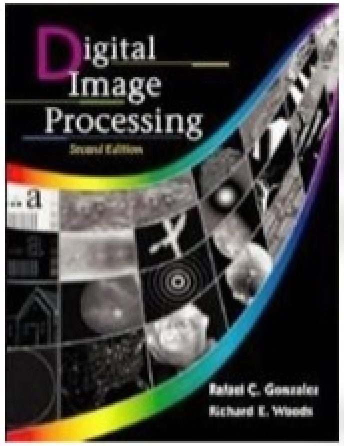
These slides should not be used as the primary source of data. Students are encouraged to learn from the core textbooks and reference books. Contents in these slides are copyrighted to the instructor and authors of original texts where applicable. –Mohan Bhandari

Image Enhancement in the Frequency Domain

REFERENCES



“Digital Image Processing”, Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002

Much of the material that follows is taken from this book

Slides by Brian Mac Namee
Brian.MacNamee@comp.dit.ie

Background

- The French mathematician Jean Baptiste Joseph Fourier
  - Born in 1768
  - Published Fourier series in 1822
  - Fourier's ideas were met with skepticism
- Fourier series
  - Any periodical function can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient

Frequency Domain

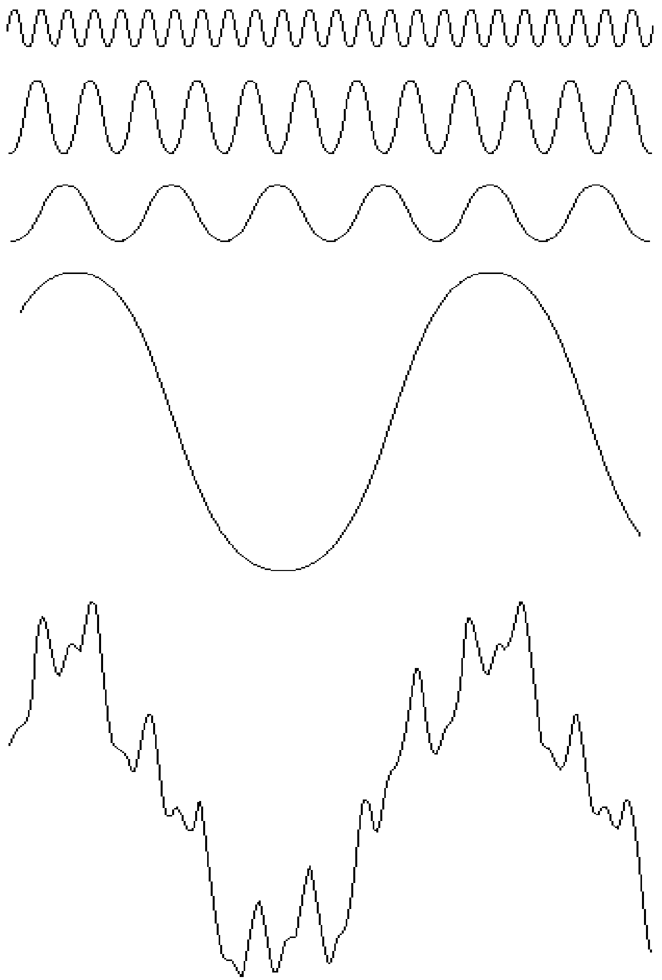


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Frequency Domain

It does not matter how complicated the function is; as long as it is periodic and meets some mild mathematical conditions, it can be represented by such a sum.

Even functions that are not periodic (but whose area under the 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 the *Fourier Transform*, and its utility is even greater than the Fourier series in most practical problem.

Both representations share the important characteristic that a function, expressed in either a Fourier series or transform, can be reconstructed completely via an inverse process with no loss of information.

Fourier Transform

- Fourier transform
  - Functions can be expressed as the integral of sines and/or cosines multiplied by a weighting function
  - Functions expressed in either a Fourier series or transform can be reconstructed completely via an inverse process with no loss of information

Fourier Transform

- Easier to remove undesirable frequencies
- Faster perform certain operations in frequency domain than in spatial domain

Fourier Transform

- Spatial, or image variables: x, y
- Transform, or frequency variables: u, v

Introduction to the Fourier Transform and the Frequency Domain

- The one-dimensional Fourier transform and its inverse
  - Fourier transform

$$F(u)=\int f(x)e^{-j2\pi \mathrm{ux}}\,\mathrm{dx}$$

  - Inverse Fourier transform

$$f(x)=\int F(u)e^{j2\pi \mathrm{ux}}\,\mathrm{du}$$

These two equations comprise the *Fourier transform pair*. These two equations exist if $f(x)$ is continuous and integrable and $F(u)$ is integrable. These conditions are almost always satisfied in practice. We are concerned with functions $f(x)$, which are real, however the Fourier transform of a real function is, generally, complex. So,

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

where $R(u)$ and $I(u)$denote the real and imaginary components of $F(u)$respectively.

Introduction to the Fourier Transform and the Frequency Domain

Expressed in exponential form, $F(u)$is: $F(u) = |F(u)|e^{j\varphi(u)}$

Where, $|F(u)| = \sqrt{R^2(u) + I^2(u)}$ $and\ \varphi(u) = \tan^{-1}\frac{I(u)}{R(u)}$

The magnitude function $|F(u)|$is called the *Fourier spectrum* of $F(x)$ and $\varphi(u)$is the phase angle.

The square of the spectrum, $P(u) = |F(u)|^2 = R^2(u) + I^2(u)$, is commonly called the *power spectrum* (or the spectral density) of f(x).

The variable u is often called the frequency variable. This name arises from the expression of the exponential term $e^{-j2\pi ux}$in terms of sines and cosines (from Euler's formula):

$$e^{-j2\pi ux} = \cos(2\pi ux) - j\sin(2\pi ux)$$

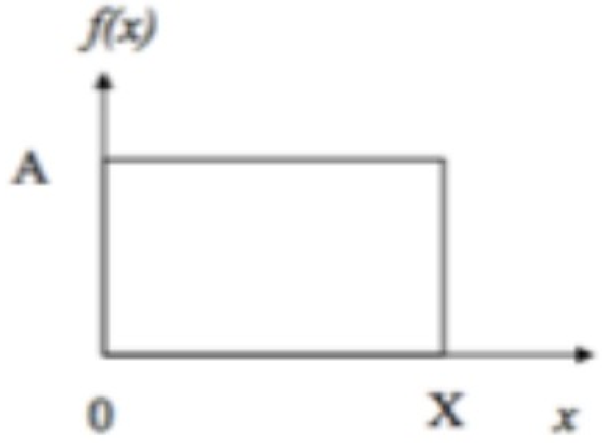
Introduction to the Fourier Transform and the Frequency Domain

Interpreting the integral in the Fourier transform equation as a limit summation of discrete terms make it obvious that:

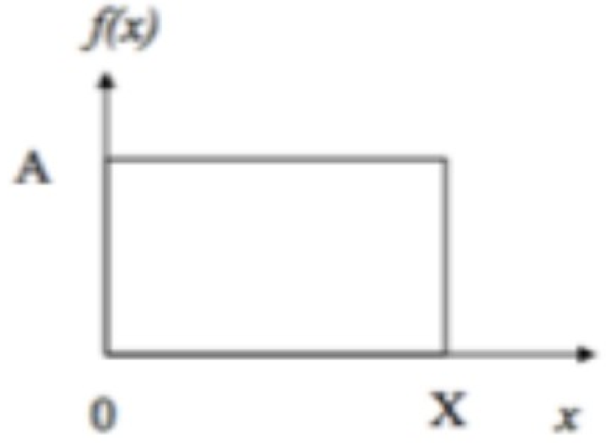
- $F(u)$is composed of an infinite sum of sine and cosine terms.
- Each value of u determines the frequency of its corresponding sine-cosine pair.

Introduction to the Fourier Transform and the Frequency Domain

Consider the following simple function.



The Fourier transform is:

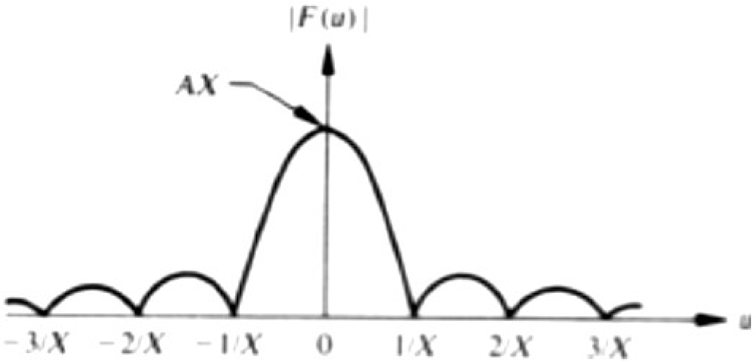


$$
\begin{aligned}
F(u) &= \int_{-\infty}^{+\infty} f(x)\exp[-j2\pi ux]dx \\
&= \int_{0}^{X} A\exp[-j2\pi ux]dx \\
&= \frac{-A}{j2\pi u}[e^{-j2\pi ux}]_0^X = \frac{-A}{j2\pi u}[e^{-j2\pi uX}-1] \\
&= \frac{A}{j2\pi u}[e^{j\pi uX}-e^{-j\pi uX}]e^{-j\pi uX} \\
&= \frac{A}{\pi u}\sin(\pi uX)e^{-j\pi uX}
\end{aligned}
$$

This is a complex function. The Fourier spectrum is:

$$|F(u)| = \left|\frac{A}{\pi u}\right| |\sin(\pi u X)||e^{-j\pi u x}|$$

$$= AX\left|\frac{\sin(\pi u X)}{(\pi u X)}\right|$$



2D Fourier Transform

The Fourier transform can be extended to 2 dimensions:

$$F(u,v) = \iint_{-\infty}^{\infty} f(x,y) e^{-j2\pi(ux+vy)}\, dx\, dy$$

and the inverse transform

$$f(x,y) = \iint_{-\infty}^{\infty} f(u,v) e^{j2\pi(ux+vy)}\, dx\, dy$$

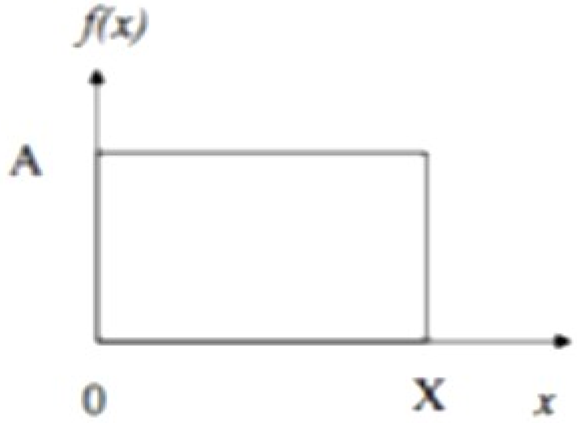
The 2-D Fourier spectrum is:

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

The phase angle is: $\varphi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$

