SLIIT

Discover Your Future

[redacted]

IT4130 - Image Understanding and Processing

Lecture 03 : Image Enhancement

SLIIT
FACULTY OF COMPUTING

Image Enhancement

On completion the students will be able to:

- Necessity of Image Enhancement
- Spatial Domain Operations
  - Point Processing
  - Histogram based techniques
  - Mask Processing
- Frequency Domain Operations

Spatial Domain vs. Frequency Domai[redacted]

- Spatial domain

  directly process the intensity values of the image plane.

- Frequency domain

  process the transform coefficients, does not directly process the intensity values of the image plane

Different Enhancement Techniques

- Enhancement techniques fall under two broad categories:
- Spatial Domain Techniques:

  - Work on Image Plane itself

  - Direct manipulation of pixels in an image
- Frequency Domain Techniques

  - Modify Fourier Transform coefficients of an image

  - Take inverse Fourier Transform of the modified coefficients to obtain the enhanced image

Image Enhancement

Processing an image to enhance certain features of the image

The result is more suitable than the original image for certain specific applications

- Processing techniques are very much problem oriented
- Best technique for enhancement of X-Ray image may not be the best for enhancement of microscopic images

Intensity Transformations

- Spatial domain processing takes the operation in the form of

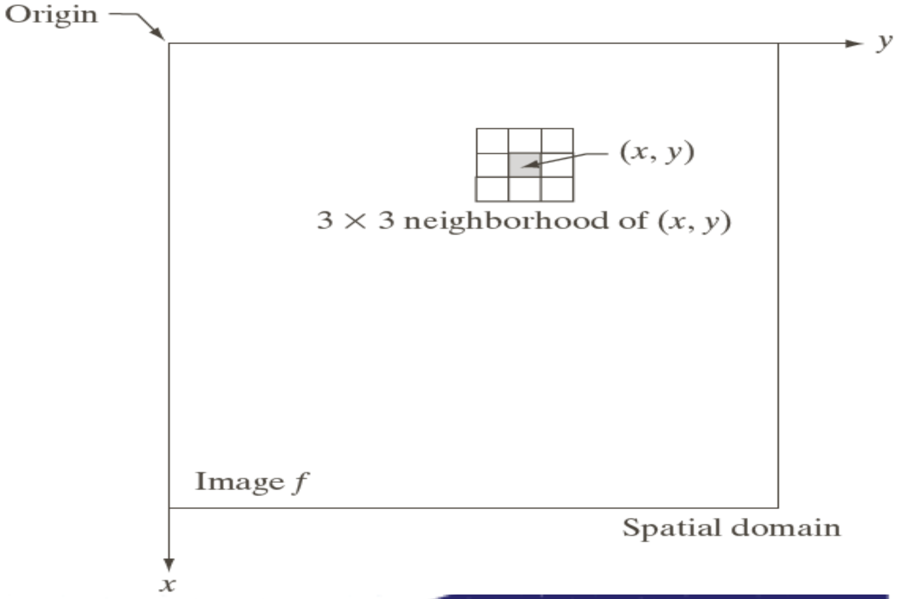
$$g(x) = T\,[\,f\,(x)]$$

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

- T is an operator on f defined over a neighborhood of point (x,y)
- The operator can apply to a single image or to a set of images

Intensity Transformations

- The region containing the point is a neighborhood of (x,y)
- Depending on the neighborhood image processing can be done as
  - Point processing
  - Local processing
  - Global processing



Point Processing

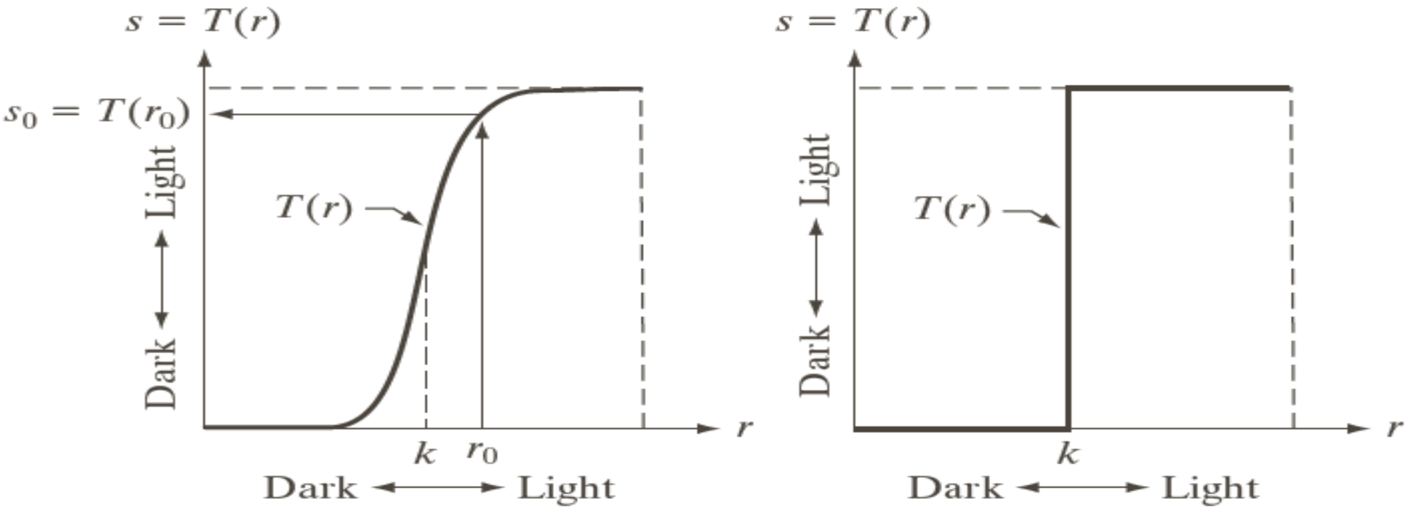
- The smallest neighborhood of a pixel is the pixel itself, 1×1 in size
- *T* becomes an *intensity transformation function* of the form

$$s = T(r)$$

where *s* and *r* represent the intensity of *g* and *f* at any point (x,y)

- T is also called a *gray-level* or *mapping function*

Example: **Point Processing**

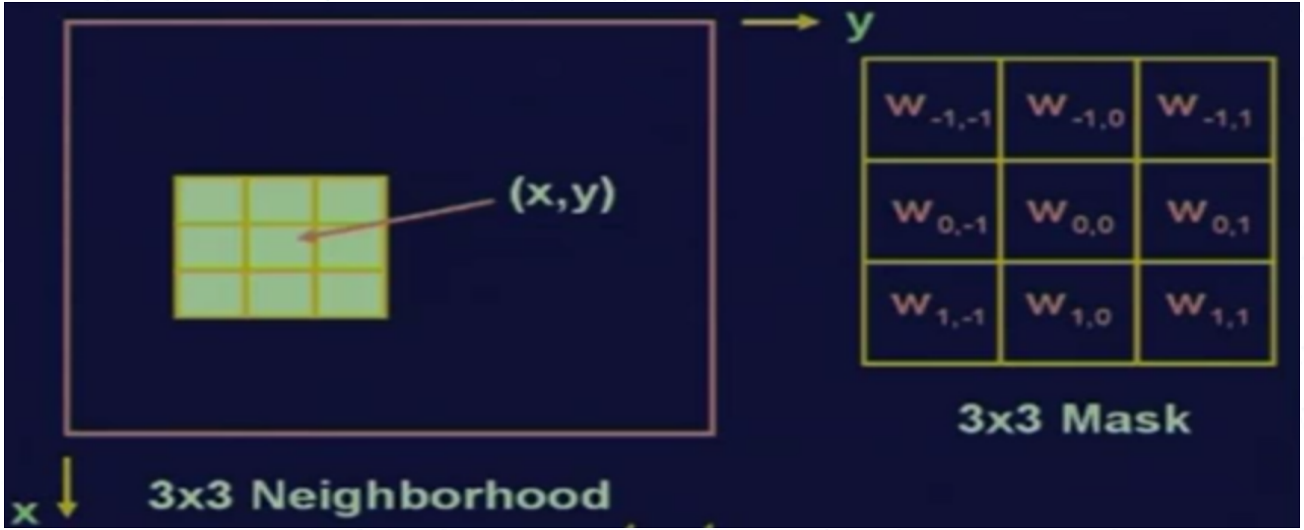


a b

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

Mask Operations

- When the neighborhood size is larger than 1.



$$g(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} w_{i,j}\, f(x+i, y+j)$$

