Last Lesson Matrix Representation

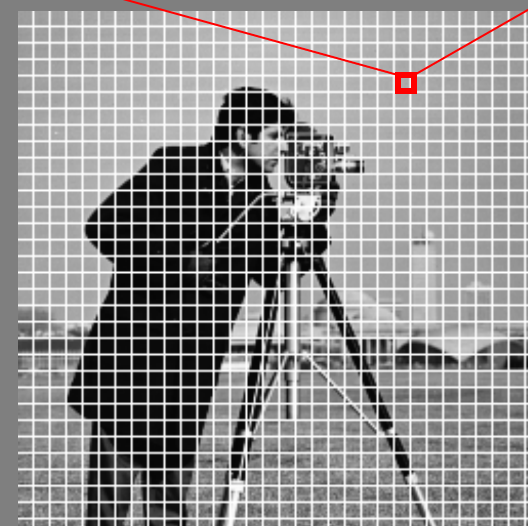
$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$\begin{bmatrix} 183 & 160 & 94 & 153 & 194 & 163 & 132 & 165 \\ 183 & 153 & 116 & 176 & 187 & 166 & 130 & 169 \\ 179 & 168 & 171 & 182 & 179 & 170 & 131 & 167 \\ 177 & 177 & 179 & 177 & 179 & 165 & 131 & 167 \\ 178 & 178 & 179 & 176 & 182 & 164 & 130 & 171 \\ 179 & 180 & 180 & 179 & 183 & 169 & 132 & 169 \\ 179 & 179 & 180 & 182 & 183 & 170 & 129 & 173 \\ 180 & 179 & 181 & 179 & 181 & 170 & 130 & 169 \end{bmatrix}$$

H=256

W=256

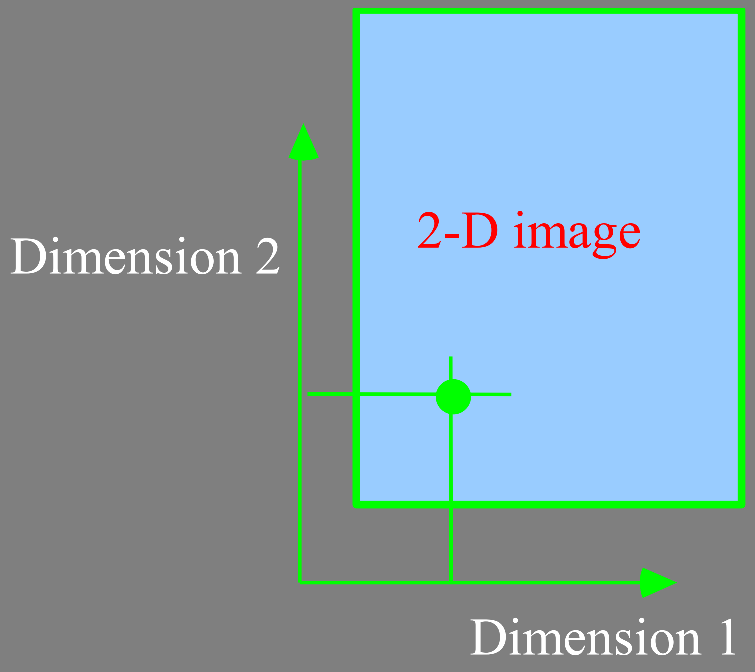
Divide into 8x8 blocks

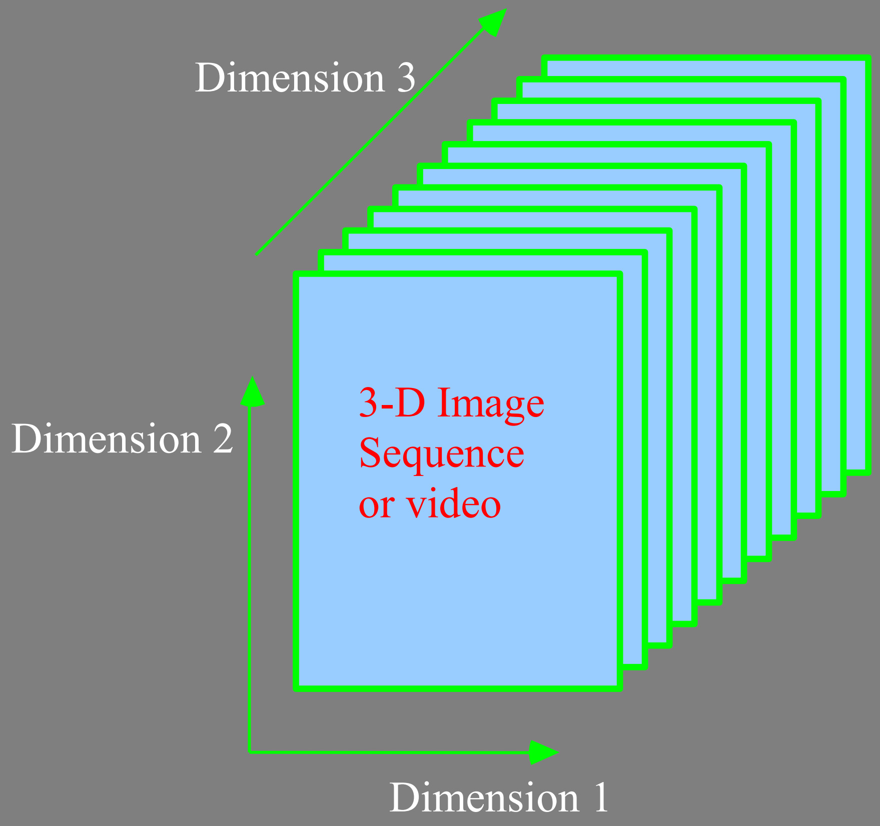


From [Gonzalez & Woods]

Last Lesson Dimensionality of Digital Images

- Images and videos are **multi-dimensional** (≥ 2 dimensions) signals.





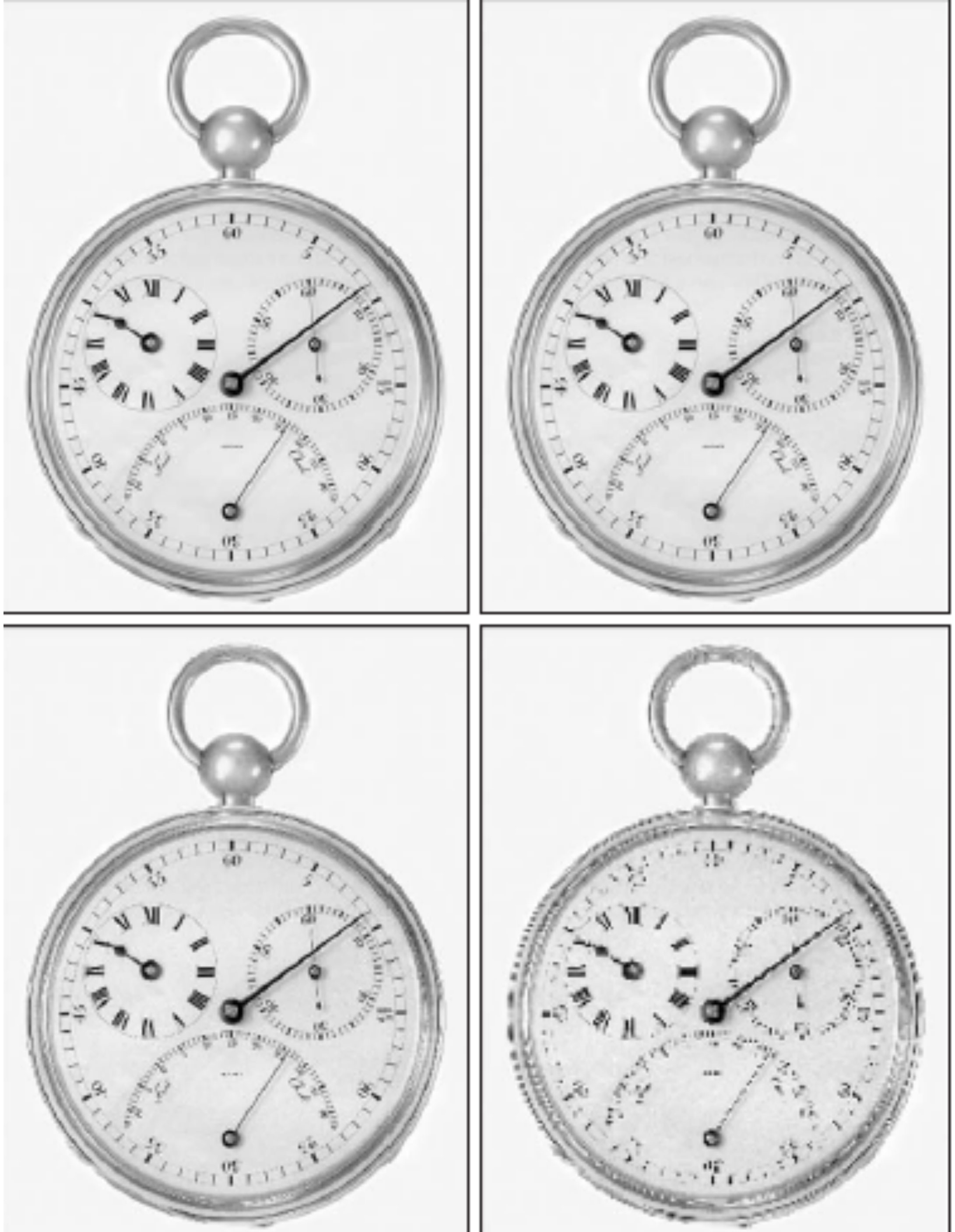
From Prof. Alan C. Bovik

TODAY

Spatial Resolution
Point processing
Bit planes
Histogram Equalizatioın
Resempling
Basic transformations
Histogram

Upon completion of this chapter, readers should:

- Understand the meaning of spatial domain processing, and how it differs from transform domain processing.
- Be familiar with the principal techniques used for intensity transformations.
- Understand the physical meaning of image histograms and how they can be manipulated for image enhancement.
- Understand the mechanics of spatial filtering, and how spatial filters are formed.
- Understand the principles of spatial convolution and correlation.
- Be familiar with the principal types of spatial filters, and how they are applied.
- Be aware of the relationships between spatial filters, and the fundamental role of lowpass filters.
- Understand how to use combinations of enhancement methods in cases where a single approach is insufficient.



Spatial Resolotion

FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

From [Gonzalez & Woods]

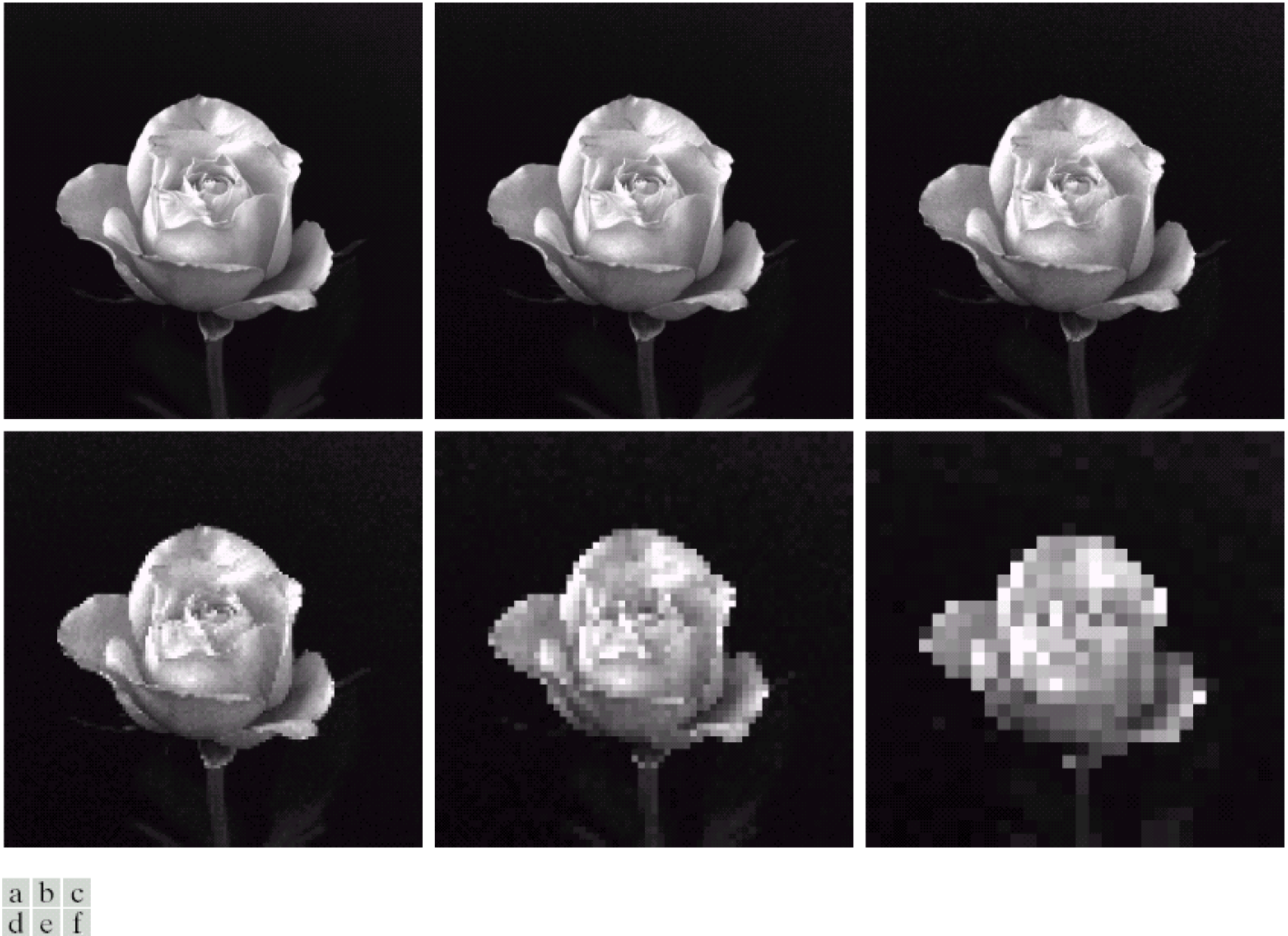


FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

From [Gonzalez & Woods]

Resolution: How Much Is Enough?

- The big question with resolution is always *how much is enough*?
  - This all depends on what is in the image and what you would like to do with it
  - Key questions include
    - Does the image look aesthetically pleasing?
    - Can you see what you need to see within the image?

Resolution: How Much Is Enough? (cont…)

- The picture on the right is fine for counting the number of cars, but not for reading the number plate

Intensity Level Resolution (cont…)

Low Detail

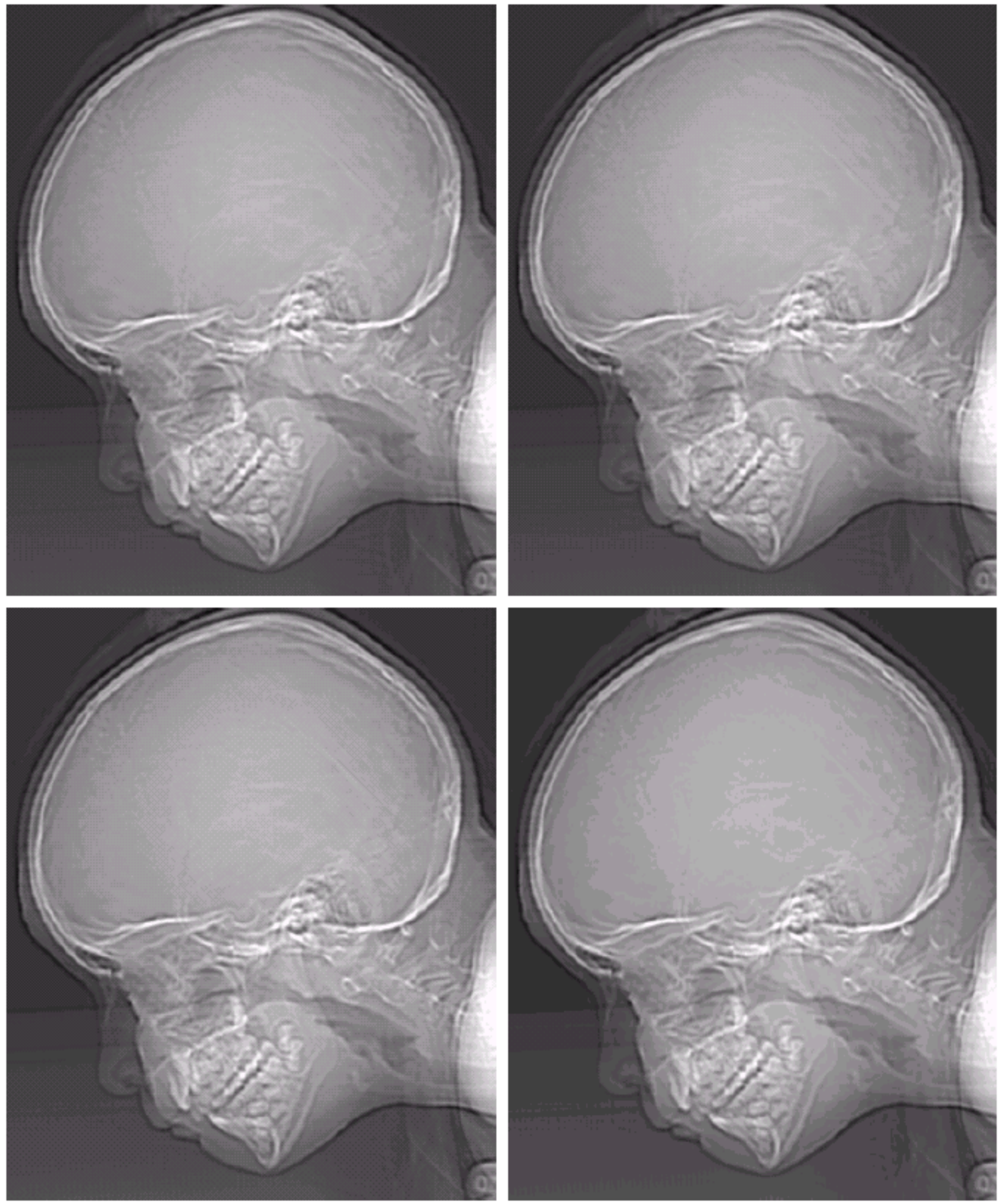
Medium Detail

High Detail

Intensity Level Resolution (cont…)

Intensity Level Resolution (cont…)

Intensity Level Resolution (cont…)



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.

