

Image Enhancement in the Spatial Domain

Course Leader

Dr. Aruna Kumar S V

Assistant Professor

FET- Computer Science and Engineering

Ramaiah University of Applied Sciences

Bengaluru

arunakumar.cs.et@msruas.ac.in



Image Enhancement in the Spatial Domain

- At the end of this session, students will be able to:
 - Describe various Gray Level Transformation techniques.
 - Understand the importance of Histogram Processing.
 - Identify different Arithmetic and Logical operations for Image Enhancement.
 - Understand the importance First and Second Order Derivatives in Image Enhancement.



Principle Objective of Enhancement

- Process an image so that the result will be more suitable than the original image for a specific application.
- The suitability is up to each application.
- A method which is quite useful for enhancing an image may not necessarily be the best approach for enhancing another images



2 Domains

- Spatial Domain : (image plane)
 - Techniques are based on direct manipulation of pixels in an image
- Frequency Domain :
 - Techniques are based on modifying the Fourier transform of an image
- There are some enhancement techniques based on various combinations of methods from these two categories.



Good images

- For human visual
 - The visual evaluation of image quality is a highly subjective process.
 - It is hard to standardize the definition of a good image.
- For machine perception
 - The evaluation task is easier.
 - A good image is one which gives the best machine recognition results.
- A certain amount of trial and error usually is required before a particular image enhancement approach is selected.



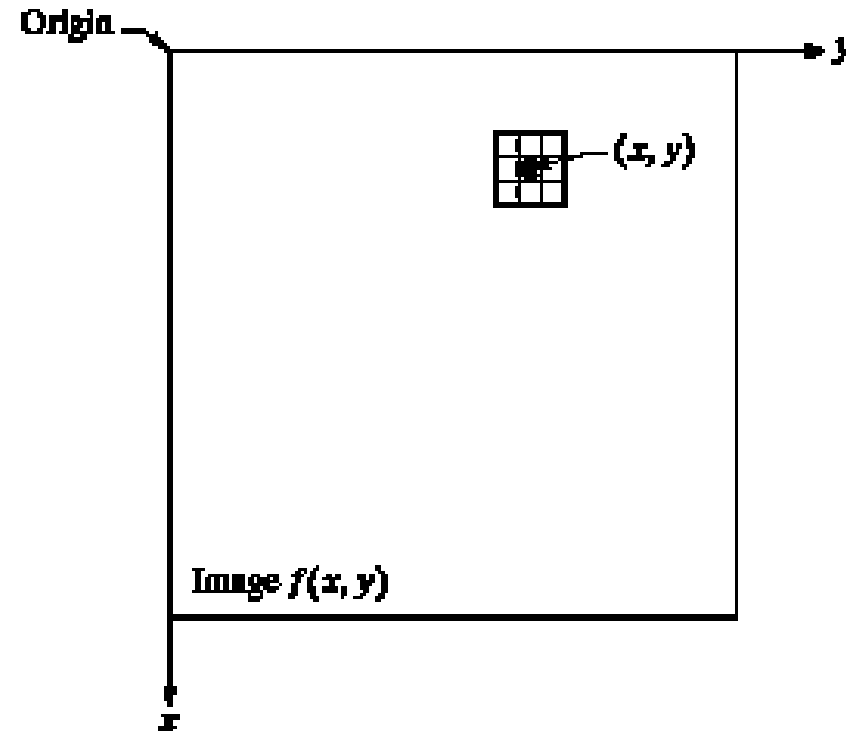
Spatial Domain

- Procedures that operate directly on pixels.

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

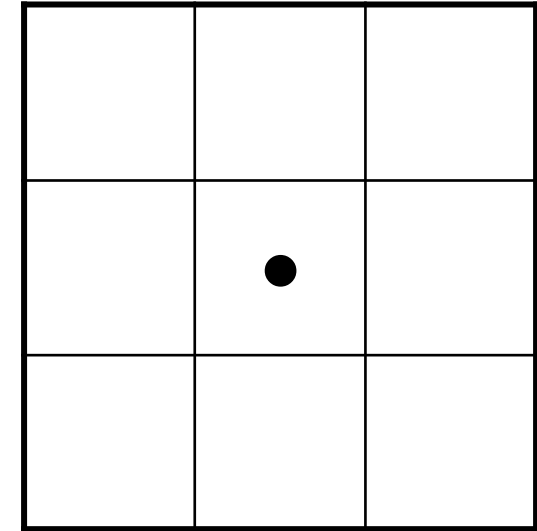
where

- $f(x,y)$ is the input image
- $g(x,y)$ is the processed image
- T is an operator on f defined over some neighborhood of (x,y)



Mask/Filter

- Neighborhood of a point (x,y) can be defined by using a square/rectangular (common used) or circular subimage area centered at (x,y)
- The center of the subimage is moved from pixel to pixel starting at the top of the corner



Point Processing

- Neighborhood = 1x1 pixel
- g depends on only the value of f at (x,y)
- T = gray level (or intensity or mapping) transformation function

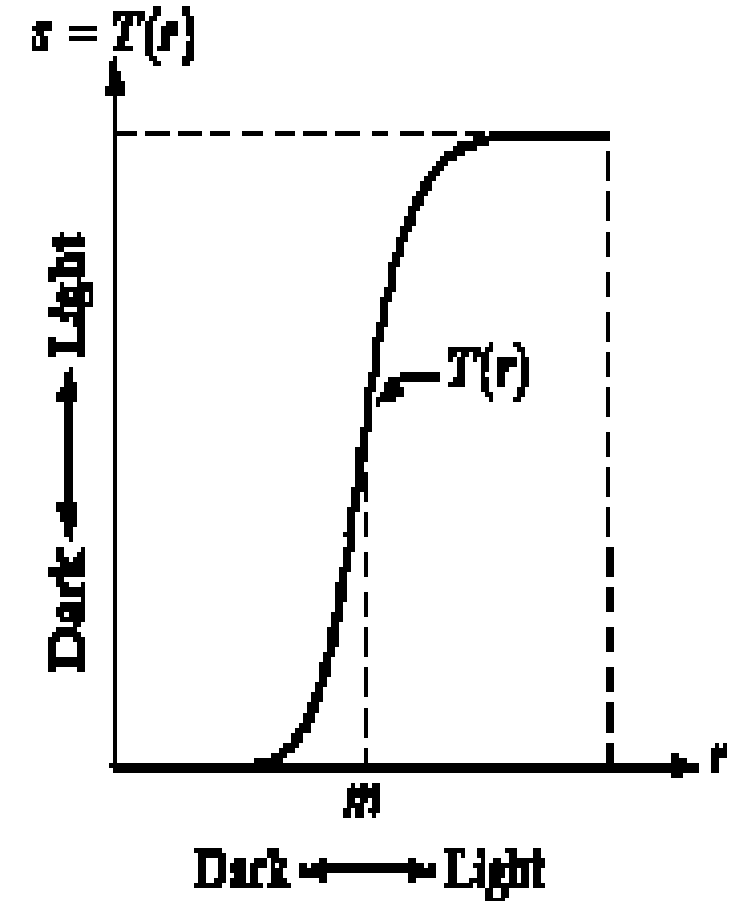
$$s = T(r)$$

- Where
 - r = gray level of $f(x,y)$
 - s = gray level of $g(x,y)$



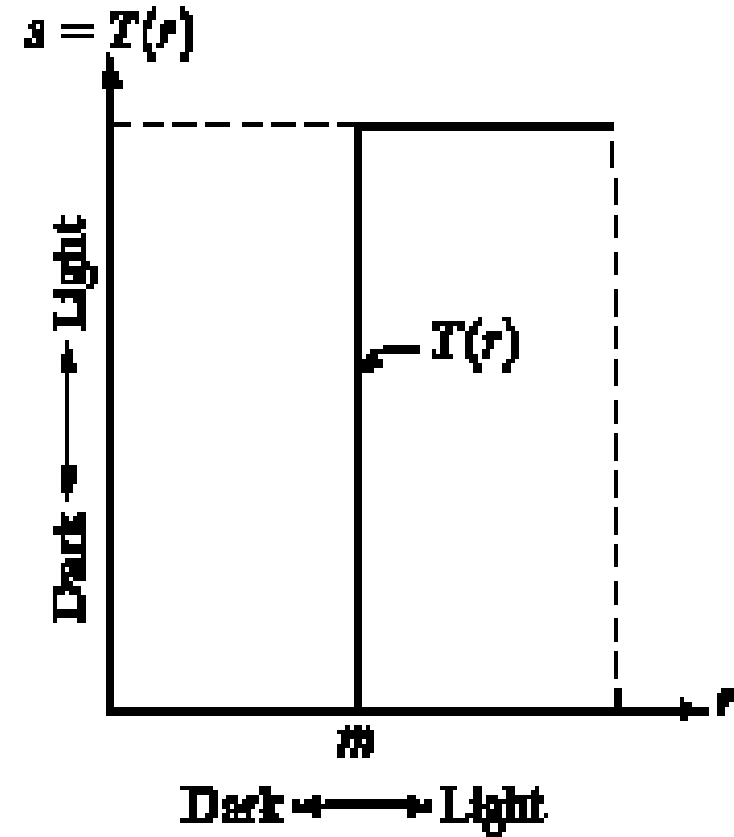
Contrast Stretching

- Produce higher contrast than the original by
 - darkening the levels below m in the original image
 - Brightening the levels above m in the original image



Thresholding

- Produce a two-level (binary) image



Mask Processing or Filter

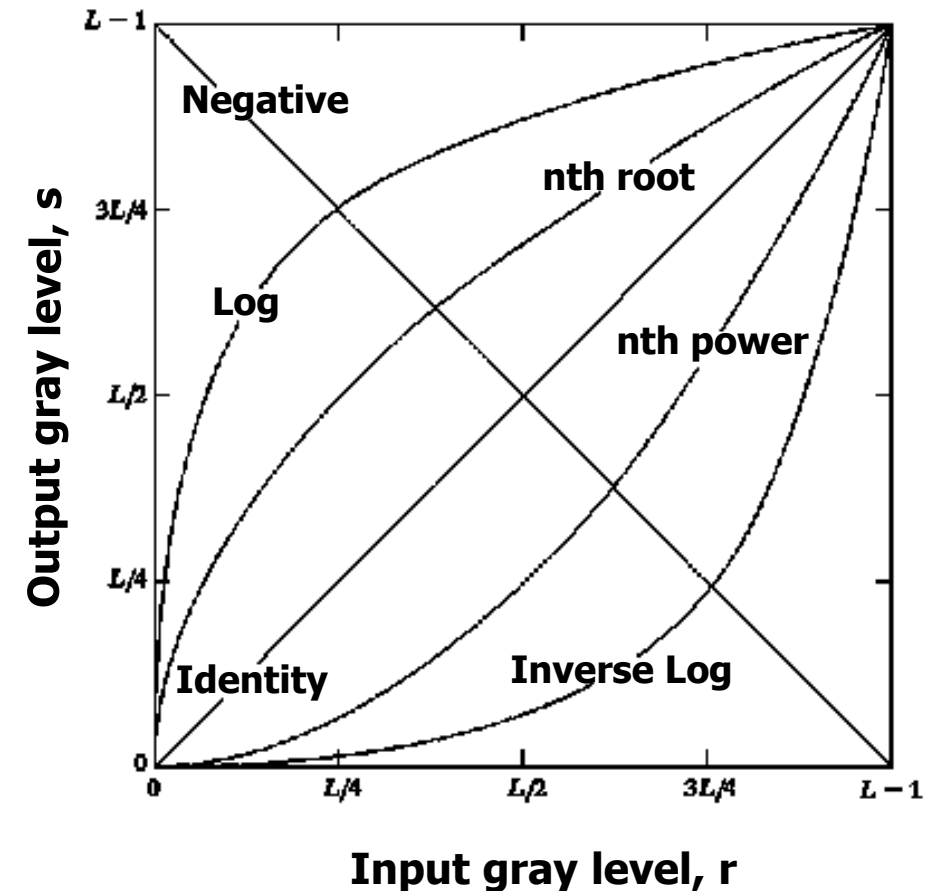
- Neighborhood is bigger than 1x1 pixel
- Use a function of the values of f in a predefined neighborhood of (x,y) to determine the value of g at (x,y)
- The value of the mask coefficients determine the nature of the process
- Used in techniques



– Image Sharpening and Image Smoothing

3 basic gray-level transformation functions

- Linear function
 - Negative and identity transformations
- Logarithm function
 - Log and inverse-log transformation
- Power-law function
 - n^{th} power and n^{th} root transformations



Identity function

- Output intensities are identical to input intensities.
- Is included in the graph only for completeness.

