# CSL7320: Digital Image Analysis

Filtering in Frequency Domain

# An overview

# Spatial-Frequency Domain Transformation

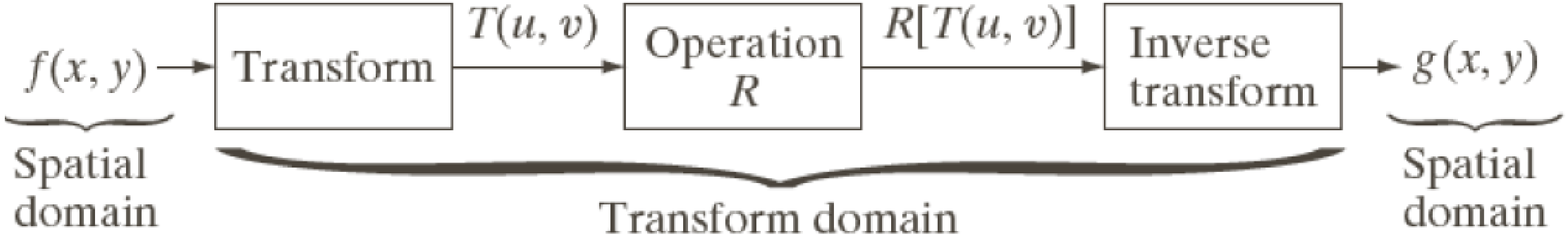



**FIGURE 2.39** General approach for operating in the linear transform domain.

2-D *linear transforms*

Input image

Forward transformation kernel

*transform pair*

$$T(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\boxed{r(x,y,u,v)},$$

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} T(u,v)\boxed{s(x,y,u,v)}$$

Inverse transformation kernel

$$u=0,1,\ldots,M-1$$
$$v=0,1,\ldots,N-1$$
$$x=0,1,\ldots,M-1$$
$$y=0,1,\ldots,N-1$$

$M$ and $N$ are the row and column dimensions of $f$

Slide credit: Yan Tong

# Spatial-Frequency Domain Transformation

The forward transformation kernel is said to be *separable* if

$$r(x, y, u, v) = r_1(x, u) r_2(y, v) \tag{2-57}$$

In addition, the kernel is said to be *symmetric* if $r_1(x,u)$ is functionally equal to $r_2(y,v)$, so that

$$r(x, y, u, v) = r_1(x, u) r_1(y, v) \tag{2-58}$$

Identical comments apply to the inverse kernel.

Slide credit: Yan Tong

# Spatial-Frequency Domain Transformation

***Fourier Transform***

Forward $$r(x,y,u,v) = e^{-j2\pi(ux/M+vy/N)}$$

Inverse $$s(x,y,u,v) = \frac{1}{MN} e^{j2\pi(ux/M+vy/N)}$$

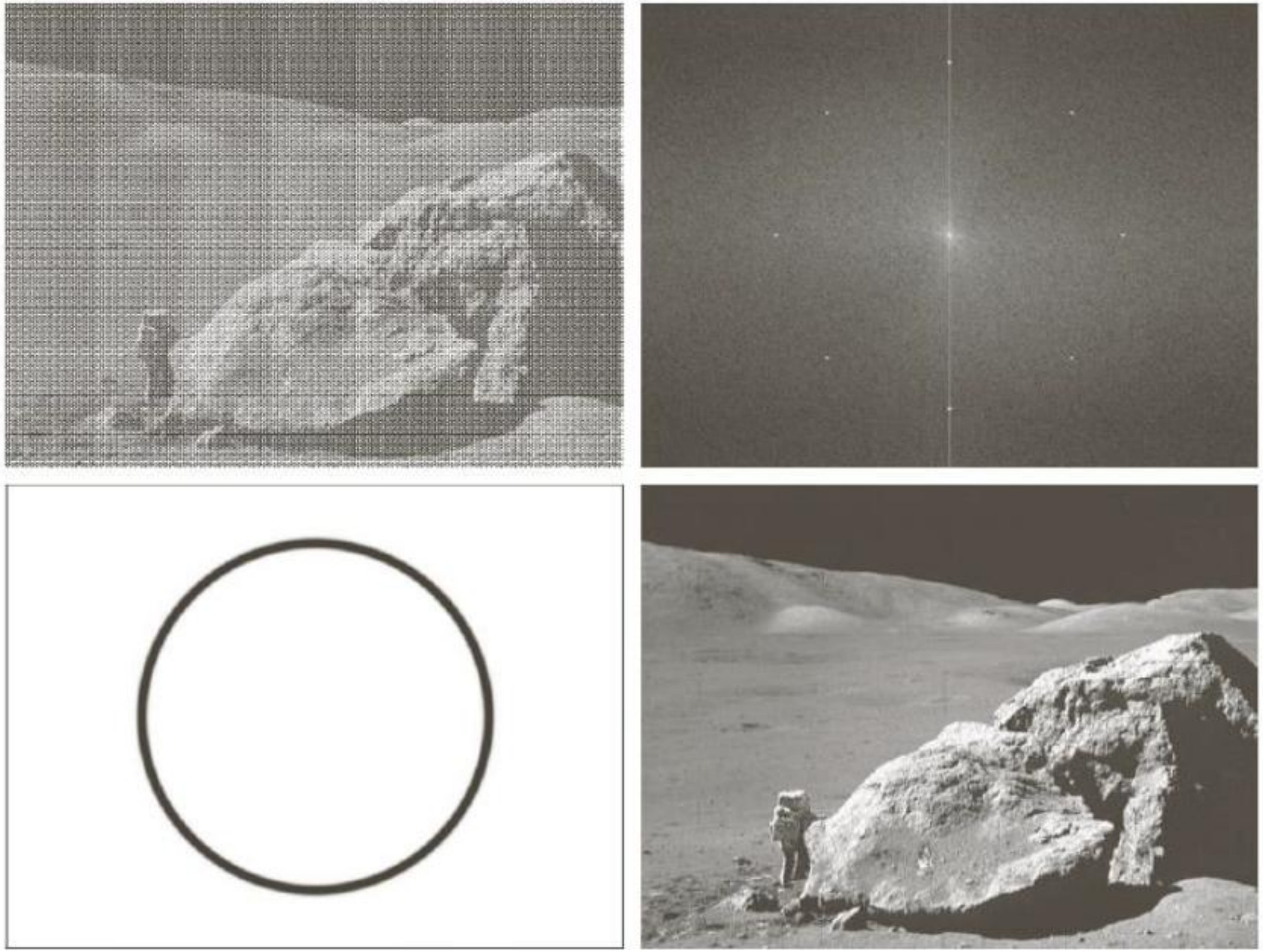
***Discrete Fourier Transform***

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

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

https://www.youtube.com/watch?v=9mLeVn8xzMw

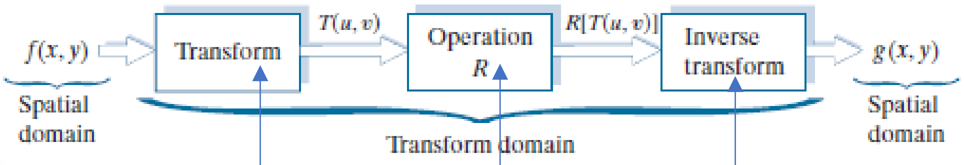
# Spatial-Frequency Domain Transformation



a b
c d

**FIGURE 2.40**
(a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy of NASA.)

Sinusoidal interference occurs when two or more sinusoidal waves overlap, resulting in a new wave with a different amplitude.



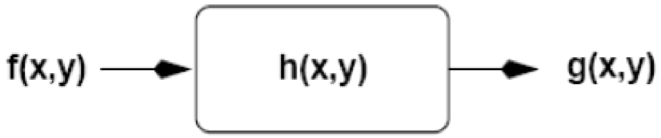

Figures (b) (b) (c)

$\underbrace{\text{Multiplied by filter (c)}}_{\text{Modified transform}}$

# Filtering in Frequency Domain

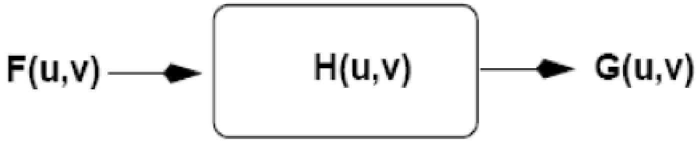
# Spatial Vs Frequency Domain

## Spatial Domain




$g(x,y) = f(x,y) * h(x,y)$

## Frequency Domain




$G(u,v) = F(u,v)\,H(u,v)$

$( g(x,y) = \mathbf{F}^{-1}(F(u,v)\,H(u,v)) )$

Slide credit: Ashish Ghosh

# 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.

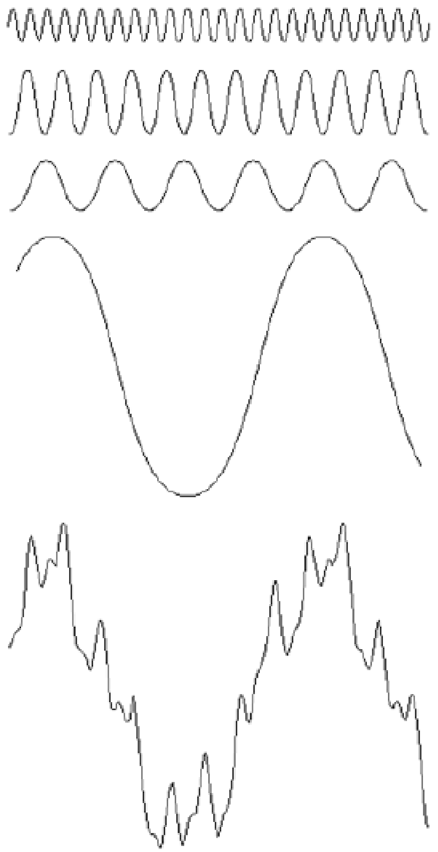


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Slide credit: Ashish Ghosh

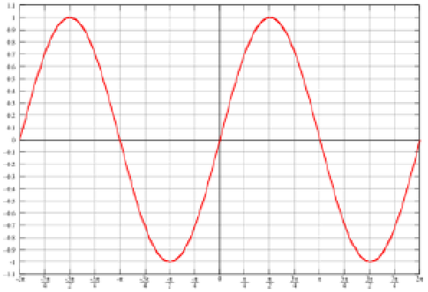
# Fourier Series

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

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

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

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



Where $T_0$ is the period.

Slide credit: Ashish Ghosh

# 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.

Slide credit: Ashish Ghosh

# Fourier Transform

- Fourier transform
  - Functions which 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 weighting function
  - Its utility is greater than the Fourier series in most practical problems
- A function, expressed in either as a Fourier series or a Fourier transform, can be reconstructed (recovered) completely via an inverse process, with no loss of information

Slide credit: Ashish Ghosh

# 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
- Each value of ***u*** determines the frequency of its corresponding sine-cosine pair.