Image Negative

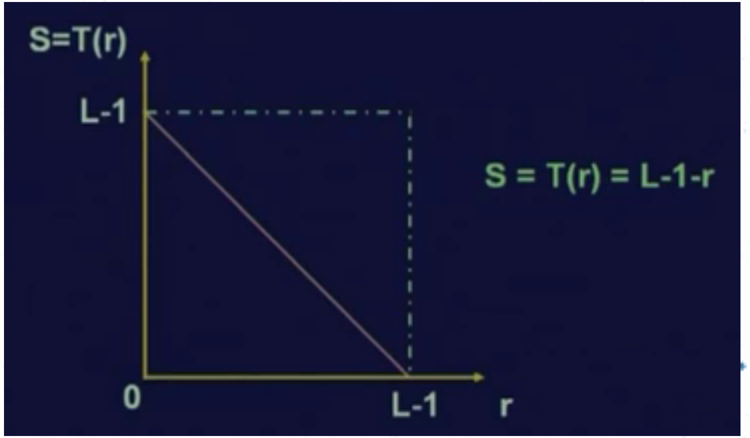
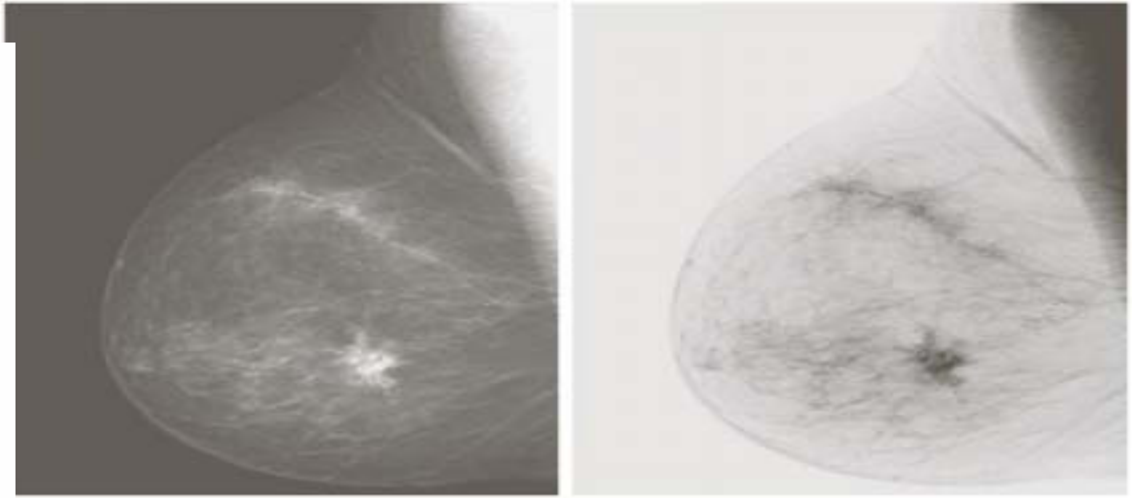


Image Negative

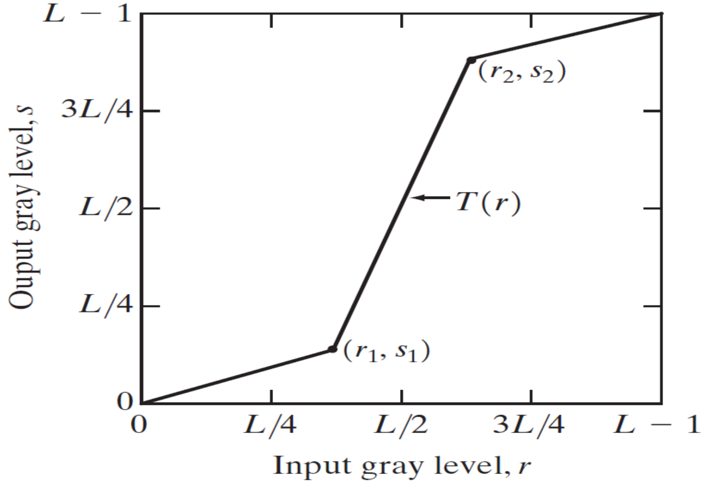
a b

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.)



Contrast stretching

- expands the range of intensity levels in an image, so it spans a given (full) intensity range
- Control points (r1,s1) and (r2,s2) control the shape of the transform T(r)
- r1=r2, s1=0 and s2=L-1 yields a *thresholding function*



Contrast stretching

- Original image
- Draw the plot for

$r_1 = p_1$, $r_2 = p_2$, $s_1=0$, $s_2 =L-1$

- Write the algorithm and draw the plot for

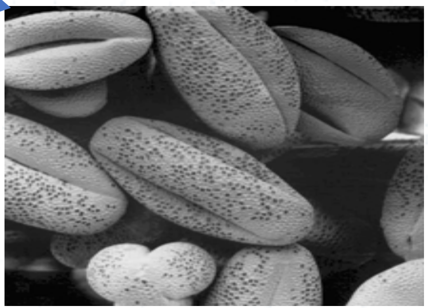
$r_1 = r_2 = m$, $s_1=0$,

$s_2 =L-1$

m is the mean gray

level in the image

- For image with intensity range [50 - 150] What should (r1,s1) and (r2,s2) be to increase the dynamic range of the image to [0 - 255]?

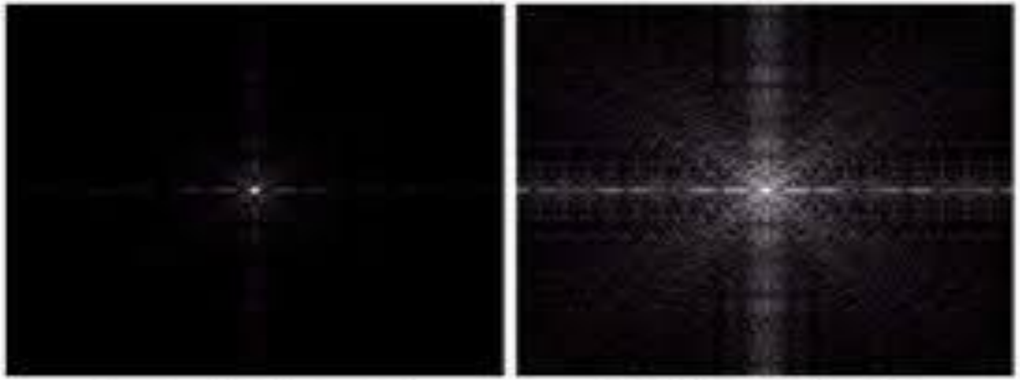


SLIIT FACULTY OF COMPUTING

Log Transformations

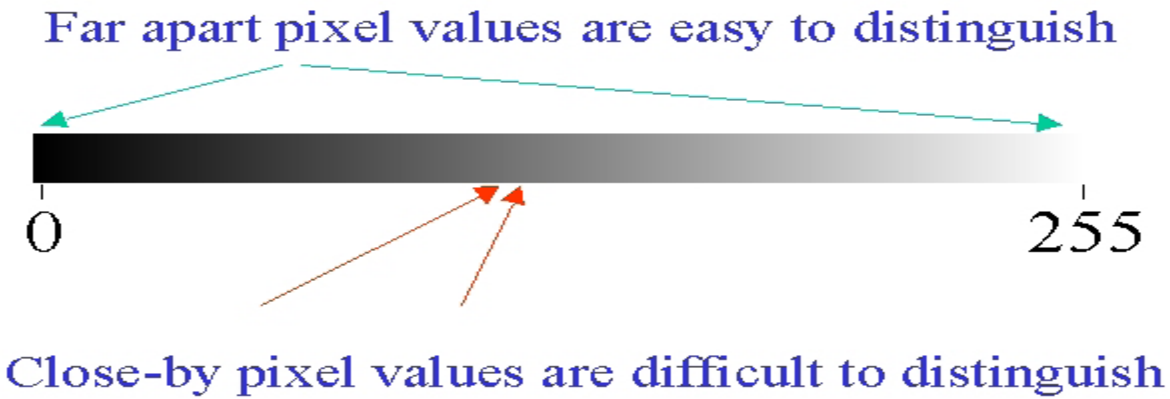
- Maps
  - a narrow range of low intensity values in input to a wider output range and
  - A wider range of high intensity values to a narrow output range

