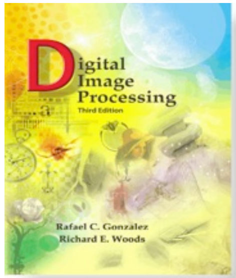
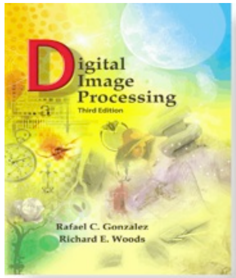


3. Intensity Transformations and Spatial Filtering

1. Background
2. Some Basic Intensity Transformation Functions
3. Histogram Processing
4. Fundamentals of Spatial Filtering
5. Smoothing Spatial Filters
6. Sharpening Spatial Filters



3.1 Background

3.1.1 The Basics of Intensity Transformations and Spatial Filtering

Spatial domain process: $g(x, y) = T\left[f(x, y)\right]$

Output image

Operator defined over a neighbourhood of point (x,y)

Input image

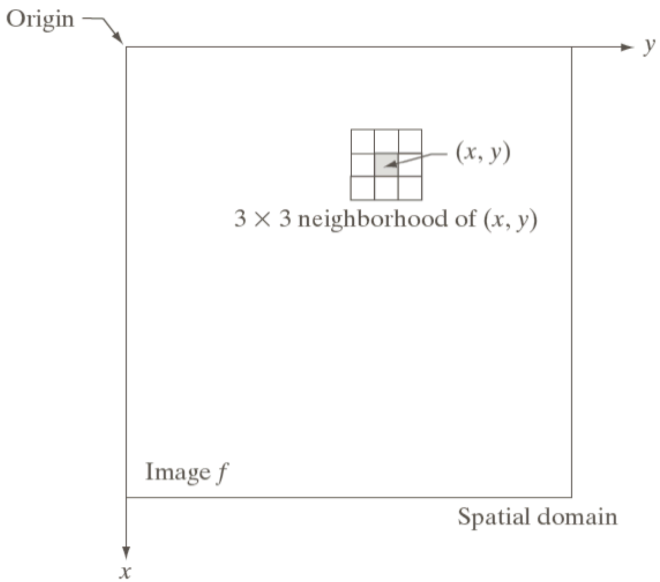
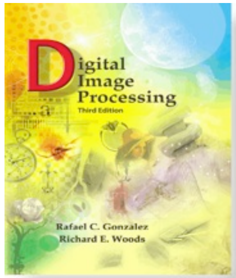


FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.



Smallest possible neighbourhood: size 1x1

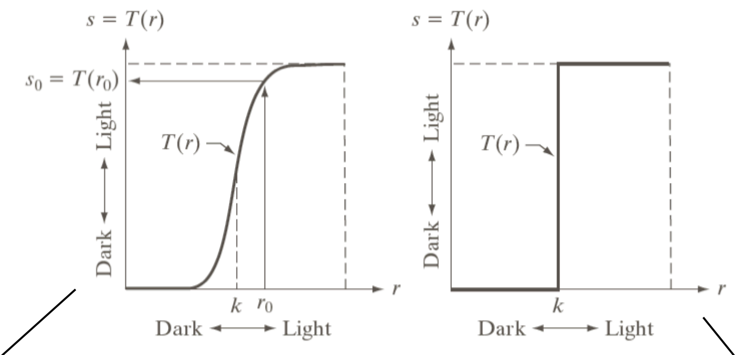
$\Rightarrow$ T becomes an *intensity* (or *gray-level*, or *mapping*) *transformation function*

$$s = T(r)$$

where s (resp. r) = intensity of g (resp. f) at (x,y)

<u>Examples</u>:

Contrast stretching　　Thresholding

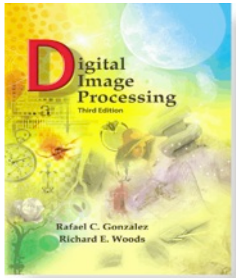


a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

Higher contrast

Two-level (binary) image



3.2 Some Basic Intensity Transformation Functions

Values of the pixels will be noted: *r before* processing, *s after* : $s = T(r)$

Digital => values of T typically stored in a one-dimension array + *mappings* from *r* to *s* implemented via table lookups

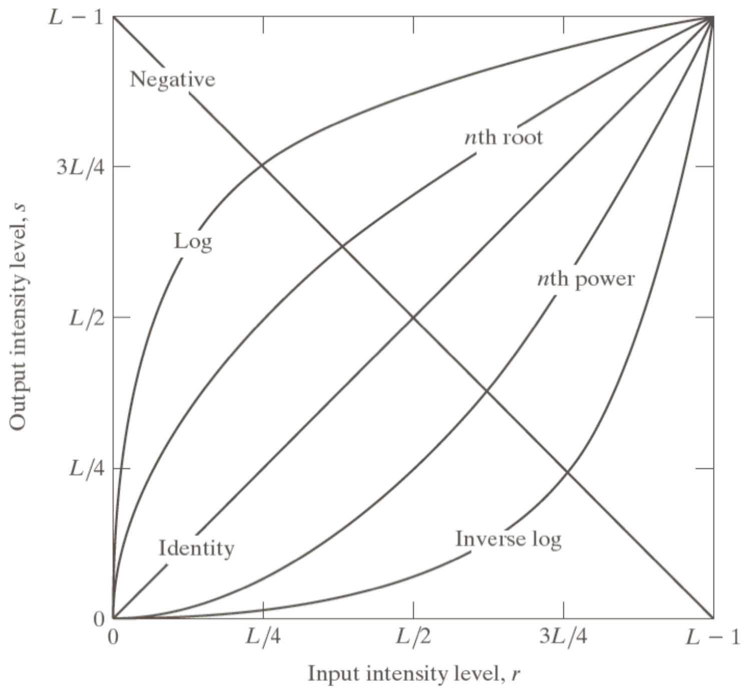
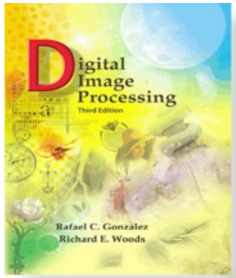
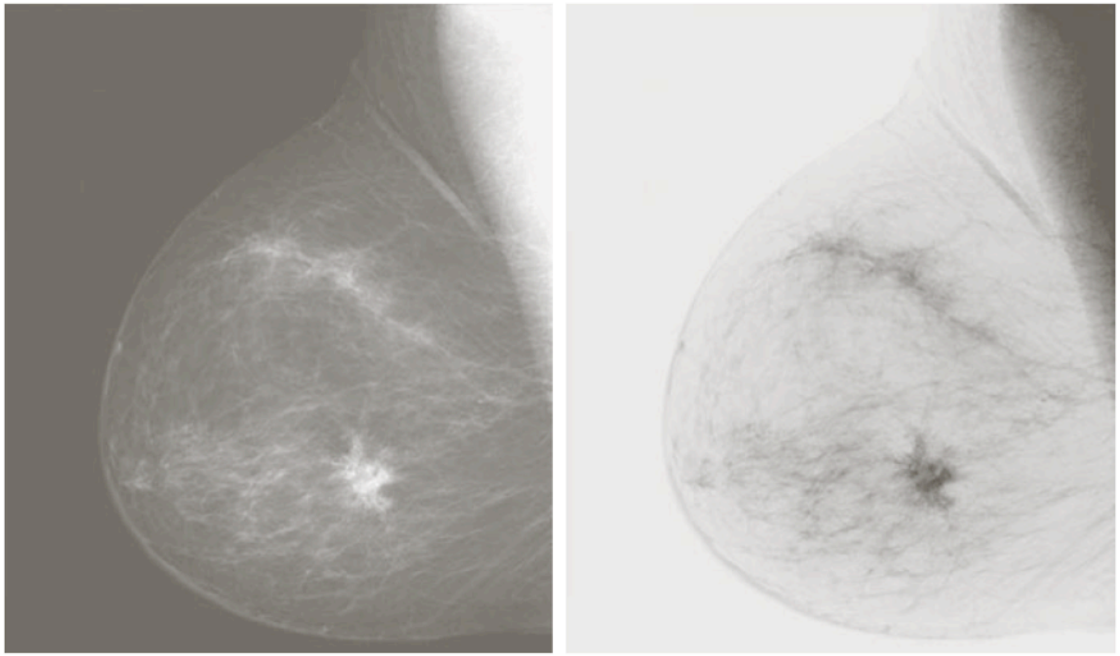


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.



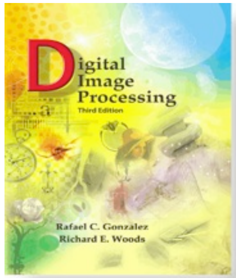
3.2.1 Image Negatives

Negative of an image with intensity levels in [0, L-1] obtained by: $s = (L-1) - r$



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.)

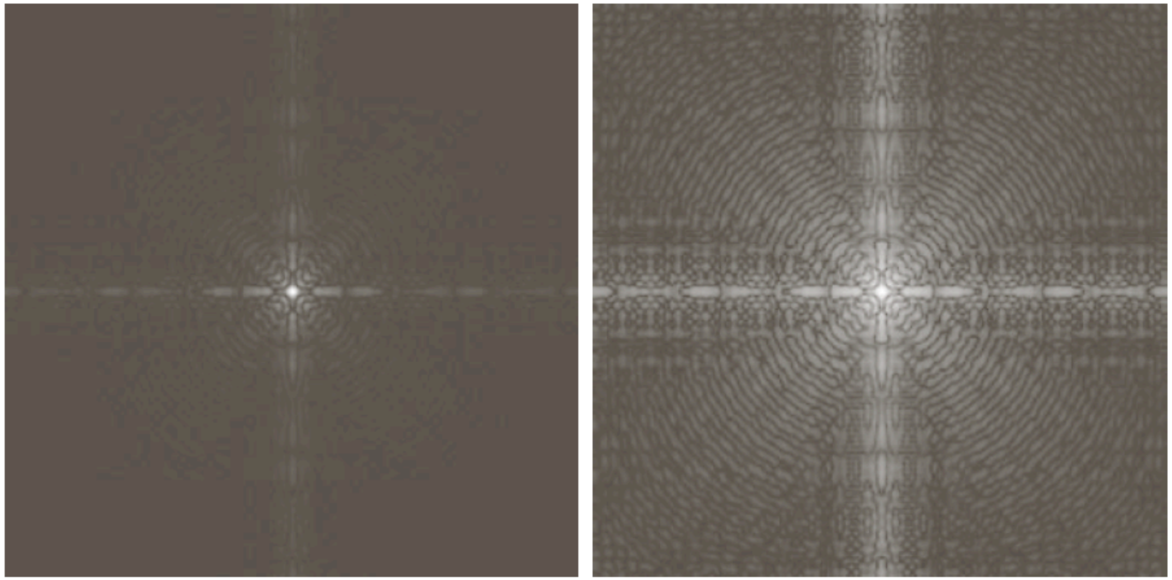


3.2.2 Log Transformations

General form : $s = c\,log(1 + r)$

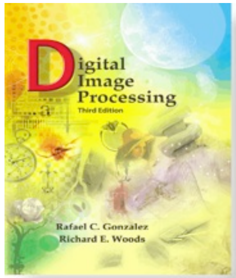
Where $c = \text{constant, and } r \geq 0$

=> Expand the values of dark pixels in an image while compressing the higher-level values



a b

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



3.2.3 Power-Law (Gamma) Transformations

Basic form: $s = cr^{\gamma}$ c and γ : positive constants

Sometimes written as $s = c(r+\epsilon)^{\gamma}$ to account for an offset

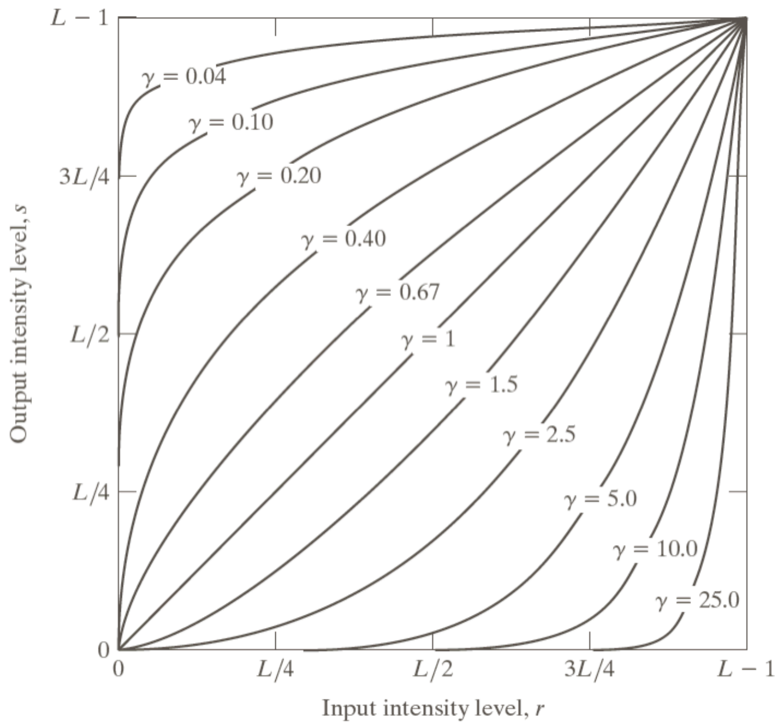
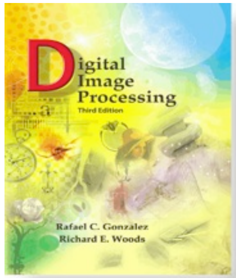


