Mahidol University *Wisdom of the Land*

Chapter 3

Image Enhancement in the Spatial Domain

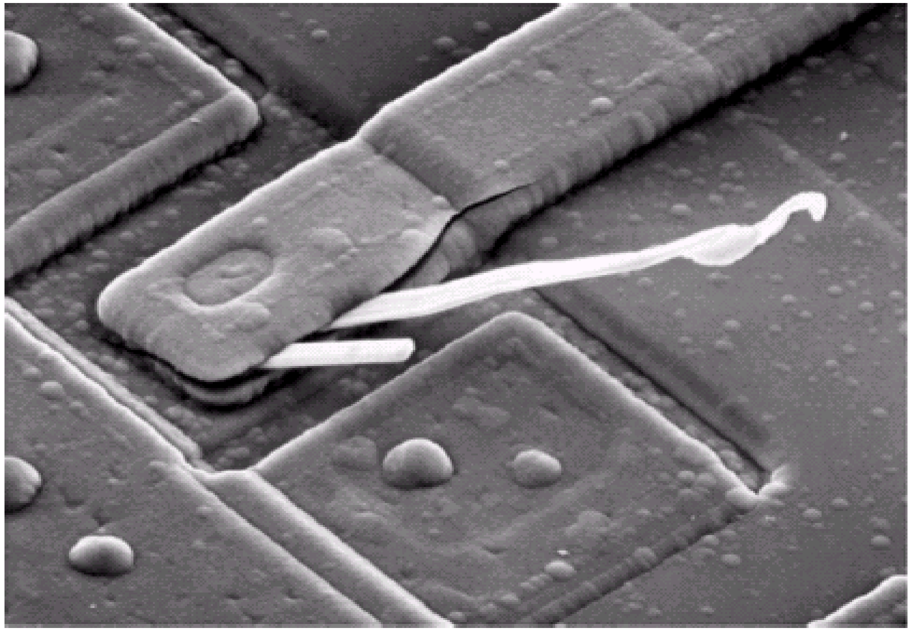
What is Image Enhancement?

- Image enhancement is the process that improves the quality of the image for a specific application.

- There are two main categories of image enhancement methods:
  - Spatial domain methods : The term spatial domain refers to the two-dimensional image plane, and the methods manipulate the pixels of a given image.
  - Frequency domain methods : The methods in frequency domain manipulate the Fourier transform of a given image.

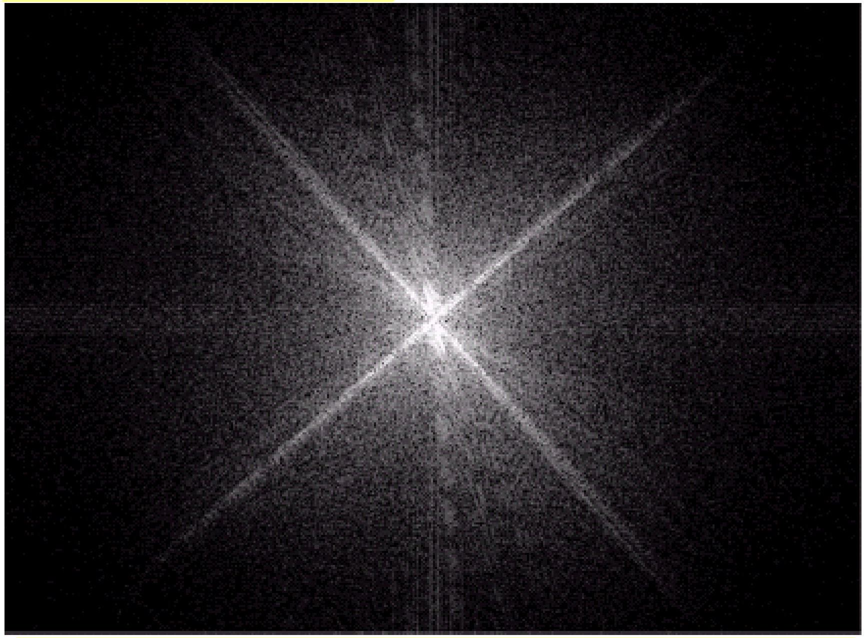
Image Enhancement Methods

Spatial domain



A given image

Fourier domain



Fourier transform of a given image

Image Enhancement Methods

- Spatial domain : Pixel - base (2-D image plane)

Input Image data f(x,y) → Transfer Function T[f(x,y)] → Output Image data g(x,y)

- Frequency domain : Frequency - base

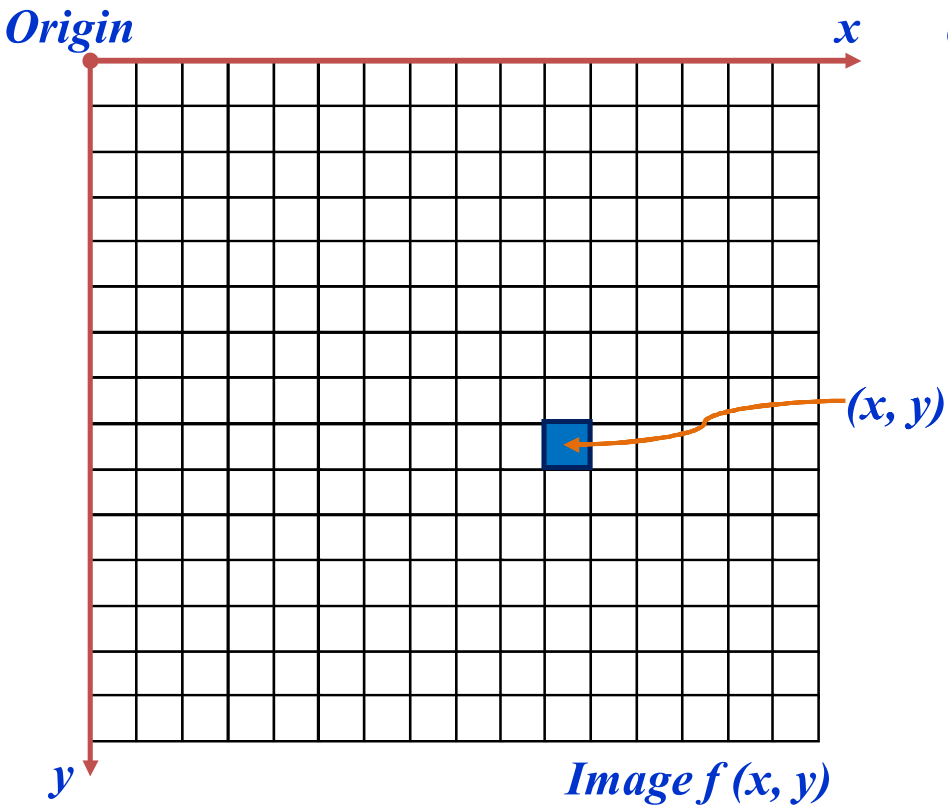
Input Image data f(x,y) → Fourier Transform F(u,v) → Filter Function F(u,v) H(u,v) → Inverse Fourier Transform → Output Image data g(x,y)

Image Enhancement in Spatial Domain

- In spatial domain, there are two types of image enhancement methods :
  - Gray-level transformations (Point processing) refer to the processing methods where the enhancement at any point in an image depends only the gray-level at that point.
  - Spatial filtering methods (Neighborhood processing) are the two-dimensional array defined around a pixel, where the values of the mask (also referred to as window/kernel/template) coefficients determine the nature of the process.

Image Enhancement in Spatial Domain

Gray-level transformation



Spatial filtering

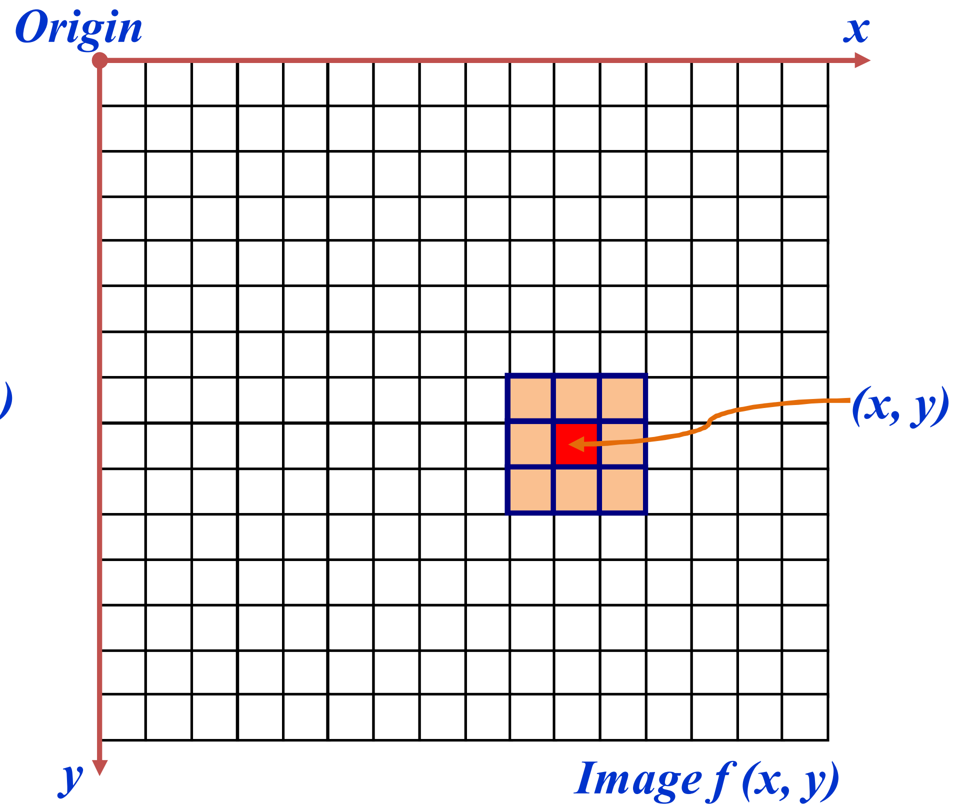
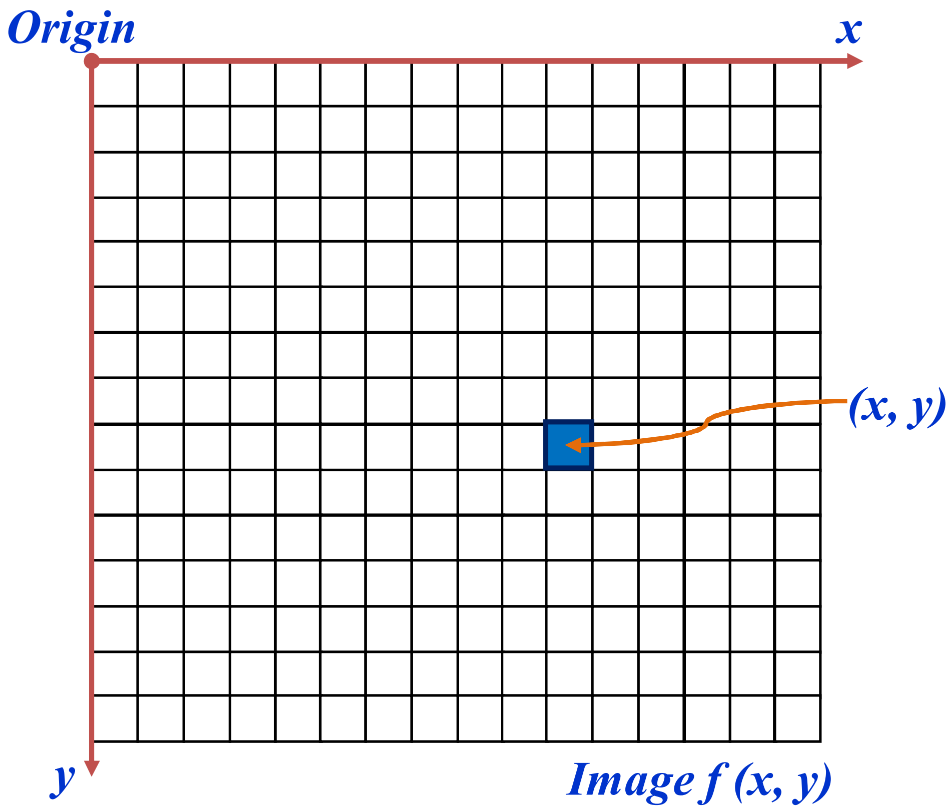


Image Enhancement in Spatial Domain

- Most spatial domain enhancement operations can be reduced to the form :

$$\boxed{g(x,y) = T\left[f(x,y)\right]}$$



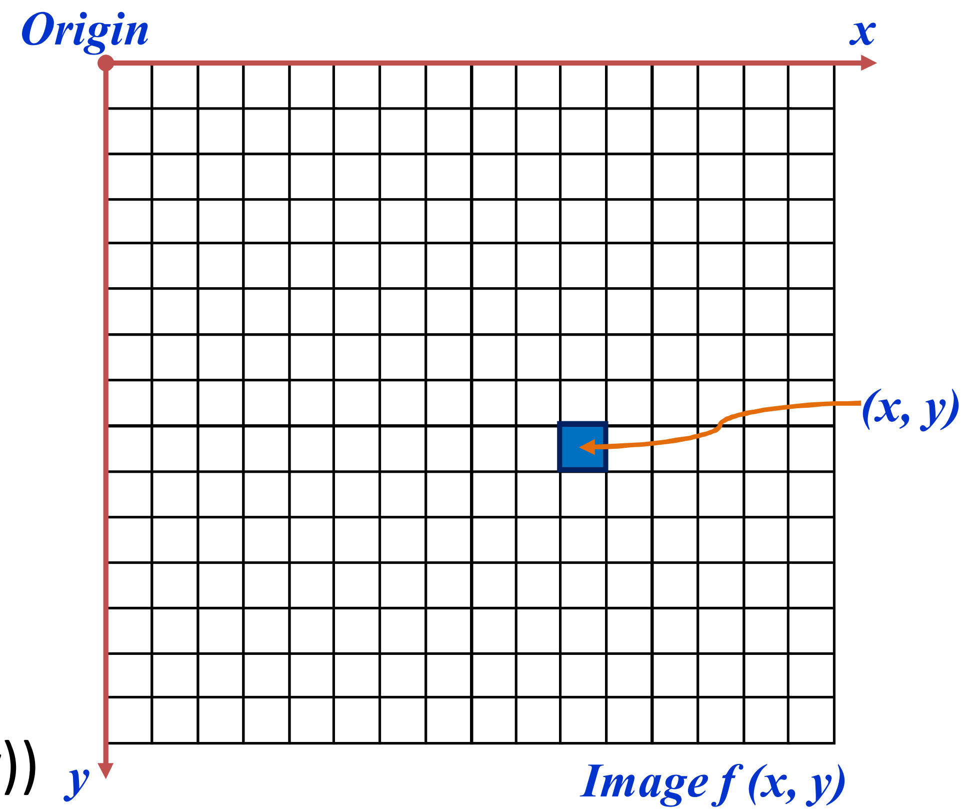
where

- $g(x, y)$ is the output image
- $f(x, y)$ is the input image
- T is the operator (Transformation function) of f, defined over a neighborhood of (x, y)

Gray-Level Transformations

- If a 1x1 neighborhood is used for the transformation function.
- Then the function is called the gray-level transformation function.

$$\boxed{s = T(r)}$$



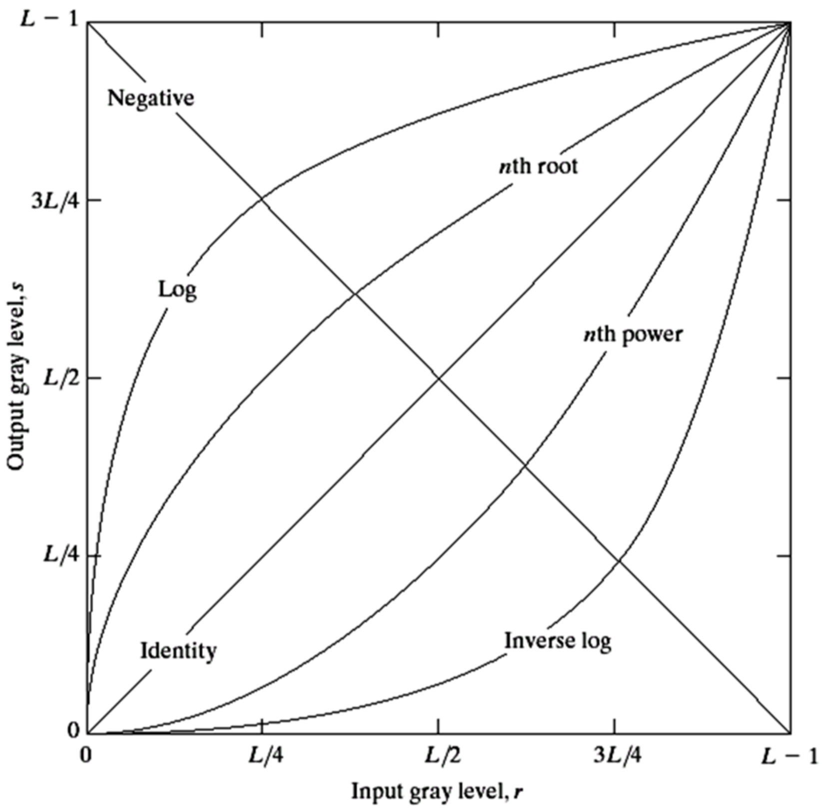
where

- s is the output image ($s = g(x, y)$)
- r is the input image ($r = f(x, y)$)

Basic Gray Level Transformations

There are many different kinds of Gray Level transformations. The most common are shown here :

- Negative Transformations
- Log Transformations
- Power-Law Transformations
- Piecewise-Linear Transformation Functions



Negative Transformations

- The negative of an image with gray levels in the range [0,L-1] can be obtained by

$$\boxed{s = L - 1 - r}$$

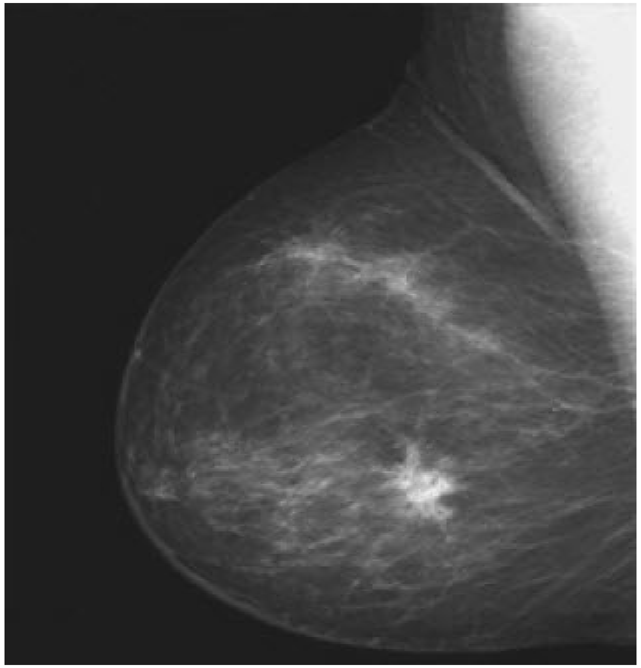
where

- s is the processed image
- r is the input image

- Negative Transformations or Image Negatives: be suited for enhancing white or grey detail embedded in dark regions of an image, especially when the black areas are dominant in size.