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



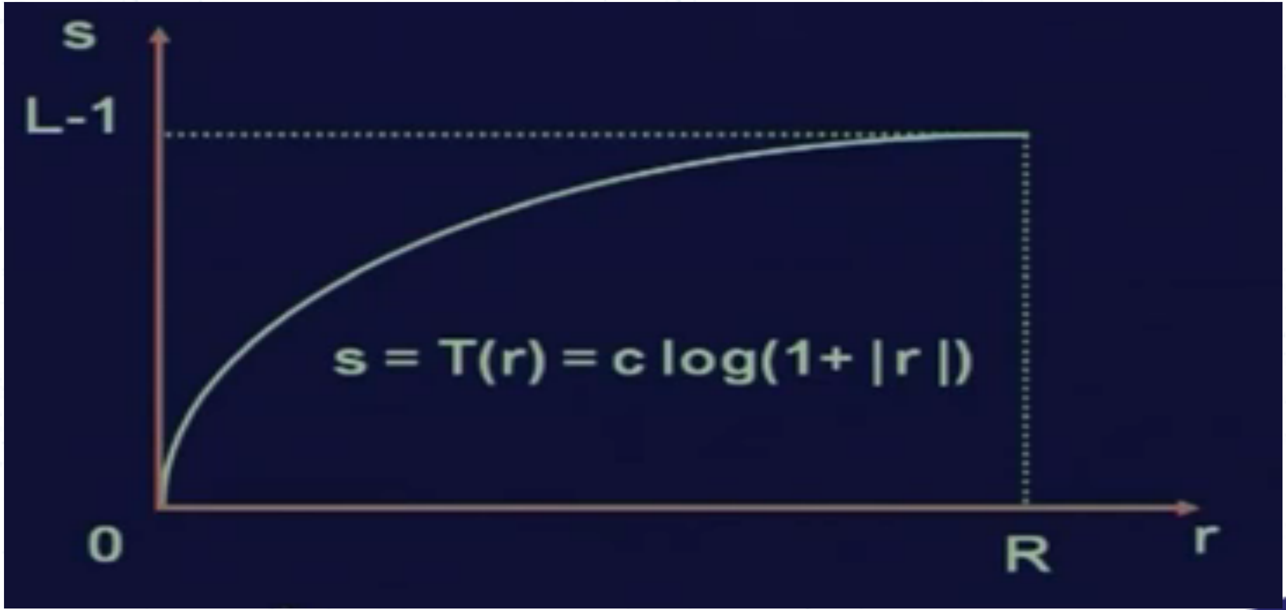
[redacted]

Contrast Enhancement

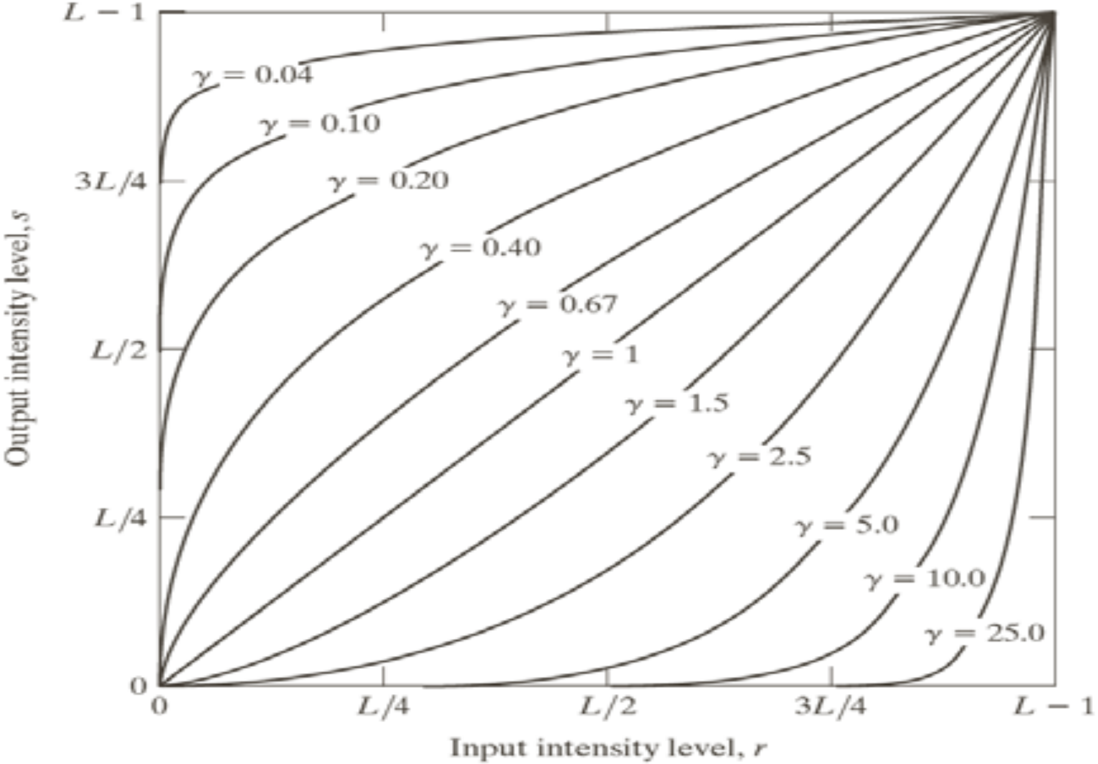


- Expanding the dynamic range of input pixels are achieved by contrast enhancement.
- Low contrast images can result from poor illumination, low density range imaging (sensors), wrong settings in image acquisition.

Dynamic Range Compression



Power-Law (Gamma) Transformations



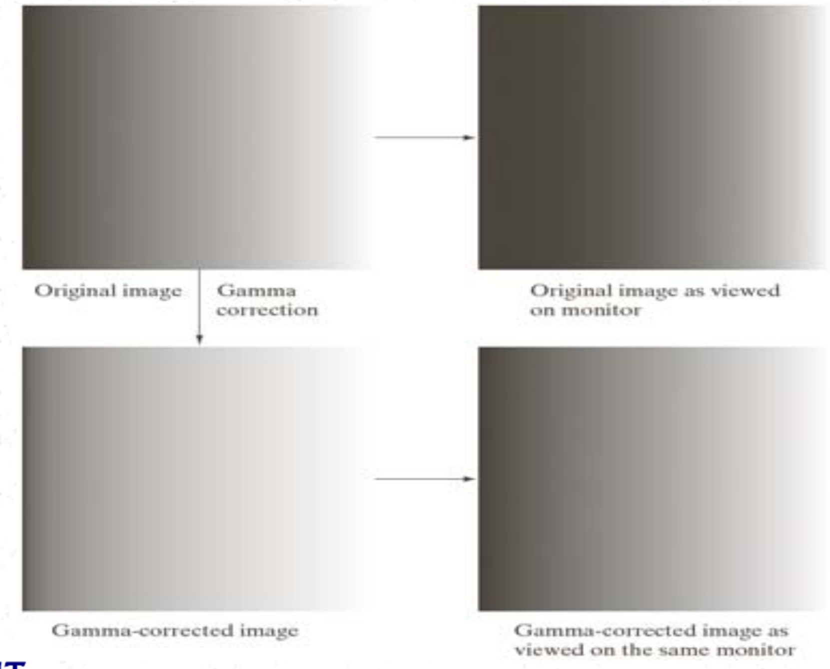
$$s = cr^{\gamma}$$

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

Gamma Correction

- Many devices used for image capture, display and printing respond according to a power law
- The process of correcting for the power-law response is referred to as *gamma correction*
- Example:
  - CRT devices have an intensity-to-voltage response that is a power function (exponents typically range from 1.8-2.5)
  - Gamma correction in this case could be achieved by applying the transformation $s=r^{1/2.5}=r^{0.4}$

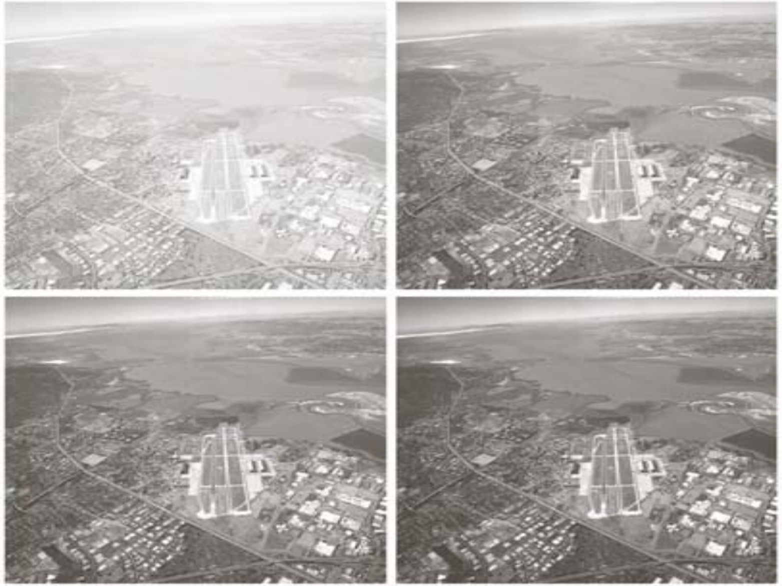
Power-Law - Gamma correction



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).