Slide credit: Ashish Ghosh

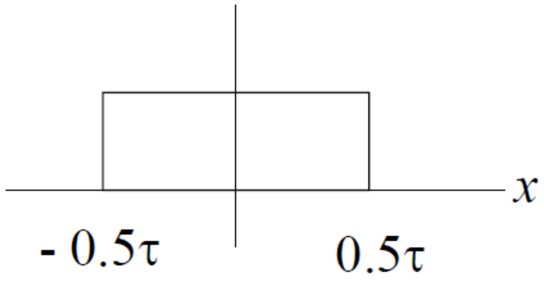
# Continuous One-Dimensional Fourier Transform and Its Inverse

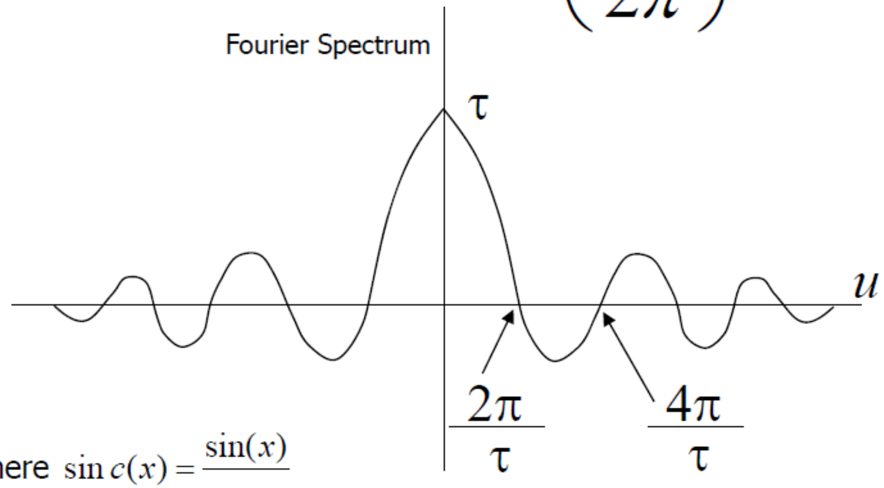
**Example**

**Find the Fourier transform of a gate function $\Pi(t)$ defined by**

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

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




where $\sin c(x) = \frac{\sin(x)}{x}$

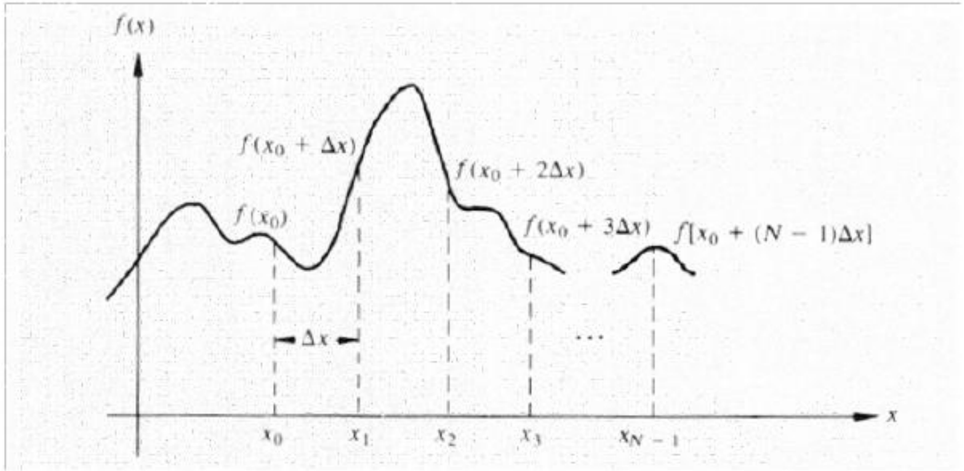
Slide credit: Ashish Ghosh

# Discrete One-Dimensional Fourier Transform and Its Inverse

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

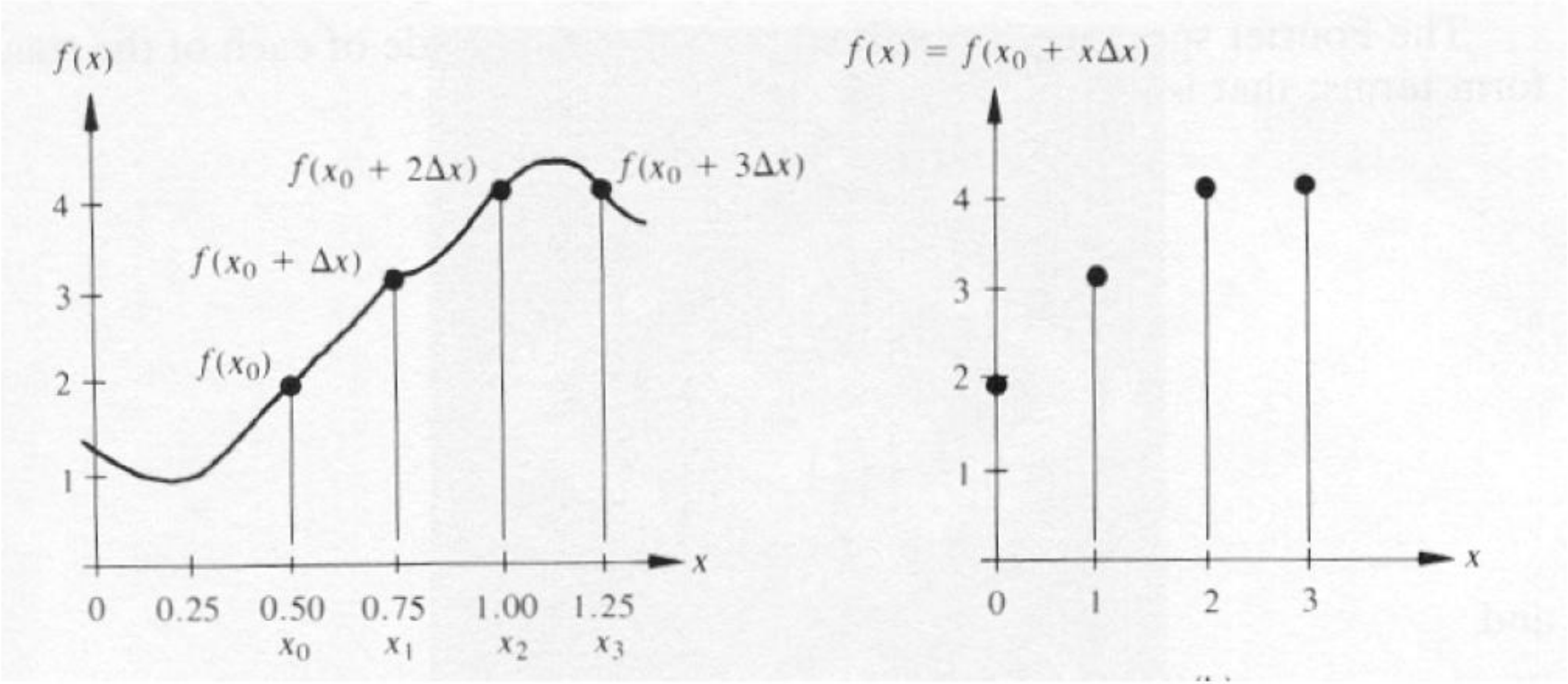
$$\{f(x_0), f(x_0+\Delta x), f(x_0+2\Delta x), \ldots, f(x_0+[M-1]\Delta x)\}$$

**by taking M samples Δx units apart.**




Slide credit: Ashish Ghosh

# Discrete One-Dimensional Fourier Transform and Its Inverse




Slide credit: Ashish Ghosh

# 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 continuous function.**

Slide credit: Ashish Ghosh

# Discrete One-Dimensional Fourier Transform and Its Inverse

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

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

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

Slide credit: Ashish Ghosh

# 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, $\Delta$u, 2$\Delta$u, …, (M-1)$\Delta$u.

  i.e. F(u) represents F(u$\Delta$u), where:

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

- Each term of the FT (F(u) for every u) is composed of the sum of all values of f(x)

Slide credit: Ashish Ghosh

# Discrete One-Dimensional Fourier Transform and Its Inverse

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

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

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

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

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

Slide credit: Ashish Ghosh

# Discrete One-Dimensional Fourier Transform and Its Inverse

- $|F(u)|$ (magnitude function) is the Fourier spectrum of f(x) and $\phi(u)$ its phase angle.

- 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).

