Lecture #5

- Point transformations (cont.)
- Histogram transformations
  - Equalization
  - Specification
  - Local vs. global operations
- Intro to neighborhoods and spatial filtering

Brightness & Contrast

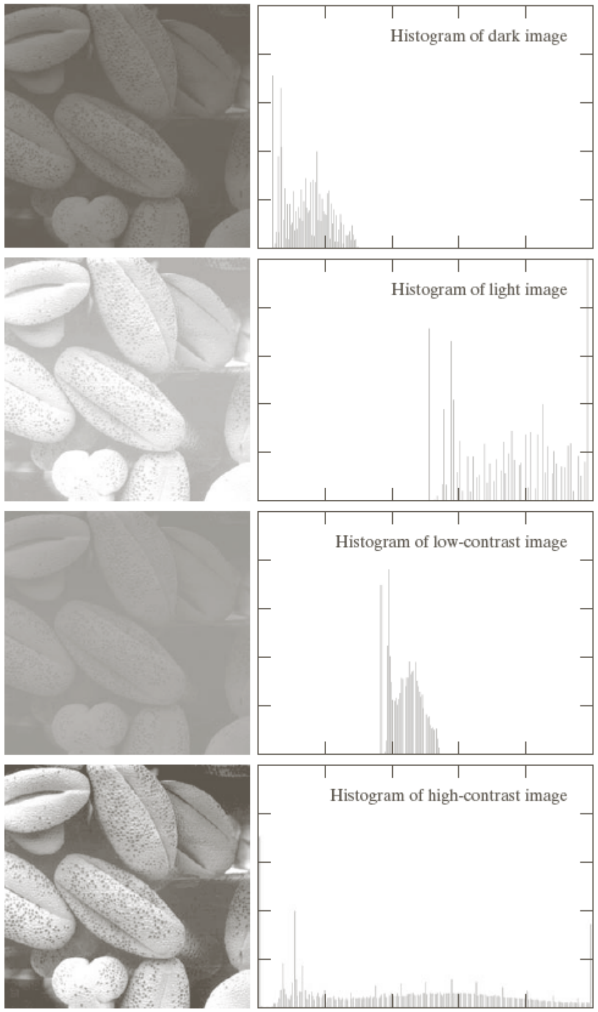
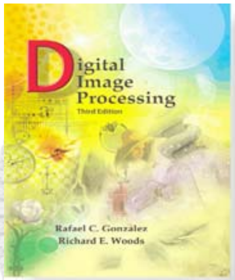
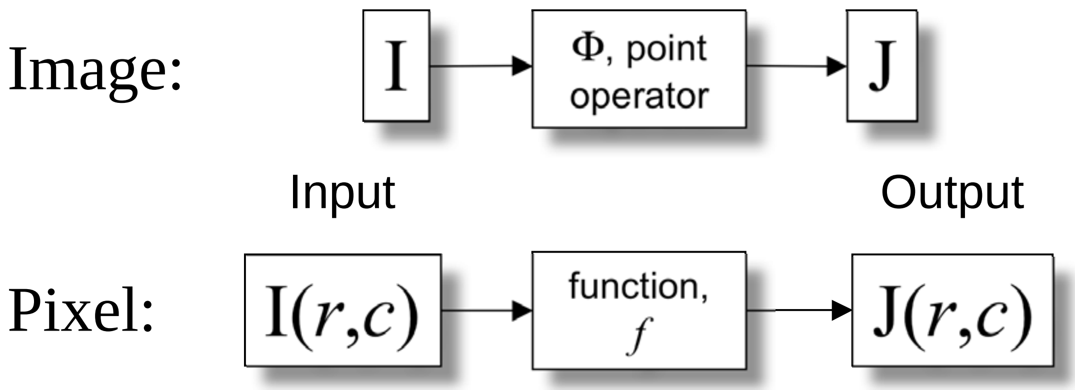


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.



Point Ops via Functional Mappings



$$J = \Phi[I]$$

If $I(r,c) = g$
and $f(g) = k$
then $J(r,c) = k$.

The transformation of image *I* into image *J* is accomplished by replacing each input intensity, *g*, with a specific output intensity, *k*, at every location (*r*,*c*) where *I*(*r*,*c*) = *g*.

The rule that associates *k* with *g* is usually specified with a function, *f*, so that *f* (*g*) = *k*.

Point Ops via Functional Mappings

One-band Image

$$J(r,c) = f\left(I(r,c)\right)$$

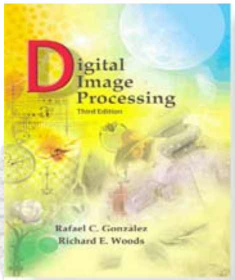
for all pixels locations (r,c)

Three-band Image

$$J(r,c,b) = f\left(I(r,c,b)\right) \text{ or}$$

$$J(r,c,b) = f_b\left(I(r,c,b)\right)$$

for $b = 1,2,3$ and all (r,c)



Point Ops via Functional Mappings

One-band Image

$$J(r,c) = f\big(I(r,c)\big)$$

for all pixels locations (r,c)

Either all 3 bands are mapped through the same function, f, or …

Three-band Image

$$J(r,c,b) = f\big(I(r,c,b)\big) \text{ or}$$

$$J(r,c,b) = f_b\big(I(r,c,b)\big)$$

for $b = 1,2,3$ and all $(r,c$

… each band is mapped through a separate function, f_b.

Point Ops via Functional Mappings

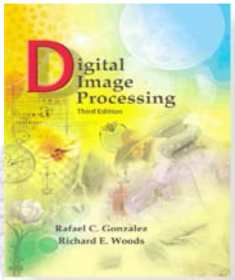
A look-up table (LUT) implements a functional mapping.

If $k = f(g)$,
for $g = 0, \ldots, 255$,
and if k takes on
values in $\{0, \ldots, 255\}$, …

… then the LUT that implements f is a 256x1 array whose $(g+1)^{th}$ value is $k = f(g)$.

To remap a 1-band image, I, to J:

$$J = \text{LUT}(I + 1)$$



Point Ops via Functional Mappings

If I is 3-band, then

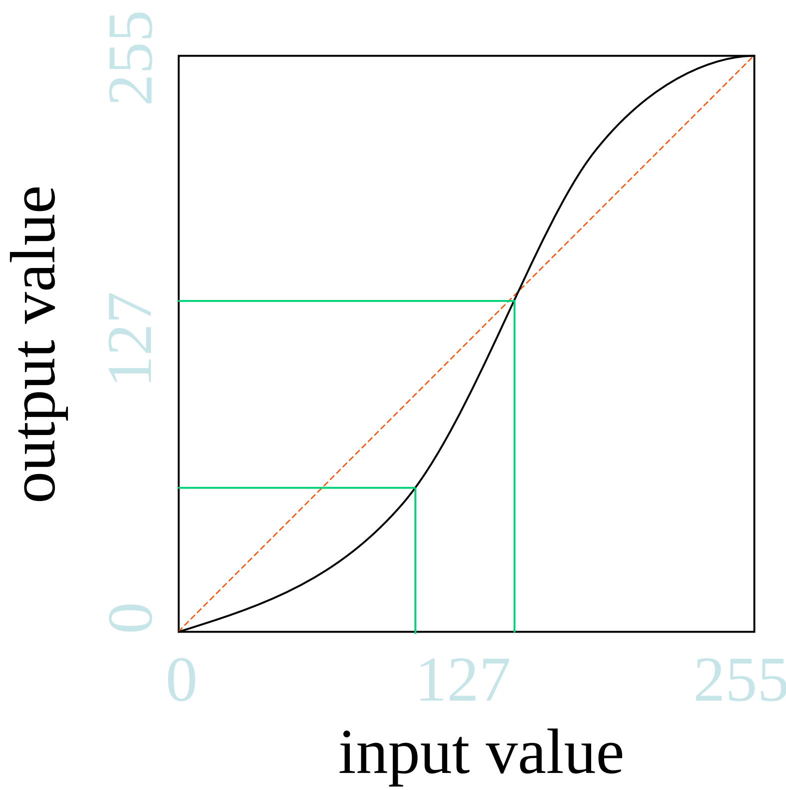
a) each band is mapped separately using the same LUT for each band *or*

b) each band is mapped using different LUTs – one for each band.

