Enhancement Using Arithmetic/Logic Operations

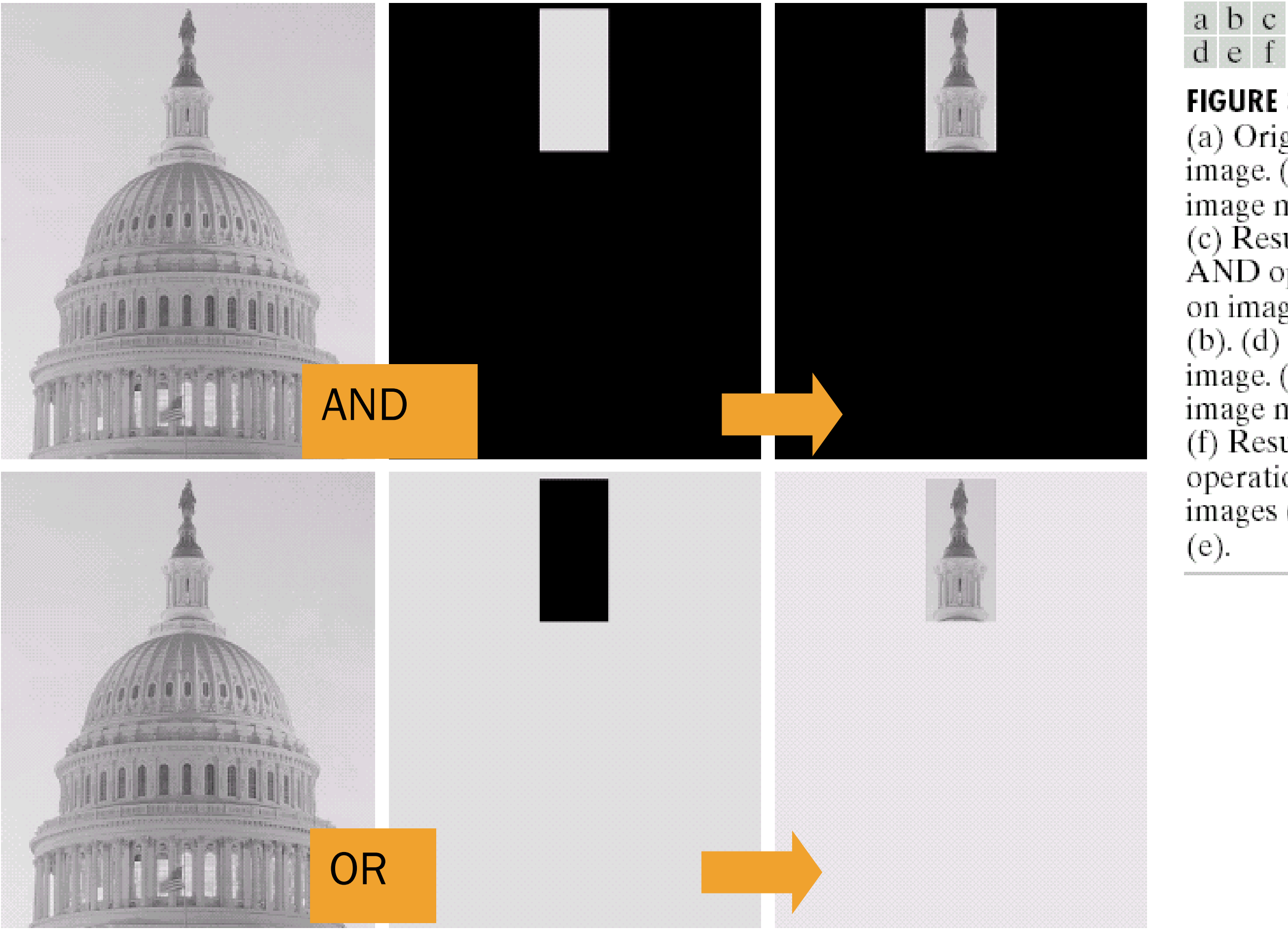


FIGURE 3.27 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

FILTERING TO REMOVE NOISE

We can use spatial filters of different kinds to remove different kinds of noise

The *arithmetic mean* filter is a very simple one and is calculated as follows:

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

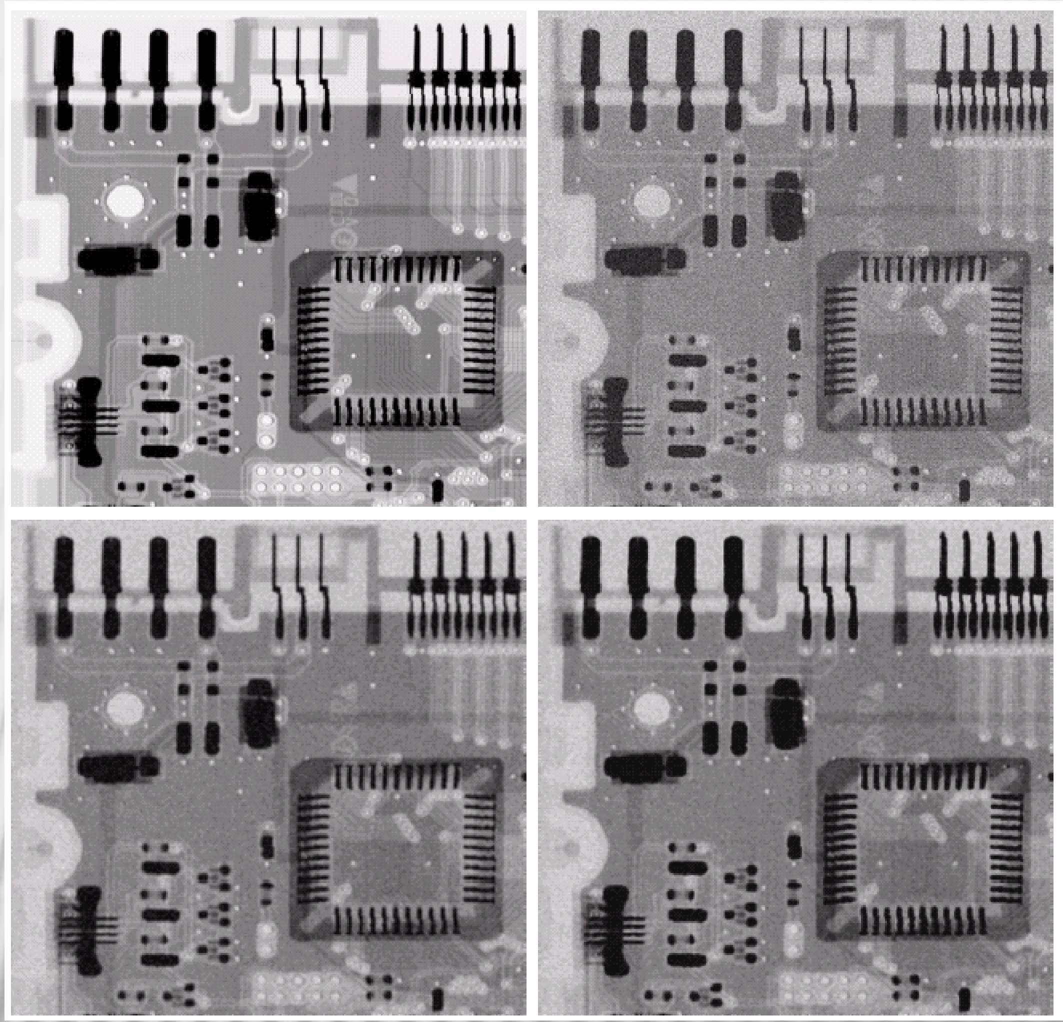
This is implemented as the simple smoothing filter

Blurs the image to remove noise

NOISE REMOVAL EXAMPLES

Original Image

Image Corrupted By Gaussian Noise



After A 3*3 Arithmetic Mean Filter

After A 3*3 Geometric Mean Filter

SHARPENING SPATIAL FILTERS

The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as an natural effect of a particular method of image acquisition.

INTRODUCTION

- The image blurring is accomplished in the spatial domain by pixel averaging in a neighborhood.
- Since averaging is analogous to integration.
- Sharpening could be accomplished by *spatial differentiation*.

FOUNDATION

- We are interested in the *behavior* of these derivatives in areas of constant gray level (flat segments), at the onset and end of discontinuities (step and ramp discontinuities), and along gray-level ramps.

- These types of discontinuities can be noise points, lines, and edges.

PROPERTIES OF A FIRST DERIVATIVE

- Sharpening filters that are based on first and second order derivatives respectively.
- Must be ***zero*** in flat segments
- Must be ***nonzero*** at the onset of a gray-level step or ramp; and
- Must be ***nonzero*** along ramps.

PROPERTIES OF A SECOND DERIVATIVE

- Must be ***zero*** in flat areas;
- Must be ***nonzero*** at the onset and end of a gray-level step or ramp;
- Must be ***zero*** along ramps of constant slope

DEFINITION OF THE 1ST-ORDER DERIVATIVE

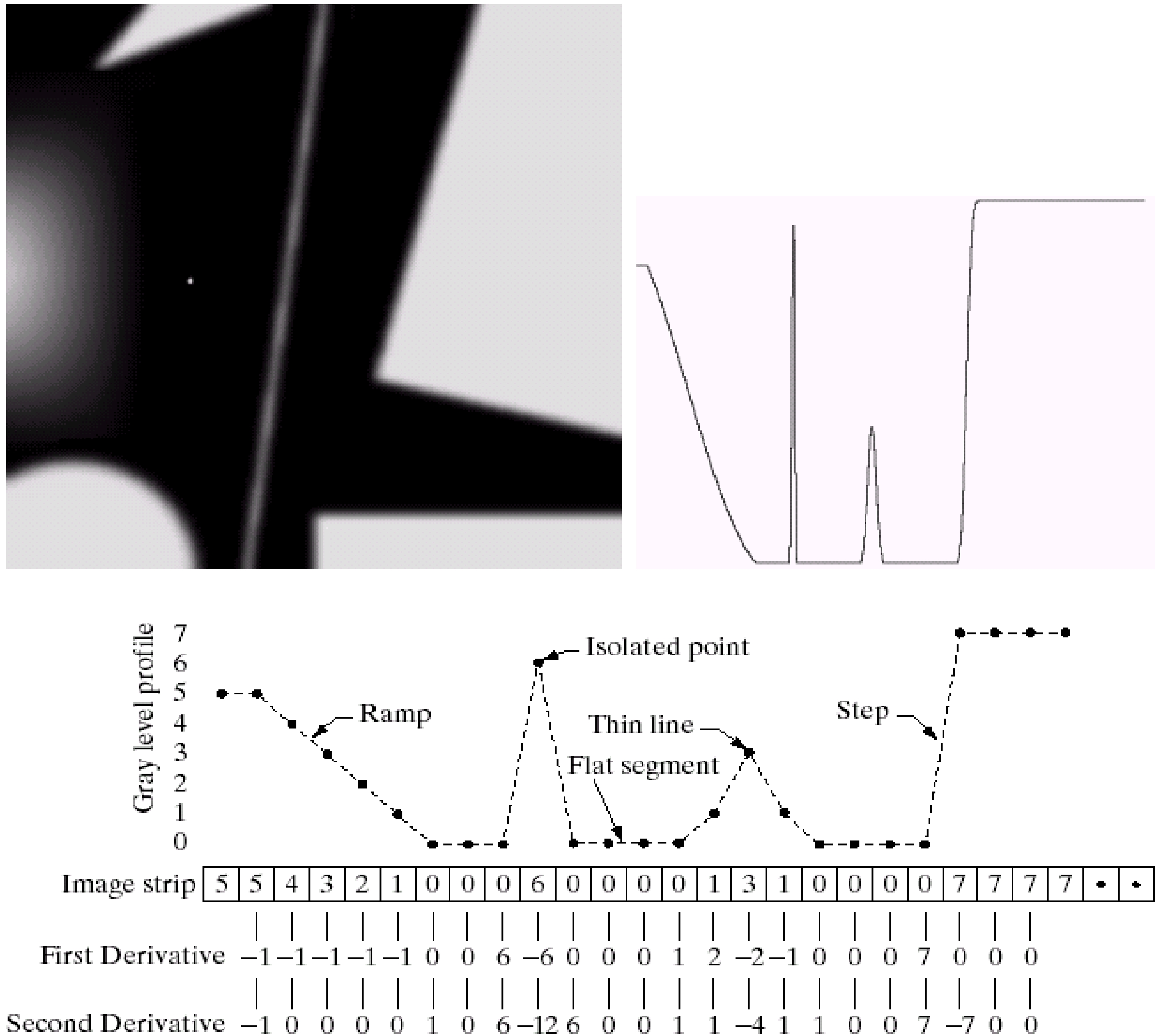
- A basic definition of the first-order derivative of a one-dimensional function f(x) is :

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

DEFINITION OF THE 2ND-ORDER DERIVATIVE

- We define a second-order derivative as the difference

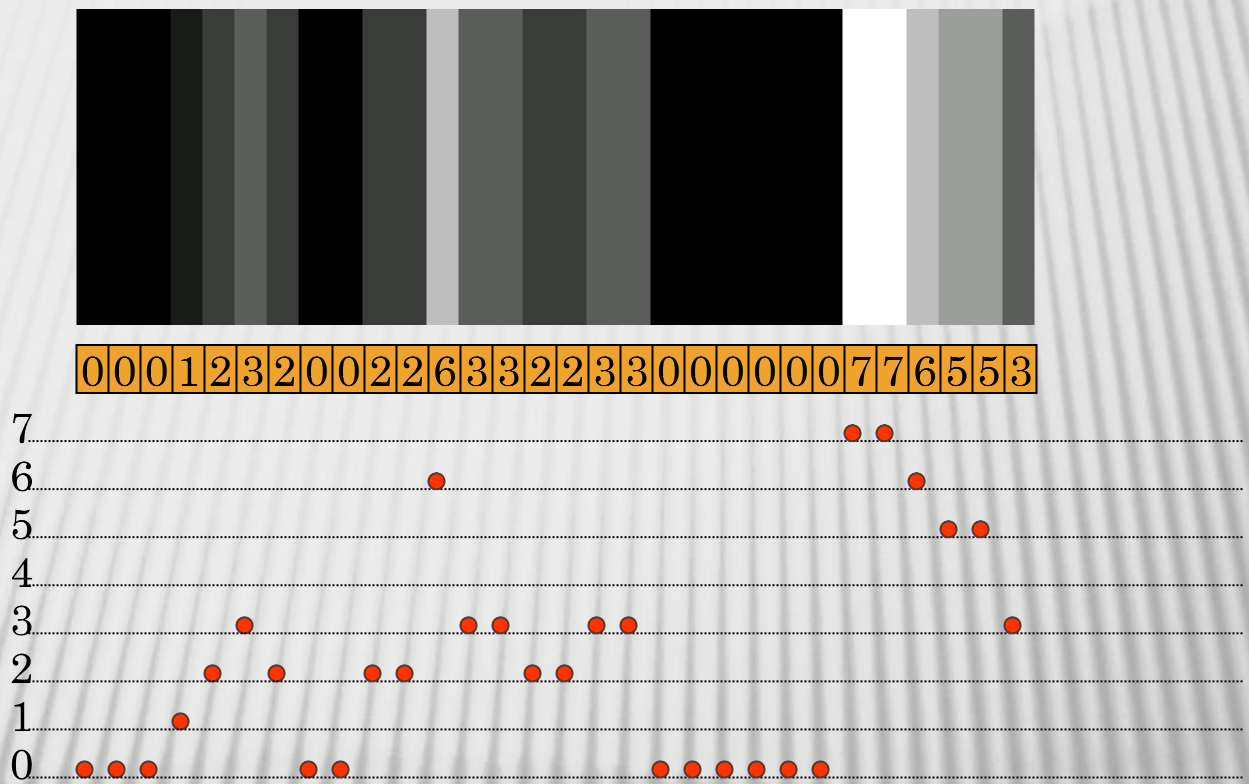
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x).$$



a b
c

FIGURE 3.38
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point.
(c) Simplified profile (the points are joined by dashed lines to simplify interpretation).