Slide credit: Ashish Ghosh

# Discrete One-Dimensional Fourier Transform and Its Inverse

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

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

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

Slide credit: Ashish Ghosh

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

Slide credit: Ashish Ghosh

# Discrete 2-Dimensional Fourier Transform and Its Inverse

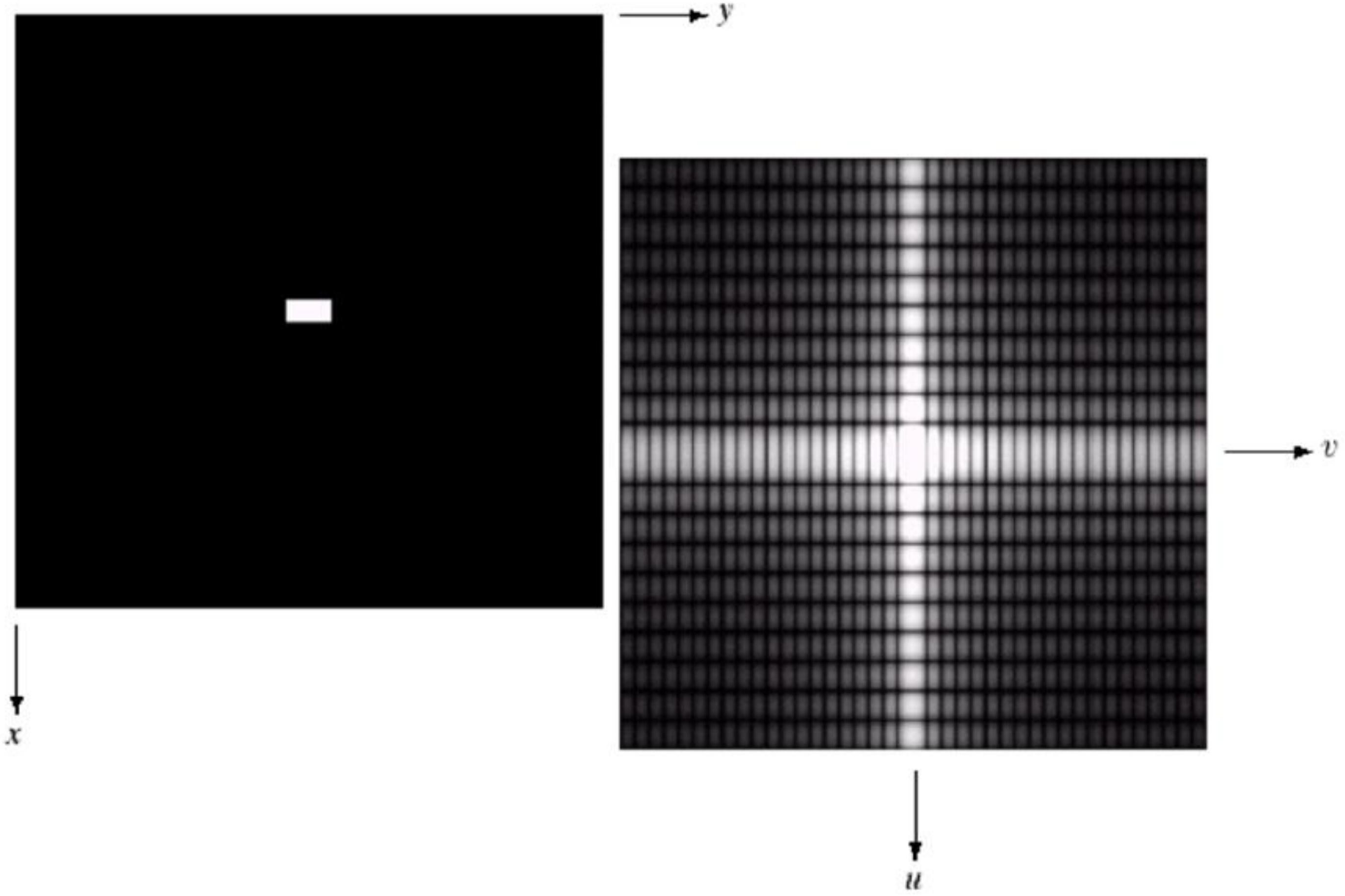
$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(u\frac{x}{M}+v\frac{y}{N})}$$

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

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

Slide credit: Ashish Ghosh

# Discrete 2-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

Slide credit: Ashish Ghosh

# Frequency Shifting Property of the Fourier Transform

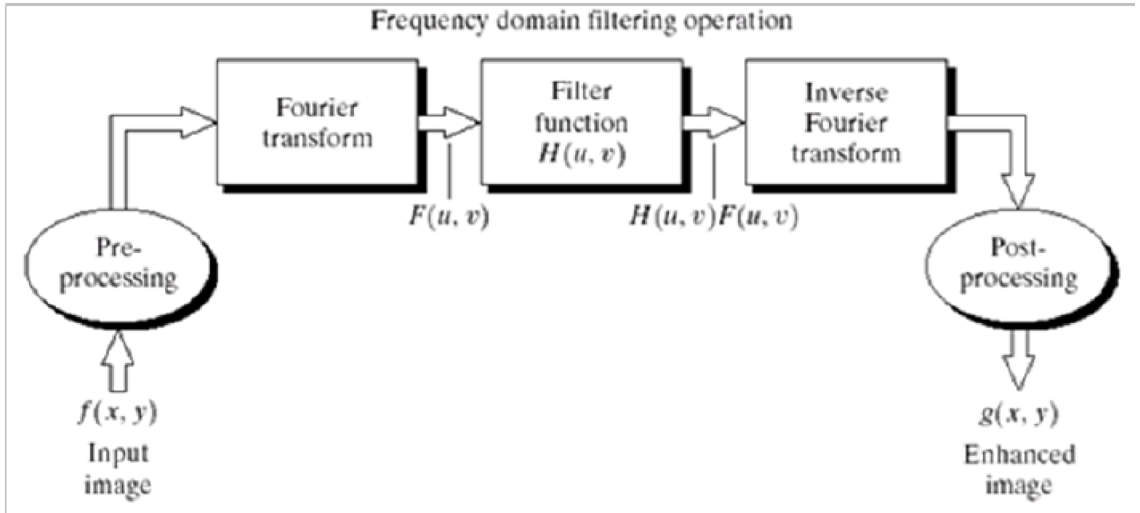
$$\text{If } f(x) \leftrightarrow F(u) \text{then}$$

$$f(x)e^{j2\pi u_0 x} \leftrightarrow F(u-u_0)$$

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

Slide credit: Ashish Ghosh

# Basic Filtering in the Frequency Domain



Basic steps for filtering in the frequency domain.

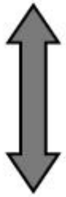
$$F\left(f(x,y)(-1)^{x+y}\right) = F(u-M/2, v-N/2)$$

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 $(-1)^{x+y}$

Slide credit: Ashish Ghosh

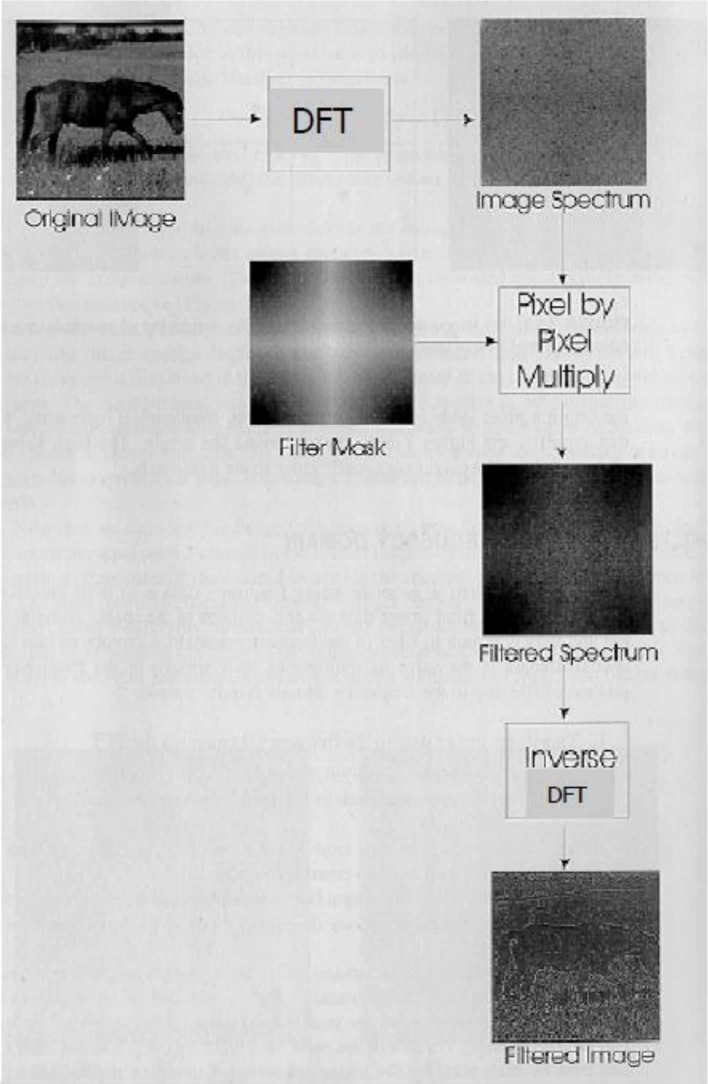
# FREQUENCY DOMAIN METHODS

$$f(x,y) * h(x,y) = g(x,y)$$



$$F(u,v)\, H(u,v) = G(u,v)$$

- ❑ H(u,v) is specified in the frequency domain.
- ❑ h(x,y) is specified in the spatial domain.




Slide credit: Ashish Ghosh

# Major Filter Categories

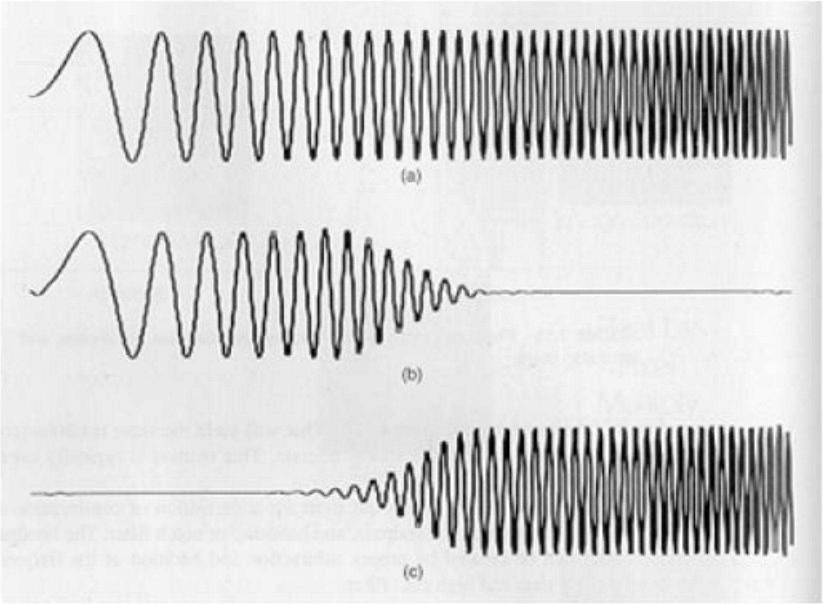
Typically, filters are classified by examining their properties in the frequency domain:

(1) Low-pass

(2) High-pass

Slide credit: Ashish Ghosh

# Example



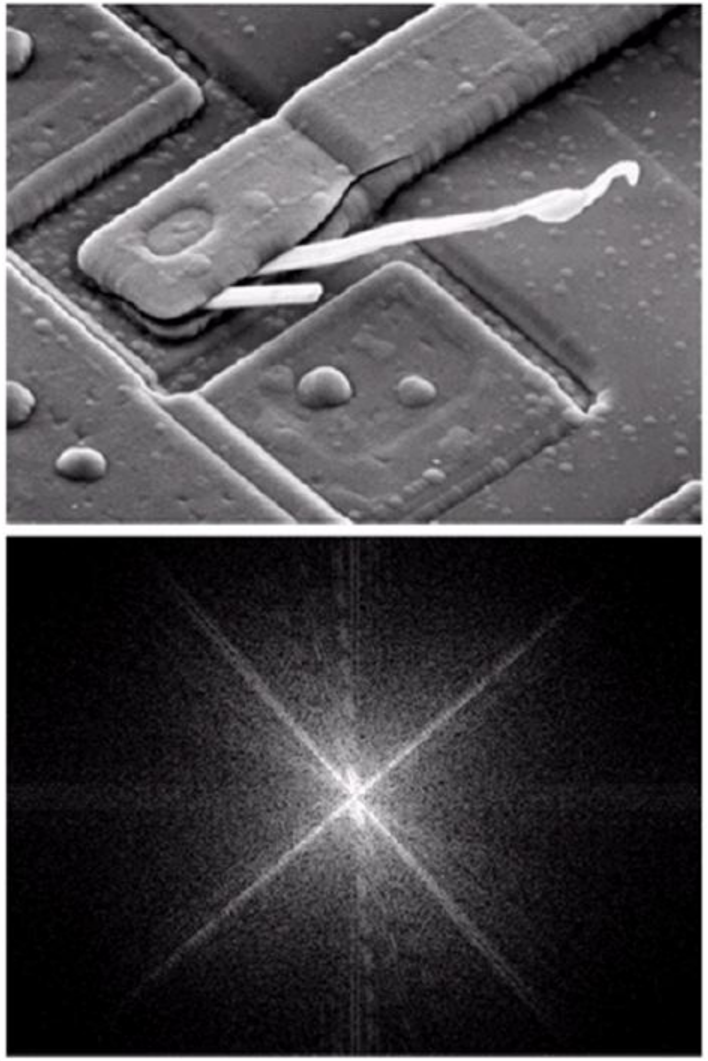

Original Signal

Low-Pass filtered output

High-Pass filtered output

Slide credit: Ashish Ghosh

# Filtering out the DC Frequency Component

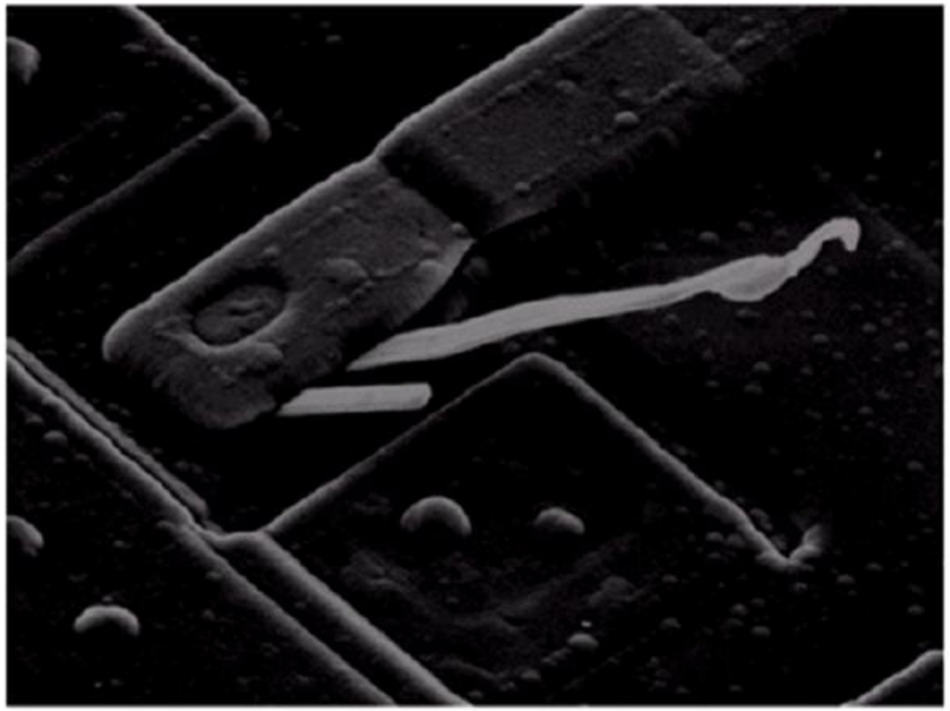


(a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

Slide credit: Ashish Ghosh

# Filtering out the DC Frequency Component

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**

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

Slide credit: Ashish Ghosh

# Low-pass and High-pass Filters

**Low Pass Filter**
attenuates high frequencies while "passing" low frequencies.

**High Pass Filter**
attenuates low frequencies while "passing" high frequencies.

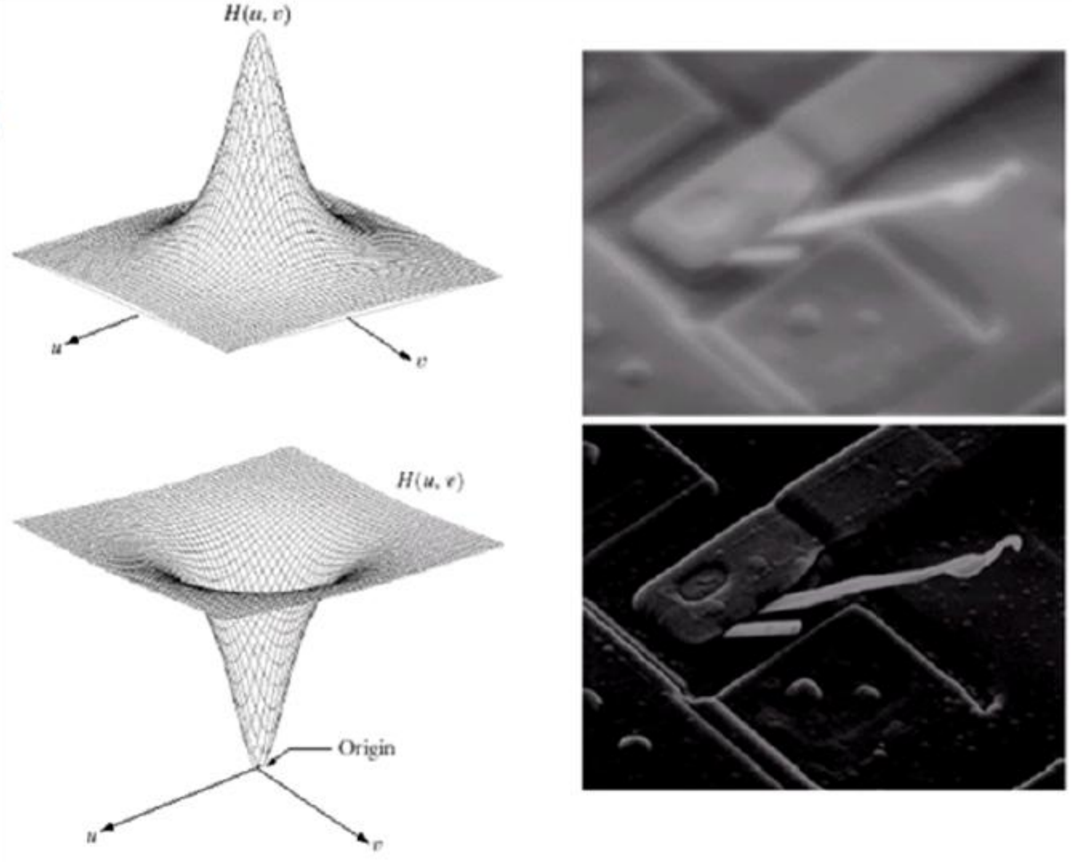



FIGURE (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

Slide credit: Ashish Ghosh

# Low-pass and High-pass Filters

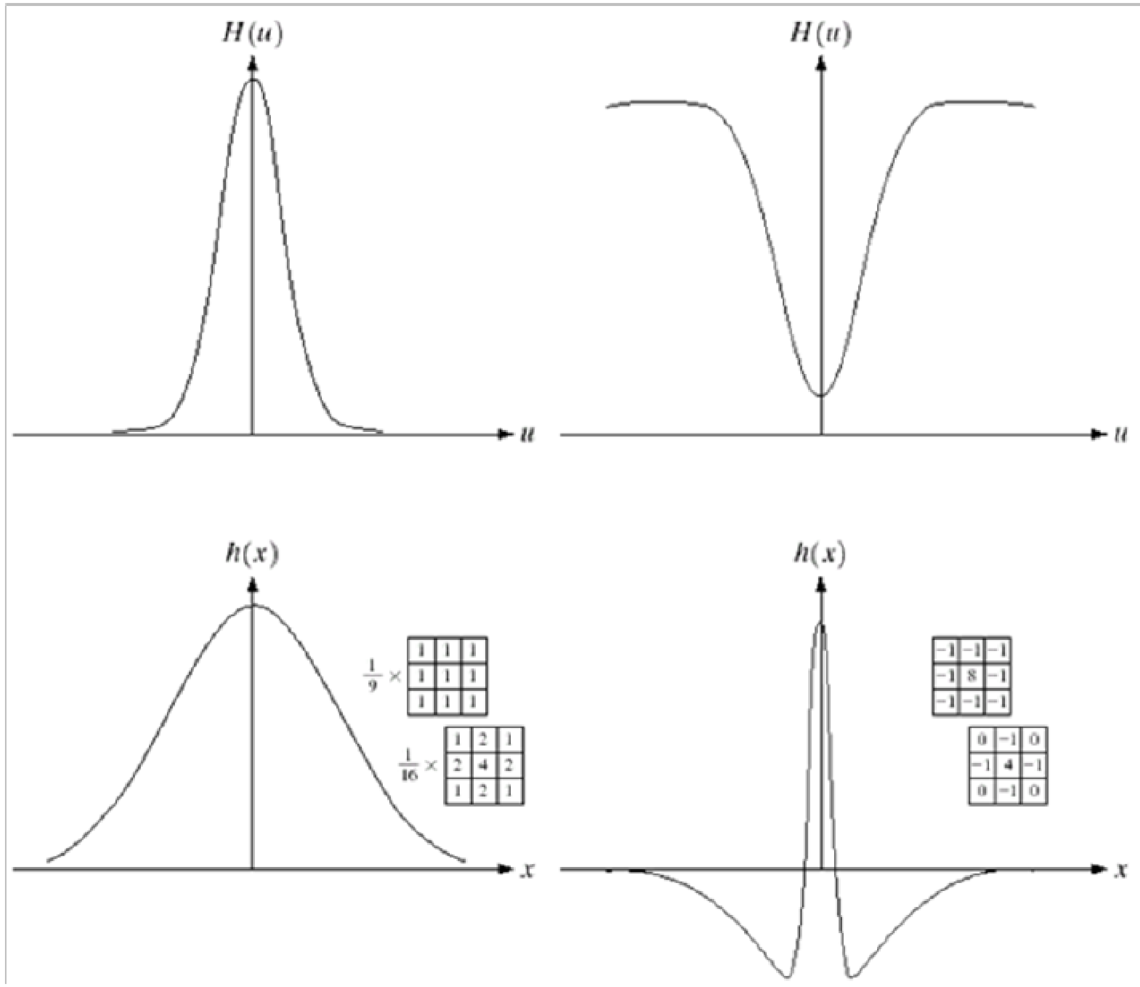



**FIGURE**
(a) Gaussian frequency domain lowpass filter. (b) Gaussian frequency domain highpass filter. (c) Corresponding lowpass spatial filter. (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