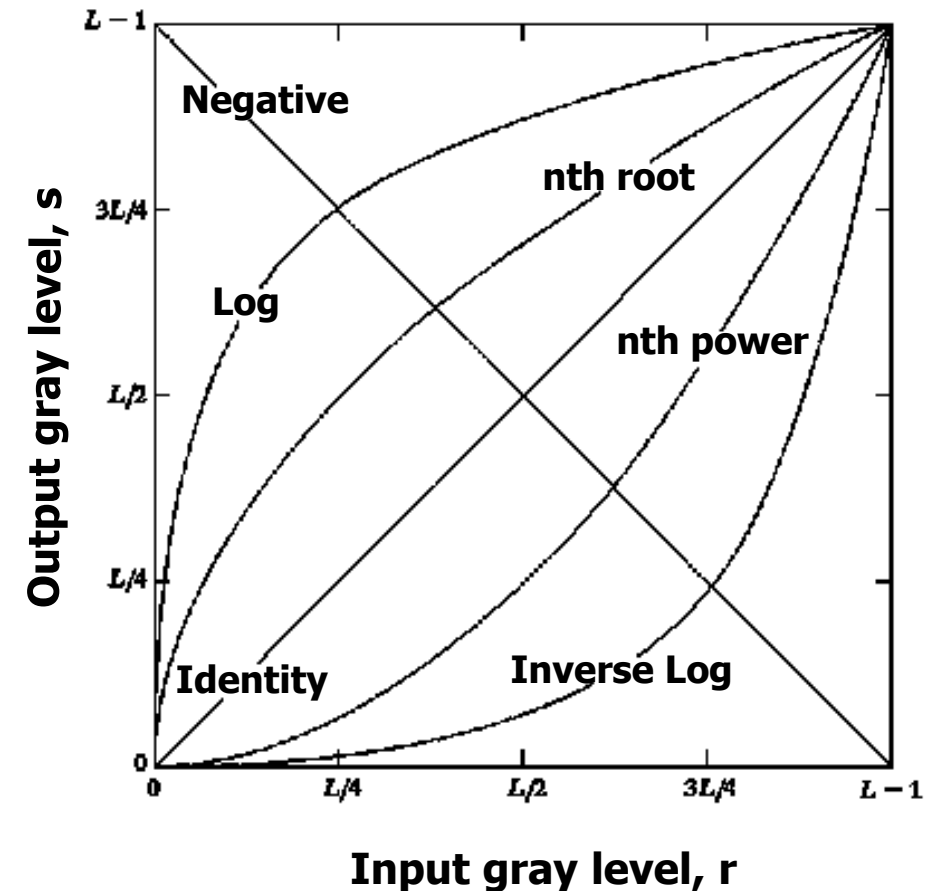


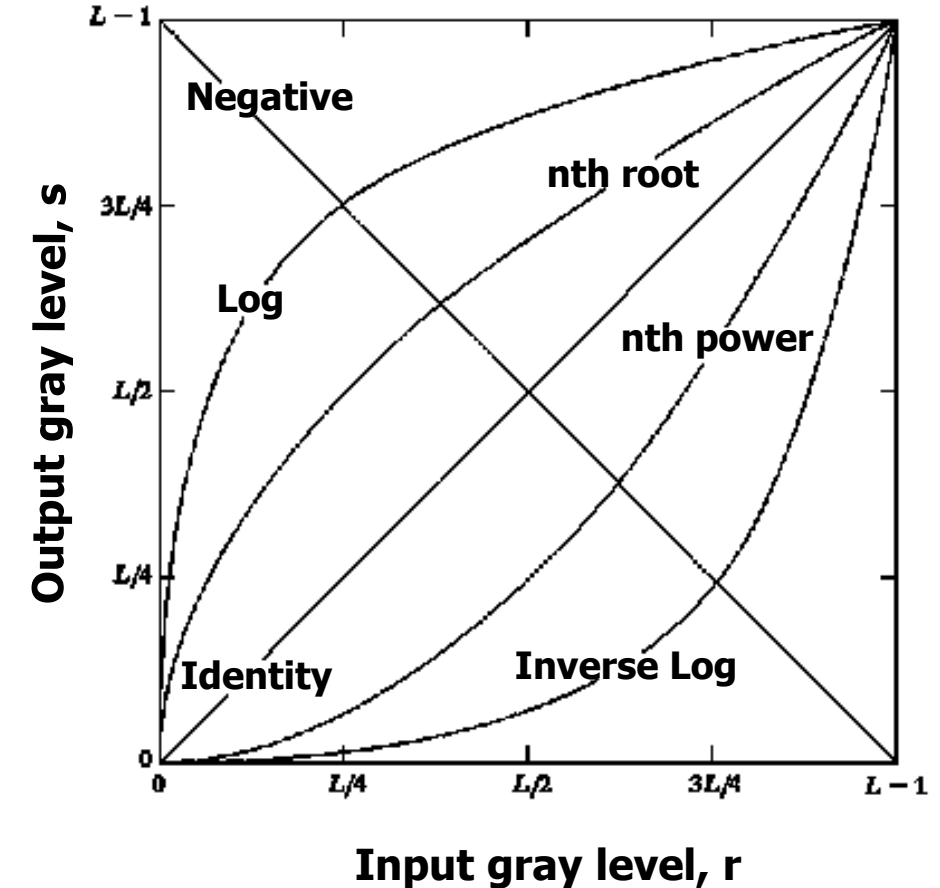
Image Negatives

- An image with gray level in the range $[0, L-1]$ where $L = 2^n$; $n = 1, 2, \dots$

- Negative transformation :

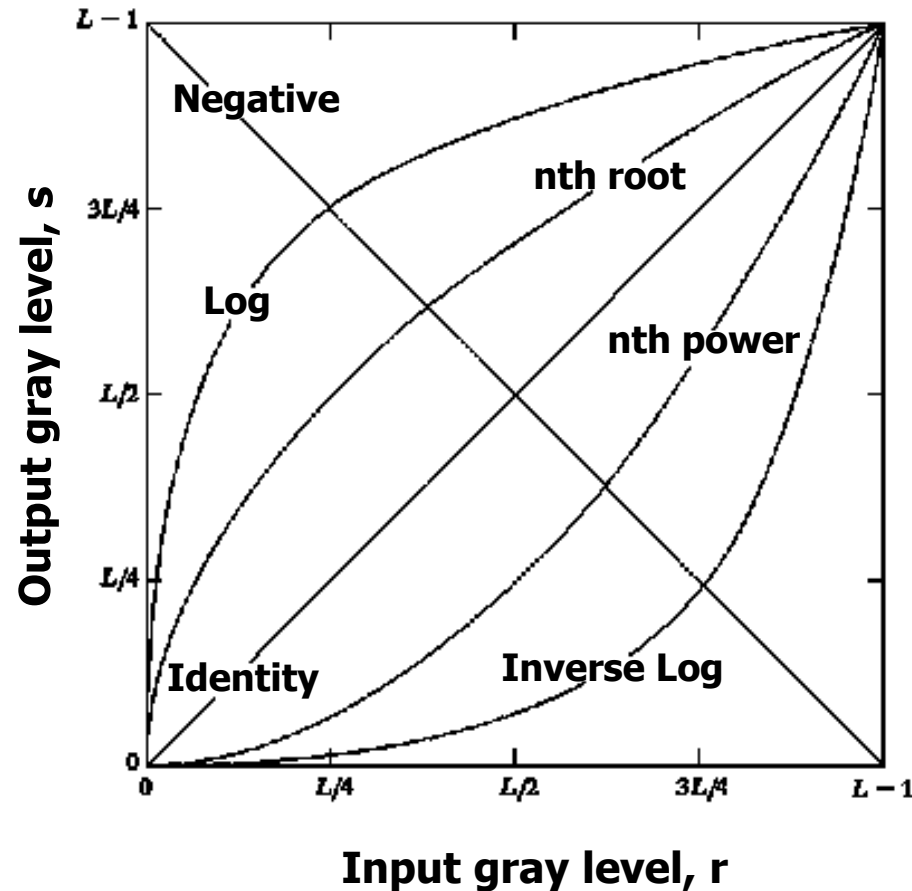
$$s = L - 1 - r$$

- Reversing the intensity levels of an image.
- Suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area dominant in size.





Log Transformations



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

c is a constant and $r \geq 0$

- Log curve maps a narrow range of low gray-level values in the input image into a wider range of output levels.
- Used to expand the values of dark pixels in an image while compressing the higher-level values.

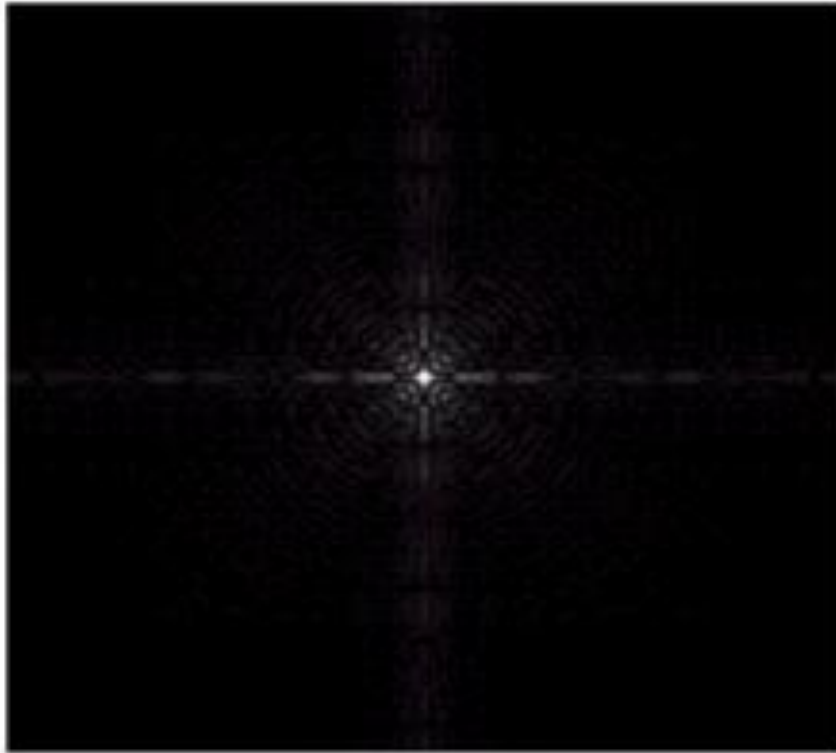


Log Transformations

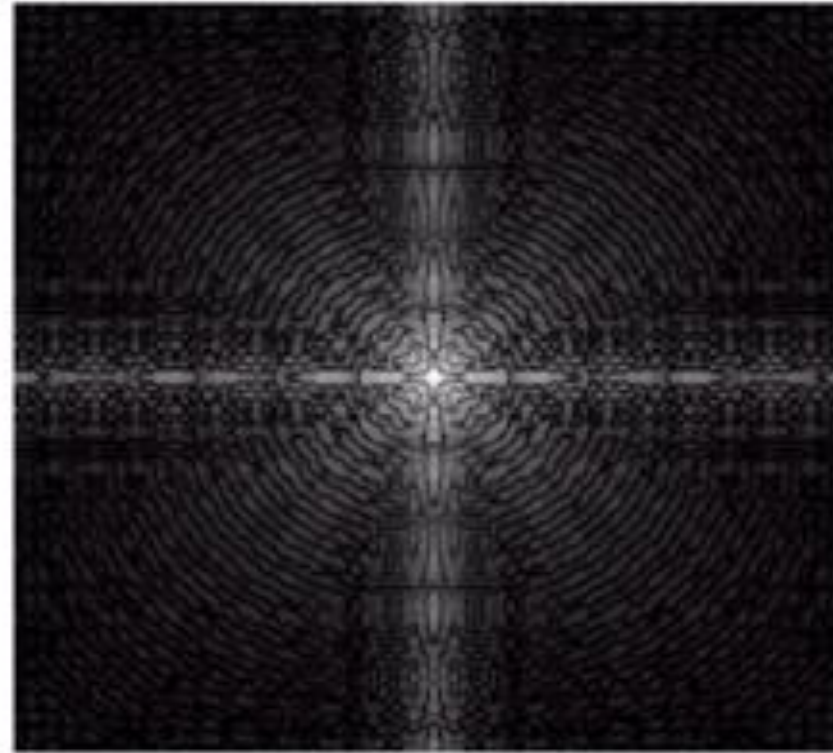
- It compresses the dynamic range of images with large variations in pixel values
- Example of image with dynamic range: Fourier spectrum image
- It can have intensity range from 0 to 10^6 or higher.
- We can't see the significant degree of detail as it will be lost in the display.



Example of Logarithm Image



**Fourier Spectrum with
range = 0 to 1.5×10^6**



**Result after apply the log
transformation with $c = 1$,
range = 0 to 6.2**



Inverse Logarithm Transformations

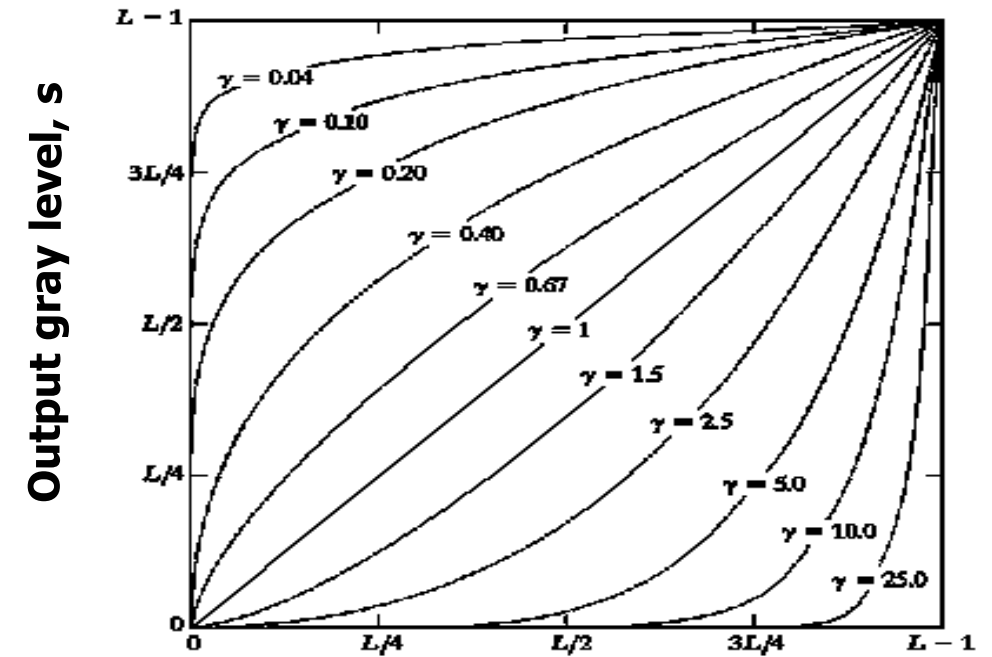
- Do opposite to the Log Transformations
- Used to expand the values of high pixels in an image while compressing the darker-level values.



