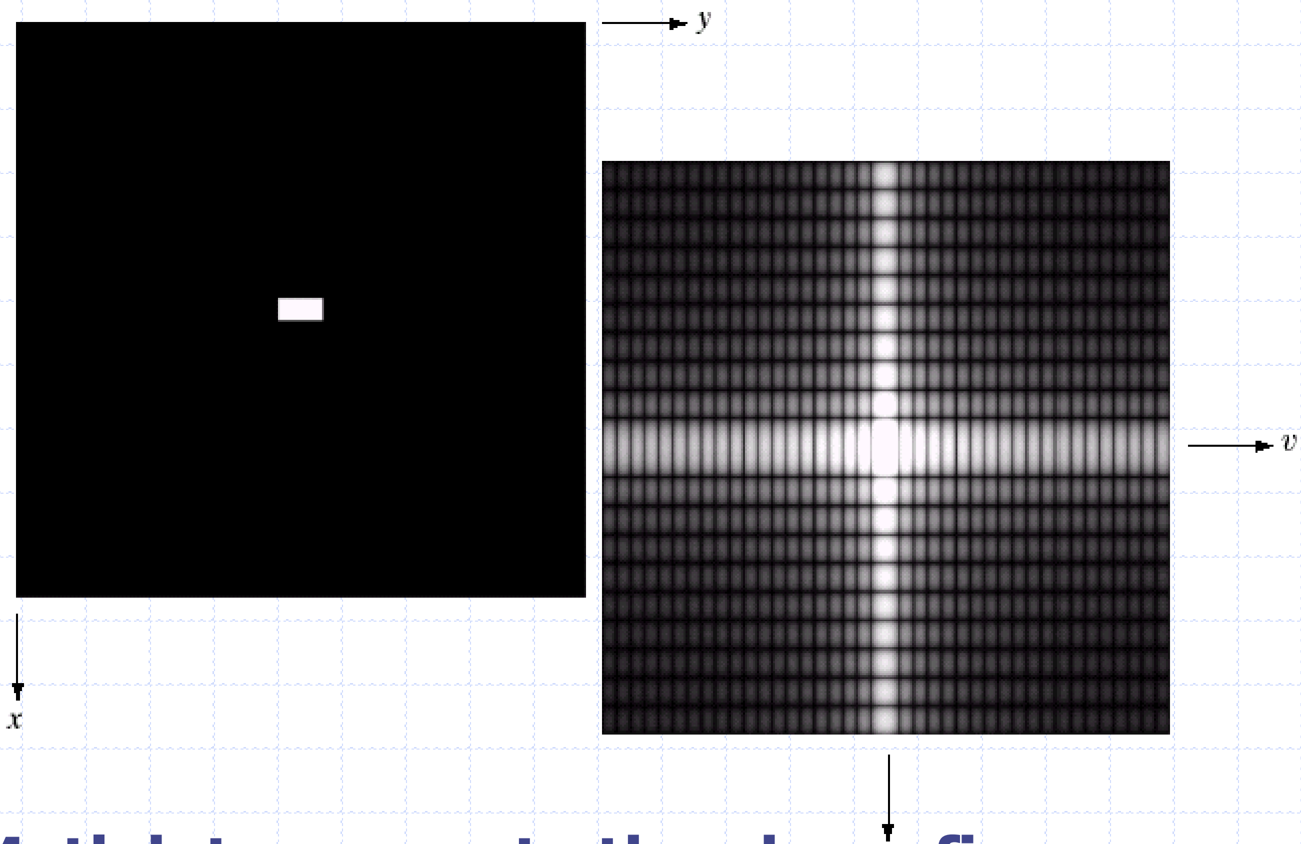
$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

$F(0,0)$ is the average intensity of an image

Discrete Two-Dimensional Fourier Transform and Its Inverse

a b

FIGURE 4.3
(a) Image of a 20 × 40 white rectangle on a black background of size 512 × 512 pixels.
(b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.



Use Matlab to generate the above figures

Frequency Shifting Property of the Fourier Transform

If

$$g(t) \leftrightarrow G(\omega)$$

then

$$g(t)e^{j\omega_0 t} \leftrightarrow G(\omega - \omega_0)$$

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

Frequency Shifting Property of the Fourier Transform

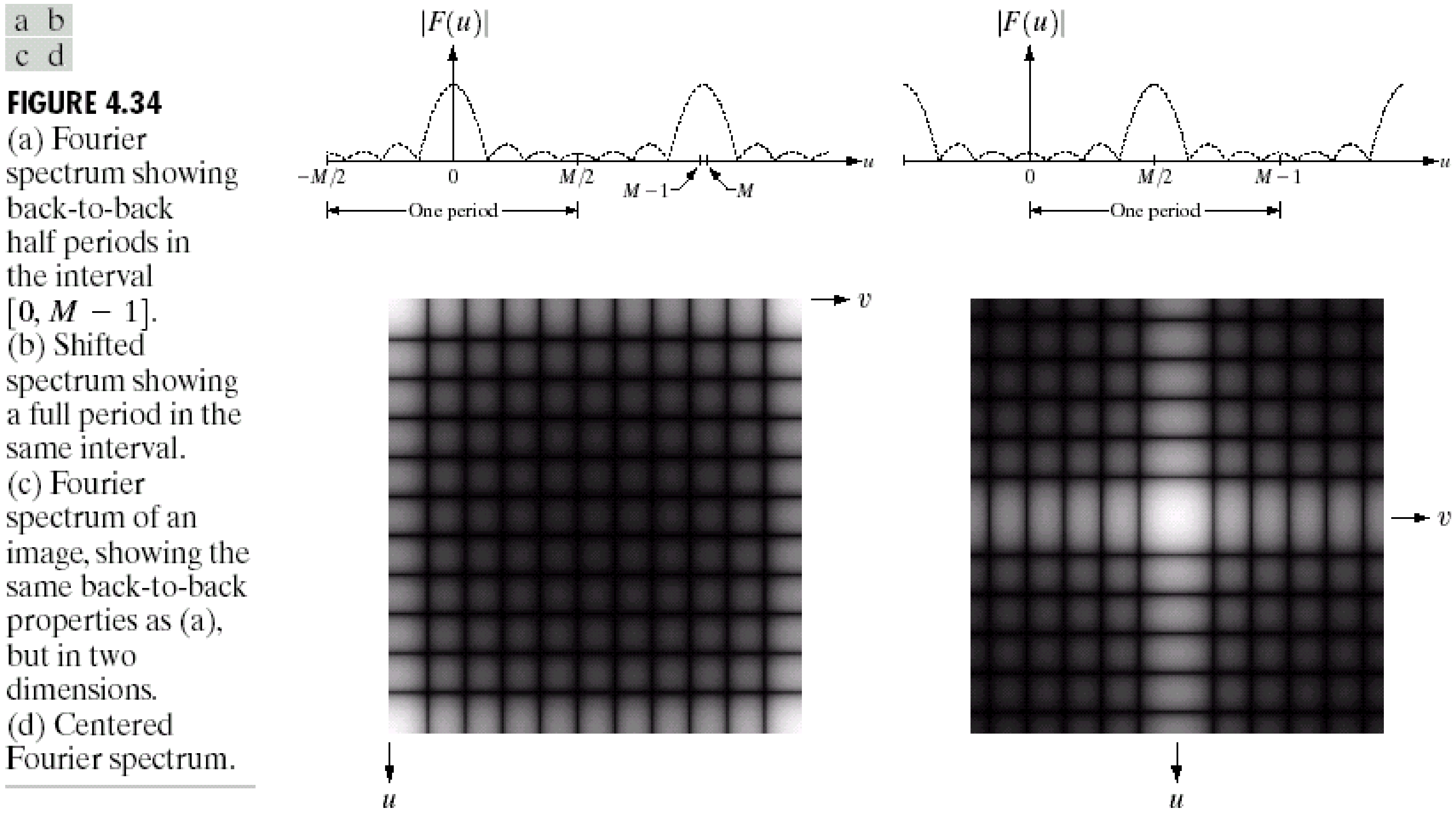


FIGURE 4.34 (a) Fourier spectrum showing back-to-back half periods in the interval $[0, M-1]$. (b) Shifted spectrum showing a full period in the same interval. (c) Fourier spectrum of an image, showing the same back-to-back properties as (a), but in two dimensions. (d) Centered Fourier spectrum.

Basic Filtering in the Frequency Domain using Matlab

```
function Normalized_DFT = Img_DFT(img)
img=double(img);     % So mathematical operations can be conducted on
                      %  the image pixels.
[R,C]=size(img);
for r = 1:R              % To phase shift the image so the DFT will be
   for c=1:C             %    centered on the display monitor
        phased_img(r,c)=(img(r,c))*(-1)^(r+c);
    end
end
fourier_img = fft2(phased_img);            %Discrete Fourier Transform
mag_fourier_img = abs(fourier_img );    % Magnitude of DFT
Log_mag_fourier_img = log10(mag_fourier_img +1);
Max = max(max(Log_mag_fourier_img ));
Normalized_DFT = (Log_ mag_fourier_img )*(255/Max);
imshow(uint8(Normalized_DFT))
```

Basic Filtering in the Frequency Domain

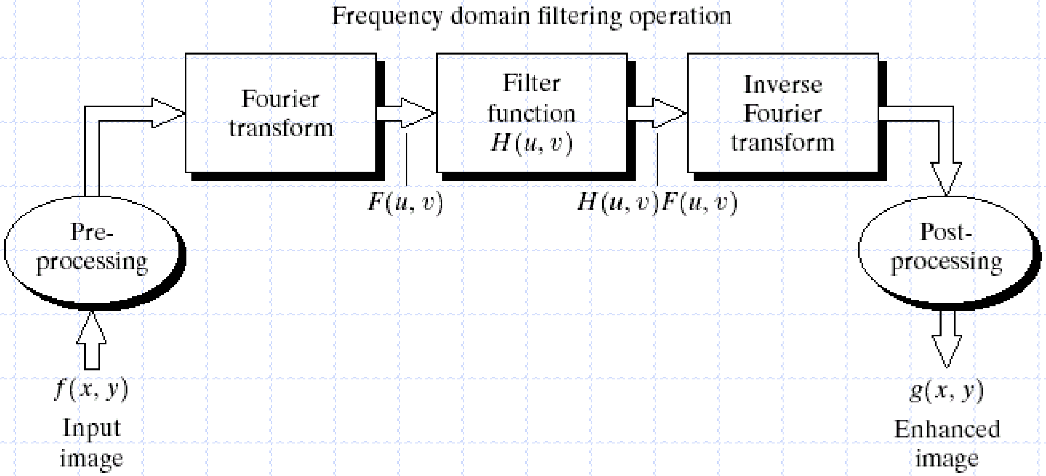
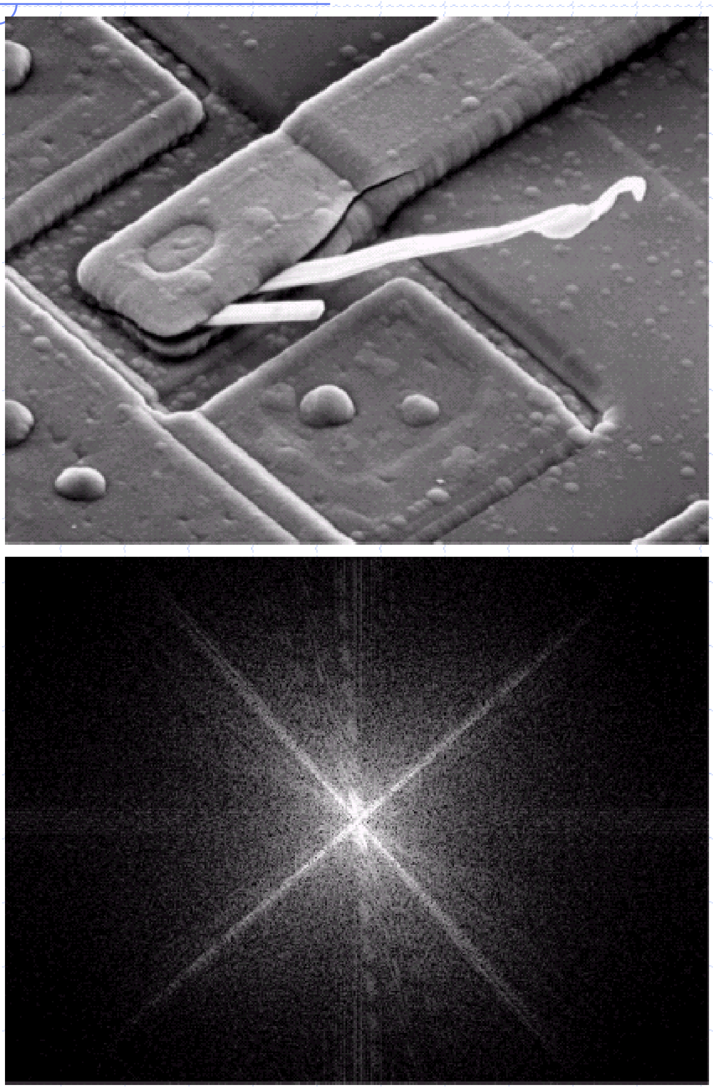


FIGURE 4.5 Basic steps for filtering in the frequency domain.

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

An image and its Frequency information

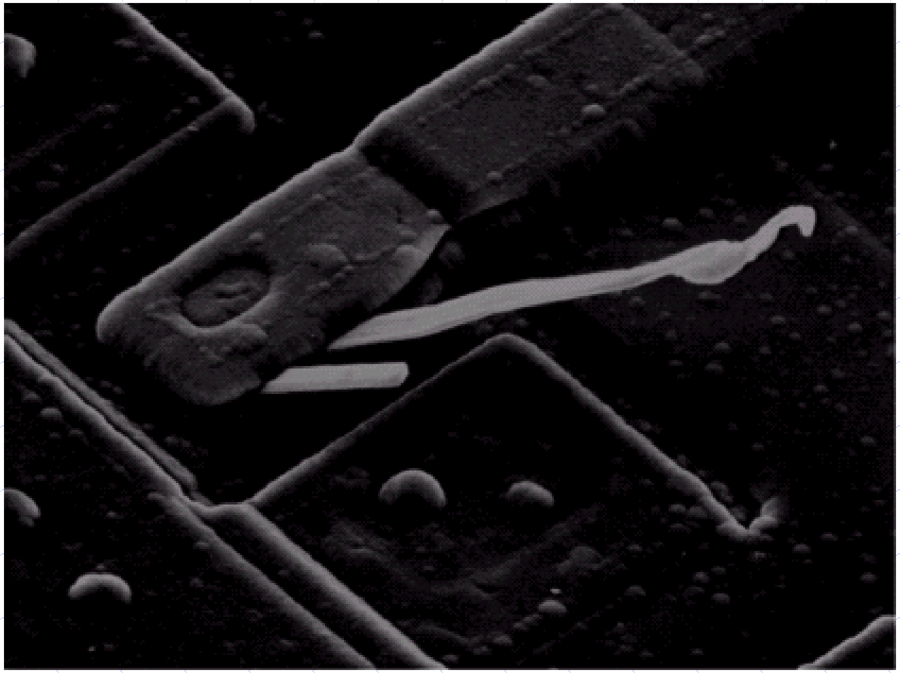


a
b

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.)