Slide credit: Ashish Ghosh

# Smoothing Frequency-Domain Filters

- The basic model for filtering in the frequency domain

$$G(u,v) = H(u,v)F(u,v)$$

where $F(u,v)$: the Fourier transform of the image to be smoothed

$H(u,v)$: a filter transfer function

- Smoothing is fundamentally a lowpass operation in the frequency domain.
- There are several standard forms of lowpass filters (LPF).
  - Ideal lowpass filter
  - Butterworth lowpass filter
  - Gaussian lowpass filter

Slide credit: Ashish Ghosh

# Smoothing Frequency Domain, Ideal Low-pass Filters

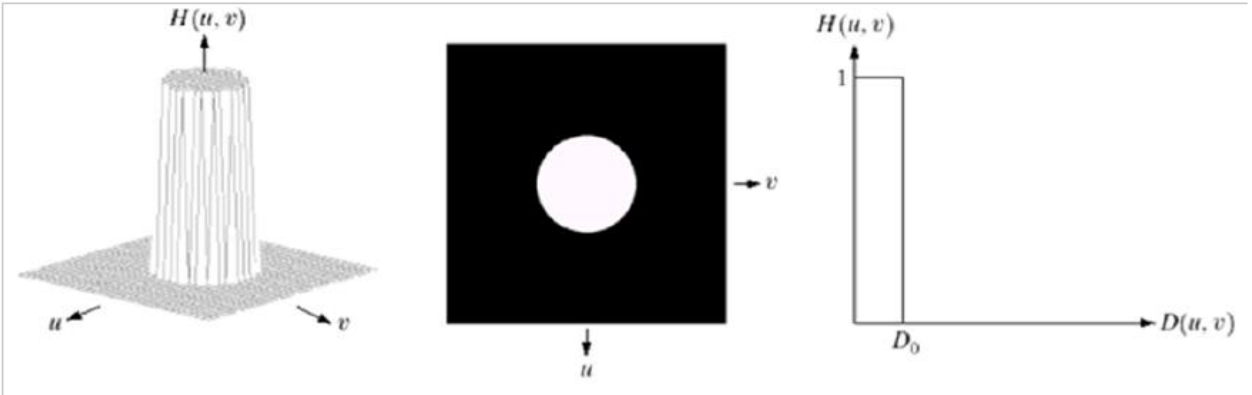


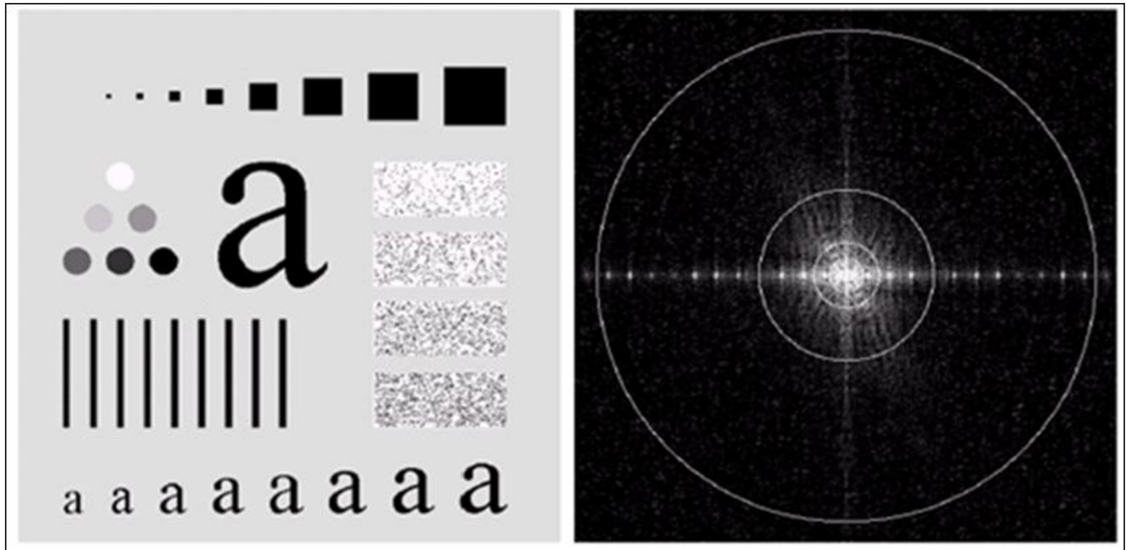

**FIGURE** (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

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

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

Slide credit: Ashish Ghosh

# Smoothing Frequency Domain, Ideal Low-pass Filters




a b

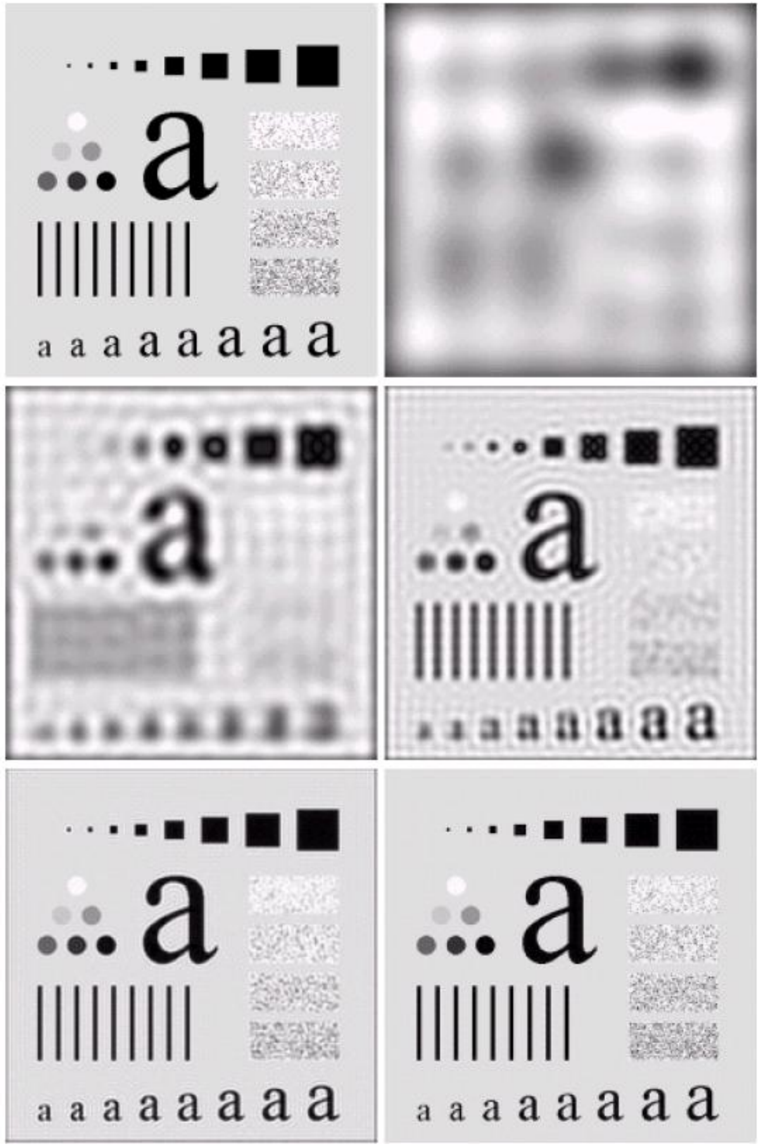
**FIGURE** [redacted] (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

Total Power ⟶ $$f_c = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left|F(u,v)\right|^2$$

The remained percentage power after filtration ⟶ $$\alpha = 100\times\left[\sum_{u}\sum_{v}\left|F(u,v)\right|/f_c\right]$$

