



## Lecture 3 Image Enhancement

Instructor: Yiming Xiao

*Email:* [yiming.xiao@concordia.ca](mailto:yiming.xiao@concordia.ca)

Department of Computer Science and Software Engineering  
Concordia University

*Slides modified from materials provided by Dr. Tien D Bui*

**Image Enhancement:** improving the quality of images (make it better) or modifying image to bring out hidden features. Two kinds: *spatial domain* and *frequency domain*.

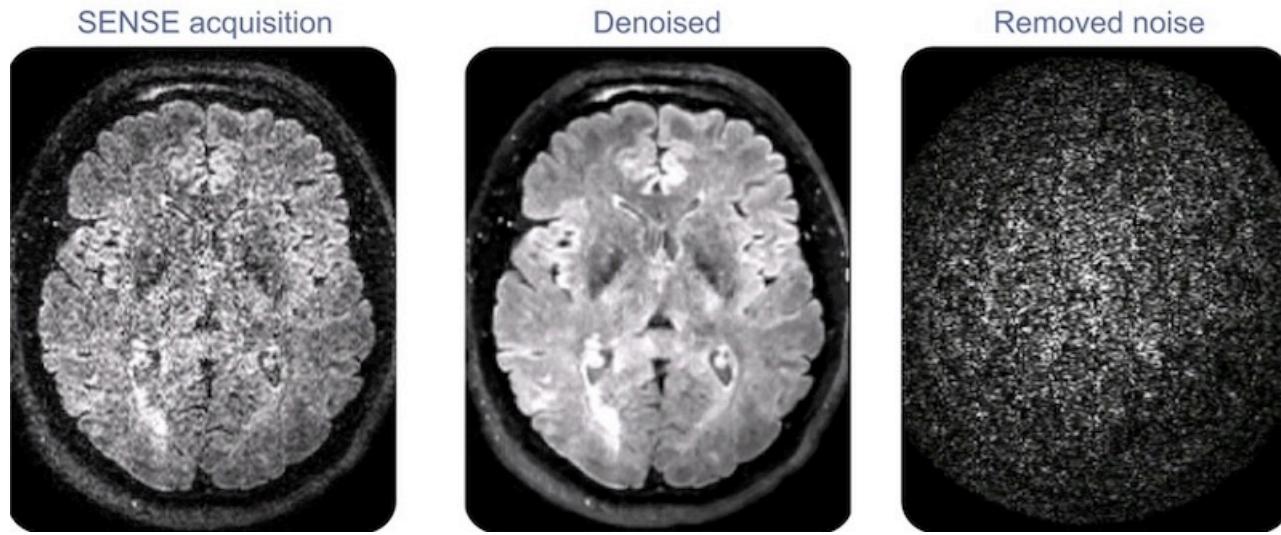
**Image Restoration:** correcting images subjected to noise, blurs, distortions, atmospheric effects, etc

## Denoising

## Histogram equalization



<https://pyimagesearch.com/>



## Restoration

## Enhancement

J. V. Manjon, et al. Journal of Magnetic Resonance Imaging, 2010.

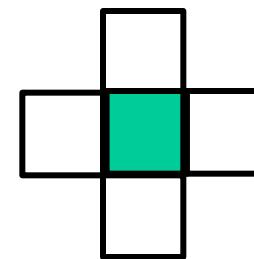
# **Image Enhancement I: Pixel-wise Operations**

# Image Enhancement: Operating on the Pixels

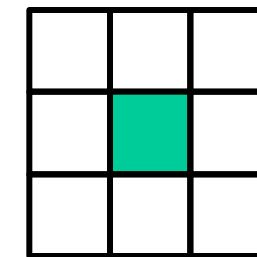
- Make image “better” for a specific application
  - The idea of “better” is somewhat subjective
- We distinguish two domains:
  - Spatial or Pixel domain:  $f(x, y)$  or  $f(m, n)$
  - Frequency Domain:  $F(w_x, w_y)$  or  $F(u, v)$
- For this section: Pixel Domain
  - Operations on single pixel at a time
  - Operations on groups of pixels (neighborhoods)



Pixel

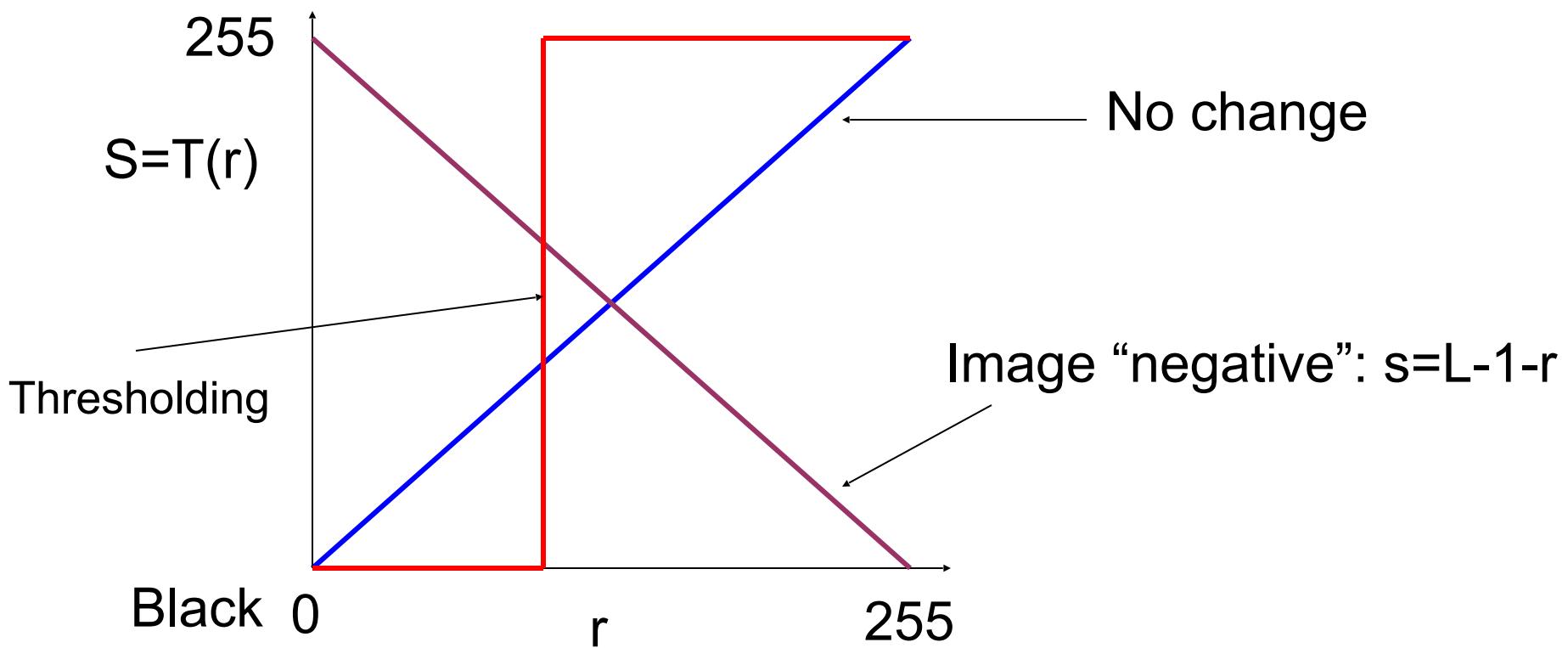
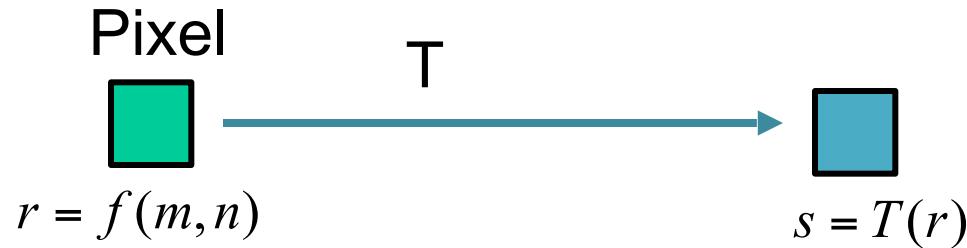


4-Neighbors

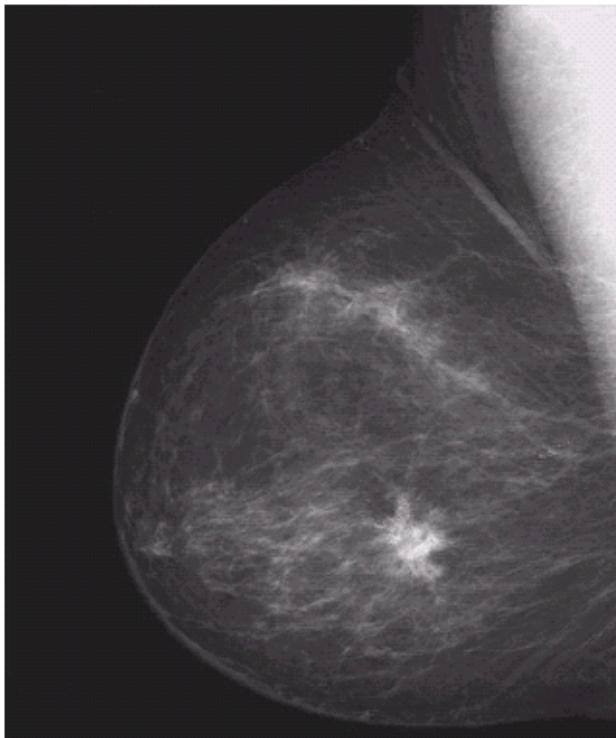


8-Neighbors

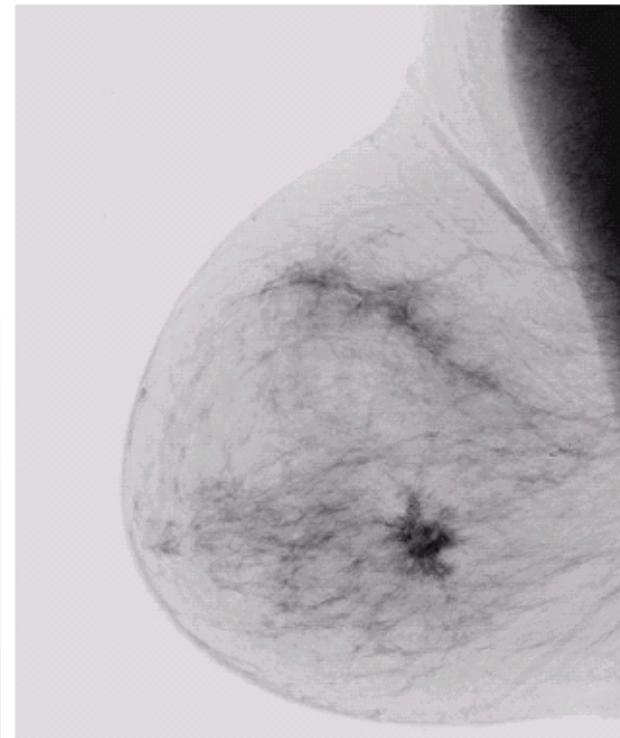
# Simplest form of processing: Point Processing



## Negative of an image



Original



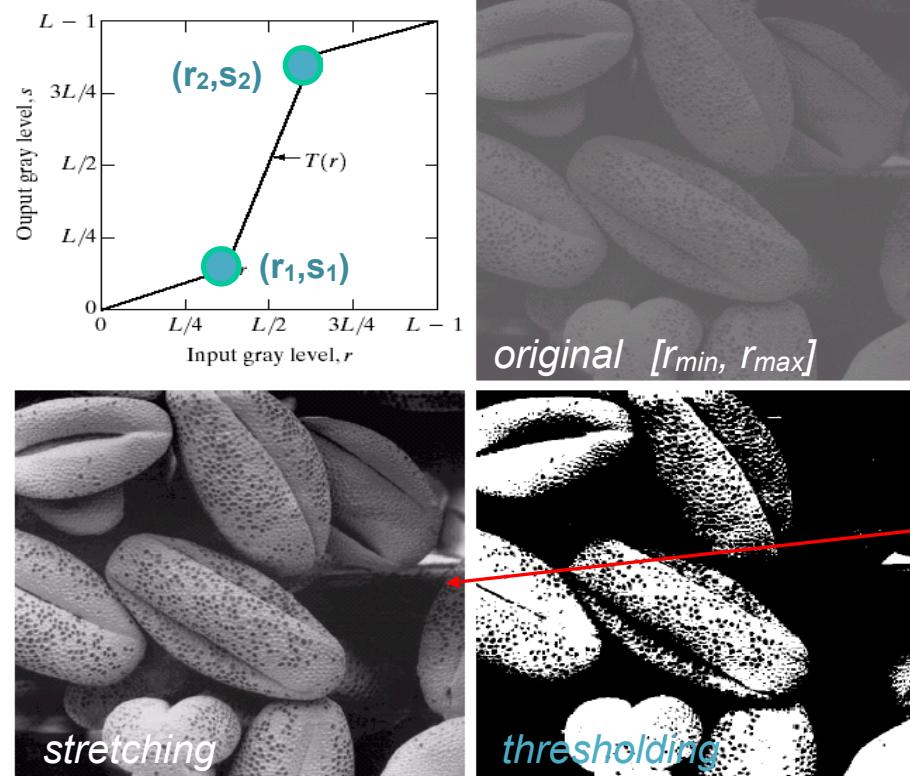
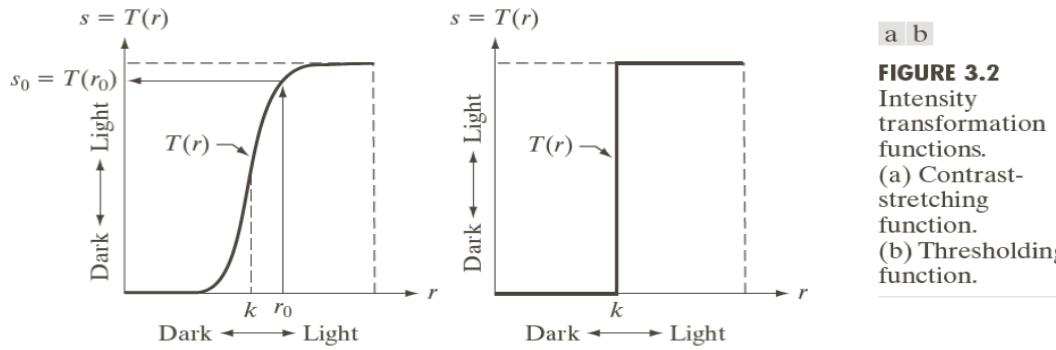
Negative

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.)

