Chapter - 2
Image Processing

Prof. Mittal Darji
IT, GCET

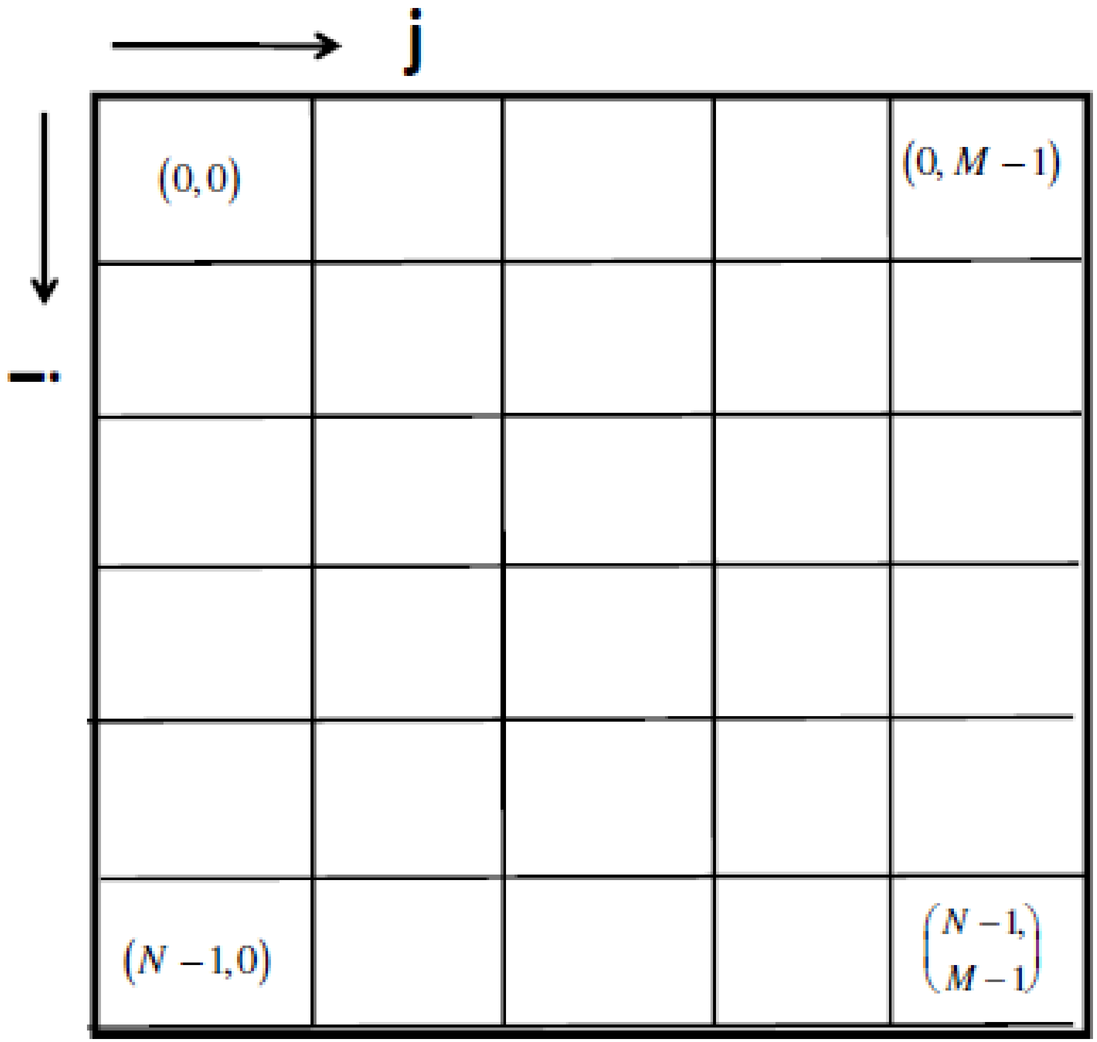
Why is DIP important? Two objectives

1) To improve the quality of pictorial information for better human interpretation
2) Processing of image data for storage, transmission and representation for automatic machine interpretation

What is an Image?

- An image may be defined as a two-dimensional function f(x ,y), where x and y are spatial coordinates and the amplitude of *f* at any point is called the intensity or gray level of the image at that point.
- When x ,y and the amplitude values of f are all finite and discrete quantities, the image is called a digital image.

Mathematical representation of DI

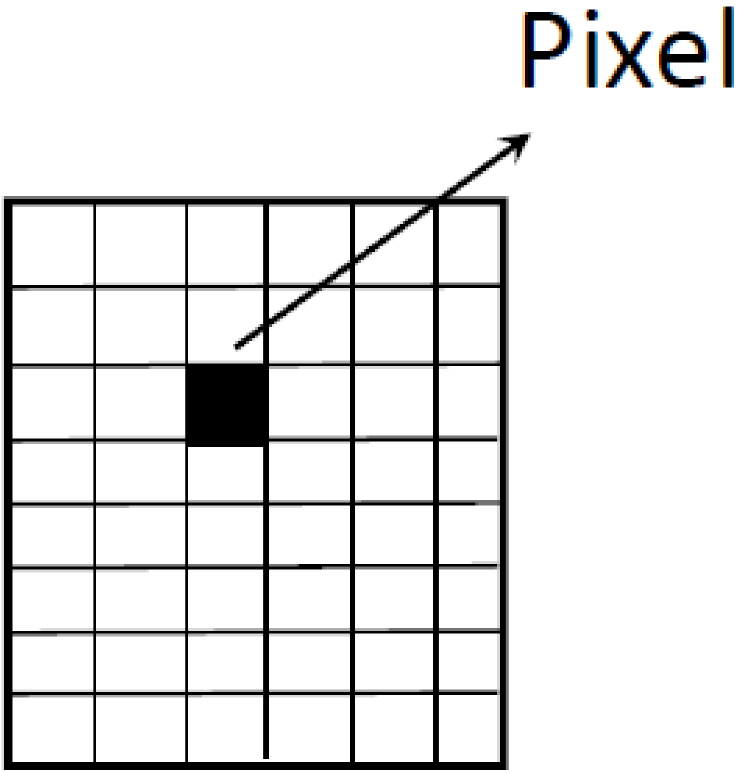


What is DIP?

- The processing of digital images by means of computer to extract meaningful information is called Digital Image Processing.
- Here, the meaningful information is dependent on the application.

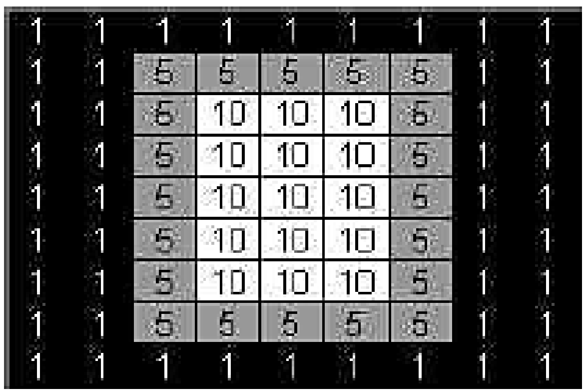
Pixel

- Digital image is composed of a finite number of elements each has a particular location and value. These elements are referred as pixels.
- Pixels are also known as picture elements, image elements or pels.

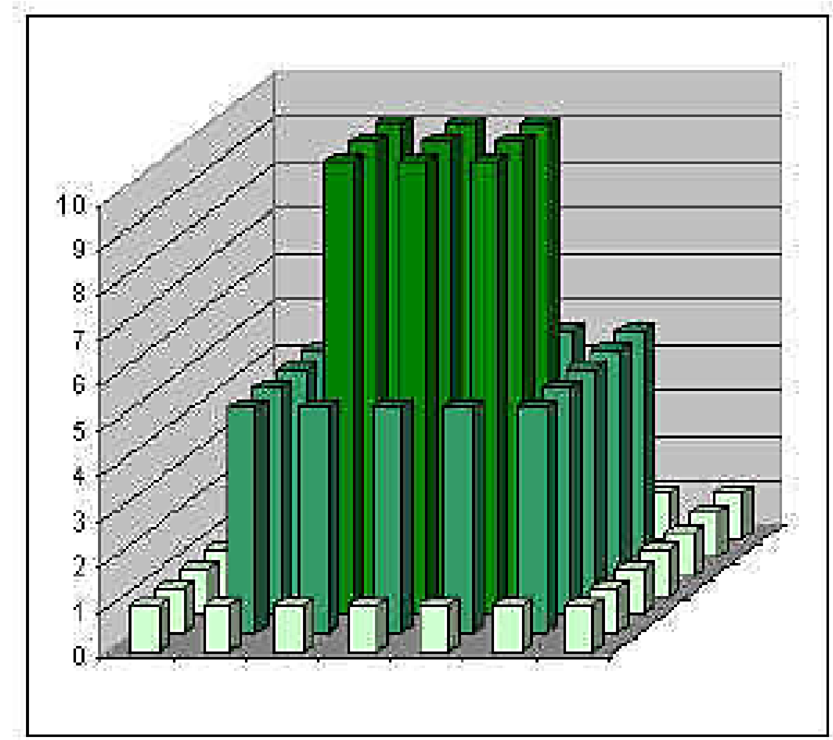


Smallest part of any image (Digital Image).

Bar diagram

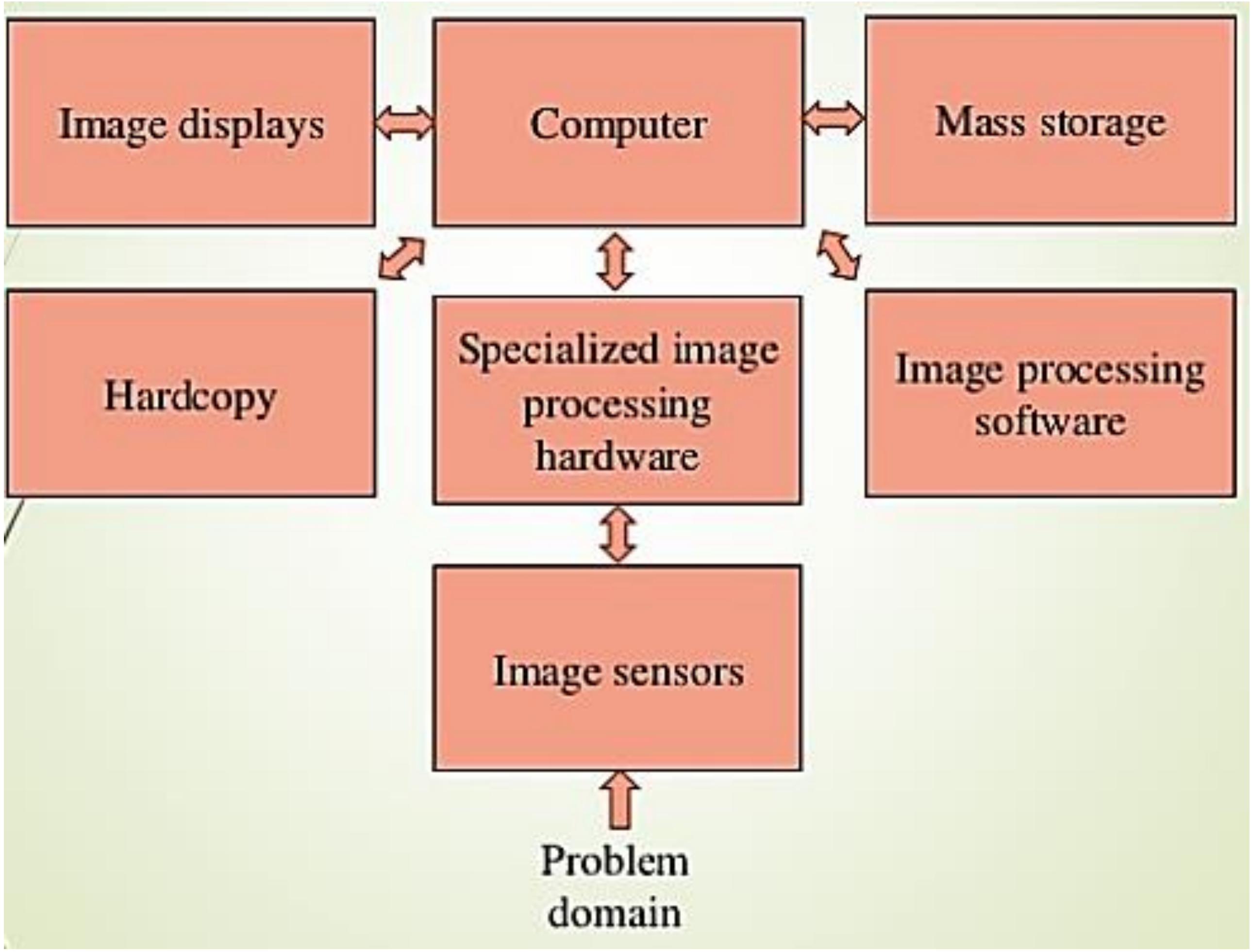


Pixel Image



Corresponding Bar Diagram

Components of DIP



Spatial resolution of image

- Spatial resolution refers to the number of pixels utilized in construction of the image.
- Images having higher spatial resolution are composed with a greater number of pixels than those of lower spatial resolution.
- Measured in form of lines per unit or dots per unit.
- DPI (dots per inch) is the commonly used measure.

Spatial resolution of image

What happens as the resolution decreases (i.e., becomes coarser)?

Full res
1 cm/pel

1/2 Res
2 cm/pel

1/4 Res
4 cm/pel

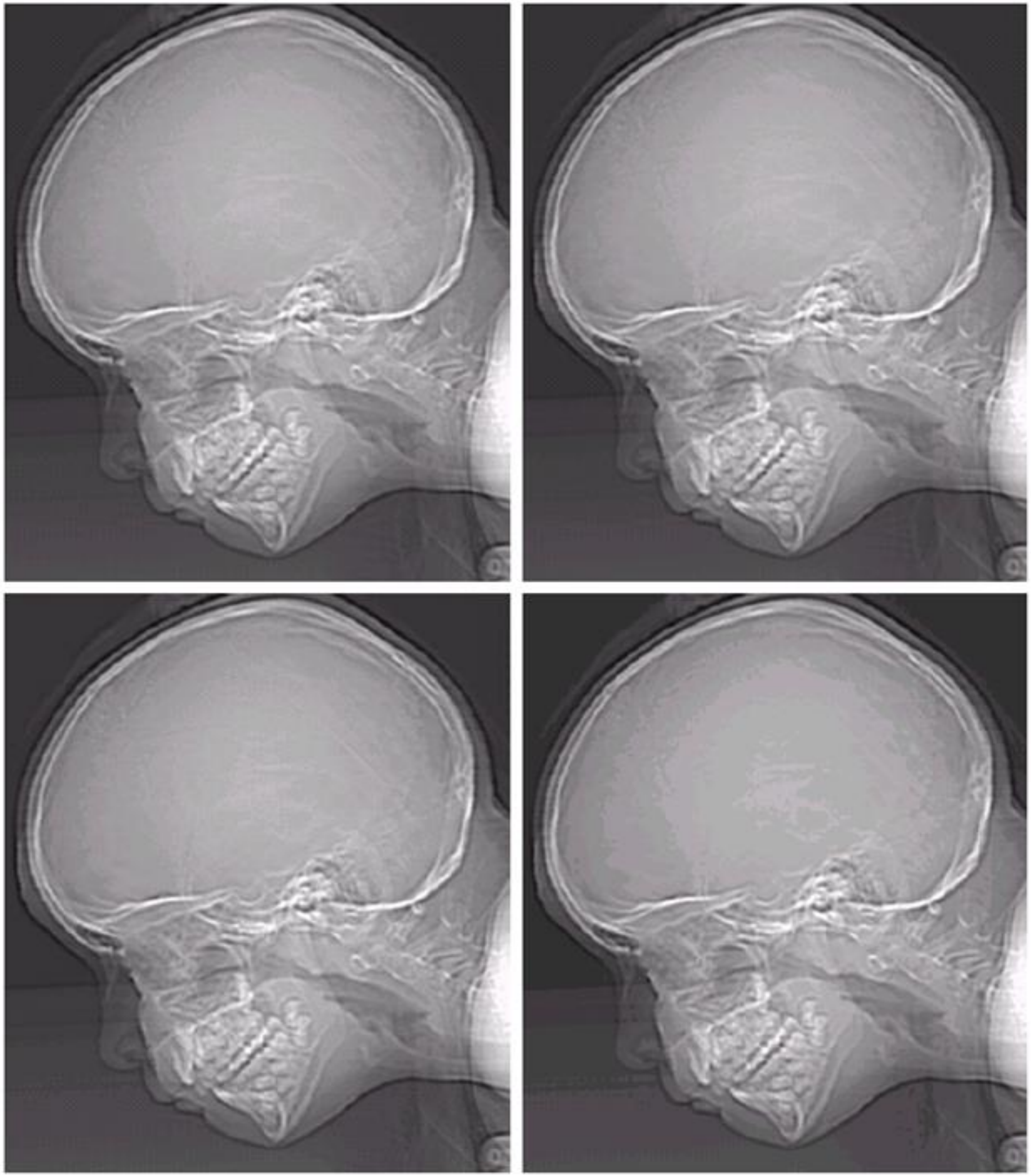
1/8 Res
8 cm/pel

a b
c d

FIGURE 2.20 Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.

Intensity/Gray level resolution of Image

- It refers to the smallest change in the intensity level.
- Gray level resolution refers to the change in the shades or levels of gray in an image.
- Gray level resolution is equal to the number of bits per pixel.



a b
c d

FIGURE 2.21
(a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

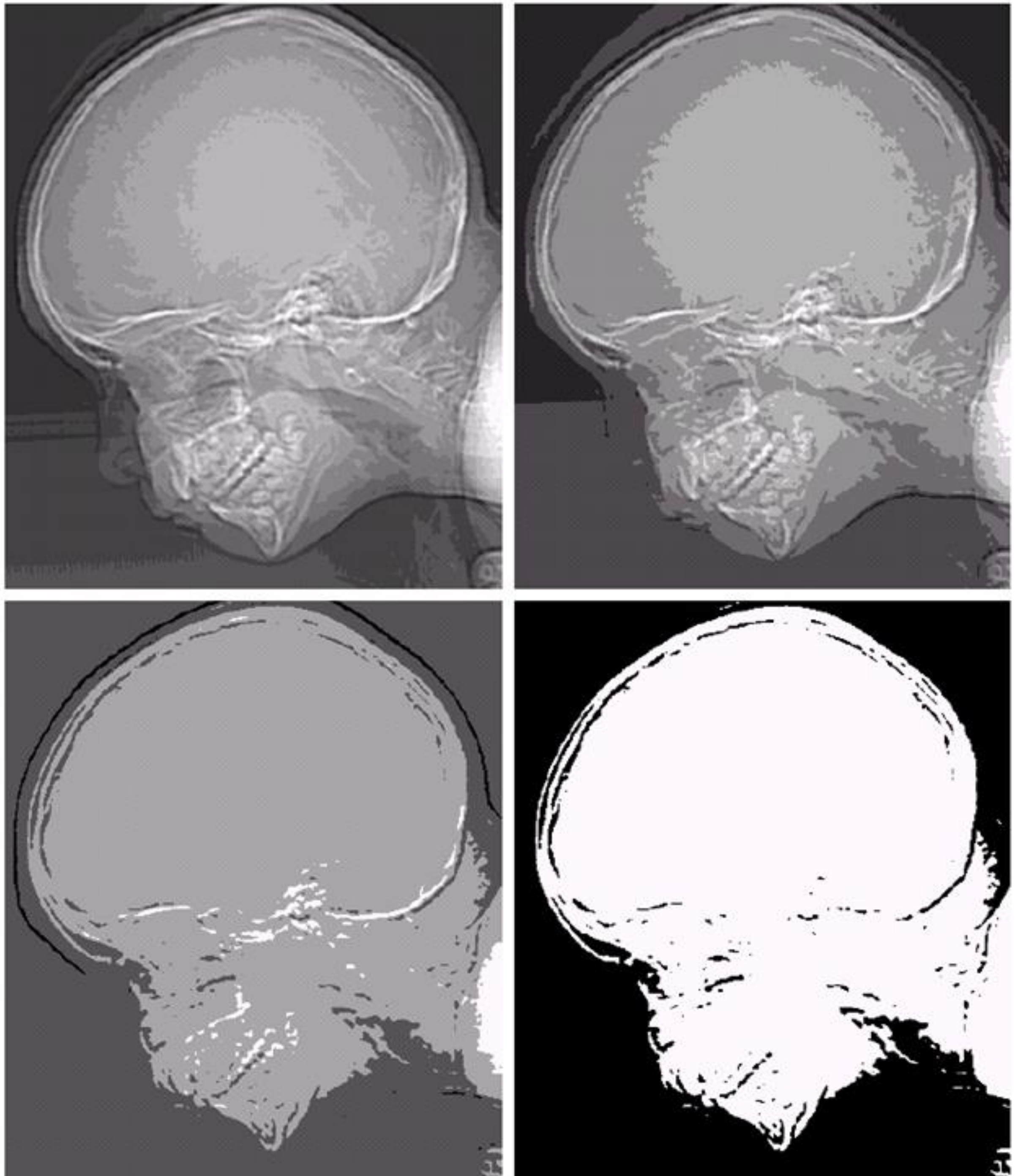


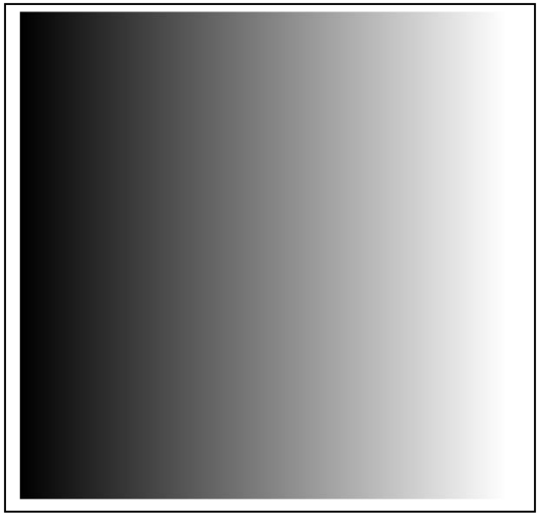
FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

False contouring

- It is the effect caused by insufficient number of intensity levels in smooth areas of a digital image.
- A set of very fine ridge like structures in area of constant or nearly constant intensity.

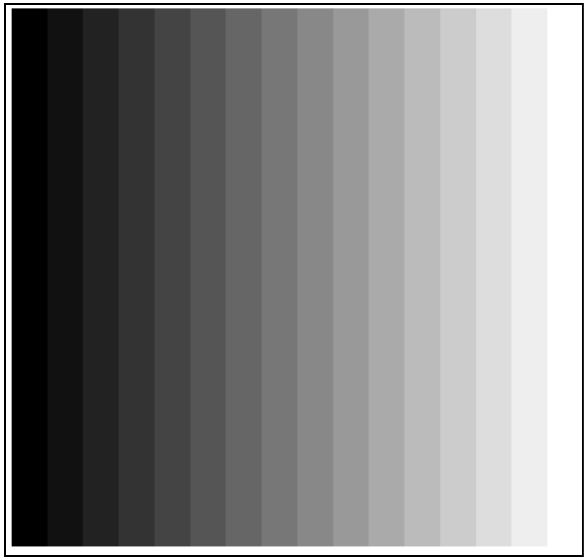
Intensity/Gray level resolution

- Number of bits available affects how accurately continuous tone image is represented.
- This figure shows image with 8 bits per pixel(256 grey levels). It looks continuous.



Intensity/Gray level resolution

- The same scene is here in 4 bits per pixel (16 gray levels).



Each kind of image has few attributes attached to it like **number of channels** and **depth .**

- Number of channels : Defines the dimension of array , each pixel is.
- Depth : Defines the maximum bit size of the number which is stored in the array

Types of Images: Binary Image

Again, as the name suggest each number associated with the pixel can have just one of the two possible values.

- Each pixel is a 1 bit number.
- It can take either 0 or 1 as its value.
- 0 corresponds to **Black**
- 1 corresponds to **White**
- Number of channels of a binary Image is 1
- Depth of a binary image is 1(bit)

Example of a binary image

Types of Images: Gray scale Image

- Each pixel is a 8 bit number
- It can take values from 0-255
- Each value corresponds to a shade between black and white(0 for black and 255 for white)
- Number of channels for a grayscale image is 1
- Depth of a gray scale image is 8(bits)

Example of a grayscale image

Types of Images: Color Image

- Each pixel stores three values:
    1. R : 0-255
    2. G : 0-255
    3. B : 0-255
- Each number between 0-255 corresponds to a shade of corresponding color
- Depth of a RGB image is 8(bits)
- Number of channels for a RGB image is 3

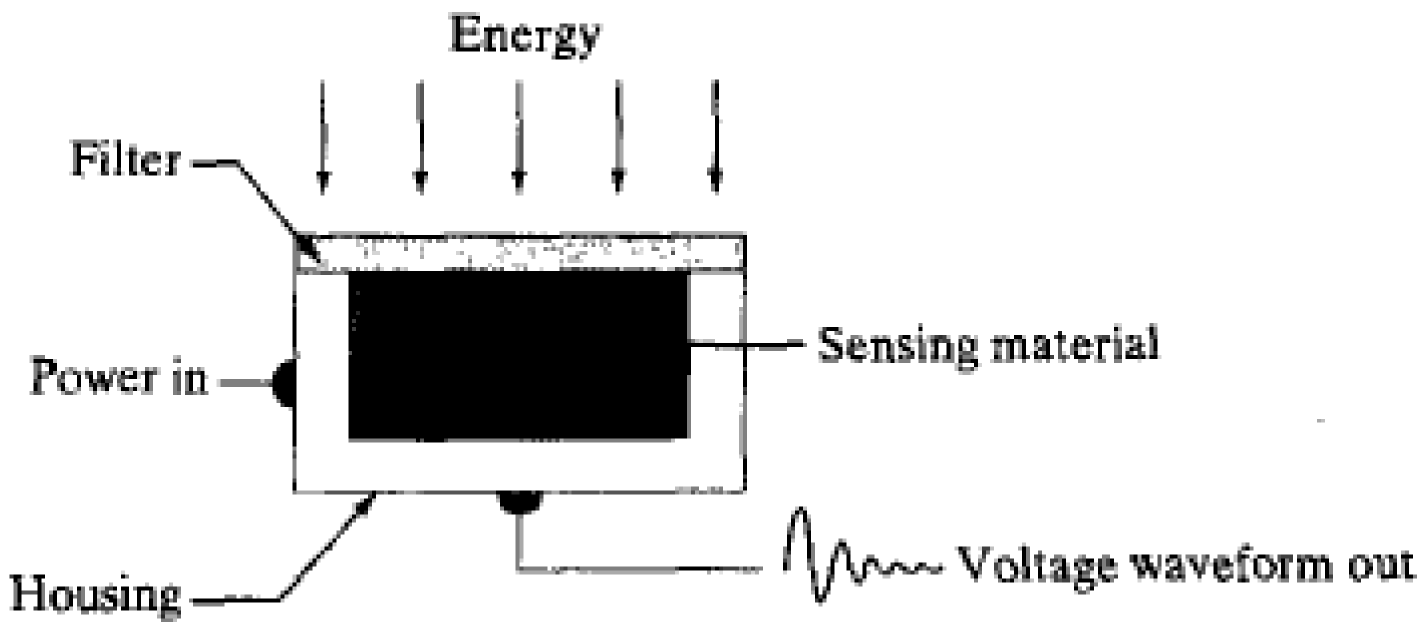
Example of a RGB image

Image acquisition

- Most of the images are generated by the combination of an illumination source and reflection or absorption of energy from that source by the elements of the scene being imaged.

Single sensor

- Used to transfer illumination energy into digital image.
- The idea is simple: incoming energy is transferred into voltage by the combination of input electrical power and sensor material.
- Output voltage waveform is the response of sensor and by digitizing it, a digital quantity is obtained.


Energy
Filter
Sensing material
Power in
Housing
Voltage waveform out

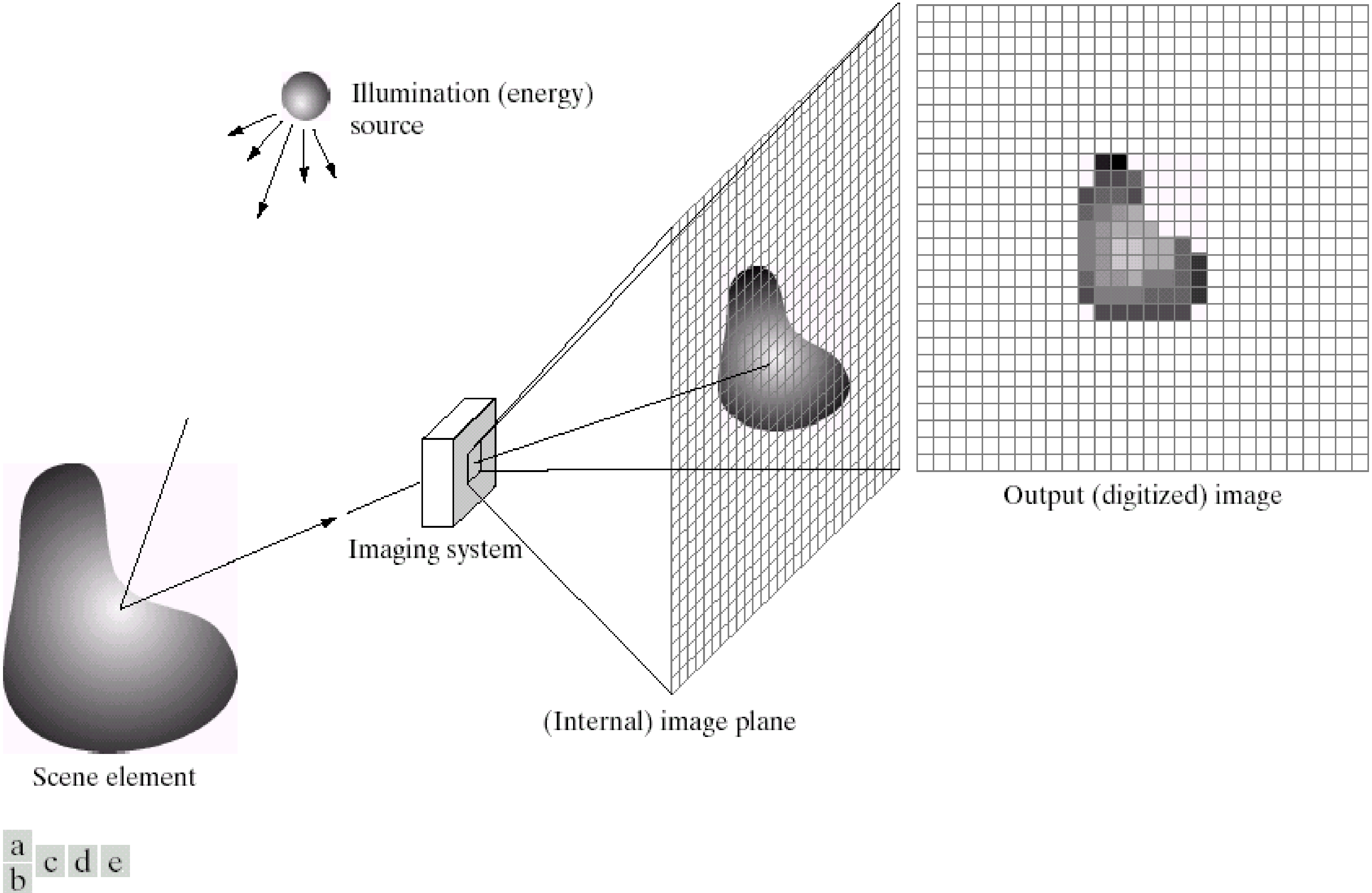


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Image sampling & quantization

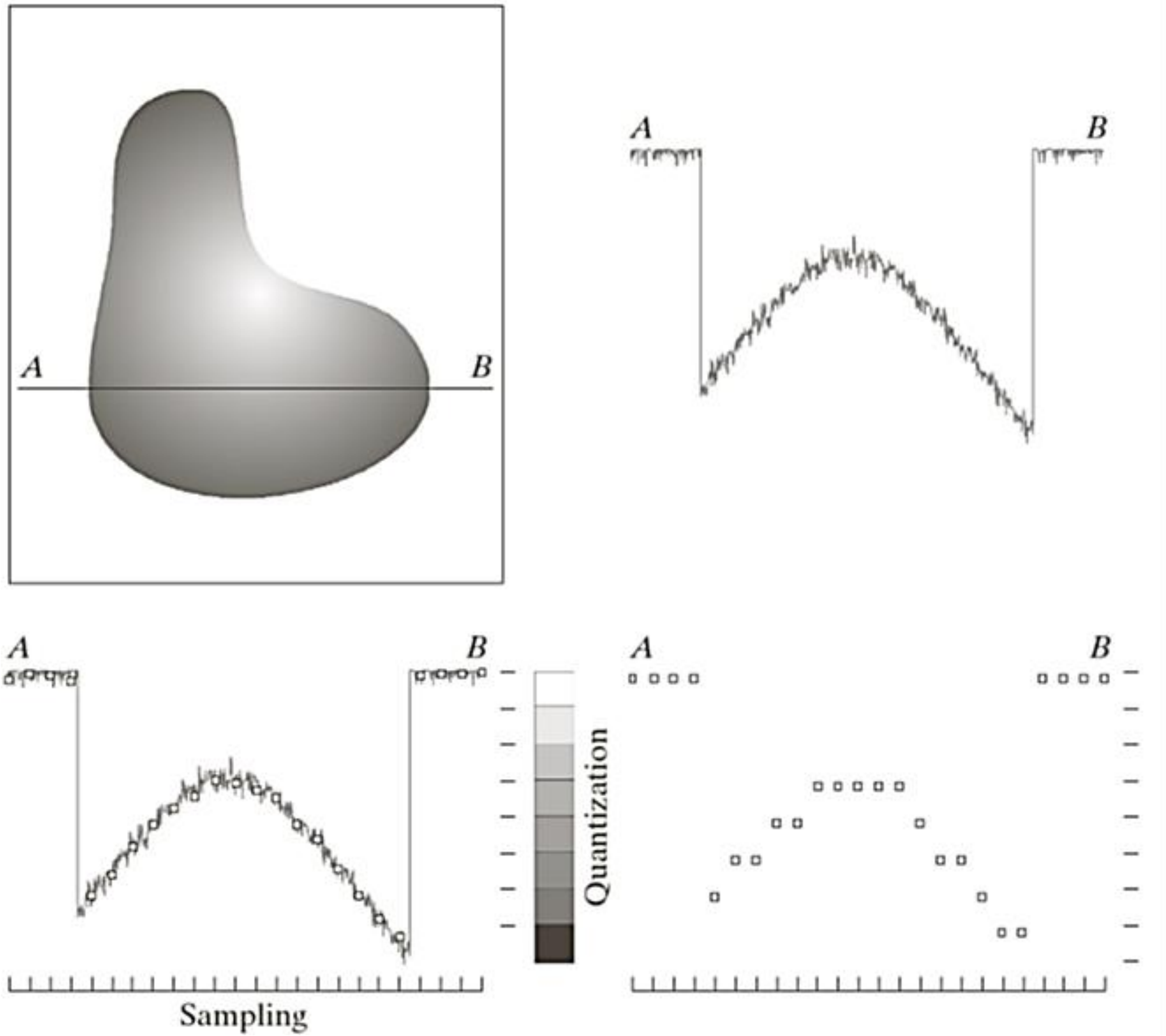
- Output of most sensors is a continuous voltage waveform.
- To create a digital image, we have to convert the analog signal into digital form using a digitizer.
- We have to sample the signal in both coordinates and amplitude.
- This involves 2 processes: sampling & quantization.

Image sampling

- Digitizing the coordinate values is called **sampling**.