Filtering out the DC Frequency Component

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.



Notch Filter

$$H(u,v) = \begin{cases} 0 & \text{if } (u,v) = (M/2, N/2) \\ 1 & \text{otherwise} \end{cases}$$

Low-pass and High-pass Filters

Low Pass Filter attenuate high frequencies while "passing" low frequencies.

High Pass Filter attenuate low frequencies while "passing" high frequencies.

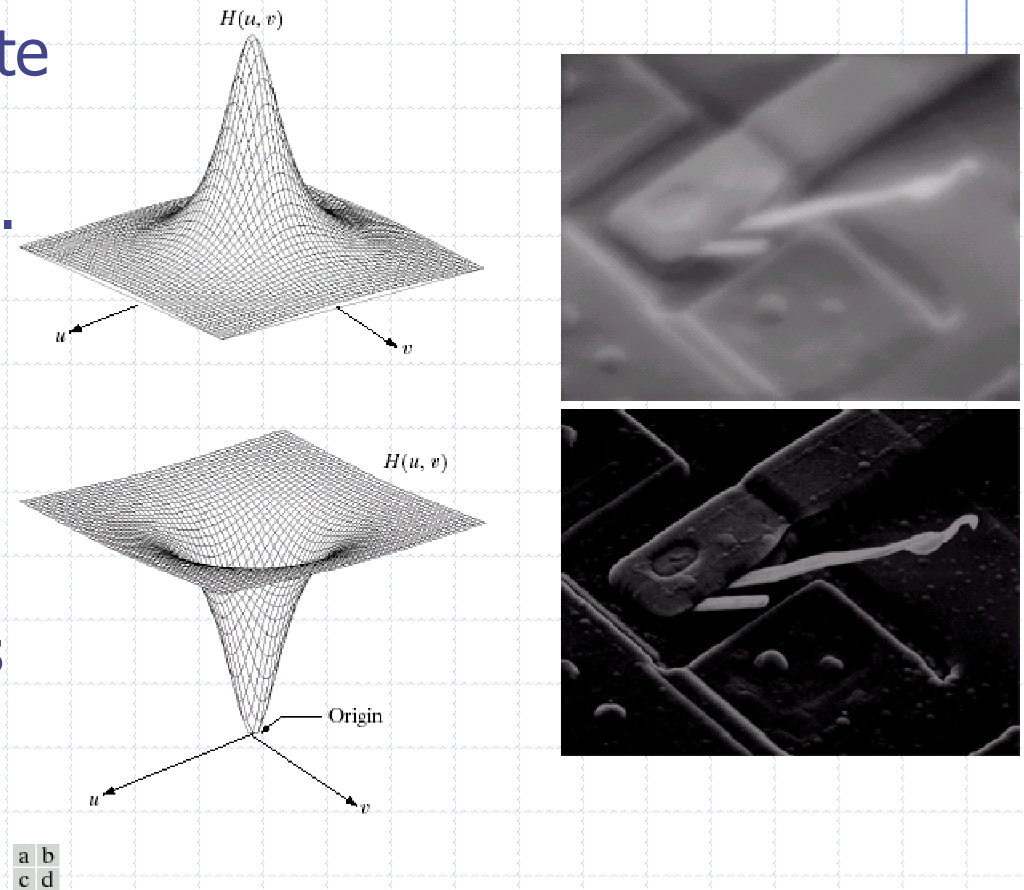
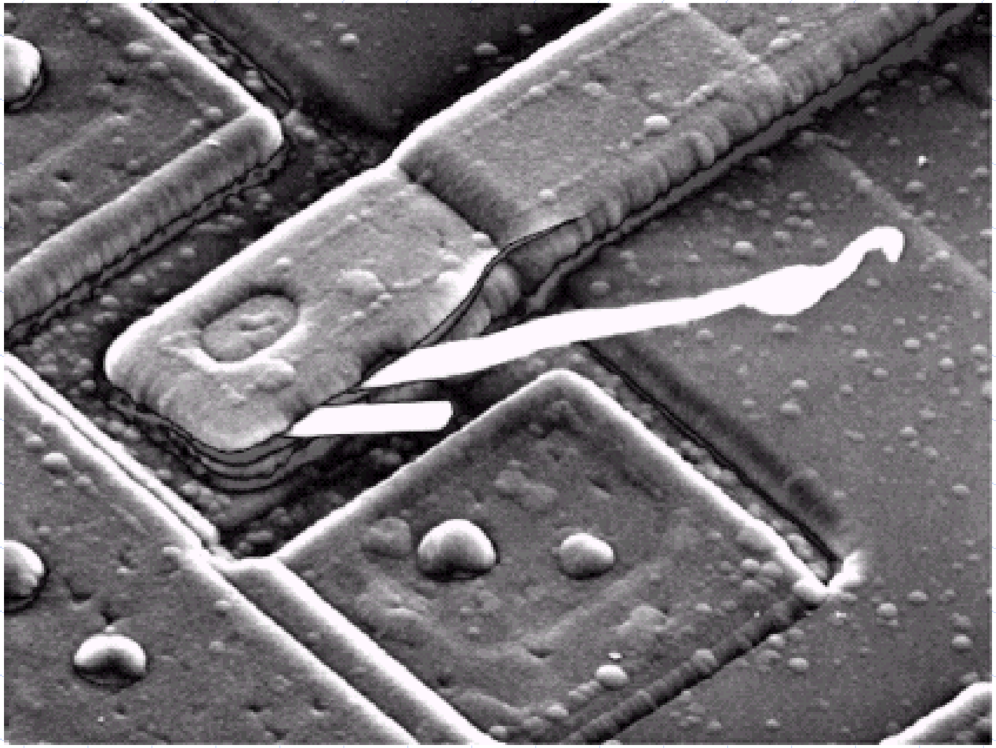


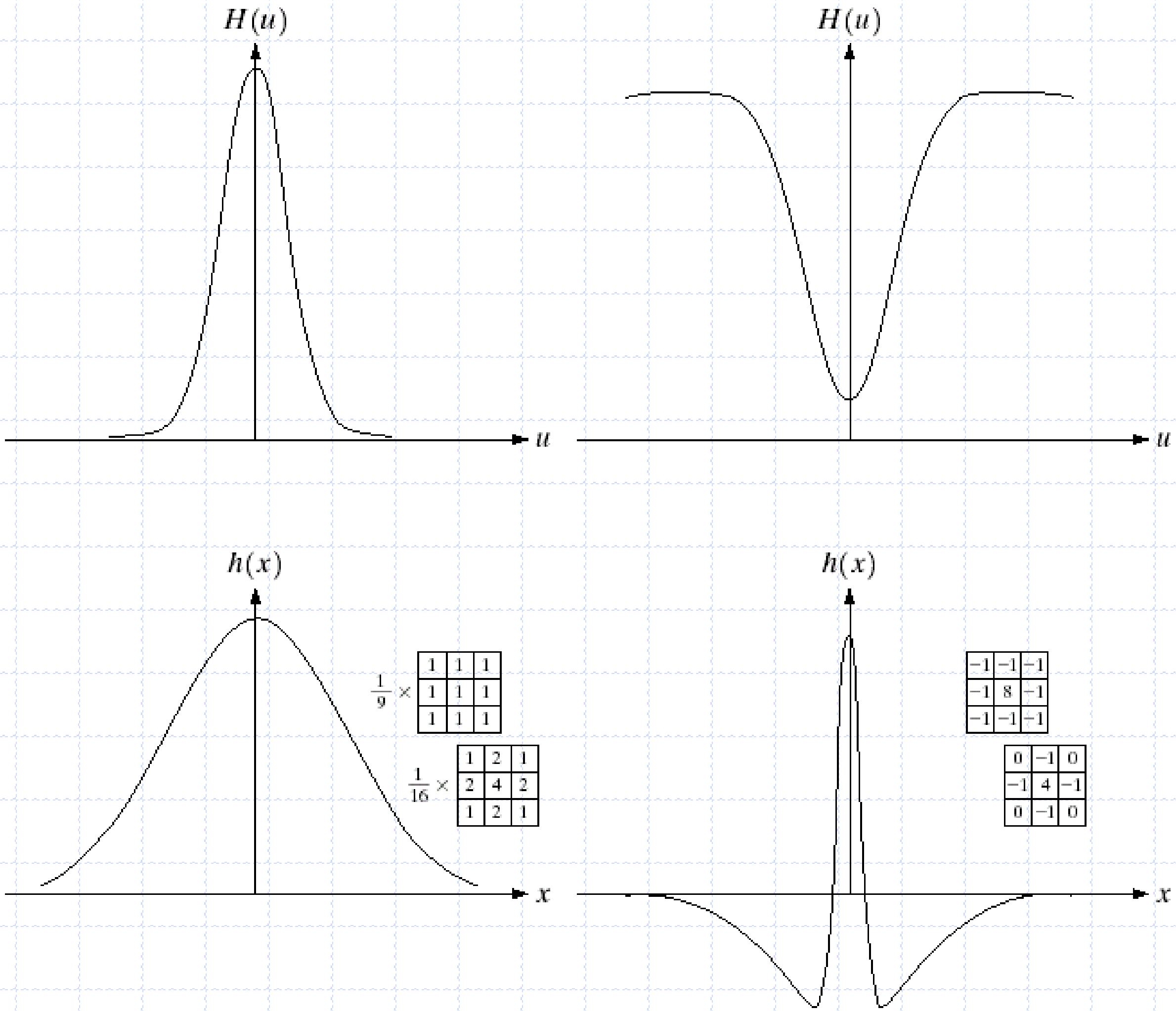
FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

High-pass Filtering

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).



Low-pass and High-pass Filters



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.

Smoothing Frequency Domain, *Ideal Low-pass Filters*

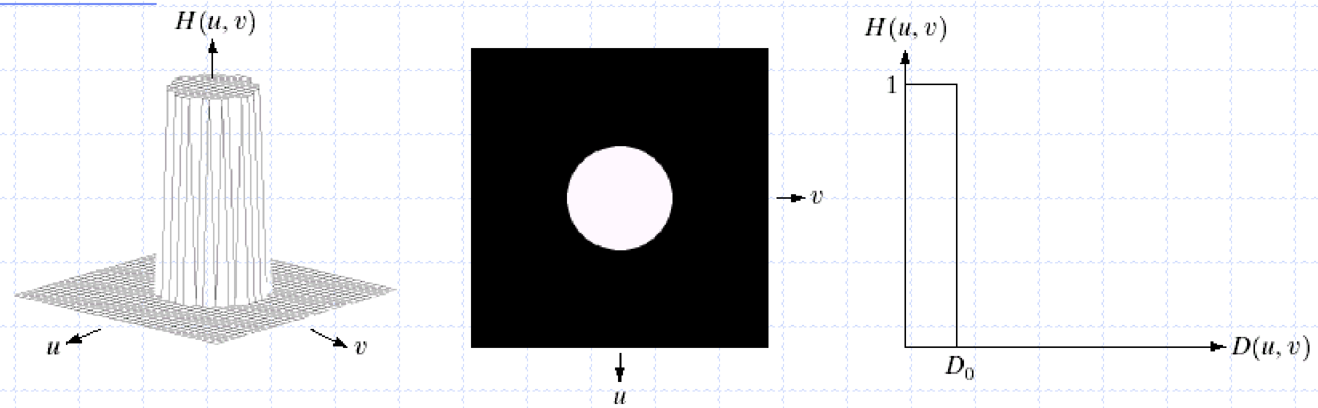
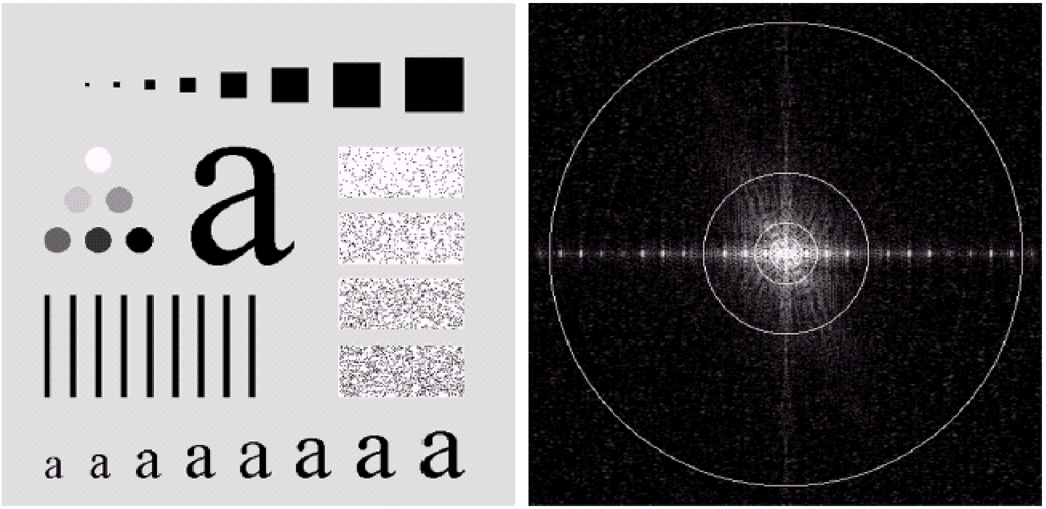