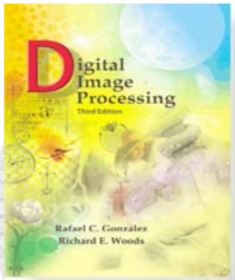
a) $J = \text{LUT}(I+1)$, *or*

b) $J(:,:,b) = \text{LUT}_b\big(I(:,:,b)+1\big)$ for $b = 1,2,3.$

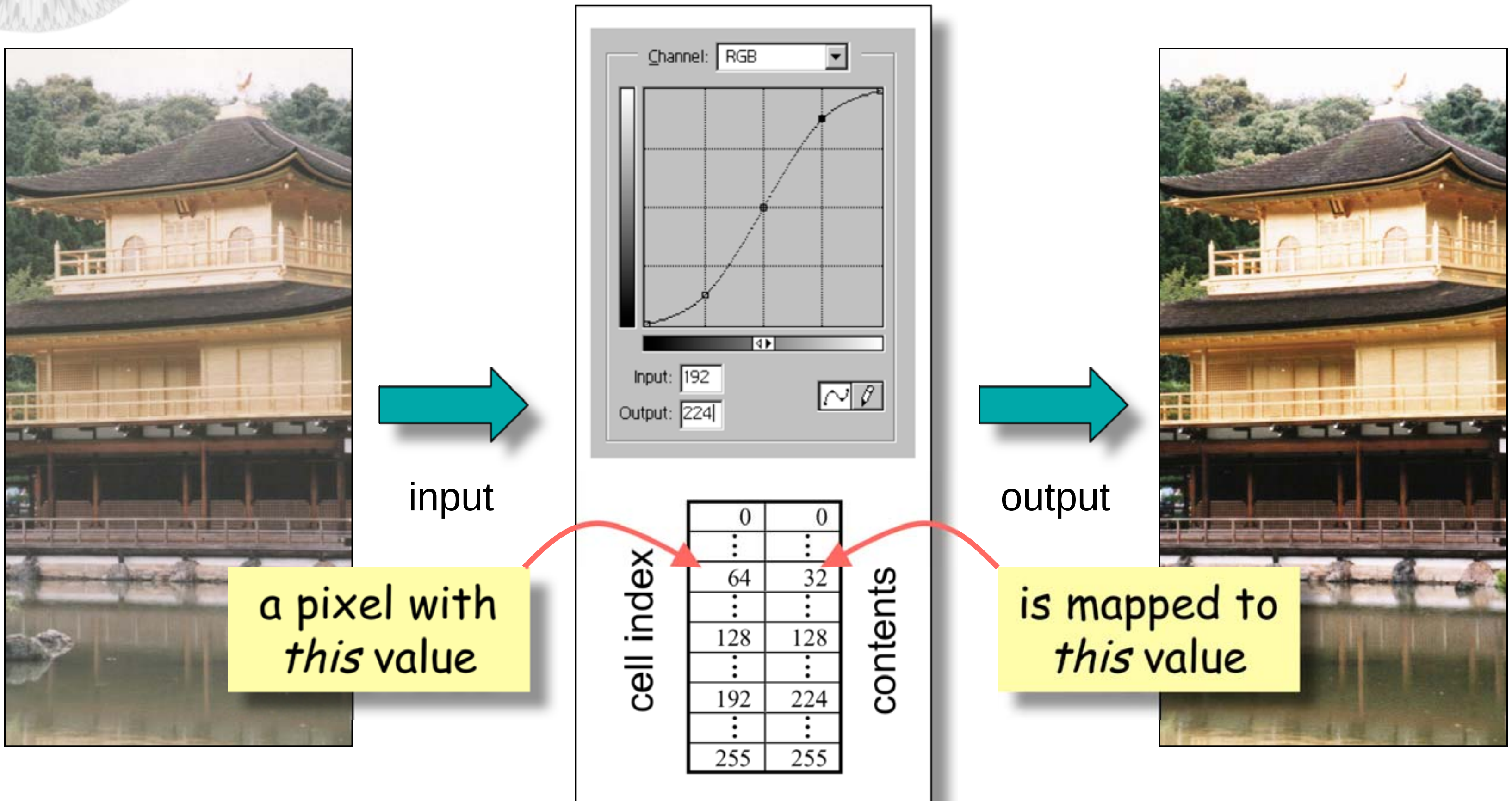
Point Ops via Functional Mappings



E.g.:	index	value
	...	...
	101	64
	102	68
	103	69
	104	70
	105	70
	106	71
	...	...
	input	output



Point Ops via Functional Mappings



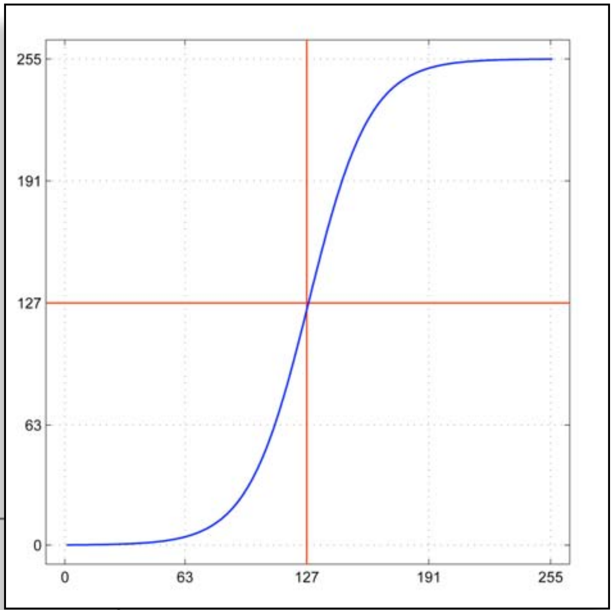
How to Generate a Look-Up Table

For example:

Let $a = 2.$

Let $x \in \{0,\ldots,255\}$

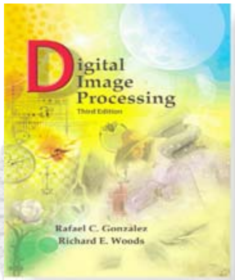
$$\sigma(x;a) = \frac{255}{1 + e^{-a(x-127)/32}}$$



Or in Matlab:

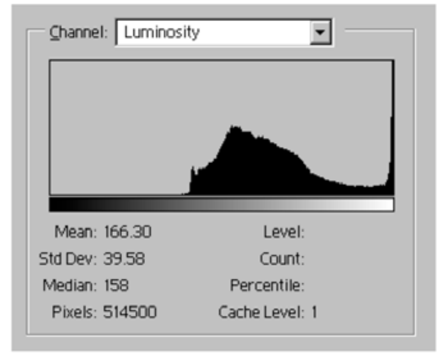
```
a = 2;
x = 0:255;
LUT = 255 ./ (1+exp(-a*(x-127)/32));
```

This is just one example.



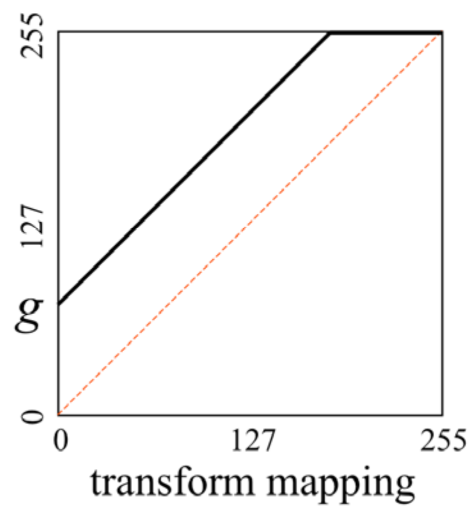
Point Processes: Increase Brightness

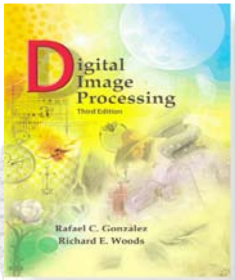
$$I = \frac{R+G+B}{3}$$



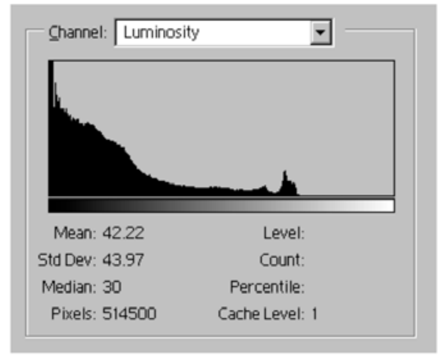
$$J_k(r,c) = \begin{cases} I_k(r,c) + g, & \text{if } I_k(r,c) + g < 256 \\ 255, & \text{if } I_k(r,c) + g > 255 \end{cases}$$

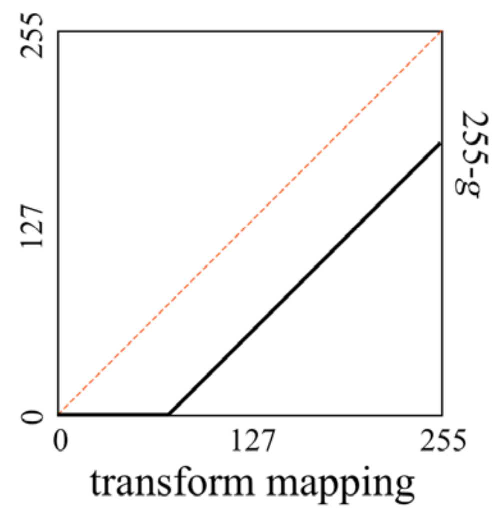
$g \geq 0$ and $k \in \{1,2,3\}$ is the band index.





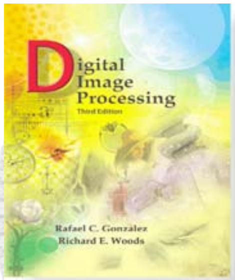
Point Processes: Decrease Brightness



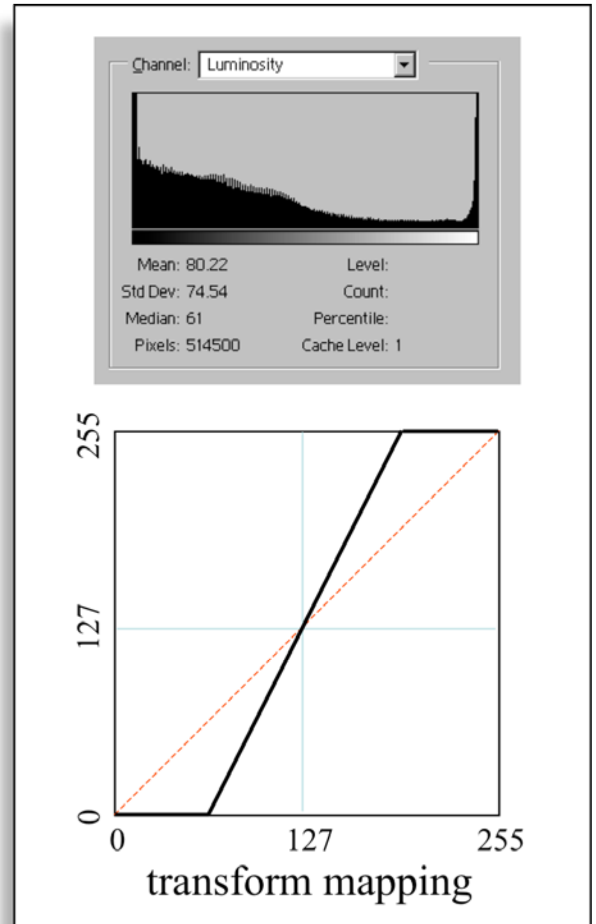


$$J_k(r,c) = \begin{cases} 0, & \text{if } I_k(r,c) - g < 0 \\ I_k(r,c) - g, & \text{if } I_k(r,c) \end{cases}$$

$g \geq 0$ and $k \in \{1,2,3\}$ is the band index.



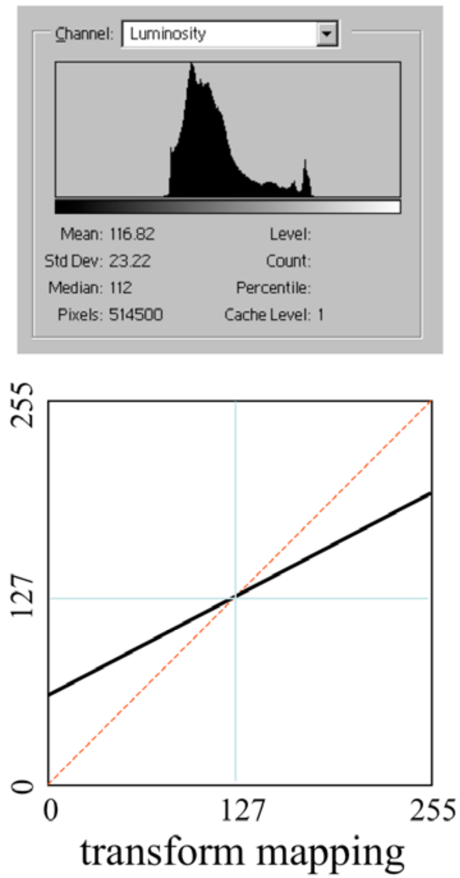
Point Processes: Increase Contrast



Let $T_k(r,c) = a\left[I_k(r,c) - 127\right] + 127$, where $a > 1.0$

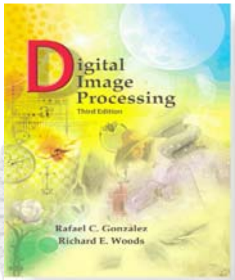
$$J_k(r,c) = \begin{cases} 0, & \text{if } T_k(r,c) < 0, \\ T_k(r,c), & \text{if } 0 \le T_k(r,c) \le 255, \\ 255, & \text{if } T_k(r,c) > 255. \quad k \in \{1,2,3\} \end{cases}$$