GRAY-LEVEL PROFILE



DERIVATIVE OF IMAGE PROFILE

0 0 0 1 2 3 2 0 0 2 2 6 3 3 2 2 3 3 0 0 0 0 0 0 7 7 6 5 5 3

first 0 0 1 1 1 -1 -2 0 2 0 4 -3 0 -1 0 1 0 -3 0 0 0 0 0 -7 0 -1 -1 0 -2

second 0 -1 0 0 -2 -1 2 2 -2 4 -7 3 -1 1 1 -1 -3 3 0 0 0 0 -7 7 -1 0 1 -2

ANALYZE

- The 1st-order derivative is nonzero along the entire ramp, while the 2nd-order derivative is nonzero only at the onset and end of the ramp.
- The response at and around the point is much stronger for the 2nd- than for the 1st-order derivative

1st makes thick edge and 2nd make thin edge

THE LAPLACIAN (2ND ORDER DERIVATIVE)

- Shown by Rosenfeld and Kak[1982] that the simplest isotropic derivative operator is the Laplacian is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

DISCRETE FORM OF DERIVATIVE

f(x-1,y)	*f(x,y)*	*f(x+1,y)*

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

f(x,y-1)
f(x,y)
f(x,y+1)

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

2-DIMENTIONAL LAPLACIAN

- The digital implementation of the 2-Dimensional Laplacian is obtained by summing 2 components

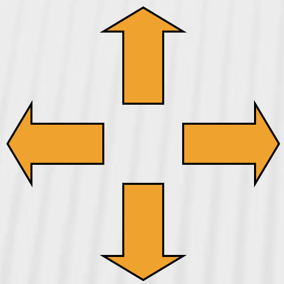
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial x^2}$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

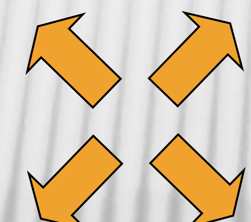
	1	
1	-4	1
	1	

LAPLACIAN

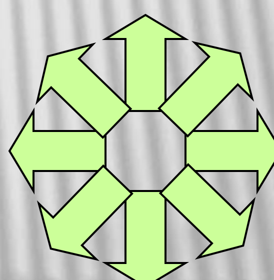
0	1	0
1	-4	1
0	1	0



1	0	1
0	-4	0
1	0	1

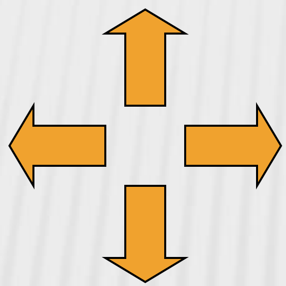


1	1	1
1	-8	1
1	1	1

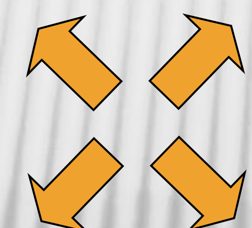


LAPLACIAN

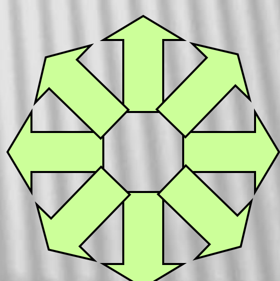
0	-1	0
-1	4	-1
0	-1	0



-1	0	-1
0	4	0
-1	0	-1



-1	-1	-1
-1	8	-1
-1	-1	-1



SECOND DERIVATIVE IN 2D: LAPLACIAN (CONT'D)

$$\frac{\partial^2 f}{\partial x^2} = f(i, j+1) - 2f(i, j) + f(i, j-1)$$

$$\frac{\partial^2 f}{\partial y^2} = f(i+1, j) - 2f(i, j) + f(i-1, j)$$

$$\nabla^2 f = -4f(i, j) + f(i, j+1) + f(i, j-1) + f(i+1, j) + f(i-1, j)$$

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 1 & -2 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 0 & -2 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

VARIATIONS OF LAPLACIAN

0.5	0.0	0.5
1.0	-4.0	1.0
0.5	0.0	0.5

+

0.5	1.0	0.5
0.0	-4.0	0.0
0.5	1.0	0.5

=

1	1	1
1	-8	1
1	1	1

1	-2	1
1	-2	1
1	-2	1

+

1	1	1
-2	-2	-2
1	1	1

=

2	-1	2
-1	-4	-1
2	-1	2

PROPERTIES OF LAPLACIAN

- It is an isotropic operator.
- It is cheaper to implement than the gradient (i.e., one mask only).
- It does not provide information about edge direction.
- It is more sensitive to noise (i.e., differentiates twice).

IMPLEMENTATION

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \textit{If the center coefficient is negative} \\ f(x,y)+\nabla^2 f(x,y) & \textit{If the center coefficient is positive} \end{cases}$$

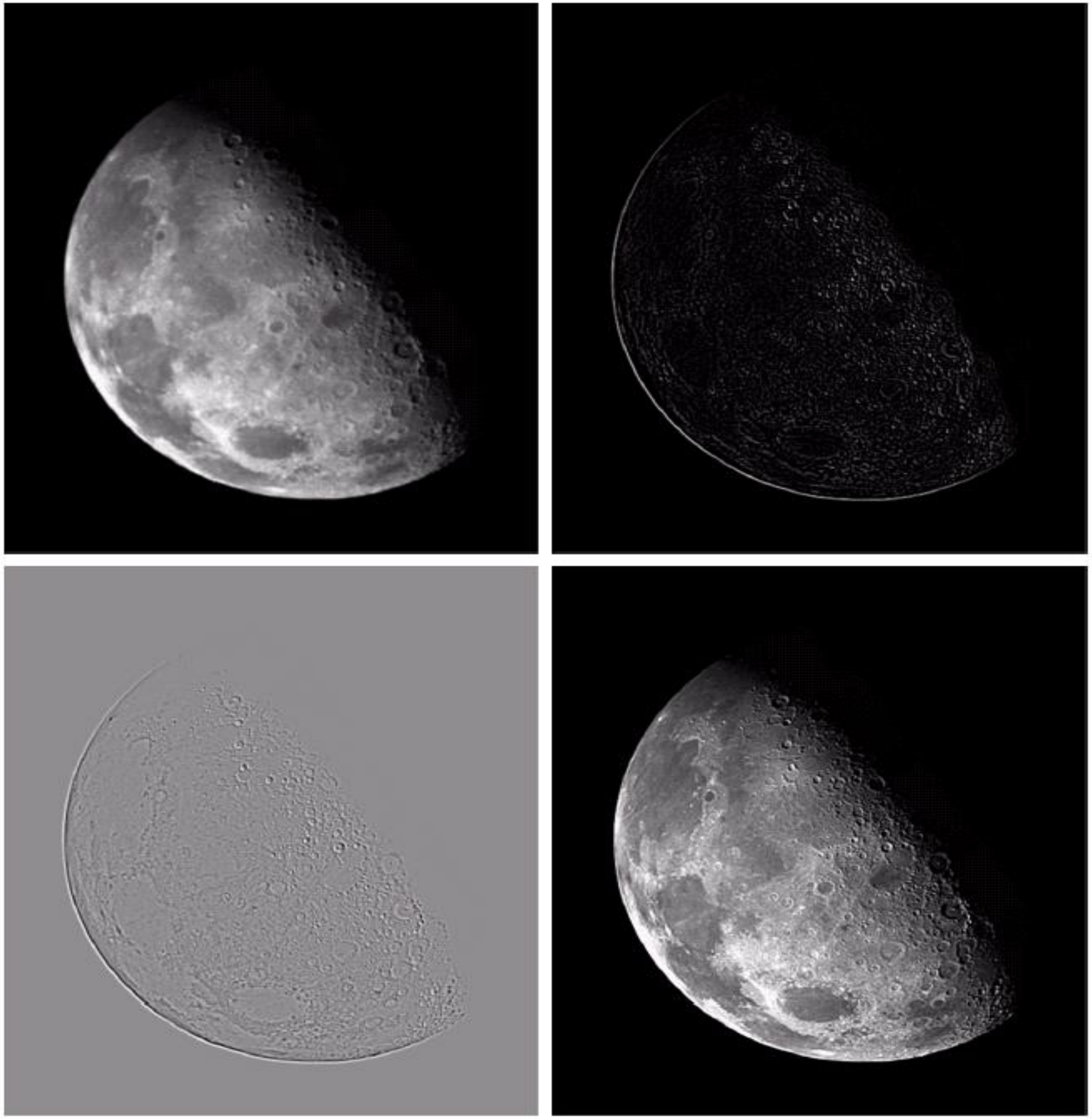
Where f(x,y) is the original image

$\nabla^2 f(x,y)$ is Laplacian filtered image

g(x,y) is the sharpen image

IMPLEMENTATION

-1	-1	-1
-1	8	-1
-1	-1	-1



a b
c d

FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)

UNSHARP MASKING

- A process to sharpen images consists of subtracting a blurred version of an image from the image itself. This process, called *unsharp masking*, is expressed as

$$f_s(x,y) = f(x,y) - \bar{f}(x,y)$$

Where $f_s(x,y)$ denotes the sharpened image obtained by unsharp masking, and $f(x,y)$ is a blurred version of $\bar{f}(x,y)$

Combining Spatial Enhancement Methods

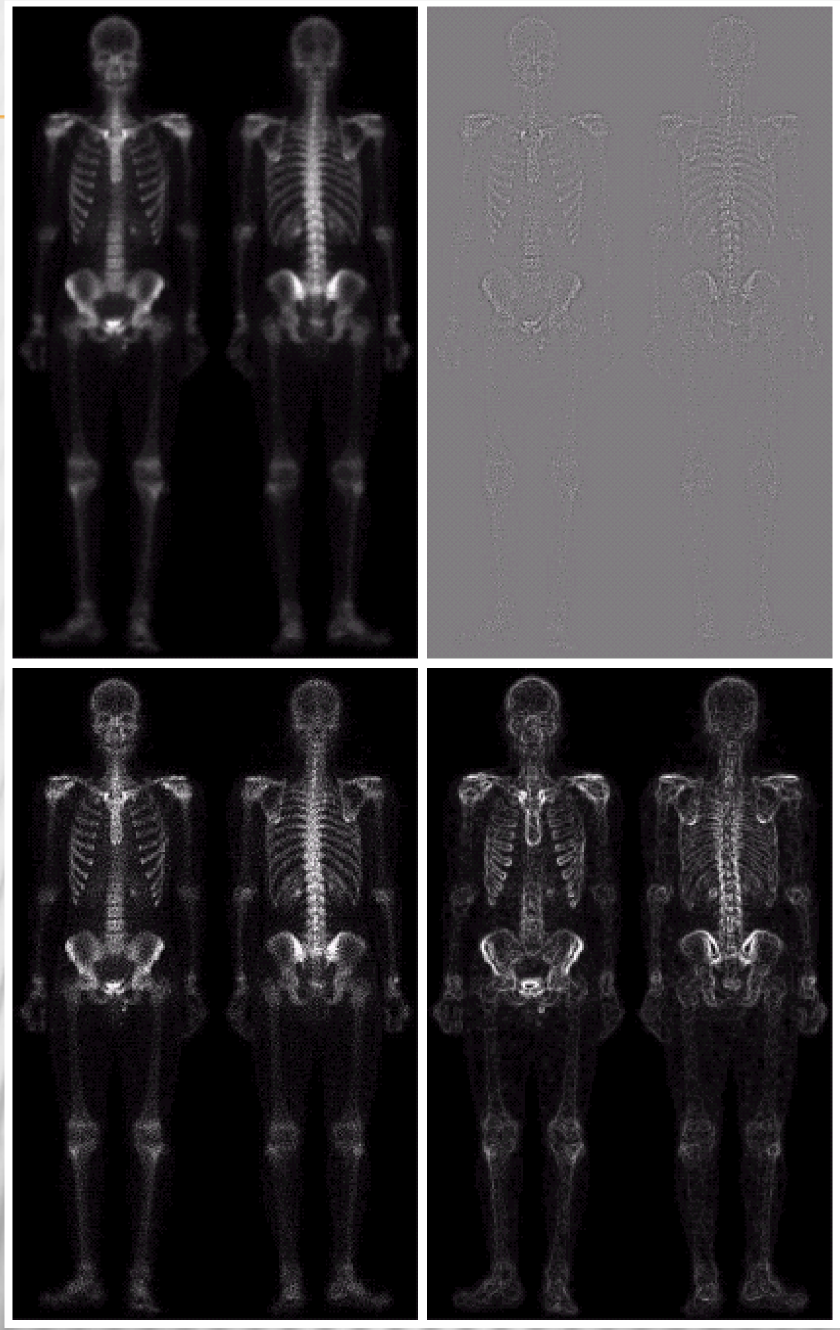
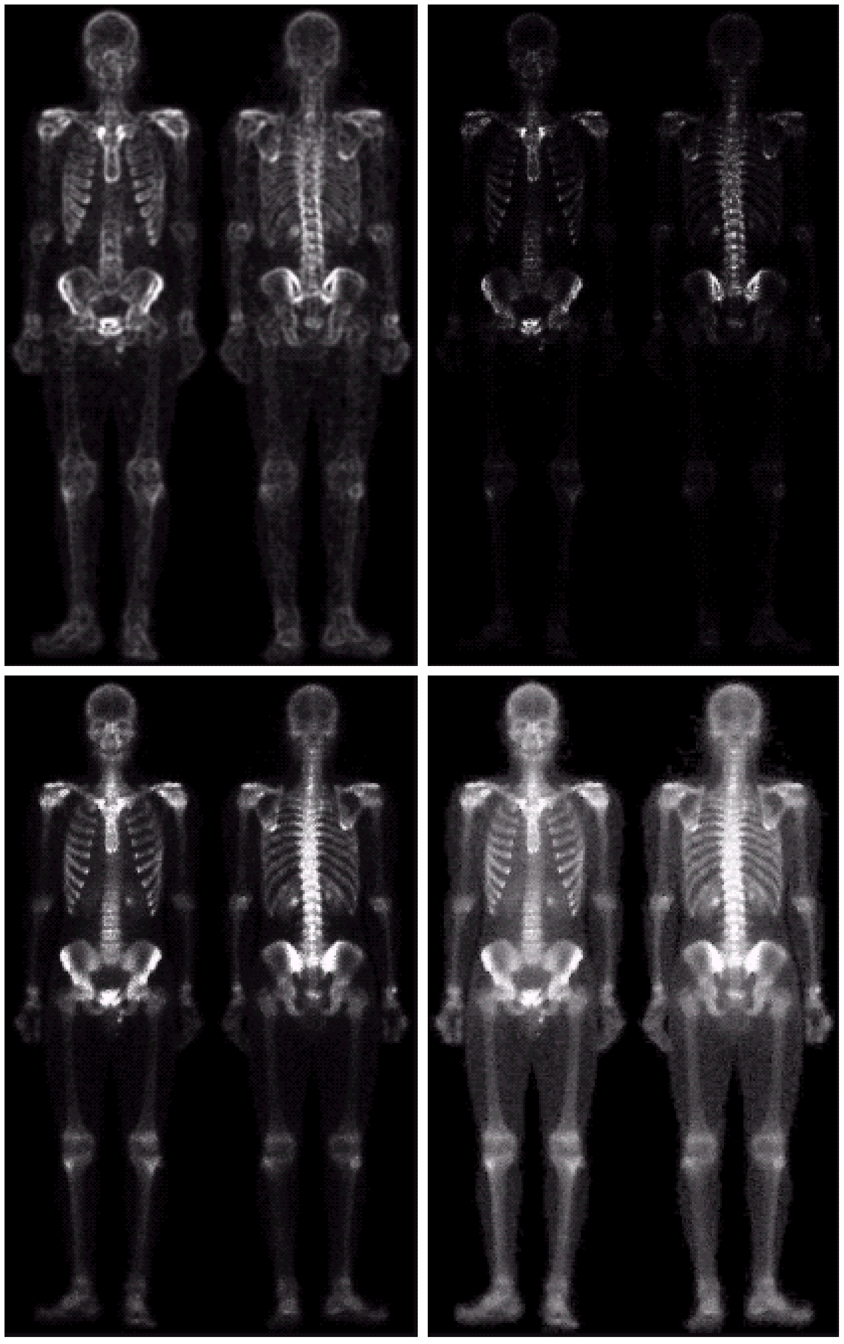