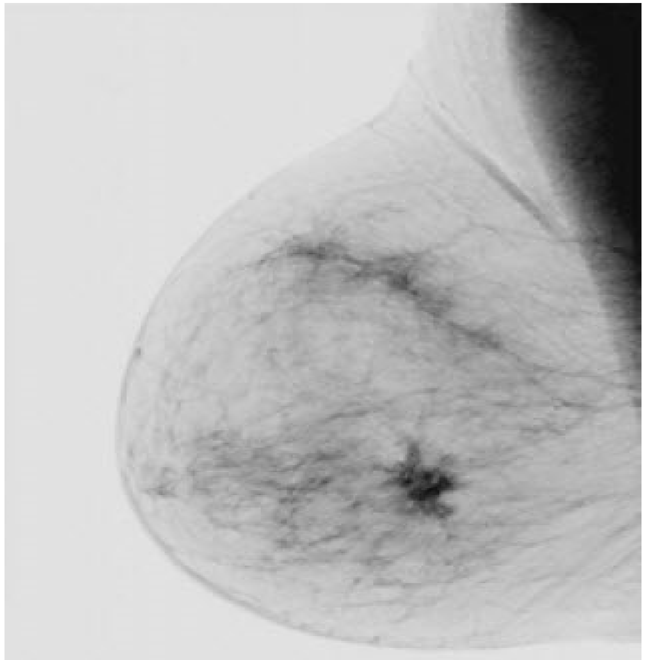
Example : Negative transformation

- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image.
- Note how much clearer the tissue is in the negative image of the mammogram below.



Original mammogram showing a small lesion of a breast

$$s = intensity_{max} - r$$



Negative image gives a better vision to analyze the image.

Log Transformations

- The general form of the log transformations :

$$\boxed{s = c\log(r+1)}$$

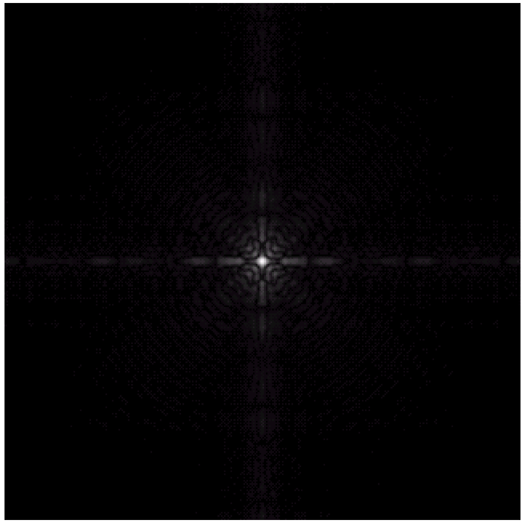
where

- c is a constant
- r positive constants ($r \geq 0$)

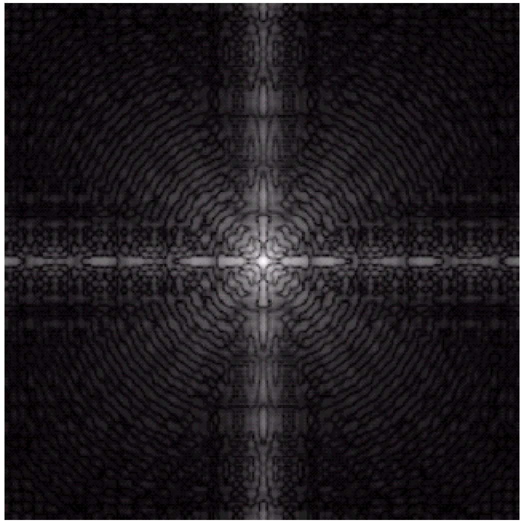
- The Log Transforms compresses the dynamic range of images with large variations in pixel values.
- Typical application is to change the large dynamic range of a Fourier Spectrum to a smaller range for a clearer inspection.

Example : Log Transformations

- Log functions are particularly useful when the input grey level values may have an extremely large range of values.
- In the following example the Fourier transform of an image is put through a log transform to reveal more detail.



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



a) Fourier Spectrum with range = 0 to 1.5 x 10^6

b) Result after apply the log transformation with c = 1, range = 0 to 6.2

Power-Law Transformations

- The general form of the Power-law transformation :

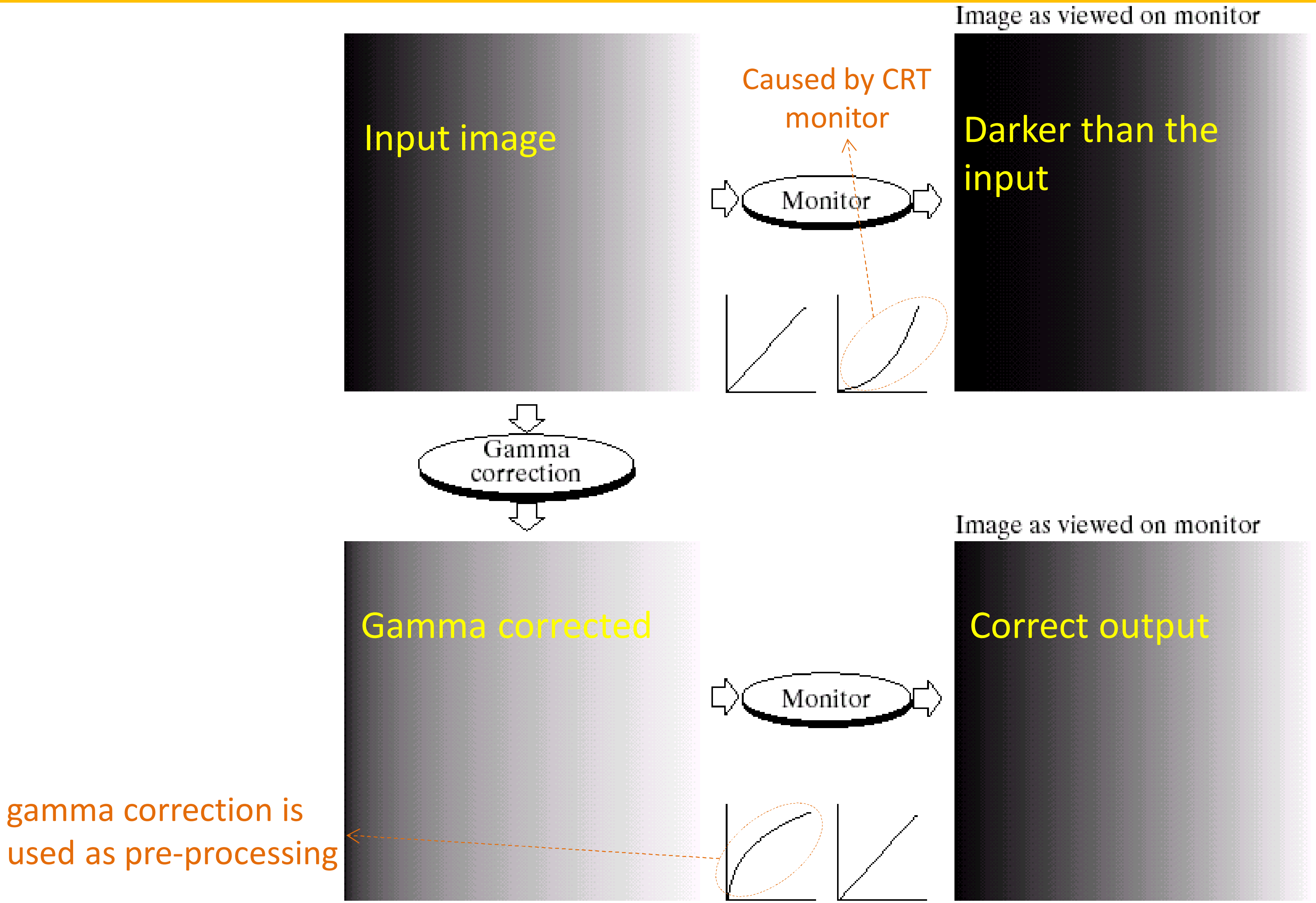
$$\boxed{s = cr^{\gamma}}$$

where

- c and γ are positive constants

- Different transformation curves are obtained by varying γ (gamma).
- Many image capturing, printing and display devices use gamma correction which enhances the given images by power-law response phenomena.

Power-Law Transformations : Gamma Correction Application

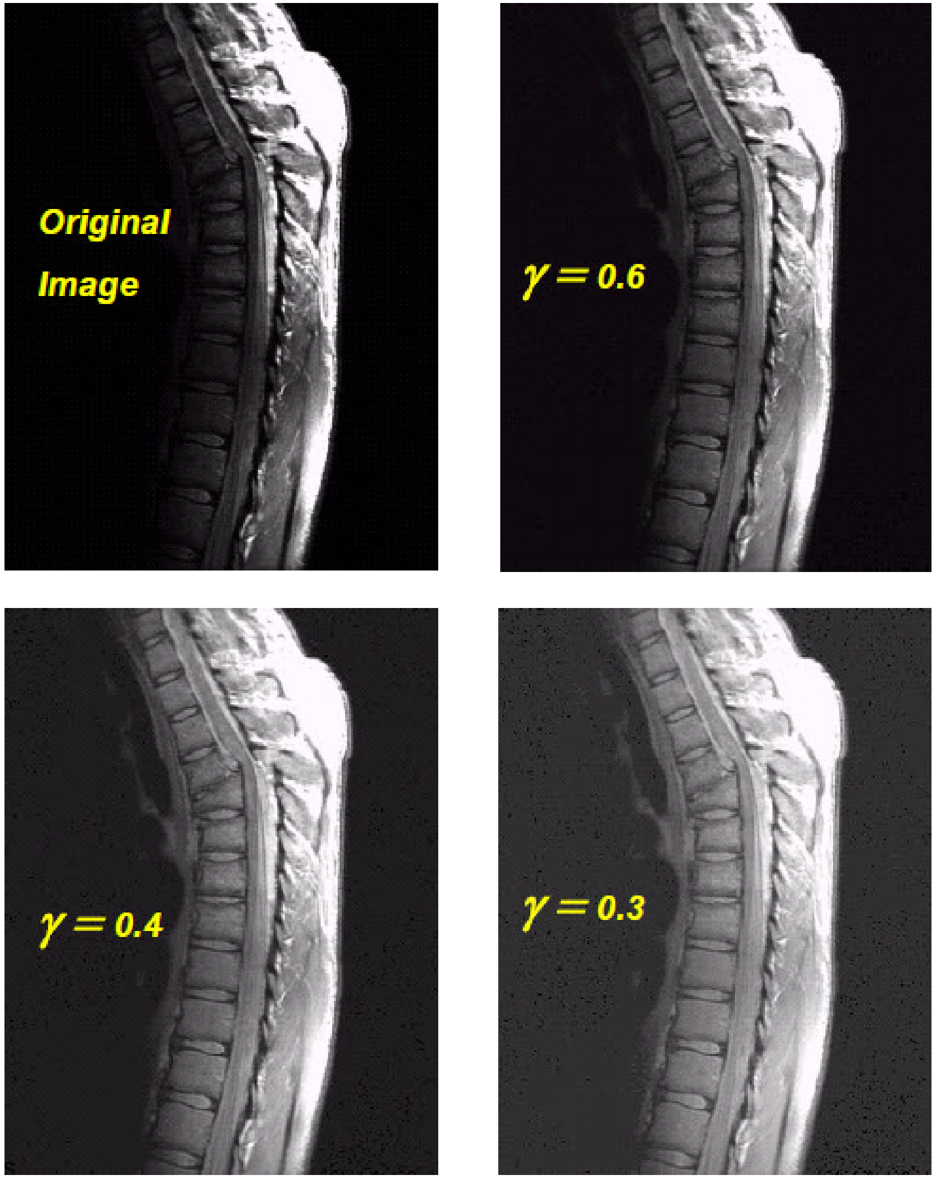


Example : Power-Law Transformations

The magnetic resonance image (MRI) of a fractured human spine.

Example : Power-Law Transformations

- Image is predominantly dark.
- Power-Law Transformation with a fractional exponent.

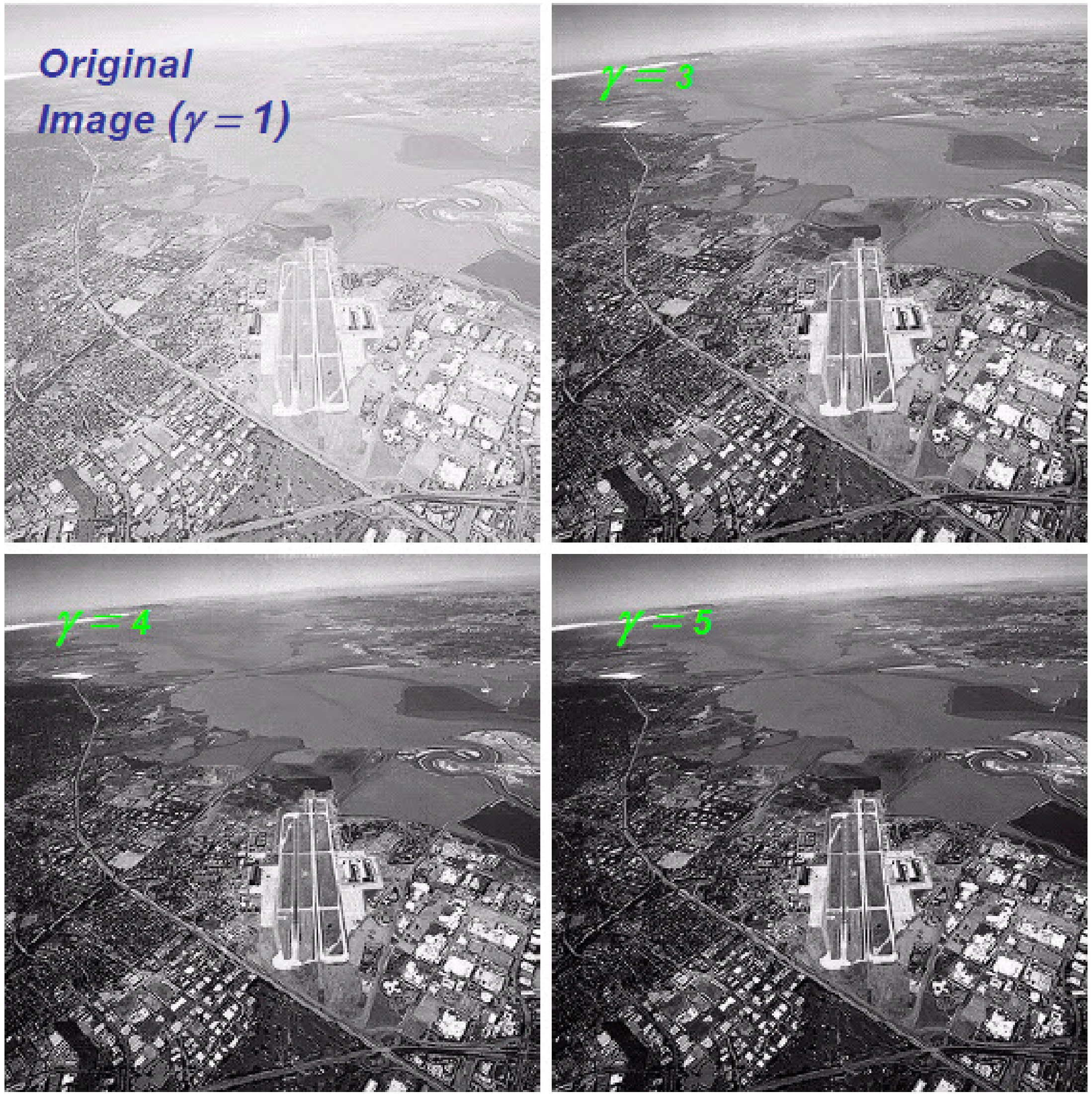


Example : Power-Law Transformations

An aerial image of a runway.