Slide credit: Ashish Ghosh

# Smoothing Frequency Domain, Ideal Low-pass Filters



$f_c = 5$
$\alpha = 92\%$

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

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

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

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

Slide credit: Ashish Ghosh

# Smoothing Frequency Domain, Butterworth Low-pass Filters

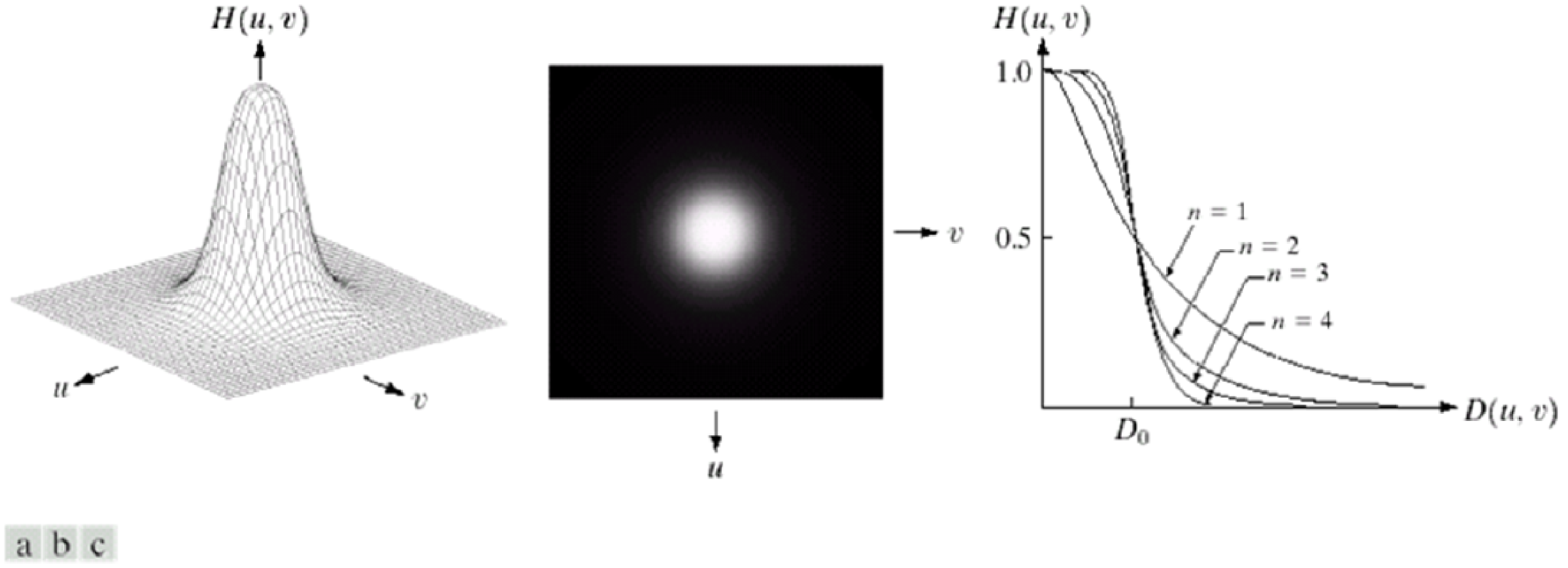


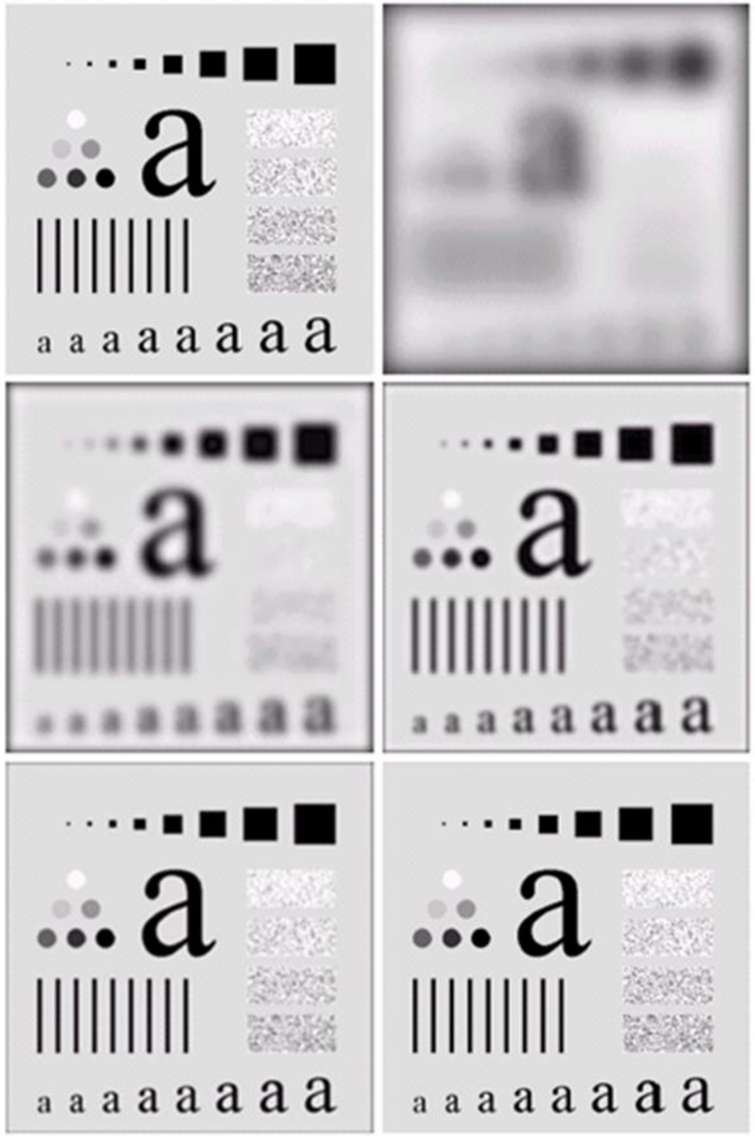

**FIGURE** (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

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

Slide credit: Ashish Ghosh

# Smoothing Frequency Domain, Butterworth Low-pass Filters

Butterworth Low-pass
Filter: *n*=2



Radii= 5

Radii= 15

Radii= 30

Radii= 80

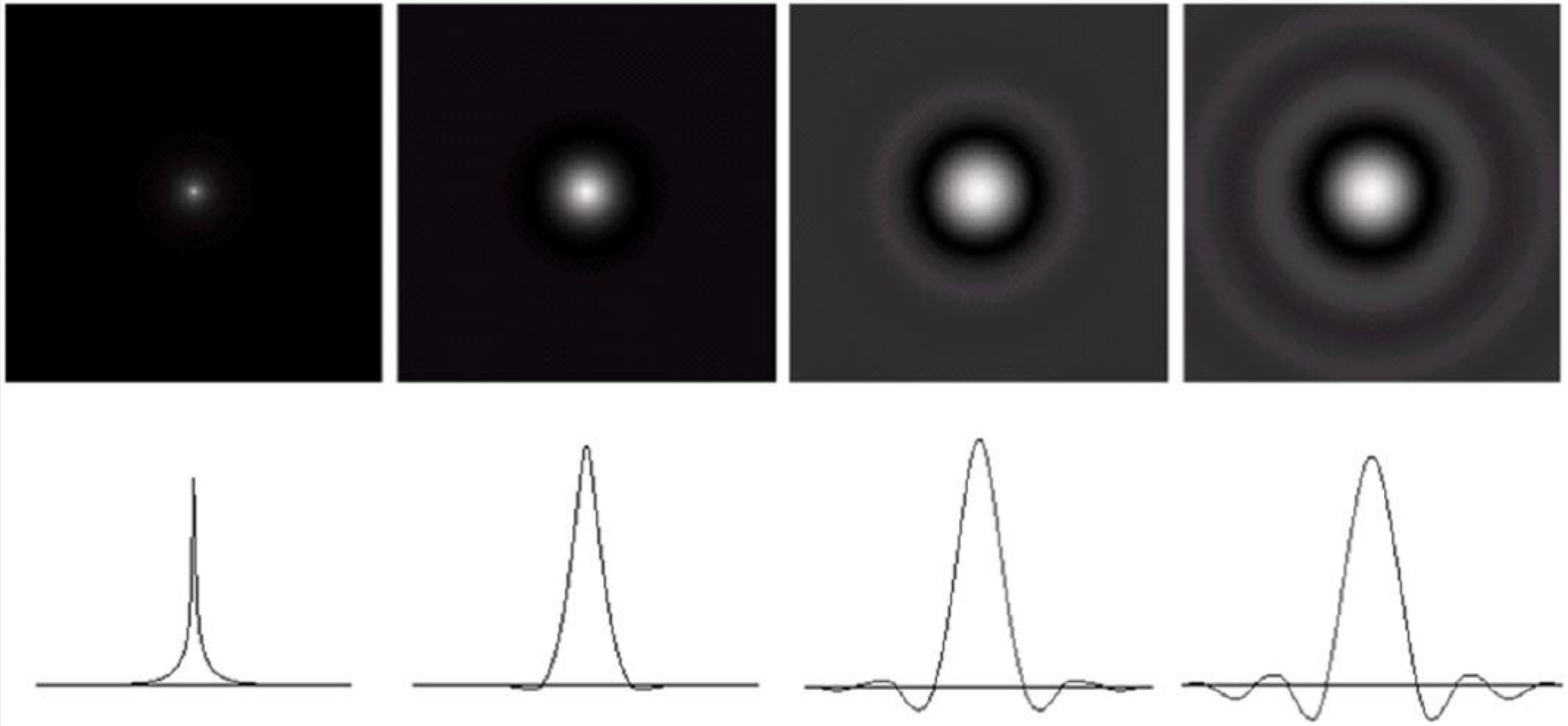
Radii= 230

a b
c d
e f

**FIGURE** (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.

Slide credit: Ashish Ghosh

