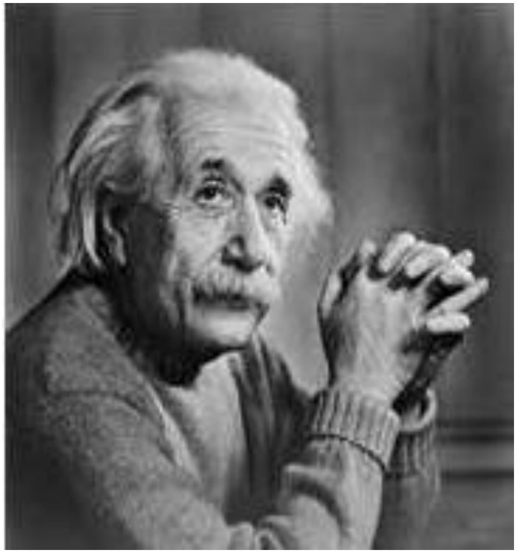
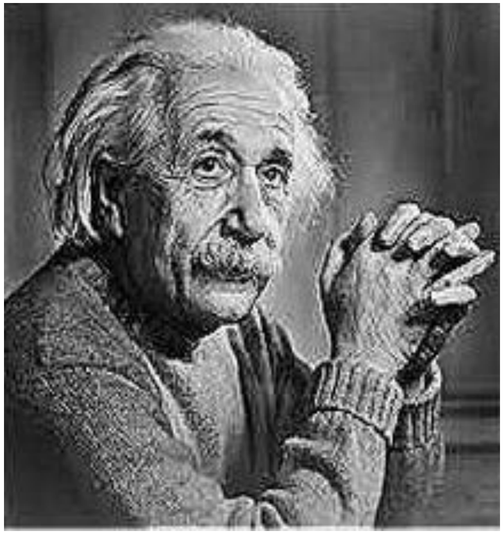
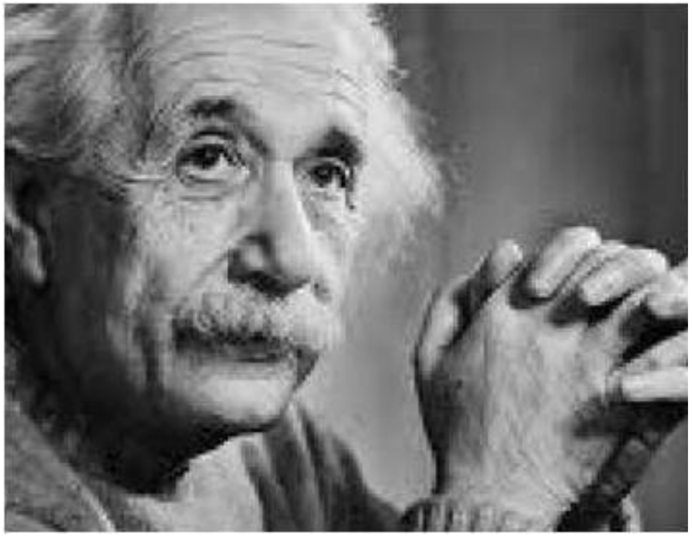
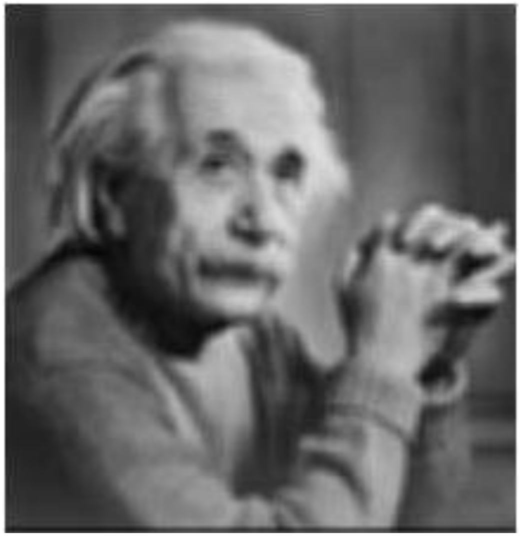
Digital Image Processing

What is DIP?

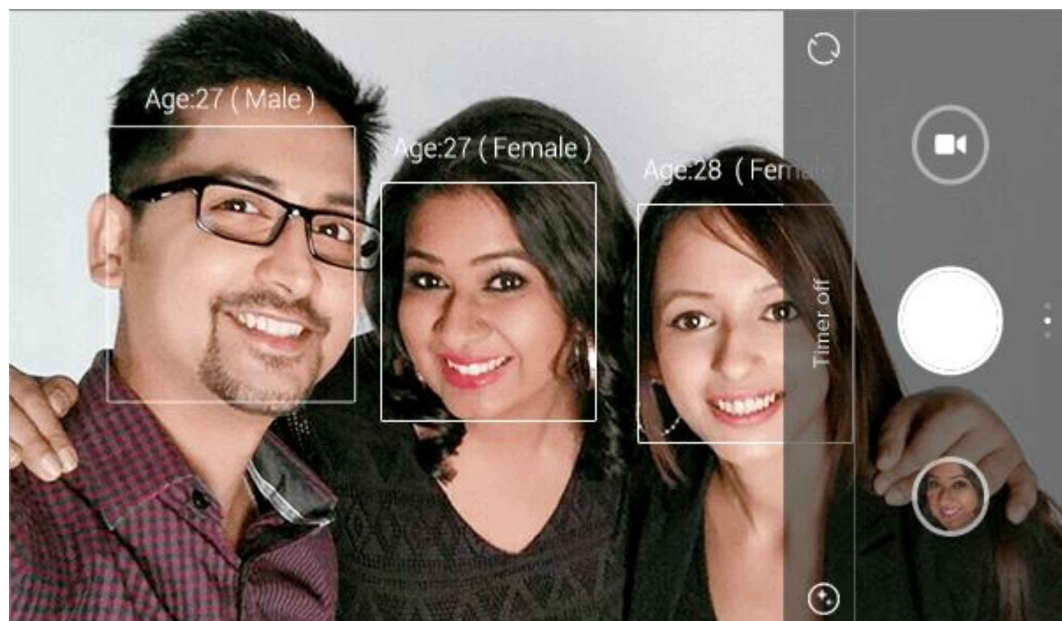
- Digital image processing deals with manipulation of digital images through a digital computer.

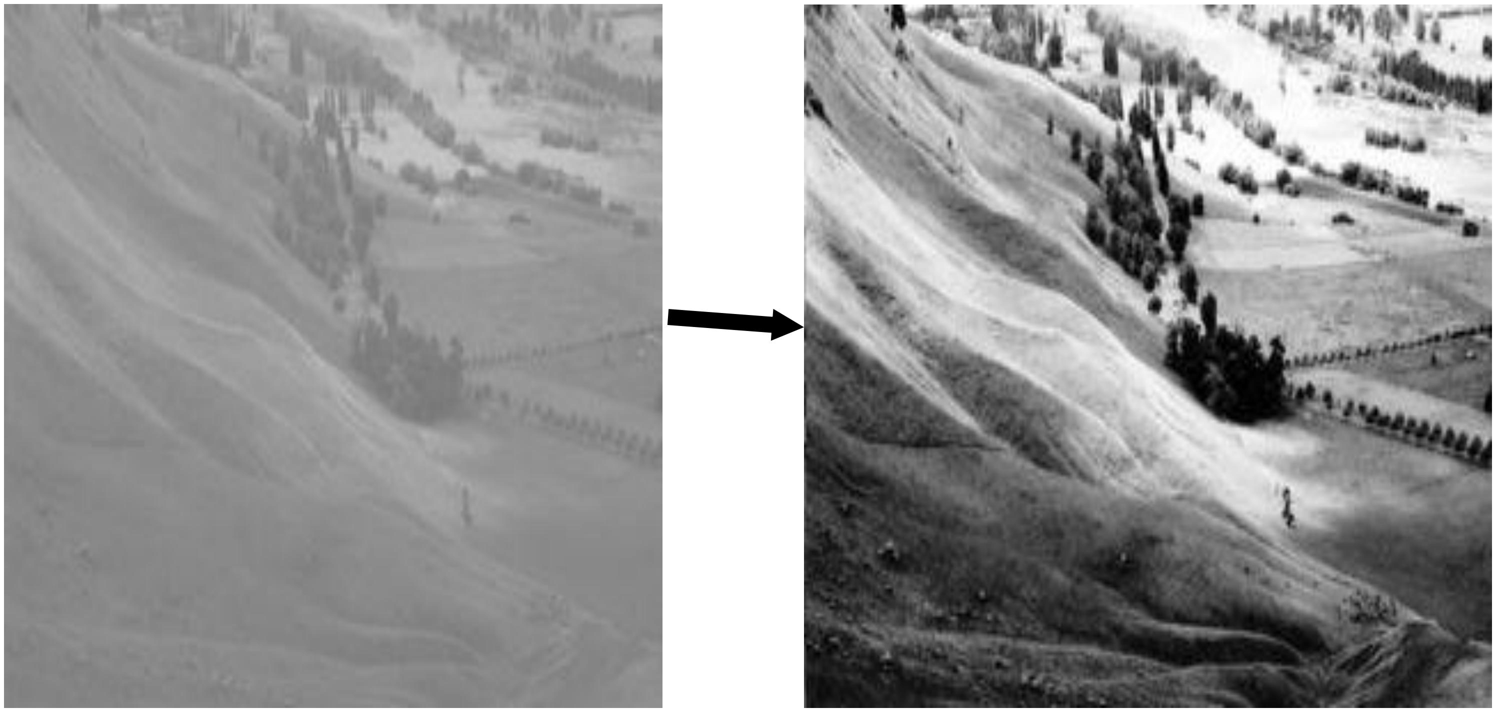



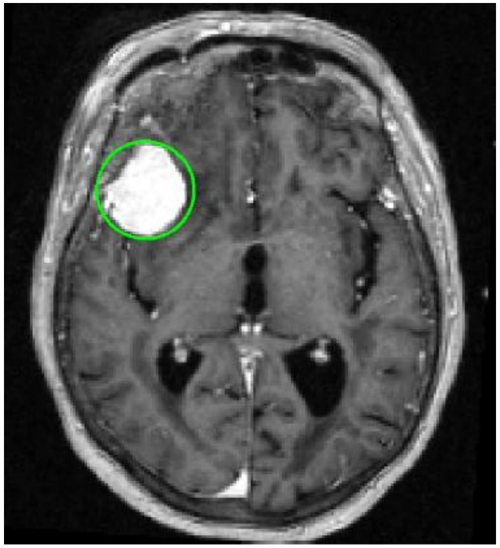

After
Before


After
Before


Age:27 (Male)
Age:27 (Female)
Timer off



Other applications:



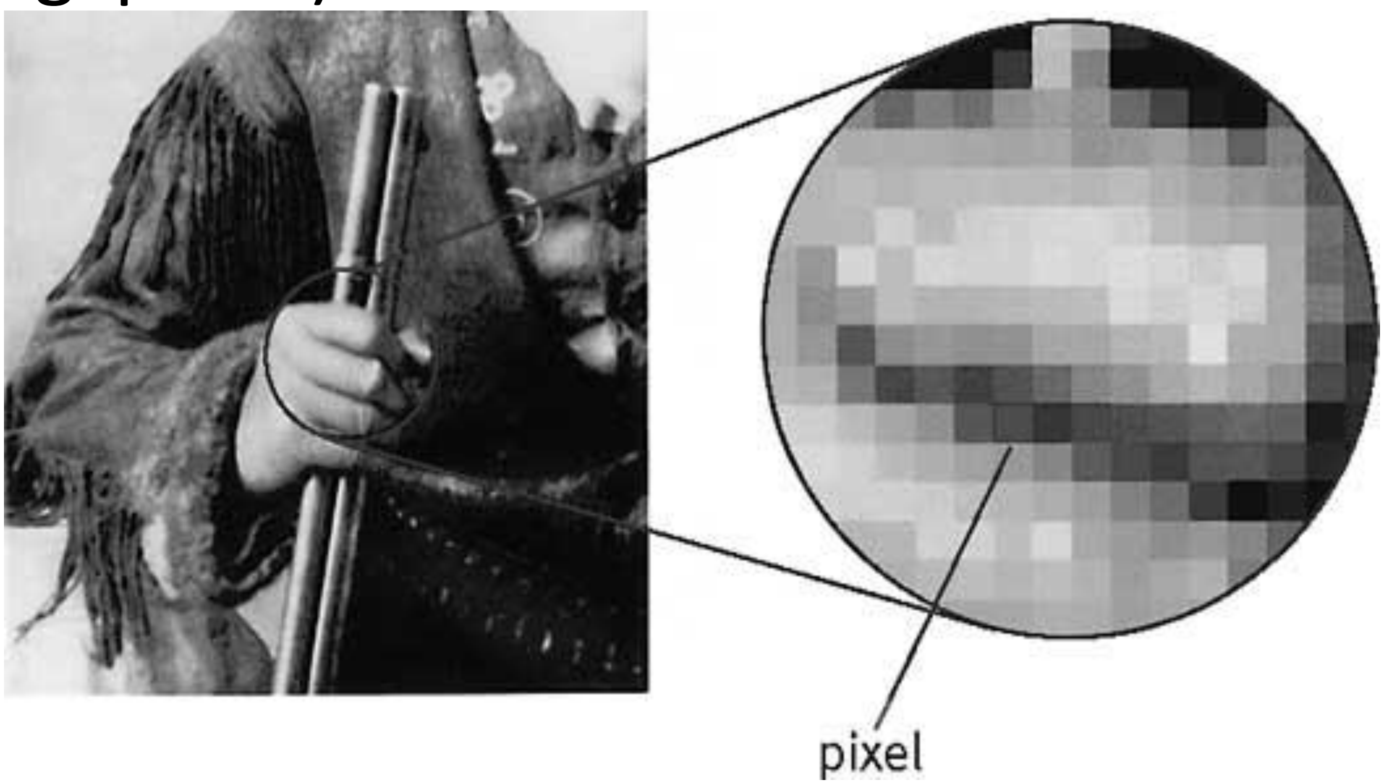
- Security: biometrics

- Medical Imaging

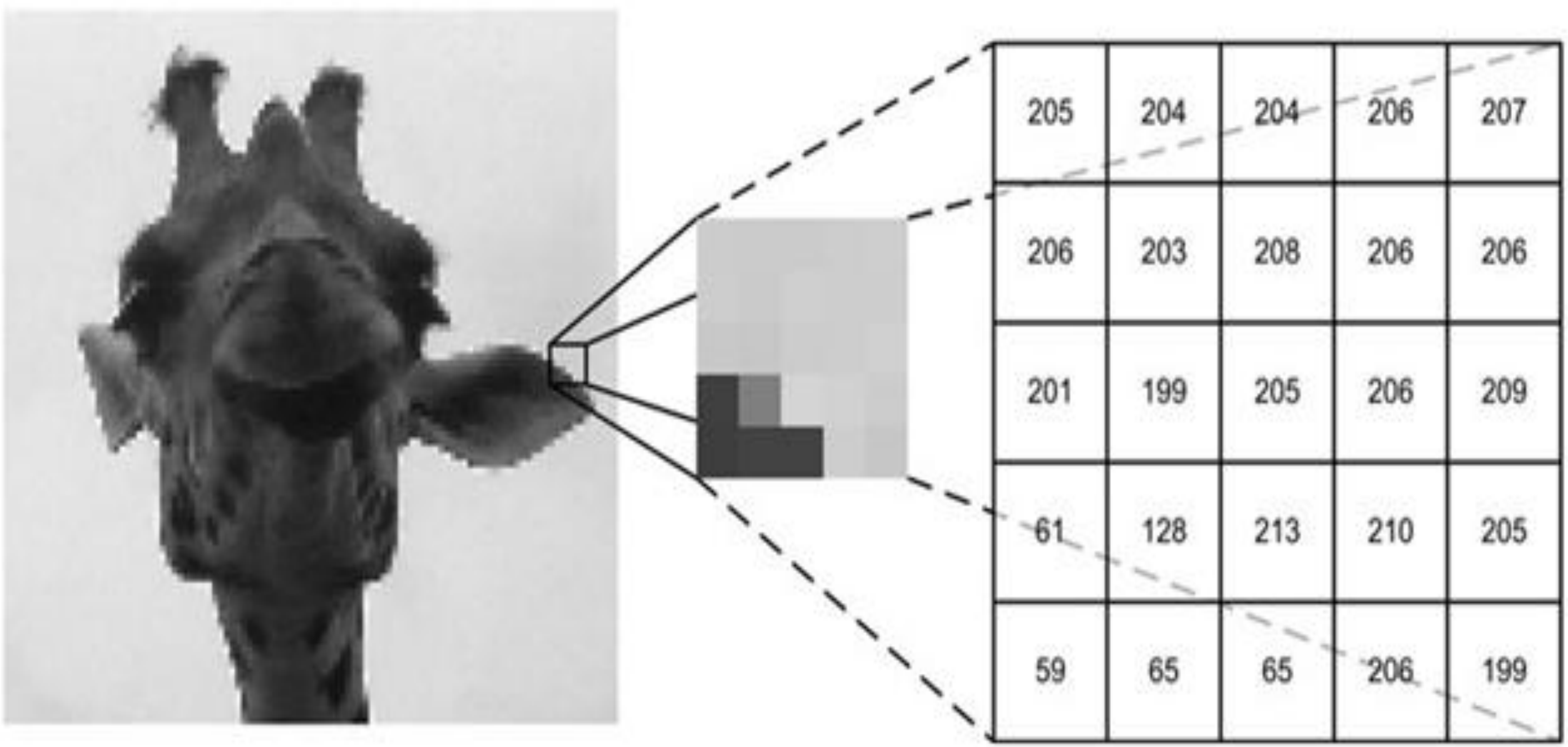
And many more!!

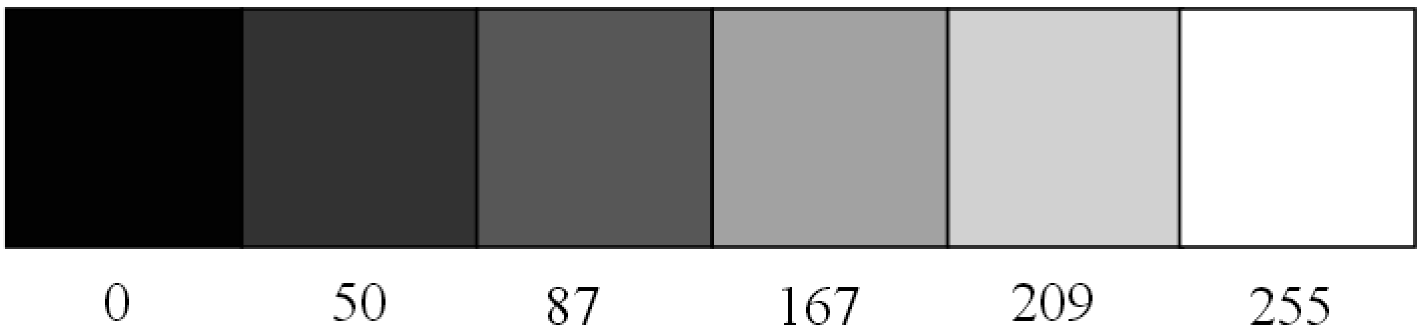
- Resolution of an Image????
- 1024 x 768, 1280 x 1024, 1920 x 1080 ???

- Resolution :Number of pixels in an image.
- 2048 x 1536 : An image that is 2048 pixels wide and 1536 pixels high.
- It contains (multiply) 3,145,728 pixels (or 3.1 Megapixels).



Information on Pixels:




0
50
87
167
209
255

Manipulation of pixel value:

- Spatial domain processing
  - Direct manipulation of pixel

- Transform domain processing
  - Transform the image in suitable domain
  - Perform suitable operation
  - Inverse transform

Which one is computationally efficient?