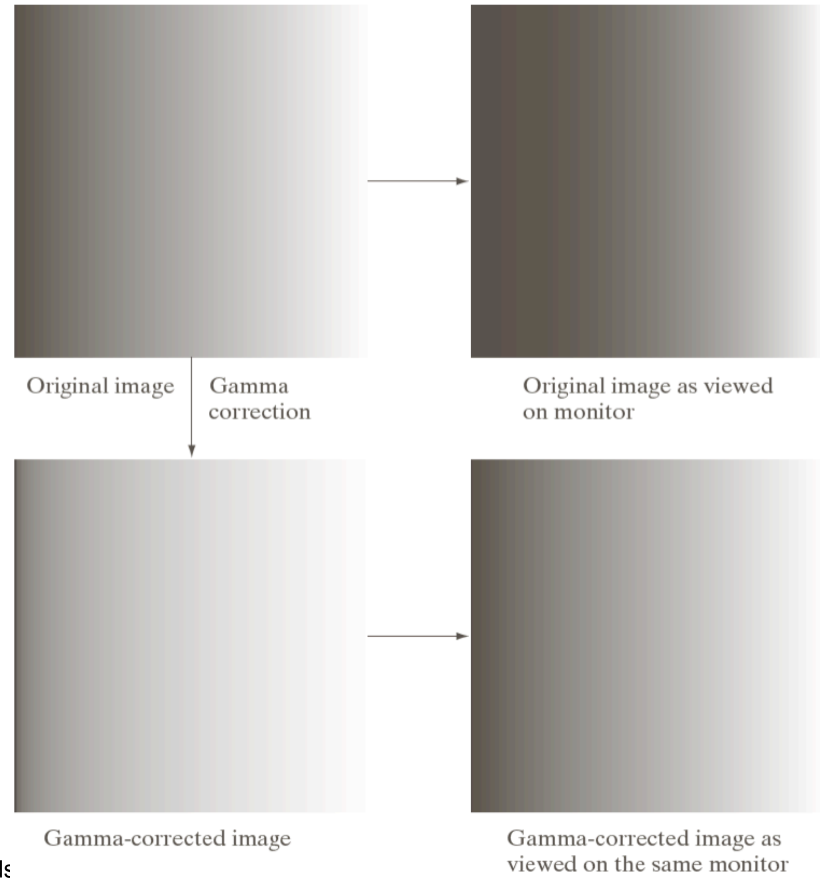
FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.



Example of gamma correction:

CRT (Cathode Ray Tube) devices have an intensity-to-voltage response that is a power function, with exponents from ~1.8 to 2.5 => images darker than intended

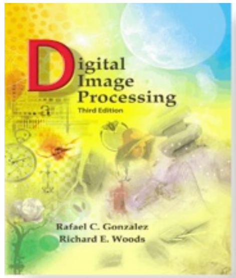
$$s = r^{1/2.5} = r^{0.4}$$



a b
c d

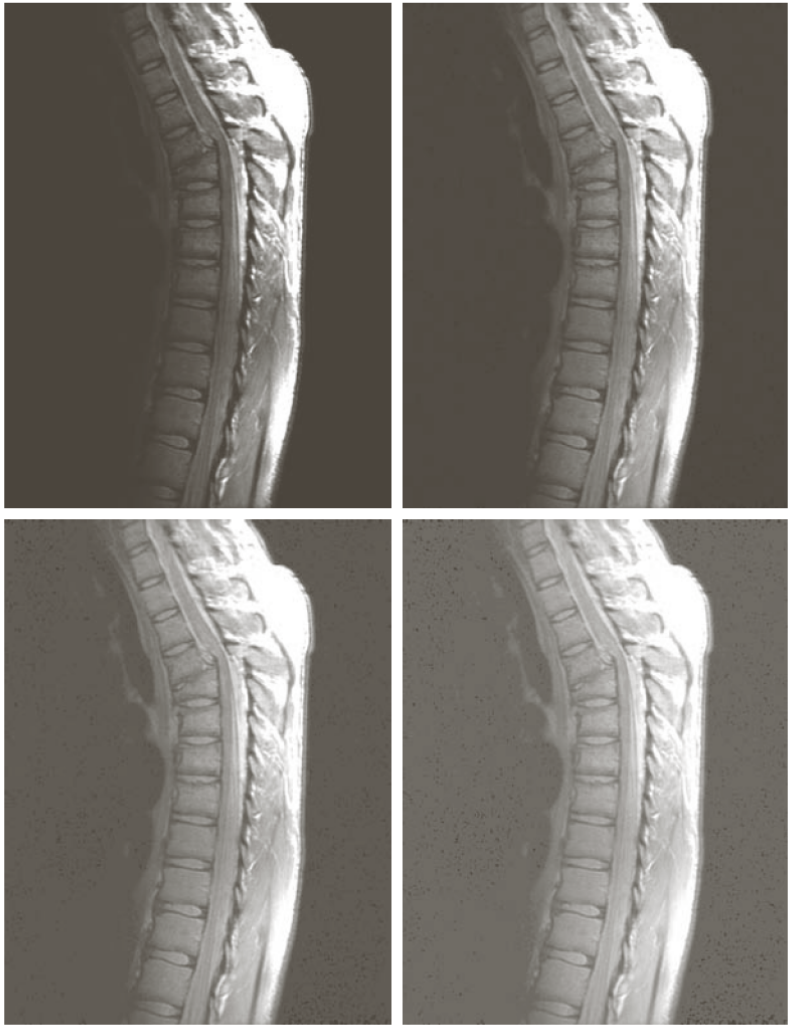
FIGURE 3.7 (a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).

NB: “Device-dependent” value of gamma



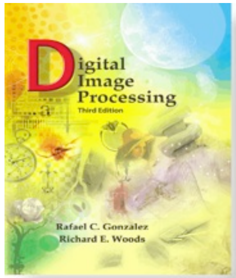
Example:

Expansion of intensity levels is desirable
=> can be accomplished with a power-law transformation with a fractional component



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.)



Opposite example:

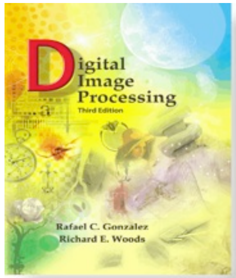
Image has a “washed-out” appearance

⇒ Compression of intensity levels is desirable

⇒ values of gamma > 1

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.)



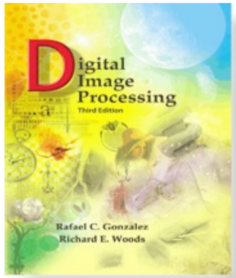
3.2.3 Piecewise-Linear Transformation Functions

Piecewise Linear Functions:

- ➚ : can be arbitrarily complex
- ➘ : specification requires much more user input

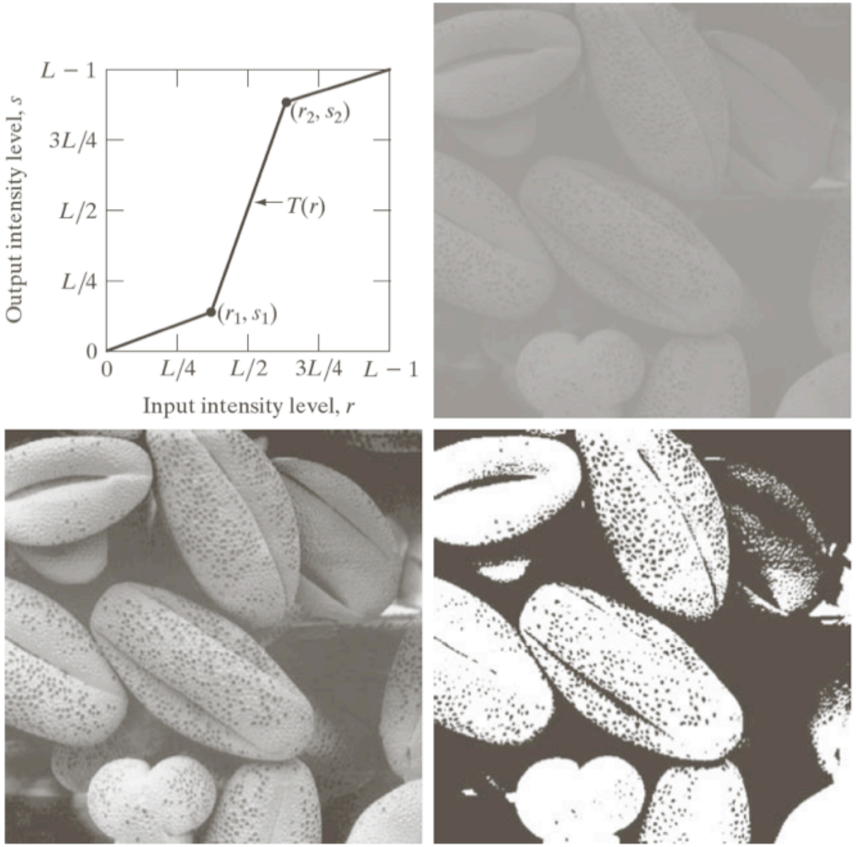
A) Contrast Stretching

Expands the range of intensity levels to span the full intensity range of the output device



A) Contrast Stretching

Original 8-bit 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.)

Contrast stretching

$(r_1, s_1) = (m, 0)$
$(r_2, s_2) = (m, L-1)$
m = mean intensity level

Special cases:

$$\begin{cases} r_1 = s_1 \\ r_2 = s_2 \end{cases}$$

=> Linear transformation (no change in intensity levels)

$$\begin{cases} r_1 = r_2 \\ s_1 = 0 \\ s_2 = L-1 \end{cases}$$

=> *Thresholding function* (=> binary image)

B) Intensity level slicing

Goal: highlight a *specific range* of intensities

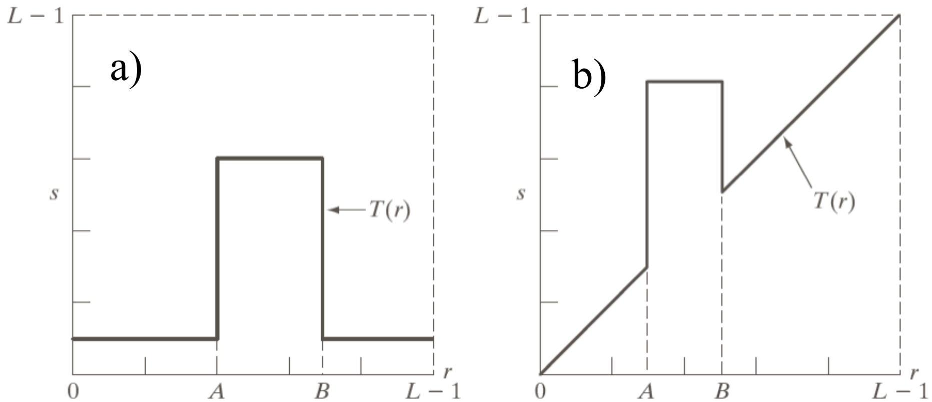
Two main approaches:

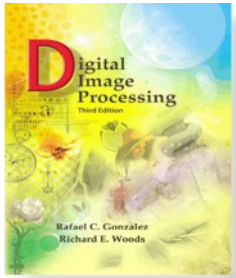
a) Display in one value (e.g. white) all values in range of interest and all others in another value (e.g. black) => binary image

b) Brighten (or darken) the desired range, leaving all other intensity levels unchanged

a b

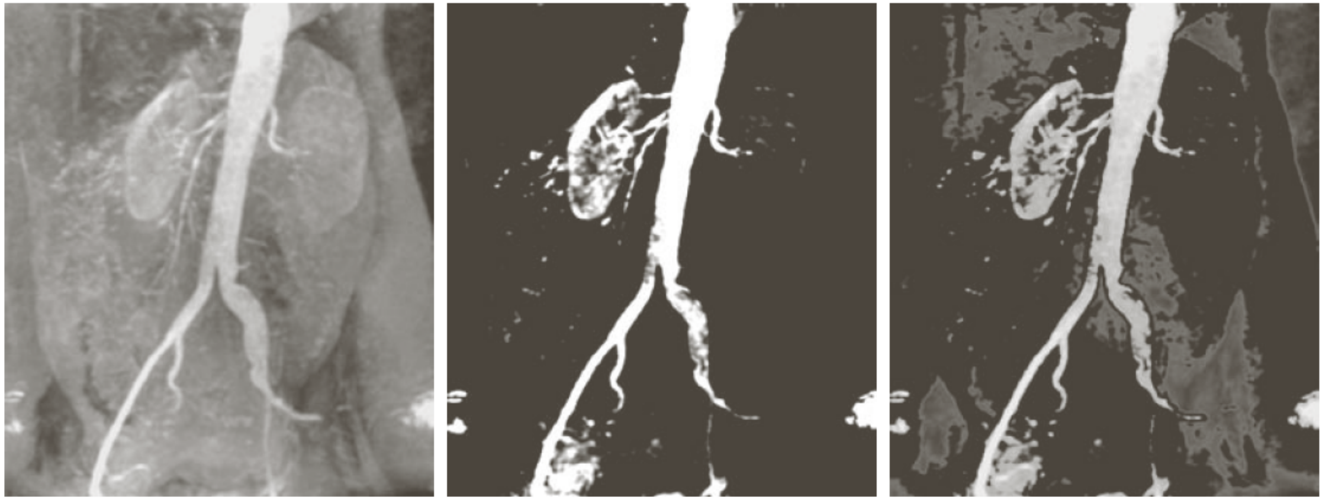
FIGURE 3.11 (a) This transformation highlights intensity range [A, B] and reduces all other intensities to a lower level. (b) This transformation highlights range [A, B] and preserves all other intensity levels.





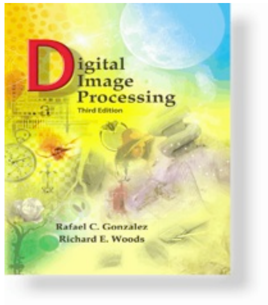
B) Intensity level slicing

approach a) approach b)



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.)



C) Bit-plane slicing

Goal: highlight the contribution made to total image appearance by specific bits

Example: 8-bit gray-scale image, composed of 8 1-bit planes

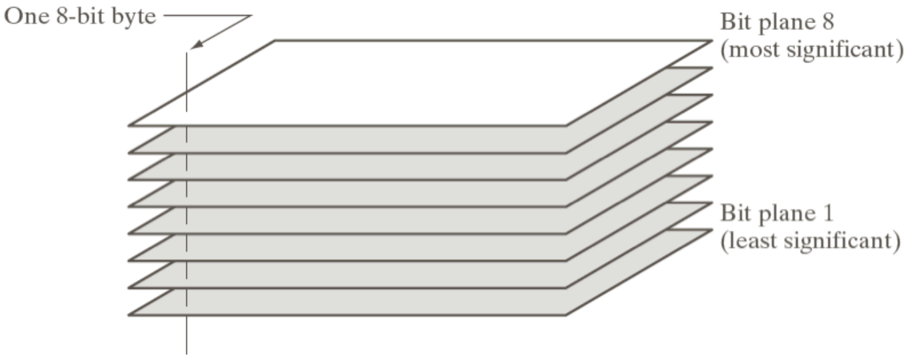


FIGURE 3.13 Bit-plane representation of an 8-bit image.

Chapter 3

Intensity Transformations & Spatial Filtering

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.

C) Bit-plane slicing

Original 8-bit image

Application to image compression:

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).

3.3 Histogram Processing

Histogram of an MxN image with intensity levels in [0,L-1]:

Discrete function: $h(r_k) = n_k$

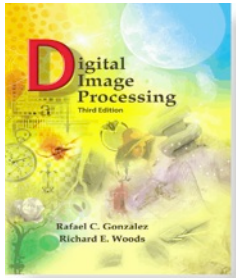
Where:

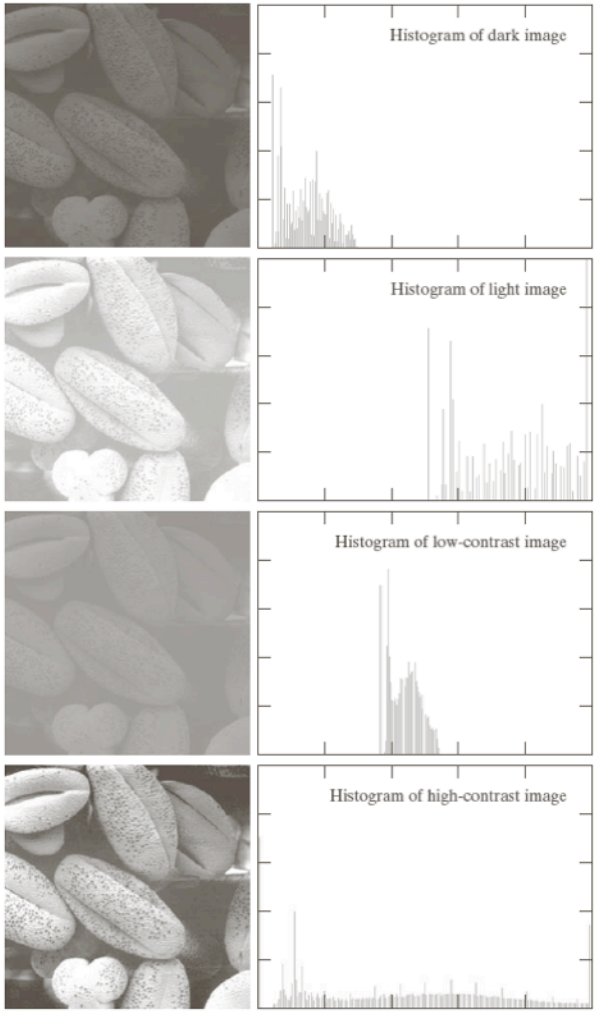
- $r_k = k^{th}$ intensity value
- n_k = number of pixels with intensity r_k

Normalized histogram: $p(r_k) = n_k/MN, \text{ for } k = 0, 1, 2, \ldots L-1$

$p(r_k)$: "estimate of the probability of occurrence of intensity level r_k in the image"

NB: in a normalized histogram, $\sum_{k=0}^{L-1} p(r_k) = 1$





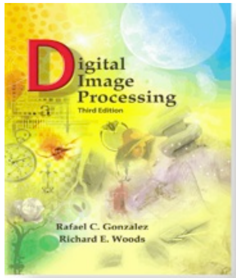
$h(r_k)$ or $p(r_k)$

intensity values: r_k

FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

Histograms:

- Basis for numerous spatial domain processing techniques
- Provides useful image statistics
- Useful for image enhancement, image compression and segmentation
- Simple to calculate in software (+ hardware implementations)



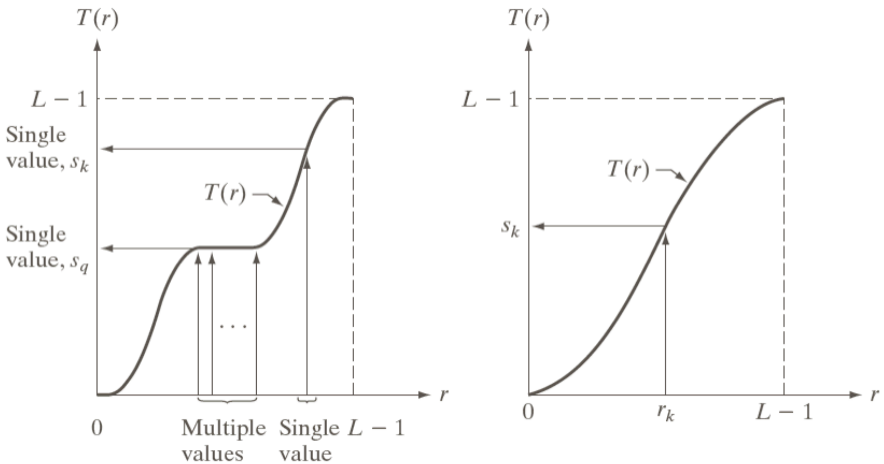
3.3.1 Histogram Equalization

Goal: obtain an output image with a uniform PDF(*) (continuous case)

Input image → output image
$r \rightarrow s$ (intensity levels)

Consider continuous intensity values

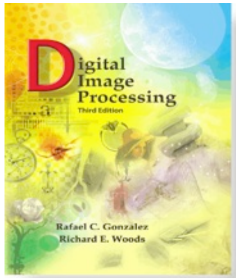
PDF of s (output image): $$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right|$$



a b

FIGURE 3.17 (a) Monotonically increasing function, showing how multiple values can map to a single value. (b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

(*) PDF: Probability Density Function



3.3.1 Histogram Equalization

Cumulative Distribution Function

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

$$\Rightarrow \frac{ds}{dr} = (L-1)\,p_r(r)$$

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

$$= \frac{1}{(L-1)} \quad 0 \le s \le L-1$$

$\rightarrow$ *Uniform* PDF

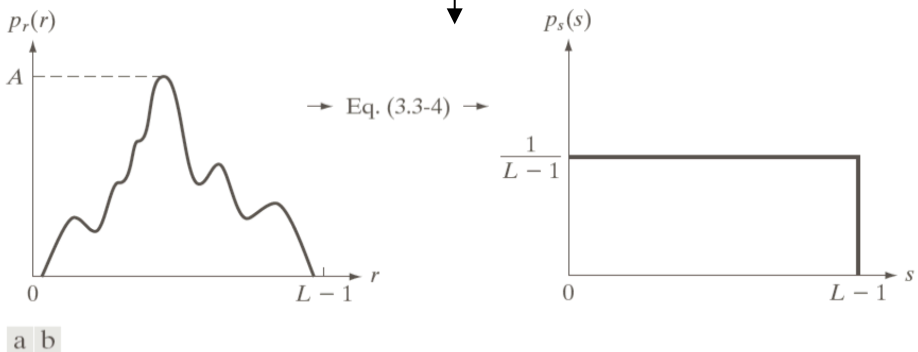


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

3.3.1 Histogram Equalization

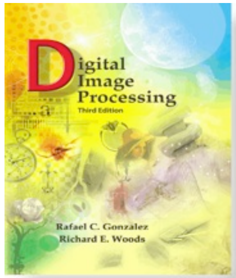
Histogram equalization or *histogram linearization* transformation (discrete form):

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

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

r_k: intensity levels in the *input* image

s_k: intensity levels in the *output* image



3.3.1 Histogram Equalization

Example: 3-bit image (L=8), 64x64 pixels:

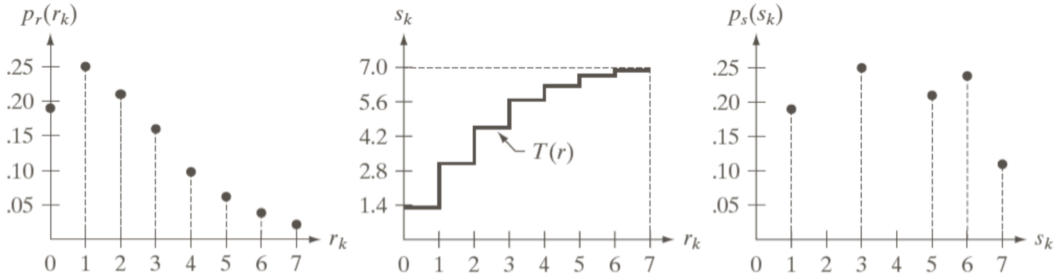
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

TABLE 3.1 Intensity distribution and histogram values for a 3-bit, 64×64 digital image.