The power spectrum is: $P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$

2D Fourier Transform



$$F(u,v)=\int_{-\infty}^{+\infty}\int f(x,y)\exp[-j2\pi(ux+vy)]dxdy$$

$$=A\int_0^X \exp[-j2\pi ux]dx\int_0^Y \exp[-j2\pi vy]dy$$

$$=A[\frac{e^{-j2\pi ux}}{-j2\pi u}]_0^X[\frac{e^{-j2\pi vy}}{-j2\pi v}]_0^Y$$

$$=\frac{A}{-j2\pi u}[e^{-j2\pi uX}-1]\frac{1}{-j2\pi y}[e^{-j2\pi vY}-1]$$

$$=AXY[\frac{\sin(\pi uX)e^{-j\pi uX}}{(\pi uX)}][\frac{\sin(\pi vY)e^{-j\pi vY}}{(\pi vY)}]$$

The spectrum is $|F(u,v)|=AXY[\frac{\sin(\pi uX)}{(\pi uX)}][\frac{\sin(\pi vY)}{(\pi vY)}]$

Discrete Fourier Transform (DFT) is purely discrete

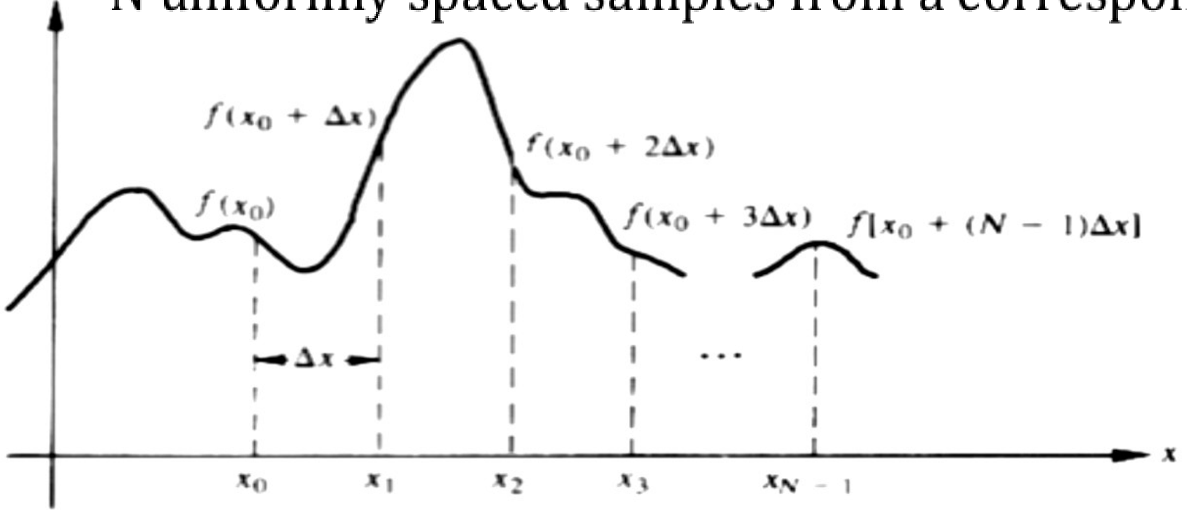
Discrete-time data sets are converted into a discrete-frequency representation.

Suppose a continuous function, f(x), is discretized into a sequence by taking N samples Δx units apart.

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

Let x refer to either a continuous or discrete value by saying $f(x) = f(x_0 + x\Delta x)$

Where x assumes the discrete values 0, 1, ... , N-1 and $f(0), f(1), \dots, f(N-1)$denotes any N uniformly spaced samples from a corresponding continuous function.



Discrete Fourier Transform

The discrete Fourier transform is given by:

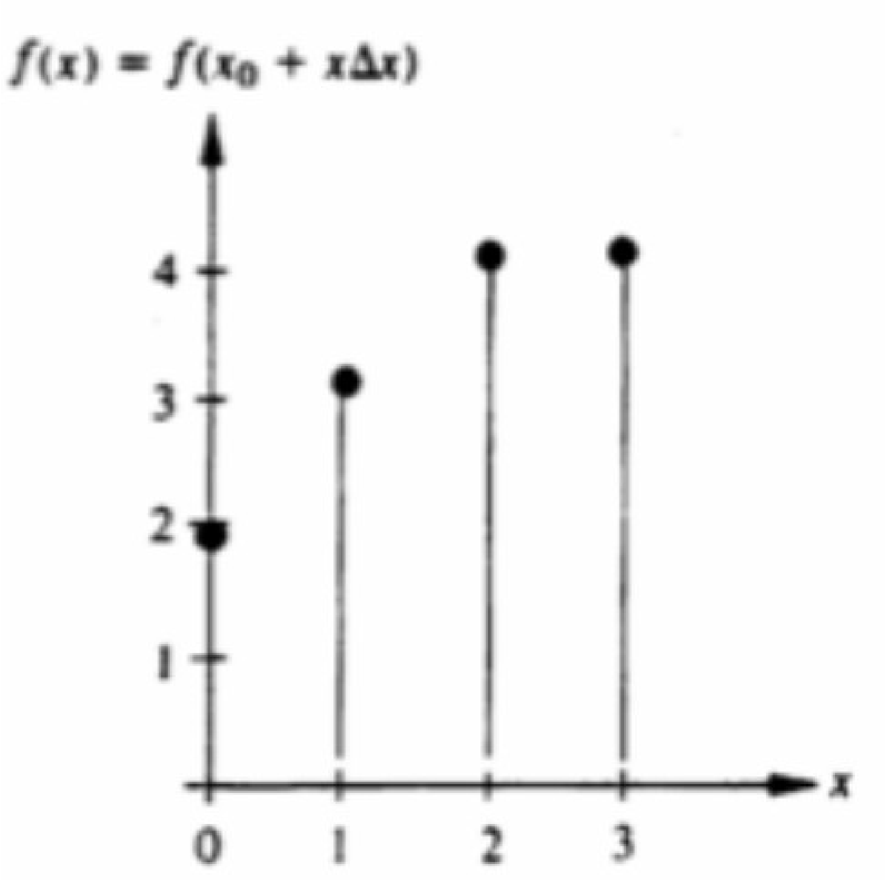
$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j2\pi ux/N}\,;\mathrm{u} = 0, 1, \dots, \mathrm{N} - 1$$

The discrete inverse Fourier transform is given by:

$$f(x) = \sum_{x=0}^{N-1} f(u)e^{j2\pi ux/N}\,; x = 0, 1, \dots, N - 1$$

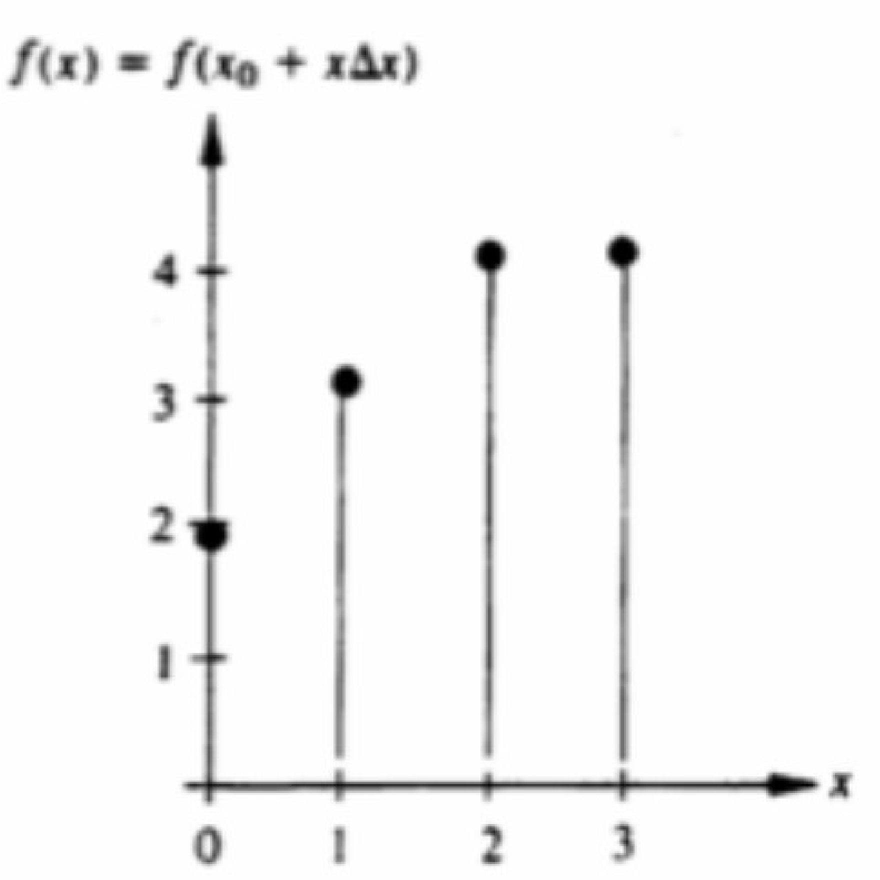
The values of $\mathrm{u} = 0, 1, \dots, \mathrm{N} - 1$in the discrete case correspond to samples of the continuous transform at $0, \Delta\mathrm{u}, 2\Delta\mathrm{u}, \dots, (\mathrm{N} - 1)\Delta u$ and $\Delta\mathrm{u} = \frac{1}{N\Delta x}$

Question



Find the Fourier Spectrum, Phase angle and Power Spectrum for N = 0 and 1

Solution



Given,

f(0) =2
f(1) =3
f(2) =4
f(3) =4

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

Where N= 4 (0,1,2,3)

Solution

For U = 0

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

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

$$F(0)=\frac{1}{4}[2+3+4+4]$$

$$F(0)=\frac{13}{4}$$

Fourier Spectrum

$$|F(u)| = \sqrt{R^2(u) + I^2(u)} \quad = \quad \sqrt{(\frac{13}{4})^2} \quad = \quad \frac{13}{4}$$

Power Spectrum

$$|F(u)|^2 = R^2(u) + I^2(u) \quad = \quad (\frac{13}{4})^2 \quad = \quad \frac{169}{16}$$

Phase Angle

$$\varphi(u) = \tan^{-1}\frac{I(u)}{R(u)} \quad = \tan^{-1}\frac{0}{13/4} \quad = \tan^{-1}\frac{4}{13} \quad = 0.2984$$