From [Gonzalez & Woods]

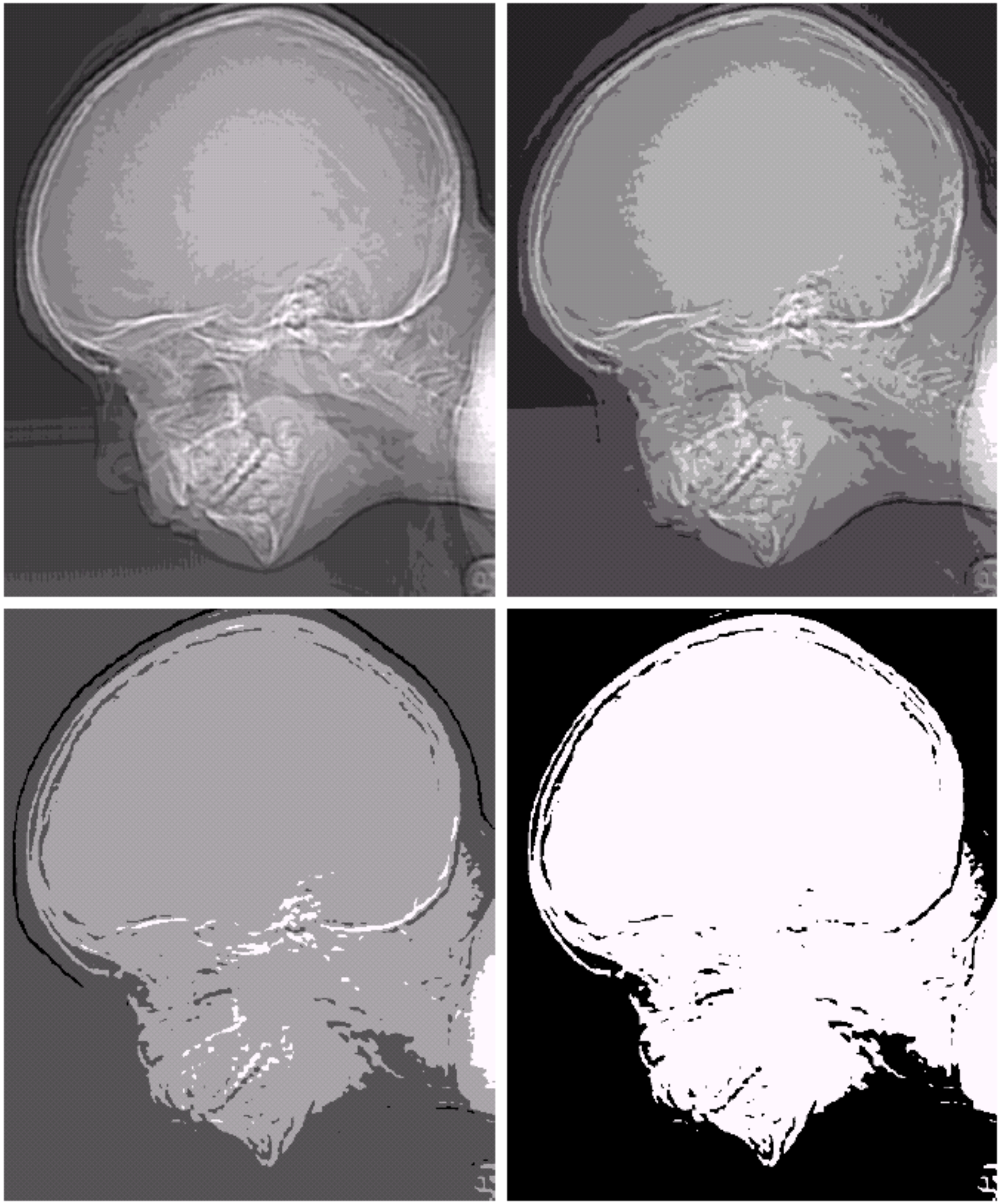
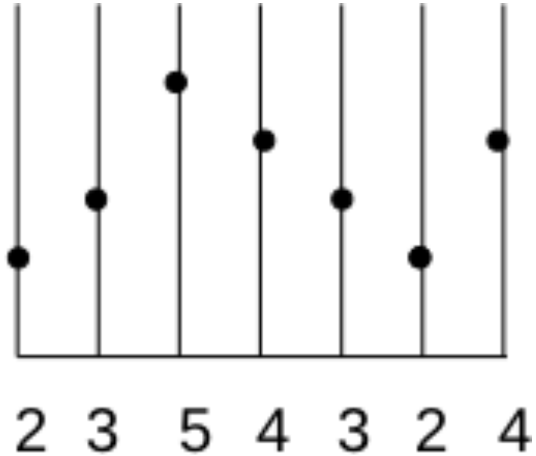
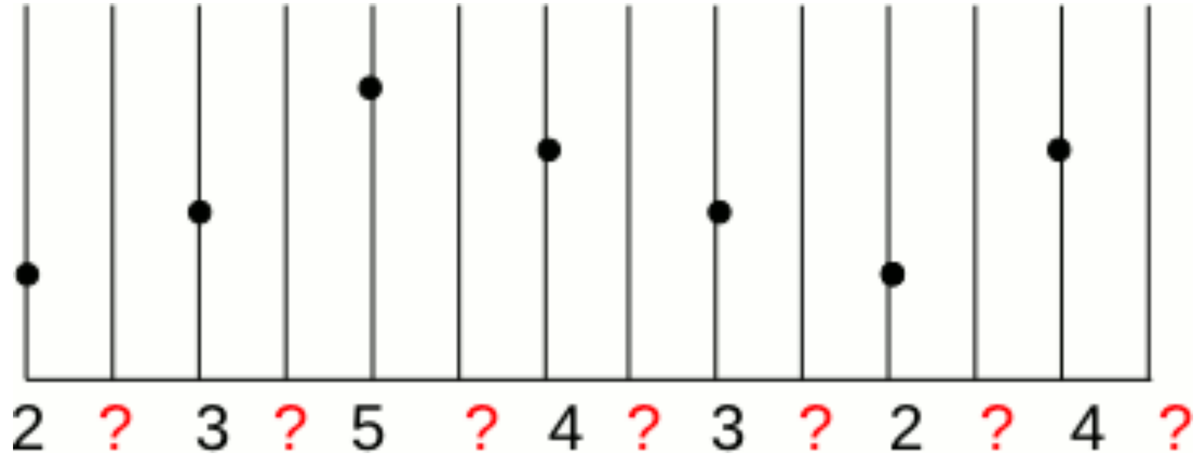
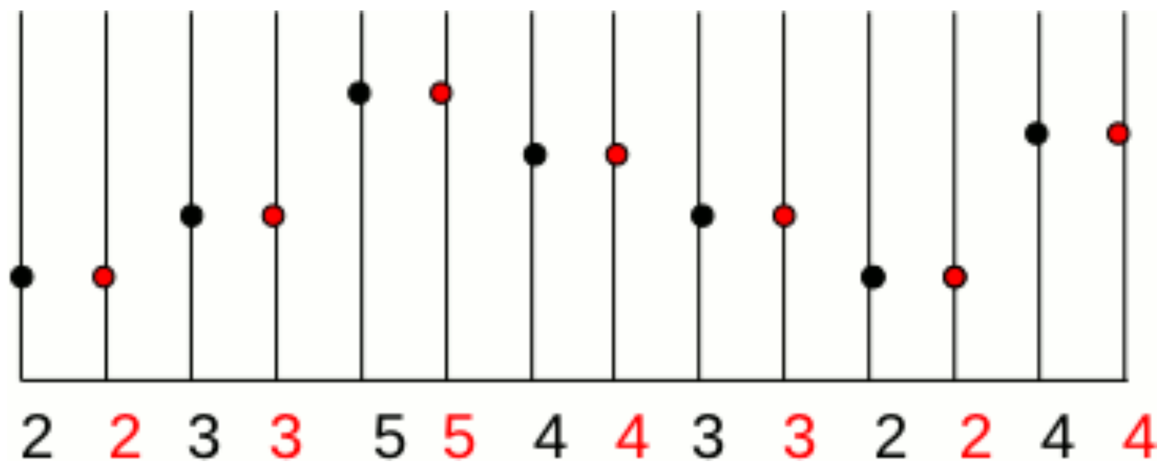


FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

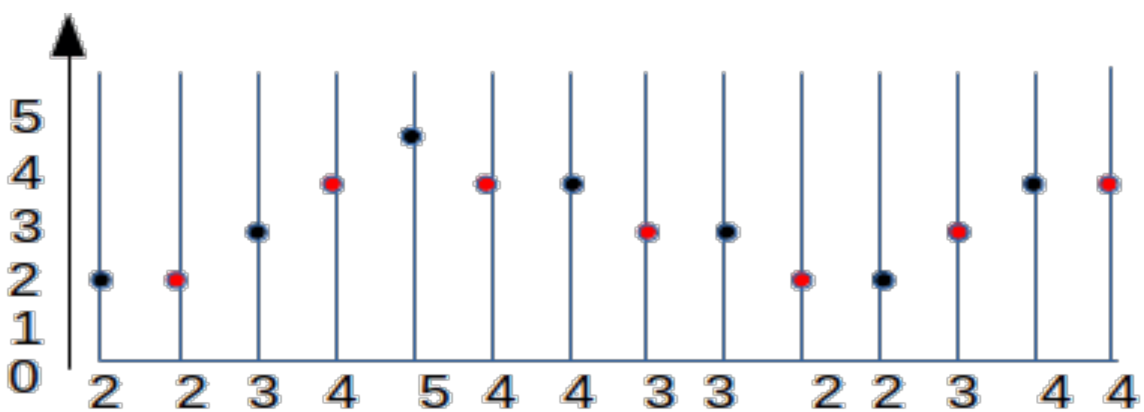
From [Gonzalez & Woods]







Nearest Neighboor interpolation



Bilinear interpolation

Resampling by convolution

Nearest neighboor kernel [1, 1, 0]

nearest neighbor kernel (tripling) [0, 1, 1, 1, 0]

Linear interpolation kernel [0.5 1 0.5] I

Linear interpolation kernel (tripling) [1/3, 2/3, 1, 2/3, 1 / 3]

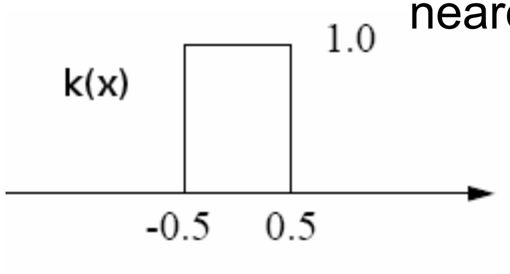
$$g(x') = \sum_i f(x+i).k(i)$$

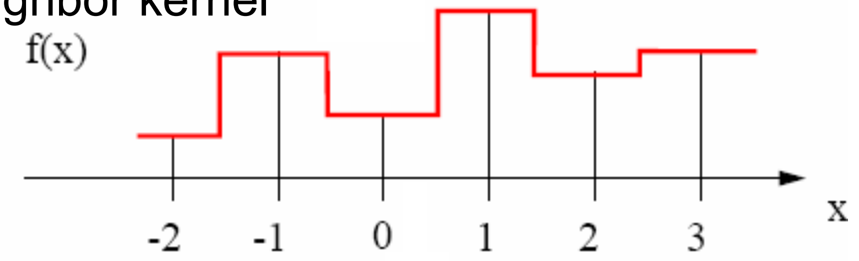
k(i)				0	1/3	2/3	1	2/3	1/3	0			
f'(x)	25	0	0	22	0	0	34	0	0	50	0	0	32
g(x)							34						
	0	1	2	3	4	5	6	7	8	9	10	11	12

k(i)				0	1/3	2/3	1	2/3	1/3	0			
f'(x)	0	0	22	0	0	34	0	0	50	0	0	32	0
g(x)						34	39						
	0	1	2	3	4	5	6	7	8	9	10	11	12

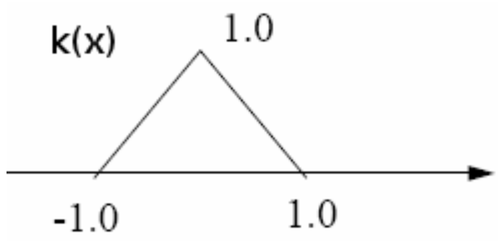
A variety of kernels

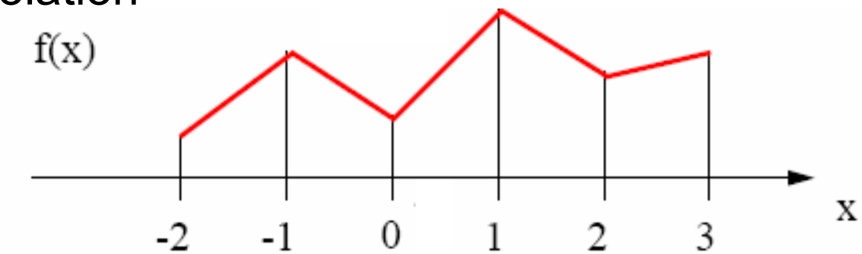
nearest neighbor kernel



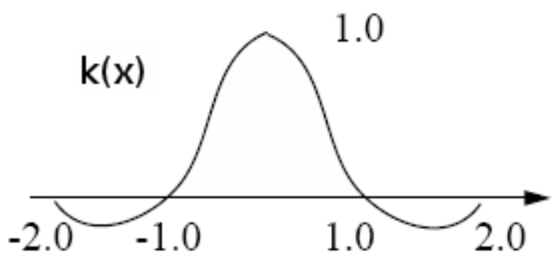


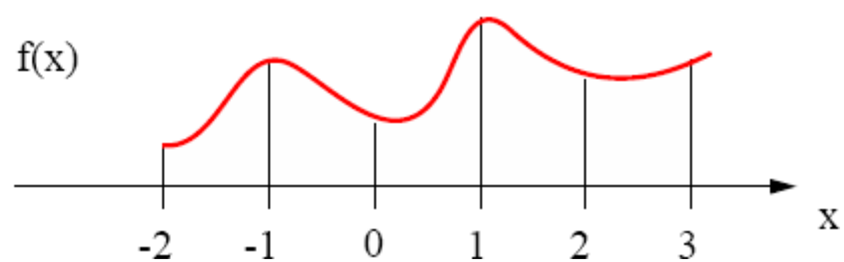
Linear interpolation





B-splines



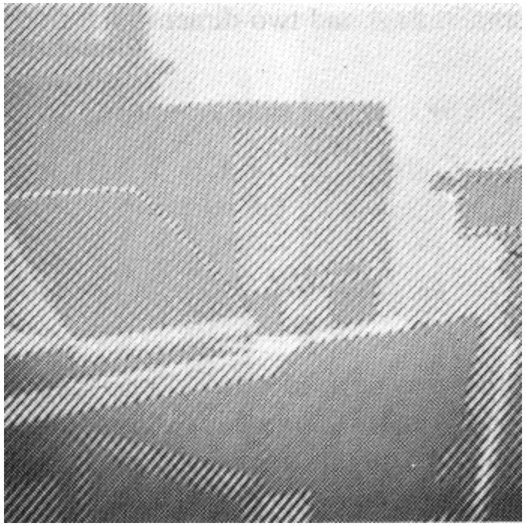


2D interpolation:

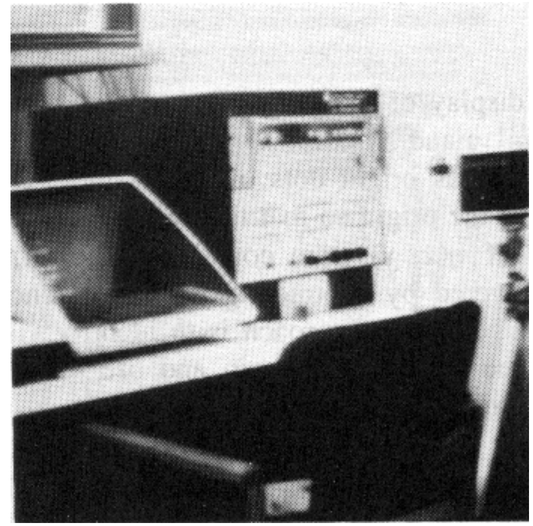
$$g(x_0,y_0)=\sum_{x_0-a+1}^{x_0+a}\sum_{y_0-a+1}^{y_0-a+1} k(x_0-i).k(y_0-j).f(x,y)$$

Intensity Transformations and Spatial Filtering

$$g(x, y) = T[f(x, y)]$$



Noisy image



Noise-cleaned image

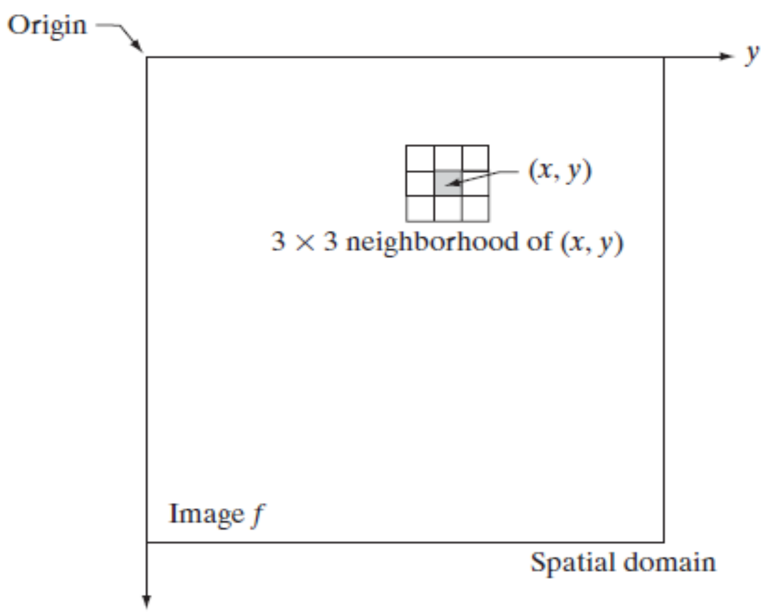


Image Enhancement

To process an image so that the result is more suitable than the original image for a *specific* application. Spatial domain methods and frequency domain methods

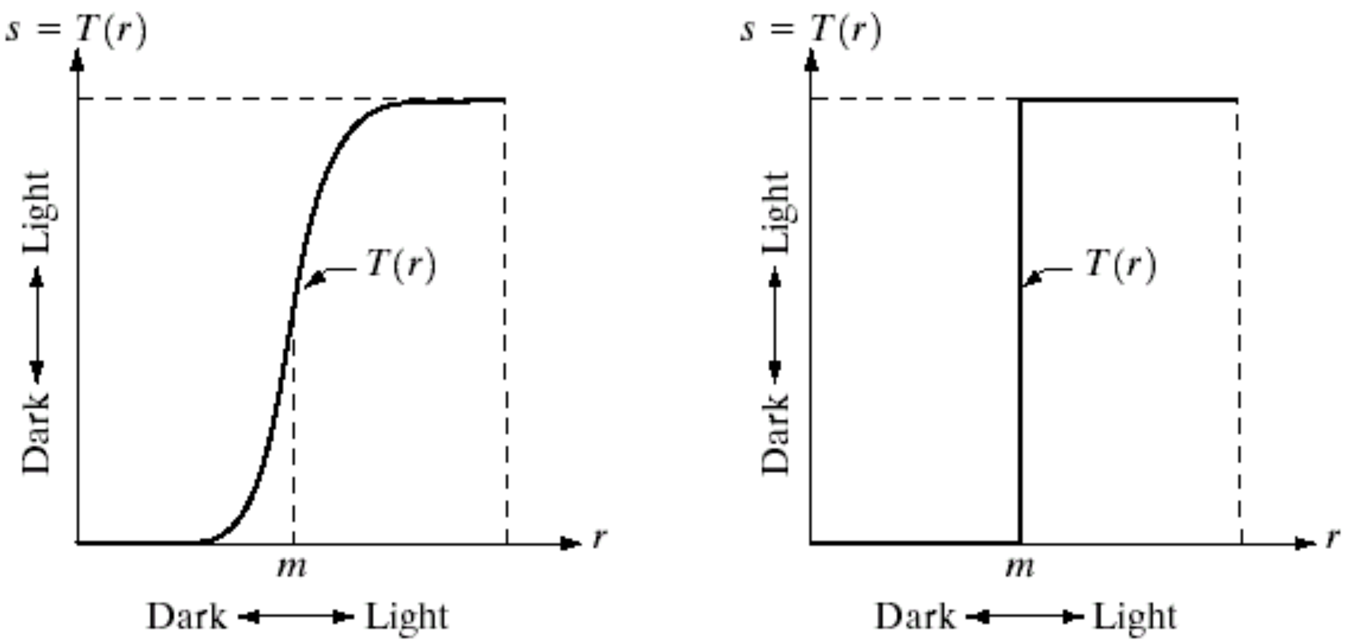
Spatial Domain Methods : Procedures that operate directly on the aggregate of pixels composing an image $g(x,y)=T[f(x,y)]$ A neighborhood about (x,y) is defined by using a square (or rectangular) subimage area centered at (x,y).