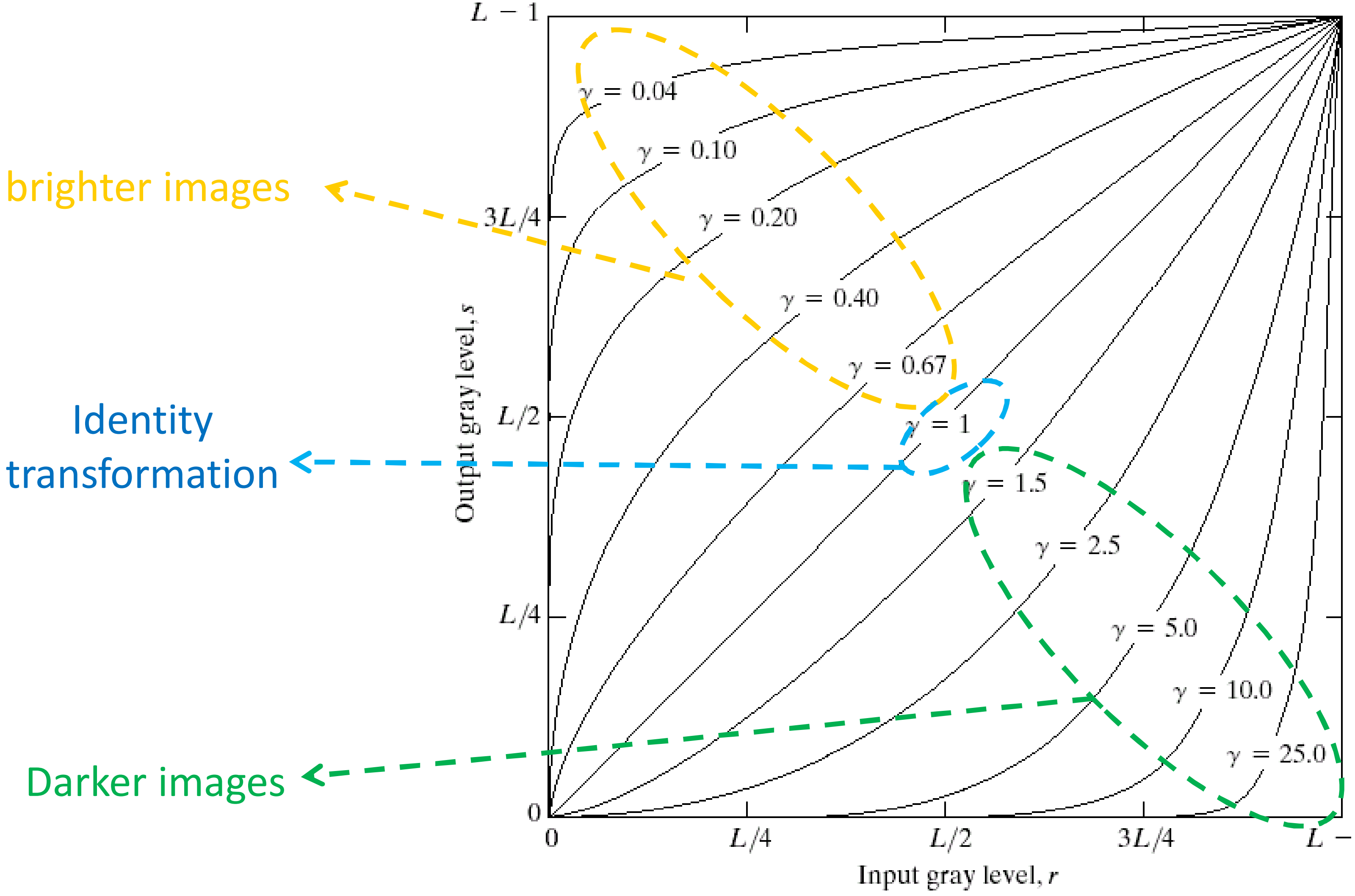
Example : Power-Law Transformations

- The image to be enhanced now has a washed-out appearance.



- Power-Law Transformation using values of $\gamma > 1$

Example : Power-Law Transformations



- Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases)

Piecewise-Linear Transformations

- Advantage : The principal advantage of piecewise linear functions over the types of functions we have discussed thus far is that the form of piecewise functions can be arbitrarily complex.
- Disadvantage : The principal disadvantage of piecewise functions is that their specification requires considerably more user input.

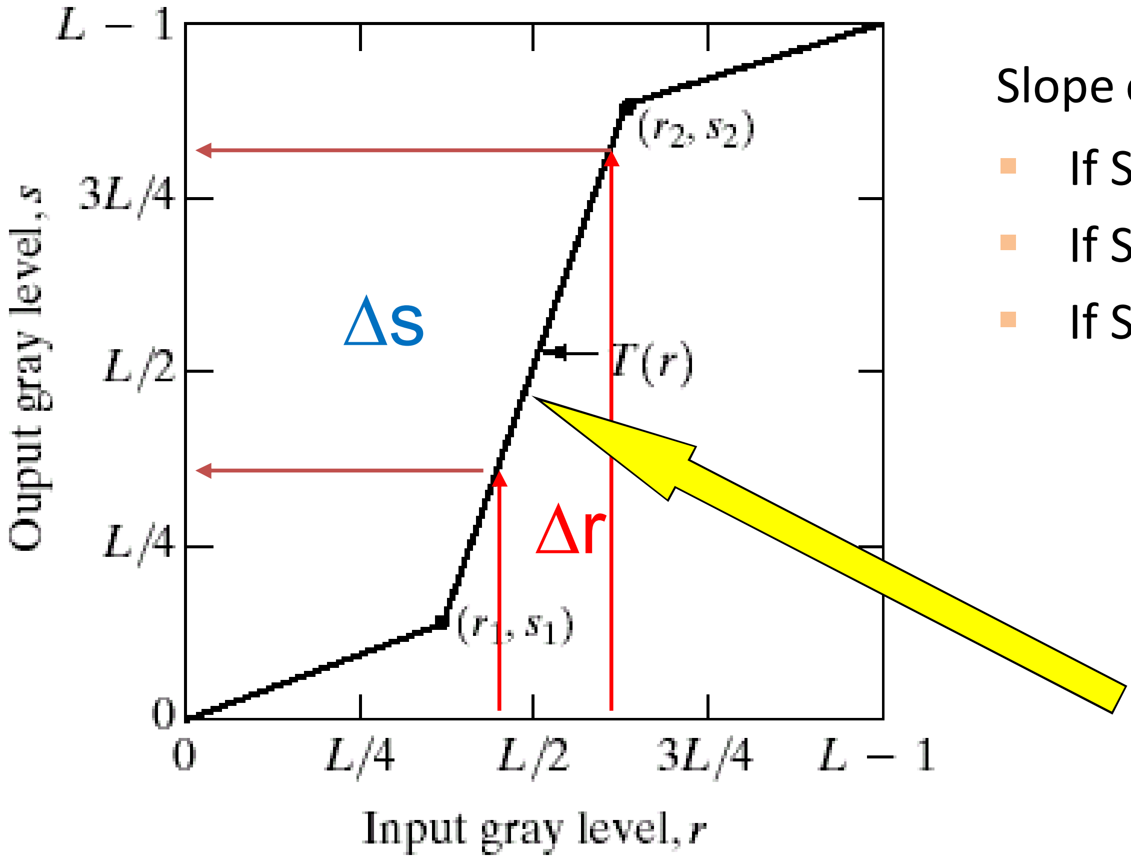
- There are three categories of piecewise-linear functions:
    - Contrast stretching
    - Gray-level slicing
    - Bit-plane slicing

Contrast Stretching

- One of the simplest piecewise functions is the contrast stretching which is used to enhance the low contrast images.
- Low-contrast images can result from :
    - poor illumination
    - lack of dynamic range in the imaging sensor
    - or even wrong setting of a lens aperture during image acquisition.
- Contrast stretching aims to increase (expand) the dynamic range of an image. It transforms the gray levels in the range [0,L-1].

How to know where the contrast is enhanced ?

- The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function.



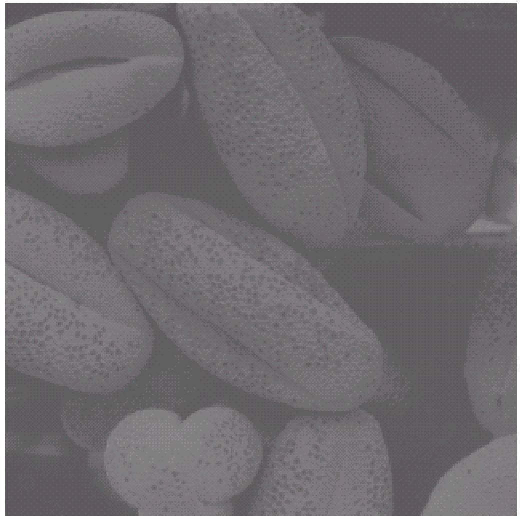
Slope of T(r)

- If Slope > 1 → increase contrast
- If Slope < 1 → decrease contrast
- If Slope = 1 → stable contrast

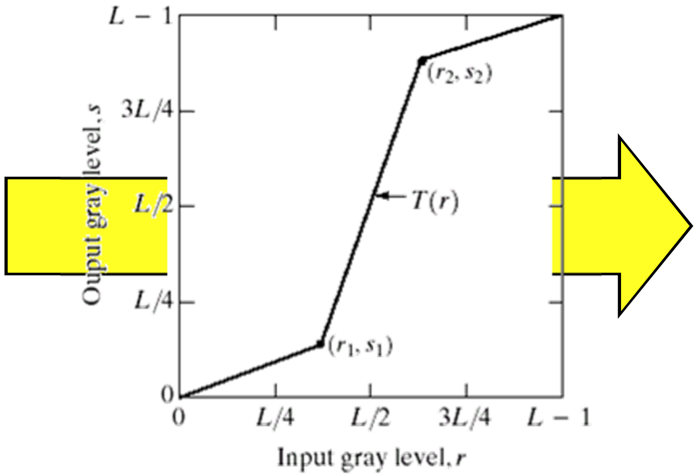
Δr narrow, get Δs wide.
that means increase contrast

Example : Contrast Stretching

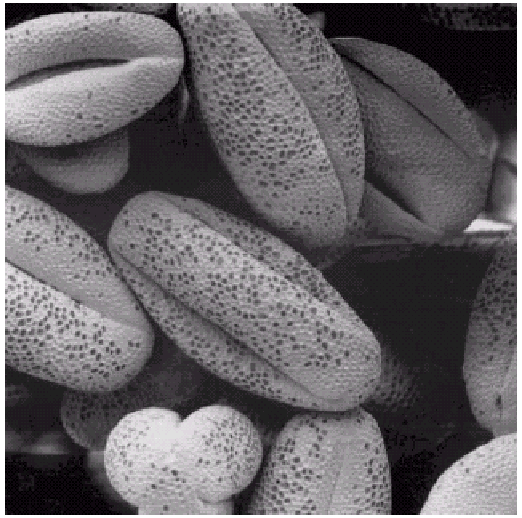
- Rather than using a well defined mathematical function we can use arbitrary user-defined transforms.
- The images below show a contrast stretching linear transform to add contrast to a poor quality image.



Low-contrast image



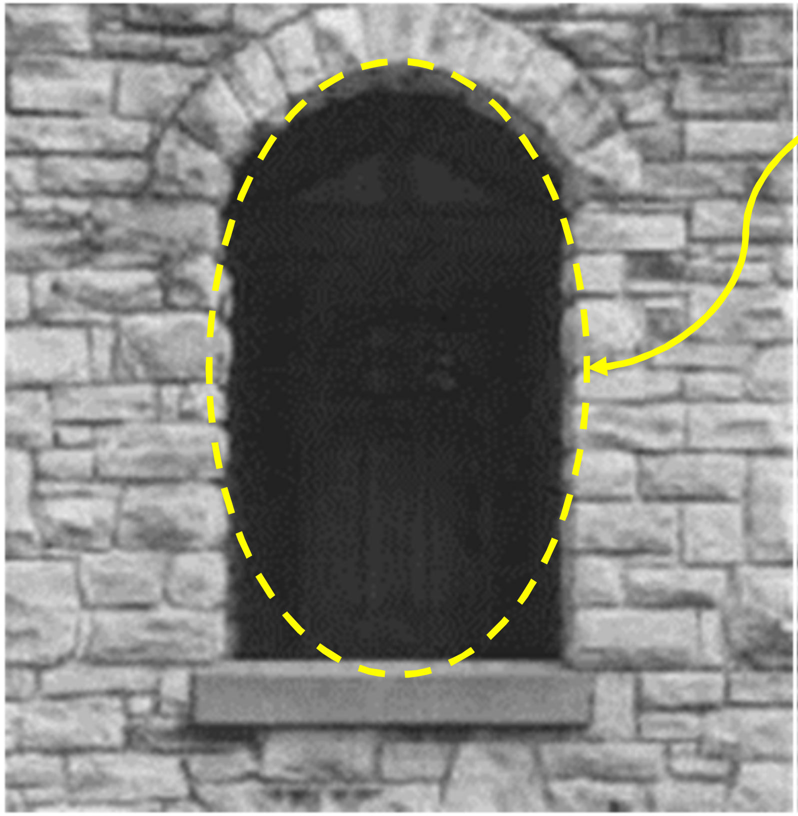
Contrast stretching



Result of contrast stretching

Example : Contrast Stretching

- An low-contrast image having 256 gray-levels is showed as following :

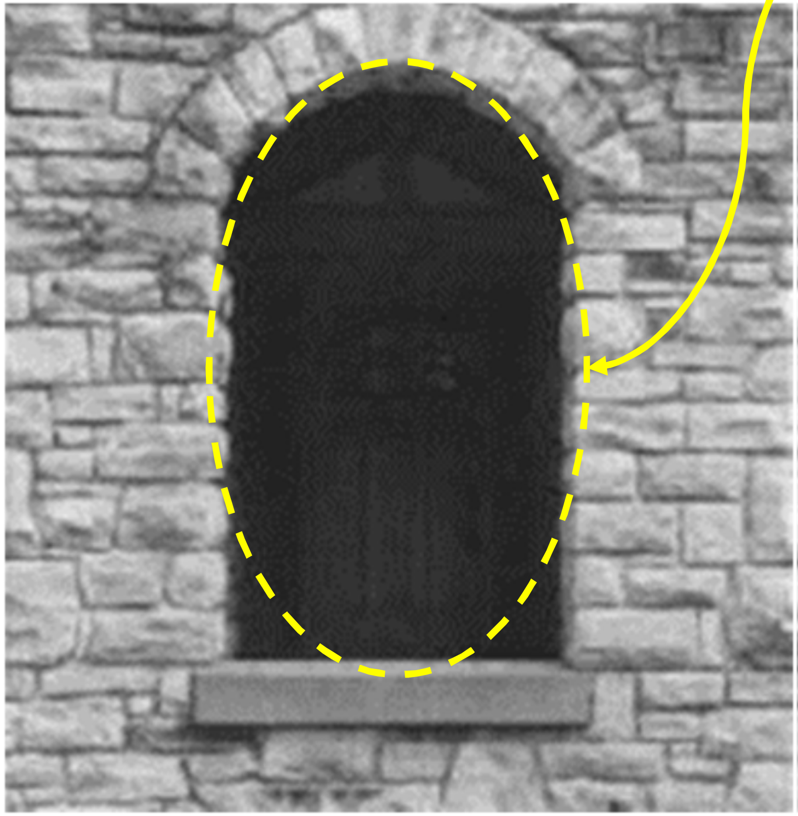


Low-contrast image

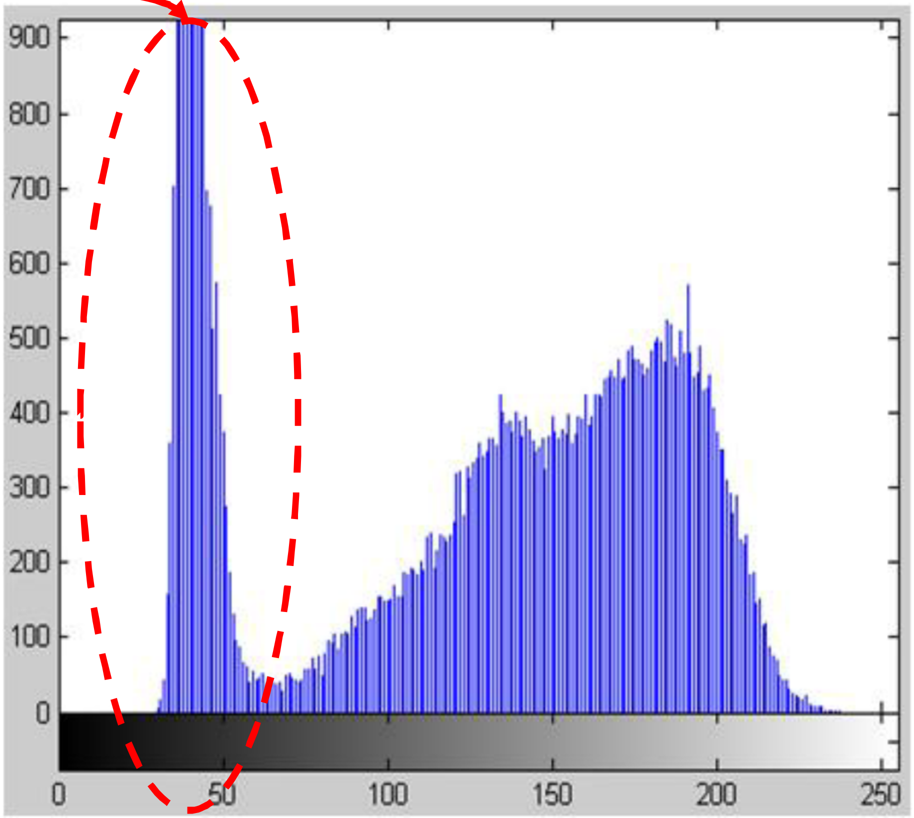
How we can highlight parts of the image?