Solution

For U = 1

$e^{-j2\pi ux} = \cos(2\pi ux) - j\sin(2\pi ux)$

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

$$F(1) = \frac{1}{4}\sum_{x=0}^{3} f(x)e^{(-j\pi*x/2)}$$

$$F(1) = \frac{1}{4}[f(0)e^{(-j\pi*0/2)} + f(1)e^{(-j\pi*1/2)} + f(2)e^{(-j\pi*2/2)} + f(3)e^{(-j\pi*3/2)}]$$

$$F(1) = \frac{1}{4}[f(0)e^{0} + f(1)e^{(-j\pi/2)} + f(2)e^{(-j\pi)} + f(3)e^{(-j\pi*3/2)}]$$

$$F(1) = \frac{1}{4}[2*1 + 3*(\cos\frac{\pi}{2} - j\sin\frac{\pi}{2}) + 4*(\cos\pi - j\sin\pi) + 4*(\cos\frac{3\pi}{2} - j\sin\frac{3\pi}{2})]$$

$$F(1) = \frac{1}{4}[2*1 + 3*(0 - j*1) + 4*(-1 - j*0) + 4*(0 - j*-1)]$$

$$F(1) = \frac{1}{4}[2 - 3j - 4 + 4j]$$

$$F(1) = \frac{1}{4}[j - 2] = -\frac{1}{2} + \frac{1}{4}j$$

Solution

For U = 1

Fourier Spectrum

$$|F(u)| = \sqrt{R^2(u) + I^2(u)} = \sqrt{\left(\frac{-1}{2}\right)^2 + \left(\frac{1}{4}\right)^2} = \frac{\sqrt{5}}{4}$$

Power Spectrum

$$|F(u)|^2 = R^2(u) + I^2(u) = \left(\frac{-1}{2}\right)^2 + \left(\frac{1}{4}\right)^2 = \frac{5}{16}$$

Phase Angle

$$\varphi(u) = \tan^{-1}\frac{I(u)}{R(u)} = \tan^{-1}\frac{1/4}{-1/2} = -0.4636$$

Practice

Calculate for U=2 and 3

$$|F(2)| = [(1/4)^2 + (0/4)^2]^{1/2} = 1/4$$

$$|F(3)| = [(2/4)^2 + (1/4)^2]^{1/2} = \sqrt{5}/4$$

2-D Discrete Fourier Transform

In the 2-D case:

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

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

2-D Discrete Fourier Transform

The discrete function f(x,y) represents samples of the continuous function at

$$f(x_0 + x\Delta x, y_0 + y\Delta y)$$

$$\Delta u = \frac{1}{M\Delta x} and \Delta v = \frac{1}{N\Delta y}$$

when N=M (such as in a square image)

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

And

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

2-D Discrete Fourier Transform

Magnitude or spectrum

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

Phase angle or phase spectrum

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

Power spectrum or spectral density

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

Distance Function

- To generate filter transfer functions, it is necessary to know the distance of each element of the transfer function to the origin (0,0). In the filter transfer function equations, this distance to the origin is denoted as ($D(u, v)$) , or simply as D .

- Since the frequency domain data ranges from 0 to 2π in both the horizontal and vertical direction, rather than between –π and π, it is convenient to use a special function to calculate the distances.

Cutoff Frequency

- The transition point between the pass and stop bands of the filter is denoted as D_0

Band Width

- For band-reject and band-pass filters, where the data between lower and upper cutoff frequencies is either rejected or passed, the width is denoted as W .

Properties Of 2D Fourier Transform

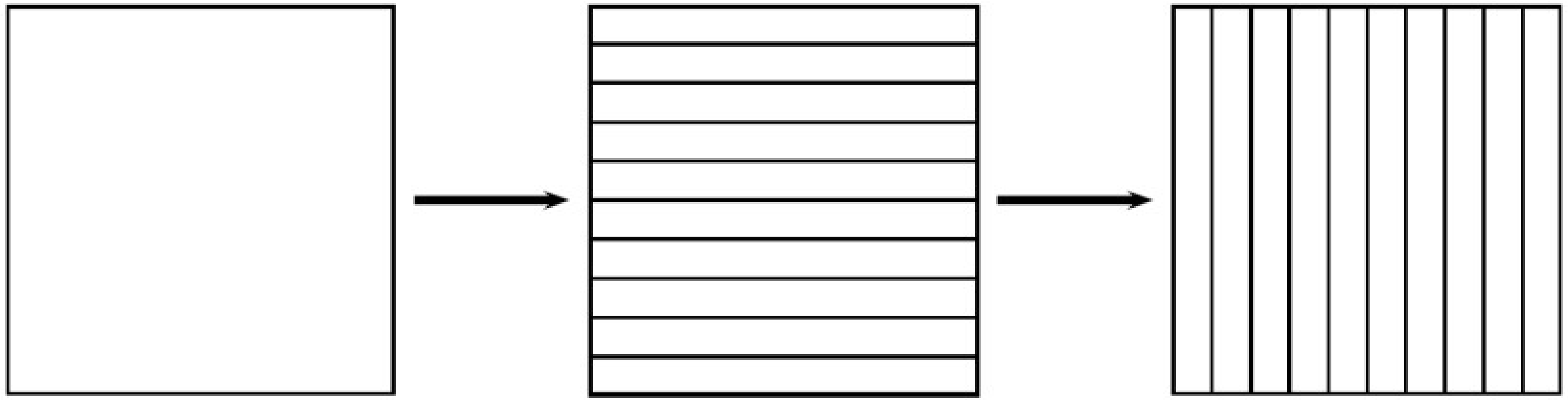
The dynamic range of the Fourier spectra is generally higher than can be displayed. A common technique is to display the function

$$D(u,v) = c\,log[1 + |F(u,v)|]$$

where c is a scaling factor and the logarithm function performs "compression" of the data. 'c' is usually chosen to scale the data into the range of the display devices [0,255]

Properties Of 2D Fourier Transform

Separability



(a) Original image (b) DFT of each row of (a) (c) DFT of each column of (b)

Properties Of 2D Fourier Transform
Separability

2D Fourier Transforms can be implemented as a sequence of 1D Fourier Transform operations performed *independently* along the two axis

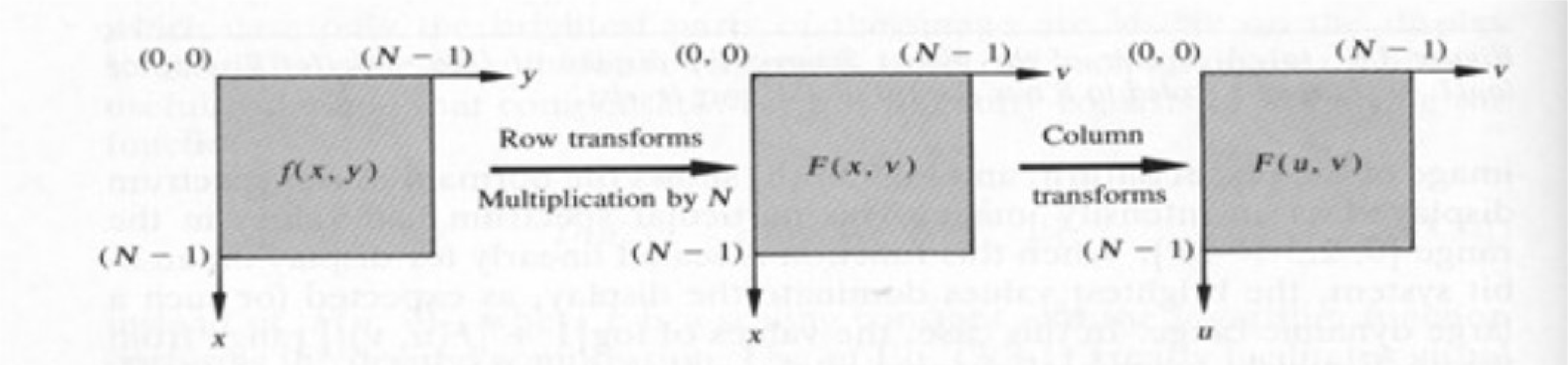
$$F(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy=$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi ux}e^{-j2\pi vy}dxdy=\int_{-\infty}^{\infty} e^{-j2\pi vy}dy\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi ux}dx=$$

$$=\int_{-\infty}^{\infty} F(u,y)e^{-j2\pi vy}dy=F(u,v)$$

2D DFT $\Rightarrow$ 1D DFT along the rows $\rightarrow$ 1D DFT along the cols

Properties Of 2D Fourier Transform
Separability



The main advantage of separability property is that a Fourier Transform of any dimension can be performed by applying 1D transform on each dimension

Properties Of 2D Fourier Transform
Translation

The translation properties of the Fourier transform pair are

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

and

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

where the double arrow indicates a correspondence between a function and its Fourier transform (or vice versa). Multiplying f(x,y) by the exponential and taking the transform results in a shift of the origin of the frequency plane to the point (u_0, v_0).

Properties Of 2D Fourier Transform

Periodicity

The discrete Fourier transform (and its inverse) are periodic with period N.

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

Although F(u,v) repeats itself infinitely for many values of u and v, only N values of each variable are required to obtain f(x,y) from F(u,v).

i.e. Only one period of the transform is necessary to specify F(u,v) in the frequency domain.

Properties Of 2D Fourier Transform
Conjugate Symmetry

If f(x,y) is real (true for all of our cases), the Fourier transform exhibits conjugate symmetry: F(u,v)=F*(-u,-v); F*(u,v) is the complex conjugate of F(u,v)

Or the more interesting: |F(u,v)|=|F(-u,-v)|

Example-1

$f(t) = e^{jt}$ $\qquad f^*(-t) = e^{(-j)(-t)} = e^{jt} = f(t)$

$f(-t) = e^{j(-t)}$ $\qquad$ Hence, $f(t) = f^*(-t)$

Properties Of 2D Fourier Transform

Implication of Periodicity and Symmetry

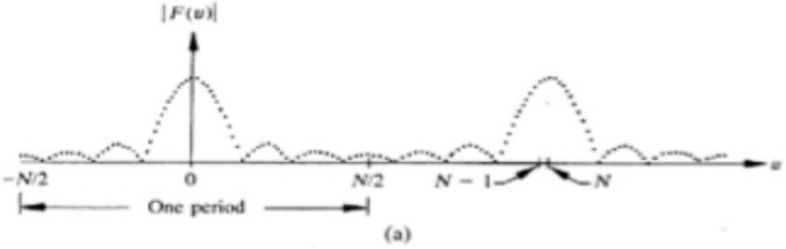
Consider a 1-D case:

$F(u)=F(u+N)$ indicates $F(u)$ has a period of length N

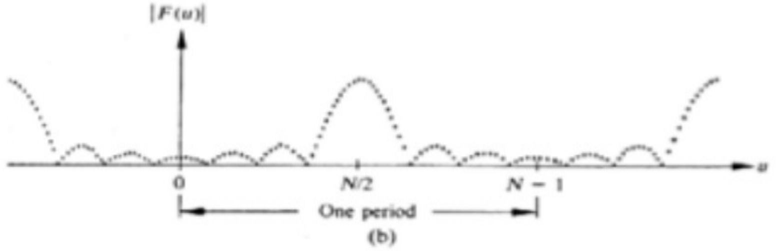
$|F(u)|=|F(-u)|$ shows the magnitude is centered about the origin.