Power-Law – Contrast enhancement



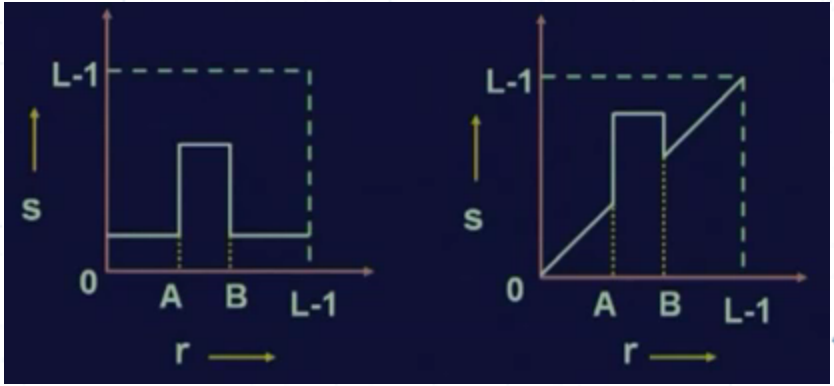
a	b
c	d

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.)

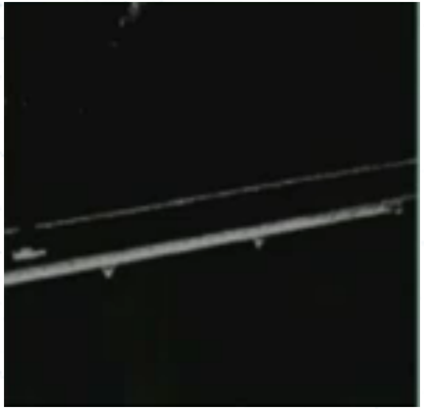
Gray-level Slicing

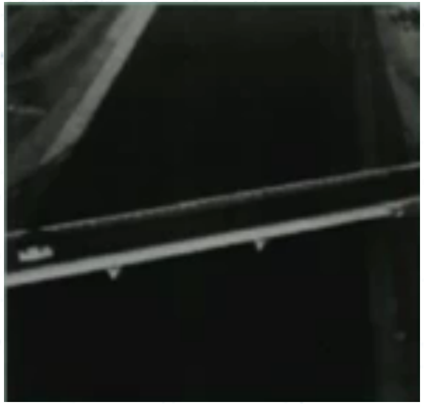
- Used to highlight a specific range of intensities in an image that might be of interest
- Two common approaches
  - Set all pixel values within a range of interest to one value (white) and all others to another value (black)
    - Produces a binary image
  - Brighten (or darken) pixel values in a range of interest and leave all others unchanged

Gray-level Slicing

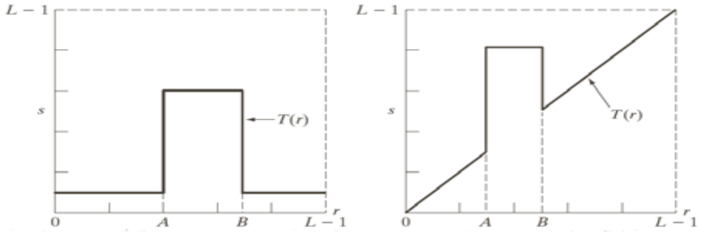


Gray-level Slicing





Intensity-level Slicing - example



a b

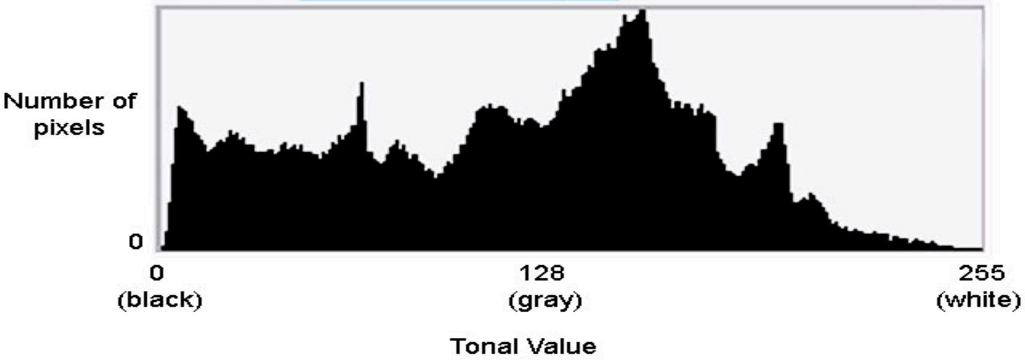
FIGURE 3.11 (a) This transformation highlights intensity range [A, B] and reduces all other intensities to a lower level. (b) This transformation highlights range [A, B] and preserves all other intensity levels.

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.)

Histogram

- The histogram charts the tonal values of pixels in an image.



Histogram Processing

- The *histogram* of a digital image, *f*, (with intensities $[0, L\text{-}1]$) is a discrete function

  $$h(r_k) = n_k$$

  $r_k = k^{\text{th}}$ intensity value

  n_k is the number of pixels in f with intensity r_k

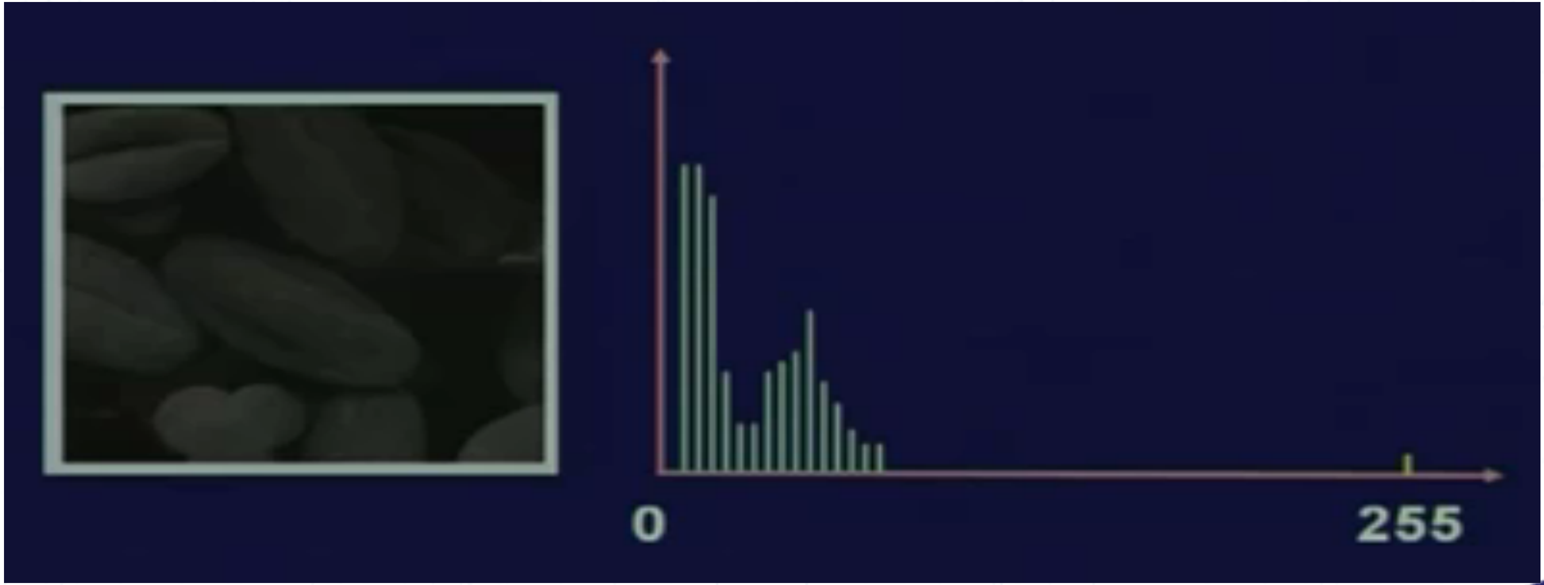
- Normalizing the histogram is common practice
  - Divide the components by the total number of pixels in the image
  - Assuming an MxN image, this yields