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

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

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

Smoothing Frequency Domain, Ideal Low-pass Filters



a b

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

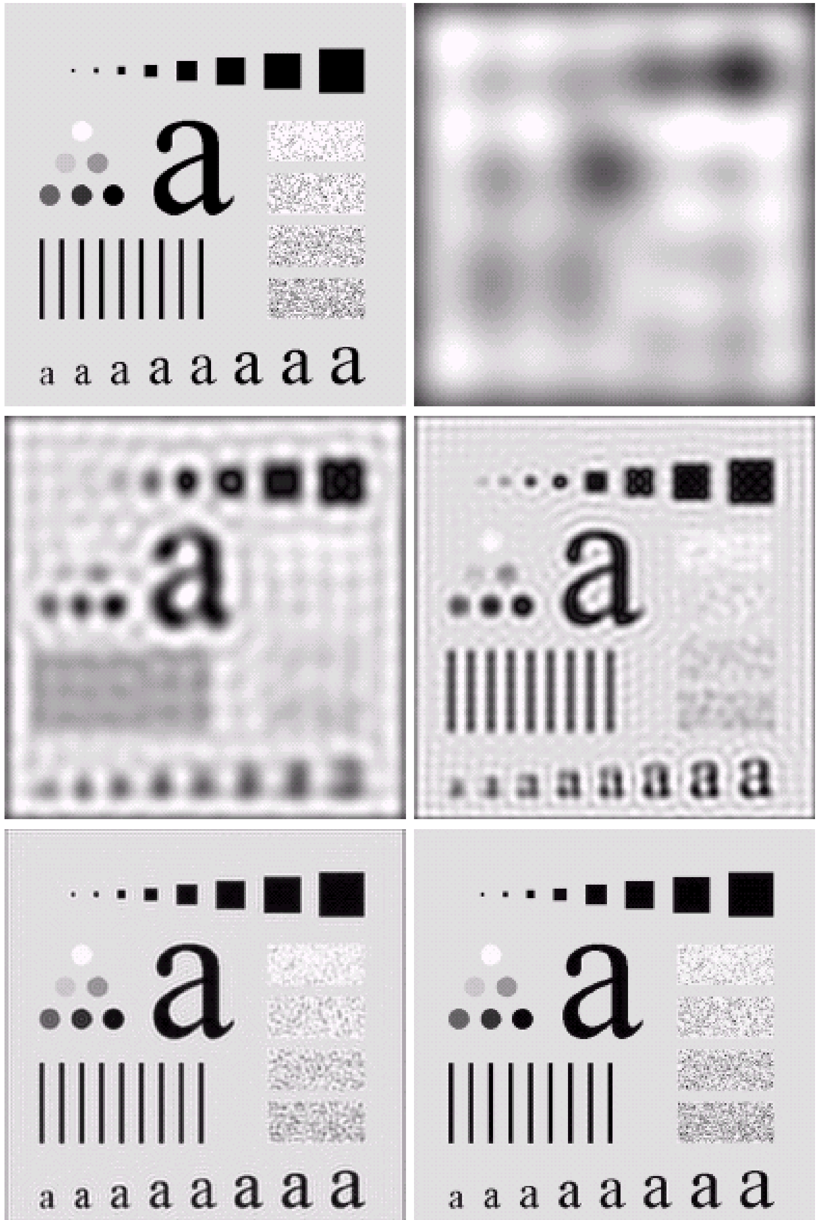
Total Power →

$$P_T = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|^2$$

The remained percentage power after filtration →

$$\alpha = 100 \times \left[\sum_u \sum_v |F(u,v)| / P_T\right]$$

Smoothing Frequency Domain,
Ideal Low-pass Filters



For circle radius $f_c = 5$

Enclose image power $\alpha = 92\%$

$f_c = 15$
$\alpha = 94.6\%$

$f_c = 30$
$\alpha = 96.4\%$

$f_c = 80$
$\alpha = 98\%$

$f_c = 230$
$\alpha = 99.5\%$

Cause of Ringing

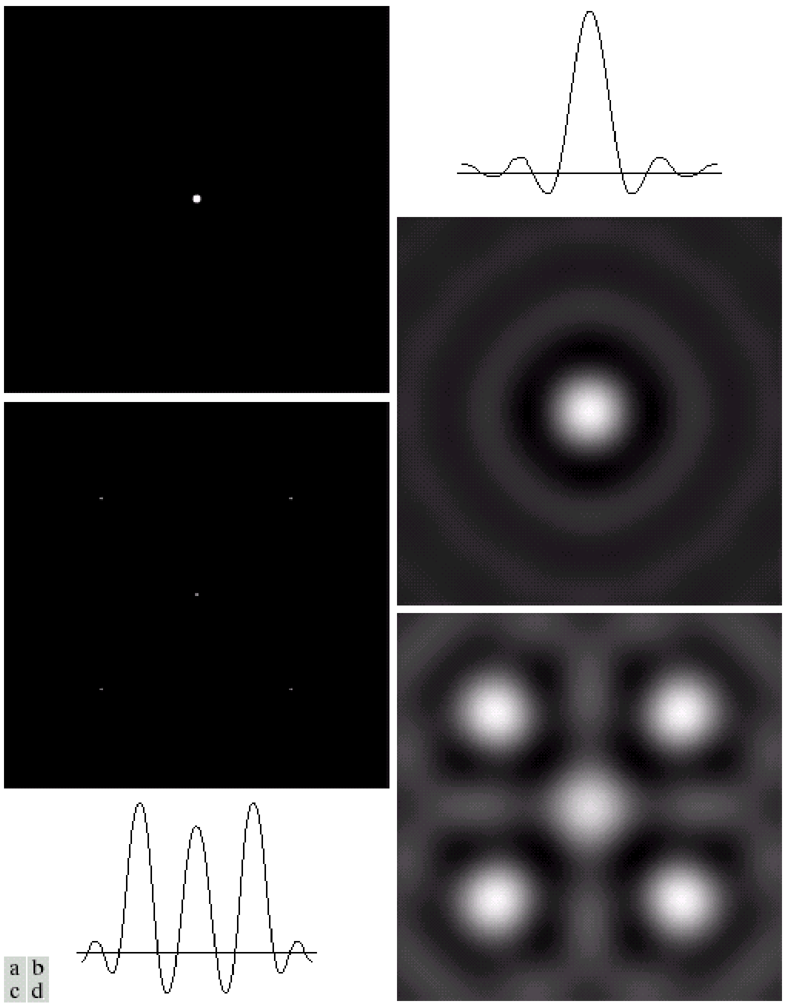
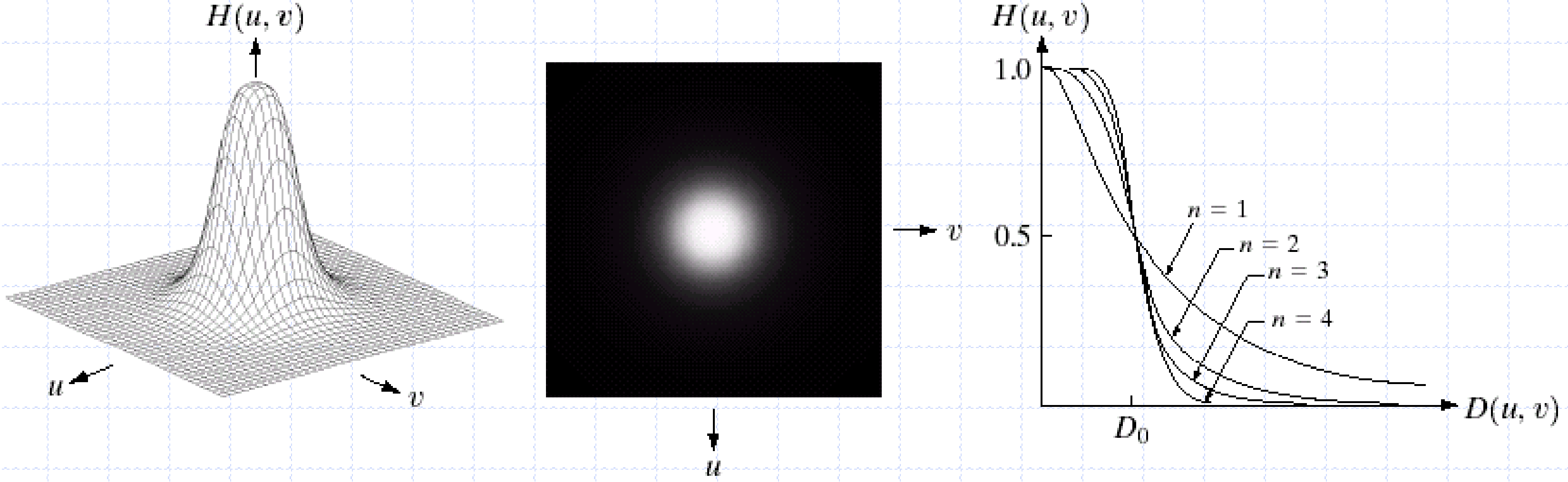


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

Smoothing Frequency Domain, Butterworth Low-pass Filters



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.

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

Smoothing Frequency Domain,
Butterworth Low-pass Filters

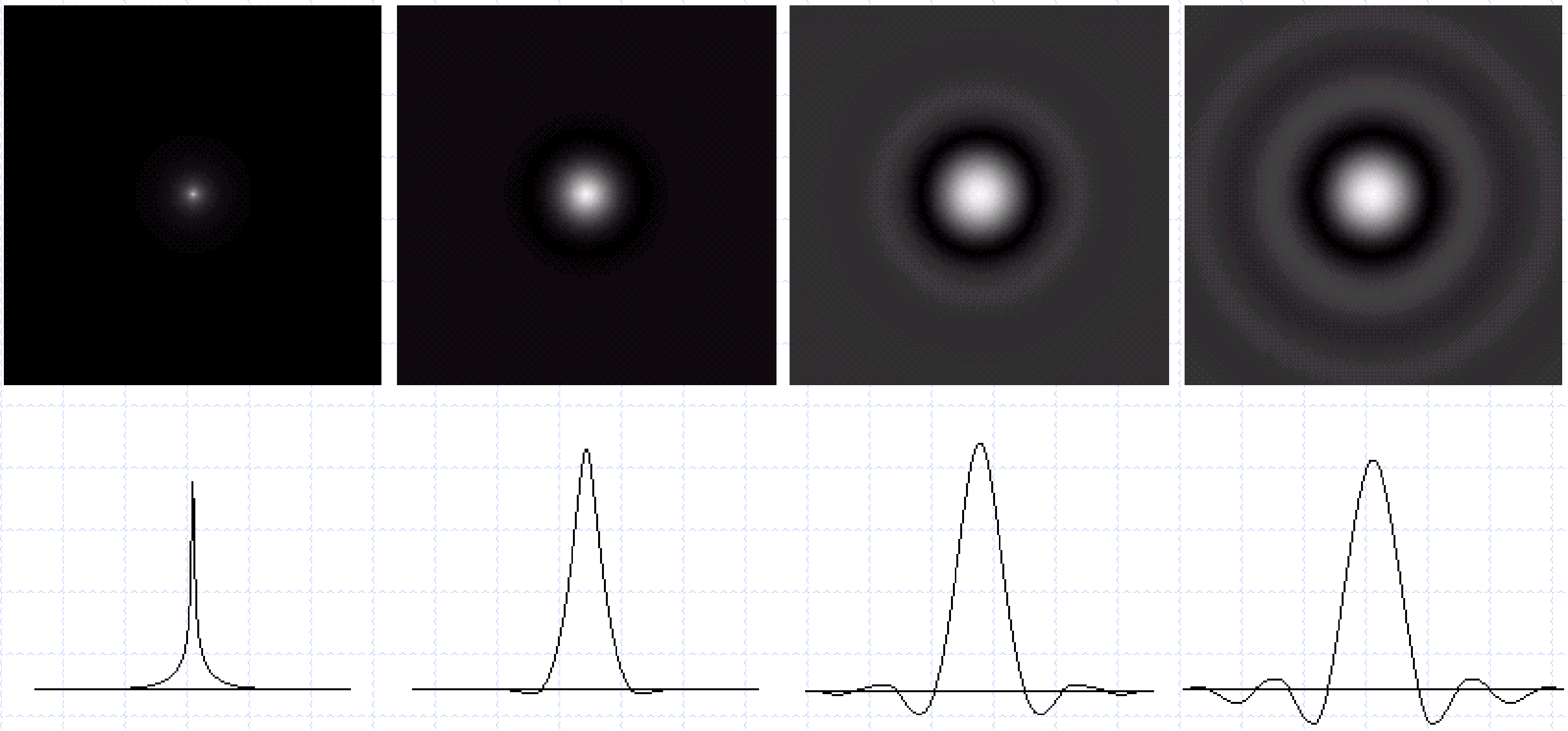
Butterworth Low-pass Filter: n=2

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.

Smoothing Frequency Domain,

Butterworth Low-pass Filters



a b c d

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.