Because the Fourier transform is formulated for values in the range from [0,N-1], the result is two back-to-back half periods in this range. To display one full period in the range, move (shift) the origin of the transform to the point $u=N/2$

Fourier spectrum with back-to-back half periods in the range [0,n-1]

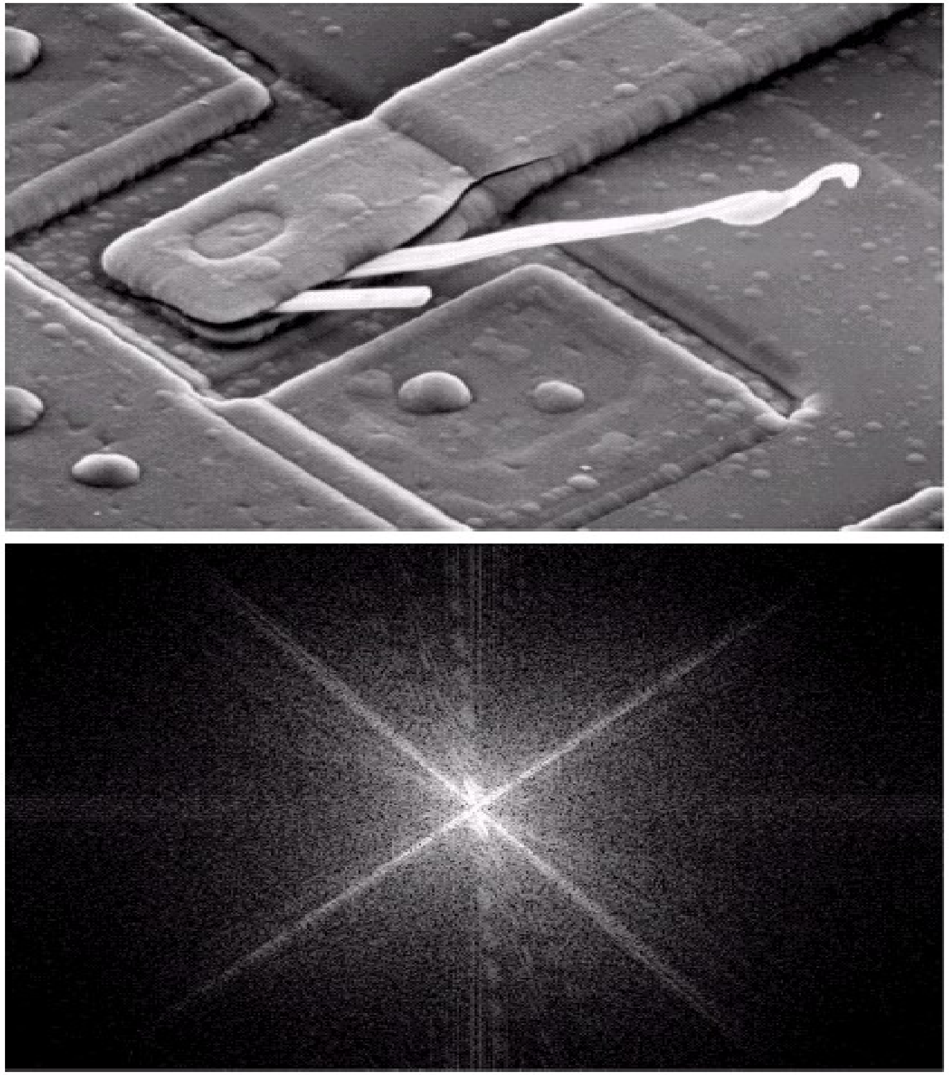


Shifted spectrum with a full period in the same range



Filtering in the frequency domain

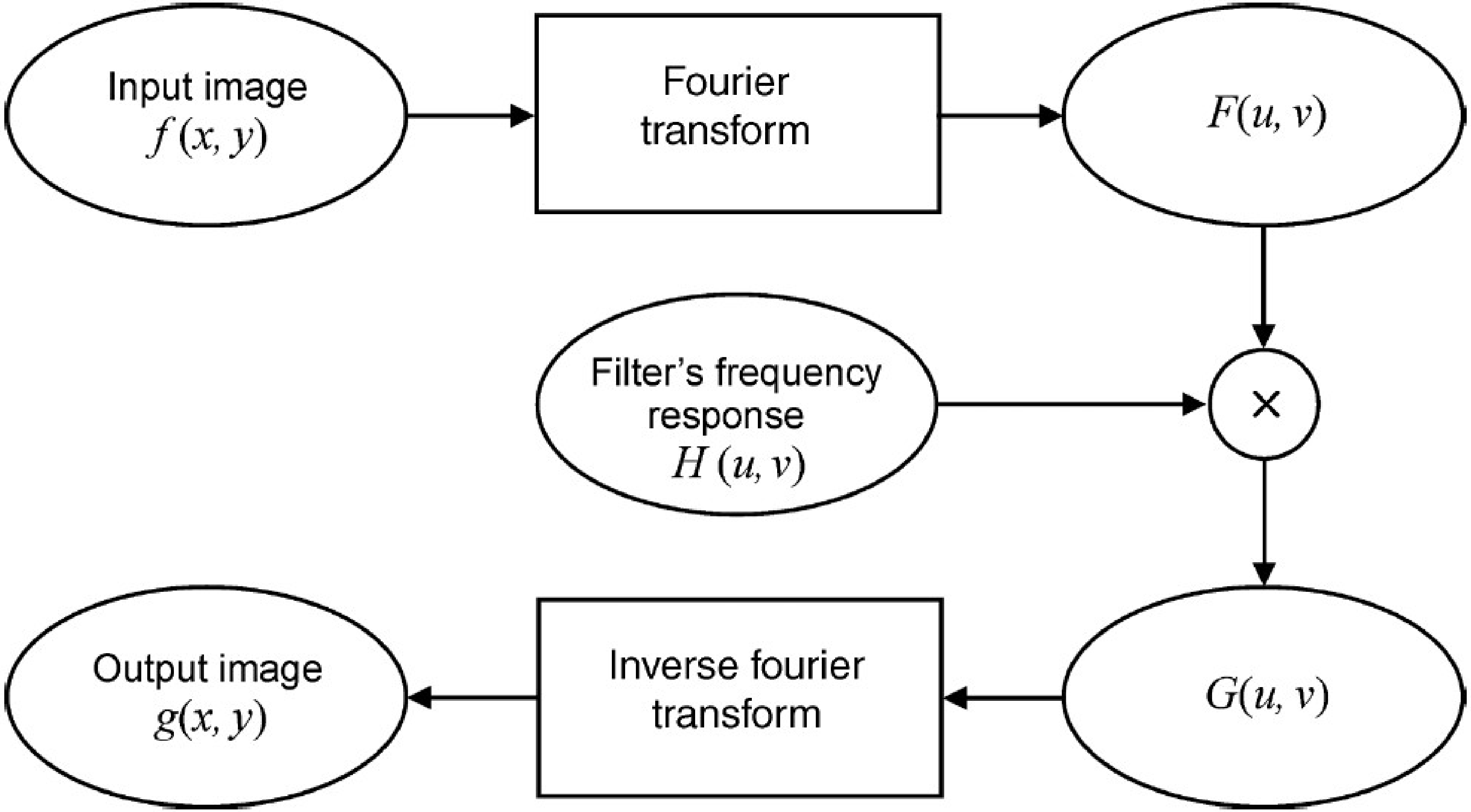
scanning electron microscope



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 in the frequency domain



Filtering in the frequency domain

For each filtering operation, we will describe the construction three types of filters in the frequency domain.

- Ideal Filters
- Gaussian Filter
- Butterworth Filter

Filtering in the frequency domain
Ideal Filters

- Ideal filters are quick and easy to implement and give an easy to understand appreciation of the filter.
- Unfortunately, ideal filters typically yield undesired results.
- Specifically, ideal filters suffer from a problem known as ringing where extra lines appear around the edges of objects in an image

Gaussian Filters

- The shape of a Gaussian filter transfer function is bell-shaped curve that models the probability distribution function
- The smooth transition between the pass-band and stop-band produces good results with no noticeable ringing artifacts. The cutoff frequency of Gaussian filters is determined by a variable (σ).

Filtering in the frequency domain
Butter worth Filters

- The shape of a butter worth filter transfer function is tune-able
- The order of the filter (n) determines the gradient of the transition between the pass-band and stop-band.
- It can assume a gentle transition like in Gaussian filters, or it can assume an abrupt transition like ideal filters.
- A nice aspect of Butter worth filters is that the cutoff frequency is a parameter of transfer function equation. Thus, it is simpler to determine the parameters from examining the image in the frequency domain.