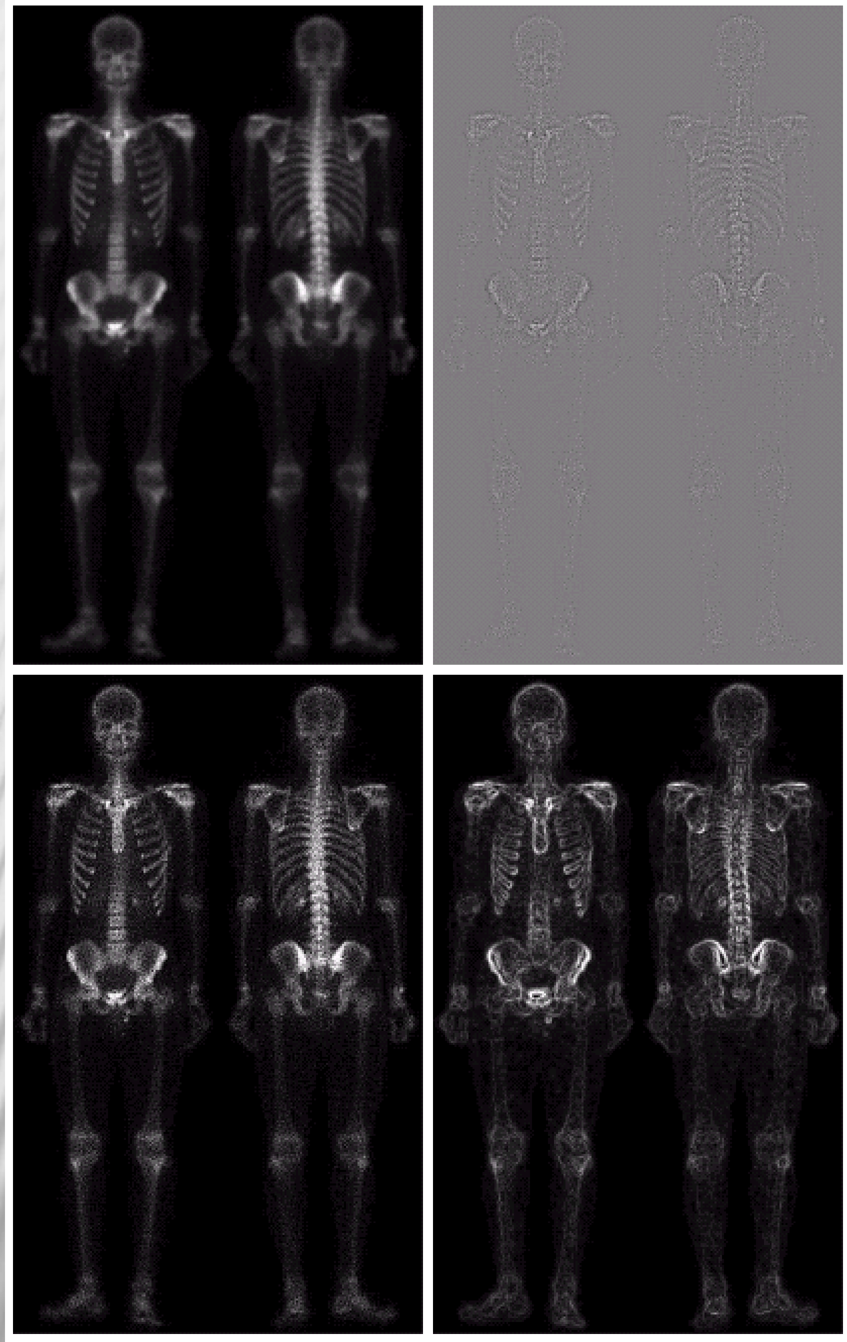
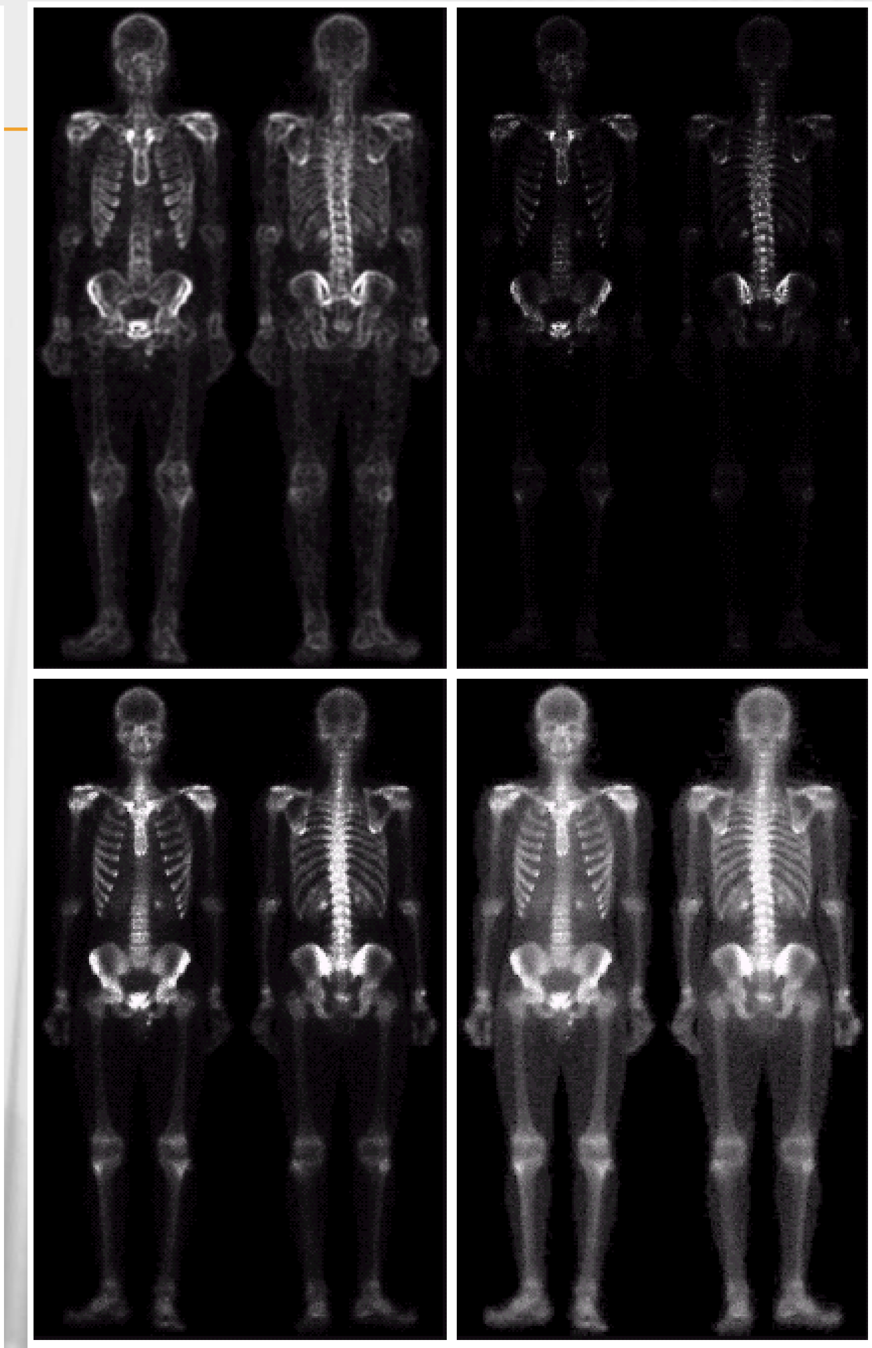
FIGURE 3.46
(a) Image of whole body bone scan.
(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).

Combining Spatial Enhancement Methods



e	f
g	h

FIGURE 3.46 *(Continued)* (e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

- EXTRA

Use of First Derivatives for Enhancement using Gradient

- Development of the Gradient method
  - The gradient of function *f* at coordinates (*x*,*y*) is defined as the two-dimensional column vector:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

  - The magnitude of this vector is given by

$$\nabla f = \mathrm{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{\frac{1}{2}} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{\frac{1}{2}}$$

$$\nabla f \approx \left|G_x\right| + \left|G_y\right|$$

Use of First Derivatives for Enhancement
The Gradient

a
b c
d e

FIGURE 3.44 A 3 × 3 region of an image (the z's are gray-level values) and masks used to compute the gradient at point labeled z_5. All masks coefficients sum to zero, as expected of a derivative operator.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

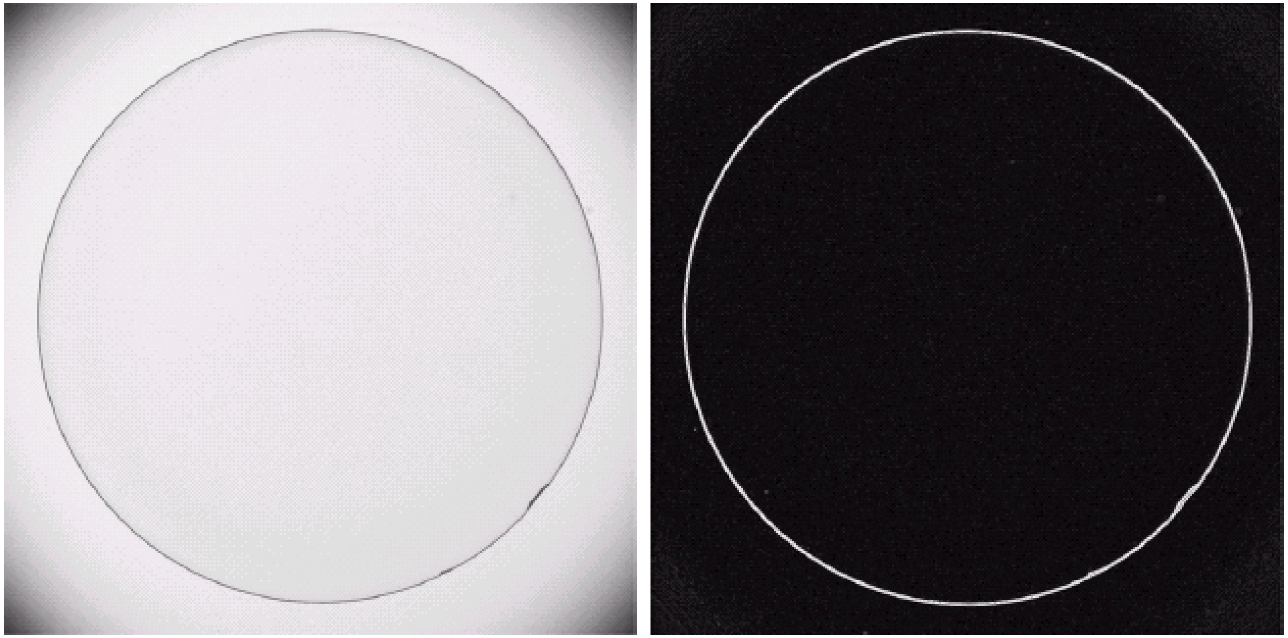
Roberts cross-gradient operators

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel operators

Use of First Derivatives for Enhancement
The Gradient: Using Sobel Operators



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

ROBERT'S METHOD

- The Roberts Edge filter is use to detect edges based applying a horizontal and vertical filter in sequence.

- Both filters are applied to the image and summed to form the final result.

ROBERT'S METHOD

- The two filters are basic convolution filters of the form:

Horizontal Filter

-1	0
0	1

Vertical Filter

0	-1
1	0

Example – Roberts operator/ filter

SOBEL'S METHOD

- The Sobel Edge filter is use to detect edges based applying a horizontal and verticle filter in sequence.
- Both filters are applied to the image and summed to form the final result.