Round intensity values to nearest integer:

$$s_0 = 1.33 \rightarrow 1 \quad s_4 = 6.23 \rightarrow 6$$
$$s_1 = 3.08 \rightarrow 3 \quad s_5 = 6.65 \rightarrow 7$$
$$s_2 = 4.55 \rightarrow 5 \quad s_6 = 6.86 \rightarrow 7$$
$$s_3 = 5.67 \rightarrow 6 \quad s_7 = 7.00 \rightarrow 7$$

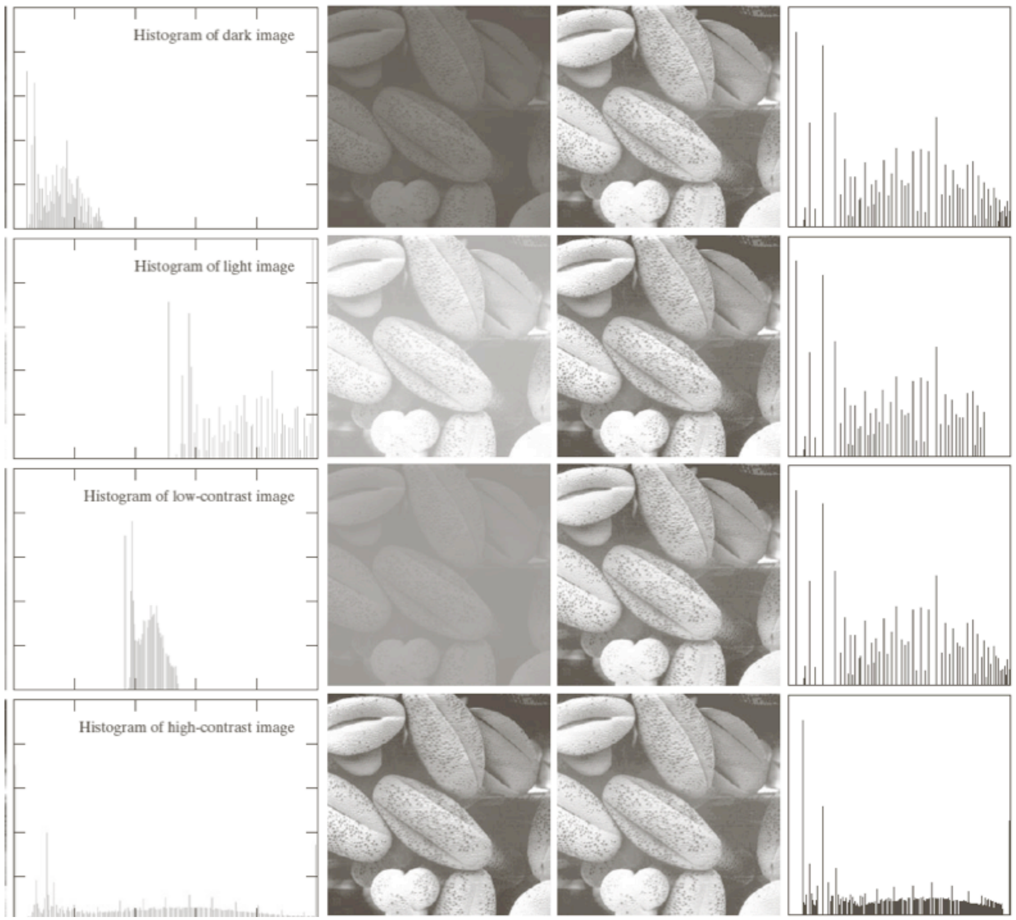
=> 5 intensity levels



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 as an *Adaptive Contrast Enhancement Tool*



a) Original Histograms *b) Original Images* *c) Histogram-equalised im.* *Histograms of c)*

Tend to spread the histogram of the input image
⇒ intensity levels of the equalized image span a wider range
⇒ *contrast enhancement*

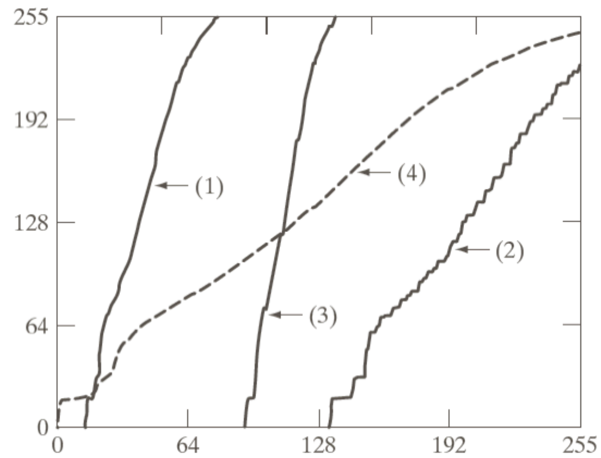
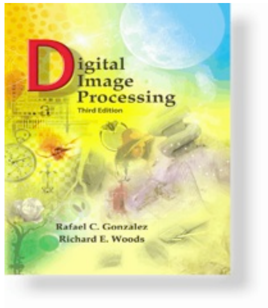


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).



3.3.2 Histogram Matching (Specification)

Histogram Matching or *Histogram Specification*

r and z: continuous intensities (random variables)

$p_r(r)$ and $p_z(z)$: corresponding PDF

$p_z(z)$: *specified* PDF

Input image → output image
$r \rightarrow z$ (intensity levels)

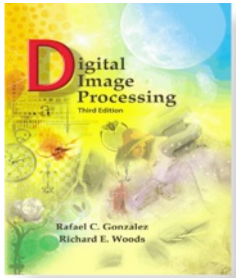
$$s = T(r) = (L-1)\int_0^r p_r(w)dw \qquad (3\text{-}3.1)$$

Transformation Function:

$$G(z) = (L-1)\int_0^z p_z(t)dt = s \qquad (3\text{-}3.2)$$

Mapping:

$$G(z) = T(r) \quad \Rightarrow z = G^{-1}[T(r)] = G^{-1}(s) \qquad (3\text{-}3.3)$$



3.3.2 Histogram Matching (Specification)

Procedure to obtain an image whose intensity levels have a specified PDF:

1. Obtain $p_r(r)$ from the input image, and use Eq. (3-3.1) to obtain the values of s
2. Use the specified PDF in Eq. (3-3.2) to obtain the transformation function $G(z)$
3. Obtain the inverse transformation $z = G^{-1}(s)$ (i.e. *mapping* from s to z)
4. Obtain the output image by equalizing the input image using Eq. (3-3.1). For each pixel with value s in the equalized image, perform the inverse mapping $z = G^{-1}(s)$ to obtain pixels in the output image.

3.3.2 Histogram Matching (Specification)

Discrete Formulation

Input image → output image
$r_k \rightarrow z_k$ (intensity levels)

$p_z(z_k)$: *specified* PDF

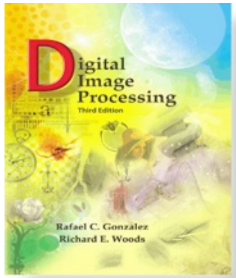
$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_k \qquad \text{(3-3.4)}$$

Transformation Function:

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) \qquad \text{(3-3.5)}$$

Mapping:

$$z_q = G^{-1}(s_k) \qquad \text{(3-3.6)}$$

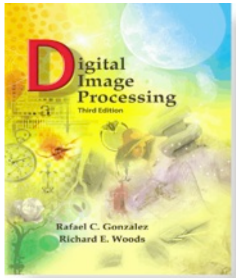


3.3.2 Histogram Matching (Specification)

s_k being the values of the histogram-equalized image,

Discrete formulation of the histogram-specification procedure:

1. Compute $p_r(r)$ of input image, and use Eq. (3-3.4) to obtain the values of s_k, *rounded* to the integer range *[0,L-1]*
2. Compute all values of G using Eq. (3-3.5) for $q=0,1,\ldots,L-1$ and the specified histogram values. Round the values of G to integers in the range *[0,L-1]* and store those values in a table.
3. For every value of s_k, use 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 mapping from s to z. When mapping not unique, choose the smallest value by convention.
4. Form the histogram-specified image by "histogram-equalizing" the input image and then mapping every equalized pixel value s_k of this image to the corresponding value z_q using the mapping found in step 3.



Example of Histogram Specification

Original histogram

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

Specified histogram

TABLE 3.2 Specified and actual histograms (the values in the third column are from the computations performed in the body of Example 3.8).

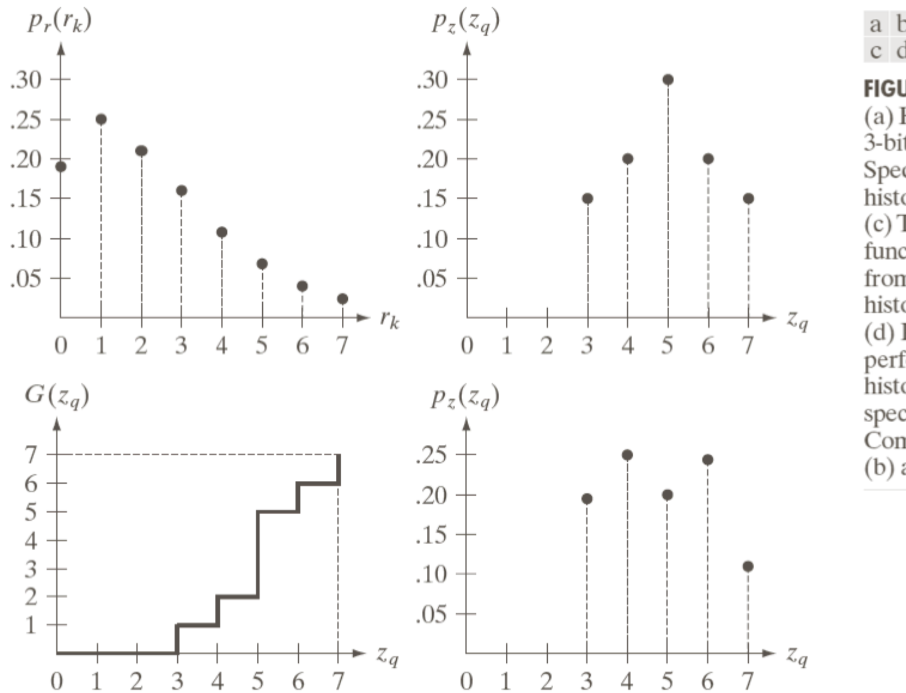
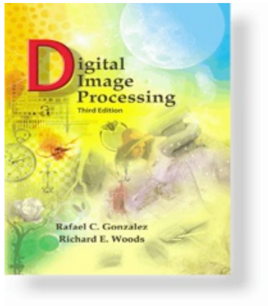


FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).



Example of Histogram Specification

Original histogram

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

Step 1: obtain scaled histogram-equalized values:

$$
\begin{array}{ll}
s_0 = 1.33 \rightarrow 1 & s_4 = 6.23 \rightarrow 6 \\
s_1 = 3.08 \rightarrow 3 & s_5 = 6.65 \rightarrow 7 \\
s_2 = 4.55 \rightarrow 5 & s_6 = 6.86 \rightarrow 7 \\
s_3 = 5.67 \rightarrow 6 & s_7 = 7.00 \rightarrow 7
\end{array}
$$

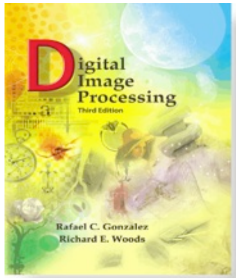
z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

TABLE 3.2 Specified and actual histograms (the values in the third column are from the computations performed in the body of Example 3.8).

Step 2: compute all values of G, rounded:

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

TABLE 3.3 All possible values of the transformation function G scaled, rounded, and ordered with respect to z.



$s_0 \rightarrow 1$
$s_1 \rightarrow 3$
$s_2 \rightarrow 5$
$s_3 \rightarrow 6$
$s_4 \rightarrow 6$
$s_5 \rightarrow 7$
$s_6 \rightarrow 7$
$s_7 \rightarrow 7$

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

Step 3: for every s_k, find the smallest value of z_q so that $G(z_q)$ is the closest to s_k :

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

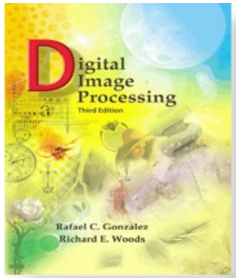
TABLE 3.4 Mappings of all the values of s_k into corresponding values of z_q.