Point Processes: Decrease Contrast

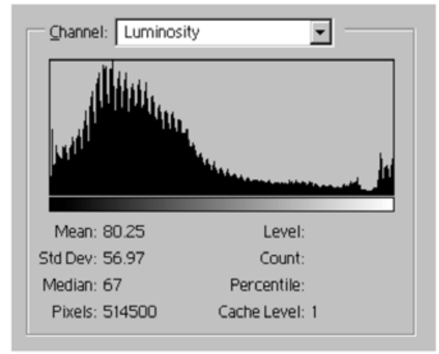


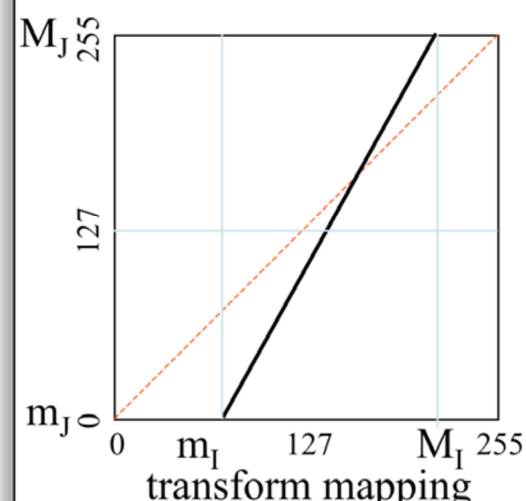
$$T_k(r,c) = a\left[I_k(r,c) - 127\right] + 127,$$

where $0 \le a < 1.0$ and $k \in \{1,2,3\}$.



Point Processes: Contrast Stretch



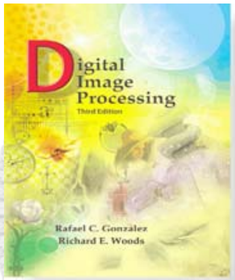


Let $m_I = \min[I(r,c)], \quad M_I = \max[I(r,c)],$

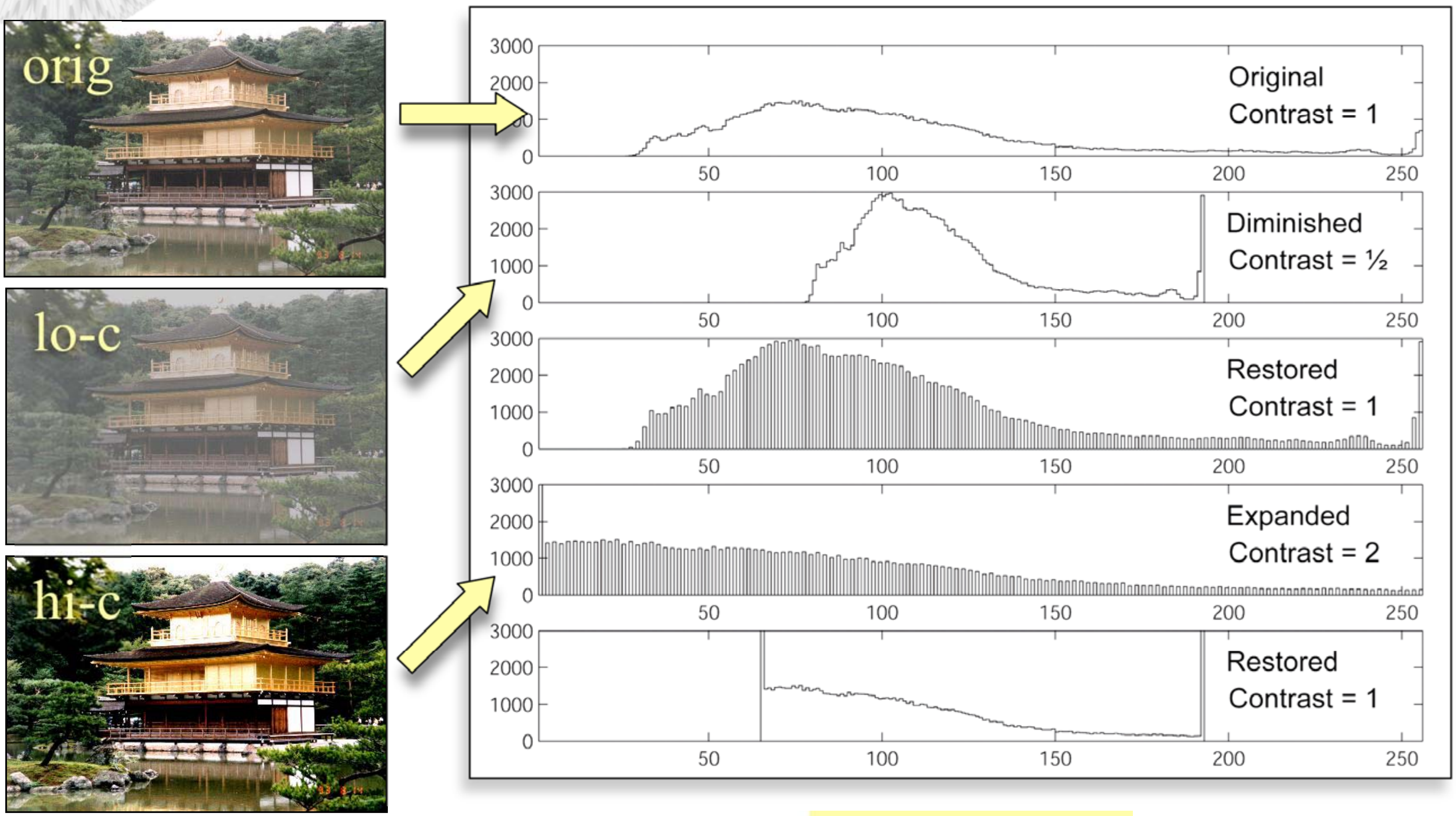
$m_J = \min[J(r,c)], \quad M_J = \max[J(r,c)].$

Then,

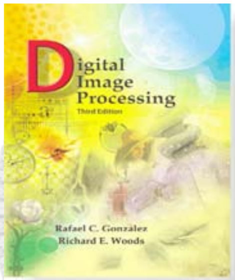
$$J(r,c) = (M_J - m_J)\frac{I(r,c) - m_I}{M_I - m_I} + m_J.$$



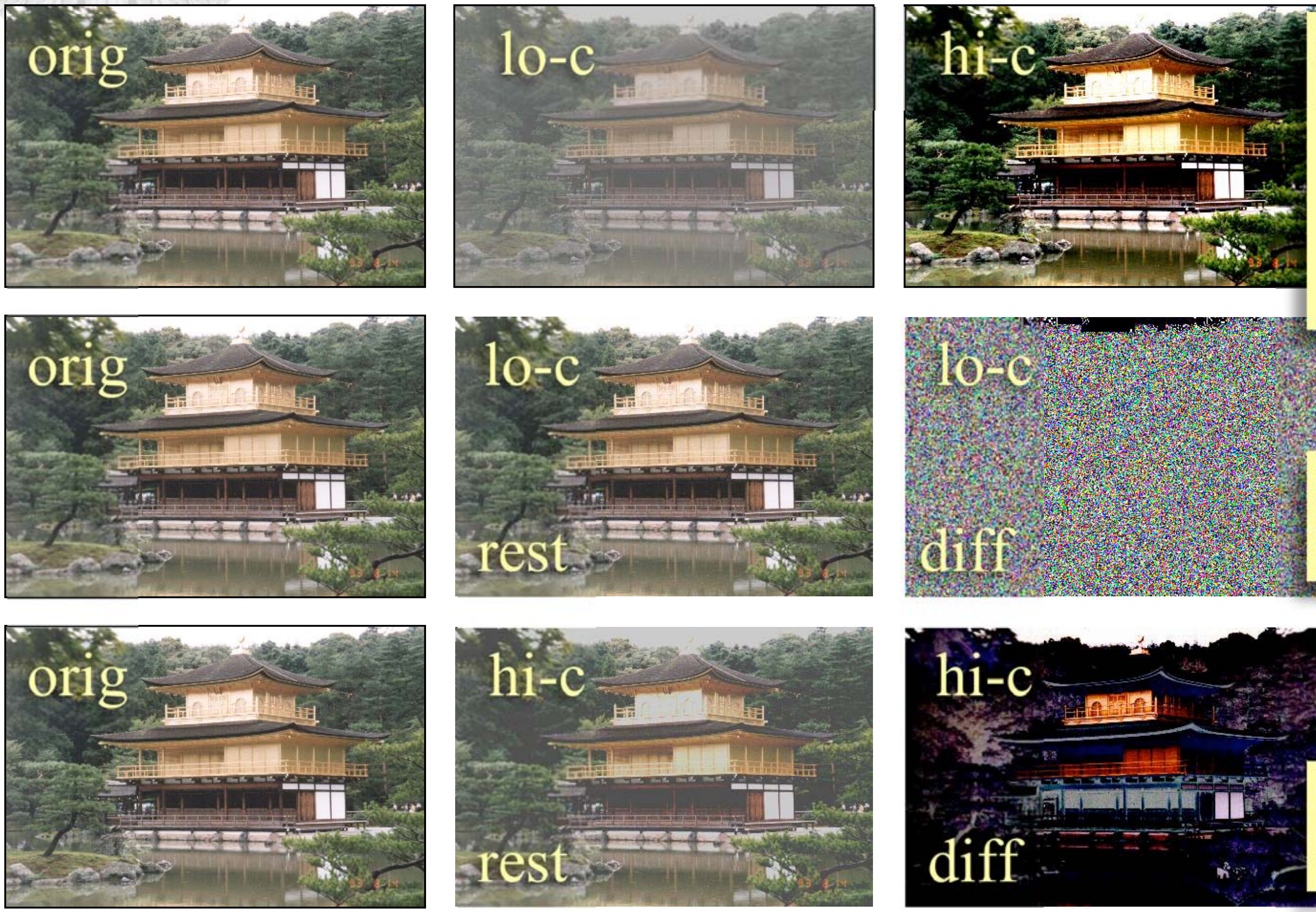
Information Loss from Contrast Adjustment



