Fourier transform fundamentals

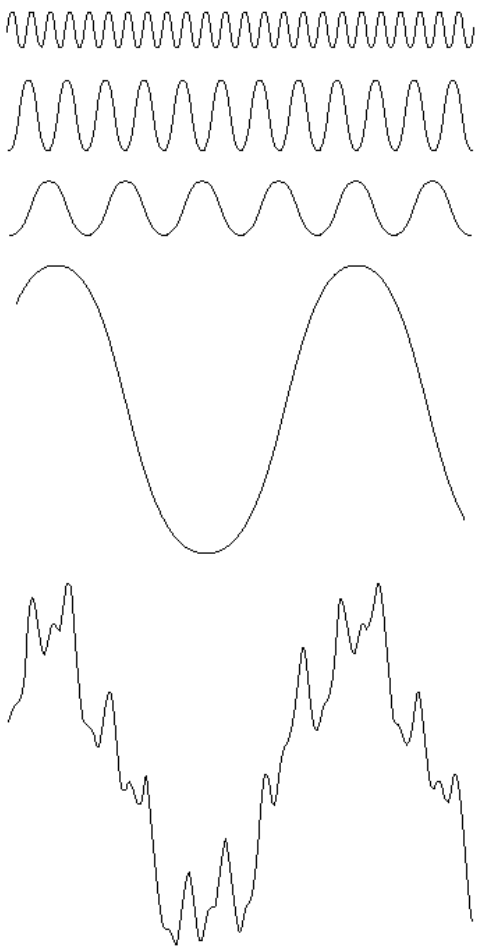


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

1-D case:

Three parameters for the sinusoidal functions:
- Frequency
- Amplitude
- Phase

Zero-frequency sine:
DC value

Fourier transform fundamentals

Let *f(x)* be a continuous (and aperiodic) function of a real variable *x*.
Its **Fourier transform (CTFT)** is

$$\Im\{f(x)\} = F(u) = \int_{-\infty}^{+\infty} f(x)\exp[-j2\pi ux]dx$$

It is a complex function of the real variable *u*:

$$F(u) = R(u) + jI(u) = |F(u)|e^{j\Phi(u)}$$

The **inverse** Fourier **transform** is given by

$$\Im^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{+\infty} F(u)\exp[j2\pi ux]du$$

Fourier transform fundamentals

For a discrete (sampled) finite-length sequence, when *x* takes integer values e.g. in the range *[0,N-1],* the **Discrete Fourier Transform (DFT)** is

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

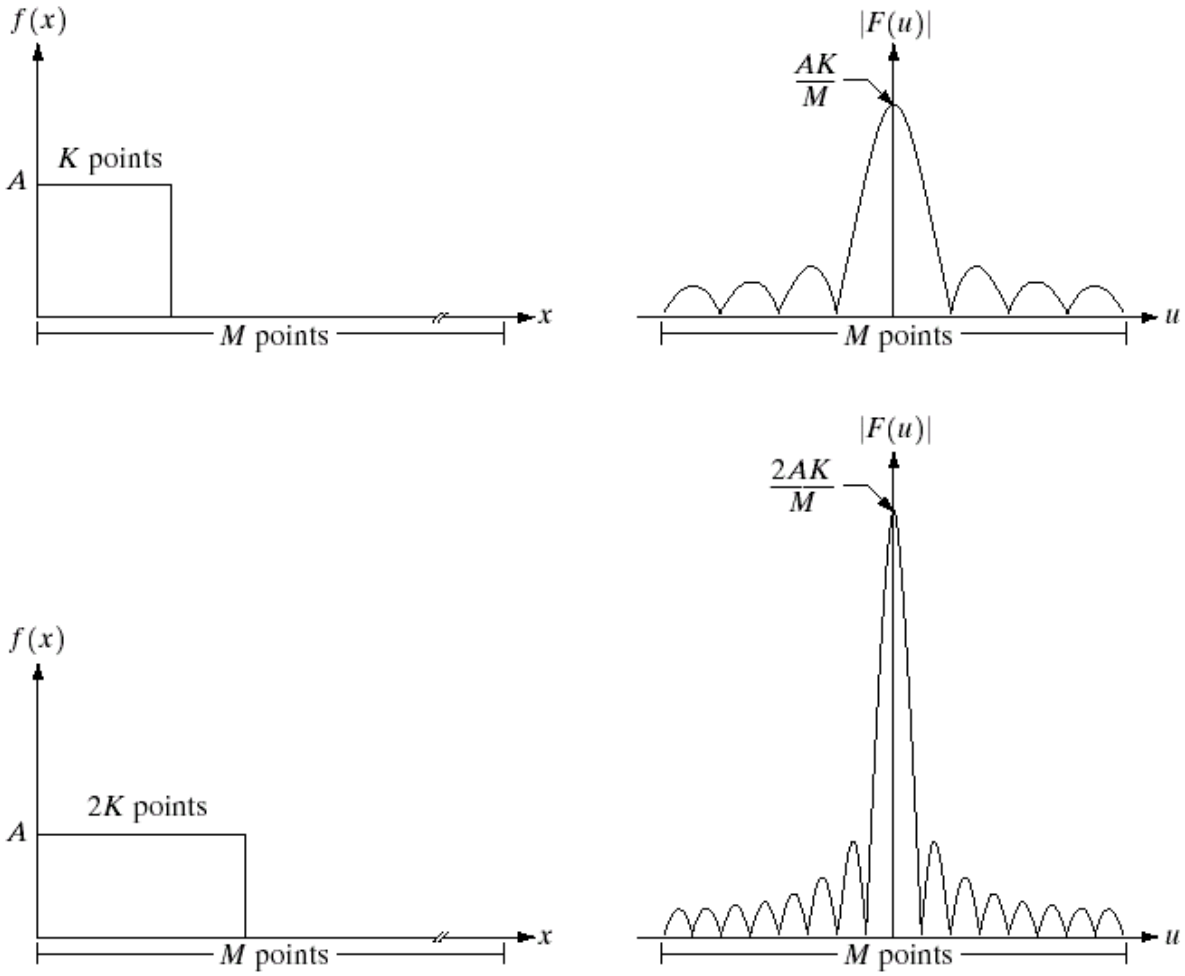
It is a complex function of the discrete variable *u = 0, 1, 2, … N-1*.
Periodic repetition of the input sequence is implied.

The **Inverse Discrete Fourier Transform (IDFT)** is

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[\, j2\pi\, ux / N]$$

Fourier transform fundamentals

E.g.:



a b
c d

FIGURE 4.2 (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.

Fourier transform fundamentals

If *f(x,y)* is a continuous function of **two** real variables *x,y*, the Fourier pair is

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

$$\Im^{-1}\{F(u,v)\} = f(x,y) = \iint_{-\infty}^{+\infty} F(u,v)\exp[j2\pi(ux+vy)]dudv$$

In the discrete, finite-size (implicitly periodic) case, the **2-D DFT** is:

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

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

Fourier transform fundamentals

We define the
- Fourier spectrum,
- phase angle,
- power spectrum

as functions of the **spatial frequencies** *u* and *v*:

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

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

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

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

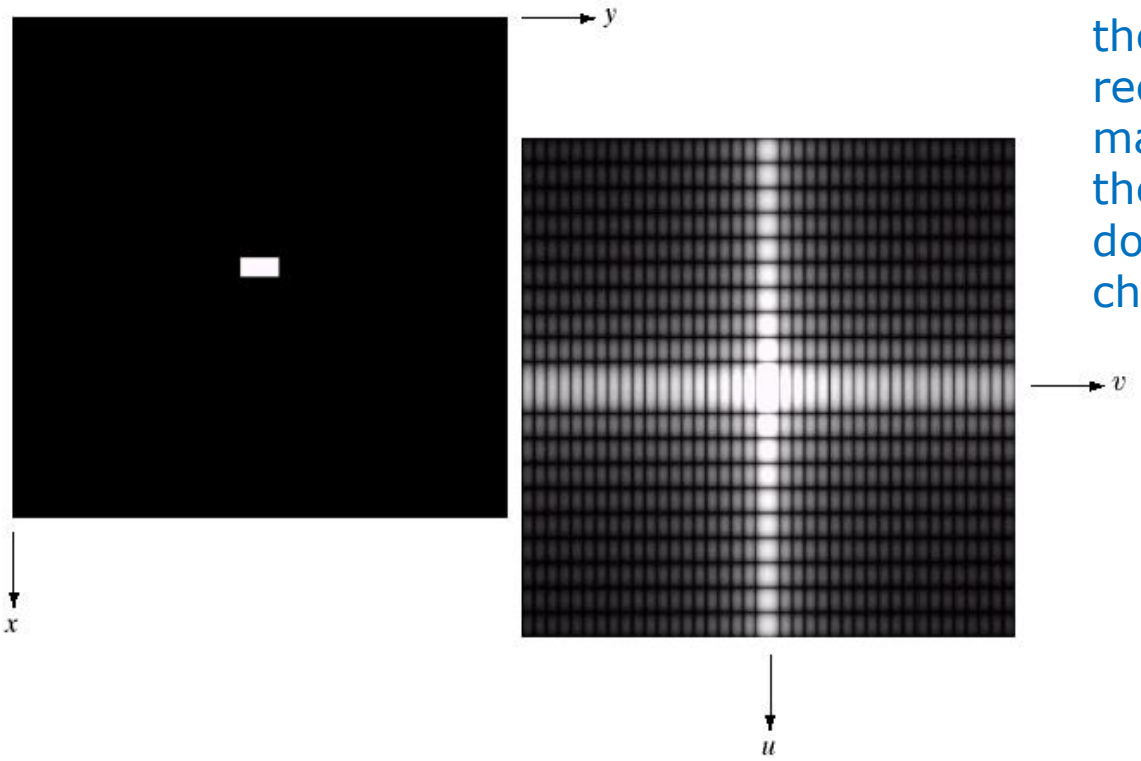
Symmetry: if *f* is real, *F* is conjugate symmetric → the spectrum is **symmetric**:

$$F(u,v) = F^*(-u,-v) \quad \rightarrow \quad |F(u,v)| = |F(-u,-v)|$$

Fourier transform fundamentals

Note: whatever the position of the white rectangle, the magnitude of the transform does not change

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



It is common practice to multiply *f(x,y)* by *(-1)^(x+y)*, which moves the origin of the Fourier transform at *u=M/2, v=N/2*:

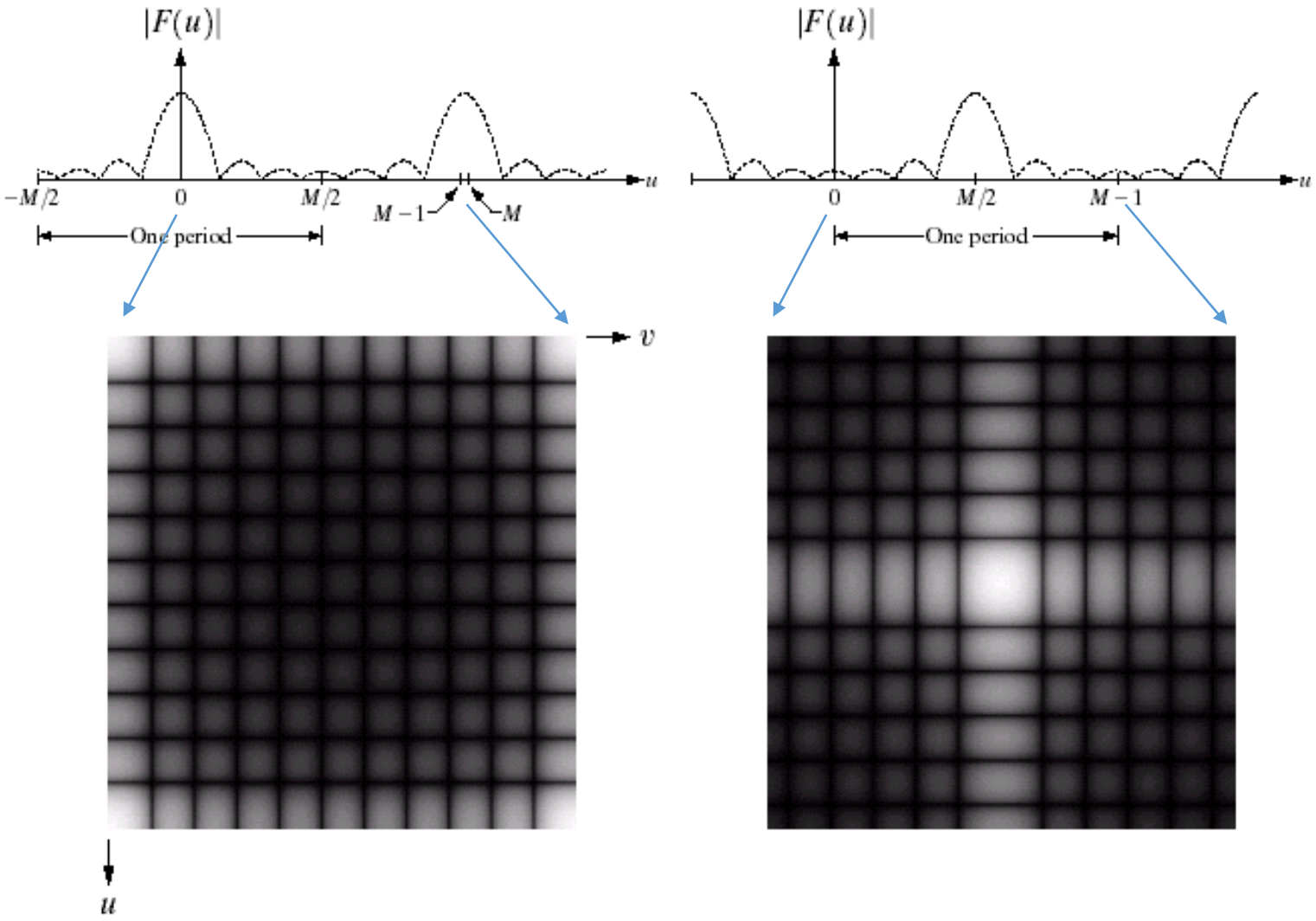
$$\Im\left\{f(x,y)(-1)^{x+y}\right\} = F(u-M/2, v-N/2)$$

(see later, Fourier transform properties)

Fourier transform fundamentals

a b
c d

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



Fourier transform fundamentals

Separability:

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

$$= \frac{1}{M}\sum_{x=0}^{M-1}\exp[-j2\pi ux/M] \quad \frac{1}{N}\sum_{y=0}^{N-1} f(x,y)\exp[-j2\pi vy/N]$$

$$= \frac{1}{M}\sum_{x=0}^{M-1} F(x,v)\exp[-j2\pi ux/M]$$

Let P = sqrt(MN)
Separability permits an $O(P^4) \rightarrow O(P^3)$ computational saving

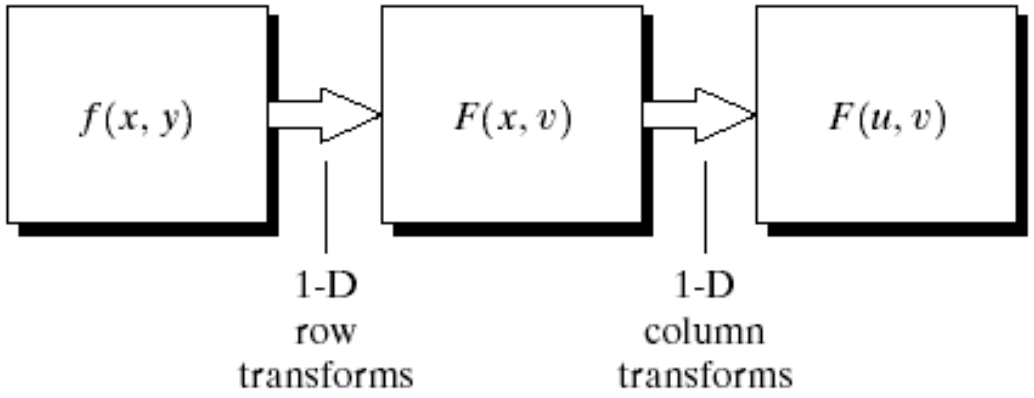


FIGURE 4.35 Computation of the 2-D Fourier transform as a series of 1-D transforms.

Matrix formulation

The **1-D DFT matrix** is defined as:

$$\boldsymbol{A}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix}$$

with $W_N = e^{-j2\pi/N}$

Its inverse is: $\boldsymbol{A}_N^{-1} = \boldsymbol{A}_N^{*T}$

Then, the matrices for the row and column 1-D transforms are

$$\boldsymbol{A}_{row} = \boldsymbol{A}_M \quad \text{and} \quad \boldsymbol{A}_{col} = \boldsymbol{A}_N$$

and the direct and inverse **2-D DFT** can be calculated as

$$\boldsymbol{F} = \boldsymbol{A}_{row}\ \boldsymbol{f}\ \boldsymbol{A}_{col}$$

$$\boldsymbol{f} = \boldsymbol{A}_{row}^{-1}\ \boldsymbol{F}\ \boldsymbol{A}_{col}^{-1}\ /(MN)$$

[Pratt p.202, Gonzalez3 p.96]

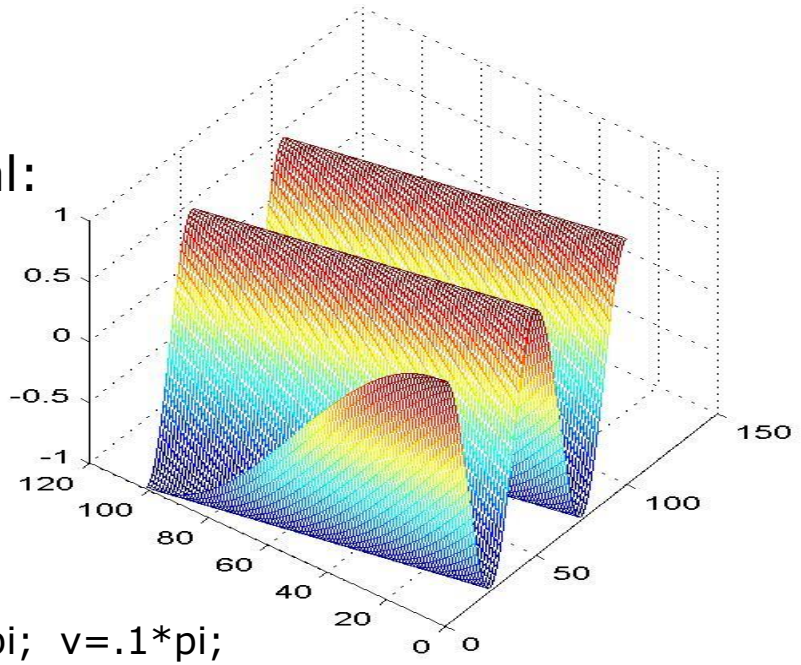
Basic signals

Define the 2-D impulse, unit step, complex exponential:

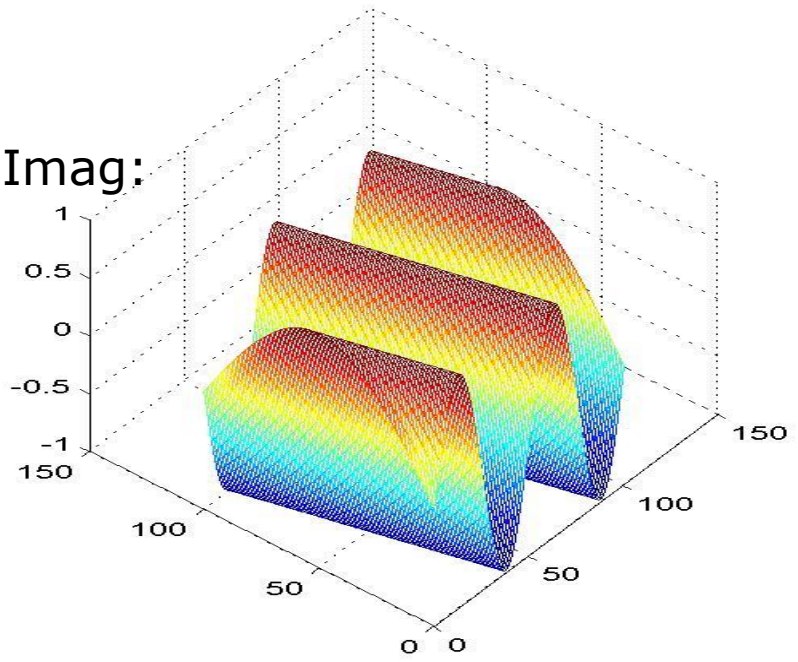
$$u_0(x,y) = \begin{cases} 1 & x = y = 0 \\ 0 & \text{elsewhere} \end{cases} \qquad u_1(x,y) = \begin{cases} 1 & x, y \geq 0 \\ 0 & \text{elsewhere} \end{cases}$$

$$f(x,y) = \exp[j(ux+vy)] = \exp(jux)\exp(jvy) \qquad -\infty \leq x, y \leq \infty$$

Real:



Imag:



u=.4*pi; v=.1*pi;

[x,y] = meshgrid([0:.1:10],[0:.1:10]);

(note step=0.1: signals are oversampled)

Fourier transform fundamentals

Convolution†

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

Convolution theorem†

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

Indeed (1-D case, infinite-length sequences, **DTFT**):

$$F(u) = \sum_{x=-\infty}^{\infty} f(x) \exp[-j u x]; \qquad f(x) * h(x) = \sum_{n=-\infty}^{\infty} f(n) h(x-n);$$

$$\Im[f(x) * h(x)] = \sum_{x=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(n) h(x-n) \exp[-j u x] =$$

$$= \sum_{n=-\infty}^{\infty} f(n) \sum_{x=-\infty}^{\infty} h(x-n) \exp[-j u x] \quad = \sum_{n=-\infty}^{\infty} f(n) \sum_{m=-\infty}^{\infty} h(m) \exp[-j u (m+n)]$$

$$= \sum_{n=-\infty}^{\infty} f(n) \exp[-j u n] \sum_{m=-\infty}^{\infty} h(m) \exp[-j u m] \quad = F(u) H(u)$$

Fourier transform fundamentals

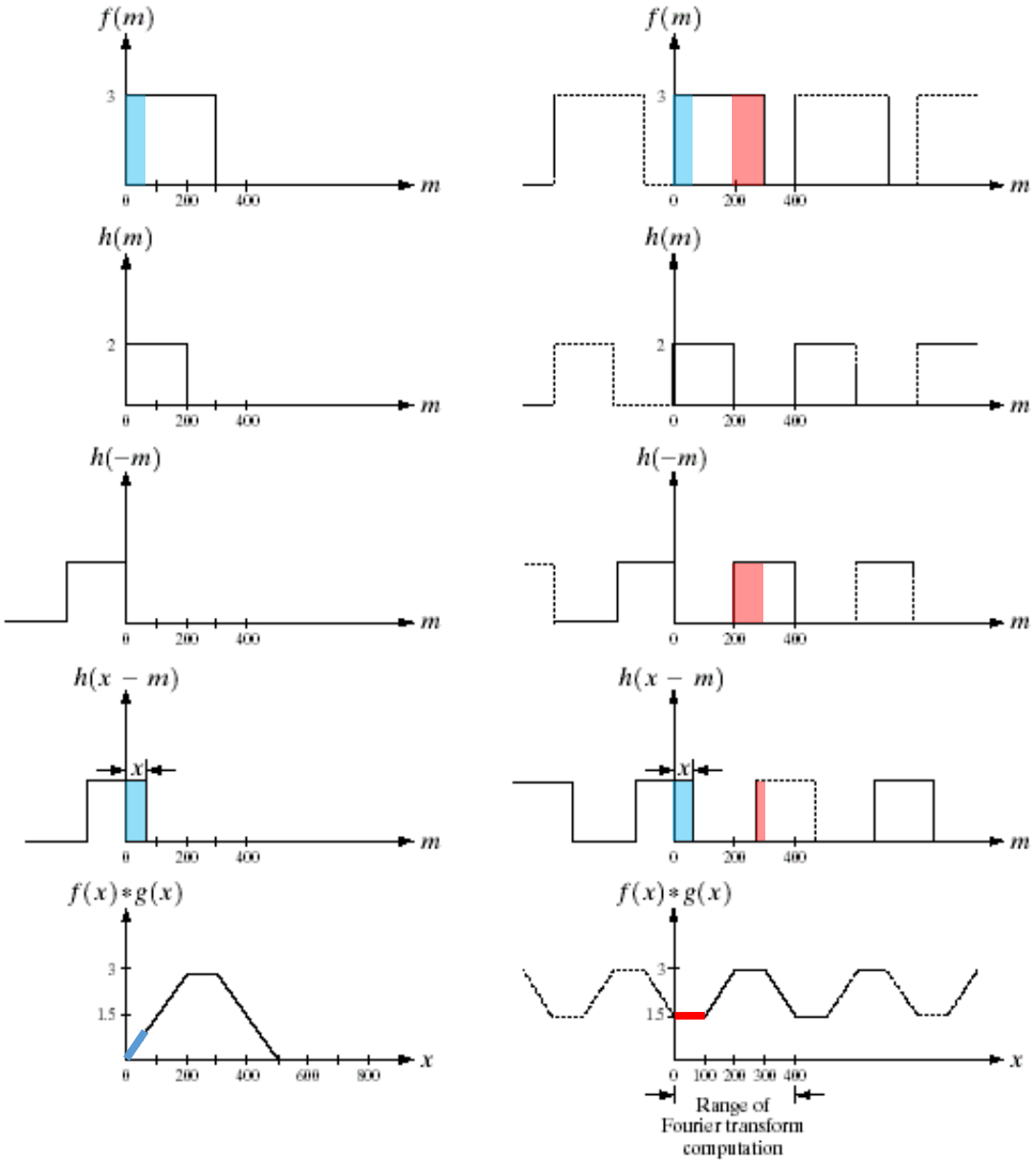
Convolution theorem†	$f(x, y) * h(x, y) \Leftrightarrow F(u, v)H(u, v);$ $f(x, y)h(x, y) \Leftrightarrow F(u, v) * H(u, v)$

Thus, we can perform the convolution as a product in the Fourier domain, and then take the inverse transform.

Note:
for finite-length sequences the product of the **DFT**s is used, which is equivalent to **circular** convolution. Zero padding is required to make it equivalent to **linear** convolution.

Indeed, the Fourier transform assumes that the *f* and *h* functions are **periodic**, with period equal to their length. This should be taken into account, otherwise a wrong result is obtained. The error source is visible even in the data domain:

Fourier transform fundamentals



a f
b g
c h
d i
e j

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

Fourier transform fundamentals

Solution: **zero padding**

$$f_e(x) = \begin{cases} f(x) & 0 \le x \le A-1 \\ 0 & A \le x \le P \end{cases}$$

$$h_e(x) = \begin{cases} h(x) & 0 \le x \le B-1 \\ 0 & B \le x \le P \end{cases}$$

$$P \ge A + B - 1$$

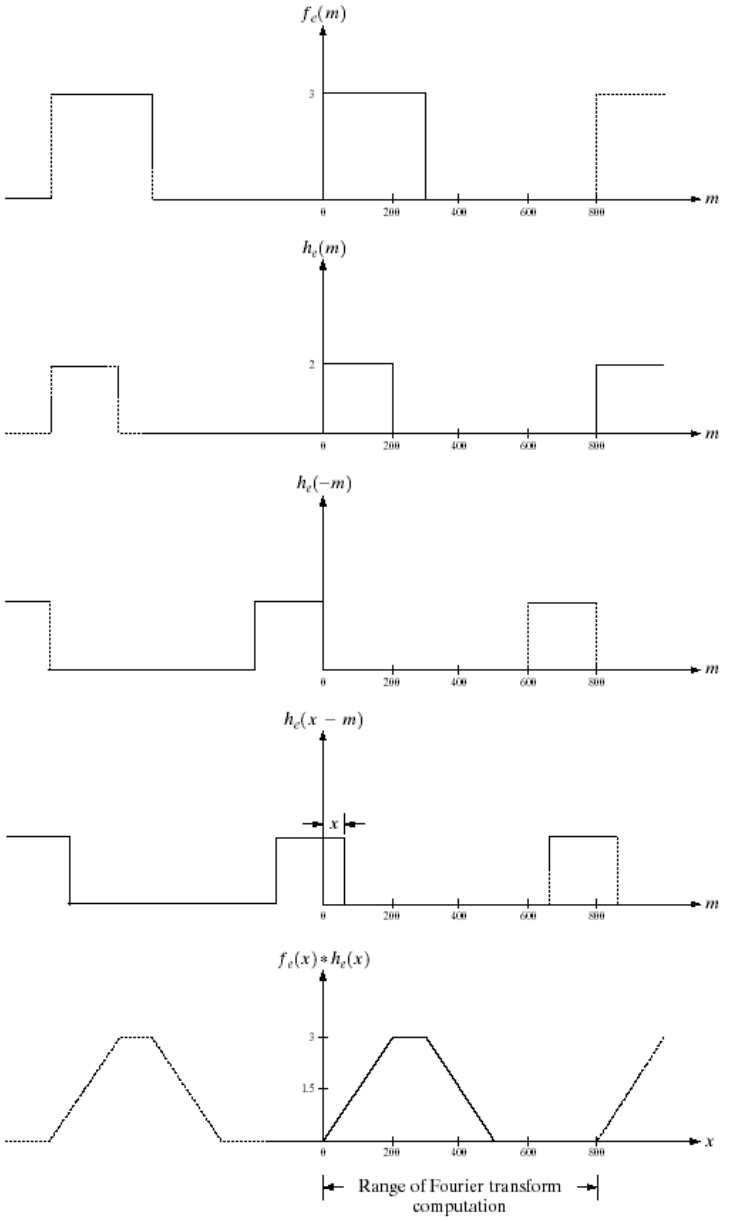
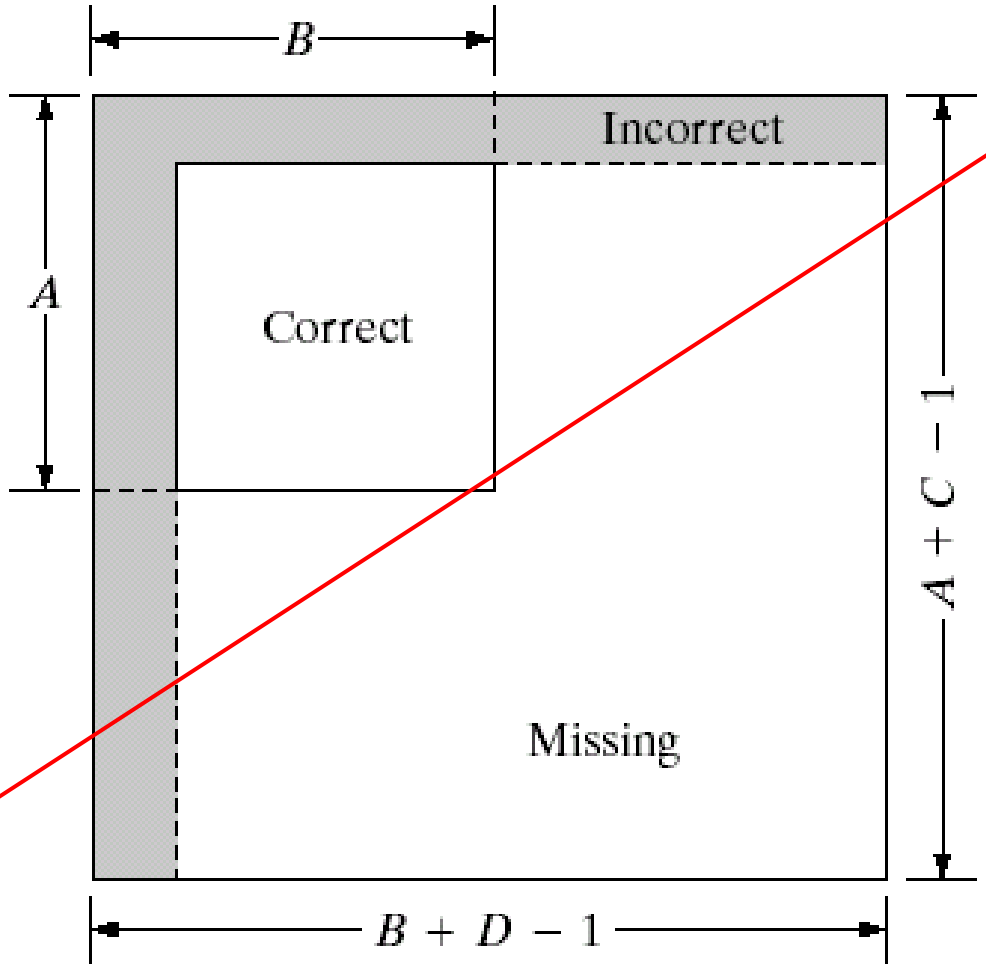


FIGURE 4.37 Result of performing convolution with extended functions.

Fourier transform fundamentals



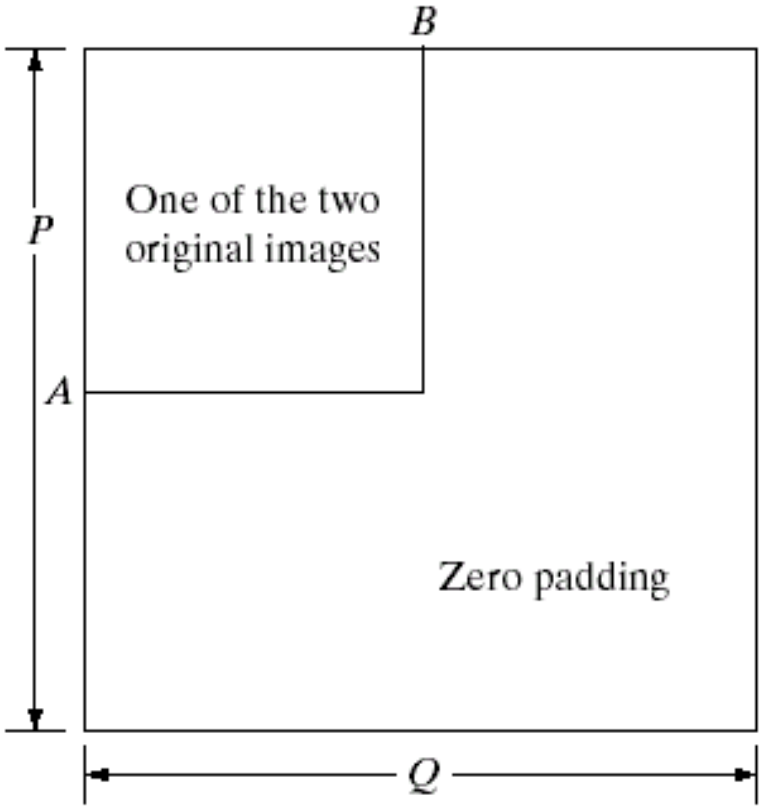
Result of filtering in the frequency domain without properly padding the input images.

a b
c

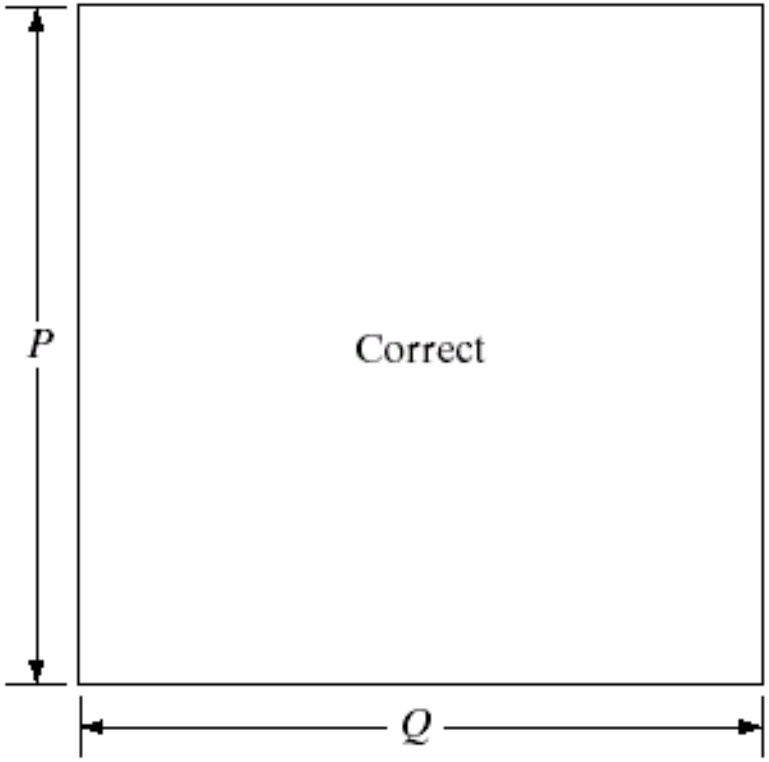
FIGURE 4.38
Illustration of the need for function padding. (a) Result of performing 2-D convolution without padding. (b) Proper function padding. (c) Correct convolution result.

Fourier transform fundamentals

$$P = A + C - 1$$
$$Q = B + D - 1$$

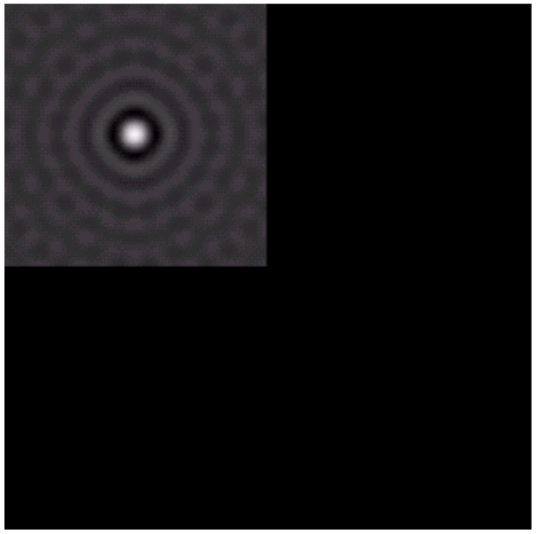


Properly extended (padded) image



Result of filtering in the frequency domain with properly padded input images.

Fourier transform fundamentals



Impulse response of an ideal lowpass filter, truncated to same size as the input image (NxN), zero-padded up to size (2N-1)x(2N-1)

An example

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.