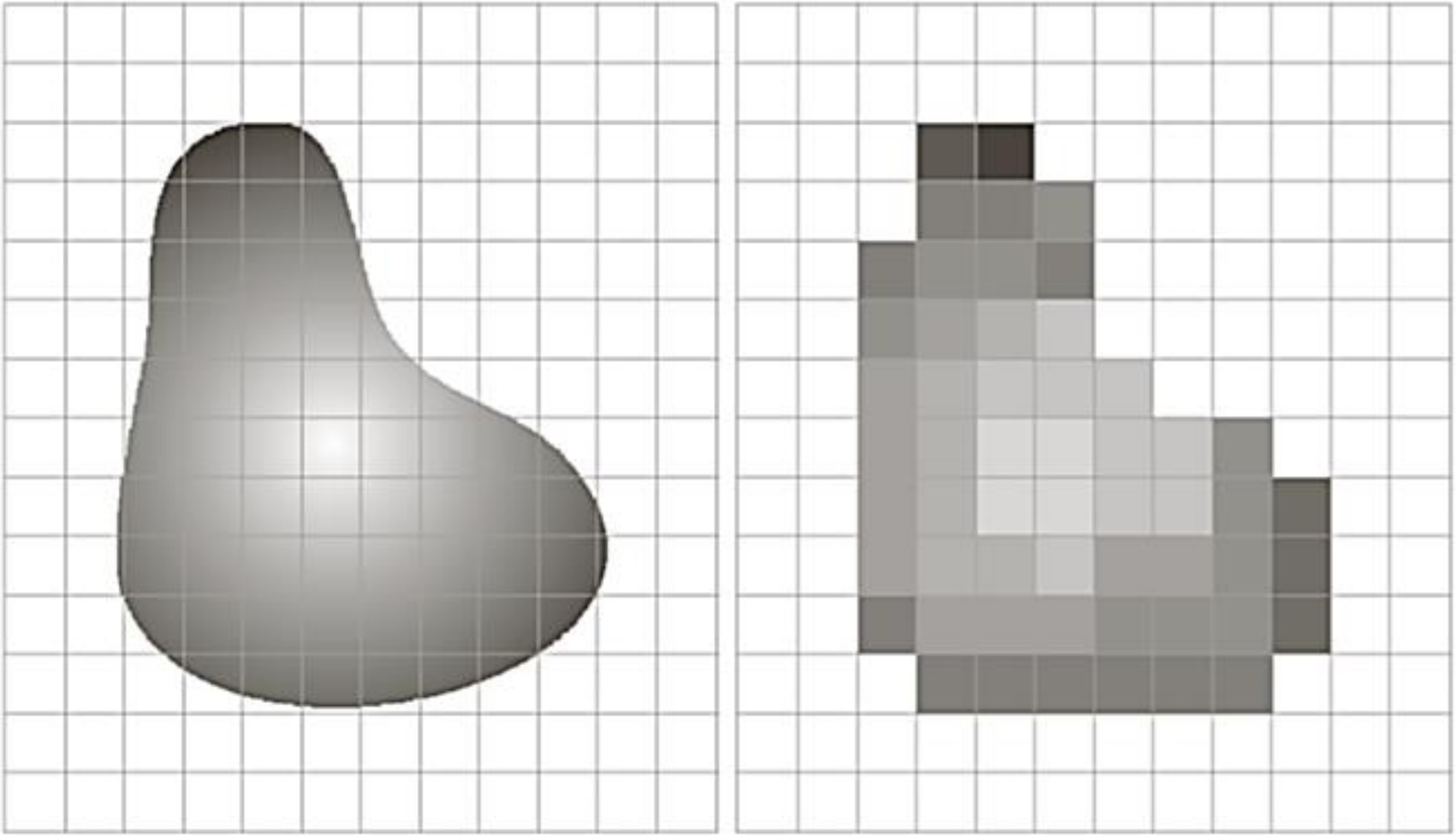
Image quantization

- Converting the sampled analog value into a discrete integer values.
- Digitizing the amplitude values is called **quantization**.



a b
c d

FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.



a b

FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Representing digital images

- Let f(s,t) represent a continuous image function of two continuous variables, *s* and *t*.
- We convert this function into a *digital image* by sampling and quantization, as explained in the previous section.
- Suppose that we sample the continuous image into a 2-D array f(x,y), containing *M* rows and *N* columns, where x and y are discrete spatial coordinates or spatial variables.

- In equation form, we write the representation of an M*N numerical array as

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix}$$

- In some discussions it is advantageous to use a more traditional matrix notation to denote a digital image and its elements:

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

- This digitization process requires that decisions be made regarding the values for *M*, *N*, and for the number, *L*, of discrete intensity levels.
- *M* and *N* have to be **positive integers**.
- However, due to storage and quantizing hardware considerations, the number of intensity levels typically is an integer power of 2:

$$L = 2^k$$

- Intensity levels are [0, L-1]

Some definitions

- **Dynamic range** – it is the ratio of the maximum measurable intensity to the minimum detectable intensity in the system.
- Upper limit is determined by **saturation**.
- Lower limit by **noise**.
- **Image contrast** – it is the difference in intensity between the highest and lowest intensity levels in an image.
- Usually, higher dynamic range images, have higher contrast and lower dynamic range indicates dull images with lower contrast.

- No. of bits required to store a digital image is

$$b = M * N * k$$

- For square images, M = N therefore,

$$b = N^2 * k$$

- Image with k bits has 2^k intensity levels.
- 8 bit image has 256 possible intensity levels.

Basic relationship between pixels

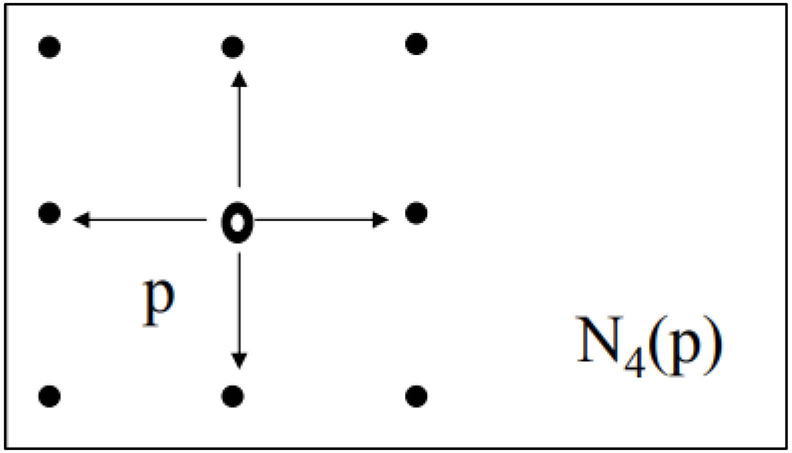
- Neighborhood
- Adjacency
- Connectivity
- Paths
- Regions and boundaries

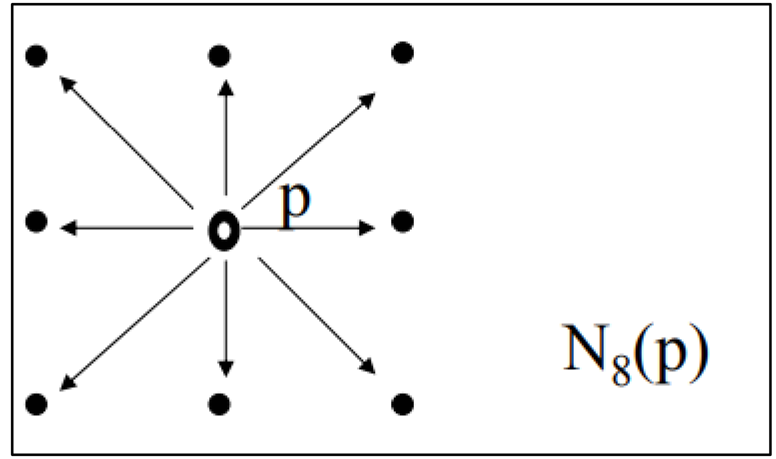
Neighbors of a Pixel

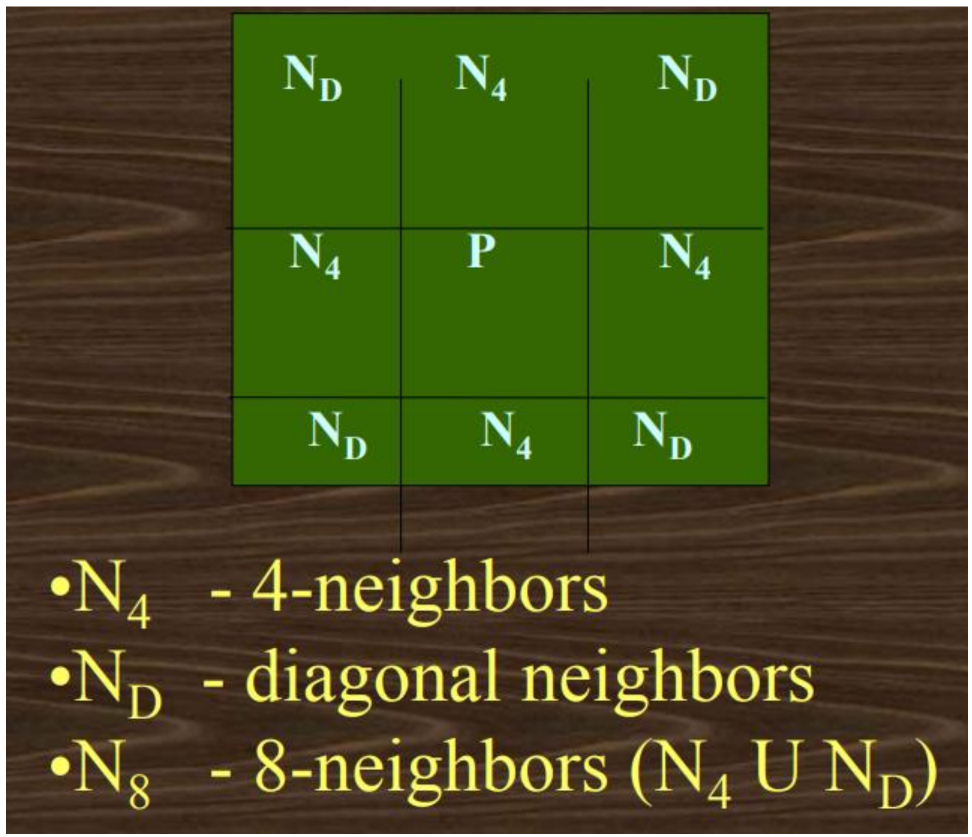
- A pixel *p* at coordinates (x,y) has four *horizontal* and *vertical* neighbors whose coordinates are given by

  (x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)
- This set of pixels, called the 4-*neighbors* of *p*, is denoted by $\mathbf{N_4(p)}$.
- The four *diagonal* neighbors of *p* have coordinates

  (x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)
- They are denoted by $\mathbf{N_D(p)}$.
- These points, together with the 4-neighbors, are called the 8-*neighbors* of *p*, denoted by $\mathbf{N_8(p)}$.


p
N4(p)


p
N8(p)


ND
N4
ND
N4
P
N4
ND
N4
ND
•N4 - 4-neighbors
•ND - diagonal neighbors
•N8 - 8-neighbors (N4 U ND)

What is image enhancement?

Process an image so that the result is more suitable than the original image for a specific application.

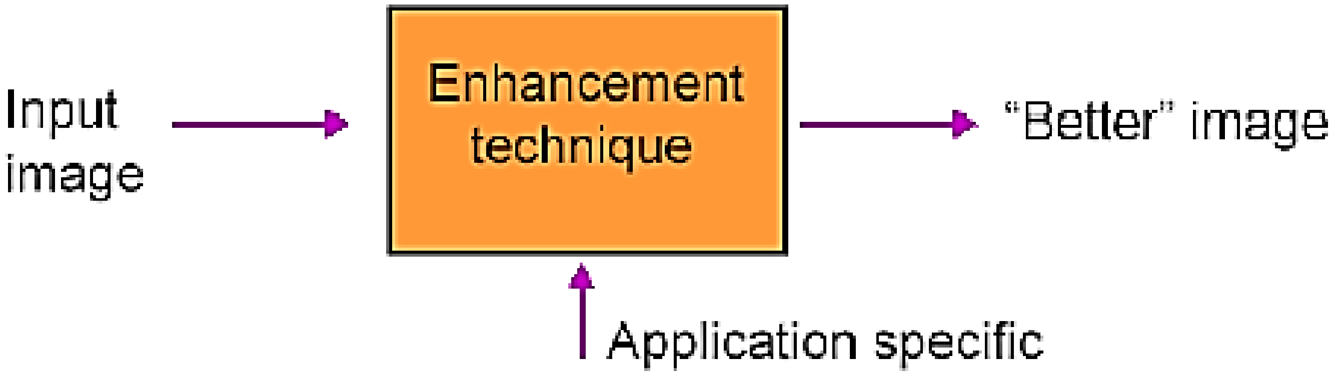


Image enhancement

- Low level processing
- Bring out detail that is obscured
- Highlight certain features of interest of an image
- It performs primitive operations such as Noise removal , contrast enhancement and image sharpening
- Inputs and outputs are images

Image Enhancement Approaches

- These approaches can be classified as

1) Spatial domain approaches

- Involves direct manipulation of pixels in an image

2) Frequency domain approaches

- Involves modifying the Fourier transform of an image

Spatial Domain Enhancement

- The approaches are further classified as

1) Intensity transformation

- Modify the gray level of a pixel independent of the nature of its neighbors e.g. thresholding, gray level transformation
- Called as Point processing

2) Spatial filtering

- Small sub-images (masks) are used in local processing to modify each pixel in the image to be enhanced e.g. image sharpening, edge detection.
- Called as Neighborhood Processing

Gray Level Transformations

- These are among the simplest of all image enhancement techniques.
- The transformation (T) maps pixel value r to a pixel value s and is denoted by

$$s = T(r)$$

Basic intensity transformations

- Three types of basic functions often used for image enhancements are
  - Linear (negative and identity transformations)
  - Logarithmic (log and inverse-log transformations)
  - Power-law (nth power and nth root transformation)

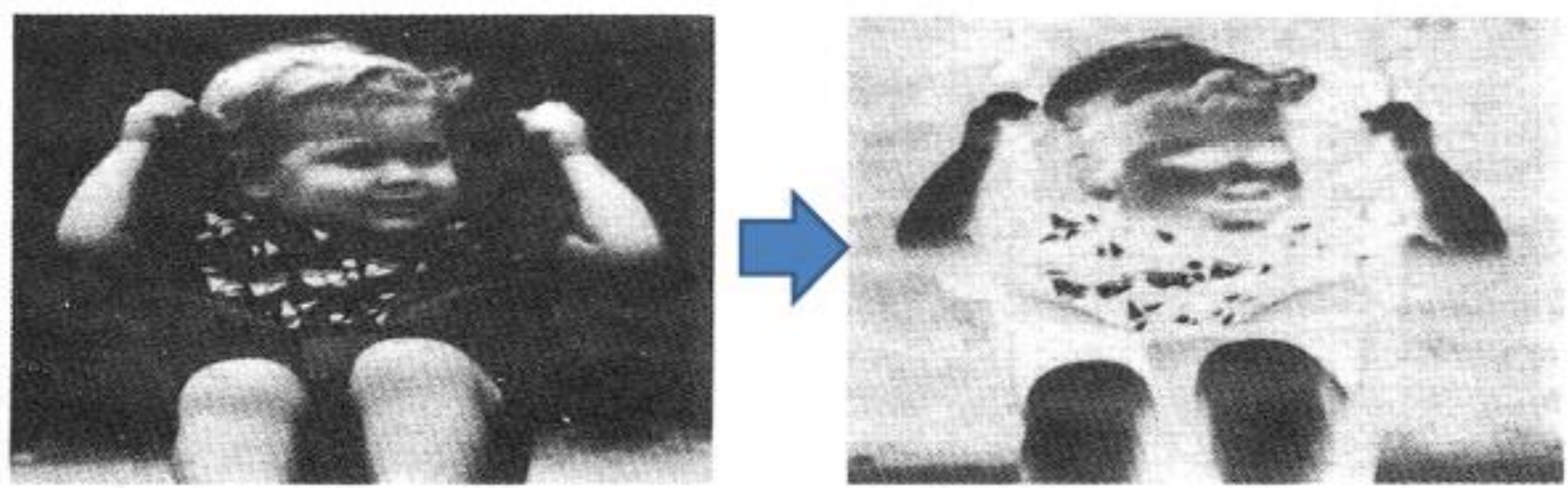
Image negative

- The negative of an image with gray level in the range [0,L-1] is obtained by using

$$s = L - 1 - r$$

- Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative.
- This is particularly suited for enhancing white or gray detail embedded in the dark regions of the image, especially when the black areas are dominant.

Image negative



(b) (c)

Image negative

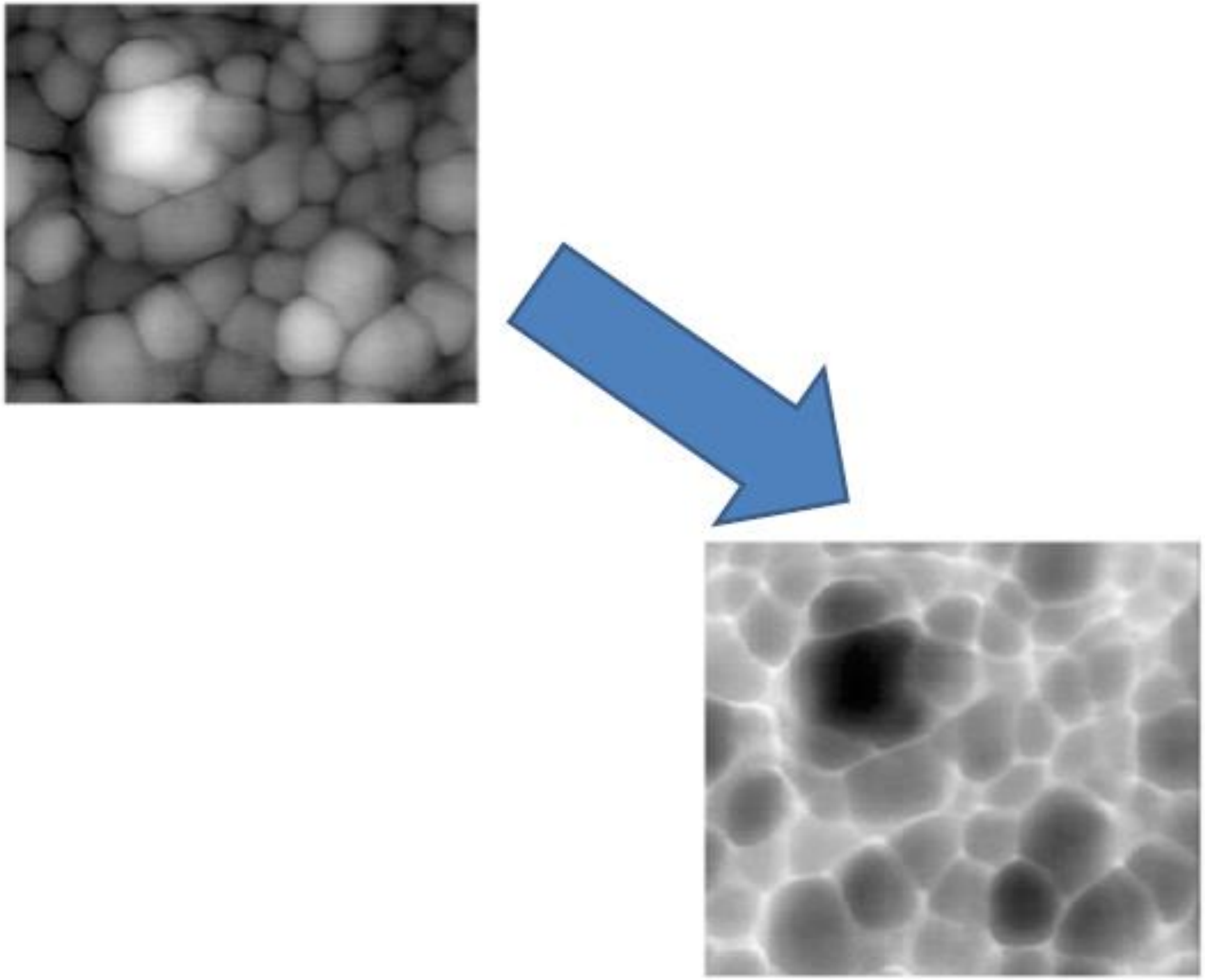


Image negative

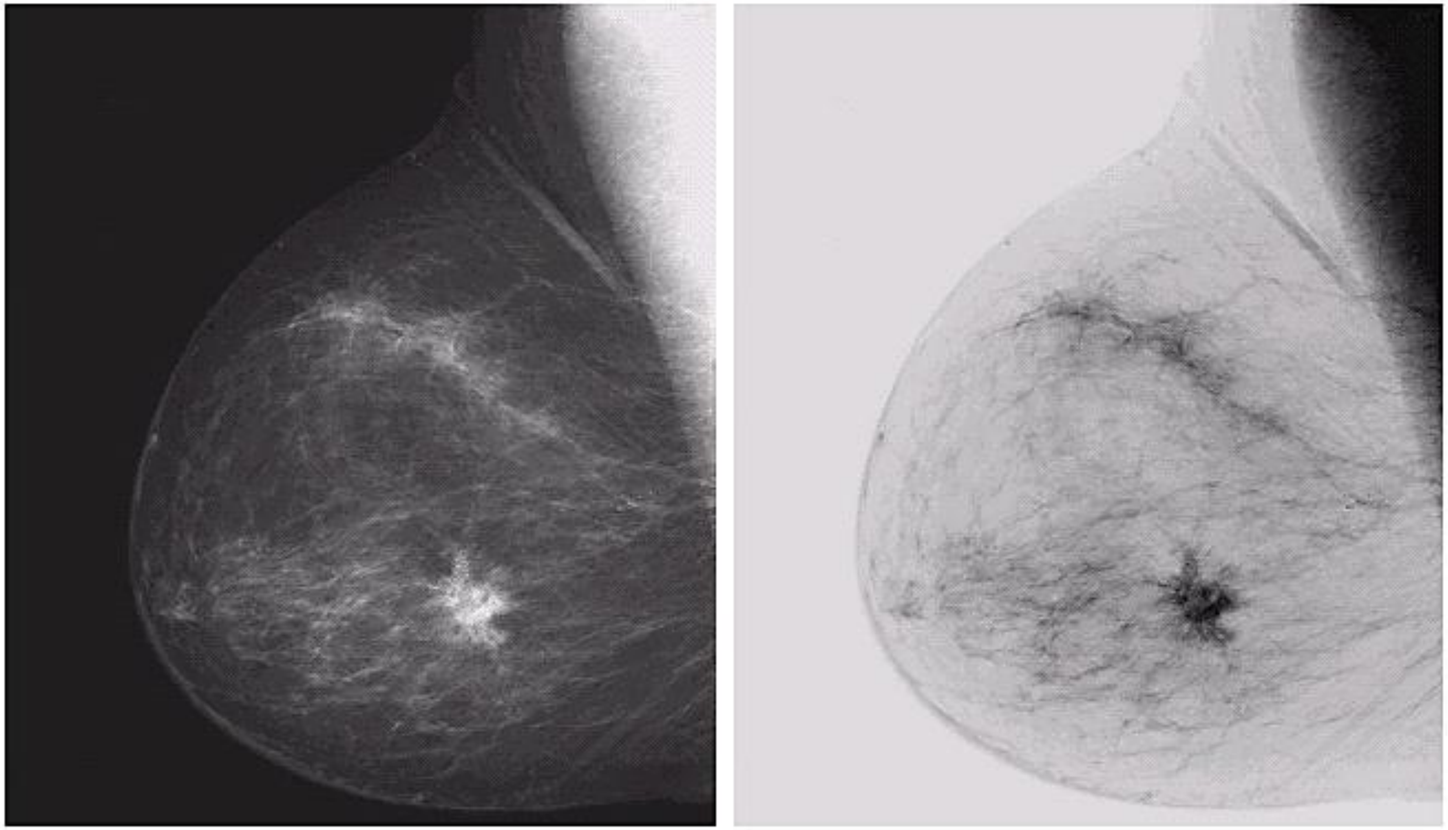
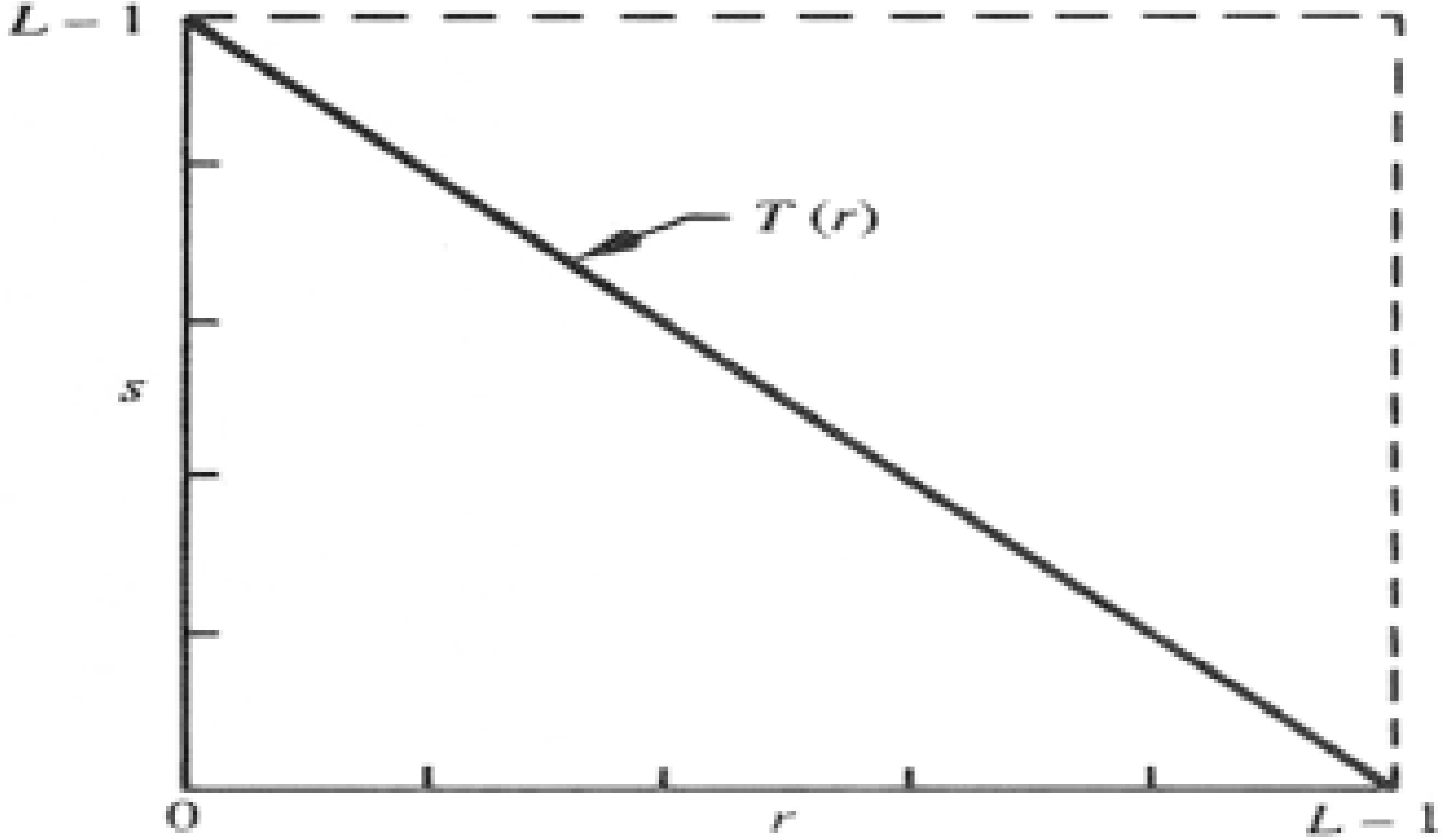


FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)



(a)

Log transformation

- The general form of the log transformation is

$$s = c \log(1 + r)$$

where c is a constant and $r \geq 0$.

- The transformation maps a narrow range of low gray level input values to wider range of output levels and the opposite for higher values of input levels.
- Used to expand the values of dark pixels while compressing the brighter pixel values.
- The opposite is true for inverse-log transformation.
- It compresses the dynamic range of images.

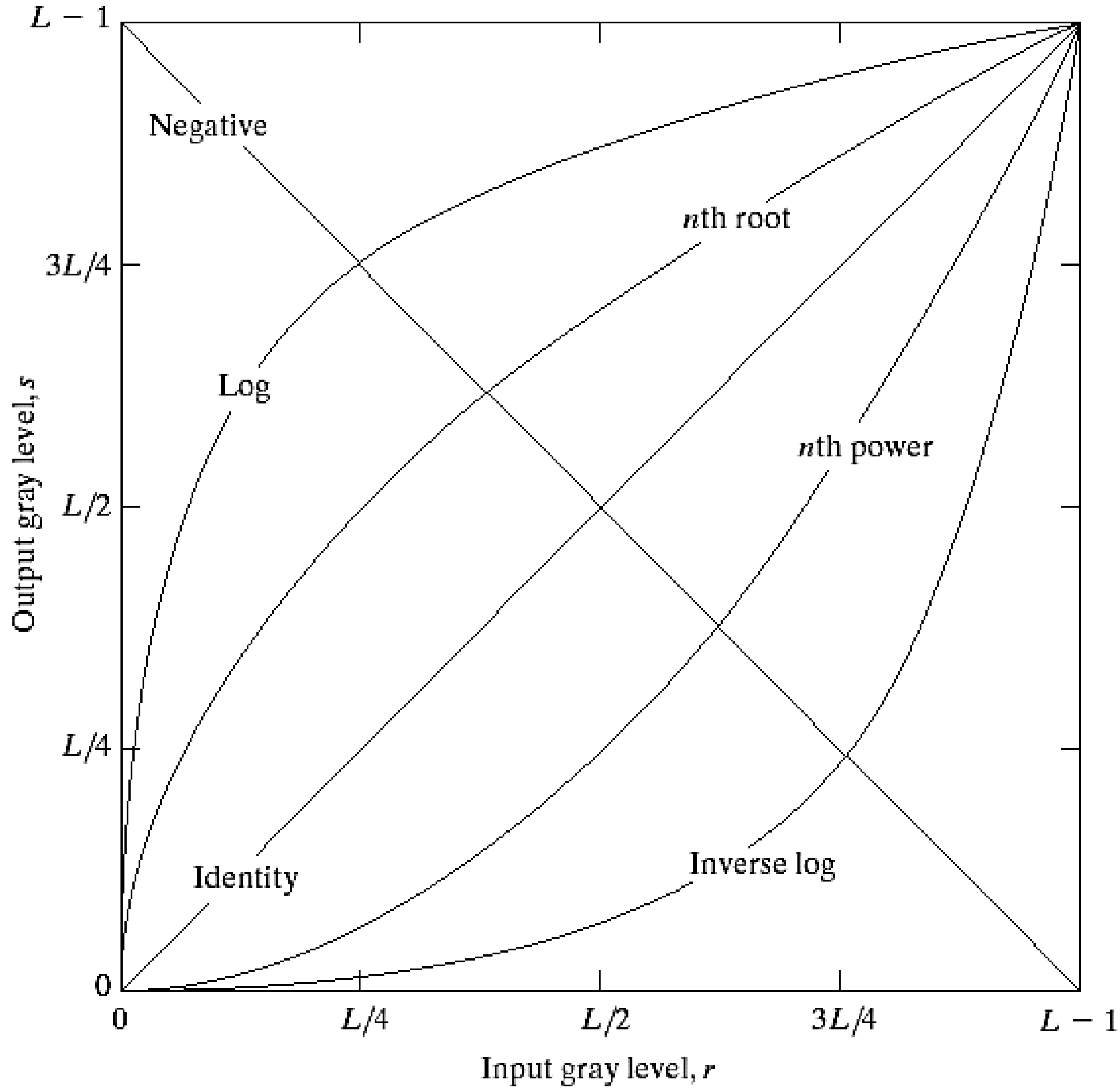
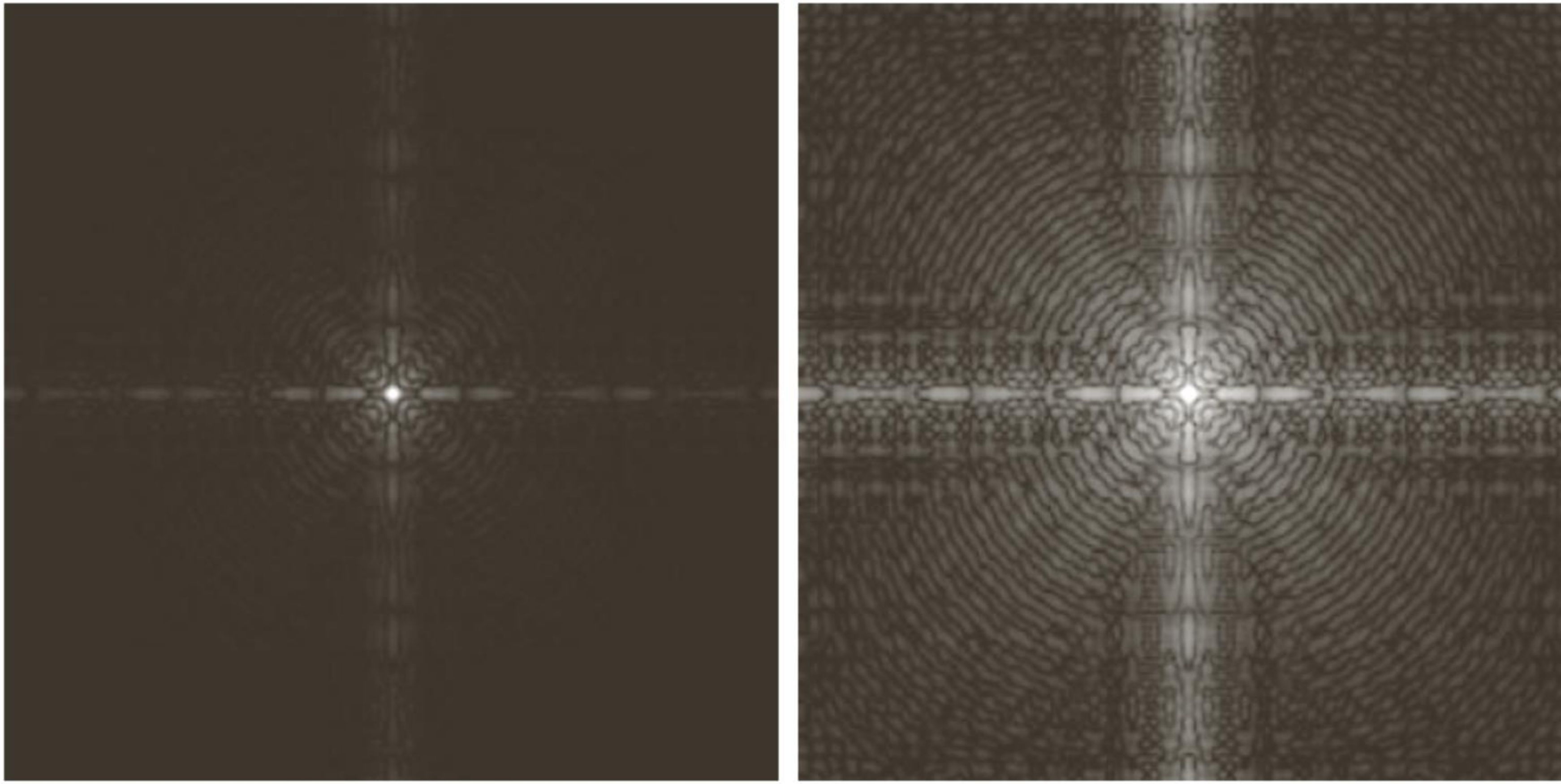


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



a b

FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation in Eq. (3.2-2) with $c = 1$.

Power-Law (Gamma) transformations

- The transformation has the basic form of

$$s = c\, r^{\gamma}$$

where c and γ are constants.

- NOTE: The above is sometimes written as

$$s = c\,(r + \epsilon)^{\gamma}$$

to account for an offset (output when input is zero)

- $\gamma < 1$ maps a narrow range of dark input values to a wider range of output values.
- $\gamma > 1$ does the opposite.