histograms



Information Loss from Contrast Adjustment

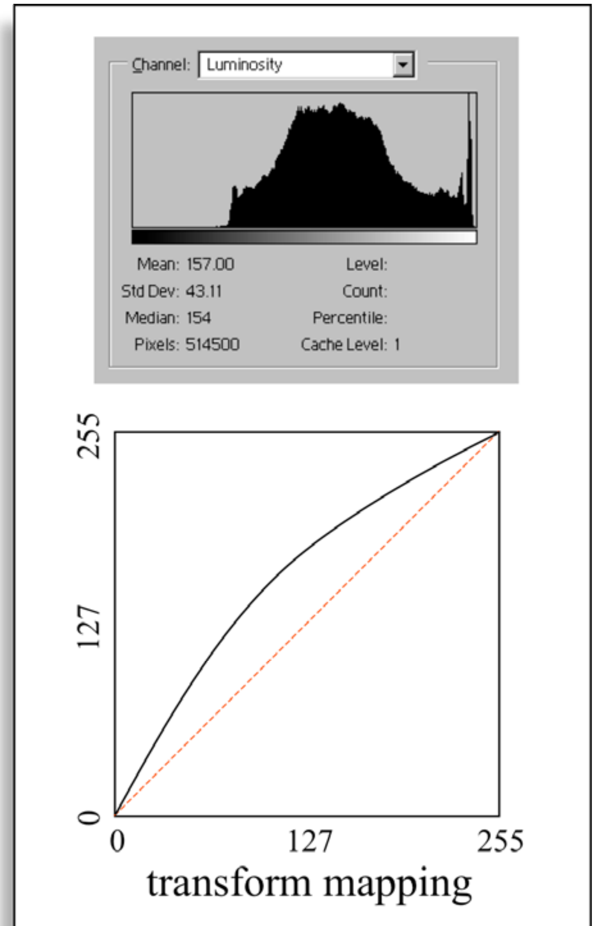


abbreviations:
- original
- low-contrast
- high-contrast
- restored
- difference

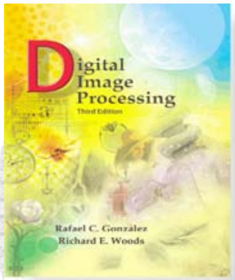
difference between original and restored low-contrast

difference between original and restored high-contrast

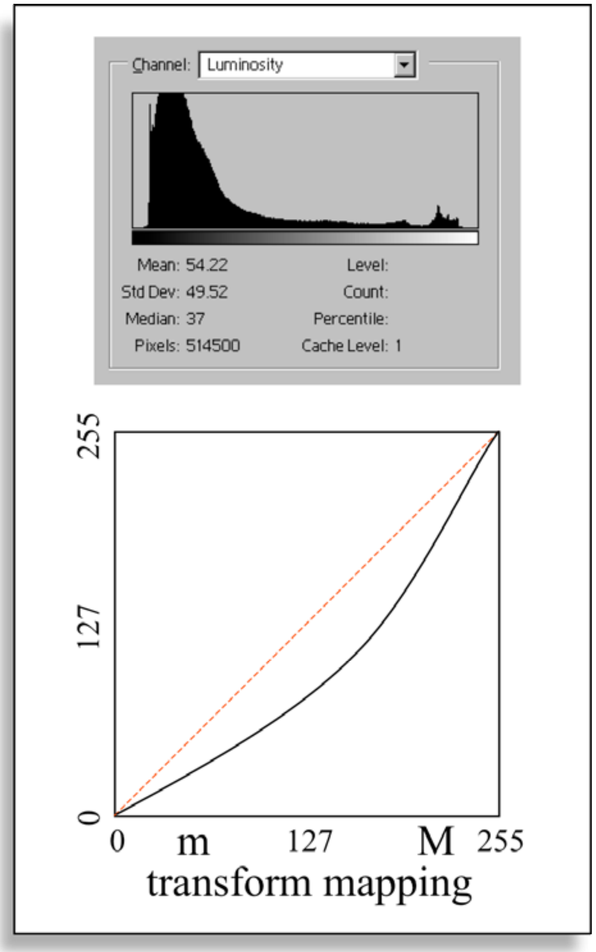
Point Processes: Increased Gamma



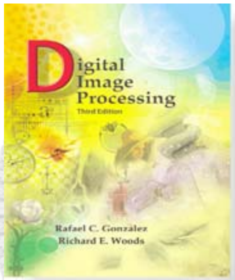
$$J(r,c) = 255 \cdot \left[\frac{I(r,c)}{255} \right]^{1/\gamma} \quad \text{for} \quad \gamma > 1.0$$



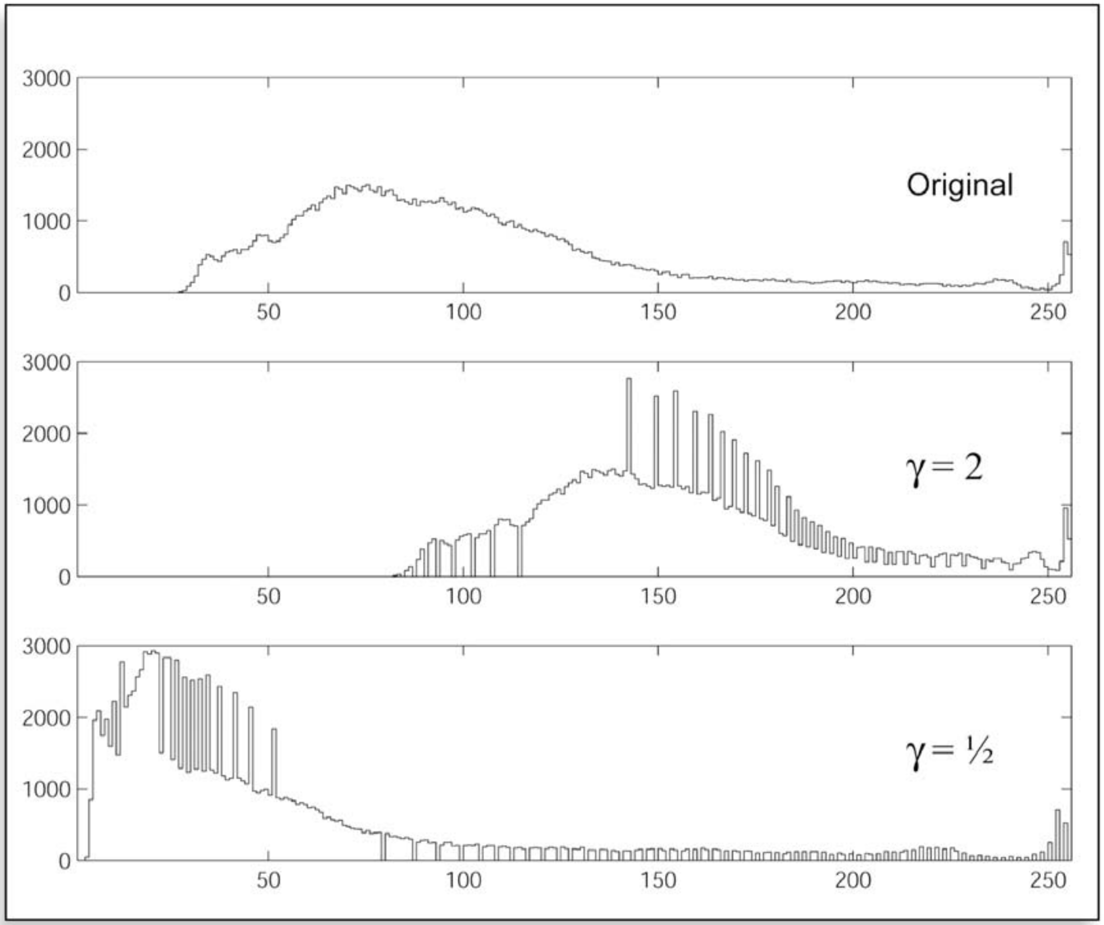
Point Processes: Decreased Gamma



$$J(r,c) = 255 \cdot \left[\frac{I(r,c)}{255} \right]^{1/\gamma} \quad \text{for} \quad \gamma < 1.0$$



Gamma Correction: Effect on Histogram



The Probability Density Function of an Image

Let $A = \sum_{g=0}^{255} h_{I_k}(g+1)$.

Note that since $h_{I_k}(g+1)$ is the number of pixels in I_k (the k th color band of image I) with value g, A is the number of pixels in I. That is if I is R rows by C columns then $A = R \times C$.

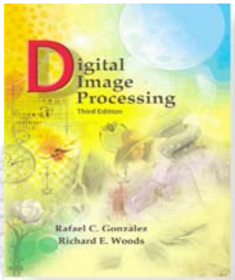
Then,

$$p_{I_k}(g+1) = \frac{1}{A} h_{I_k}(g+1)$$

is the graylevel probability density function of I_k.

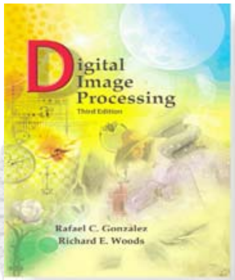
pdf [lower case]

This is the probability that an arbitrary pixel from I_k has value g.



The Probability Density Function of an Image

- $p_{\text{band}}(g+1)$ is the fraction of pixels in (a specific band of) an image that have intensity value g.
- $p_{\text{band}}(g+1)$ is the probability that a pixel randomly selected from the given band has intensity value g.
- Whereas the sum of the histogram $h_{\text{band}}(g+1)$ over all g from 1 to 256 is equal to the number of pixels in the image, the sum of $p_{\text{band}}(g+1)$ over all g is 1.
- p_{band} is the normalized histogram of the band.



The Probability Distribution Function of an Image

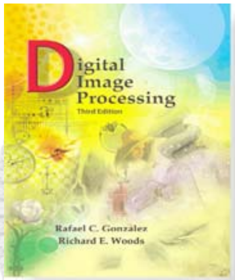
Let $\mathbf{q} = [q_1\ q_2\ q_3] = I(r,c)$ be the value of a randomly selected pixel from I. Let g be a specific gray level. The probability that $q_k \leq g$ is given by

$$P_{I_k}(g+1) = \sum_{\gamma=0}^{g} p_{I_k}(\gamma+1) = \frac{1}{A}\sum_{\gamma=0}^{g} h_{I_k}(\gamma+1) = \frac{\sum_{\gamma=0}^{g} h_{I_k}(\gamma+1)}{\sum_{\gamma=0}^{255} h_{I_k}(\gamma+1)},$$

where $h_{Ik}(\gamma+1)$ is the histogram of the kth band of I.

PDF
[upper case]

This is the probability that any given pixel from I_k has value less than or equal to g.



The Probability Distribution Function of an Image

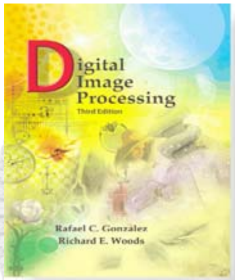
Let $\mathbf{q} = [q_1\ q_2\ q_3] = I(r,c)$ be the value of a randomly selected pixel from I. Let g be a specific gray level. The probability that $q_k \leq$ g is given by

$$P_{I_k}(g+1) = \sum_{\gamma=0}^{g} p_{I_k}(\gamma+1) = \frac{1}{A}\sum_{\gamma=0}^{g} h_{I_k}(\gamma+1) = \frac{\sum_{\gamma=0}^{g} h_{I_k}(\gamma+1)}{\sum_{\gamma=0}^{255} h_{I_k}(\gamma+1)},$$

where $h_{Ik}(\gamma+1)$ is the histogram of the kth band of I.