When the neighborhood is 1 x 1 then g depends only on the value of f at (x,y) and T becomes a gray-level transformation (or mapping) function:

$$s=T(r)$$

r,s: gray levels of f(x,y) and g(x,y) at (x,y)

Point processing techniques (e.g. contrast stretching, thresholding)

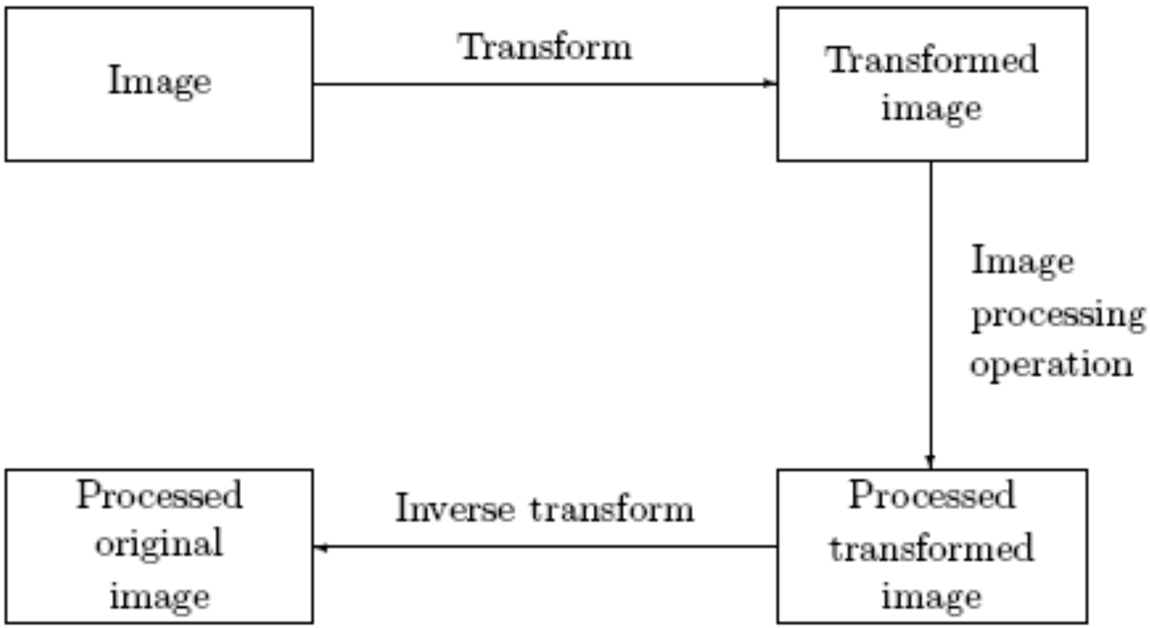


a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

POINT PROCESSING

Transform: represents the pixel values in some other, but equivalent form



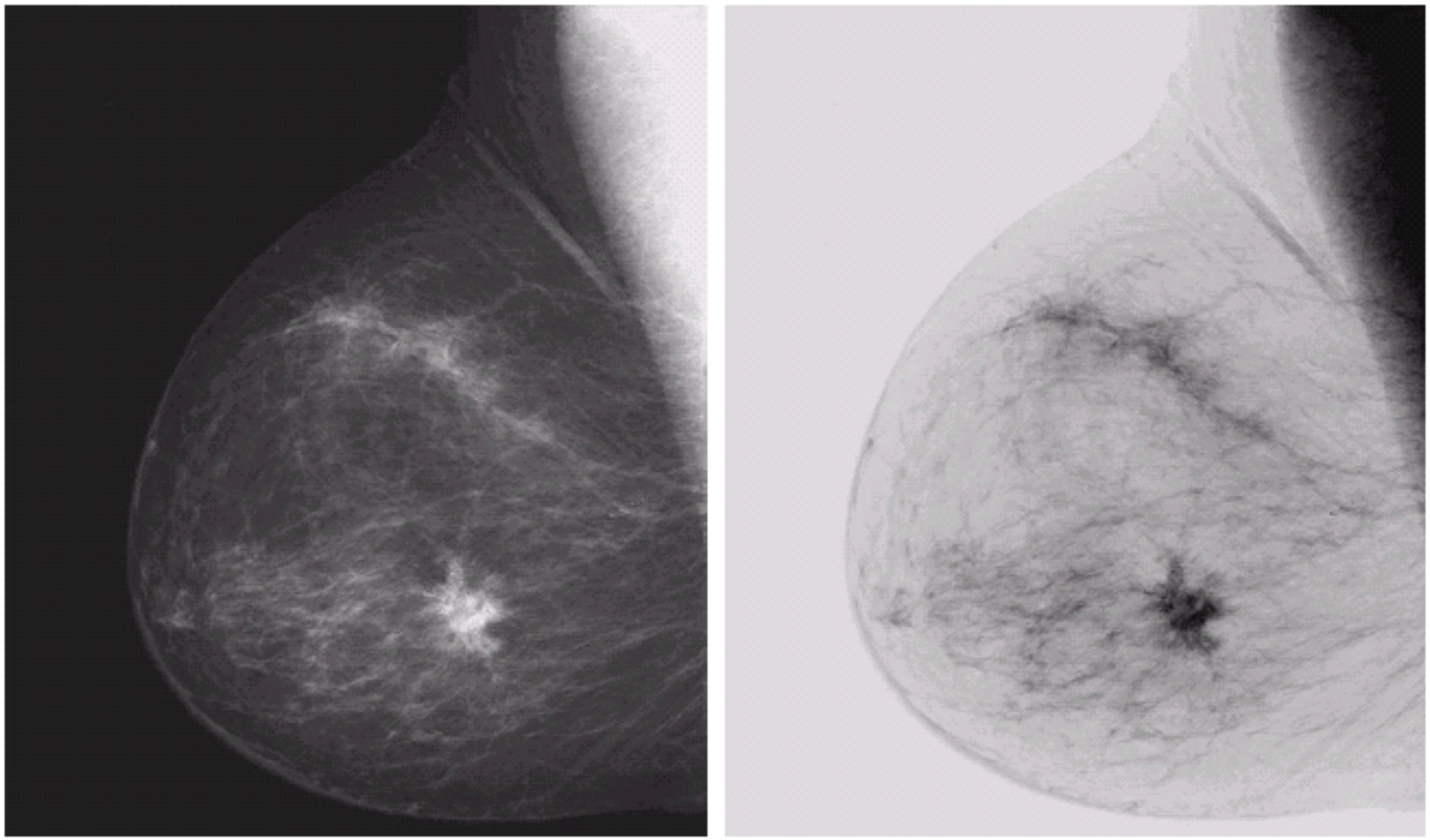
Schema for transform processing

Neighbourhood processing. To change the grey level of a given pixel we need only know the value of the grey levels in a small neighbourhood of pixels around the given pixel.

Point operations. A pixel's grey value is changed without any knowledge of its surrounds.

Image Negatives

Function reverses the order from black to white so that the intensity of the output image decreases as the intensity of the input increases. Used mainly in medical images and to produce slides of the screen.



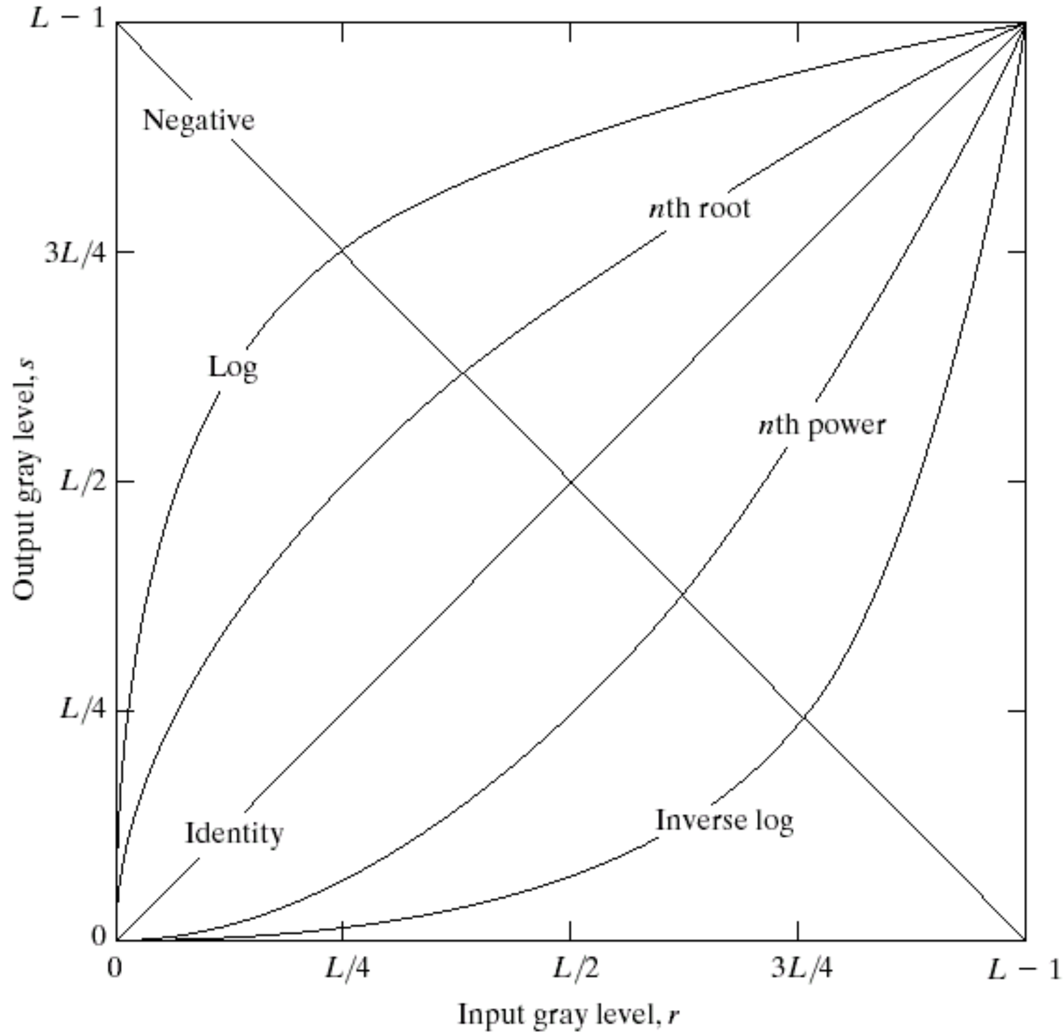
a b

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.)

$$s = L - 1 - r$$

Basic Transformations

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



Negative:

$$s = L - 1 - r$$

Log:

Inverse Log:

$$s = e^{cr} - 1$$

Power-law:

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

……

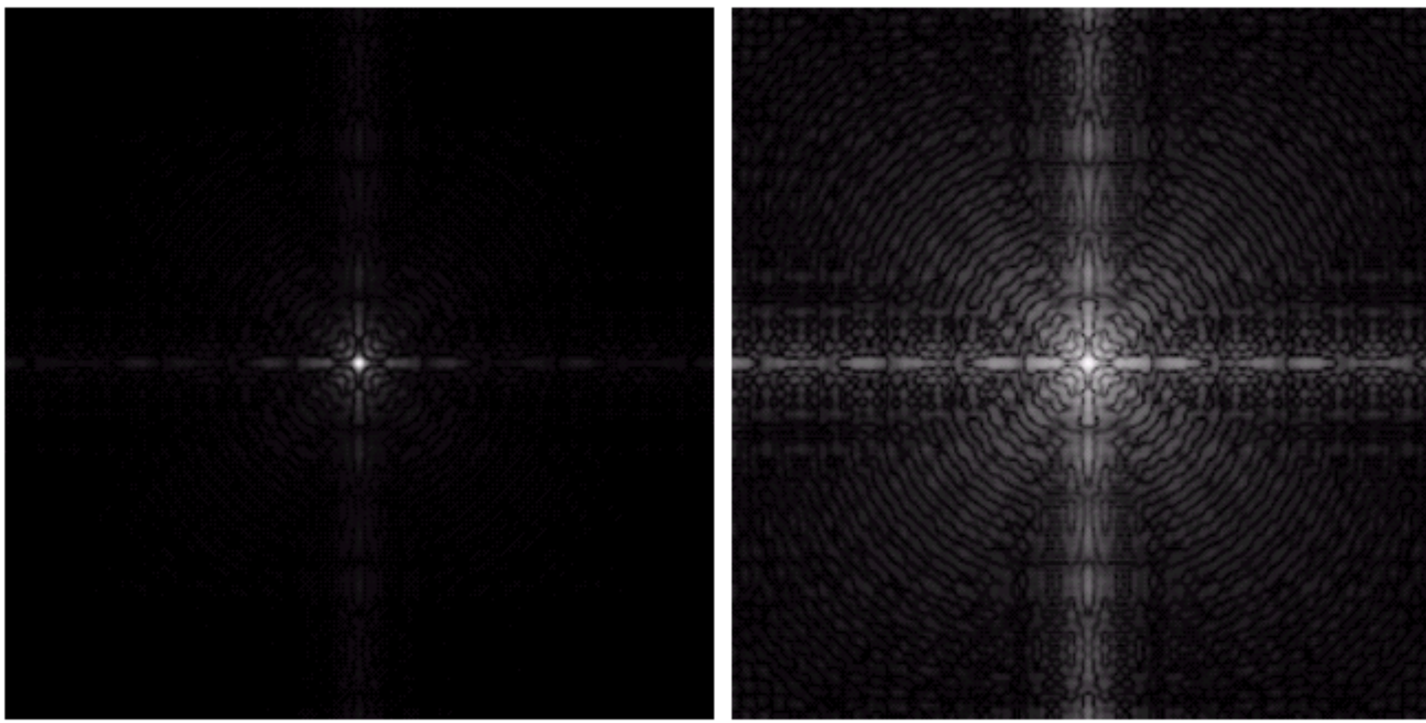
Log Transformation

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

- Compresses the dynamic range of images with large variations in pixel values

a b

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



From [Gonzalez & Woods]

Power-law (Gamma) Transformation

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

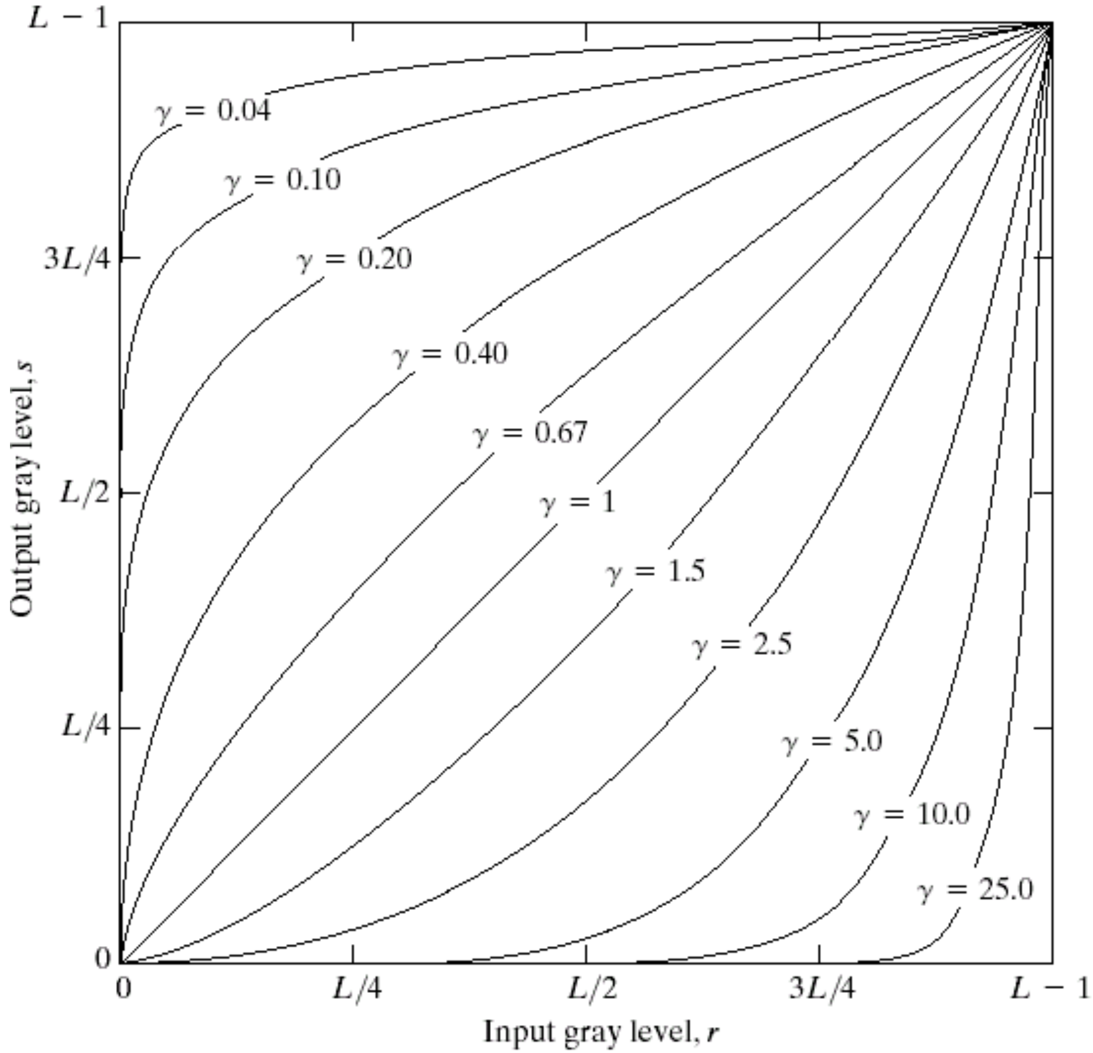


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

From [Gonzalez & Woods]

Power-law (Gamma) Transformation

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

a	b
c	d

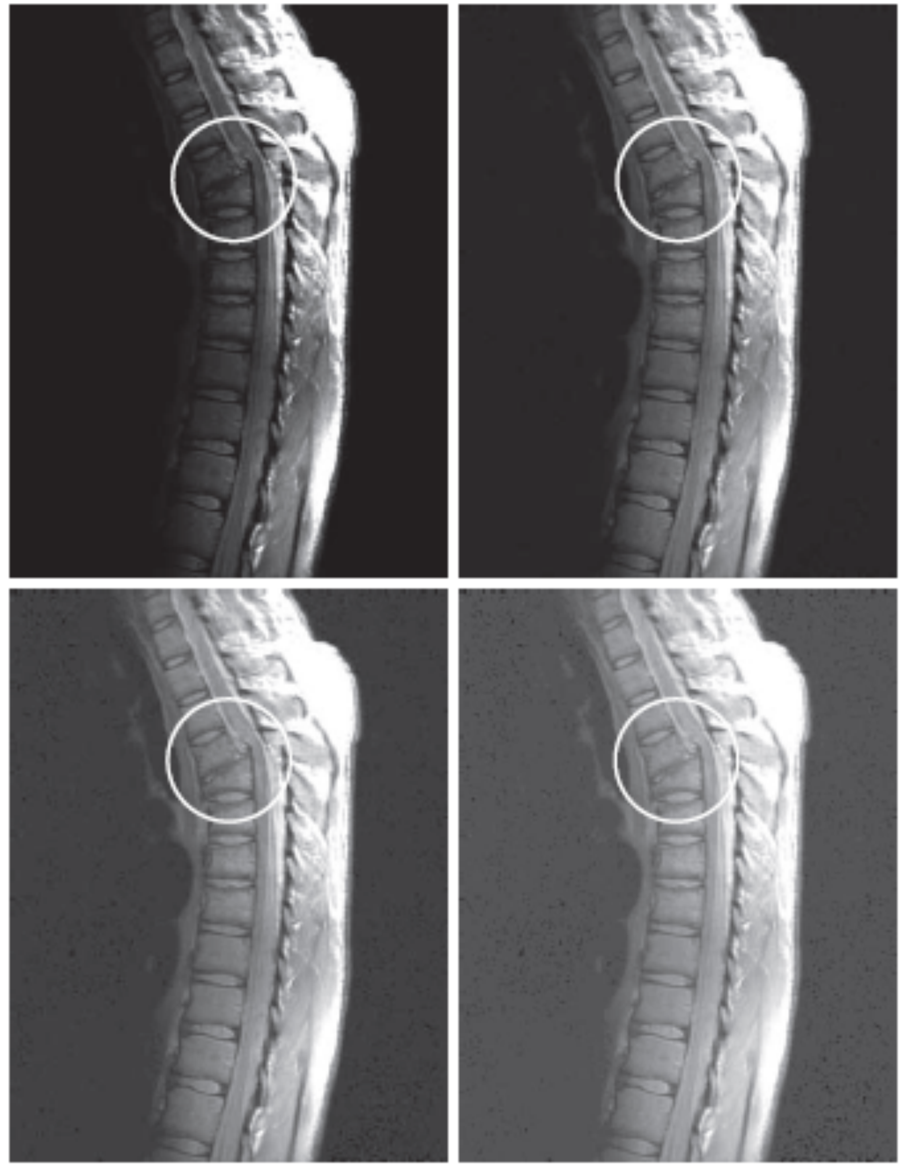


FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine (the region of the fracture is enclosed by the circle). (b)–(d) Results of applying the transformation in Eq. (3-5) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

From [Gonzalez & Woods]