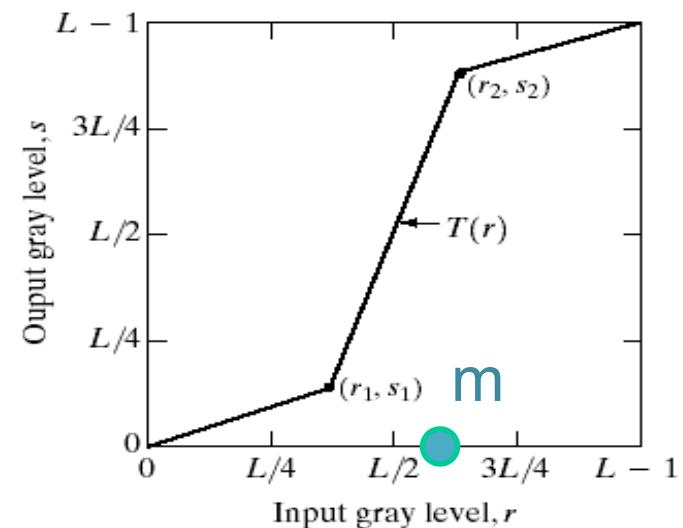
# Intensity Transformation: Contrast Stretching



8-bit image, low contrast

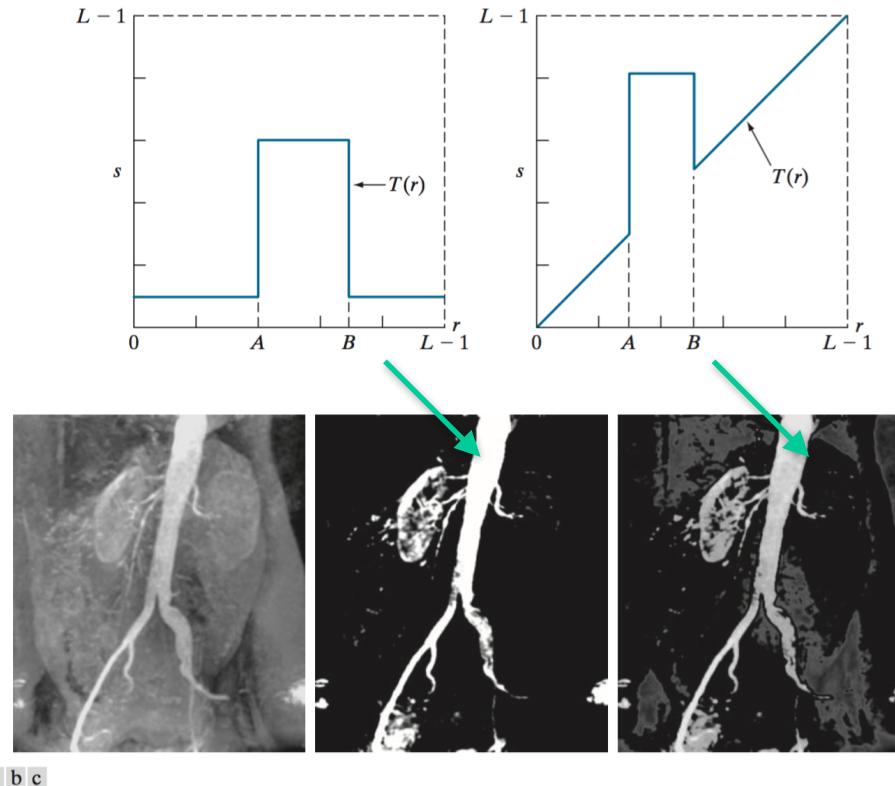
The locations of  $(r_1, s_1)$  and  $(r_2, s_2)$  control the shape of the transformation function.

- a) If  $r_1 = s_1$  and  $r_2 = s_2$  the transformation is a linear function: no change in intensity.
- b) If  $r_1 = r_2 = m$ ,  $s_1 = 0$  and  $s_2 = L - 1$  the transformation is a thresholding function: a binary image.



# Intensity Transformation: Intensity Slicing

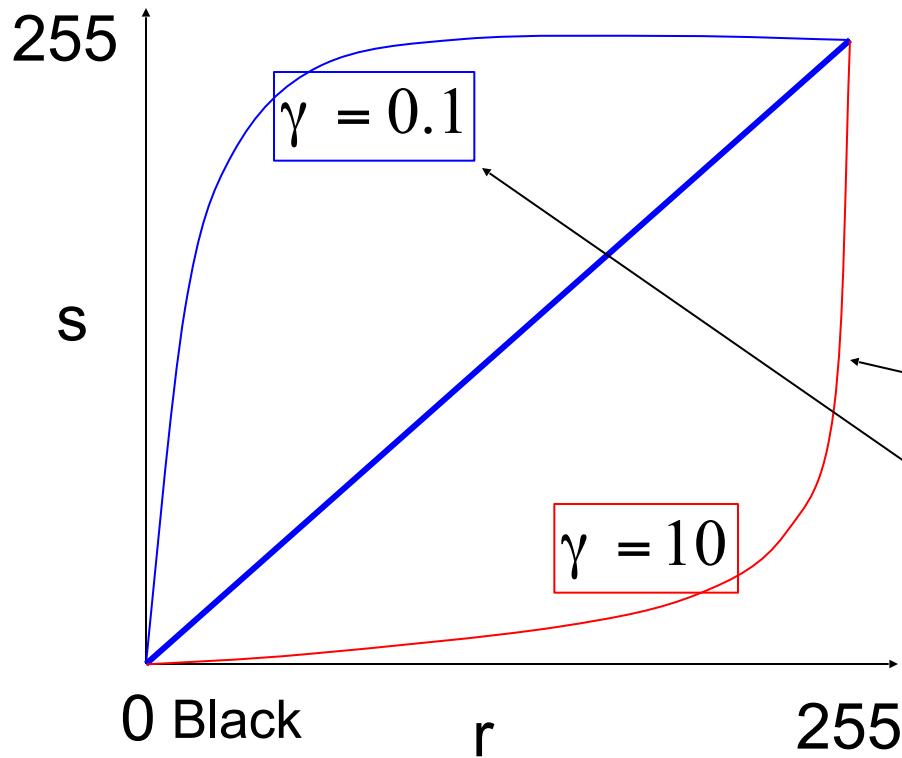
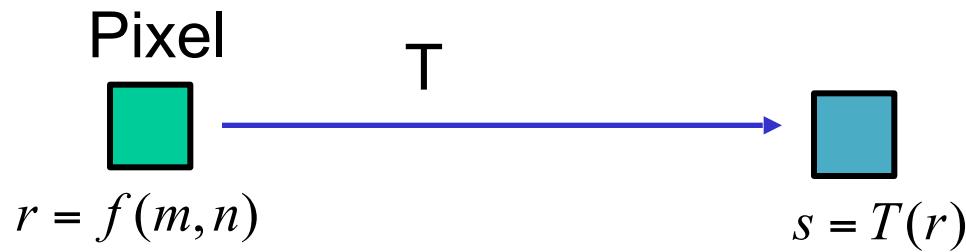
The aim is to highlight specific range of intensity of interest.



**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 range set near black, so that the 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.)

See also Fig. 3.12 (p. 117 of the 3<sup>rd</sup> Ed., or p. 145 of the 4<sup>th</sup> Ed.)

# Simplest form of processing: Point Processing



Common Examples:

Gamma Correction

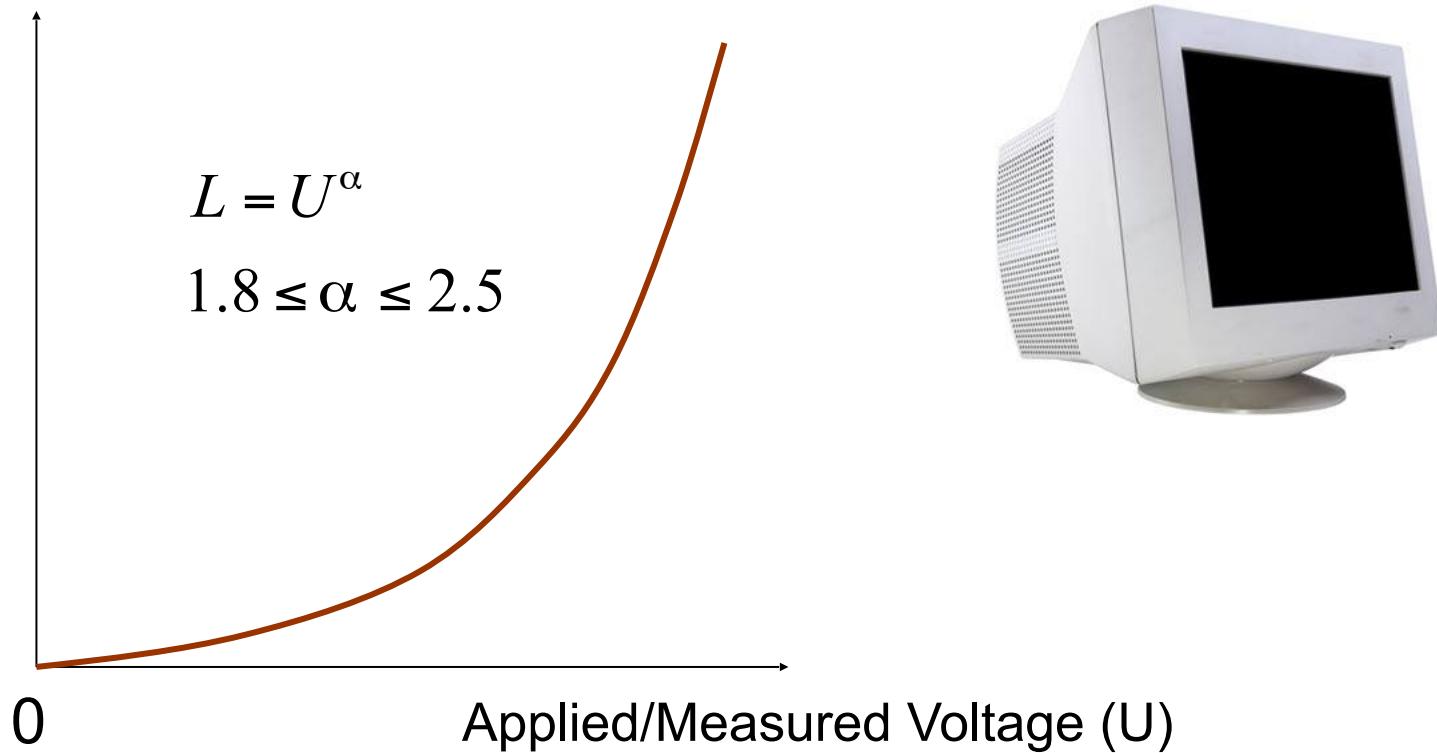
$$T(r) = cr^\gamma$$

Narrow range of “dark” gets mapped to broad range of “gray”