ideal low-pass filter

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

Gaussian low-pass filter

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

Butterworth low-pass filter

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

Smoothing Frequency-Domain Filter
Ideal Low Pass

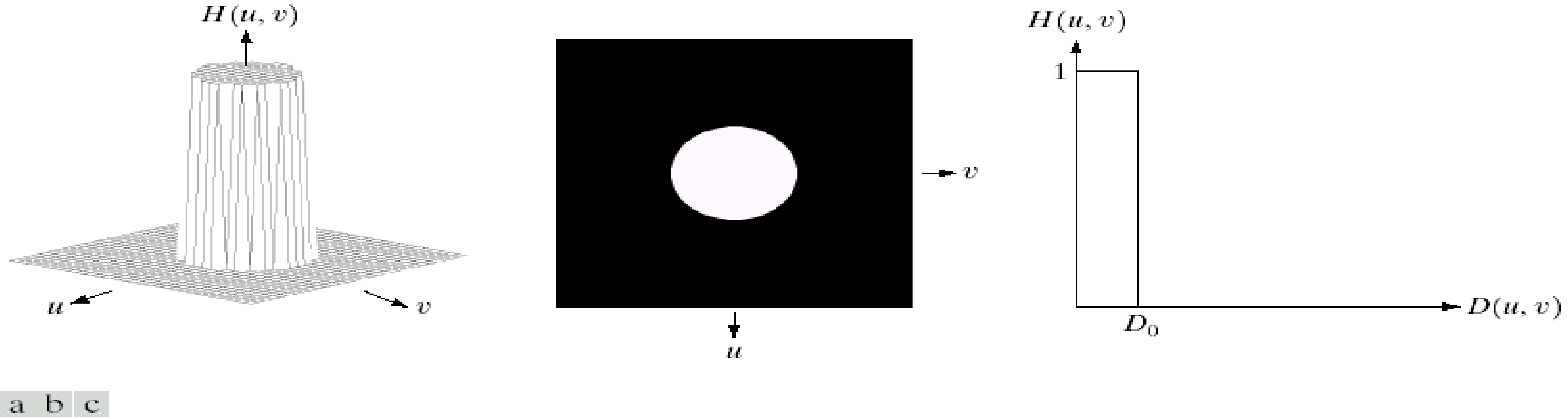
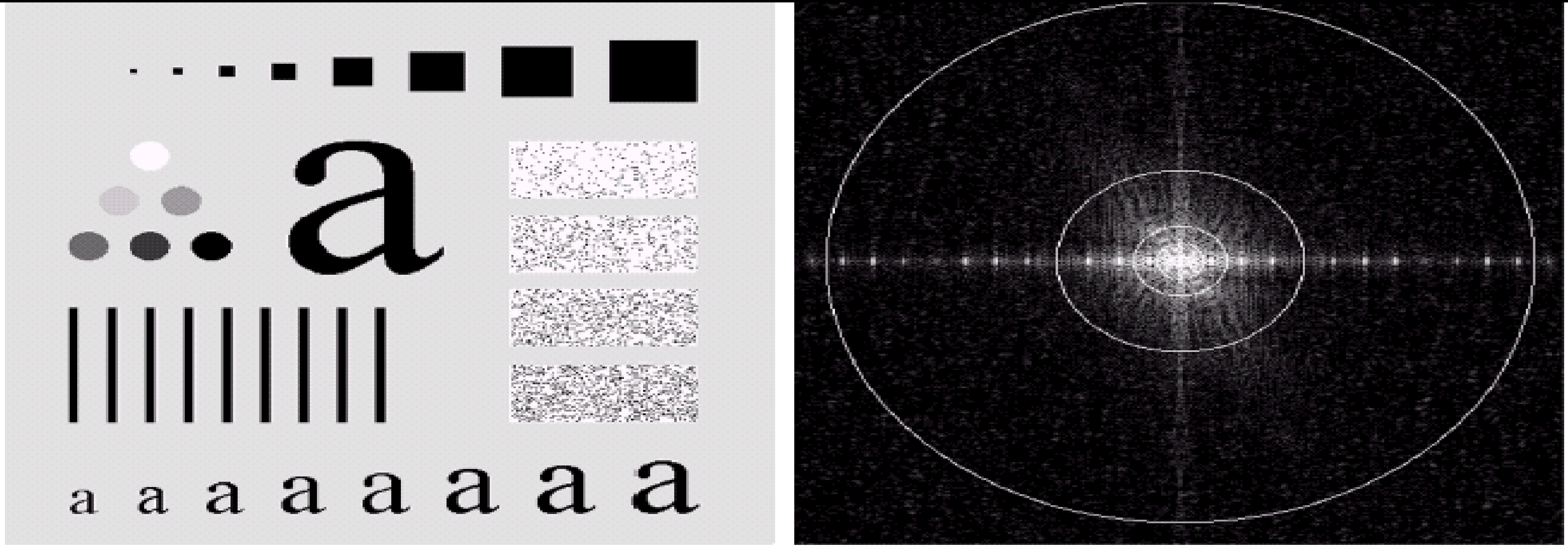


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Smoothing Frequency-Domain Filter

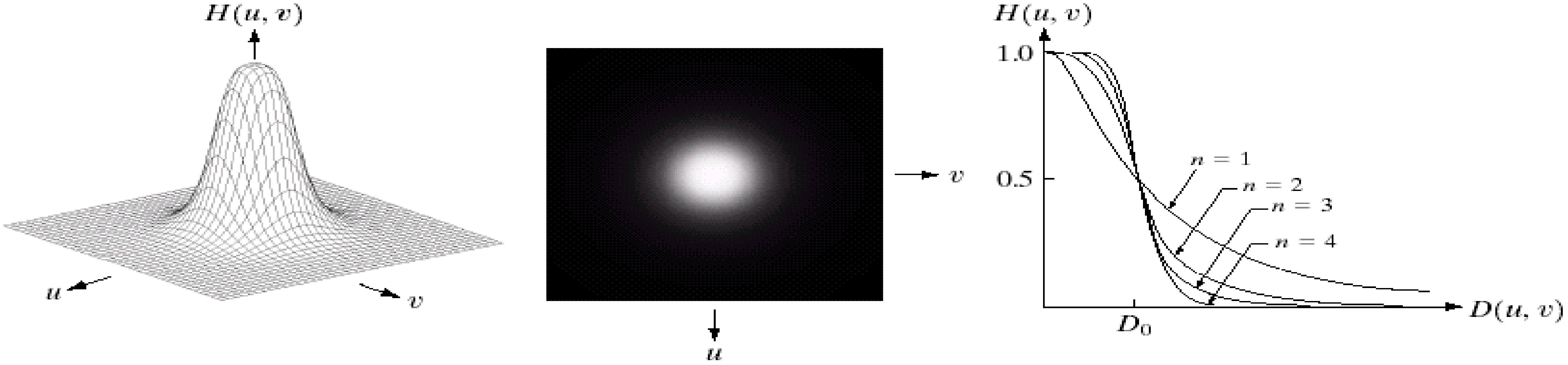
Ideal Low Pass



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.

Smoothing Frequency-Domain Filter
Butterworth Low Pass



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.

Smoothing Frequency-Domain Filter

Butterworth Low Pass

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 Filter
Butterworth Low Pass

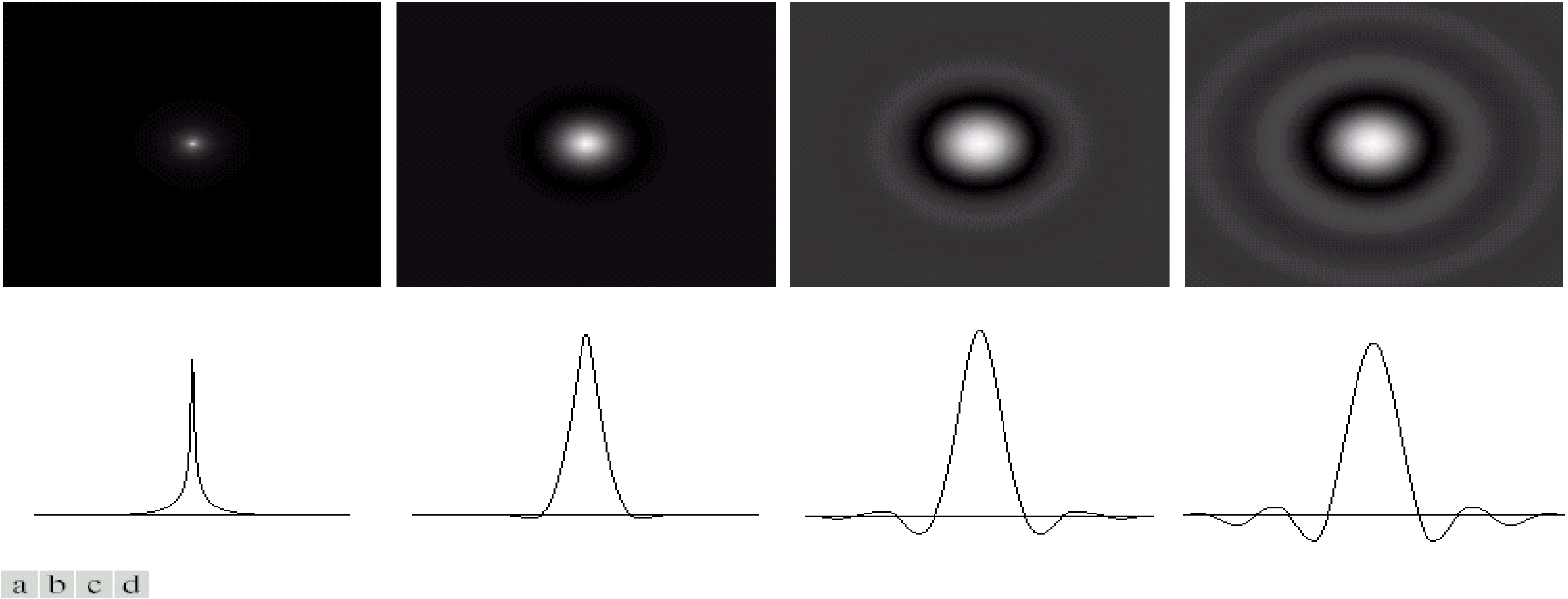
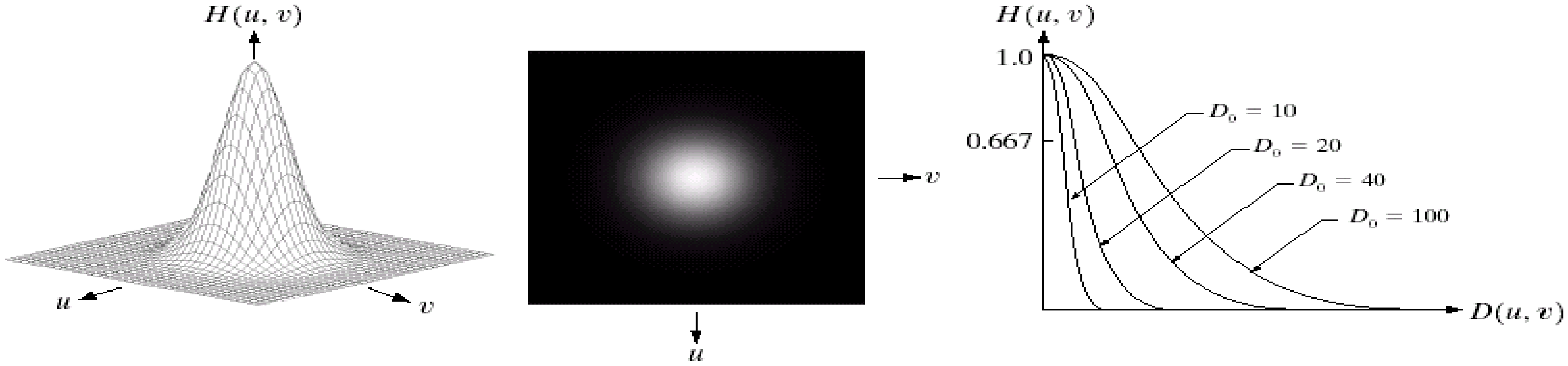


FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Smoothing Frequency-Domain Filter
Gaussain Low Pass



a b c

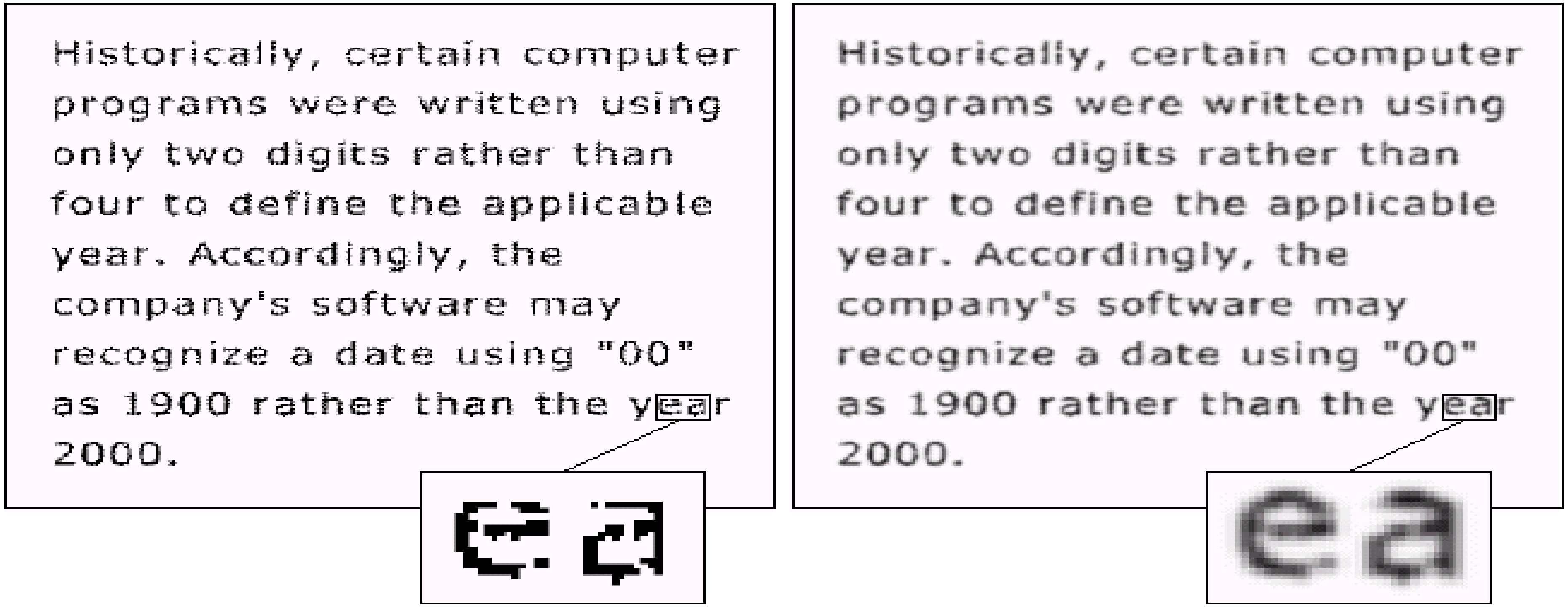
FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Smoothing Frequency-Domain Filter
Gaussain Low Pass

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 Filter
Gaussain Low Pass



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 Filter
Gaussain Low Pass

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).

- High pass filter

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

ideal high-pass filter

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

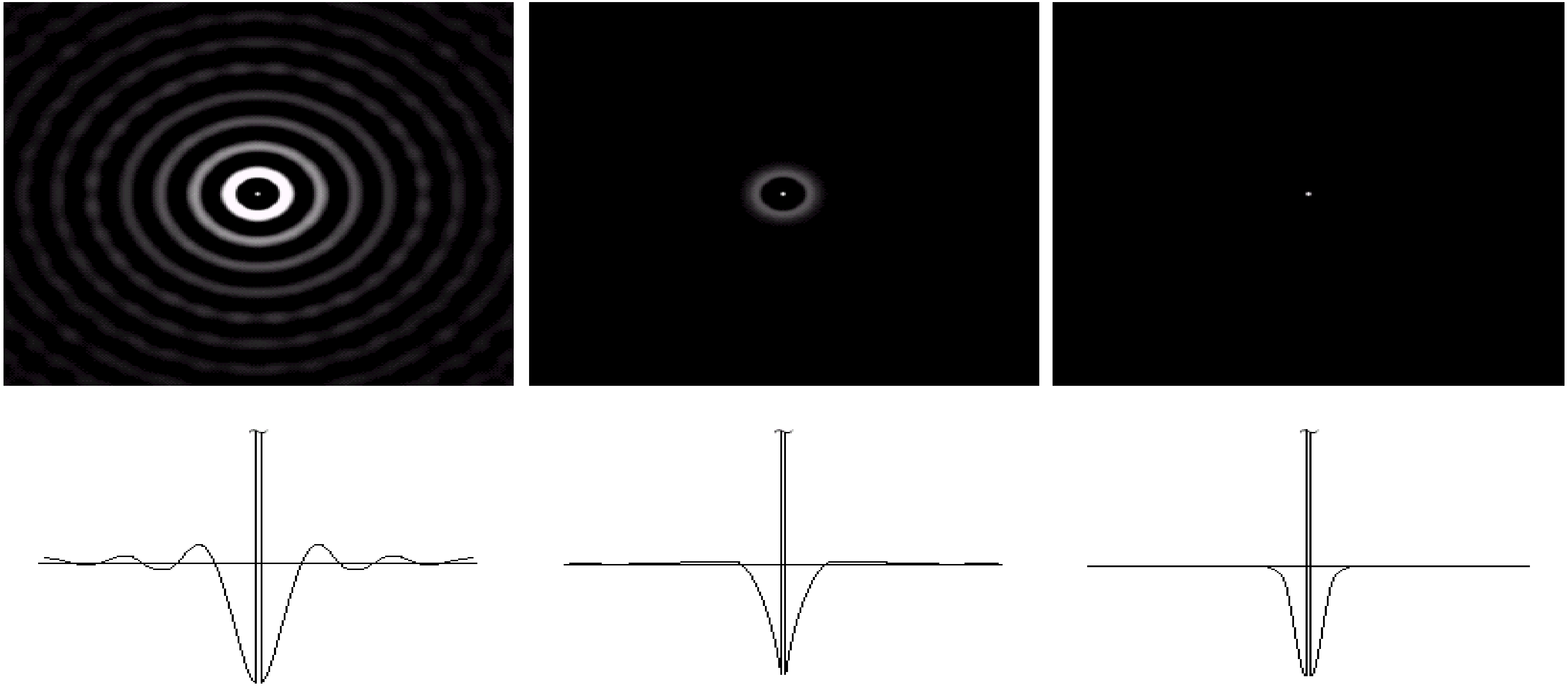
Gaussian high-pass filter

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

Butterworth high-pass filter

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

Sharpening Frequency-Domain Filter
High Pass

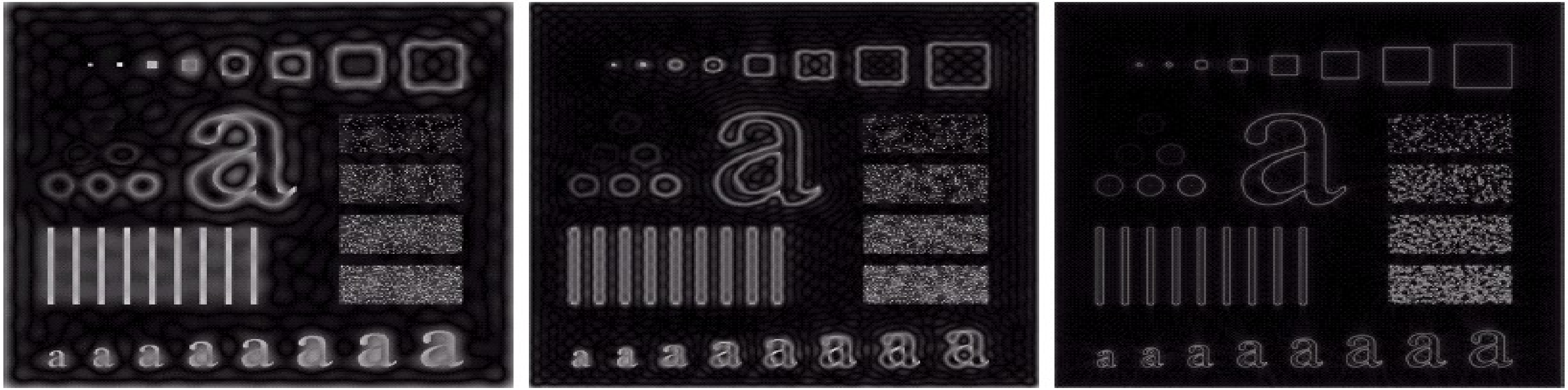


a b c

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 Filter

Ideal High Pass



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).

Sharpening Frequency-Domain Filter
Butterworth High Pass

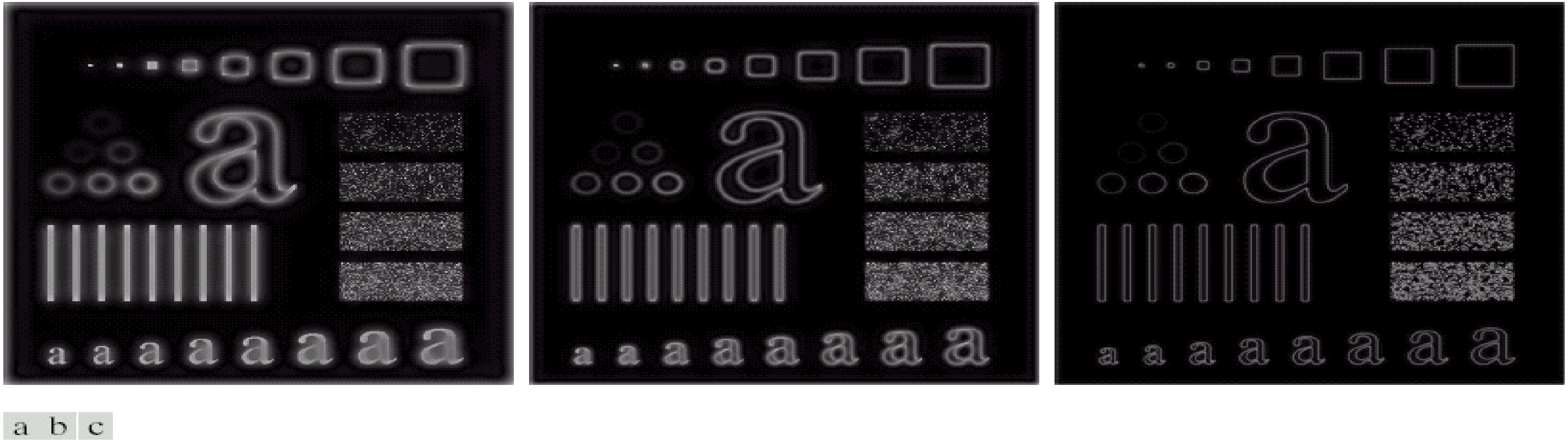
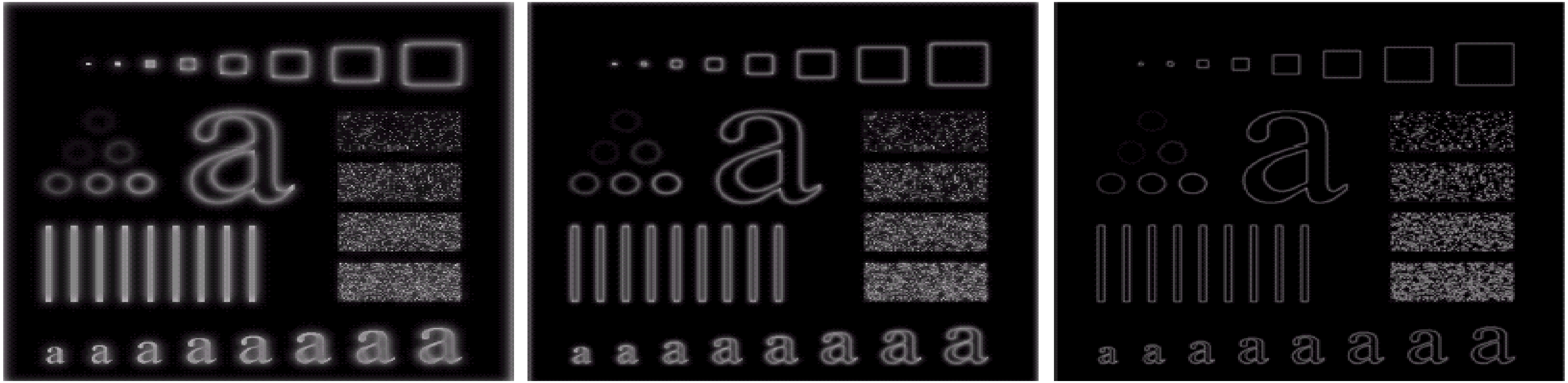


FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

Sharpening Frequency-Domain Filter
Gaussian High Pass



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.