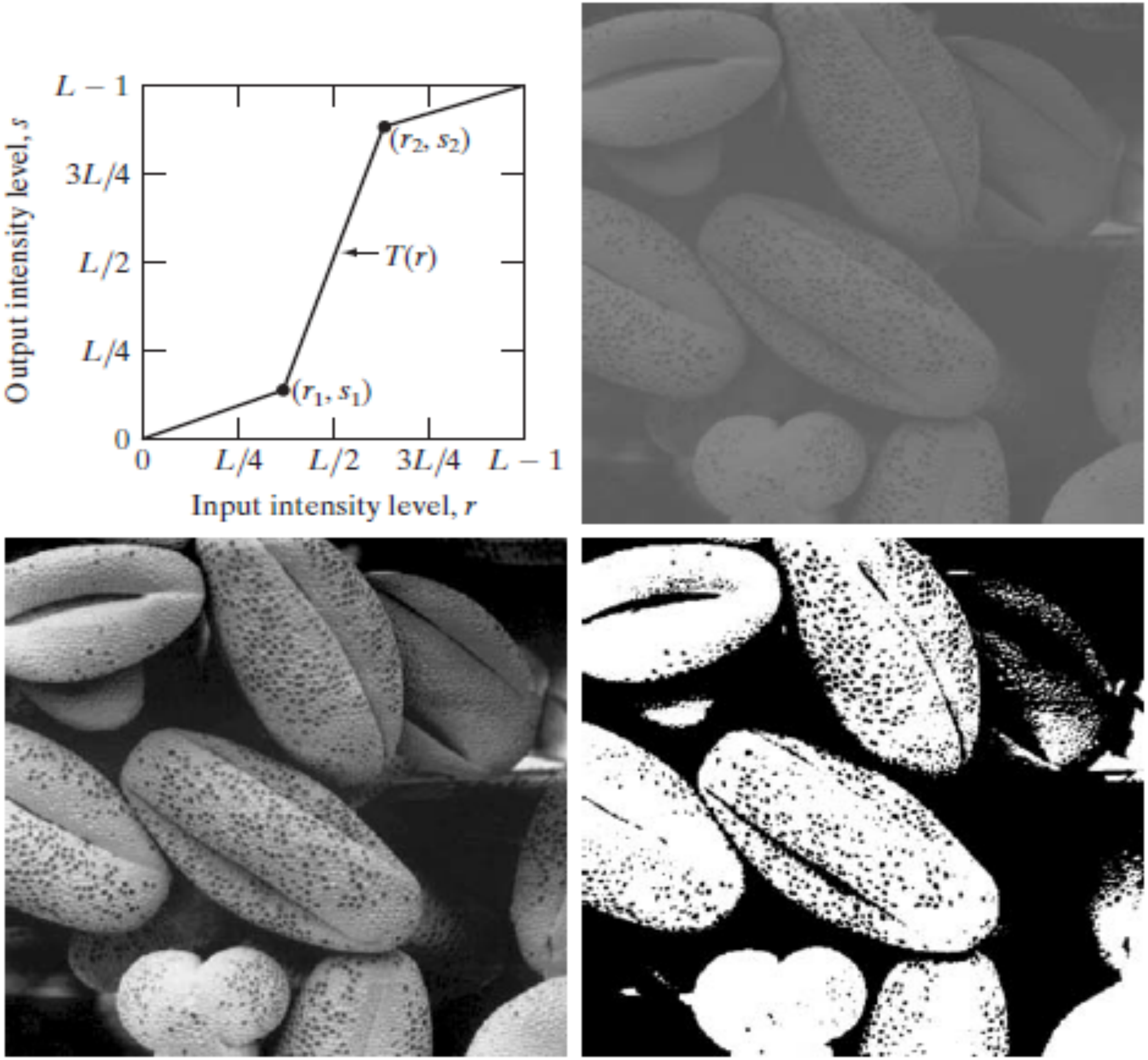
Power-law (Gamma) Transformation

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

a b
c d

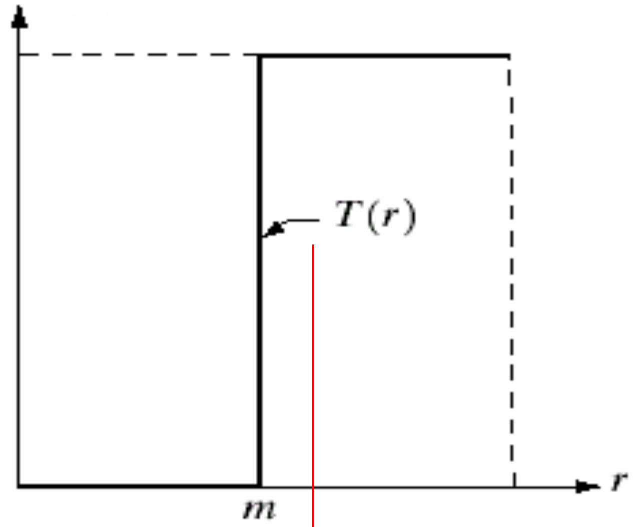
FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

From [Gonzalez & Woods]



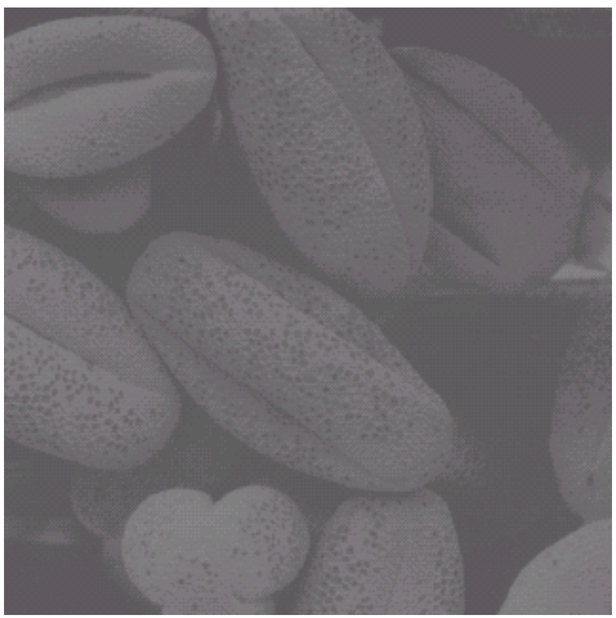
$L-1$
$3L/4$
$L/2$
$L/4$
0
(r_2, s_2)
$T(r)$
(r_1, s_1)
0
$L/4$
$L/2$
$3L/4$
$L-1$
Output intensity level, s
Input intensity level, r

Thresholding

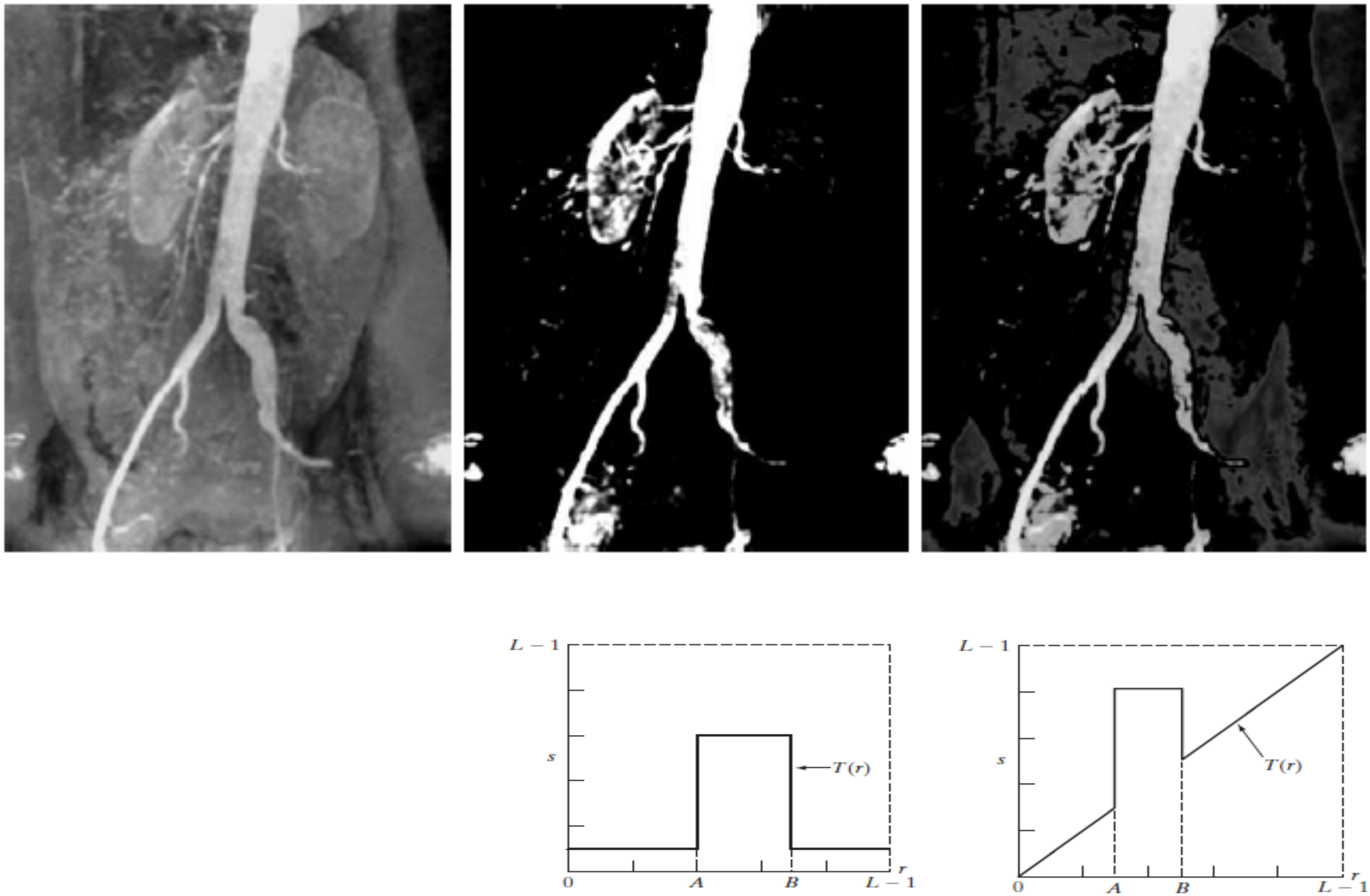


$$s = \begin{cases} c & if\ r \le m \\ 0 & if\ r > m \end{cases}$$

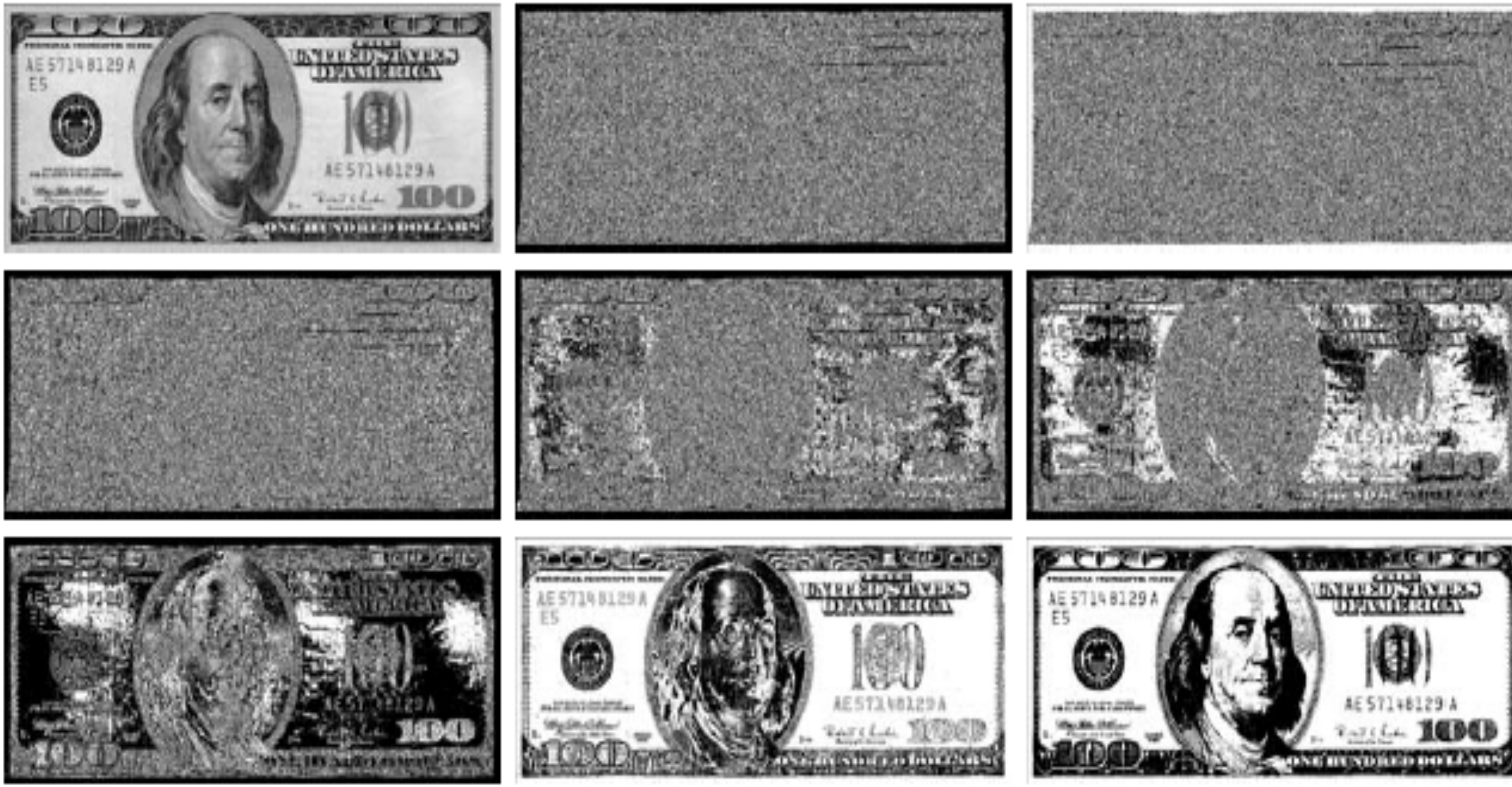
m : threshold



From [Gonzalez & Woods]


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

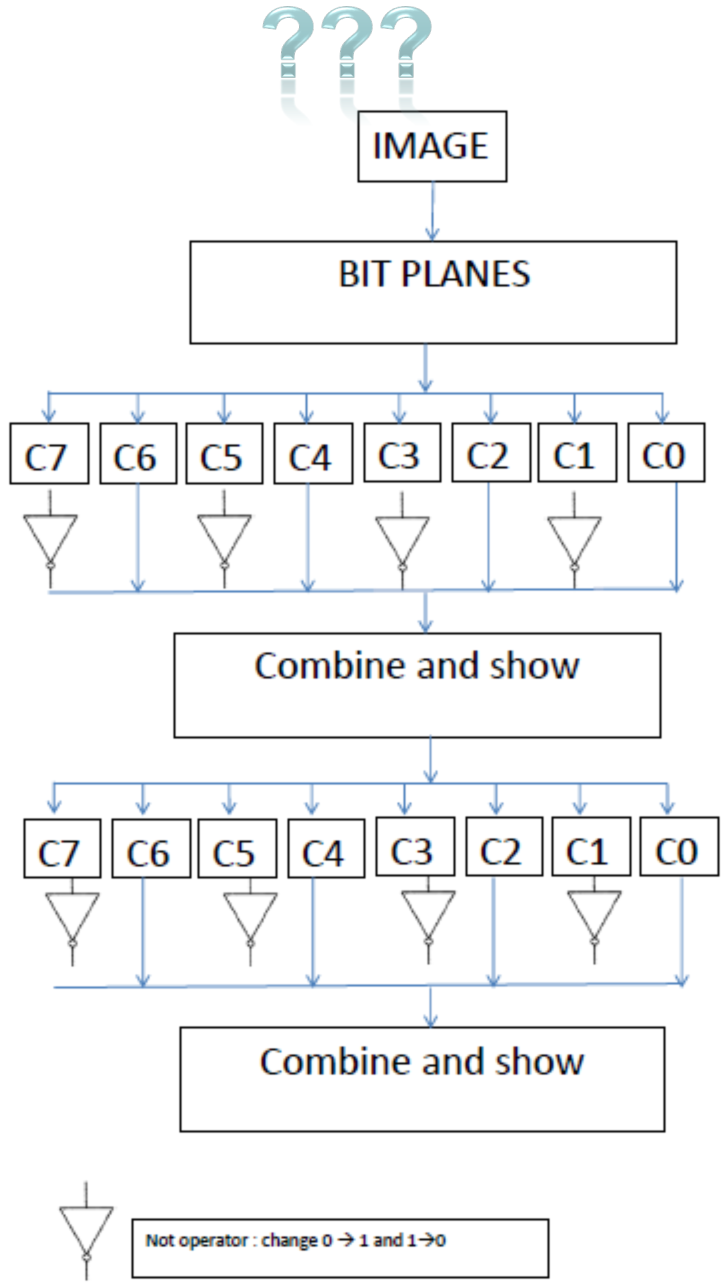
Bit planes



100
THE
UNITED STATES
OF AMERICA
AE 57148129 A
E5
AE 57148129 A
100
ONE HUNDRED DOLLARS

100
THE
UNITED STATES
OF AMERICA
AE 57148129 A
E5
AE 57148129 A
100
ONE HUNDRED DOLLARS

100
THE
UNITED STATES
OF AMERICA
AE 57148129 A
E5
AE 57148129 A
100
ONE HUNDRED DOLLARS


???
IMAGE
BIT PLANES
C7
C6
C5
C4
C3
C2
C1
C0
Combine and show
C7
C6
C5
C4
C3
C2
C1
C0
Combine and show
Not operator : change 0 → 1 and 1→0

Example: Fixed Intensity Transformation

- A 4x4, 4bits/pixel image

1	8	6	6
6	3	11	8
8	8	9	10
9	10	10	7

passes through an intensity transformation $s = T(r) = round\left(\frac{1}{15} r^2\right)$

1 → round(0.0667) = 0;

3 → round(0.6) = 1;

6 → round(2.4) = 2;

7 → round(3.2667) = 3;

8 → round(4.2667) = 4;

9 → round(5.4) =5;

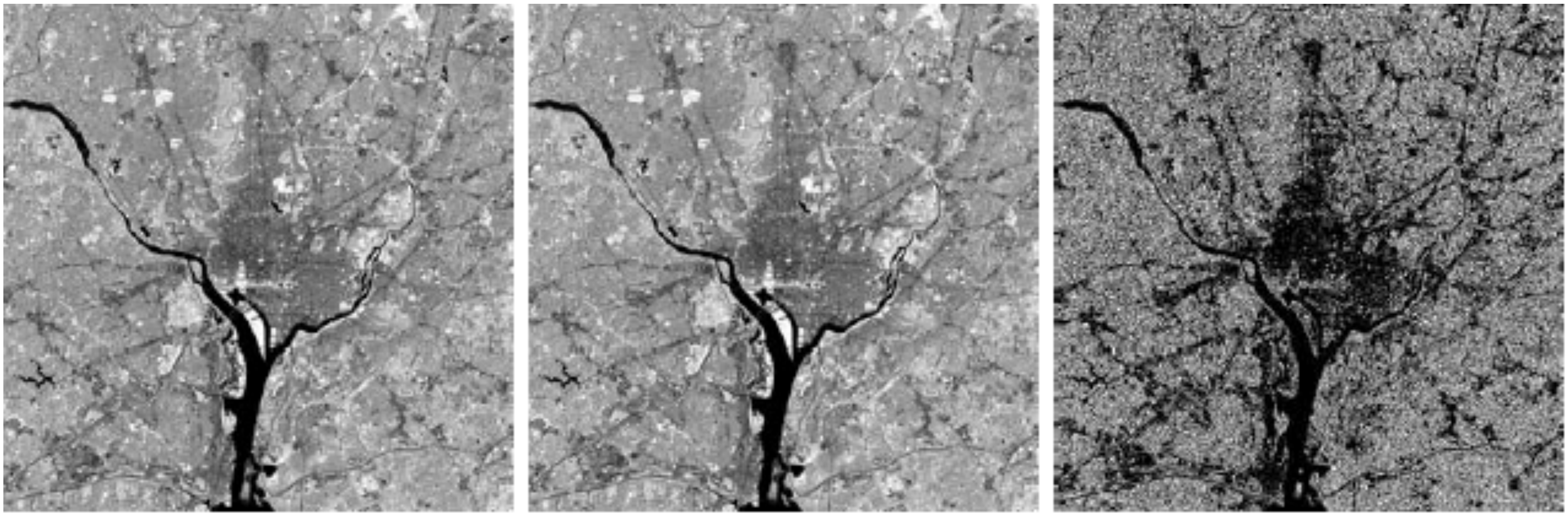
10 → round(6.6667) = 7;

11 → round(8.0667) = 8;

The resulting image is:

0	4	2	2
2	1	8	4
4	4	5	7
5	7	7	3

Image Subtraction

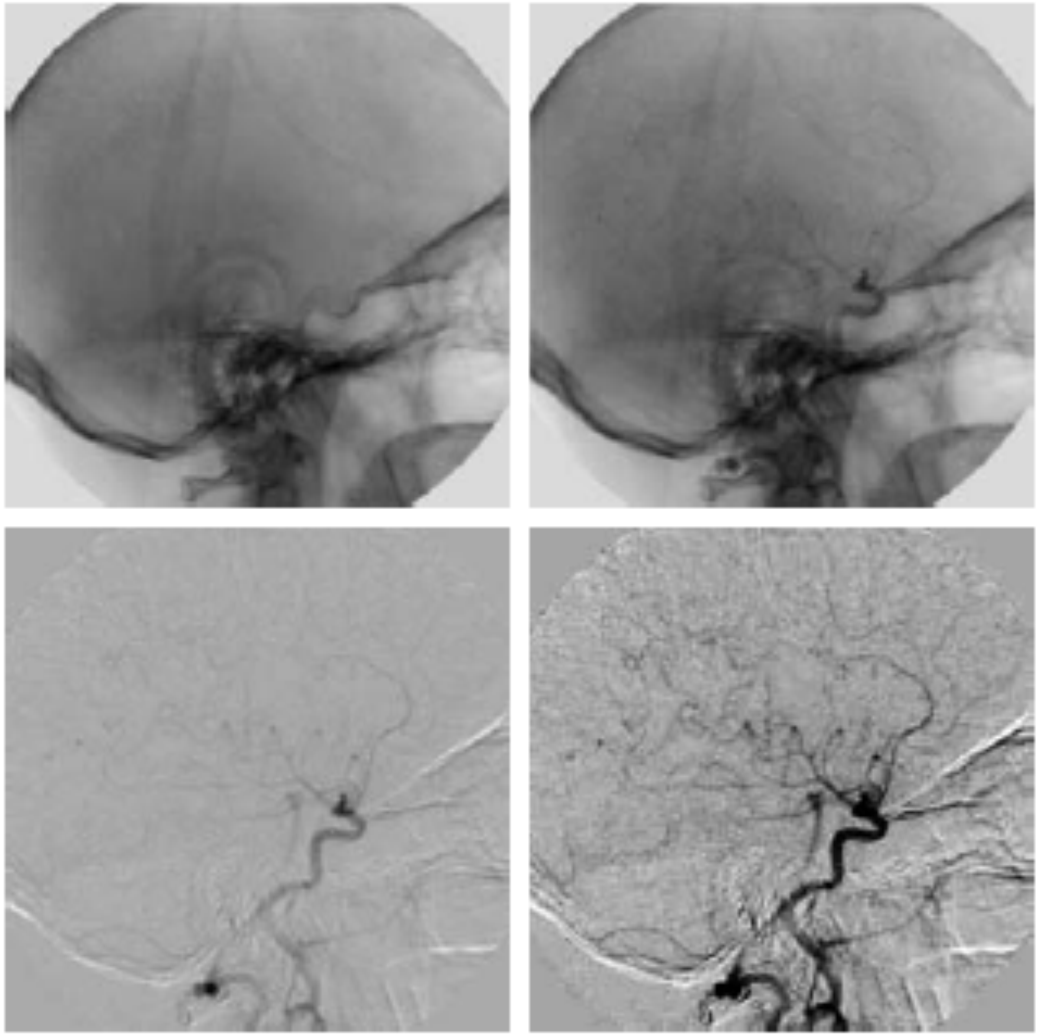


a b c

FIGURE 2.27 (a) Infrared image of the Washington, D.C. area. (b) Image obtained by setting to zero the least significant bit of every pixel in (a). (c) Difference of the two images, scaled to the range [0, 255] for clarity.