Example : Contrast Stretching

How we can highlight parts of the image?



Low-contrast image



Corresponding histogram

Example : Contrast Stretching

- To increase the dynamic of image, contrast stretching is used to transform the gray-level values in the interval [28 to 75].
- We use the equation of a straight line to compute the contrast stretching for each line:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

- For the input gray-level values in the interval [28 to 75], we get:

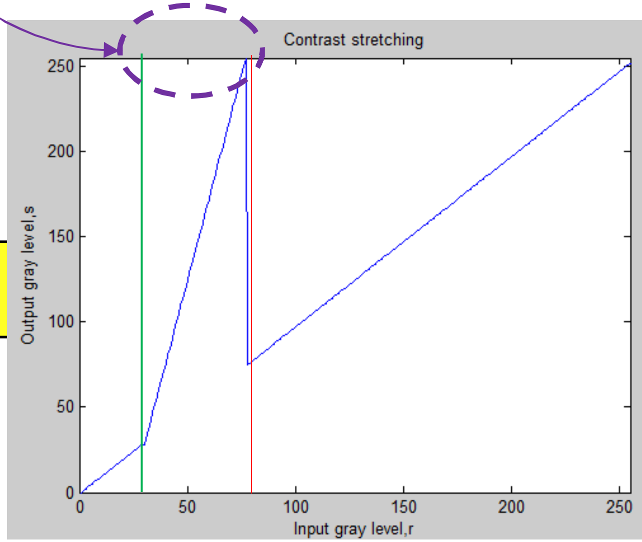
$$y - 28 = \frac{255 - 28}{75 - 28}(x - 28)$$

$$y = \frac{(227 * x - 5040)}{47}$$

- For the following contrast stretching function

$$s = \begin{cases} (227 * x - 5040)/47, & if\ 28 \le r \le 75 \\ r, & otherwise \end{cases}$$

Low-contrast image



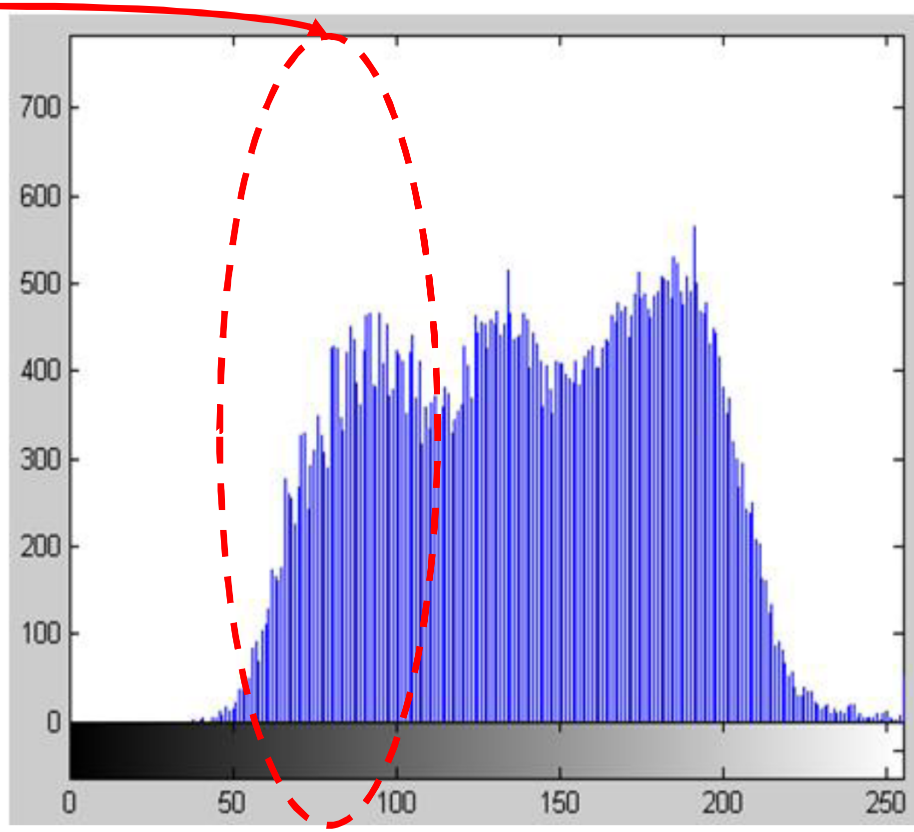
Contrast stretching

Result of contrast stretching

Example : Contrast Stretching

After using contrast stretching, the contrast of the image is shown as below.

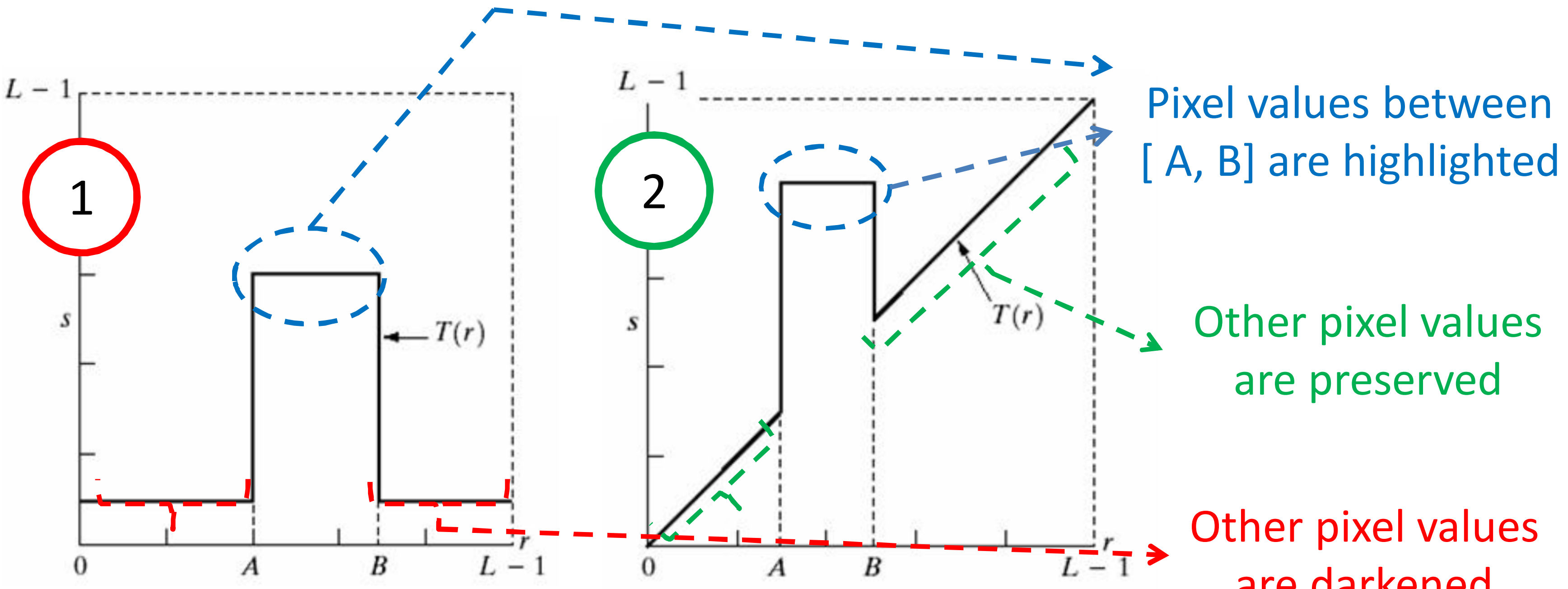
Result of contrast stretching



Corresponding histogram

Gray-Level Slicing

- This method is used to highlight a specific range of gray levels [A, B] in a given image.
- Applications include enhancing features such as masses of water in satellite imagery and flaws in X-ray images.



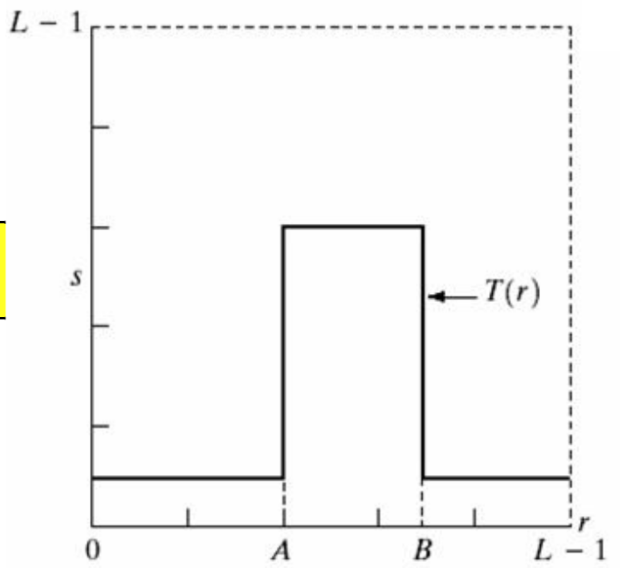
(Background changed to back) (Background unchanged)

Example : Gray-level slicing

Background changed to back

- Highlights a specific range of gray-levels [A, B].
- Other levels can be darkened.

Input image



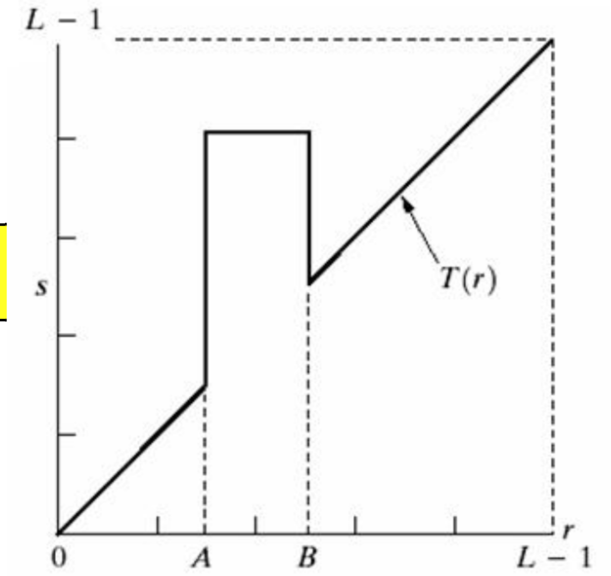
Gray-level slicing

Result of applying gray-level slicing on the input image

Background unchanged

- Highlights a specific range of gray-levels [A, B].
- Other levels can be preserved.

Input image

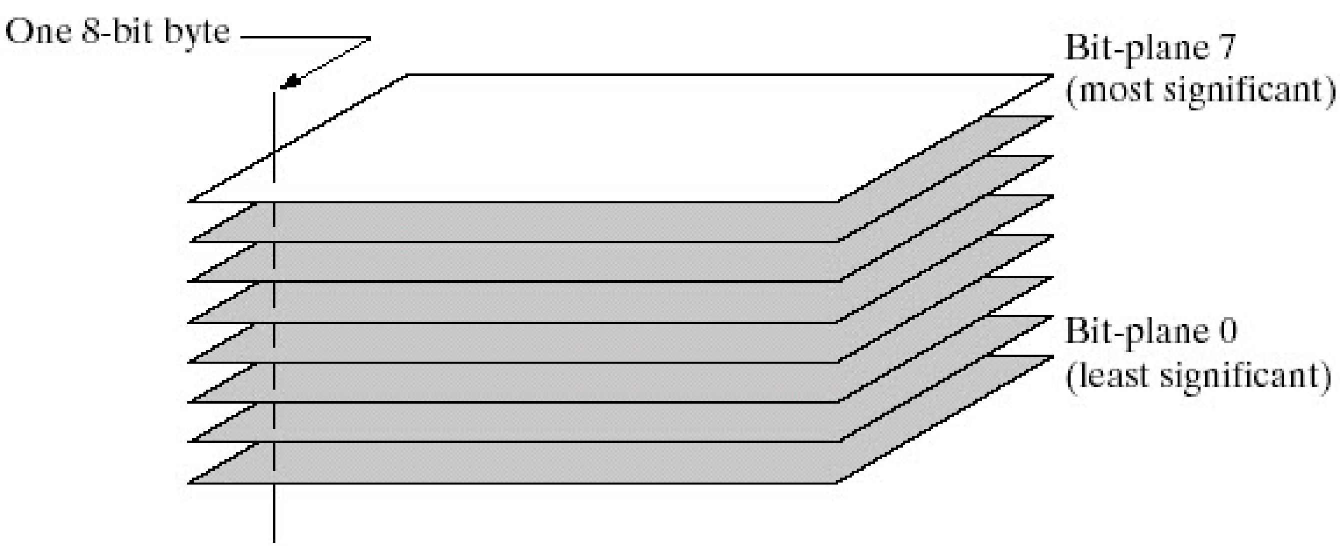


Gray-level slicing

Result of applying gray-level slicing on the input image

Bit-Plane Slicing

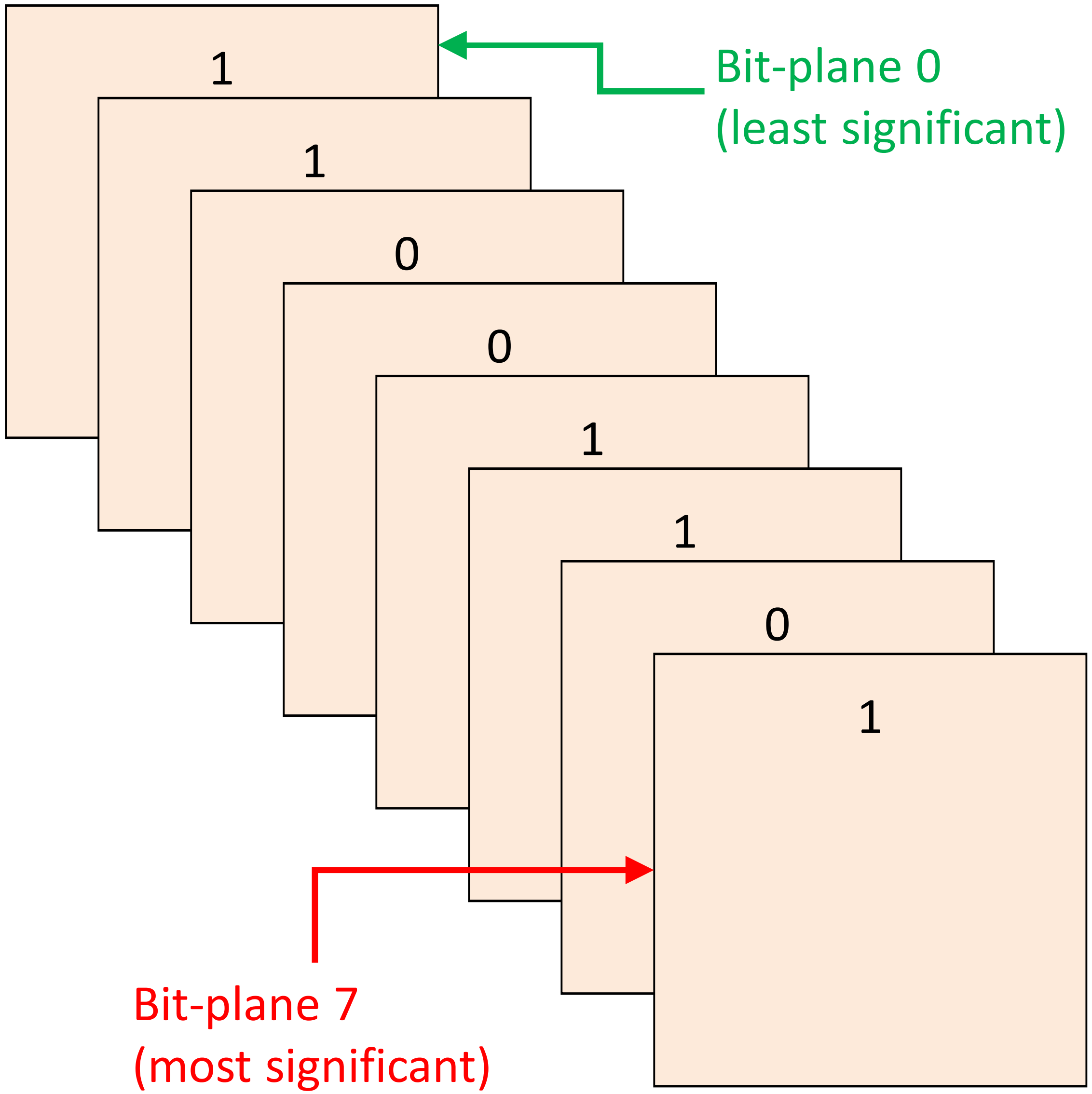
- Instead of highlighting gray-level ranges, highlighting the contribution made by each bit is useful and used in image compression.
- Most significant bits contains the majority of the visually significant data.