If padding is not performed, periodic-repetition artifacts affect the output image (see Matlab examples)

Correlation in the transform domain

Correlation†

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

Correlation theorem†

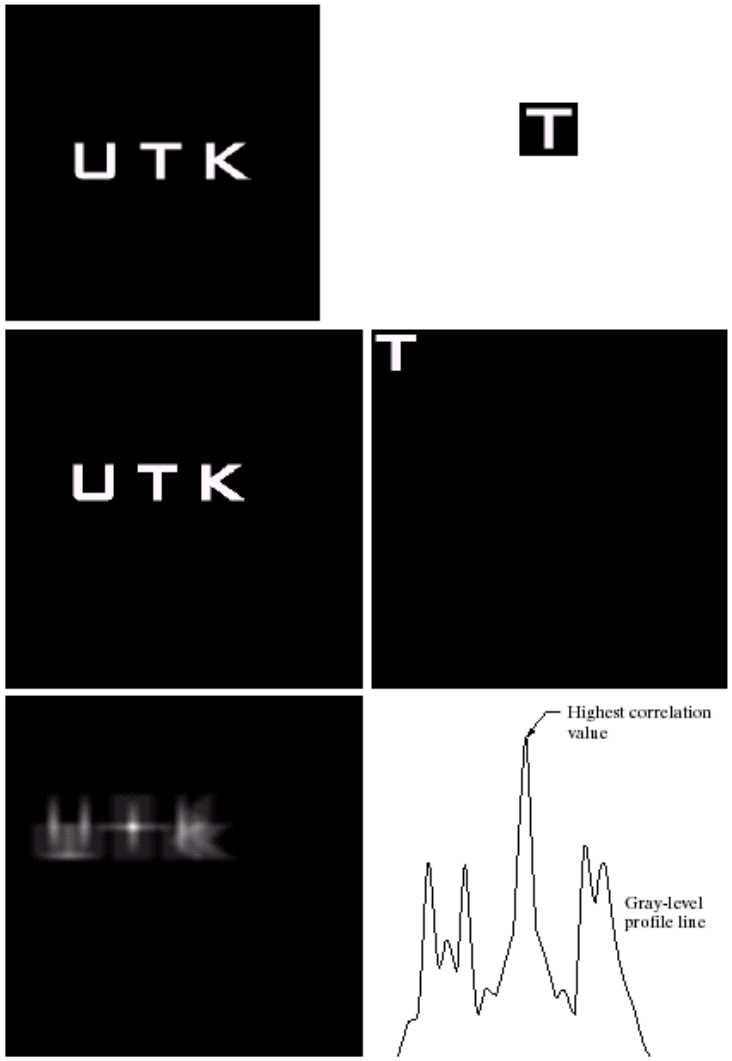
$$f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v) H(u, v);$$
$$f^*(x, y) h(x, y) \Leftrightarrow F(u, v) \circ H(u, v)$$

When dealing with real images, $f=f^*$: correlation coincides with convolution (apart from the + sign in the shifted data: no mirroring of the *h* function).

Correlation can be performed in the Fourier domain, after suitable zero padding.

Correlation in the transform domain

Correlation application: **template matching**



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.

Fourier transform fundamentals

Summary of properties

Property	Expression(s)
Fourier transform	$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$
Inverse Fourier transform	$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$
Polar representation	$F(u,v) = \lvert F(u,v)\rvert e^{-j\phi(u,v)}$
Spectrum	$\lvert F(u,v)\rvert = [R^2(u,v) + I^2(u,v)]^{1/2}$, $R = \text{Real}(F)$ and $I = \text{Imag}(F)$

Fourier transform fundamentals

Phase angle

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

Power spectrum

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

Average value

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

Translation

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

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

When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$, then

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

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

Fourier transform fundamentals

Conjugate symmetry	$F(u, v) = F^*(-u, -v)$ $\lvert F(u, v)\rvert = \lvert F(-u, -v)\rvert$
Differentiation	$\dfrac{\partial^n f(x, y)}{\partial x^n} \Leftrightarrow (ju)^n F(u, v)$ $(-jx)^n f(x, y) \Leftrightarrow \dfrac{\partial^n F(u, v)}{\partial u^n}$
Laplacian	$\nabla^2 f(x, y) \Leftrightarrow -(u^2 + v^2)F(u, v)$
Distributivity	$\Im[f_1(x, y) + f_2(x, y)] = \Im[f_1(x, y)] + \Im[f_2(x, y)]$ $\Im[f_1(x, y) \cdot f_2(x, y)] \neq \Im[f_1(x, y)] \cdot \Im[f_2(x, y)]$

Fourier transform fundamentals

Scaling	$af(x, y) \Leftrightarrow aF(u, v), f(ax, by) \Leftrightarrow \frac{1}{\lvert ab\rvert} F(u/a, v/b)$
Rotation	$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$ $f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$
Periodicity	$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ $f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$
Separability	See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.

Fourier transform fundamentals

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

Fourier transform fundamentals

Some useful FT pairs:

Impulse	$\delta(x, y) \Leftrightarrow 1$
Gaussian	$A\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2(x^2+y^2)} \Leftrightarrow Ae^{-(u^2+v^2)/2\sigma^2}$
Rectangle	$\text{rect}[a, b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow \dfrac{1}{2}[\delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0)]$
Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow j\dfrac{1}{2}[\delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0)]$

† Assumes that functions have been extended by zero padding.

2-D Discrete **Cosine** Transform

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

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

$$\alpha(u) = \begin{cases} \sqrt{1/M} & \text{if } u = 0 \\ \sqrt{2/M} & \text{if } u > 0 \end{cases} \qquad \alpha(v) = \begin{cases} \sqrt{1/N} & \text{if } v = 0 \\ \sqrt{2/N} & \text{if } v > 0 \end{cases}$$

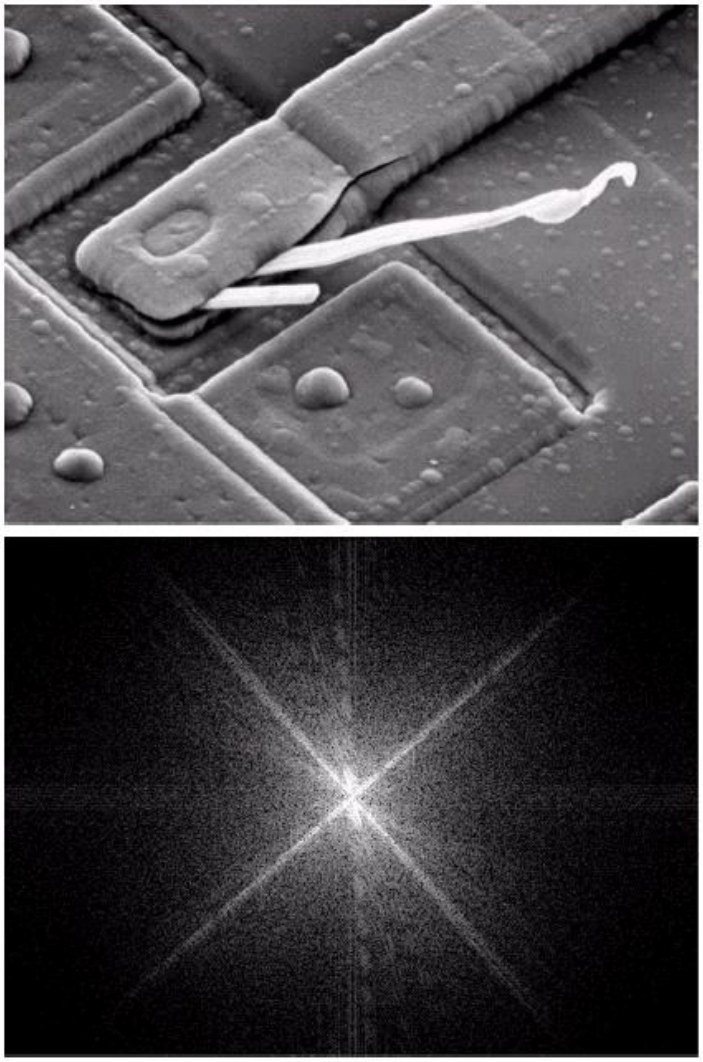
Note:

fast 2-D DCT computation is feasible, e.g. based on separability and FFT

With respect to the DFT:

- The transform is real. No symmetries.
- Reduced artifacts from periodic repetition.
- Similar energy compaction performance (ok for highly correlated data).
- Easily quantized to perform compression.
  (See Matlab for coeff. ablation example)

Enhancement in the Frequency Domain



a
b

FIGURE 4.4
(a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a).

Enhancement in the Frequency Domain

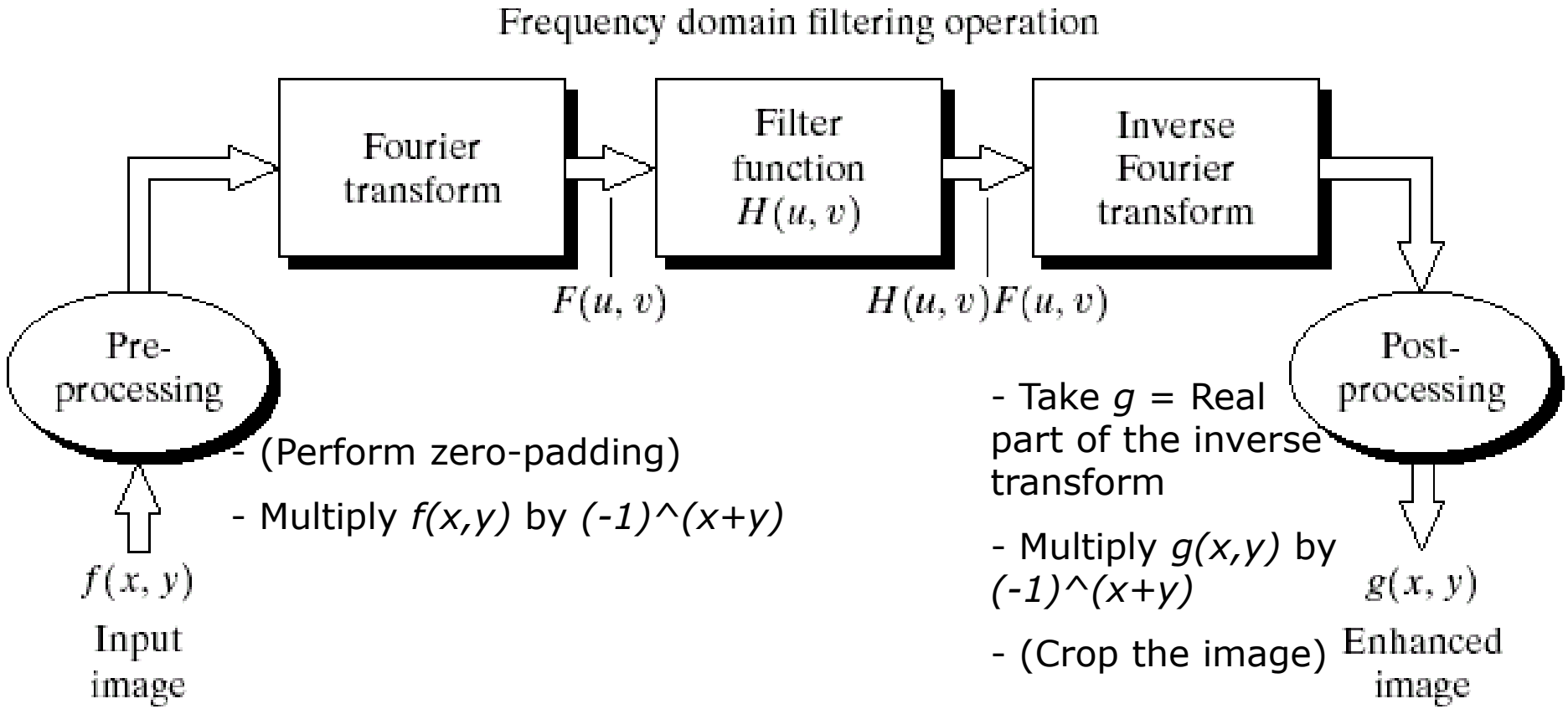
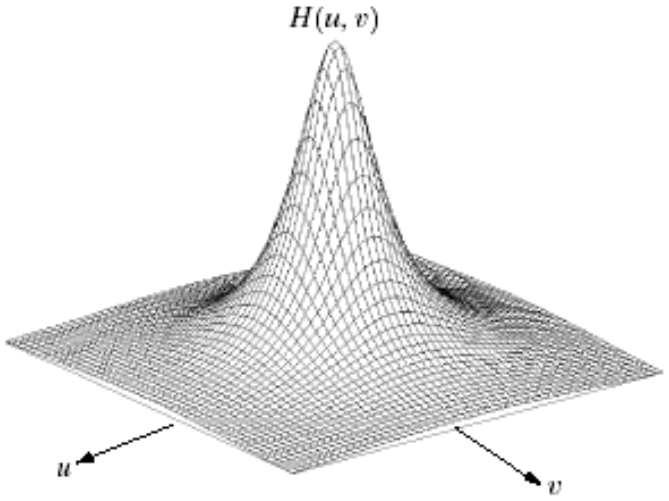


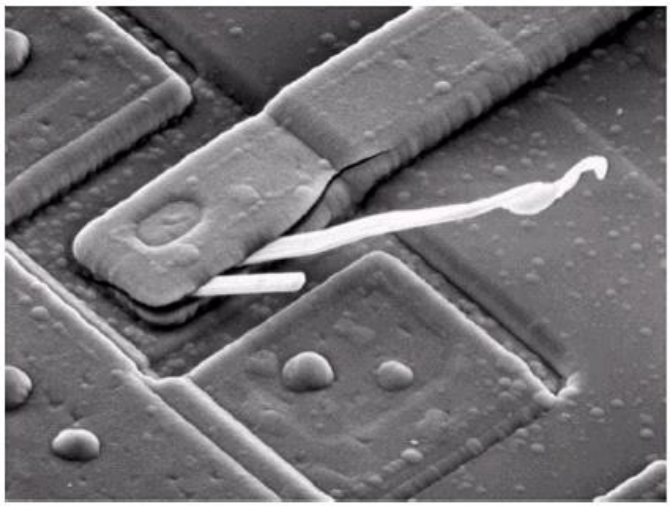
FIGURE 4.5 Basic steps for filtering in the frequency domain.

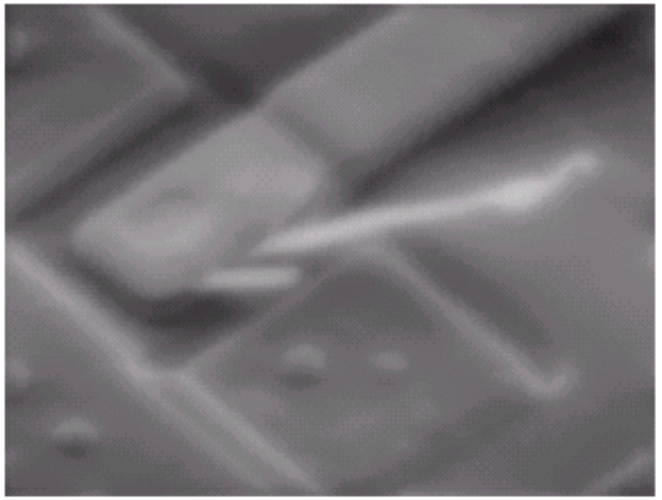
Enhancement in the Frequency Domain

Note: Real freq. responses are shown in the next slides, but in general $H(u,v)$ is complex:

→ $|H(u,v)|$

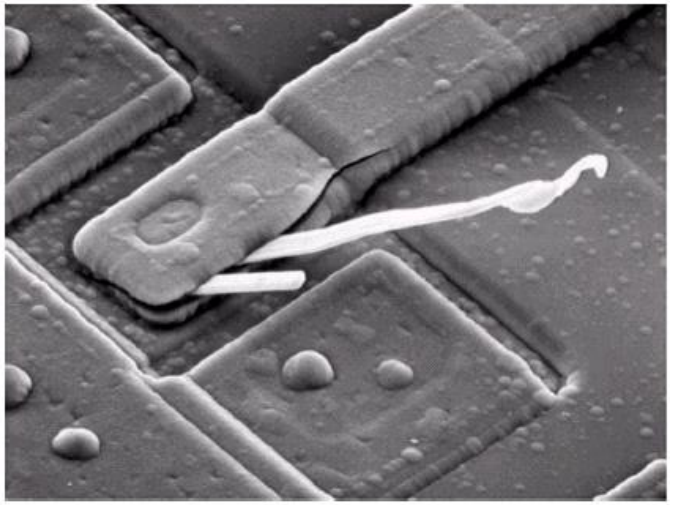


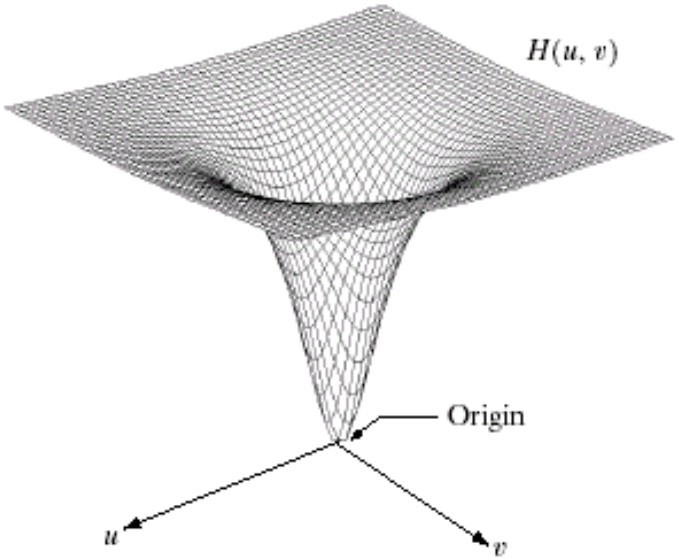


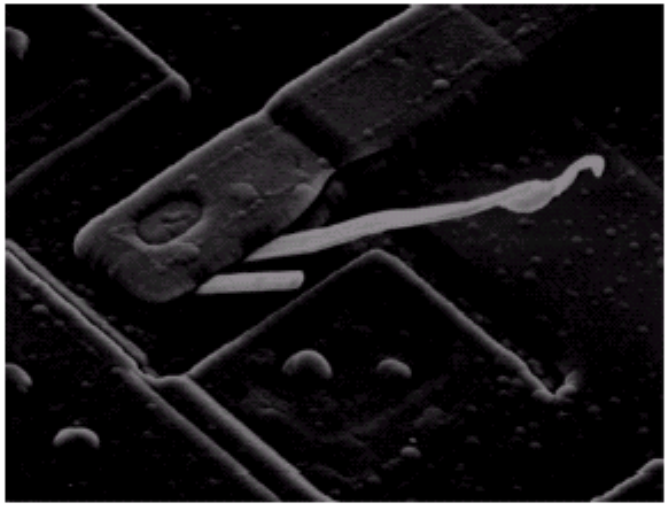


(a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a).