The Laplacian in the frequency domain

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

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

$$= (ju)^2 F(u,v) + (jv)^2 F(u,v)$$

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

Sharpening Frequency-Domain Filter
Laplacian High Pass

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

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

- After centering

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

Fast Fourier Transform

The Discrete Fourier Transform is

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x)\, e^{-i2\pi ux/M}$$

Time Complexity :

$O(M^2)$

Cannot it be made Faster?

Fast Fourier Transform

- In image processing, the most common way to represent pixel location is in the spatial domain by column (x), row (y), and z(value). But sometimes image processing routines may be slow or inefficient in the spatial domain, requiring a transformation to a different domain that offers compression benefits.

- A common transformation is from the spatial to the frequency (or Fourier) domain. The frequency domain is the basis for many image filters used to remove noise, sharpen an image, analyze repeating patterns, or extract features. In the frequency domain, pixel location is represented by its x- and y-frequencies and its value is represented by amplitude.

- The Fast Fourier Transform (FFT) is commonly used to transform an image between the spatial and frequency domain.

  - FFT method **preserves all original data.**

  - FFT fully **transforms images into the frequency domain**

  - FFT **decomposes an image into sines and cosines of varying amplitudes and phases**, which reveals repeating patterns within the image.

Fast Fourier Transform

- **Direct computation takes time:** 2^{2n} multiplications
- **FFT method takes:** $n2^n$ multiplications
- **Time savings:** $2^n/n$

2^n	Direct arithmetic	FFT	Increase in speed
4	16	8	2.0
8	84	24	2.67
16	256	64	4.0
32	1024	160	6.4
64	4096	384	10.67
128	16384	896	18.3
256	65536	2048	32.0
512	262144	4608	56.9
1024	1048576	10240	102.4

The Fast Fourier Transform (FFT) is a way to reduce the complexity of the Fourier transform computation from $O(n^2)$ to $O(n \log n)$ which is a dramatic improvement.

1. Radix-2 is the first FFT algorithm
2. It was proposed by Cooley and Tukey in 1965
3. Though it is not the most common algorithm, it laid the foundation for efficient DFT Calculation.
4. The algorithm appear in
   - Decimation In Time (DIT) , or
   - Decimation in Frequency (DIF)

A radix-2 decimation-in-time (DIT) FFT is the simplest and most common form of the Cooley-Tukey algorithm, although highly optimized Cooley-Tukey implementations typically use other forms of the algorithm as described below. Radix-2 DIT divides a DFT of size N into two interleaved DFTs (hence the name "radix-2") of size N/2 with each recursive stage.

Assignment (FFT → DIT)

$x(n) = \{1,1,1,0,0,0,0,0\}$

If we let

$$W_M = e^{-i2\pi/M}$$

the Discrete Fourier Transform can be written as

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x)\ {W_M}^{ux}$$

If M is a multiple of 2, $M = 2K$ for some positive number K.

Substituting $2K$ for M gives

$$F(u) = \frac{1}{2K} \sum_{x=0}^{2K-1} f(x)\ {W_{2K}}^{ux}$$

Separating out the K even terms and the K odd terms,

$$F(u) = \frac{1}{2}\left\{\frac{1}{K}\sum_{x=0}^{K-1} f(2x){W_{2K}}^{u(2x)} + \frac{1}{K}\sum_{x=0}^{K-1} f(2x+1){W_{2K}}^{u(2x+1)}\right\}$$

Notice that

$$W_{2K}^{2u} = W_K^u$$

$$W_{2K}^{2(u+1)} = W_{2K}^{2u}\, W_{2K}^{u} = W_K^u W_{2K}^u$$

So,

$$F(u) = \frac{1}{2}\left\{\frac{1}{K}\sum_{x=0}^{K-1} f(2x)W_K^{ux} + \frac{1}{K}\sum_{x=0}^{K-1} f(2x+1)W_K^{ux}W_{2K}^u\right\}$$

$$F(u) = \frac{1}{2}\left\{\frac{1}{K}\sum_{x=0}^{K-1} f(2x)W_K^{ux} + \frac{1}{K}\sum_{x=0}^{K-1} f(2x+1)W_K^{ux}W_{2K}^{u}\right\}$$

$$F(u) = \frac{1}{2}\left\{F_{\text{even}}(u) + F_{\text{odd}}(u)W_{2K}^{u}\right\}$$

Use this to get the first K terms ($u = 0..K-1$), then re-use these parts to get the last K terms ($u = K..2K-1$):

$$F(u+K) = \frac{1}{2}\left\{F_{\text{even}}(u) - F_{\text{odd}}(u)W_{2K}^{u}\right\}$$

Convolution

Mathematical operation on two functions (f and g) that produces a third function (f*g) that expresses how the shape of one is modified by the other.

Finite length signals (N_0 samples) → circular or periodic convolution

- the summation is over 1 period
- the result is a N_0 period sequence

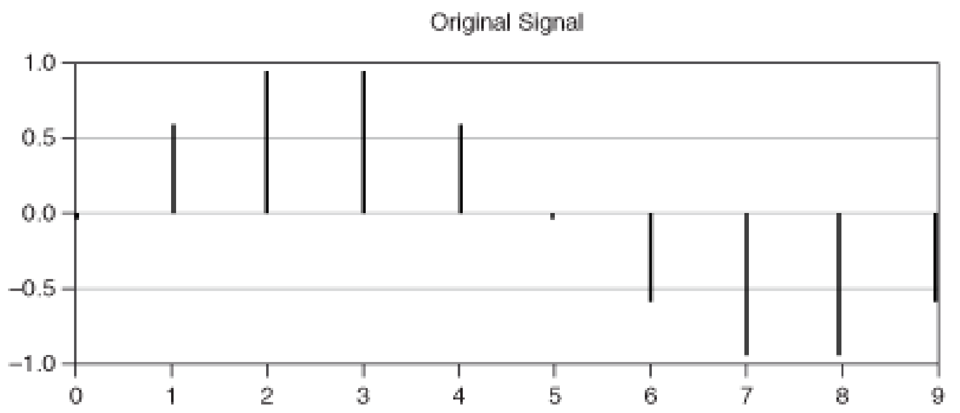
$$\boxed{c[k] = f[k] \otimes g[k] = \sum_{n=0}^{N_0-1} f[n]g[k-n]}$$

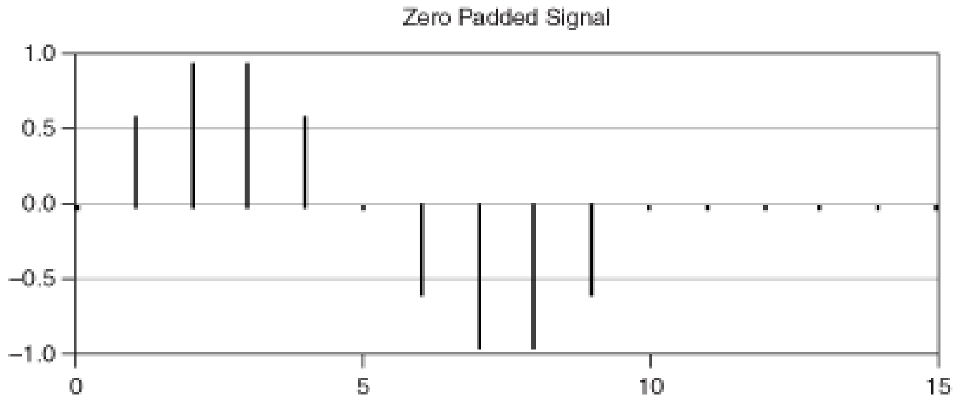
Zero Padding

Zero padding is a technique typically employed to make the size of the input sequence equal to a power of two.

In zero padding, you add zeros to the end of the input sequence so that the total number of samples is equal to the next higher power of two.

For example, if you have 10 samples of a signal, you can add six zeros to make the total number of samples equal to 16, or 32, which is a power of two





Correlation

The correlation theorem says that multiplying the Fourier transform of one function by the complex conjugate of the Fourier transform of the other gives the Fourier transform of their correlation.

That is, take both signals into the frequency domain, form the complex conjugate of one of the signals, multiply, then take the inverse Fourier transform. This is expressed by:

$$f(x) \star g(x) \leftrightarrow F^*(u)G(u)$$

Hadamard Transform

The Hadamard transform H_m is a 2^m x 2^m matrix, the Hadamard matrix, that transforms 2^m real numbers x_n into 2^m real numbers X_k

The Hadamard transform can be defined in two ways:

- recursively,

- binary(base-2) representation of the indices n and k

Hadamard Transform

We define the 1 × 1 Hadamard transform H_0 by the identity $H_0 = 1$, and then define H_m for $m > 0$ by:

$$H_m = \frac{1}{\sqrt{2}}\begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix}$$

$$(H_m)_{k,n} = \frac{1}{2^{m/2}}(-1)^{\sum_j k_j n_j}$$

$$H_0 = +1$$

$$H_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$H_2 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$H_3 = \frac{1}{2^{3/2}}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

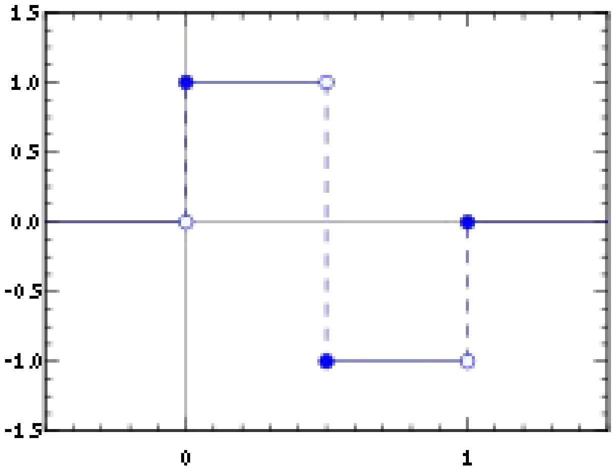
Haar Wavelets

The haar wavelet is a sequence of rescaled "square-shaped" functions which together form a wavelet family or basis.