Smoothing Frequency Domain, Gaussian Low-pass Filters

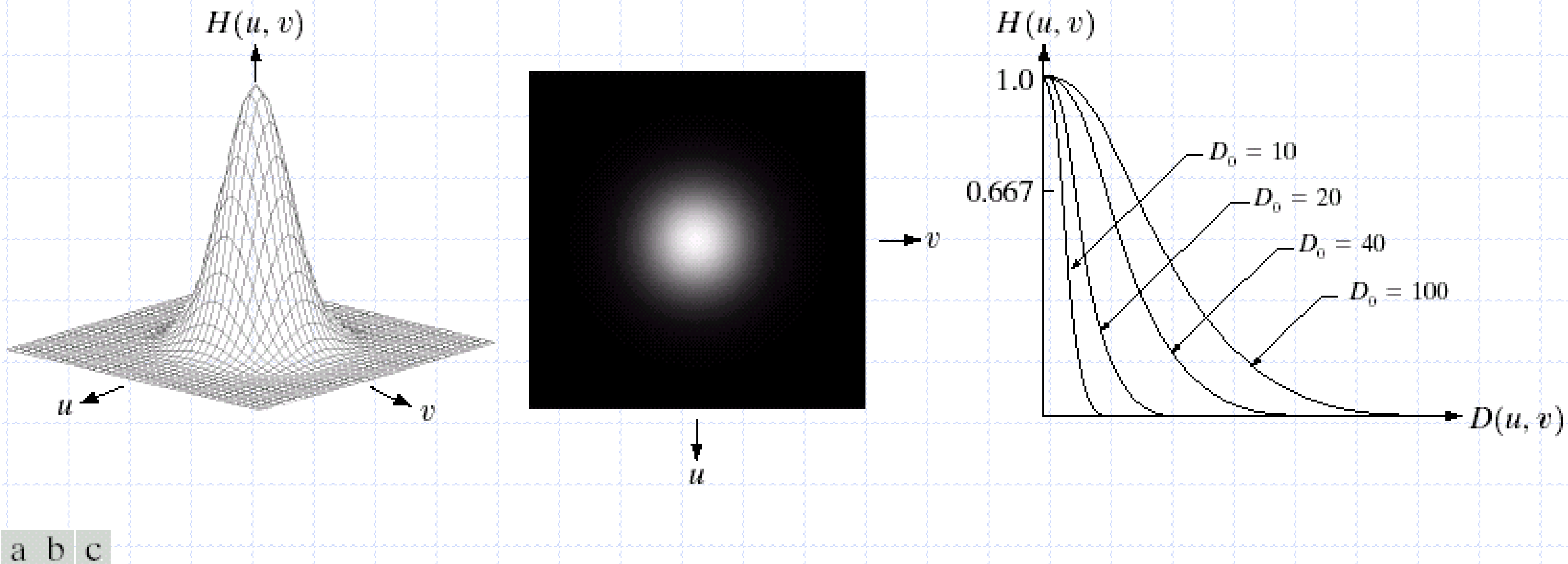


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

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

Smoothing Frequency Domain, Gaussian Low-pass Filters

Gaussian Low-pass

Radii= 5

Radii= 15

Radii= 30

Radii= 80

Radii= 230

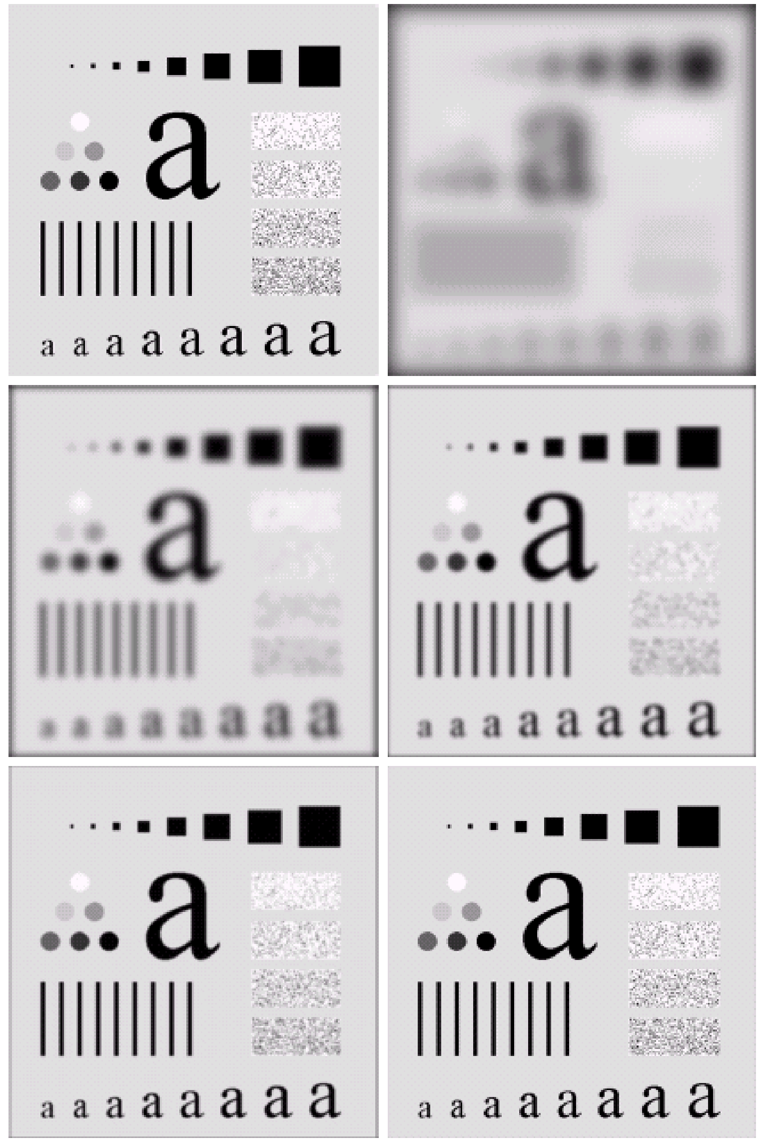


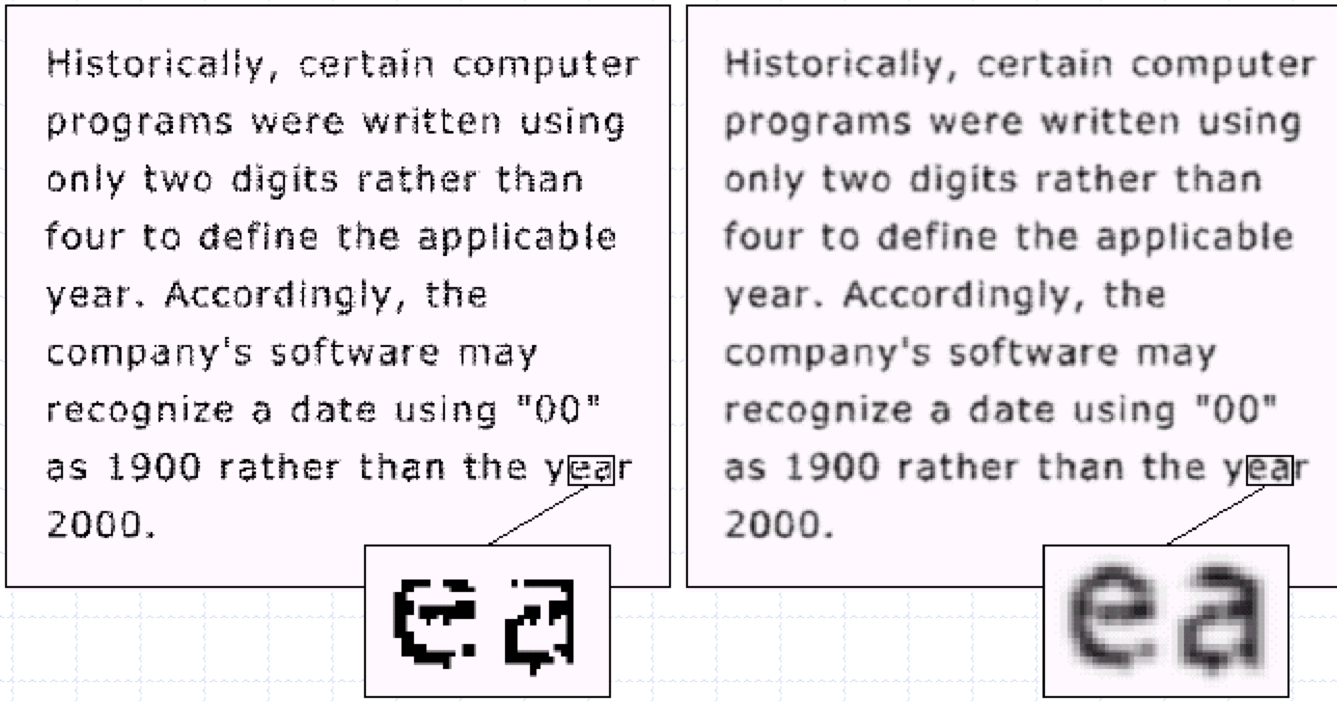
FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

a b
c d
e f

Smoothing Frequency Domain, Gaussian Low-pass Filters

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).

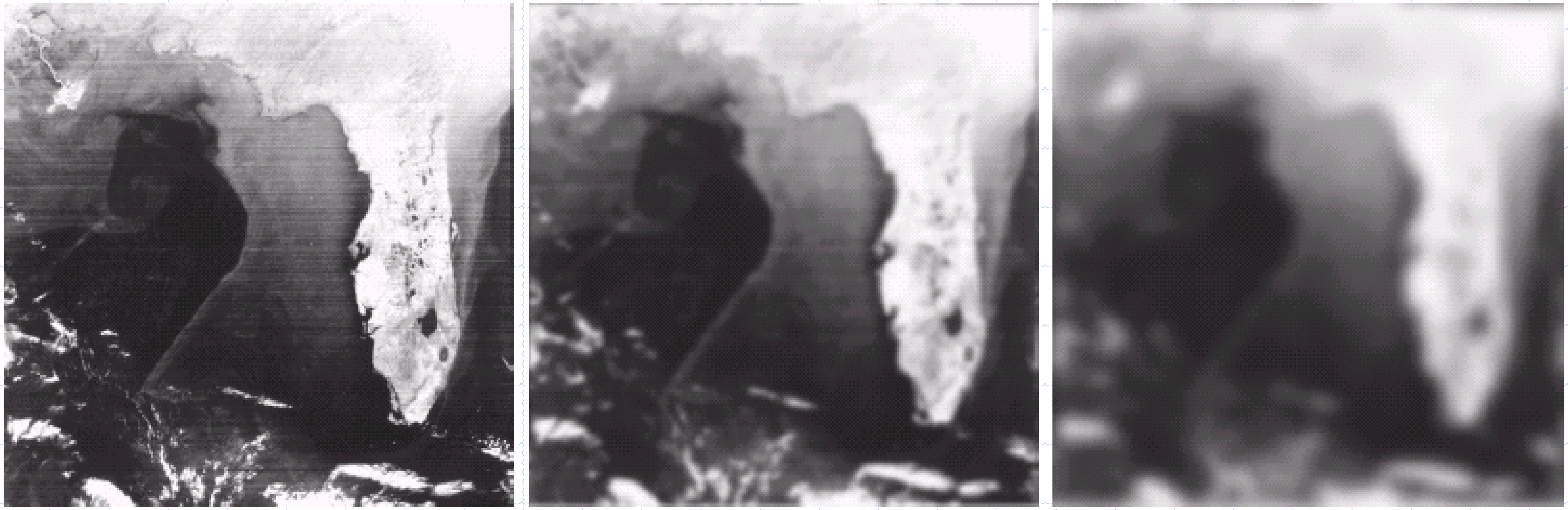


Smoothing Frequency Domain,
Gaussian Low-pass Filters

a b c

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

Smoothing Frequency Domain,
Gaussian Low-pass Filters



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.)

Sharpening Frequency Domain Filters

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

Ideal HPF

Butterworth HPF

Gaussian HPF

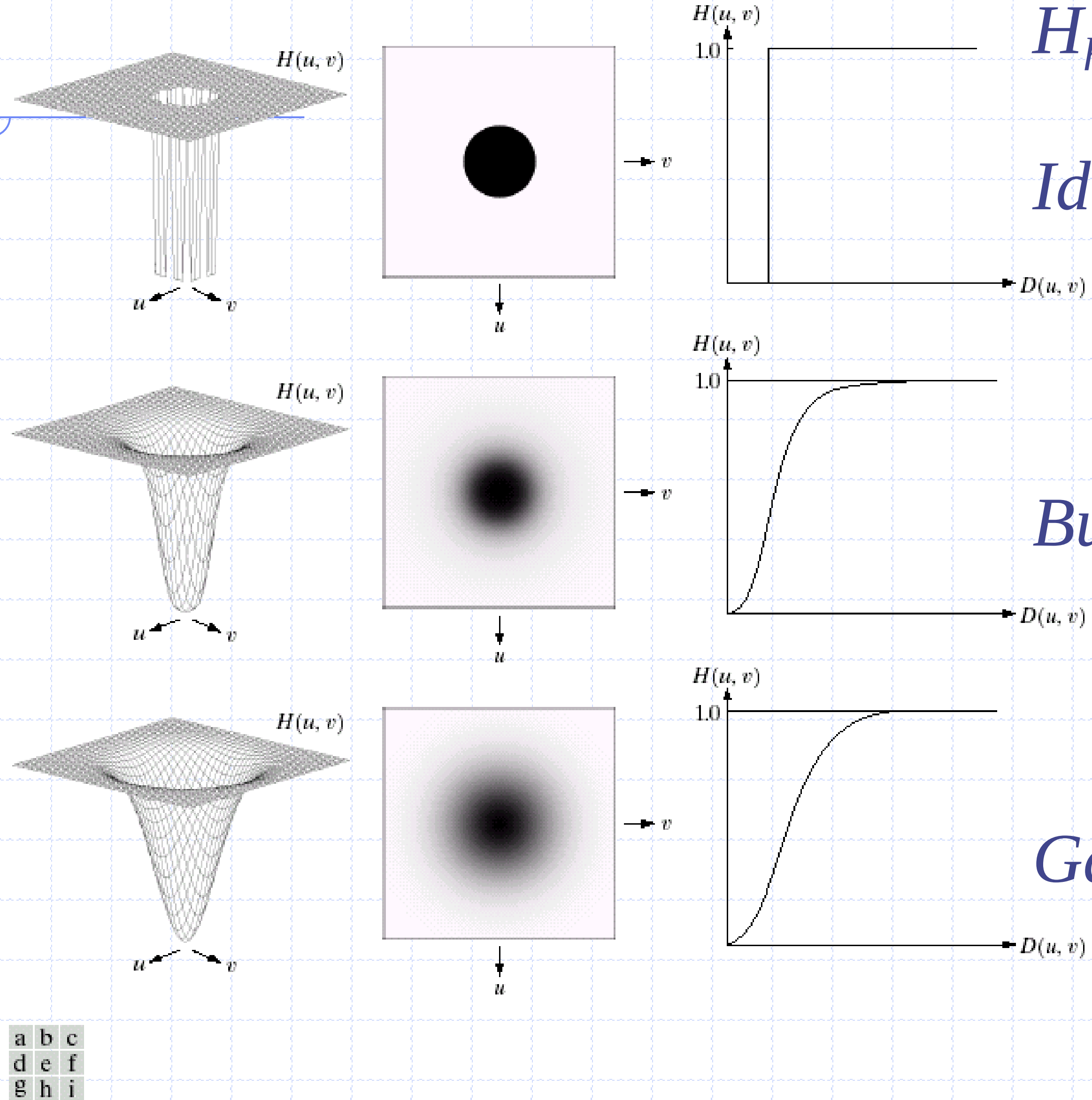
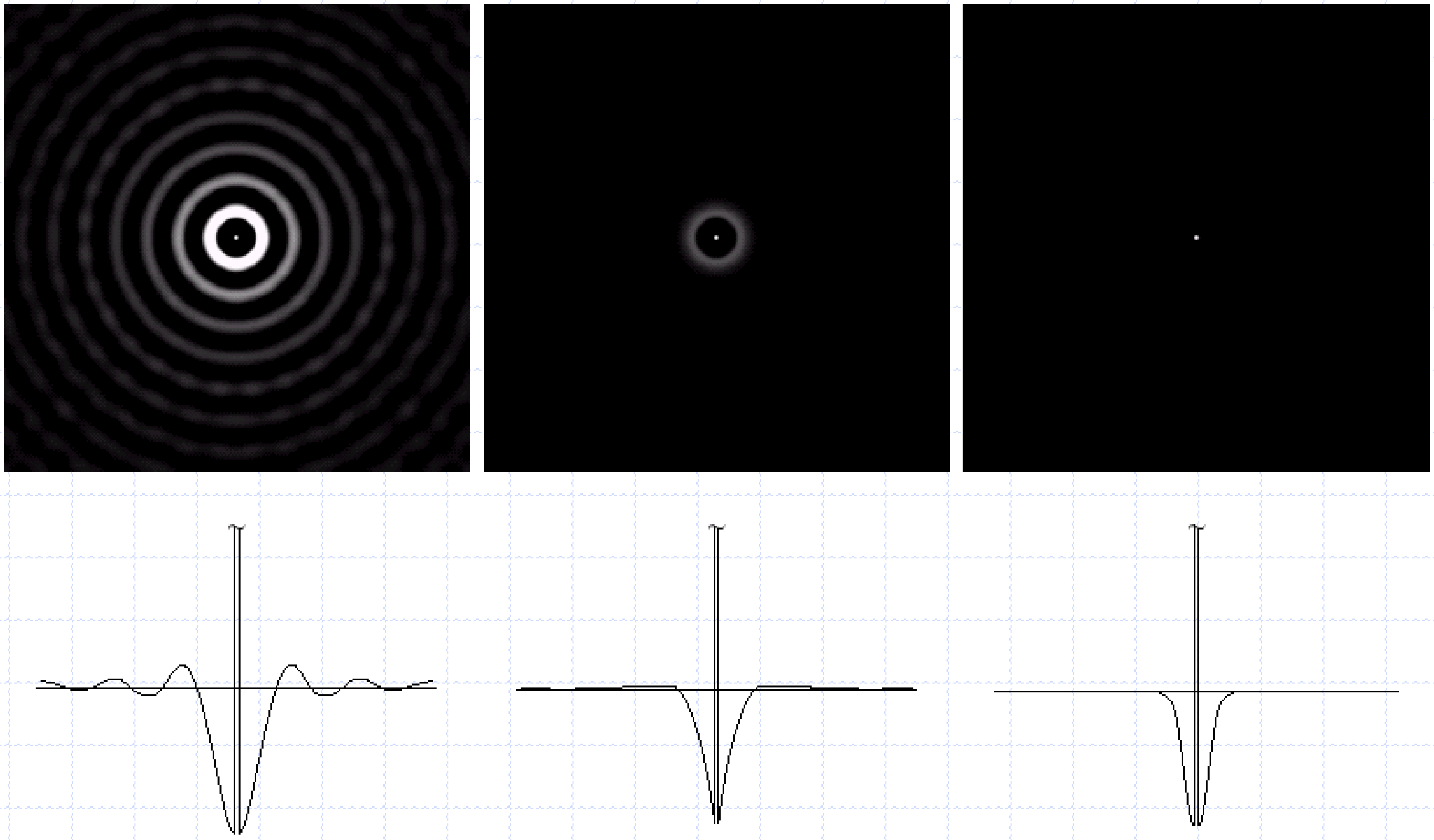