Also called CDF for "Cumulative Distribution Function".

This is the probability that any given pixel from I_k has value less than or equal to g.

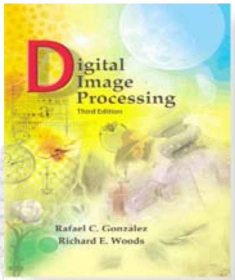


The Probability Distribution Function of an Image

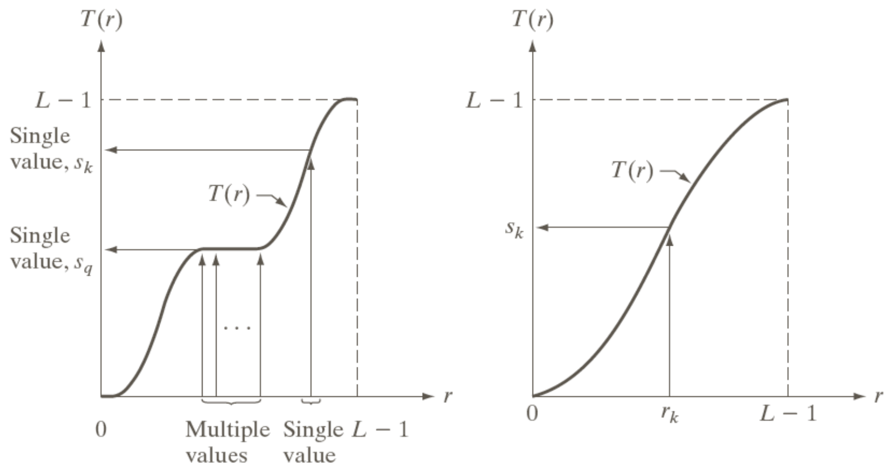
A.k.a. Cumulative Distribution Function.

- $P_{\text{band}}(g+1)$ is the fraction of pixels in (a specific band of) an image that have intensity values less than or equal to g.
- $P_{\text{band}}(g+1)$ is the probability that a pixel randomly selected from the given band has an intensity value less than or equal to g.
- $P_{\text{band}}(g+1)$ is the cumulative (or running) sum of $p_{\text{band}}(g+1)$ from 0 through g inclusive.
- $P_{\text{band}}(1) = \text{p}_{\text{band}}(1)$ and $P_{\text{band}}(256) = 1$; $P_{\text{band}}(g+1)$ is non-decreasing.

Note: the Probability Distribution Function (PDF, capital letters) and the Cumulative Distribution Function (CDF) are exactly the same things. Both PDF and CDF will refer to it. However, pdf (small letters) is the *density* function.



Intensity Transform Requirements

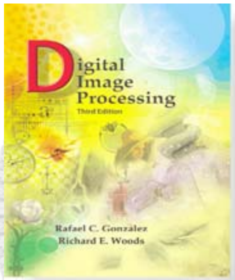


a b

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.

a) A monotonically increasing intensity transformation prevents an intensity reversal which could cause artifacts in the transformed image.

b) A strictly monotonically increasing intensity transformation guarantees that the inverse transformation (from s=T(r) back to r) will be 1:1 preventing ambiguities



Uniform Histogram Transform

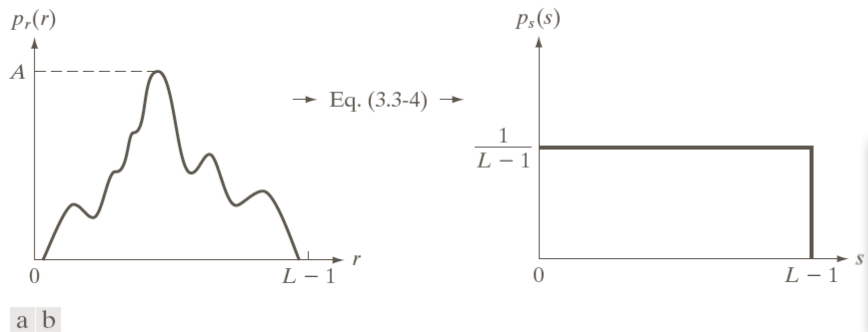


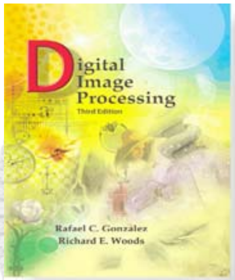
FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

These are pdf's. Gonzalez uses PDF instead of pdf.

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

L is the number of gray levels.

Cumulative Distribution Function (CDF)

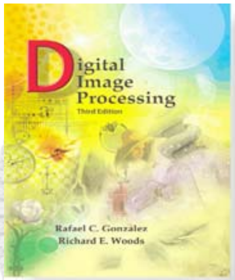


Histogram Equalization Example

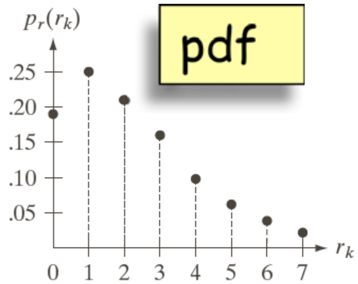
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

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

Note: this same 3-bit (8 gray level) 64x64 pixel image will be used for several examples



Histogram Equalization Example



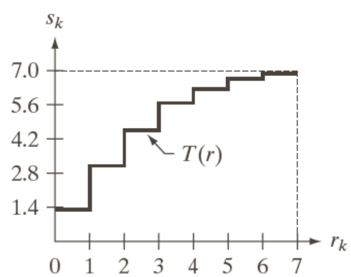
pdf

$$s_0 = T(r_0) = (8-1)\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 7(0.19) = 1.33$$

$$s_1 = T(r_1) = (8-1)\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 7(0.19) + 7(0.25) = 1.33 + 1.75 = 3.08$$

$$s_2 = T(r_2) = (8-1)\sum_{j=0}^{2} p_r(r_j) = 7(0.19) + 7(0.25) + 7(0.21) = 1.33 + 1.75 + 1.47 = 4.55$$

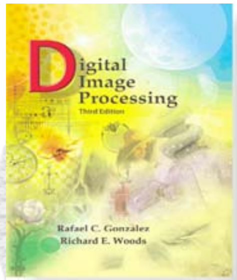
$s_3 = 5.67$ $\quad s_4 = 6.23$ $\quad s_5 = 6.65$ $\quad s_6 = 6.86$ $\quad s_7 = 7.00$



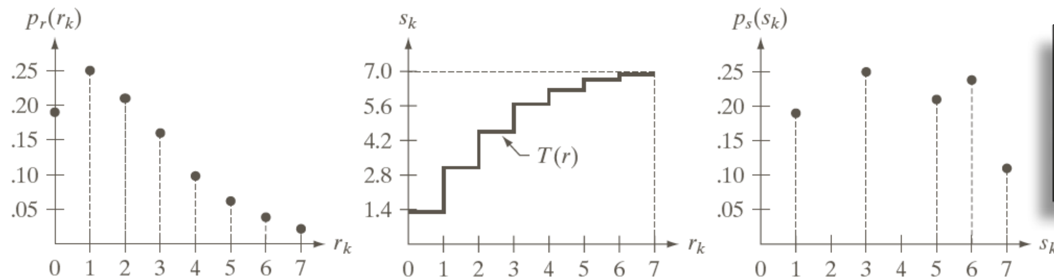
Computed PDF

Discrete CDF (transformation)

Input	Output	Discrete Output
0	1.33	1
1	3.08	3
2	4.55	5
3	5.67	6
4	6.23	6
5	6.65	7
6	6.86	7
7	7.00	7



Histogram Equalization



Resulting histogram when trying to achieve uniform* equalized histogram

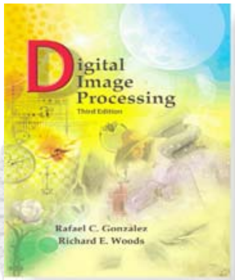
a b c

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

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

Must round off to nearest integer value since only L integer levels in output

*Resulting histogram is usually not uniform due to discrete nature of transform. In this case we only have five output levels.



Histogram Equalization

dark

light

Low contrast

high contrast

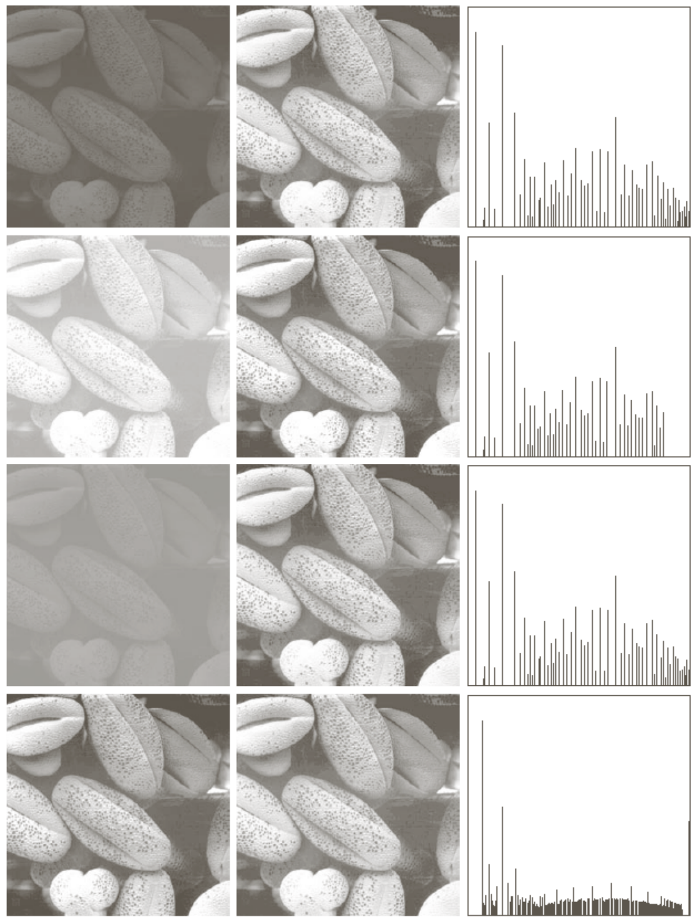


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

Histogram Equalization

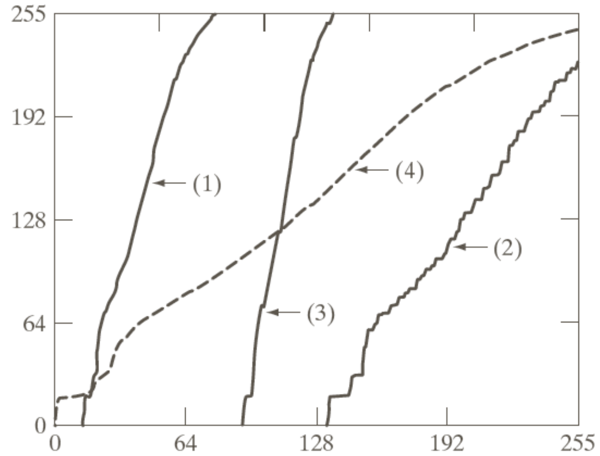
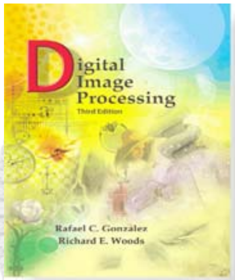