Enhancement in the Frequency Domain







Output image not shifted

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

Enhancement in the Frequency Domain

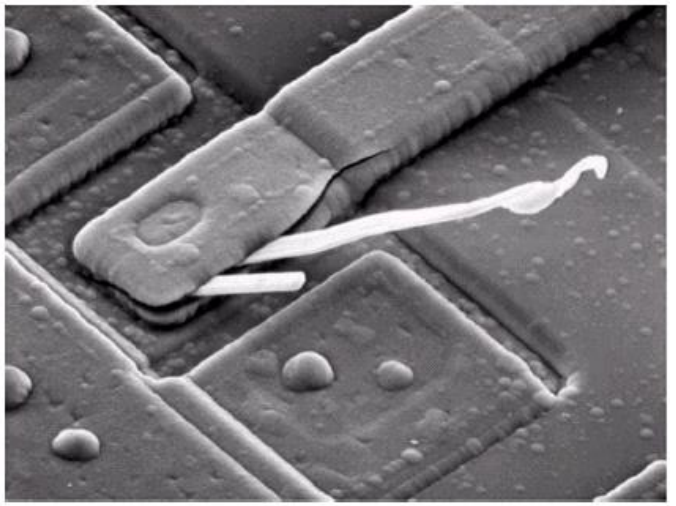
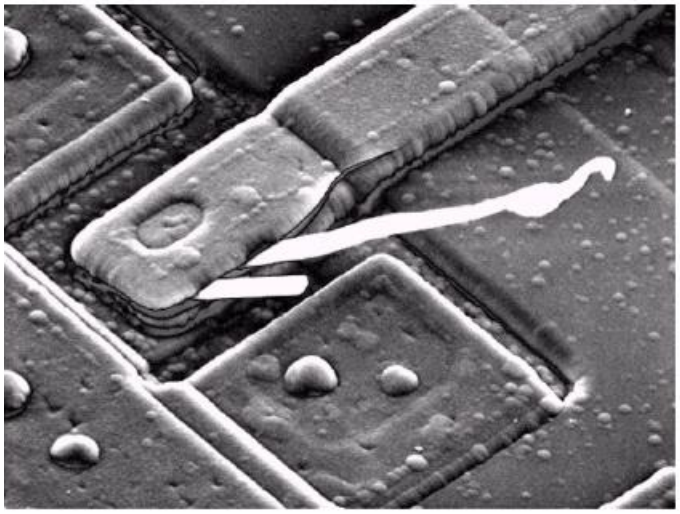
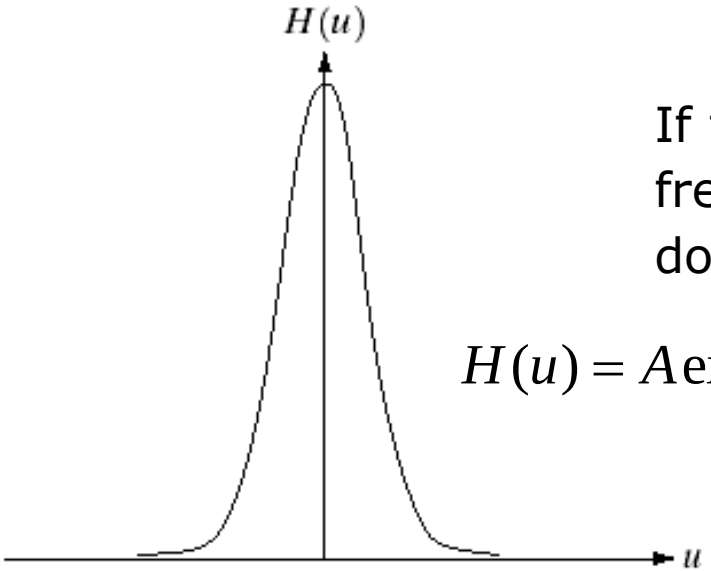


FIGURE 4.8
Result of highpass filtering the image in Fig. 4.4(a) with the filter in Fig. 4.7(c), modified by adding a constant of one-half the filter height to the filter function.



i.e.: unsharp masking

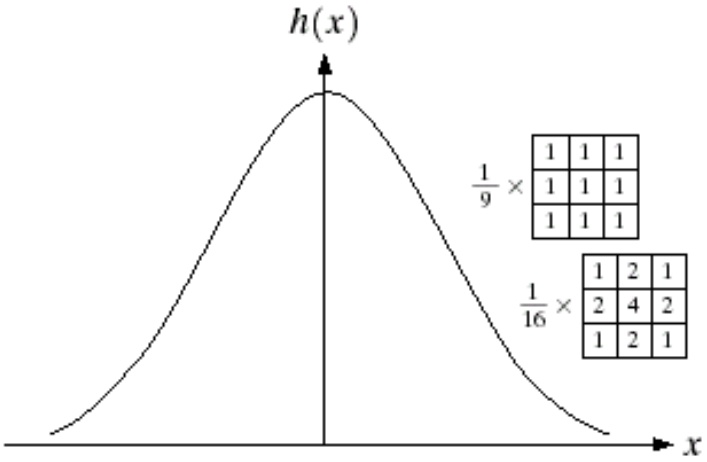
Enhancement in the Frequency Domain
Gaussian lowpass filter



If the filter has a (real) Gaussian shape in the frequency domain, its coefficients in the data domain are real and have a Gaussian shape too:

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

When σ is large (wide passband in the frequency domain), the coefficients profile is narrow, and viceversa.



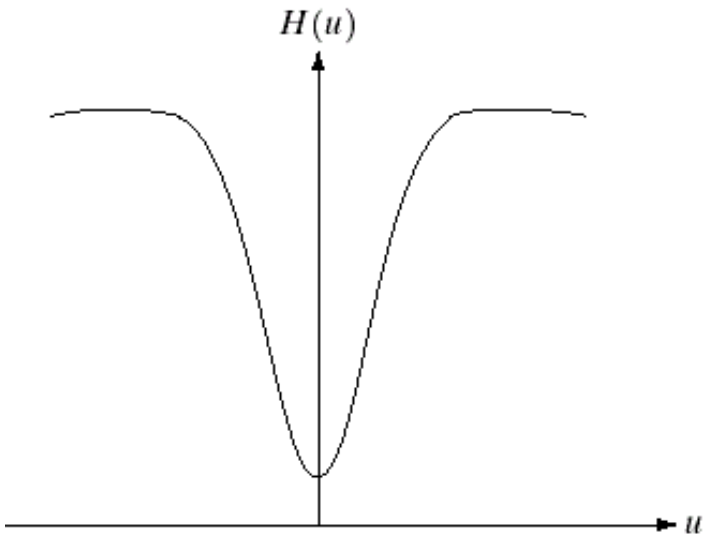
Typical 3x3 lowpass filter masks in the data domain can be considered very rough approximations of a Gaussian shape.

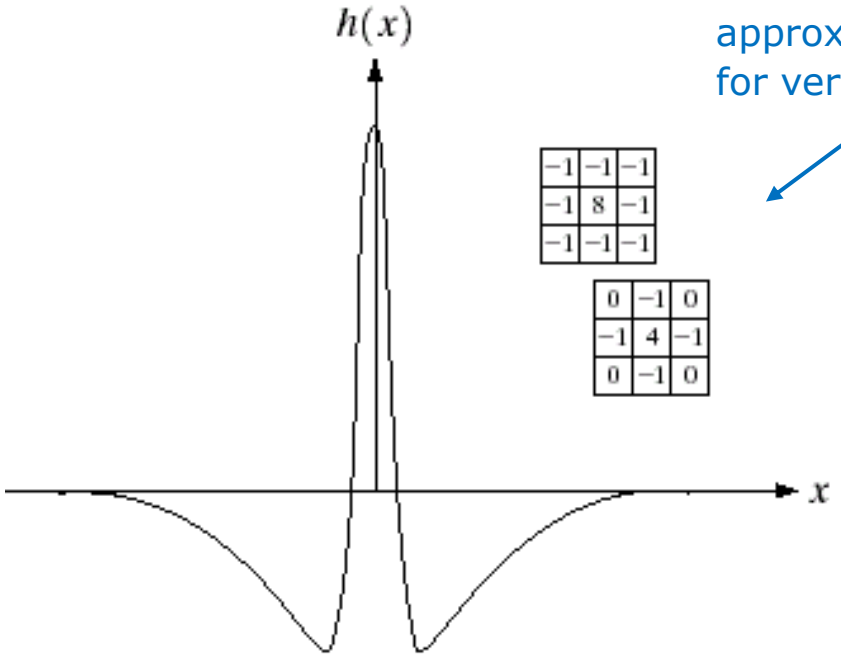
Enhancement in the Frequency Domain
Difference-of-Gaussians (**DoG**) filter

$$H(u) = A_1 \exp(-u^2 / 2\sigma_1^{\,2}) - A_2 \exp(-u^2 / 2\sigma_2^{\,2})$$

$$h(x) = \sqrt{2\pi}\left(\sigma_1 A_1 \exp(-2\pi^2\sigma_1^{\,2}x^2) - \sigma_2 A_2 \exp(-2\pi^2\sigma_2^{\,2}x^2) \right)$$

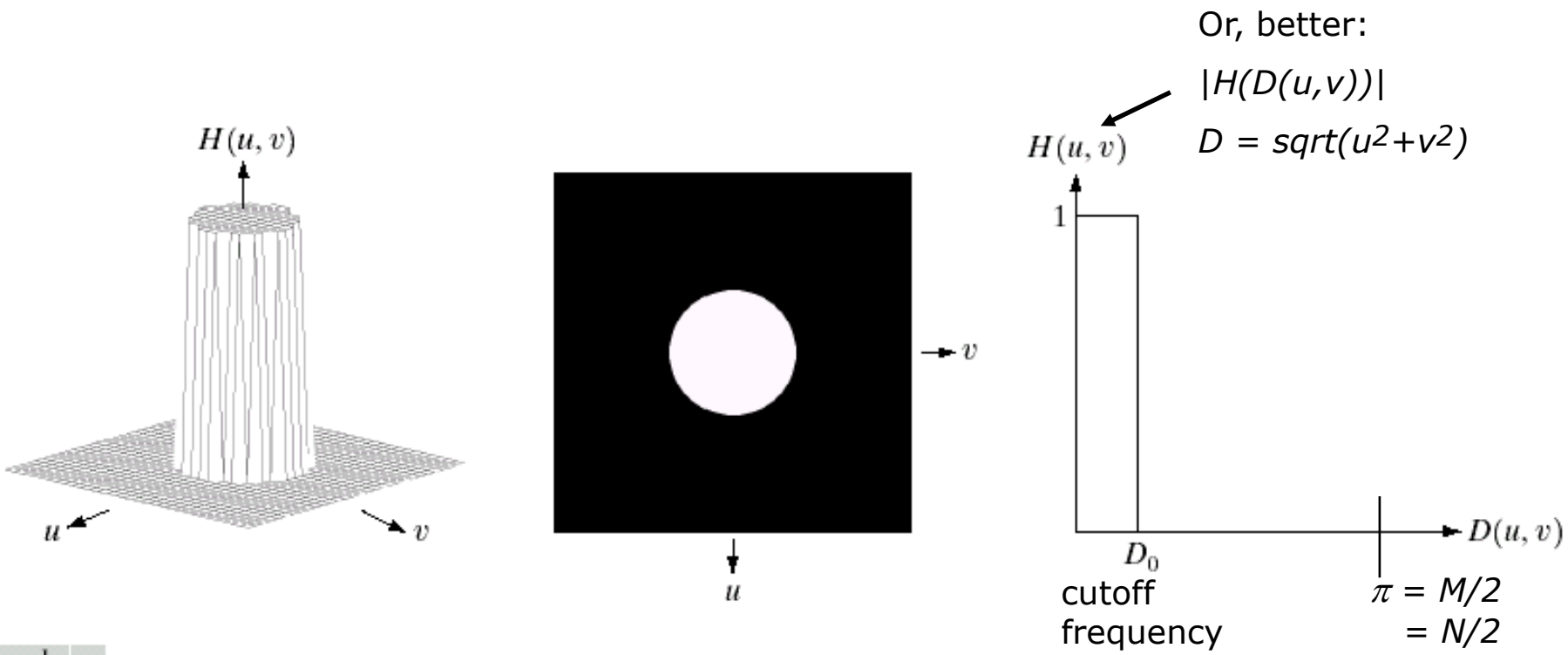
$A_1 \geq A_2$; $\sigma_1 > \sigma_2$ (A_1=A_2 to obtain a purely highpass response)





approximations for very small σ

Enhancement in the Frequency Domain
Ideal lowpass filter



a b c

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

It is a *sinc* function in the coefficients domain → **ringing**

Enhancement in the Frequency Domain
Ideal lowpass filter

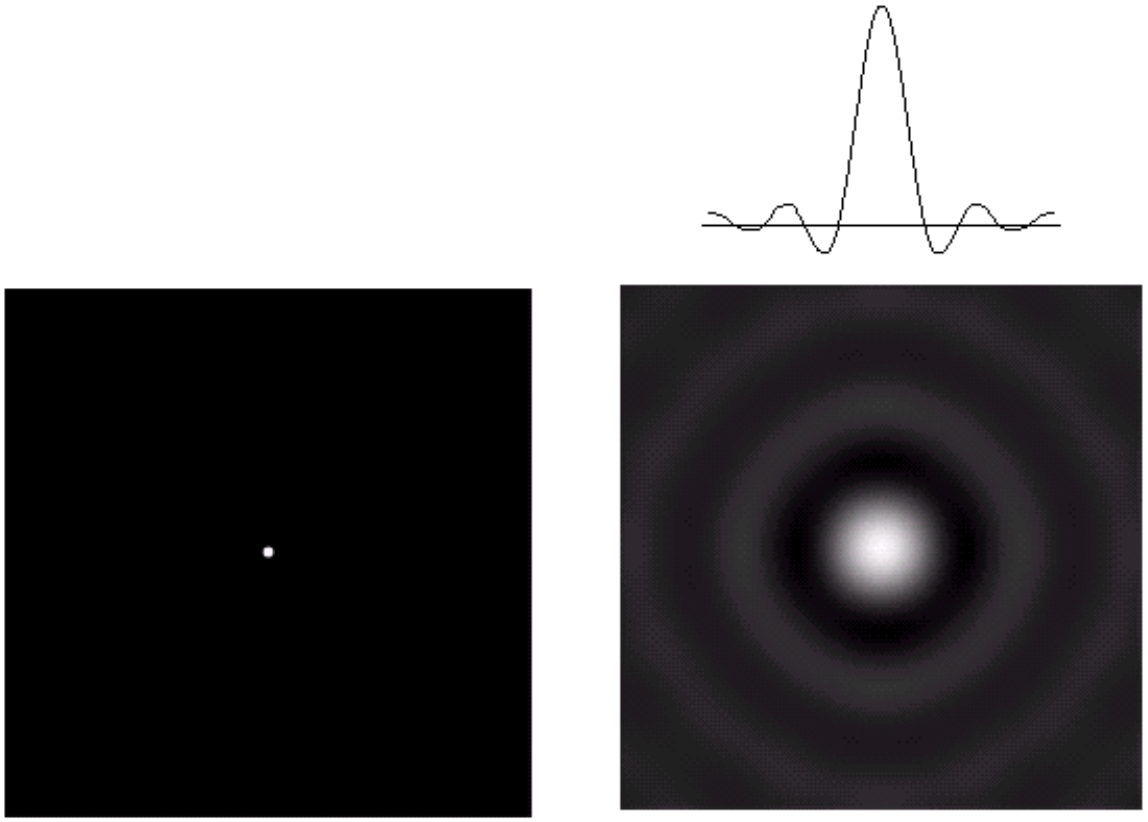


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing).

Enhancement in the Frequency Domain
Ideal lowpass filter

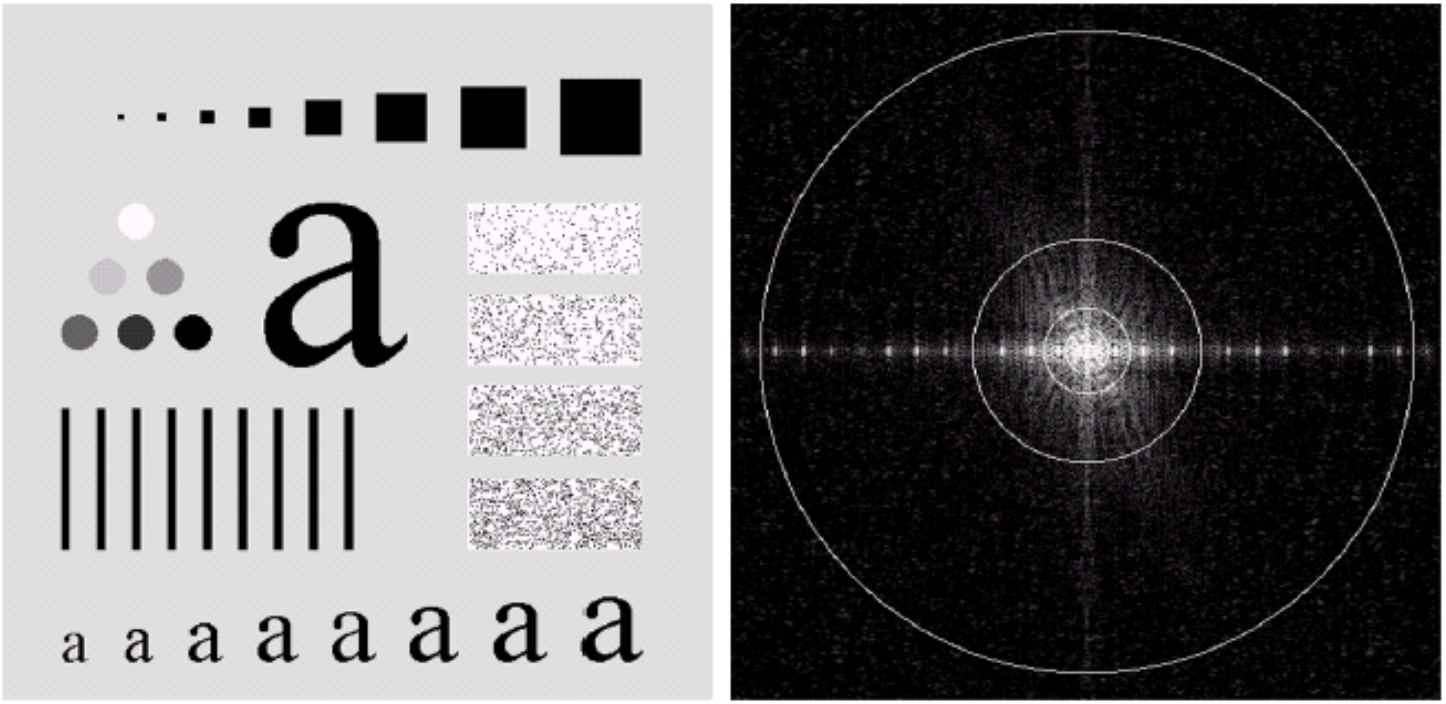
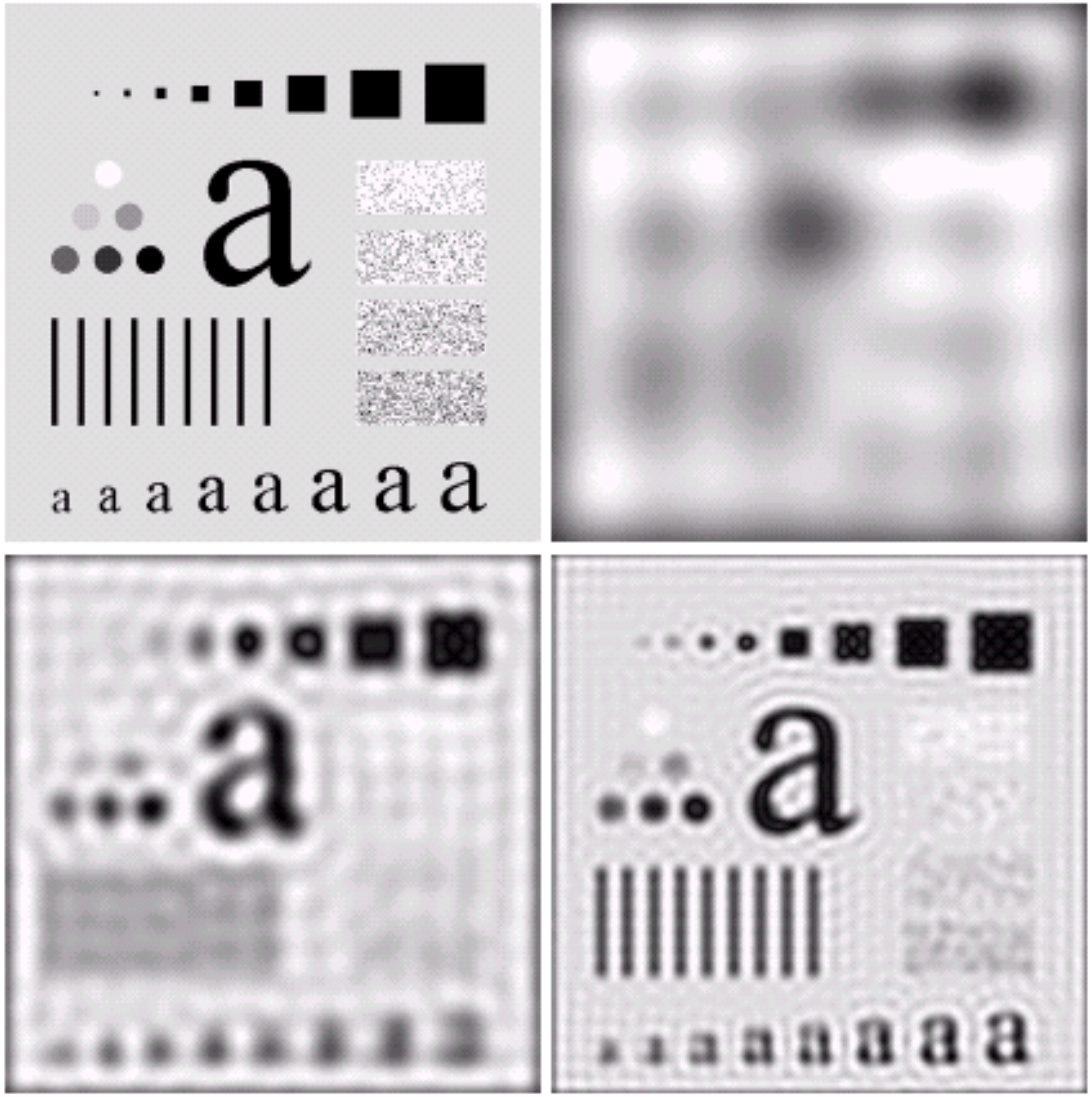


FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

Enhancement in the Frequency Domain
Ideal lowpass filter



Original image. (b)–(f) Re
t radii values of 5, 15, 30,

Enhancement in the Frequency Domain
Ideal lowpass filter

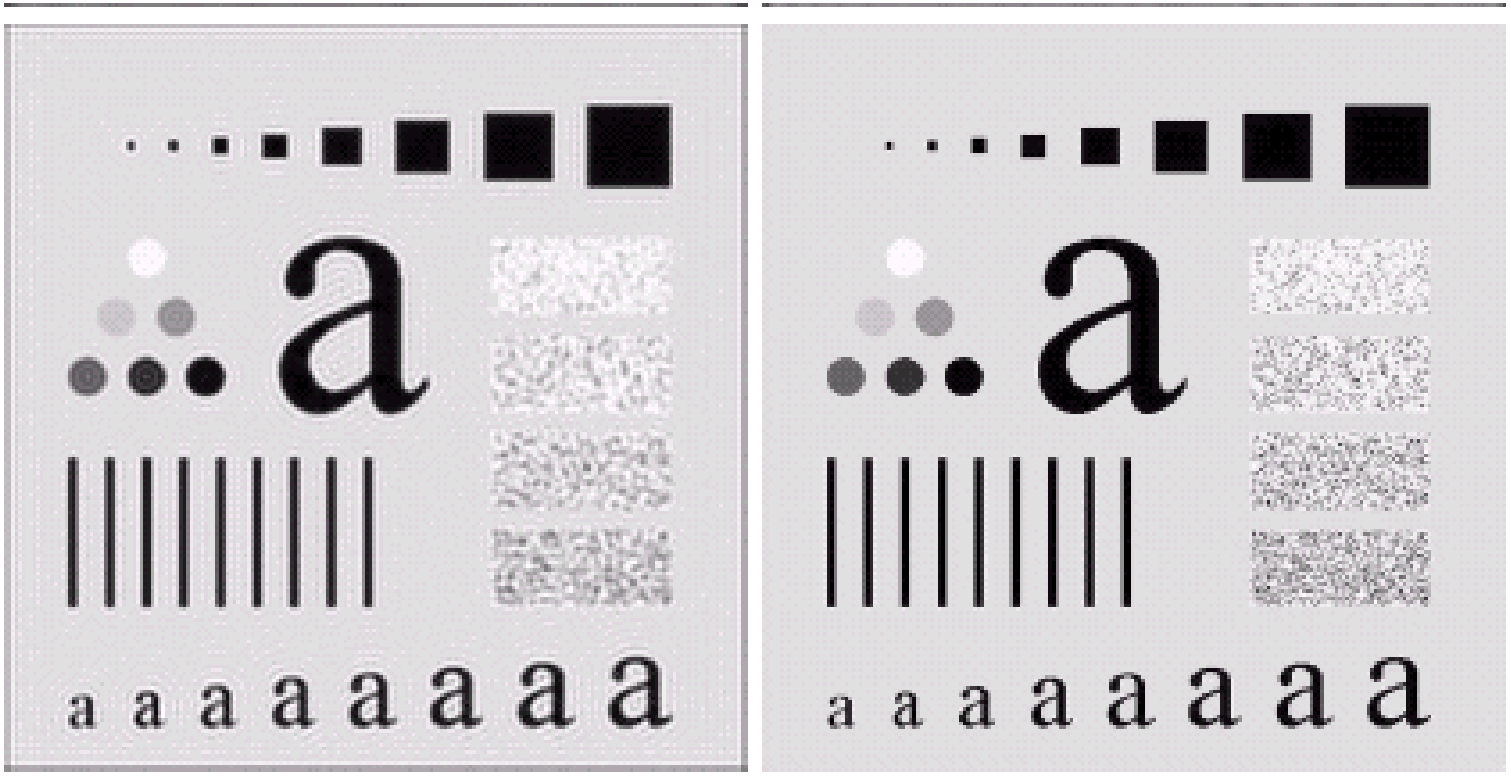


FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

Enhancement in the Frequency Domain
Butterworth lowpass filter

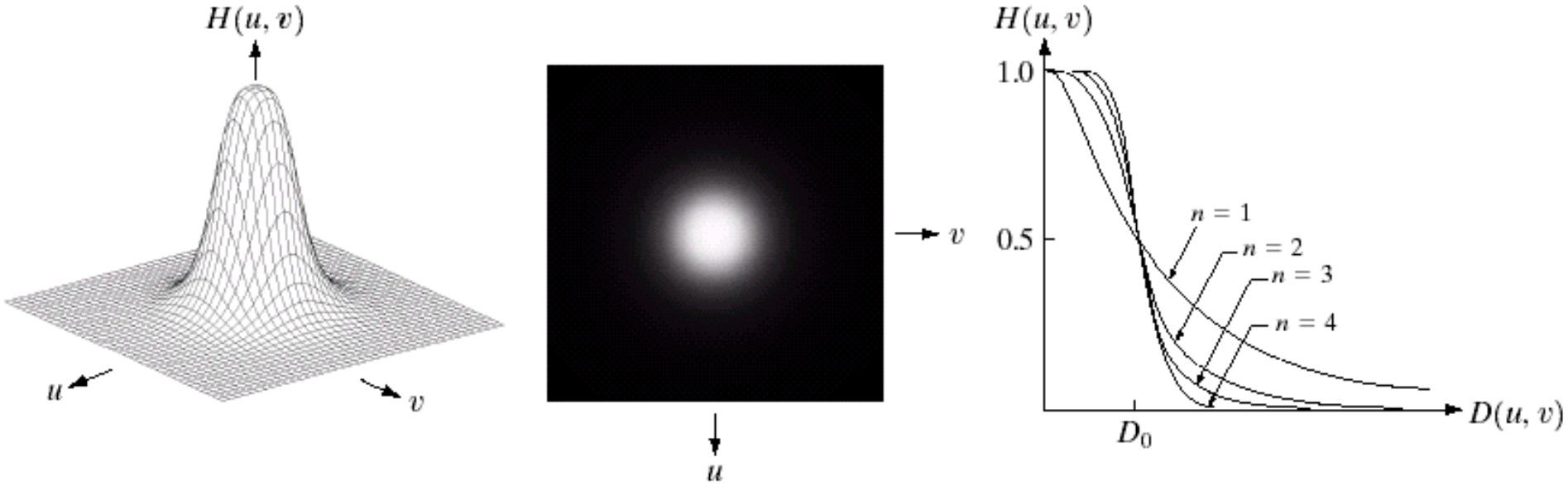


FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

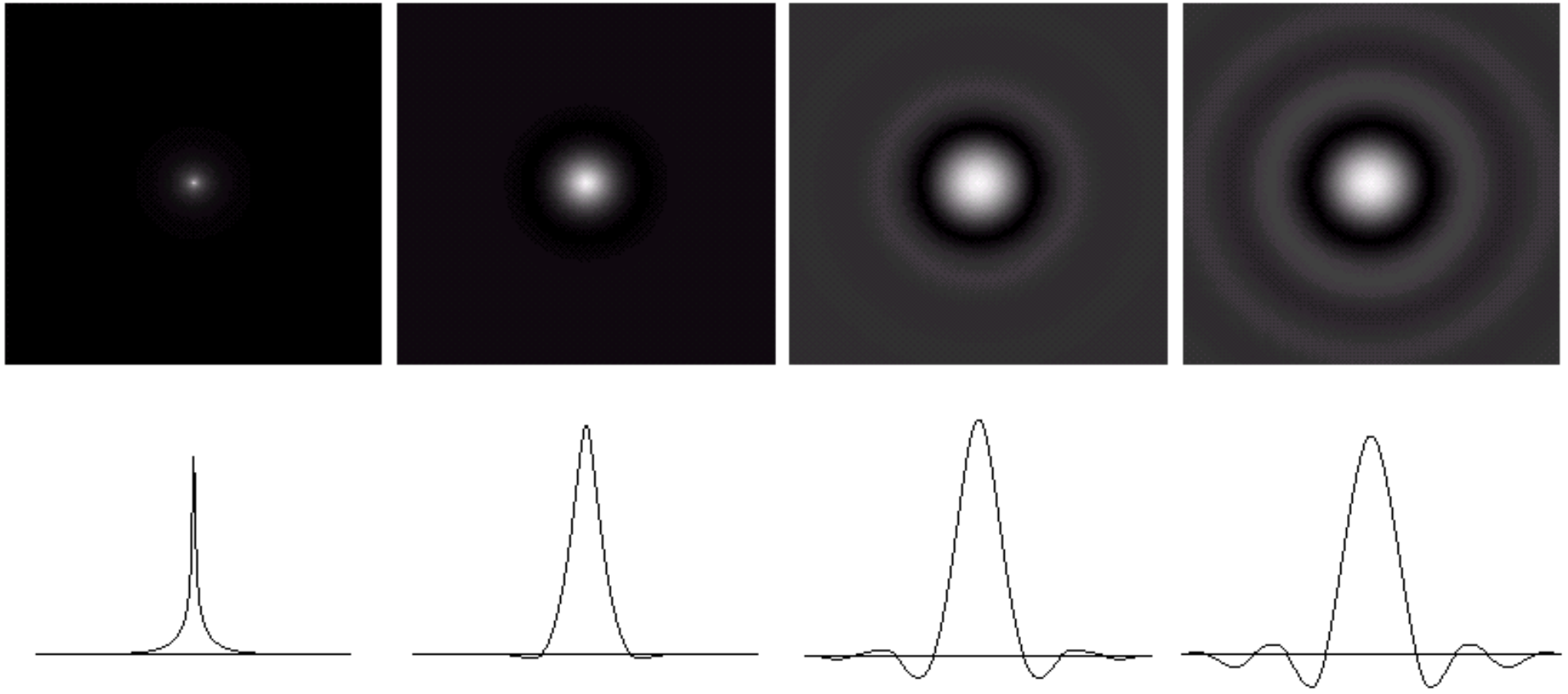
$$H(u,v) = \frac{1}{1+(D(u,v)/D_0)^{2n}}$$

n: filter order

D(u,v): Euclidean distance from center of transform

D_0: cutoff frequency ($|H(D_0)|=0.5$)

Enhancement in the Frequency Domain
Butterworth lowpass filter



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.

Enhancement in the Frequency Domain
Butterworth lowpass filter

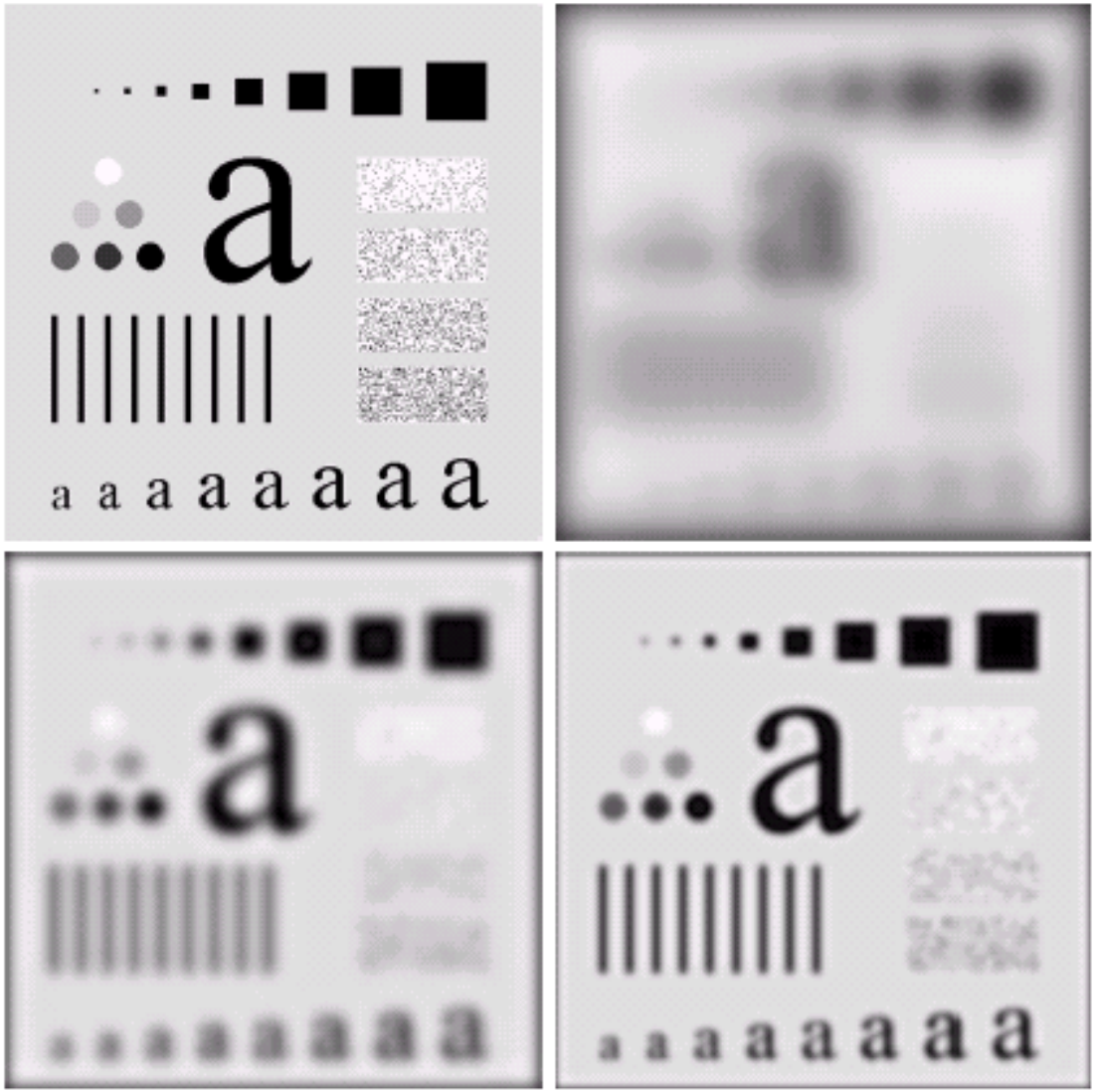


Fig.4.15 (a) original; (b-d) Butterworth filter of order 2, cutoff freqs. 5, 15, 30.