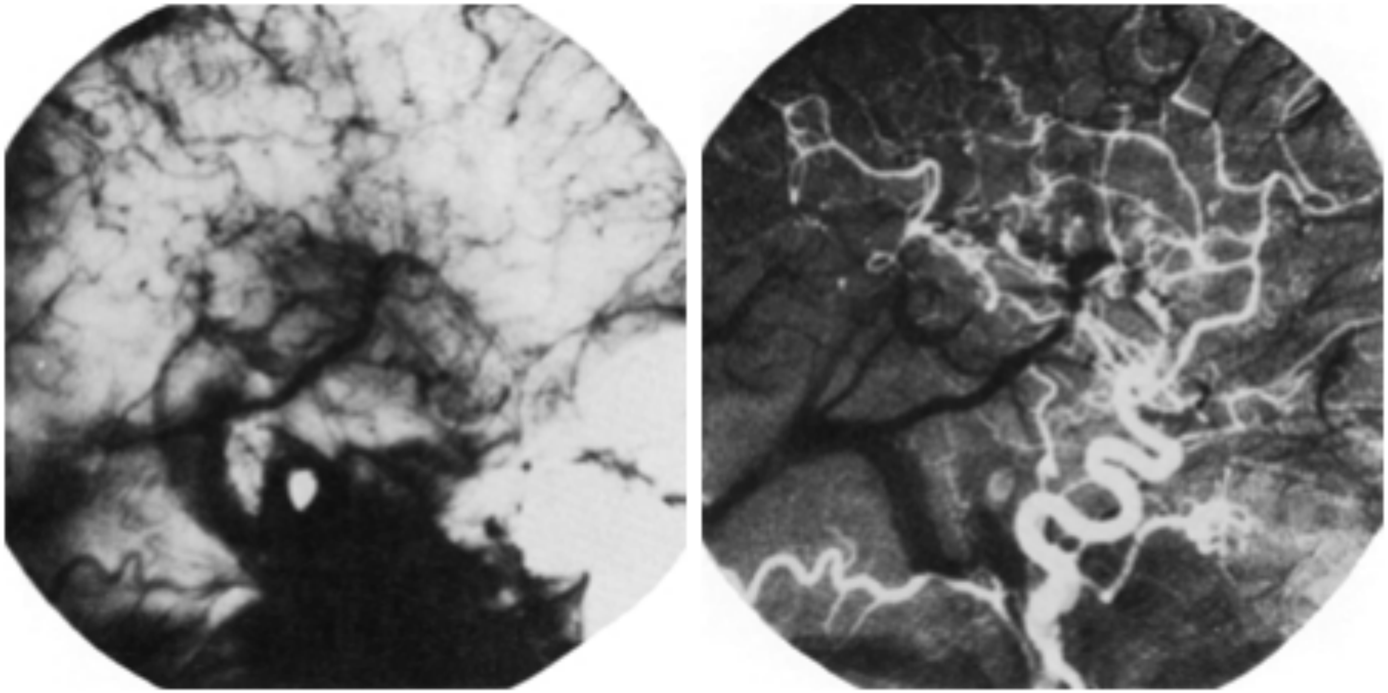
Image Enhancement Subtraction

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



a b
c d

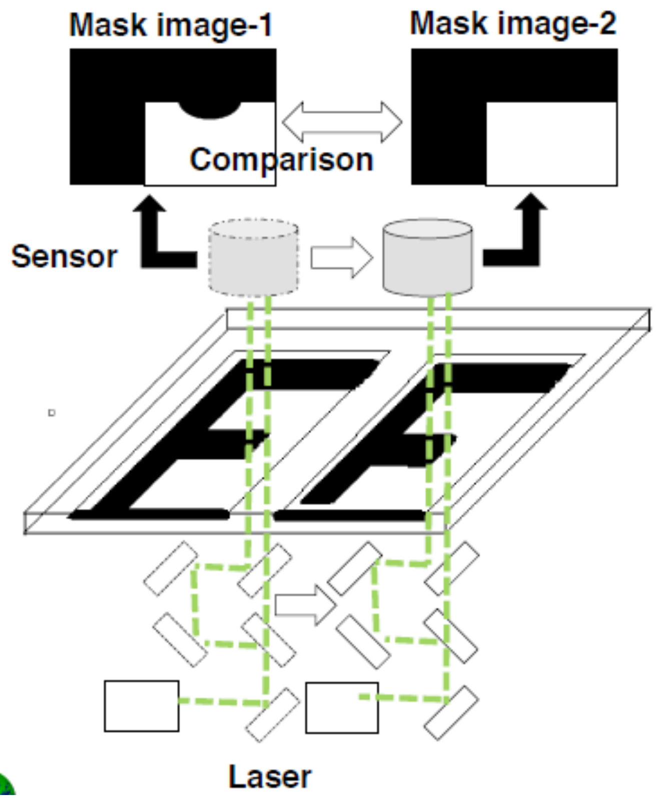
FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

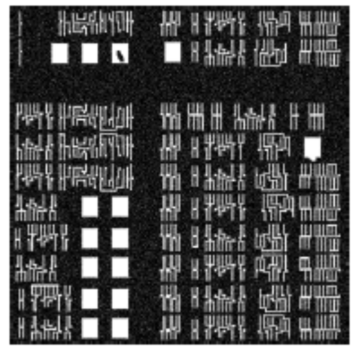


a b

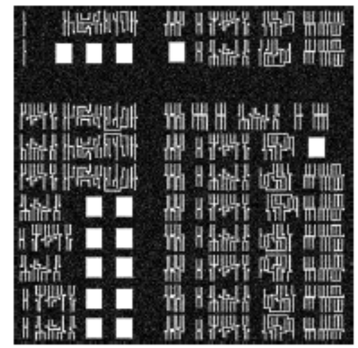
FIGURE 3.29 Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

Image subtraction in IC manufacturing: inspection of photomasks

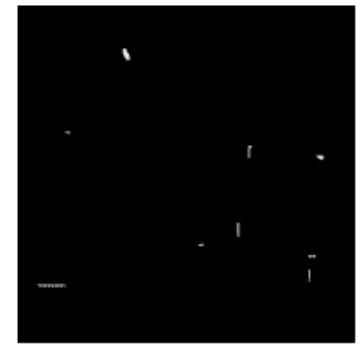




Mask image-1



Mask image-2



Difference image

Where is the defect?

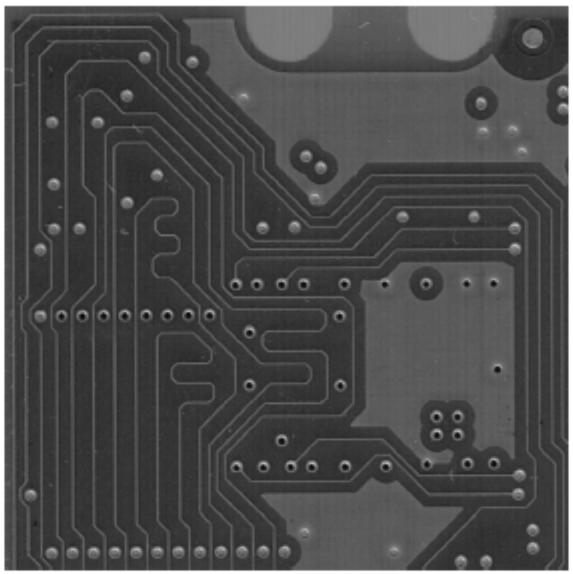


Image $g[x,y]$ (no defect)

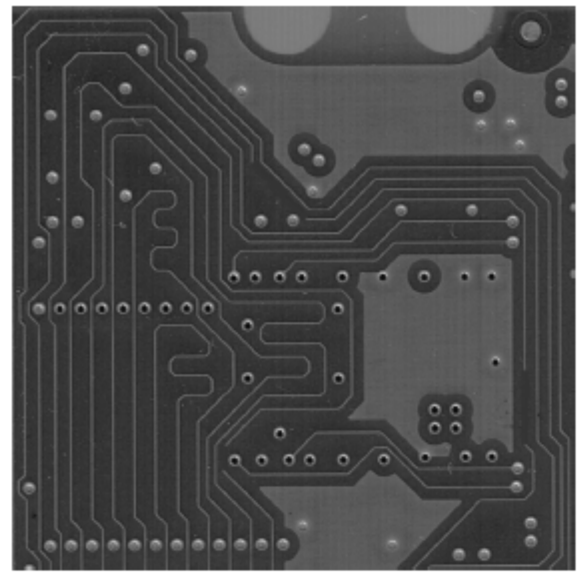
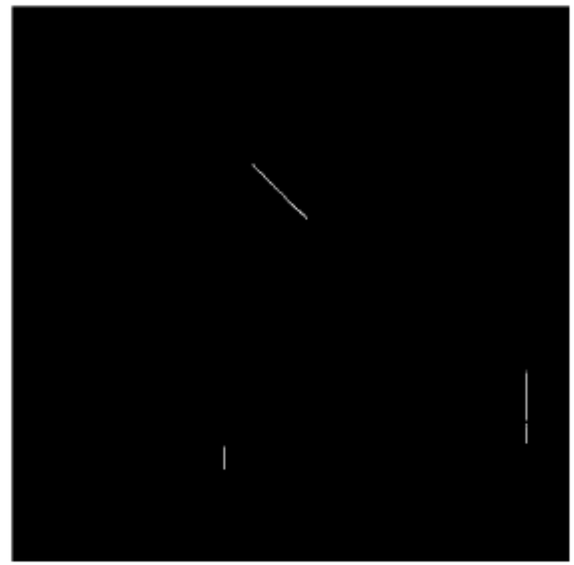


Image $f[x,y]$ (w/ defect)

Absolute difference between two images

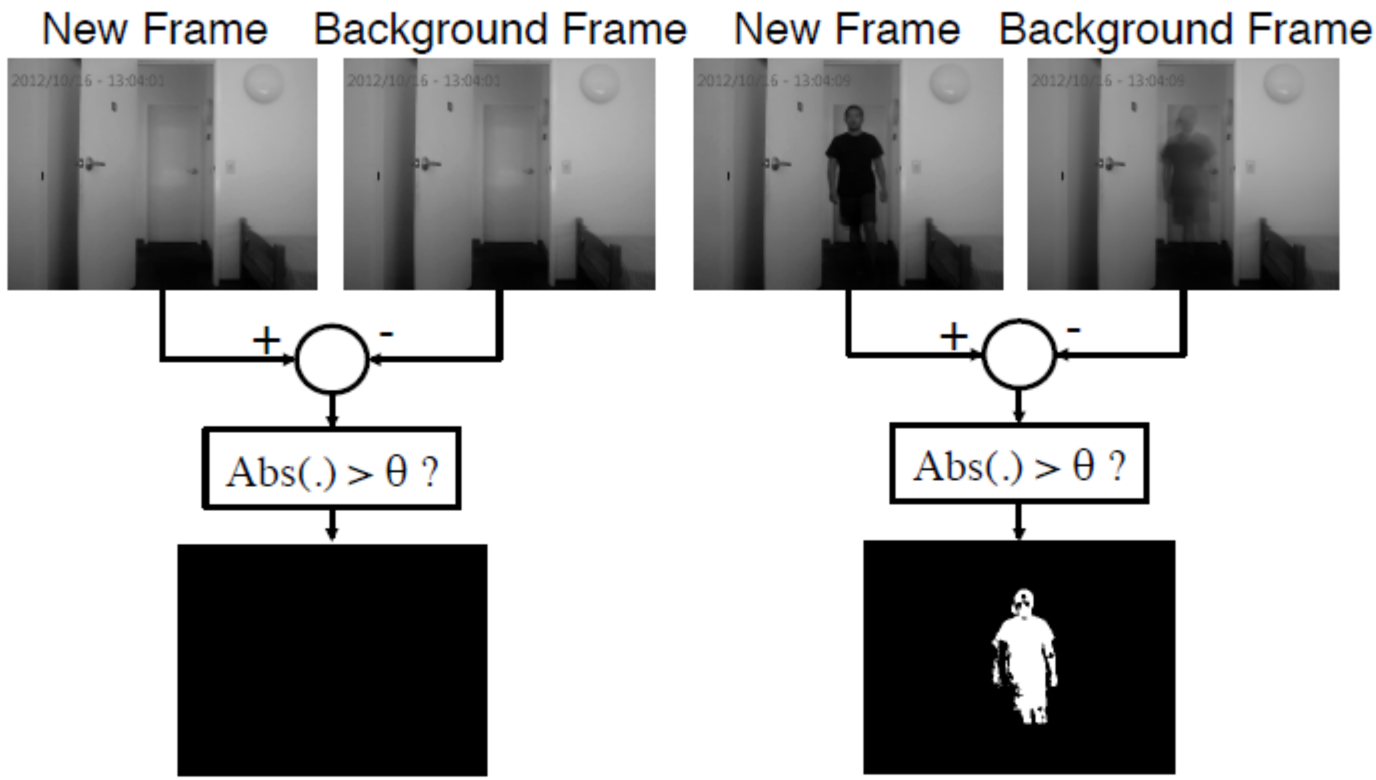
$|f\text{-}g|$ w/o alignment



$|f\text{-}g|$ w/ alignment

MATLAB EXAMPLE: DEFECT DETECTİON

Video background subtraction



Update:

Background[t] := α Background[t-1] + (1- α) New[t]

Image Averaging

A noisy image

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

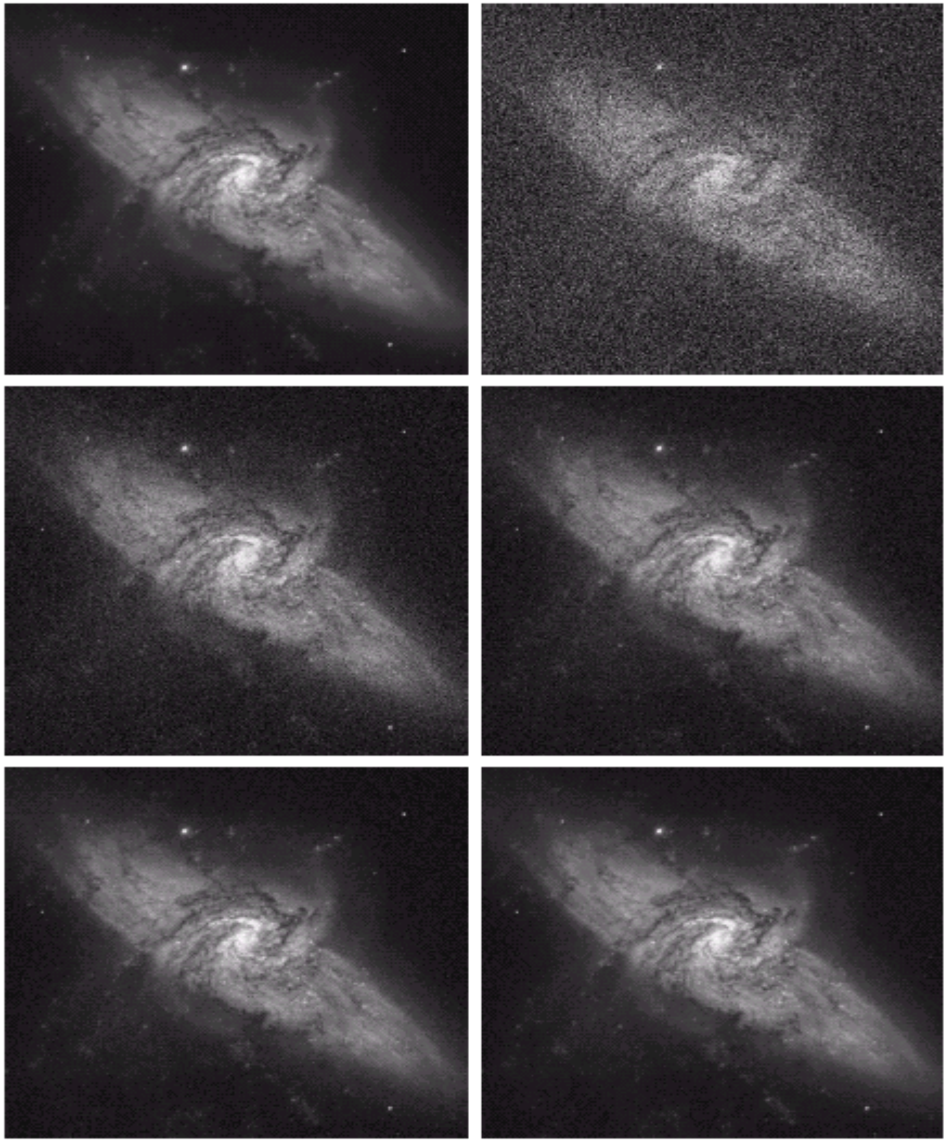
Averaging M different noisy images:

$$\bar{g}(x,y) = \frac{1}{M}\sum_{i=1}^{M} g_i(x,y)$$

As M increases, the variability of the pixel values at each location decreases.

This means that g(x,y) approaches f(x,y) as the number of noisy images used in the averaging process increases.

Registering of the images is necessary to avoid blurring in the output image.



a b
c d
e f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64$, and 128 noisy images. (Original image courtesy of NASA.)