SOBEL'S METHOD

- Using this equation

$$\nabla f \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

The two filters are basic convolution filters of the form:

Horizontal Filter

-1	-2	-1
0	0	0
1	2	1

Verticle Filter

-1	0	1
-2	0	2
-1	0	1

- For example, if a 3x3 window is used as such

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

- Z5 pixel value is calculated based on previous equation

$$Z5=|(Z7+2Z8+Z9)-(Z1+2Z2+Z3)|+|(Z3+2Z6+Z9)-(Z1+2Z4+Z7)|$$


- FREQUENCY DOMAIN

Basic Filtering in the Frequency Domain

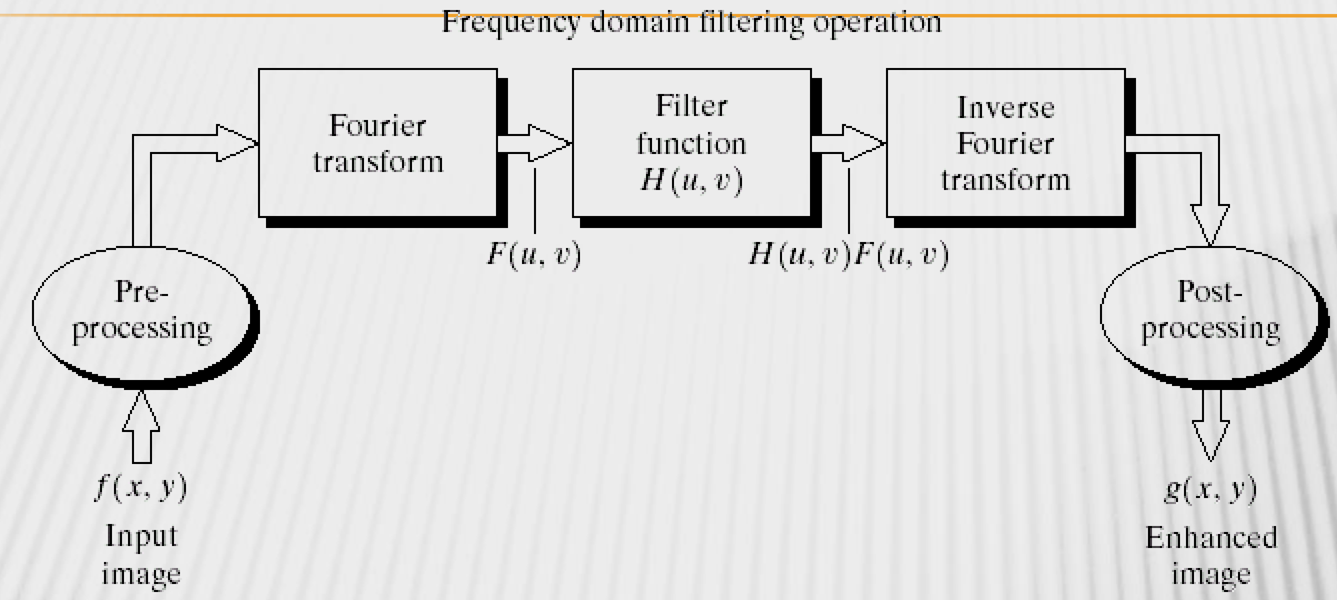
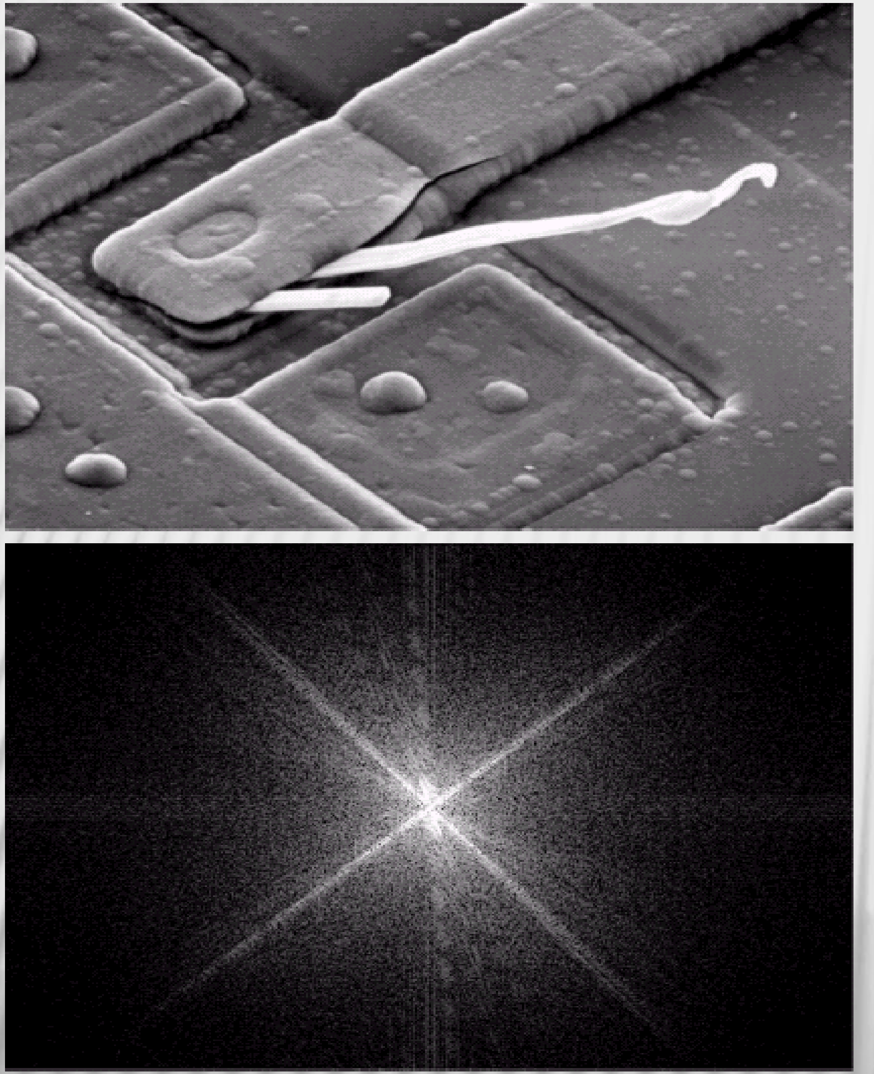


FIGURE 4.5 Basic steps for filtering in the frequency domain.

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

Filtering out the DC Frequency Component

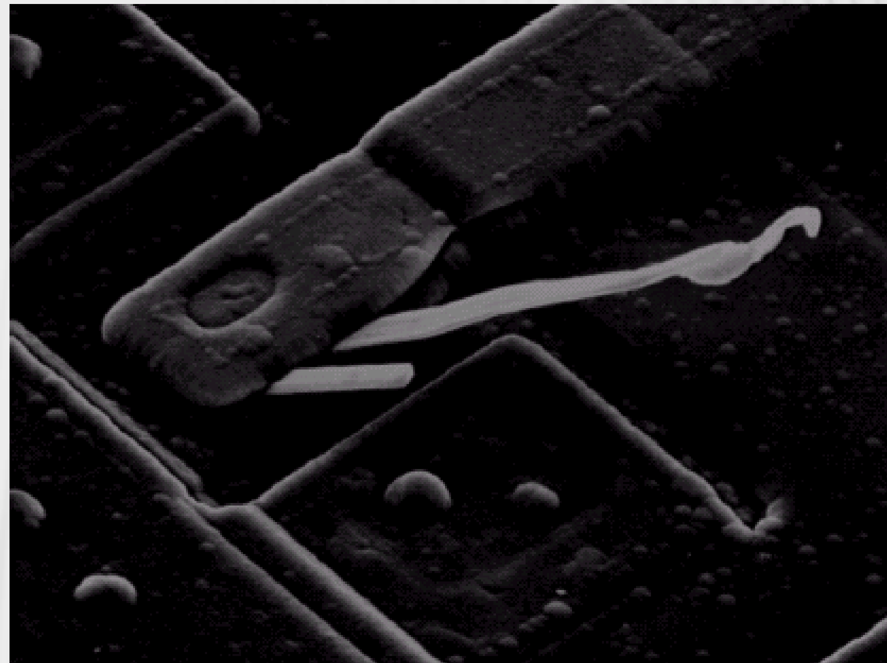


a
b

FIGURE 4.4
(a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

Filtering out the DC Frequency Component

FIGURE 4.6
Result of filtering the image in Fig. 4.4(a) with a notch filter that set to 0 the $F(0, 0)$ term in the Fourier transform.



Notch Filter

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

Low-pass and High-pass Filters

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

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

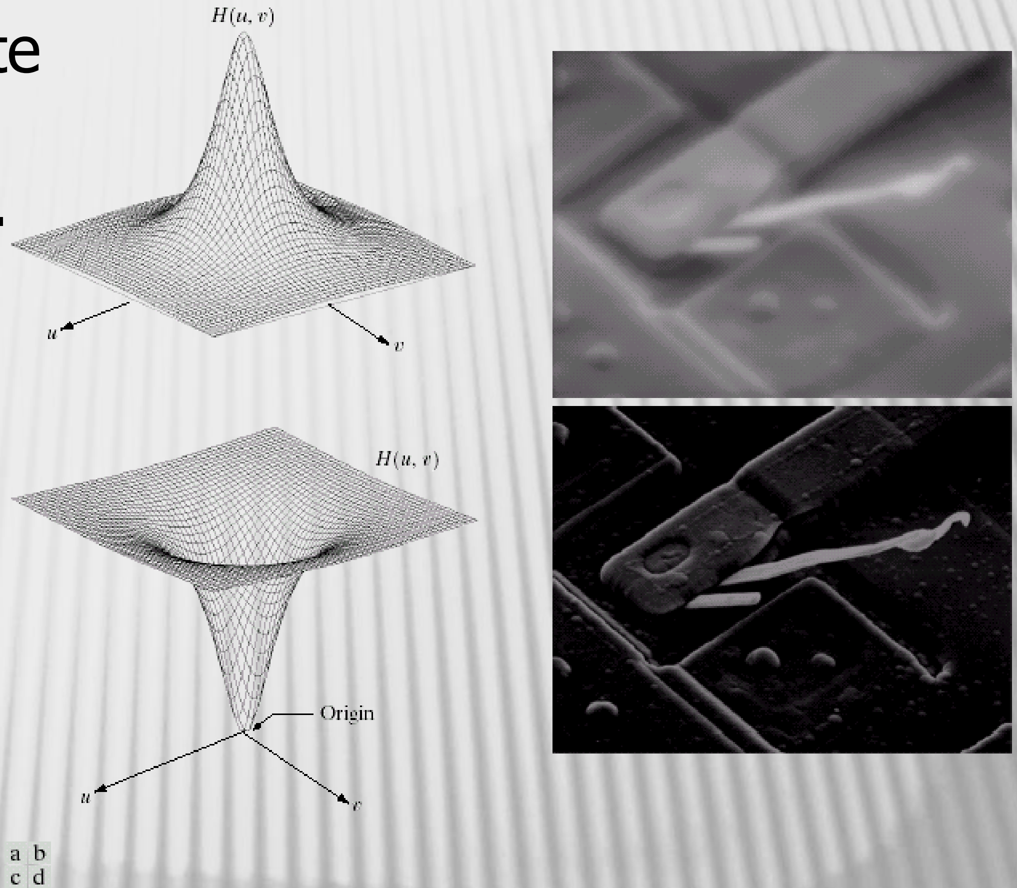
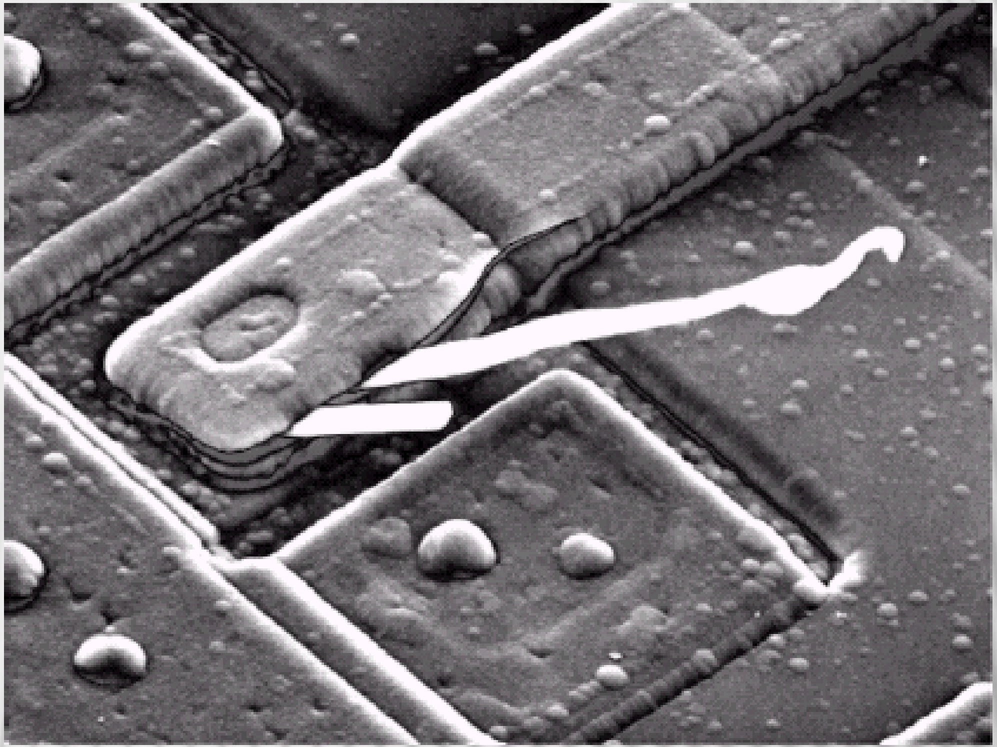


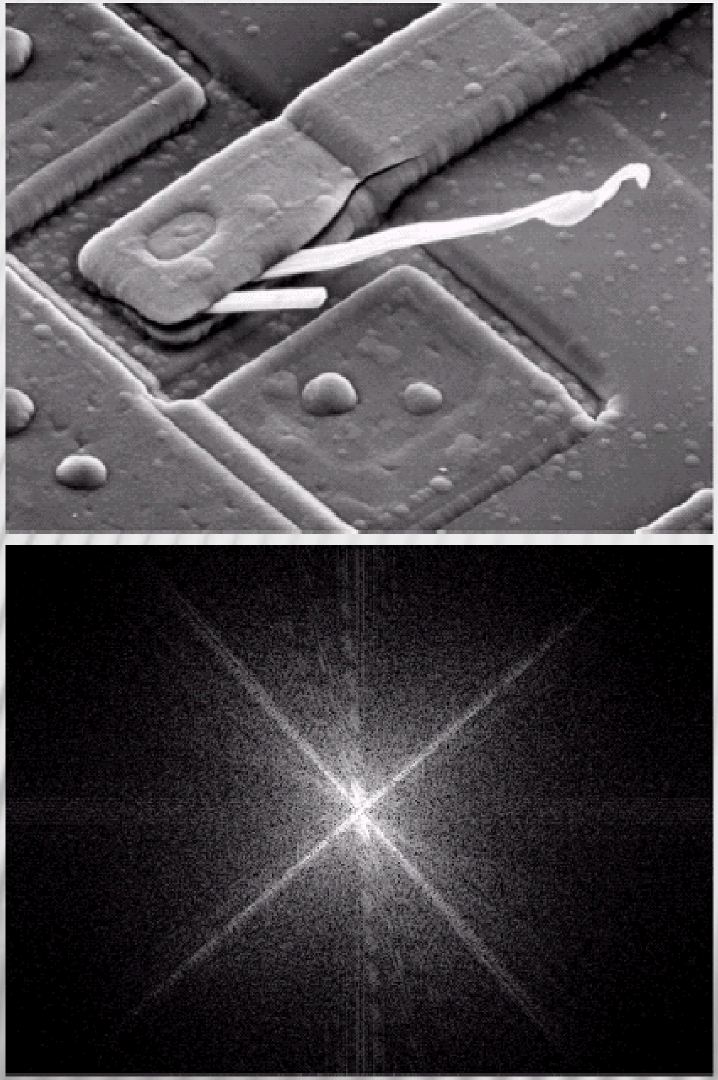
FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