Enhancement in the Frequency Domain
Gaussian lowpass filter

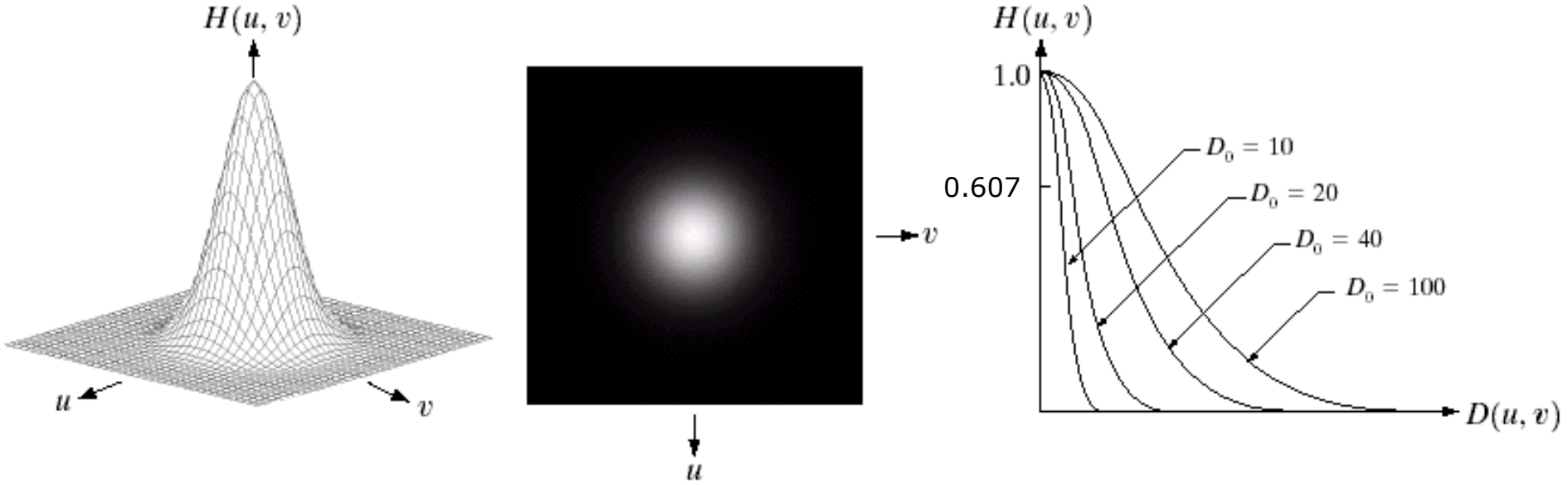


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

$$H(u,v) = \exp\left[-D^2(u,v)/2D_0^{\,2}\right]$$

D(u,v): Euclidean distance from center of transform

D_0: cutoff frequency (σ in slide 33)
$H(D_0)=0.607$

Enhancement in the Frequency Domain
Gaussian lowpass filter

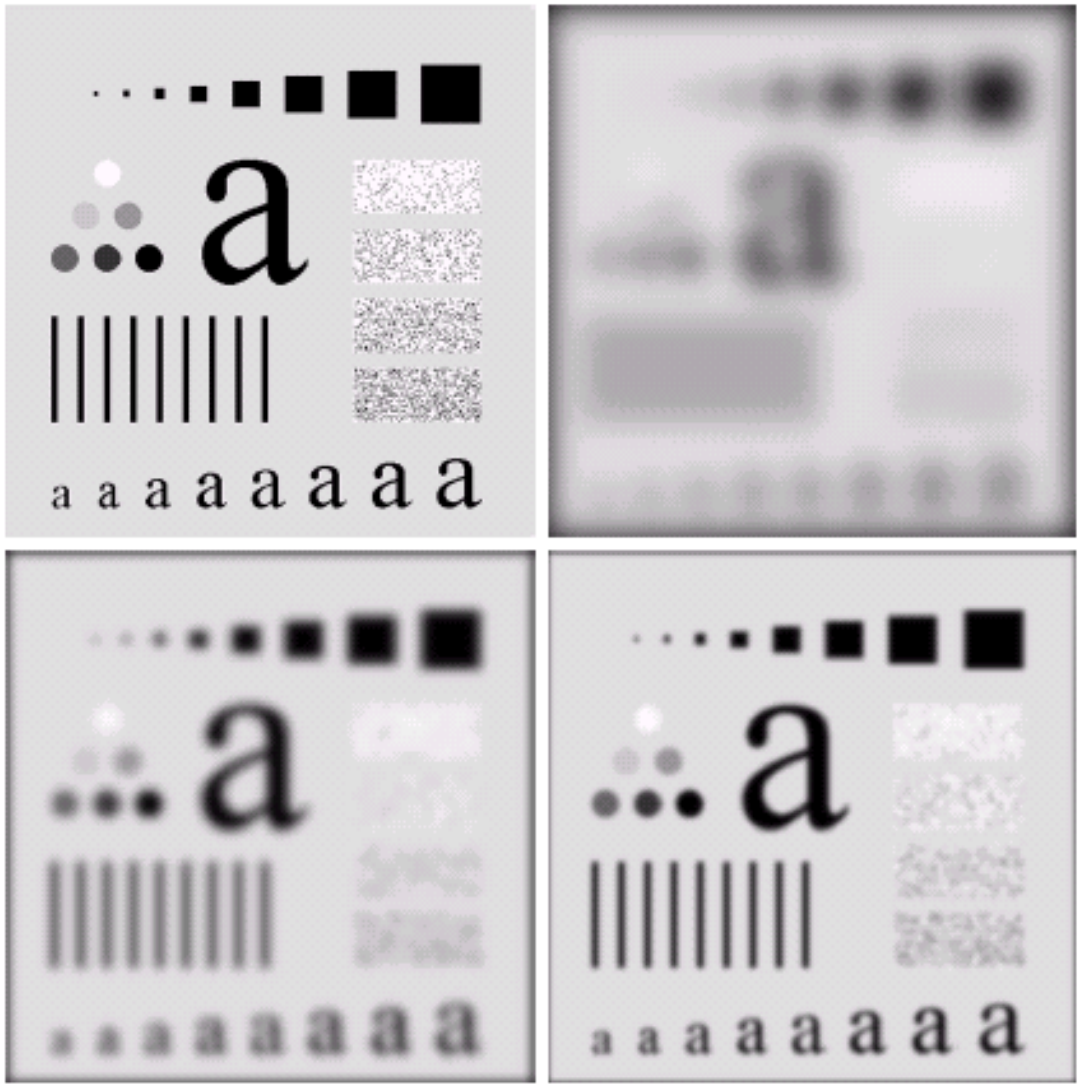
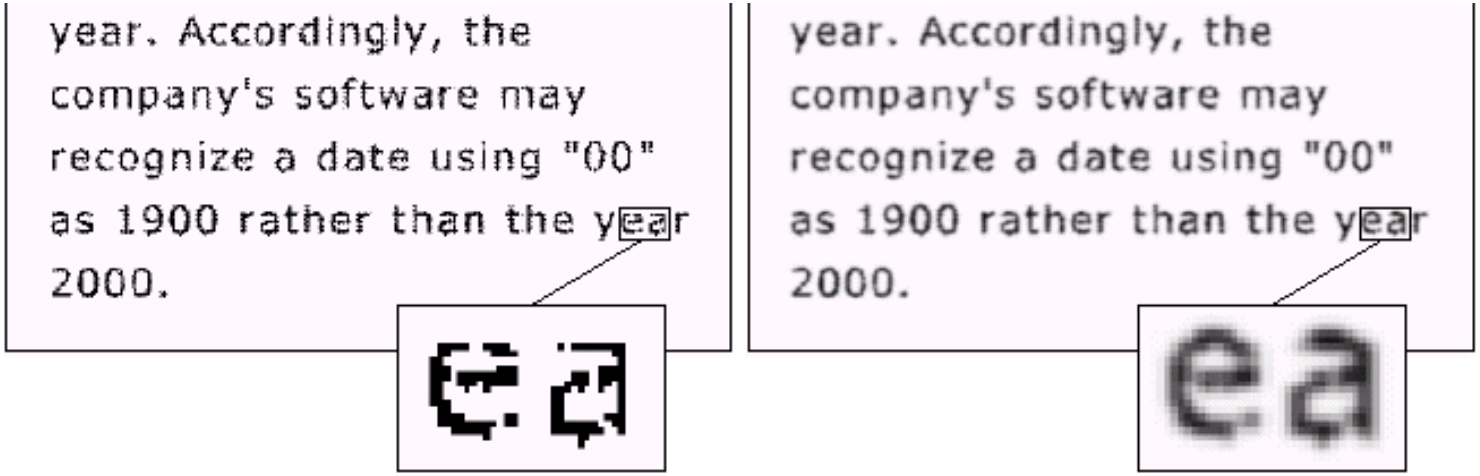


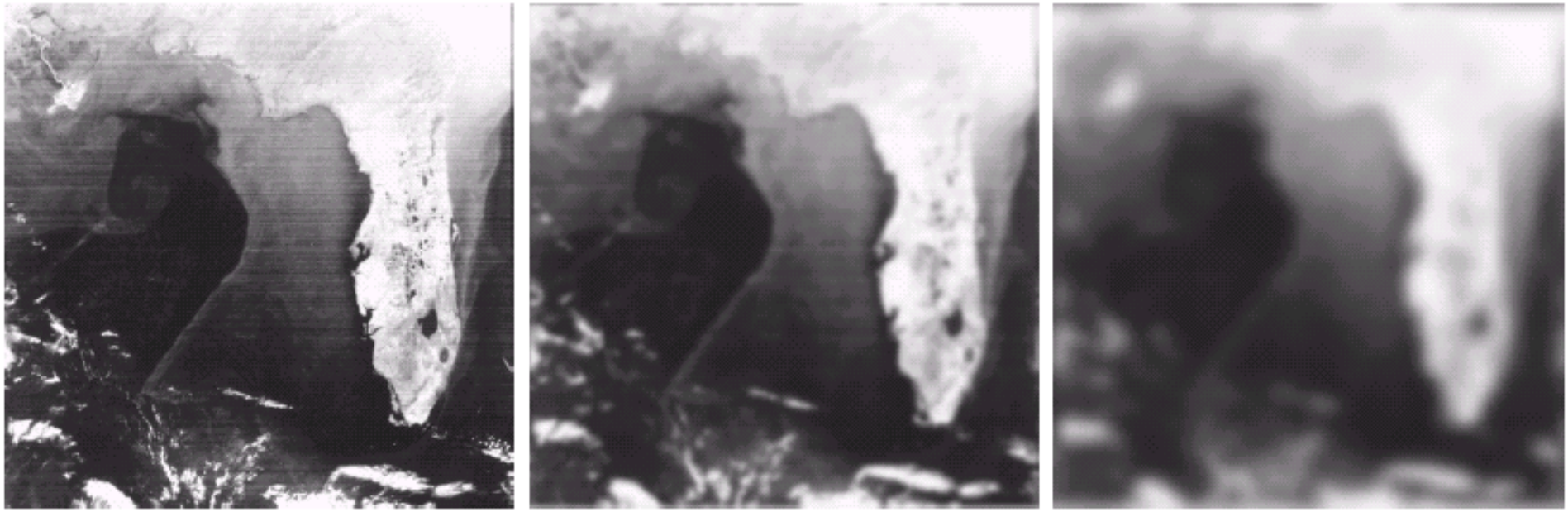
Fig.4.18 (a) original; (b-d) Gaussian filter with cutoff freqs. 5, 15, 30.

Enhancement in the Frequency Domain
Gaussian lowpass filter

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).



Enhancement in the Frequency Domain
Gaussian lowpass filter



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.)

National Oceanic and Athmospheric Administration

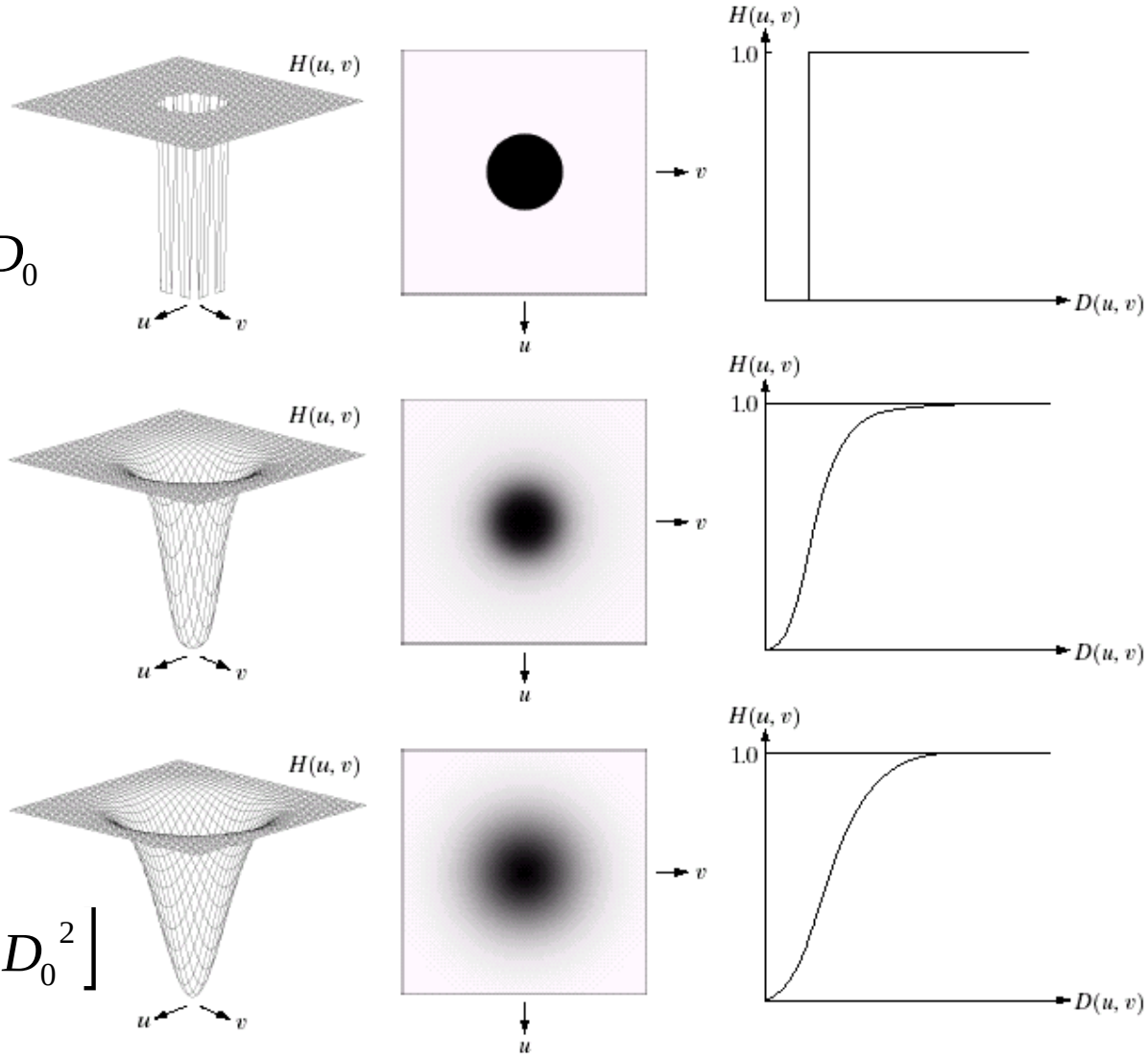
Enhancement in the Frequency Domain
Highpass filters

In general, **Hhp = 1 - Hlp**
(for unitary-gain Hlp)

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

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

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



Enhancement in the Frequency Domain
Highpass filters

Note: images here and in the following slides have not been shifted by 128 levels

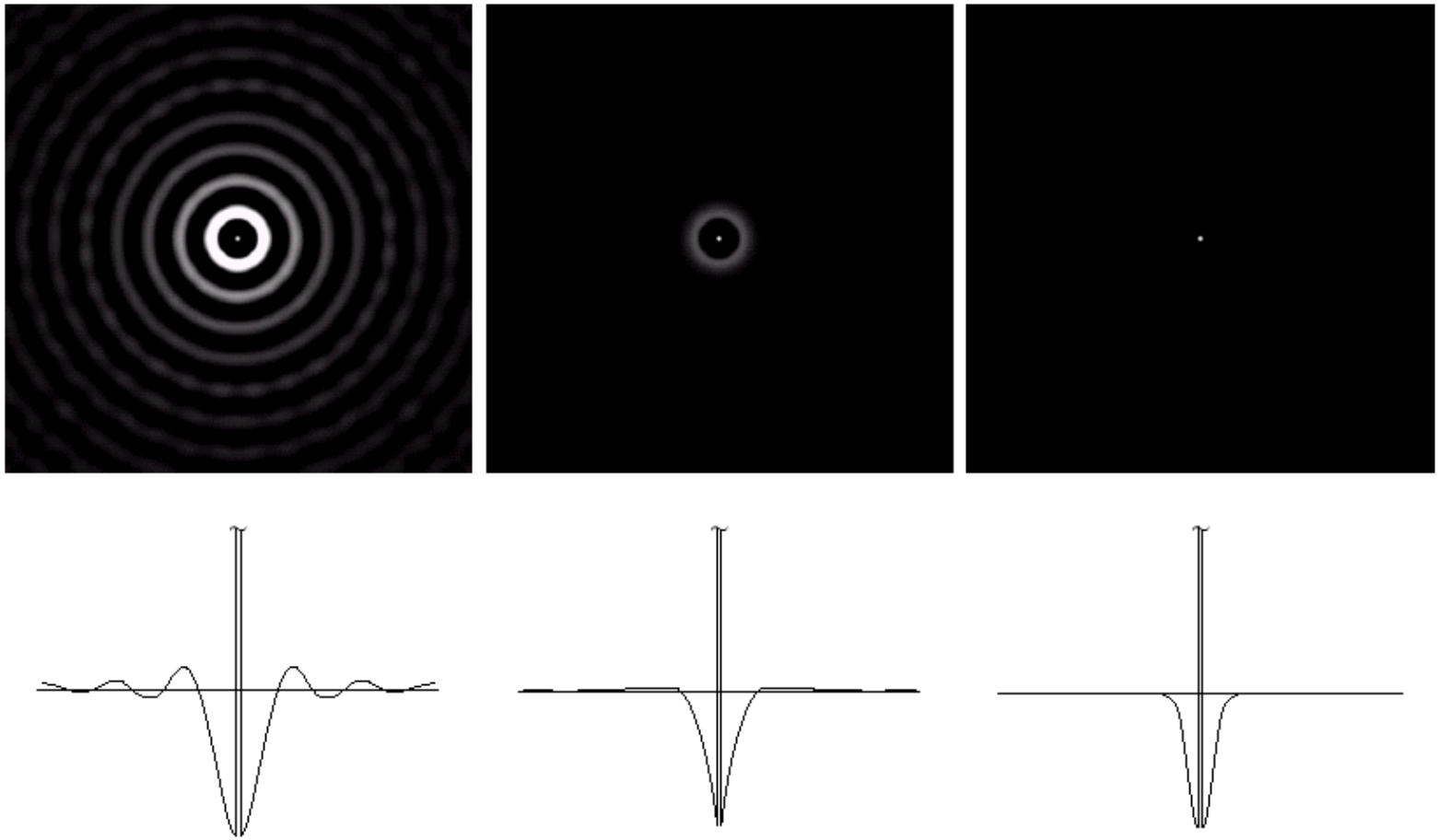
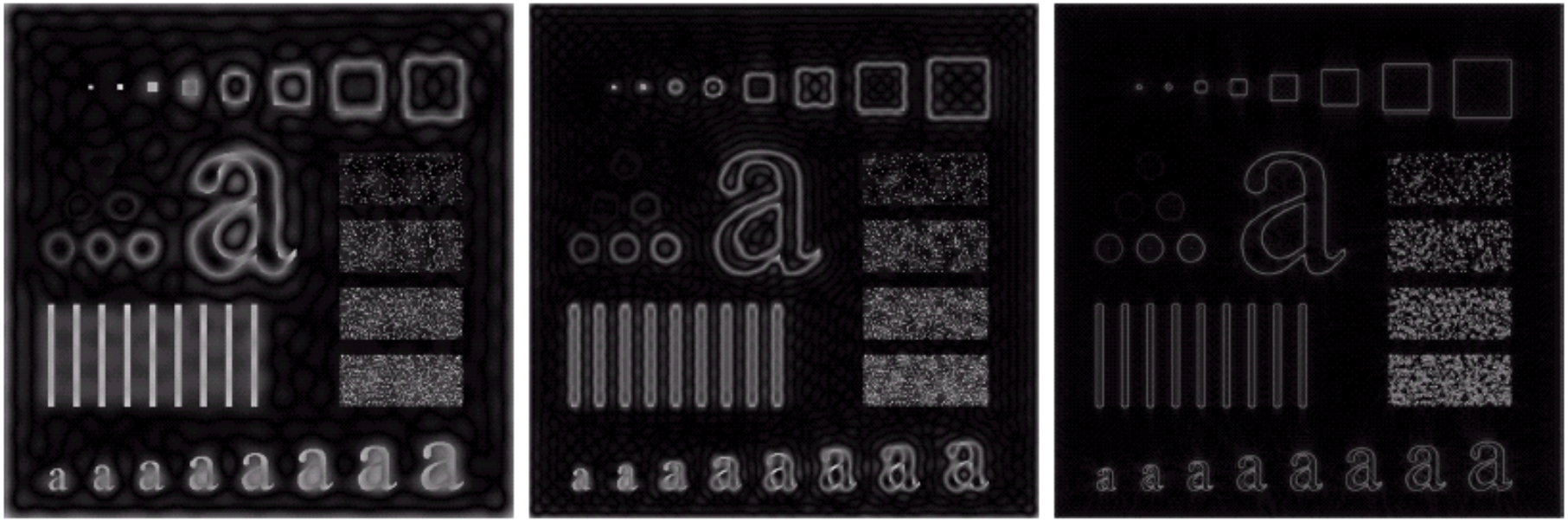


FIGURE 4.23 Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

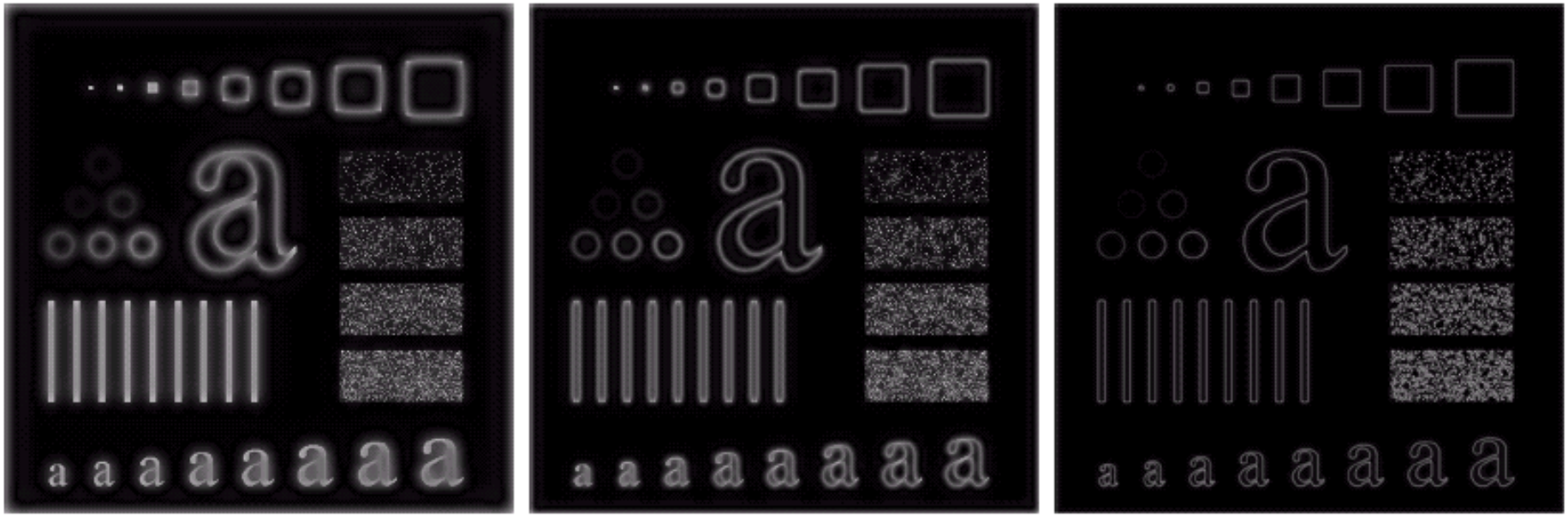
Enhancement in the Frequency Domain
Ideal highpass filter



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).