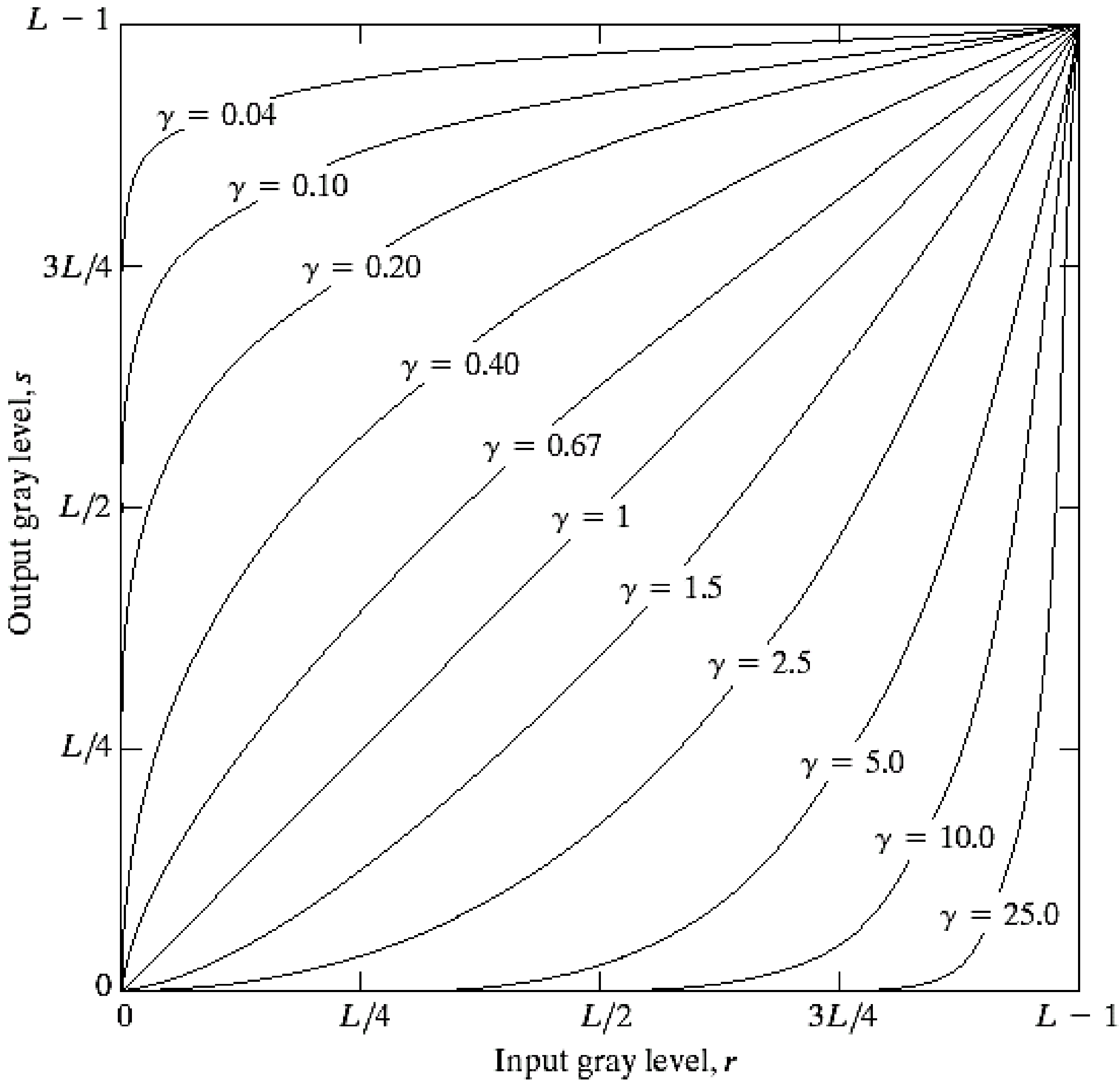
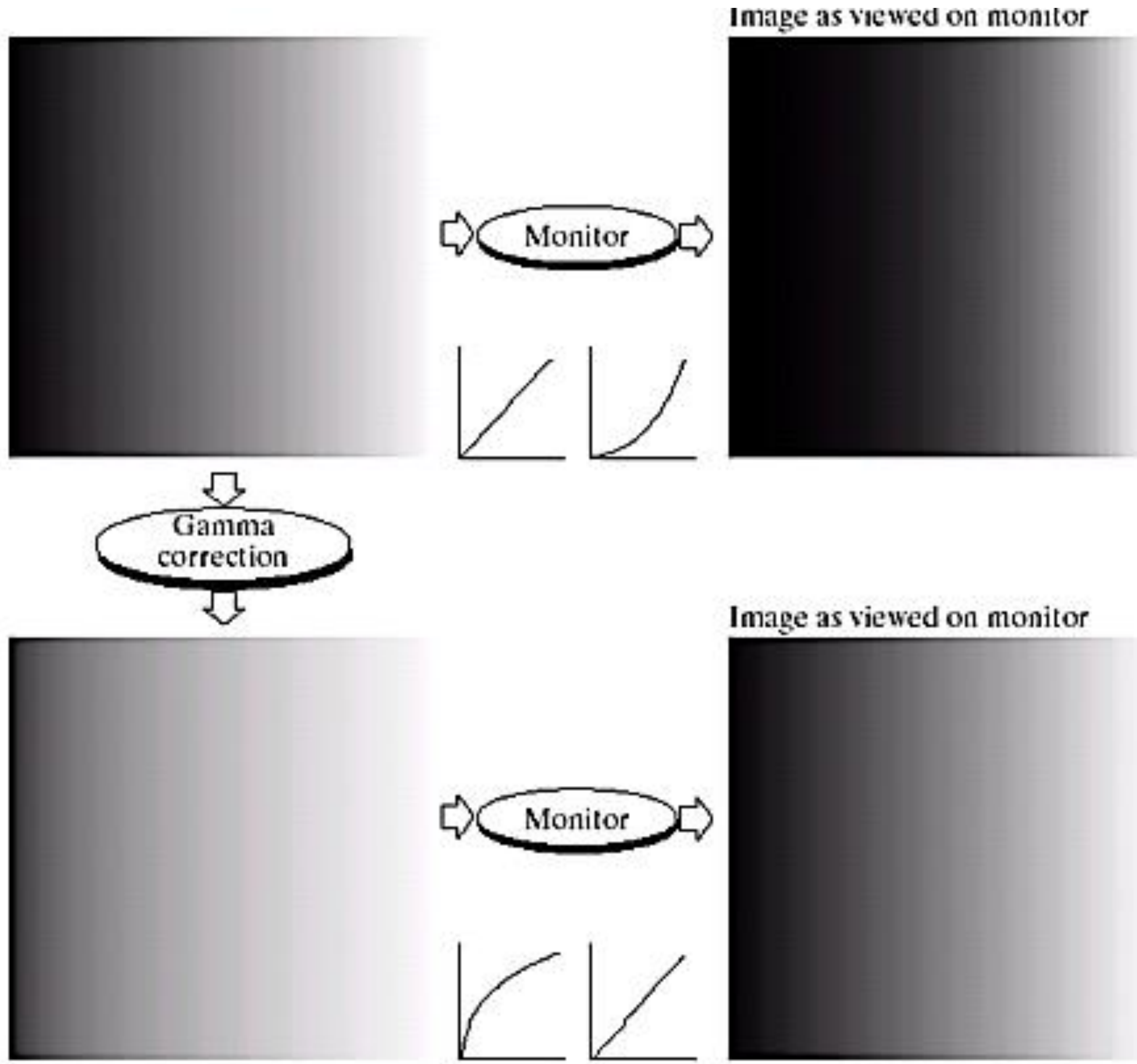


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Gamma corrections

- The process to correct the power law response phenomena is called gamma correction.
- Which is used by most of the display and printing devices.
- All we need to do is pre process the image before sending it to the monitor by performing power law transformations.
- We have to use device dependent value of gamma.



a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

Power law for contrast manipulation

a b
c d

FIGURE 3.8 (a) Magnetic resonance (MR) image of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

a b
c d

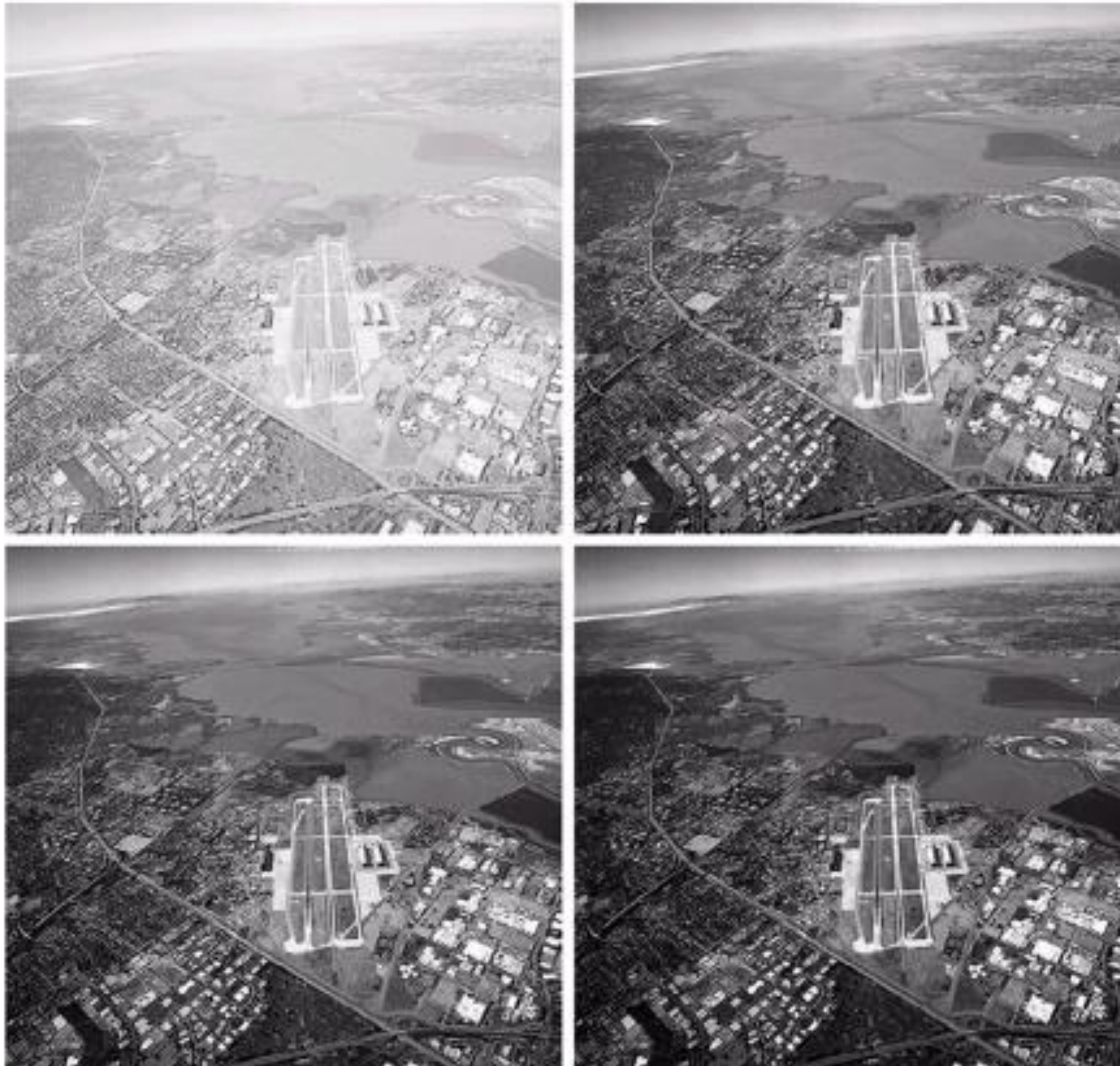


FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

Piecewise-Linear transformations

- These functions are arbitrarily complex.
- The disadvantage of piecewise functions is that their specification requires considerably more user inputs.
- Examples:

– Contrast stretching

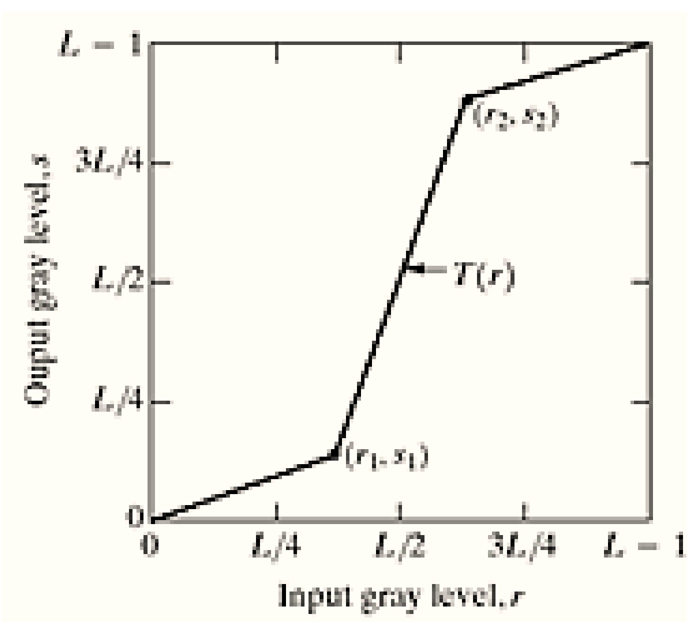
– Gray level slicing

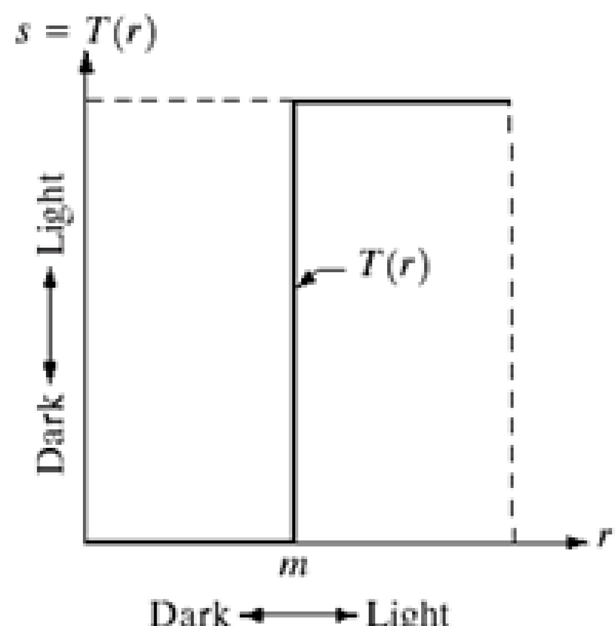
– Bit plane slicing

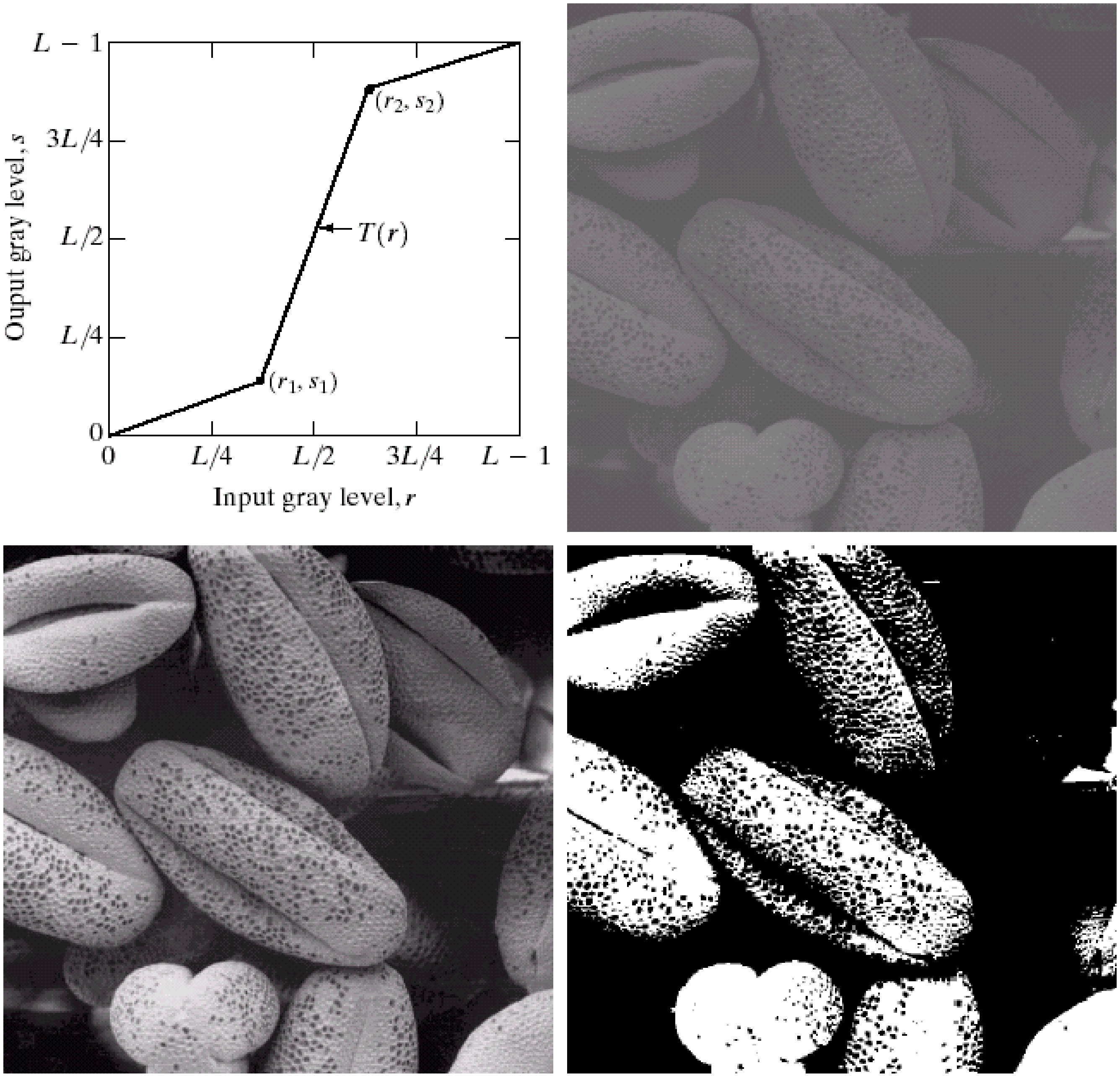
Contrast stretching

- The low-contrast image can be caused due to poor illumination, lack of dynamic range of the sensor, wrong setting of the lens etc.
- Contrast stretching is the process that expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device.
- Increase dynamic range of the grey levels.

- The locations (r1,s1) and (r2,s2) control the shape of the transformation
- If r1=s1 and r2=s2 no change in grey level.
- r1=r2 , s1=0 and s2=L-1 then transformation becomes a thresholding function that creates binary image.







a	b
c	d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

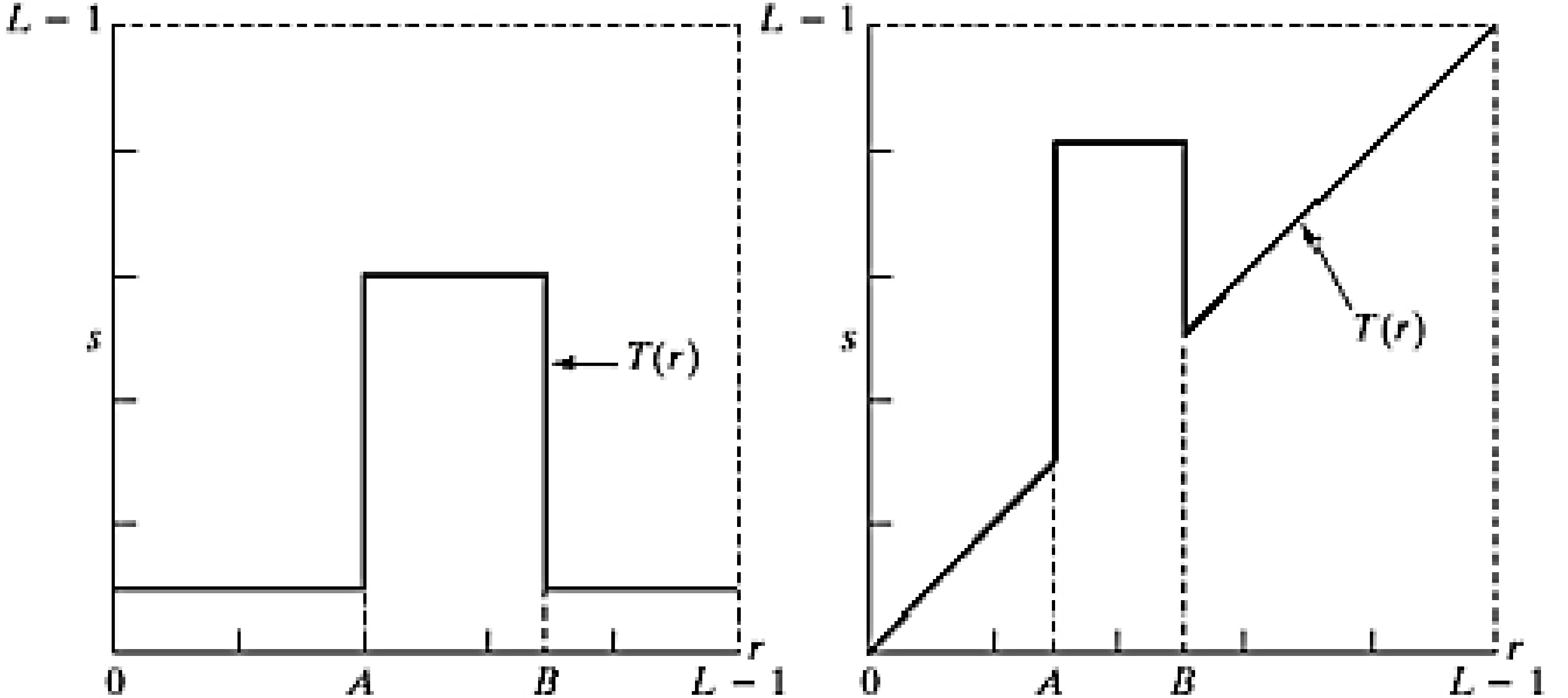
- In general $r1<=r2$ and $s1<=s2$ so the transformation function is single valued and monotonically increasing.
- If $(r1,s1)=(r_{min},0)$ and $(r2,s2) = (r_{max},L-1)$ then input image is stretched to the full range.
- Fig (d) shows the result of thresholding where $(r1,s1)=(m,0)$ and $(r2,s2) = (m,L-1)$ where is the mean intensity of the image.


Intensity level slicing

- Also called as grey level slicing
- Used for Highlighting a specific range of grey levels in an image.
- This is achieved by:

1. Display a high value for all gray levels in the range of interest and a low value for all other gray levels.

2. Brighten (or darken) the desired range of gray levels but leave all other intensities unchanged I image.

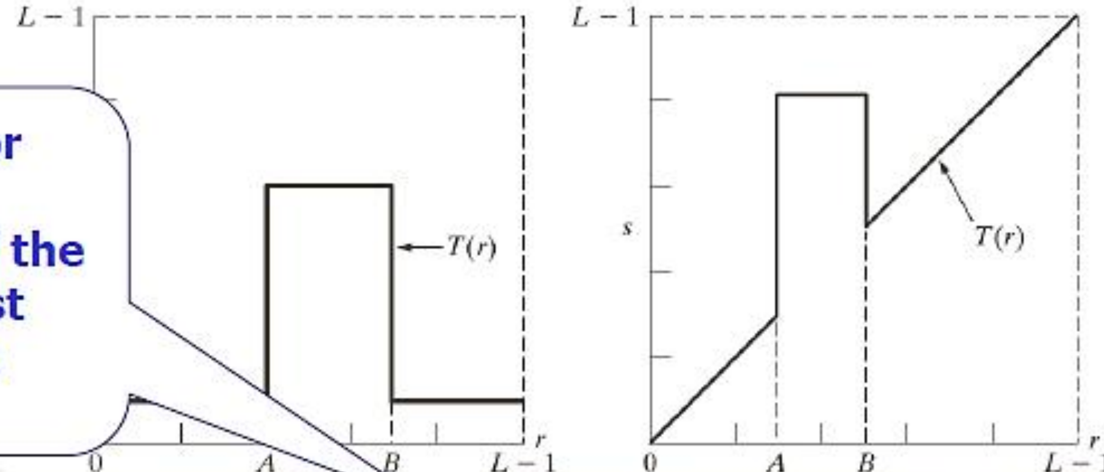


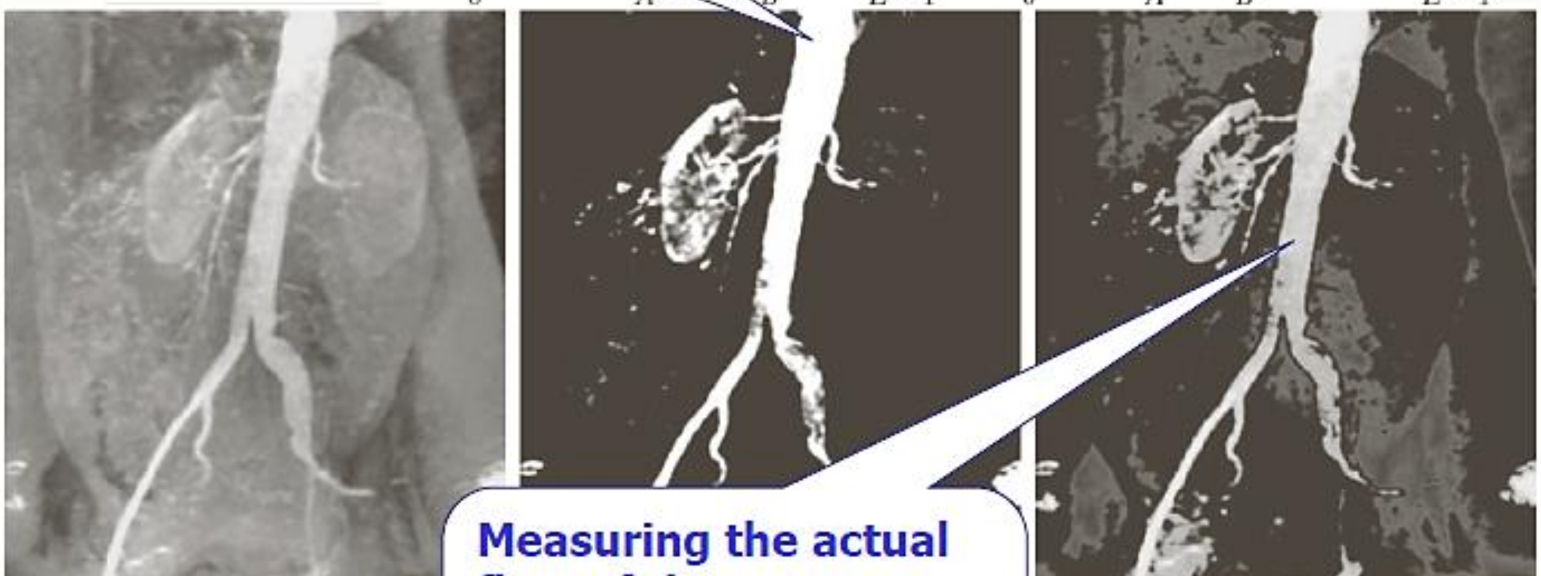
L − 1
s
T(r)
r
0
A
B
L − 1
L − 1
s
T(r)
r
0
A
B
L − 1

a b

FIGURE 3.11 (a) This

Highlight the major blood vessels and study the shape of the flow of the contrast medium (to detect blockages, etc.)





Measuring the actual flow of the contrast medium as a function of time in a series of images

a b c

FIGURE 3.12 (a) Aortic angiog mation of the type illustrated in Fig. 3.11(a), with the range of inte end of the gray scale. (c) Result of using the transformation in Fig ack, so that grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

Bit-plane slicing

- Highlighting the contribution made to the total image appearance by specific bits.
- For an 8-bit image, eight 1-bit planes images, ranging from bit plane 0 (LSB) to bit plane 7 (MSB) are constructed.
- NOTE: Higher bits contain the majority of the visually significant data and lower bit planes contribute to more subtle details in the image.

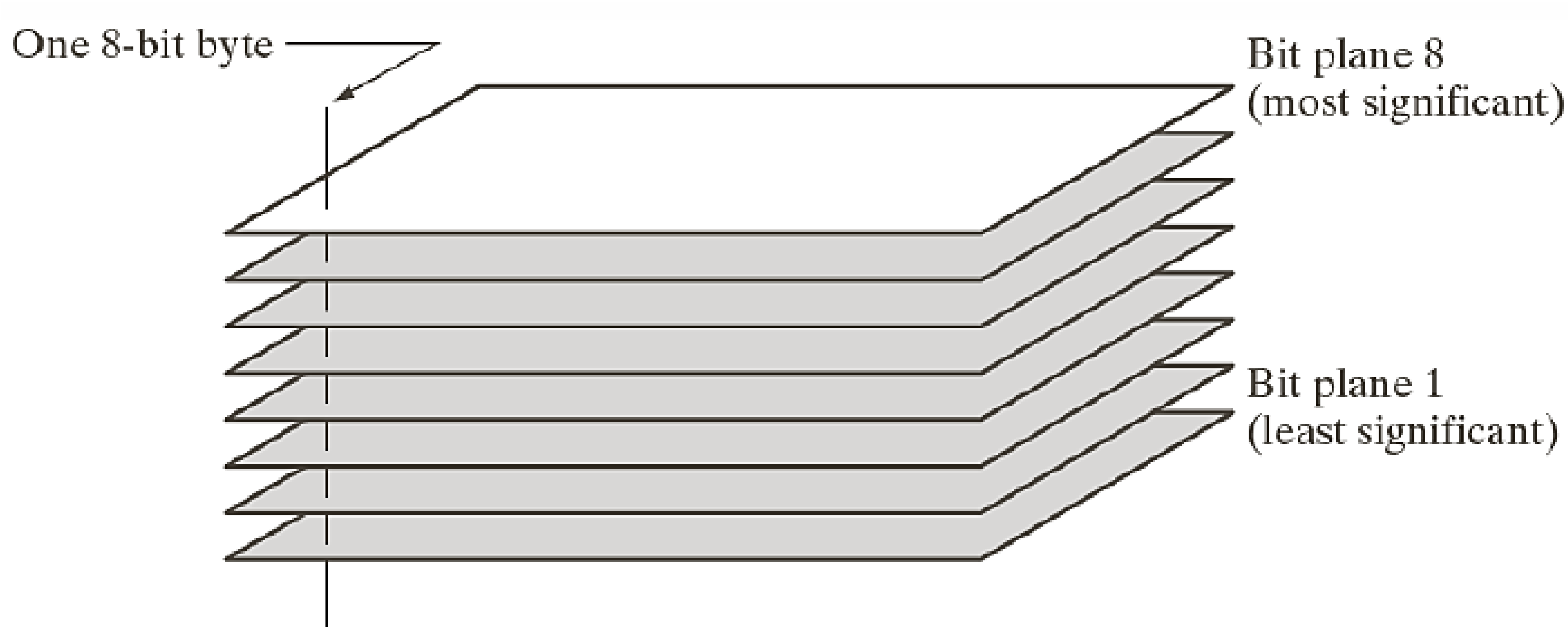


FIGURE 3.13
Bit-plane representation of an 8-bit image.

a	b	c
d	e	f
g	h	i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500×1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

a b c

FIGURE 3.15 Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5. Compare (c) with Fig. 3.14(a).

- Used to analyze the relative importance of each bit in image.
- Also used in compression.

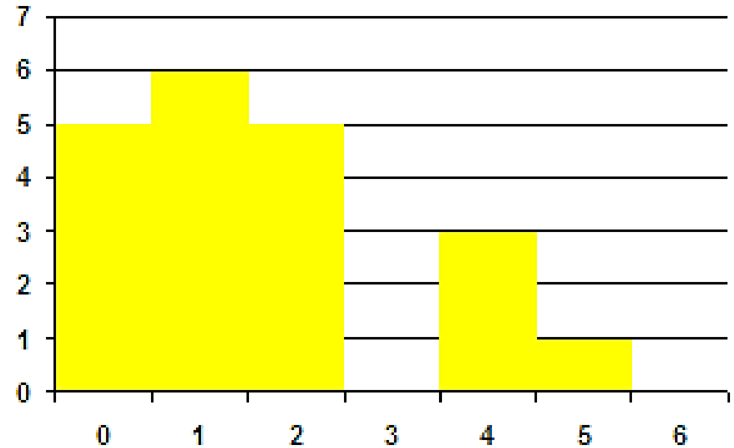
Histogram and Histogram processing

Histogram

- Find out the number of pixels having grey level r_k
- where r_k is in the range 0 to L-1
- The histogram shows how many times a particular grey level (intensity) appears in an image.

0	1	1	2	4
2	1	0	0	2
5	2	0	0	4
1	1	2	4	1

image



histogram

- **Histogram** is a discrete function denoted as

$$h(r_k)=n_k$$

Where r_k is the k^{th} gray level and

n_k is the number of pixels in the image having gray level r_k

- Histogram of the image tells us whether it is dark, light ,low contrast or high contrast. We can enhance image by applying appropriate transformation function.
- Histogram provide image statistics which is useful for image compression and segmentation

Normalized Histogram

- Normalized histogram is given as

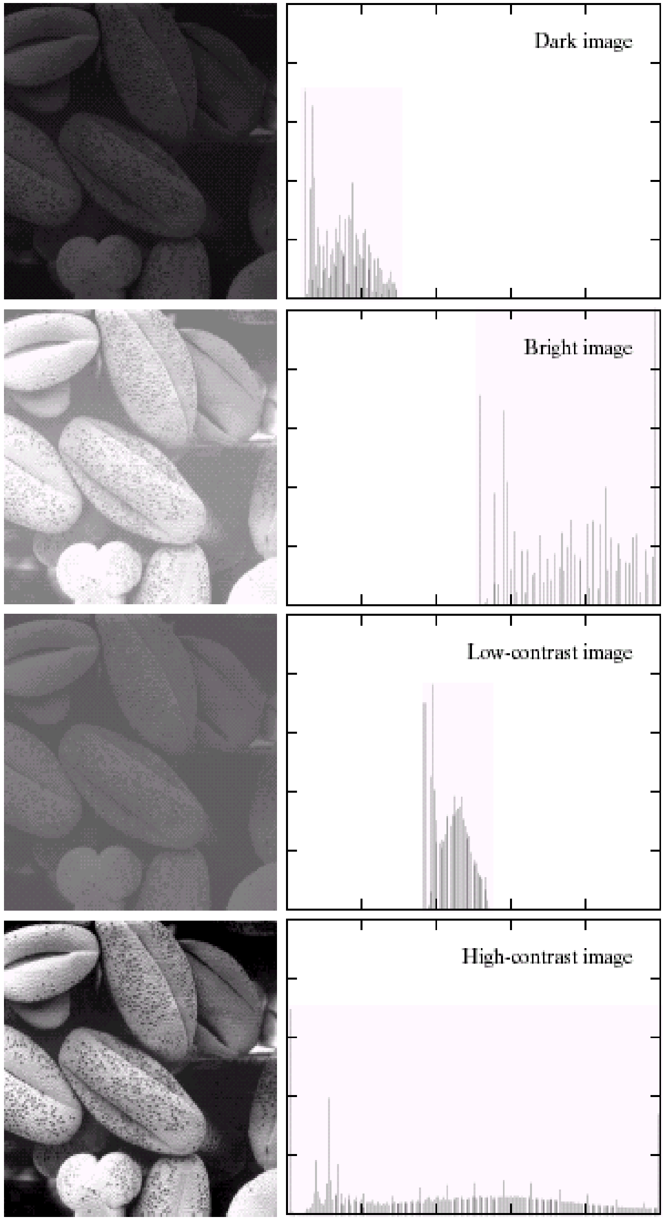
$$p(r_k) = n_k/n$$

where n is total number of pixels in image.

$$n = MXN$$

- $P(r_k)$ is an estimate of the probability of occurrence of intensity r_k in an image
- Sum of all components of normalize histogram =1
- Histogram is simple viewed graphically as a plot of $h(r_k)$ versus r_k OR $p(r_k)$ versus $r_{k.}$

Histogram and image appearance

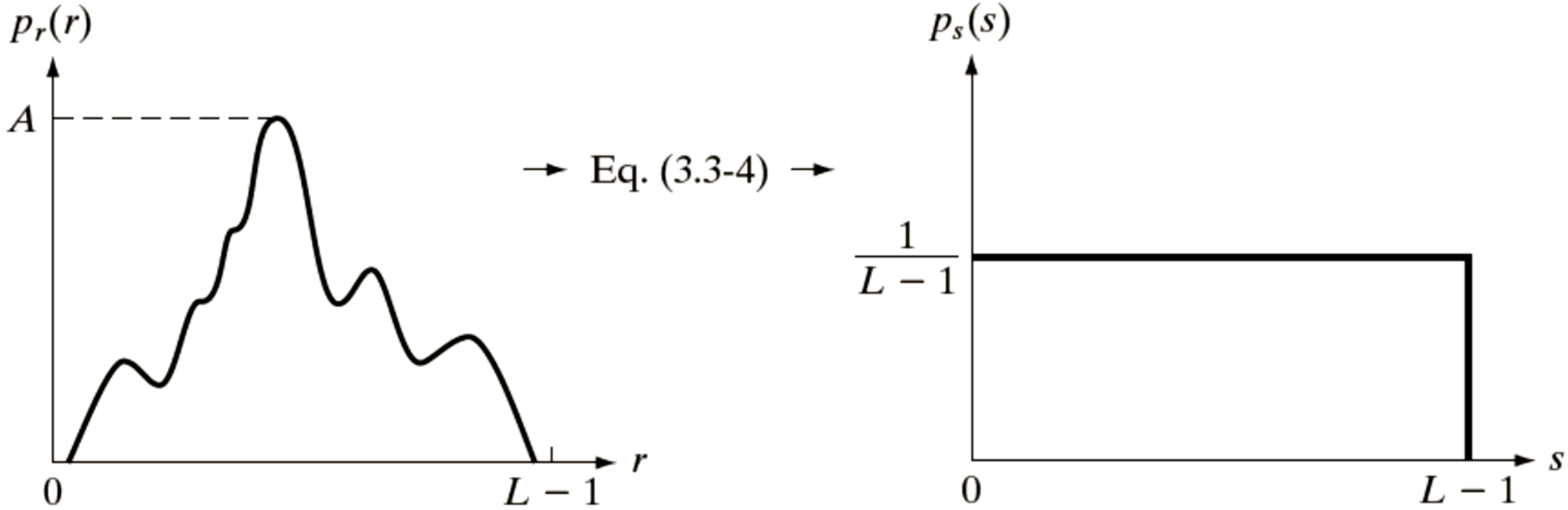


- Dark Image: components of the histogram are concentrated on the low (left) side of the gray scale.
- Bright Image: on the high (right) side of the gray scale.
- Poor Contrast: Narrow & will be centered towards the middle.
- High Contrast: Occupies full gray scale & components are uniformly distributed.