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Sharpening Frequency Domain Filters

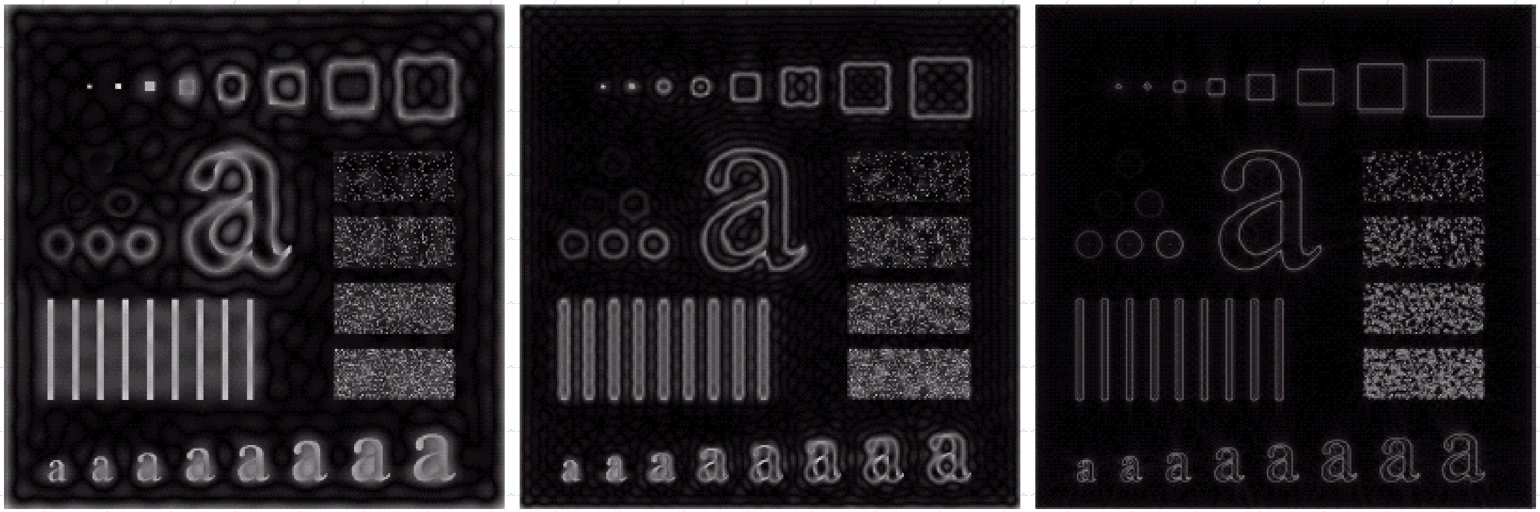


a b c

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

Sharpening Frequency Domain,

Ideal High-pass Filters



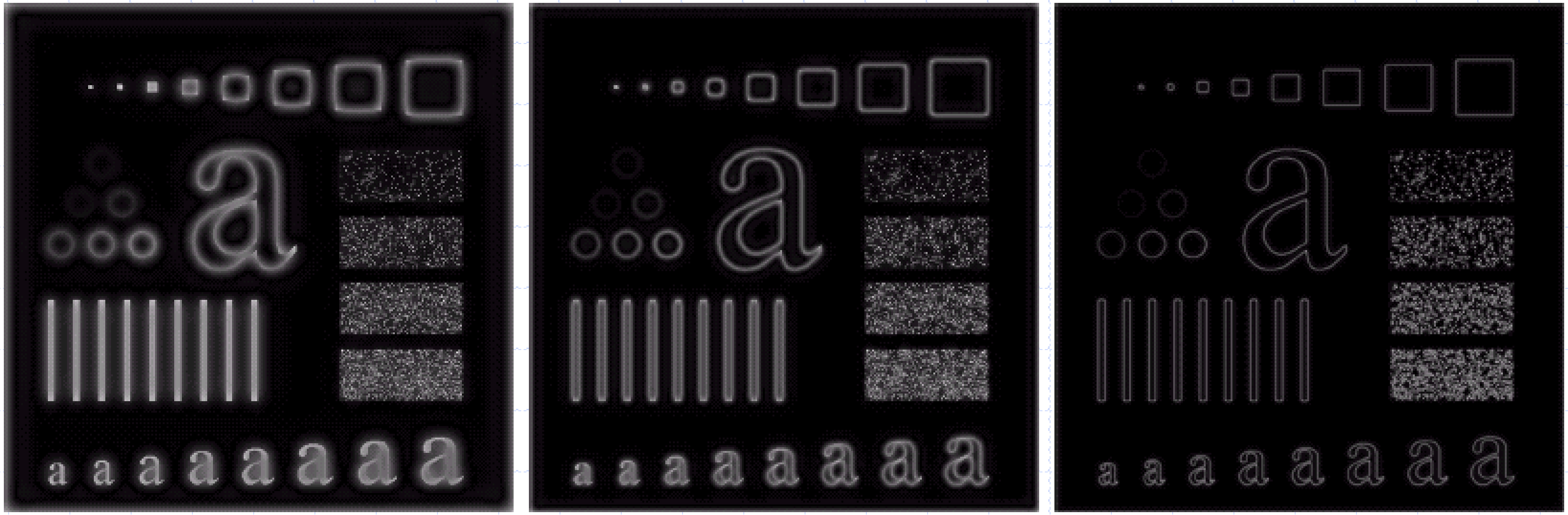
a b c

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

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

Sharpening Frequency Domain,

Butterworth High-pass Filters

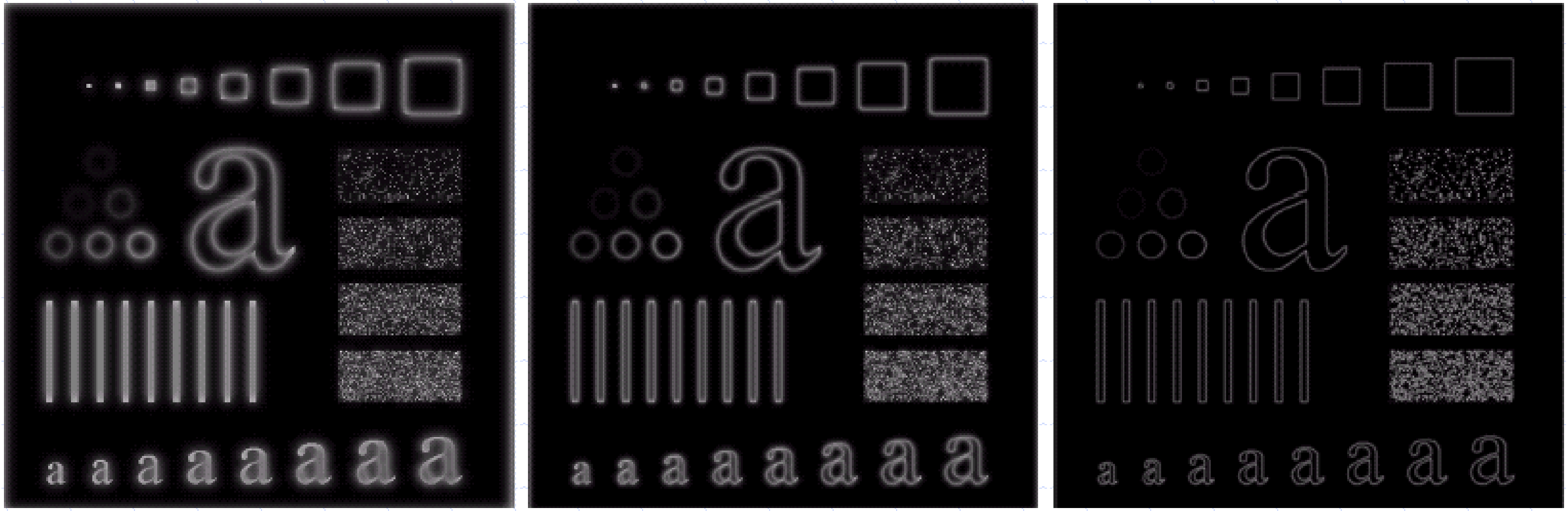


a b c

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

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

Sharpening Frequency Domain,
Gaussian High-pass Filters



a b c

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

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

Homomorphic Filtering

Homomorphic Filtering

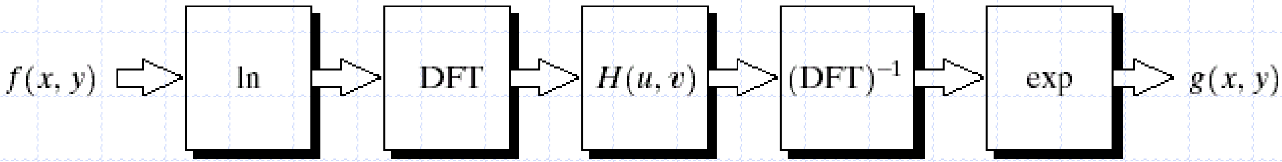
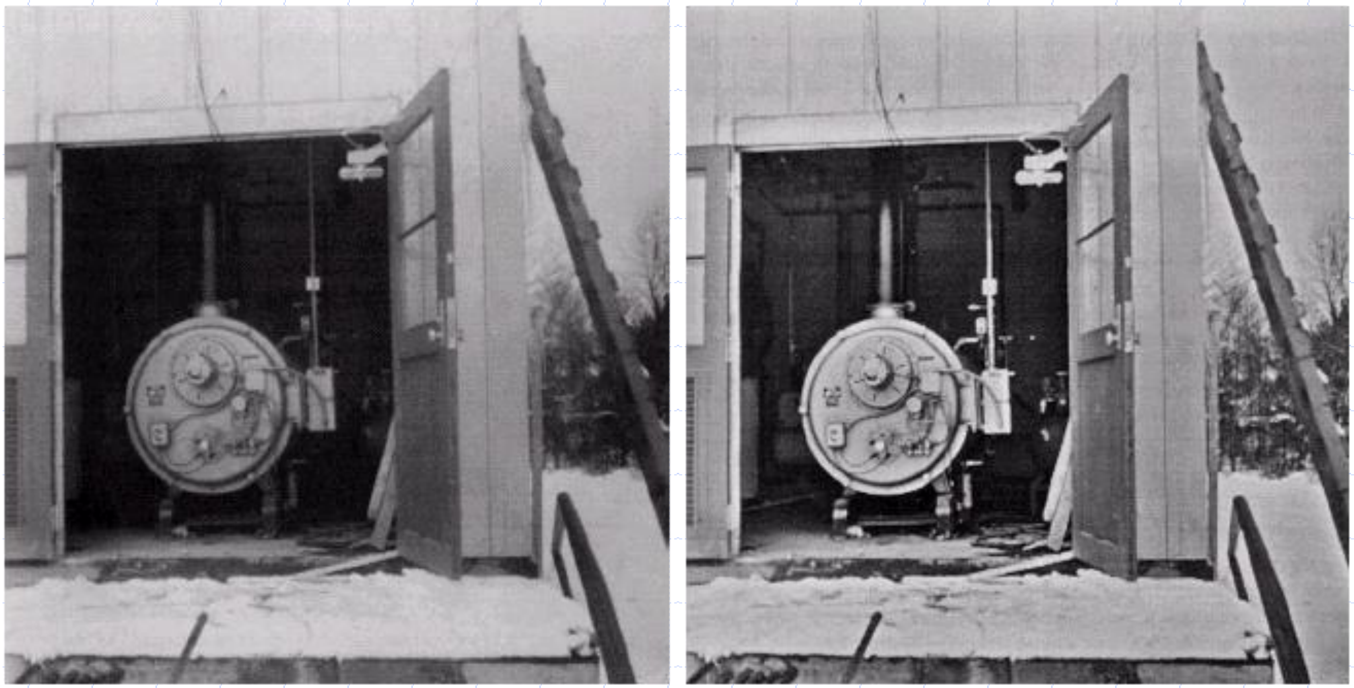


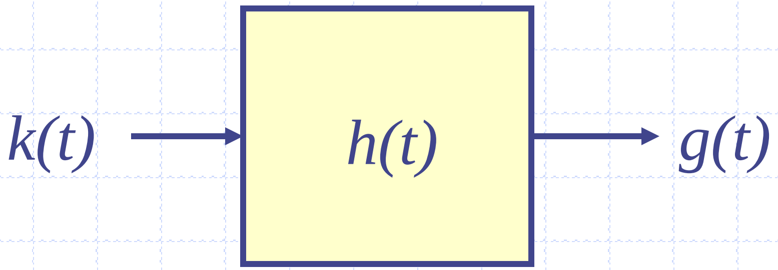
FIGURE 4.31 Homomorphic filtering approach for image enhancement.