Wavelet analysis is similar to Fourier analysis in that it allows a target function over an interval to be represented in terms of an orthonormal *(Mutual perpendicular along a line)* basis.

The Haar wavelet's mother wavelet function φ(t) can be described as

$$\psi(t) = \begin{cases} 1 & 0 \le t < \frac{1}{2}, \\ -1 & \frac{1}{2} \le t < 1, \\ 0 & \text{otherwise.} \end{cases}$$



Haar Transform

- The Haar transform is the simplest of the wavelet transforms.
- This transform cross-multiplies a function against the Haar wavelet with various shifts and stretches
- It is found effective in applications such as signal and image compression in electrical and computer engineering as it provides a simple and computationally efficient approach for analysing the local aspects of a signal.

The starting point for the definition of the Haar transform is the Haar functions $h_k(z)$, which are defined in the closed interval [0, 1]. The order k of the function is uniquely decomposed into two integers p, q

$$k = 2^p + q - 1,\ k = 0,\ 1,\ \ldots,\ L - 1, \text{ and } L = 2^n$$

where

$$0 \leq p \leq n - 1,\ 0 \leq q \leq 2^p \text{ for } p \neq 0 \text{ and } q = 0 \; or \; 1 \text{ for } p = 0$$

Haar Transform

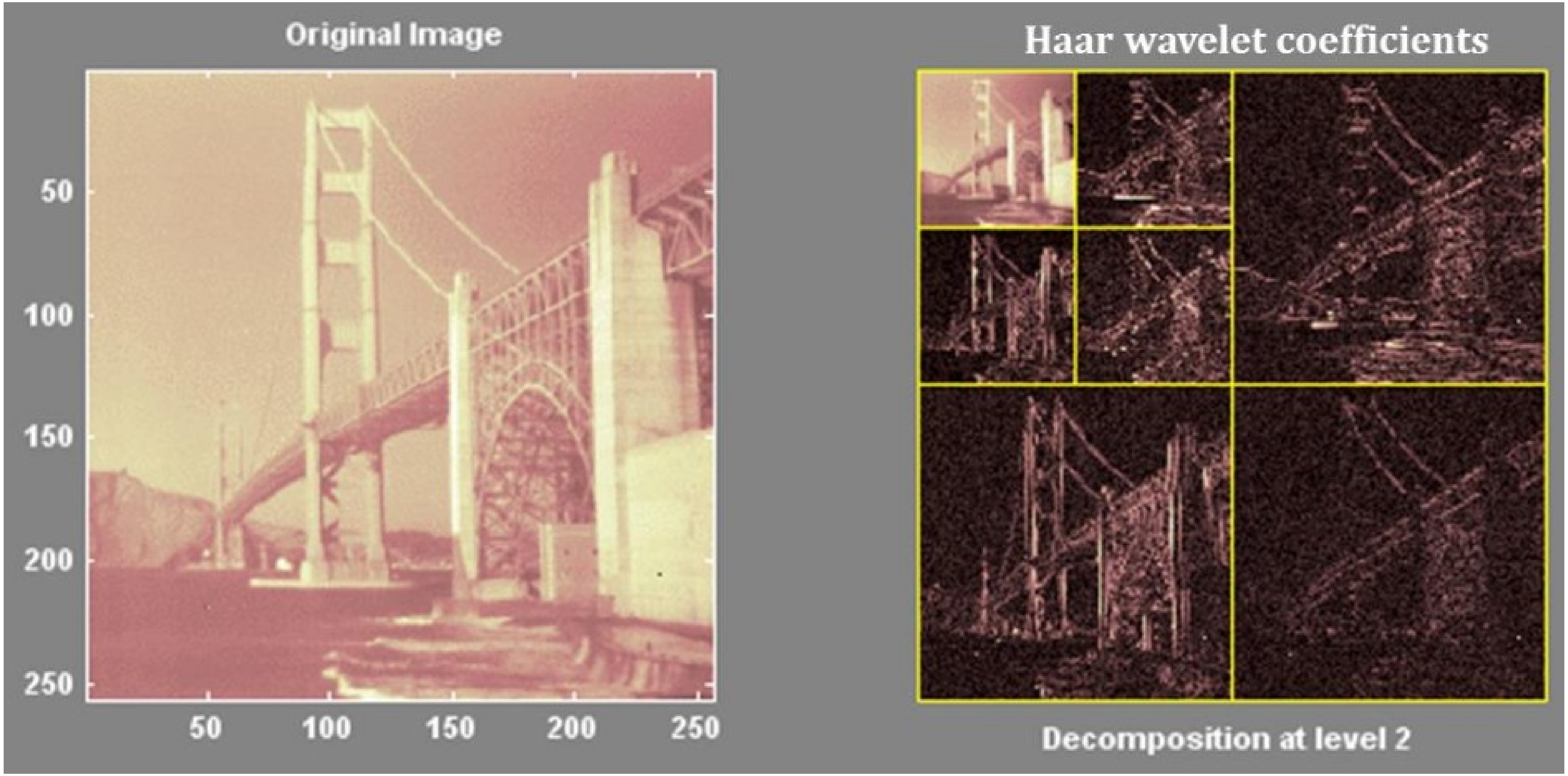
$$h_0(z) \equiv h_{00}(z) = \frac{1}{\sqrt{L}},\ z \in [0,\ 1]$$

$$h_k(z) \equiv h_{pq}(z) = \frac{1}{\sqrt{L}} \begin{cases} 2^{\frac{p}{2}} & \frac{q-1}{2p} \leq z < \frac{q-\frac{1}{2}}{2p} \\ -2^{\frac{p}{2}} & \frac{q-\frac{1}{2}}{2p} \leq z < \frac{q}{2p} \\ 0 & \text{otherwise in } [0,1] \end{cases}$$

The Haar transform matrix of order L consists of rows resulting from the preceding functions computed at the points $z = m/L$, $m = 0, 1, 2, \ldots, L - 1$. For example, the 8 × 8 transform matrix is

$$H = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

Haar Transform



Discrete Cosine Transform (DCT)

based on most common form for 1D DCT

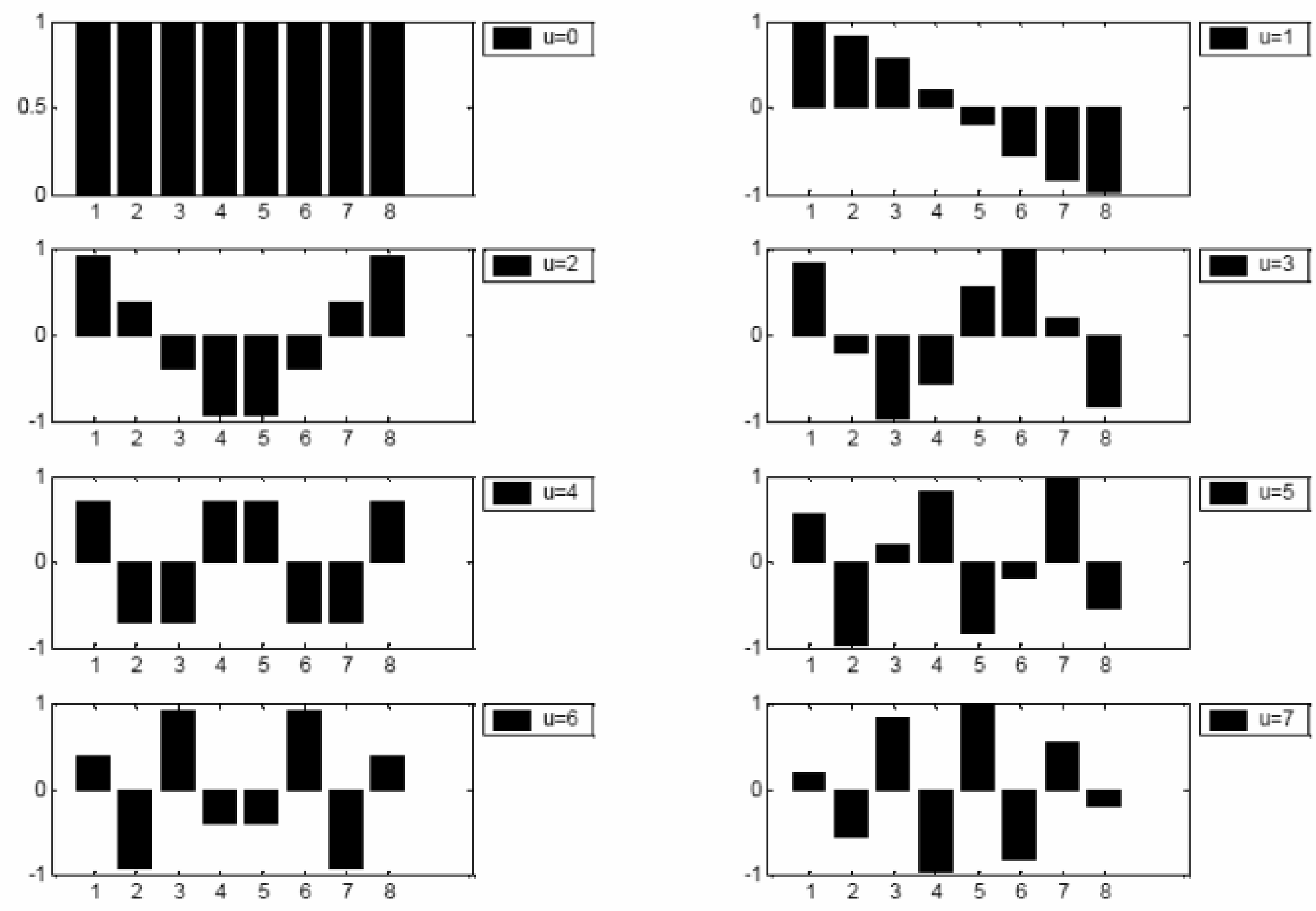
$$C(u)=\alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)u}{2N}\right]; \qquad u,x=0,1,\ldots, N-1$$

$$f(x)=\sum_{u=0}^{N-1}\alpha(u)C(u)\cos\left[\frac{\pi(2x+1)u}{2N}\right],$$

$$\alpha(u)=\begin{cases}\sqrt{\frac{1}{N}} & for \quad u=0\\ \sqrt{\frac{2}{N}} & for \quad u\neq 0.\end{cases}$$

$$C(u=0)=\sqrt{\frac{1}{N}}\sum_{x=0}^{N-1} f(x). \qquad \text{“mean” value}$$

Discrete Cosine Transform (DCT)



Cosine basis functions are orthogonal

Discrete Cosine Transform (DCT)

- Corresponding 2D formulation

direct $$C(u,v)=\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right],$$

u,v=0,1,...., N-1

$$\alpha(u)=\begin{cases}\sqrt{\frac{1}{N}} & for \quad u=0\\ \sqrt{\frac{2}{N}} & for \quad u\neq 0.\end{cases}$$

inverse $$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)C(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right],$$