Low-pass and High-pass Filters

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



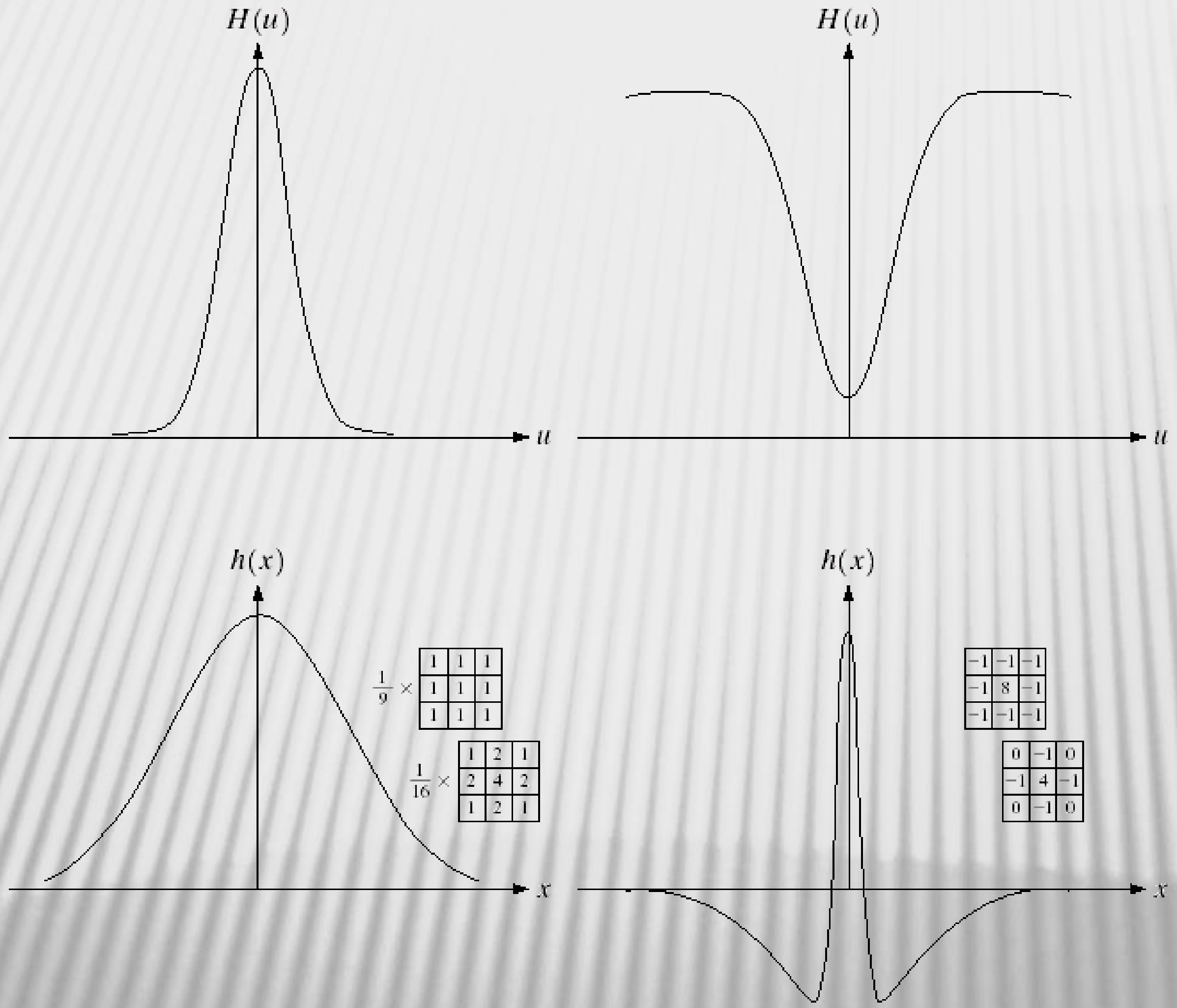
Low-pass and High-pass Filters



a
b

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.)

Low-pass and High-pass Filters



a b
c d

FIGURE 4.9
(a) Gaussian frequency domain lowpass filter.
(b) Gaussian frequency domain highpass filter.
(c) Corresponding lowpass spatial filter.
(d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

Smoothing Frequency-Domain Filters

- The basic model for filtering in the frequency domain

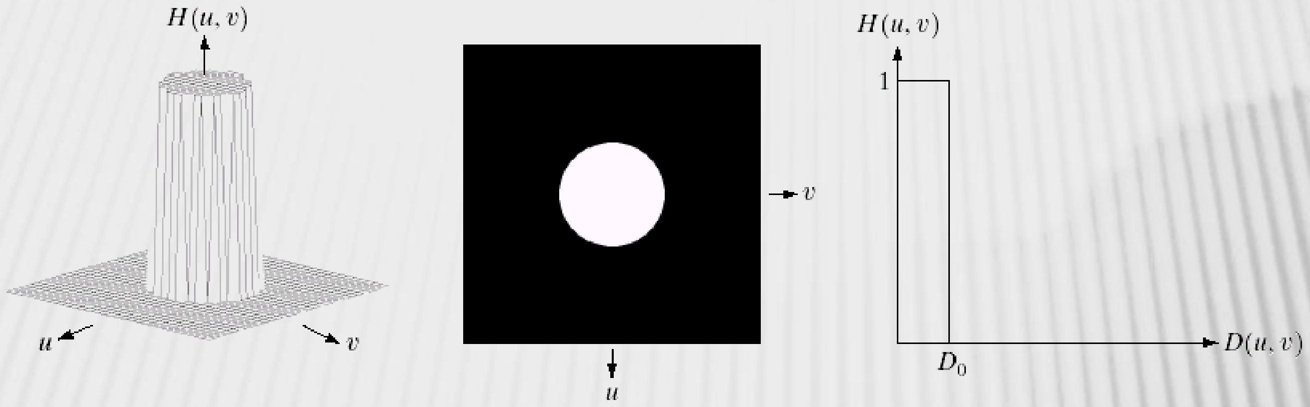
$$G(u,v) = H(u,v)F(u,v)$$

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

Smoothing Frequency Domain, *Ideal Low-Pass Filters (IPLF)*



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.

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

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

Smoothing Frequency Domain, Ideal Low-pass Filters

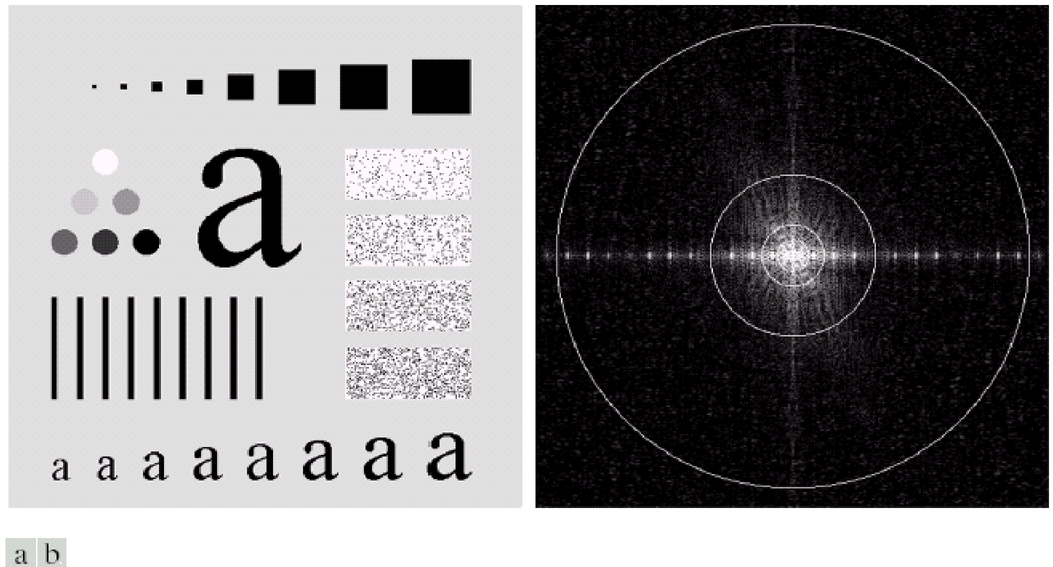
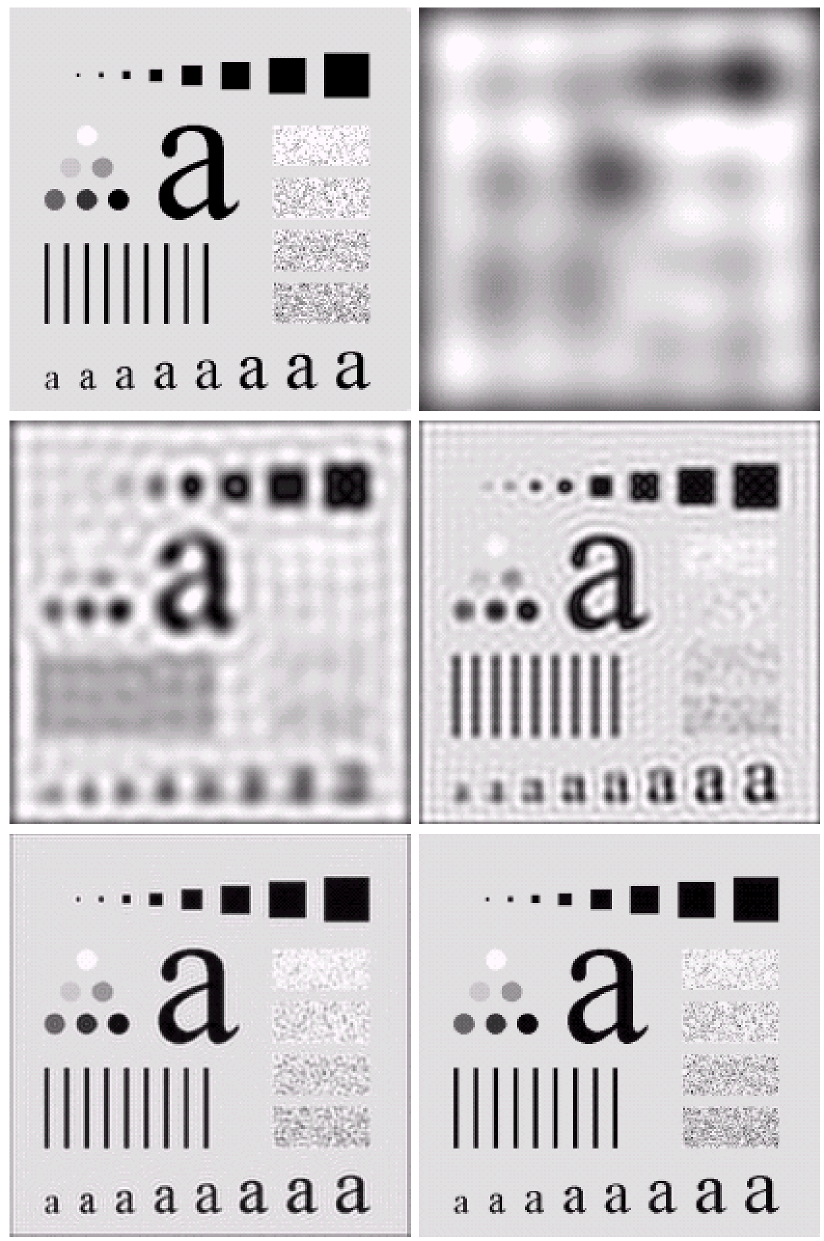


FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

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

The remained percentage power after filtration ⟶ $$\alpha = 100 \times \left[\sum_u \sum_v |F(u,v)| / P_T \right]$$

Smoothing Frequency Domain, Ideal Low-pass Filters



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

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

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

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

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

Cause of Ringing

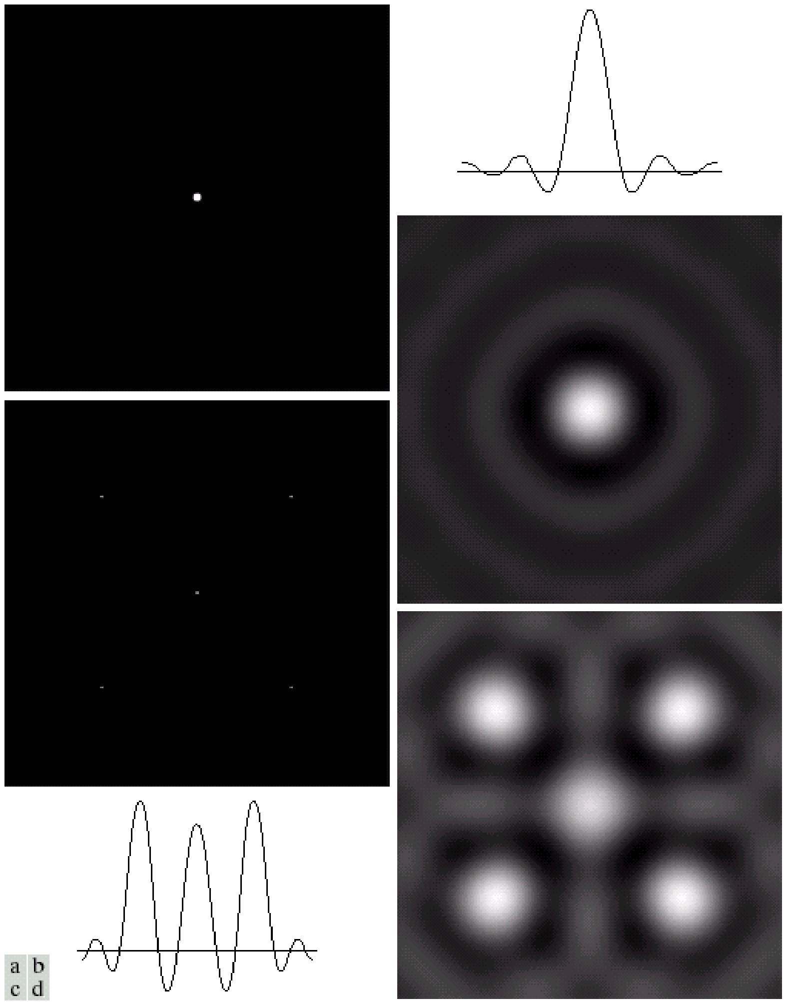


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

Ideal Low-pass Filter

Implement in Matlab the Ideal low-pass filter in the following equation.