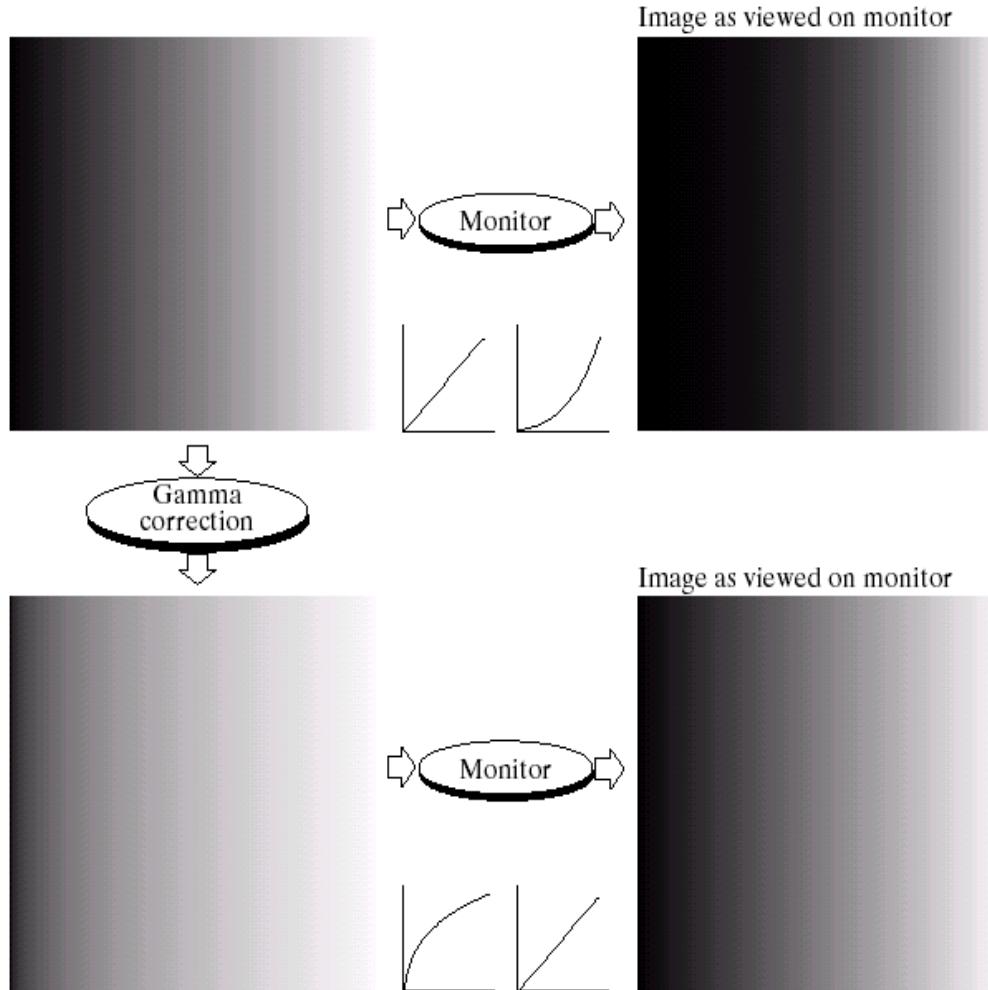
## Origins of gamma correction:

- Nonlinear response of CRT's and imagers
- To correct for this in image display, the images or commands to the CRT are “pre-distorted”

Luminance



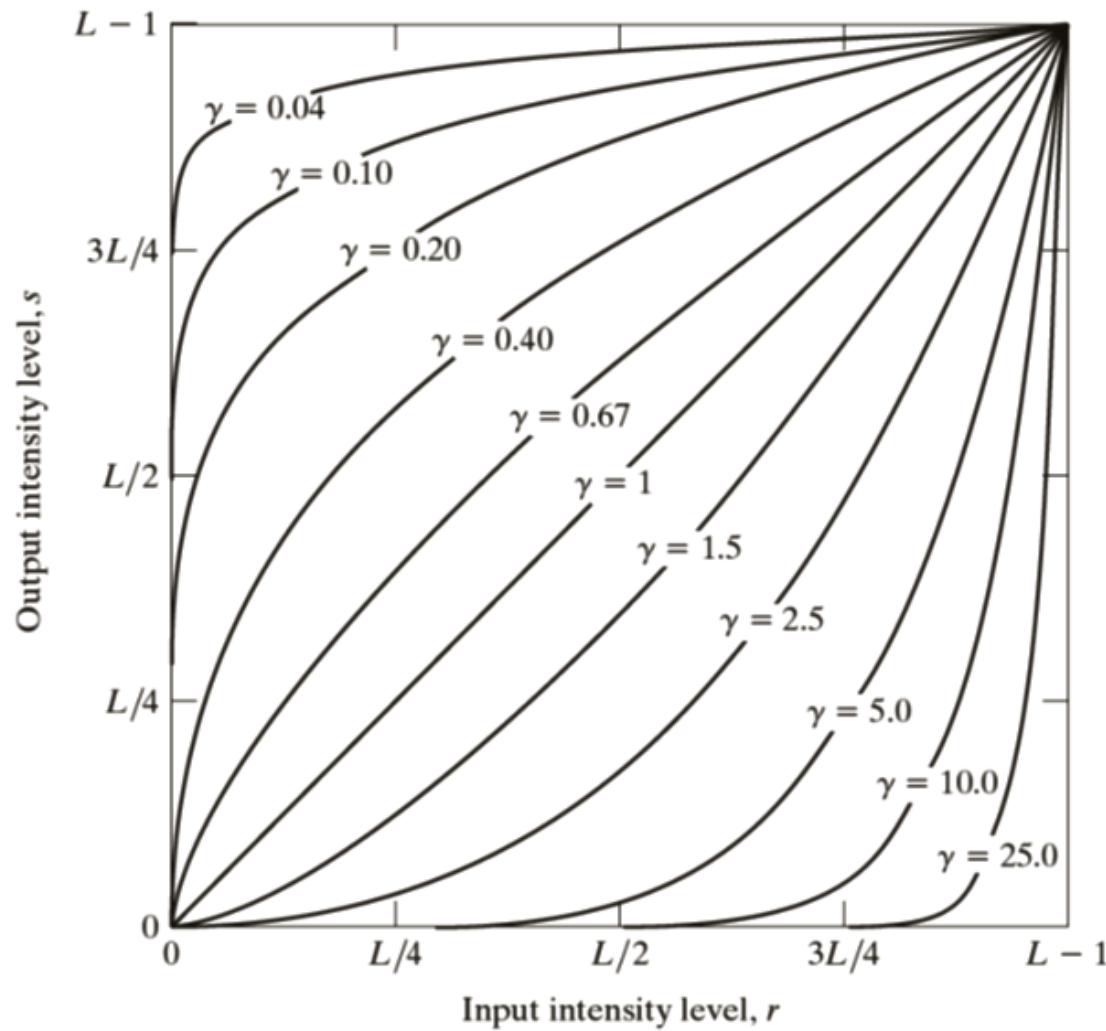
# Gamma Correction



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).



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

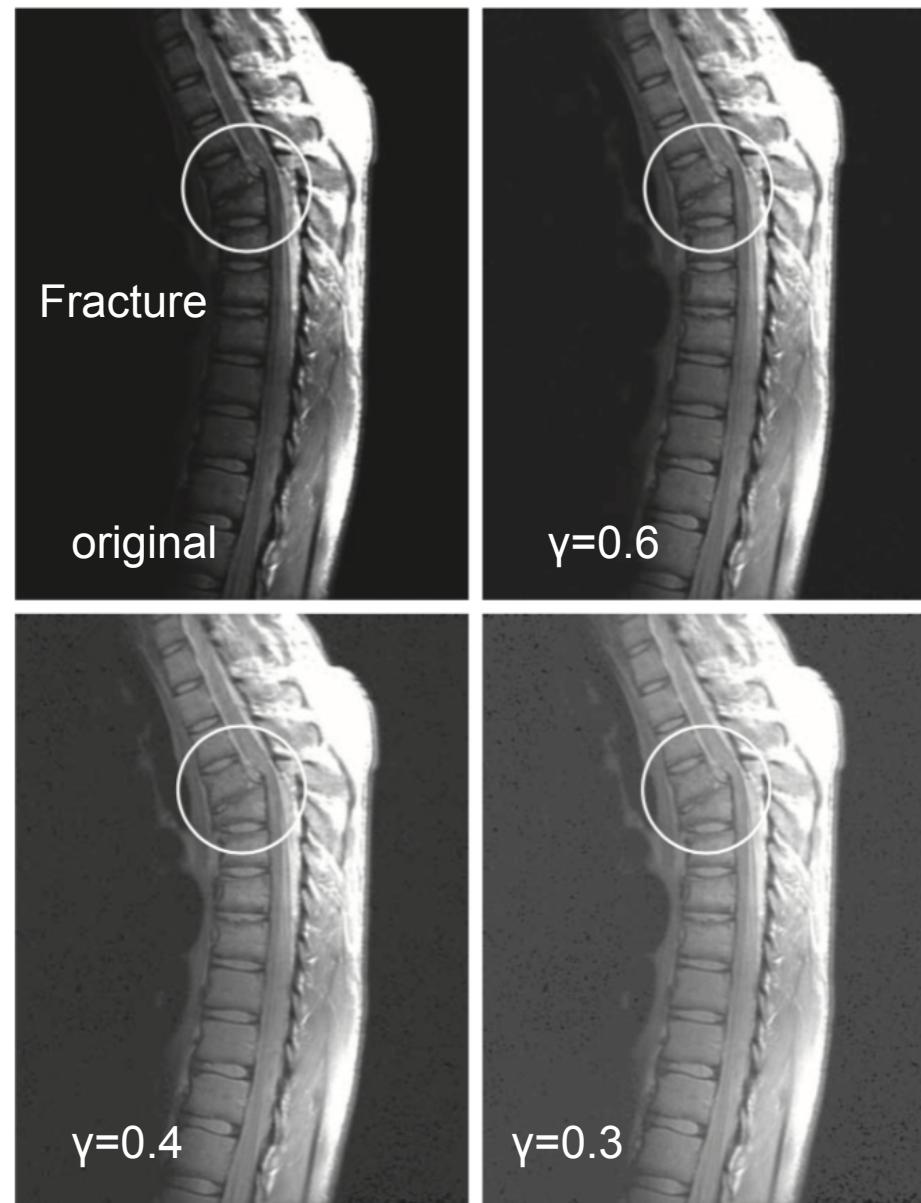
a  
b  
c  
d

**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.)

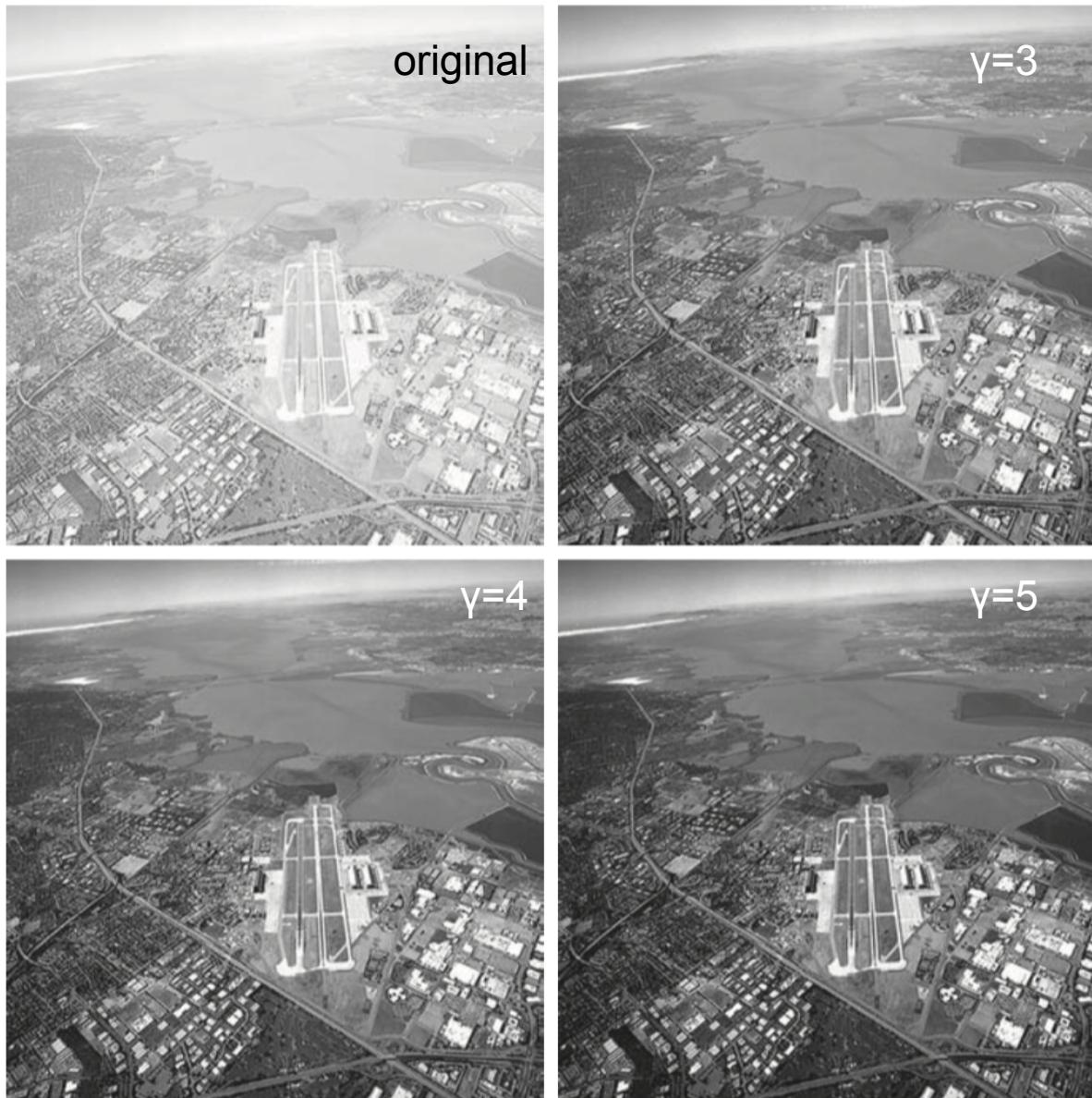


a	b
c	d

**FIGURE 3.9**

(a) Aerial image.  
(b)–(d) Results  
of applying the  
transformation  
in Eq. (3-5) with  
 $\gamma = 3.0, 4.0,$  and  
 $5.0,$  respectively.  
( $c = 1$  in all cases.)  
(Original image  
courtesy of  
NASA.)