Enhancement in the Frequency Domain
Butterworth highpass filter



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.

Enhancement in the Frequency Domain
Gaussian highpass filter

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.

Enhancement in the Frequency Domain
Laplacian filter

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

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

Thus, a Laplacian filter in the Fourier domain is described as:

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

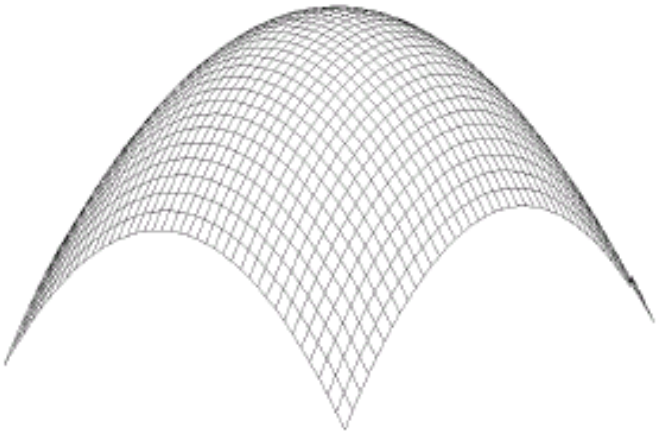
Or, when shifted:

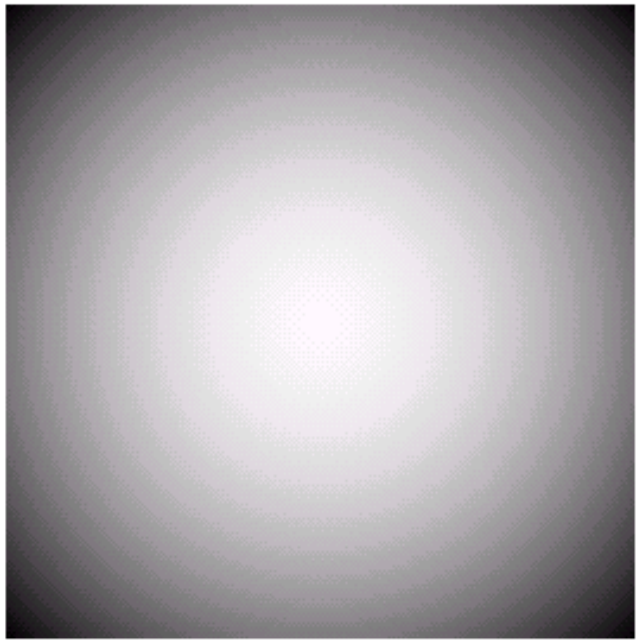
$$H(u,v) = -((u - M/2)^2 + (v - N/2)^2)$$

Enhancement in the Frequency Domain
Laplacian filter

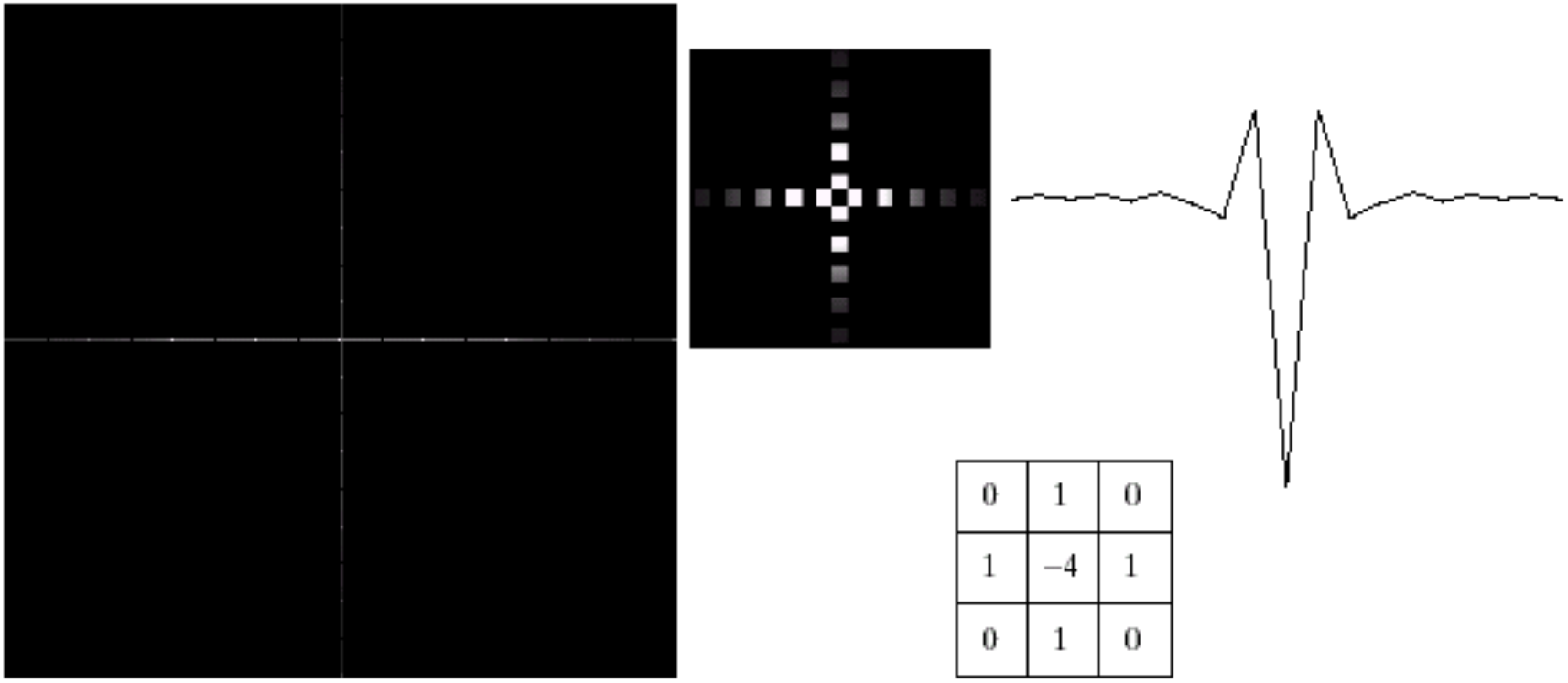
FIGURE 4.27 (a) 3-D plot of Laplacian in the frequency domain. (b) Image representation of (a).

(value at the top of the dome is zero)





Enhancement in the Frequency Domain
Laplacian filter

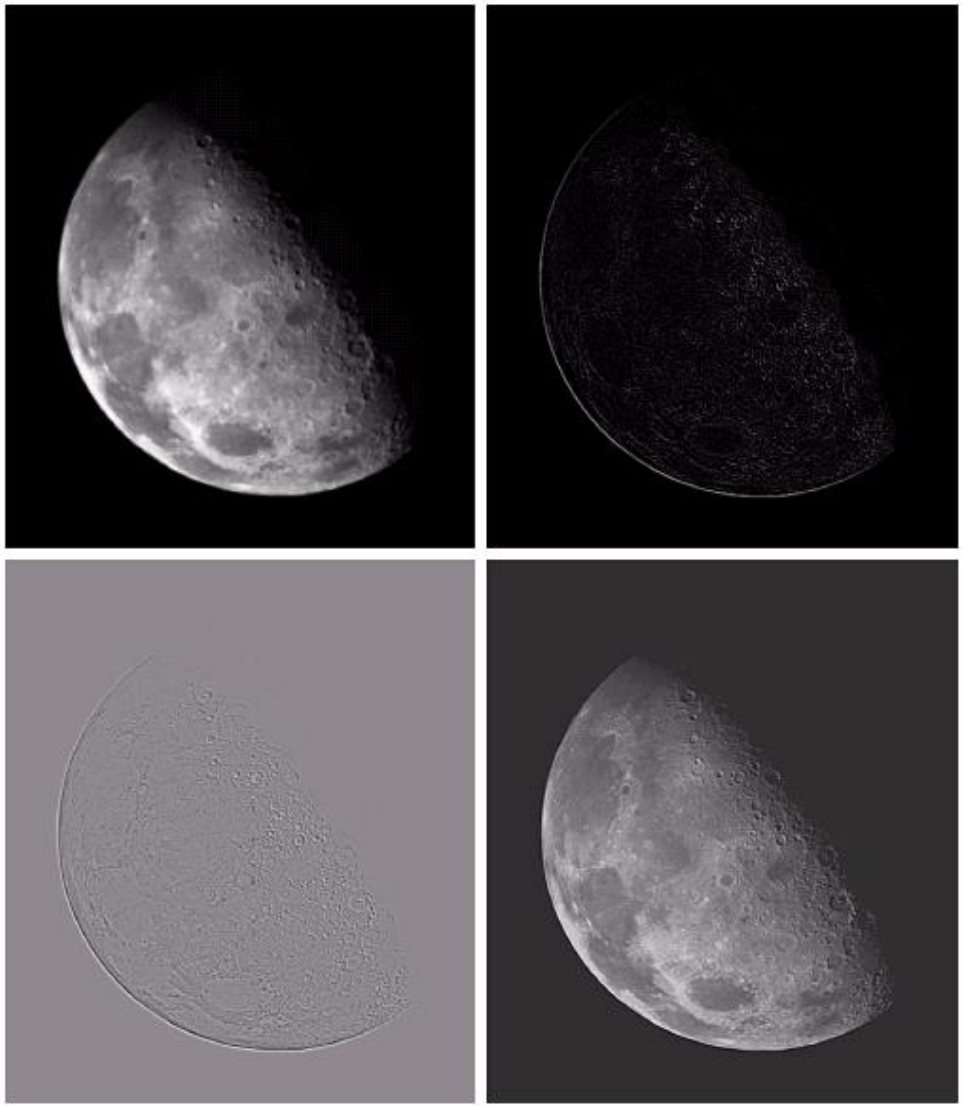


0	1	0
1	−4	1
0	1	0

a b
c d e
f

FIGURE 4.27 (a) 3-D plot of Laplacian in the frequency domain. (b) Image representation of (a). (c) Laplacian in the spatial domain obtained from the inverse DFT of (b). (d) Zoomed section of the origin of (c). (e) Gray-level profile through the center of (d). (f) Laplacian mask used in Section 3.7.

Enhancement in the Frequency Domain
Using the Laplacian filter



a b
c d

FIGURE 4.28
(a) Image of the North Pole of the moon.
(b) Laplacian filtered image.
(c) Laplacian image scaled.
(d) Image enhanced by using Eq. (4.4-12).
(Original image courtesy of NASA.)

Unsharp masking

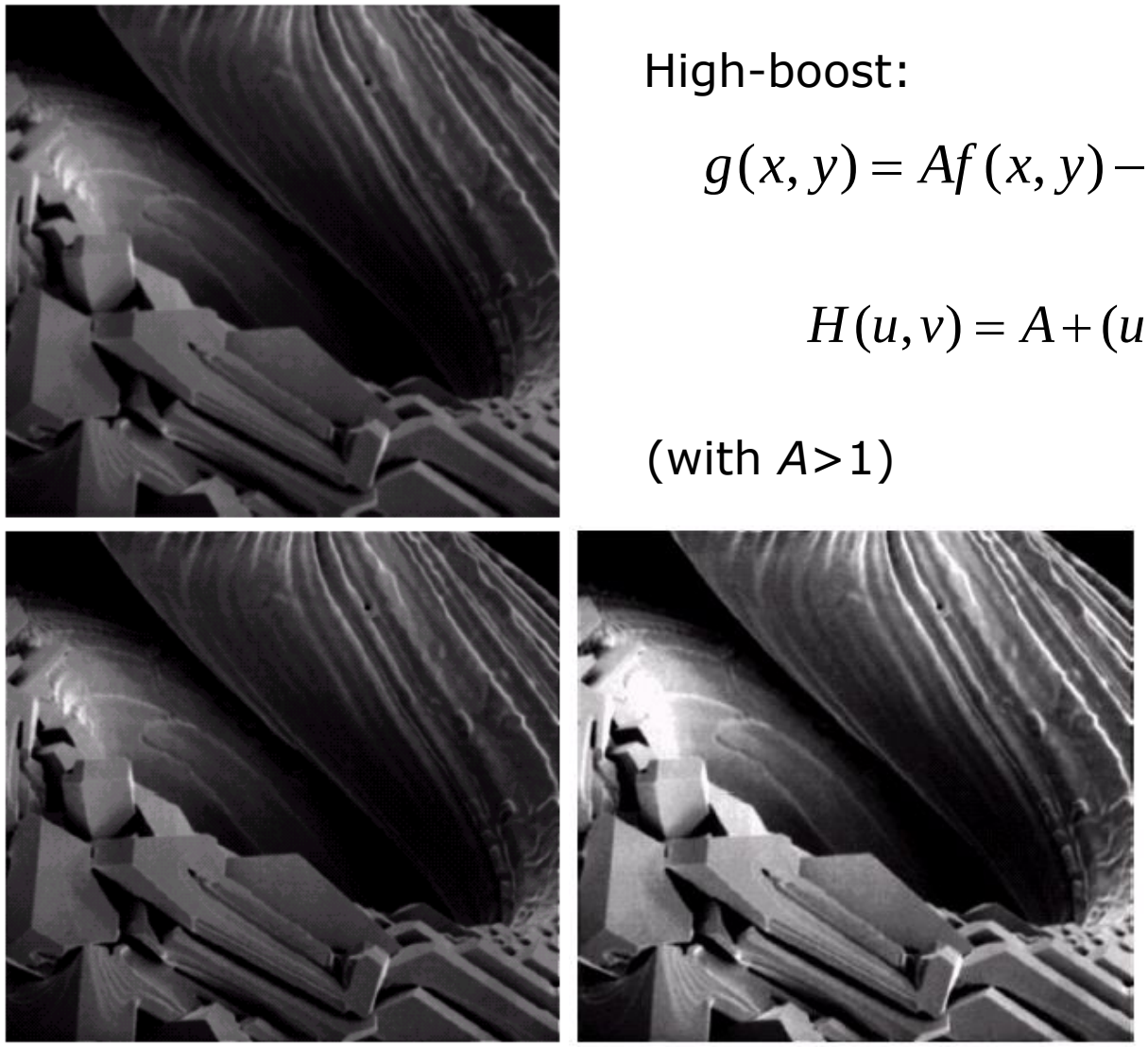
$$g(x,y) = f(x,y) - \nabla^2 f(x,y)$$

Or, equivalently:

$$H(u,v) = 1 + (u^2 + v^2)$$

(Beware of highpass filter amplification)

Enhancement in the Frequency Domain
Using the Laplacian filter



a b
c d

FIGURE 4.29
Same as Fig. 3.43, but using frequency domain filtering. (a) Input image. (b) Laplacian of (a). (c) Image obtained using Eq. (4.4-17) with $A = 2$. (d) Same as (c), but with $A = 2.7$. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

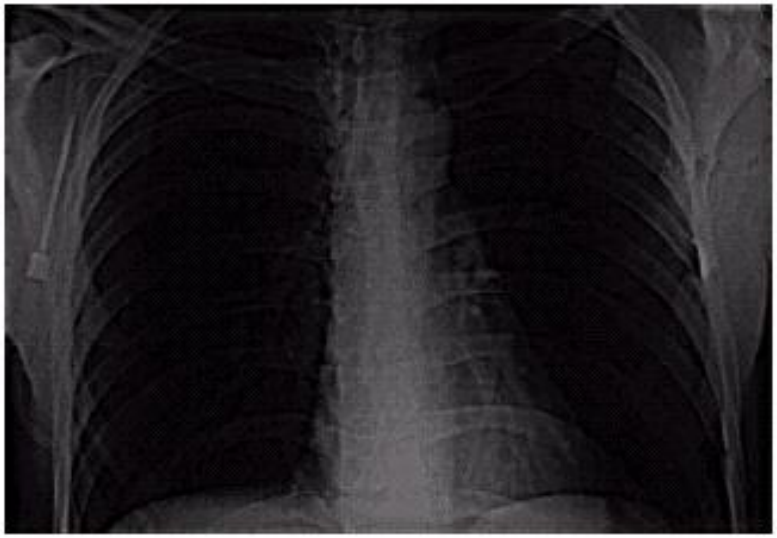
High-boost:

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

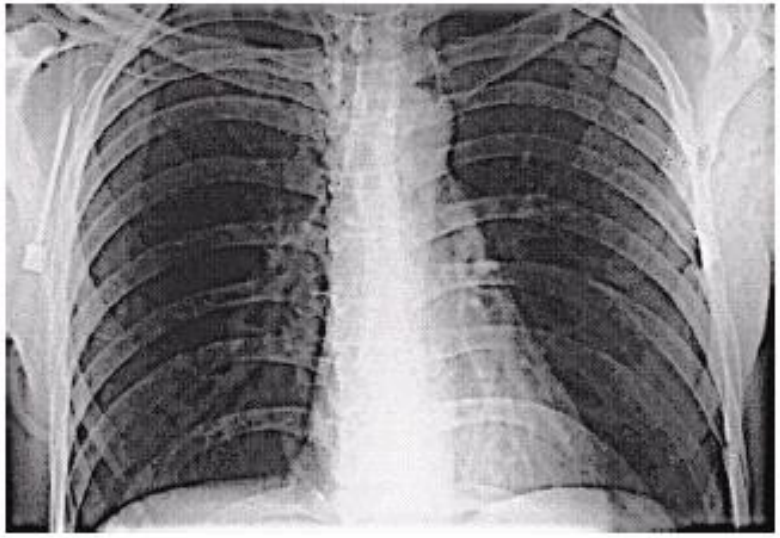
$$H(u,v) = A + (u^2 + v^2)$$

(with $A>1$)

Enhancement in the Frequency Domain Using the Laplacian filter



Unsharp masking followed by histogram equalization



Design of linear filters

[J.S. Lim]

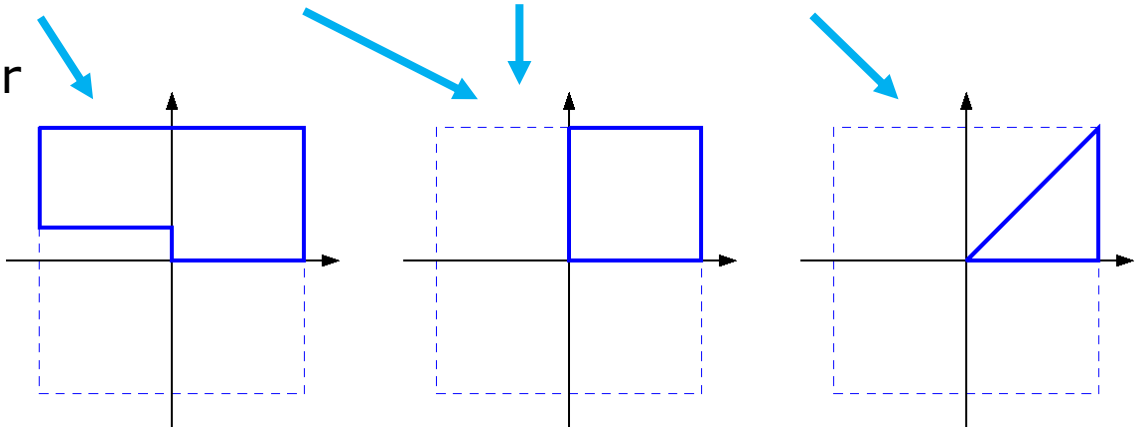
We restrict ourselves to **FIR** filters with **real** coefficients and **zero** (or linear) phase. Nonlinear-phase filters distort the image: different frequency components that make up edges and details lose proper registration.