Step 4: use Table 3.4 to map pixels in the histogram equalized image into a pixel in the new histogram-specified image

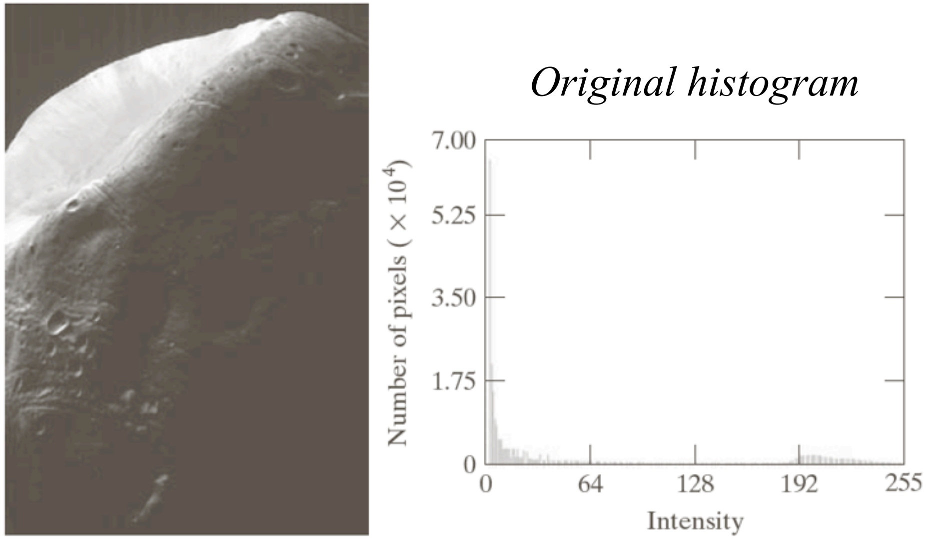
Values of the resulting histogram:

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

TABLE 3.2 Specified and actual histograms (the values in the third column are from the computations performed in the body of Example 3.8).



Example: Comparison between histogram equalization and histogram matching



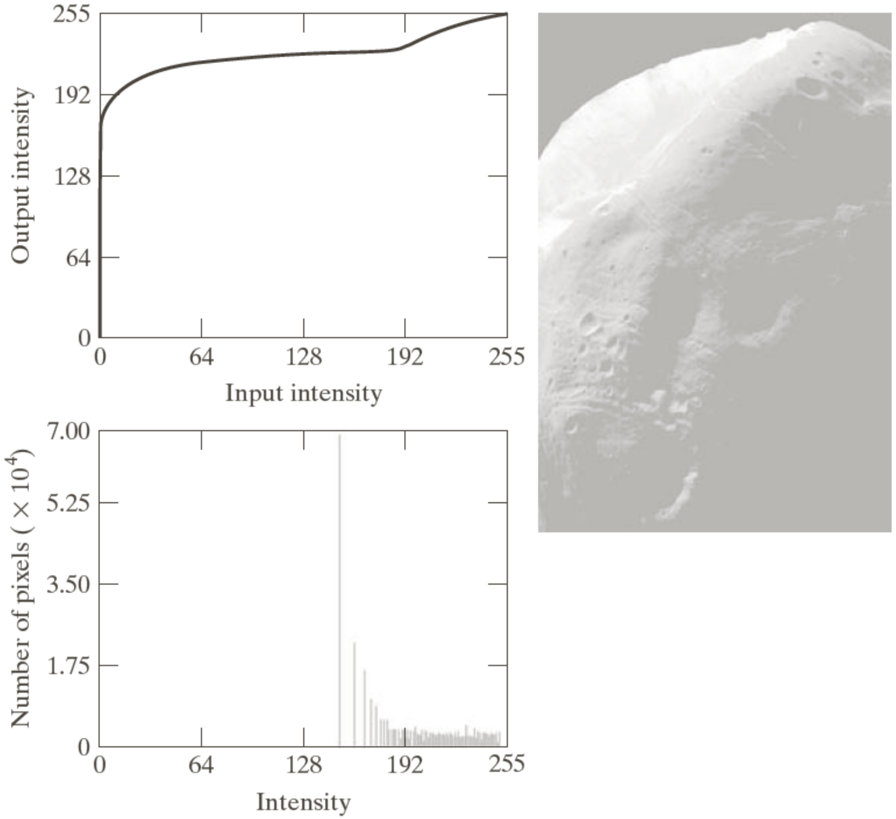
a b

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: Comparison between histogram equalization and histogram matching

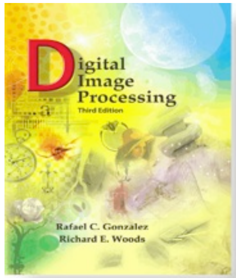
Histogram equalization

Original image



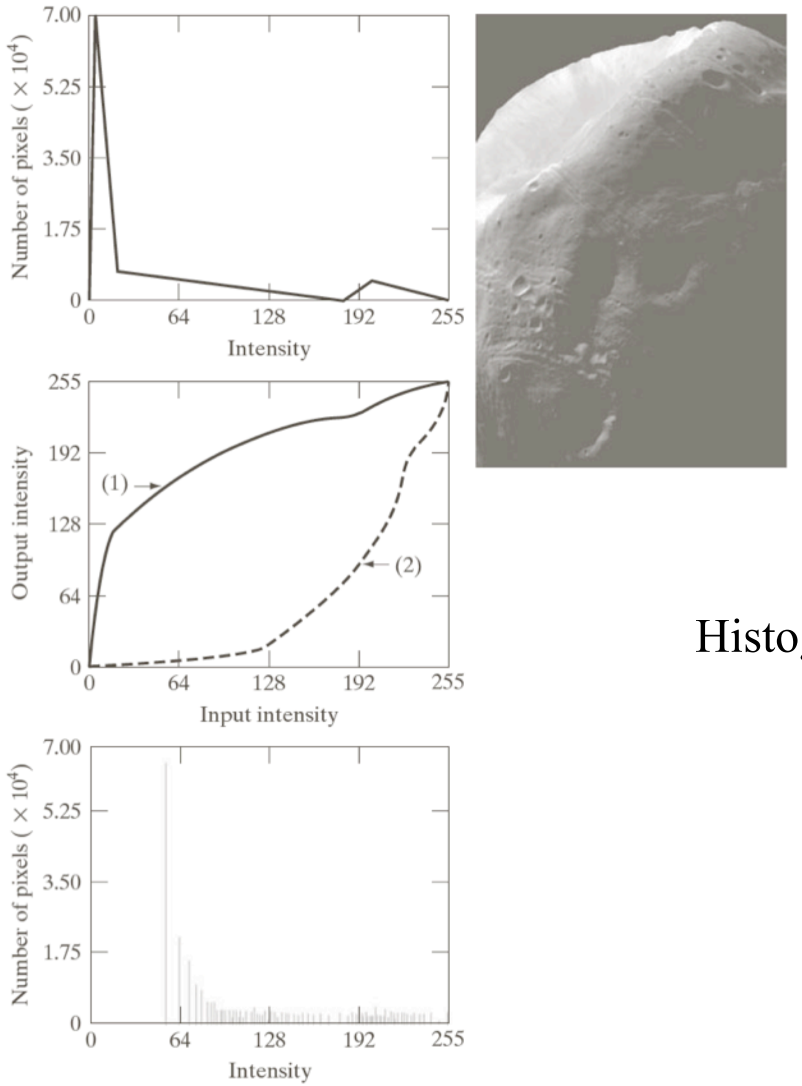
a b
c

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



Chapter 3
Intensity Transformations & Spatial Filtering

Original image



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

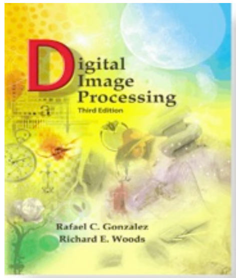


Image $f(x,y)$ → Image $g(x,y)$

3.4 Fundamentals of Spatial Filtering

Spatial filters (also called spatial *masks*, *kernels, templates* or *windows*)

3.4.1 The Mechanics of Spatial Filtering

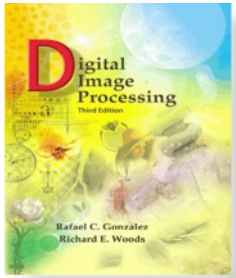
Filtering creates a new pixel with coordinates equal to the coordinates of the center of the neighbourhood, with value = result of the filtering operation.

Example : linear spatial filter using a 3x3 neighbourhood:

At any point *(x,y)* in the original image, response *g(x,y)* of the filter:

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

Centre coefficient of filter



3.4.1 The Mechanics of Spatial Filtering

For a mask of size *mxn*, we assume:

m = 2a + 1 and *n = 2b + 1*

where *a* and *b* are positive integers (*odd size*)

General expression of a linear spatial filtering of an image of size M x N with a filter of size m x n :

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

(each pixel in *w* visiting every pixel in *f*)

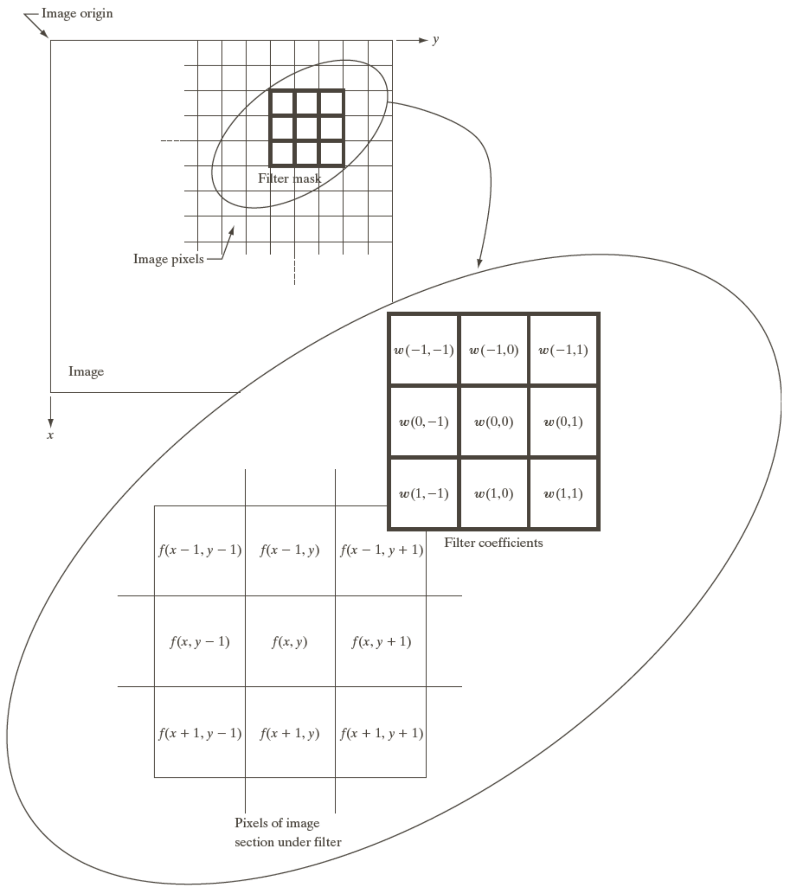
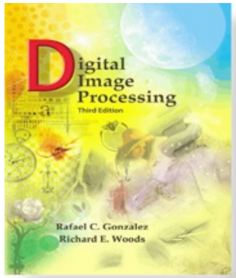


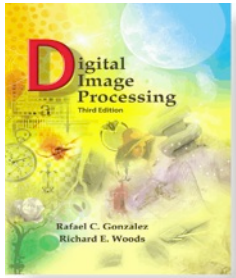
FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.