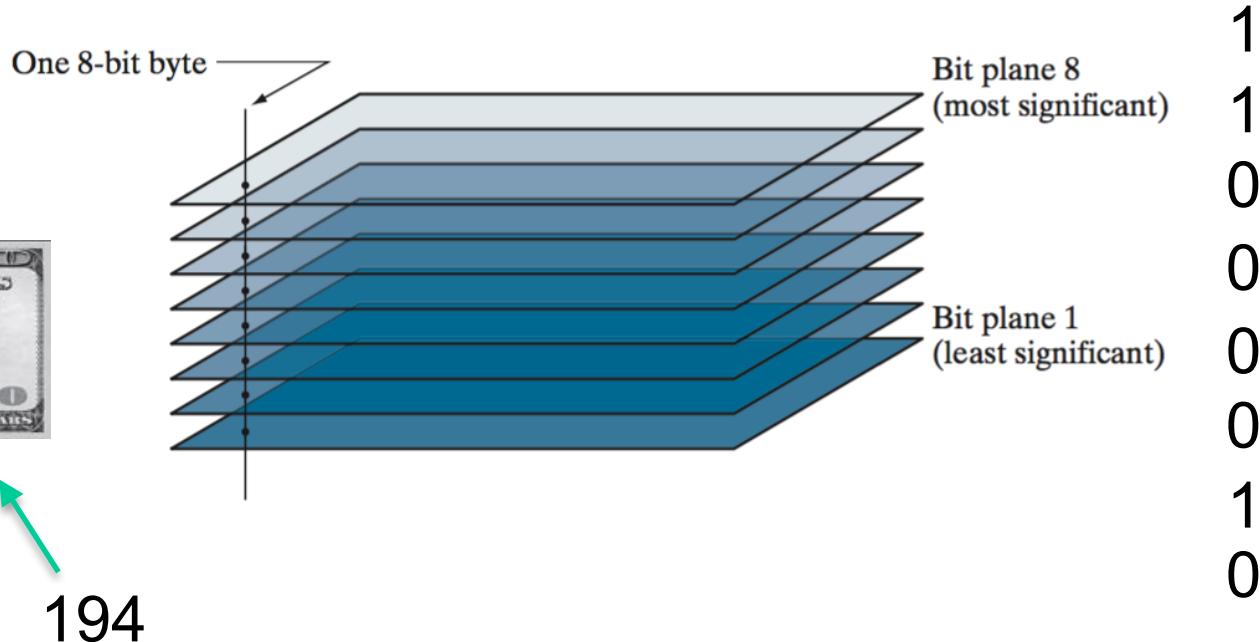
---



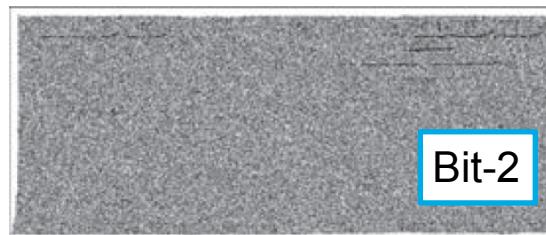
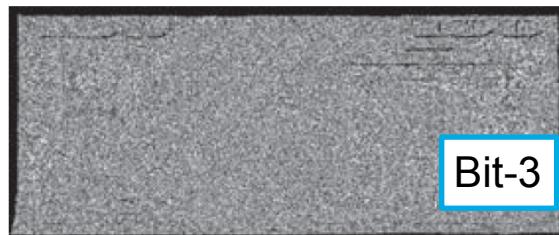
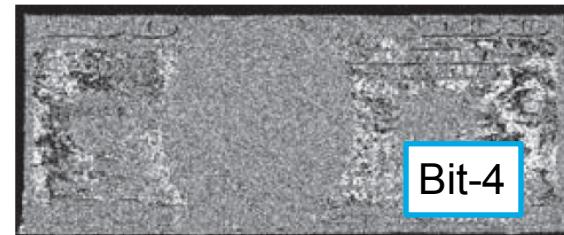
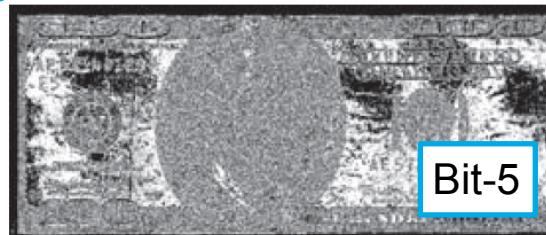
# Intensity Transformation: Bit-plane Slicing

Ex. In 8-bit image, grey level 194 is (1 1 0 0 0 0 1 0).  
Grey levels (0-127) have the 8th bit = 0, and levels  
(128-255) have the 8th bit = 1 (counting bits from right to left)

**FIGURE 3.13**  
Bit-planes of an  
8-bit image.

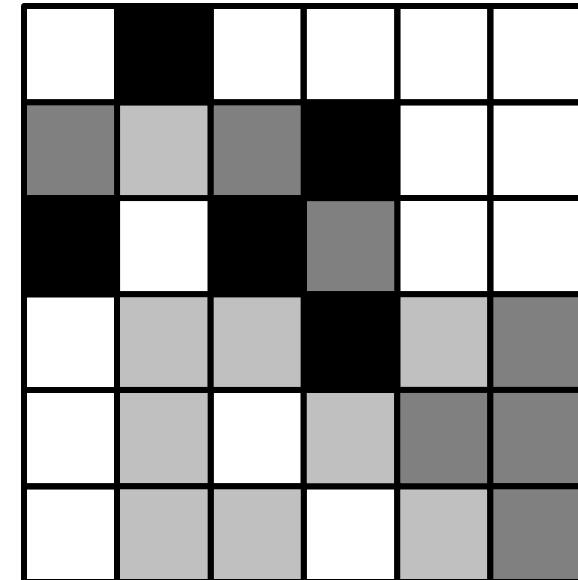
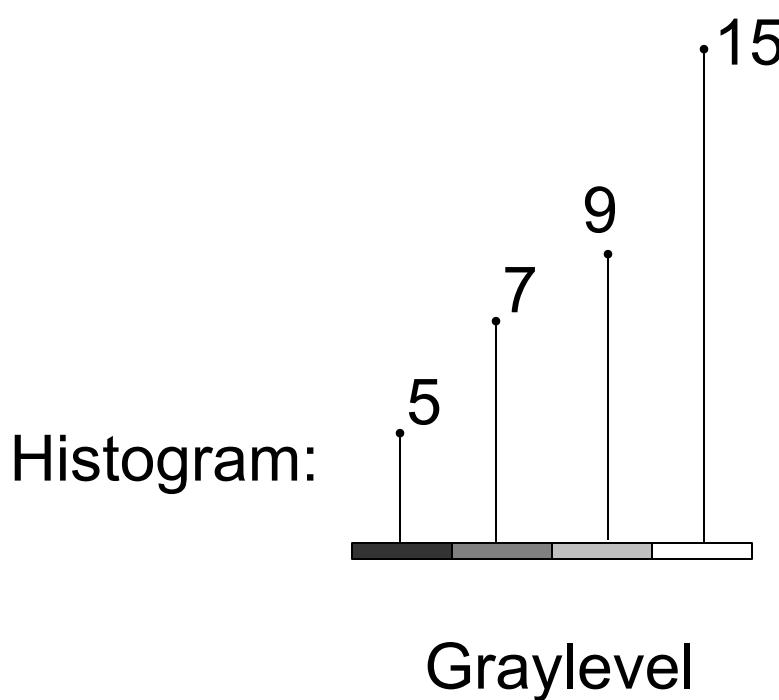


# Intensity Transformation: Bit-plane Slicing

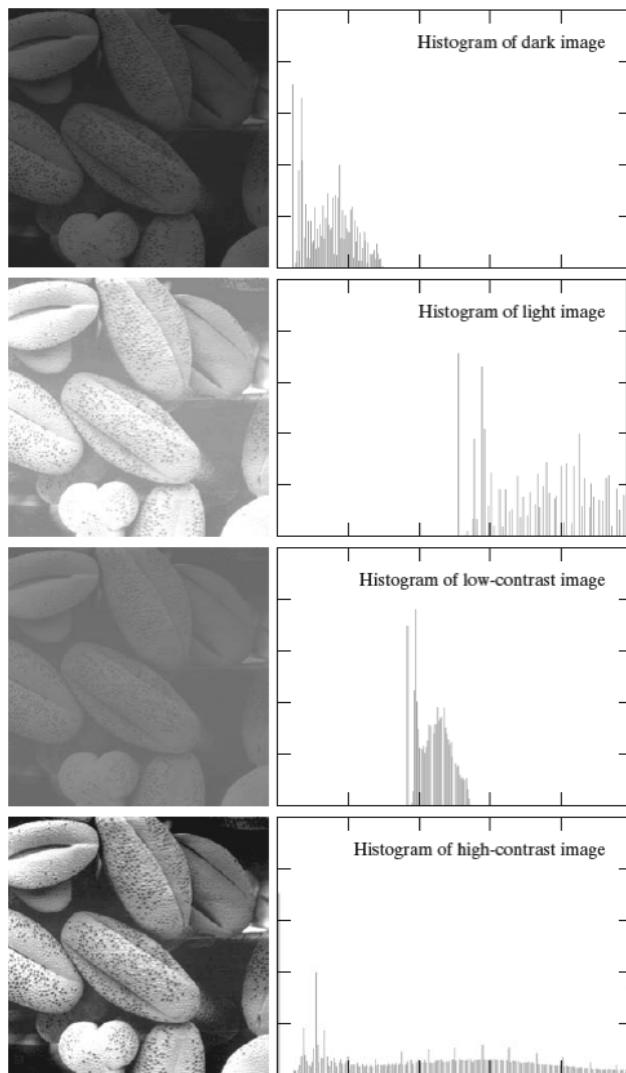


# Histogram Processing

- Distribution of gray-levels can be judged by measuring a Histogram



# Histogram Processing



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

# Histogram Processing:

For B-bit image,

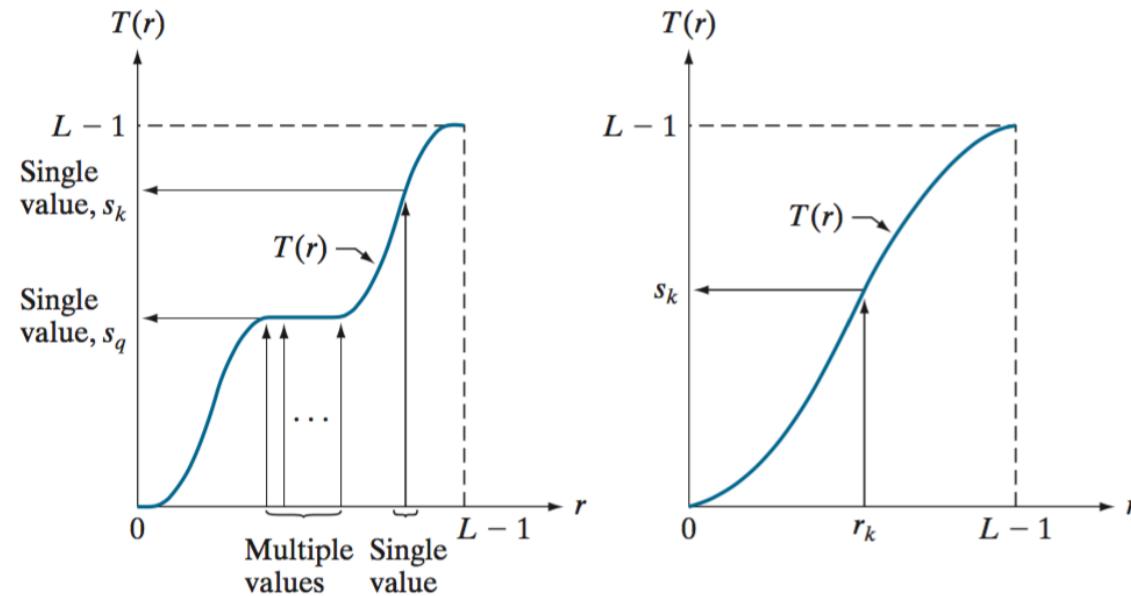
- Initialize  $2^B$  counters with 0
- Loop over all pixels  $x,y$
- When encountering gray level  $f(x,y)=i$ , increment the counter number  $i$
- With proper normalization, the histogram can be interpreted as an estimate of the probability density function (pdf) of the underlying random variable (the gray-level)
- You can also use fewer, larger bins to trade off amplitude