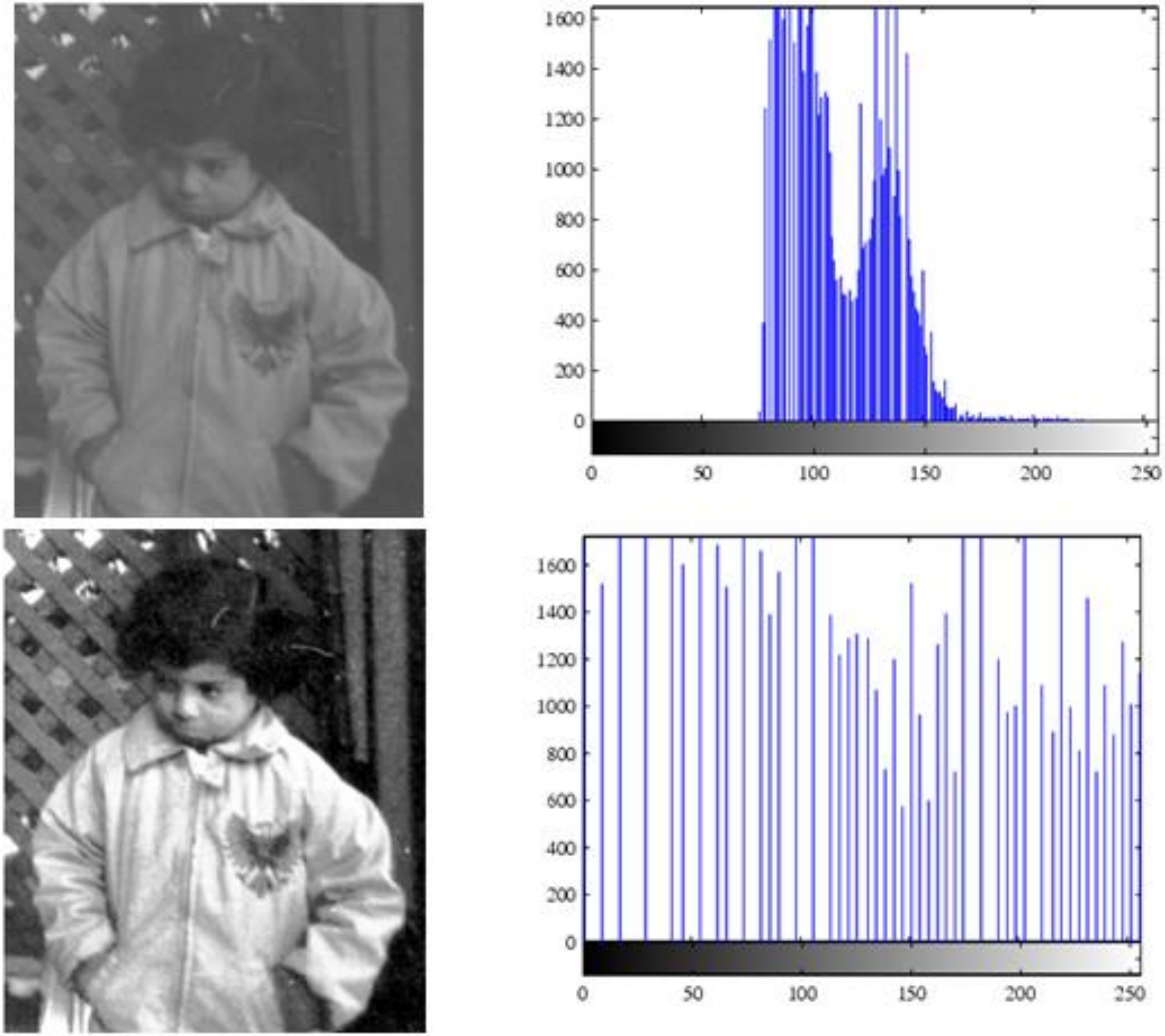
Histogram Processing

Histogram equalization

- An image whose pixels tend to occupy the entire range of possible intensity levels and in addition tend to be distributed uniformly - Such image shows a great deal of gray level detail and has high dynamic range.
- It is possible to develop a transformation function that can automatically achieve this effect based only on information available in the histogram of the input image.
- It is called Histogram equalization.



- Performing a histogram equalization on the image spreads the peaks out while compressing other parts of the histogram by assigning the same, or very close, brightness values to those pixels that are few in number and have intermediate brightness.
- This equalization makes it possible to see minor variations within regions that appeared nearly uniform in the original image.


1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250
1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250

Histogram equalization

- Equalization operation can be represented as

$$s=T(r)$$

- Equalization function has two characteristics:

1. T(r) is a monotonically increasing function
2. $0<=T(r)<=L-1$ for $0<=r<=L-1$

- The probability of occurrence of gray level r_k in an image is

$$p_r(r_k) = n_k/n \quad k=0,1...L-1$$

Histogram equalization function

Continuous case:

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

Discrete values:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= (L-1)\sum_{j=0}^{k}\frac{n_j}{MN} = \frac{L-1}{MN}\sum_{j=0}^{k} n_j \qquad k=0,1,\ldots, L\text{-}1$$

L: number of grey levels in image (e.g., 256)

n_j: number of times j-th grey level appears in image

Example

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in following table.
Get the histogram equalization transformation function and give the $p_s(s_k)$ for each s_k.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

- Find the values of equalization transformation functions using the equation.

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7 \times 0.19 = 1.33 \quad \rightarrow 1$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7 \times (0.19 + 0.25) = 3.08 \quad \rightarrow 3$$

$$s_2 = 4.55 \quad \rightarrow 5 \qquad s_3 = 5.67 \quad \rightarrow 6$$

$$s_4 = 6.23 \quad \rightarrow 6 \qquad s_5 = 6.65 \quad \rightarrow 7$$

$$s_6 = 6.86 \quad \rightarrow 7 \qquad s_7 = 7.00 \quad \rightarrow 7$$

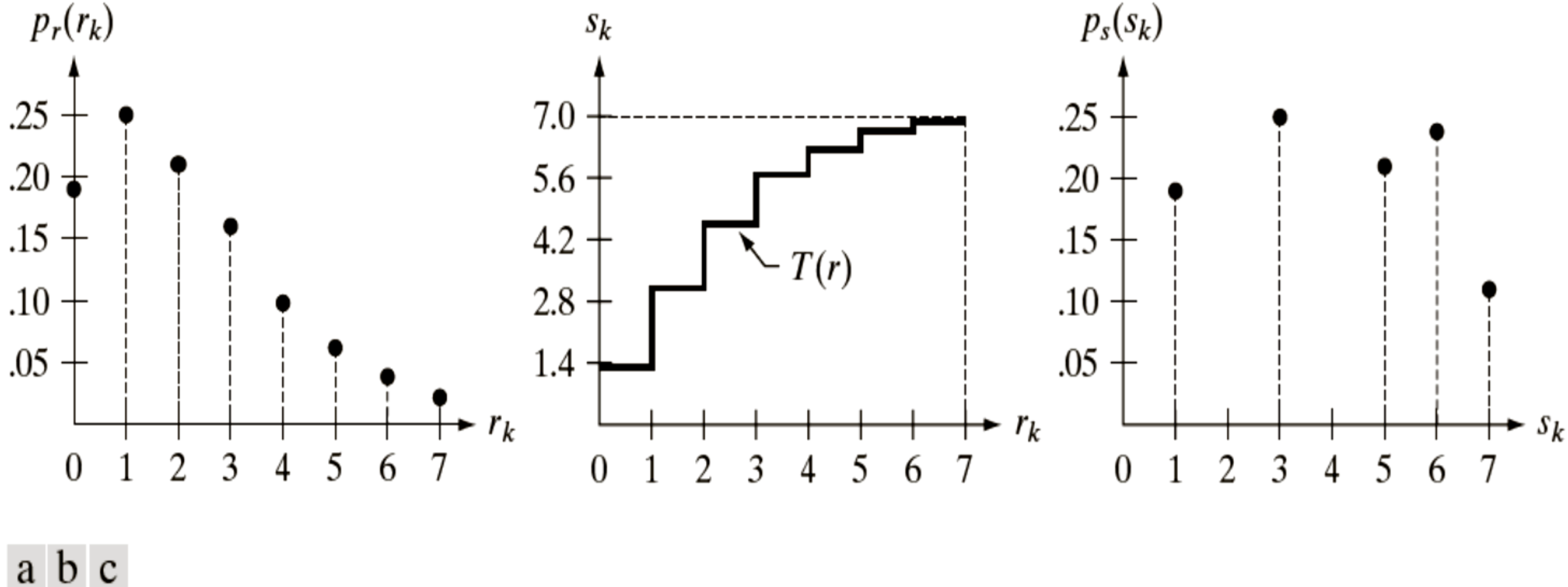
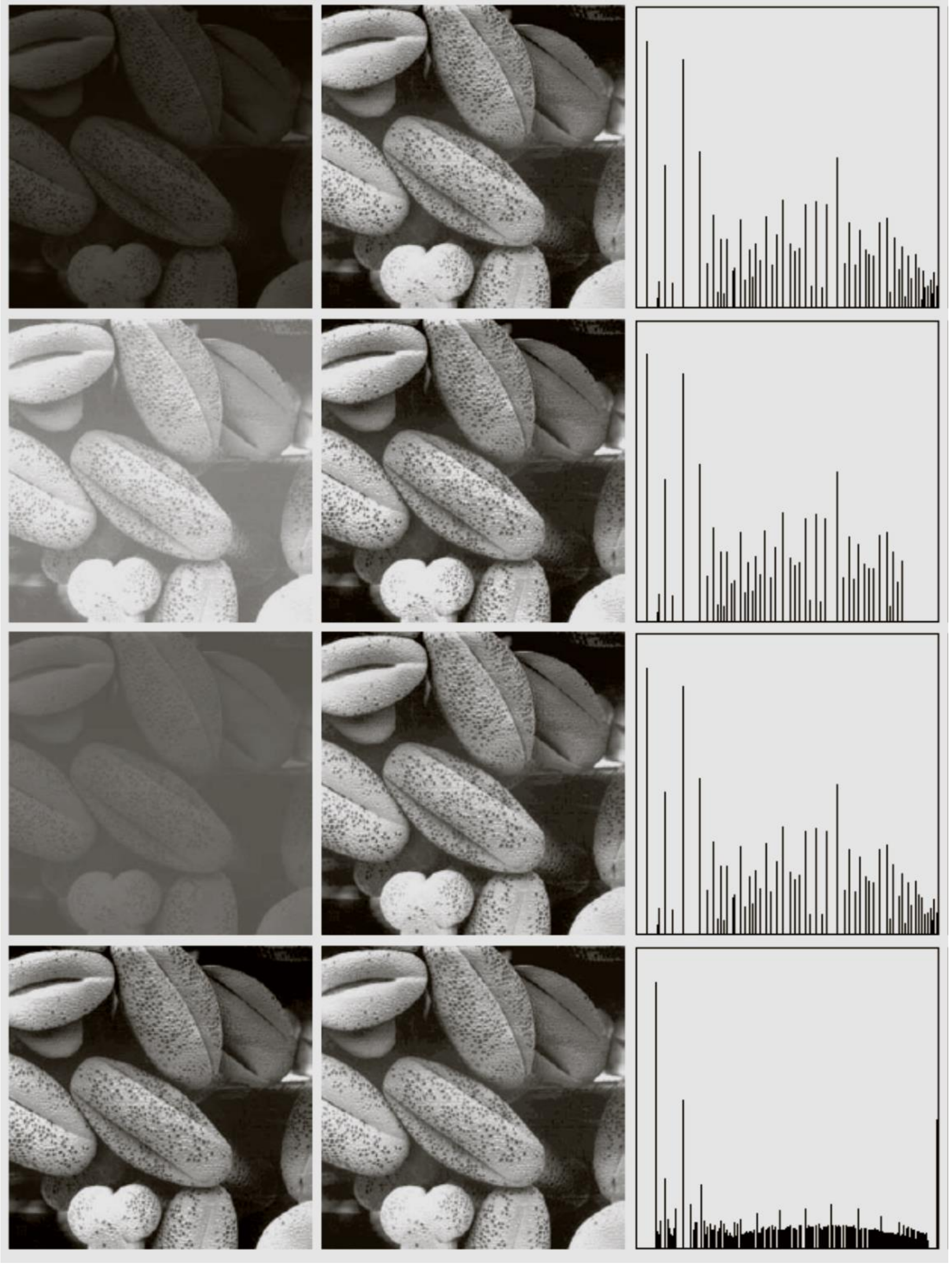
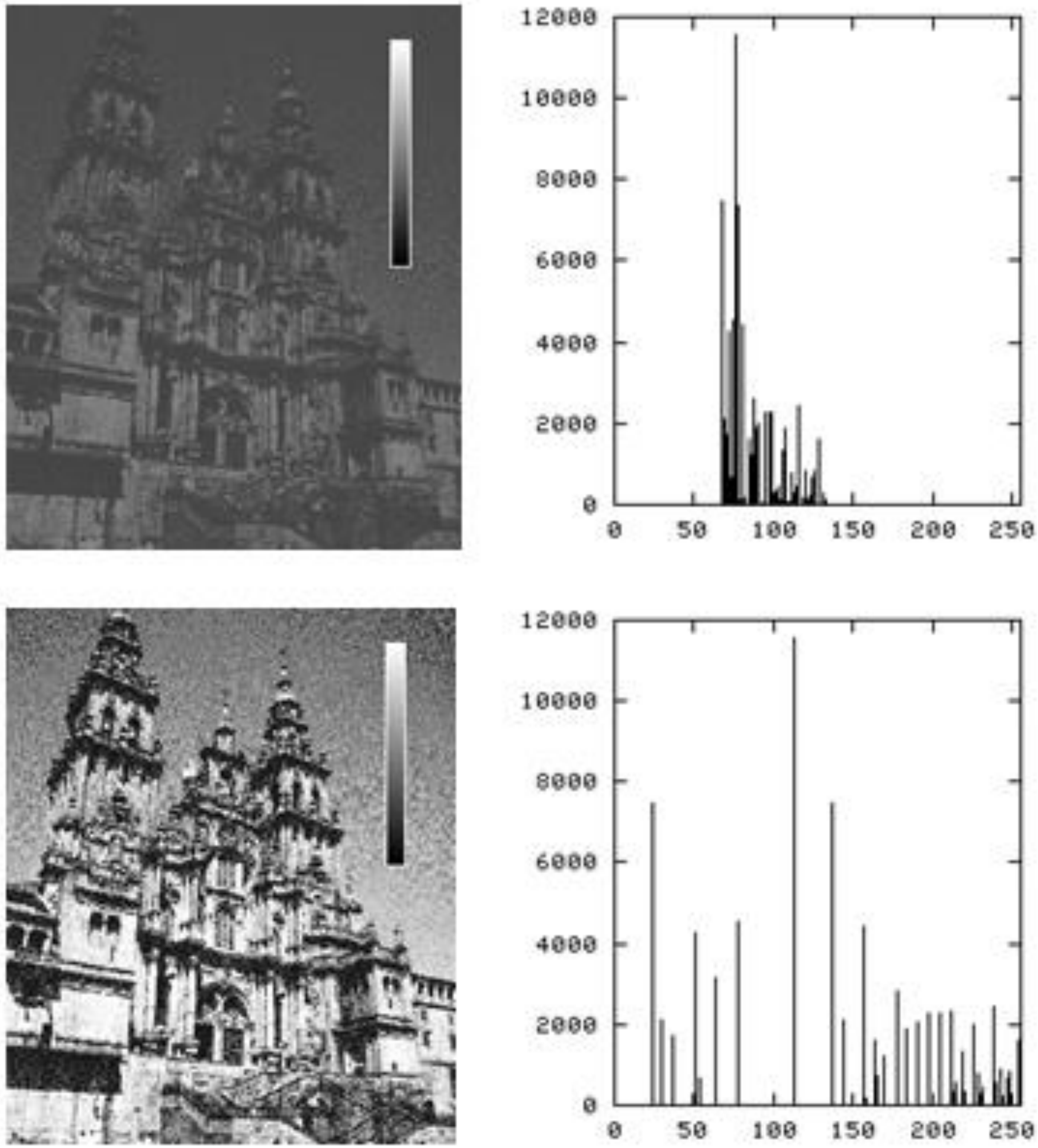


FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.




12000
10000
8000
6000
4000
2000
0
0 50 100 150 200 250
12000
10000
8000
6000
4000
2000
0
0 50 100 150 200 250

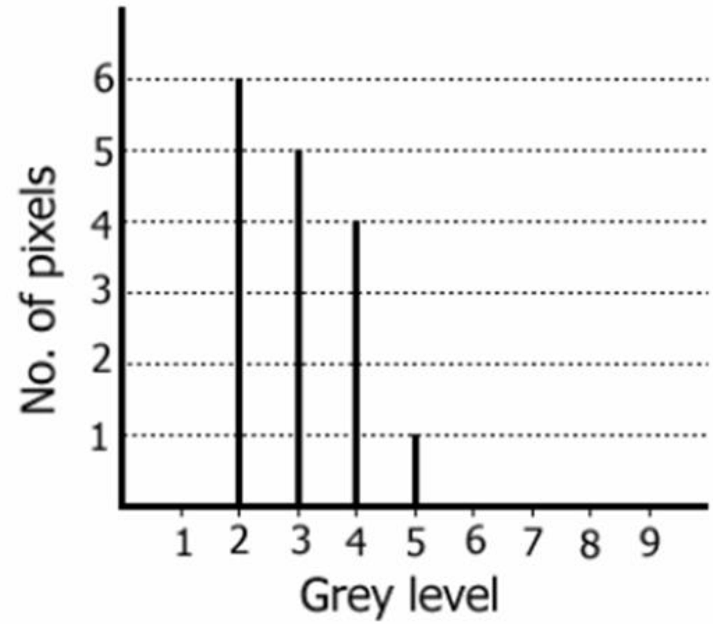
Example 2

HE-method

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Grey level=[0,9]



Original Histogram

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n}(L-1)$$

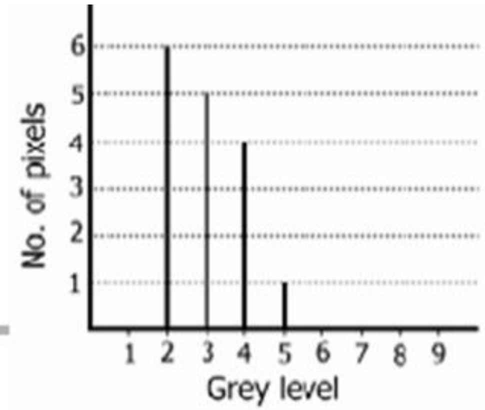
2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

HE-method

Grey level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
Σn_j	0	0	6	11	15	16	16	16	16	16
$\sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6/16	11/16	15/16	16/16	16/16	16/16	16/16	16/16
$s_k = \sum_{j=0}^{k} \frac{n_j}{n}(L-1)$	0	0	3	6	8	9	9	9	9	9

Note that L=10.

HE-method

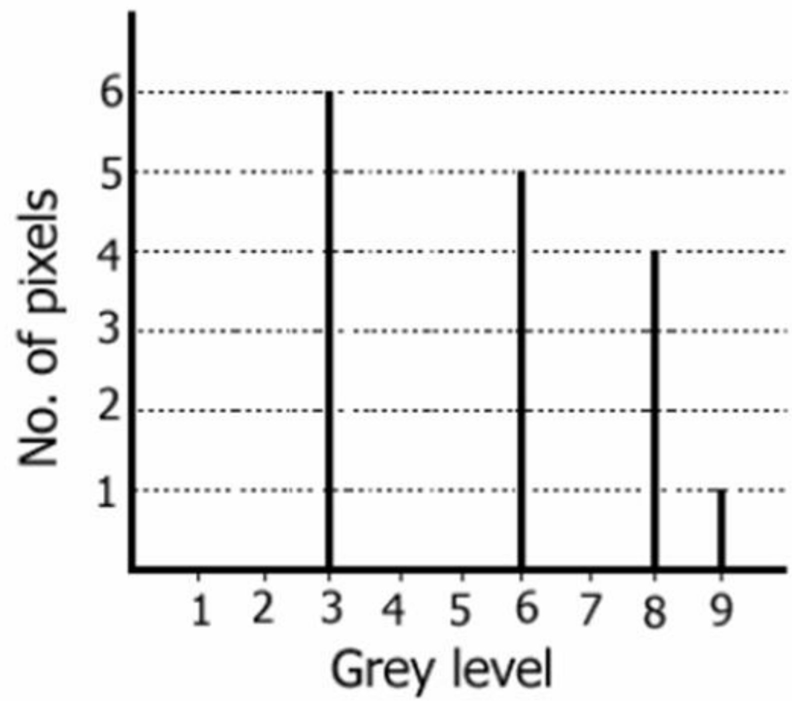


Original Histogram

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Grey level=[0,9]



New Histogram

Spatial Filtering

Filtering

- Removing noise (unwanted data) or highlighting specific detail in an image using subimage is called filtering
- Can be done both in frequency domain & spatial domain
- Subimage is operated on neighborhood of each pixel
- Subimage is called filter, kernel, mask
- Values in subimage are called coefficients

Spatial Filtering

- we have discussed enhancement techniques in which enhancement at any point in image depends only on gray level at that point. These techniques are known as point processing.
- Another approach is spatial filtering in which to use a function in predefined neighborhood of (x,y) to determine the value of g at (x,y).
- A spatial filter consists of (a) **a neighborhood**, and (b) **a predefined operation**
- Size of mask determine size of neighborhood and mask coefficient determine the nature of process.

- Filter mask is moved from point to point in an image.
- At each point (x,y), the response of filter is calculated using predefined operation
- If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter.
- Otherwise the filter is nonlinear.

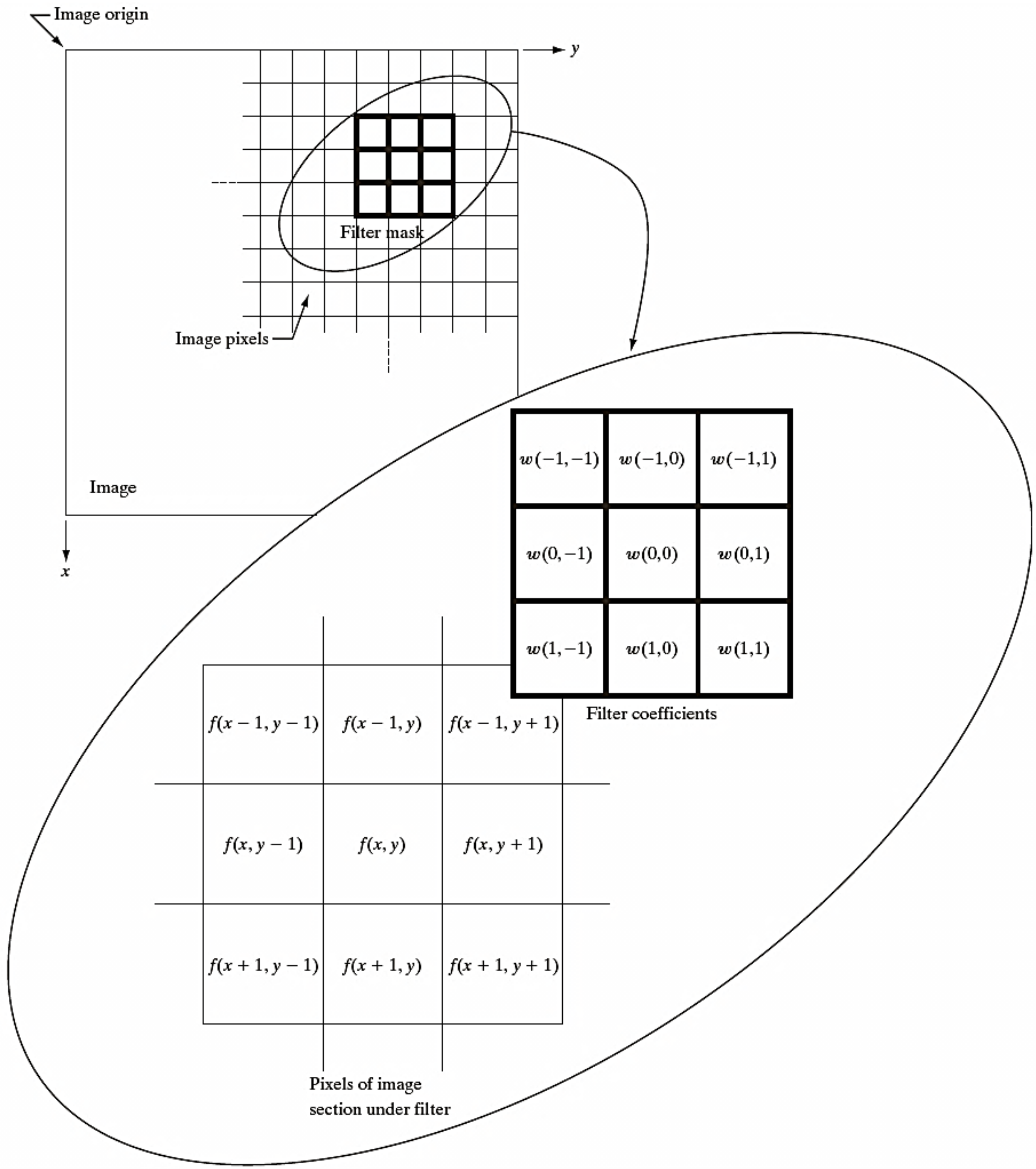


Image origin
y
x
Image
Filter mask
Image pixels
w(−1,−1)
w(−1,0)
w(−1,1)
w(0,−1)
w(0,0)
w(0,1)
w(1,−1)
w(1,0)
w(1,1)
Filter coefficients
f(x − 1, y − 1)
f(x − 1, y)
f(x − 1, y + 1)
f(x, y − 1)
f(x, y)
f(x, y + 1)
f(x + 1, y − 1)
f(x + 1, y)
f(x + 1, y + 1)
Pixels of image
section under filter

Linear spatial filtering of an image of size MxN with a filter of size mxn is given by the expression

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

FIGURE 3.33 Another representation of a general 3×3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

filtering for border pixels

- 3 ways

1. Move center of mask always less than (n-1)/2 pixels from the border
2. Padding input image with 0
3. Replicating Rows & Columns

Spatial Correlation

- Correlation is the process of moving a filter mask over the image and compute the sum of products at each pixel location.

The correlation of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Origin f

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} & w & \\ \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{7} & \mathbf{8} & \mathbf{9} \end{matrix}$$

(a)

Padded f

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(b)

Initial position for w

$$\begin{matrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & 0 & 0 & 0 & 0 \\ \mathbf{4} & \mathbf{5} & \mathbf{6} & 0 & 0 & 0 & 0 \\ \mathbf{7} & \mathbf{8} & \mathbf{9} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(c)

Correlation result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{9} & \mathbf{8} & \mathbf{7} & 0 \\ 0 & \mathbf{6} & \mathbf{5} & \mathbf{4} & 0 \\ 0 & \mathbf{3} & \mathbf{2} & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(d)

Full correlation result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{9} & \mathbf{8} & \mathbf{7} & 0 & 0 \\ 0 & 0 & \mathbf{6} & \mathbf{5} & \mathbf{4} & 0 & 0 \\ 0 & 0 & \mathbf{3} & \mathbf{2} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(e)

- Correlation of a function with a unit impulse gives a rotated version of a function at the point of impulse.
- If we rotate the filter first by 180 degree and perform the operation it will give the same result as the original filter on place of impulse.

Spatial Convolution

- Convolution is the function in which we pre rotate the filter by 180 degree and perform correlation on image.

The convolution of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

$$w(x, y) \star f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x-s, y-t)$$

Rotated w

9	**8**	**7**	0	0	0	0
6	**5**	**4**	0	0	0	0
3	**2**	**1**	0	0	0	0
0	0	0	**1**	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(f)

Convolution result

0	0	0	0	0
0	**1**	**2**	**3**	0
0	**4**	**5**	**6**	0
0	**7**	**8**	**9**	0
0	0	0	0	0

(g)

Full convolution result

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	**1**	**2**	**3**	0	0
0	0	**4**	**5**	**6**	0	0
0	0	**7**	**8**	**9**	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(h)

Types of spatial filters

- Smoothing Spatial Filters
  - Smoothing Linear Filters
    - Average Filter
    - Weighted Average Filter
  - Order-Statistics Filters
    - Median Filter
    - Minimum Filter
    - Maximum Filter
- Sharpening Spatial Filters
  - First derivative
  - Second Derivative

Smoothing filters

- Smoothing filters are used for blurring and for noise reduction
- Removal of small details in an image prior to large object extraction
- Bridging of small gaps in lines and curves
- Smoothing false contouring

Smoothing Linear filters

- The output of smoothing , linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.
- Also known as Averaging filters or low pass filters
- It replaces the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask.

- This reduces sharp transitions in gray levels
- Random noise typically consists of Sharp transitions in gray levels, so it will be reduced by average filter.
- Edges also belong to sharp transition in gray level, so filters blur the edges
- The filter in which all coefficients are equal is called a box filter.
- Weighted average filter can reduce the blurring in smoothing.

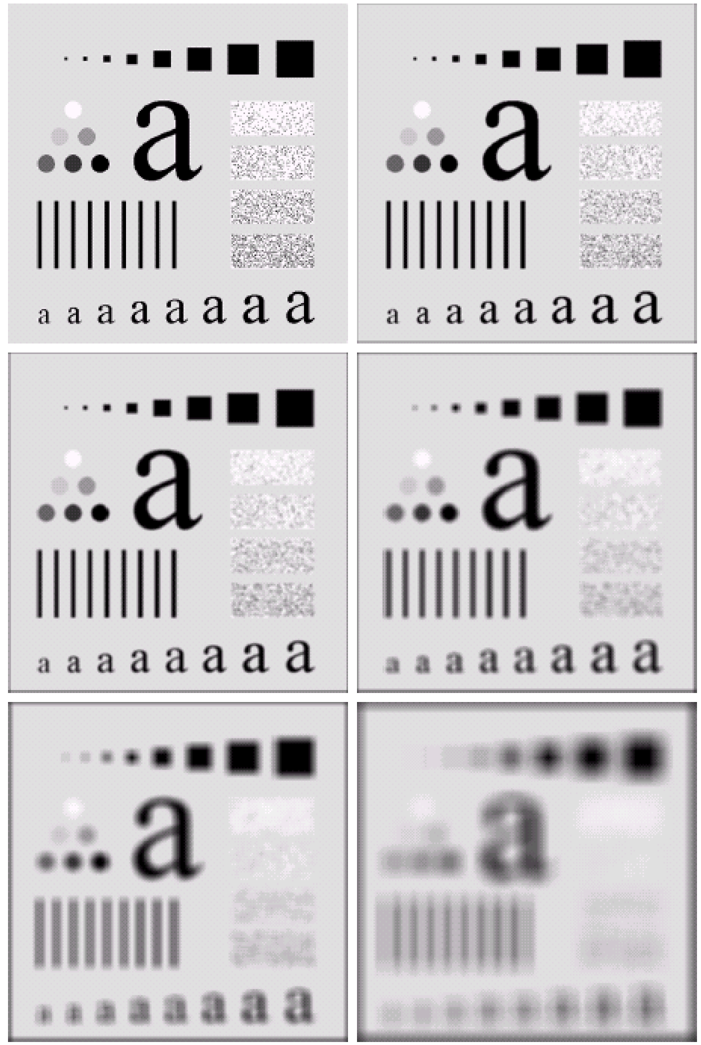
Box filter and Weighted Average Filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

The general implementation for filtering an M × N image with a weighted averaging filter of size m × n is given

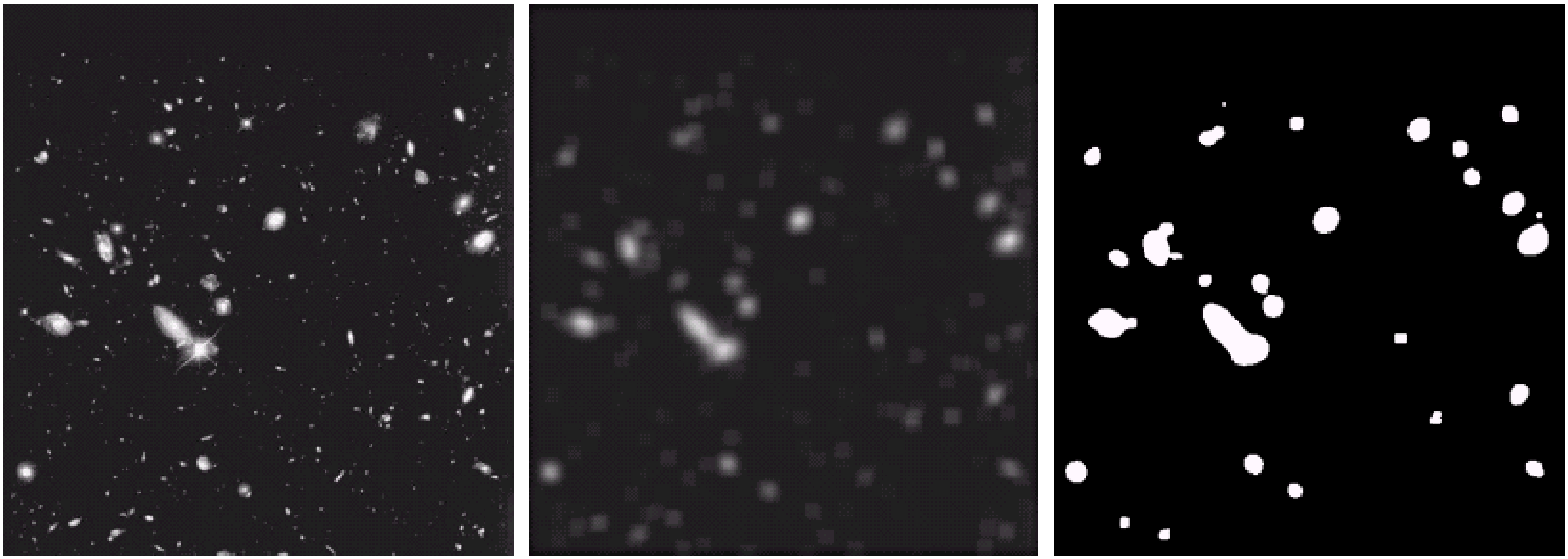
$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

where $m = 2a+1, \quad n = 2b+1.$



a b
c d
e f

FIGURE 3.35 (a) Original image, of size 500×500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15,$ and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50×120 pixels.



a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Order-Statistics Filters

- Nonlinear
- Based on ordering (ranking) the pixels contained in the filter mask
- Replacing the value of the center pixel with the value determined by the ranking result
- 3 types:
- Min filter
- Max Filter
- Median Filter

Min Filter

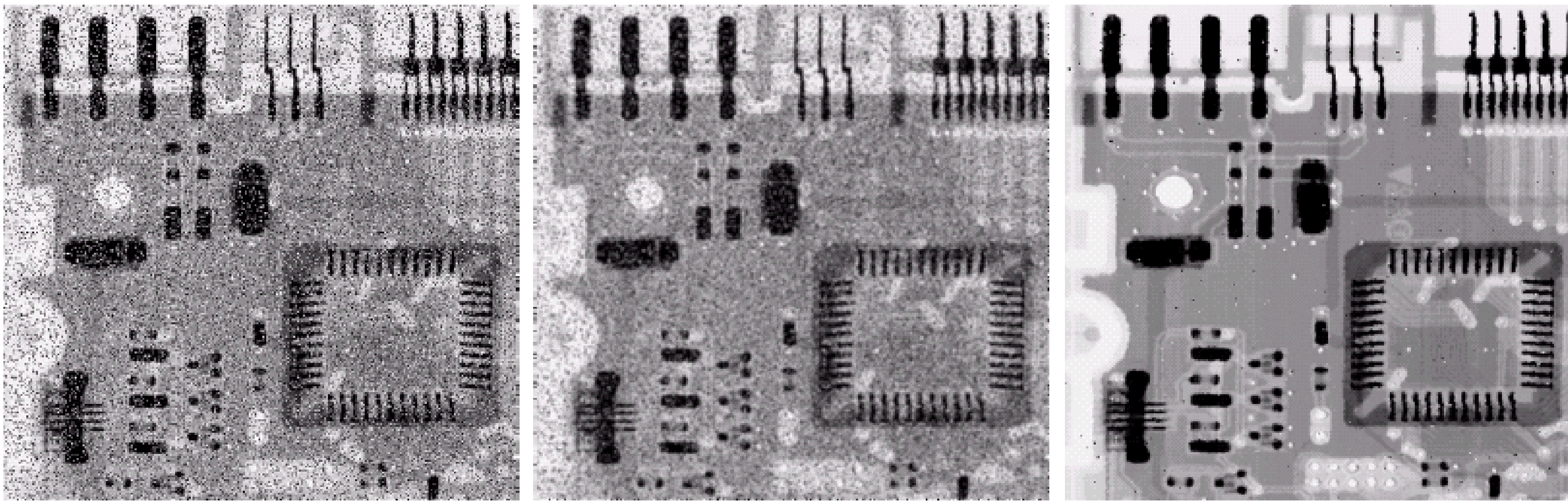
- For any pixel at (x, y) consider n x n neighborhood
- Sort pixel values
- Place the lowest (minimum) gray value in the output image at location (x, y)
- Repeat this process for all pixels
- Good in removing salt noise

Max Filter

- For any pixel at (x, y) consider n x n neighborhood
- Sort pixel values
- Place the highest (maximum) gray value in the output image at location (x, y)
- Repeat this process for all pixels
- Good in removing pepper noise

Median Filter

- For any pixel at (x, y) consider n x n neighborhood
- Sort pixel values
- Place the median gray value in the output image at location (x, y)
- Repeat this process for all pixels
- Good in removing salt & pepper noise
- Median value for 3 x 3 filter is value at location 5 after sorting 9 pixels



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Sharpening Filters

- Objective: To highlight fine detail in an image or to enhance detail that has been blurred
- Also known as high pass filter
- Produces opposite effect than smoothing filters
- Smoothing is achieved by pixel averaging.
- Averaging is analogous to integration
- Sharpening is achieved by differentiation
- Image differentiation enhances edges and other discontinuities and deemphasizes areas with slowly varying gray-levels
- First & second derivative of an image are used.