Bit-plane representation of an 8-bit image

Bit-Plane Slicing

- The 8-bit valve is 179 (decimal)
- The bit planes of 8-bit valve 10110011 (binary)

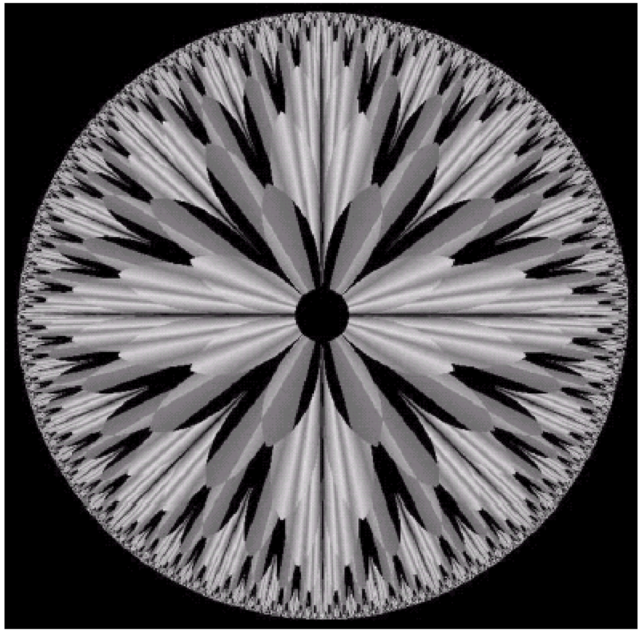


Bit-Plane Slicing

- The 8-bit valve is 179 (decimal)
- The bit planes of 8-bit valve 10110011 (binary)

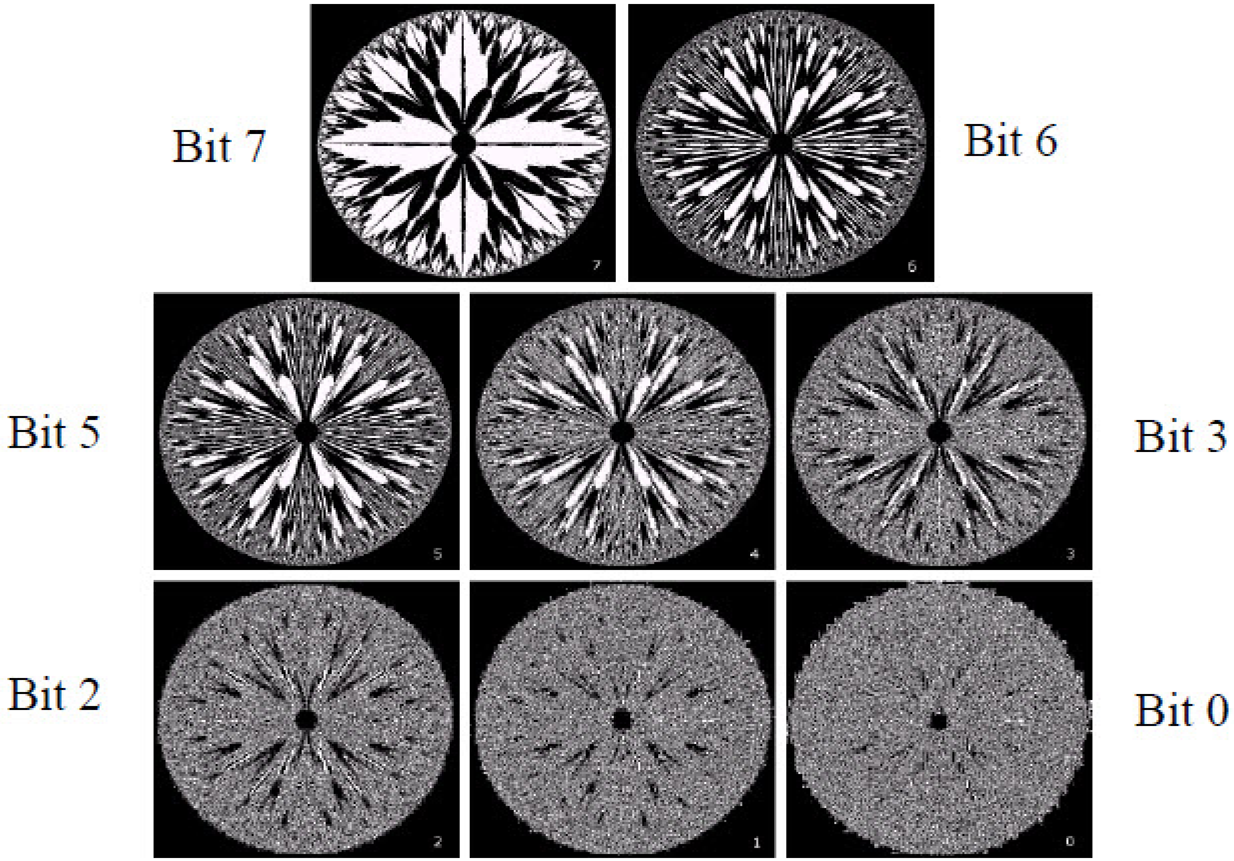
Bit Plane	Value	Contribution	Running Total
7	1	1 * 2^7 = 128	128
6	0	0 * 2^6 = 0	128
5	1	1 * 2^5 = 32	160
4	1	1 * 2^4 = 16	176
3	0	0 * 2^3 = 0	176
2	0	0 * 2^2 = 0	176
1	1	1 * 2^1 = 2	178
0	1	1 * 2^0 = 1	179

Example1 : Bit-plane slicing



An 8-bit fractal image.
(A fractal is an image generated from mathematical expressions).

Example1 : Bit-plane slicing



The eight bit planes of the image.
The number at the bottom, right of each image identifies the bit plane

Example2 : Bit-plane slicing

An 8-bit Original image.

Example2 : Bit-plane slicing

An 8-bit Original image

The eight bit planes of the image.

The number at the bottom, center of each image identifies the bit plane.

Example2 : Bit-plane slicing

Reconstructed image using only bit planes 7 and 6

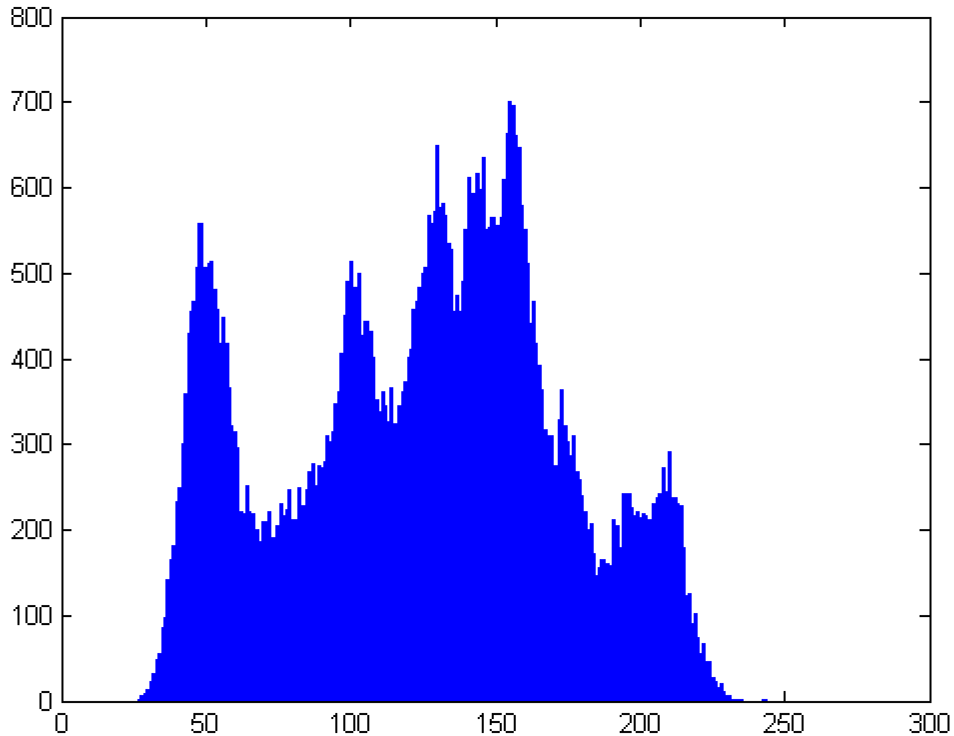
Reconstructed image using only bit planes 7, 6 and 5

Reconstructed image using only bit planes 6, 5 and 4

Histogram Processing

- The histogram of an image is a plot that records the frequency distribution of gray levels in that image.
- Massively useful in image processing, especially in segmentation.

Lena image



Histogram of Lena image

Histogram Processing

- The histogram of an image with gray levels in the range [0, L-1] is a discrete function :

$$\boxed{h(r_k) = n_k}$$

where

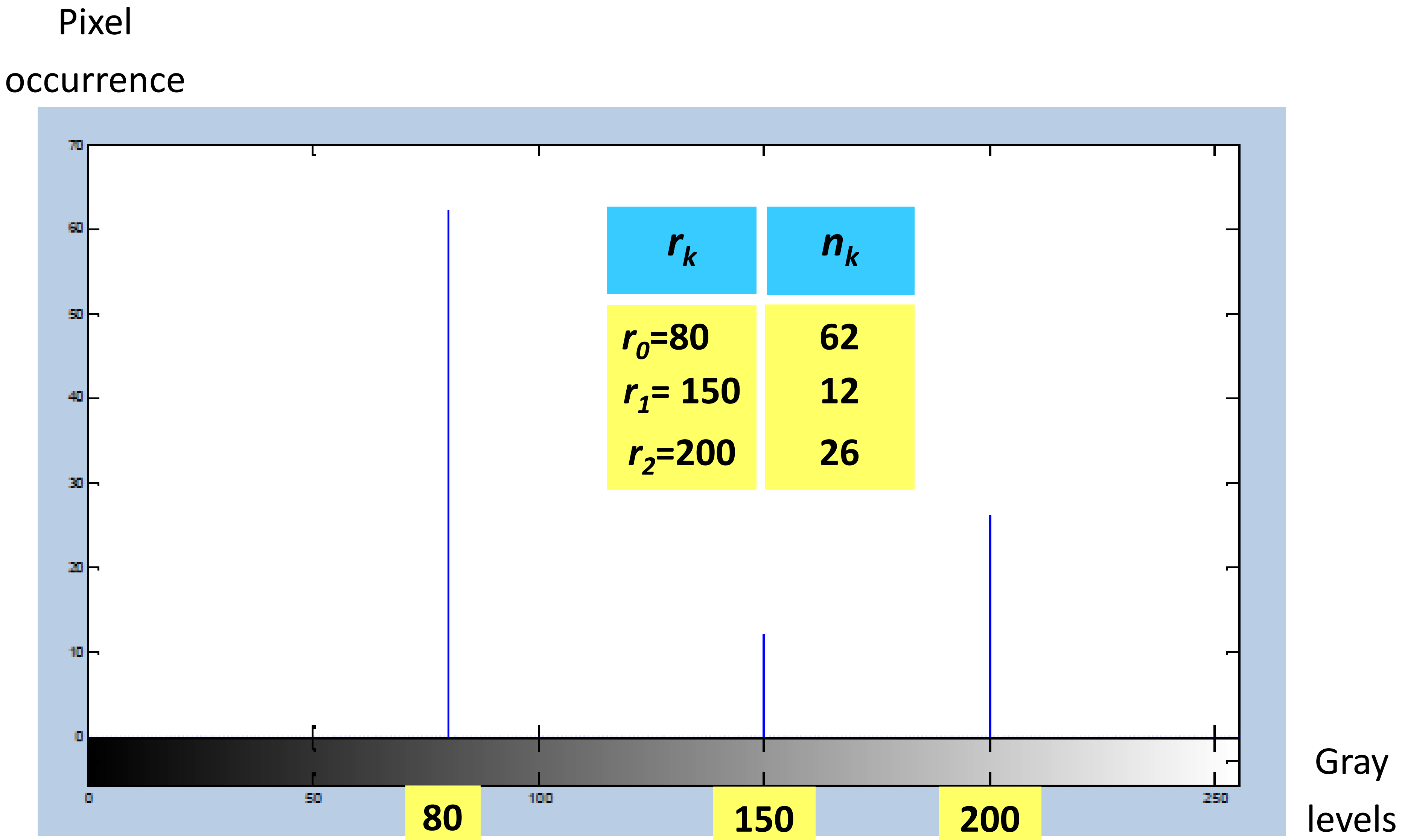
- k = 0, 1, …, L-1
- r_k is the k^{th} gray levels
- n_k is the number of pixels in the image having gray level r_k

Example : Histogram Processing

- An 8-bit image having size 10x10 pixels (MxN = 100) is showed as following :

80	80	80	80	80	80	80	80	80	80
80	200	200	200	200	80	80	80	80	80
80	200	200	200	200	80	80	80	80	80
80	200	200	200	200	80	80	80	80	80
80	200	200	200	200	80	80	80	80	80
80	200	200	200	200	80	80	80	80	80
80	200	200	200	150	150	150	150	80	80
80	200	200	200	150	150	150	150	80	80
80	80	80	80	150	150	150	150	80	80
80	80	80	80	80	80	80	80	80	80