# Histogram Processing

a b

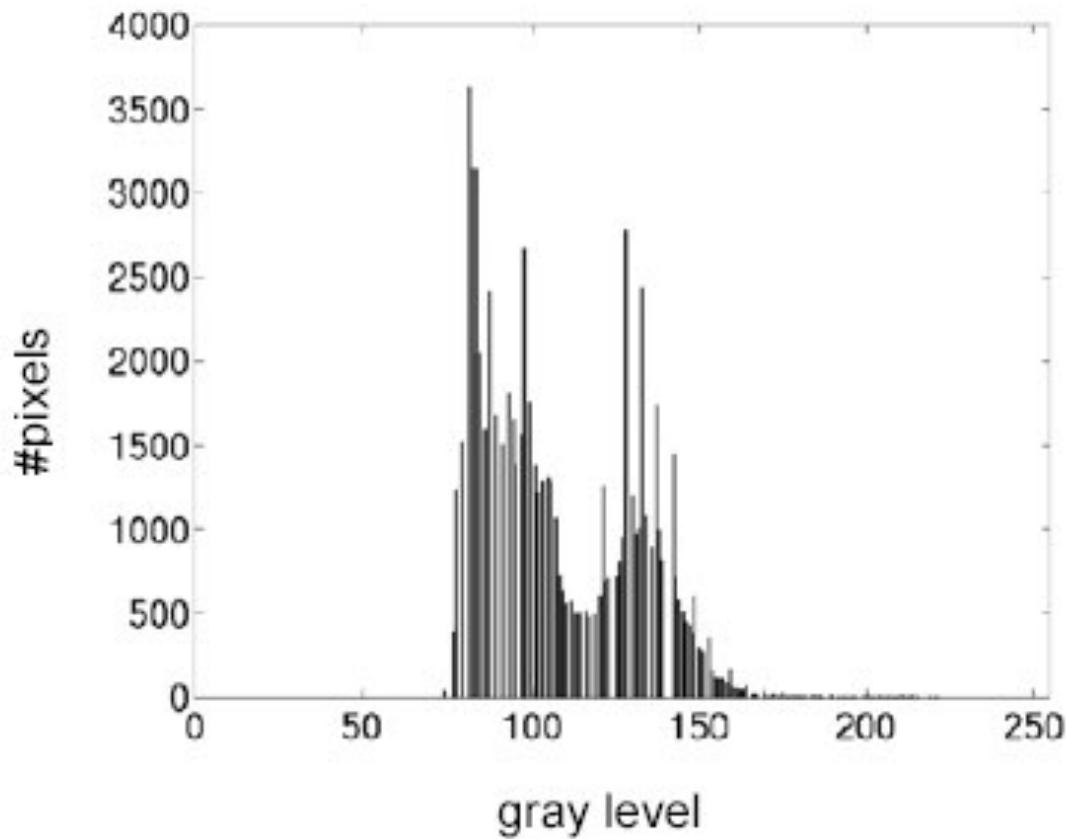
**FIGURE 3.17**

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



One-to-one mapping: no information destruction

## Example:



*Pout*  
image

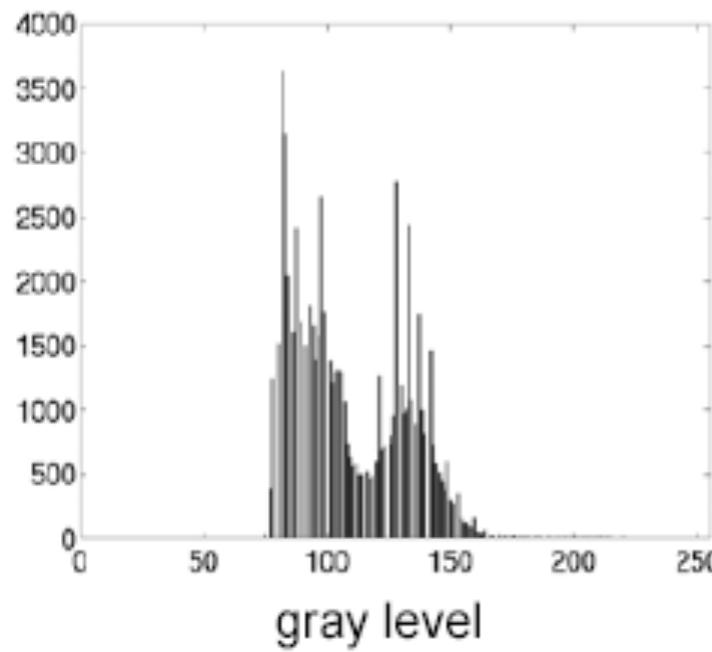


Original image *Pout*

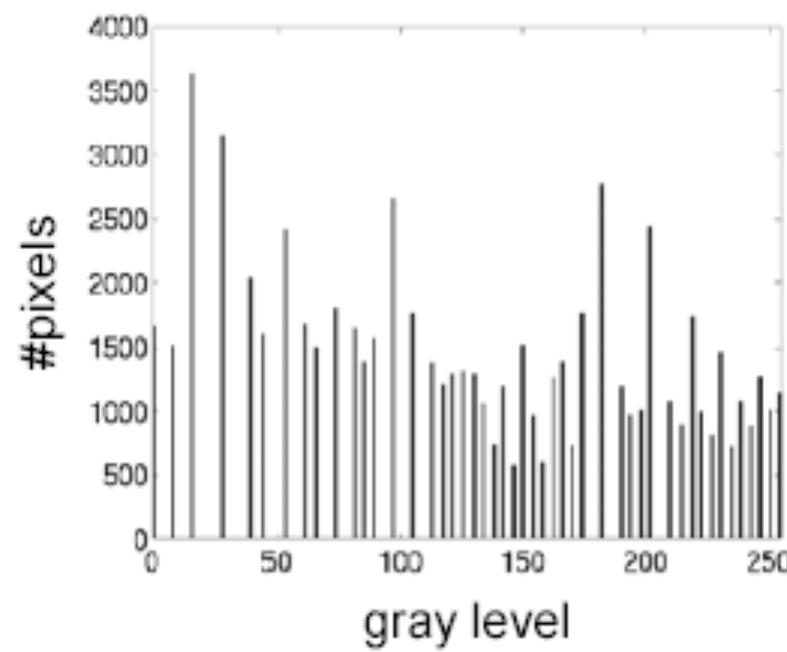


*Pout*  
after histogram equalization

Original image *Pout*



... after histogram equalization





Take a break!

# Histogram Equalization

- The general idea: map the histogram of the given image to a **flat histogram** by performing a *nonlinear operation on the gray value at each pixel.*
- Nonlinear Transformation:  $s = T(r)$
- Questions:
  - What is the right transformation?
  - How do we find it, given a particular image.
- Analysis for the continuous grayscale first

# Histogram Equalization

- Consider the histogram of the given (continuous grayscale) image as a pdf  $p(r)$ , where  $r$  is in the interval  $[0, L-1]$ .  
– Recall that as a pdf we have  $p(r) \geq 0$  and  $\int_0^{L-1} p(r)dr = 1$   
– Any pixel operation  $T(r)$  should map  $[0, L-1]$  to  $[0, L-1]$
- Desired properties of  $T(r)$ :
  - Keep the black/white order ( $T(r)$  should be monotonic increasing)
  - $T(r)$  should be single valued (one-to-one), hence invertible.
- Question: Given image with histogram  $p(r)$ , what does histogram of  $s=T(r)$  look like in general?

## Histogram Equalization

- Consider continuous values of intensity
- $r$  = gray level of image to be equalized in the range  $[0, L-1]$

**0 : black**

**L-1: white**

Goal is to design a transformation

$$s = T(r) : 0 \leq r \leq L - 1, \quad 0 \leq s \leq L - 1$$

Assumption about  $T(r)$  :

(a)  $T(r)$  is single valued and monotonically increasing in the interval  $0 \leq r \leq L - 1$

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

Condition (a) assures that  $T^{-1}(s) = r$

- From Probability theory:

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| \quad (\text{See Review Slides p. 34})$$

- We consider **Cumulative Distribution Function** (CDF) as a Transformation

$$\underline{s = T(r) = (L-1) \int_0^r p_r(\omega) d\omega}$$

$$\frac{ds}{dr} = (L-1) \frac{d}{dr} \left\{ \int_0^r p_r(\omega) d\omega \right\} = (L-1)p_r(r)$$

Hence:  $p_s(s) = p_r(r) \left| \frac{1}{(L-1)p_r(r)} \right| = \frac{1}{L-1}$  for  $0 \leq s \leq L-1$

- If  $T(r)$  is just a CDF or the integral of the input pdf ( $p_r(r)$ ) then applying  $s = T(r)$  results in an image whose pdf is uniform.

*(For PDF and CDF see pages 30-32 of the Review Slides)*

## Discrete Case

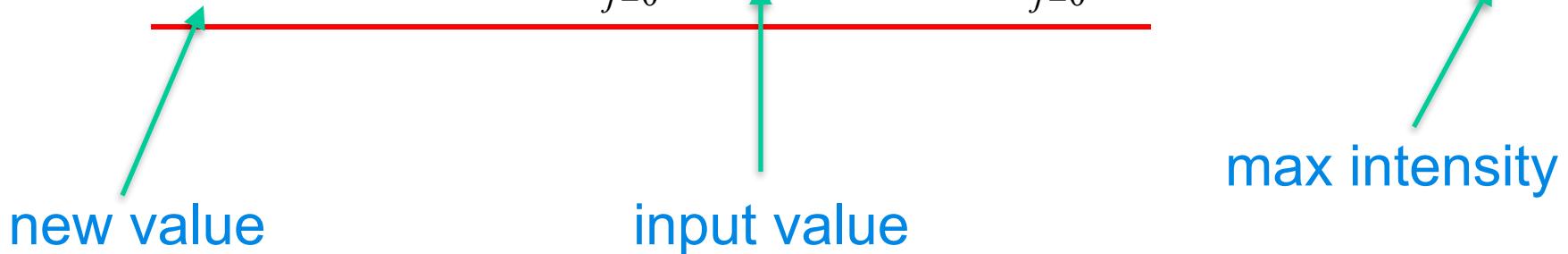
$r_k$  = discrete intensity,  $k = 0, \dots, L - 1$

$$p_r(r_k) = \frac{n_k}{n}$$

$n_k$  = number of pixels that have intensity  $r_k$

$n$  = total number of pixels of the image

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

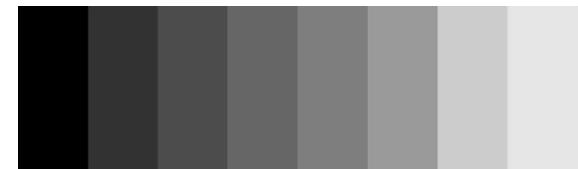


# Example

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

**TABLE 3.1**

Intensity distribution and histogram values for a 3-bit,  $64 \times 64$  digital image.



8 grey levels

$$n = M \times N$$

**Example:** 3-bit image,  $L = 8$ , image size =  $n = 64 \times 64 = 4096$ , intensity levels [0,7].

Therefore,  $p_0(r_0) = 790/4096 = 0.19$ , etc...

$$s_k = (L-1) \sum_{j=0}^k \left( \frac{n_j}{n} \right), \quad k = 0, \dots, L-1$$

Therefore  $s_0 = 7 \times 0.19 = 1.35$ ;  $s_1 = 7 \times (0.19 + 0.25) = 3.1$

$$s_2 = 4.55, \quad s_3 = 5.67, \quad s_4 = 6.23, \quad s_5 = 6.65, \quad s_6 = 6.86, \quad s_7 = 7.00$$

Note that the gray levels are integers so the values of the equalized histogram are:  $s_0 = 1, s_1 = 3, s_2 = 5, s_3 = s_4 = 6, s_5 = s_6 = s_7 = 7$ .