a b

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

Convolution

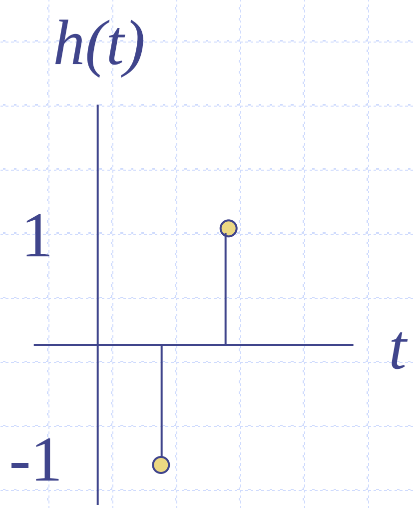


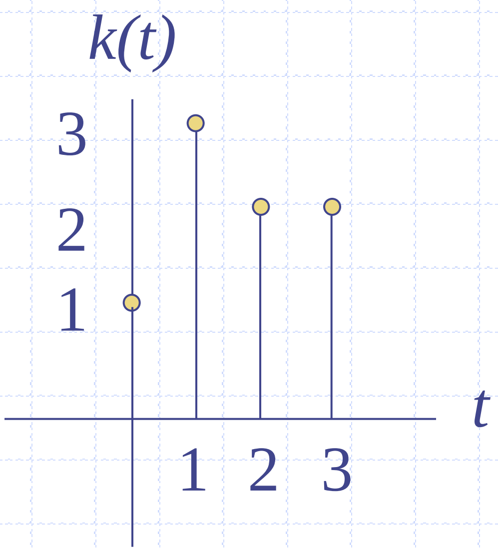
$g(t) = k(t)*h(t)$

$G(f) = K(f)H(f)$

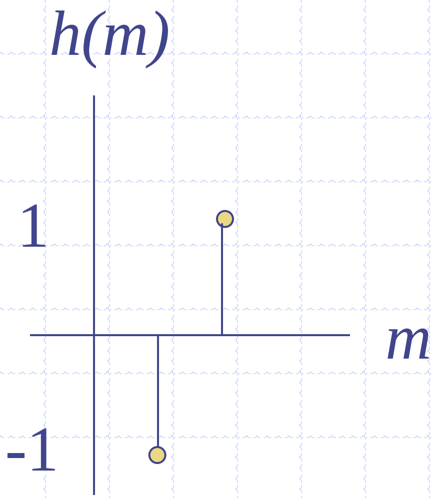
* is a convolution operator and not multiplication

$$g(t) = k(t) * h(t) = \frac{1}{M}\sum_{m=0}^{M-1} k(t-m)h(m)$$

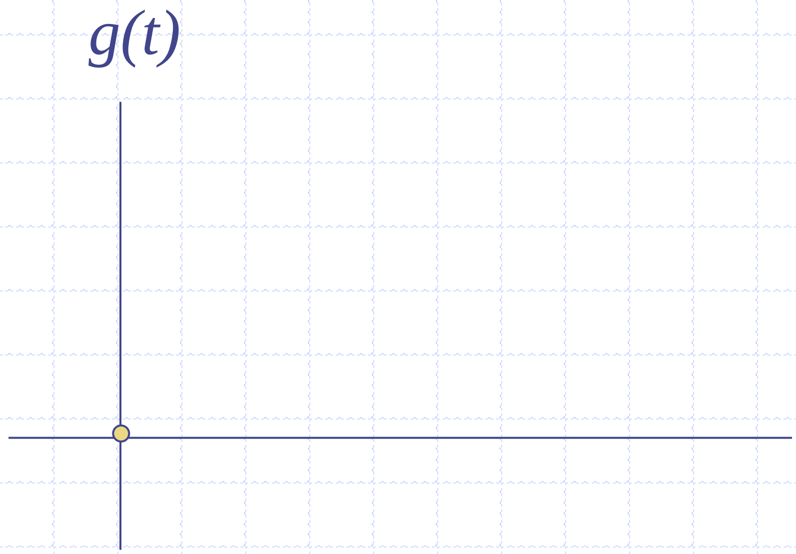


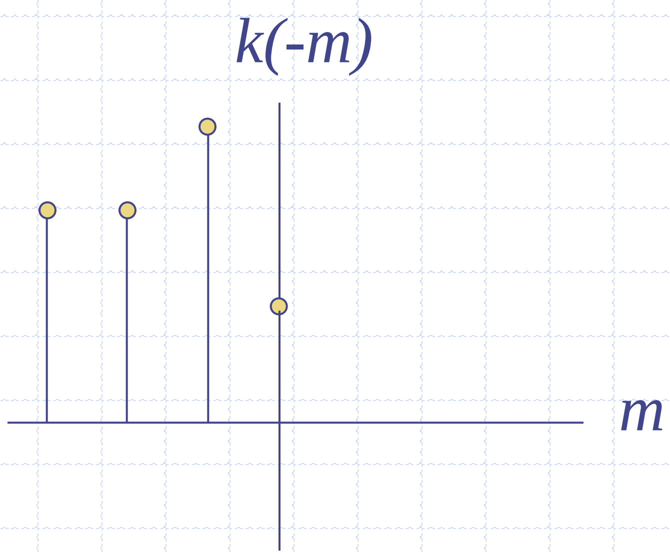


Convolution

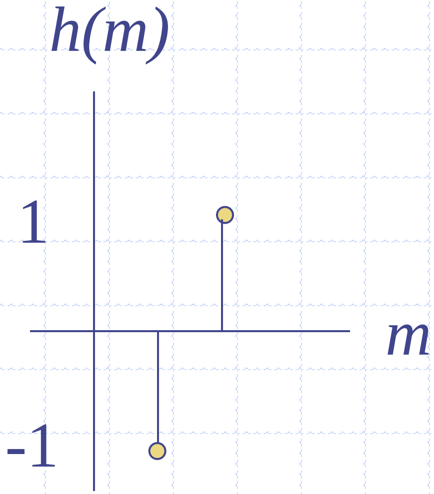


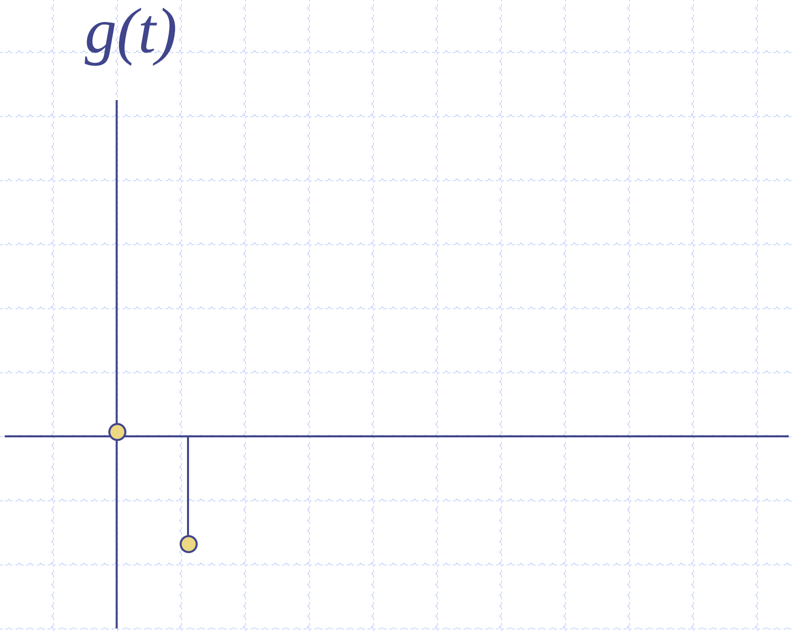
$$g(0) = \frac{1}{M}\sum_{m=0}^{M-1} k(0-m)h(m)$$