Power-Law Transformations

$$s = cr^\gamma$$

- c and γ are positive constants
- Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.
- $c = \gamma = 1 \Rightarrow$ Identity function



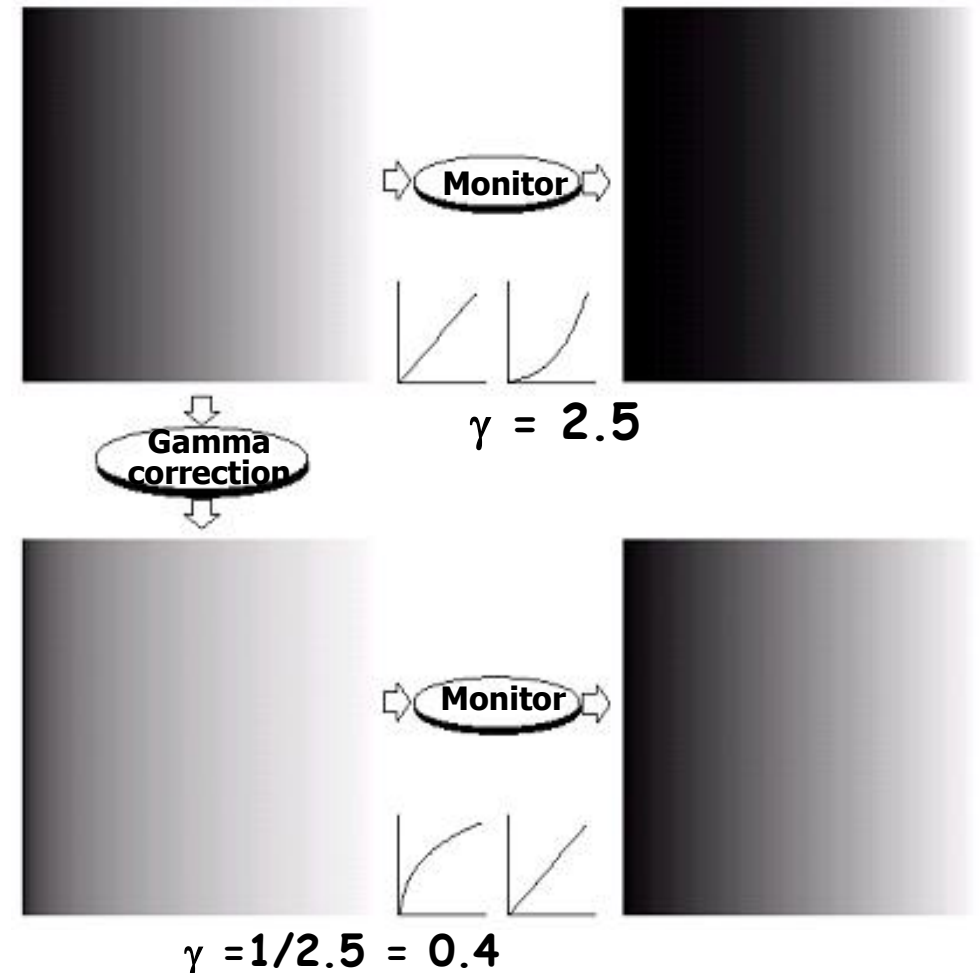
Input gray level, r
Plots of $s = cr^\gamma$ for various values of γ
($c = 1$ in all cases)



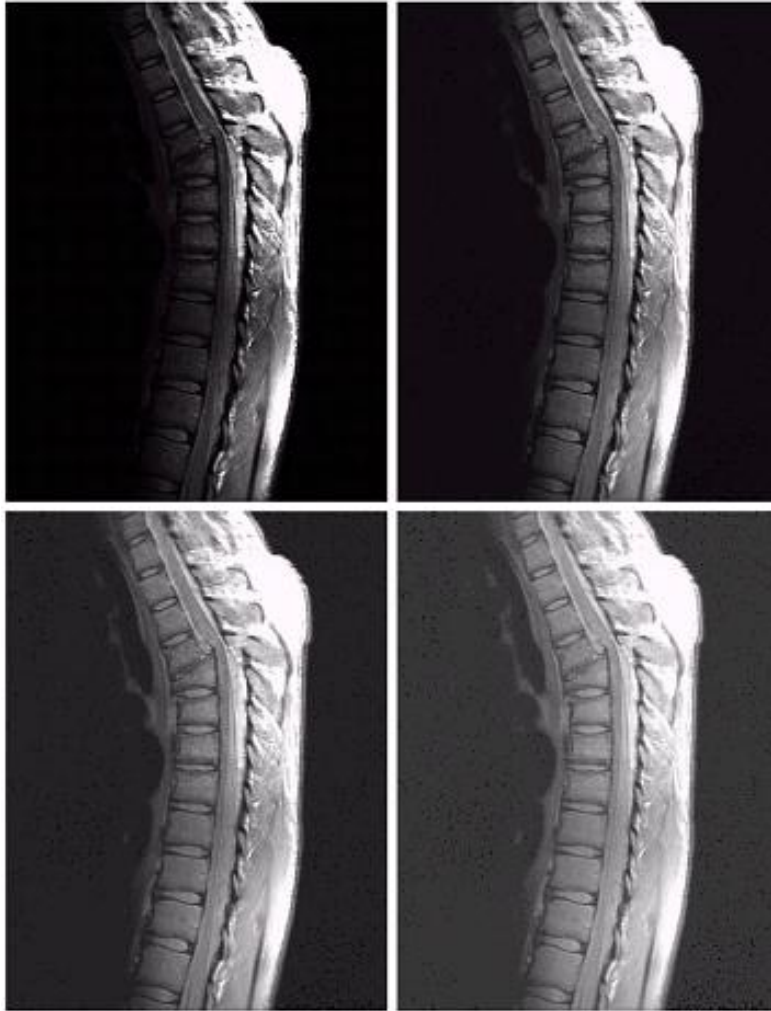
Gamma correction

- Cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with γ varying from 1.8 to 2.5
- The picture will become darker.
- Gamma correction is done by preprocessing the image before inputting it to the monitor with $s =$

$$cr^{1/\gamma}$$



Another example : MRI



(a) a magnetic resonance image of an upper thoracic human spine with a fracture dislocation and spinal cord impingement

- The picture is predominately dark
- An expansion of gray levels are desirable
⇒ needs $\gamma < 1$

(b) result after power-law transformation with $\gamma = 0.6$, $c=1$

(c) transformation with $\gamma = 0.4$
(best result)

(d) transformation with $\gamma = 0.3$
(under acceptable level)

a	b
c	d



Effect of decreasing gamma

- When the γ is reduced too much, the image begins to reduce contrast to the point where the image started to have very slight “wash-out” look, especially in the background



Another example



a	b
c	d

(a) image has a washed-out appearance,
it needs a compression of gray levels

⇒ needs $\gamma > 1$

(b) result after power-law

transformation with $\gamma = 3.0$ (suitable)

(c) transformation with $\gamma = 4.0$

(suitable)

(d) transformation with $\gamma = 5.0$

(high contrast, the image has areas
that are too dark, some detail is lost)

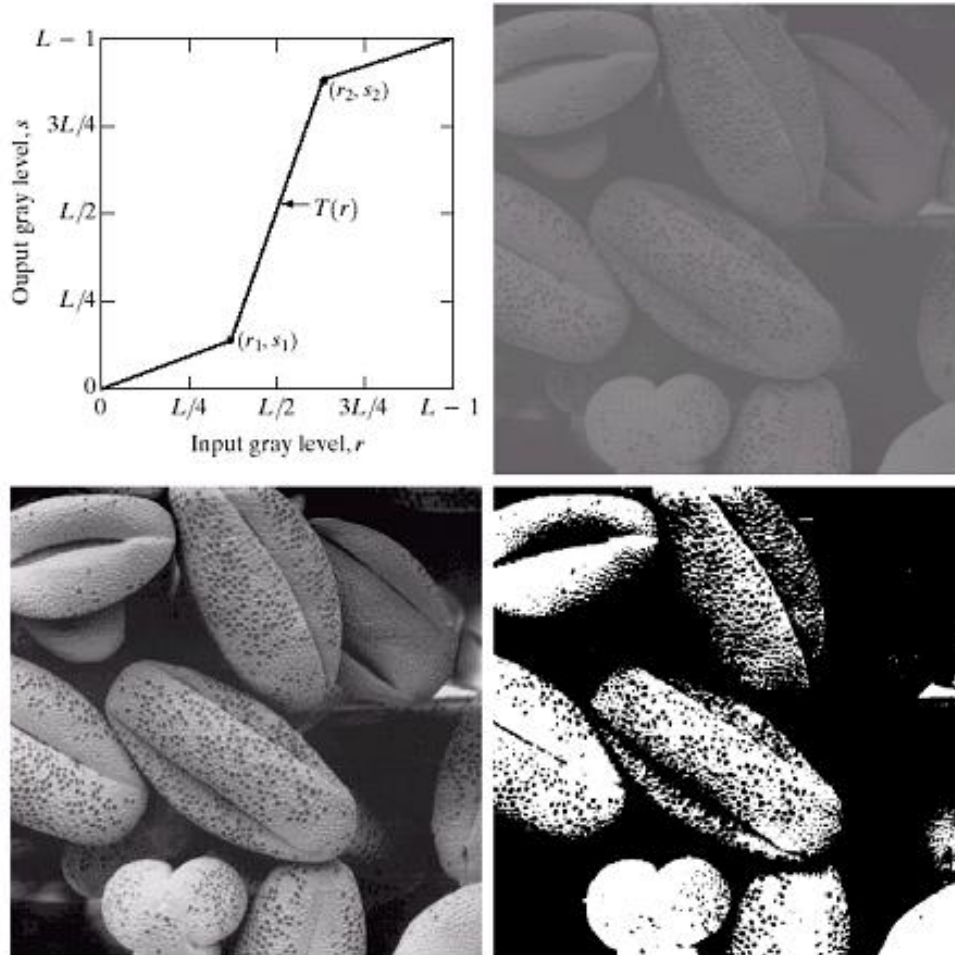


Piecewise-Linear Transformation Functions

- Advantage:
 - The form of piecewise functions can be arbitrarily complex
- Disadvantage:
 - Their specification requires considerably more user input



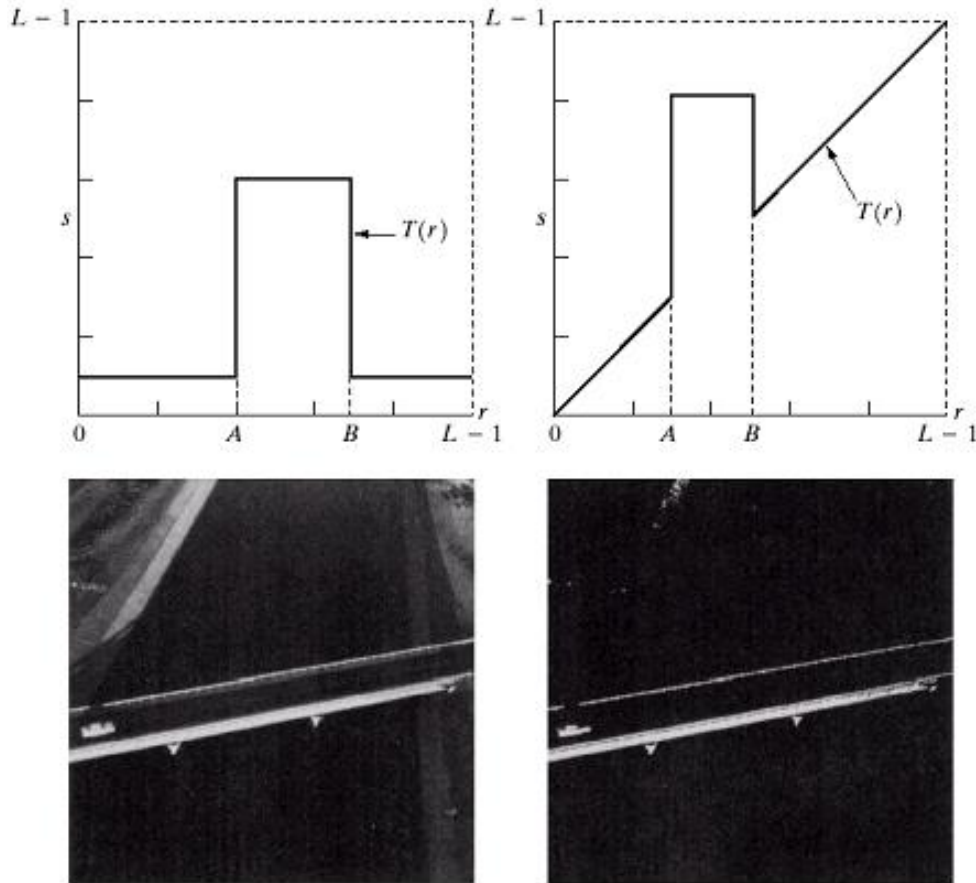
Contrast Stretching



- (a) increase the dynamic range of the gray levels in the image
- (b) a low-contrast image : result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture of image acquisition
- (c) result of contrast stretching: $(r_1, s_1) = (r_{\min}, 0)$ and $(r_2, s_2) = (r_{\max}, L-1)$
- (d) result of thresholding