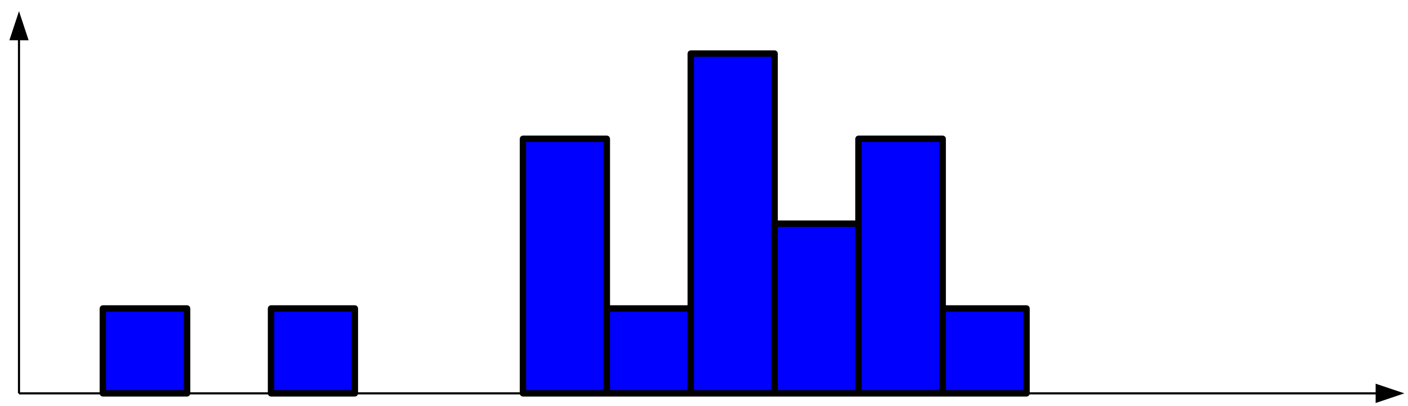
HISTOGRAMS

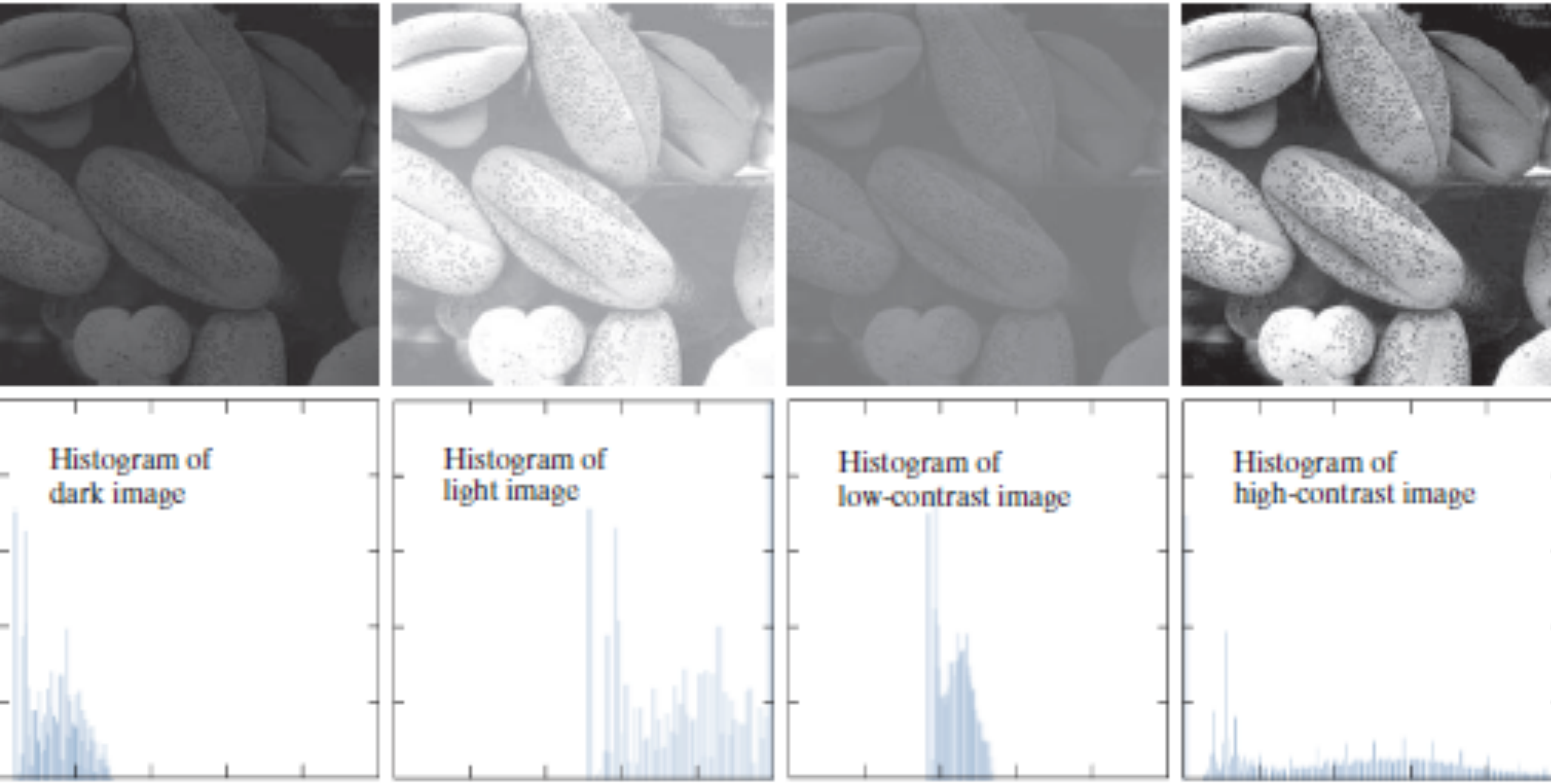
given a grayscale image, its histograms consists of the histogram of its grey levels, that is, a graph indicating the number of times each grey level occurs in the image.

- **Example** a 4x4, 4bits/pixel image

1	8	6	6
6	3	11	8
8	8	9	10
9	10	10	7

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H(k)	0	1	0	1	0	0	3	1	4	2	3	1	0	0	0	0



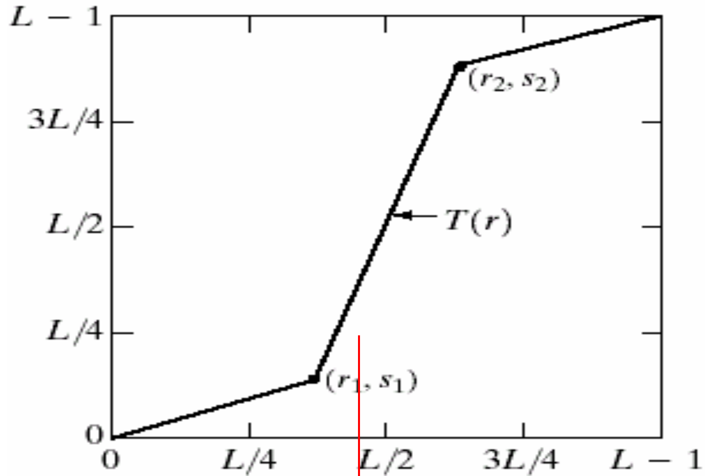


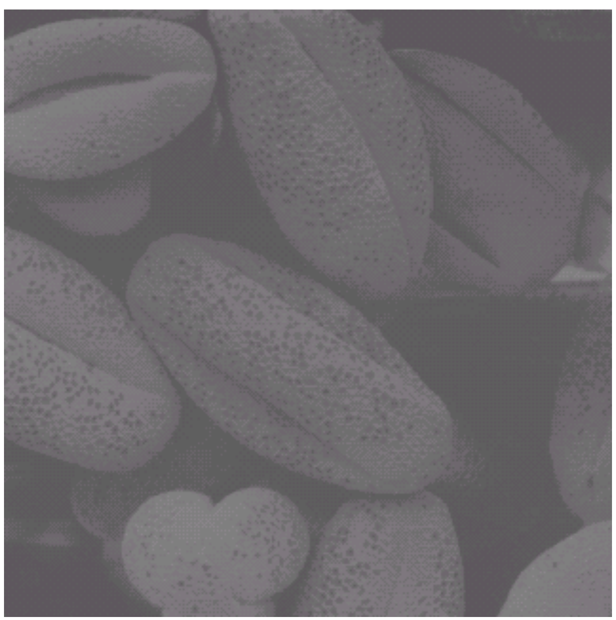
a b c d

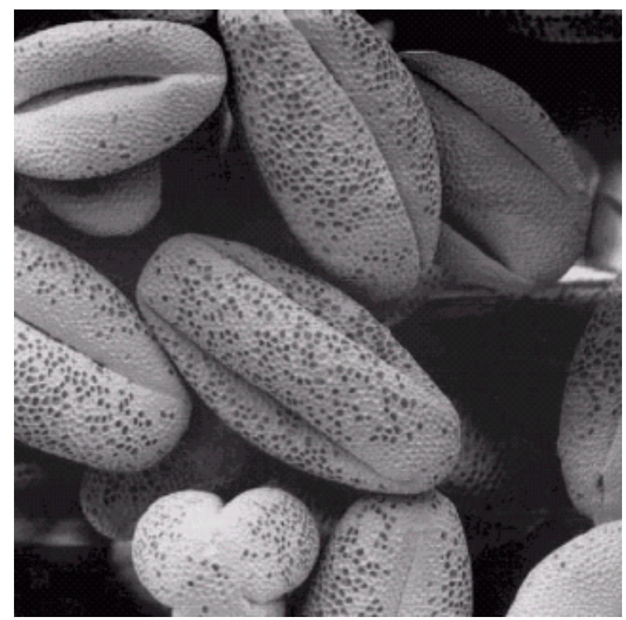
FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

Contrast Stretch

General Idea: Make Best Use of the Dynamic Range







From [Gonzalez & Woods]

Contrast Stretch

General form:

$$s = \begin{cases} \dfrac{s_1}{r_1} \cdot r & 0 \le r < r_1 \\ \dfrac{s_2 - s_1}{r_2 - r_1} \cdot r + \dfrac{s_1 r_2 - s_2 r_1}{r_2 - r_1} & r_1 \le r \le r_2 \\ \dfrac{2^B - 1 - s_2}{2^B - 1 - r_2} \cdot r + (2^B - 1) \cdot \dfrac{s_2 - r_2}{2^B - 1 - r_2} & r_2 < r \le 2^B - 1 \end{cases}$$

Special case → Full-scale contrast stretch:

$$\begin{matrix} r_1 = r_{\min} & s_1 = 0 \\ r_2 = r_{\max} & s_2 = 2^B - 1 \end{matrix} \longrightarrow s = (2^B - 1) \cdot \frac{r - r_{\min}}{r_{\max} - r_{\min}}$$

Typically used: $s = round\left((2^B - 1) \cdot \dfrac{r - r_{\min}}{r_{\max} - r_{\min}} \right)$

Example: Full-Scale Contrast Stretch

- Full-scale contrast stretch of a 4x4, 4bits/pixel image

4	8	6	6
6	4	11	8
8	8	9	10
8	11	10	7

- Find $r_{\min} = 4$ $r_{\max} = 11$ $2^B - 1 = 15$

$$s = round\left((2^B - 1)\cdot\frac{r - r_{\min}}{r_{\max} - r_{\min}}\right) = round\left(15\cdot\frac{r-4}{11-4}\right) = round\left(\frac{15}{7}(r-4)\right)$$

4 → round(0) = 0;

6 → round(4.29) = 4;

7 → round(6.43) = 6;

8 → round(8.57) = 9;

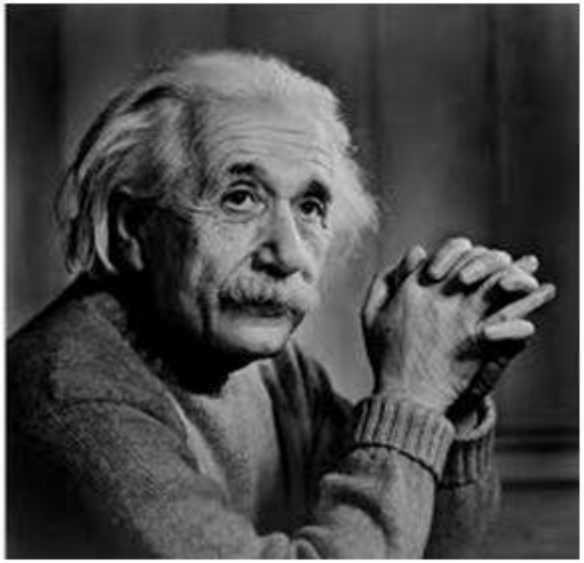
9 → round(10.71) = 11;

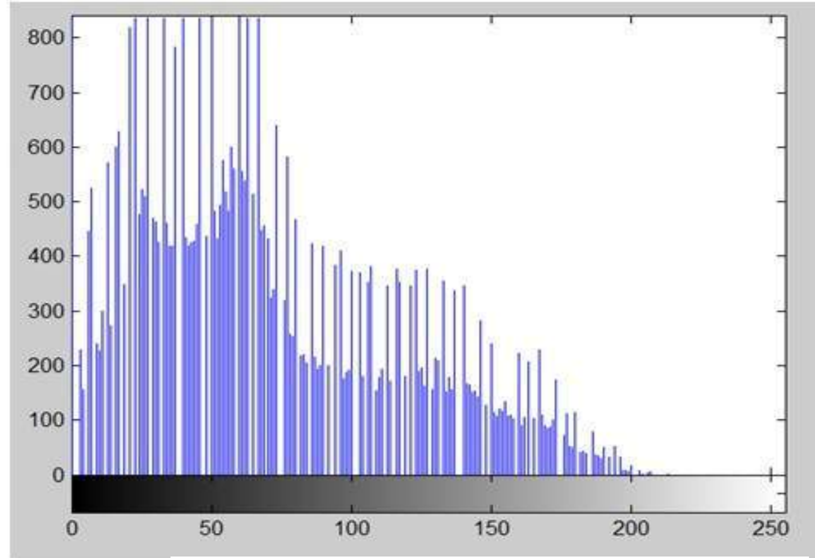
10 → round(12.86) = 13;

11 → round(15) = 15;

The resulting image is:

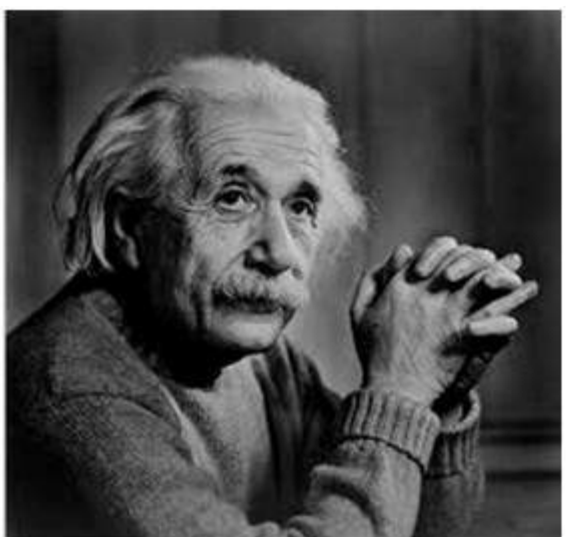
0	9	4	4
4	0	15	9
9	9	11	13
9	15	13	6

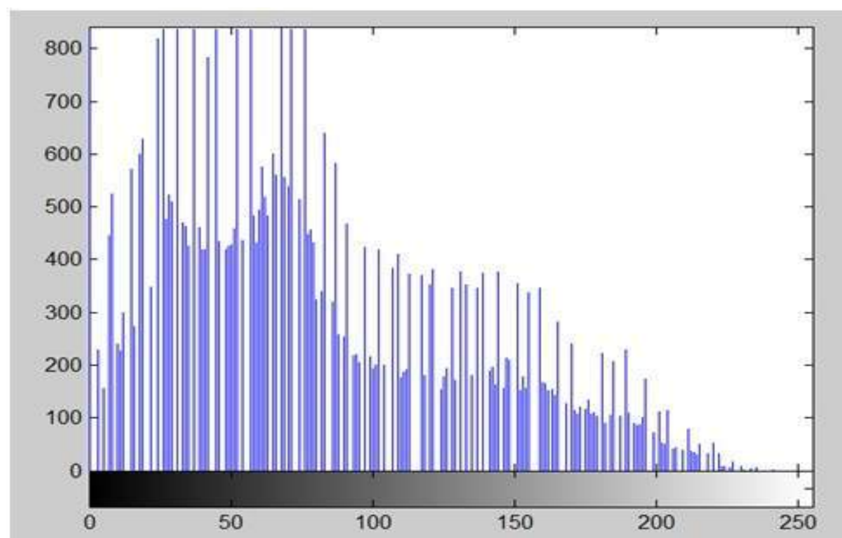




$$g(x,y) = \frac{f(x,y)-fmin}{fmax-fmin} * 2^{bpp}$$

$$g(x,y) = \frac{f(x,y)-0}{225-0} * 255$$





$$g(x,y) = \frac{f(x,y) - 0}{255 - 0} * 255$$

$$g(x,y) = \frac{f(x,y)}{255} * 255$$

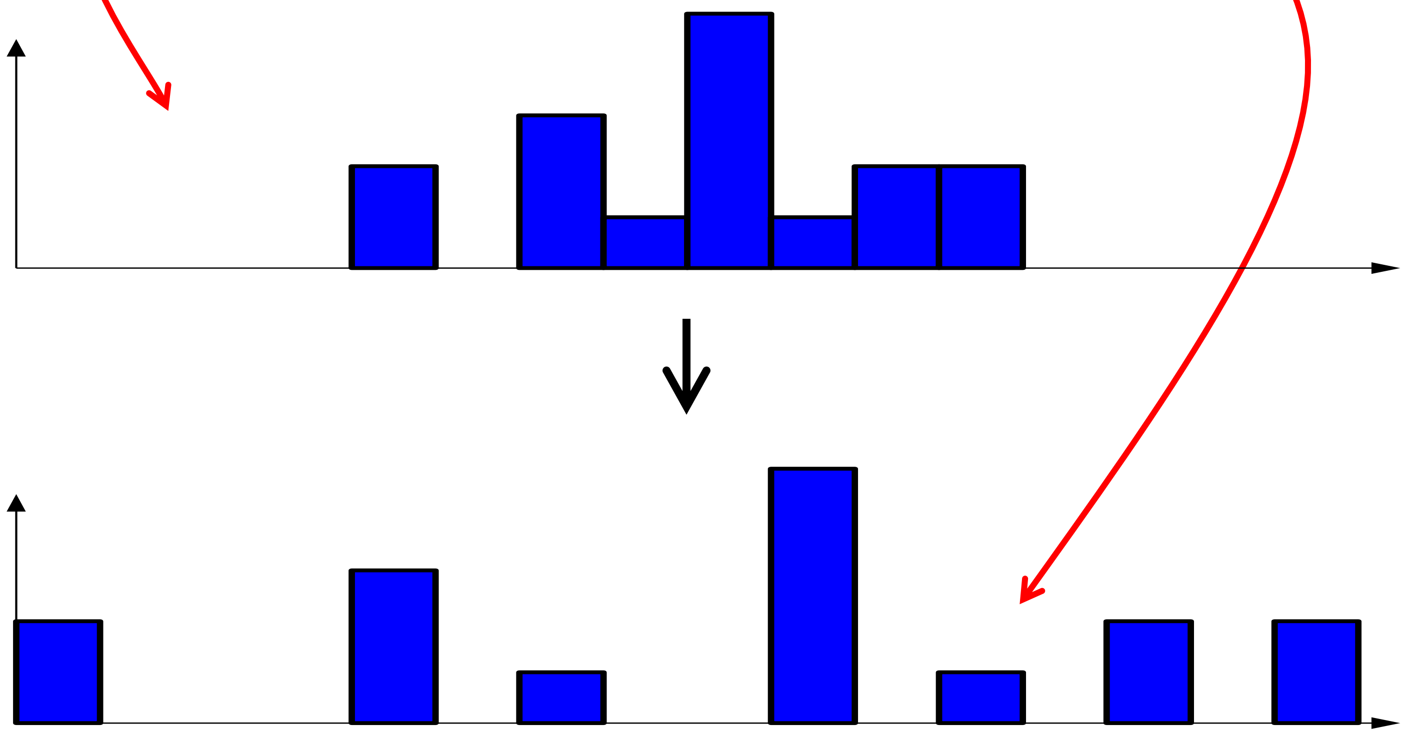
$$g(x,y) = f(x,y)$$

Example: Histogram Change

4	8	6	6
6	4	11	8
8	8	9	10
8	11	10	7

→

0	9	4	4
4	0	15	9
9	9	11	13
9	15	13	6



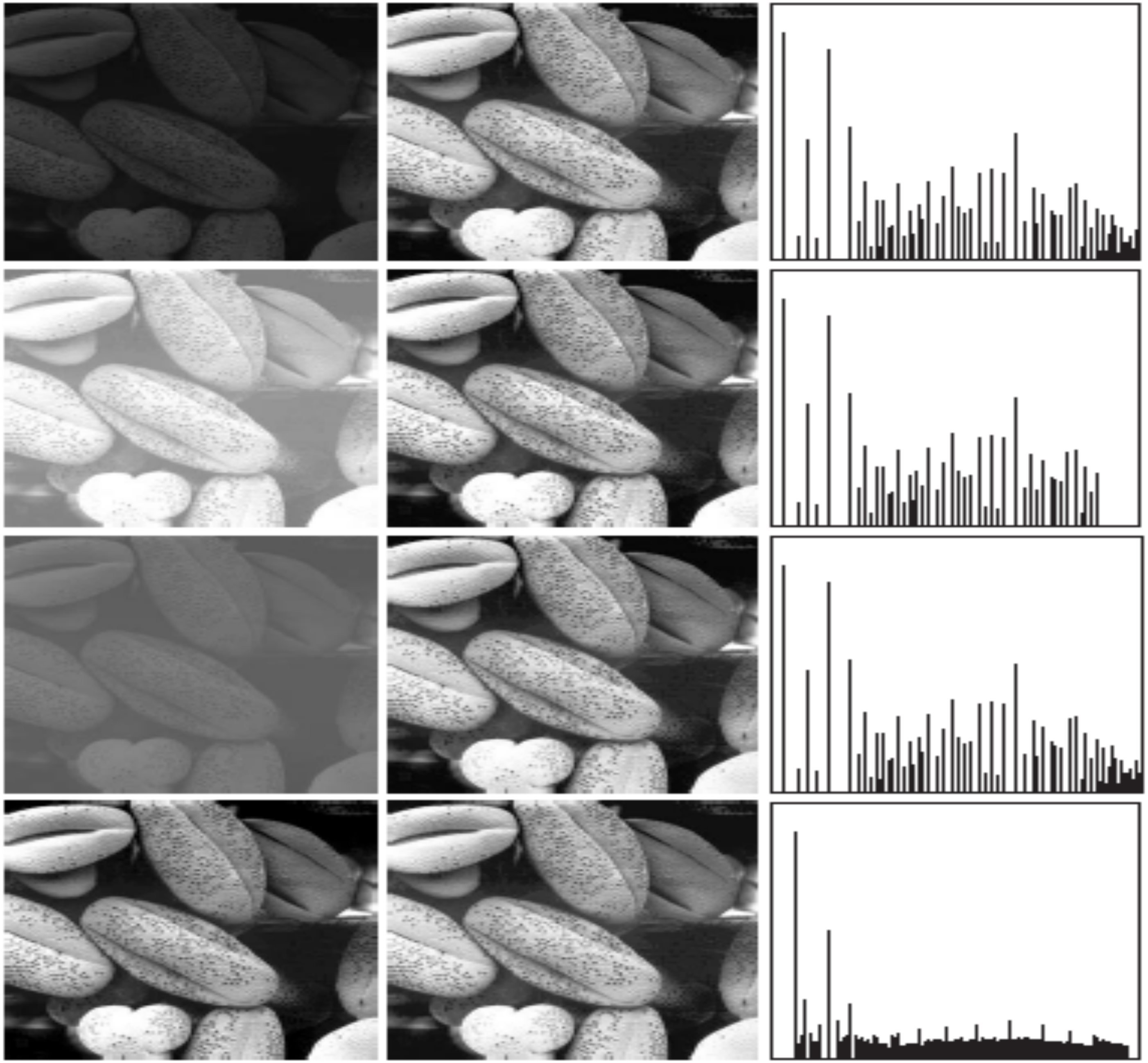
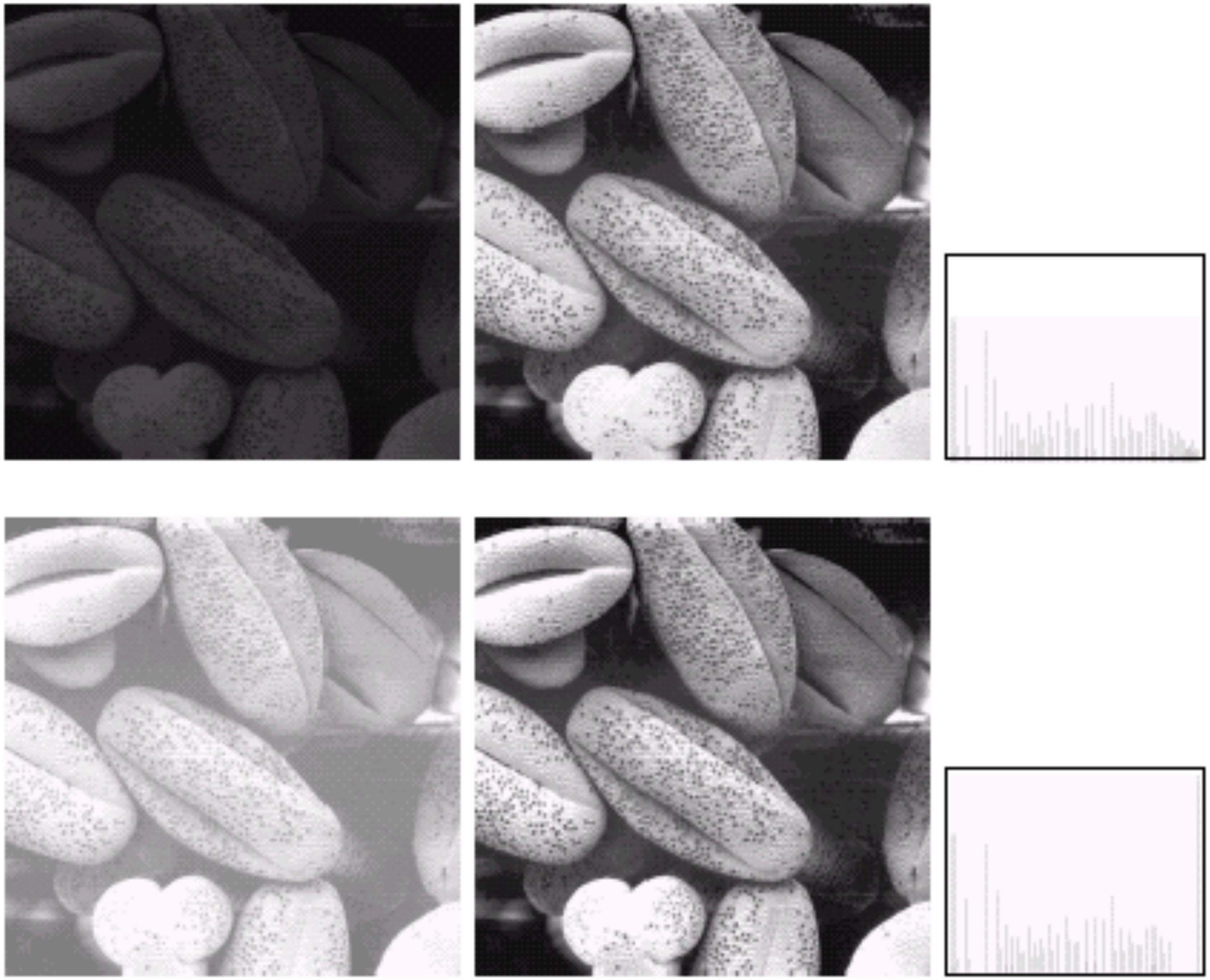


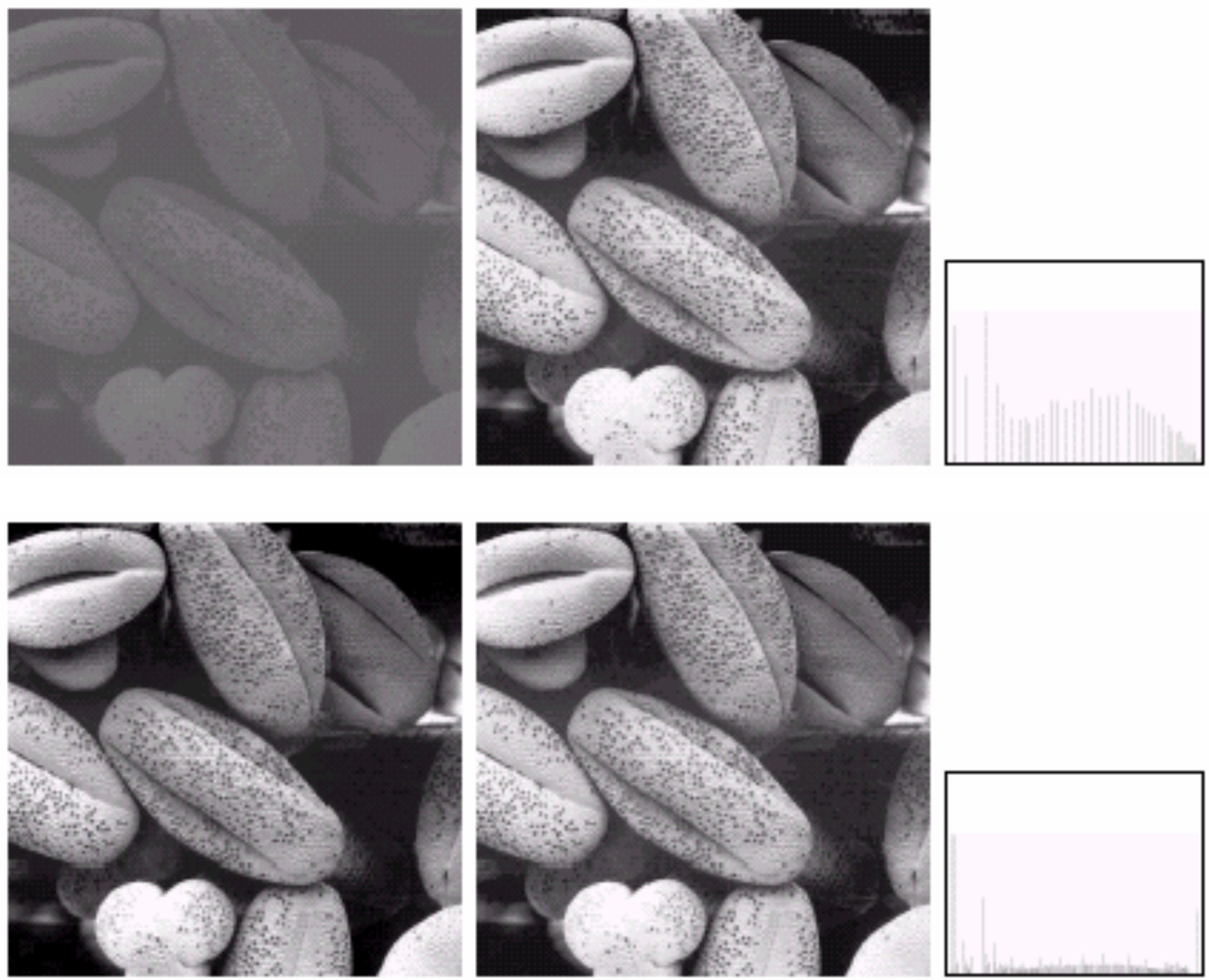
FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

Histogram Equalization



From [Gonzalez & Woods]

Histogram Equalization



From [Gonzalez & Woods]

Example

- A 4x4, 4bits/pixel image

2	8	9	9
2	3	10	9
8	3	3	11
8	3	10	11

- First try: full-scale contrast stretch $r_{\min} = 2$ $r_{\max} = 11$

$$s = round\left((2^B - 1) \cdot \frac{r - r_{\min}}{r_{\max} - r_{\min}} \right) = round\left(15 \cdot \frac{r-2}{11-2} \right) = round\left(\frac{5}{3}(r-2) \right)$$

2 → round(0) = 0;
3 → round(1.67) = 2;
8 → round(10.00) = 10;
9 → round(11.67) = 12;
10 → round(13.33) = 13;
11 → round(15) = 15;

The resulting image is:

0	10	12	12
0	2	13	12
10	2	2	15
10	2	13	15

Example: Histogram Change

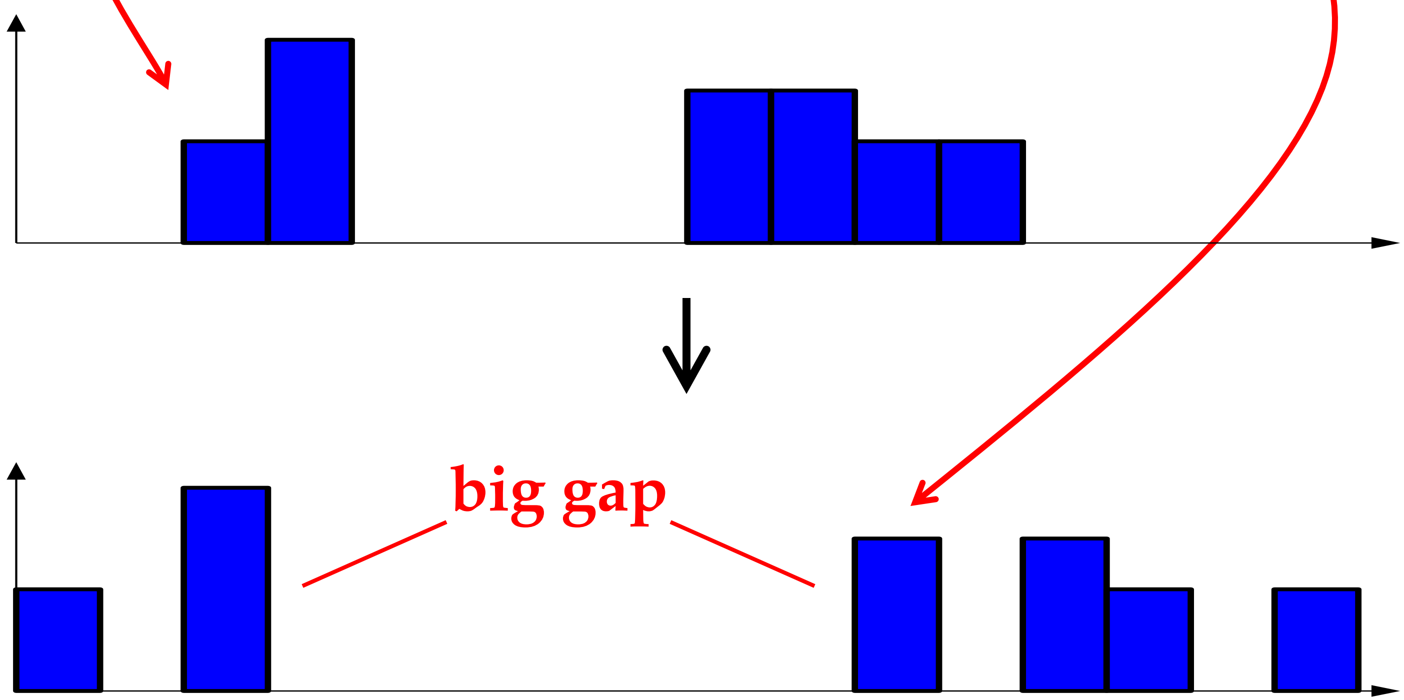
original

2	8	9	9
2	3	10	9
8	3	3	11
8	3	10	11

→

0	10	12	12
0	2	13	12
10	2	2	15
10	2	13	15

full-scale contrast stretch

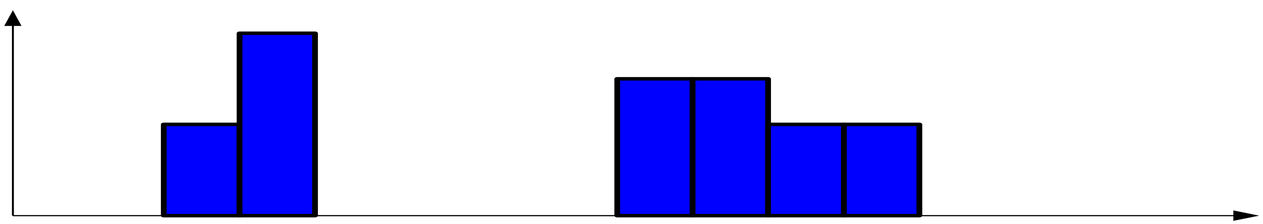


Cumulative Histogram

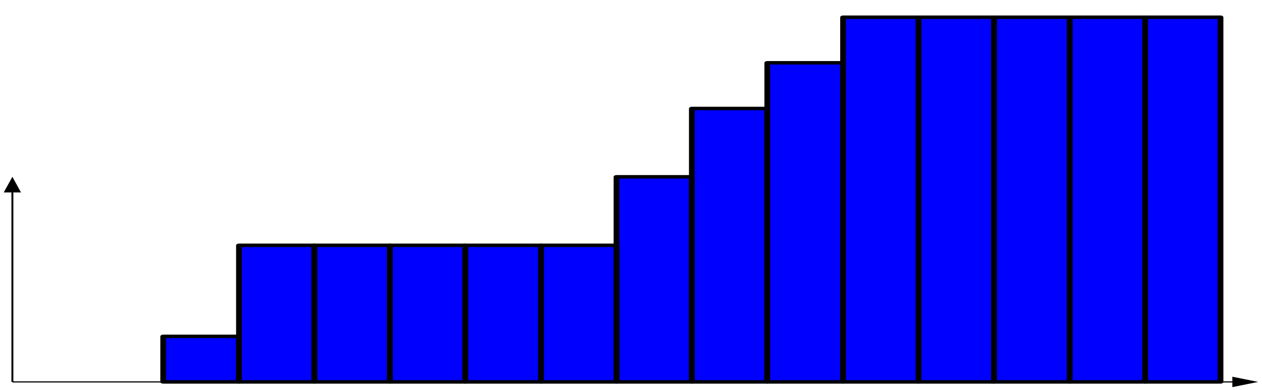
2	8	9	9
2	3	10	9
8	3	3	11
8	3	10	11

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	probability mass function															
H(k)	0	0	2	4	0	0	0	0	3	3	2	2	0	0	0	0
	cumulative distributive function															
Q(k)	0	0	2	6	6	6	6	6	9	12	14	16	16	16	16	16

H(k)



Q(k)



Intermediate Image

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
H(k)	0	0	2	4	0	0	0	0	3	3	2	2	0	0	0	0
Q(k)	0	0	2	6	6	6	6	6	9	12	14	16	16	16	16	16

original

2	8	9	9
2	3	10	9
8	3	3	11
8	3	10	11

→

2	9	12	12
2	6	14	12
9	6	6	16
9	6	14	16

intermediate image

Full-Scale Contrast Stretch of Intermediate Image

intermediate image

2	9	12	12
2	6	14	12
9	6	6	16
9	6	14	16

$$s = round\left((2^B - 1) \cdot \frac{r - r_{\min}}{r_{\max} - r_{\min}} \right) = round\left(15 \cdot \frac{r-2}{16-2} \right) = round\left(\frac{15}{14}(r-2) \right)$$

2 → round(0) = 0;

6 → round(4.29) = 4;

9 → round(7.50) = 8;

12 → round(10.71) = 11;

14 → round(12.86) = 13;