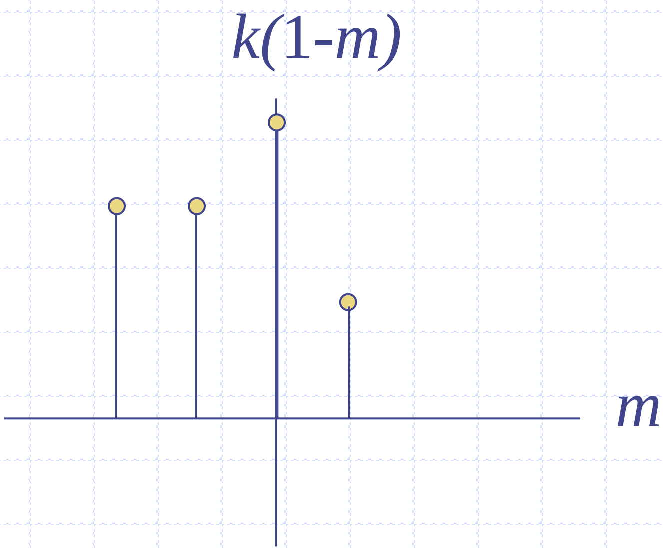




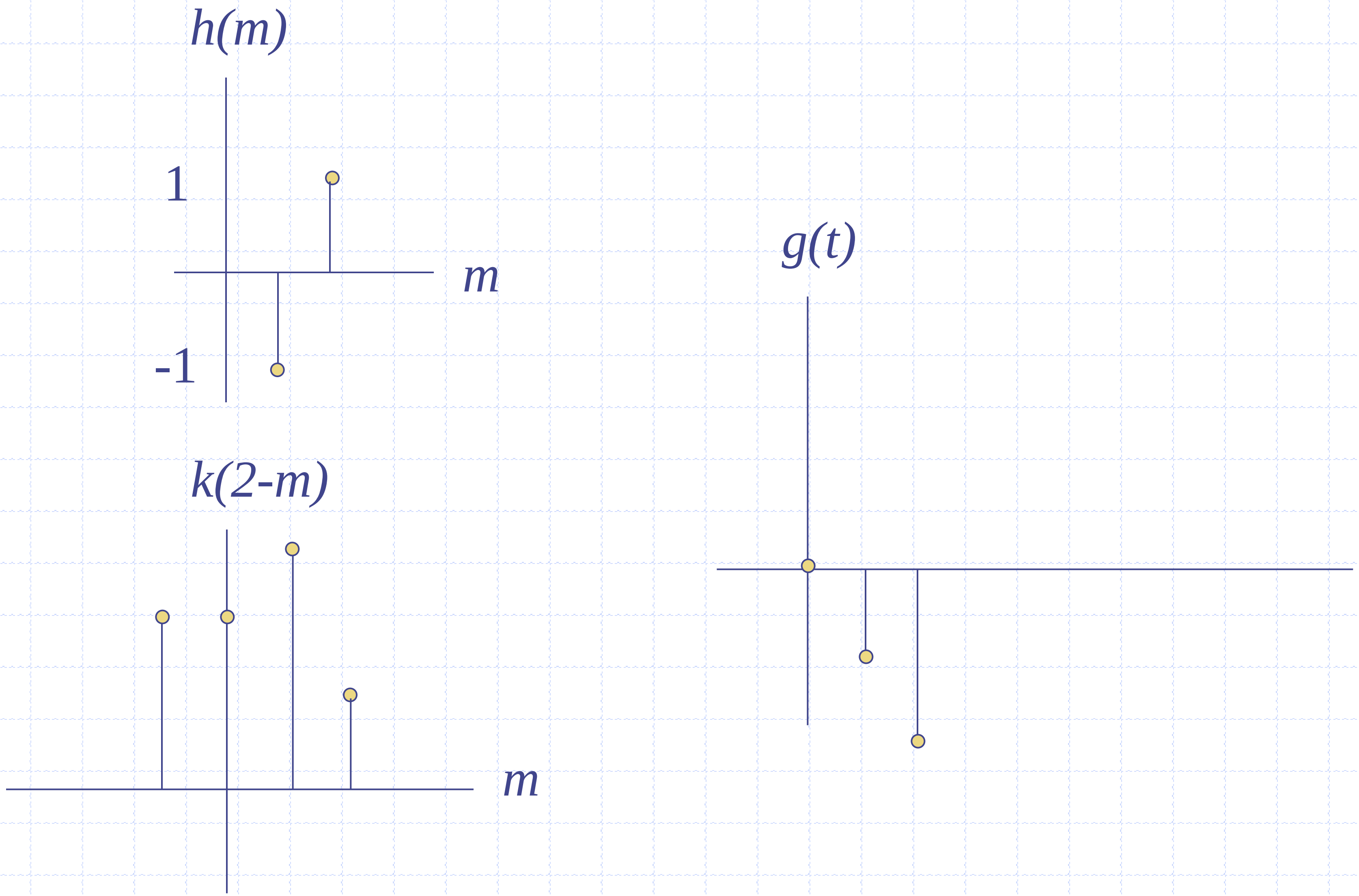
Convolution



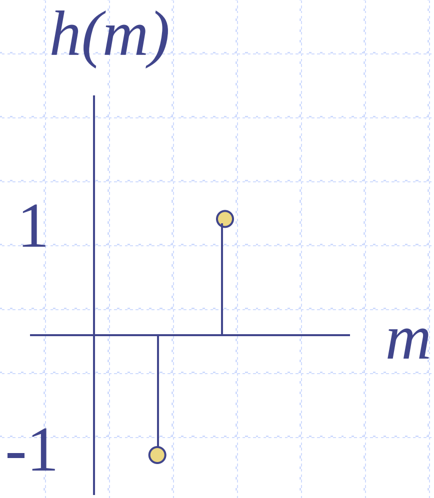


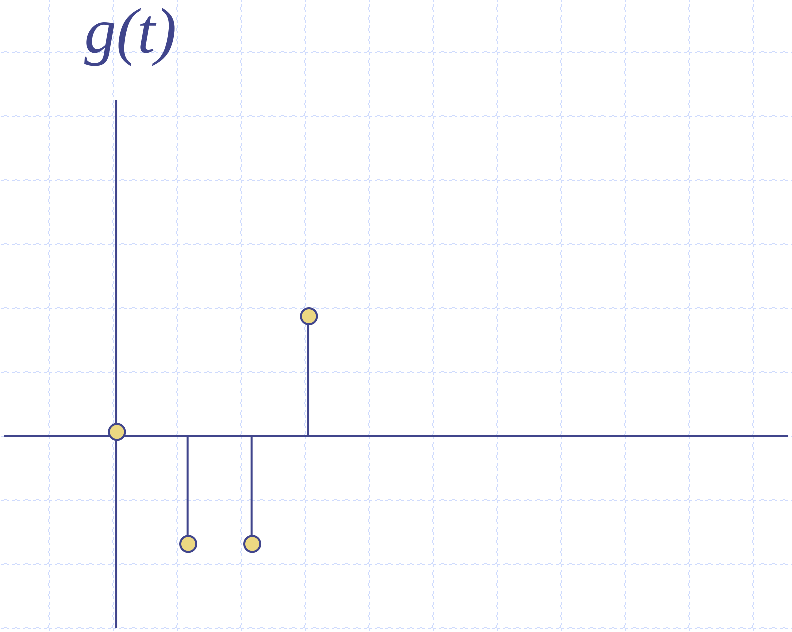


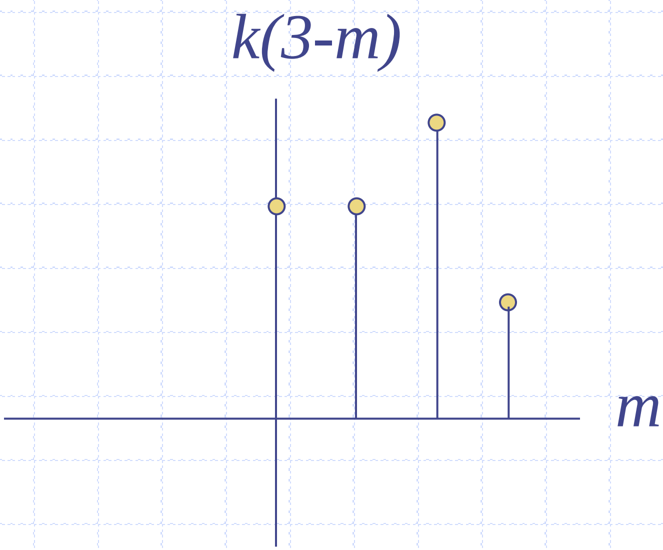
Convolution



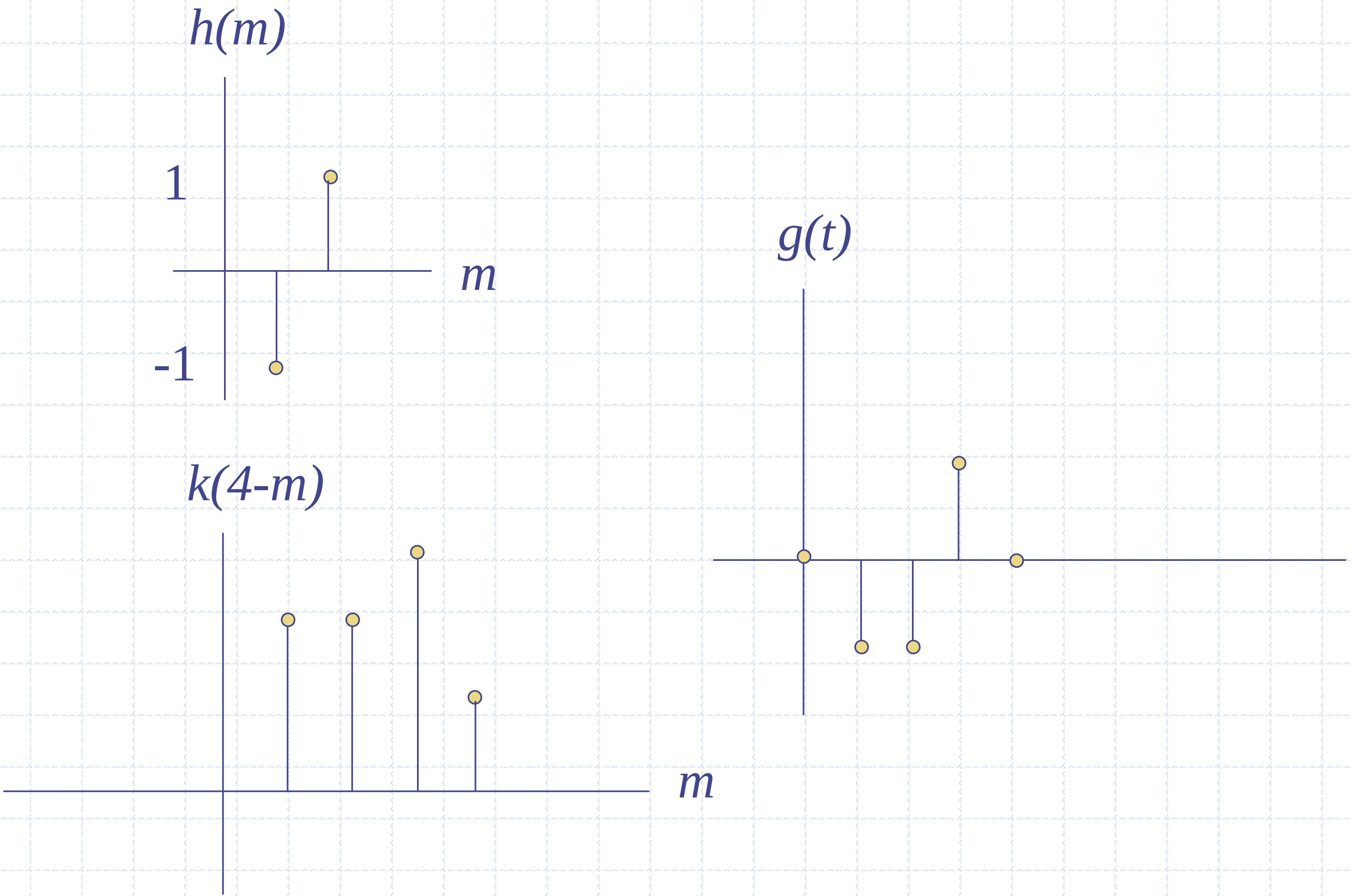
Convolution



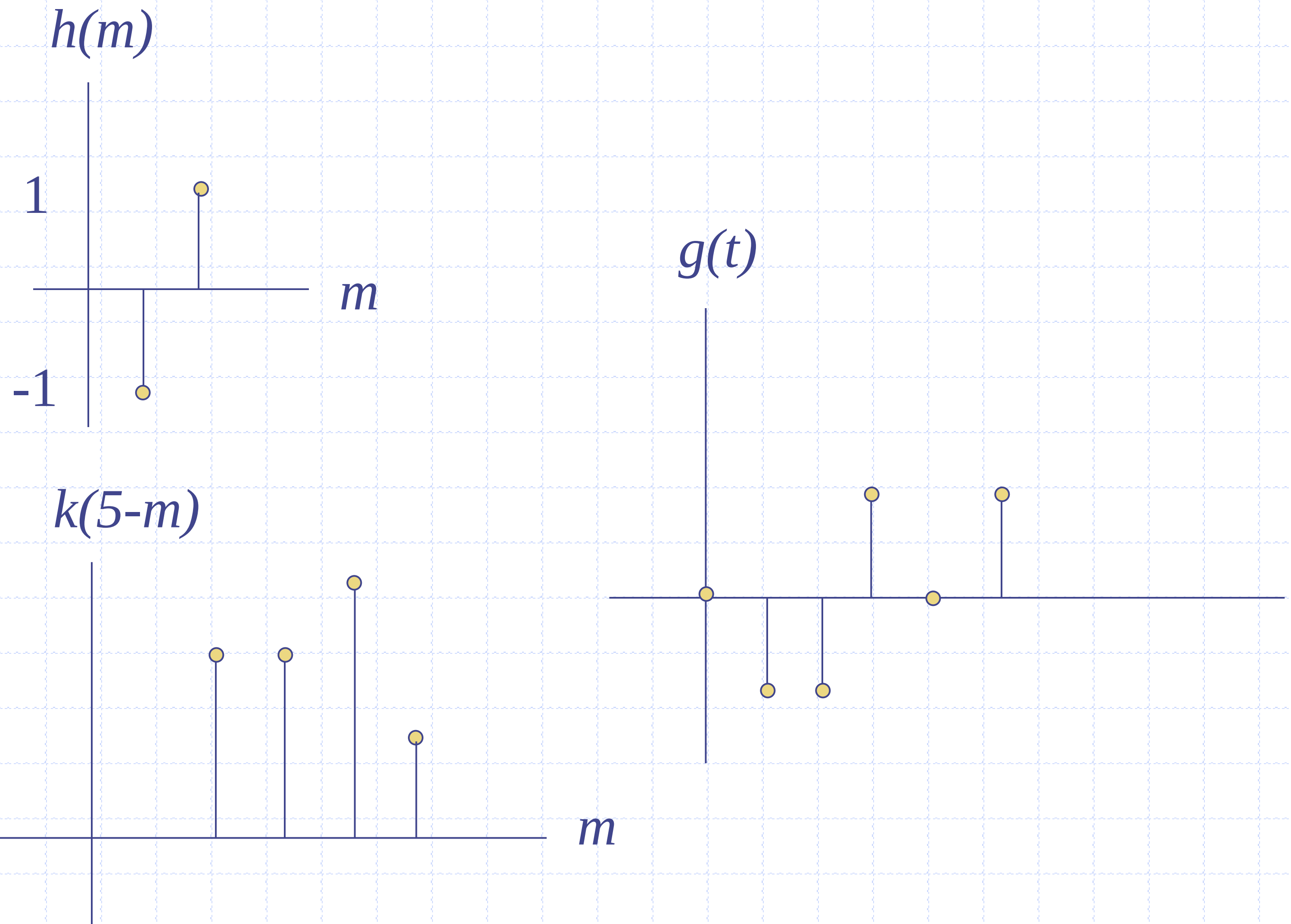




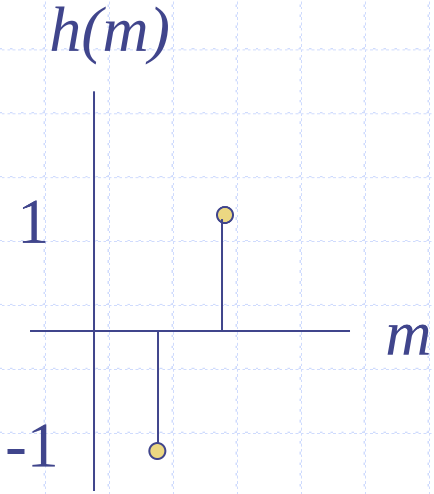
Convolution

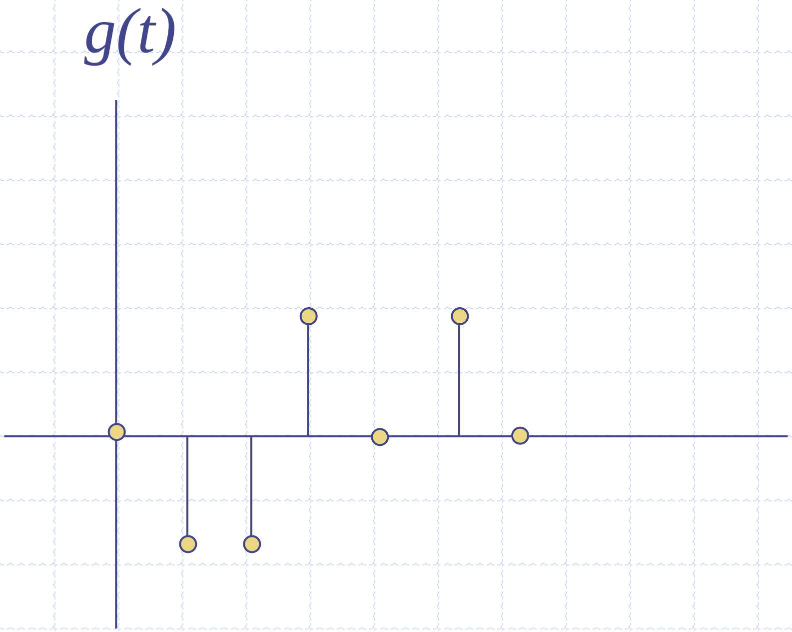


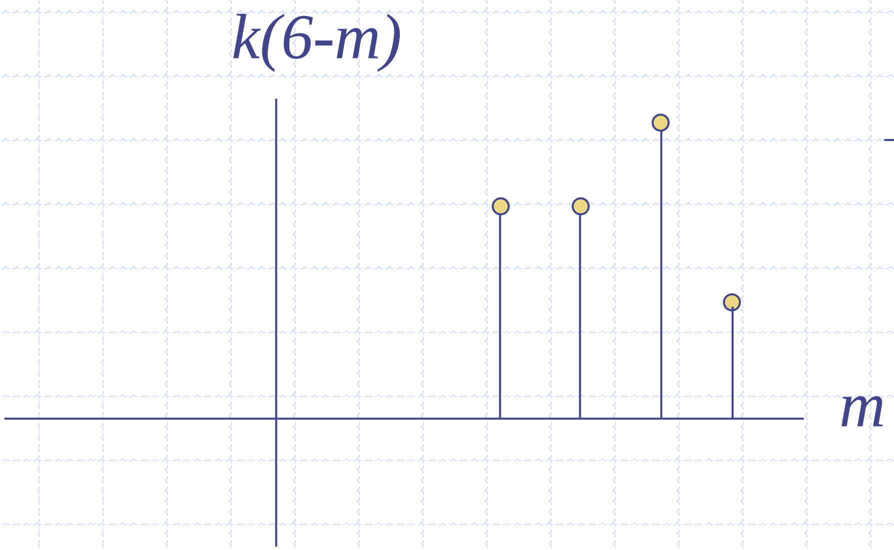
Convolution



Convolution



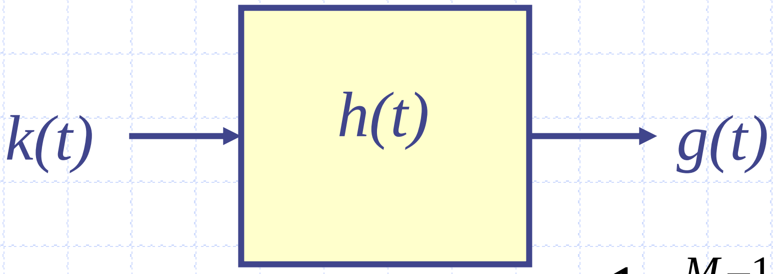




2-Dimensions Convolution

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

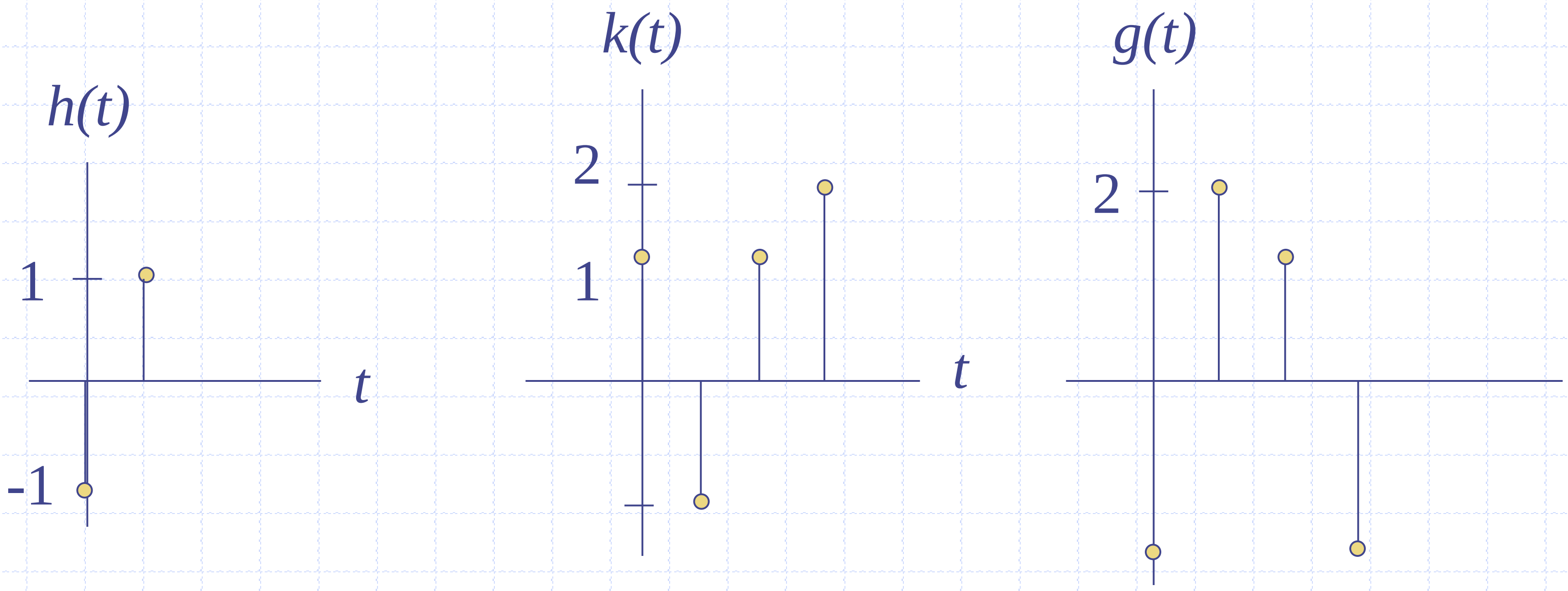
Correlation



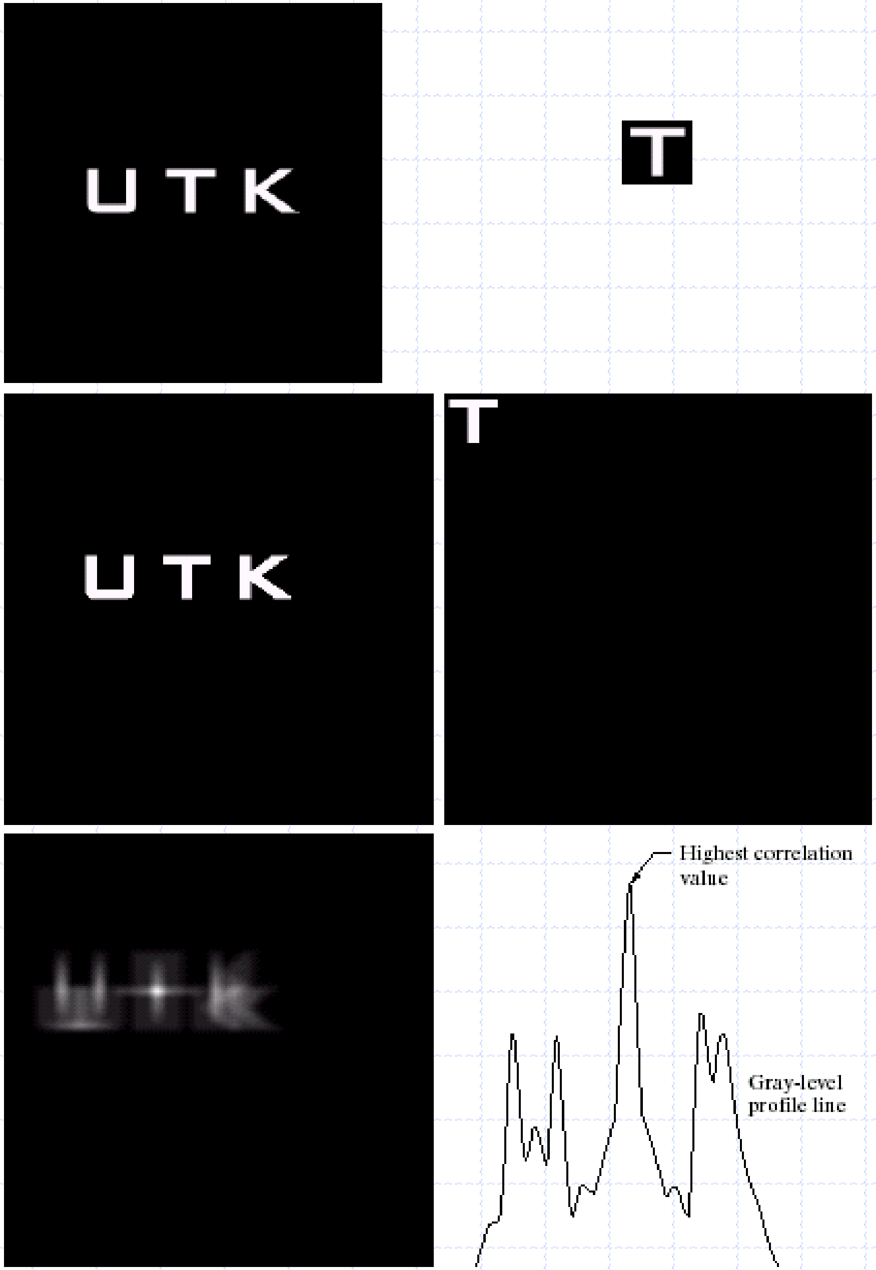
$g(t) = k(t) \circ h(t)$

$G(f) = K(f)^* H(f)$

$$g(t) = k(t) \circ h(t) = \frac{1}{M} \sum_{m=0}^{M-1} k(t+m)h(m)$$