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

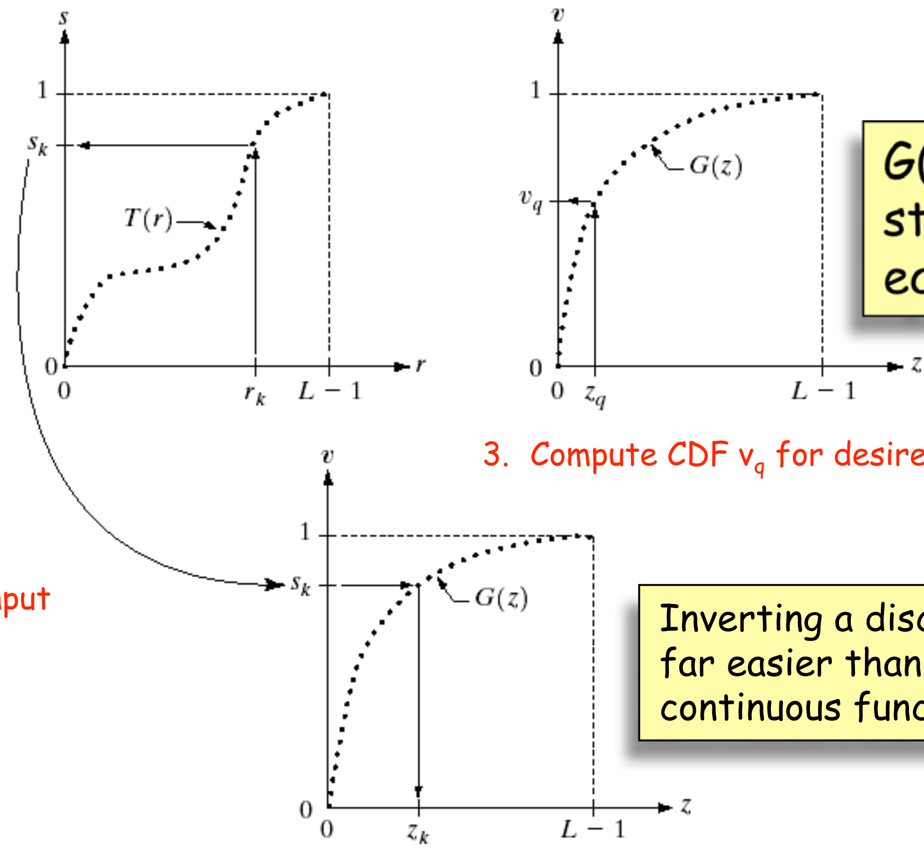
$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$



Histogram Specification

a b
c

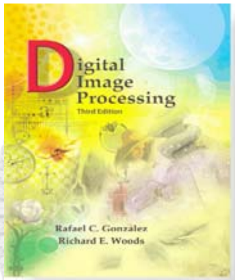
FIGURE 3.19 (a) Graphical interpretation of mapping from r_k to s_k via $T(r)$. (b) Mapping of z_q to its corresponding value v_q via $G(z)$. (c) Inverse mapping from s_k to its corresponding value of z_k.



1. Compute pdf (histogram) for input image
2. Compute CDF s_k for this PDF
3. Compute CDF v_q for desired histogram
4. Match CDF's for each s_k to create mapping table by inverting G

G(z) would be a straight line for equalization.

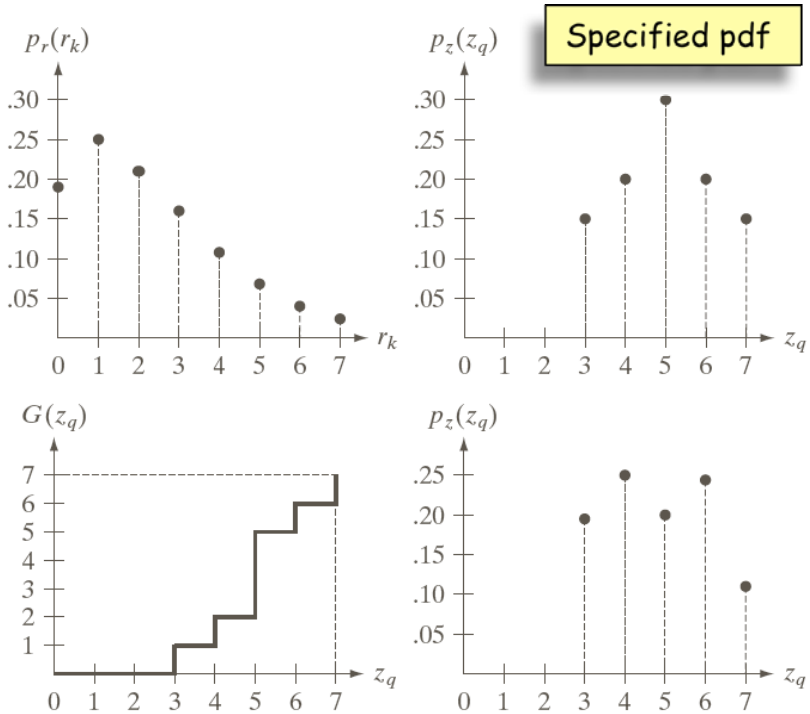
Inverting a discrete function is far easier than inverting a continuous function



Histogram Specification

Original pdf

Specified pdf



a b
c d

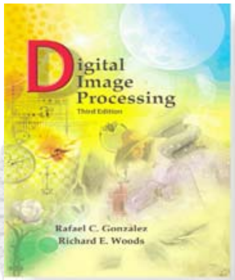
FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

CDF of specified pdf

Actual transformed image pdf

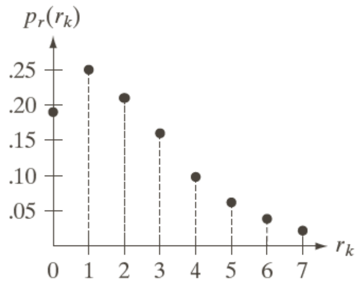
z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

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).



Histogram Specification Example

1. Compute the PDF of the image to be transformed

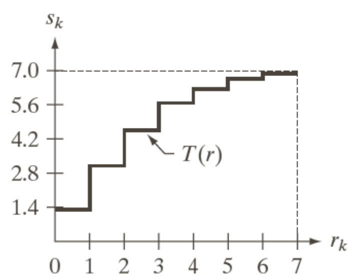


$$s_0 = T(r_0) = (8-1)\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 7(0.19) = 1.33$$

$$s_1 = T(r_1) = (8-1)\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 7(0.19) + 7(0.25) = 1.33 + 1.75 = 3.08$$

$$s_2 = T(r_2) = (8-1)\sum_{j=0}^{2} p_r(r_j) = 7(0.19) + 7(0.25) + 7(0.21) = 1.33 + 1.75 + 1.47 = 4.55$$

$s_3 = 5.67$ $s_4 = 6.23$ $s_5 = 6.65$ $s_6 = 6.86$ $s_7 = 7.00$

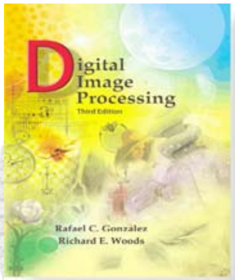


Output CDF (not discrete)

Discrete CDF of input histogram

Input	Output	Discrete Output
0	1.33	1
1	3.08	3
2	4.55	5
3	5.67	6
4	6.23	6
5	6.65	7
6	6.86	7
7	7.00	7

Rounded off



Histogram Specification Example

2. Compute the PDF for the specified histogram

$$G(z_0) = (8-1)\sum_{j=0}^{0} p_z(z_j) = 7p_z(z_0) = 7(0) = 0$$

$$G(z_1) = (8-1)\sum_{j=0}^{1} p_z(z_j) = 7p_z(z_0) + 7p_z(z_1) = 7(0) + 7(0) = 0$$

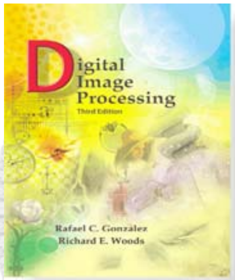
$$G(z_2) = (8-1)\sum_{j=0}^{2} p_z(z_j) = 7(0) + 7(0) + 7(0) = 0$$

$$G(z_3) = 1.05 \quad G(z_4) = 2.45 \quad G(z_5) = 4.55 \quad G(z_6) = 5.95 \quad G(z_7) = 7.00$$

Discrete CDF of desired histogram

Input	Output	Discrete Output
0	0	0
1	0	0
2	0	0
3	1.05	1
4	2.45	2
5	4.55	5
6	5.95	6
7	7.00	7

Rounded off



Histogram Specification Example

3. Match the PDFs to get the transformation

Discrete CDF of input histogram

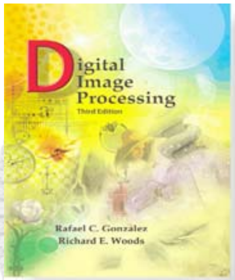
Input	CDF	Discrete CDF
0	1.33	1
1	3.08	3
2	4.55	5
3	5.67	6
4	6.23	6
5	6.65	7
6	6.86	7
7	7.00	7

Discrete CDF desired

Pixel	CDF	Discrete CDF
0	0	0
1	0	0
2	0	0
3	1.05	1
4	2.45	2
5	4.55	5
6	5.95	6
7	7.00	7

Discrete CDF of input histogram

Input	CDF input	CDF desired	New Value
0	1 (1.33)	1 (1.05)	3
1	3 (3.08)	2 (2.45)	4
2	5 (4.55)	5 (4.55)	5
3	6 (5.67)	6 (5.95)	6
4	6 (6.23)	6 (5.95)	6
5	7 (6.65)	7 (7.00)	7
6	7 (6.86)	7 (7.00)	7
7	7 (7.00)	7 (7.00)	7



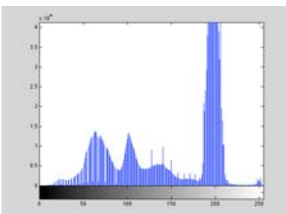
Histogram Specification Example

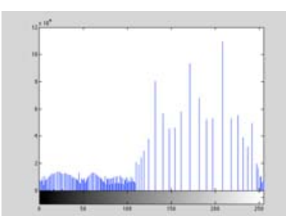
4. Put mapping in a LUT

Input		Transformed Value
0	-->	3
1	-->	4
2	-->	5
3	-->	6
4	-->	6
5	-->	7
6	-->	7
7	-->	7