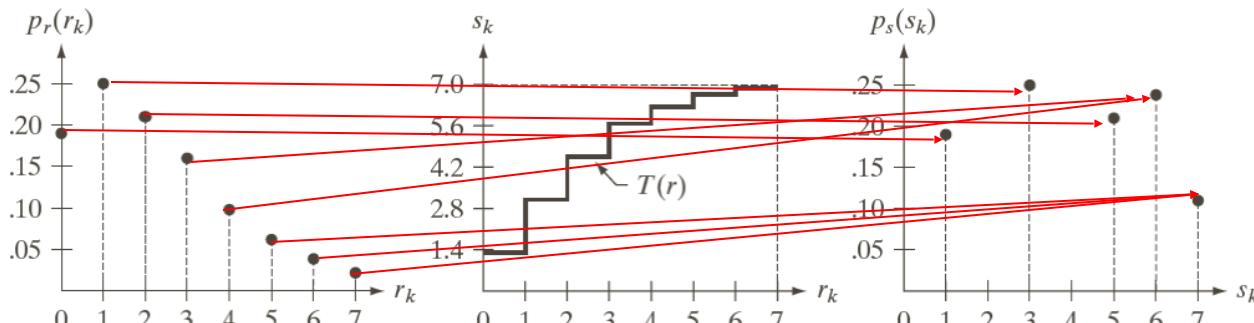
$r_0 = 0$  is mapped to  $s_0 = 1 \Rightarrow n_{s_0} = n_{r_0} = 790 \Rightarrow p_s(s_0) = \frac{790}{4096} = 0.19$

$r_1 = 1$  is mapped to  $s_1 = 3 \Rightarrow n_{s_1} = n_{r_1} = 1023 \Rightarrow p_s(s_1) = \frac{1023}{4096} = 0.25$

$r_2 = 2$  is mapped to  $s_2 = 5 \Rightarrow n_{s_2} = n_{r_2} = 850 \Rightarrow p_s(s_2) = \frac{850}{4096} = 0.21$

But  $r_3$  and  $r_4$  both mapped to  $s_3 = s_4 = 6 \Rightarrow n_{s_3} = n_{s_4} = 656 + 392 \Rightarrow p_s(s_3) = \frac{1048}{4096} = 0.26$

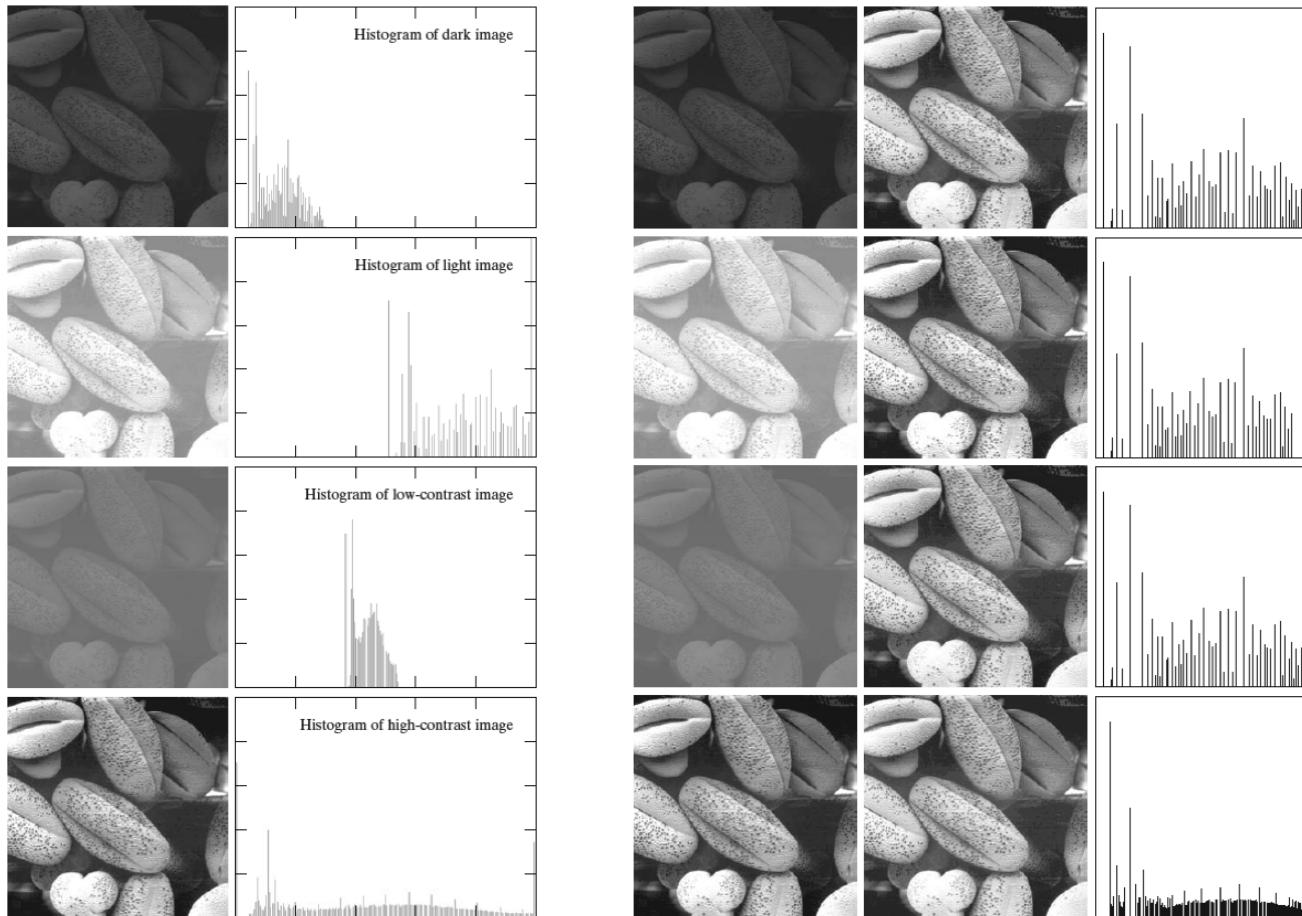
$r_5, r_6$  and  $r_7$  all mapped to  $s_5 = s_6 = s_7 = 7 \Rightarrow n_{s_5} = 245 + 122 + 81 \Rightarrow p_s(s_5) = \frac{448}{4096} = 0.11$



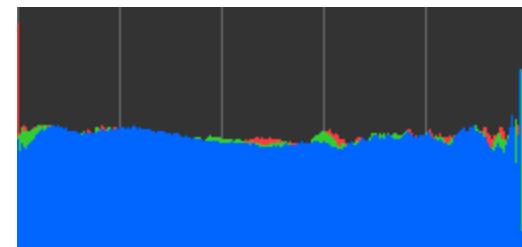
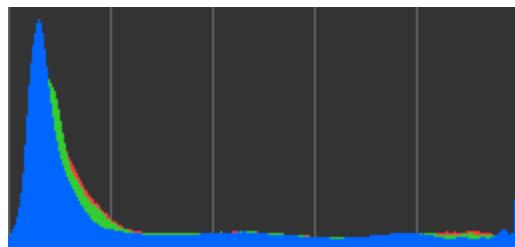
a b c

**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 (examples)



**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.



## Generalizations

- Adaptive Histogram Equalization
- Histogram Matching
  - Instead of flattening the histogram, we want to make the histogram of **A** look like the histogram of another image **B**.
  - Procedure (given two images):
    - Find transformation ( $T$  and  $U$ ) that flatten images **A**, **B**
    - Required transformation is  $U^{-1}(T(\mathbf{A}))$

# Histogram Matching

- Rather than  $p_s(s)$  uniform we want  $p_s(s)$  to match as closely as possible to a desired, given *pdf*.
- $r$  = pixel value before matching
- $z$  = pixel value after matching
- We can compute  $p_r(r)$  from the given image, and  $p_z(z)$  is given. What is the Transformation from  $r$  to  $z$  ?
- In the continuous case:

$$s = T(r) = (L - 1) \int_0^r p_r(\omega) d\omega = CDF \text{ of } r$$

$$v = G(z) = (L - 1) \int_0^z p_z(t) dt = CDF \text{ of } z$$

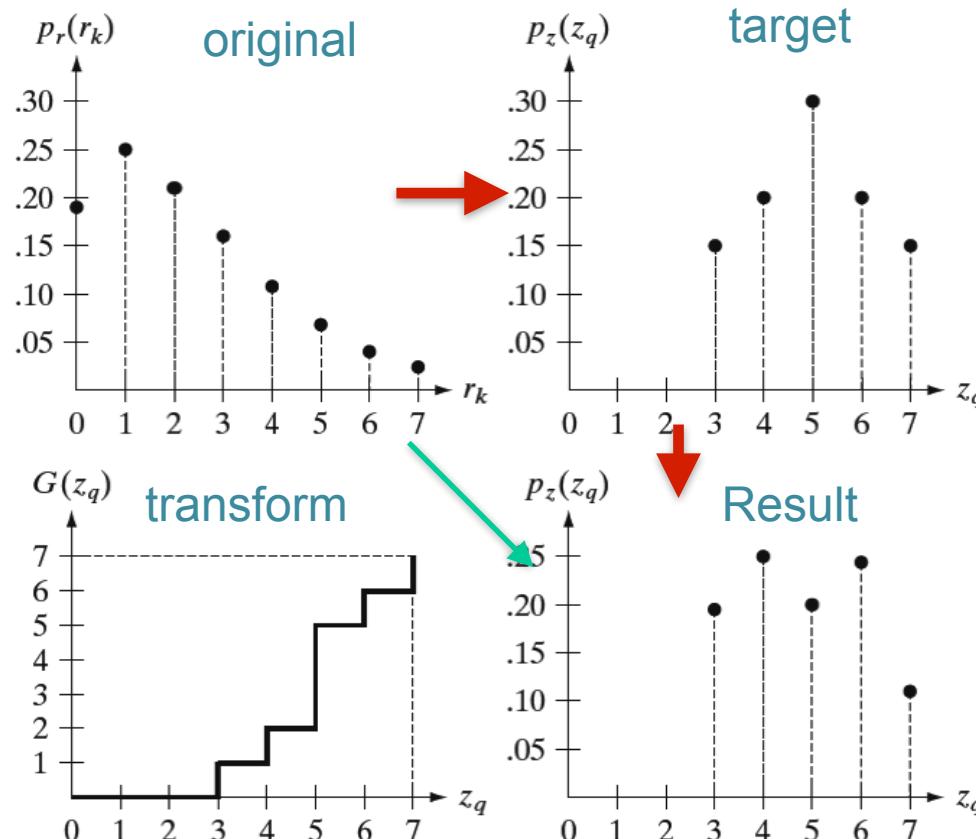
- Can form Look-up table

## Approach

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

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

*Matching:*  $G(z) = T(r) \rightarrow z = G^{-1}\{T(r)\} = G^{-1}(s) \quad (3)$



## Approach

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

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

Matching:  $G(z) = T(r) \rightarrow z = G^{-1}\{T(r)\} = G^{-1}(s) \quad (3)$

### Method:

1. Compute  $p_r(r)$  from input image and use(1) to find values of  $s$ .

Round the resulting values to the integer range  $[0, L-1]$ .

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

2. Use the given PDF  $p_z(z_i)$  to obtain transformation  $G(z)$  as in(2).

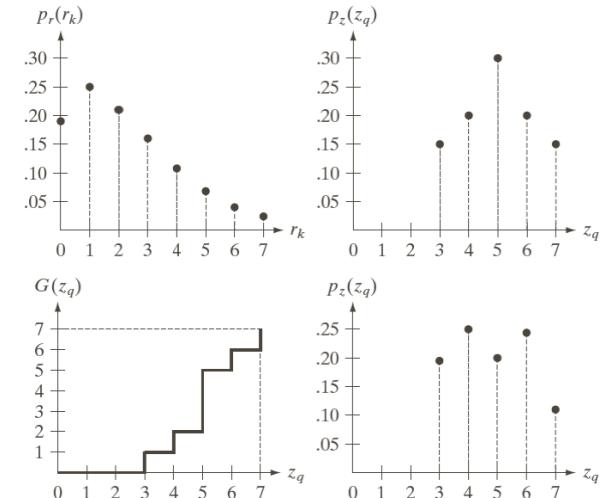
Round the values of  $G$  as above.

$$G_0 = G_1 = G_2 = 0, G_3 = 1 \text{ (round-off)}, G_4 = 2, G_5 = 5, G_6 = 6, G_7 = 7.$$

Create a table of  $G$ .

3. For every value of  $s_k$  use the stored values of  $G$  to find the corresponding of  $z_q$  so that  $G(z_q) \approx s_k$  and store these mappings from  $s$  to  $z$ . When more than one  $z_q$  satisfy the given  $s_k$  use the smallest.