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

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

1. You must give the user the ability to specify the cutoff frequency D_0
2. Calculate the The remained percentage power after filtration.
3. Display the image after filtration
4. Use Elaine image to test your program

Smoothing Frequency Domain, Butterworth Low-pass Filters

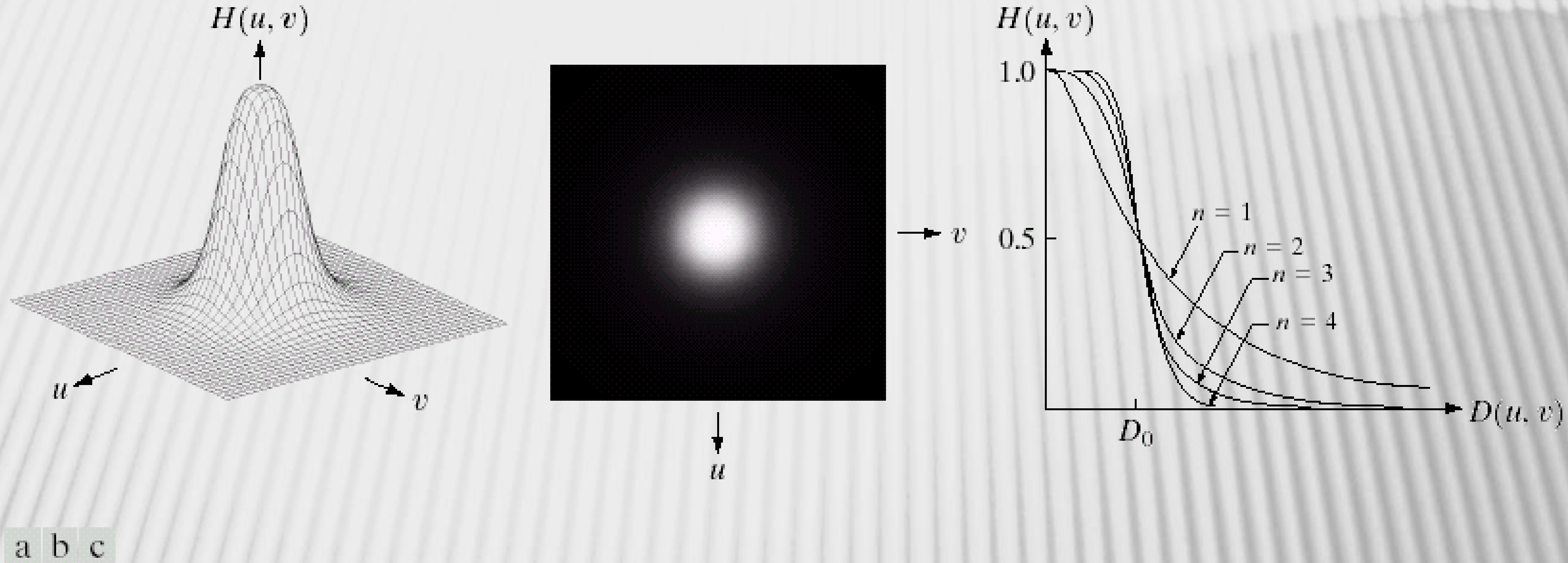


FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

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

Smoothing Frequency Domain, Butterworth Low-pass Filters

Butterworth Low-pass Filter: n=2

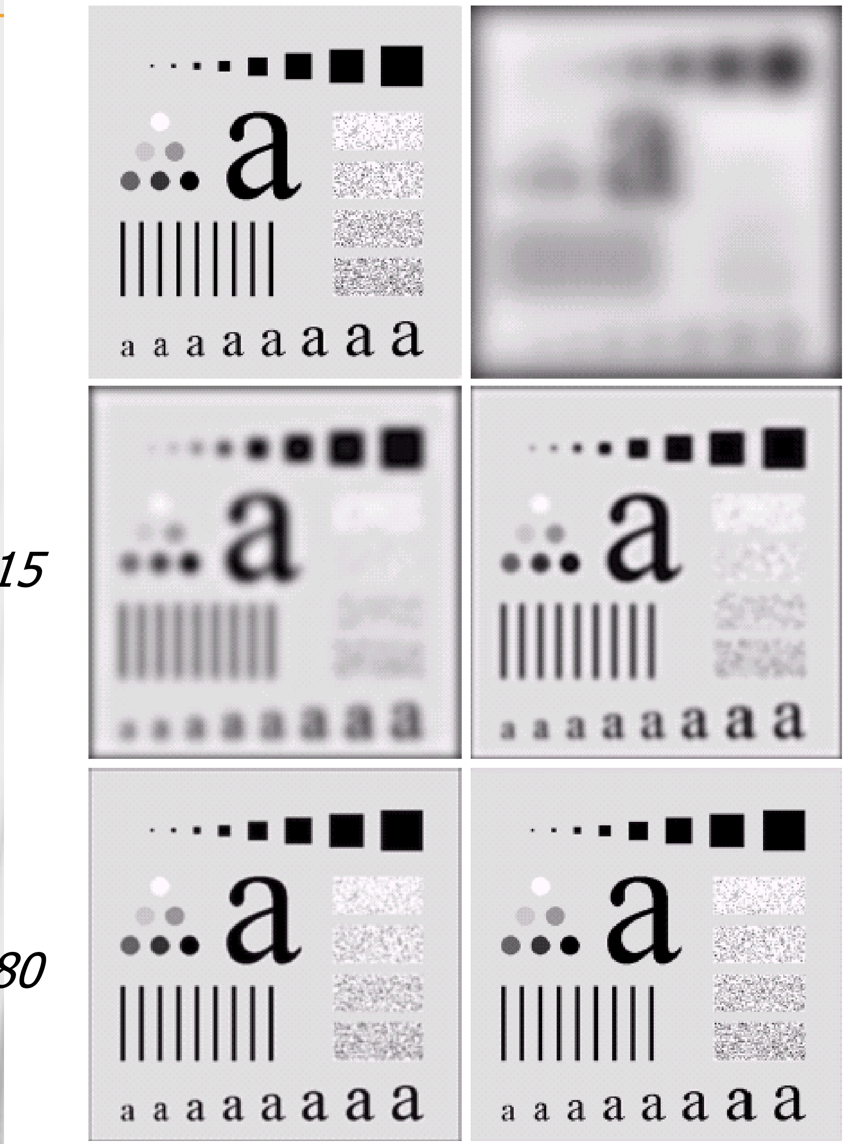
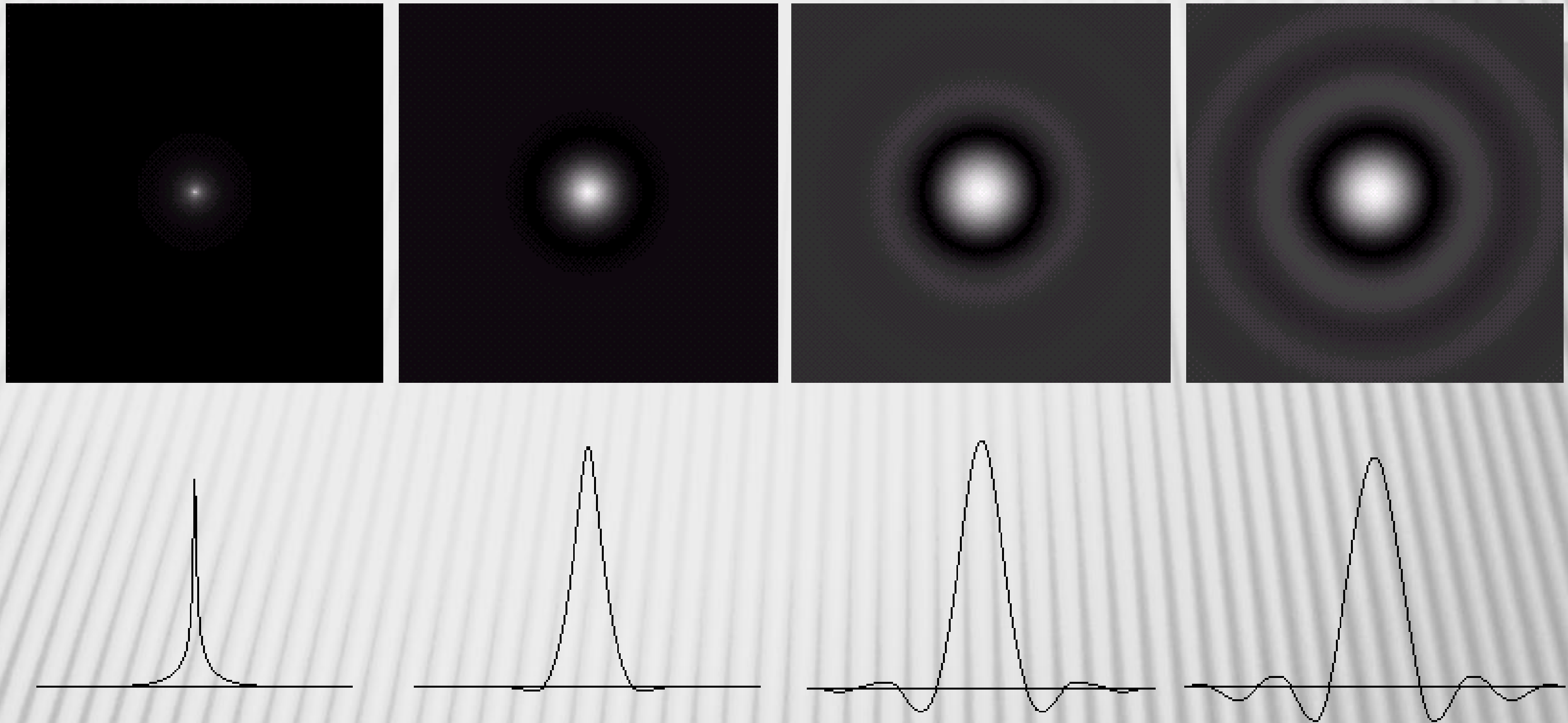


FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

Smoothing Frequency Domain, Butterworth Low-pass Filters



a b c d

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

Smoothing Frequency Domain, Gaussian Low-pass Filters

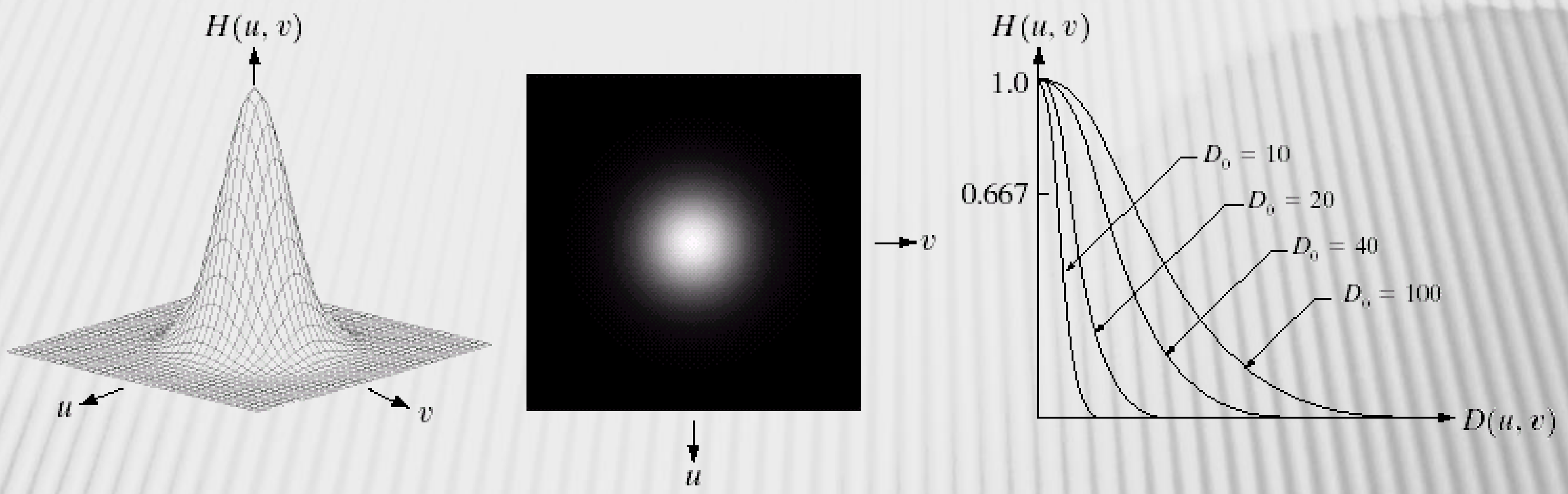


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

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

Smoothing Frequency Domain, Gaussian Low-pass Filters

- These properties make the Gaussian filter very useful for lowpass filtering an image. The amount of blur is controlled by σ^2. It can be implemented in either the spatial or frequency domain.

- Other filters besides lowpass can also be implemented by using two different sized Gaussian functions.

Smoothing Frequency Domain, Gaussian Low-pass Filters

Gaussian Low-pass

Radii= 5

Radii= 15

Radii= 30

Radii= 80

Radii= 230

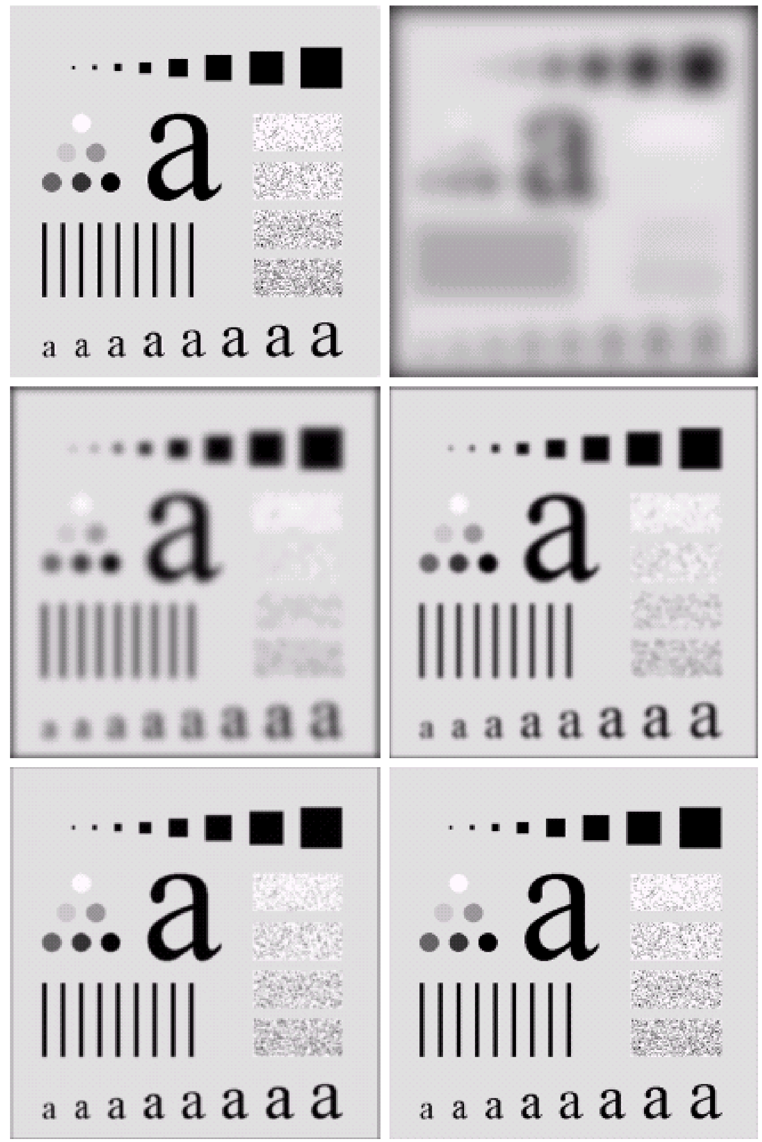


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

Smoothing Frequency Domain, Gaussian Low-pass Filters

a b c

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

Smoothing Frequency Domain, Gaussian Low-pass Filters

a b c

FIGURE 4.21 (a) Image showing prominent scan lines. (b) Result of using a GLPF with $D_0 = 30$. (c) Result of using a GLPF with $D_0 = 10$. (Original image courtesy of NOAA.)