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

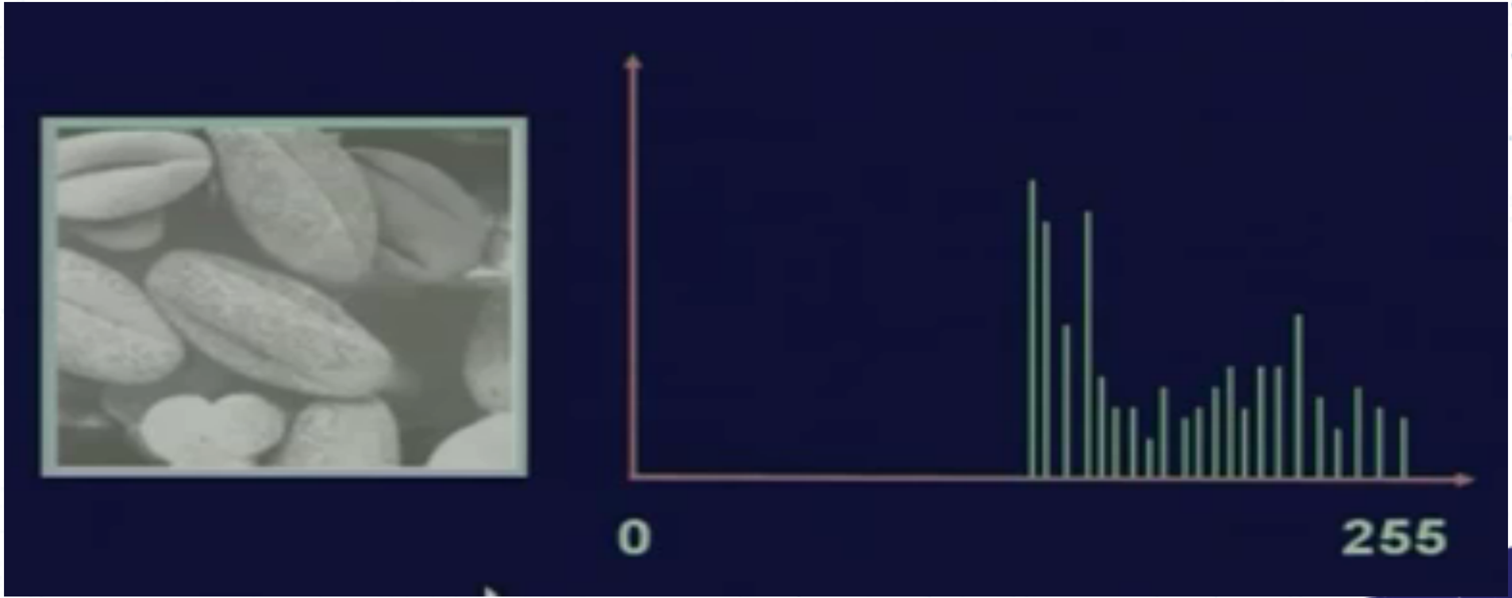
  - $p(r_k)$ is an estimate of the probability of occurrence of intensity level r_k in an image