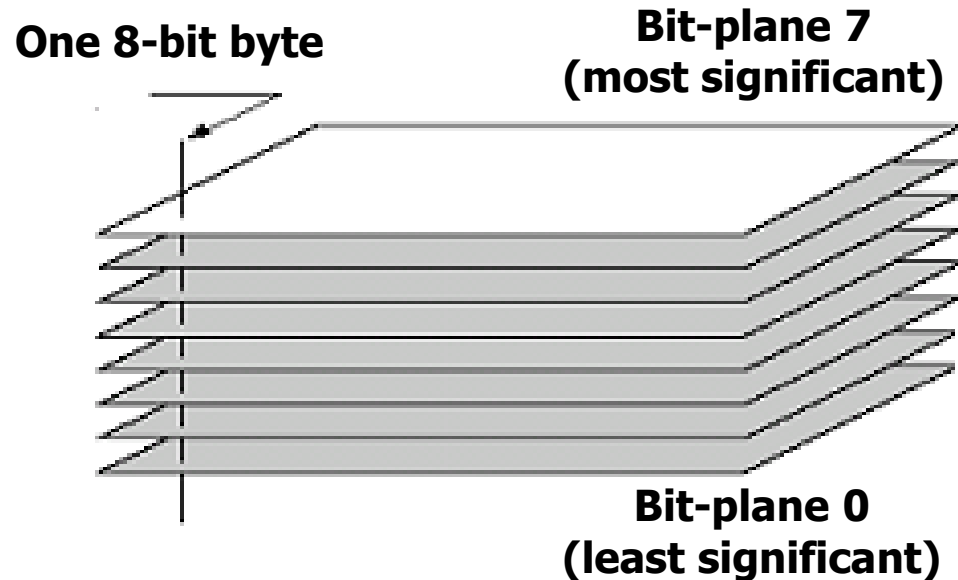
Gray-level slicing



- Highlighting a specific range of gray levels in an image
 - Display a high value of all gray levels in the range of interest and a low value for all other gray levels
- (a) transformation highlights range $[A,B]$ of gray level and reduces all others to a constant level
- (b) transformation highlights range $[A,B]$ but preserves all other levels



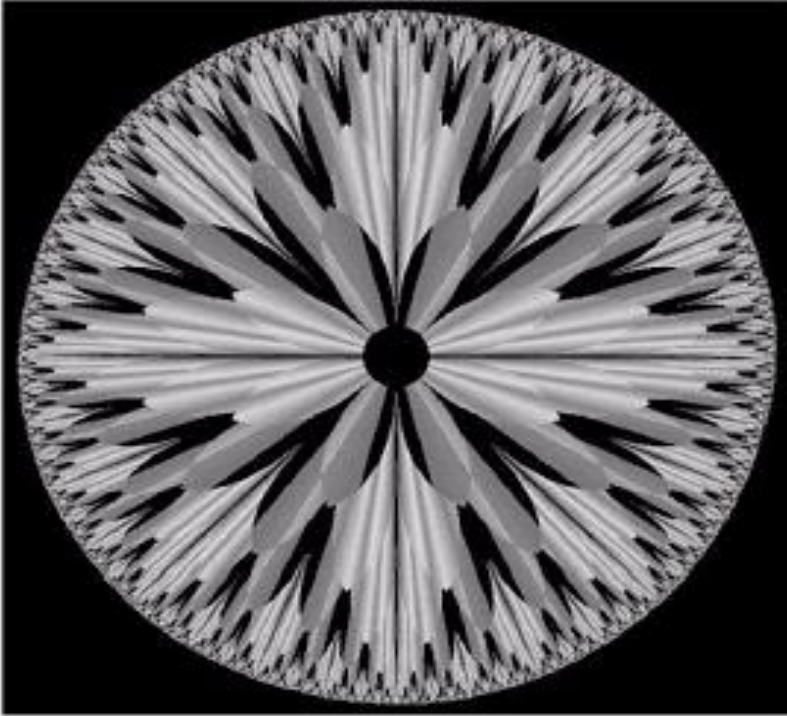
Bit-plane slicing



- Highlighting the contribution made to total image appearance by specific bits
- Suppose each pixel is represented by 8 bits
- Higher-order bits contain the majority of the visually significant data
- Useful for analyzing the relative importance played by each bit of the image



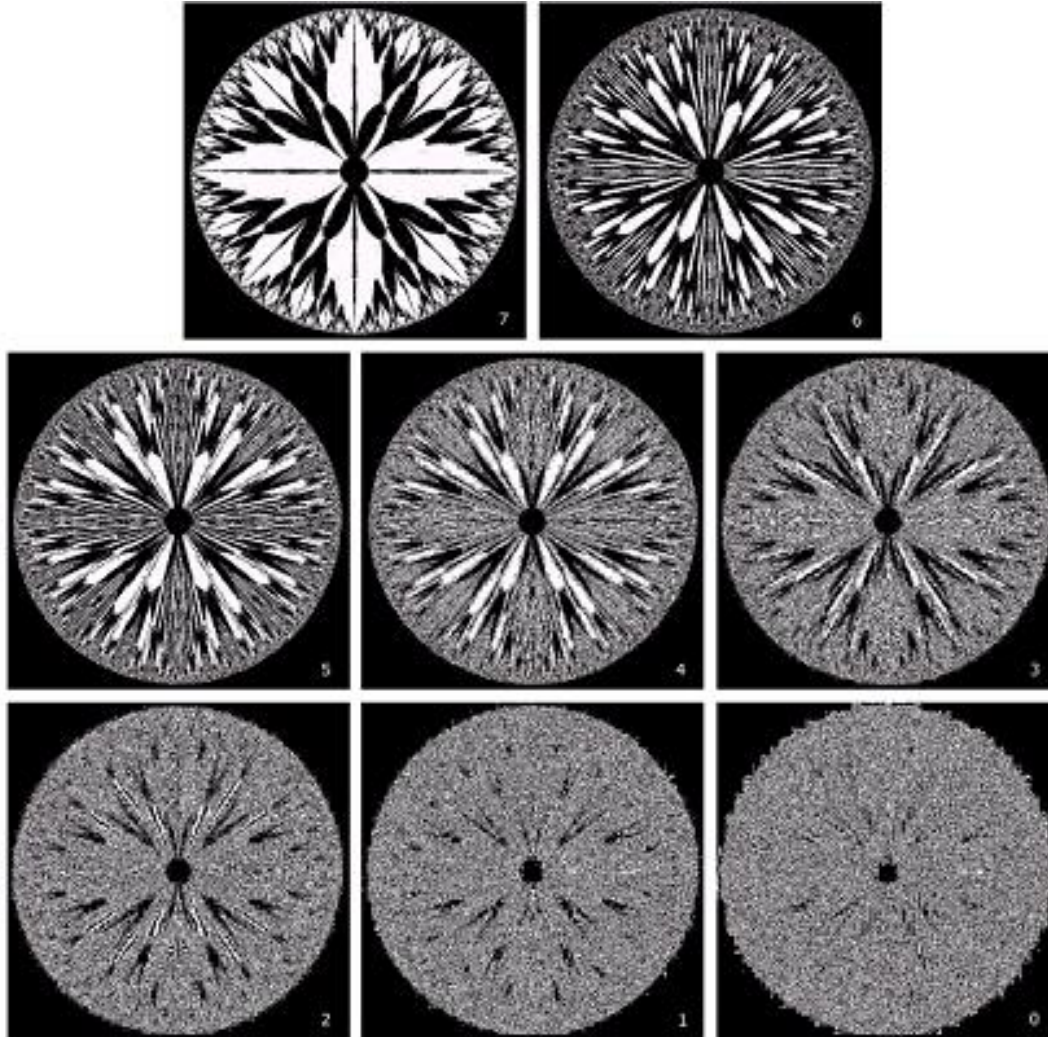
Example



An 8-bit fractal image



8 bit planes



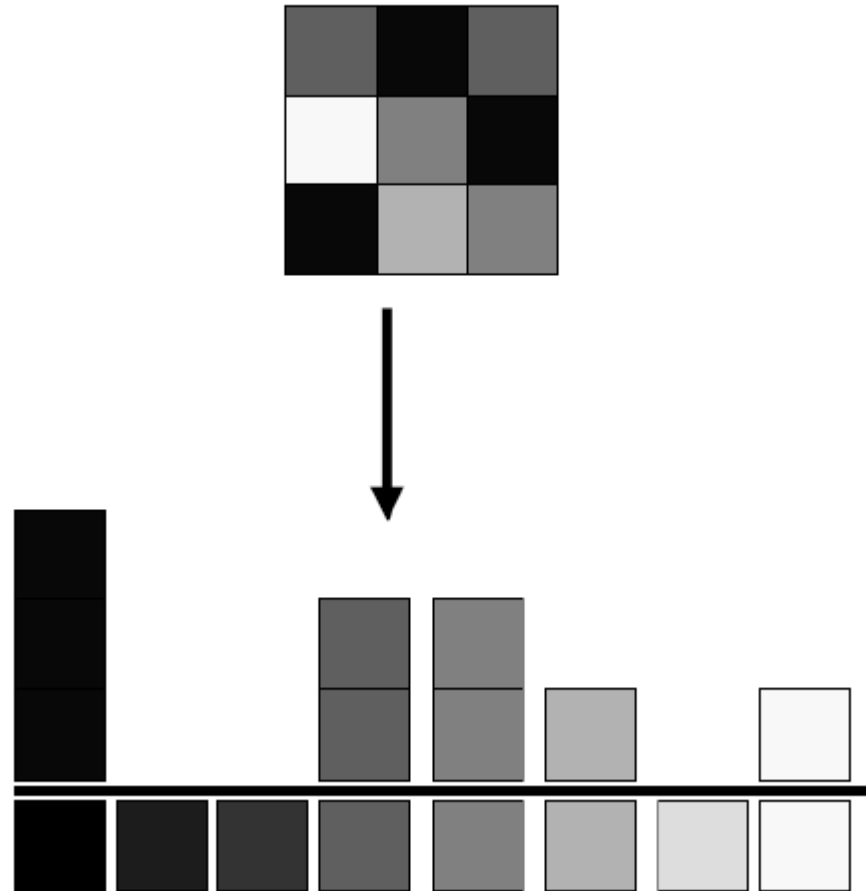
Bit-plane 7		Bit-plane 6	
Bit-plane 5	Bit-plane 4	Bit-plane 3	
Bit-plane 2	Bit-plane 1	Bit-plane 0	



Histogram Processing



Histogram



Histogram Processing

- Histogram of a digital image with gray levels in the range $[0, L-1]$ is a discrete function

$$h(r_k) = n_k$$

- Where
 - r_k : the k^{th} gray level
 - n_k : the number of pixels in the image having gray level r_k
 - $h(r_k)$: histogram of a digital image with gray levels r_k



Normalized Histogram

- dividing each of histogram at gray level r_k by the total number of pixels in the image, n

$$p(r_k) = n_k / n$$

- For $k = 0, 1, \dots, L-1$
- $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k
- The sum of all components of a normalized histogram is equal to 1