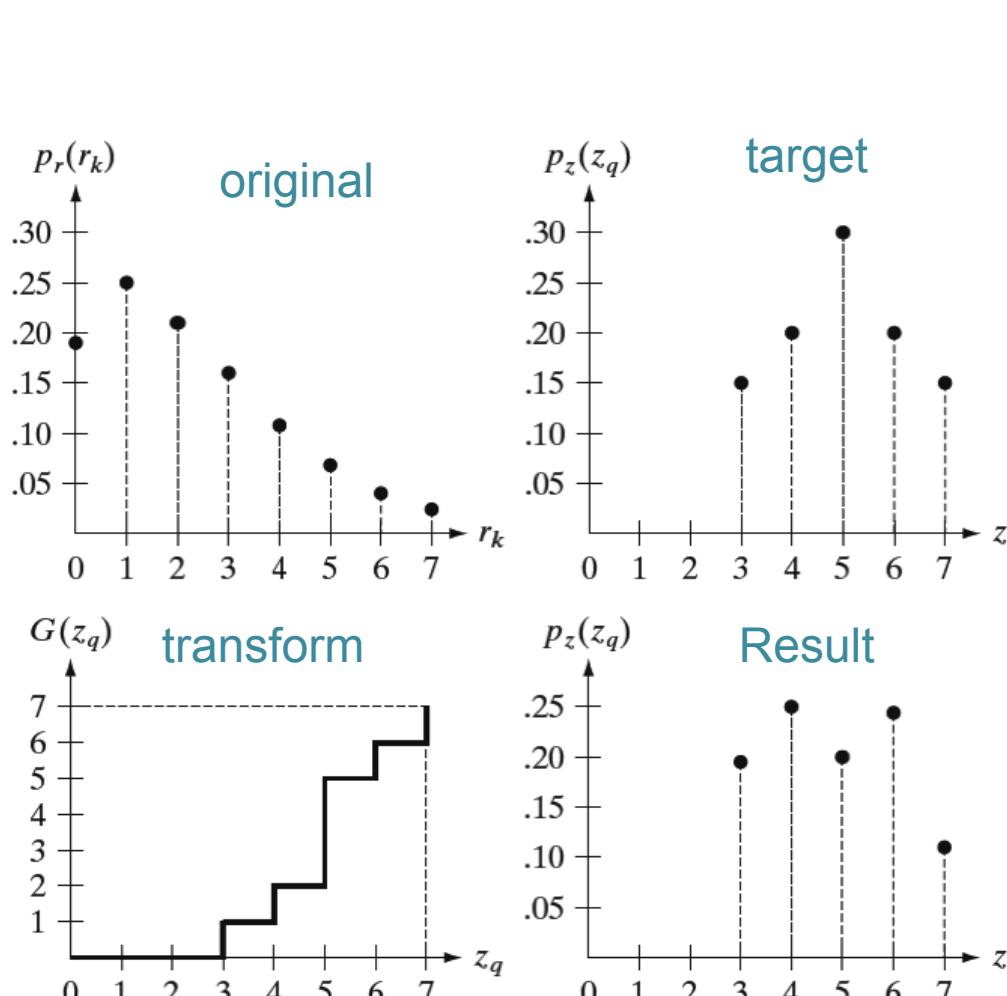
4. Obtain output image by first equalizing input image by using(1) ; pixel values in this image are  $s$  values. For each pixel with value  $s$  in the equalized image, perform inverse mapping(3) to obtain the corresponding pixel in output image.



$r_k$	$s_k$	$z_k$
0	1	3
1	3	4
2	5	5
3	6	6
4	6	6
5	7	7
6	7	7
7	7	7

$$\begin{aligned}
 s_0 = 1 &\rightarrow G_3 = 1 \rightarrow z = 3 \\
 s_1 = 3 &\rightarrow G_4 = 2 \rightarrow z = 4 \\
 s_2 = 5 &\rightarrow G_5 = 5 \rightarrow z = 5 \\
 s_3 = s_4 = 6 &\rightarrow G_6 = 6 \rightarrow z = 6 \\
 s_5 = s_6 = s_7 = 7 &\rightarrow G_7 = 7 \rightarrow z = 7
 \end{aligned}$$

# Histogram Matching (example)



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).

$r_k$	$s_k$	$z_k$
0	1	3
1	3	4
2	5	5
3	6	6
4	6	6
5	7	7
6	7	7
7	7	7

# Histogram Matching (example)

$z_q$	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

$z_q$	$G(z_q)$
$z_0 = 0$	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

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

**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).

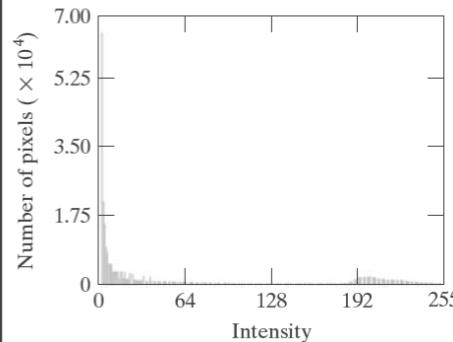
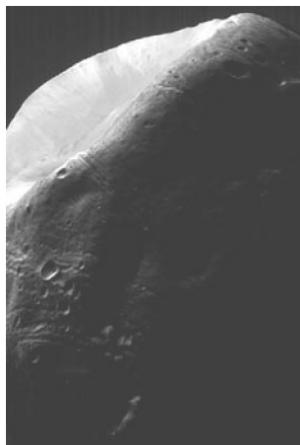
**TABLE 3.3**

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

**TABLE 3.4**

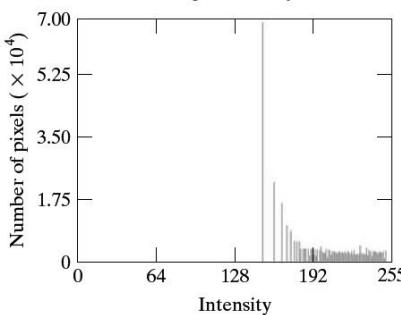
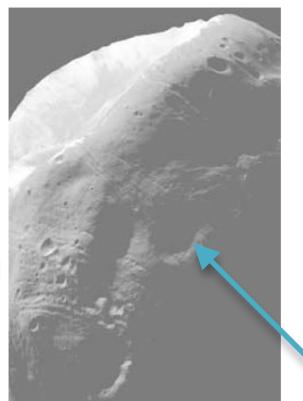
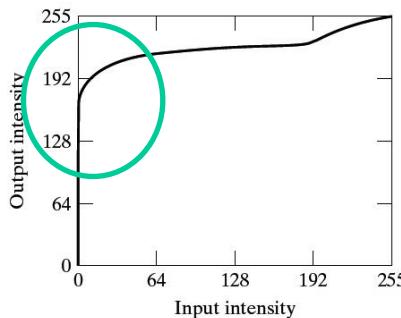
Mappings of all the values of  $s_k$  into corresponding values of  $z_q$ .

# Comparison of 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.)

a b  
c

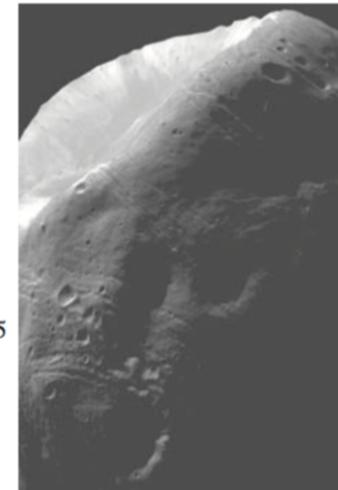
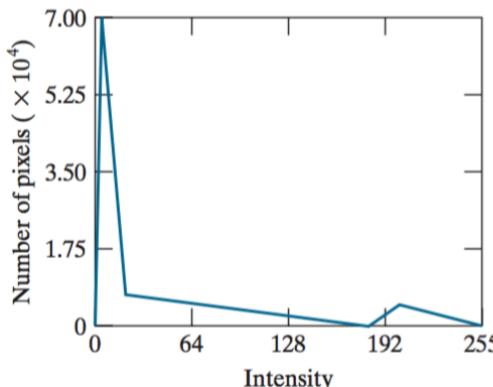
**FIGURE 3.24**  
 (a) Transformation function for histogram equalization.  
 (b) Histogram-equalized image (note the washed-out appearance).  
 (c) Histogram of (b).

## Histogram Equalization

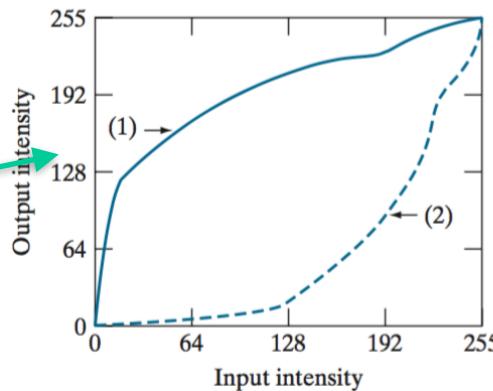
More details

# Histogram Matching (example)

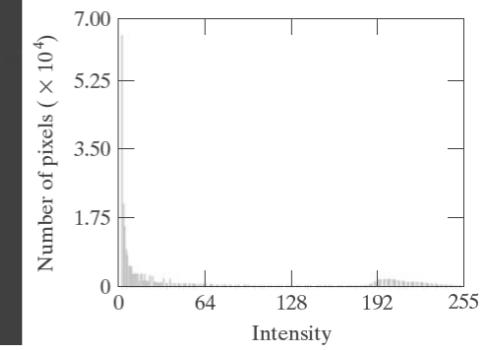
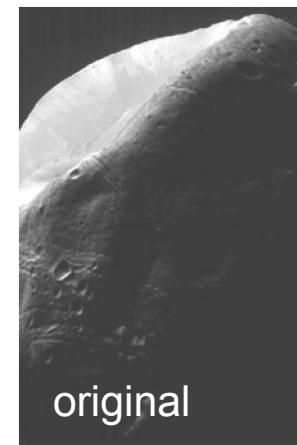
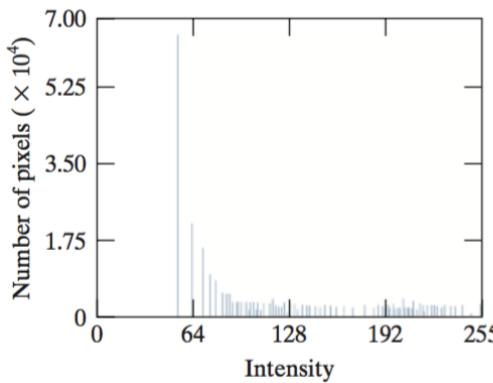
target histogram



Transform



Result



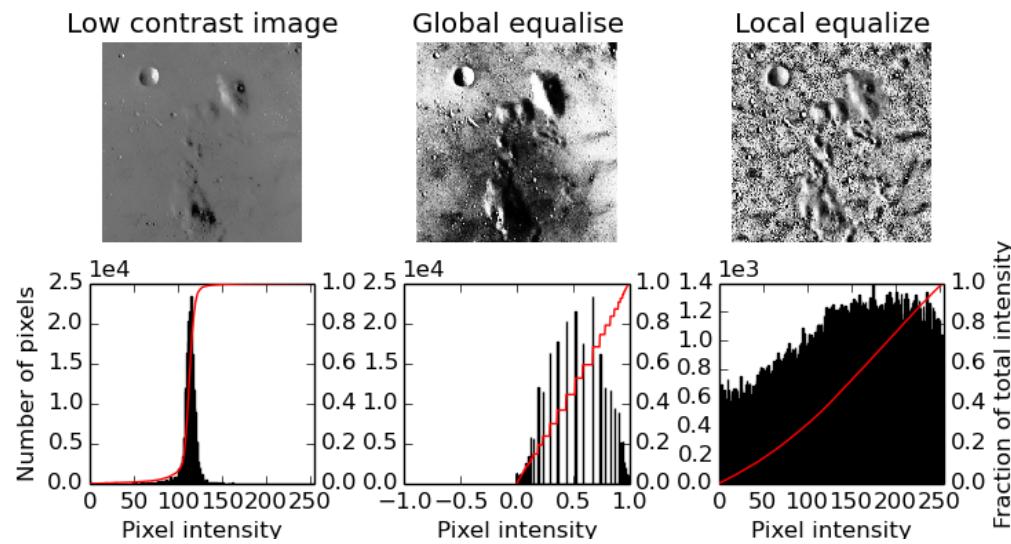
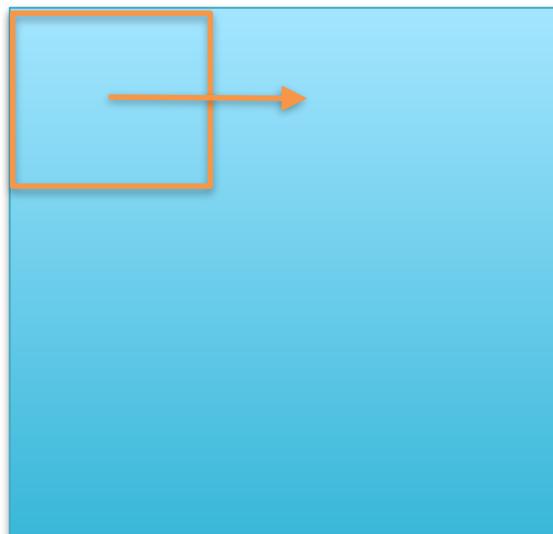
a c  
b d

**FIGURE 3.25**  
Histogram specification.  
(a) Specified histogram.  
(b) Transformation  $G(z_q)$ , labeled (1), and  $G^{-1}(s_k)$ , labeled (2).  
(c) Result of histogram specification.  
(d) Histogram of image (c).