    $$\Sigma\, p(r_k) = 1$$

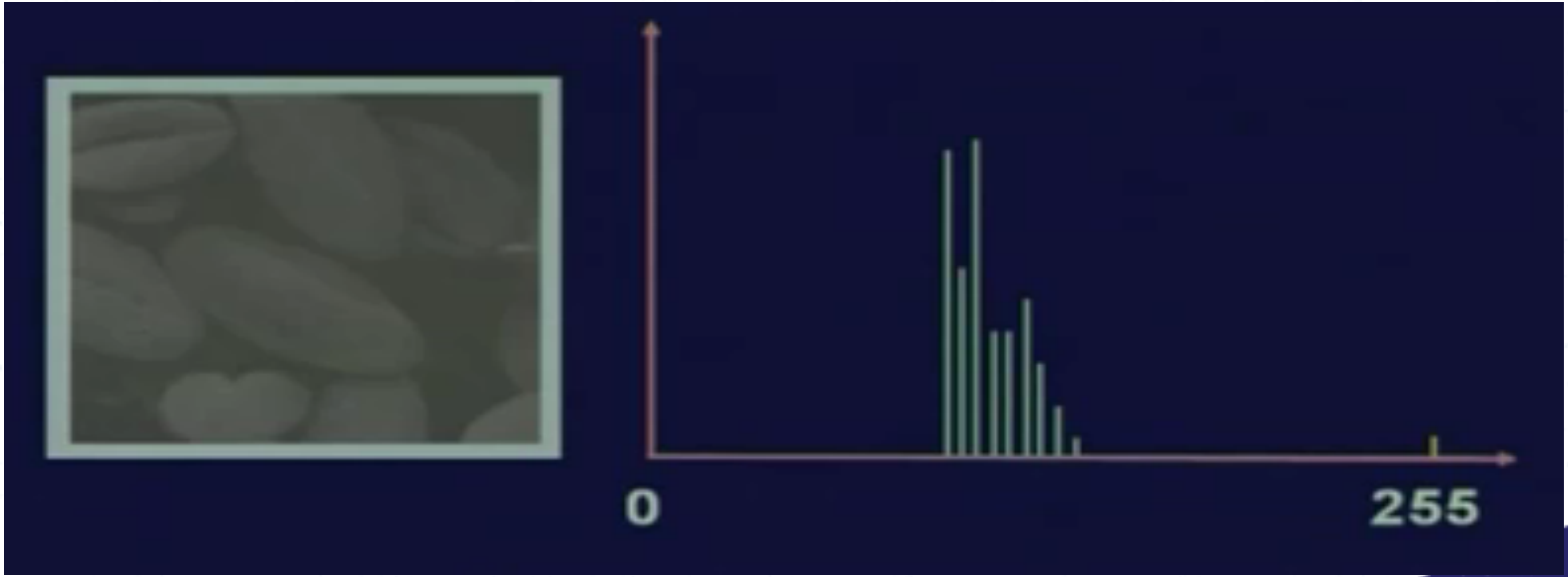
Dark Image



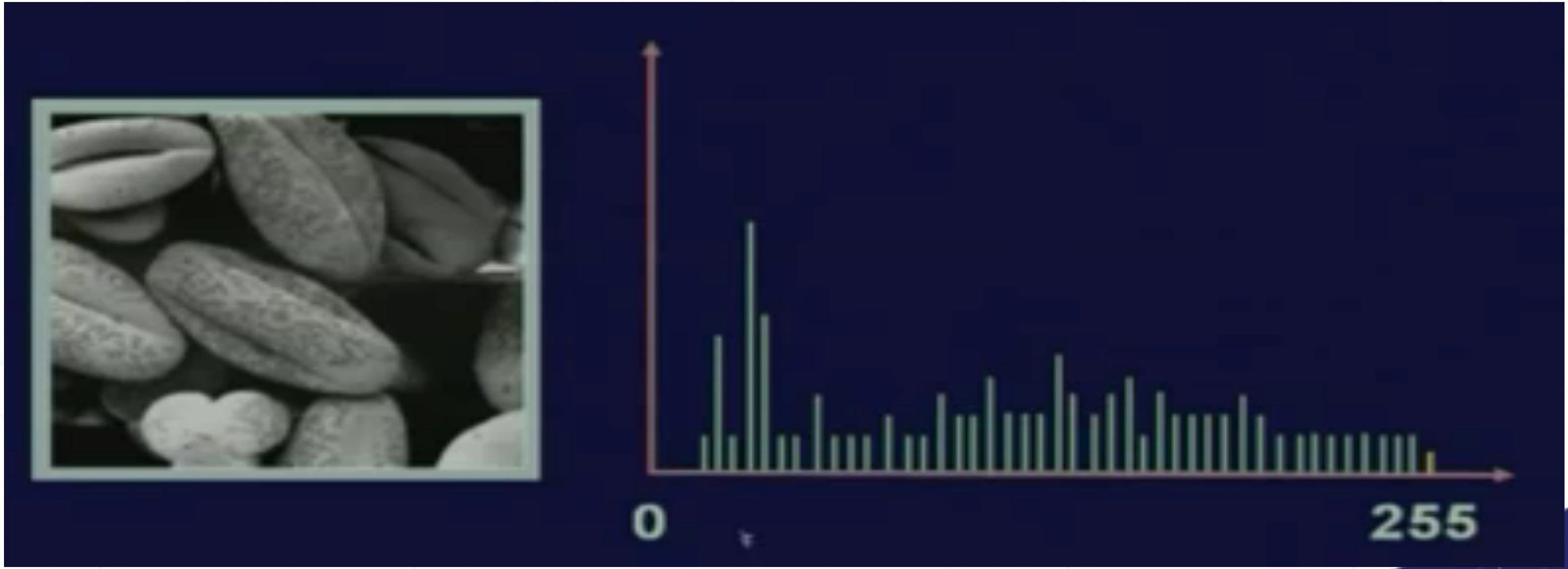
Bright Image



Low Contrast Image



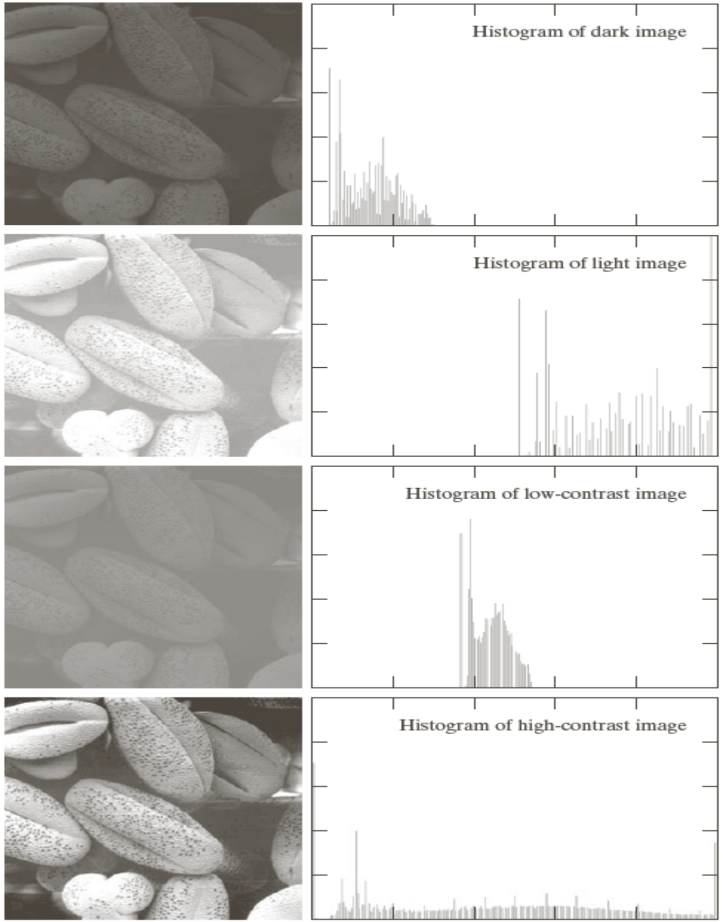
High Contrast Image



- An image has low contrast when the complete range of possible
- values is not used. Inspection of the histogram shows this lack of contrast.
- Histograms commonly viewed in plots as

$$h(r_k) = n_k \text{ versus } r_k$$

$$p(r_k) = n_k / MN \text{ versus } r_k$$



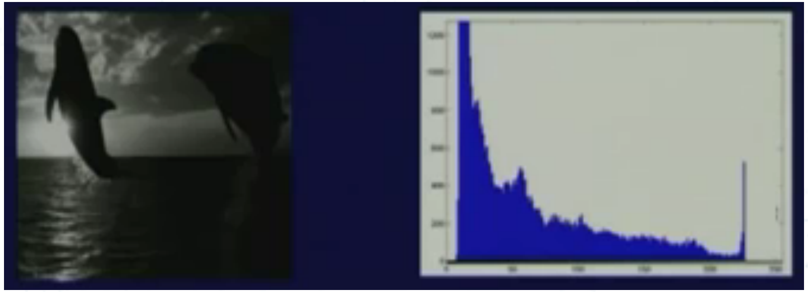
Applications of Histogram Processin[redacted]

- Image enhancements
- Image statistics
- Image compression
- Image segmentation

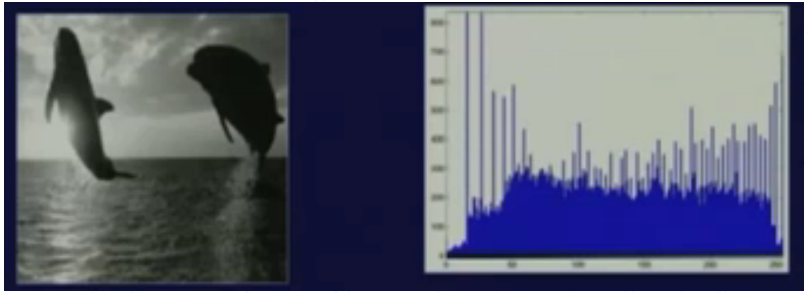
- Gives useful information in analyzing the properties of images.
- Simple to calculate in software
- Economic hardware implementations
  - Popular tool in real-time image processing

Histogram Equalization

Input Image



Equalized Image



Not exactly Uniform !!!!!!

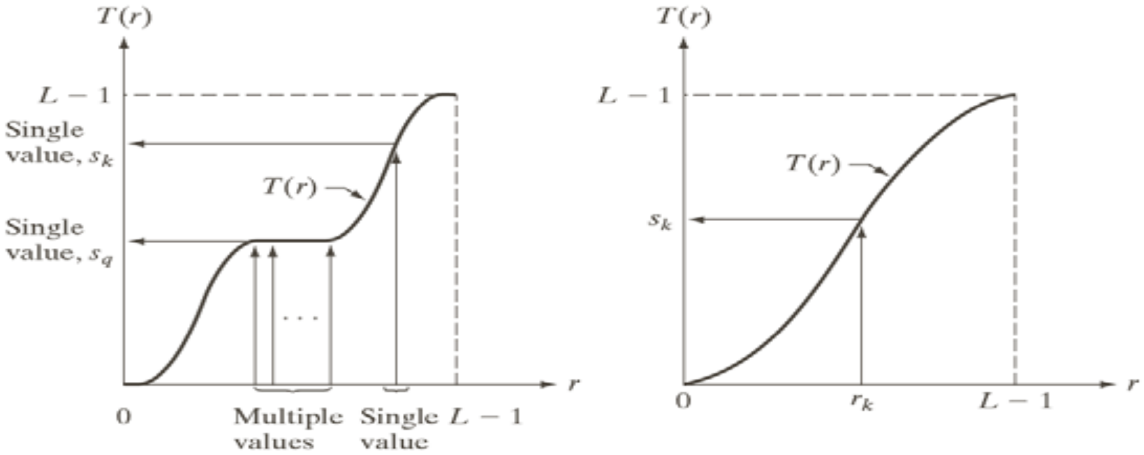
Histogram Equalization

- The intensity transformation function is of the form

  $s = T(r) \; 0 \leq r \leq L - 1$

- An output intensity level s is produced for every pixel in the input image having intensity r

  $0 \leq T(r) \leq L\text{-}1 \text{ for } 0 \leq r \leq L\text{-}1$



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.

Histogram Equalization

The intensity levels in an image may be view[redacted] random variables in the interval [0, L-1].

Let $p_r(r)$ and $p_s(s)$ denote the probability density function (PDF) of random variables r and s.

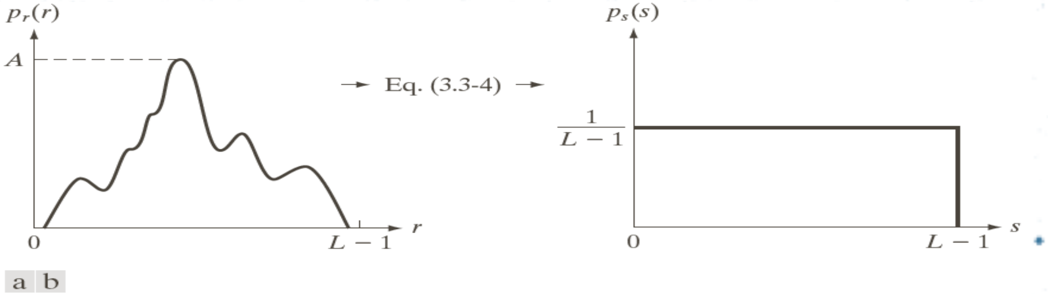


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

Histogram Equalization

- Histogram equalization requires construction of a transformation function s_k

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

$$s_k = T(r_k) = \frac{(L-1)}{M \times N} \sum_{j=0}^{k} n_j$$

- This yields an s with as many elements as the original image's histogram.
- The values of s will be in the range [0,1]. For constructing a new image, s would be scaled to the range [1,256]

Example: Histogram Equalization

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in following table.

Get the histogram equalization transformation function and give the $p_s(s_k)$ for each s_k.