Spatial Domain Processing:

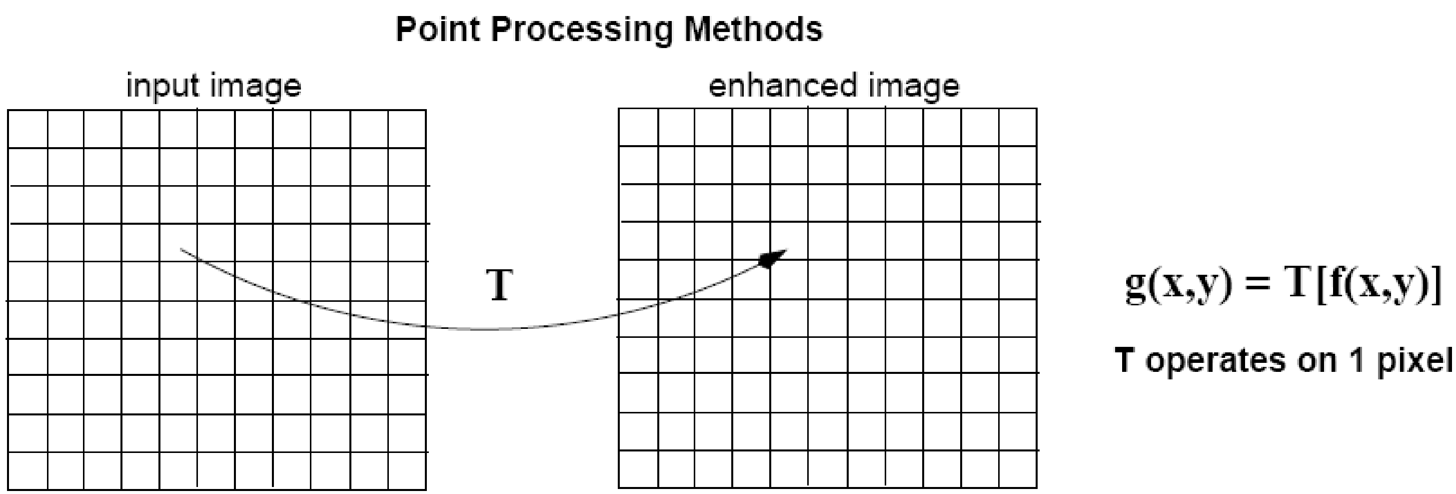
Intensity transformation

&

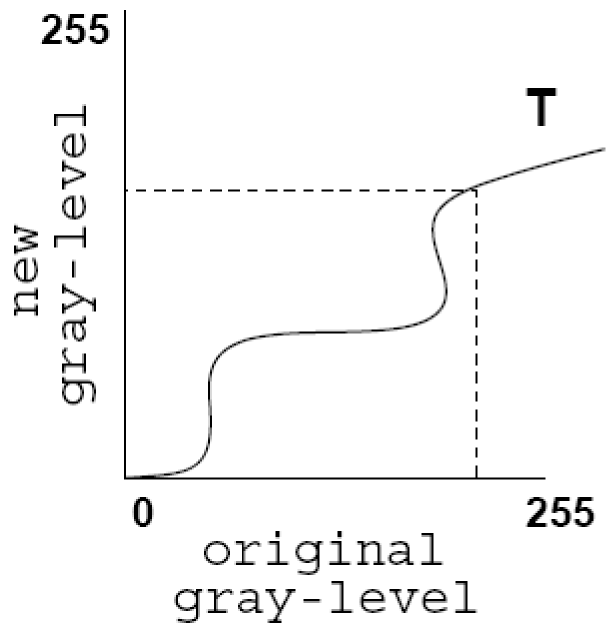
Spatial filtering

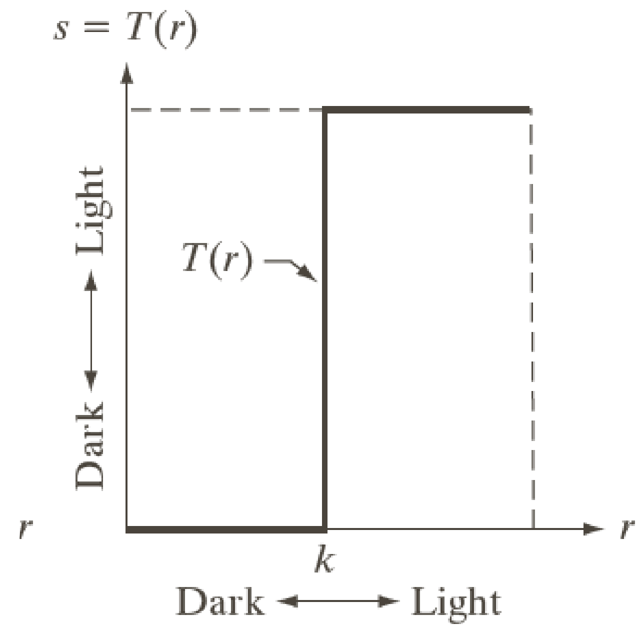
Spatial Domain processing:

- Intensity transformations:



Convert a given pixel value to a new pixel value based on some predefined function.





Which operation is this?

Some Notations:

- ‘r’ is Input image.
- ‘s’ is Output image.
- s(x,y) and r(x,y) represents arbitrary pixel value of position (x,y) in output and input image respectively.
- No of intensity levels present in an image is

  0 toL-1.
- S(x,y)=T(r(x,y)) where T is intensity transformation function .
- This can be represented as s=T(r) for simplicity.

Spatial filtering:

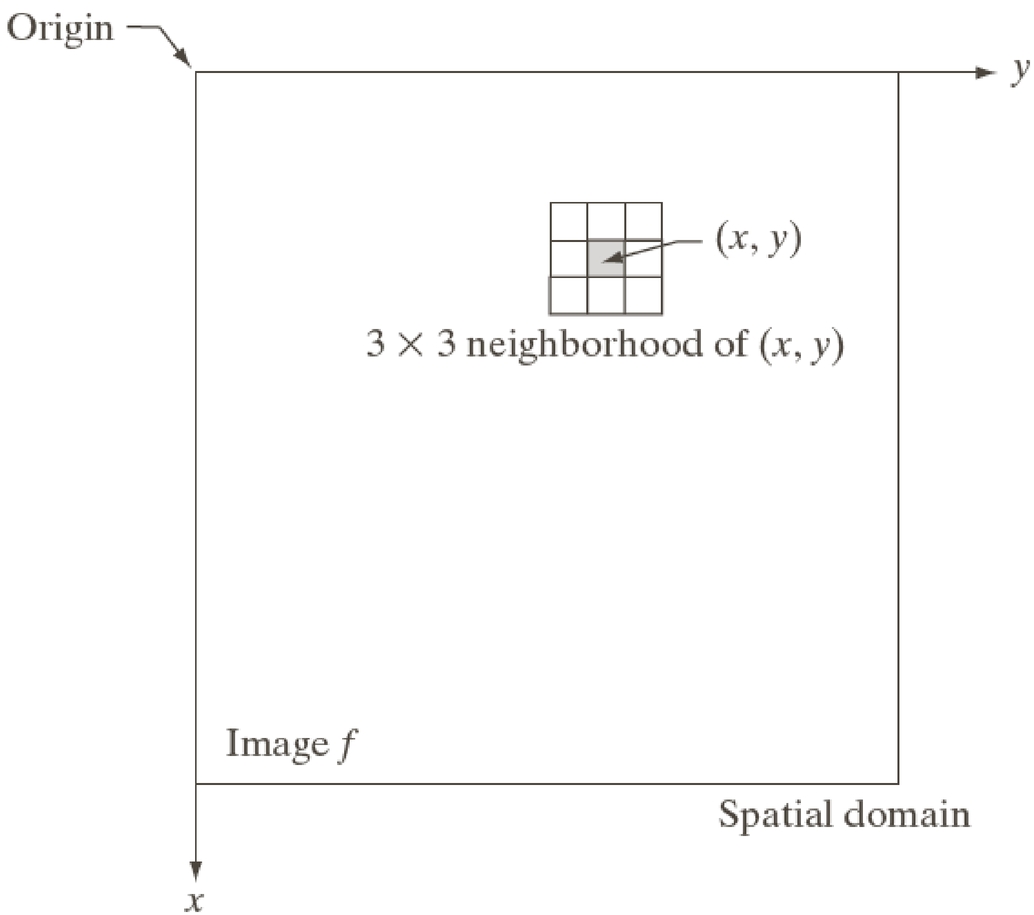
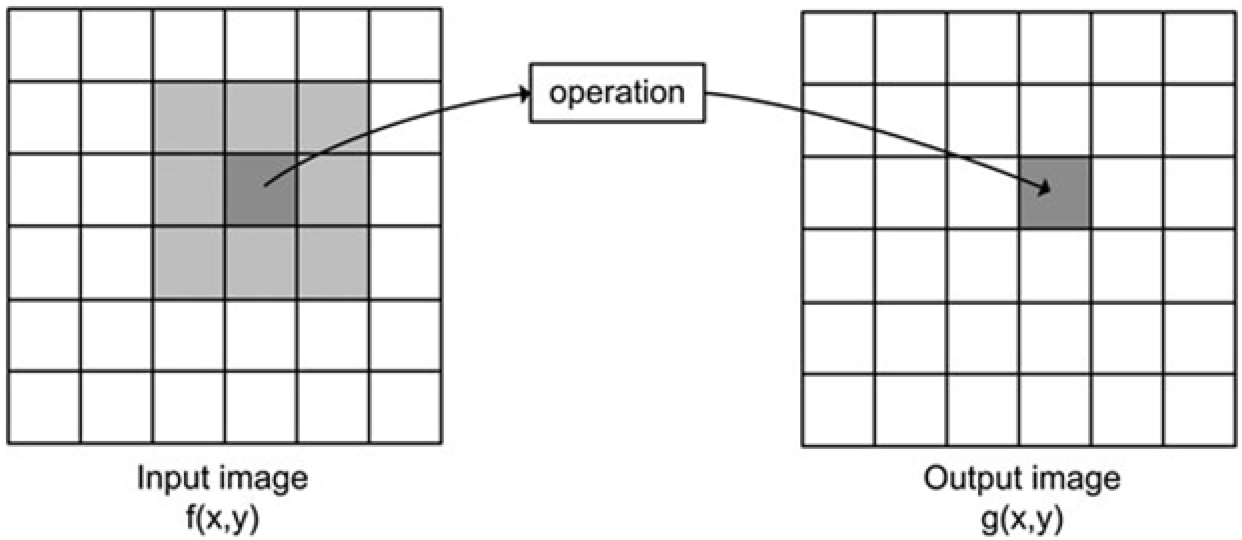
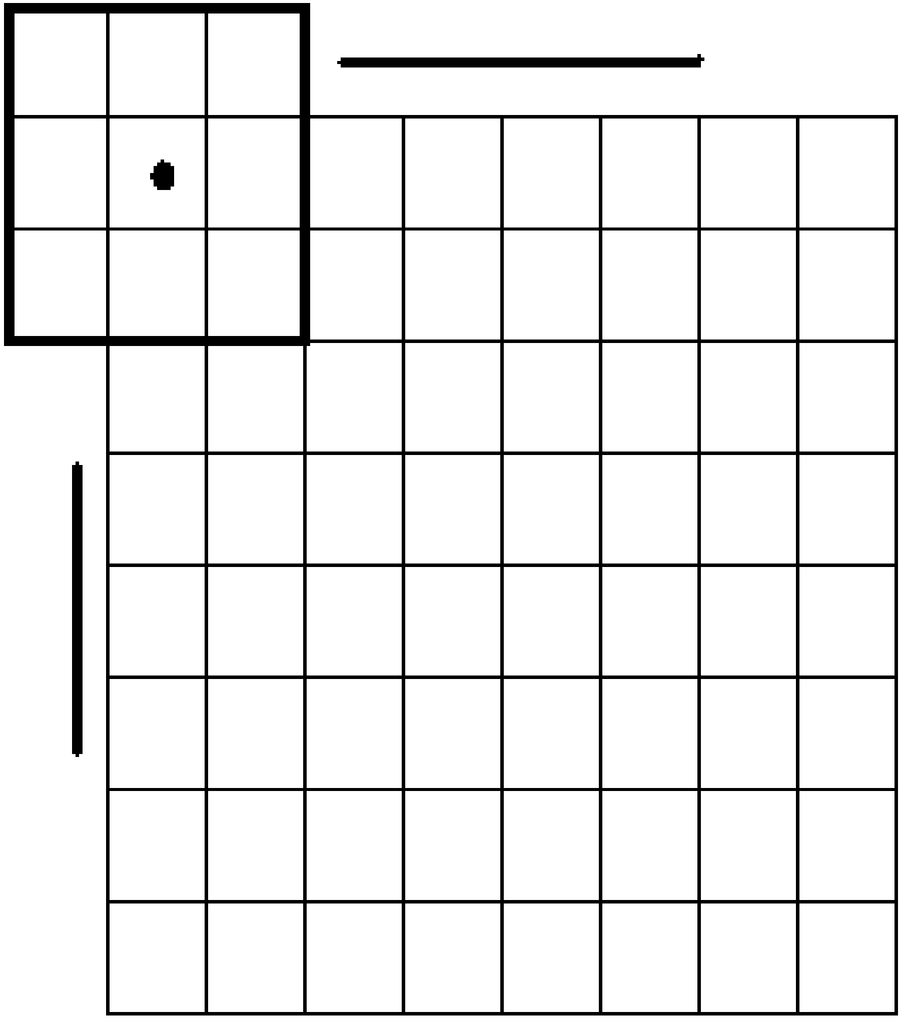


FIGURE 3.1
A 3×3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

Spatial filtering:



Averaging??



Padding?

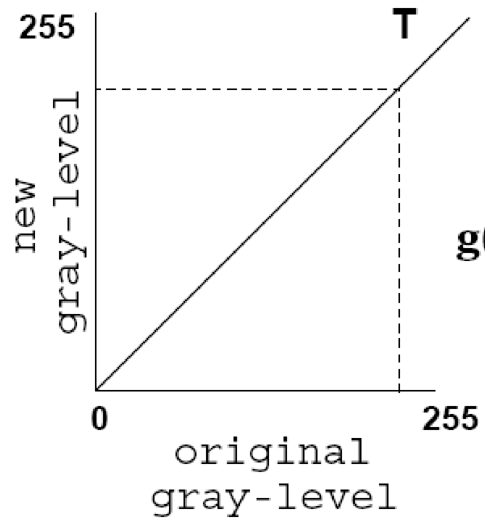
Image Enhancement:

- Image Enhancement is process of manipulating an image so that result is more suitable than the original to a *specific* application.

Basic intensity Transformation functions:

- Used for image enhancement
- Basic types:
  - Linear (identity and negative transformations)
  - Logarithmic(Log and inverse log transformations)
  - Power law(nth power and nth root transformations)

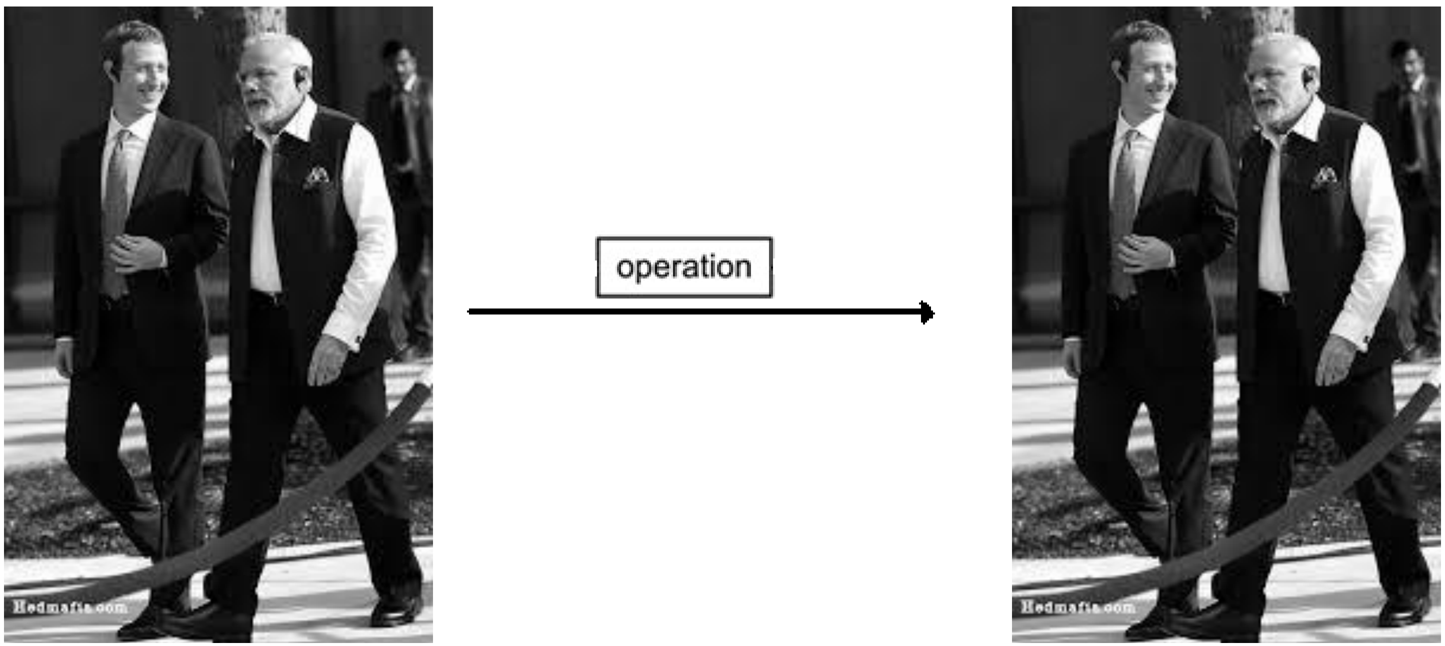
Identity transformation:



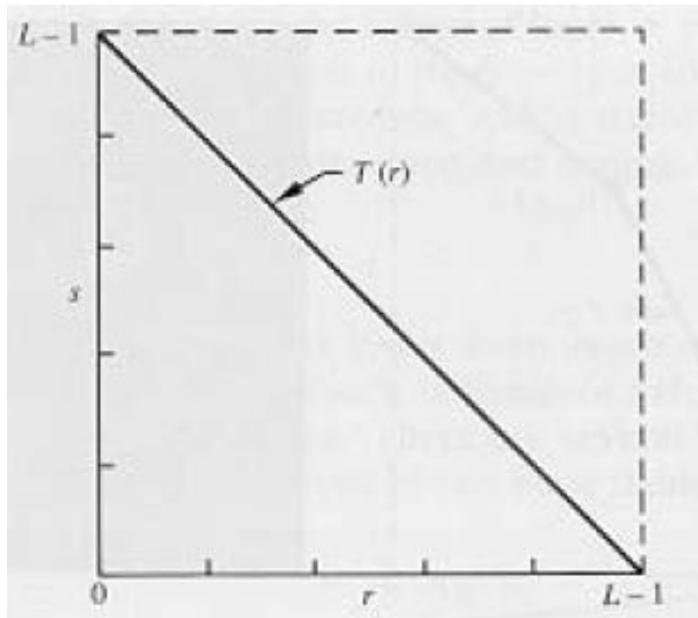
$$g(x,y) = f(x,y)$$

What is expected result??

Identity Transformation:



Negative transformation:



$$s = (L-1)-r$$

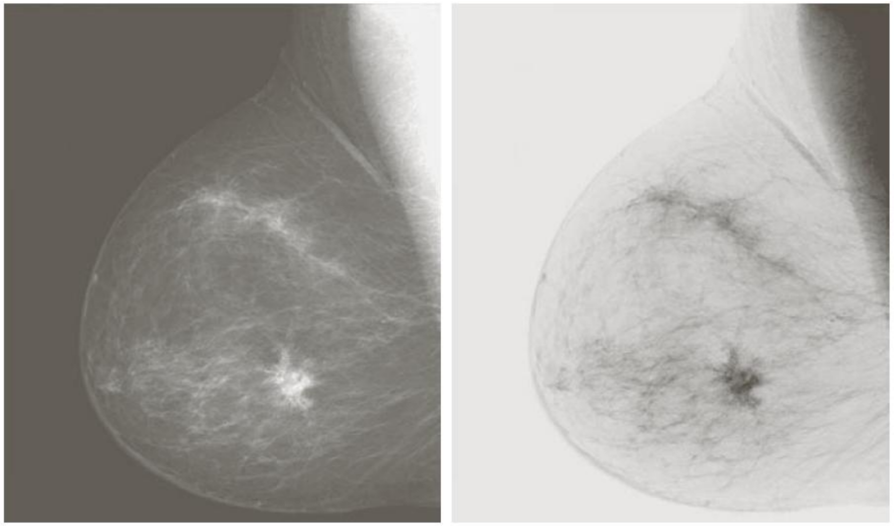
What is expected result??


Hedmafia.com


Hedmafia.com

Image negative:

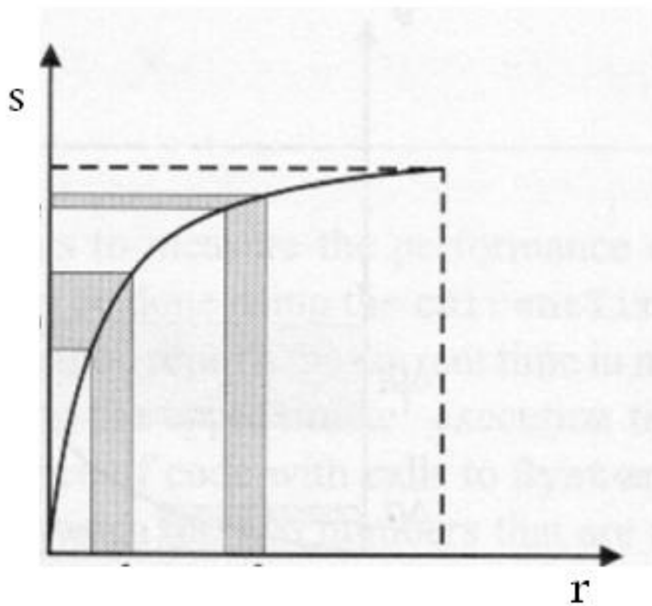
- Used for enhancing white or gray detail embedded in dark regions of image.



Log Transformations:

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

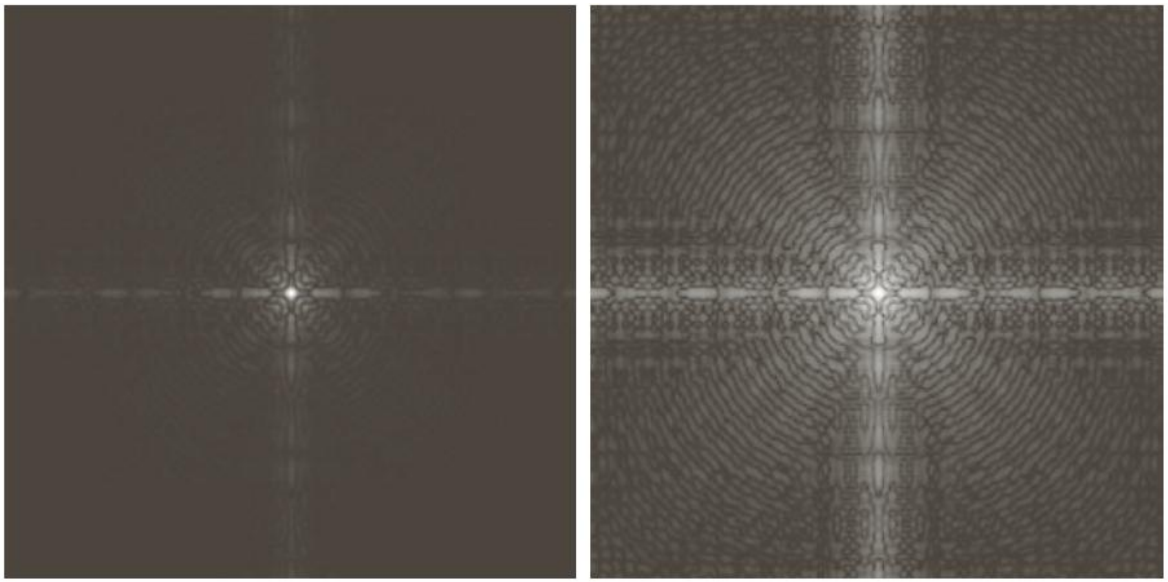
where c is a constant.