3.4.2 Spatial Correlation and Convolution

Correlation = process of moving a filter mask over the image and computing the sum of products at each location

Convolution = same mechanics but with a filter rotated by 180°



3.4.2 Spatial Correlation and Convolution

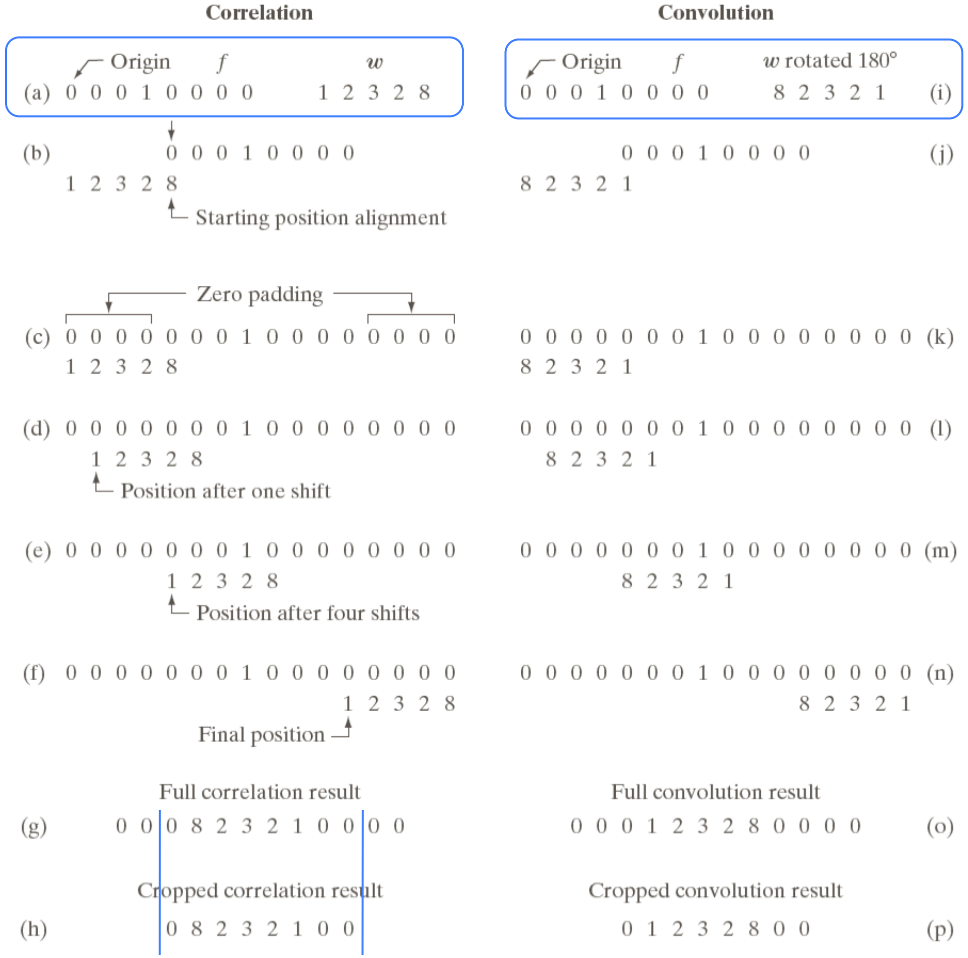


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

3.4.2 Spatial Correlation and Convolution

- Correlation is a function of *displacement* of the filter
- If *f* contains all 0s and only one 1 (*discrete unit impulse*) => copy of *w* rotated by 180°
- i.e. correlation of a function with a discrete unit impulse => rotated version of the function at the location of the impulse
- Convolving a function with a unit impulse => copy of *w* at the location of the impulse
- Convolution: rotate one function by 180° and perform same operations as in correlation

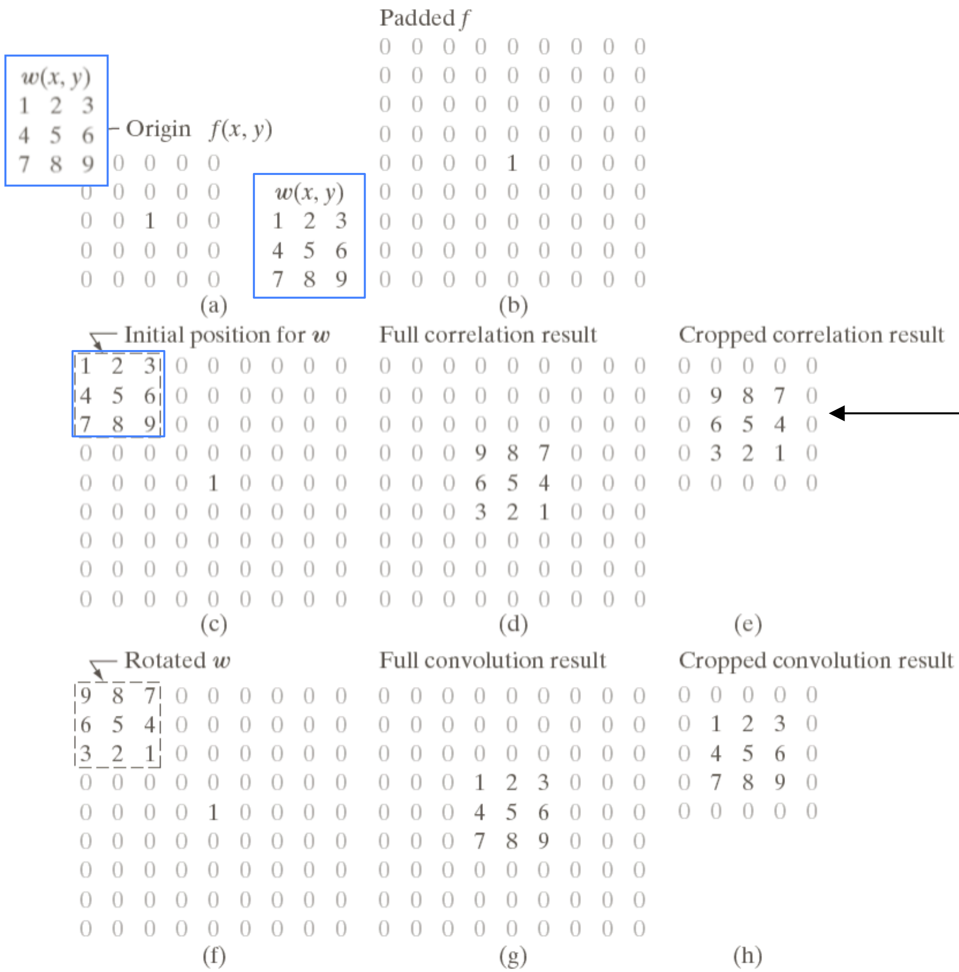
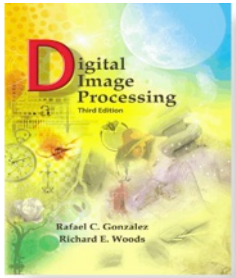


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Result rotated 180°

NB: if filter mask symmetric, correlation and convolution => same result



3.4.2 Spatial Correlation and Convolution

- Correlation of a filter *w(x,y)* with an image *f(x,y)*:

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

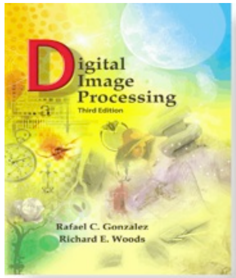
For all values of x and y so that all elements of *w* visit every pixel in *f* (*f padded* appropriately)

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

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

flipping

For all values of x and y so that all elements of *w* visit every pixel in *f* (*f* padded appropriately)



3.4.3 Vector Representation of Linear Filtering

It is convenient sometimes to write the sum of product as:

$$R = w_1 z_1 + w_2 z_2 + \ldots w_{mn} z_{mn}$$

$$= \sum_{k=1}^{mn} w_k z_k \quad = \boldsymbol{w}^T \boldsymbol{z}$$

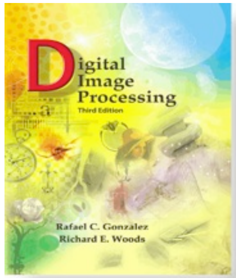
Where:

- w_k : coefficients of an *m*x*n* filter
- z_k : corresponding image intensities encompassed by the filter

Example:

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

FIGURE 3.31 Another representation of a general 3×3 filter mask.



3.4.4 Generating Spatial Filter Masks

To generate a *mxn linear* filter, need to specify *mn* mask coefficients, selected based on what the filter is supposed to do

Example 1 : average filter with 3x3 neighbourhood: $R = \frac{1}{9} \sum_{i=1}^{9} z_i \quad \Rightarrow \quad w_i = \frac{1}{9}$

Example 2 : spatial filter mask based on a continuous function of two variables
e.g. Gaussian Function:

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

3x3 mask:

$$w_1 = h(-1,-1)$$
$$w_2 = h(-1,0)$$
$$\vdots$$
$$w_9 = h(1,1)$$

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

Averaging filter

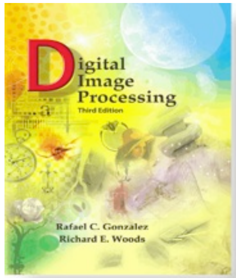
$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Weighted average

a b

FIGURE 3.32 Two 3 × 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.



3.5 Smoothing Spatial Filters

3.5.1 Smoothing Linear Filters

Output = average of pixels contained in the neighbourhood of the filter mask

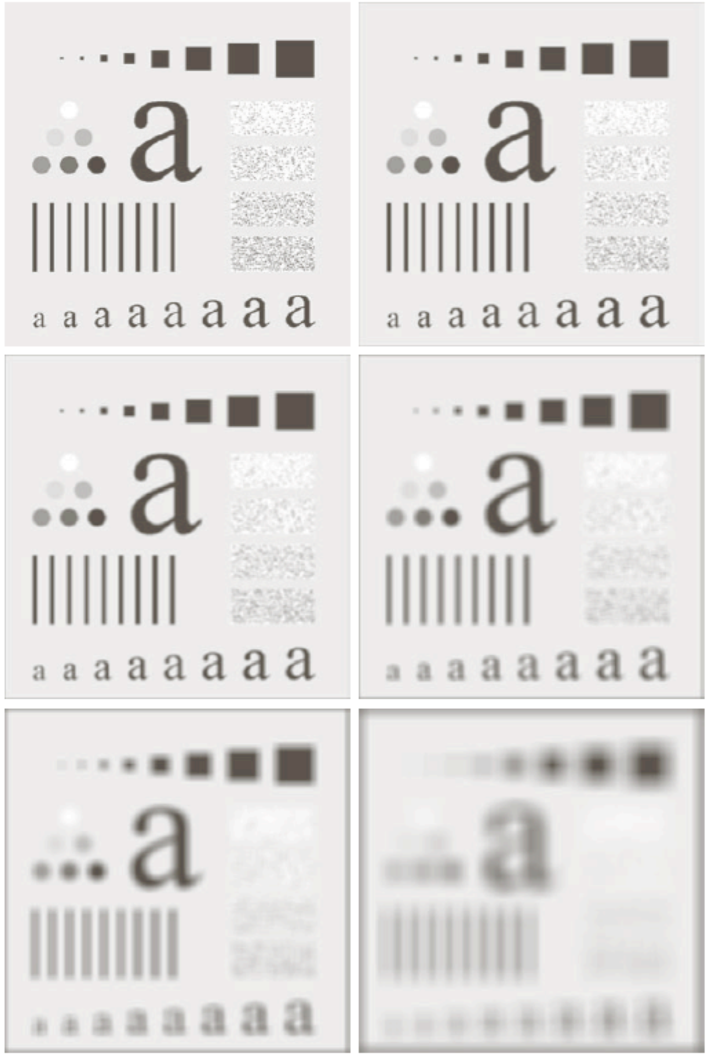
Averaging filters or *lowpass filters*

⇒ Reduce "sharp" transitions in intensities

⇒ Application 1: noise reduction

⇒ Application 2: smoothing of false contours

⇒ Blurring effect



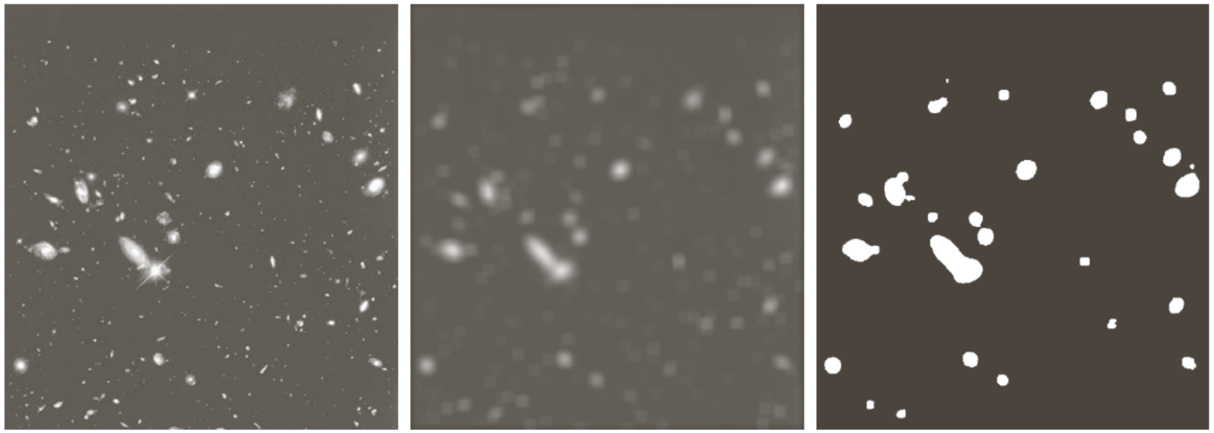
a b
c d
e f

FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, 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 intensity 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.