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

Example: Histogram Equalization

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

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

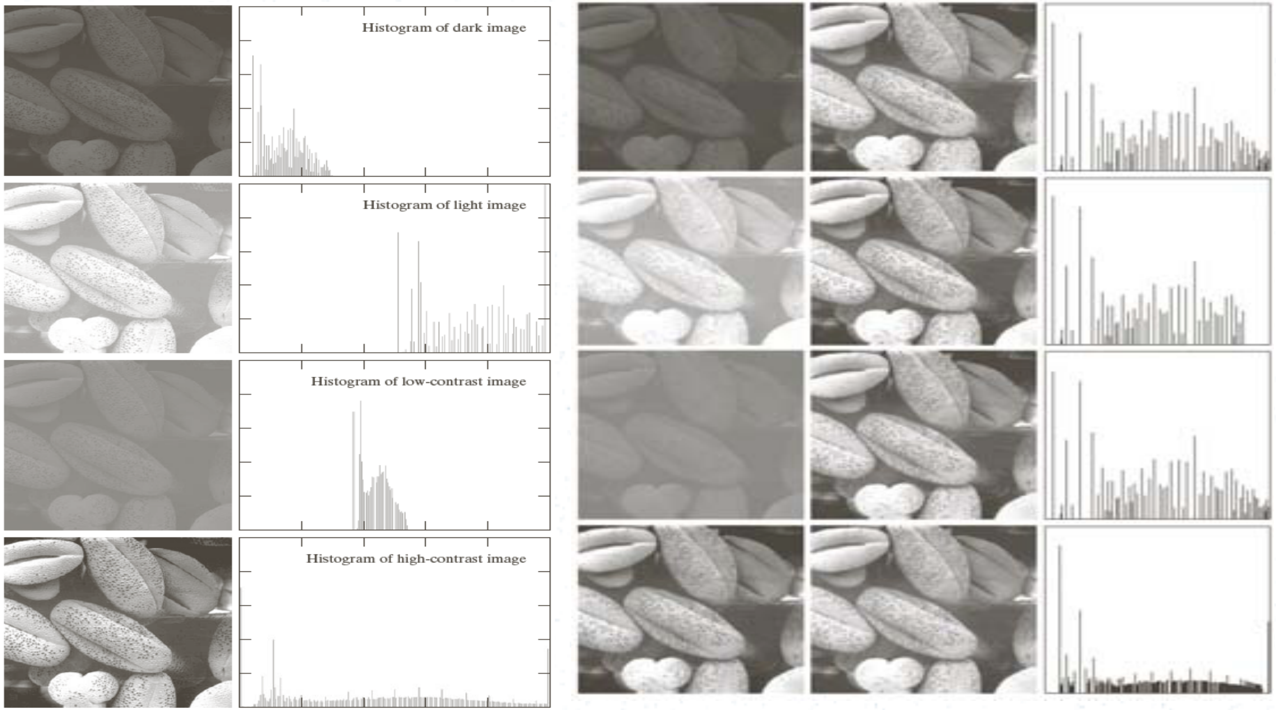
$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7 \times (0.19 + 0.25) = 3.08 \quad \rightarrow 3$$

$$s_2 = 4.55 \quad \rightarrow 5 \qquad s_3 = 5.67 \quad \rightarrow 6$$

$$s_4 = 6.23 \quad \rightarrow 6 \qquad s_5 = 6.65 \quad \rightarrow 7$$

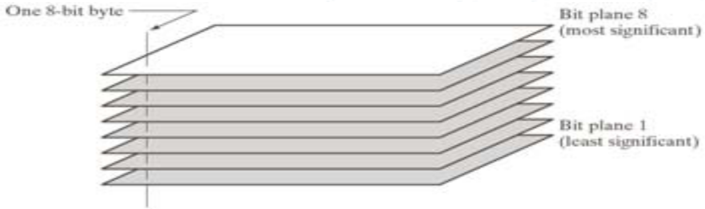
$$s_6 = 6.86 \quad \rightarrow 7 \qquad s_7 = 7.00 \quad \rightarrow 7$$

Example: Histogram Equalization



Bit-plane Slicing

- Pixels are digital values composed of bits
- For example, a pixel in a 256-level gray-scale image is comprised of 8 bits
- We can highlight the contribution made to total image appearance by specific bits
- For example, we can display an image that only shows the contribution of a specific bit plane



SLIIT
FACULTY OF COMPUTING

Bit-plane Slicing - Example

a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

Bit-plane Slicing (continued)

- Bit-plane slicing is useful in
  - Determining the number of bits necessary to quantize an image
  - Image compression
  - Feature extraction [2]

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).

Arithmetic operations

- Arithmetic & logic operations on images used extensively in most image processing applications
  - May cover the entire image or a subset
- Arithmetic operation between pixels *p and q are defined as:*
  - Addition: (*p+q)*
    - Used often for image averaging to reduce noise
  - Subtraction: (*p-q)*
    - Used often for static background removal
  - Multiplication: (*p*q) (or pq, p×q)*
    - Used to correct gray-level shading
  - Division: (*p÷q) (or p/q)*
    - As in multiplication

SUBTRACTION

a b
c d

FIGURE 3.28
(a) Original fractal image.
(b) Result of setting the four lower-order bit planes to zero.
(c) Difference between (a) and (b).
(d) Histogram-equalized difference image.
(Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).

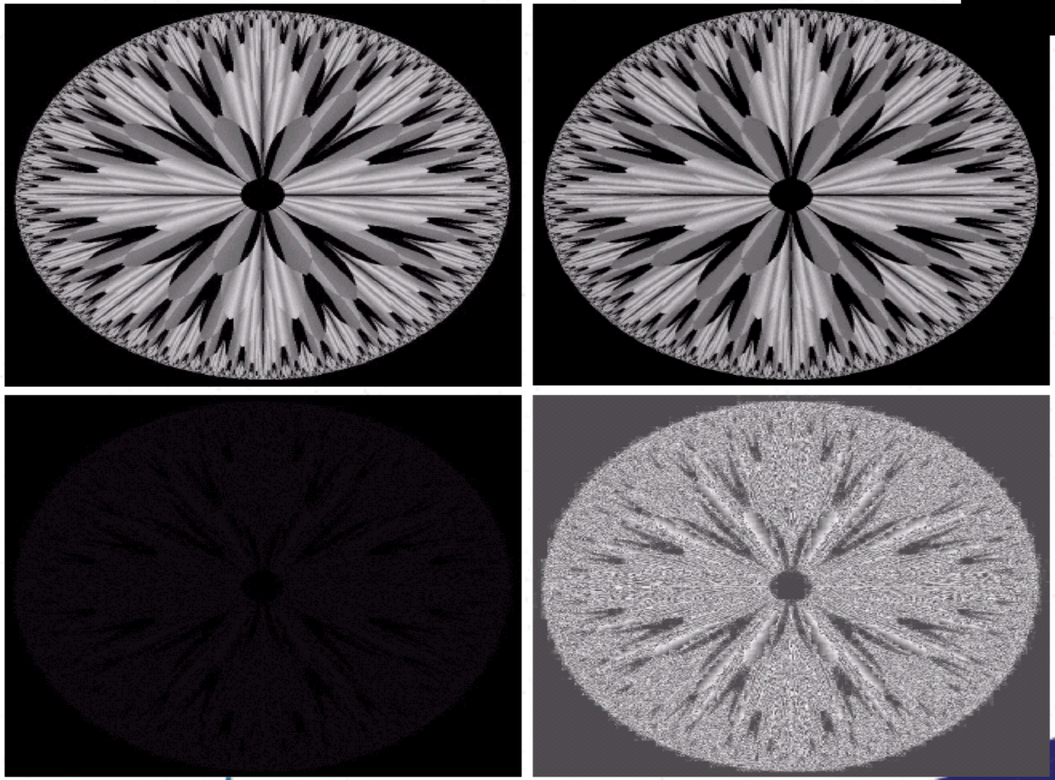
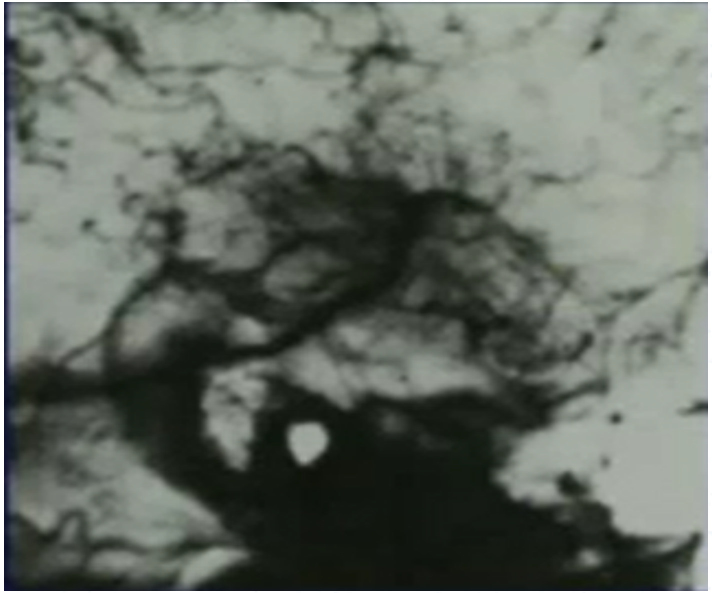
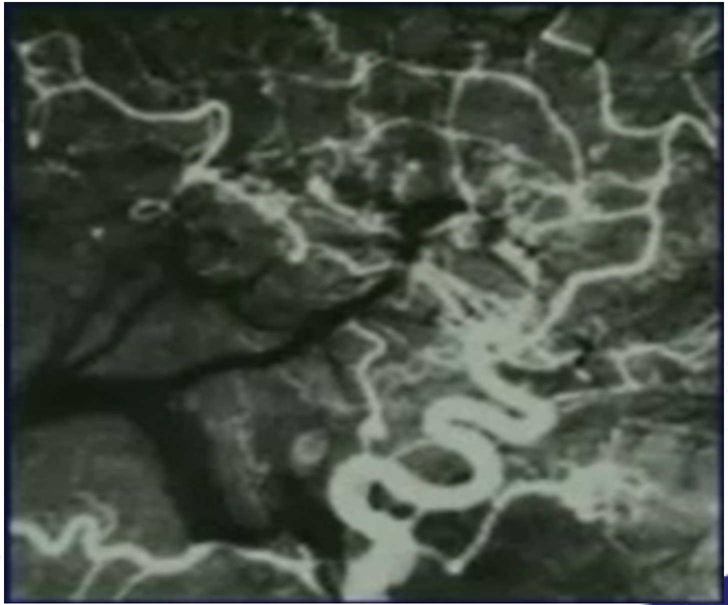