- Image derivative can be obtained from differences. But it must follow some rules:
- First derivative
  - Must be zero in flat segment
  - Must be non zero at the onset of gray level ramp and step
  - Must be non zero along ramps
- Second derivative
  - Must be zero in flat segment
  - Must be non zero at the onset and end of a gray level ramp and step
  - Must be zero along ramps of constant slope

- The first-order derivative of a one-dimensional function f(x) is the difference

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

- The second-order derivative of f(x) as the difference

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

a b
c

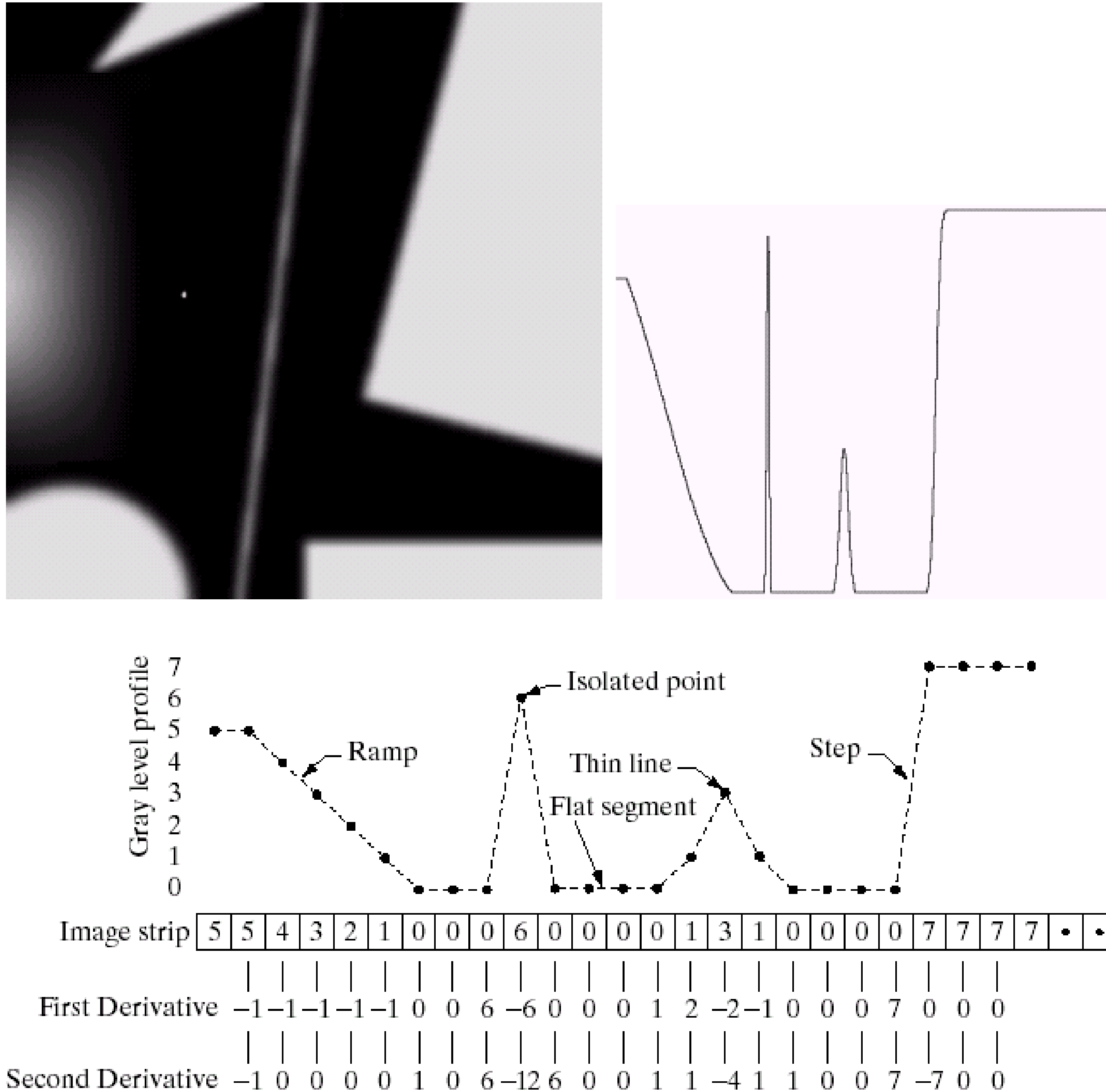
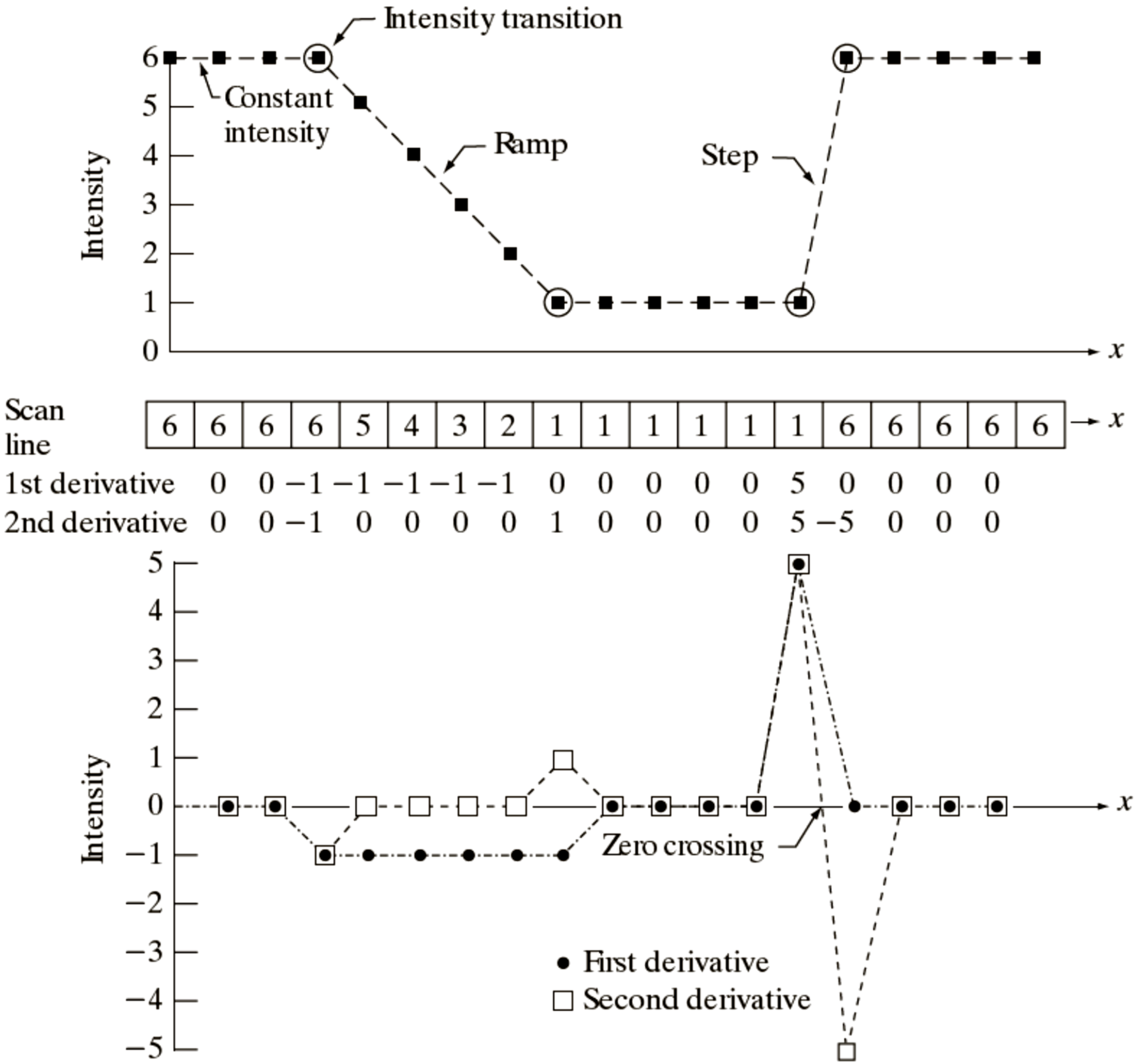


FIGURE 3.38 (a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).


Intensity transition
Constant intensity
Ramp
Step
Intensity
x
Scan line
6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6
1st derivative 0 0 −1 −1 −1 −1 −1 0 0 0 0 0 5 0 0 0 0
2nd derivative 0 0 −1 0 0 0 0 1 0 0 0 0 5 −5 0 0 0
Zero crossing
First derivative
Second derivative

Observations

1) First order derivatives generally produce thick edges in image
2) Second order derivatives have a stronger response to fine detail such as thin lines , isolated points and noise
3) Second order derivatives produce a double-edge response at step and ramp transitions in intensity

- The sign of Second order derivative can be used to determine whether a transition into an edge from light to dark or dark to light
- Second order derivatives are better for image enhancement because they enhance fine detail in image
- First order derivatives are used for edge extraction

Second order derivative OR Laplacian opreator

- It is defined as an isotropic i.e rotational invariant function of 2 variables f(x,y) (image)

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

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

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

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

a b
c d

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

FIGURE 3.37 (a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

- (a) Mask for isotropic result for rotations in increments of 90°. Diagonal Neighbors are not considered
- (b) Mask for isotropic result for rotations in increments of 45°. Diagonal Neighbors are considered
- Because the Laplacian is a derivative operator, It is used to highlight small gray-level discontinuities in an image and deemphasizes regions with slowly varying gray Levels.

- This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background.
- Background features can be "recovered" while still preserving the sharpening effect of the Laplacian operation simply by adding the original and Laplacian images.
- It is important to keep in mind which definition of the Laplacian is used. if the definition used has a negative center coefficient, then we *subtract,* rather than add, the Laplacian image to obtain a sharpened result.

Laplacian Function

Image sharpening in the way of using the Laplacian:

$$g(x, y) = f(x, y) + c\left[\nabla^2 f(x, y)\right]$$

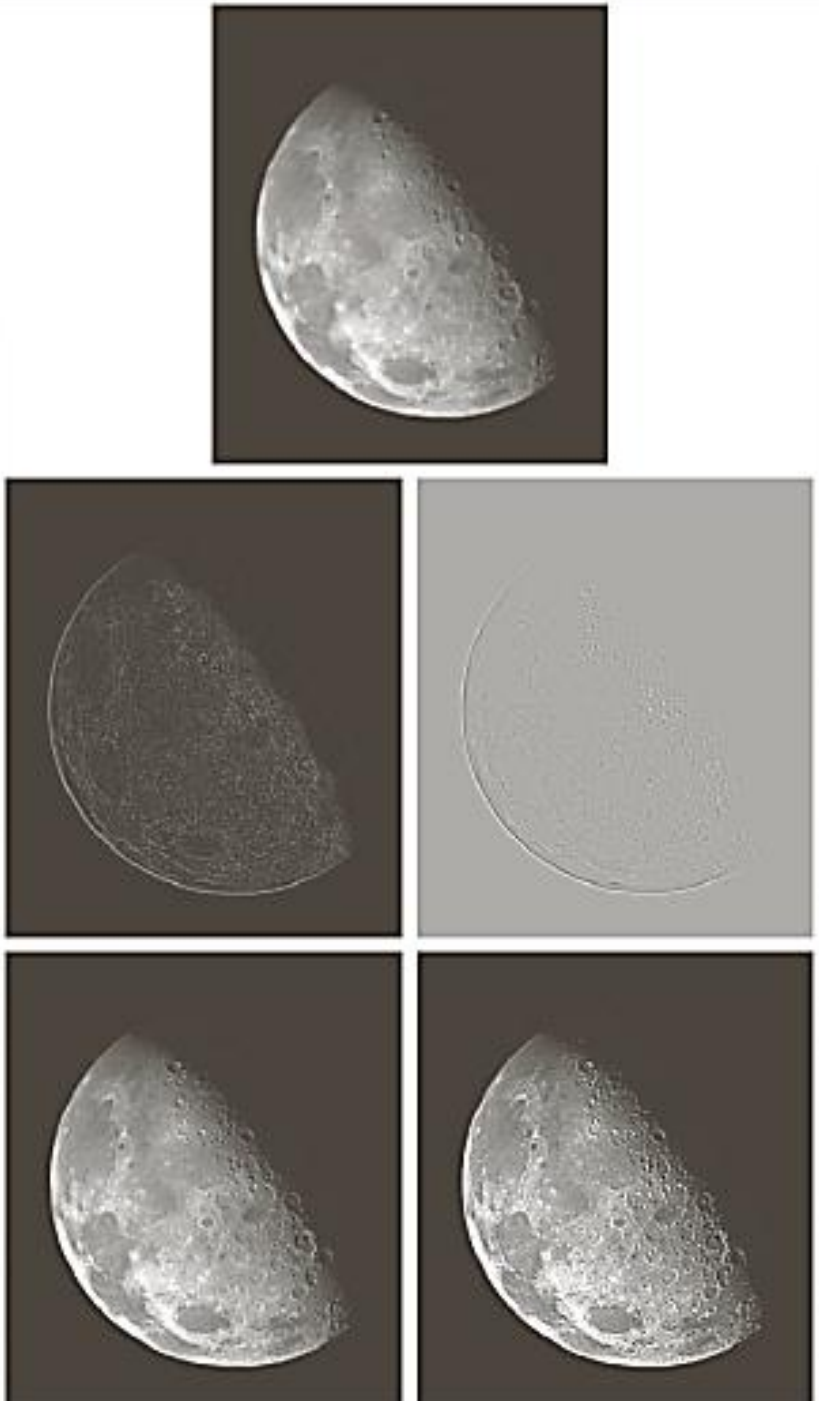
where,

$f(x, y)$ is input image,

$g(x, y)$ is sharpenend images,

$c = -1$ if $\nabla^2 f(x, y)$ corresponding to Fig. 3.37(a) or (b)

and $c = 1$ if either of the other two filters is used.



a
b c
d e

FIGURE 3.38 (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

- The Laplacian image contains both positive and negative values, Scaling is required before display.
- All negative values are clipped at 0 by display.
- A typical way to scale is to add its minimum value to bring new minimum as zero.
- Then scale the result to [0,L-1]

- Adding the image to the Laplacian restored the overall gray level variations in the image
- Laplacian increases the contrast at the locations of gray-level discontinuities

Unsharp masking & Highboost filtering

- Unsharp masking

  Sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image

  e.g., printing and publishing industry

- Steps

  1. Blur the original image
  2. Subtract the blurred image from the original
  3. Add the mask to the original

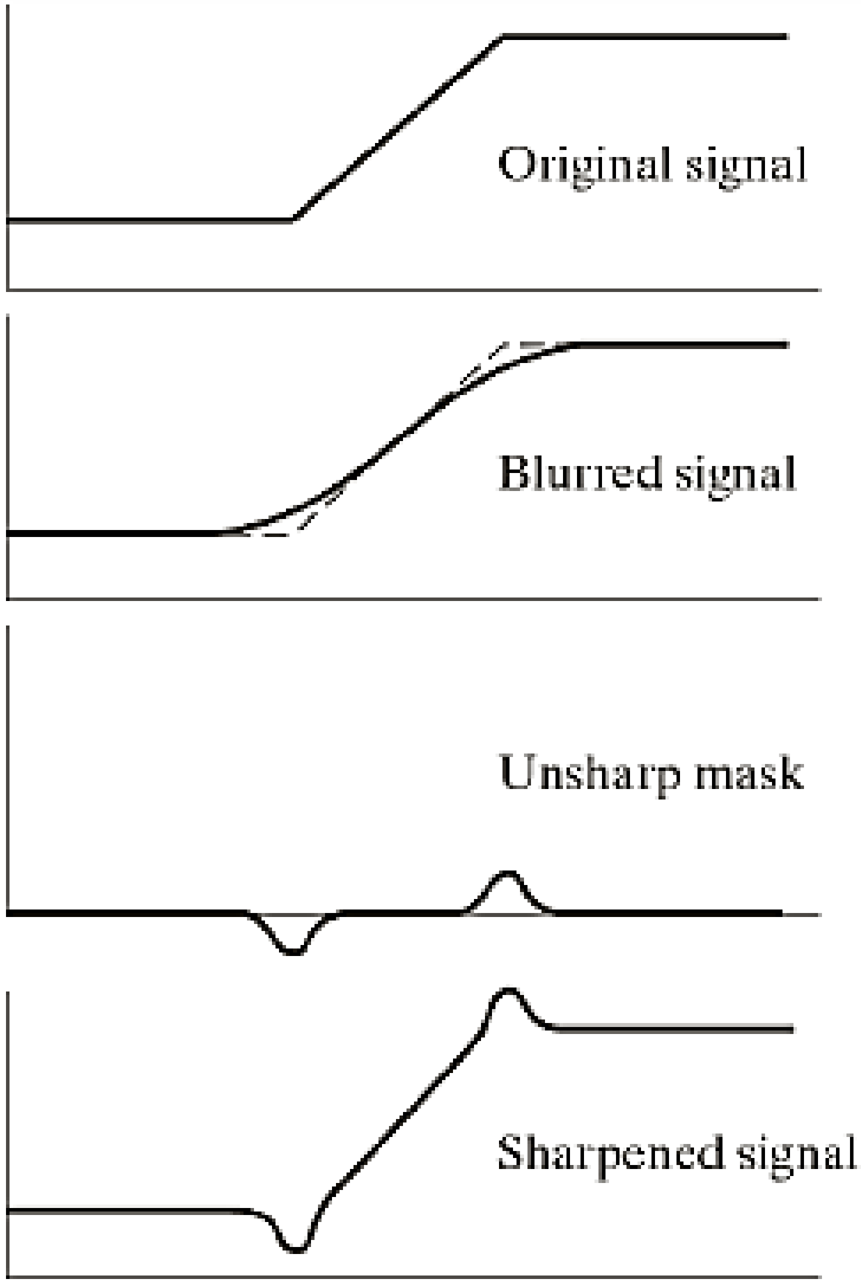
Let $\overline{f}(x, y)$ denote the blurred image, unsharp masking is

$$g_{mask}(x, y) = f(x, y) - \overline{f}(x, y)$$

Then add a weighted portion of the mask back to the original

$$g(x, y) = f(x, y) + k * g_{mask}(x, y) \qquad k \geq 0$$

when $k > 1$, the process is referred to as highboost filtering.



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

Image Sharpening based on First-Order Derivatives

- First order derivative of an image is obtained by gradient.
- Gradient is used in industry for inspection for detection of defects
- Gradient high lights small discontinuities and eliminates slowly changing background features

For function $f(x, y)$, the gradient of f at coordinates (x, y) is defined as

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

The *magnitude* of vector ∇f, denoted as $M(x, y)$

Gradient Image $\quad M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x{}^2 + g_y{}^2}$

$$M(x,y) = \text{mag}(\nabla f) = \sqrt{g_x^{\,2} + g_y^{\,2}}$$

$$M(x,y) \approx |g_x| + |g_y|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$M(x,y) = |z_8 - z_5| + |z_6 - z_5|$$

Roberts Cross-gradient Operator

Roberts Cross-gradient Operators

$$M(x,y) \approx |z_9 - z_5| + |z_8 - z_6|$$

- These masks are even sized and hence awkward to implement because they do not have a center of symmetry.
- The approximation of gradient with 3*3 filter mask is given by Sobel.

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

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

a
b c
d e

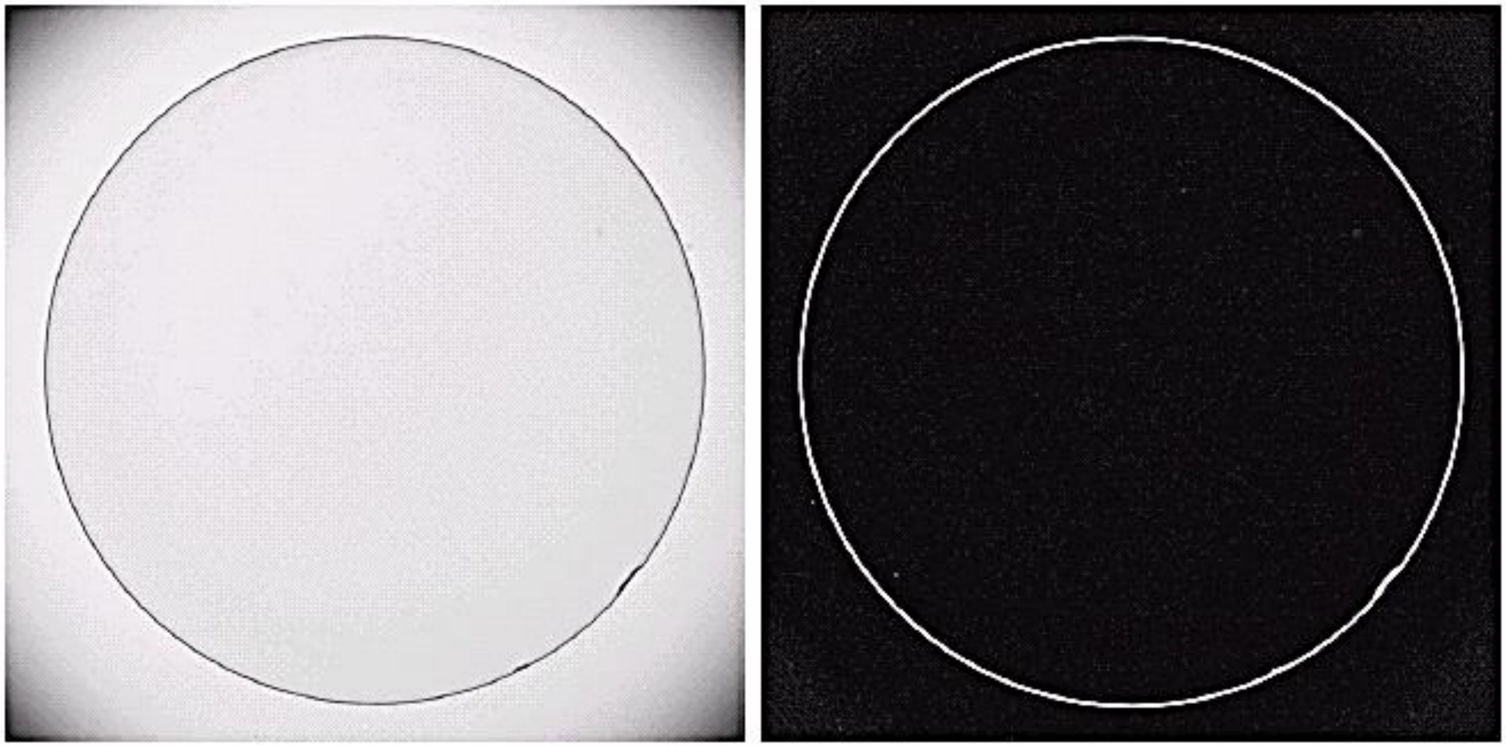
FIGURE 3.41 A 3×3 region of an image (the zs are intensity values). (b)–(c) Roberts cross gradient operators. (d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

Sobel Operator

Sobel Operators

$$M(x,y) \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)|$$
$$+ |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

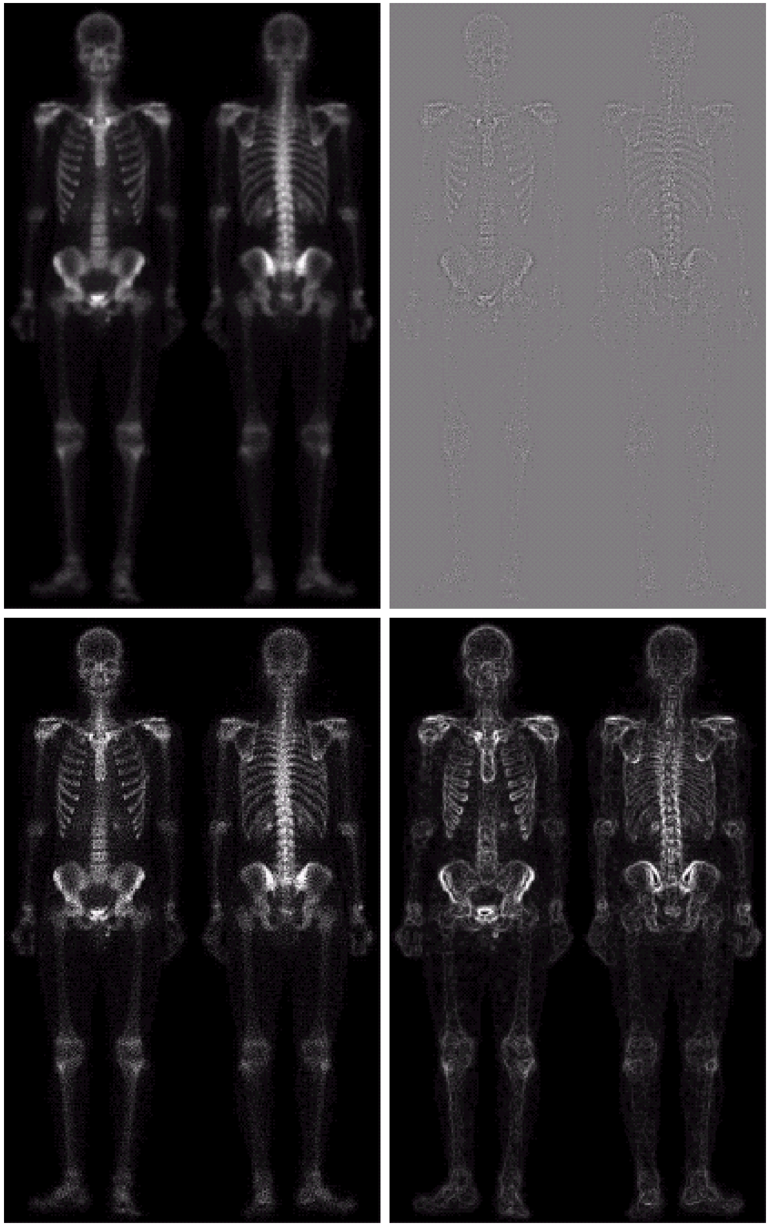
- Sum of coefficients in all masks are zero which indicates that they would give a response of zero in an area of constant intensity, as expected from derivative operators.
- Also, weight 2 is given to center pixels to achieve smoothning.



a b

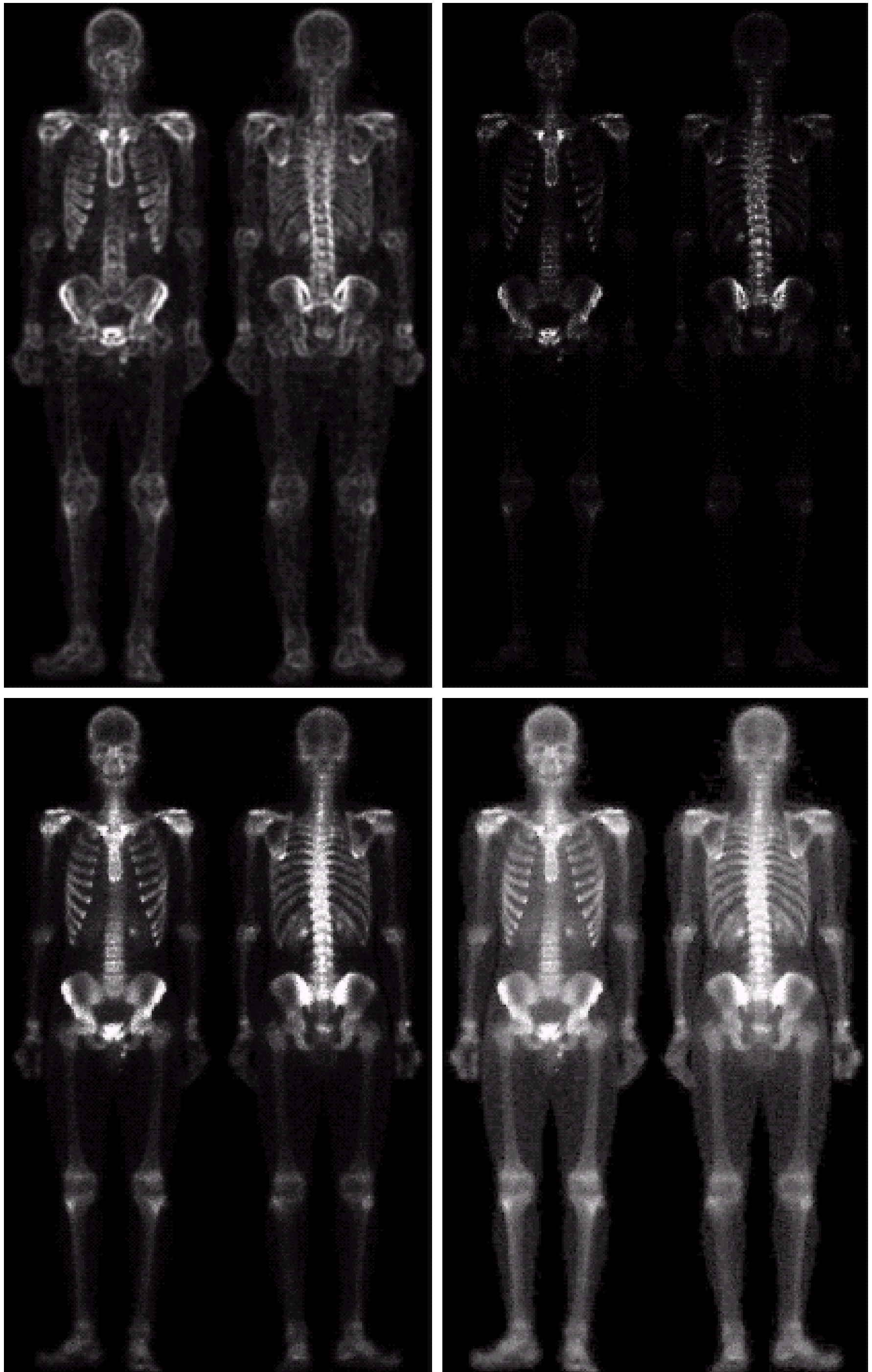
FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Combining spatial enhancement methods



a	b
c	d

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).



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.)

Frequency Domain

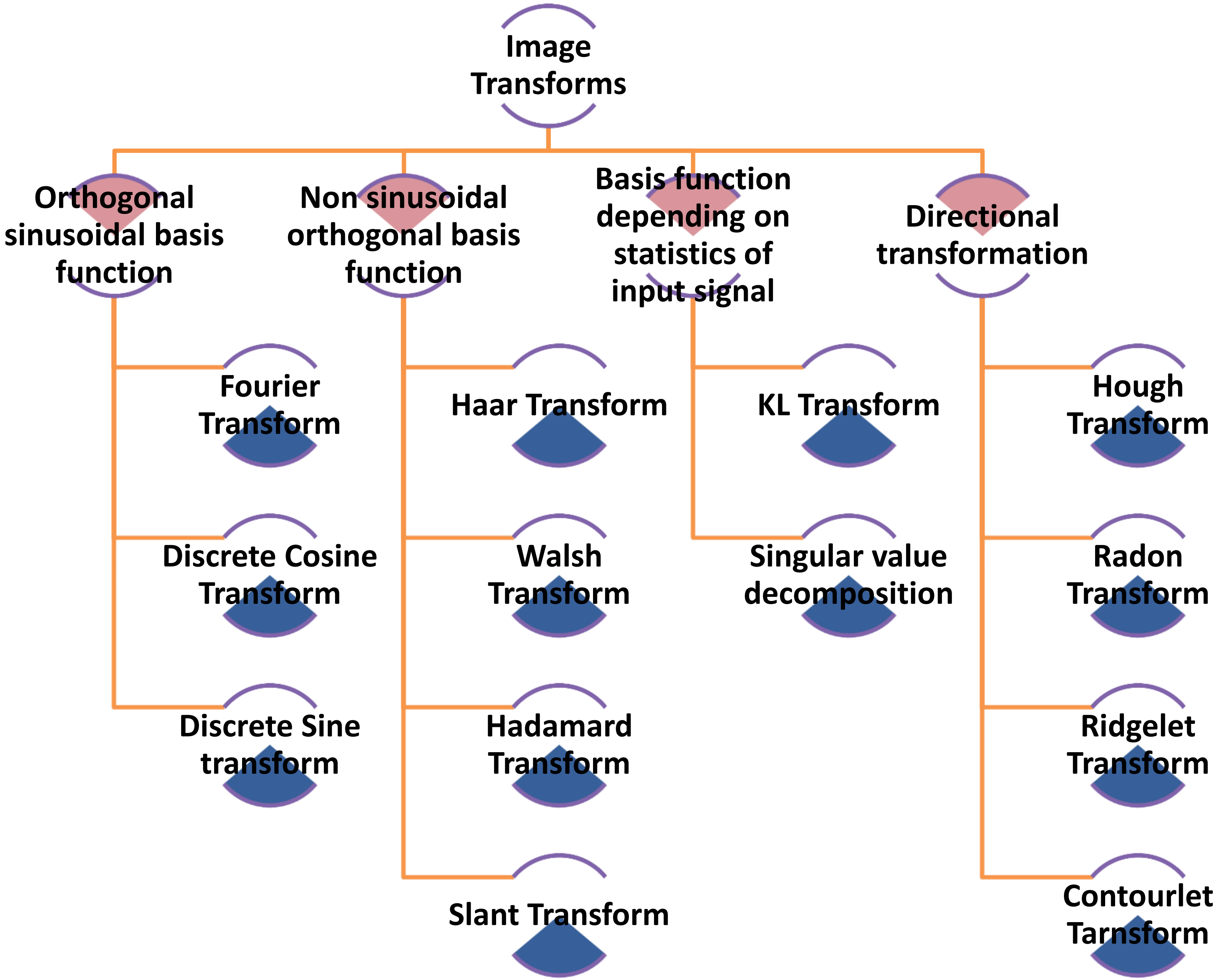
What is image transform?

- It is a mathematical tool which allows us to move from one domain to another domain.
- The reason to migrate is to perform certain tasks in easier manner.

Why Image transform?

- **Mathematical convenience** – complex process of convolution in spatial domain is equal to simple multiplication in frequency domain.
- **To extract more information** - It isolate the critical components of the image pattern so that they are directly accessible for analysis.
- **Efficiency** - The transformation places the data in more compact form so that they can be stored and transmitted efficiently.

Classification of Image Transformation



Fourier Transform

- Fourier Series and Fourier Transforms are tools to covert signal from spatial domain to frequency domain .
- Fourier series : When signal is periodic it can be represented as the sum of sines and /or cosines of different frequency multiplied by a different coefficients
- Fourier Transform : when signal is not periodic but finite duration like images it can be represented as integral of sines and or cosines multiplied by a weighting function
- Original signal is composed of infinite frequencies of varying amplitudes and the Fourier transform tells us what these frequencies are

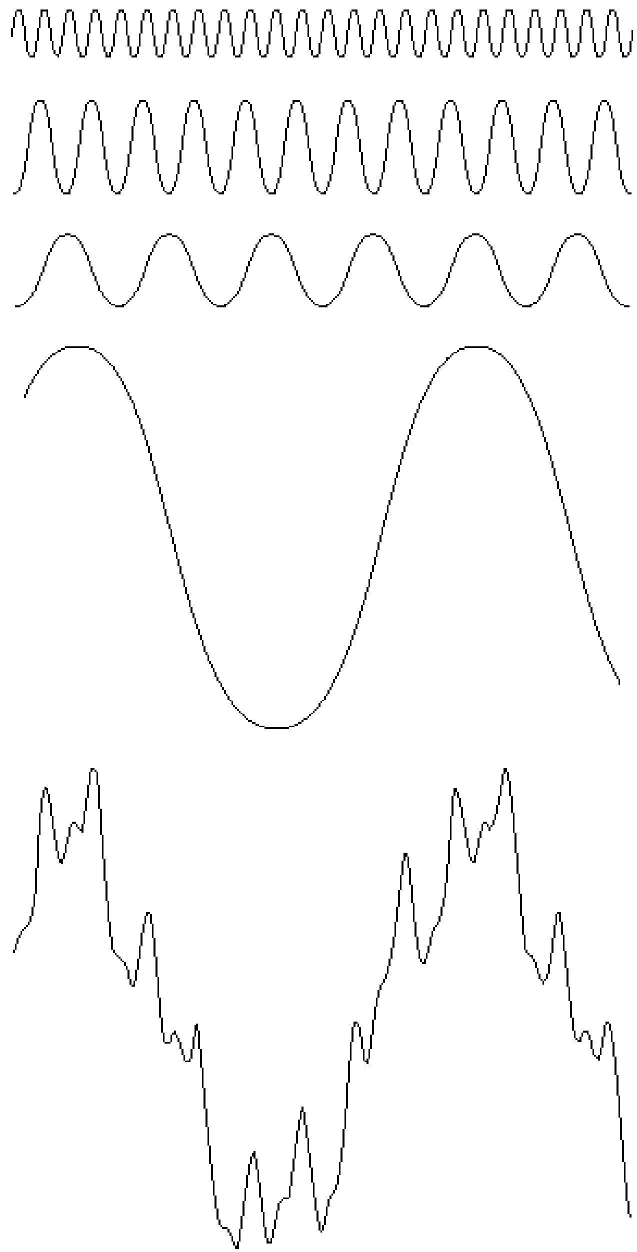


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

- Transitions in gray levels are considered as a frequency in digital image.
- Using inverse fourier transform, the original image can be recovered completely with no loss of information.