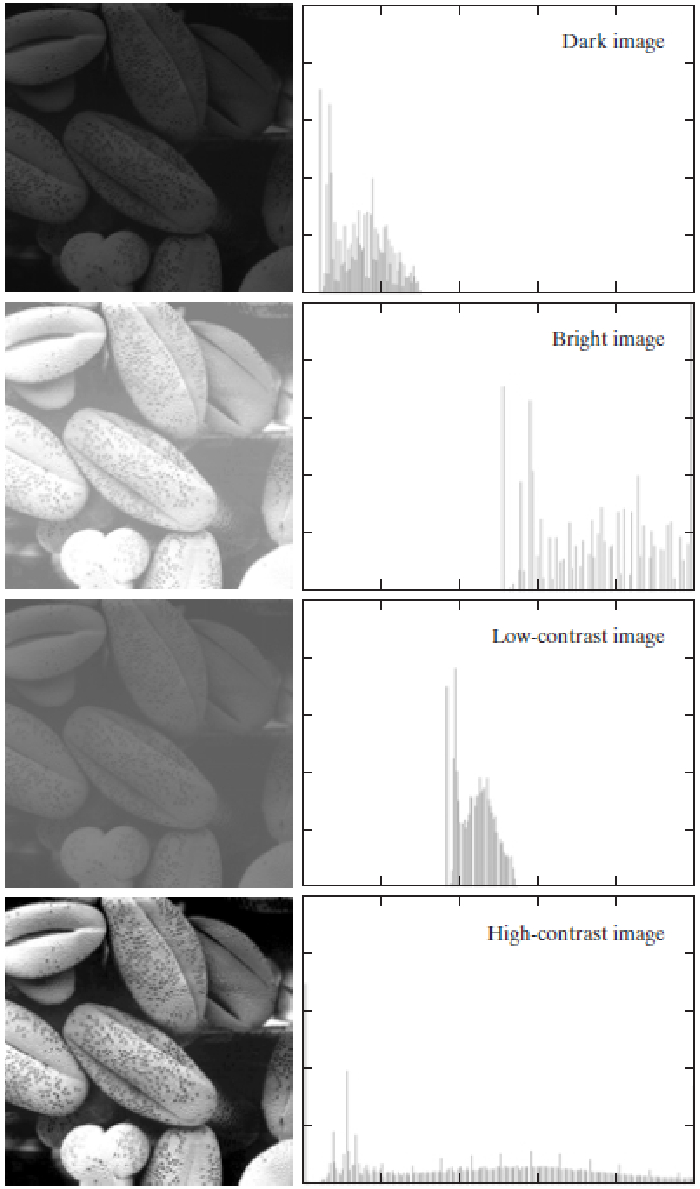
Example : Histogram of 8-bit image



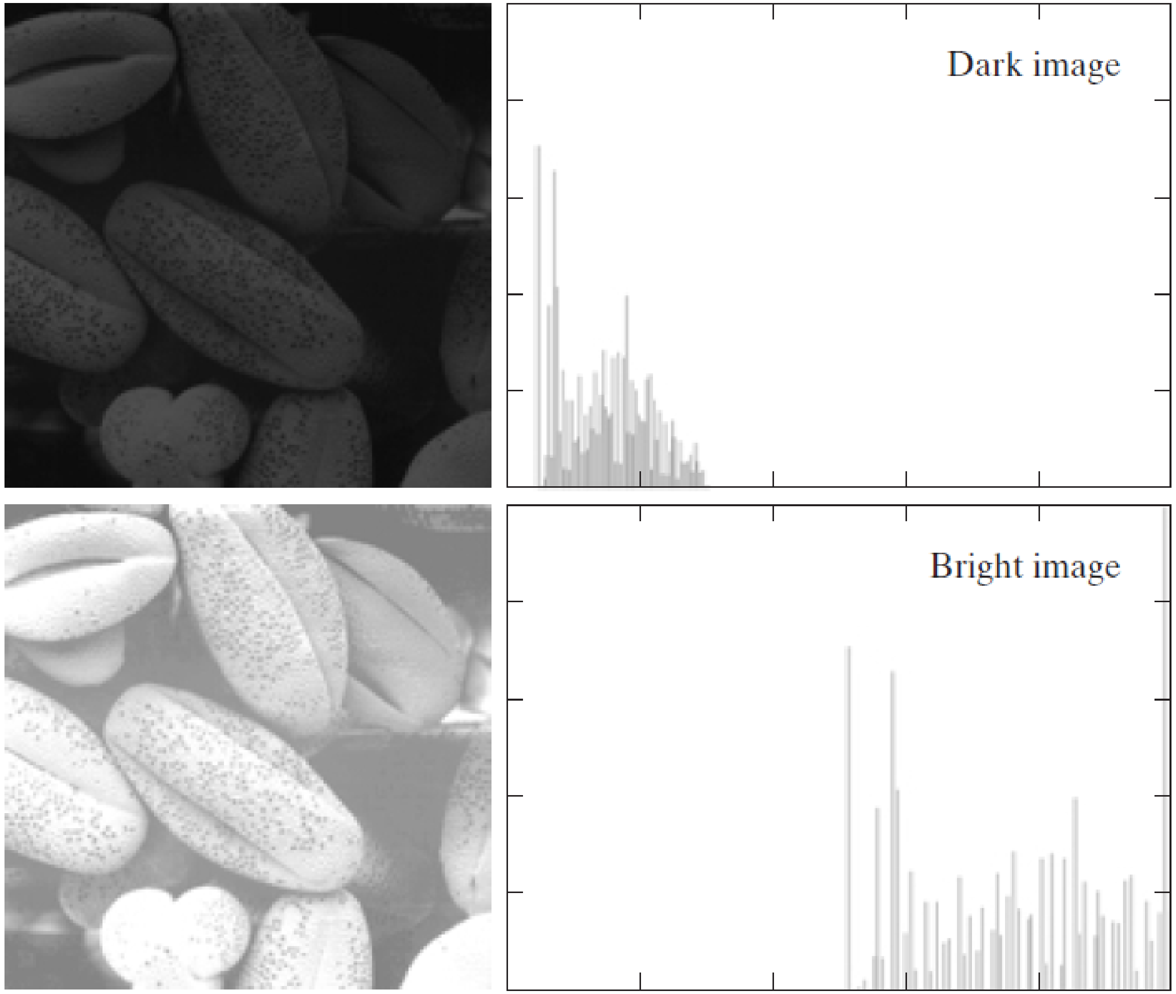
r_k	n_k
r_0=80	62
r_1= 150	12
r_2=200	26

Histogram Processing

- Histograms of four basic gray-level characteristics.
- The x axis shows the full range of the gray levels in the given image and the y axis corresponds to the number of occurrences of each gray $h(r_k)$ or the probability of a pixel to have that gray-level value, $p(r_k)$.

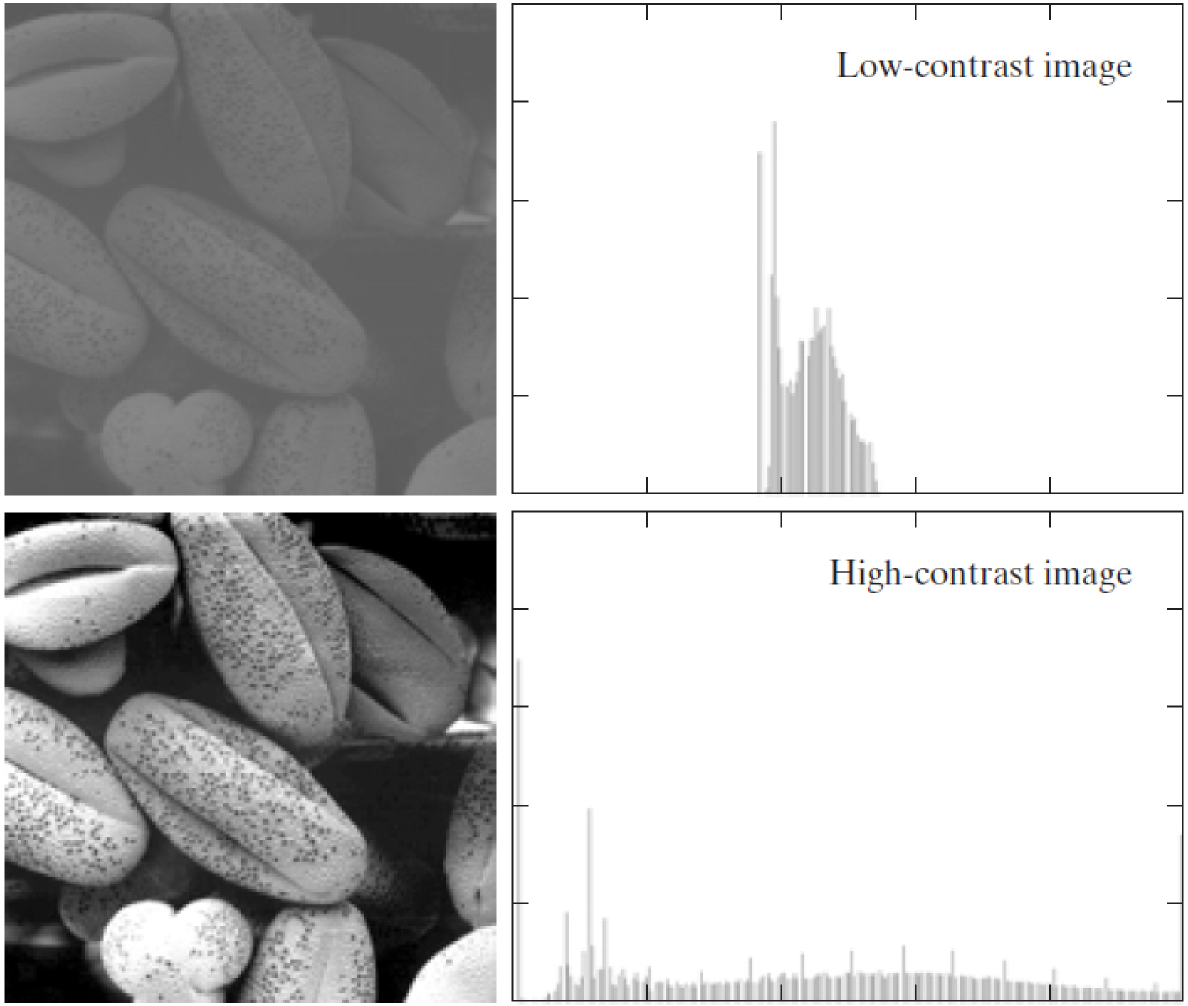


Histogram Processing



Histogram of Different Contrast and Brightness Variation

Histogram Processing

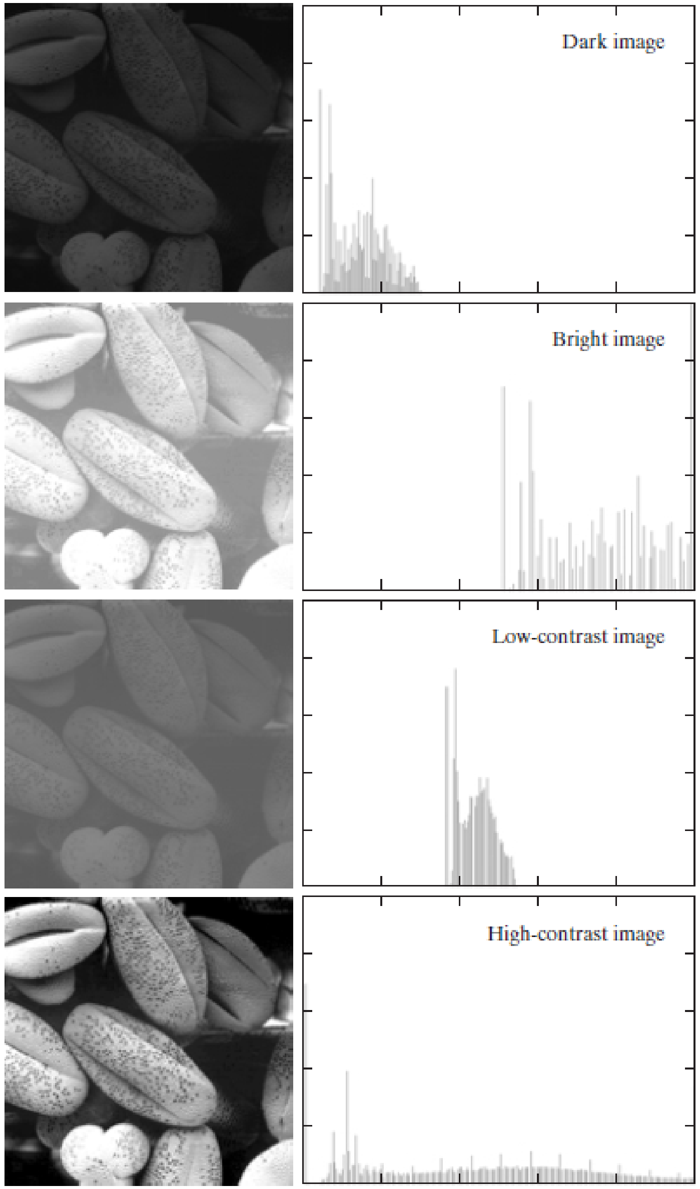


Histogram of Different Contrast and Brightness Variation

Histogram Processing

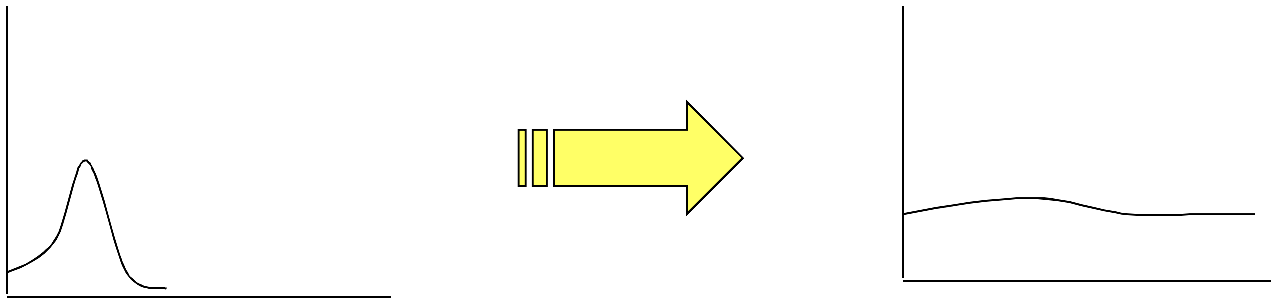
A selection of images and their histograms.

- Notice the relationships between the images and their histograms.
- Note that the high contrast image has the most evenly spaced histogram.



Histogram Equalization

- Histogram equalization is a method which increases the dynamic range of the gray-levels in a low-contrast image to cover full range of gray-levels.

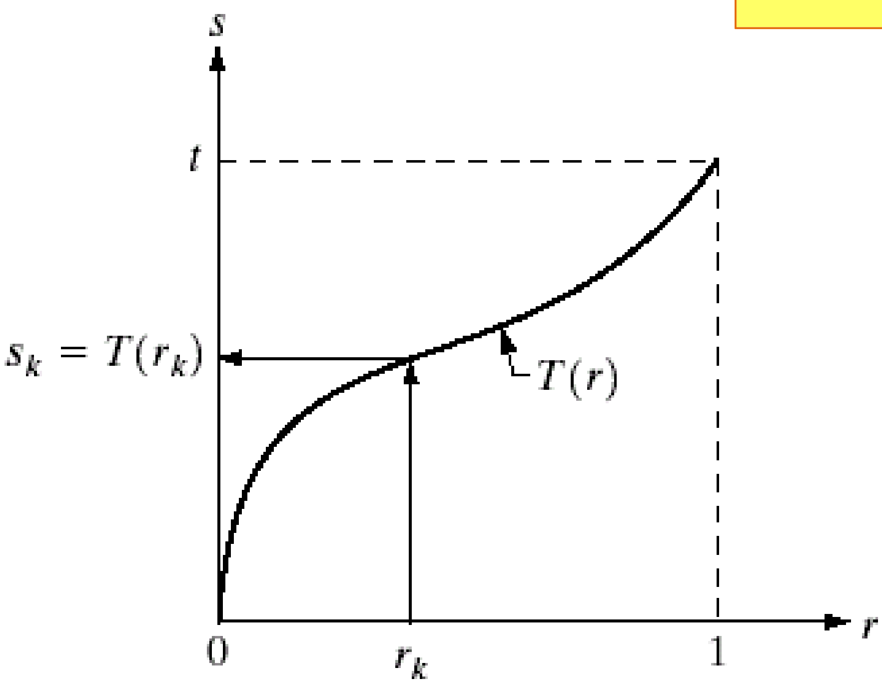


- span a fuller range of the gray scale.
- "automatic", no need for further parameter specifications.
- Also, the simplicity of the computations that would be required to implement the technique.

Monotonically Increasing Function

- Monotonically Increasing Function refers to the function that when r has increased, then T(r) is increased or constant only, no decrease.
- Histogram processing have the following form :

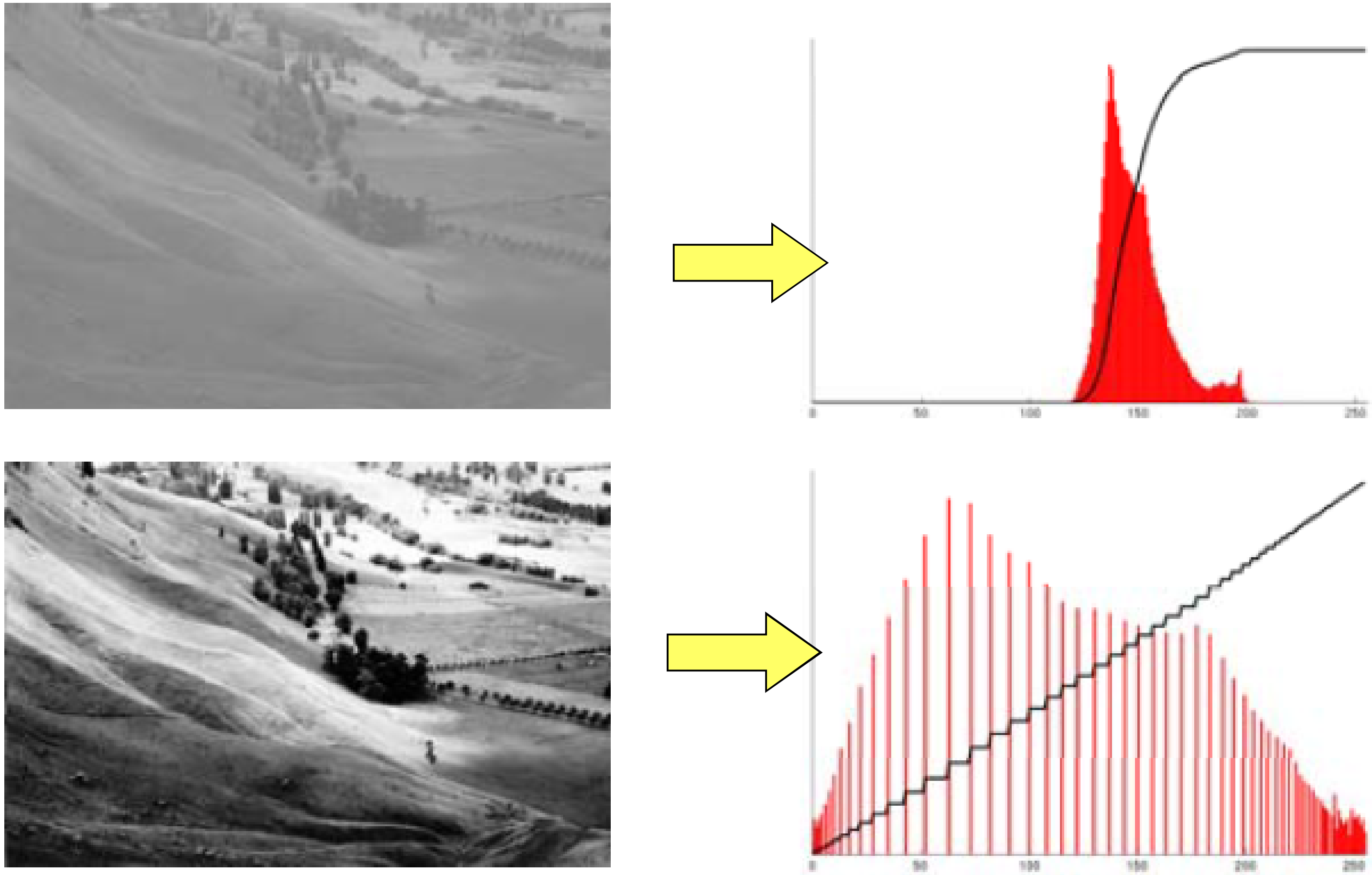
$$\boxed{s = T(r)}$$



1. Monotonically increasing function

2. $0 \le T(r) \le 1 \quad \text{for} \quad 0 \le r \le 1$

Example : Histogram Equalization



Histogram Equalization

- Histogram equalization is an automatic enhancement technique which produces an output image that has a near uniformly distributed histogram.
- The formula for histogram Equalization is given