Image Difference

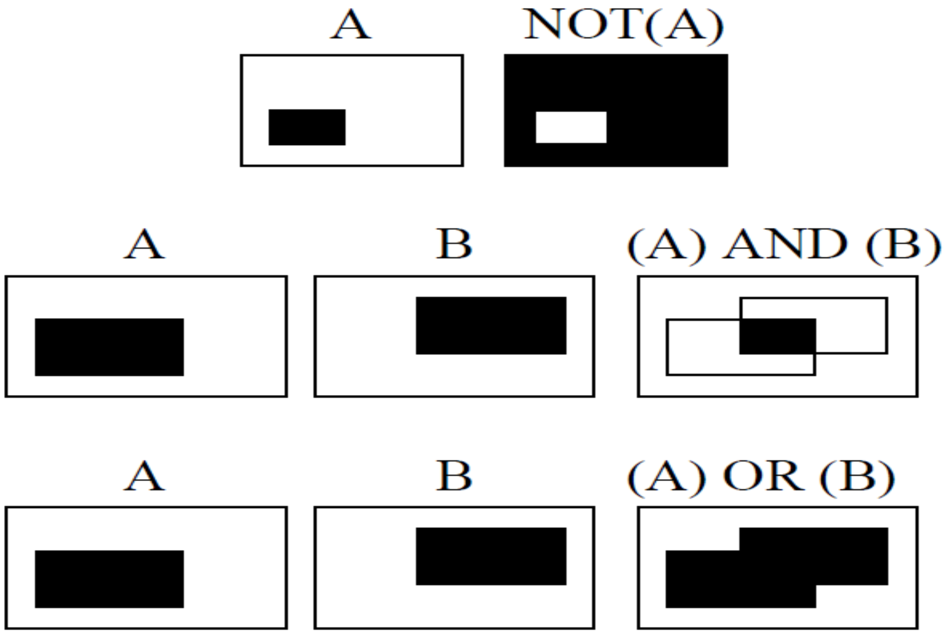




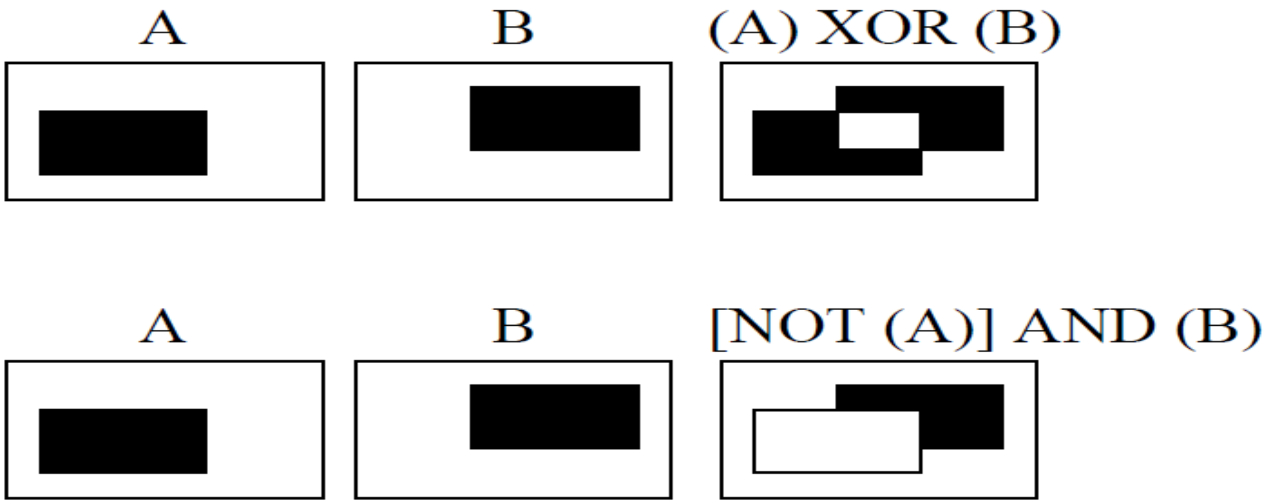
Logic operations

- Logic operation between pixels *p and q are defined as:*
  - AND: *pANDq (also p·q)*
  - OR: *pORq (also p+q)*
  - COMPLEMENT: NOT*q (also q')*
- Form a *functionally complete set*
- Applicable to binary images
- Basic tools in binary image processing, used for:
  - Masking
  - Feature detection
  - Shape analysis

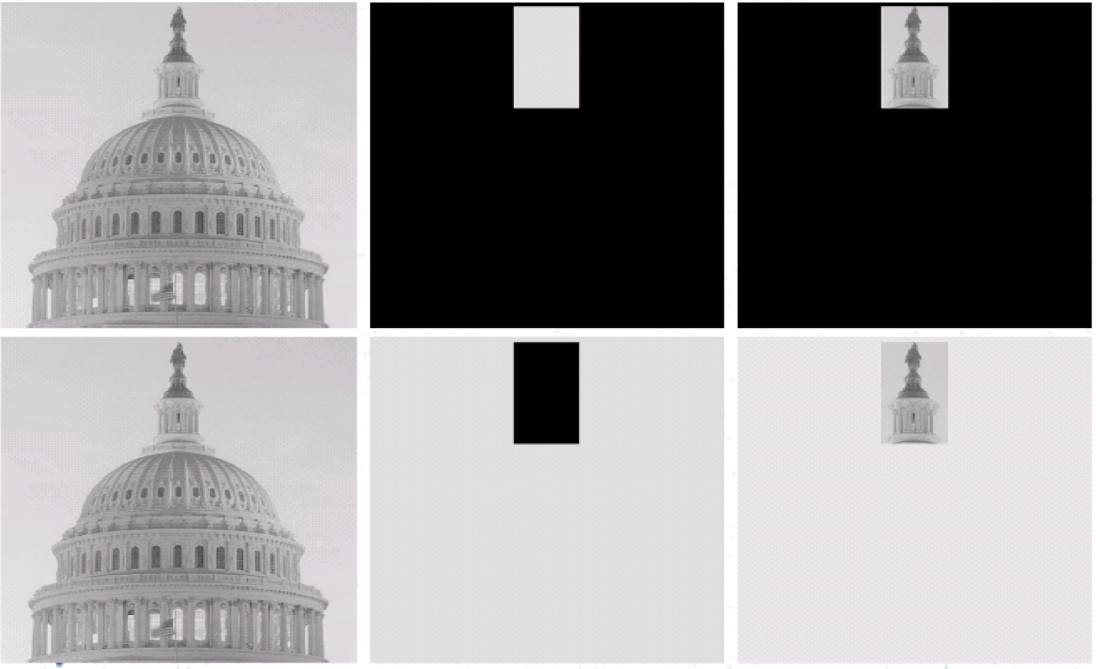
Examples of logic operations



Examples of logic operations (conti[redacted]



AND and OR



a b c
d e f

FIGURE 3.27 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).