General implementation for filtering an *M*x*N* image with a weighted averaging filter of size *m*x*n*:

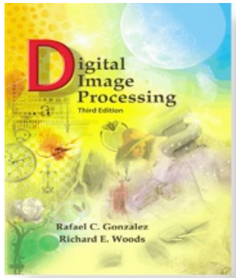
$$g(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)}$$

Sum of the mask coefficients: constant computed only once



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

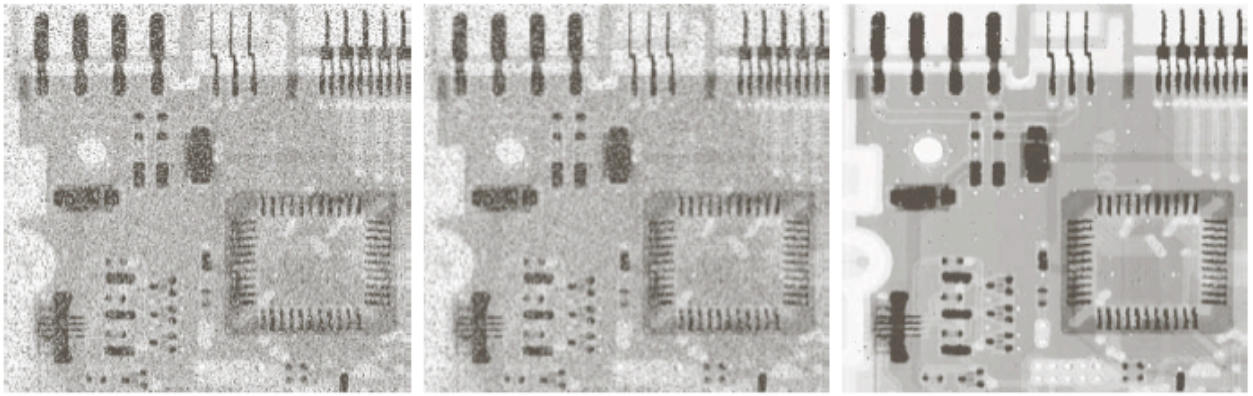


3.5.2 Order-Statistic (Nonlinear) Filters

Ordering (ranking) the pixels contained in the image area encompasses by the filter, then replacing the value of the centre pixel with the value of ranking result

Median filter: replaces the value of a pixel by the median of the intensity values in the neighbourhood. Good noise-reduction capabilities with less smoothing (e.g. *impulse noise*, or *salt-and-pepper noise*)

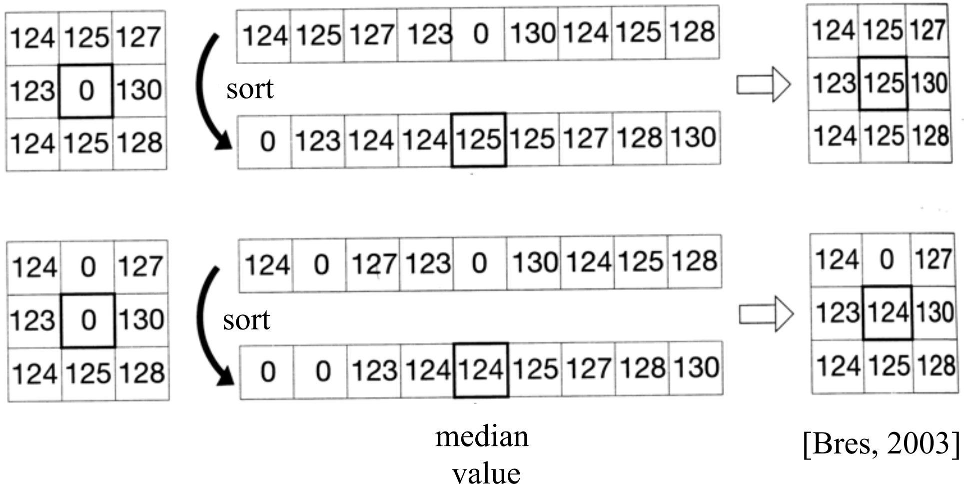
Max filter, *min filter*



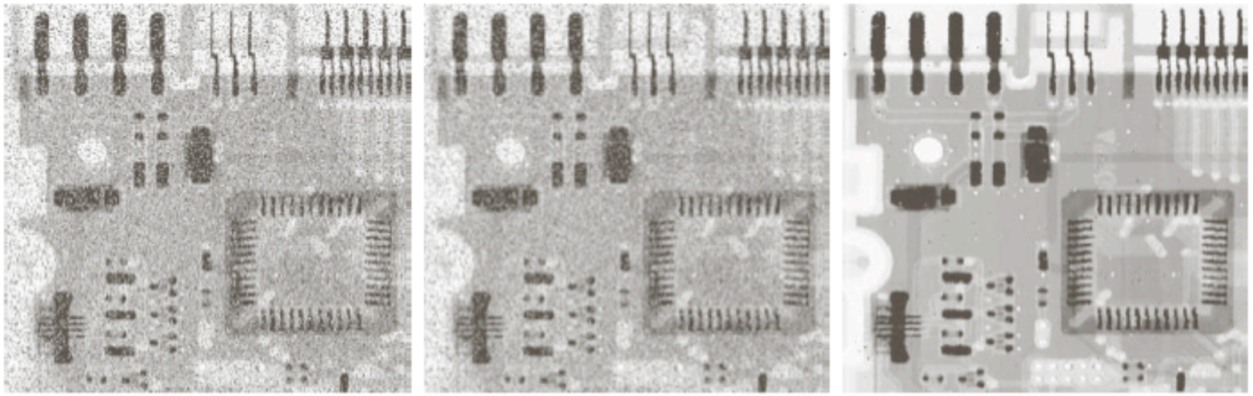
a b c

FIGURE 3.35 (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.)

Median filter :

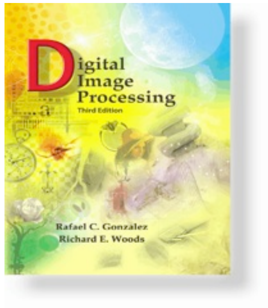


[Bres, 2003]



a b c

FIGURE 3.35 (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.)



3.6 Sharpening Spatial Filters

Objective: highlight transitions in intensity

Spatial (digital) differentiation => enhances edges and other discontinuities

3.6.1 Foundation

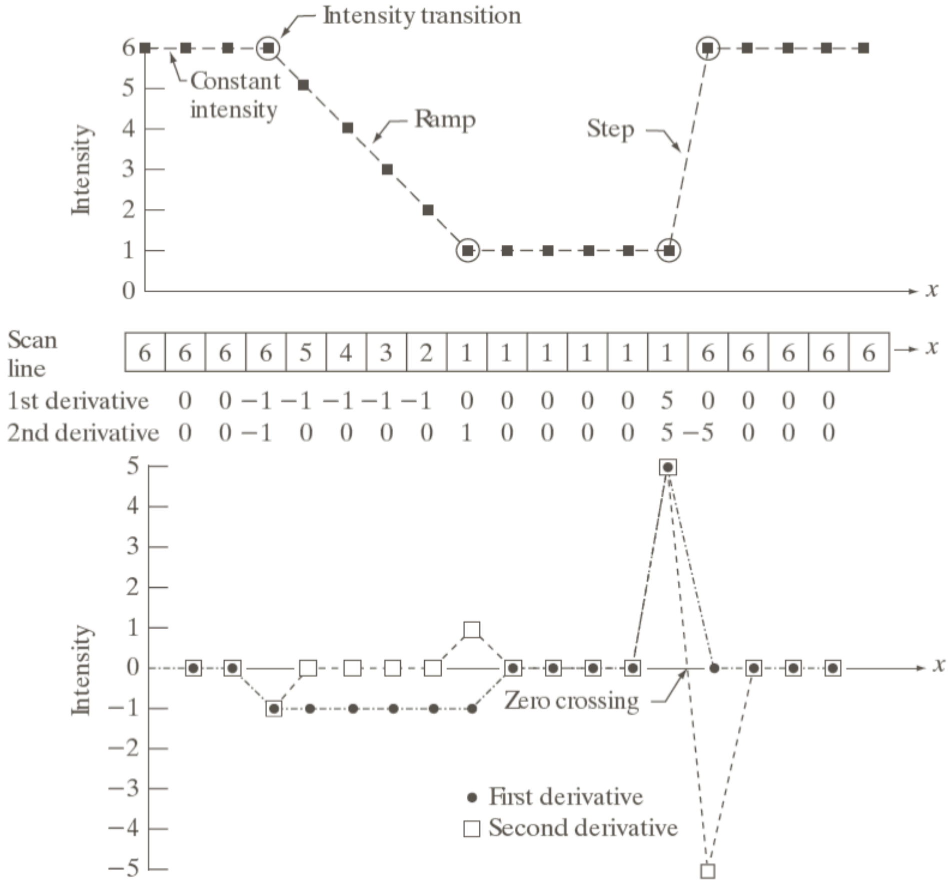
Let's study the behavior of first- and second-order derivatives in areas of constant intensities, at the onset and end of discontinuities, and along intensity ramps

Basic definition of first-order derivative of a one-dimensional function f(x):

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

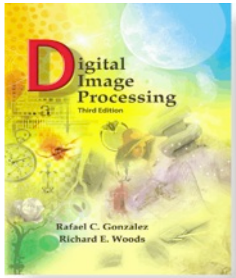
Second derivative: $$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

3.6.1 Foundation



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 Using the Second Derivative for Image Sharpening - The Laplacian

Discrete formulation of second-order derivative => construct filter mask

Isotropic filters: response independent of the direction of the discontinuities in the image to which the filter is applied (*rotation invariant*)

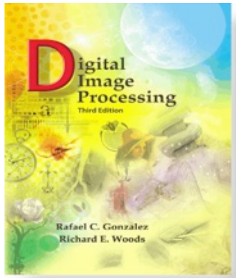
Simplest isotropic derivative operator: Laplacian: $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$ *(Linear operator)*

Discrete form:

In the *x*-direction: $$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y) \qquad (3.6\text{-}1)$$

In the *y*-direction: $$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y) \qquad (3.6\text{-}2)$$

=> $$\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y) \qquad (3.6\text{-}3)$$



Laplacian Masks

After incorporation of the diagonal directions

Eq. (3.6-3) =>

(a)

0	1	0
1	−4	1
0	1	0

=> Isotropic result for rotations in increments of 90º

(b) ⇓

1	1	1
1	−8	1
1	1	1

=> Isotropic result for rotations in increments of 45º

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

(Negative definitions of the second derivative)

a b
c d

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.

"Recovering" the background:

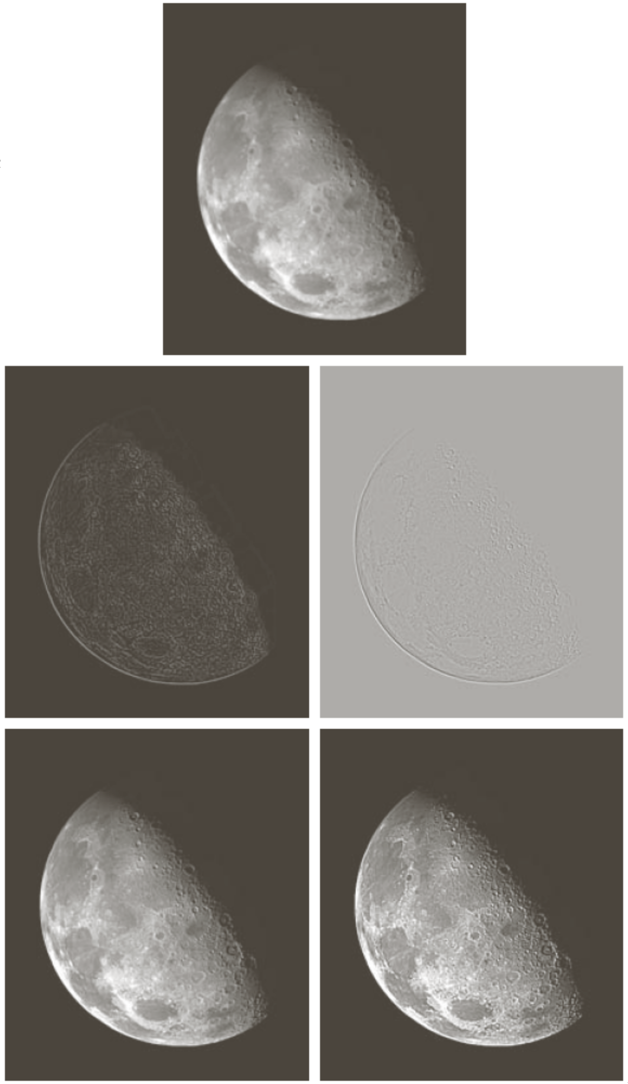
Add (or substract) the Laplacian image to the original

=> *Sharpened* image:

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

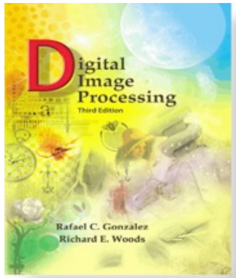
Where:

- c = -1 if filter (a) or (b)
- c = 1 if filter (c) or (d)



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.)