Sharpening Frequency Domain :
High Pass Filters

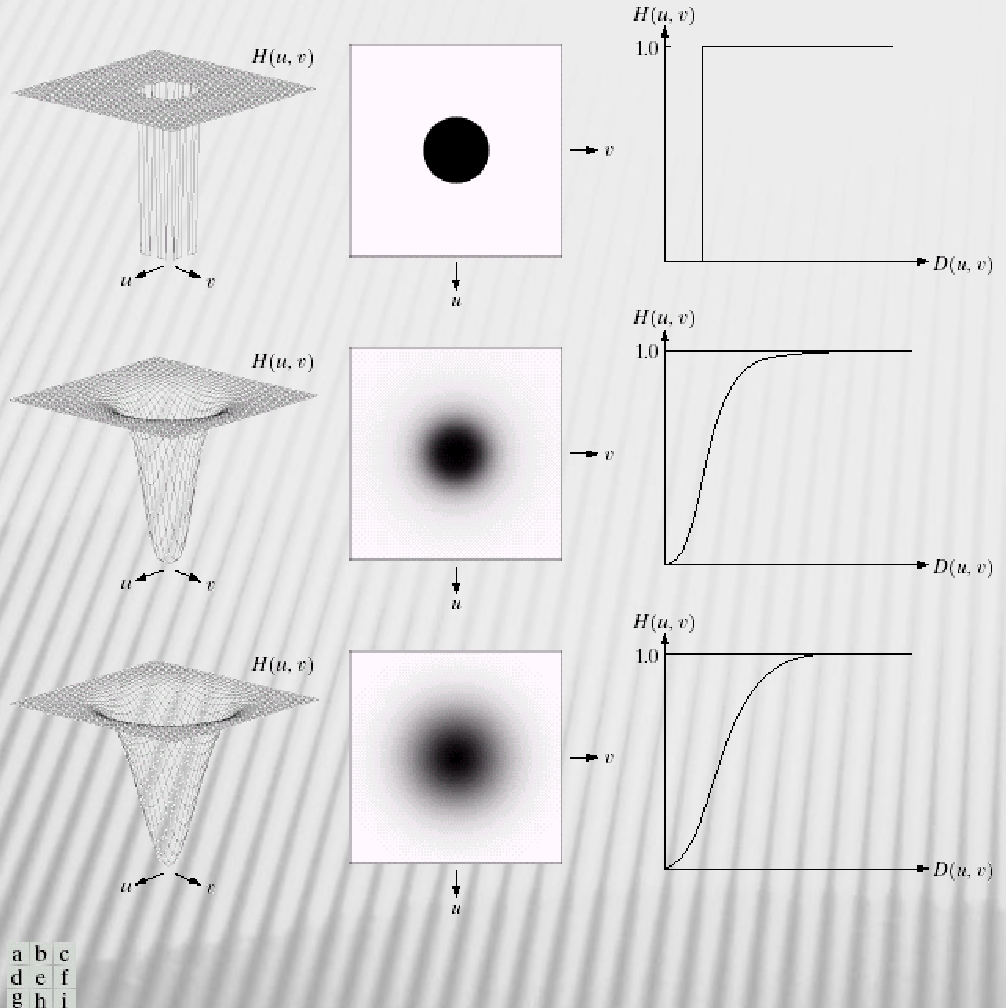


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

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

Sharpening Frequency Domain Filters

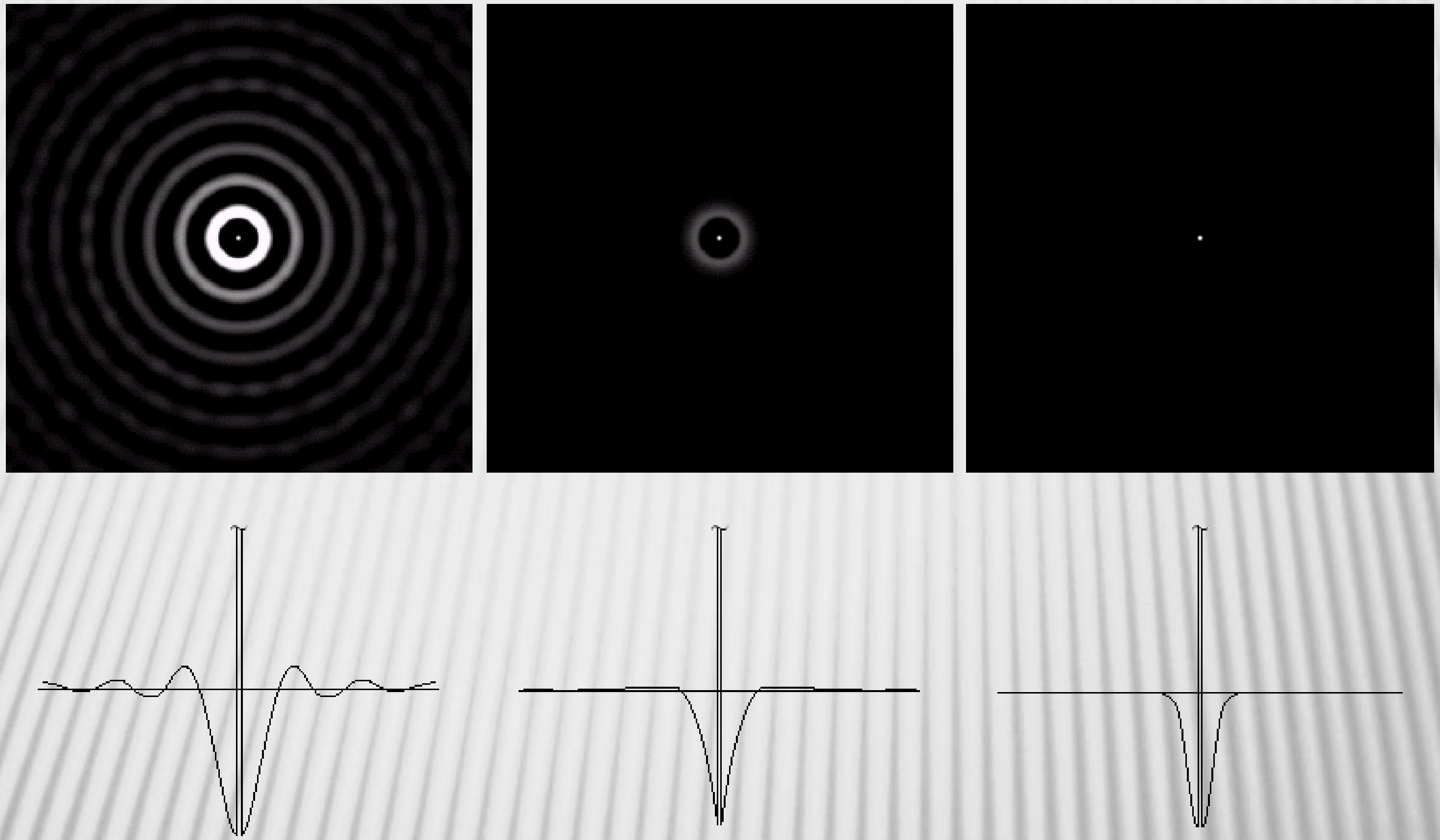
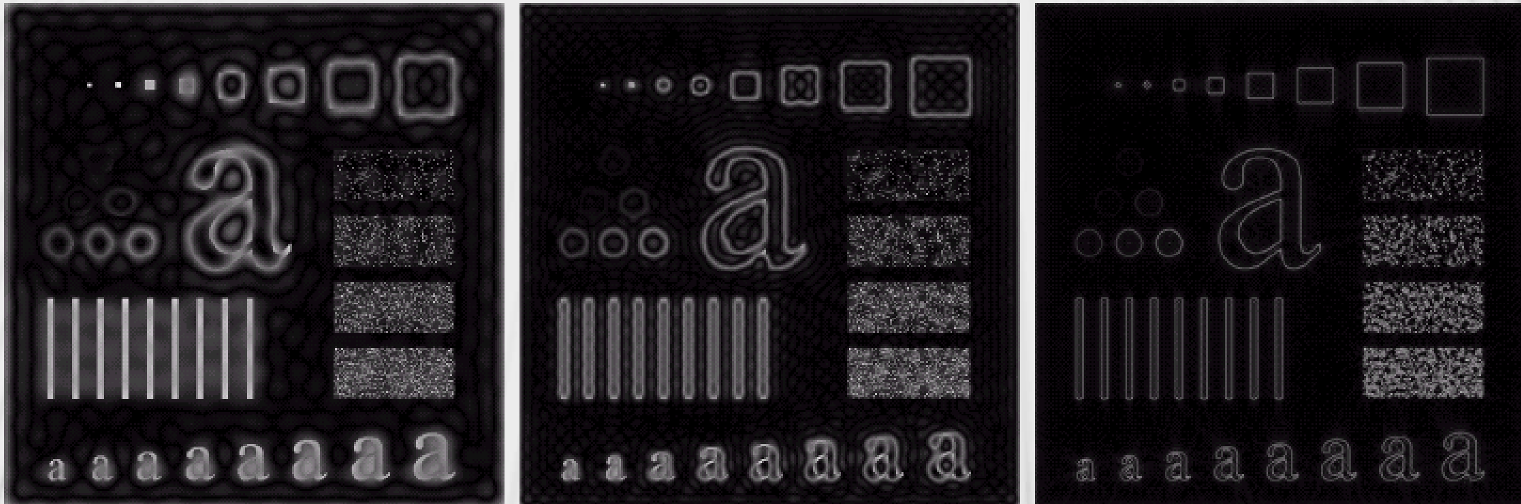


FIGURE 4.23 Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

Sharpening Frequency Domain, Ideal High-pass Filters

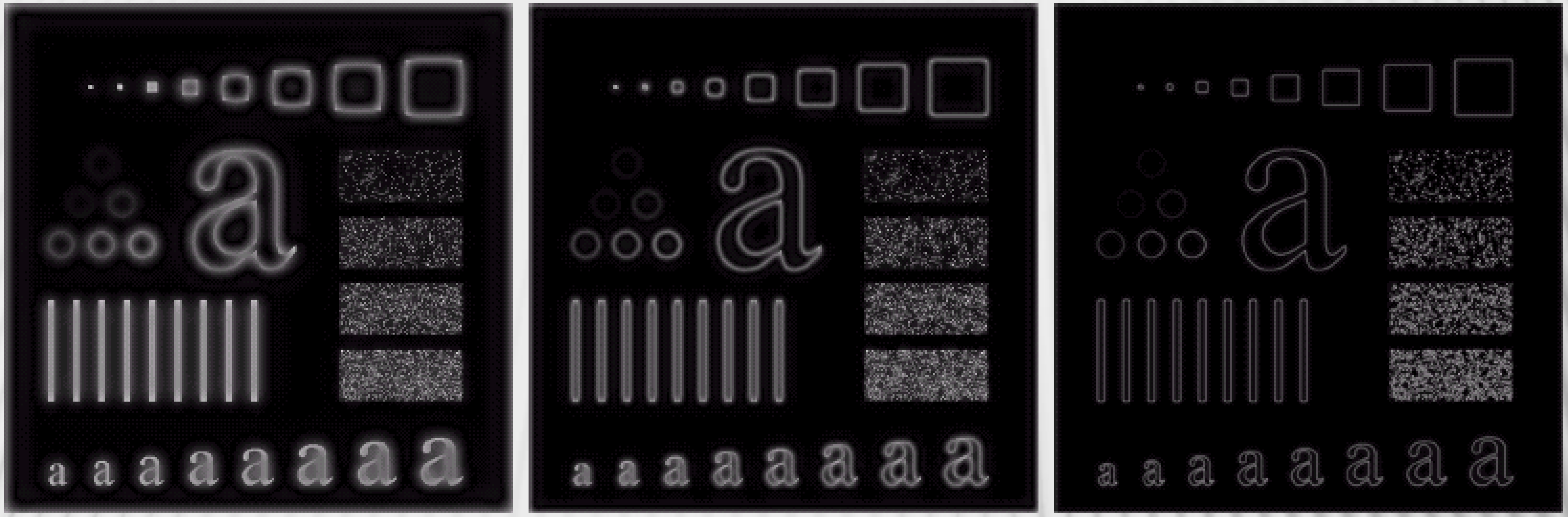


a b c

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

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

Sharpening Frequency Domain, Butterworth High-pass Filters

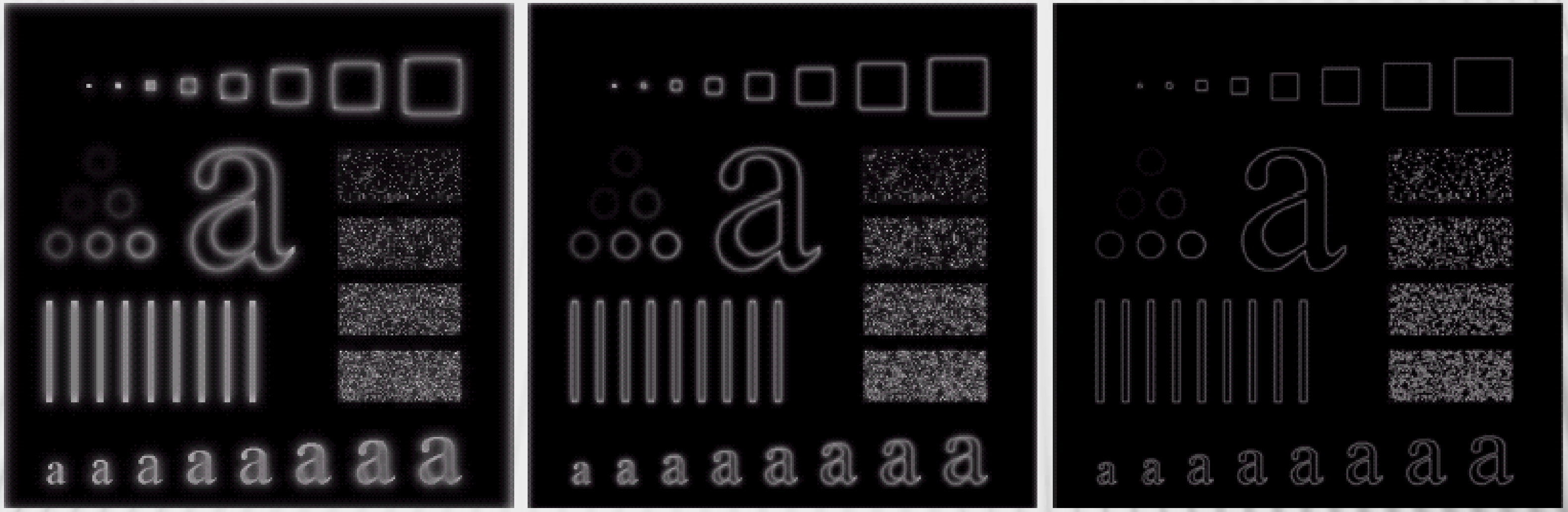


a b c

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

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

Sharpening Frequency Domain, Gaussian High-pass Filters



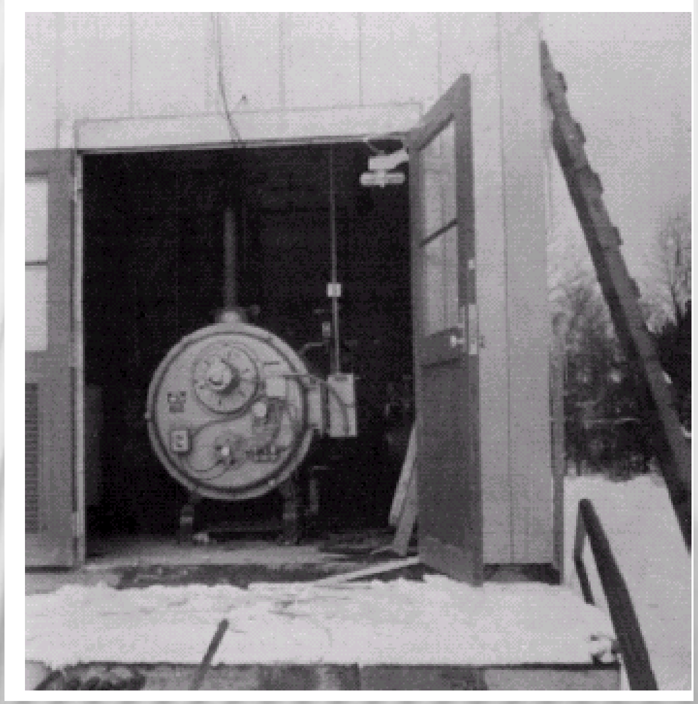
a b c

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

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

HOMOMORPHIC FILTERING

- Many times, we want to remove shading effects from an image (i.e., due to uneven illumination)
  - Enhance high frequencies
  - Attenuate low frequencies but preserve fine detail.