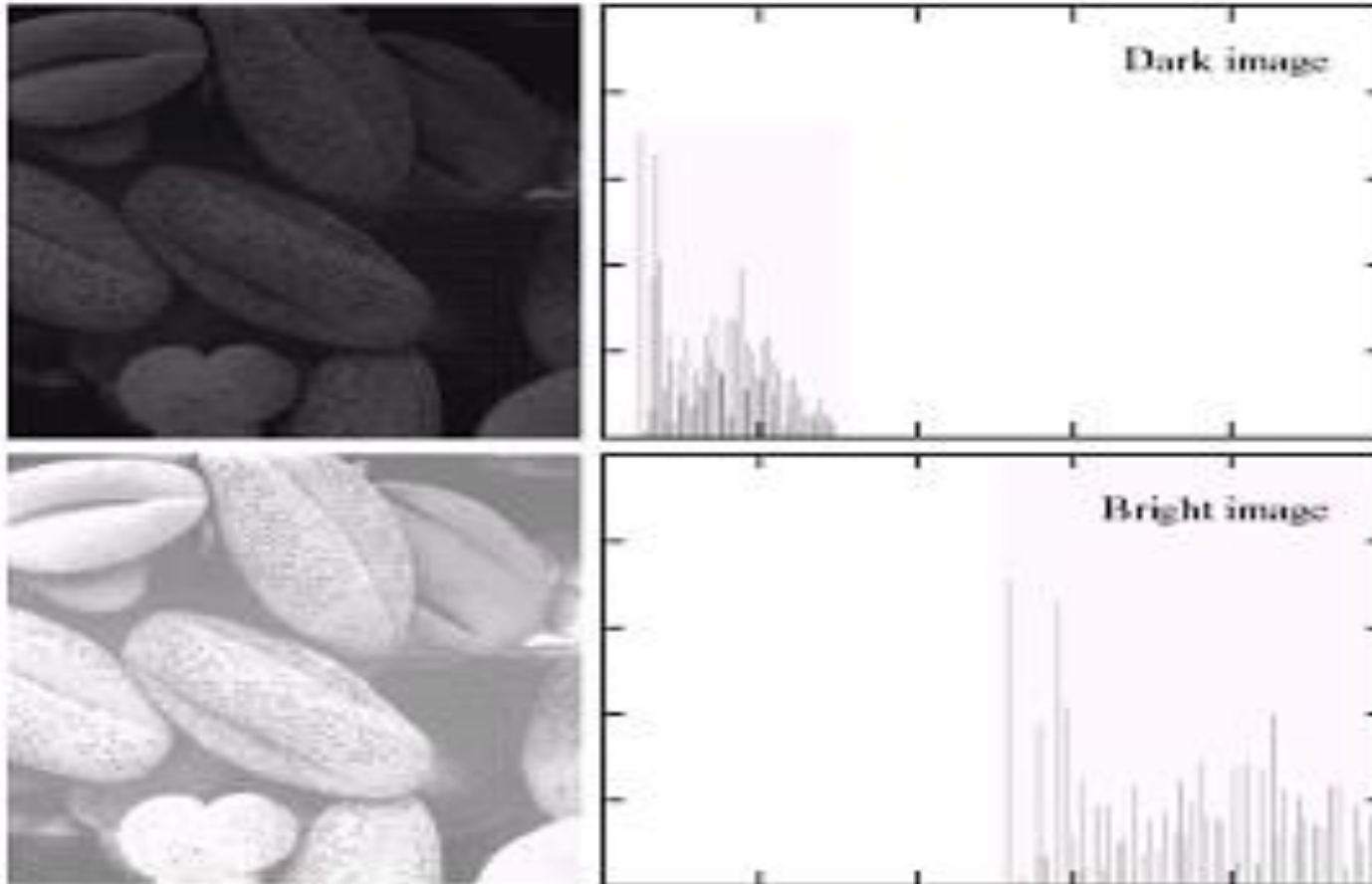
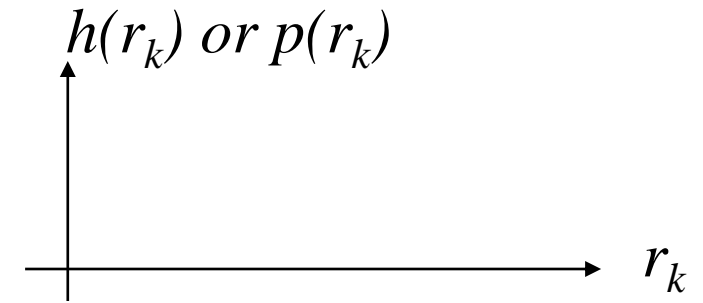


Histogram Processing

- Basic for numerous spatial domain processing techniques
- Used effectively for image enhancement
- Information inherent in histograms also is useful in image compression and segmentation



Example



Dark image

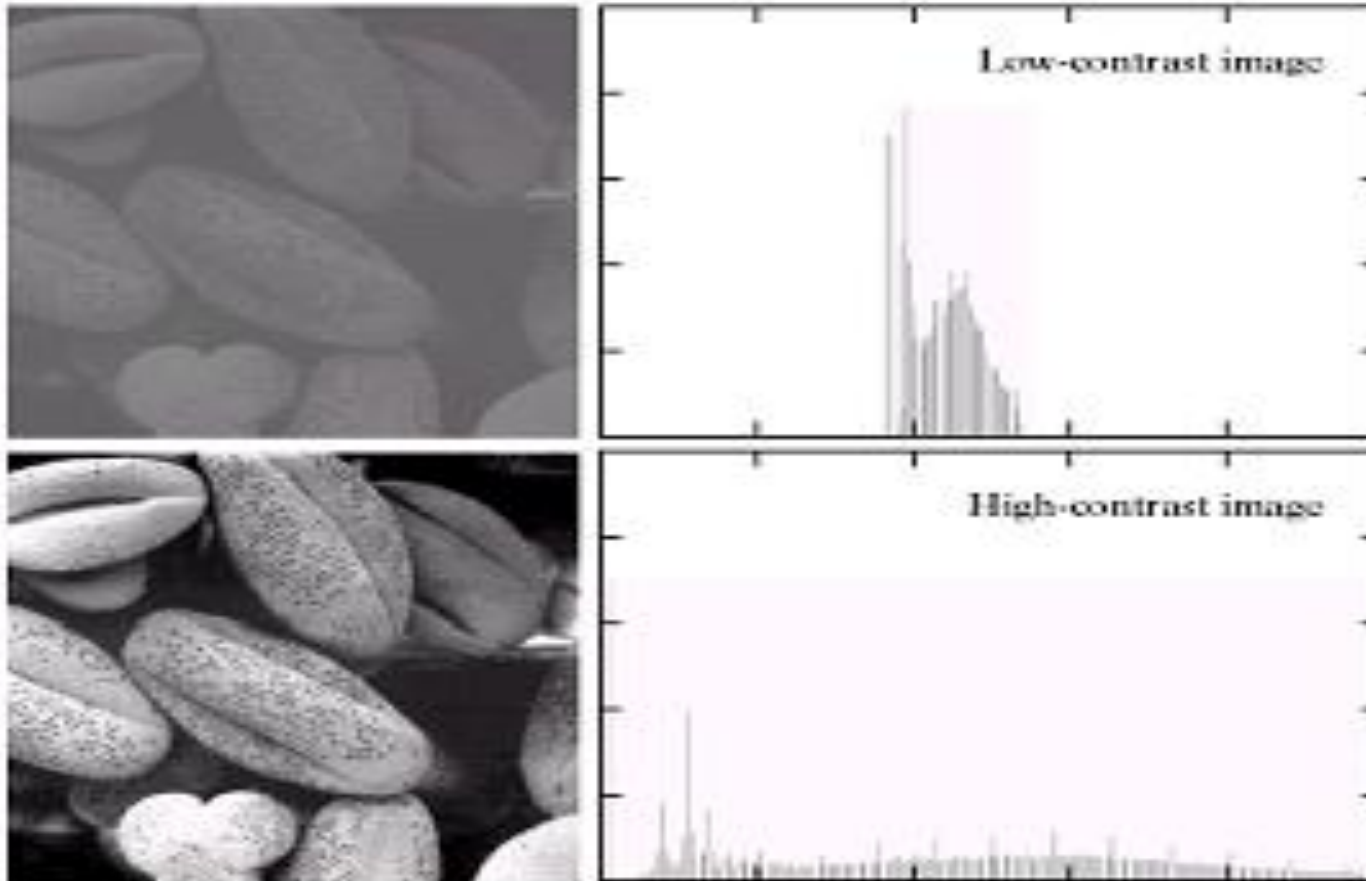
Components of histogram are concentrated on the low side of the gray scale.

Bright image

Components of histogram are concentrated on the high side of the gray scale.



Example



Low-contrast image

histogram is narrow and centered toward the middle of the gray scale

High-contrast image

histogram covers broad range of the gray scale and the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others

Histogram Equalization

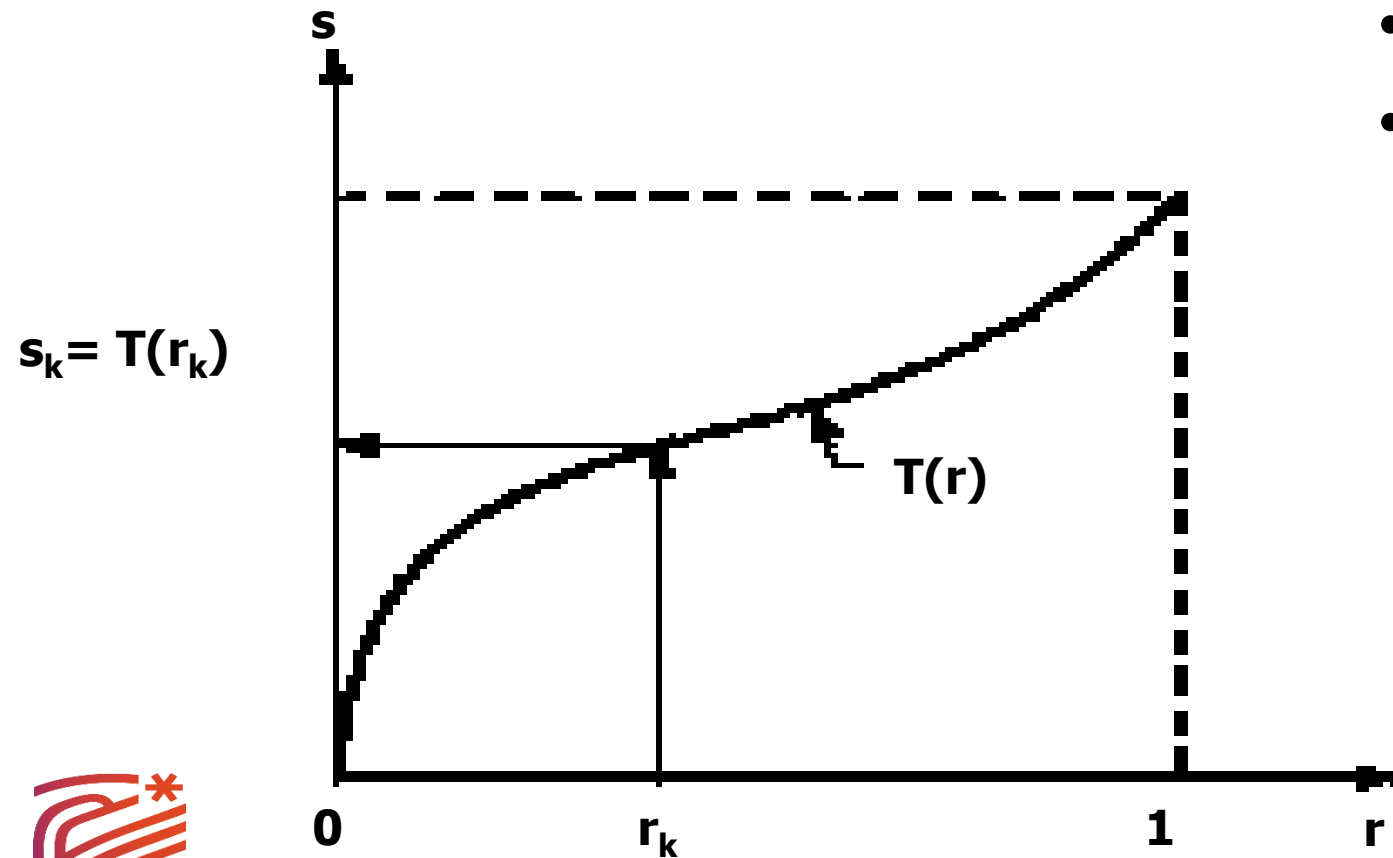
- As the low-contrast image's histogram is narrow and centered toward the middle of the gray scale, if we distribute the histogram to a wider range the quality of the image will be improved.
- We can do it by adjusting the probability density function of the original histogram of the image so that the probability spread equally



Histogram transformation

$$s = T(r)$$

- Where $0 \leq r \leq 1$
- $T(r)$ satisfies
 - (a). $T(r)$ is single-valued and monotonically increasing in the interval $0 \leq r \leq 1$
 - (b). $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$



2 Conditions of $T(r)$

- Single-valued (one-to-one relationship) guarantees that the inverse transformation will exist
- Monotonicity condition preserves the increasing order from black to white in the output image thus it won't cause a negative image
- $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$ guarantees that the output gray levels will be in the same range as the input levels.
- The inverse transformation from s back to r is

$$r = T^{-1}(s) \quad ; \quad 0 \leq s \leq 1$$



Probability Density Function

- The gray levels in an image may be viewed as random variables in the interval $[0,1]$
- PDF is one of the fundamental descriptors of a random variable



Histogram Equalization

Continues Values

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

Discrete Values

$$\begin{aligned} s_k = T(r_k) &= (L-1) \sum_{j=0}^k p_r(r_j) \\ &= (L-1) \sum_{j=0}^k \frac{n_j}{n} = \frac{L-1}{n} \sum_{j=0}^k n_j \quad k=0,1,\dots, L-1 \end{aligned}$$