Zero phase implies **symmetry wrt the origin** for the coefficients:

$$H(u,v) = H^*(u,v) \quad \Leftrightarrow \quad h(x,y) = h^*(-x,-y) = h(-x,-y)$$

Additional symmetries may be expedient to obtain more isotropic responses: $h(x,y) = h(-x,-y) = h(-x,y) = h(x,-y) = h(y,x)$

Independent coefficients for twofold (origin), fourfold, eightfold symmetries are:



e.g. 5x5 case:

```
xxxxx   ooxxx   oooox
xxxxx   ooxxx   oooxx
ooxxx   ooxxx   ooxxx
ooooo   ooooo   ooooo
ooooo   ooooo   ooooo
```

Design of linear filters

Data-domain coefficients of 2-D FIR filters can be determined by extending methods known for the 1-D case:

Windowing

- The desired frequency response $Hd(u,v)$ is assumed known;
- by IDFT/IDTFT the desired impulse response $hd(x,y)$ is determined, which may have large or infinite extent;
- $hd(x,y)$ is weighted and cropped via multiplication by a suitable window function: $h(x,y)=hd(x,y)w(x,y)$.
- In the Fourier domain one obtains: $H(u,v)=conv[Hd(u,v),W(u,v)]$.

The window is derived from a 1-D window (rectangular, Hamming, Kaiser,…) via a separable approach (square window) or by rotation.

Frequency sampling

Hd is sampled at equally spaced points on the Cartesian grid, and an IDFT is performed. It is expedient to avoid sharp changes in *Hd*: define a *transition region* to do this.

E.g., for a lowpass:

```
00000000000
00000x00000
0000x1x0000
000x111x000
00x11111x00
000x111x000
0000x1x0000
00000x00000
00000000000
```

Design of linear filters

A 2-D function $h(x,y)$ is said to be ***separable*** if $h1$, $h2$ exist such that

$$h(x,y)=h1(x)h2(y)$$

e.g., the *impulse* and *unit step* functions are separable

In general, a 2-D function is NOT separable; note indeed that an NxM nonseparable function has NxM degrees of freedom, $N+M$ if separable.

Separable filters permit a convenient implementation of the 2-D convolution:

$$g(x,y)=\sum_{s=-N/2}^{N/2}\sum_{t=-M/2}^{M/2}h(s,t)f(x-s,y-t)=\sum_{s=-N/2}^{N/2}h_1(s)\sum_{t=-M/2}^{M/2}f(x-s,y-t)h_2(t)$$

i.e., one can perform a sequence of two 1-D convolutions, first by image rows (columns) then by columns (rows)

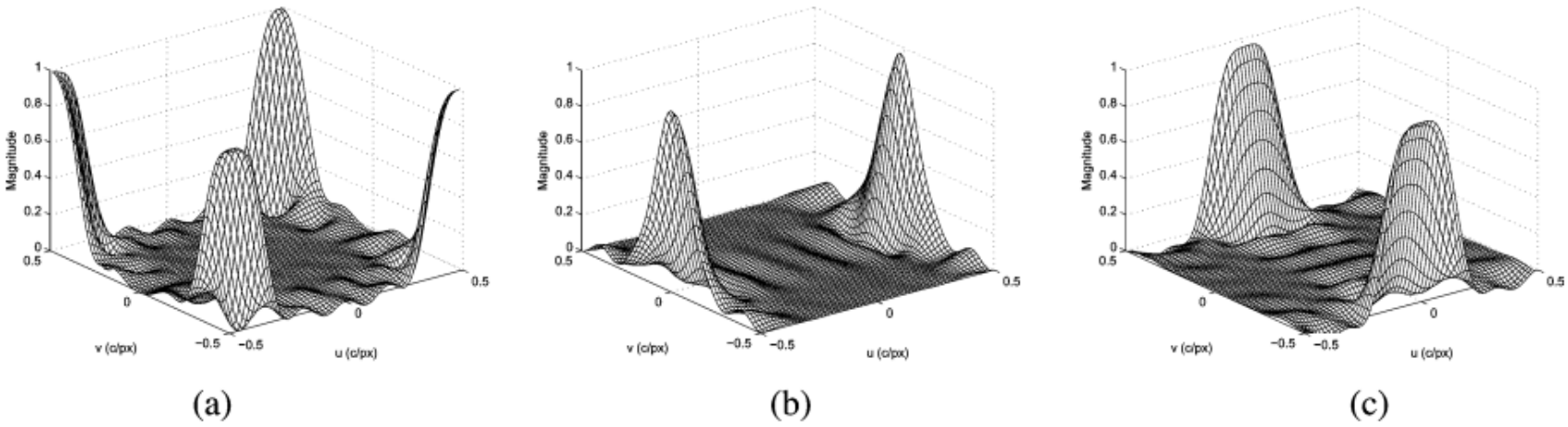
Design of linear filters

The simplest way to design a 2-D filter is to start from a suitable pair of 1-D filters and realize a **separable** 2-D filter.

Example: 11x11 anisotropic bandpass filters for MMSE *demosaicking*:

LEUNG *et al.*: LEAST-SQUARES LUMA–CHROMA DEMULTIPLEXING ALGORITHM FOR BAYER DEMOSAICKING

zz07_Demosaicking



Other examples of anisotropic and semi-isotropic filters (Matlab)…

If max. **isotropy** is desired →

Design of linear filters

McClellan transformation

Suppose we have a *zero-phase, symmetrical 1-D FIR filter*, whose frequency response is expressed as:

$$H(\omega) = \sum_{n=-N}^{N} h(n)\exp(-j\omega n) = h(0) + \sum_{n=1}^{N} 2h(n)\cos\omega n = \sum_{n=0}^{N} a(n)\cos\omega n = \sum_{n=0}^{N} b(n)(\cos\omega)^n$$

The *Chebyshev polynomials* of the first kind, T_n, are used above.

They are defined as $\cos(\omega n) = T_n(\cos\omega)$, and can be built iteratively:

$$T_0(x) = 1; \quad T_1(x) = x; \quad T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$$

Design of linear filters

Determine a mapping for the frequency domain such that:

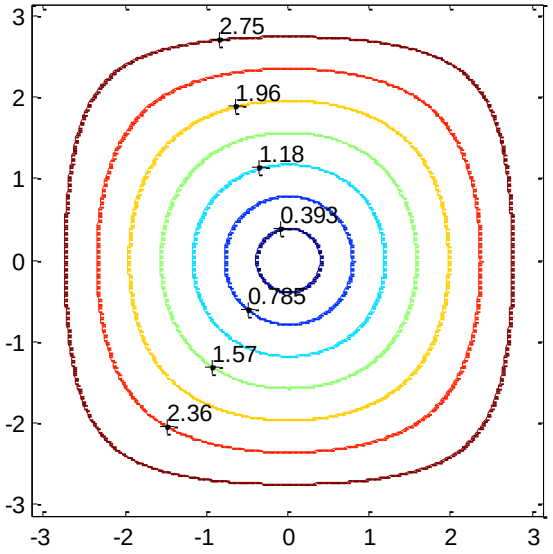
$$H(u,v) = H(\omega)\,|_{\omega=G(u,v)}$$

The function *G* maps each frequency into a 2-D contour. McClellan chose the mapping: $\omega = G(u,v) = \cos^{-1} Q(u,v)$

with $Q(u,v) = -\frac{1}{2} + \frac{1}{2}\cos(u) + \frac{1}{2}\cos(v) + \frac{1}{4}\cos(u+v) + \frac{1}{4}\cos(u-v)$

The resulting mapping shows good circular symmetry, especially in the low-frequency region

[mcclellan.m]



Design of linear filters

We shall not deal with the design of **IIR** 2-D filters:

the fundamental theorem of algebra does not extend to 2-D!

- → It is not possible in general to factorize a 2-D polynomial
- → The pole-zero stability analysis cannot be performed
- → Simple cascade realizations of low-order sections cannot be obtained

Theorems exist however to determine the stability of an IIR filter, based on its 2-D z-transform

It is also possible to constrain the design to filters having separable denominator. These however constitute a very restricted class

Enhancement in the Frequency Domain
Nonlinear filters: alpha rooting

Alpha rooting is a simple method originally proposed by Aghagolzadeh and Ersoy [32] and later modified by Agaian [28], [29] that can be used in combination with many different orthogonal transforms such as the Fourier, Hartley, and Haar wavelet and cosine transforms. The mathematical form of this operation can be seen as follows:

$$\widehat{X}(p,s) = X(p,s)\,|X(p,s)|^{\alpha-1} = |X(p,s)|^{\alpha}\, e^{j\theta(p,s)}$$

where $X(p,s)$ is the transform coefficients of the image x *(m,n)* . α is a user defined operating parameter, and $\theta(\mathrm{p},\mathrm{s})$ is the phase of the transform coefficients. The resulting output shows an emphasis on the high frequency content of the image without changing the phase of the image results in an overall contrast enhancement of the entire image. This enhancement is not without its consequences, sometimes resulting in ugly artifacts [17].

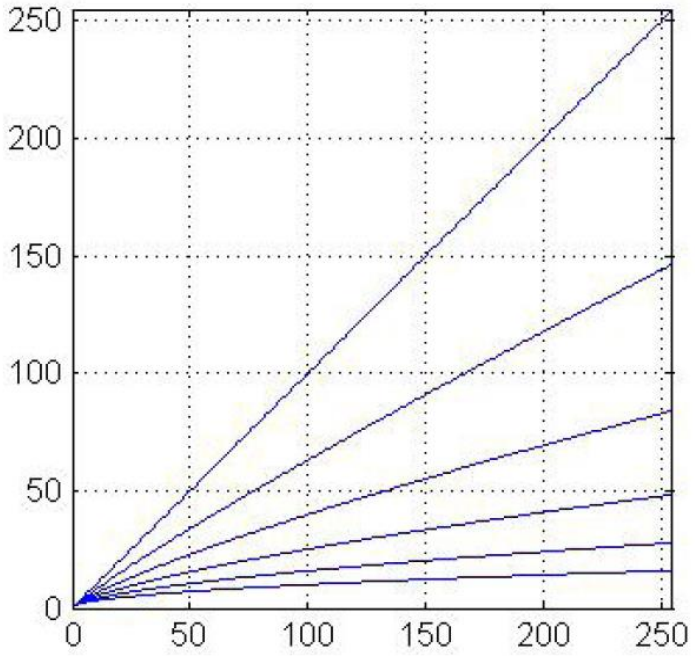
S. Agaian

Trans. IP
March 07

Enhancement in the Frequency Domain
Nonlinear filters: alpha rooting

Smaller transform coefficients (= tipically, higher frequencies) are *less* attenuated.

```
x = 0:1:255;
for alpha=.5:.1:1;
    y = x .^ alpha;
    plot(x,y); hold on; grid on;
end;
```



-- Note: the signal power is reduced. It may be recovered via a suitable scaling; e.g.:

$$g_{scaled} = g \sqrt{\overline{f^2} / \overline{g^2}}$$

-- The actual scaling factor *depends on the input image content*

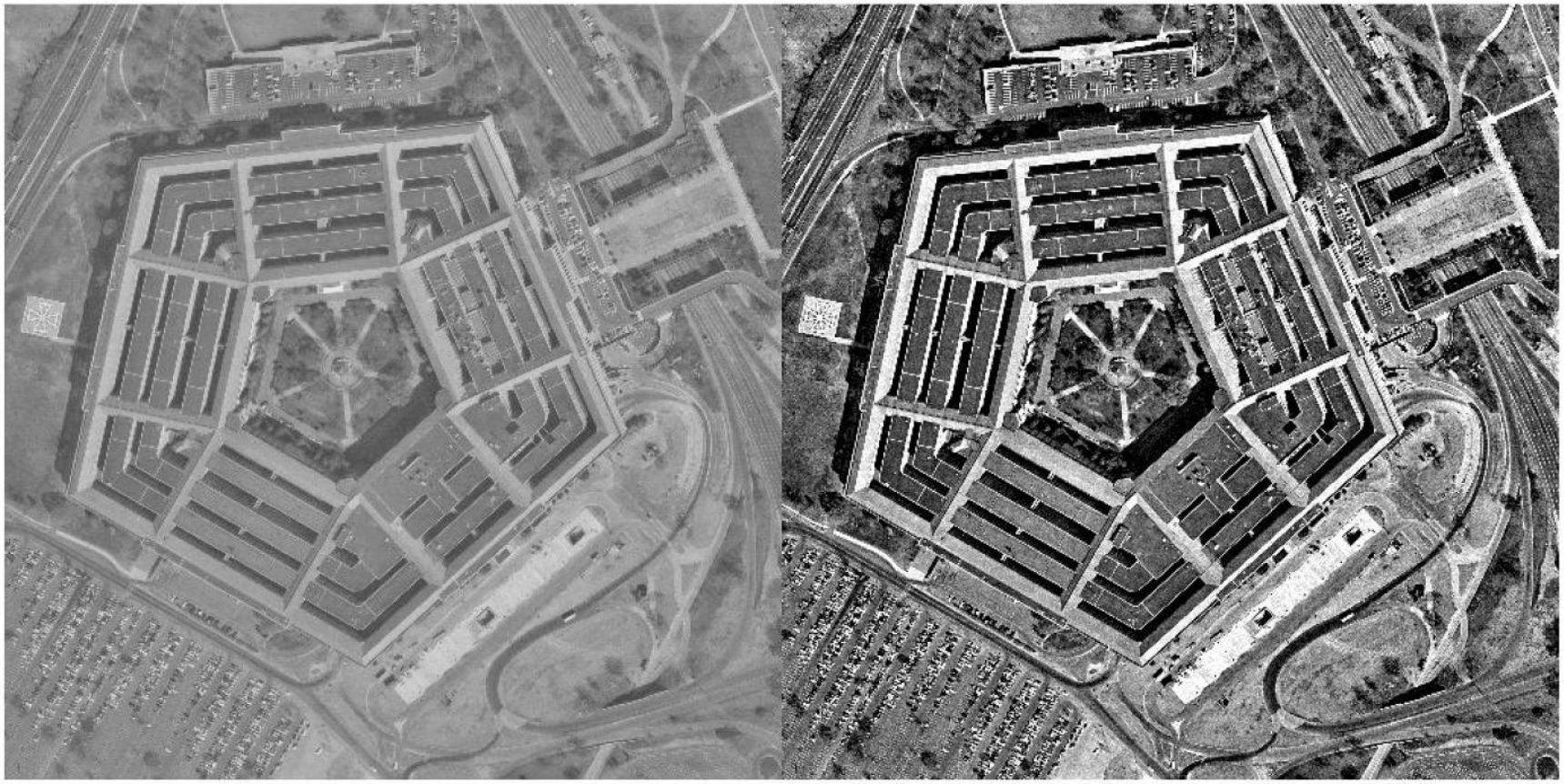
-- The output image is darker than the original, since the most strongly attenuated component is the largest: the dc one.

-- The output range may include *negative* values, which must be taken care of

Enhancement in the Frequency Domain
Nonlinear filters: alpha rooting

Alpha = 1

Alpha = 0.8



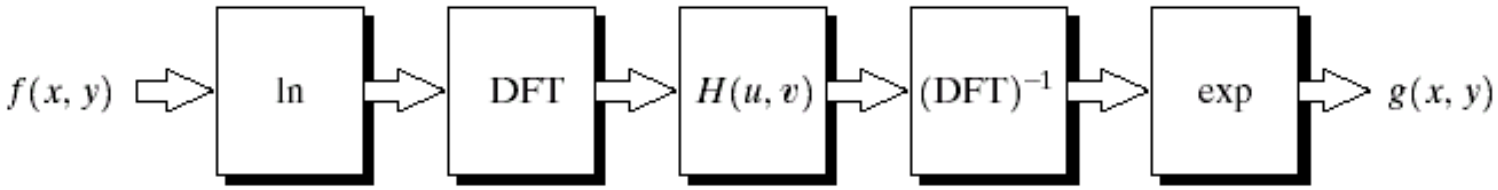
Enhancement in the Frequency Domain
Nonlinear filters: **Homomorphic filter**

A precursor of the Retinex filter, in the transform domain

Image formation model based on illumination and reflectance:

$$f(x,y) = i(x,y)r(x,y) \quad \rightarrow \quad \ln(f) = \ln(i) + \ln(r)$$

Illumination and reflectance can be separated in the log domain;



the former is supposed to vary smoothly and is attenuated, the latter is supposed to vary sharply and is amplified

(γ_L<1<γ_H)

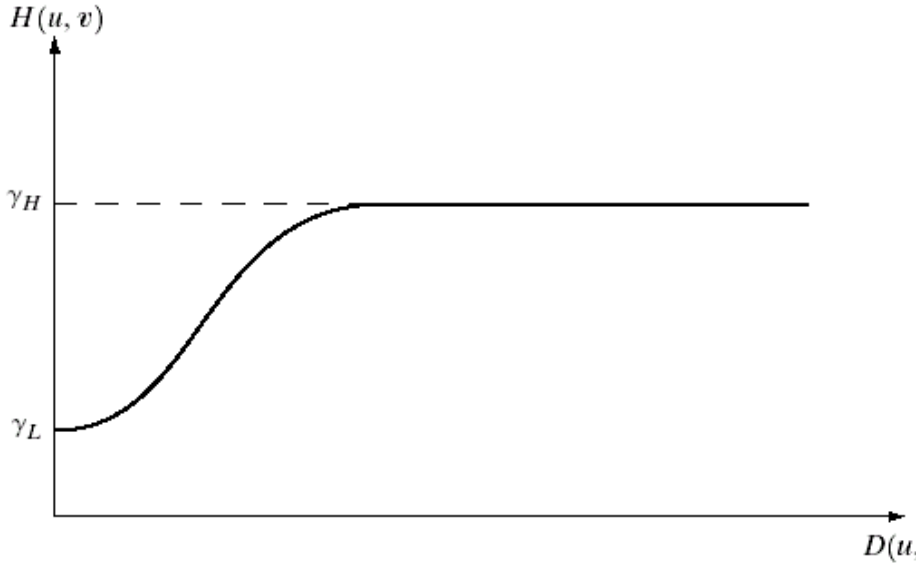


FIGURE 4.32 Cross section of a circularly symmetric filter function. $D(u, v)$ is the distance from the origin of the centered transform.

Enhancement in the Frequency Domain
Nonlinear filters: Homomorphic filter