3.6.3 Unsharp Masking and Highboost Filtering

Unsharp Masking:

1. Blur the original image *f(x,y)*

2. Substract the blurred image $\overline{f}(x,y)$ from the original (result called the *mask*):

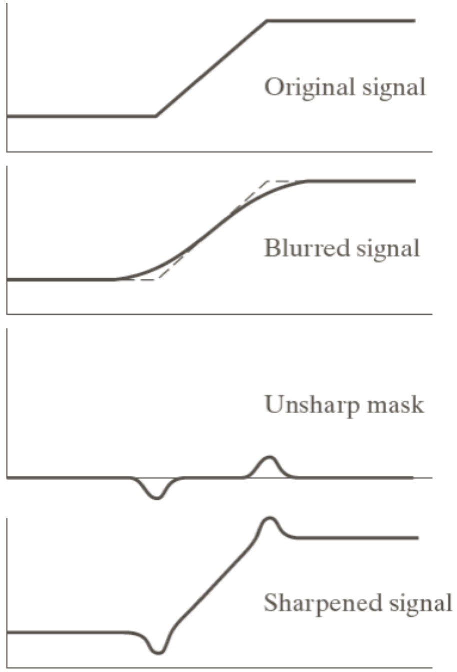
$$g_{mask}(x,y) = f(x,y) - \overline{f}(x,y)$$

3. Add the mask to the original:

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

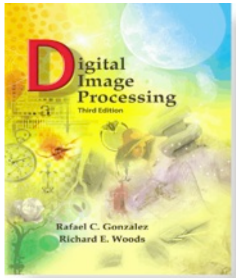
$$k \geq 0$$

- $k = 1$ => *unsharp masking*
- $k > 1$ => *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).



3.6.3 Unsharp Masking and Highboost Filtering

Example:

Blurred image

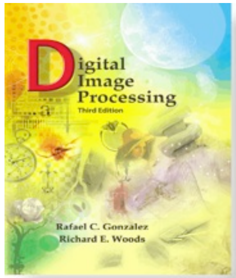
mask

Unsharp masking

Highboost filtering

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.



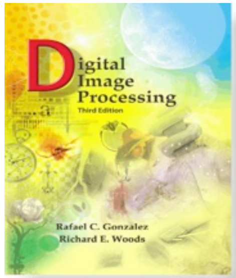
3.6.4 Using First-Order Derivatives for (Nonlinear) Image Sharpening - The Gradient

- *Gradient* of *f* at *(x,y)*: $$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$
- This vector points in the direction of the greatest rate of change of *f* at location *(x,y)*
- *Magnitude* (*length*) of *grad(f)*: $$M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}$$
- *M(x,y)*, with *x* and *y* varying over all *f*, is called the *gradient image*

NB: The Gradient is a linear operator but not rotation invariant (isotropic)

The Magnitude is not linear, but is isotropic

- Approximation: $M(x,y) \approx |g_x| + |g_y|$ => Isotropic property lost (in general)



3.6.4 Using First-Order Derivatives for (Nonlinear) Image Sharpening - The Gradient

Definitions proposed by Roberts [1965] using cross differences:

$$g_x = (z_9 - z_5) \text{ and } g_x = (z_8 - z_6)$$

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

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

=> *Roberts cross-gradient operators* →

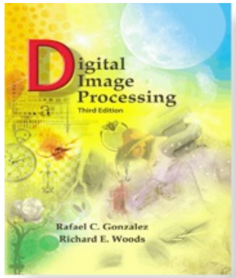
z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

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.



3.6.4 Using First-Order Derivatives for (Nonlinear) Image Sharpening - The Gradient

Approximations to g_x and g_y using a 3x3 neighbourhood centered on z_5:

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

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

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

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

Sobel operators :

$\frac{\partial f}{\partial x}$

−1	−2	−1
0	0	0
1	2	1

Horizontal edges

−1	0	1
−2	0	2
−1	0	1

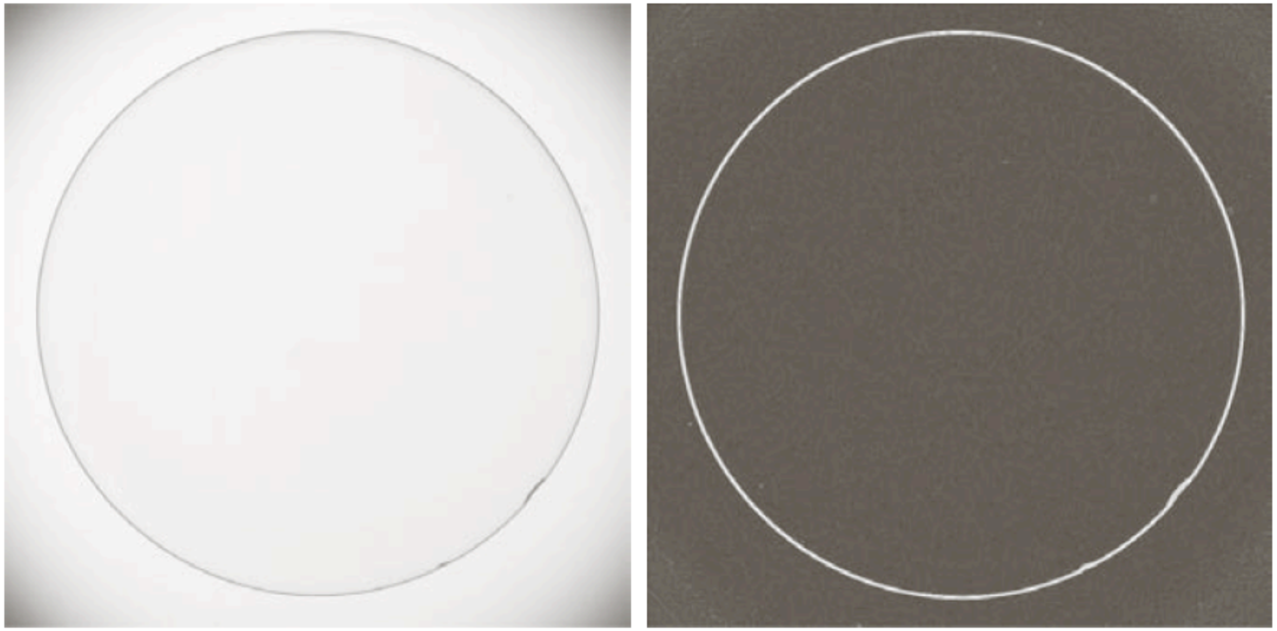
Vertical edges

$\frac{\partial f}{\partial y}$

a
b c
d e

FIGURE 3.41 A 3 × 3 region of an image (the *z*s 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.

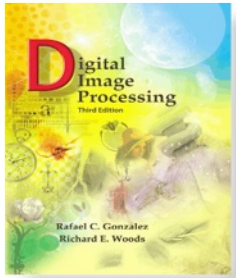
Example:



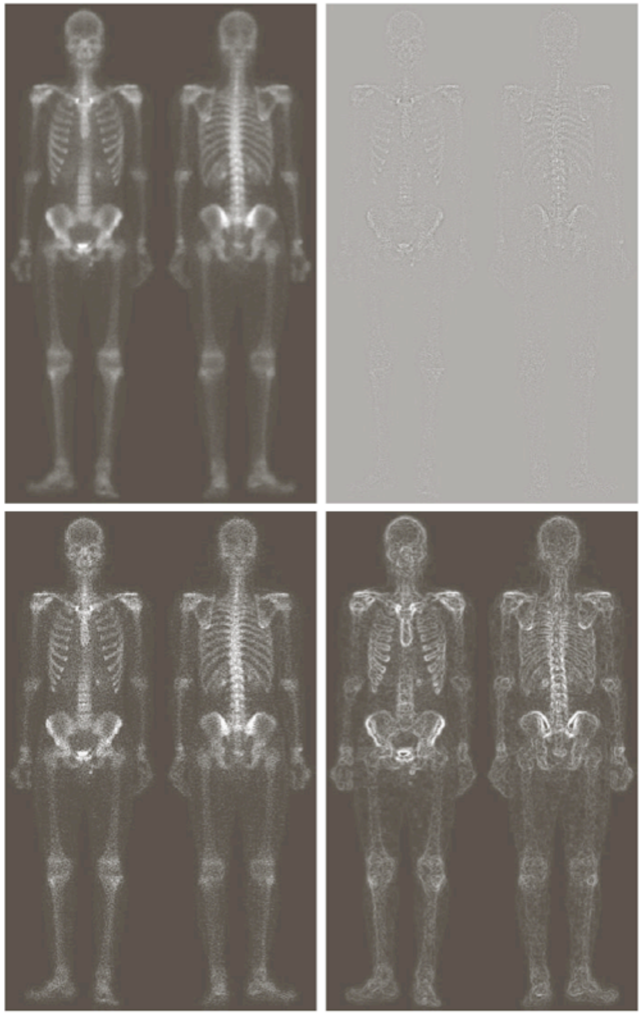
a b

FIGURE 3.42 (a) Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Pete Sites, Perceptics Corporation.)

Sobel image

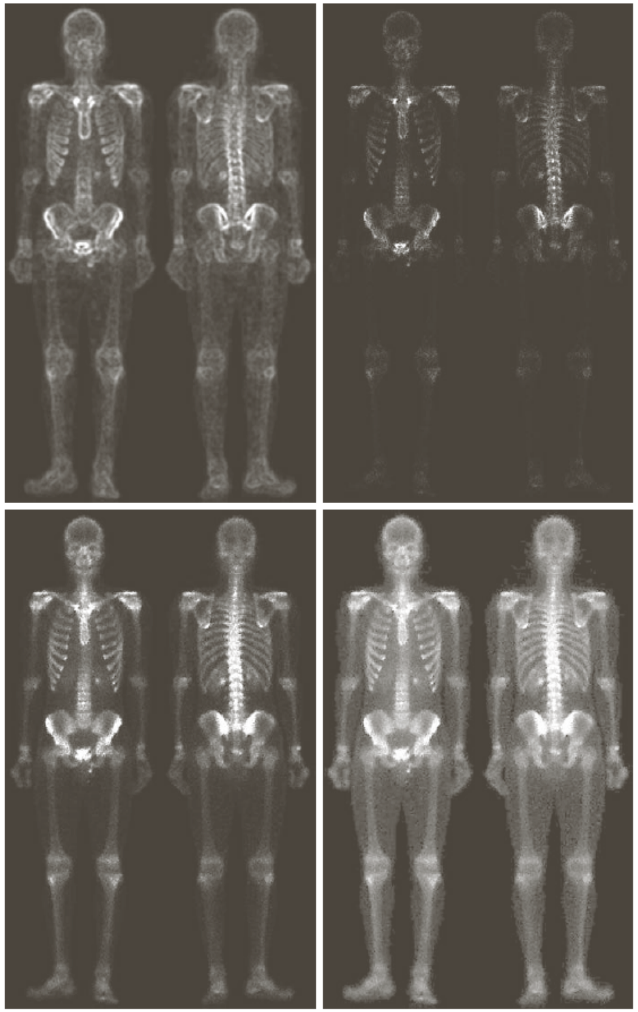


3.7 Combining Spatial Enhancement Methods



a b
c d

FIGURE 3.43 (a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel gradient of (a).



e f
g h

FIGURE 3.43 *(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.)

Chapter 3

Intensity Transformations & Spatial Filtering

References:

- R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 3rd Edition, Prentice Hall, 2008
- D.A. Forsyth and J. Ponce, *Computer Vision – A Modern Approach*, Prentice Hall, 2003
- S. Brès, J.M. Jolion and F. Lebourgeois, *Traitement et Analyse des Images Numériques*, Hermès, 2003