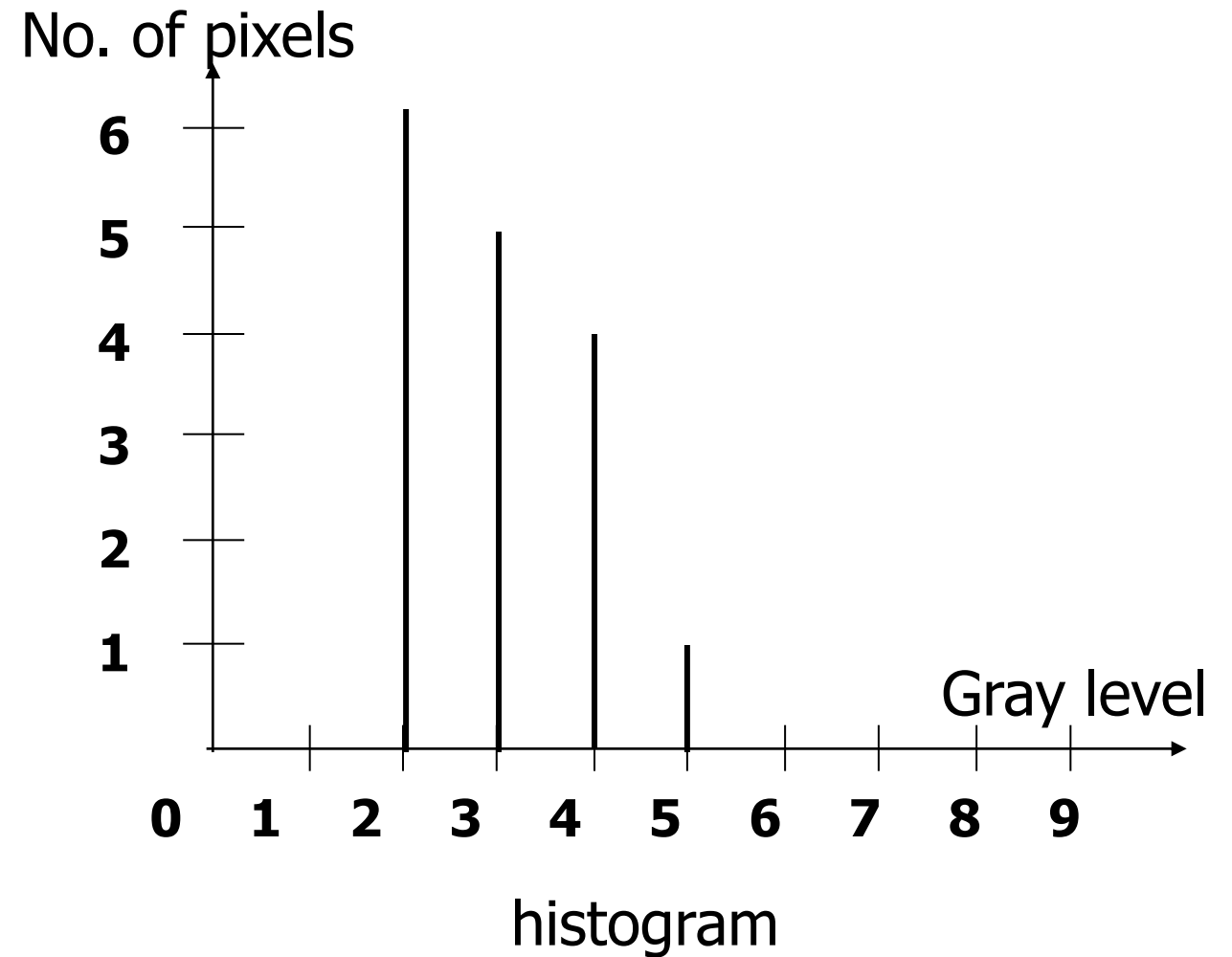


Example

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^k n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^k \frac{n_j}{n}$	0	0	$\frac{6}{16}$	$\frac{11}{16}$	$\frac{15}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$
$s \times 9$	0	0	$\frac{3.3}{\approx 3}$	$\frac{6.1}{\approx 6}$	$\frac{8.4}{\approx 8}$	9	9	9	9	9



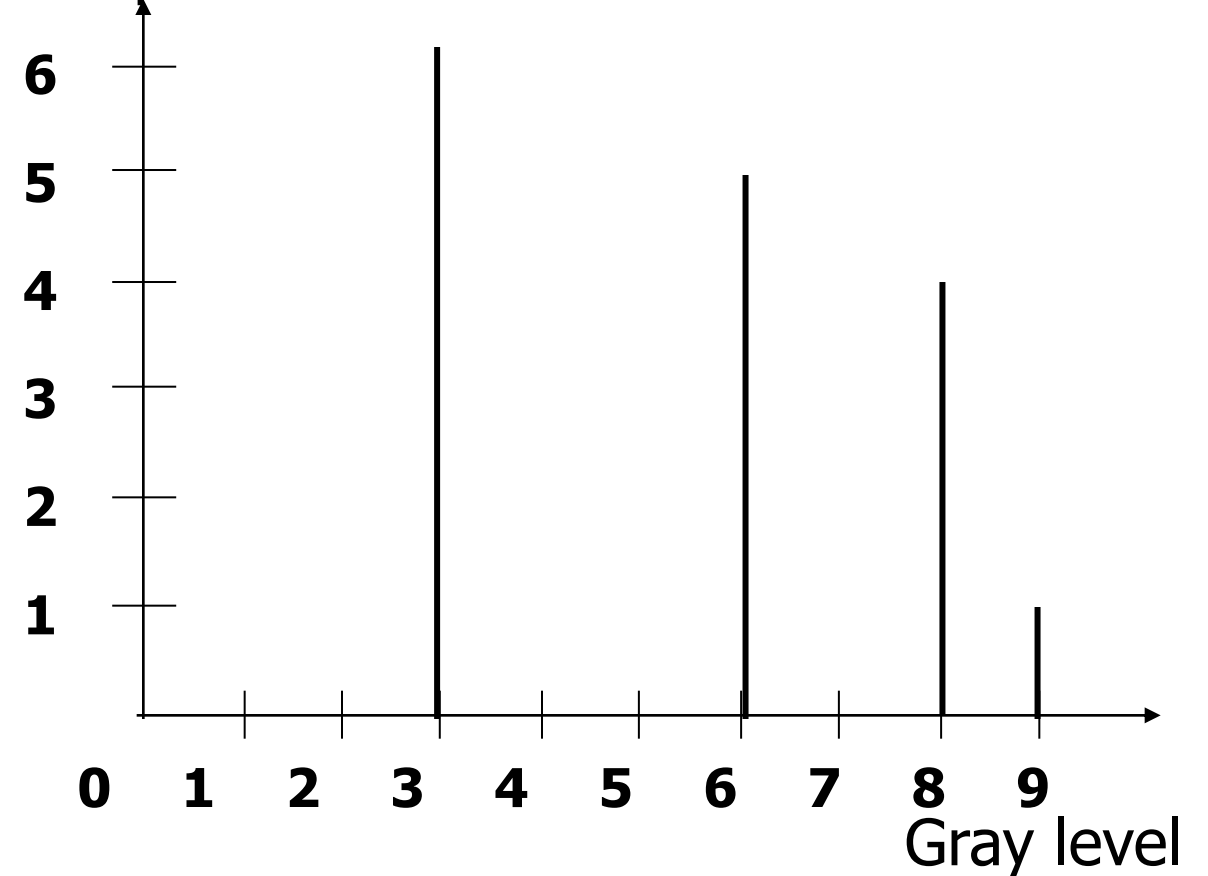
Example

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]

No. of pixels



Histogram equalization



Histogram Equalization

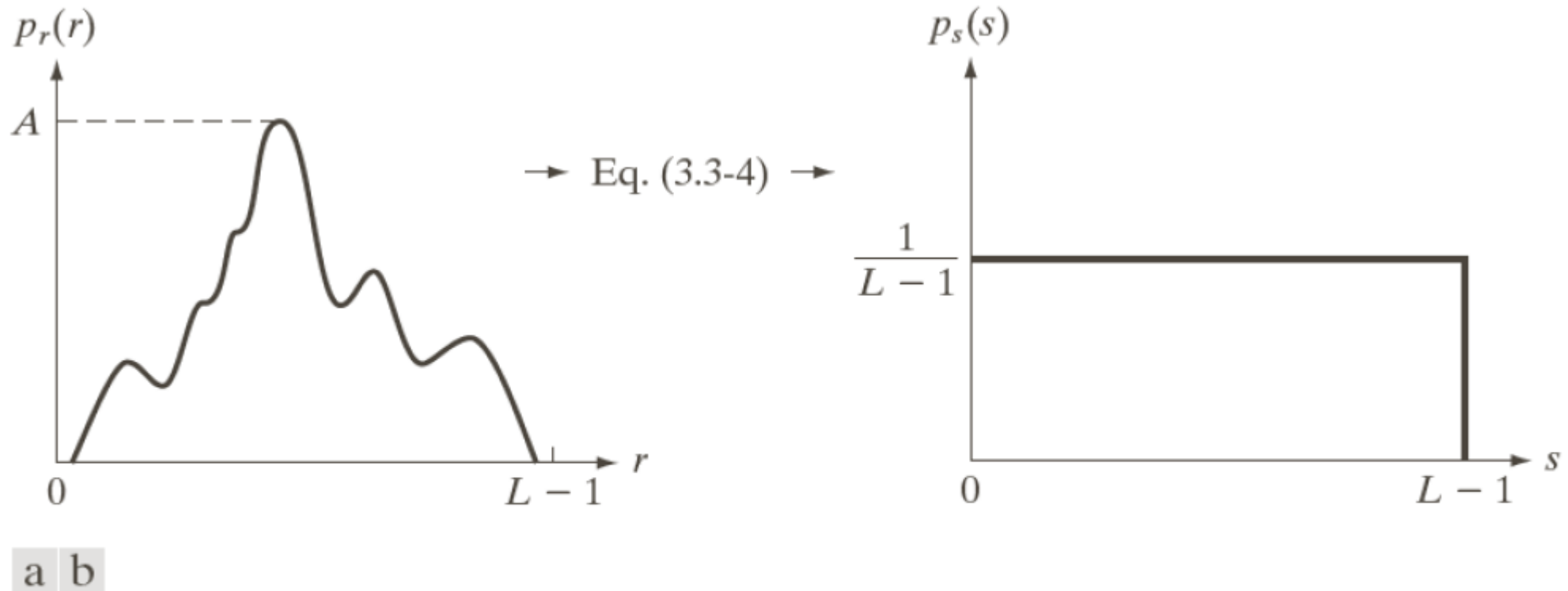
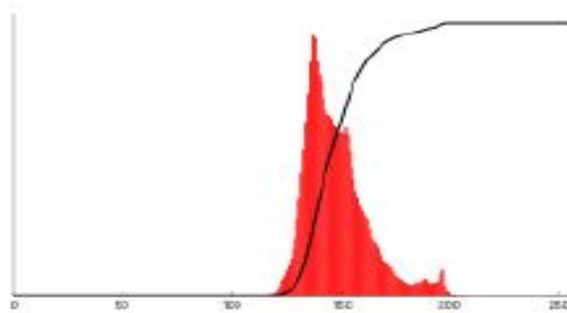


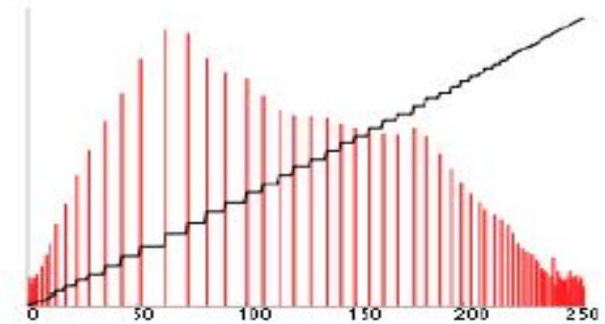
FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r . The resulting intensities, s , have a uniform PDF, independently of the form of the PDF of the r 's.



Histogram Equalization



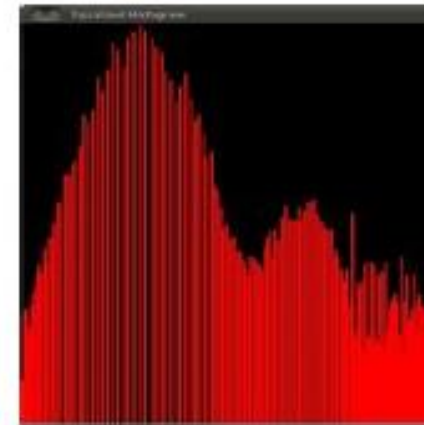
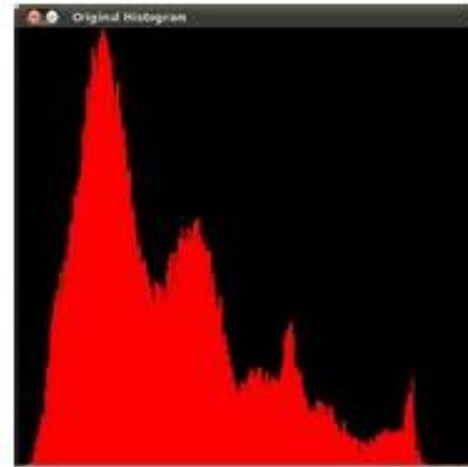
**Histogram &
Cumulative
Histogram**