Narrow range of low intensity to wider range of output.

Wider range of high intensity to narrow range of output.

Dynamic range compression.



a b

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

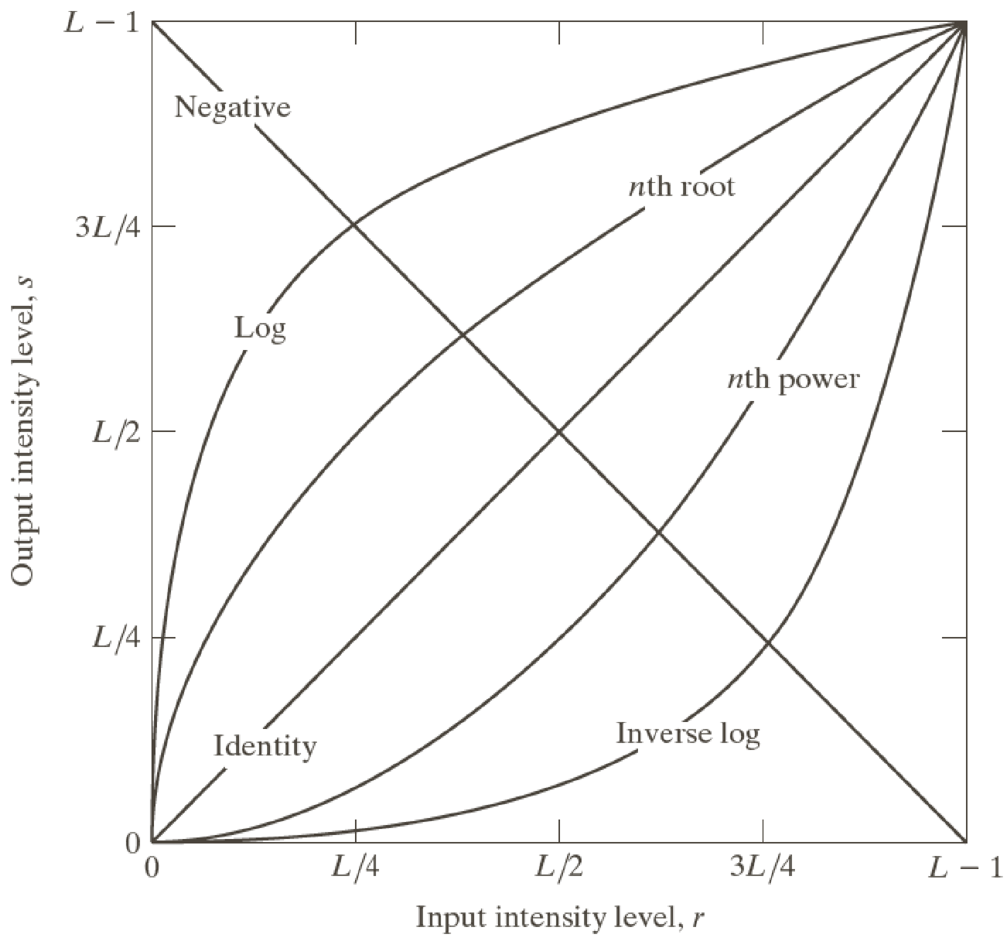


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Power-Law Transformations:

$$s = cr^{\gamma}$$

Where c and γ are positive constant.

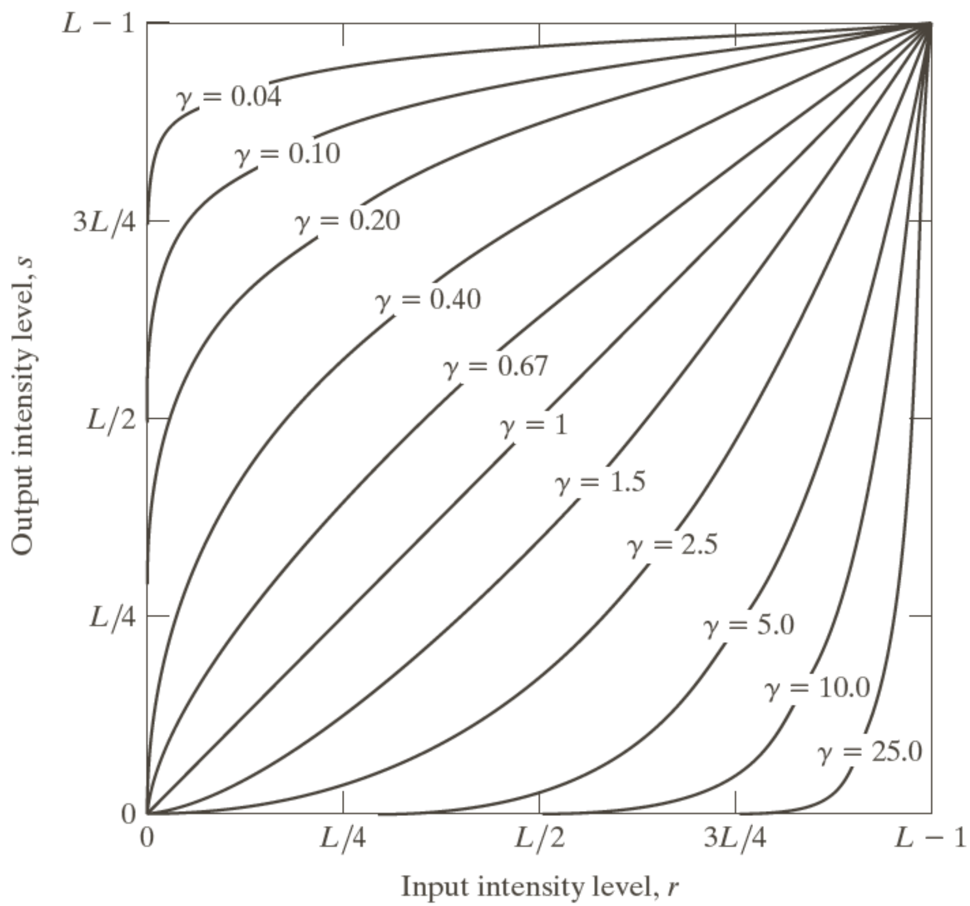
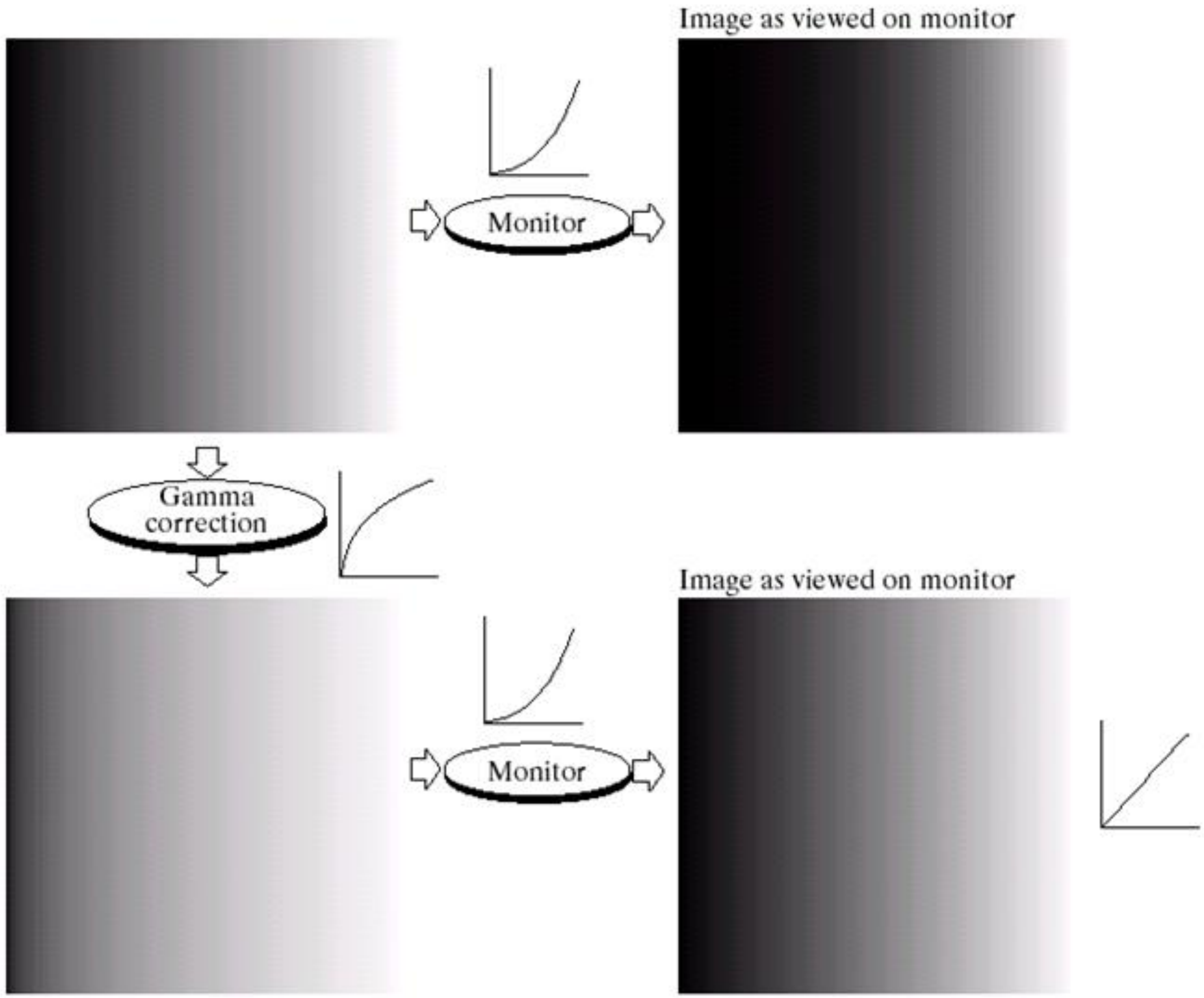


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

Gamma correction:

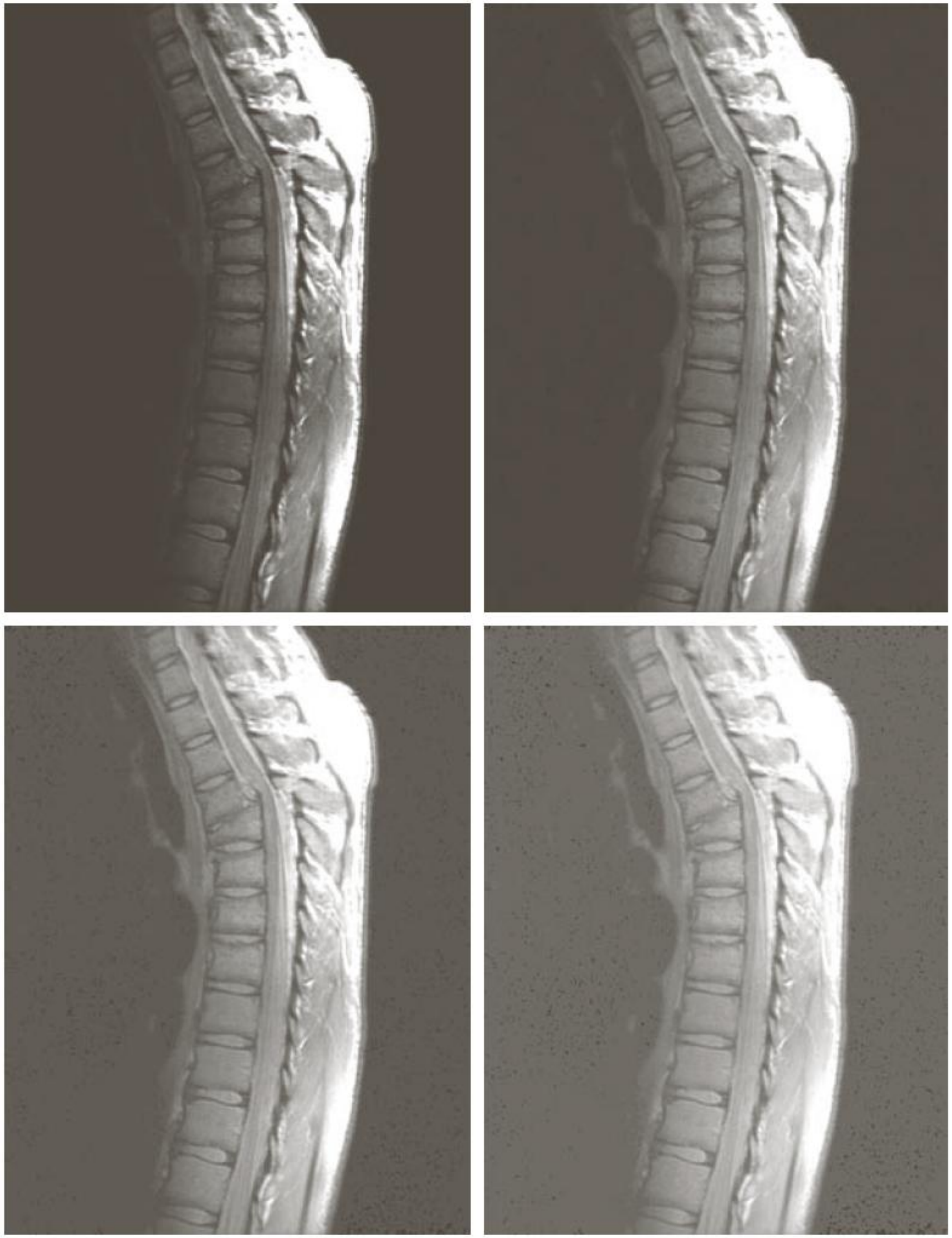
- A cathode ray tube (CRT), for example, converts a video signal to light in a nonlinear way. The light intensity *I* is proportional to a power (γ) of the source voltage *VS*
- For a computer CRT, γ is about 1.8 to 2.5
- Viewing images properly on monitors requires γ-correction

$$s = cr^{1/2.5} = cr^{0.4}$$



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.



a b
c d

FIGURE 3.8
(a) Magnetic resonance image (MRI) 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 courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

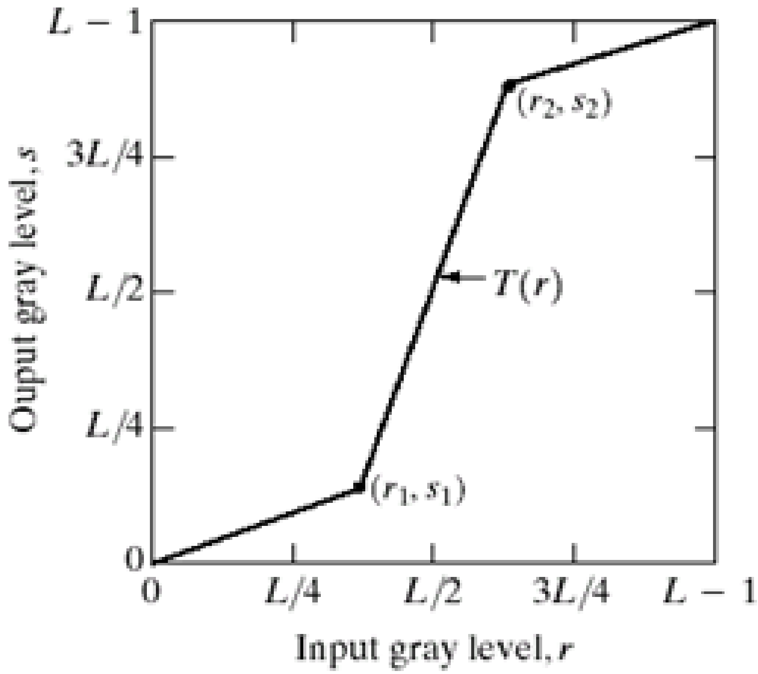
a	b
c	d

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 Transformation:

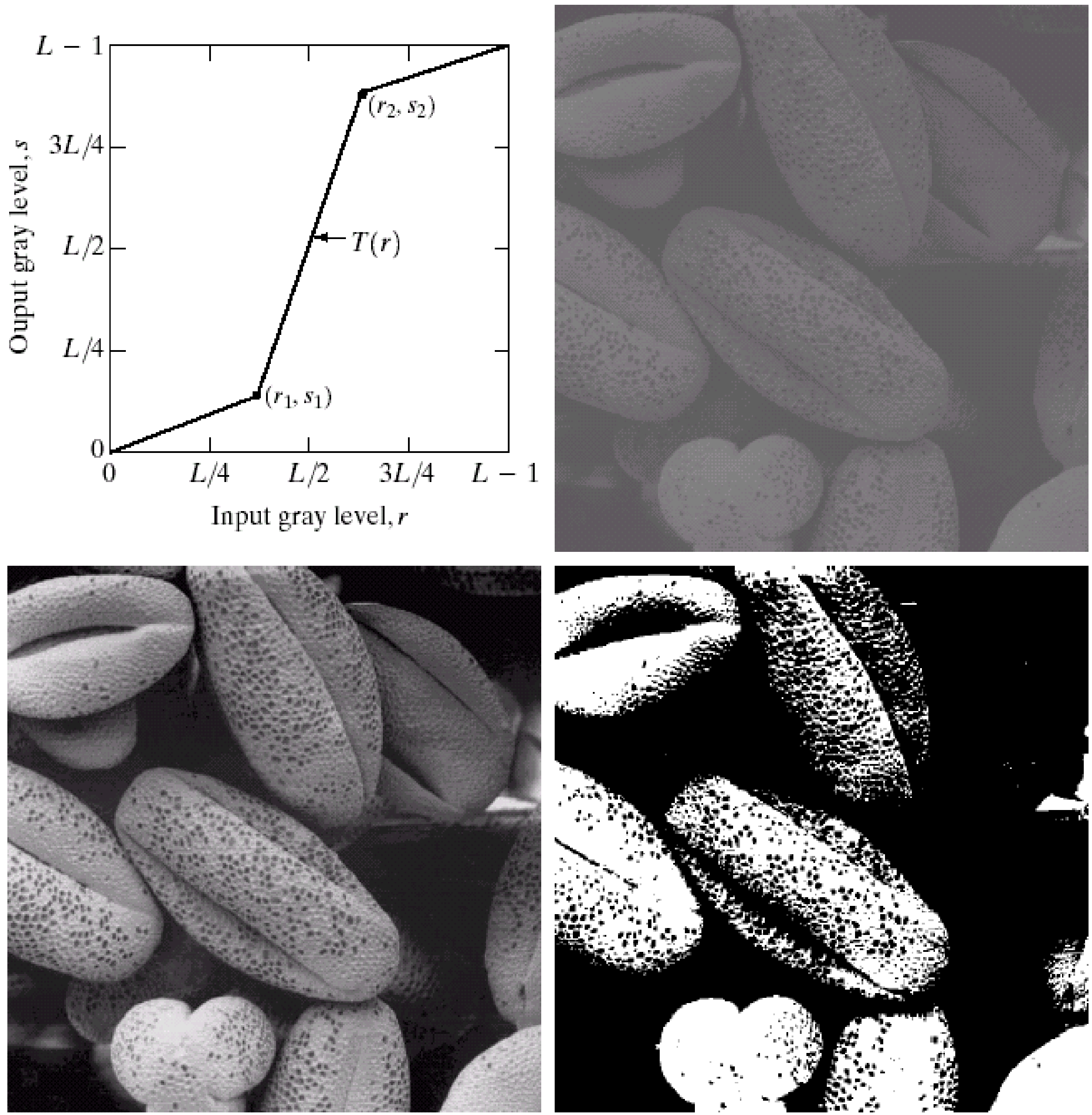
- Contrast stretching
- Intensity level slicing
- Bit plane slicing

Contrast stretching:



- (r1,s1) and (r2,s2) controls the shape of transformation.
- What if r1=r2,s1=0 and s2=L-1 ??
- Its thresholding.
- What if (r1,s1)=(rmin,0) and (r2,s2)=(rmax,L-1) ??
- Its contrast stretching.

Contrast stretching is a process that expands the range of intensity levels in an image so that it spans the available full intensity range.