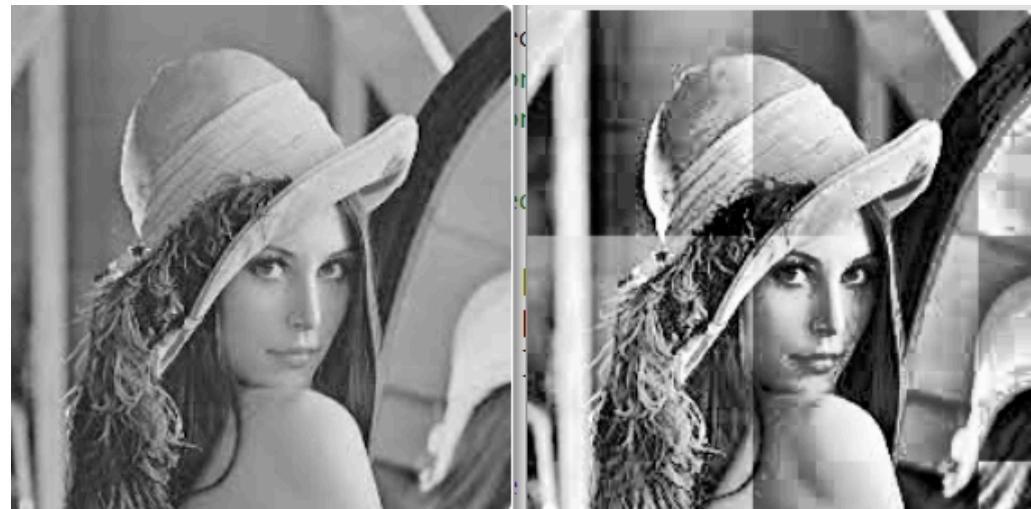
Original histogram

# Local Enhancement using Local Histogram

- Define a “window” as a rectangle neighborhood.
- Move centre of windows across the image. At each location, compute histogram in the window and do histogram equalization.



# Local Enhancement using Local Histogram



<https://www.geeksforgeeks.org/adaptive-histogram-equalization-in-image-processing-using-matlab/>

- *Ridge artifacts due to window selection*
- *Some constraints may be needed to ensure results*

## Multi-Image “Averaging”

- It may be possible to obtain multiple noise-corrupted images of a scene and average them to obtain a **less noisy result**.
- It is important that there be **no motion** between the frames, or that the frames be aligned first.

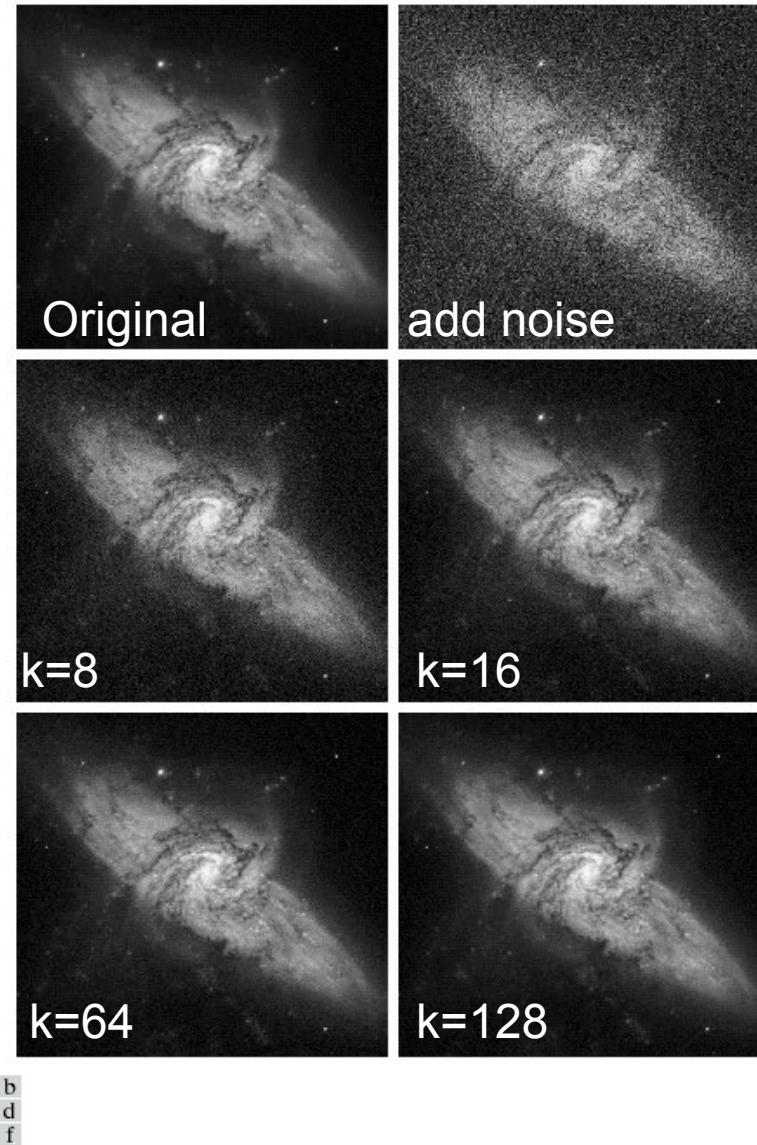
Given:  $g_k(x, y) = f(x, y) + n_k(x, y)$

Compute:  $\hat{f}(x, y) = \sum_{k=1}^N \alpha_k g_k(x, y)$

$$\sum_{k=1}^N \alpha_k = 1$$

## Simple Averaging Example:

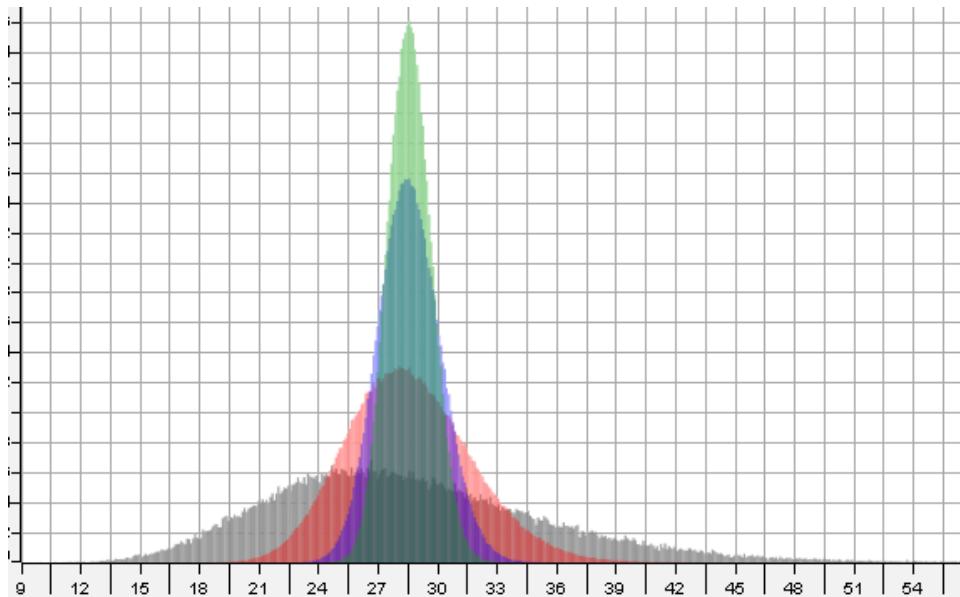
Averaging k times



**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.)

**Additive noise =**  
Gaussian with zero  
mean & standard  
deviation of 64 grey  
levels

# Character of image noise via averaging



$$\mu_{\bar{x}} = \mu$$

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

Large N for averaging is desirable but costly

## Optimal Multi-image “Averaging”

Given:  $g_k(x, y) = f(x, y) + n_k(x, y)$

Compute:  $\hat{f}(x, y) = \sum_{k=1}^N \alpha_k g_k(x, y)$

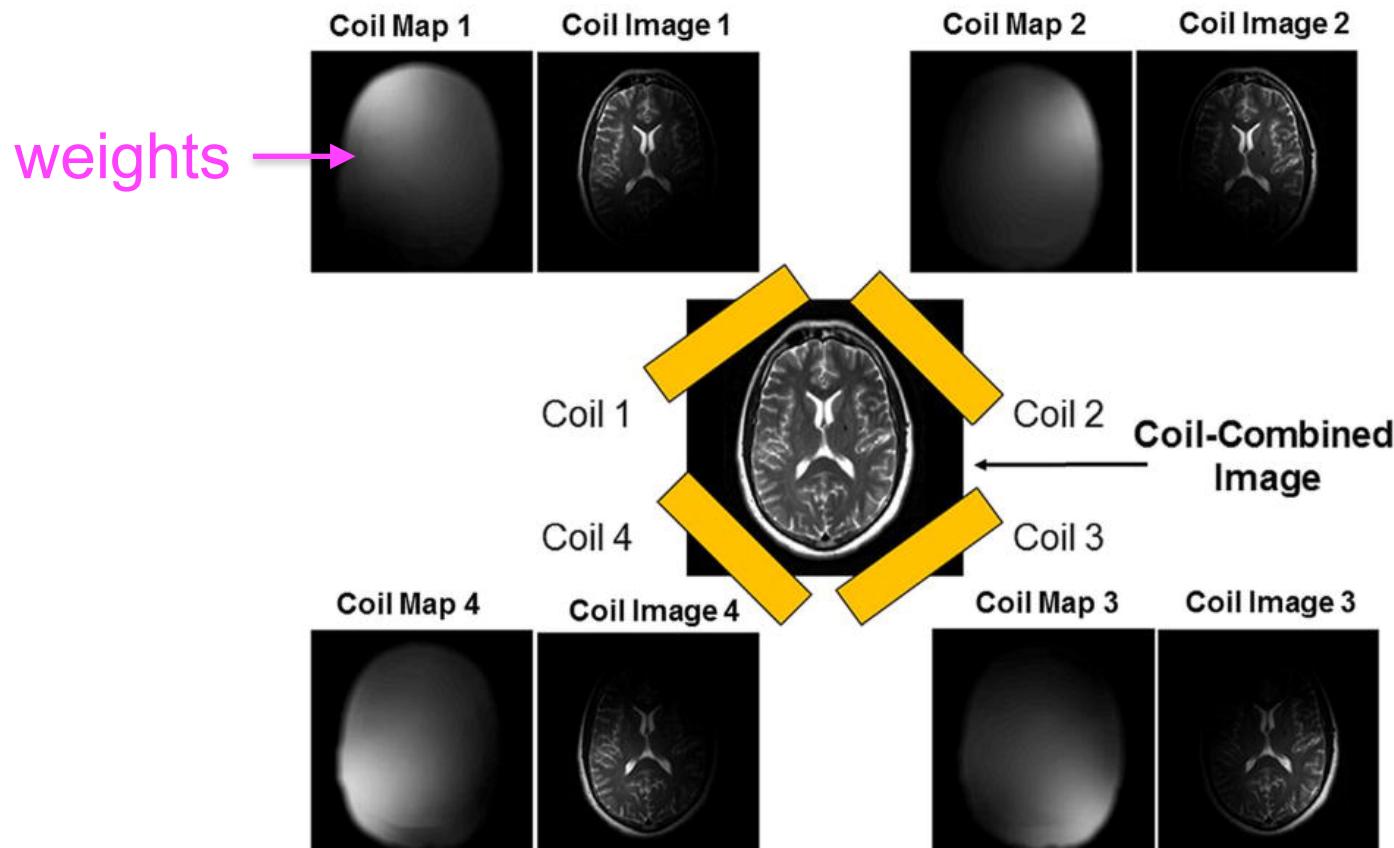
$$\sum_{k=1}^N \alpha_k = 1$$

Given the noise statistics of the measured images, what should these coefficients be to yield the “best” results?

Inversely proportional to the variance of noise in each image

# MRI scan reconstruction from multiple sensor data

*Adaptive averaging for image reconstruction*



(Hamilton et al. Europe PMC, 2017 )

# Summary

- **Point processing**
  - ✓ Contrast stretching
  - ✓ Intensity slicing
  - ✓ Gamma correction
  - ✓ Bit-plane slicing
- **Histogram processing**
  - ✓ Histogram equalization (global & local)
  - ✓ Histogram matching
- **Multi-image averaging**
  - ✓ Simple averaging
  - ✓ Optimal/adaptive averaging

## Reading from textbook

Chapter 3: Page 120-153

## Food for thought

- For the bit map, how will the image histogram change if we remove the lowest-order bit?
- When we perform histogram equalization, why is it very rare that a perfectly flat histogram can be obtained for the resulting image?
- Why histogram matching often won't result in the exact histogram as the template image?
- When using averaging of N images to reduce noise, if the standard deviation of the Gaussian noise is  $S$ , what is the standard deviation of the noise in the noise-reduced image?

<https://www.youtube.com/watch?v=-RDYHXwJILY>

- In what scenario is local histogram equalization beneficial?
- If we want to improve image contrast, will gamma correction be a good choice? compared to histogram equalization?