MATLAB: Histogram Equalization

HISTOGRAM EQUALIZATION IN MATLAB





```
>> f=imread('fig10.10(a).jpg');  % load in figure 3.15(a)
>> imshow(f)                     % show image in a window
>> figure, imhist(f)             % show histogram in a new window
                                 % This histogram is not normalized.
>> ylim('auto')                  % set histogram tick marks and
                                 % axis limits automatically
>> g=histeq(f,256);              % Create histogram equalized image
                                 % you can also do this with
                                 % the cumsum function
>> hnorm=imhist(f)./numel(f);    % compute normalized histogram
>> figure, imshow(g)             % show this figure in a new window
>> figure, imhist(g)             % generate a histogram of
                                 % the equalized image
>> ylim('auto')                 % set limits again
```

SEE GWE, Section 3.3.3 for a discussion of histogram specification using MATLAB

MATLAB: Useful Transformations

THRESHOLDING

```
>> f=imread('fig10.10(a).jpg');     % load in an image
>> g=(f>100)*255                    % show figure in a new window
```

(f>100) evaluates to 1 (true) if the pixel is >100, or 0 (false) if the pixel is ≤100

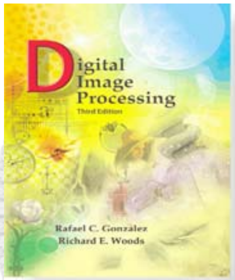
Multiplying by 255 is necessary to have a uint8 255 level image.

ROTATION

bilinear

```
>> f=imread('fig10.10(a).jpg');     % load in an image
>> g=imrotate(f,angle,'method');    % rotate it by angle
```

%Supports methods: 'nearest', 'bilinear', 'bicubic'



MATLAB: Getting Picture Coordinates

INPUT CURSOR VALUES

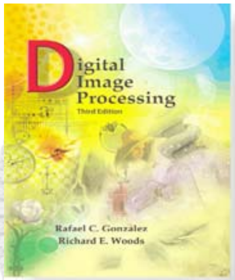
```
%(Very useful for specifying reference points in an image)
>> [x,y]=ginput
%Displays the graph window, displays a cross hair.  Will input
%coordinate pairs from the graph window until you type return.
%Position cursor using mouse and click to input each coordinate pair.

x =
  778.3362
  619.7249
  844.4242
  465.5196

y =
  213.0831
  327.6357
  470.8264
  609.6112

>> [x,y]=ginput(n)
```

Will input n coordinate pairs until you press return



Examples of Point Processing

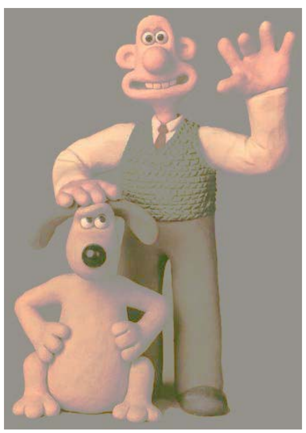
- gamma

- brightness

original

+ brightness

+ gamma



histogram mod

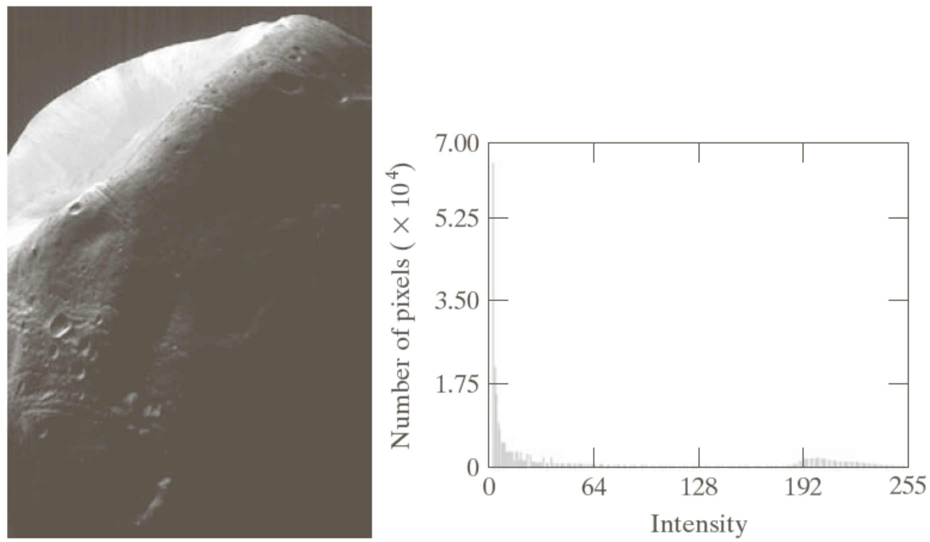
- contrast

original

+ contrast

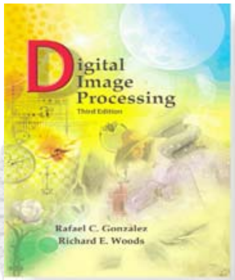
histogram EQ

Histogram

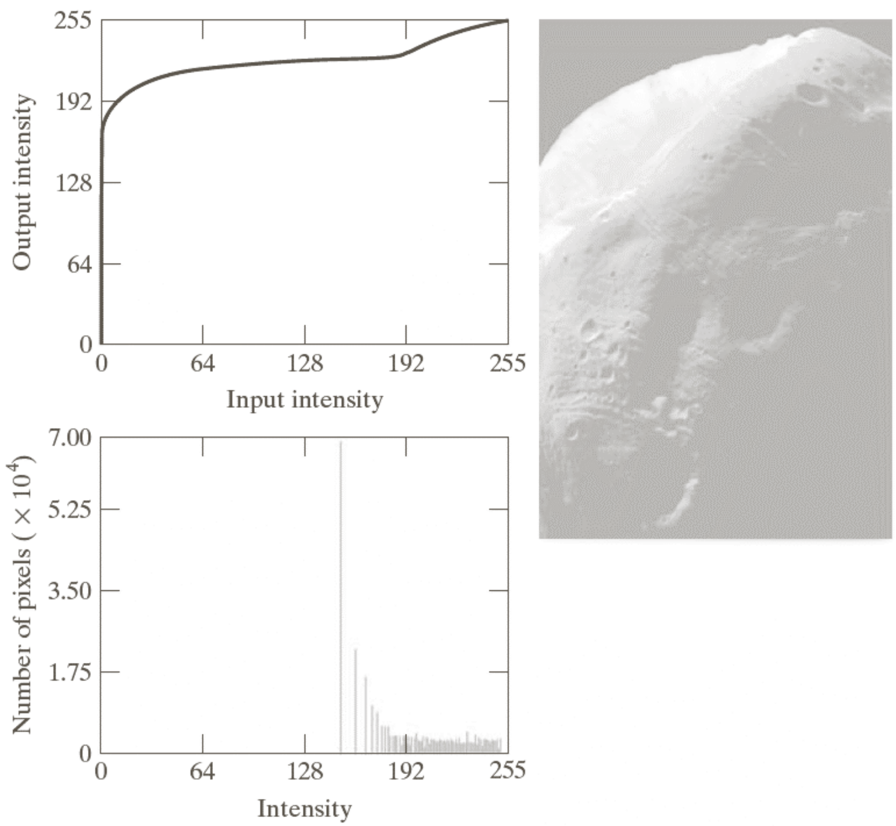


a b

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



Histogram Equalization



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).

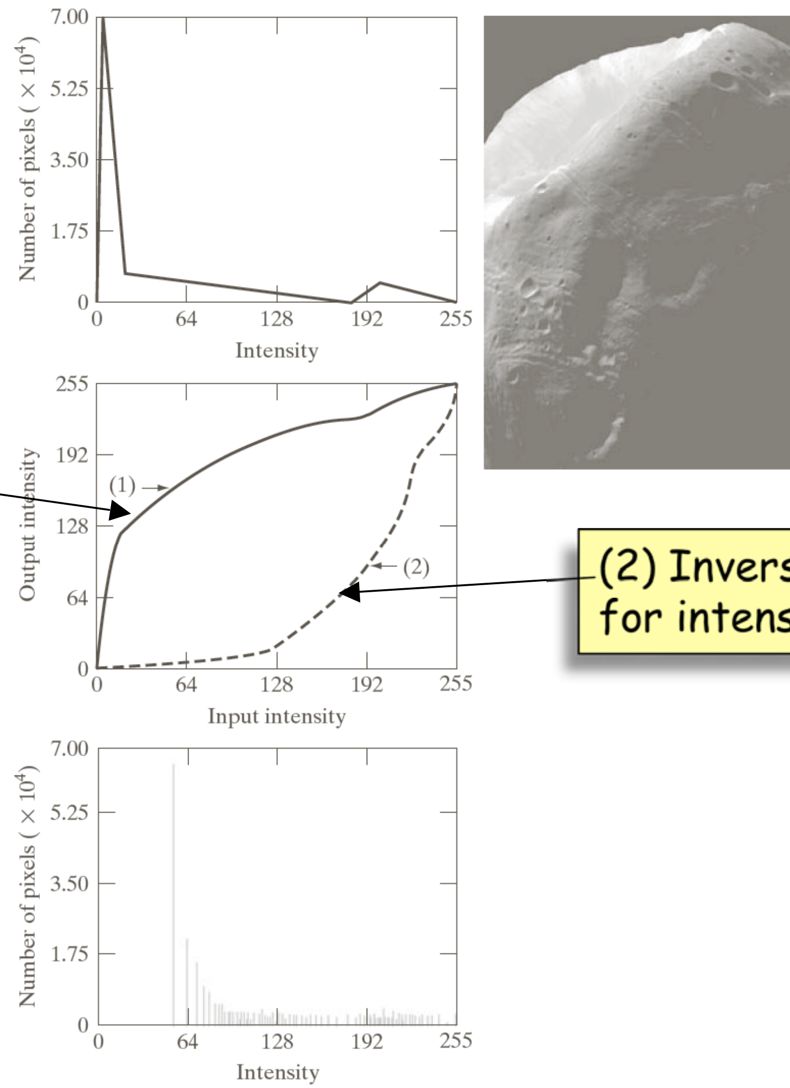
Histogram Specification

Manually specified histogram h

(1) CDF $s_k=G(z_q)$ corresponding to h

(2) Inverse transform $z=G^{-1}(s_k)$ for intensity mapping

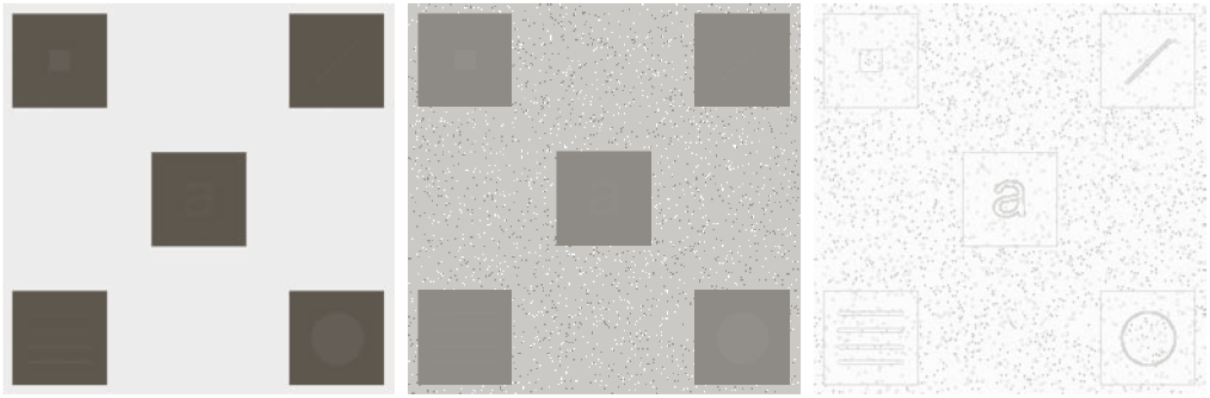
Resulting histogram



a c
b
d

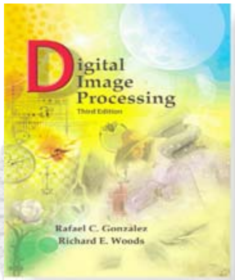
FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