# Smoothing Frequency Domain, Butterworth Low-pass Filters

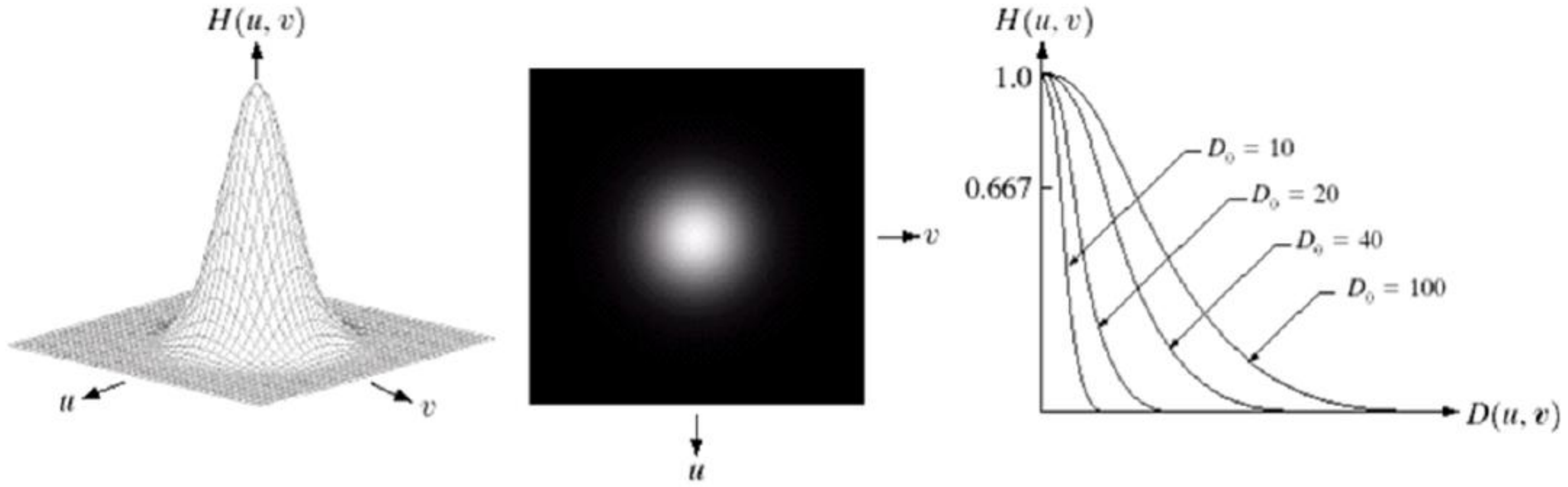


a b c d

**FIGURE** (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.

Slide credit: Ashish Ghosh

# Smoothing Frequency Domain, Gaussian Low-pass Filters




a b c

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

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

Slide credit: Ashish Ghosh

# Smoothing Frequency Domain, Gaussian Low-pass Filters

Gaussian Low-pass

*Radii= 5*

*Radii= 15*

*Radii= 30*

*Radii= 80*

*Radii= 230*

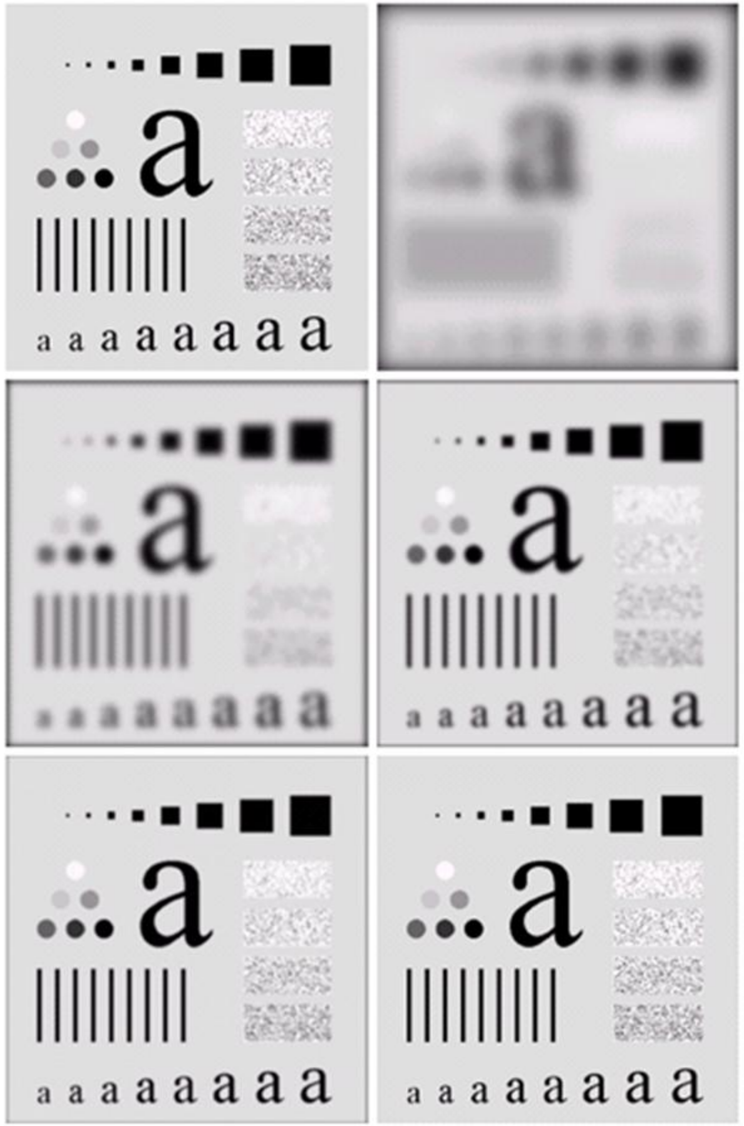


**FIGURE** (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

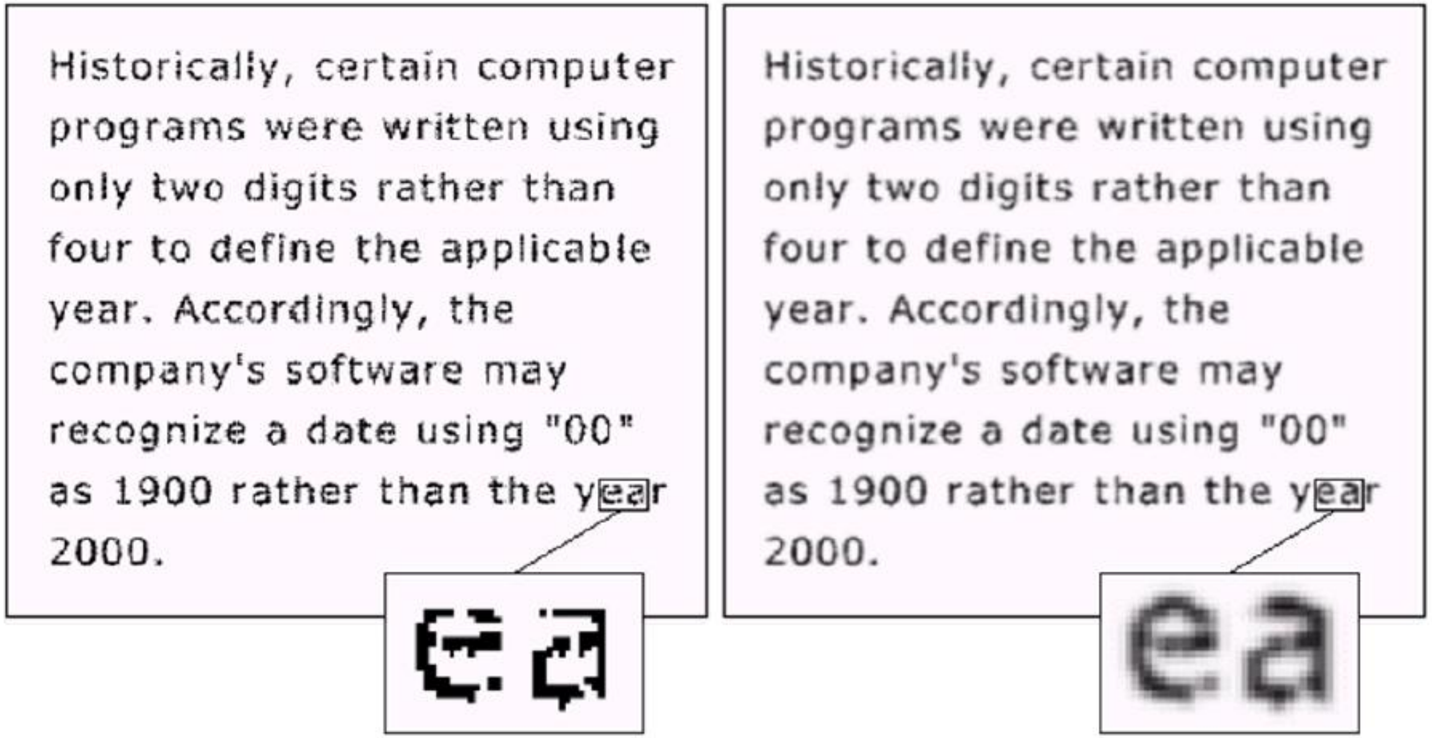
a b
c d
e f

Slide credit: Ashish Ghosh

# 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).




Slide credit: Ashish Ghosh

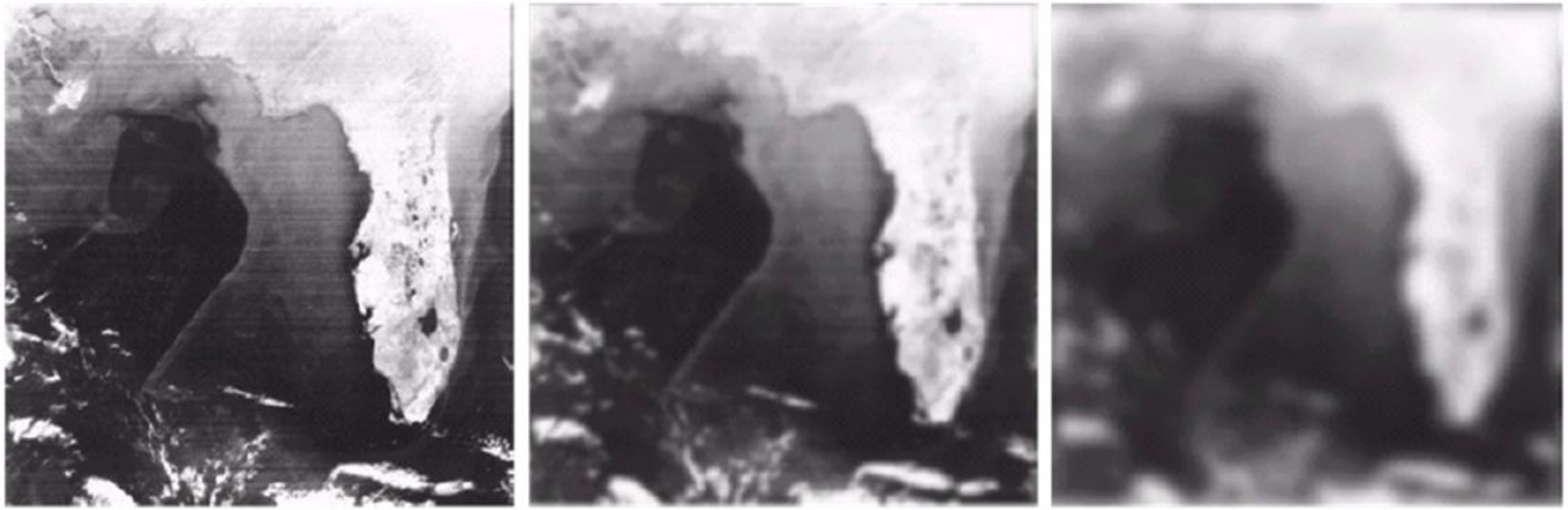
# Smoothing Frequency Domain, Gaussian Low-pass Filters