HOMOMORPHIC FILTERING (CONT'D)

- Consider the following model of image formation:

$$f(x, y) = i(x, y)\, r(x, y)$$

i(x,y): illumination
r(x,y): reflection

- In general, the illumination component i(x,y) varies **slowly** and affects low frequencies mostly.
- In general, the reflection component r(x,y) varies **faster** and affects high frequencies mostly.

<u>IDEA:</u> separate low frequencies due to i(x,y)
from high frequencies due to r(x,y)

HOW ARE FREQUENCIES MIXED TOGETHER?

- Low and high frequencies from **i(x,y)** and **r(x,y)** are mixed together.

$$f(x, y) = i(x, y)\, r(x, y)$$

- When applying filtering, it is difficult to handle low/high frequencies separately.

CAN WE SEPARATE THEM?

- Idea:

Take the ln( ) of $f(x, y) = i(x, y)\, r(x, y)$

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

STEPS OF HOMOMORPHIC FILTERING

(1) Take $$ln(f(x,y)) = ln(i(x,y)) + ln(r(x,y))$$

(2) Apply FT: $$F(ln(f(x,y))) = F(ln(i(x,y))) + F(ln(r(x,y)))$$

or

$$Z(u,v) = Illum(u,v) + Refl(u,v)$$

(3) Apply H(u,v)

$$Z(u,v)H(u,v) = Illum(u,v)H(u,v) + Refl(u,v)H(u,v)$$

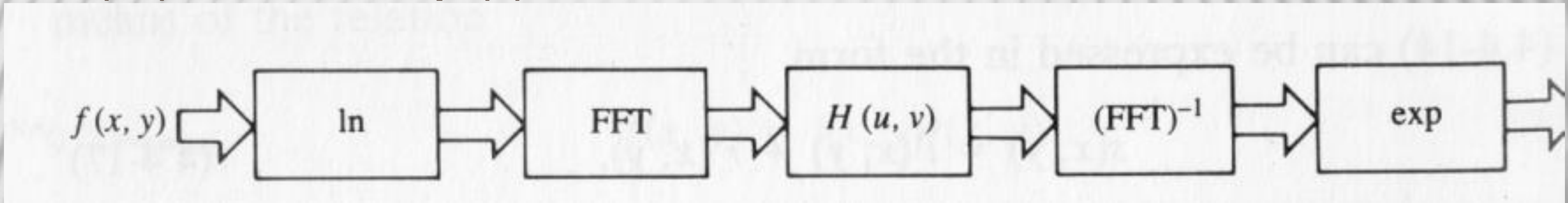
STEPS OF HOMOMORPHIC FILTERING (CONT'D)

(4) Take Inverse FT:

$$F^{-1}(Z(u,v)H(u,v)) = F^{-1}(Illum(u,v)H(u,v)) + F^{-1}(Refl(u,v)H(u,v))$$

$$e^{s(x,y)} = e^{i'(x,y)}e^{r'(x,y)}$$

(5) Take exp( )



EXAMPLE USING HIGH-FREQUENCY EMPHASIS

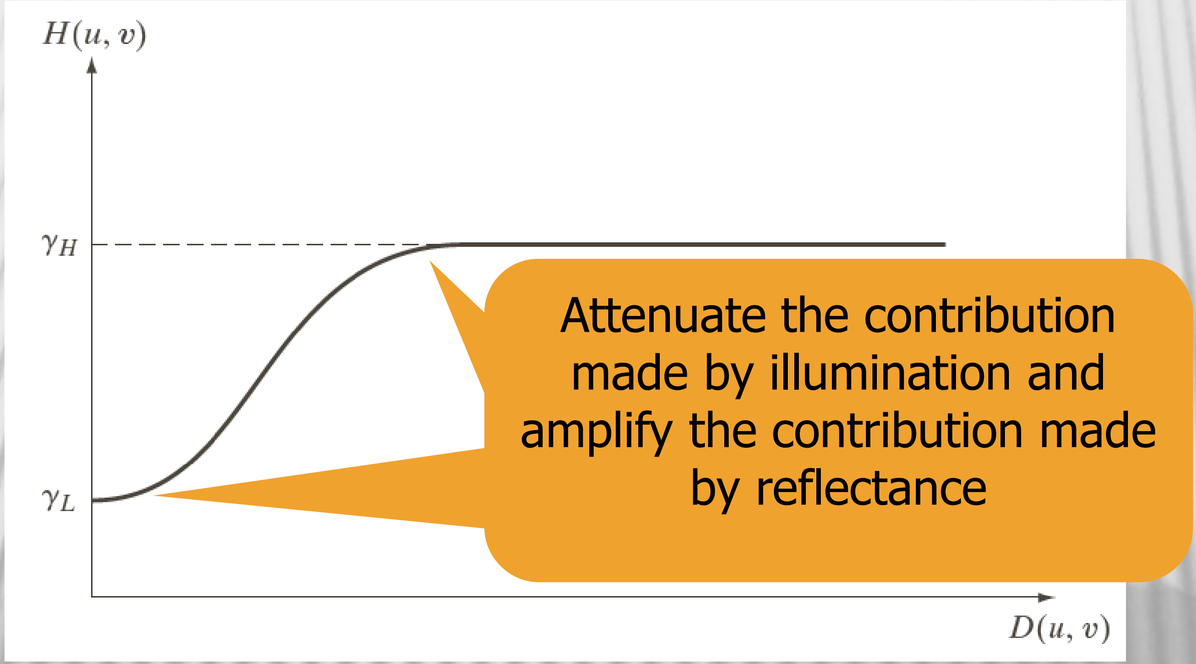
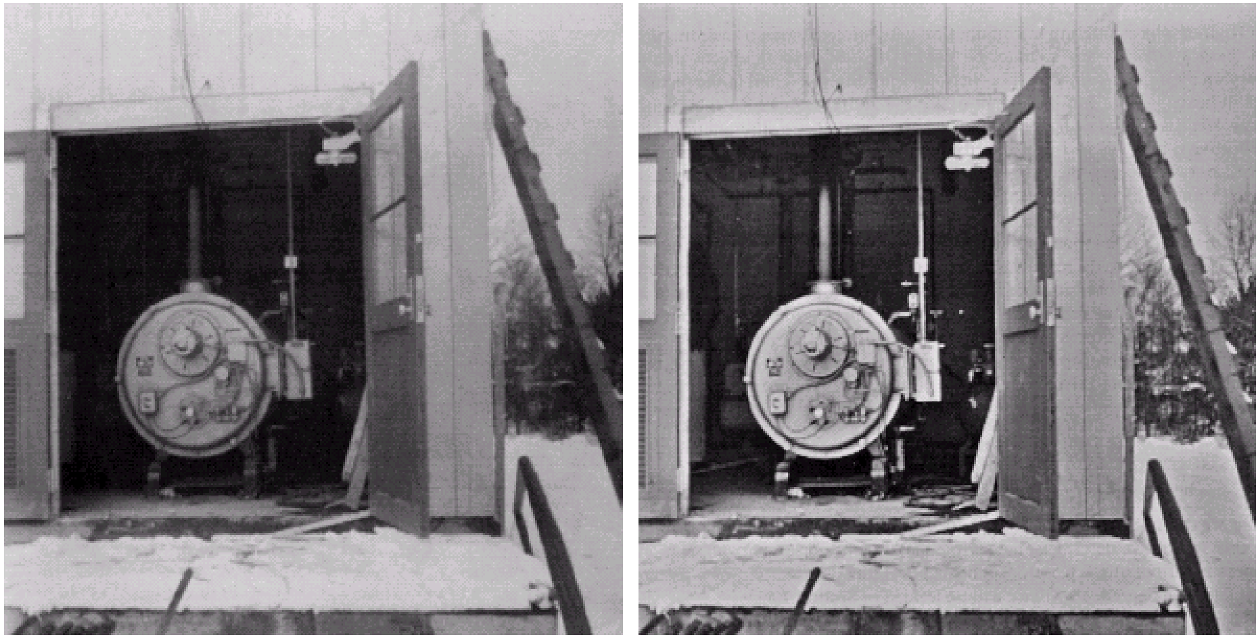


FIGURE 4.61 Radial cross section of a circularly symmetric homomorphic filter function. The vertical axis is at the center of the frequency rectangle and $D(u, v)$ is the distance from the center.

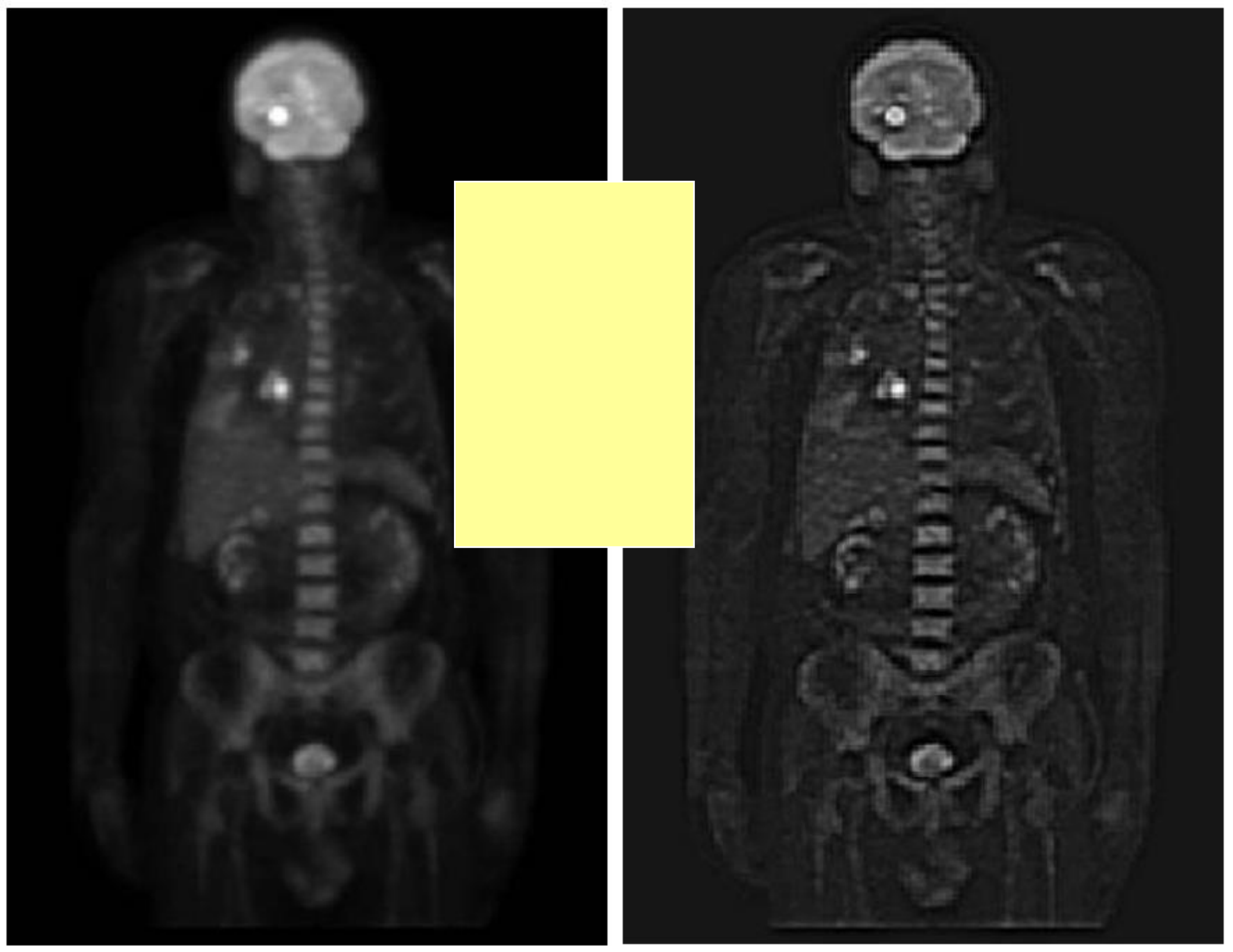
HOMOMORPHIC FILTERING: EXAMPLE



a b

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

HOMOMORPHIC FILTERING: EXAMPLE



a b

FIGURE 4.62
(a) Full body PET scan. (b) Image enhanced using homomorphic filtering. (Original image courtesy of Dr. Michael E. Casey, CTI PET Systems.)

Summary of Some Important Properties of the 2-D Fourier Transform

TABLE 4.1
Summary of some important properties of the 2-D Fourier transform.

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	$\lvert F(u,v)\rvert = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$, $R = \text{Real}(F)$ and $I = \text{Imag}(F)$
Phase angle	$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$
Power spectrum	$P(u,v) = \lvert F(u,v)\rvert^2$
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	$f(x,y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$ $f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M+vy_0/N)}$ When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$, then $f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$ $f(x-M/2, y-N/2) \Leftrightarrow F(u,v)(-1)^{u+v}$

Summary of Some Important Properties of the 2-D Fourier Transform

TABLE 4.1 *(continued)*

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	$\Im[f_1(x,y)+f_2(x,y)] = \Im[f_1(x,y)] + \Im[f_2(x,y)]$ $\Im[f_1(x,y)\cdot f_2(x,y)] \neq \Im[f_1(x,y)]\cdot\Im[f_2(x,y)]$
Scaling	$af(x,y) \Leftrightarrow aF(u,v),\ f(ax,by) \Leftrightarrow \dfrac{1}{\lvert ab\rvert}F(u/a, v/b)$
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.

Summary of Some Important Properties of the 2-D Fourier Transform

TABLE 4.1 *(continued)*

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

Summary of Some Important Properties of the 2-D Fourier Transform

TABLE 4.1 *(continued)*

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	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow \dfrac{1}{2}\left[\delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0)\right]$
Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow j\dfrac{1}{2}\left[\delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0)\right]$

† Assumes that functions have been extended by zero padding.