Correlation



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.

Convolution

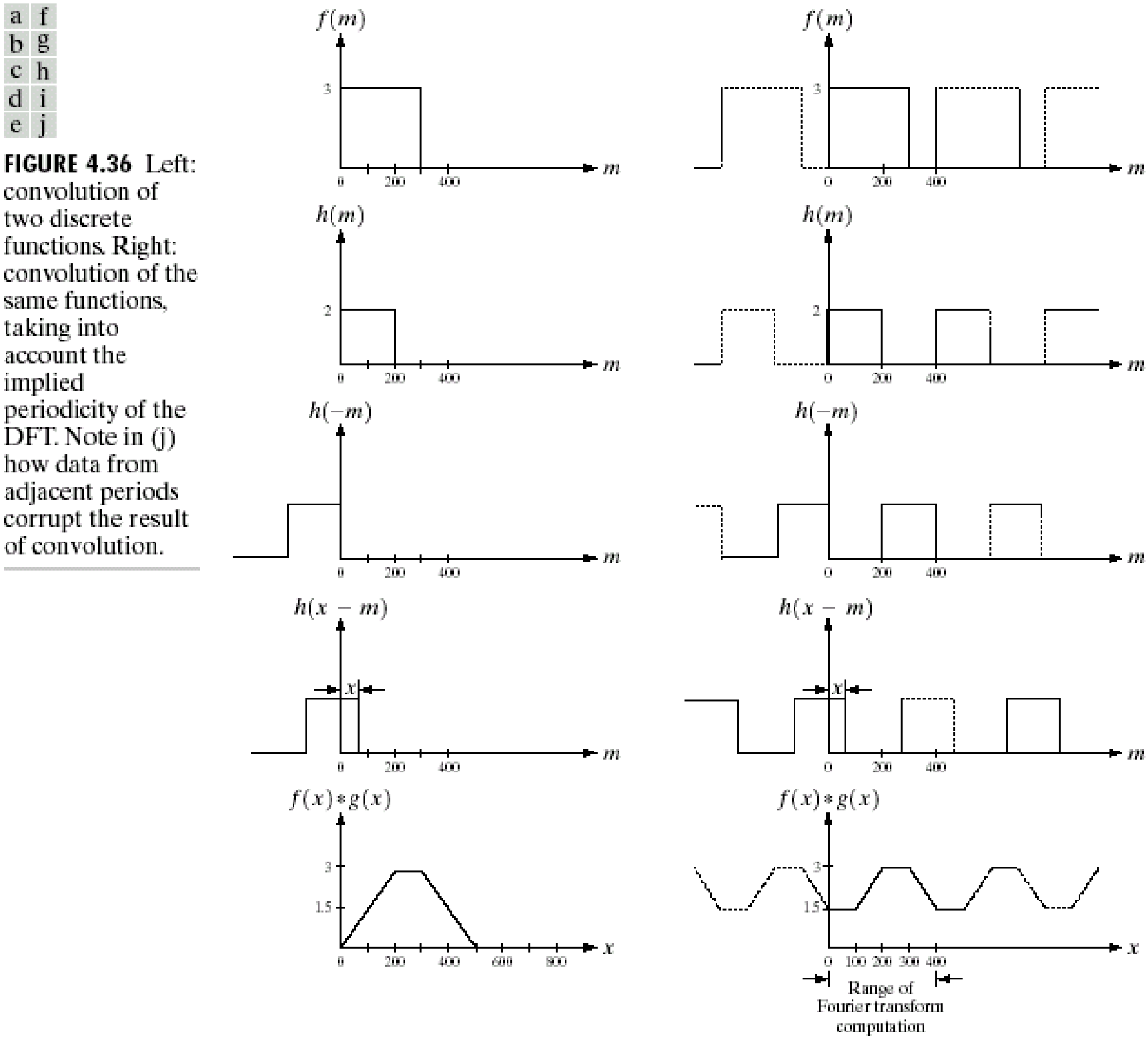


FIGURE 4.36 Left: convolution of two discrete functions. Right: convolution of the same functions, taking into account the implied periodicity of the DFT. Note in (j) how data from adjacent periods corrupt the result of convolution.

Convolution

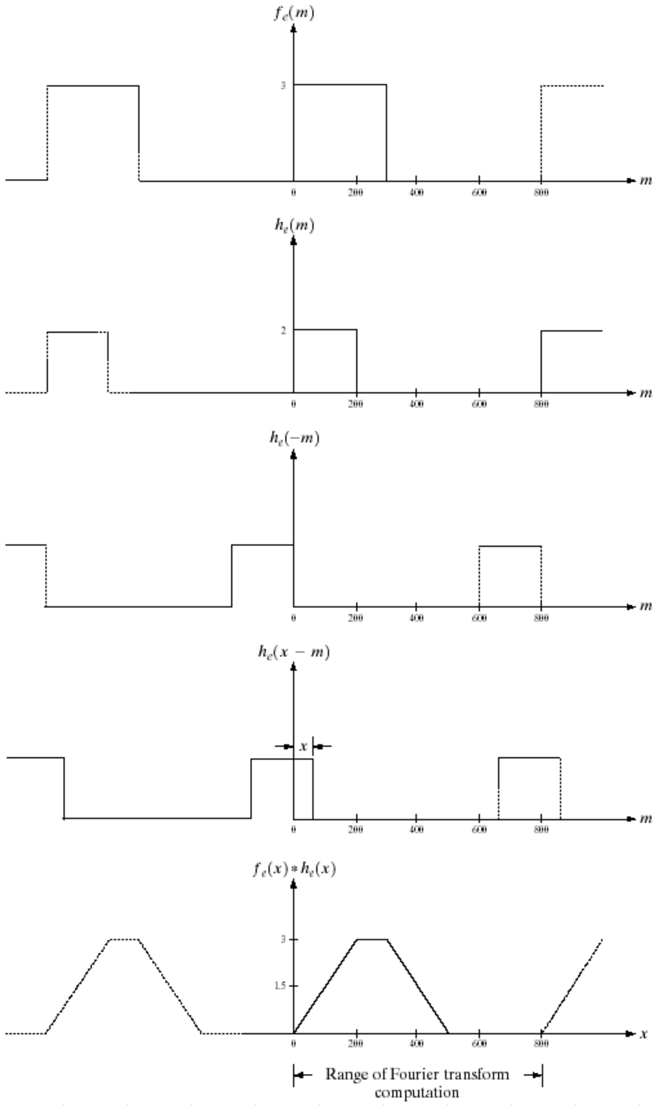
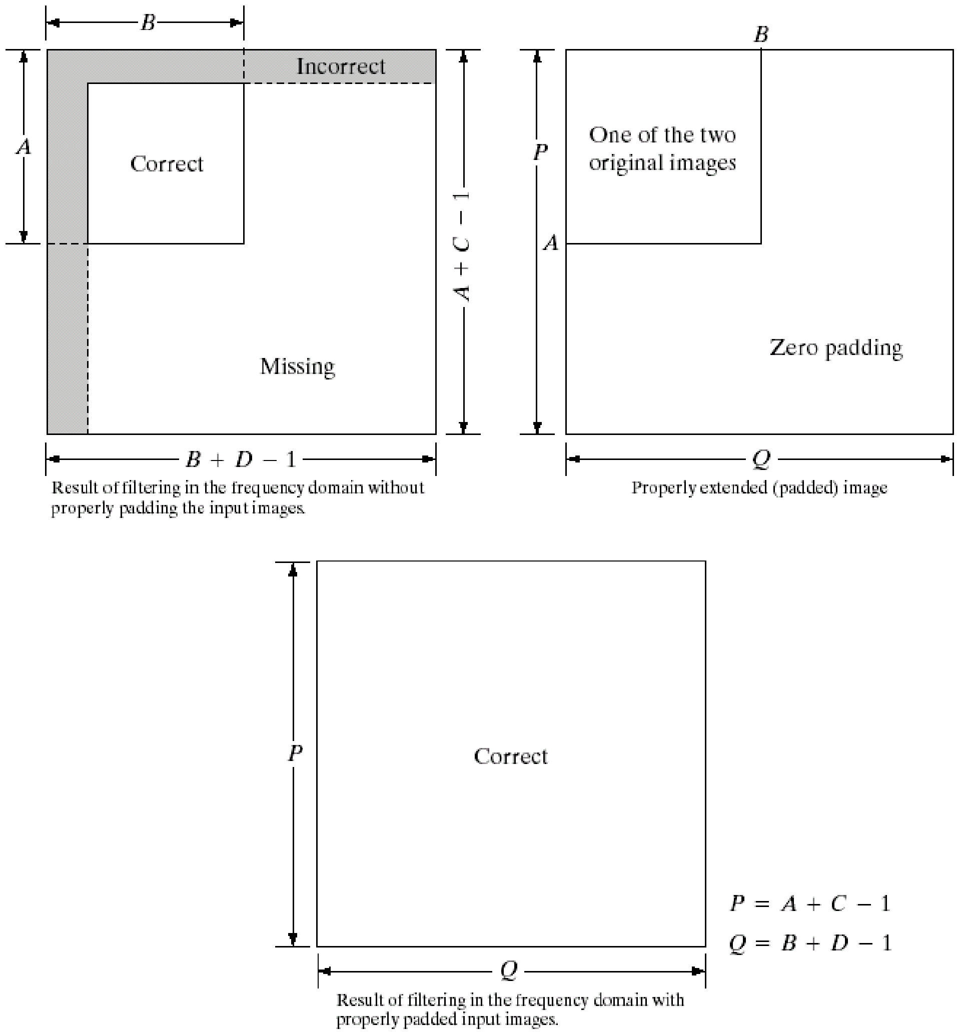


FIGURE 4.37 Result of performing convolution with extended functions. Compare Figs. 4.37(e) and 4.36(e).

Convolution



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.

Convolution

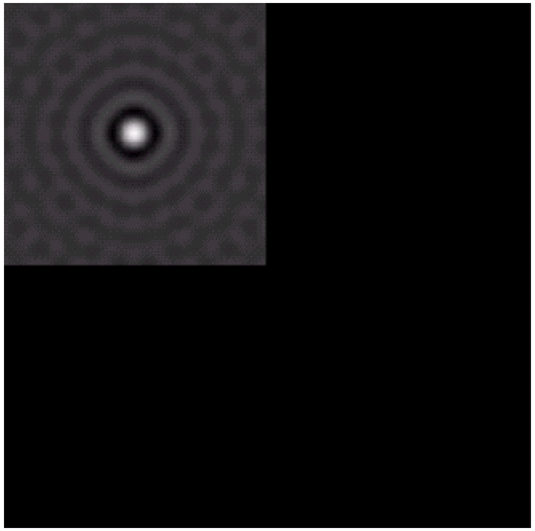


FIGURE 4.39 Padded lowpass filter is the spatial domain (only the real part is shown).

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.