$$s_k = T(r_k) = \sum_{j=1}^{k} p_r(r_j) = \sum_{j=1}^{k} \frac{n_j}{n}$$

where

r_k: the k^{th} gray levels

k: the gray levels range (k = 0, 1, …, L-1)

n_j: the number of pixels in the image j

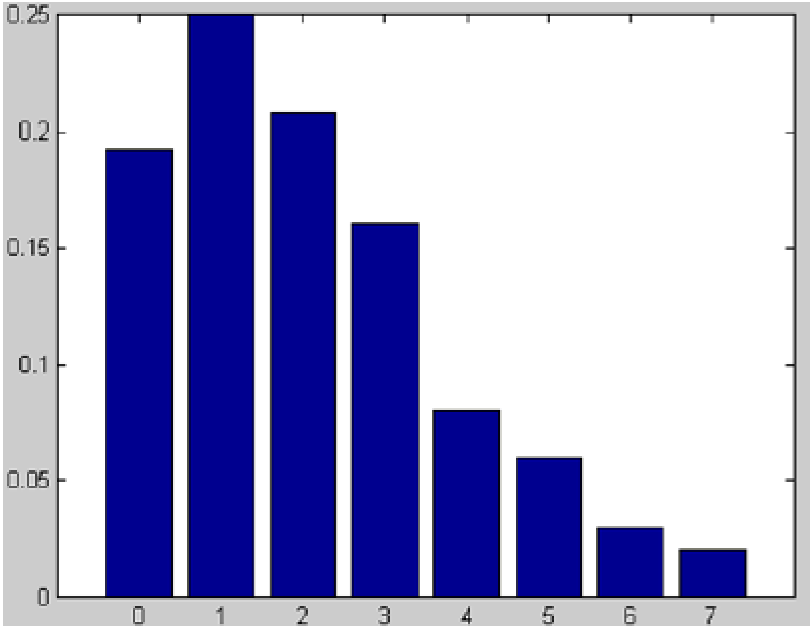
n: the total number of pixels

Example : Histogram Equalization

- Assuming there is an image 3 bit (L=8) which having size 64x64 pixels (MxN = 4096). There are gray levels distribution values as Table in which the gray levels range [0, L-1] = [0, 7].

r_k	n_k	$p_k(r_k) = n_k/MN$
$r_0=0$	790	0.1929
$r_1= 1$	1023	0.2498
$r_2=2$	850	0.2075
$r_3=3$	656	0.1602
$r_4=4$	329	0.0803
$r_5=5$	245	0.0598
$r_6=6$	122	0.0298
$r_7=7$	81	0.0198

$p_k(r_k)$



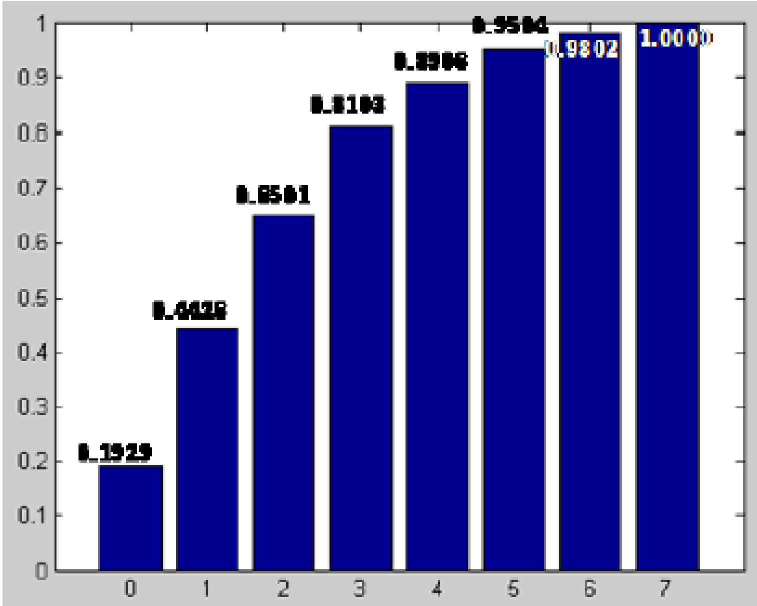
The cumulative of all probabilities of the normalized histogram is equal to 1.

Example : Histogram Equalization

- Then we find the cumulative of all probabilities of the normalized histogram :

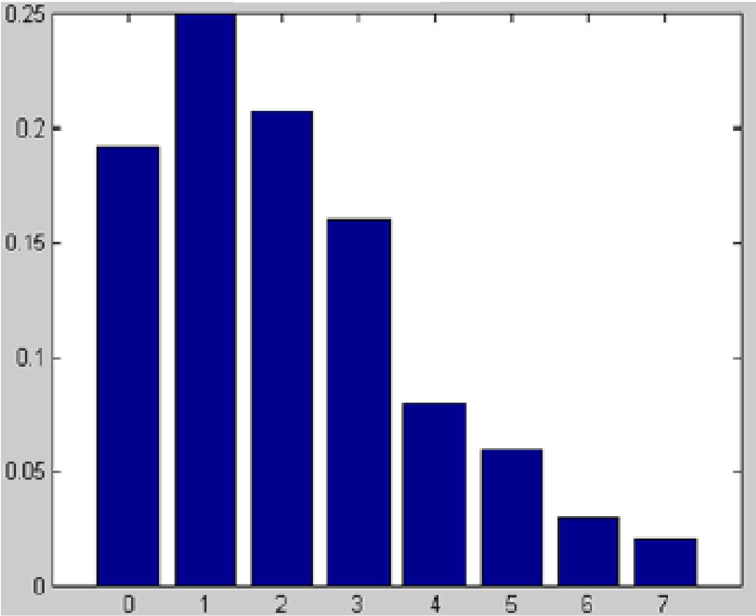
r_k	$p_k(r_k) = n_k/MN$	$c_k(r_k)$
$r_0=0$	0.1929	0.1929
$r_1= 1$	0.2498	0.4426
$r_2=2$	0.2075	0.6501
$r_3=3$	0.1602	0.8103
$r_4=4$	0.0803	0.8906
$r_5=5$	0.0598	0.9504
$r_6=6$	0.0298	0.9802
$r_7=7$	0.0198	1.0000

$c_k(r_k)$

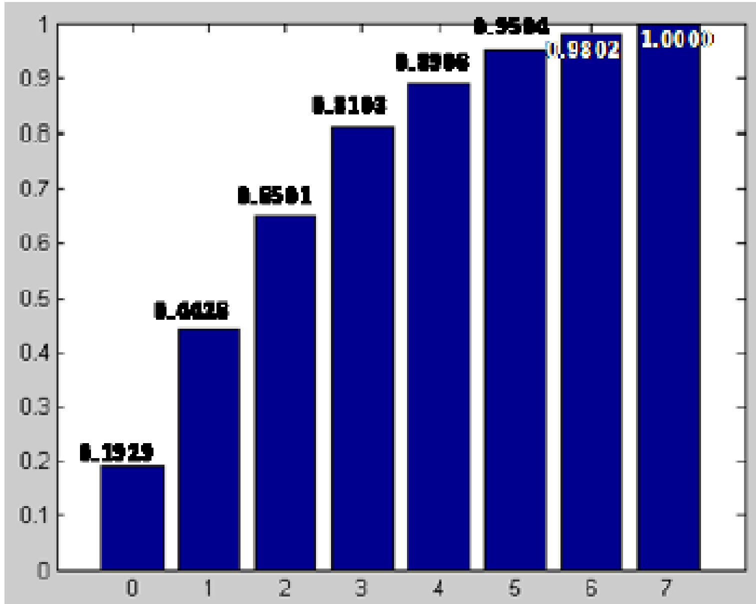


Example : Histogram Equalization

$p_k(r_k)$



$c_k(r_k)$



Example : Histogram Equalization

- Next, we find the transformation function:

$$s_0 = T(r_0) = \sum_{j=1}^{k} p_r(r_j) = 7p_r(r_0) = 1.3501$$

$$s_1 = T(r_1) = \sum_{j=1}^{k} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.0984$$

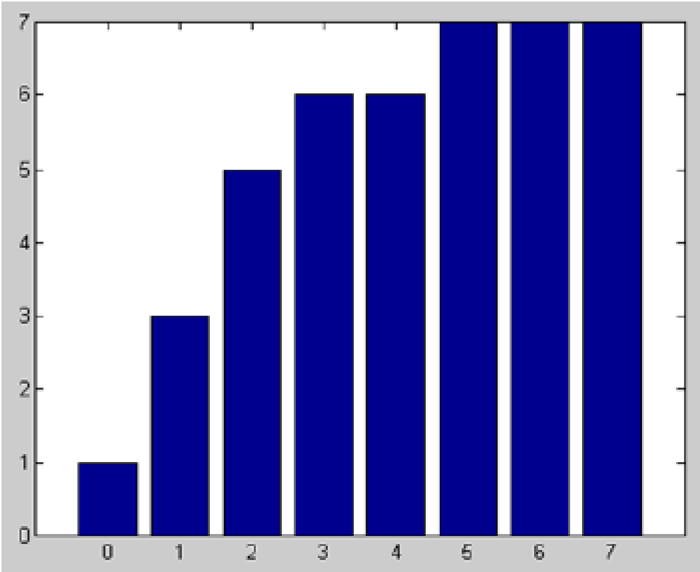
- and s_2 = 4.5510, s_3 = 5.6721, s_4 = 6.2344, s_5 = 6.6531, s_6 = 6.8616, s_7 = 7.00

Example : Histogram Equalization

- We round the values of *s* to the nearest integer:

r_k	$c_k(r_k)$	s_k
r_0=0	0.1929	1.3501 → 1
r_1= 1	0.4426	3.0984 → 3
r_2=2	0.6501	4.5510 → 5
r_3=3	0.8103	5.6721 → 6
r_4=4	0.8906	6.2344 → 6
r_5=5	0.9504	6.6531 → 7
r_6=6	0.9802	6.8616 → 7
r_7=7	1.0000	7.0000 → 7

s_k



Example : Histogram Equalization

- These are the values of the equalized histogram. Note that there are only five gray levels.

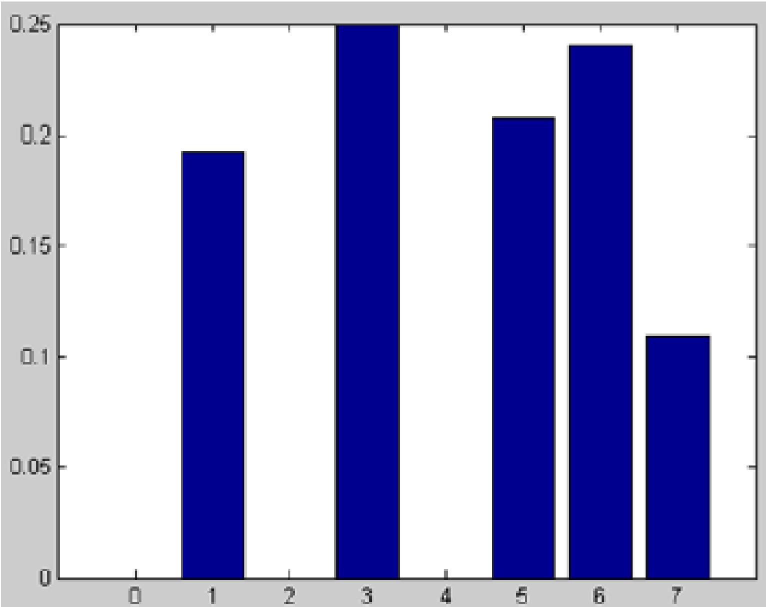
r_k	n_k	s_k	New n_k	$p_s(s_k)$ = New n_k/MN
$r_0=0$	790	$s_0=1$	790	0.1929
$r_1= 1$	1023	$s_1= 3$	1023	0.2498
$r_2=2$	850	$s_2=5$	850	0.2075
$r_3=3$	656	$s_3=6$	985 (656+329)	0.2405
$r_4=4$	329	$s_4=6$		
$r_5=5$	245	$s_5=7$	448 (245+122+81)	0.1094
$r_6=6$	122	$s_6=7$		
$r_7=7$	81	$s_7=7$		

Example : Histogram Equalization

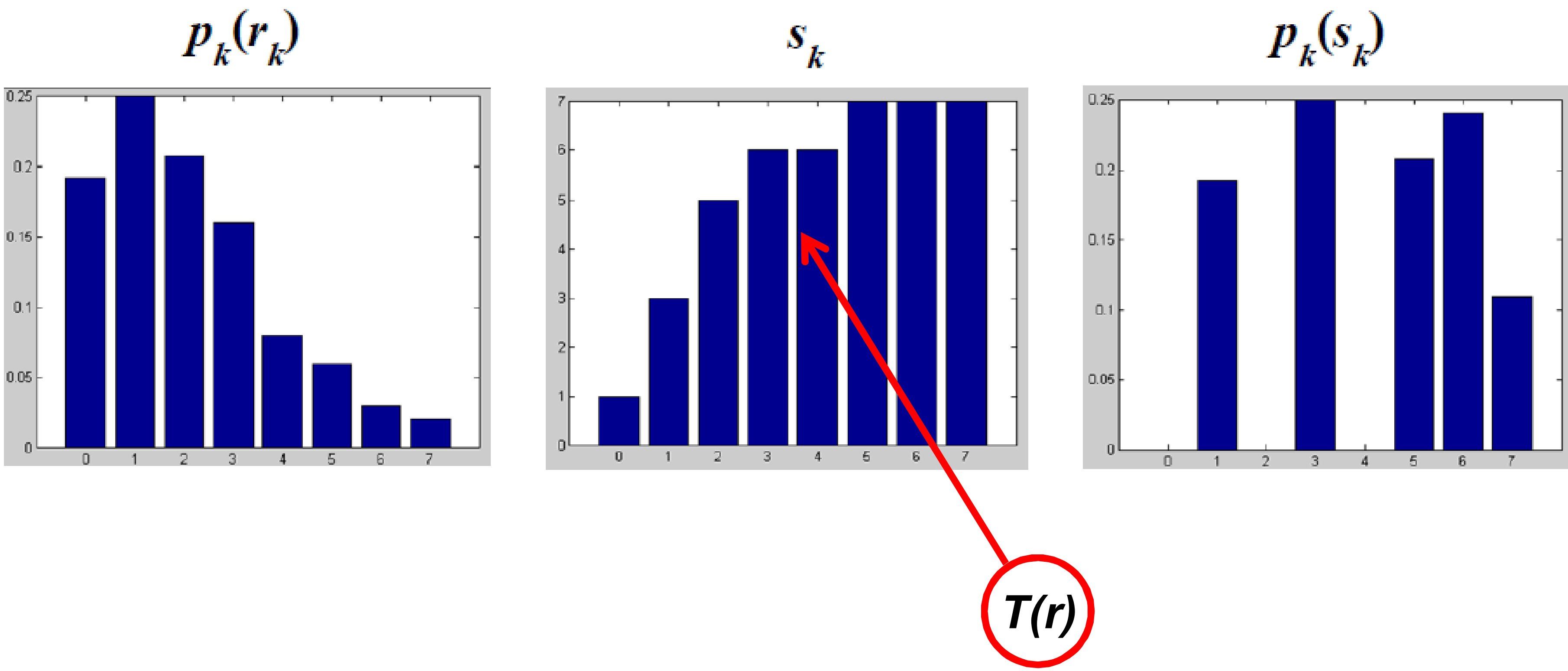
- Therefore, the histogram of the equalized image can be drawn as follows:

S_k	New n_k	$p_s(s_k)$
s_0=1	790	0.1929
s_1= 3	1023	0.2498
s_2=5	850	0.2075
s_3=6	985	0.2405
s_4=6		
s_5=7	448	0.1094
s_6=7		
s_7=7		