Local Histogram Operations

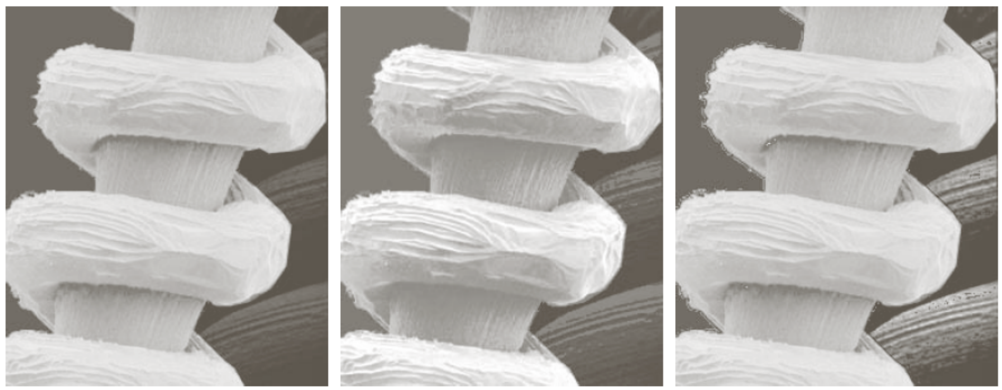


a b c

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.



Local Histogram Operations



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.)

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

compare to global mean m_G and standard deviation σ_G

Spatial Neighborhoods

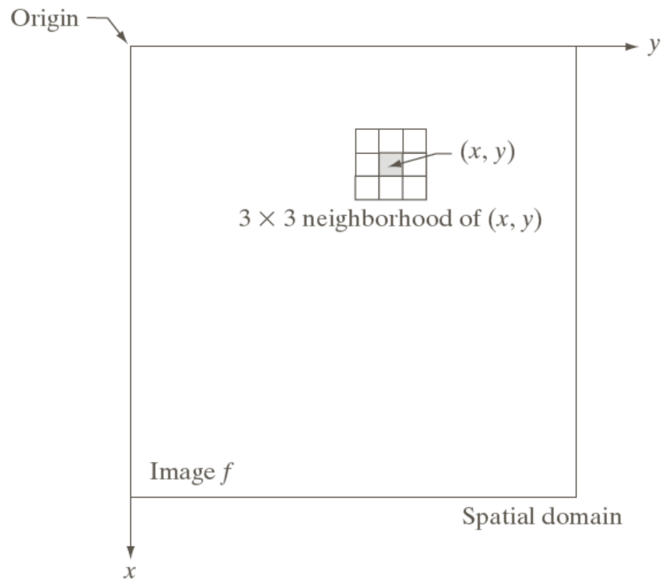


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

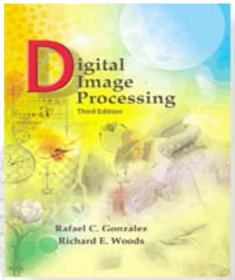
Spatial Filtering

Let I and J be images such that $J = T[I]$.

$T[\cdot]$ represents a transformation, such that,

$$J(r,c) = T[I](r,c) =$$
$$f\left(\left\{I(u,v)\,\middle|\, u \in \{r-s,\ldots,r,\ldots r+s\},\, v \in \{c-d,\ldots,c,\ldots c+d\}\right\}\right)$$

That is, the value of the transformed image, J, at pixel location (r,c) is a function of the values of the original image, I, in a $2s+1 \times 2d+1$ rectangular neighborhood centered on pixel location (r,c).



Moving Windows

- The value, $J(r,c) = \mathrm{T}[I](r,c)$, is a function of a rectangular neighborhood centered on pixel location (r,c) in I.
- There is a different neighborhood for each pixel location, but if the dimensions of the neighbor-hood are the same for each location, then trans-form T is sometimes called a *moving window transform.*

Spatial Filtering

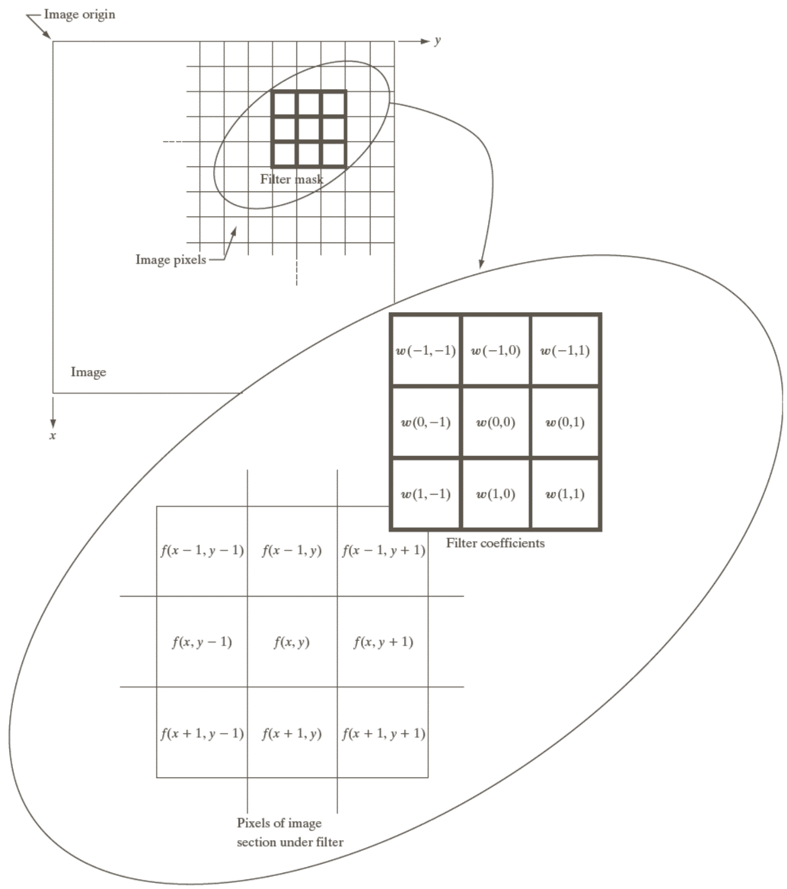
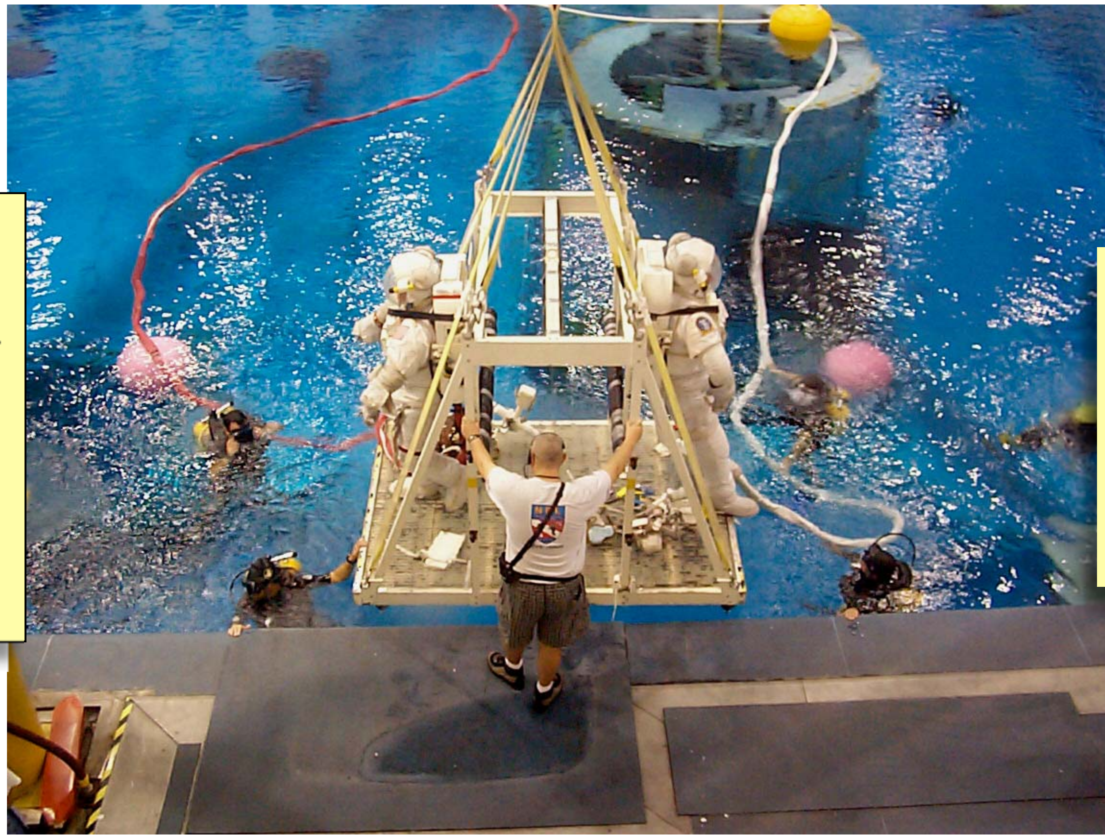


FIGURE 3.28 The mechanics of linear spatial filtering using a 3×3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

Moving-Window Transformations

Neutral Buoyancy Facility at NASA Johnson Space Center



We'll take a section of this image to demonstrate the MWT

photo: R.A.Peters II, 1999