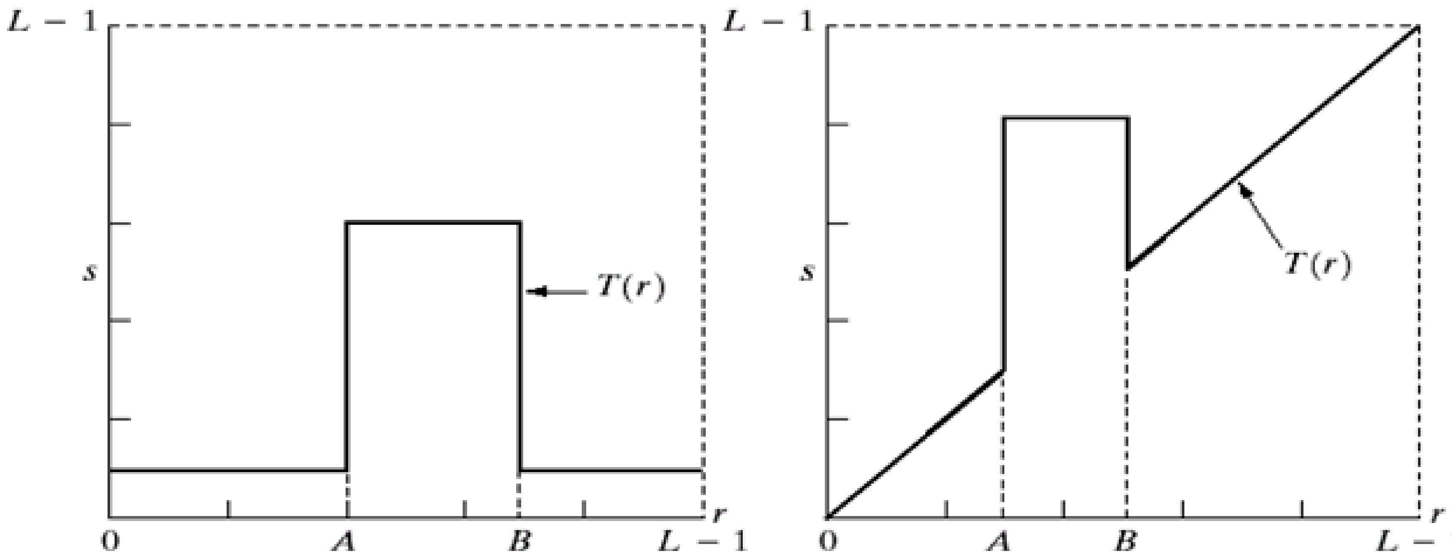
a b
c d

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.)

Intensity Level slicing:

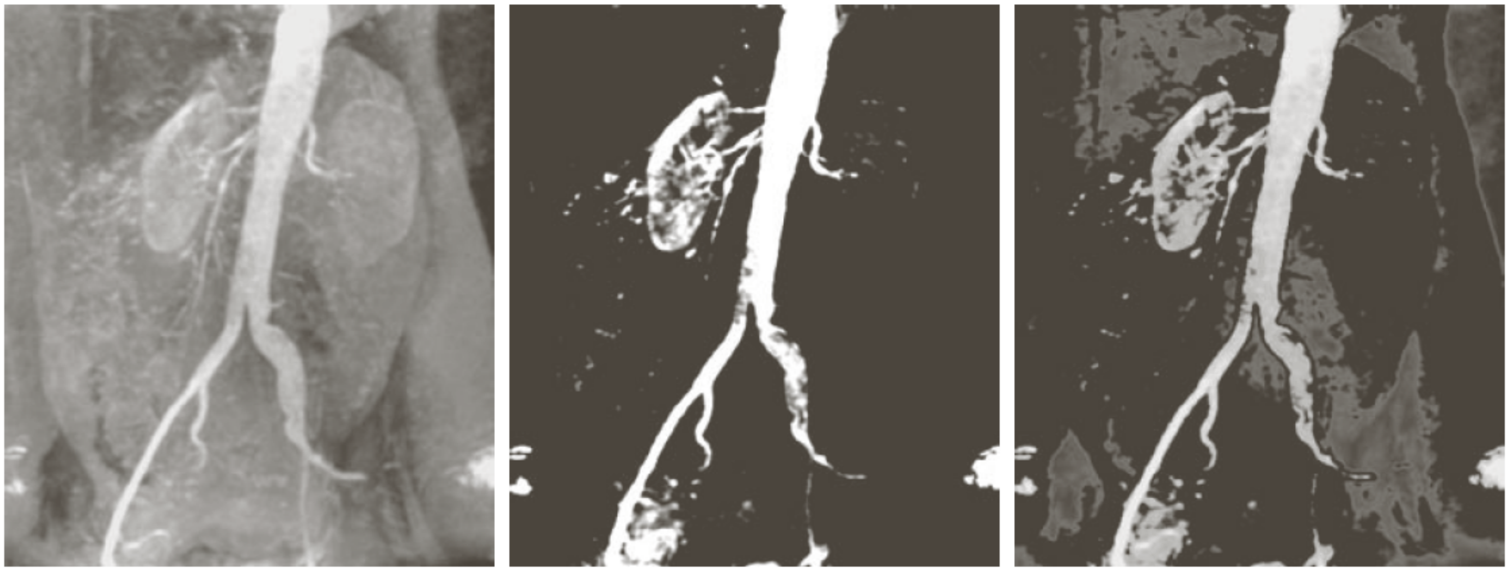
- Highlighting a specific range of intensities in an image.
- Two basic themes:

1. All the values of interest ➔ one intensity value(say white)
   All other values ➔ another intensity value(say black).
2. All the values of interest are brighten or darken
   keeping all other intensity levels unchanged.



a b

FIGURE 3.11 (a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level. (b) This transformation highlights range $[A, B]$ but preserves all other levels.



a b c

FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected area set to black, 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:

- 256 intensity levels ➔8 bits to represent a pixel
- It highlights the contribution made to total image appearance by specific bits.

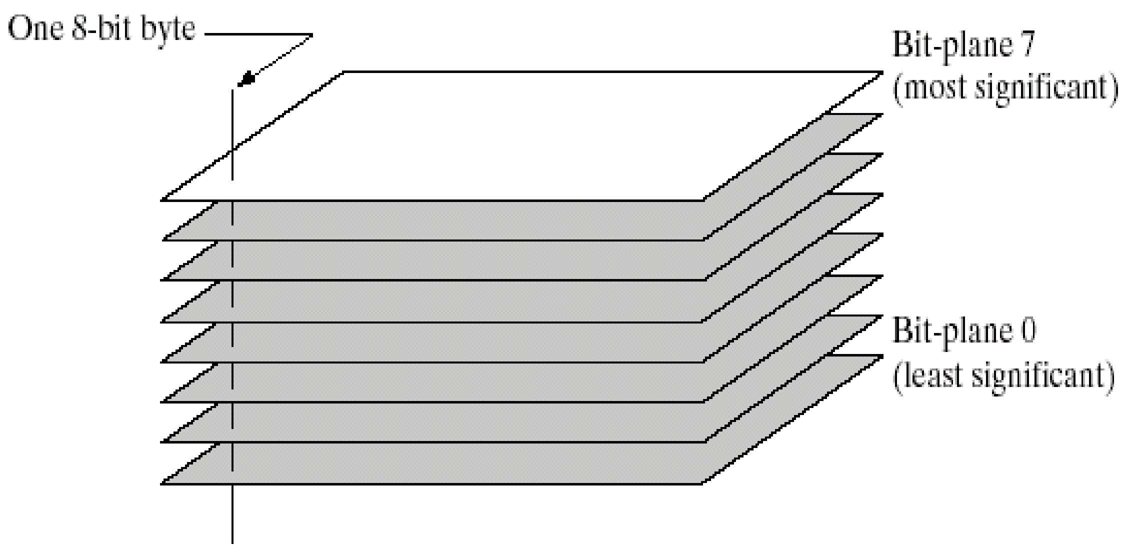
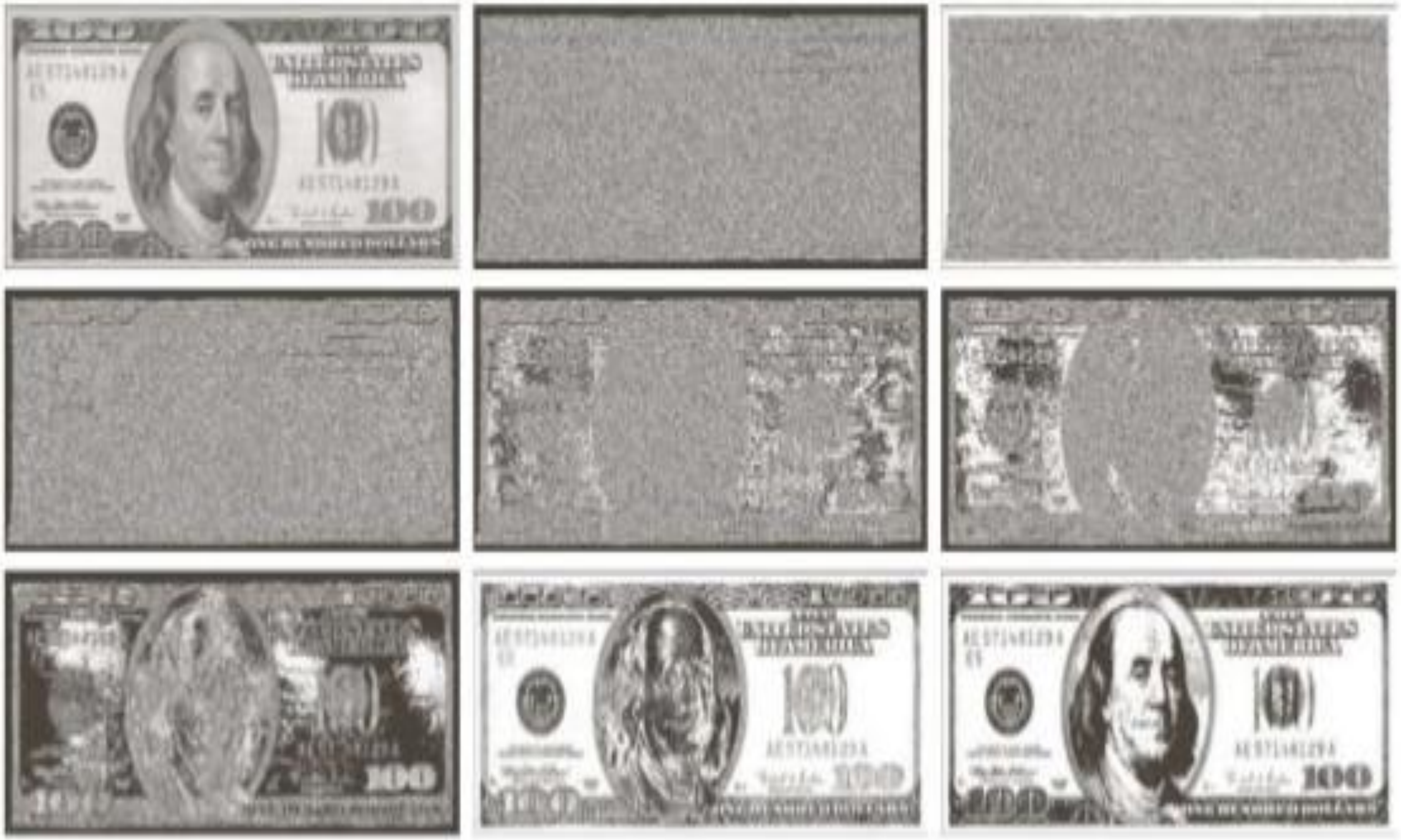


FIGURE 3.12 Bit-plane representation of an 8-bit image.



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.

Bit plan slicing:

- How to obtain 8-bit plane?
- Set all intensity values from 0-127 to 0 and 128 to 255 to 1.
- How to obtain others?
- 7 bit plane: 64-127 and 192-255 ➔1 o/w 0
- Border value $=(194)_d=(11000010)_b$

Why it is useful?

bit planes
8 and 7

8, 7 and 6

8, 7, 6
and 5

Reconstruction:

- Nth plane bit multiplied by 2^(n-1) constant.
- Addition of all such planes.
- Bit plane slicing is useful in compression and storage

Image Histogram:

- An image histogram is a plot of the gray-level frequencies (i.e., the number of pixels in the image that have that gray level).

0	0	1	0	2	0
1	0	7	7	7	0
0	7	0	0	7	0
1	0	0	7	2	0
0	0	7	1	0	1
1	0	7	7	7	0

frequencies

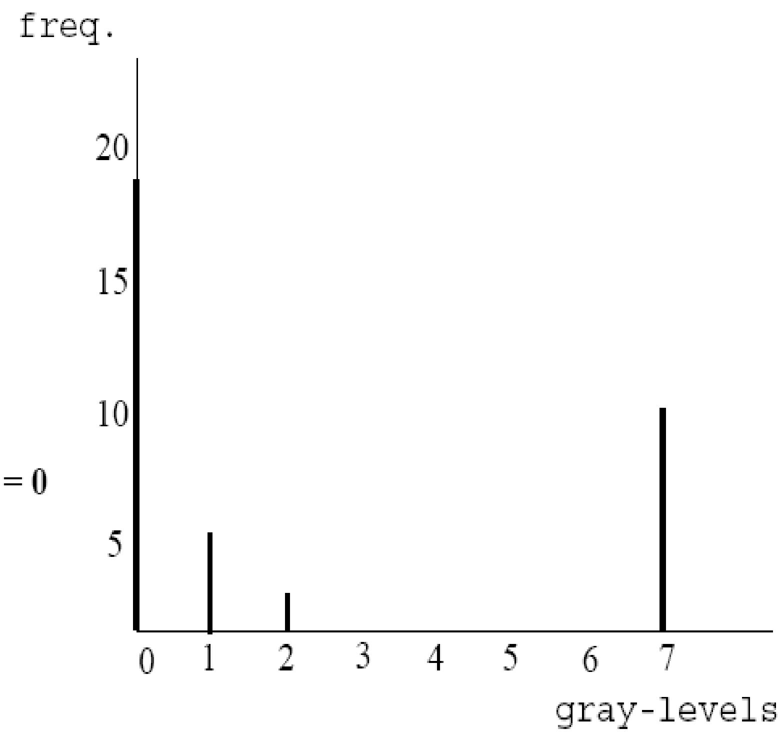
f(0)= 18

f(1)= 6

f(2)= 2

f(3)=f(4)=f(5)=f(6) = 0

f(7) = 10



- Divide frequencies by total number of pixels (m x n image size)to represent as probabilities.

$$p_k = n_k / N = n_k / mn$$

$$P(0) = \frac{f(0)}{36} = \frac{1}{2} \qquad P(1) = \frac{f(1)}{36} = \frac{1}{6}$$

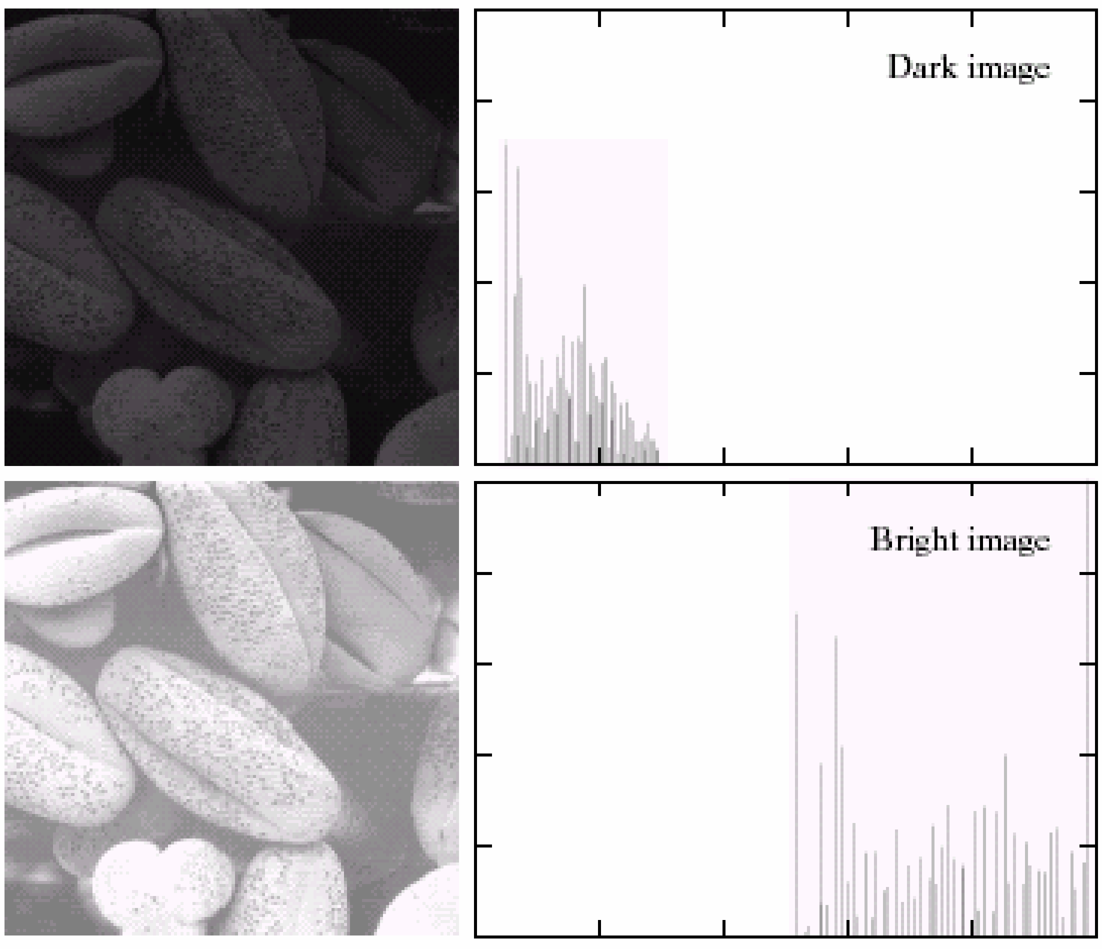
$$P(2) = \frac{f(2)}{36} = \frac{1}{18} \qquad P(3) = P(4) = P(5) = P(6) = 0$$

$$P(7) = \frac{f(7)}{36} = \frac{5}{18}$$

Sum of all these probabilities?

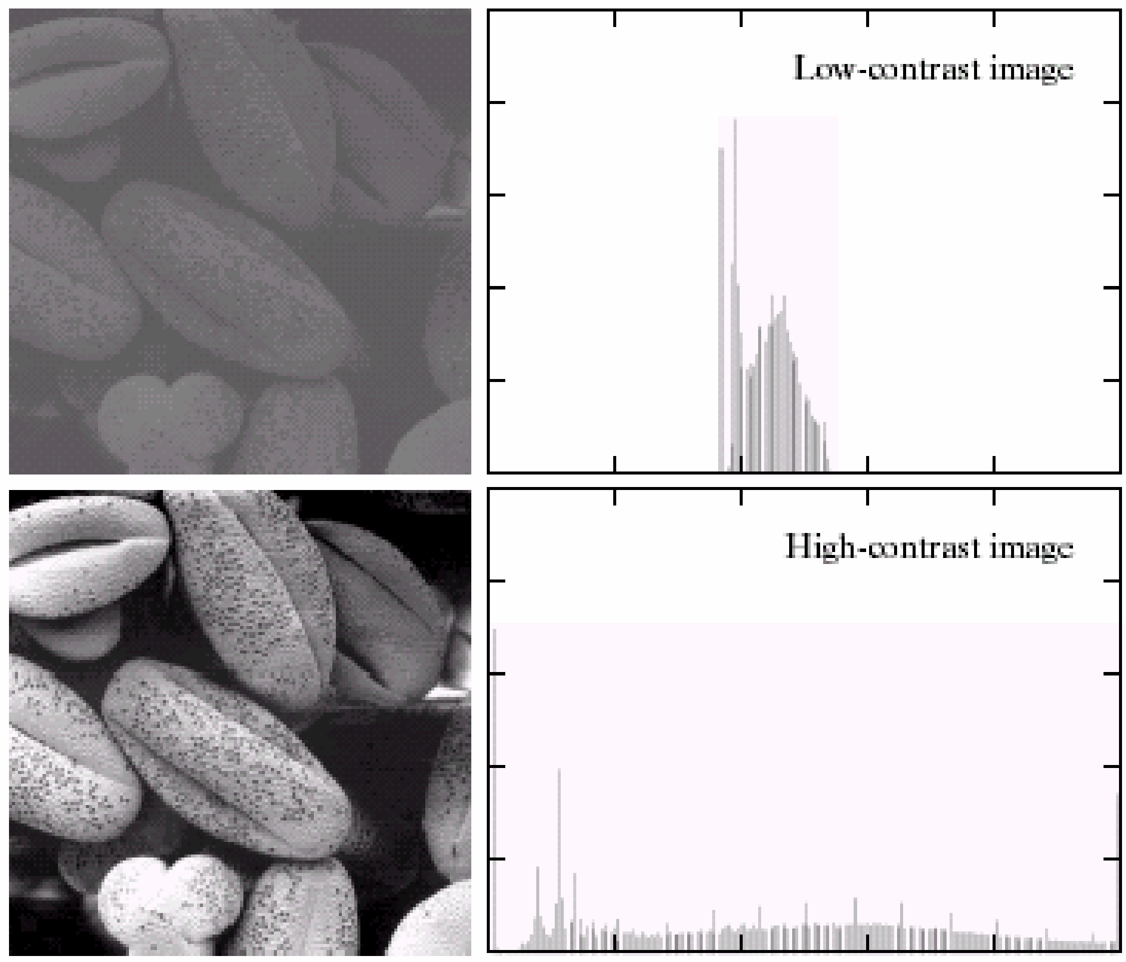
Plot of these probabilities is probability density function of input image pr(r)

Histogram:



x-axis – values of intensities
y-axis – their occurrence

Histogram:



x-axis – values of intensities
y-axis – their occurrence

Histogram:

- Histogram manipulation can be used for image enhancement.
- Information inherent in histogram also is quite useful in other image processing applications, such as image compression and segmentation.