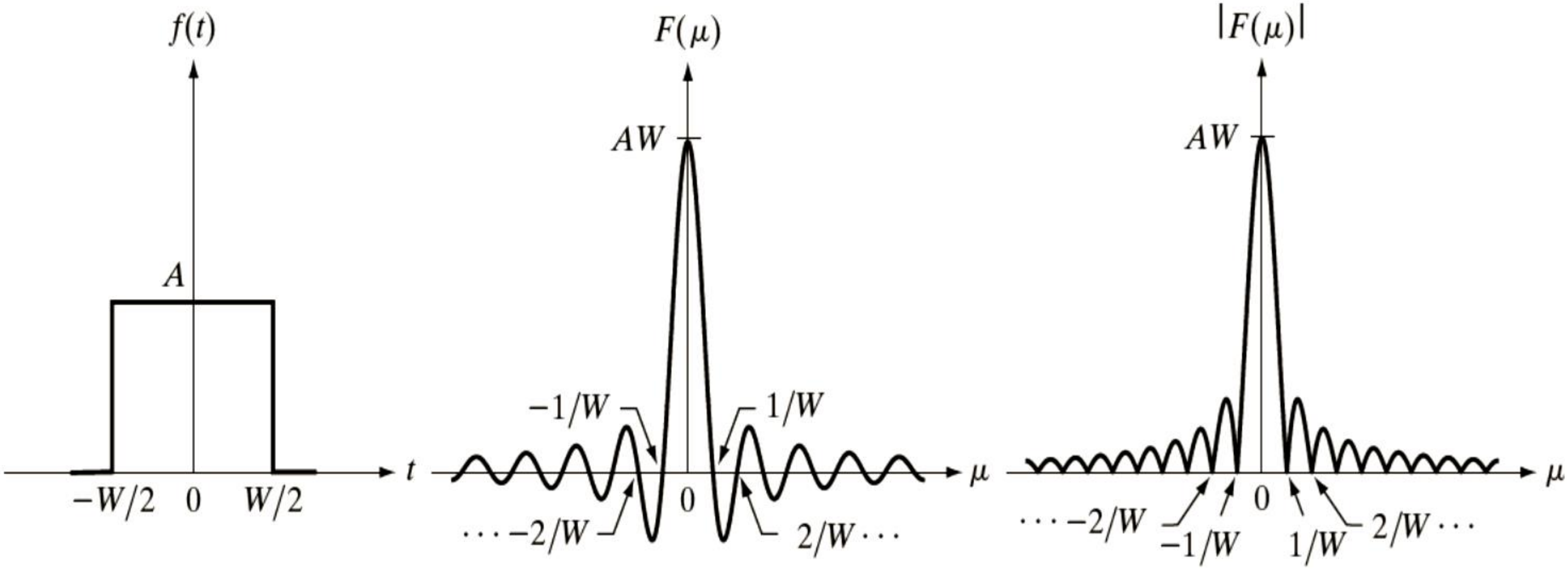
- A useful analogy is to compare the Fourier transform to a glass prism, The prism is a physical device that separates light into various color components, each depending on its wavelength (or frequency) content.
- The Fourier transform may be viewed as a "mathematical prism" that separates a function into various components, also based on frequency content.

1-D Fourier transform & its inverse

The Fourier transform, *F(u)*, of a single variable, continuous function, *f(x)*, is defined by the equation

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

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



a b c

FIGURE 4.4 (a) A simple function; (b) its Fourier transform; and (c) the spectrum. All functions extend to infinity in both directions.

- Locations of zeros are inversely proportional to the width W and the height of the lobes decreasing as a function of distance from the origin.

1-D DFT

- Suppose $\{f(0), f(1), \ldots, f(M-1)\}$ is a sequence/vector/1-D image of length M. Its M-point DFT is defined as

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j\frac{2\pi}{M}ux}, u = 0,1,2,\ldots\ldots, M-1$$

- Inverse DFT

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{j\frac{2\pi}{M}ux}, x = 0,1,2,\ldots\ldots, M-1$$

- Recall: $e^{j\theta} = \cos\theta + j\sin\theta$

Fourier Spectrum and phase angle : representation of intensity as a function of frequency is called spectrum

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

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

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

Power Spectrum: $$P(u) = |F(u)|^2$$

1-D function

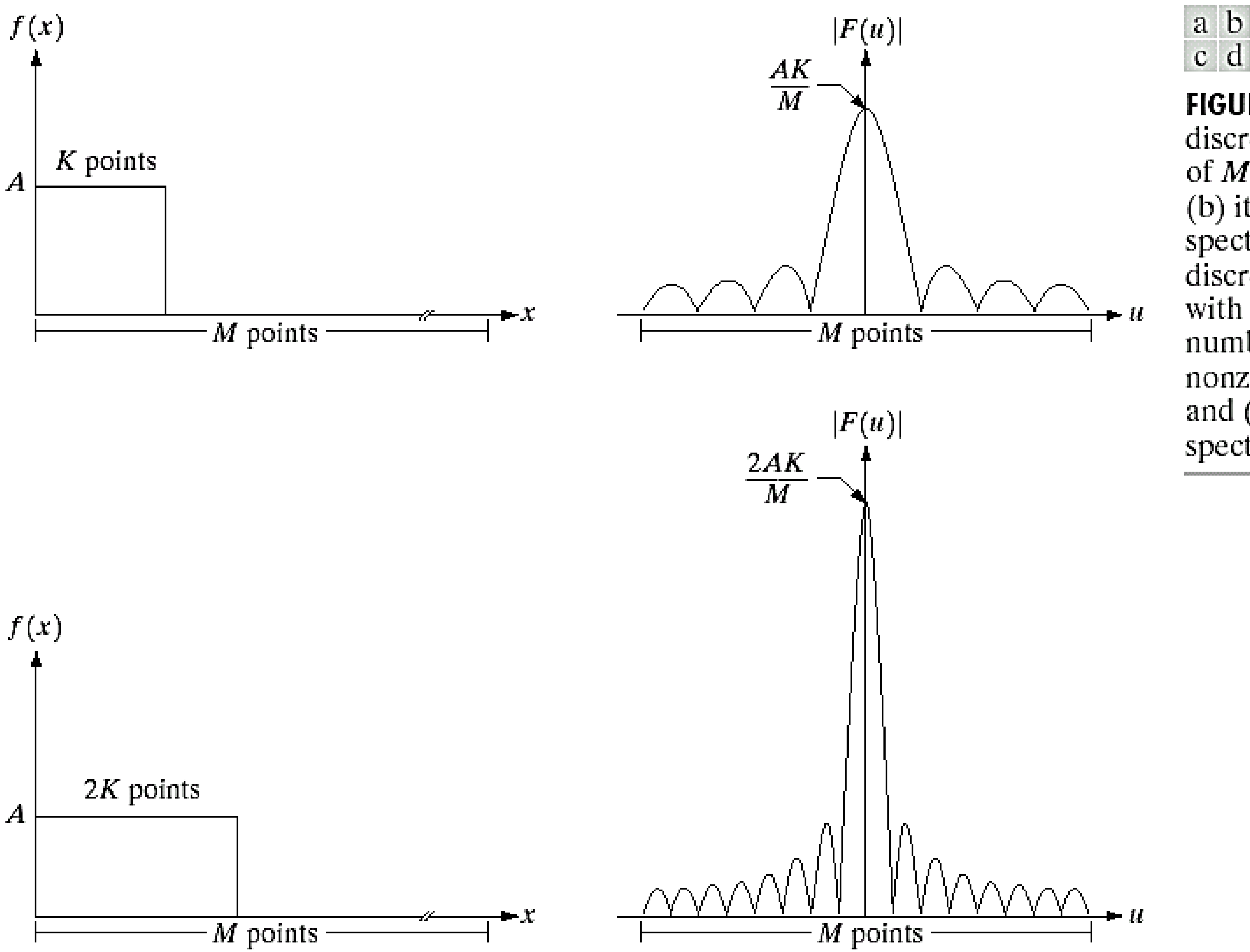


FIGURE 4.2 (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.

2D Fourier Transform & Its Inverse

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

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

2D DFT & its Inverse

DFT

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

$$u=0,1,2,\ldots,M-1,\quad v=0,1,2,\ldots,N-1$$

IDFT

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

$$x=0,1,2,\ldots,M-1,\quad y=0,1,2,\ldots,N-1$$

2D DFT in Polar

Magnitude is known as Fourier spectrum

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

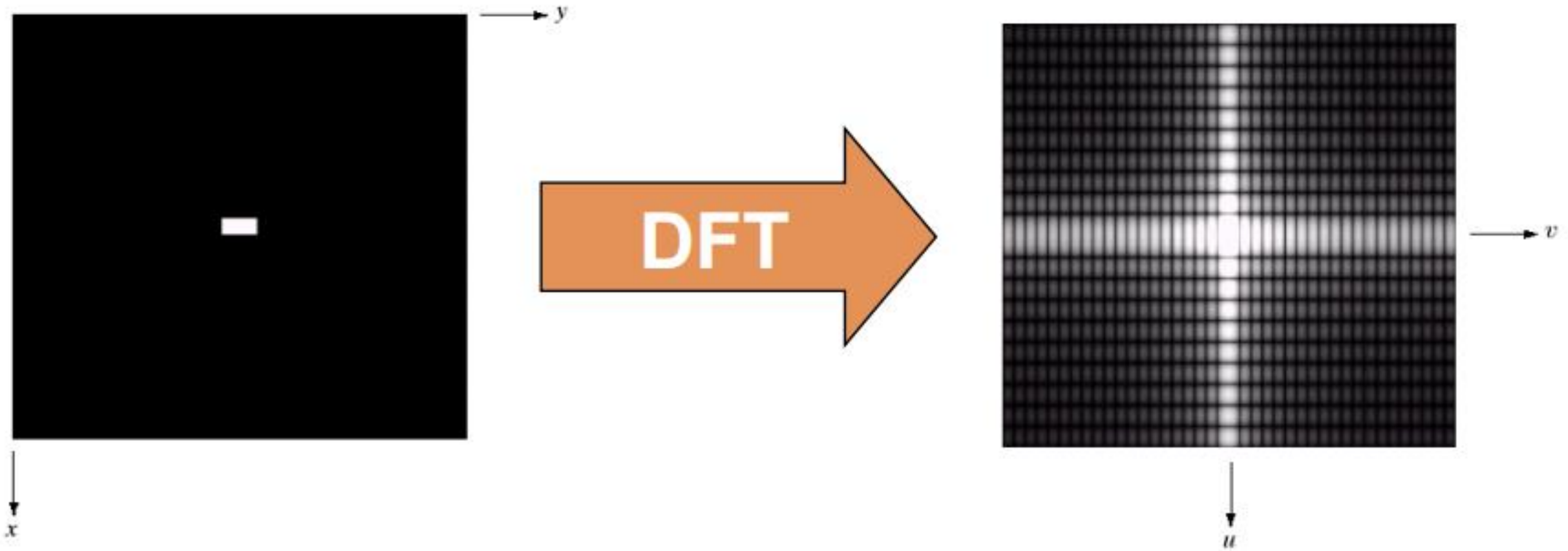
Magnitude: $|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$

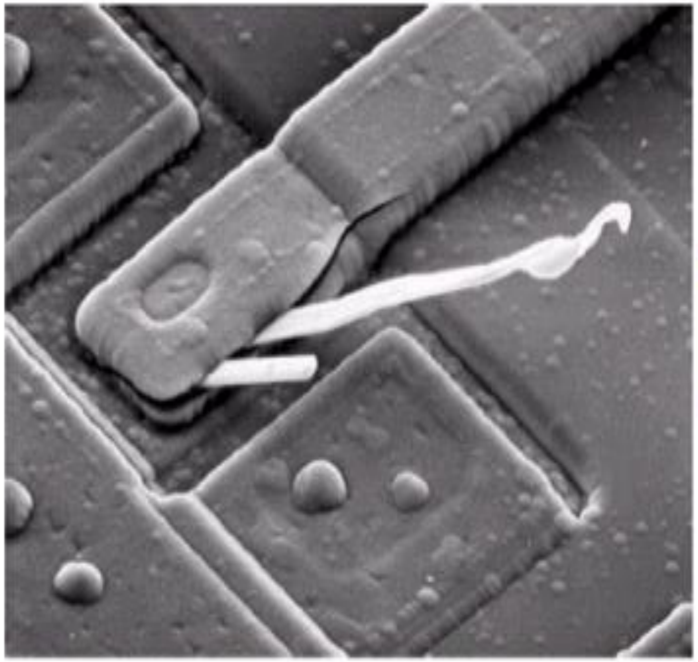
Phase: $\phi(u,v) = \tan^{-1}\left(\frac{I(u,v)}{R(u,v)}\right)$

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

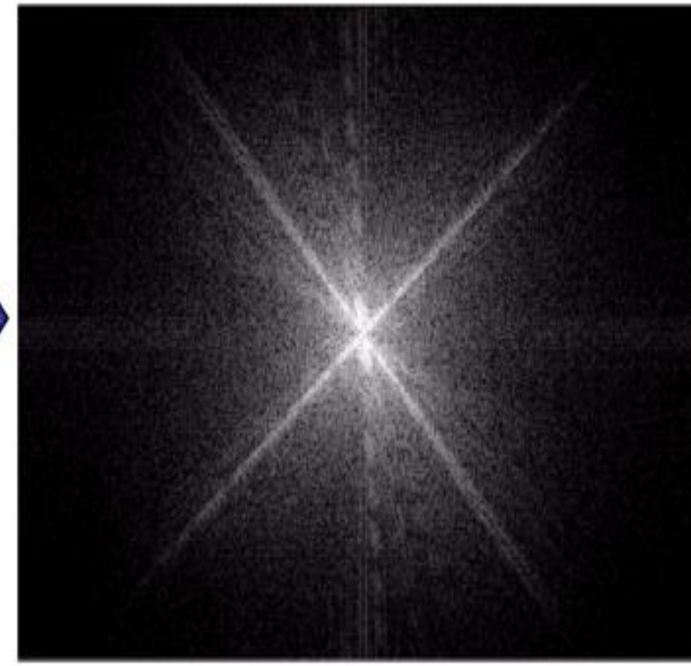
2-D function

The DFT of a two dimensional image can be visualised by showing the spectrum of the images component frequencies



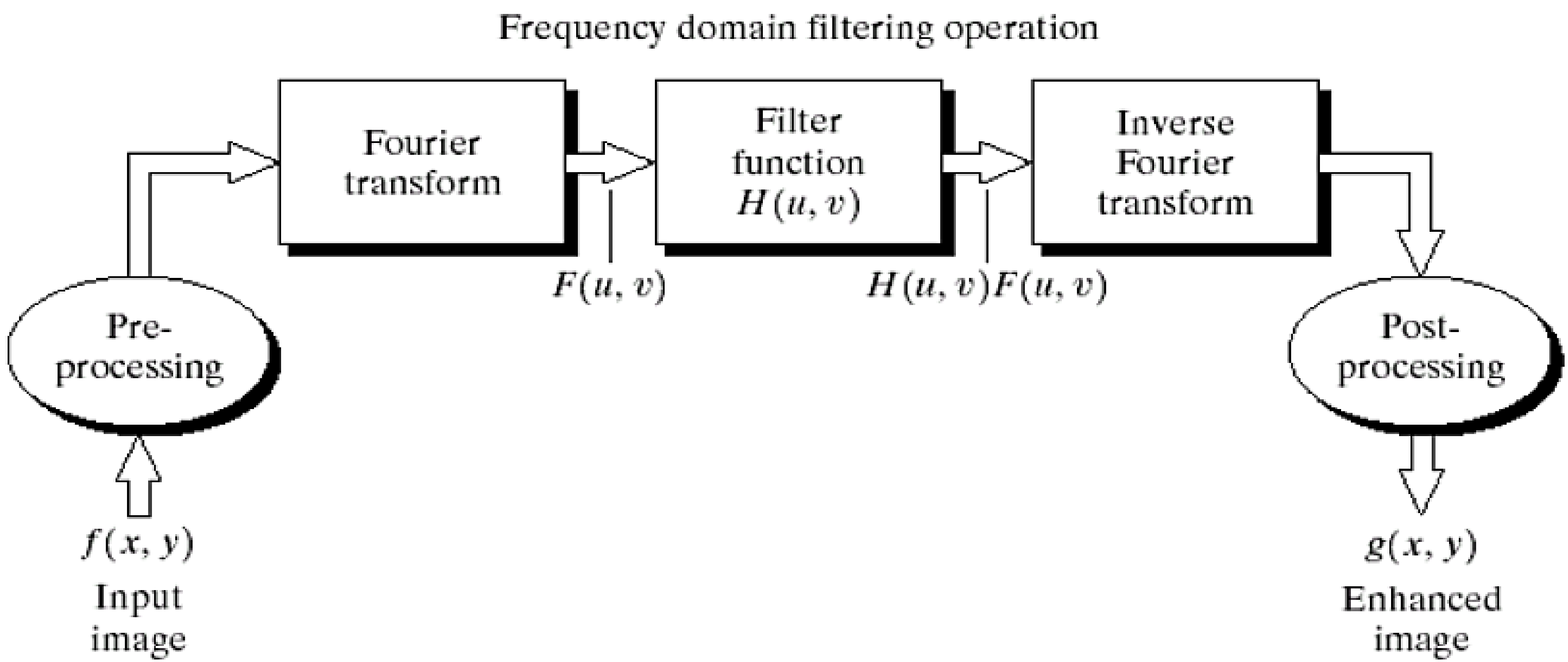


Scanning electron microscope image of an integrated circuit magnified ~2500 times



Fourier spectrum of the image

Filtering in frequency domain



Frequency domain filtering operation
Fourier transform
Filter function $H(u, v)$
Inverse Fourier transform
$F(u, v)$
$H(u, v)F(u, v)$
Pre-processing
Post-processing
$f(x, y)$
Input image
$g(x, y)$
Enhanced image

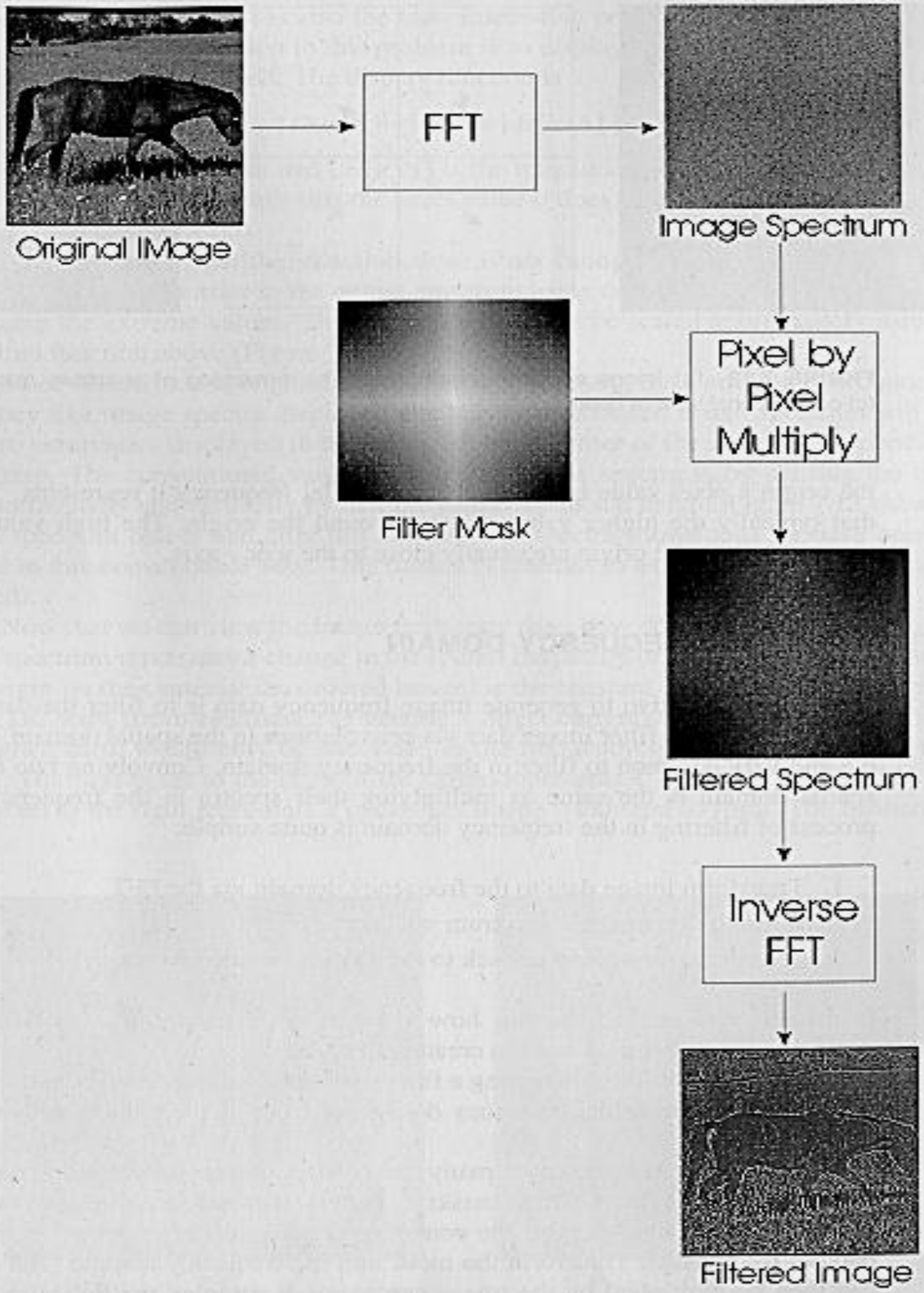
Convolution theorem

Convolution in spatial domain is multiplication in frequency domain.

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

$$\Updownarrow$$

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


Original IMage
FFT
Image Spectrum
Filter Mask
Pixel by Pixel Multiply
Filtered Spectrum
Inverse FFT
Filtered Image

Let f(x,y) and h(x,y) be two image arrays of size A*B & C*D pixels, respectively. Wraparound error can be avoided by padding these functions with zeros.

$$f_p(x,y) = \begin{cases} f(x,y) & 0 \le x \le A-1 \text{ and } 0 \le y \le B-1 \\ 0 & A \le x \le P \text{ or } B \le y \le Q \end{cases}$$

and

$$h_p(x,y) = \begin{cases} h(x,y) & 0 \le x \le C-1 \text{ and } 0 \le y \le D-1 \\ 0 & C \le x \le P \text{ or } D \le y \le Q \end{cases}$$

with

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

and

$$Q \ge B + D - 1$$

The resulting padded images are of size P* Q. If both arrays are of the same size,M*N, then we require that

$$P \ge 2M - 1$$

$$Q \ge 2N - 1$$

Frequency domain filtering: steps

1. Given an input image $f(x, y)$ of size $M \times N$, obtain the padding parameters P and Q from Eqs. (4.6-31) and (4.6-32). Typically, we select $P = 2M$ and $Q = 2N$.
2. Form a padded image, $f_p(x, y)$, of size $P \times Q$ by appending the necessary number of zeros to $f(x, y)$.
3. Multiply $f_p(x, y)$ by $(-1)^{x+y}$ to center its transform.
4. Compute the DFT, $F(u, v)$, of the image from step 3.

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

5. Generate a real, symmetric filter function, $H(u, v)$, of size $P \times Q$ with center at coordinates $(P/2, Q/2)$.[†] Form the product $G(u, v) = H(u, v)F(u, v)$

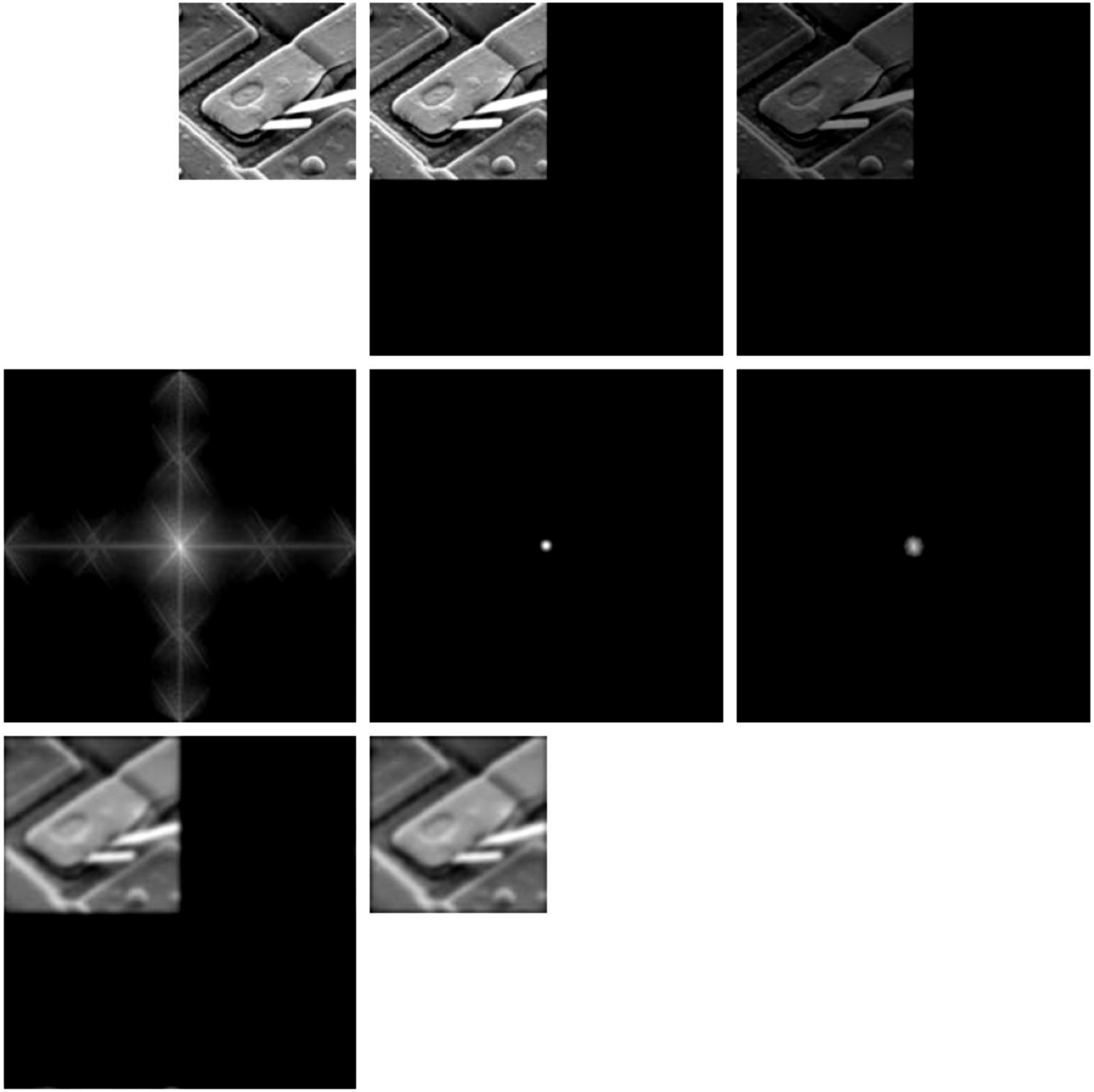
$$G(u,v) = F(u,v)H(u,v) = H(u,v)\,R(u,v) + jH(u,v)I(u,v)$$

6. Obtain the processed image:

$$g_p(x, y) = \left\{\text{real}\left[\Im^{-1}[G(u, v)]\right]\right\}(-1)^{x+y}$$

where the real part is selected in order to ignore parasitic complex components resulting from computational inaccuracies, and the subscript p indicates that we are dealing with padded arrays.

7. Obtain the final processed result, $g(x, y)$, by extracting the $M \times N$ region from the top, left quadrant of $g_p(x, y)$.



a	b	c
d	e	f
g	h	

FIGURE 4.36
(a) An $M \times N$ image, f.
(b) Padded image, f_p of size $P \times Q$.
(c) Result of multiplying f_p by $(-1)^{x+y}$.
(d) Spectrum of F_p. (e) Centered Gaussian lowpass filter, H, of size $P \times Q$.
(f) Spectrum of the product HF_p.
(g) g_p, the product of $(-1)^{x+y}$ and the real part of the IDFT of HF_p.
(h) Final result, g, obtained by cropping the first M rows and N columns of g_p.

- $h(x,y)$ specified in spatial domain: How to generate $H(u,v)$ from $h(x,y)$?
- If $h(x,y)$ is given in the spatial domain, we can generate $H(u,v)$ as follows:

1. Form $h_p(x,y)$ by padding with zeroes.
2. Multiply by $(-1)^{x+y}$ to center its spectrum.
3. Compute its DFT to obtain $H(u,v)$

Types of filters

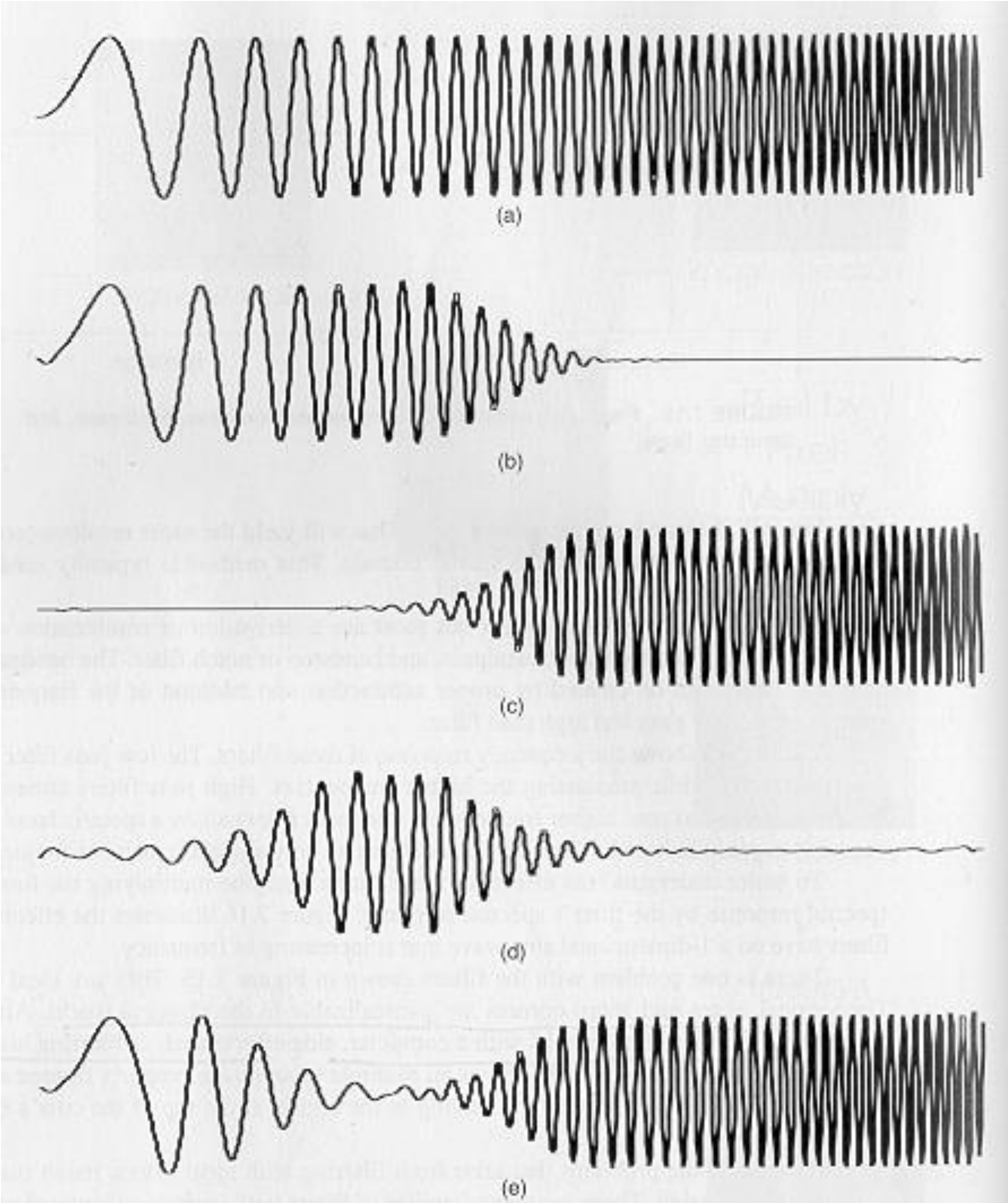
- Typically, filters are classified by examining their properties in the frequency domain:

  (1) Low-pass

  (2) High-pass

  (3) Band-pass

  (4) Band-stop



Original signal

Low-pass filtered

High-pass filtered

Band-pass filtered

Band-stop filtered

Image Smoothing-
Lowpass filtering

Smoothing

- Smoothing (blurring) in frequency domain is achieved by attenuating the high frequencies, that is, by passing only low frequencies.
- Such filters are called lowpass filters.
- We will consider 3 types of lowpass filters:

1. Ideal lowpass filter
2. Butterworth lowpass filter
3. Gaussian lowpass filter

- These three filters cover the range from very sharp (ideal) to very smooth (Gaussian) filtering.

- Butterworth filter has a parameter called the **order** of filtering.
- For higher order values, it approaches to ideal filter. For lower order values, it is more like Gaussian filter.
- Thus, Butterworth filter provides the transition between the two extremes.