16 → round(15) = 15;

final result:

histogram equalized image

0	8	11	11
0	4	13	11
8	4	4	15
8	4	13	15

Histogram Comparison

4	8	6	6
6	4	11	8
8	8	9	10
8	11	10	7

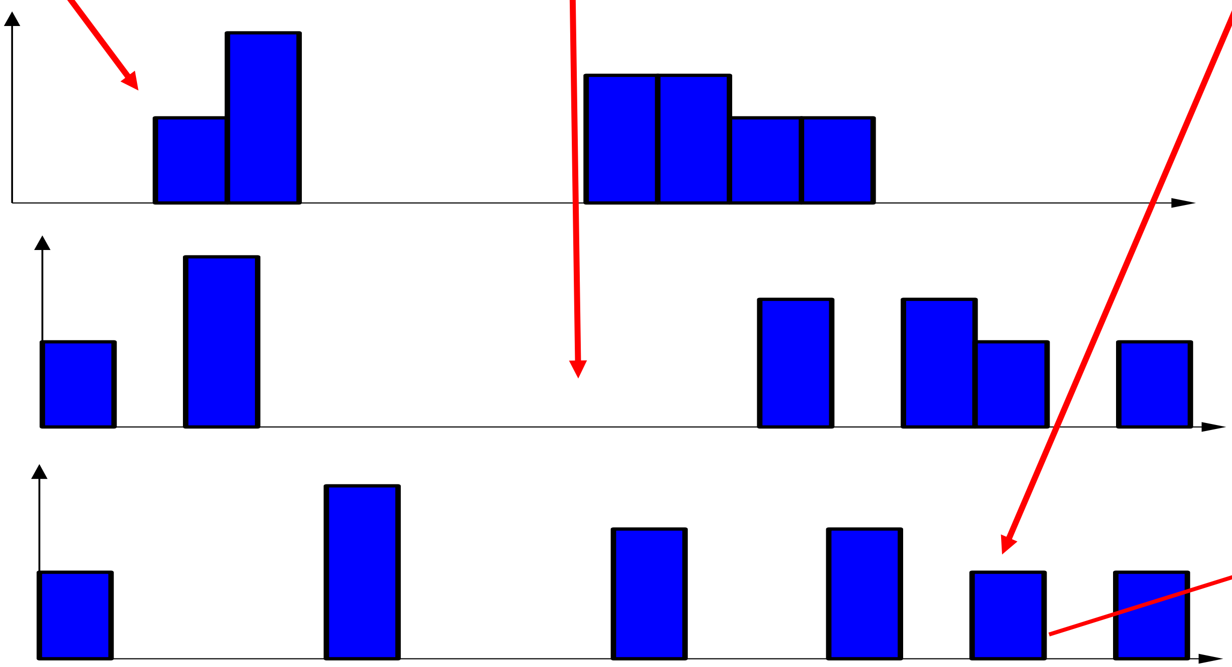
original

0	10	12	12
0	2	13	12
10	2	2	15
10	2	13	15

direct full-scale contrast stretch

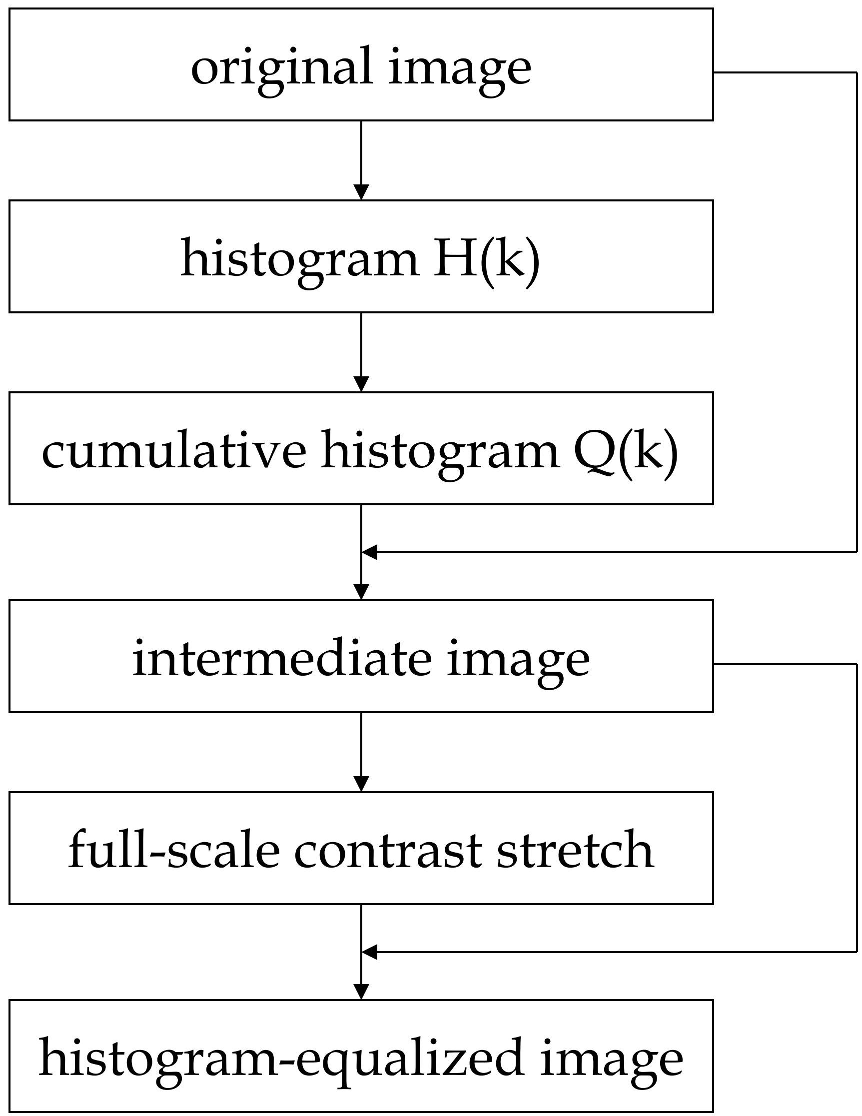
0	8	11	11
0	4	13	11
8	4	4	15
8	4	13	15

histogram-equalized



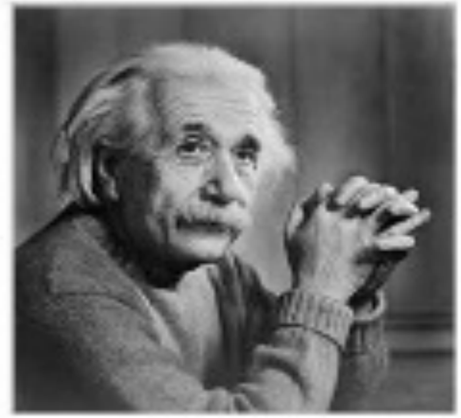
more equalized

Summary of the Histogram Equalization Algorithm

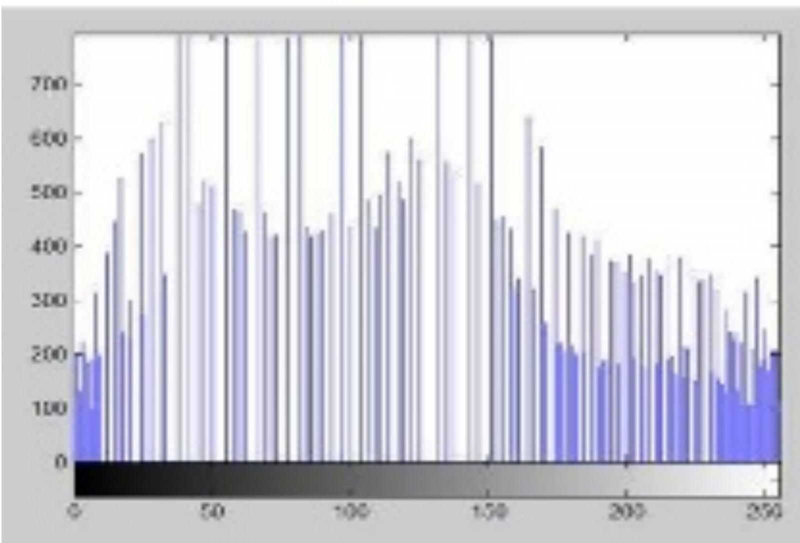


New Image

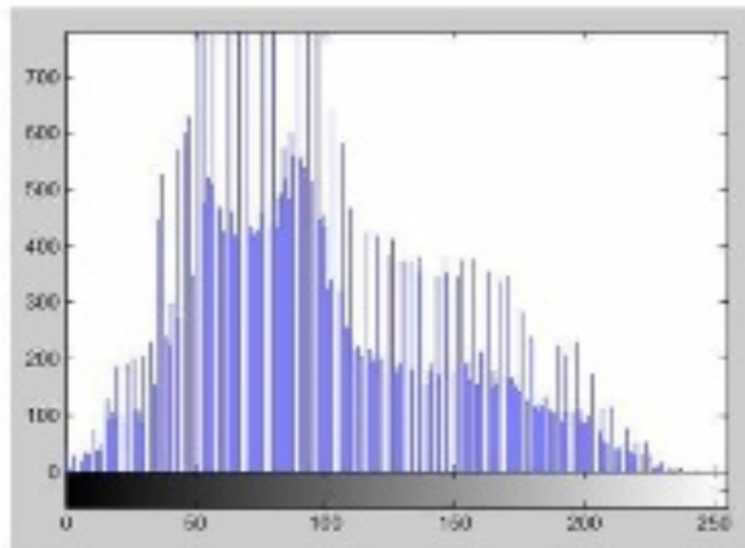
Old image



New Histogram



Old Histogram

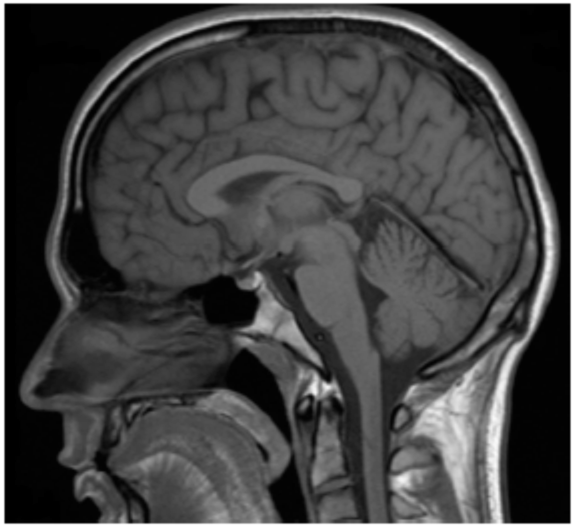


Histogram equalization example

Original image *Bay*

... after histogram equalization

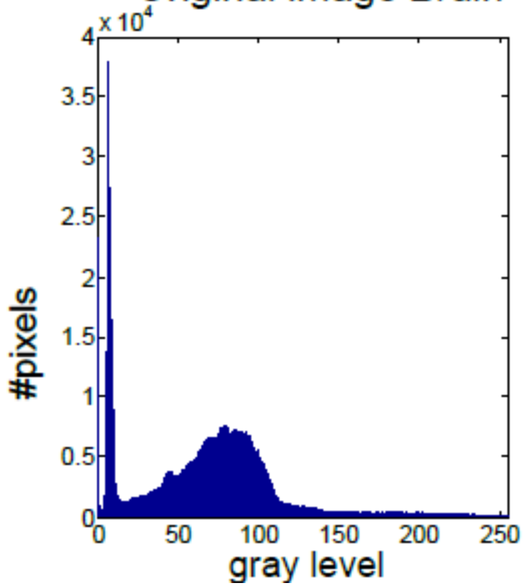
Histogram equalization example



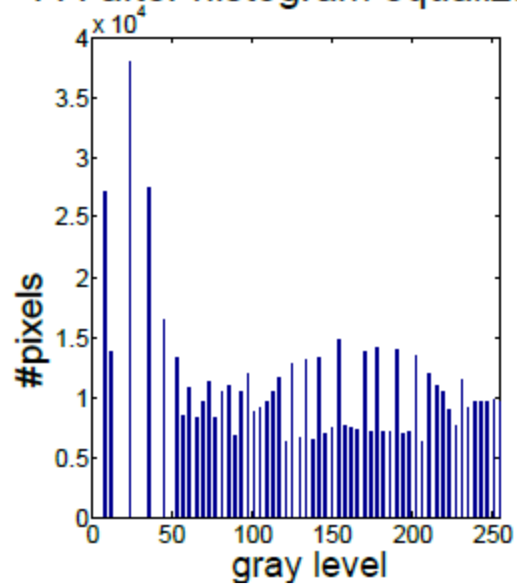
Original image *Brain*

... after histogram equalization

Original image *Brain*



. . . after histogram equalization

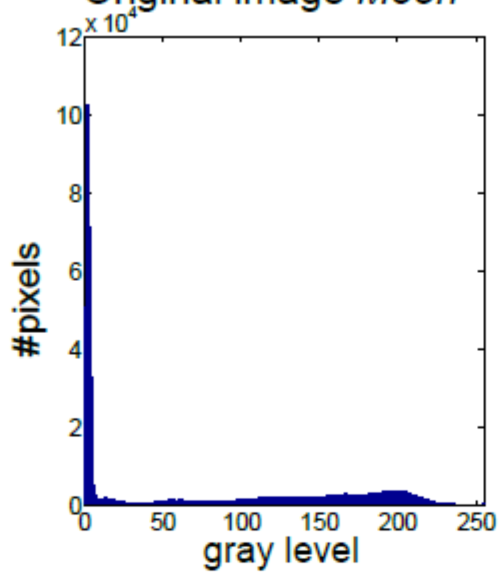


Histogram equalization example

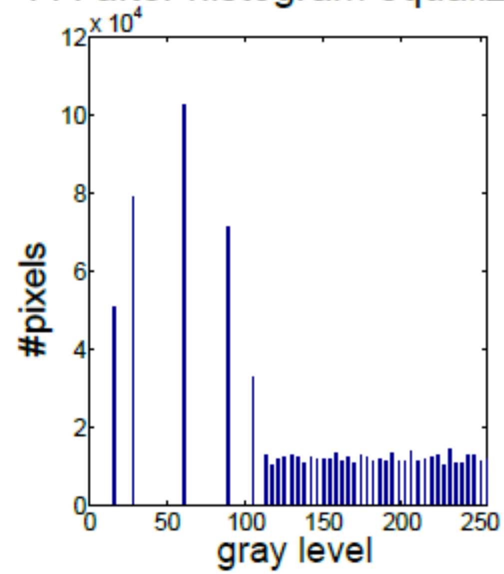
Original image *Moon*

... after histogram equalization

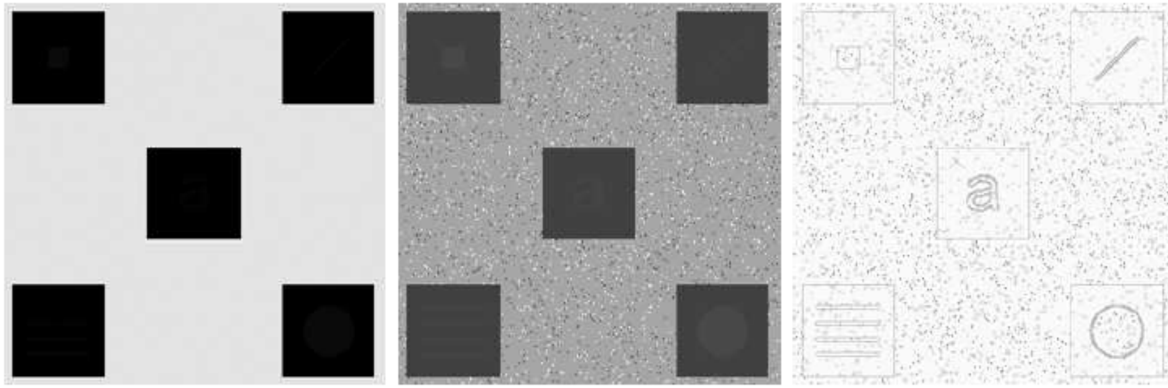
Original image *Moon*



. . . after histogram equalization



Local Histogram Processing

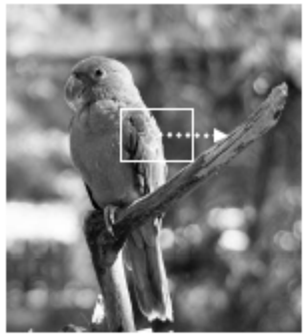


a b c

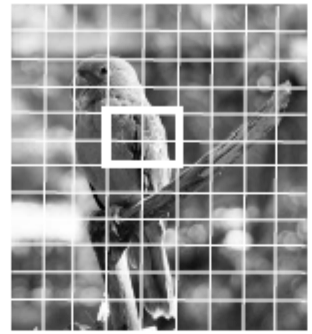
FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3×3.

Adaptive histogram equalization

- Histogram equalization based on a histogram obtained from a portion of the image



Sliding window **approach:**
different histogram (and
mapping) for every pixel



Tiling approach:
subdivide into overlapping
regions, mitigate blocking
effect by smooth blending
between neighboring tiles

Adaptive histogram equalization

Original image
Parrot

Global histogram
equalization

Adaptive histogram
equalization, 8x8 tiles

Adaptive histogram
equalization, 16x16 tiles

Adaptive histogram equalization

Original image
Dental Xray

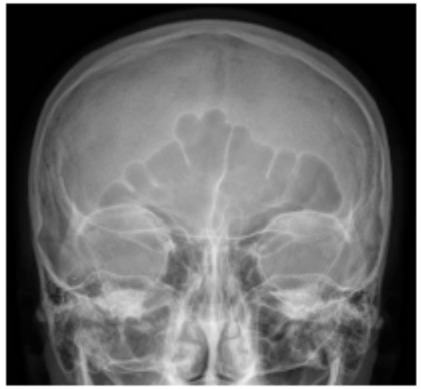
Global histogram
equalization

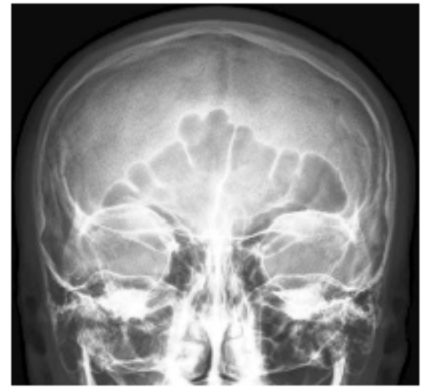
Adaptive histogram
equalization, 8x8 tiles

Adaptive histogram
equalization, 16x16 tiles

Adaptive histogram equalization

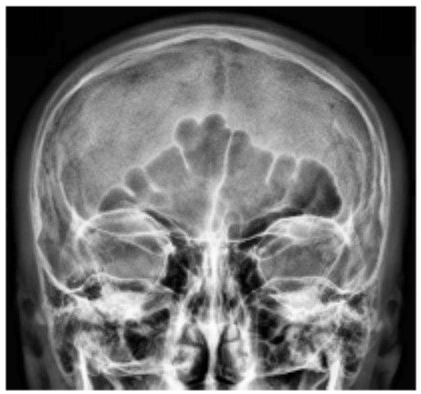
Original image
Skull Xray

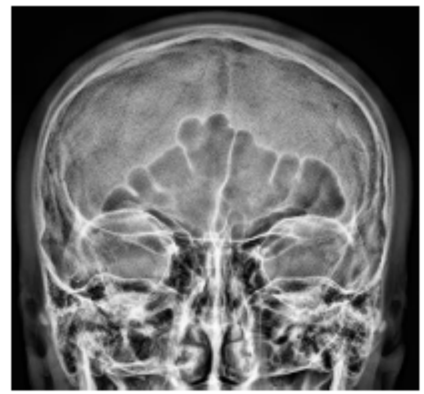




Global histogram
equalization

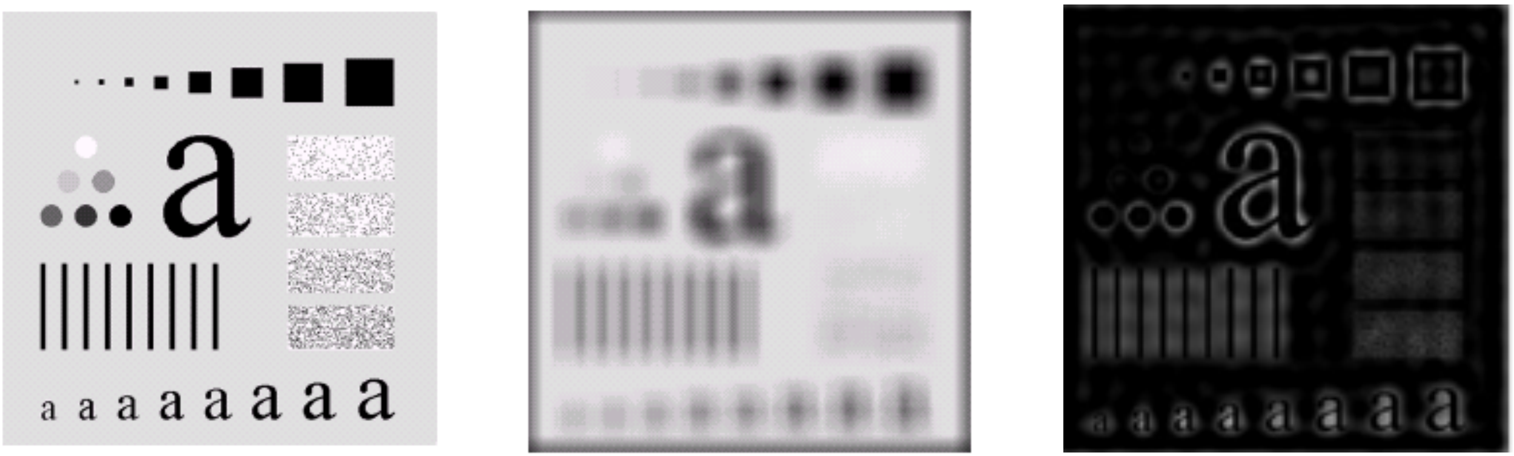
Adaptive histogram
equalization, 8x8 tiles





Adaptive histogram
equalization, 16x16 tiles

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



Linear image processing

- Image processing system $S(.)$ is linear, iff superposition principle holds:

$$S\left(\alpha \cdot f\left[x,y\right] + \beta \cdot g\left[x,y\right]\right) = \alpha \cdot S\left(f\left[x,y\right]\right) + \beta \cdot S\left(g\left[x,y\right]\right) \text{ for all } \alpha, \beta \in \mathbb{R}$$

- Any linear image processing system can be written as

$$\vec{g} = H\vec{f}$$

Note: matrix H need not be square.

by sorting pixels into a column vector

$$\vec{f} = \begin{pmatrix} f\left[0,0\right] & f\left[1,0\right] & \cdots & f\left[N-1,0\right] & f\left[0,1\right] & \cdots & f\left[N-1,1\right] & \cdots & \cdots & f\left[0,L-1\right] & \cdots & f\left[N-1,L-1\right] \end{pmatrix}^T$$