Histogram equalization:

- To improve the contrast of an image
- To transform an image in such a way that the transformed image has a nearly *uniform distribution* of pixel values

Histogram equalization:

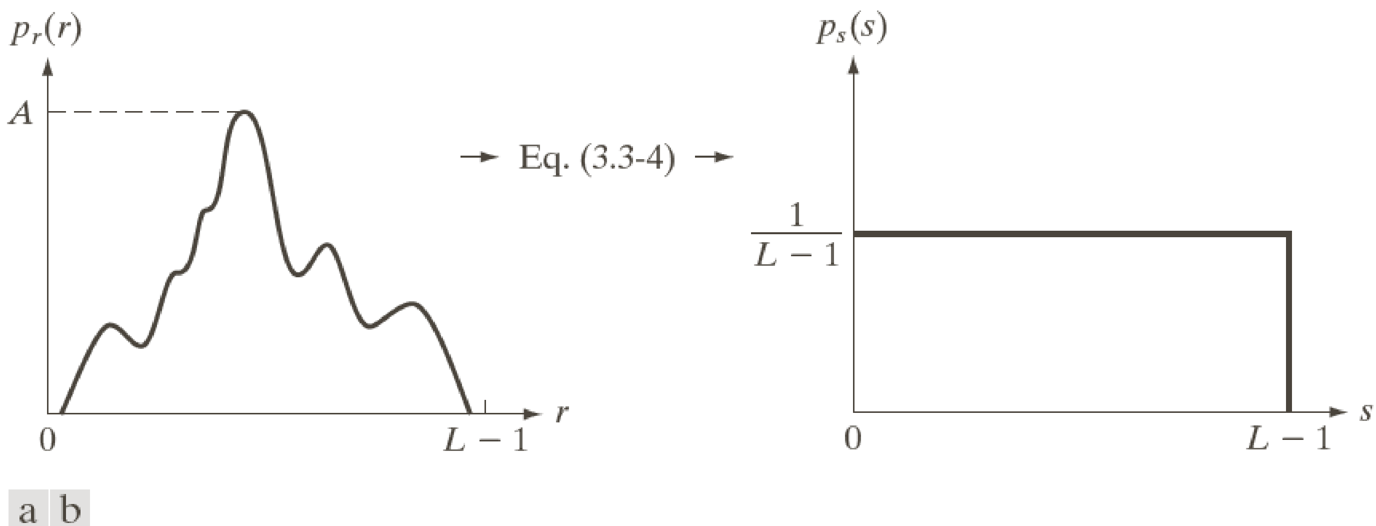


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

Histogram Equalization:

- Intensity mapping form

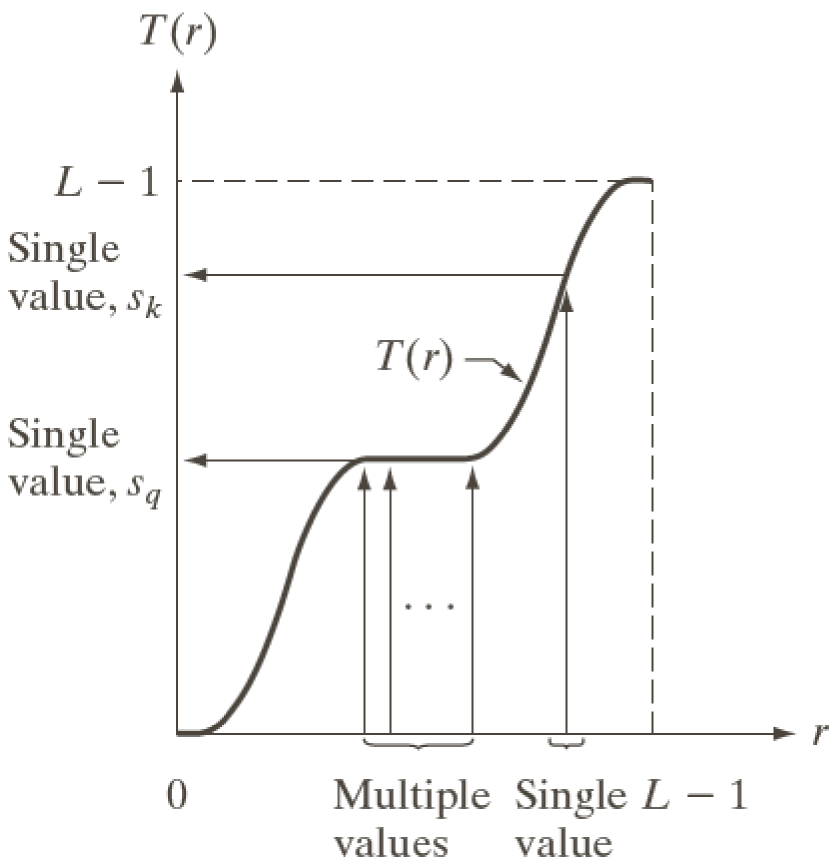
$$s = T(r),\ 0 \le r \le L-1$$

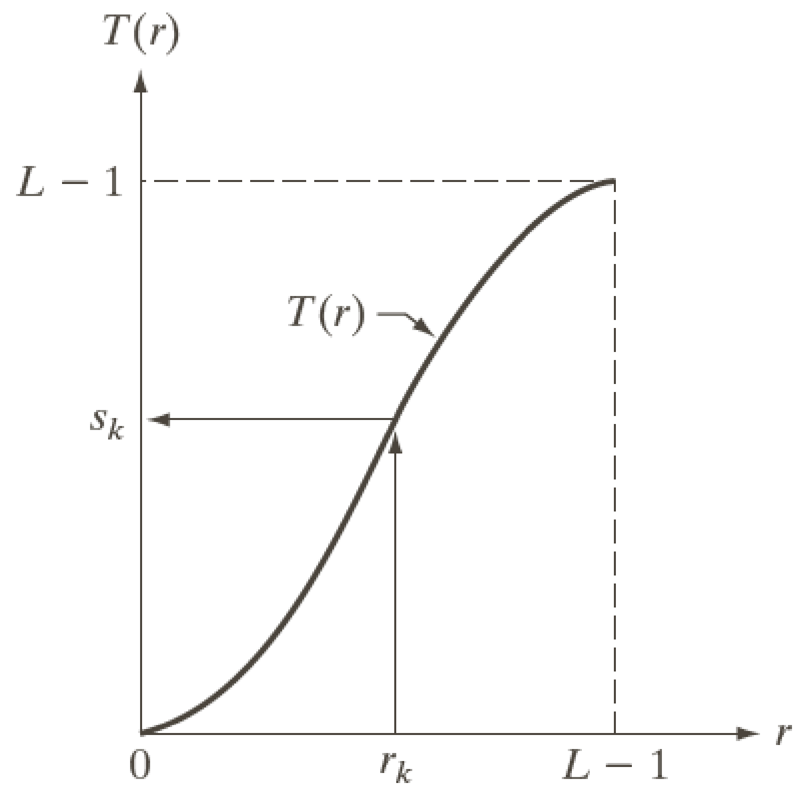
Conditions:

a) T(r) is a monotonically increasing function in the interval [0, L-1] and

b) $0 \le T(r) \le L-1 \qquad for\ 0 \le r \le L-1$

Histogram equalization:





Histogram Equalization:

In some formulations, we use the inverse in which case

$$r = T^{-1}(s),\ 0 \leq s \leq 1$$

(a) change to

a') T(r) is a strictly monotonically increasing function in the interval [0, L-1]

- Let $p_r(r)$ and $p_s(s)$ denote the PDFs of r and s respectively

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

- Consider following transform function:

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

where *w* is a dummy variable of integration and the right side of this equation is the *cumulative distribution function* of random variable *r*.

Does it satisfies condition a and b?

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)dw\right] = (L-1)p_r(r)$$

By Leibniz's rule that the derivative of a definite integral with respect to its upper limit is simply the integrand evaluated at that limit.

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right| = \frac{1}{L-1} \qquad 0 \le s \le L-1$$

Example

Suppose that the (continuous) intensity values in an image have the PDF

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2}, & \text{for } 0 \le \text{r} \le \text{L-1} \\ 0, & \text{otherwise} \end{cases}$$

Find the transformation function for equalizing the image histogram.

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = (L-1)\int_0^r \frac{2w}{(L-1)^2}dw \quad = \frac{r^2}{L-1}$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = \frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right|$$

$$= \frac{2r}{(L-1)^2}\left|\left[\frac{d}{dr}\frac{r^2}{L-1}\right]^{-1}\right|$$

$$= \frac{2r}{(L-1)^2}\left|\frac{(L-1)}{2r}\right| = \frac{1}{L-1}$$

- In discrete version:
  - The probability of occurrence of gray level r_k in an image is

$$p_r(r_k) = \frac{n_k}{n} \quad k = 0,1,2,\ldots,L-1$$

n : the total number of pixels in the image

n_k : the number of pixels that have gray level r_k

L : the total number of possible gray levels in the image

  - The transformation function is

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = (L-1)\sum_{j=0}^{k} \frac{n_j}{n} \quad k = 0,1,2,\ldots,L-1$$

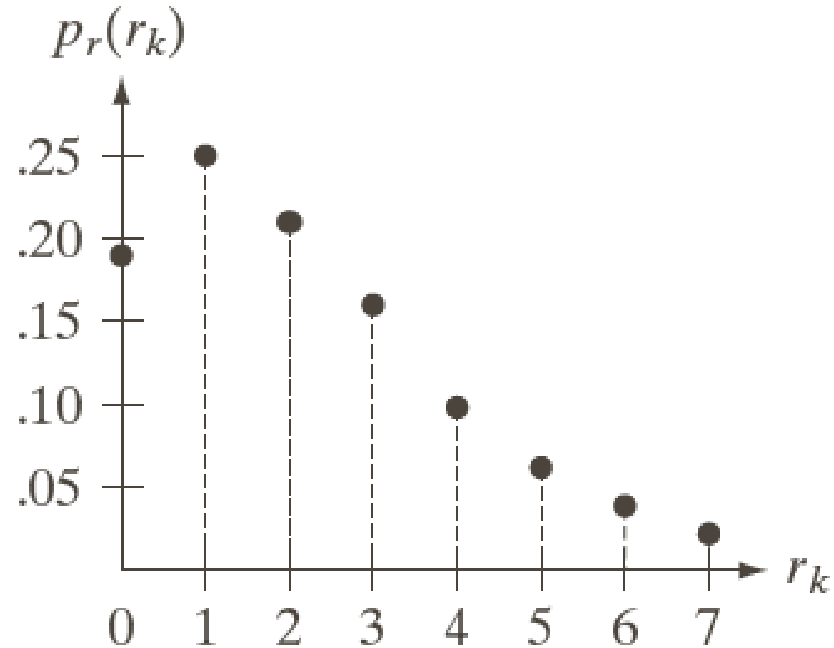
  - Thus, an output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k.

Example: Histogram Equalization

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

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



$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = (L-1)\sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0,1,2,\ldots,L-1$$

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

$s_2 = 4.55$

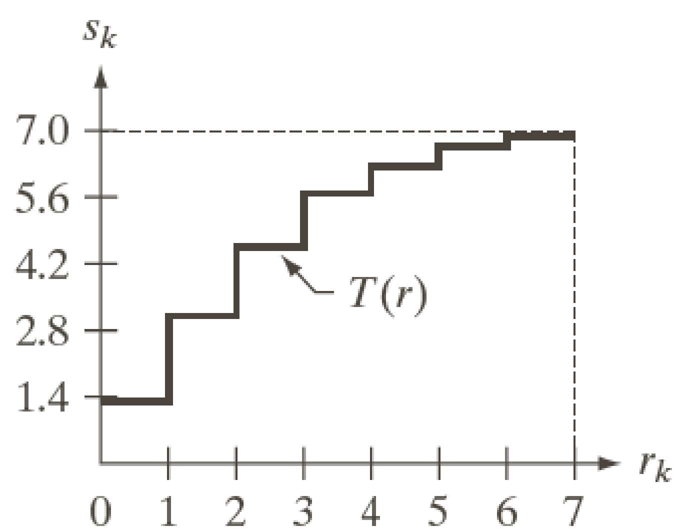
$s_3 = 5.67$

$s_4 = 6.23$

$s_5 = 6.65$

$s_6 = 6.86$

$s_7 = 7.00$



$s_0 = 1.33 \rightarrow 1$ $s_4 = 6.23 \rightarrow 6$

$s_1 = 3.08 \rightarrow 3$ $s_5 = 6.65 \rightarrow 7$

$s_2 = 4.55 \rightarrow 5$ $s_6 = 6.86 \rightarrow 7$

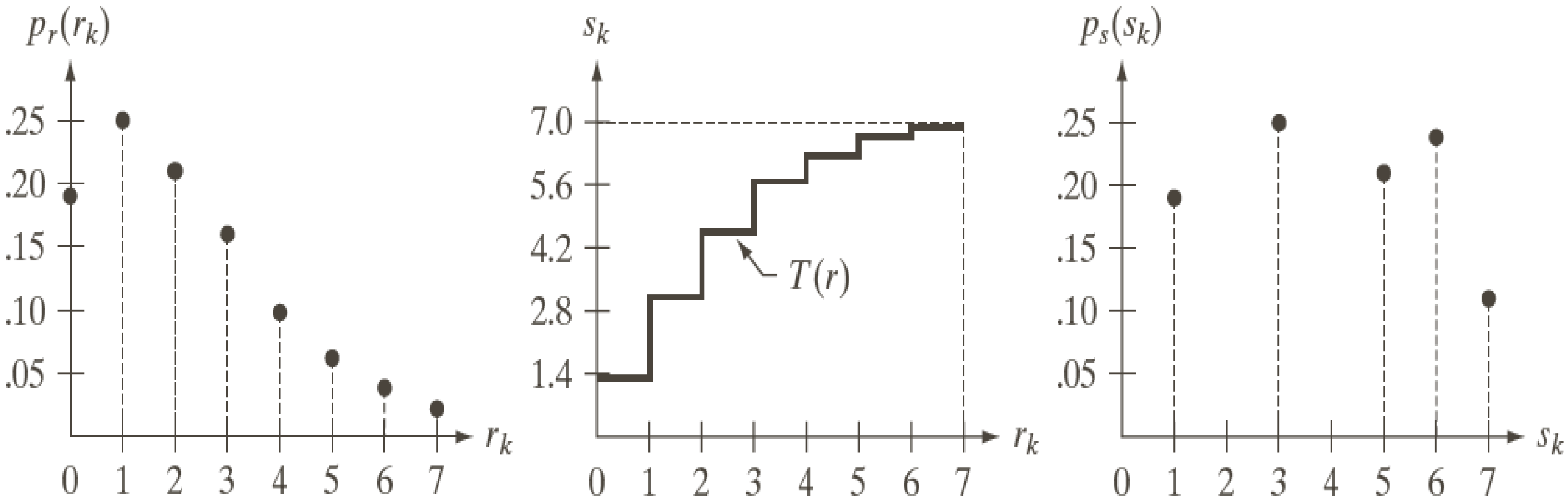
$s_3 = 5.67 \rightarrow 6$ $s_7 = 7.00 \rightarrow 7$

$r_0=0 \Rightarrow s_0=1$
$r_1=1 \Rightarrow s_1=3$
$r_2=2 \Rightarrow s_2=5$
$r_3=3 \Rightarrow s_3=6$
$r_4=4 \Rightarrow s_4=6$
$r_5=5 \Rightarrow s_5=7$
$r_6=6 \Rightarrow s_6=7$
$r_7=7 \Rightarrow s_7=7$

$r_0=0$	790	$s_0=1$
$r_1=1$	1023	$s_1=3$
$r_2=2$	850	$s_2=5$
$r_3=3$	656	$s_3= s_4=6$
$r_4=4$	329	
$r_5=5$	245	$s_5=s_6= s_7=7$
$r_6=6$	122	
$r_7=7$	81	

ps(sk)?

$r = T^{-1}(s)$?



a b c

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

Histogram Equalization

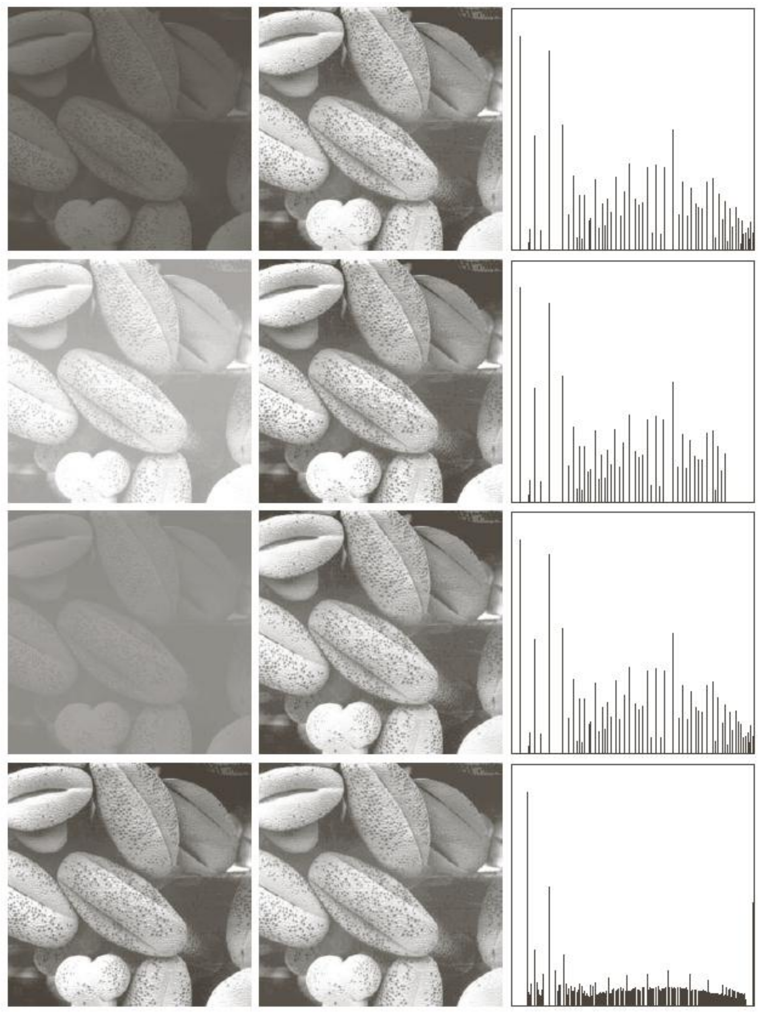
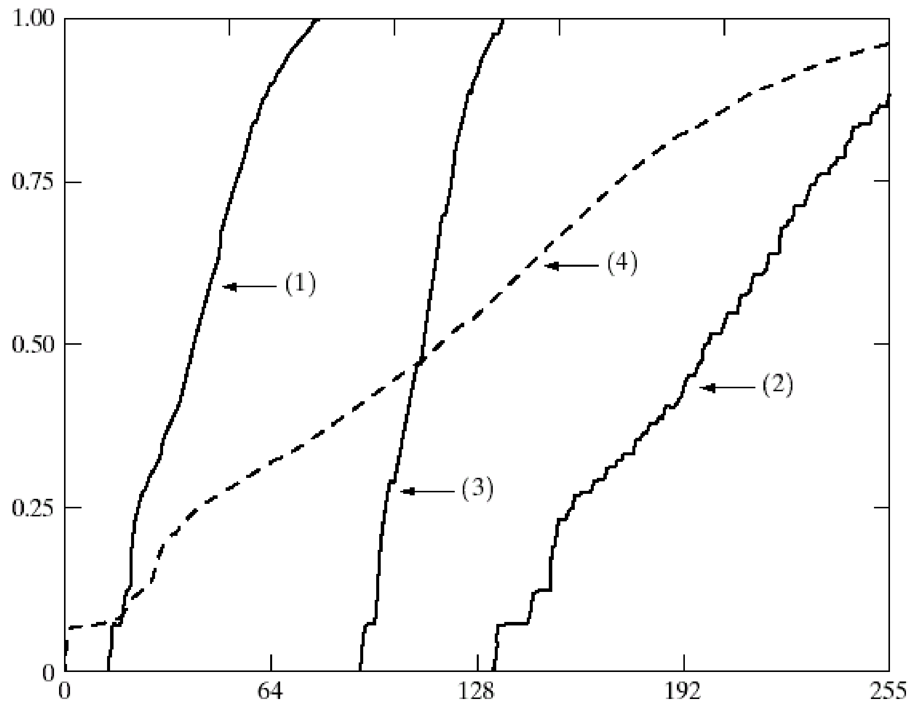


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

Cumulative Histograms

FIGURE 3.18 Transformation functions (1) through (4) were obtained from the histograms of the images in Fig.3.17(a), using Eq. (3.3-8).



HISTOGRAM MATCHING

- Histogram equalization yields an image whose pixels are (in theory) uniformly distributed among all gray levels.
- Sometimes, this may not be desirable. Instead, we may want a transformation that yields an output image with a pre-specified histogram. This technique is called **histogram specification.**

Let $p_r(r)$ denote continuous probability density function of gray-level of input image, r

Let $p_z(z)$ denote desired (specified) continuous probability density function of gray-level of output image, z

- Equalize the initial histogram of the image:

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

$$G(z) = T(r)$$

- Equalize the target histogram:

$$s = G(z) = (L-1)\int_0^r p_z(w)\,dw$$

- Obtain the inverse transform: $z = G^{-1}(s) = G^{-1}(T(r))$

Histogram Matching (example)

Assuming continuous intensity values, suppose that an image has the intensity PDF

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2}, & \text{for } 0 \le r \le L\text{-}1 \\ 0\ , & \text{otherwise} \end{cases}$$

Find the transformation function that will produce an image whose intensity PDF is

$$p_z(z) = \begin{cases} \dfrac{3z^2}{(L-1)^3}, & \text{for } 0 \le z \le (L\text{-}1) \\ 0, & \text{otherwise} \end{cases}$$

Histogram Matching: Example

Find the histogram equalization transformation for the input image

$$s = T(r) = (L-1)\int_0^r p_r(w)dw = (L-1)\int_0^r \frac{2w}{(L-1)^2}dw = \frac{r^2}{L-1}$$

Find the histogram equalization transformation for the specified histogram

$$G(z) = (L-1)\int_0^z p_z(t)dt = (L-1)\int_0^z \frac{3t^2}{(L-1)^3}dt = \frac{z^3}{(L-1)^2} = s$$

The transformation function

$$z = \left[(L-1)^2 s\right]^{1/3} = \left[(L-1)^2 \frac{r^2}{L-1}\right]^{1/3} = \left[(L-1)r^2\right]^{1/3}$$

Histogram Matching (Specification)

- Histogram Specification Procedure for discrete:

Compute the histogram $p_r(r)$ of the given image and use it to find the histogram equalization transformation in equation

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} \frac{n_j}{MN}, \quad k = 0,1,2,\ldots,L-1$$

and round the resulting values to the integer range [0, L-1]

Histogram Matching (Specification)

Compute all values of the transformation function G using same equation

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(r_i), \quad q = 0,1,2,\ldots,L-1$$

and round values of G

Histogram Matching (Specification)

For every value of s_k, k = 0,1,…,L-1, use the stored values of G to find the corresponding value of z_q so that $G(z_q)$ is closest to s_k and store these mappings from s to z.