Ideal lowpass filter

- Simply cut off all high frequency components that are at specified distance D_0 from the origin of the transform.
- Where D_0 is a positive constant and D(u,v) is the distance between a point (u,v) and the center of the frequency rectangle.

where $D(u,v)$ is given as:

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

The transfer function for the ideal low pass filter can be given as:

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

where $D(u,v)$ is given as:

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

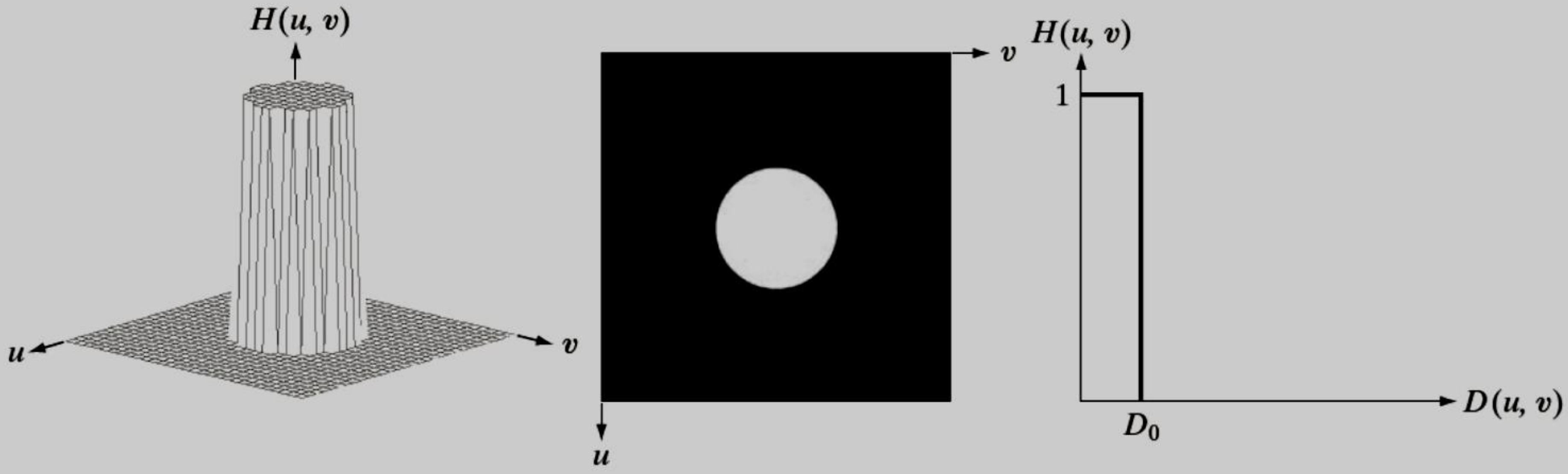
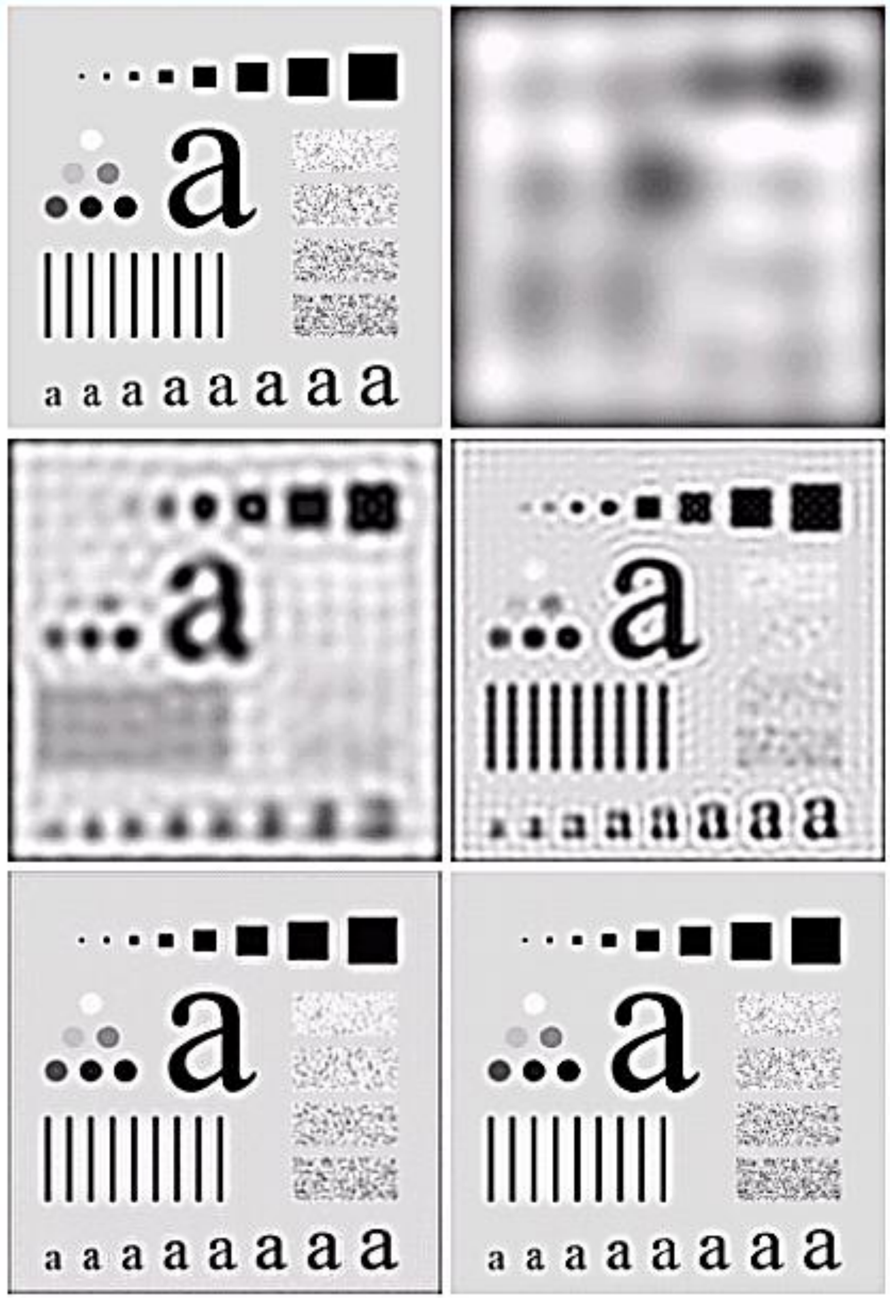


FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.



Original image

Result of filtering with ideal low pass filter of radius 5

Result of filtering with ideal low pass filter of radius 15

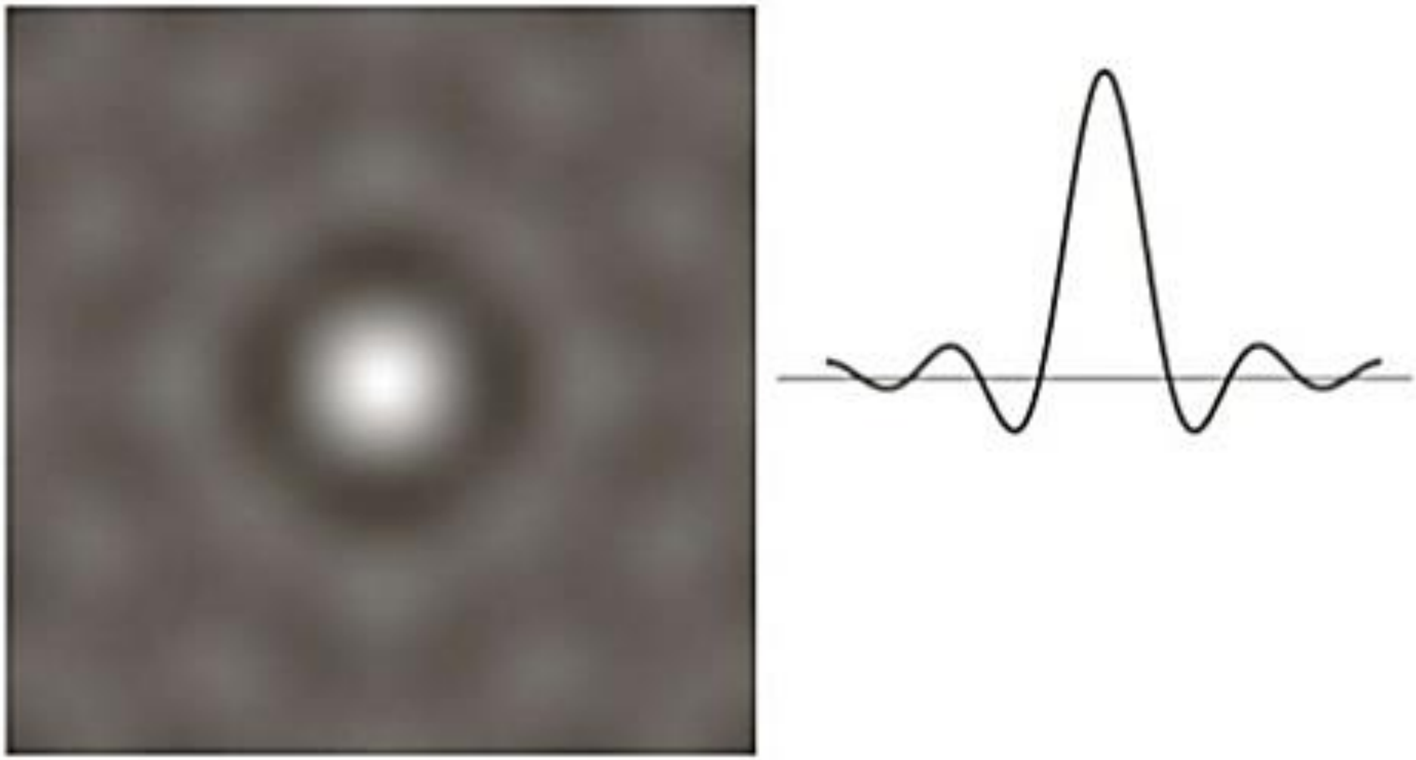
Result of filtering with ideal low pass filter of radius 30

Result of filtering with ideal low pass filter of radius 80

Result of filtering with ideal low pass filter of radius 230

- **Ringing effect: disadvantage of ideal filter**
- ringing artifacts are artifacts that appear as spurious signals ("rings") near sharp transitions in a signal. Visually, they appear as "rings" near edges.

$\uparrow D_0 \longrightarrow \downarrow$ Ringing radius + $\downarrow$ blur



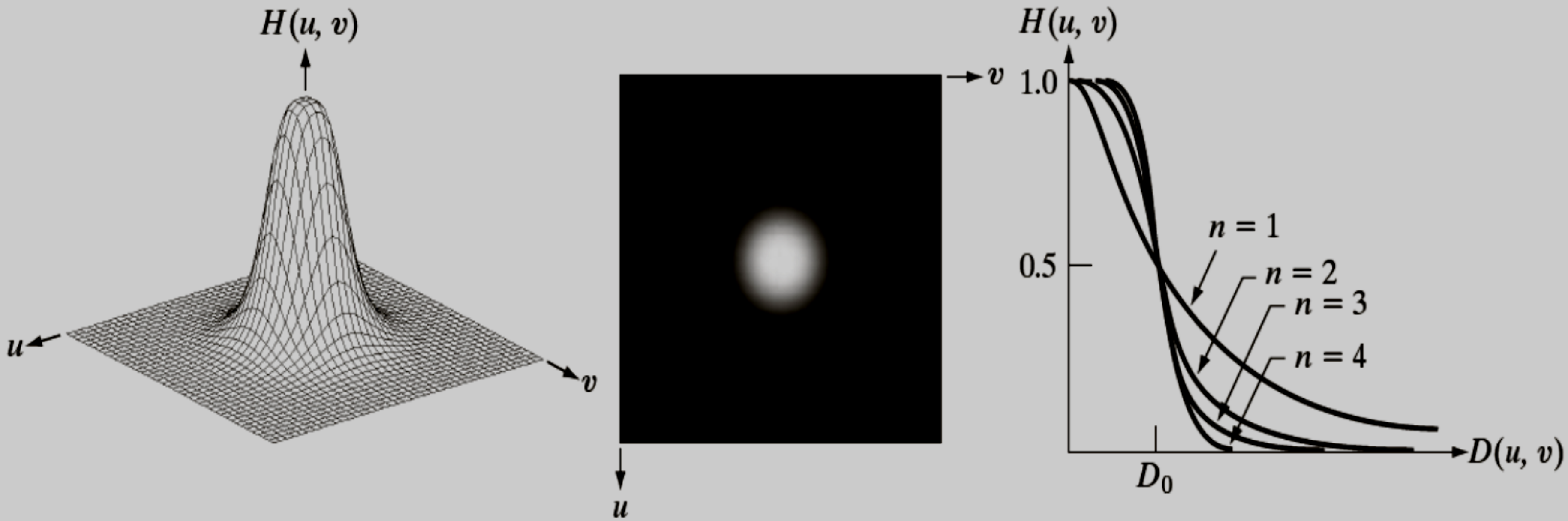
a b

FIGURE 4.43 (a) Representation in the spatial domain of an ILPF of radius 5 and size 1000×1000. (b) Intensity profile of a horizontal line passing through the center of the image.

Butterworth lowpass filter

The transfer function of a Butterworth lowpass filter of order n with cutoff frequency at distance D_0 from the origin is defined as:

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



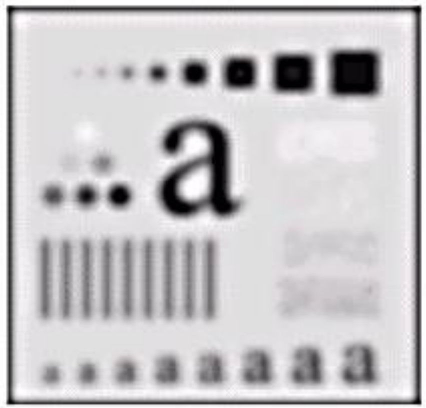
a b c

FIGURE 4.44 (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.

Original image

Result of filtering with Butterworth filter of order 2 and cutoff radius 5

Result of filtering with Butterworth filter of order 2 and cutoff radius 15

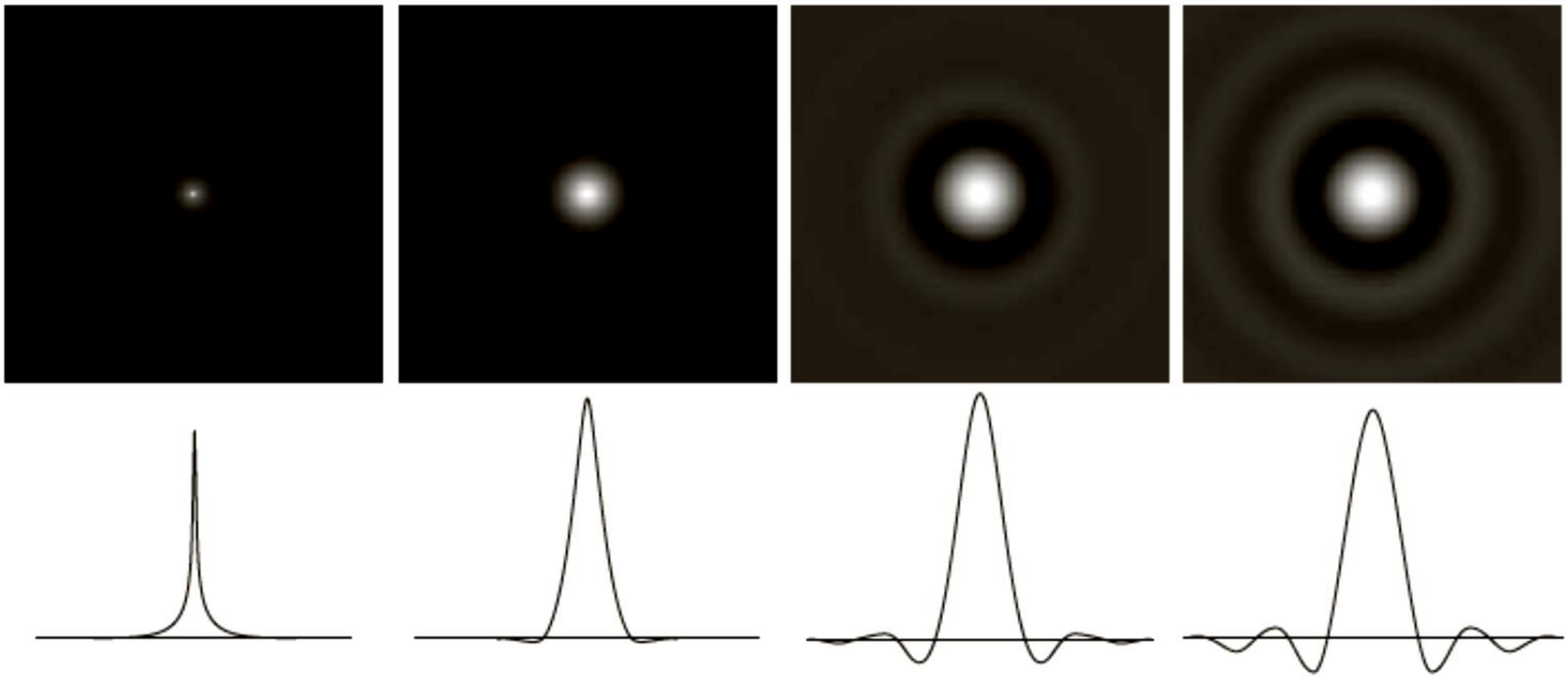


Result of filtering with Butterworth filter of order 2 and cutoff radius 30

Result of filtering with Butterworth filter of order 2 and cutoff radius 80

Result of filtering with Butterworth filter of order 2 and cutoff radius 230

- No ringing effect with n=1 and n=2



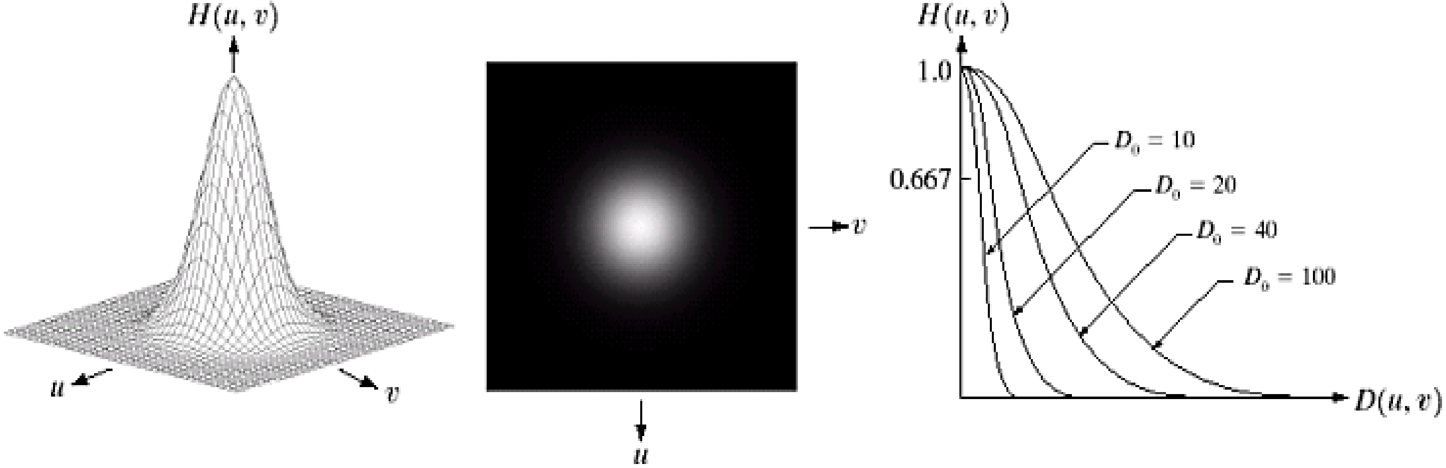
a b c d

FIGURE 4.46 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding intensity profiles through the center of the filters (the size in all cases is 1000 × 1000 and the cutoff frequency is 5). Observe how ringing increases as a function of filter order.

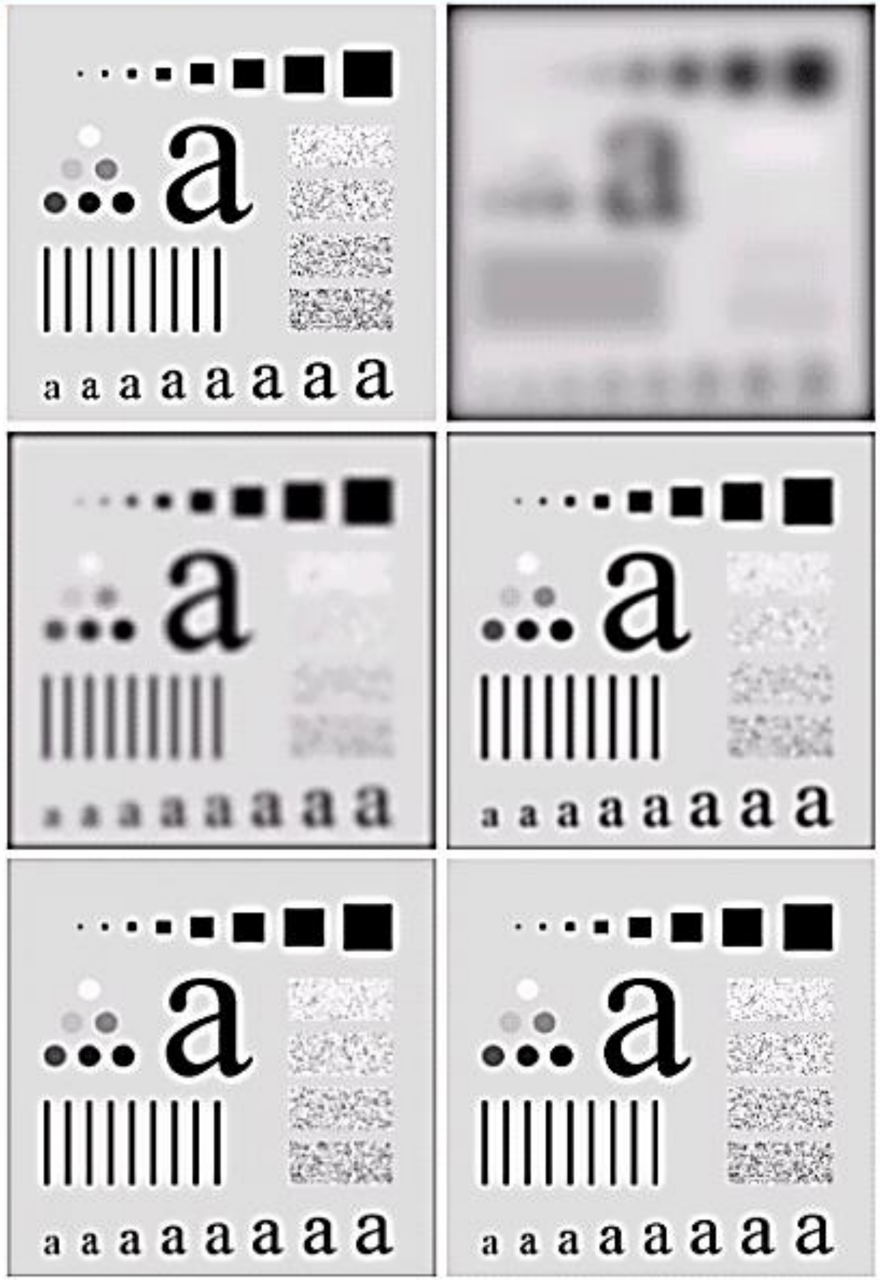
Gaussian lowpass filter

The transfer function of a Gaussian lowpass filter is defined as:

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



Original image



Result of filtering with Gaussian filter with cutoff radius 5

Result of filtering with Gaussian filter with cutoff radius 15

Result of filtering with Gaussian filter with cutoff radius 30

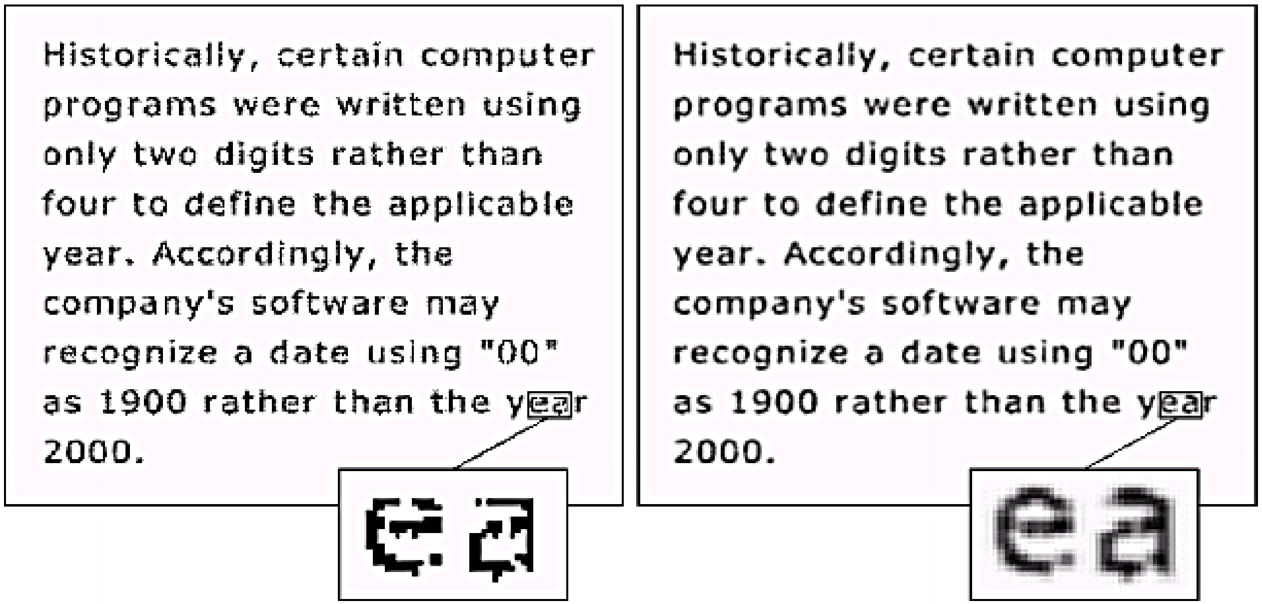
Result of filtering with Gaussian filter with cutoff radius 85

Result of filtering with Gaussian filter with cutoff radius 230

- The main advantage of Gaussian over Butterworth is that there will not be ringing effects no matter what order we choose to work with.

Softer Blurring + no Ringing

A low pass Gaussian filter is used to connect broken text



Different lowpass Gaussian filters used to remove blemishes in a photograph

Image Sharpening- Highpass filtering

Image Sharpening

- Edges and fine detail in images are associated with high frequency components.
- Therefore, attenuating certain low frequency components and preserving the high frequency components can result in image sharpening.
- High pass filters – only pass the high frequencies, drop the low ones.
- High pass frequencies are precisely the reverse of low pass filters, so:

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

- Highpass filters:

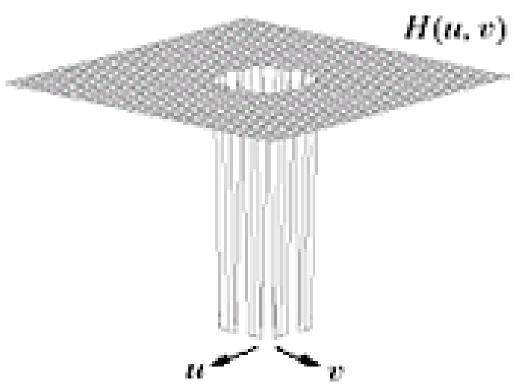
1. Ideal high-pass filter (IHPF)
2. Butterworth high-pass filter (BHPF)
3. Gaussian high-pass filter (GHPF)

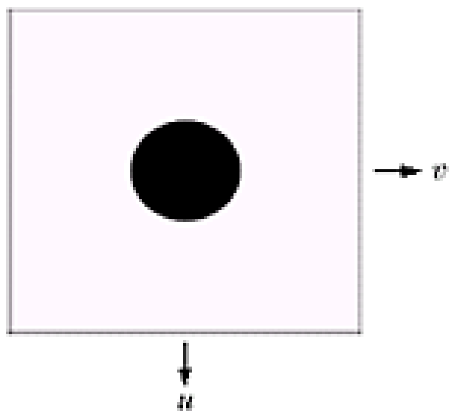
Ideal highpass filter

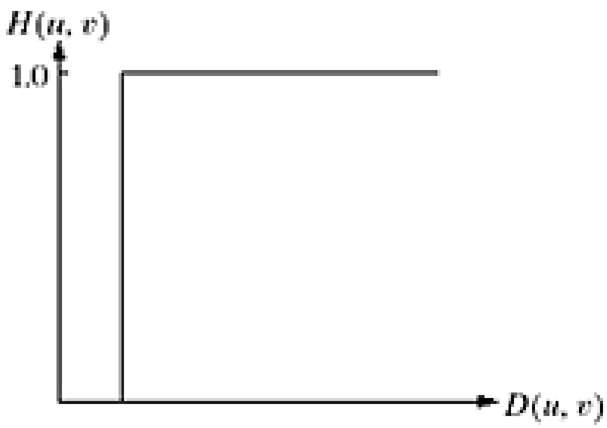
The ideal high pass filter is given as:

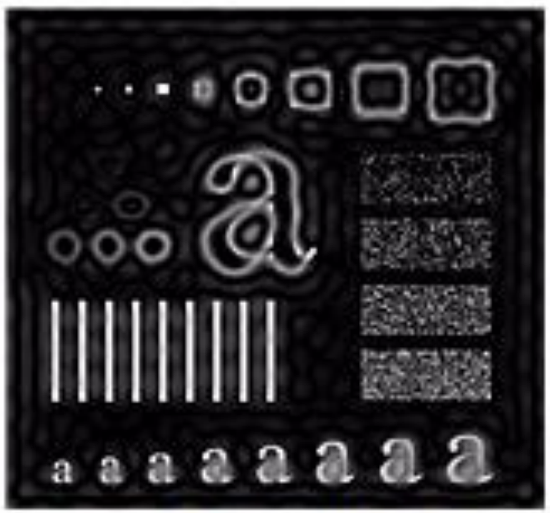
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

where D_0 is the cut off distance as before

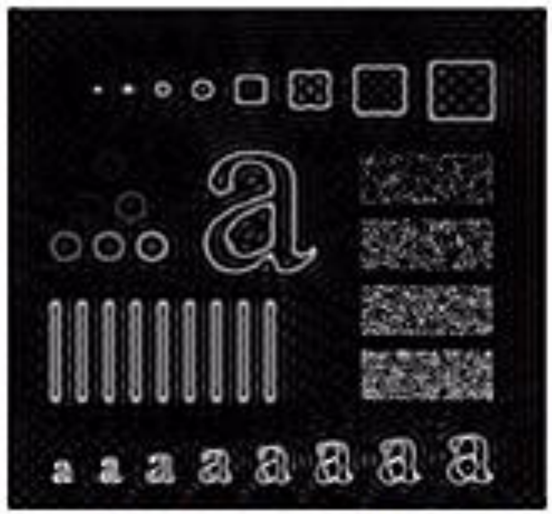








Results of ideal high pass filtering with $D_0 = 15$



Results of ideal high pass filtering with $D_0 = 30$

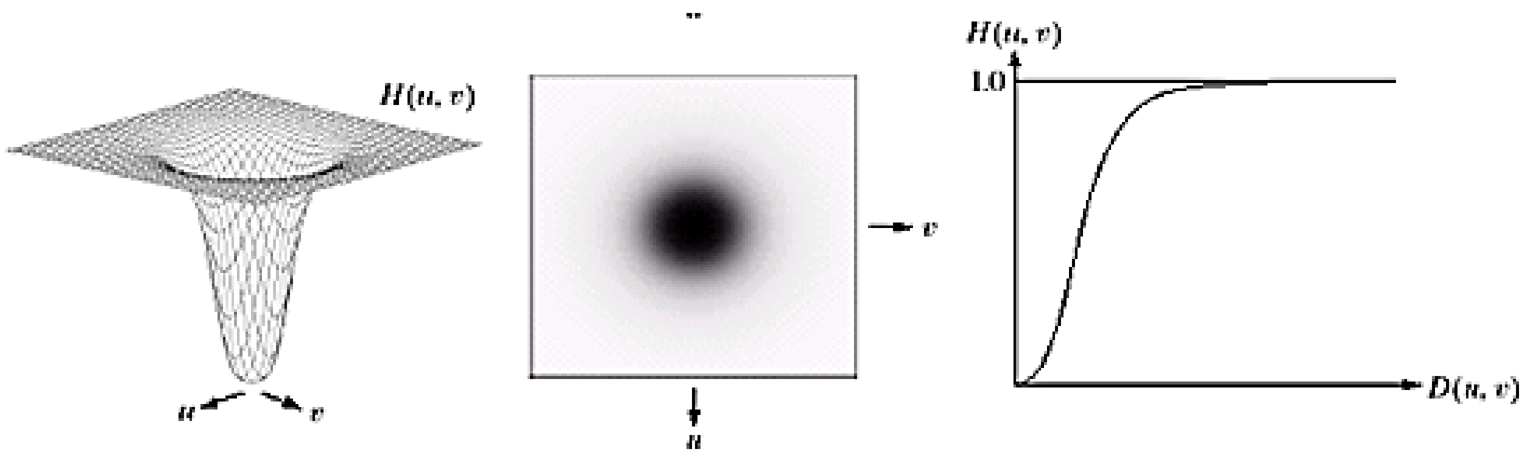
Results of ideal high pass filtering with $D_0 = 80$

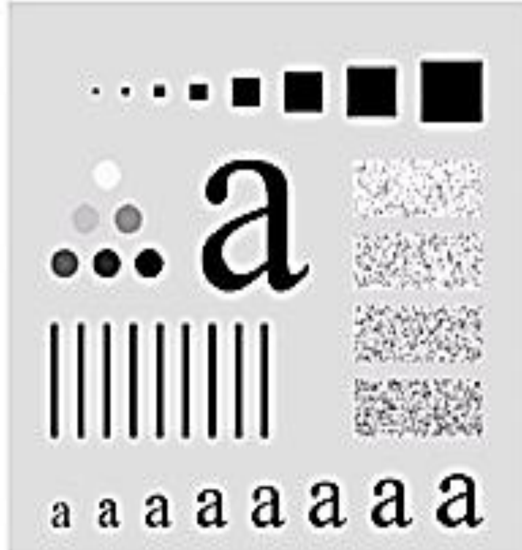
Butterworth highpass filter

The Butterworth high pass filter is given as:

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

where n is the order and D_0 is the cut off distance as before





Results of Butterworth high pass filtering of order 2 with $D_0 = 15$

Results of Butterworth high pass filtering of order 2 with $D_0 = 80$

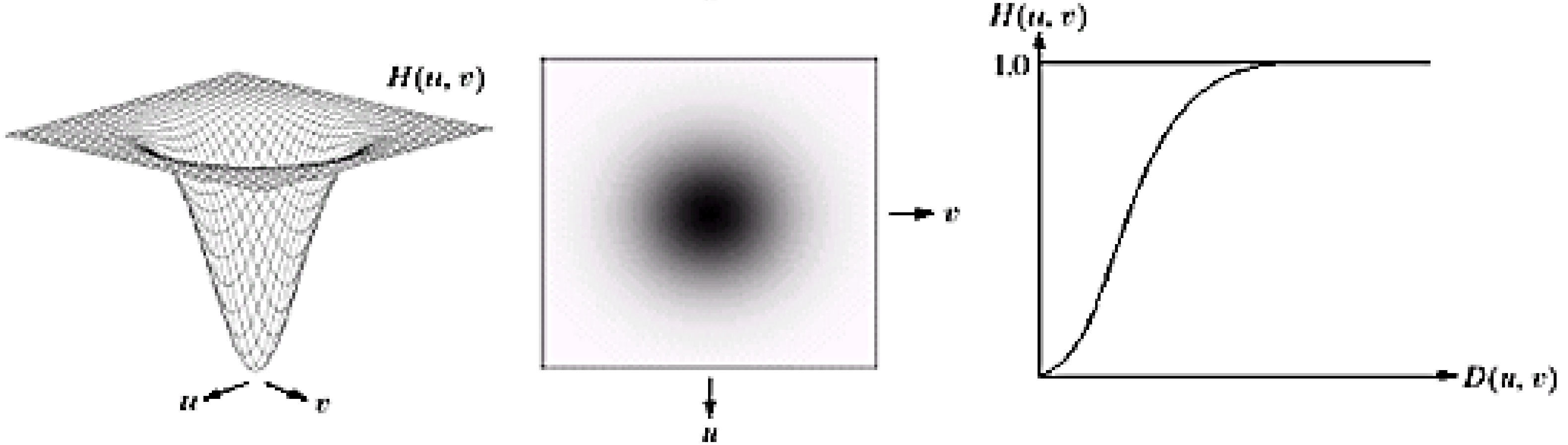
Results of Butterworth high pass filtering of order 2 with $D_0 = 30$

Gaussian High Pass Filter

The Gaussian high pass filter is given as:

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

where D_0 is the cut off distance as before

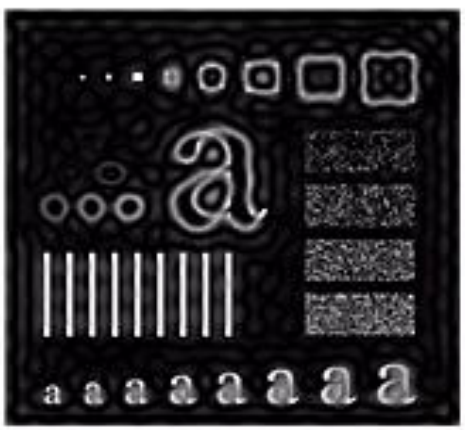


Results of Gaussian high pass filtering with $D_0 = 15$

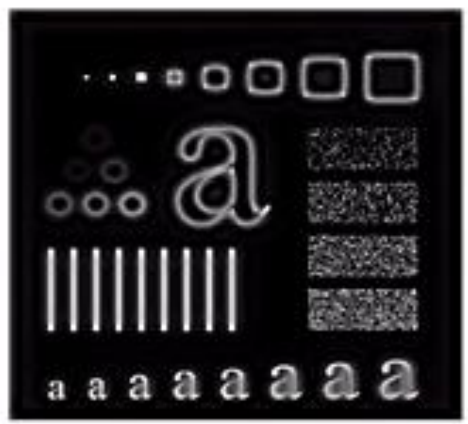
Results of Gaussian high pass filtering with $D_0 = 30$