Before & After Histogram Equalization



Source: https://en.wikipedia.org/wiki/Histogram_equalization



Before & After Histogram Equalization

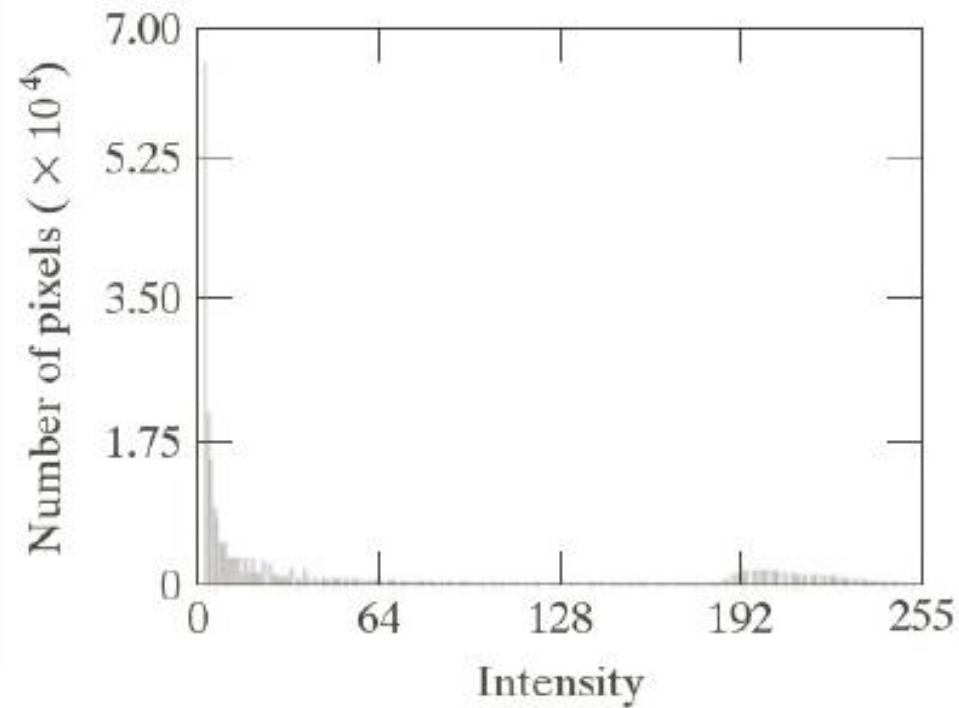
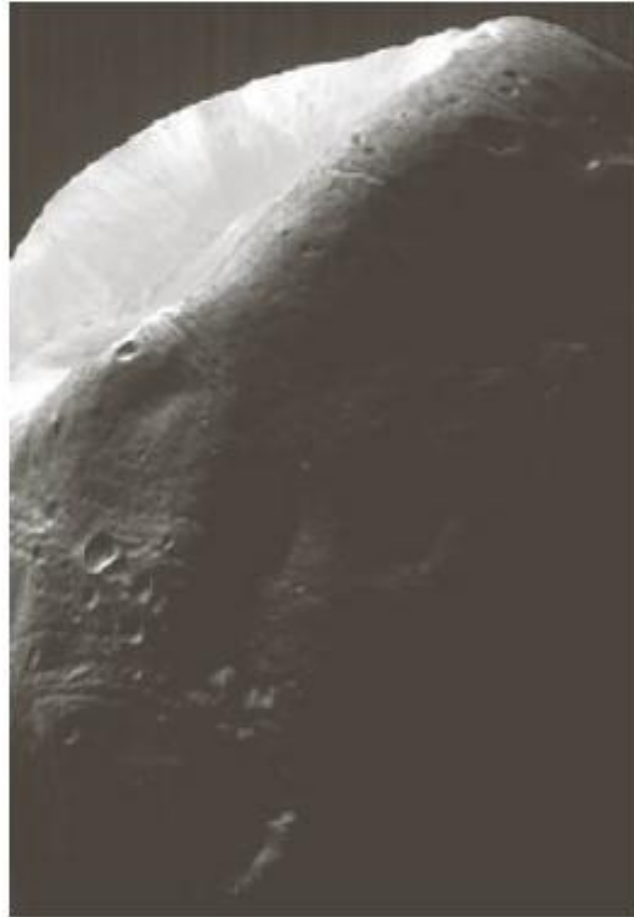
Source: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html



Note

- It is clearly seen that
 - Histogram equalization distributes the gray level to reach the maximum gray level (white) because the cumulative distribution function equals 1 when $0 \leq r \leq L-1$
 - If the cumulative numbers of gray levels are slightly different, they will be mapped to little different or same gray levels as we may have to approximate the processed gray level of the output image to integer number
 - Thus the discrete transformation function can't guarantee the one to one mapping relationship

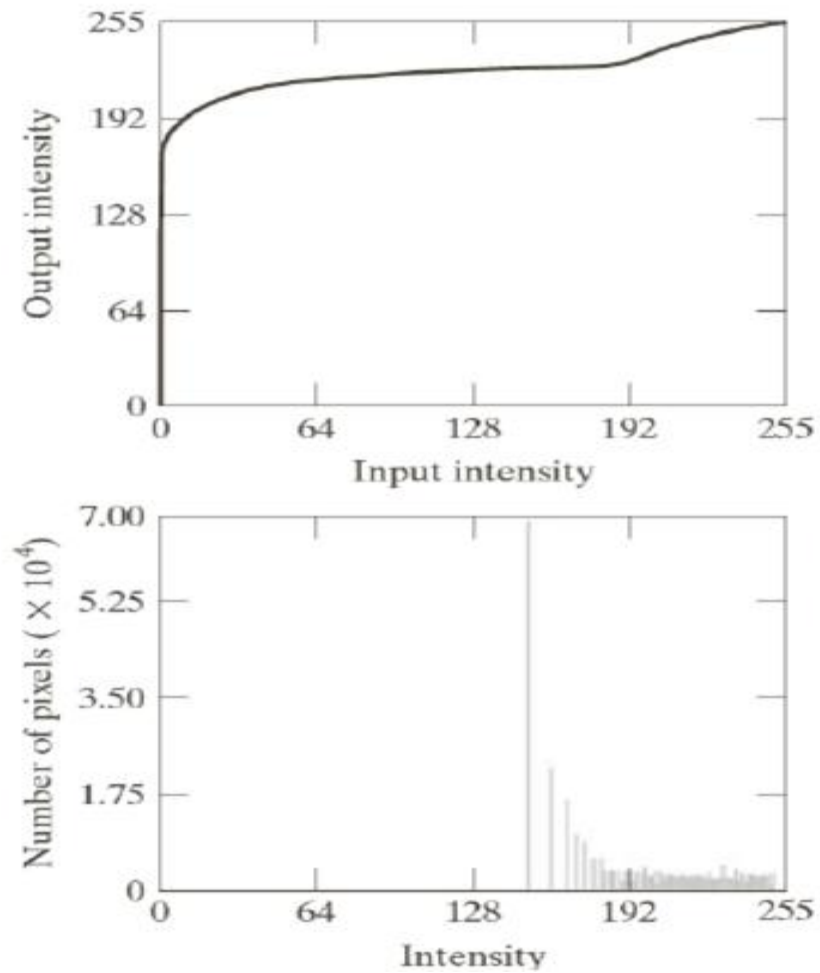




a b

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

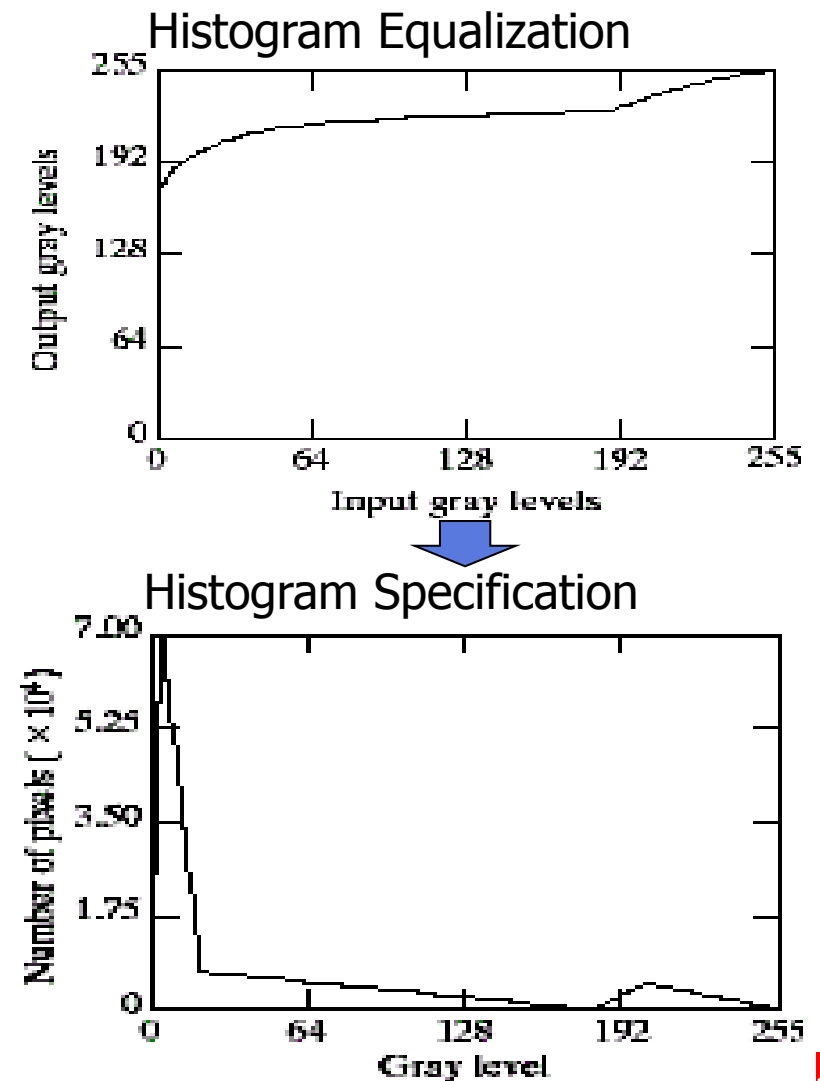
FIGURE 3.24
(a) Transformation function for histogram equalization.
(b) Histogram-equalized image (note the washed-out appearance).
(c) Histogram of (b).

- Histogram Equalization – Failure Example
 - If the histogram is heavily skewed, equalization may not produce good result
 - Then we need to find transformation to a ‘desired’ histogram



Solve the problem

- Since the problem with the transformation function of the histogram equalization was caused by a large concentration of pixels in the original image with levels near 0
- A reasonable approach is to modify the histogram of that image so that it does not have this property



Histogram Matching/ Specification



Histogram Matching/ Specification

- Goal: Specify the shape of the histogram:

$$p_r(r) \xrightarrow{?} p_z(z)$$

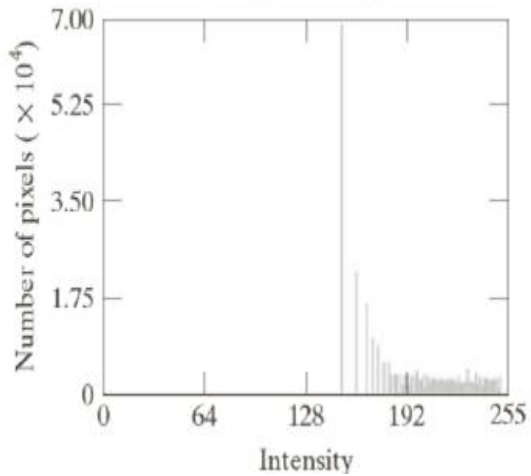
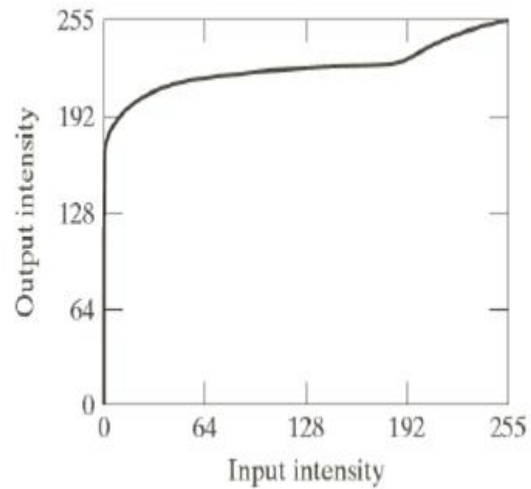
$$s = T(r) = (L-1) \int_0^r p_r(w) dw$$

$$G(z) = (L-1) \int_0^z p_z(t) dt = s$$

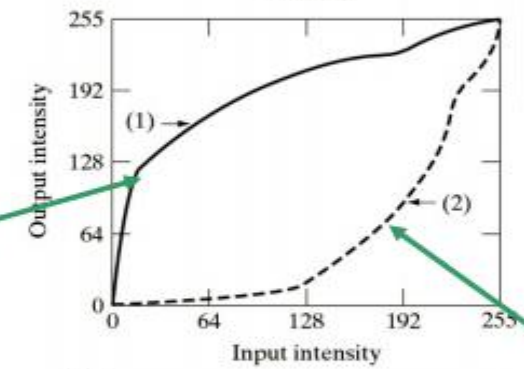
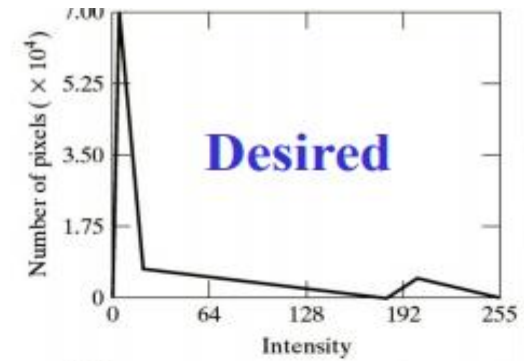
$$z = G^{-1}[T(r)] = G^{-1}(s)$$

where $p_r(r)$ and $p_z(z)$ are i/p & o/p PDFs

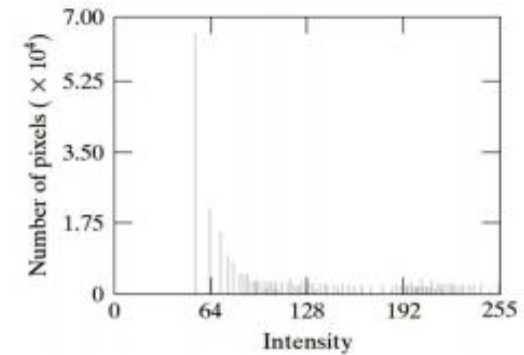




Transform using
Initial CDF



Transform using
Initial Modified CDF



- Histogram processing methods are global processing, in the sense that pixels are modified by a transformation function based on the gray-level content of an entire image.
- Sometimes, we may need to enhance details over small areas in an image, which is called a local enhancement.