When more than one value of zq satisfies the given sk(i.e mapping is not unique), choose the smallest value by convention.

Example: Histogram Matching

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

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

z_q	**Specified** $p_z(z_q)$	**Actual** $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

Example: Histogram Matching

- Obtain the scaled histogram-equalized values,

$$s_0 = 1, s_1 = 3, s_2 = 5, s_3 = 6, s_4 = 6,$$
$$s_5 = 7, s_6 = 7, s_7 = 7.$$

- Compute all the values of the transformation function G,

$$G(z_0) = 7\sum_{j=0}^{0} p_z(z_j) = 0.00 \rightarrow 0$$

$$G(z_1) = 0.00 \rightarrow 0 \qquad G(z_2) = 0.00 \rightarrow 0$$
$$G(z_3) = 1.05 \rightarrow 1 \qquad G(z_4) = 2.45 \rightarrow 2$$
$$G(z_5) = 4.55 \rightarrow 5 \qquad G(z_6) = 5.95 \rightarrow 6$$
$$G(z_7) = 7.00 \rightarrow 7$$

Example: Histogram Matching

$$s_0 = 1, s_1 = 3, s_2 = 5, s_3 = 6, s_4 = 6,$$
$$s_5 = 7, s_6 = 7, s_7 = 7.$$

r_k

0

1

2

3

4

5

6

7

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

Example: Histogram Matching

$r_k \rightarrow z_q$

$0 \rightarrow 3$

$1 \rightarrow 4$

$2 \rightarrow 5$

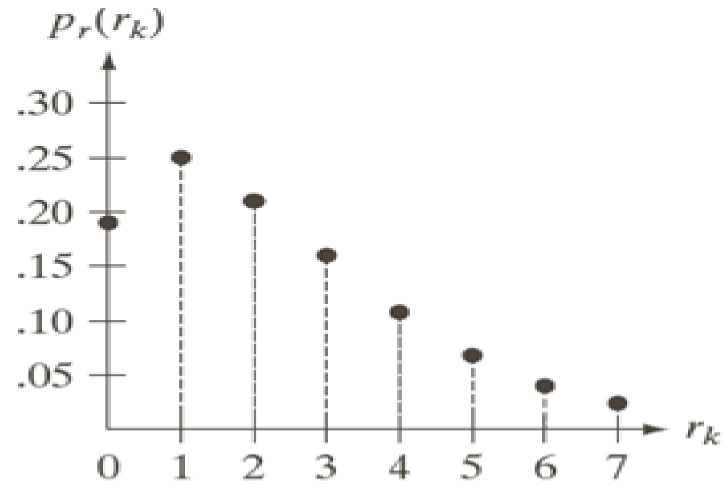
$3 \rightarrow 6$

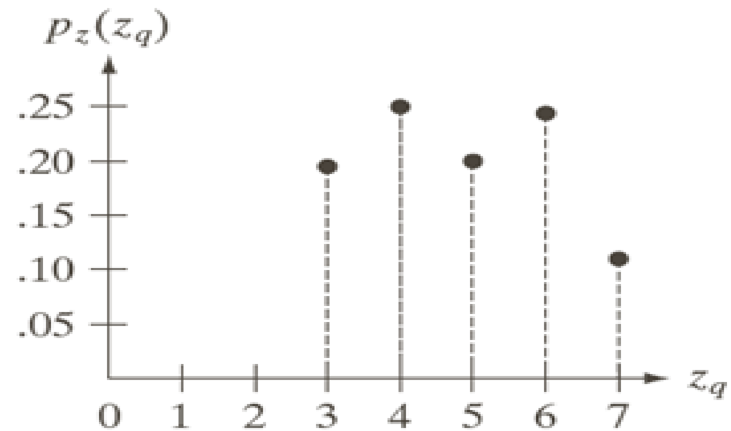
$4 \rightarrow 6$

$5 \rightarrow 7$

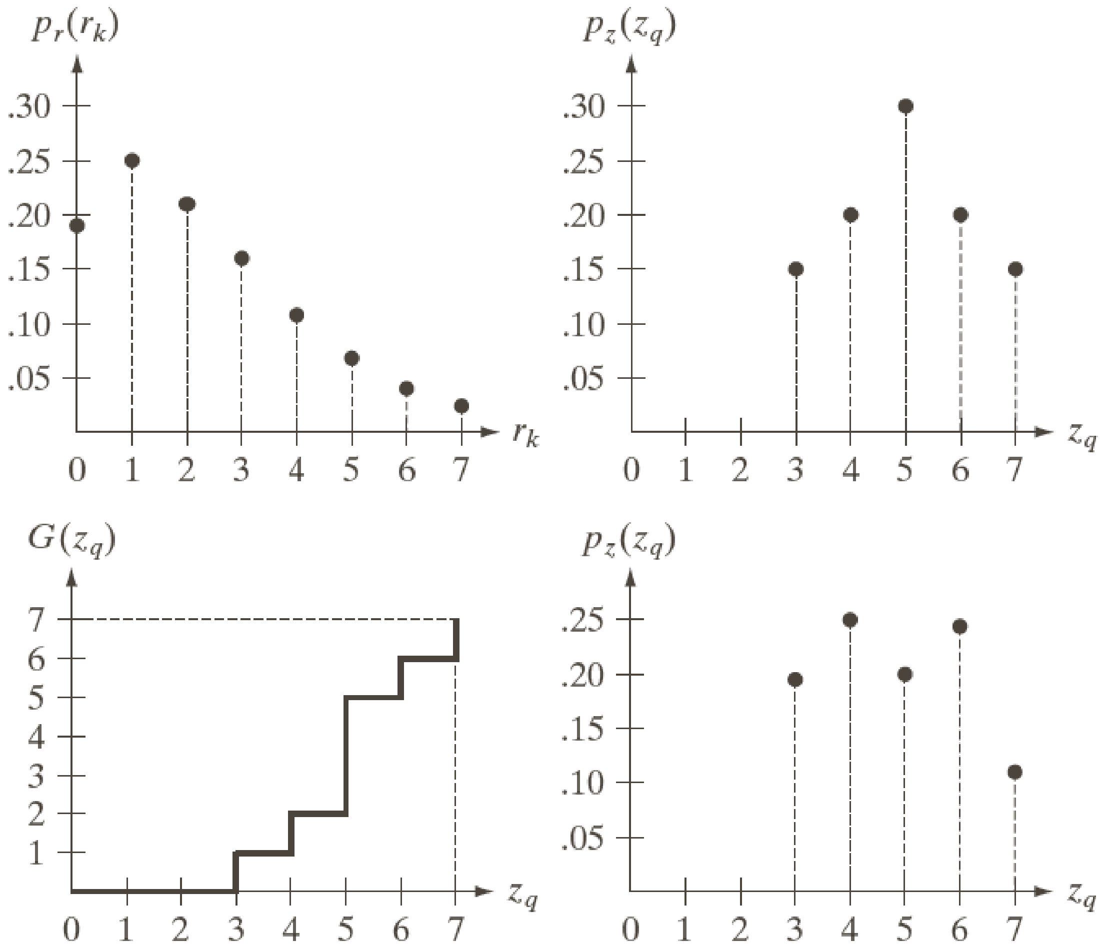
$6 \rightarrow 7$

$7 \rightarrow 7$





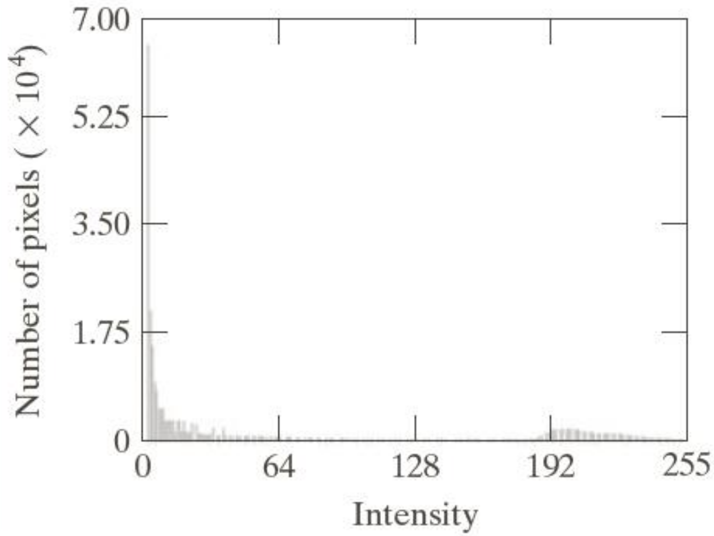
Example: Histogram Matching



a	b
c	d

FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

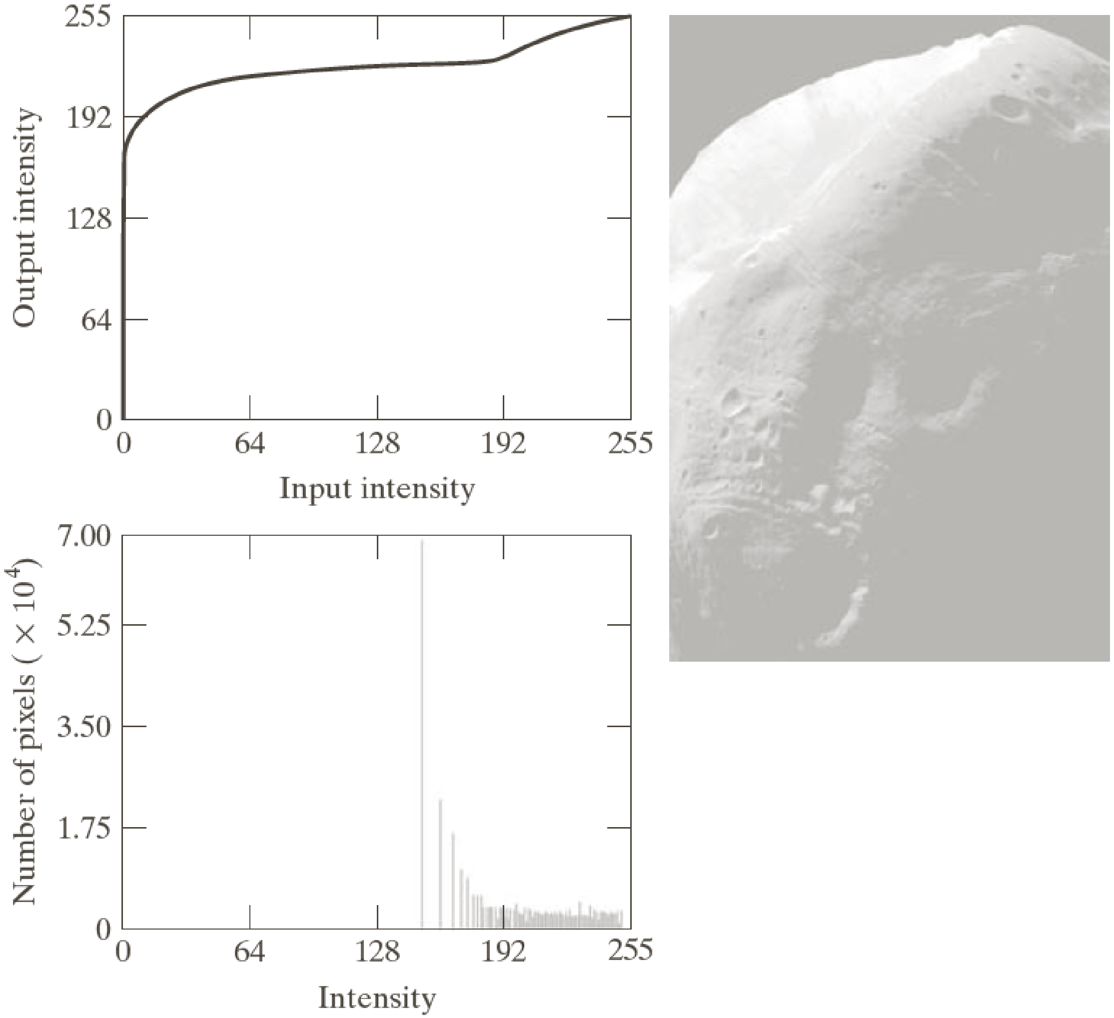
Example: Histogram Matching



a b

FIGURE 3.23
(a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor.* (b) Histogram. (Original image courtesy of NASA.)

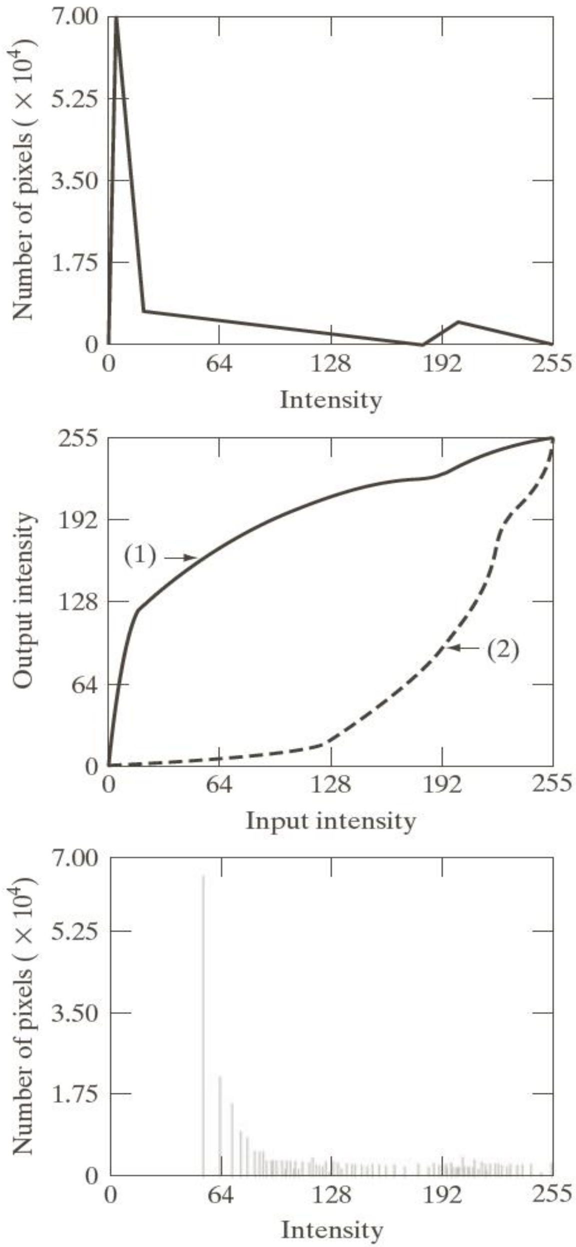
Example: Histogram Matching



a b
c

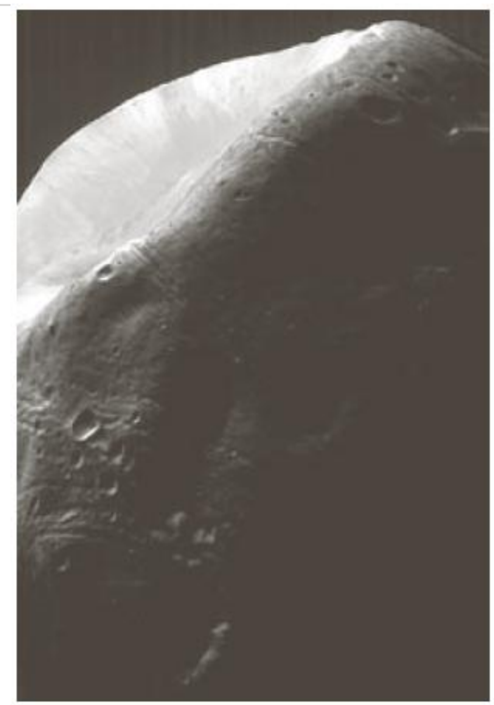
FIGURE 3.24
(a) Transformation function for histogram equalization.
(b) Histogram-equalized image (note the washed-out appearance).
(c) Histogram of (b).

Example: Histogram Matching



a c
b
d

FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).



Histogram Matching (Specification)

- Trial and error process
- No specific rules!!!

Local Histogram processing:

- Histogram Processing methods discussed in the previous two sections are *Global.*
- What if it is necessary to enhance detail over small areas in an image?
- *This procedure is to define a neighborhood and move its center pixel to pixel.*

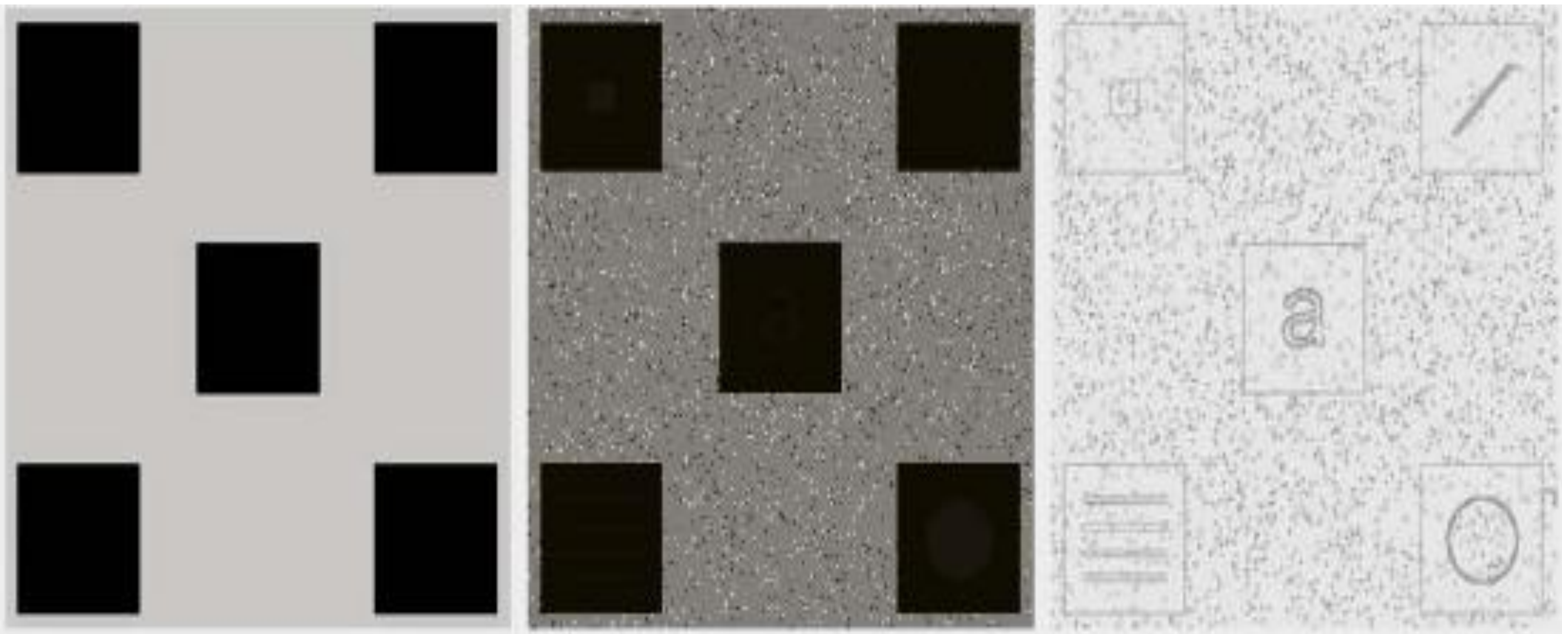
Local Histogram processing

- At each location, the histogram of the points in the neighborhood is computed and either a histogram equalization or histogram specification transformation function is obtained.

Local Histogram processing:

- Map the intensity of the pixel centered in the neighborhood.
- Center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated
- Sometimes to reduce computation is to utilize non overlapping regions, but this method usually produces an undesirable *“blocky”* effect.

Local Histogram processing:



a b c

FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3×3.

Statistics:

Given a set of numbers, for example,

S= {1, 1, 1, 1, 3, 3, 4, 5, 5 , 6 }

To calculate the average of these 10 numbers, you can use

$$\frac{1+1+1+1+3++3+4+5+5+6}{10}=\frac{30}{10}=3$$

Or you can get it this way

$$\frac{(1)(4)+(3)(2)+(4)(1)+(5)(2)+(6)(1)}{10}$$

$$=(1)(\frac{4}{10})+(3)(\frac{2}{10})+(4)(\frac{1}{10})+(5)(\frac{2}{10})+(6)(\frac{1}{10})$$

$$=3.$$

$$\overline{x}=\sum_{x}P\left(x\right)x$$

Statistics:

- Variance of numbers is defined as:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \quad \text{Where} \quad \mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Alternately :

$$V(X) = \sigma^2 = \sum_{all\ x}(x-\mu)^2 P(x)$$

Histogram statistics:

r: discrete random variable representing intensity values in the range [0, L-1]

$p(r_i)$: normalized histogram component corresponding to value r_i

m is the mean value of r

$$m = \sum_{i=0}^{L-1} r_i \, p(r_i)$$

- Alternately from sample values:

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

sample mean

Histogram statistics:

the second moment (also the variance of r) is

$$\mu_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 \, p(r_i)$$

- Alternately from sample values:

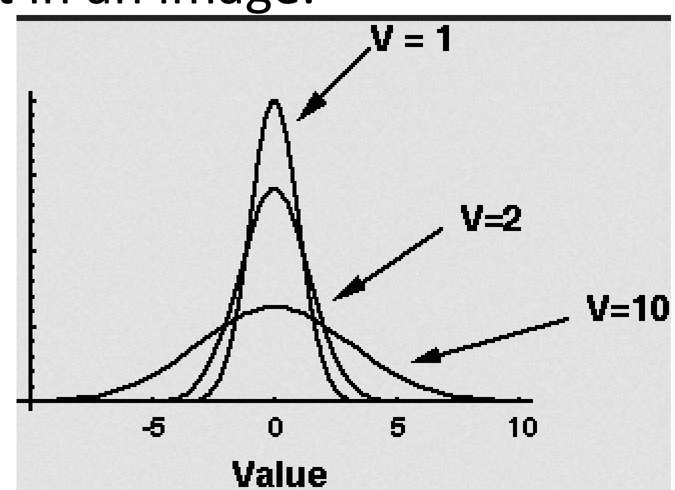
$$\sigma^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - m]^2$$ sample variance

Histogram statistics:

the nth moment of r about its mean is defined as

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n \, p(r_i)$$

Mean is a measure of average intensity and the variance is a measure of contrast in an image.