Results of Gaussian high pass filtering with $D_0 = 80$

Highpass Filter Comparison

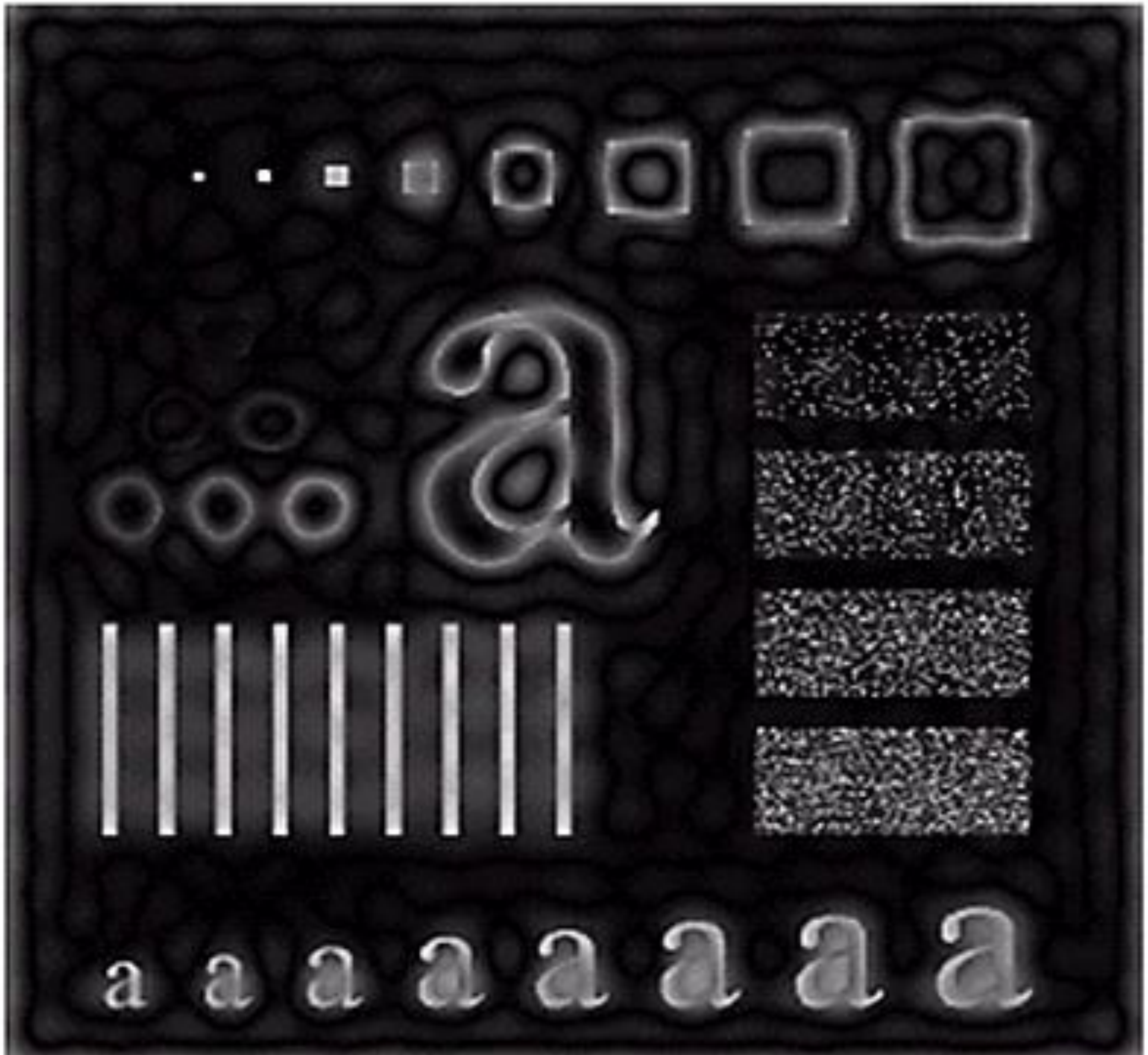


Results of ideal high pass filtering with $D_0 = 15$

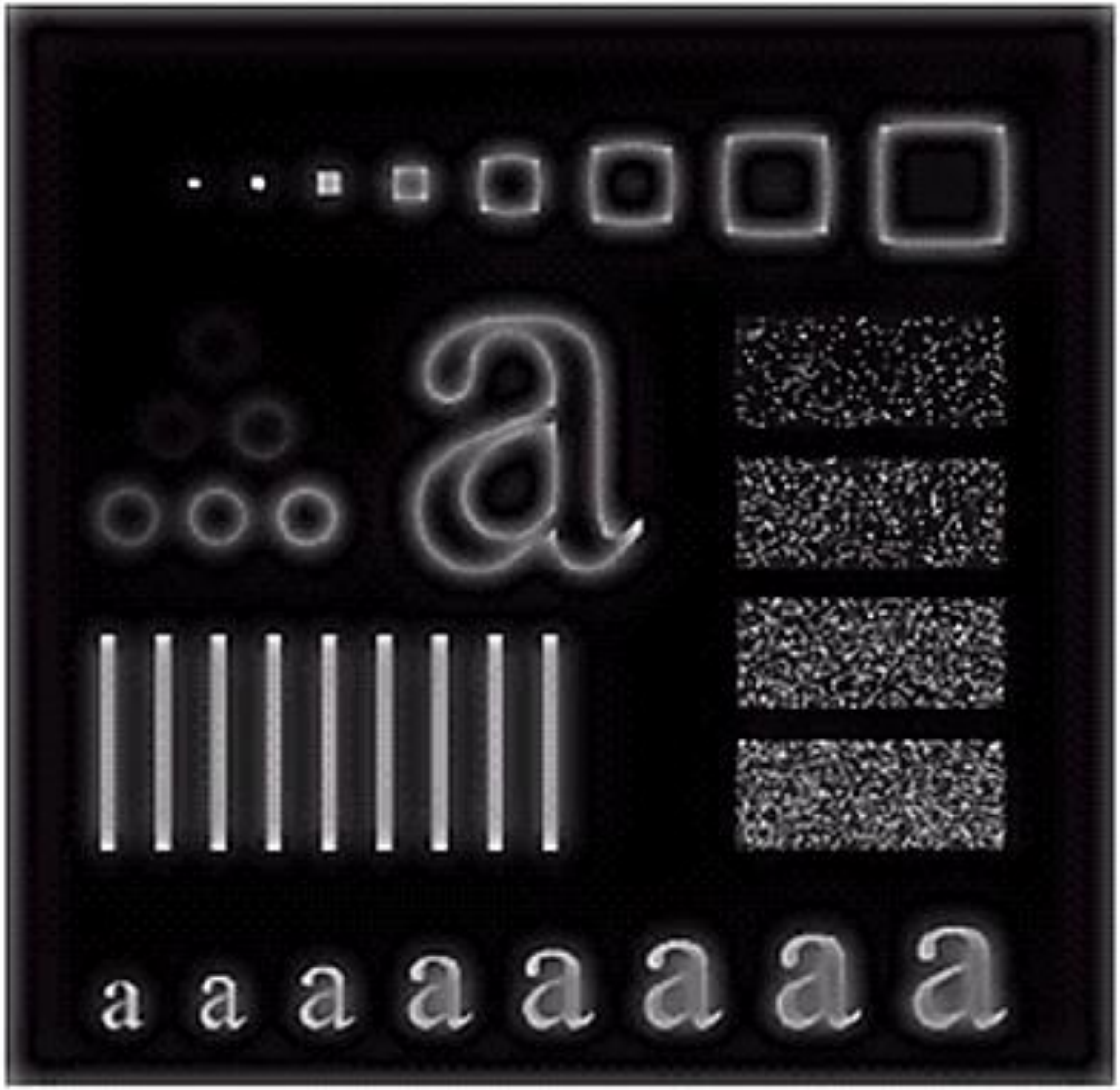


Results of Butterworth high pass filtering of order 2 with $D_0 = 15$

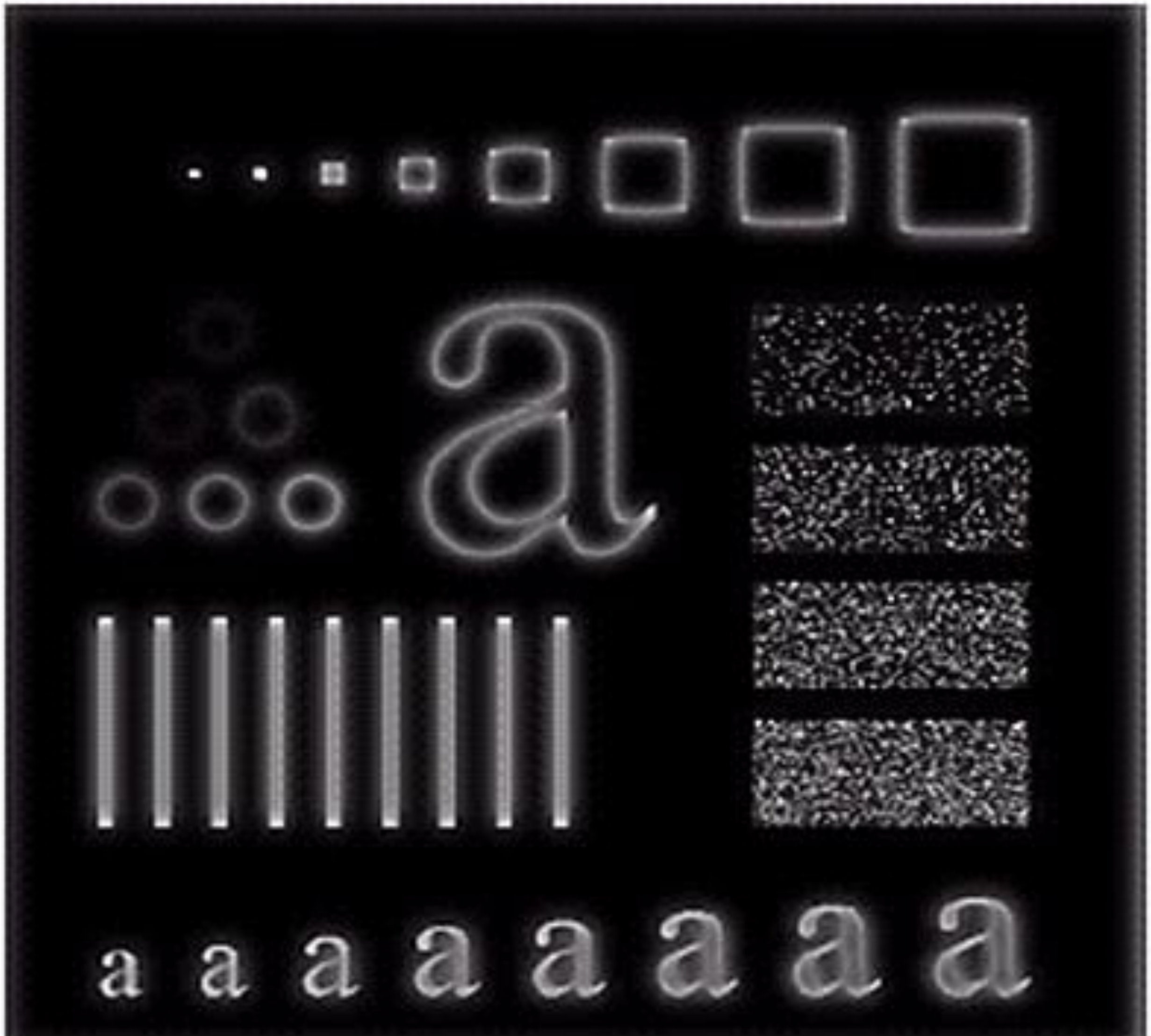
Results of Gaussian high pass filtering with $D_0 = 15$



Results of ideal high pass filtering with $D_0 = 15$



Results of Butterworth high pass filtering of order 2 with D_0 = 15



Results of Gaussian high pass filtering with $D_0 = 15$

a b c

FIGURE 4.57 (a) Thumb print. (b) Result of highpass filtering (a). (c) Result of thresholding (b). (Original image courtesy of the U.S. National Institute of Standards and Technology.)

Laplacian in frequency domain

It can be shown that Laplacian can be implemented in frequency domain using the filter

$$H(u,v) = -4\pi^2(u^2 + v^2)$$

With respect to the center of frequency rectangle

$$\begin{aligned} H(u,v) &= -4\pi^2[(u - P/2)^2 + (v - Q/2)^2] \\ &= -4\pi^2 D^2(u,v) \end{aligned}$$

Laplacian filtered image in spatial domain is obtained by computing the inverse Fourier transform of $H(u,v)F(u,v)$ as

$$\nabla^2 f(x,y) = \Im^{-1}\{H(u,v)F(u,v)\}$$

Subtracting the Laplacian from the original image gives you the enhanced image

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

Example


Unsharp masking & highboost filtering

The mask for unsharp masking in frequency domain is given by

$$g_{\text{mask}}(x, y) = f(x, y) - f_{\text{LP}}(x, y)$$

where

$$f_{\text{LP}}(x, y) = \Im^{-1}[H_{\text{lp}}(u, v)F(u, v)]$$

* $H_{\text{LP}}(u, v)$ is a lowpass filter
* $F(u, v)$ is the Fourier transform of $f(x, y)$
* $f_{\text{LP}}(x, y)$ is smoothed image version of $\bar{f}(x, y)$

We have

$$g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y)$$

* We have unsharp masking when $k = 1$ and highboost filtering if $k > 1$

$$g(x,y) = f(x,y) + k(f(x,y) - f_{LP}(x,y))$$

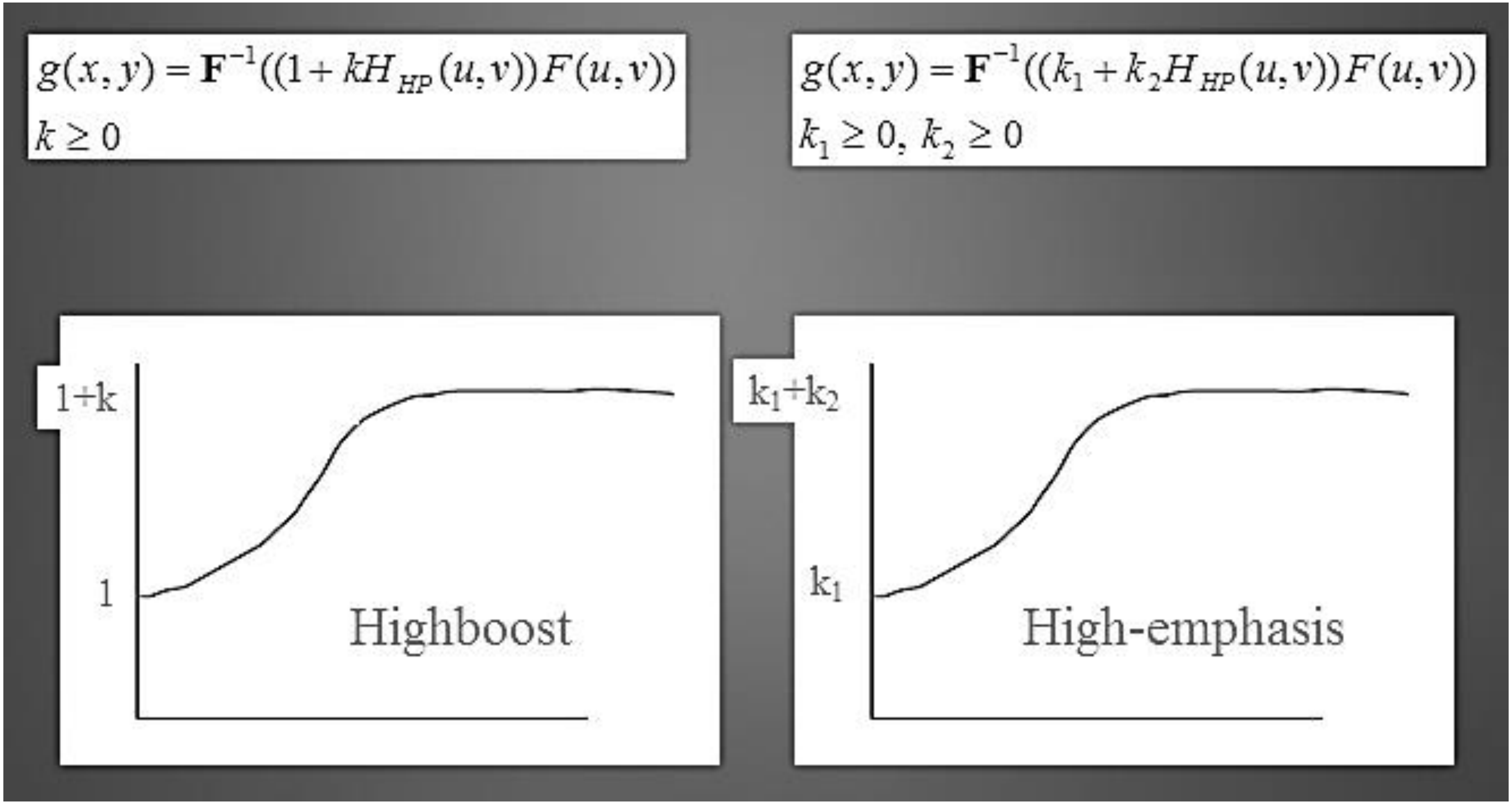
$$G(u,v) = \mathbf{F}\{f(x,y) + k(f(x,y) - f_{LP}(x,y))\}$$

$$= F(u,v) + k(F(u,v) - H_{LP}(u,v)F(u,v)) =$$

$$= [1 + k(1 - H_{LP}(u,v))]F(u,v) = [1 + kH_{HP}(u,v)]F(u,v)$$

$$so: G(u,v) = [1 + kH_{HP}(u,v)]F(u,v) \ or \ g(x,y) = \mathbf{F}^{-1}\{[1 + kH_{HP}(u,v)]F(u,v)\}$$

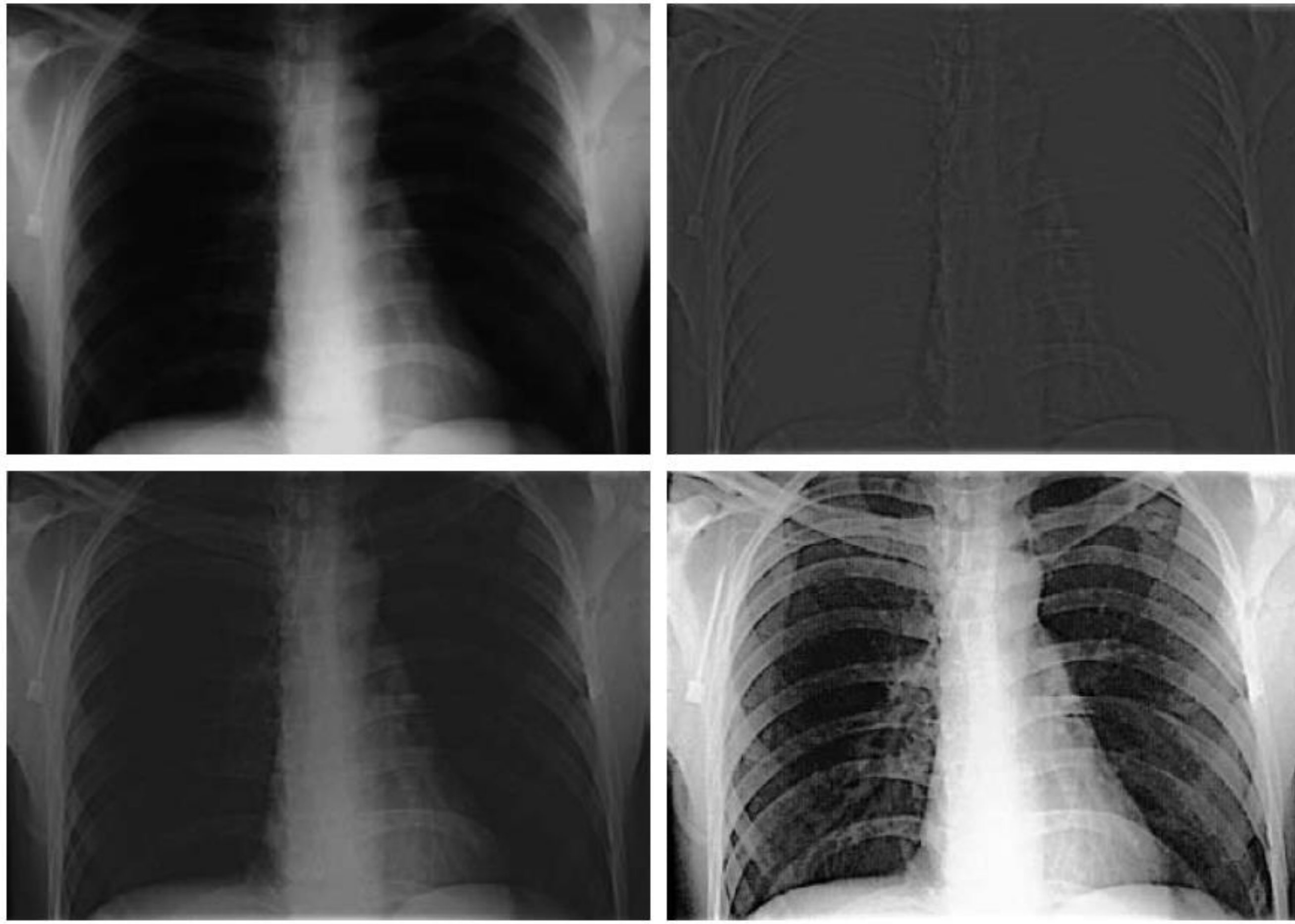
High-Frequency-Emphasis Filters



K1>=0 gives the controls of the offset from origin
k2>=0 controls the high frequencies

High-Frequency Emphasis filtering Using Gaussian filter
$k_1=0.5$, $k_2=0.75$

$D_0=40$



a b
c d

FIGURE 4.59 (a) A chest X-ray image. (b) Result of highpass filtering with a Gaussian filter. (c) Result of high-frequency-emphasis filtering using the same filter. (d) Result of performing histogram equalization on (c). (Original image courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)

Selective filtering

1. Bandreject filters
2. Bandpass filters
3. Notch filter

Bandreject filters

- A band reject filter is useful when the general location of the noise in the frequency domain is known.
- A band reject filter blocks frequencies within the chosen range and lets frequencies outside of the range pass through.
- In filter equations, D_0 is the radial center of the band, W is the width of the band.

1. Ideal Bandreject filter:

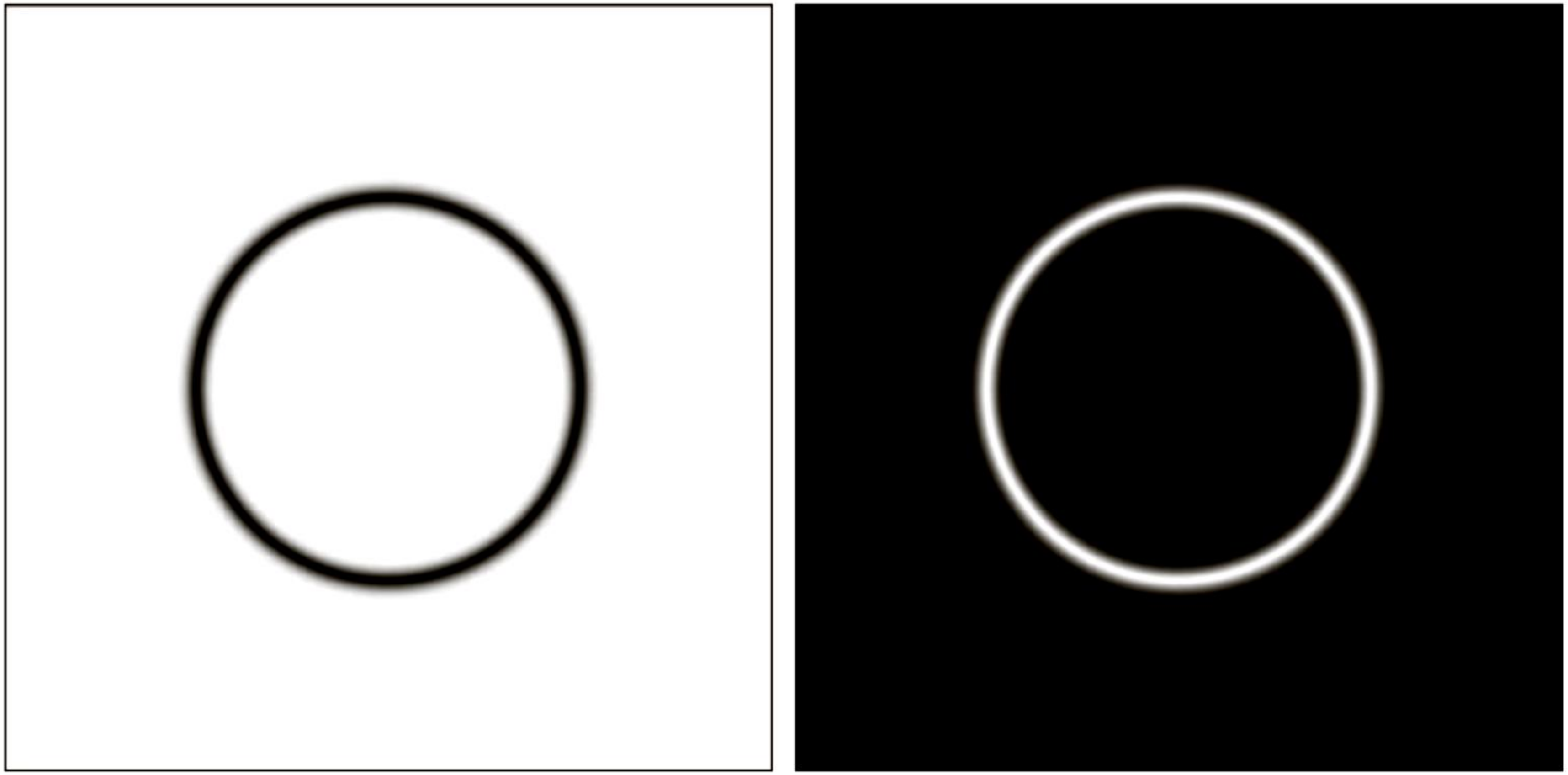
$$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \dfrac{W}{2} \le D \le D_0 + \dfrac{W}{2} \\ 1 & \text{Otherwise} \end{cases}$$

2. Butterworth bandreject filter:

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

3. Gaussian bandreject filter:

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



Bandreject and Bandpass Gaussian filter

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Ideal

Butterworth

Gaussian

Bandpass filters

$$H_{BP}(u,v)=1-H_{BR}(u,v)$$

Notch filter

- Reject (or pass) frequencies in a predefined neighborhood about the center of the frequency rectangle.
- Constructed as products of highpass filters whose centers have been translated to the centers of the notches.
- A notch filter a center at (u0,v0) must have a corresponding notch at location (-u0,-v0).

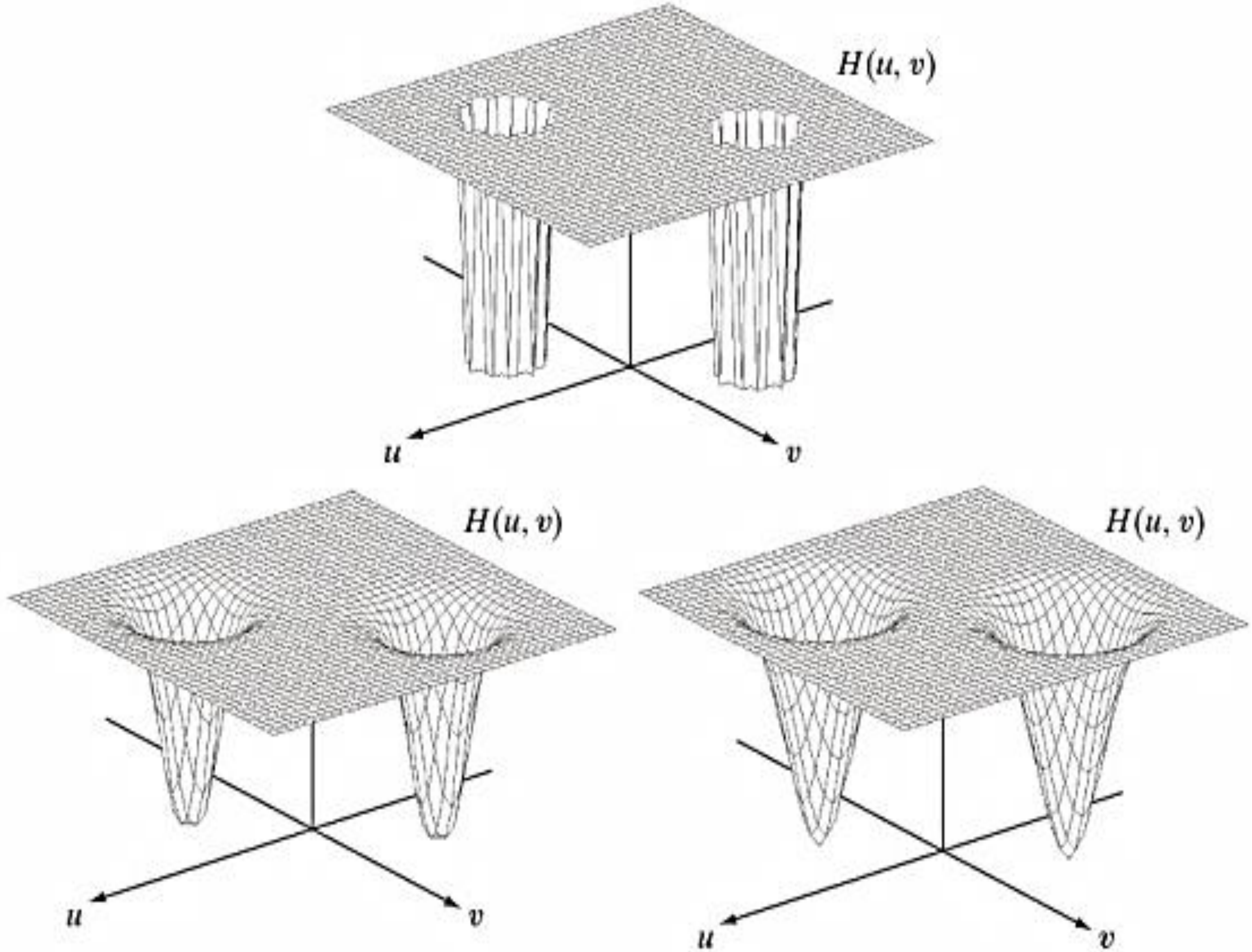


FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

Gonzalez & Woods - Digital Image Processing (3rd Edition)

$$H_{NR}(u,v) = \prod_{k=1}^{Q} H_k(u,v)H_{-k}(u,v)$$

centre at (u_k, v_k)

centre at $(-u_k, -v_k)$

Distances computations:

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

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

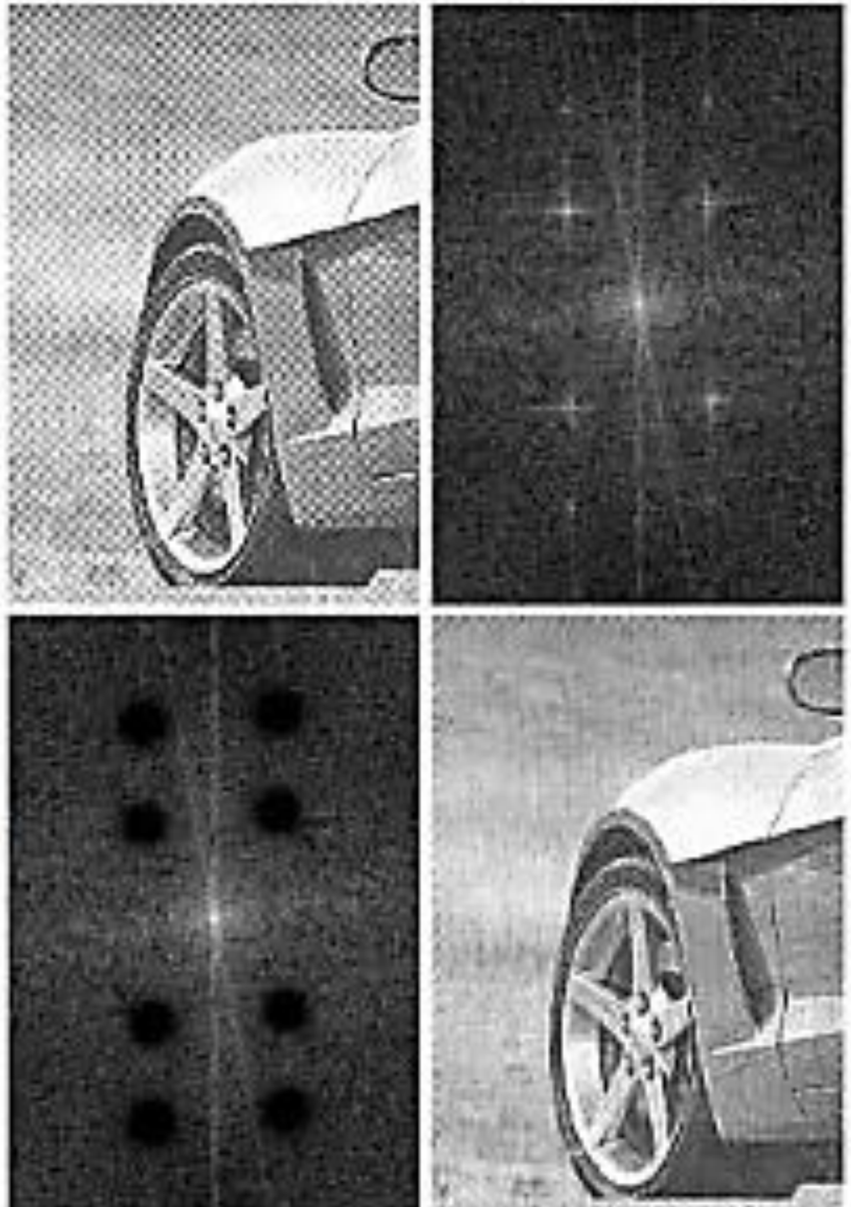
Example: Butterworth notch reject filter of order n, containing 3 notch pairs:

$$H_{NR}(u,v) = \prod_{k=1}^{3}\left[\frac{1}{1+[D_{0k}/D_k(u,v)]^{2n}}\right]\left[\frac{1}{1+[D_{0k}/D_{-k}(u,v)]^{2n}}\right]$$

A *Notch Pass filter* (NP) is obtained from a *Notch Reject filter* (NR) using:

$$H_{NP}(u,v) = 1 - H_{NR}(u,v)$$

Newspaper image FT Spectrum

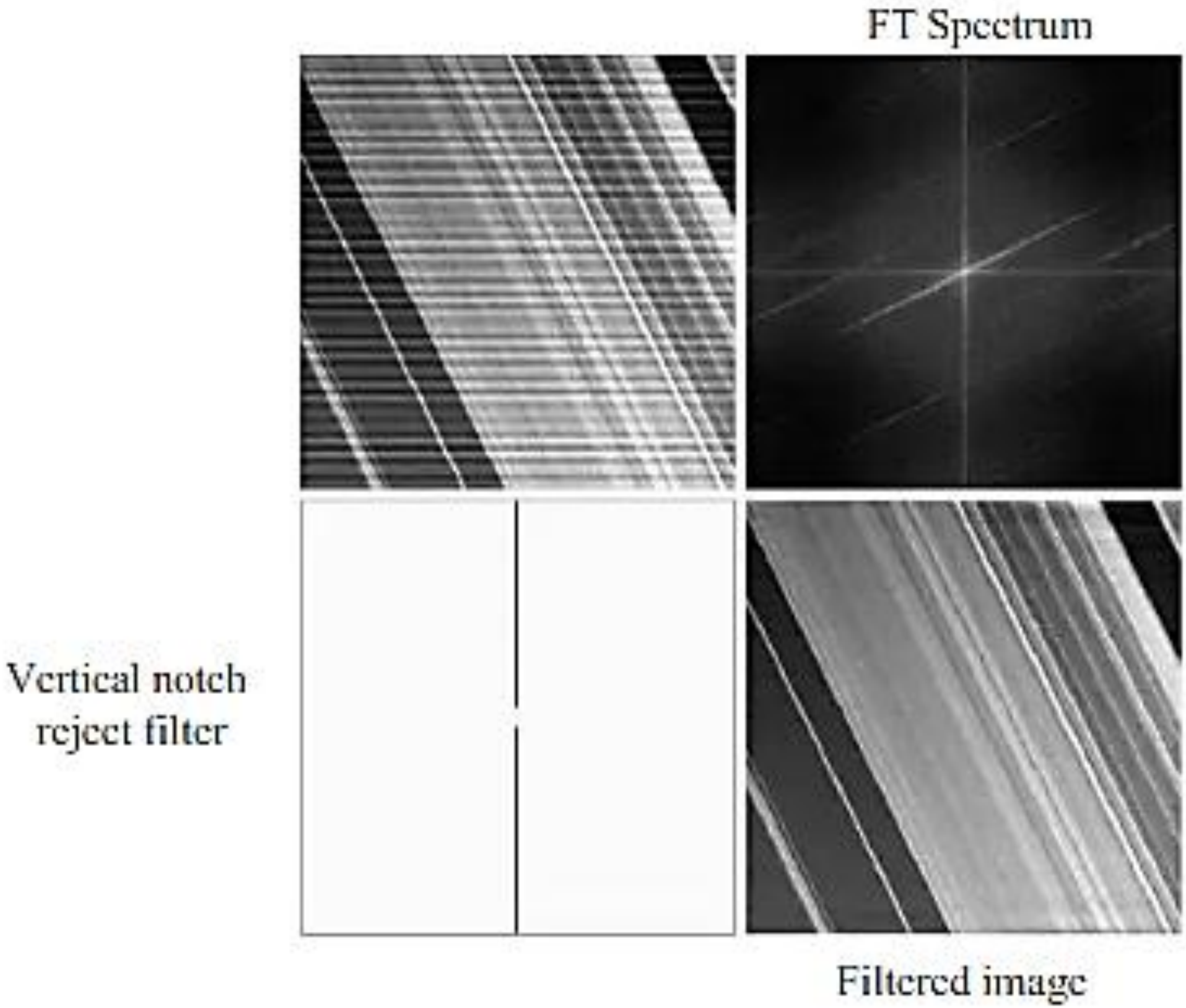


a b
c d

FIGURE 4.64 (a) Sampled newspaper image showing a moiré pattern. (b) Spectrum. (c) Butterworth notch reject filter multiplied by the Fourier transform. (d) Filtered image.

Butterworth notch reject filter multiplied by FT

Filtered image


FT Spectrum
Vertical notch
reject filter
Filtered image