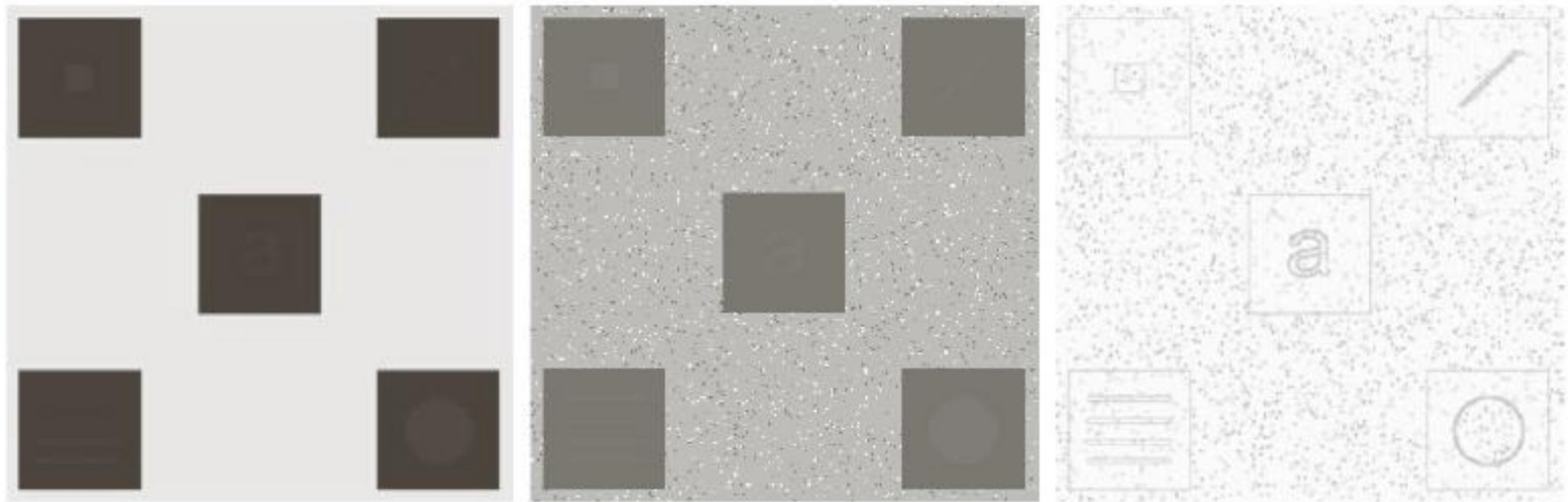
Local Histogram Processing



Local Processing Steps

- Define a Neighborhood
- Move its center from pixel to pixel
- Apply histogram equalization / matching @ center
- Non-overlapping computation is fast but blocky





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 .



Histogram Statistics For Image Enhancement

Use of Global Statistical Measures

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i) \approx \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x, y) - m]^n$$

$$m = \sum_{i=0}^{L-1} r_i p(r_i) \approx \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N f(x, y)$$

Gross adjustments in overall intensity (m) and contrast (μ_2)



Histogram Statistics For Image Enhancement

Local mean and local variance:

$$m_{S_{xy}}(x, y) = \sum_{i=0}^{L-1} r_i p_{S_{xy}}(r_i) \approx \frac{1}{|S_{xy}|} \sum_{(s,t) \in S_{xy}} f(s, t)$$

$$\sigma_{S_{xy}}^2(x, y) = \sum_{i=0}^{L-1} (r_i - m_{S_{xy}}(x, y))^2 p_{S_{xy}}(r_i) \approx \frac{1}{|S_{xy}|} \sum_{(s,t) \in S_{xy}} [f(s, t) - m_{S_{xy}}(x, y)]^2$$

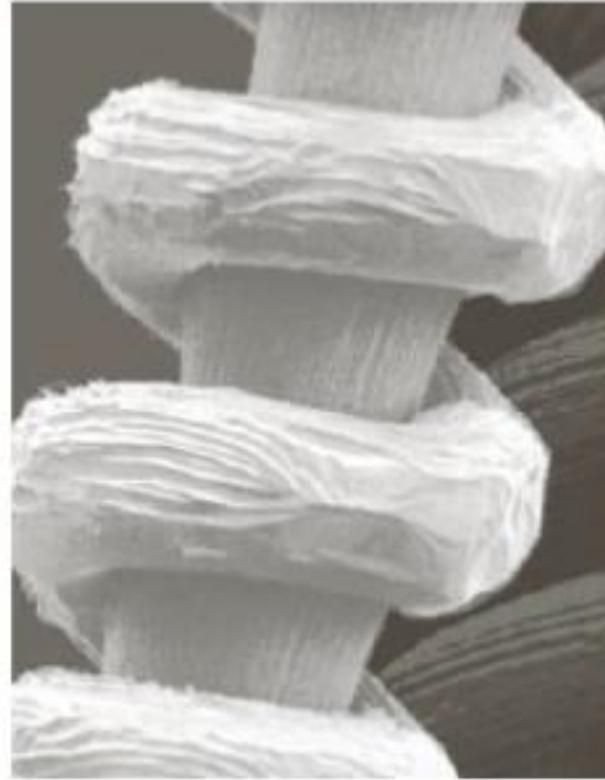
S_{xy} : Neighborhood centered on (x, y)

Local information intensity and contrast
(edges)





Original



Local Histogram



Local Statistics

Enhancement using Arithmetic/Logic Operations



Enhancement using Arithmetic/Logic Operations

- Arithmetic/Logic operations perform on pixel by pixel basis between two or more images
- except NOT operation which perform only on a single image

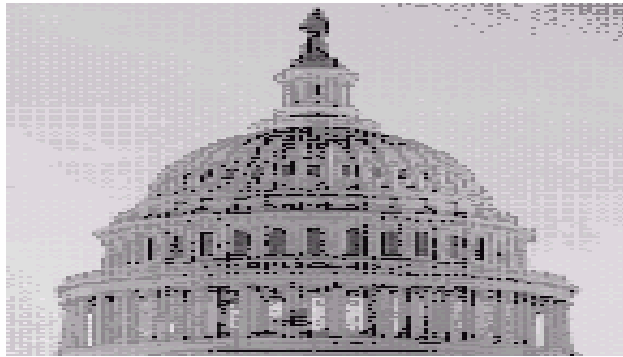


Logic Operations

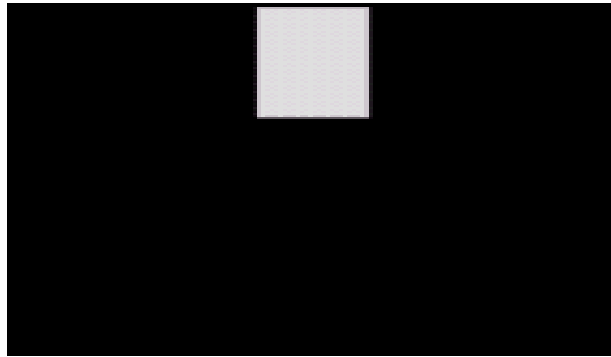
- Logic operation performs on gray-level images, the pixel values are processed as binary numbers
- light represents a binary 1, and dark represents a binary 0
- NOT operation = negative transformation



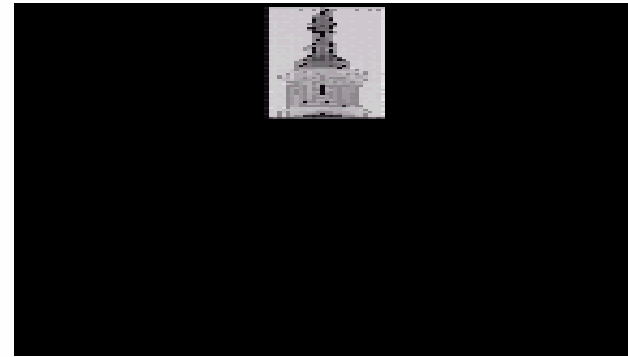
Example of AND Operation



original image



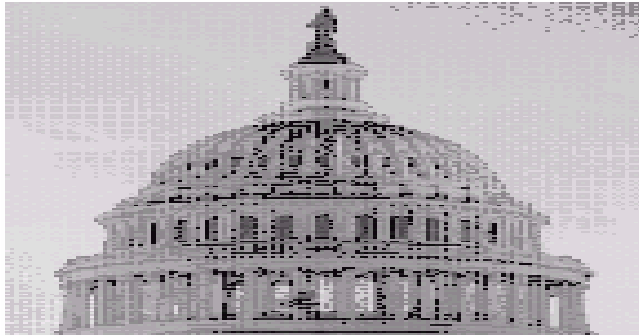
AND image
mask



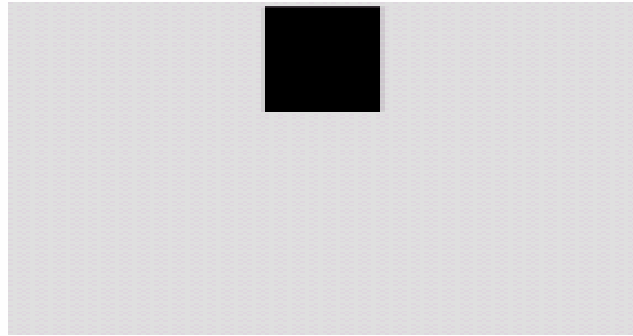
result of AND
operation



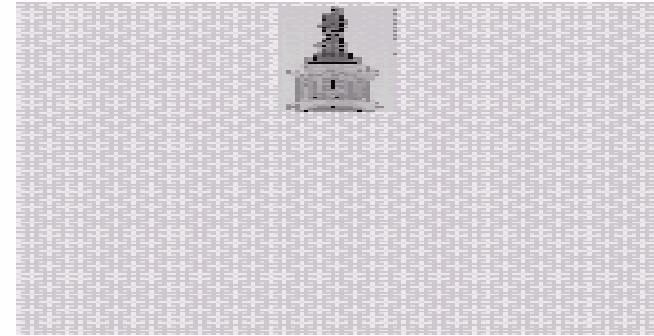
Example of OR Operation



original image



OR image
mask



result of OR
operation



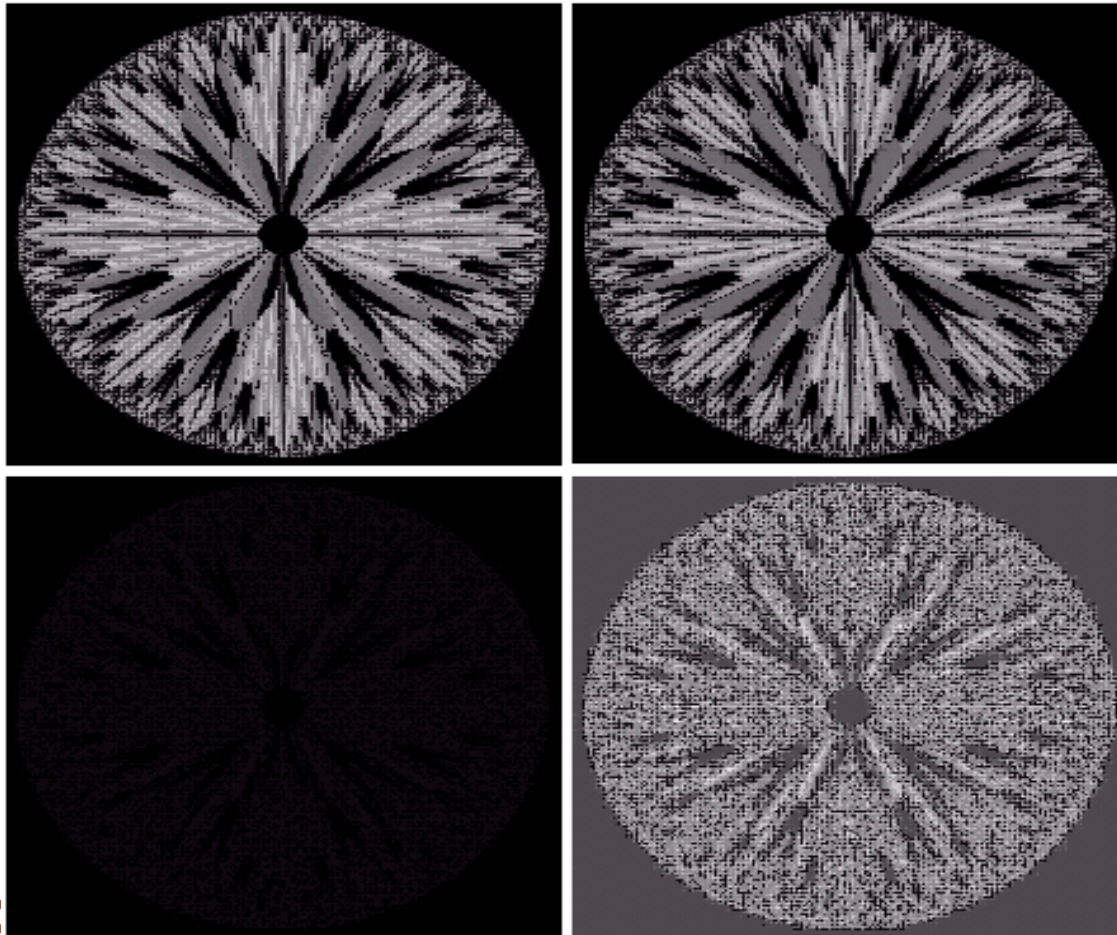
Image Subtraction

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



Image Subtraction

a	b
c	d



a). original fractal image

b). result of setting the four lower-order bit planes to zero

- refer to the bit-plane slicing
- the higher planes contribute significant detail
- the lower planes contribute more to fine detail
- image b). is nearly identical visually to image a), with a very slightly drop in overall contrast due to less variability of the gray-level values in the image.

c). difference between a). and b). (nearly black)

d). histogram equalization of c). (perform contrast stretching transformation)



Summary

- Gray Level Transformation techniques.
- Histogram Processing.
- Arithmetic and Logical operations



*Thank
you!*