a b c

**FIGURE** (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).

Slide credit: Ashish Ghosh

# Smoothing Frequency Domain, Gaussian Low-pass Filters



a b c

**FIGURE** ... (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.)

Slide credit: Ashish Ghosh

# Sharpening Frequency Domain Filters: High-pass Filters

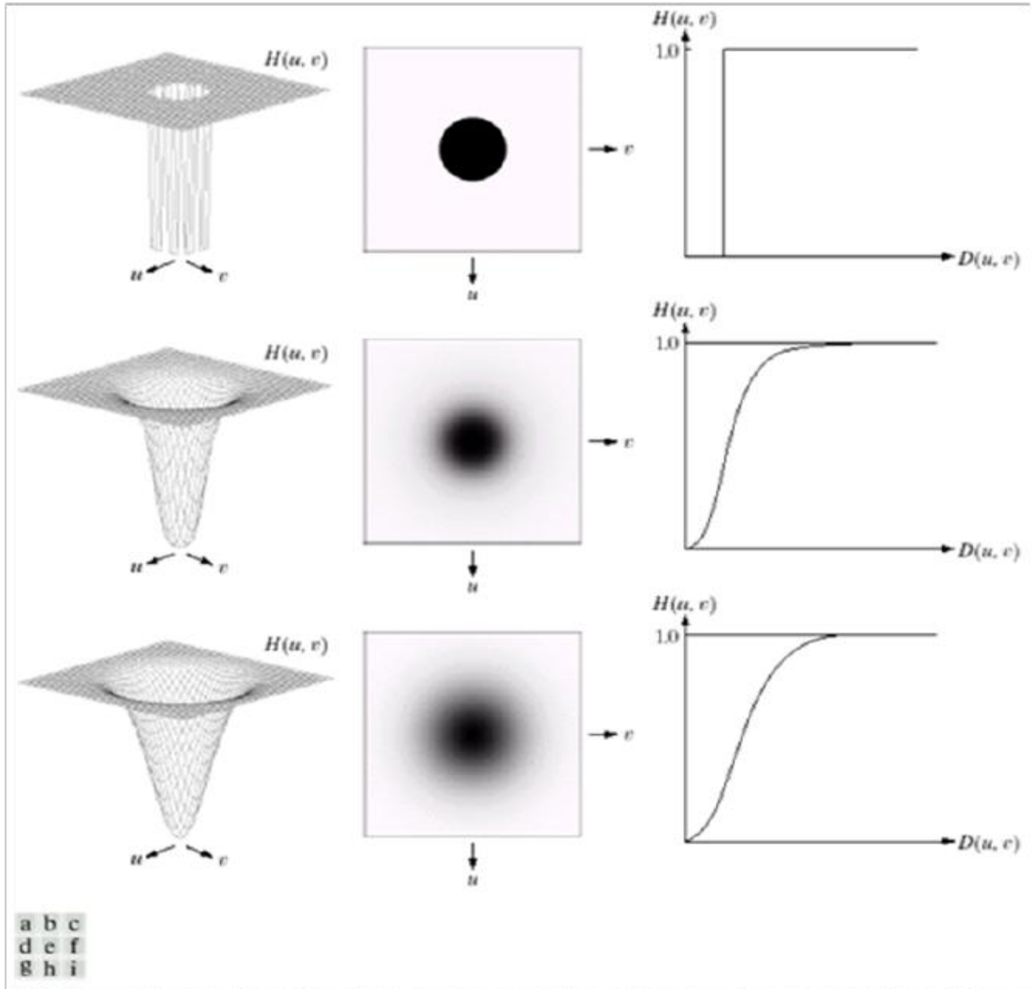
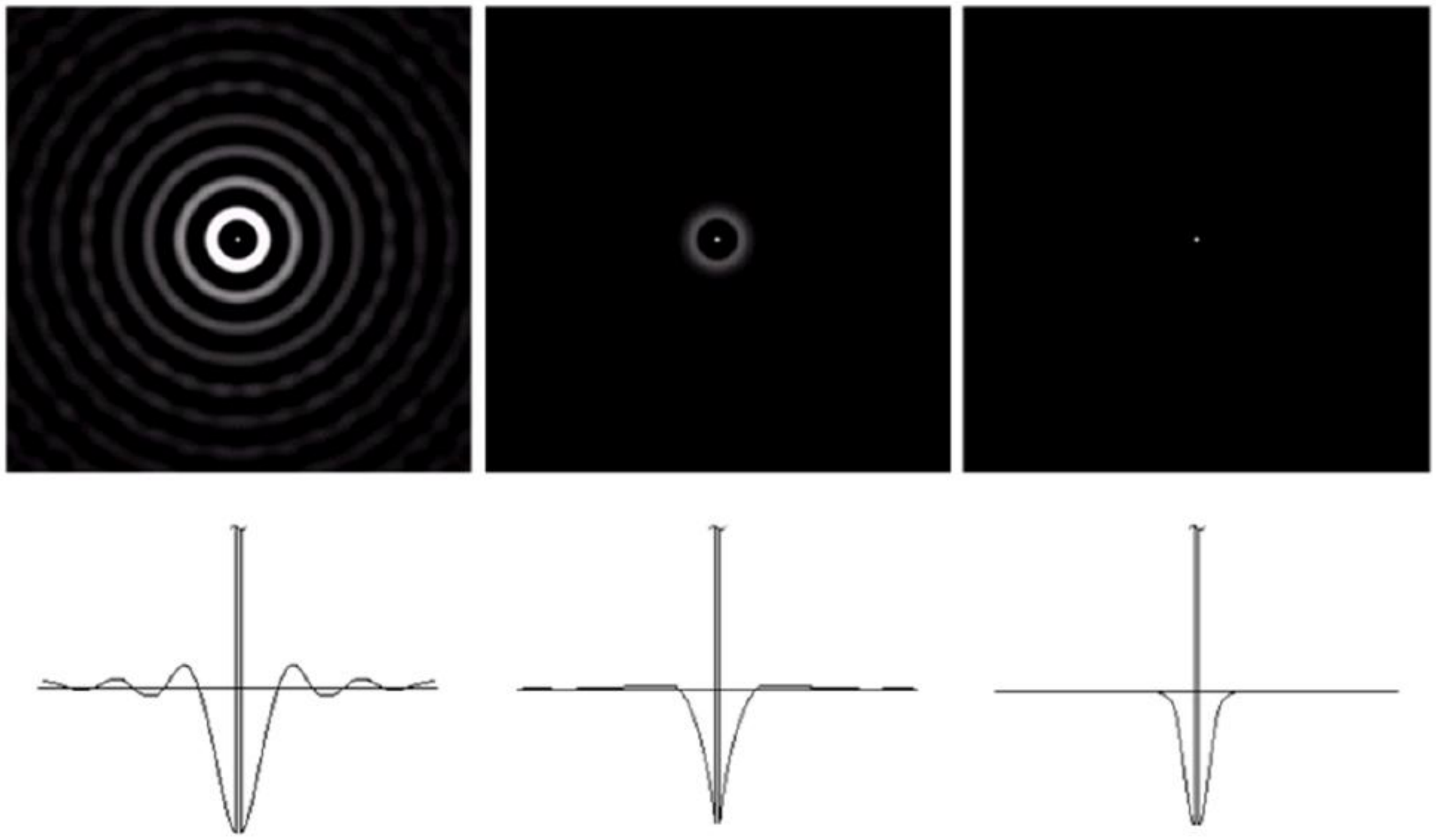



**FIGURE** Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

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

Slide credit: Ashish Ghosh

# Sharpening Frequency Domain Filters: High-pass Filters

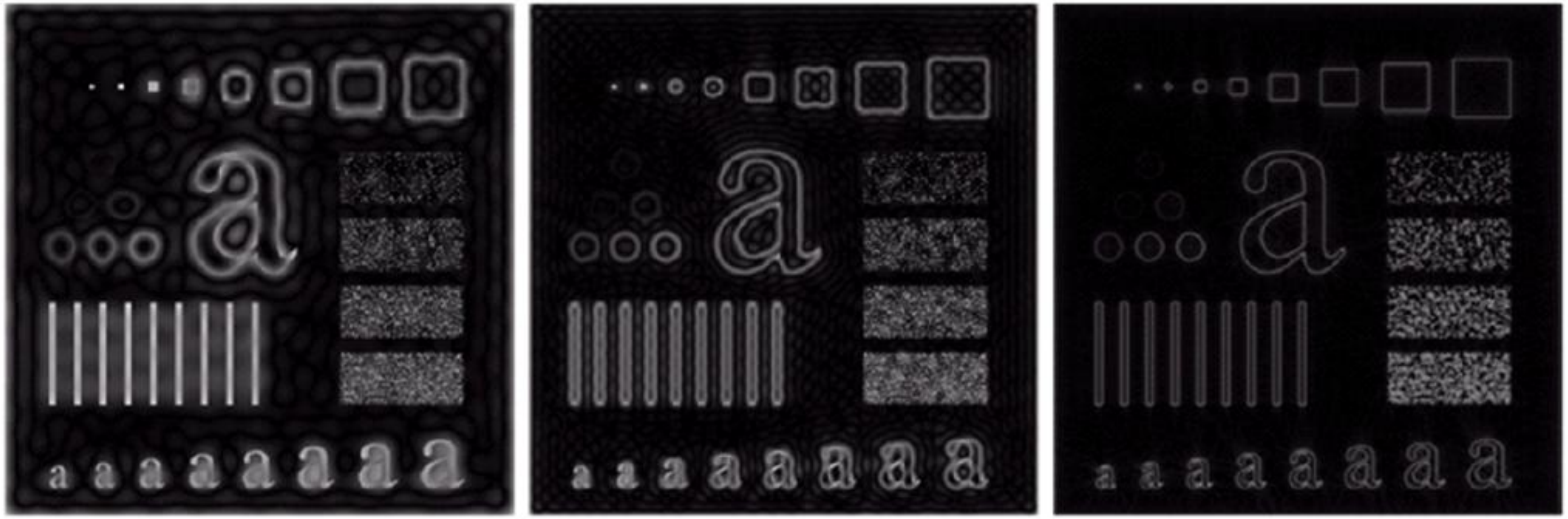


a b c

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

Slide credit: Ashish Ghosh

# Sharpening Frequency Domain Filters: Ideal High-pass Filters



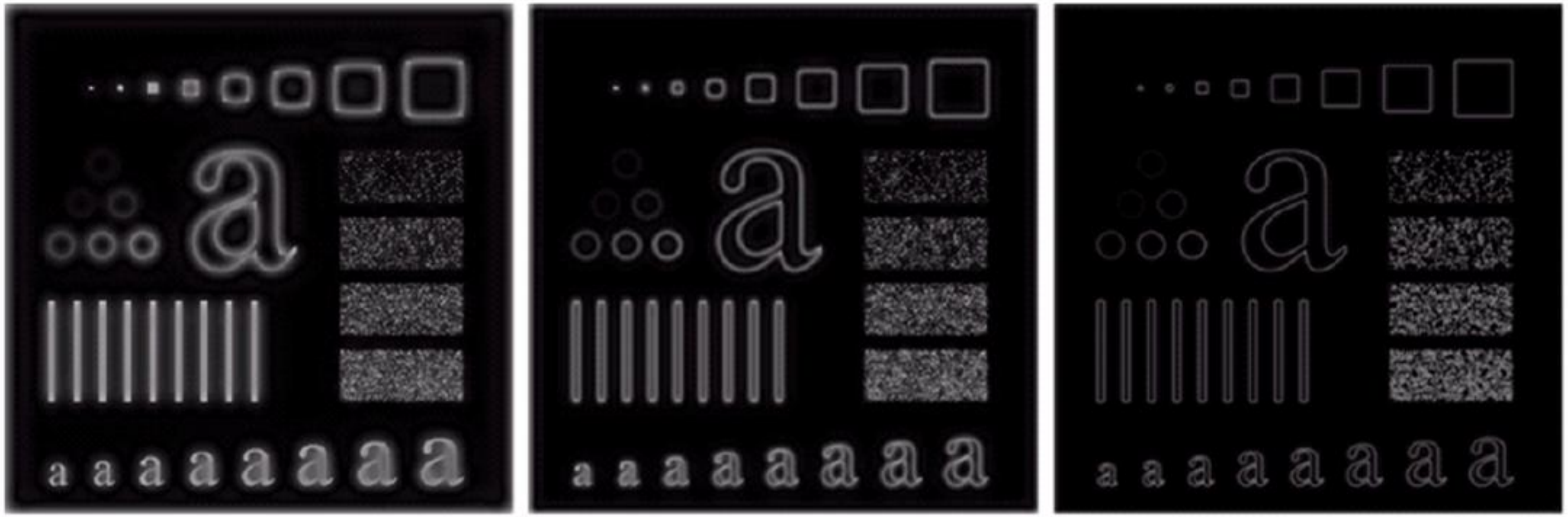
a b c

**FIGURE** 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).

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

Slide credit: Ashish Ghosh

# Sharpening Frequency Domain Filters: Butterworth High-pass Filters



a b c

**FIGURE** 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.

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

Slide credit: Ashish Ghosh

# Sharpening Frequency Domain Filters: Gaussian High-pass Filters


**FIGURE** 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.

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

Slide credit: Ashish Ghosh