■ Before proceeding, it will be useful to work through a simple numerical example to fix ideas. Consider the following 2-bit image of size 5 × 5:

$$
\begin{array}{ccccc}
0 & 0 & 1 & 1 & 2 \\
1 & 2 & 3 & 0 & 1 \\
3 & 3 & 2 & 2 & 0 \\
2 & 3 & 1 & 0 & 0 \\
1 & 1 & 3 & 2 & 2
\end{array}
$$

The pixels are represented by 2 bits; therefore, $L = 4$ and the intensity levels are in the range $[0, 3]$. The total number of pixels is 25, so the histogram has the components

$$p(r_0) = \frac{6}{25} = 0.24; \quad p(r_1) = \frac{7}{25} = 0.28;$$

$$p(r_2) = \frac{7}{25} = 0.28; \quad p(r_3) = \frac{5}{25} = 0.20$$

where the numerator in $p(r_i)$ is the number of pixels in the image with intensity level r_i. We can compute the average value of the intensities in the image using Eq. (3.3-18):

$$
\begin{aligned}
m &= \sum_{i=0}^{3} r_i \, p(r_i) \\
&= (0)(0.24) + (1)(0.28) + (2)(0.28) + (3)(0.20) \\
&= 1.44
\end{aligned}
$$

Letting $f(x, y)$ denote the preceding 5×5 array and using Eq. (3.3-20), we obtain

$$
\begin{aligned}
m &= \frac{1}{25} \sum_{x=0}^{4} \sum_{y=0}^{4} f(x, y) \\
&= 1.44
\end{aligned}
$$

- Global mean and variance are measured over entire image

  Used for gross adjustment of overall intensity and contrast
- Local mean and variance are measured locally

  Used for local adjustment of local intensity and contrast

(x,y): coordinate of a pixel

S_{xy}: neighborhood (subimage), centered on (x,y)

$r_0, \ldots, r_{L-1}$: L possible intensity values

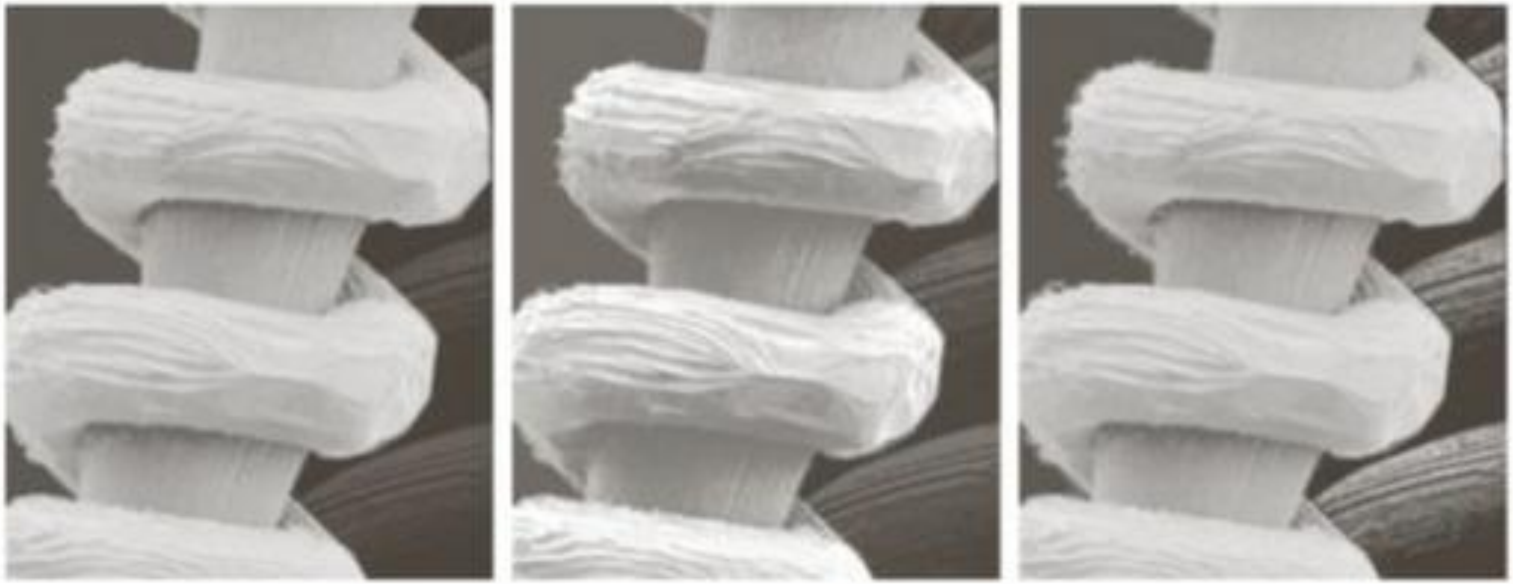
$p_{S_{xy}}$: histogram of pixels in region S_{xy}

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i \, p_{S_{xy}}(r_i)$$

local mean

$$\sigma^2_{S_{xy}} = \sum_{i=0}^{L-1} (r_i - m_{S_{xy}})^2 \, p_{S_{xy}}(r_i)$$

local variance



a b c

FIGURE 3.27 (a) SEM image of a tungsten filament magnified approximately 130×. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

- measure whether an area is relatively light or dark at (x,y)
  Compare the local average gray level $m_{S_{xy}}$ to the global mean m_G
  (x,y) is a candidate for enhancement if $m_{S_{xy}} \le k_0 m_G$
- Enhance areas that have low contrast
  Compare the local standard deviation $\sigma_{S_{xy}}$ to the global standard deviation σ_G
  (x,y) is a candidate for enhancement if $\sigma_{S_{xy}} \le k_2 \sigma_G$
- Restrict lowest values of contrast
  (x,y) is a candidate for enhancement if $k_1 \sigma_G \le \sigma_{S_{xy}}$
- Enhancement is processed simply multiplying the gray level by a constant E

$$g(x,y) = \begin{cases} E \cdot f(x,y) & \text{if } m_{S_{xy}} \le k_0 m_G \;\; AND \;\; k_1\sigma_G \le \sigma_{S_{xy}} \le k_2\sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

Spatial filtering:

- It consist:
  - Image neighborhood and sub image
  - Predefined operation

The sub-image is called a filter, mask, kernel, template or window

If operation is linear, then filter is called a linear spatial filter otherwise nonlinear.

Basics of Spatial Filtering

Sometime we need to manipulate values obtained from neighboring pixels

Example: How can we compute an average value of pixels in a 3x3 region center at a pixel z?

Pixel z

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

Image

Basics of Spatial Filtering (cont.)

Step 1: Select a pixel z.
Step 2: Select predefined neighborhood.
Step 3: select a sub image which will give u suitable result.
Step 4: Multiply elements of sub image with corresponding elements of image neighborhood

Pixel z

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

$$\times \; \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \;\rightarrow\;$$

Sub image

$$y = \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 9 + \frac{1}{9}\cdot 7 + \frac{1}{9}\cdot 6 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 6 + \frac{1}{9}\cdot 1$$

Question: How to compute the 3x3 average values at every pixels?

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

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

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

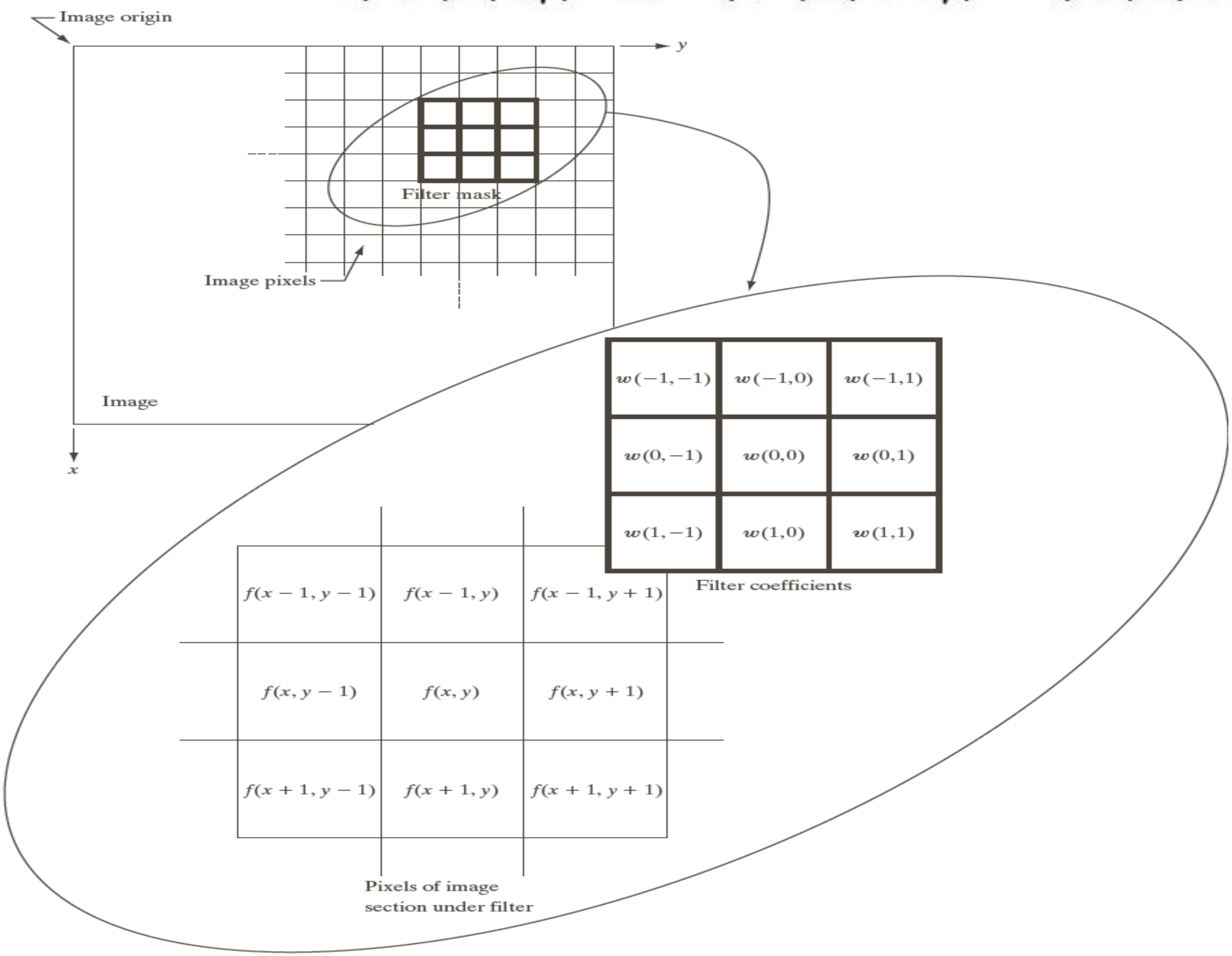
Solution: Imagine that we have a 3x3 window that can be placed everywhere on the image

Spatial filtering:

- Select a pixel(x, y)
- Select neighborhood of predefined size of that pixel.
- Select a sub image of same size as selected neighborhood
- Multiply element of image neighborhood and corresponding element of sub-image.
- Sum of that product: linear operation.
- Replace value of (x,y) with result.

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots$$

$$+ w(0,0)f(x,y) + \ldots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$$



Linear spatial filtering:

- Linear Filtering of an image f of size MxN filter mask of size mxn is given by the expression

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

where a = (m-1)/2 and b = (n-1)/2

To generate a complete filtered image this equation must be applied for x = 0, 1, 2, … , M-1 and y = 0, 1, 2, … , N-1

Correlation **Convolution**

	Correlation	Convolution	
	Origin f w	Origin f w rotated 180°	
(a)	0 0 0 1 0 0 0 0 1 2 3 2 8	0 0 0 1 0 0 0 0 8 2 3 2 1	(i)
(b)	0 0 0 1 0 0 0 0 1 2 3 2 8 Starting position alignment	0 0 0 1 0 0 0 0 8 2 3 2 1	(j)
(c)	Zero padding 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(k)
(d)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8 Position after one shift	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(l)
(e)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8 Position after four shifts	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(m)
(f)	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8 Final position	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 8 2 3 2 1	(n)
	Full correlation result	Full convolution result	
(g)	0 0 0 8 2 3 2 1 0 0 0 0	0 0 0 1 2 3 2 8 0 0 0 0	(o)
	Cropped correlation result	Cropped convolution result	
(h)	0 8 2 3 2 1 0 0	0 1 2 3 2 8 0 0	(p)

Origin $f(x, y)$

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \qquad \begin{array}{ccc} & w(x, y) & \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

(a)

Padded f

$$\begin{array}{ccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

(b)

Initial position for w

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

(c)

Full correlation result

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

(d)

Cropped correlation result

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

(e)

Rotated w

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

(f)

Full convolution result

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

(g)

Cropped convolution result

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

(h)

FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Rotation of 2D mask:

- Flip wrt one axes and then wrt other.

1	2	3
4	5	6
7	8	9

w

7	8	9
4	5	6
1	2	3

9	8	7
6	5	4
3	2	1

w1

w1(-1,-1)=w(1,1),   w1(-1,0)=w(1,0) ,...,   w1(1,1)=w(-1,-1)

w1(s,t)=w(-s,-t) where s and t varies from -1 to 1

3.4.2 Spatial Correlation and Convolution

- Correlation of a filter w(x,y) of size m x n with an image f(x,y)

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

- Convolution of w(x,y) and f(x,y)

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

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

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

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

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

Vector representation of Linear filtering:

$$
\begin{aligned}
R &= w_1 z_1 + w_2 z_2 + \ \dots \ + w_{mn} z_{mn} \\
&= \sum_{k=1}^{mn} w_k z_k \\
&= \mathbf{w}^T \mathbf{z}
\end{aligned}
$$

3.4.4 Generating Spatial Filter Masks

- Linear spatial filtering by 3 x 3 filter

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i \quad = \mathbf{w}^T \mathbf{z}$$

- Average value in 3 x 3 neighborhood

$$R = \frac{1}{9} \sum_{i=1}^{9} z_i$$

- Gaussian function

$$h(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Non linear filter :

- It requires :
  - Size of neighbor hood
  - Operation performed
- Operations like max, min etc

Smoothing spatial filters:

- Linear spatial filters for smoothing: averaging filters, lowpass filters
  - Noise reduction
  - Undesirable side effect: blur edges

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Standard average→ Box filter

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

weighted average

Which one will have less blurring??

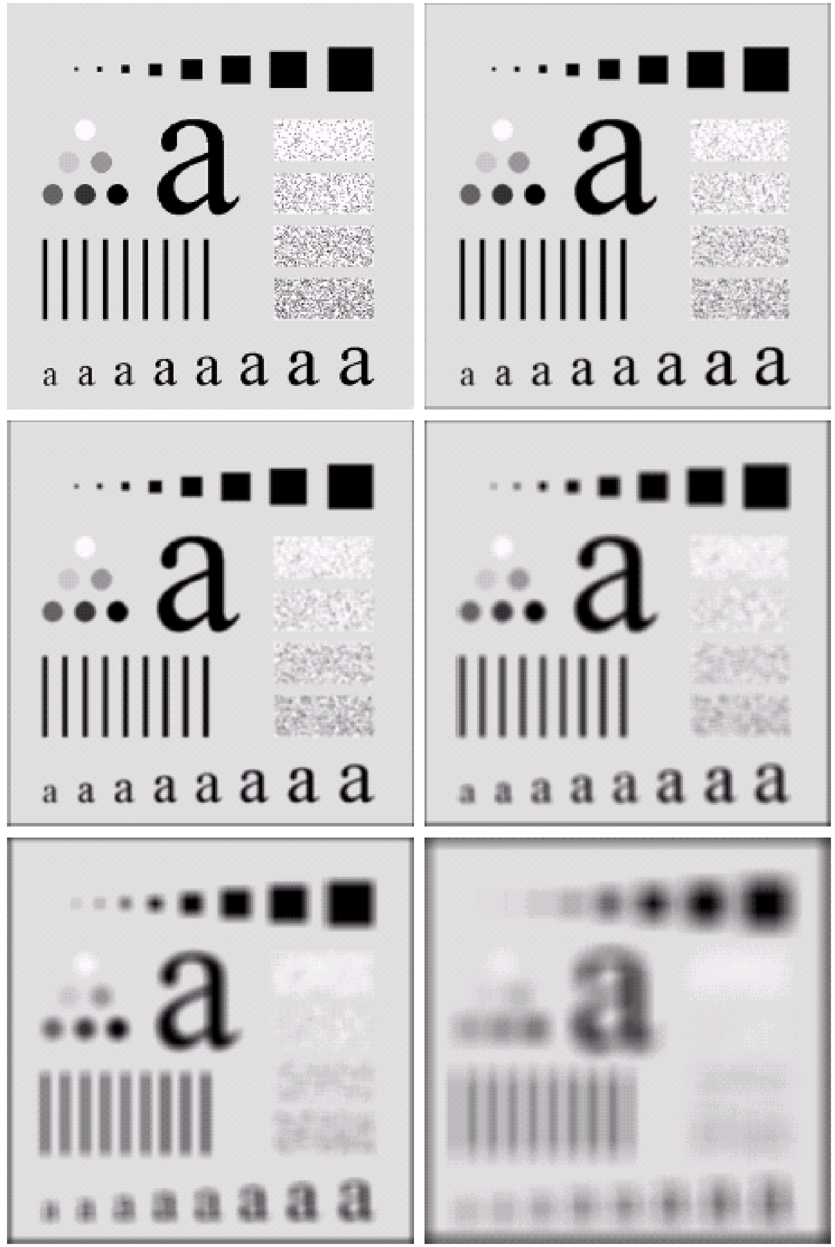
Smoothing spatial filters:

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

Result of smoothing with square averaging filter masks

Original

n=3



n=5

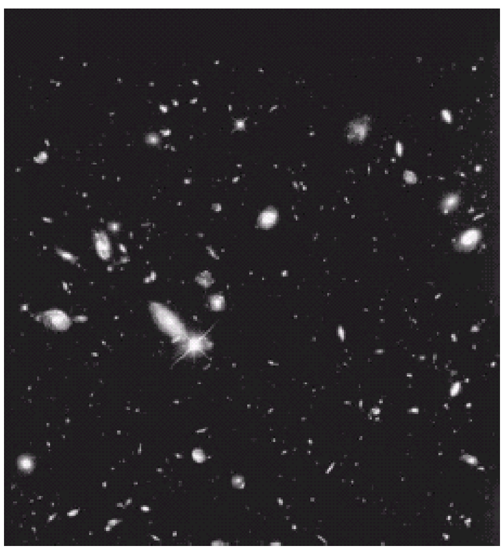
n=9

n=15

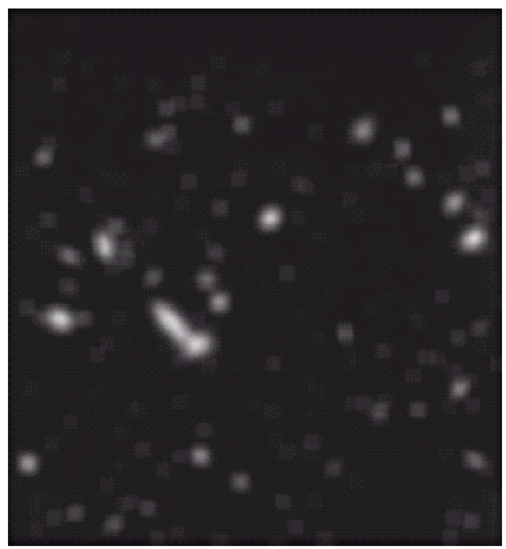
n=35

Another Smoothing Example

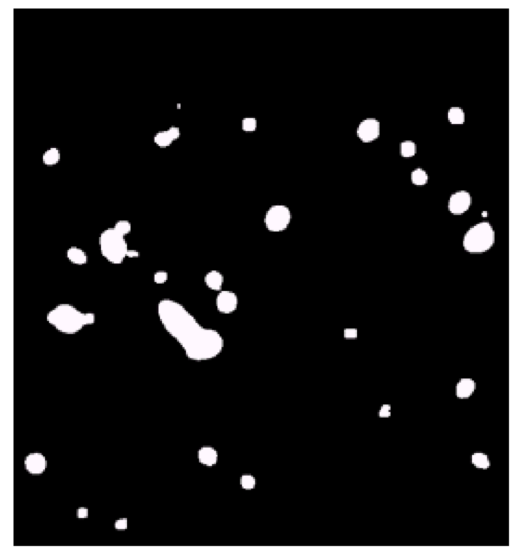
By smoothing the original image we get rid of lots of the finer detail which leaves only the gross features for thresholding



Original Image



Smoothed Image 15x15 mask



Thresholded Image

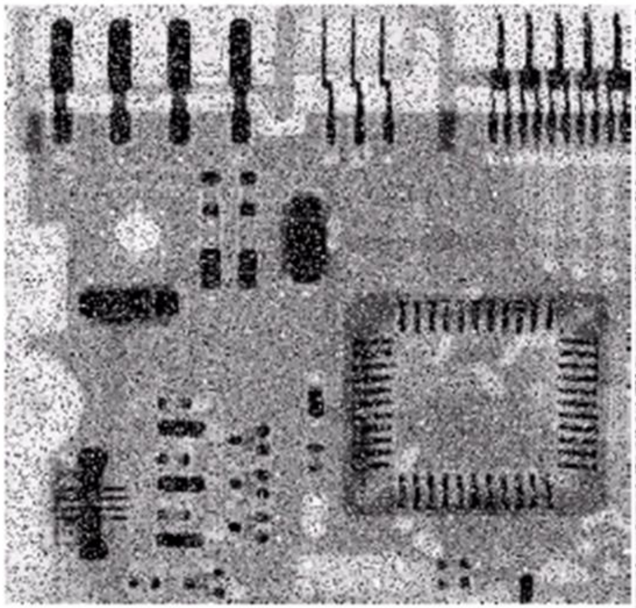
3.5.2 Order-Statistic (Nonlinear) Filters

- Order-statistic filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels
- Median filter
  - Replaces the pixel value by the median of the gray levels in the neighborhood of that pixel
  - Effective for impulse noise (salt-and-pepper noise)
  - 3x3 neighborhood: 5th largest value
  - 5x5 neighborhood: 13th largest value

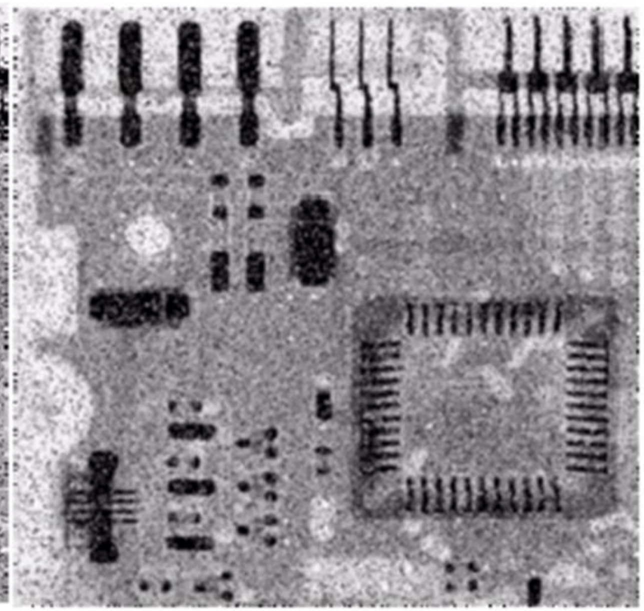
  - Isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2/2$, are eliminated by an n x n median filter

- Max filter: select maximum value in the neighborhood
- Min filter: select minimum value in the neighborhood

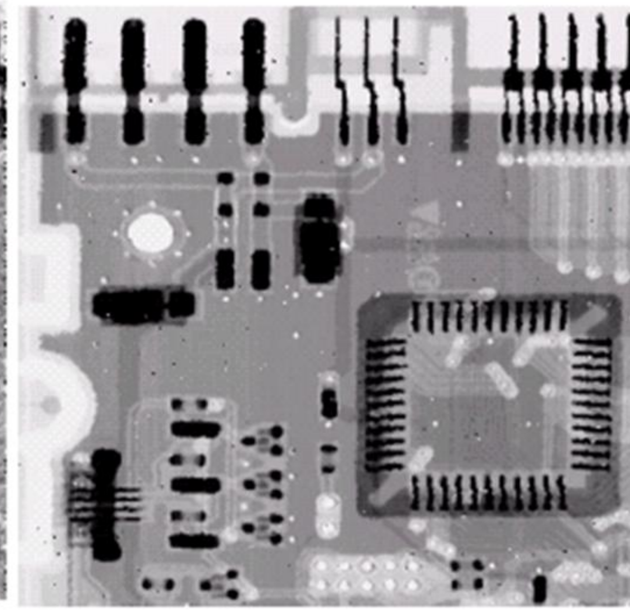
Order statistic filters:



Original Image



Averaging Mask



Median filter

Sharpening Spatial Filters

Previously we have looked at smoothing filters which remove fine detail

Sharpening spatial filters seek to highlight fine detail

- Remove blurring from images
- Highlight edges
- Enhances the noise

Sharpening filters are based on *spatial differentiation*

3.6.1 Foundation

- Image sharpening by first- and second-order derivatives
- Derivatives are defined in terms of differences
- Requirement of first derivative

1) Must be zero in flat areas
2) Must be nonzero at the onset (start) of step and ramp
3) Must be nonzero along ramps

- Requirement of second derivative

1) Must be zero in flat areas
2) Must be nonzero at the onset (start) of step and ramp
3) Must be zero along ramps of constant slope

3.6.1 Foundation

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

first-order derivative

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

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

second-order derivative

3.6.1 Foundation

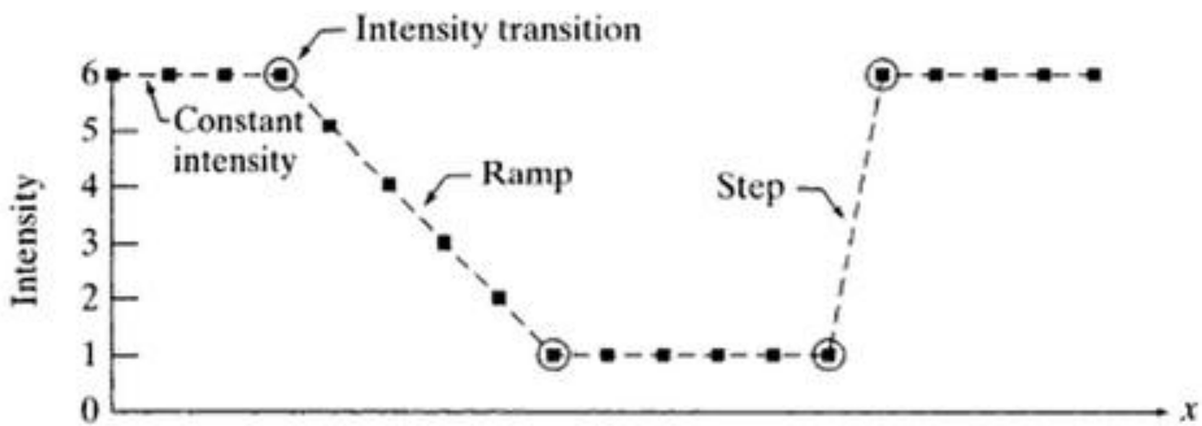
$$\frac{\partial f}{\partial x}(x) = f(x+1) - f(x)$$

first-order derivative

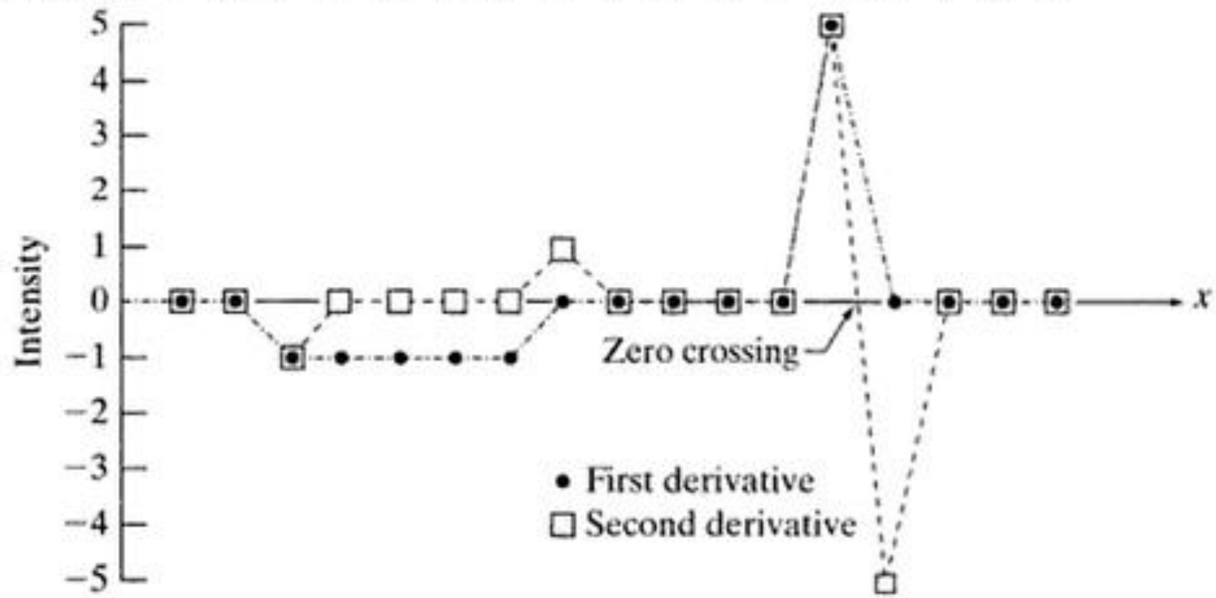
$$\frac{\partial f}{\partial x}(x-1) = f(x) - f(x-1)$$

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

second-order derivative



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6 → x
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	



a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

3.6.2 Use of Second Derivatives for Enhancement

- Isotropic filters: rotation invariant
- Simplest isotropic second-order derivative operator: Laplacian

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

2-D Laplacian operation

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

x-direction

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

y-direction

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

0	1	0
1	-4	1
0	1	0

Fig: a

1	1	1
1	-8	1
1	1	1

Fig: b

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

Fig: c

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

Fig: d

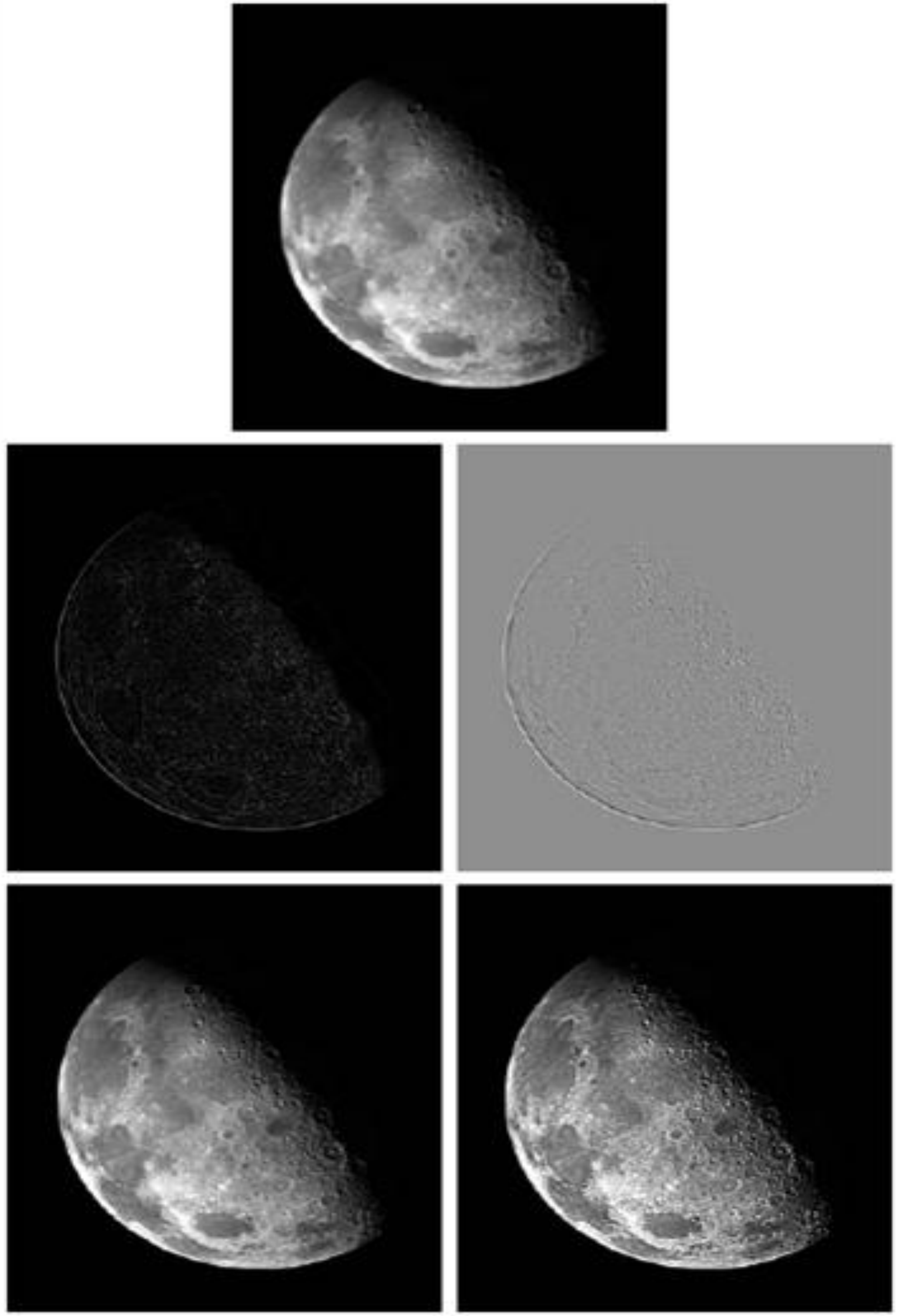
3.6.2 Use of Second Derivatives for Enhancement

- Image enhancement (sharpening) by Laplacian operation

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is positive} \end{cases}$$

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

where $f(x, y)$ and $g(x, y)$ are the input and sharpened images, respectively. The constant is $c = -1$ if the Laplacian filters in Fig. 3.37(a) or (b) are used, and $c = 1$ if either of the other two filters is used.



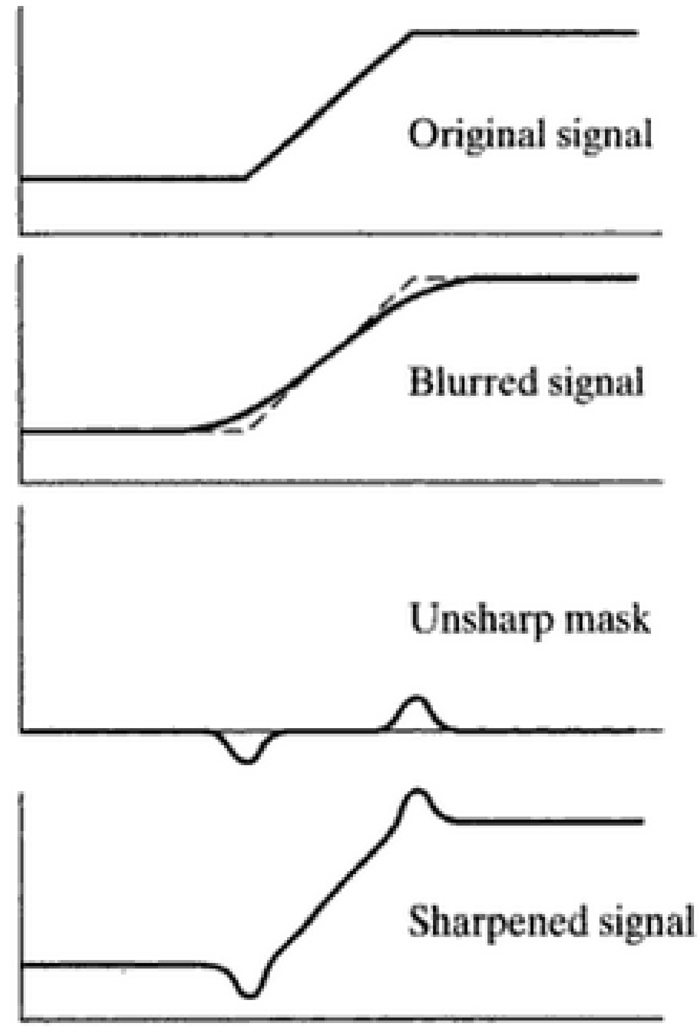
Unsharp Masking and Highboost Filtering :

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

original image – blurred image

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

- When k=1, unsharp masking
- When k > 1, 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).

a
b
c
d
e

FIGURE 3.40
(a) Original image.
(b) Result of blurring with a Gaussian filter.
(c) Unsharp mask. (d) Result of using unsharp masking.
(e) Result of using highboost filtering.

First Derivative – The Gradient

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\[2ex] \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$\begin{aligned} \nabla f &= \operatorname{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} \end{aligned}$$

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

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.

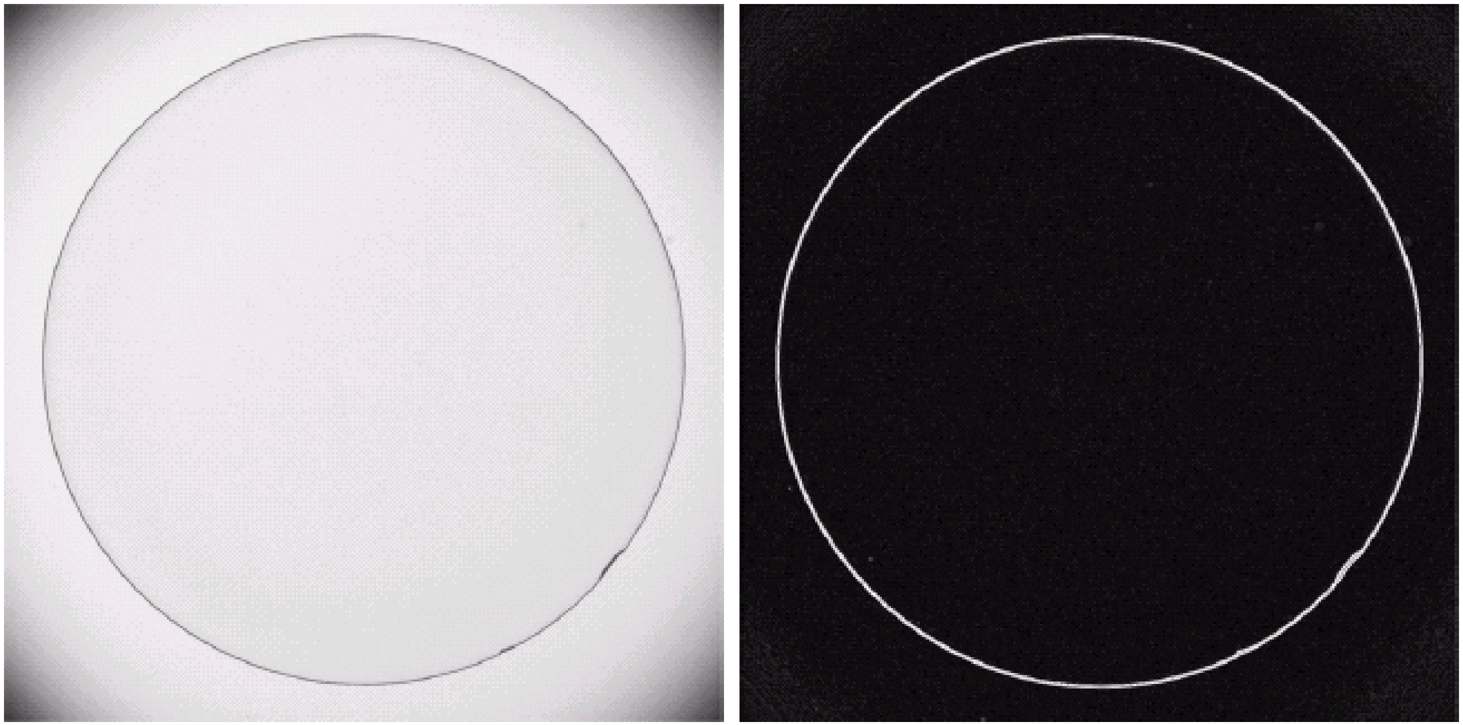
$$M(x, y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2\right]^{1/2}$$

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

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$\begin{aligned} M(x, y) \approx {} & |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| \\ & + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)| \end{aligned}$$



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.)

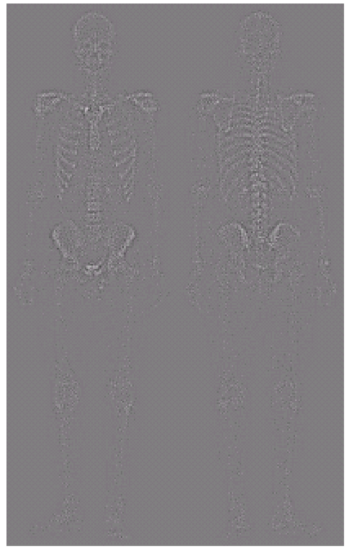
Combining Spatial Enhancement Methods:

Successful image enhancement is typically not achieved using a single operation

Rather we combine a range of techniques in order to achieve a final result

Combining Spatial Enhancement Methods

(a)



(b)

Laplacian filter of bone scan (a)

(c)

Sharpened version of bone scan achieved by subtracting (a) and (b)

(d)

Sobel filter of bone scan (a)

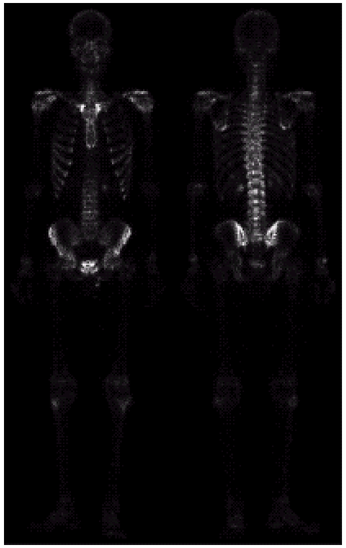
Combining Spatial Enhancement Methods

Image (d) smoothed with a 5*5 averaging filter

(e)

The product of (c) and (e) which will be used as a mask

(f)



Sharpened image which is sum of (a) and (f)

(g)

Result of applying a power-law trans. to (g)

(h)

Combining Spatial Enhancement Methods

Compare the original and final images