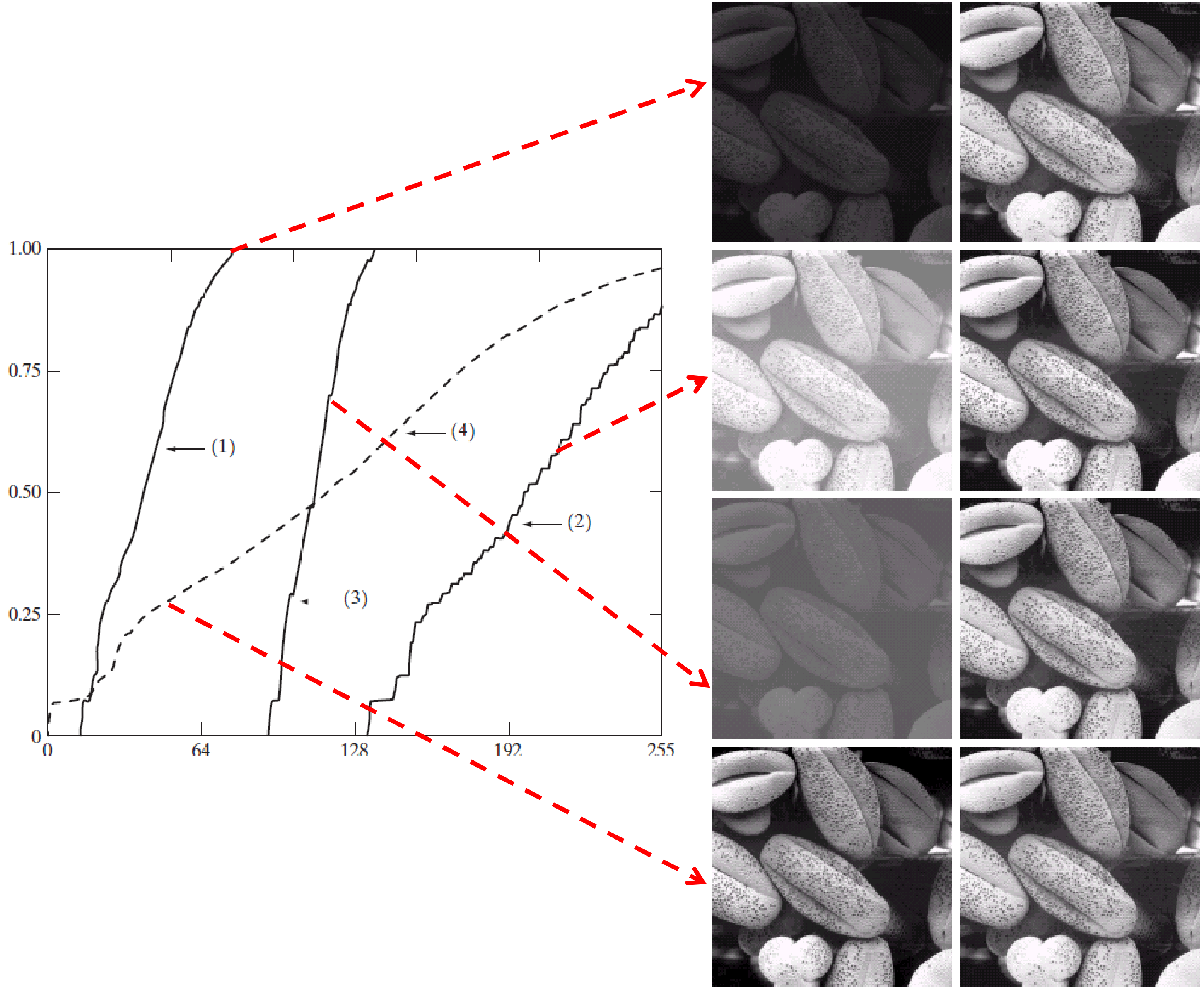
$p_k(s_k)$



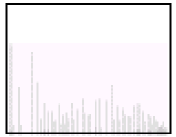
Example : Histogram Equalization



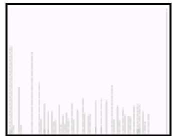
Example : Histogram Equalization Results



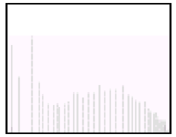
Dark image enhancement



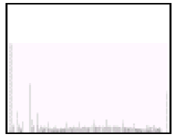
Bright image enhancement



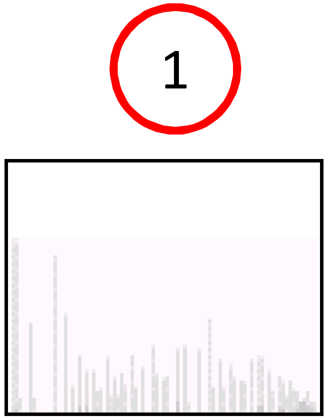
Low contrast image enhancement

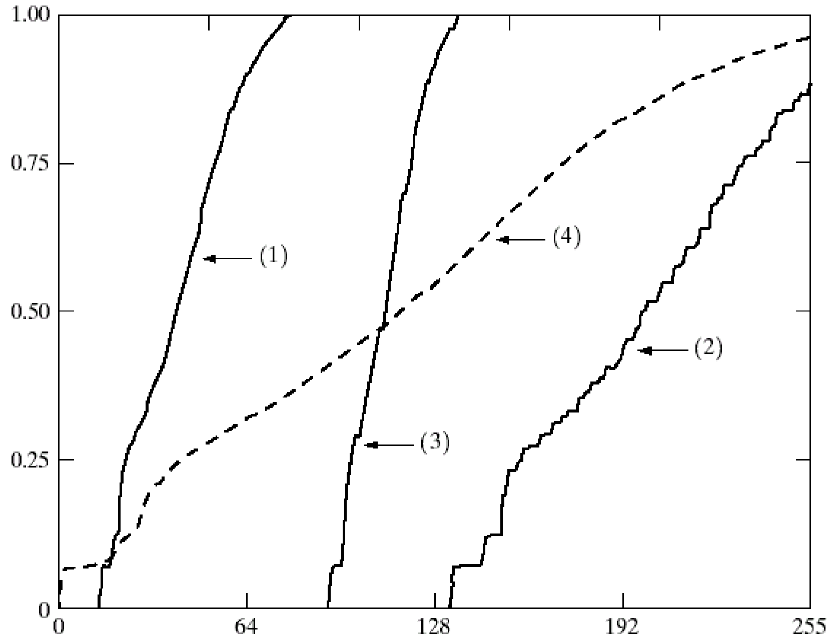


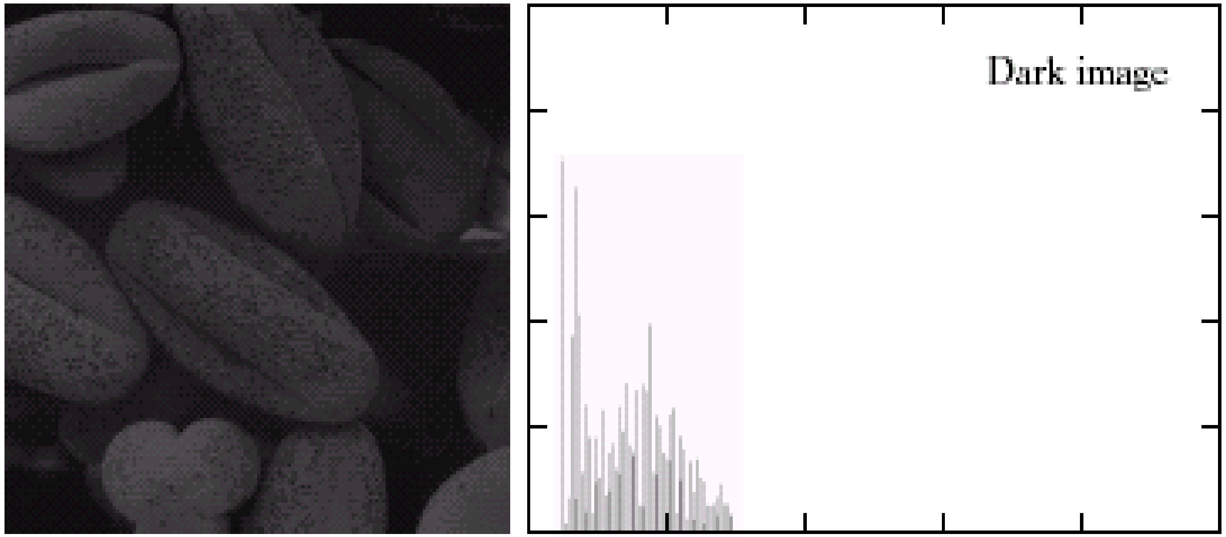
High contrast image almost no enhancement

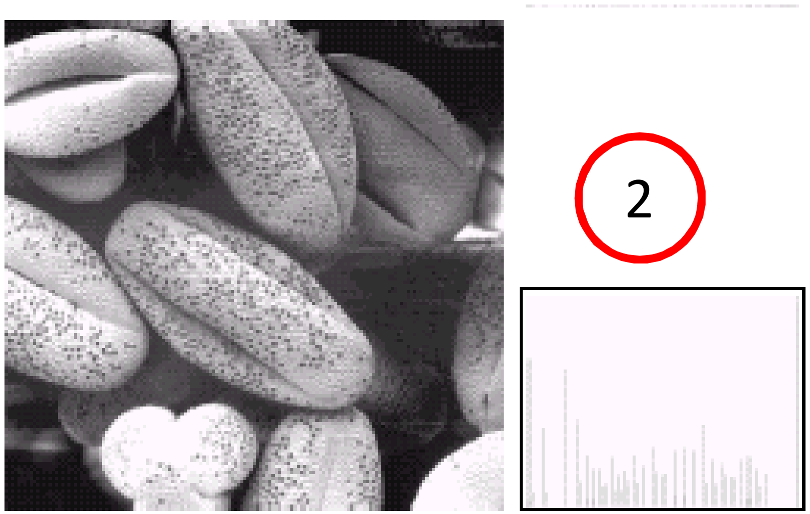


Example : Histogram Equalization Results

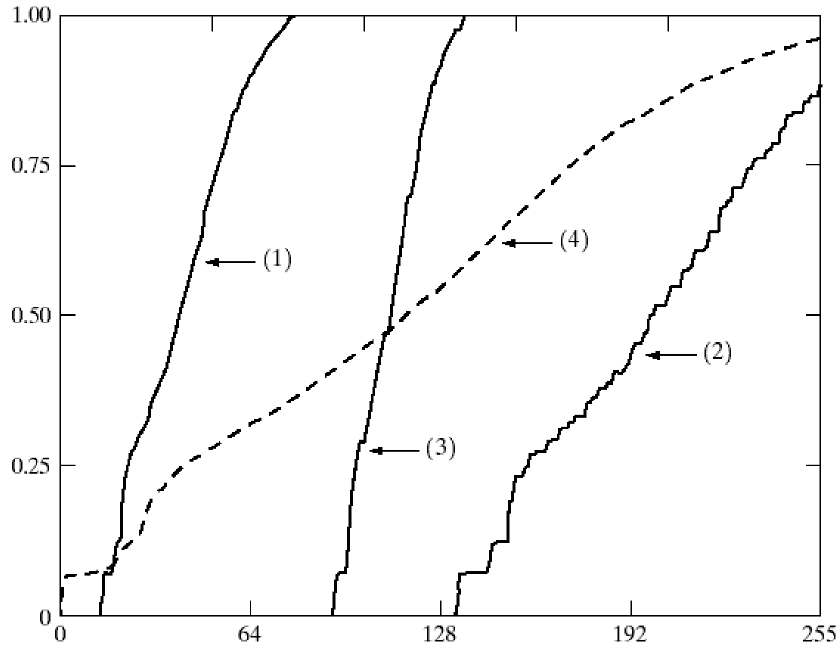


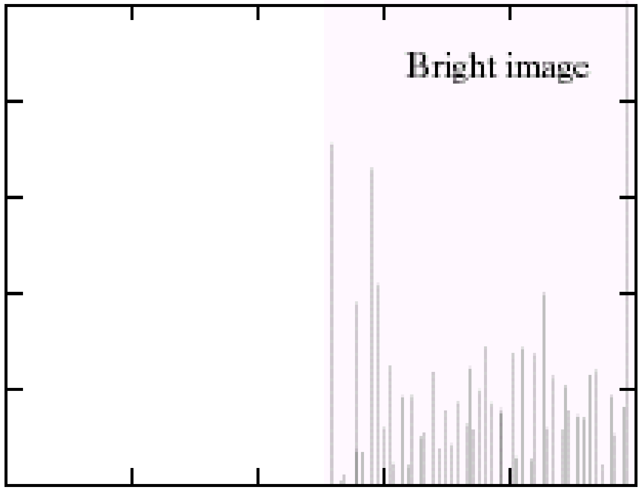




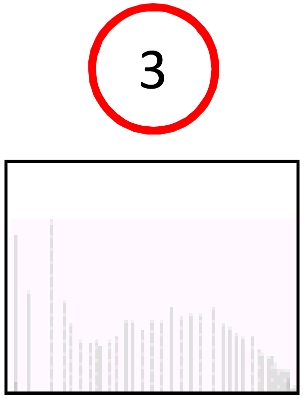


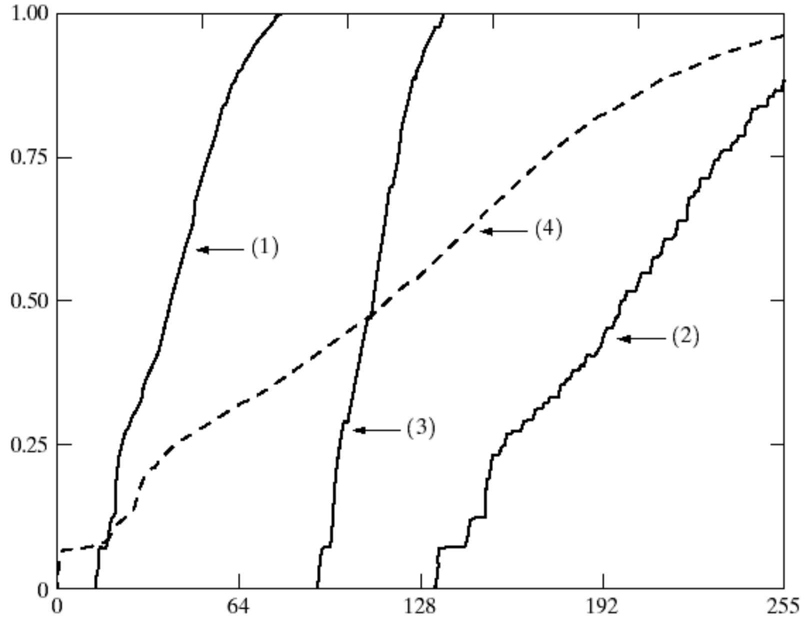
2

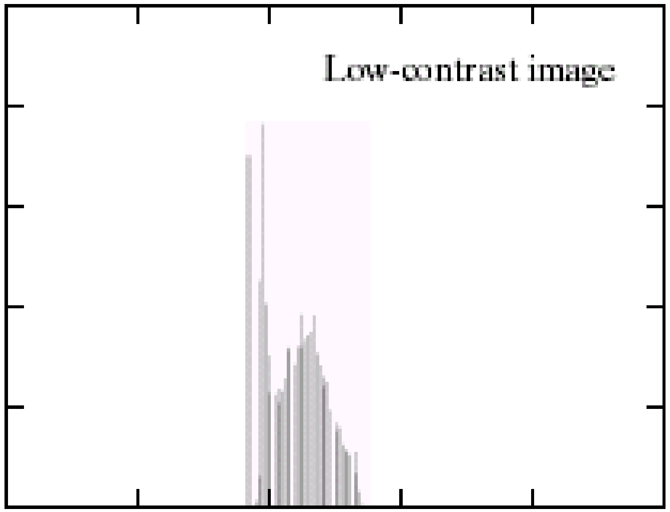




Example : Histogram Equalization Results

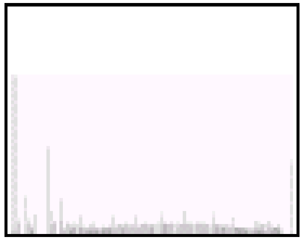


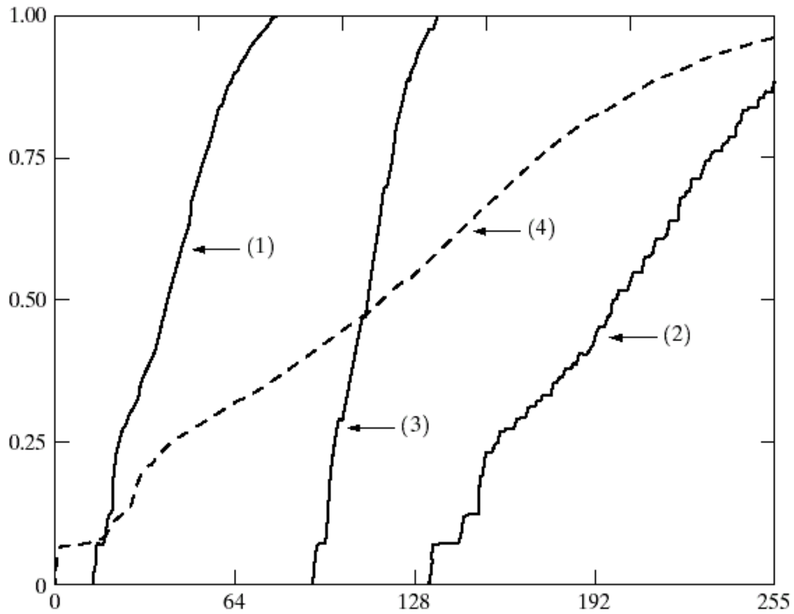


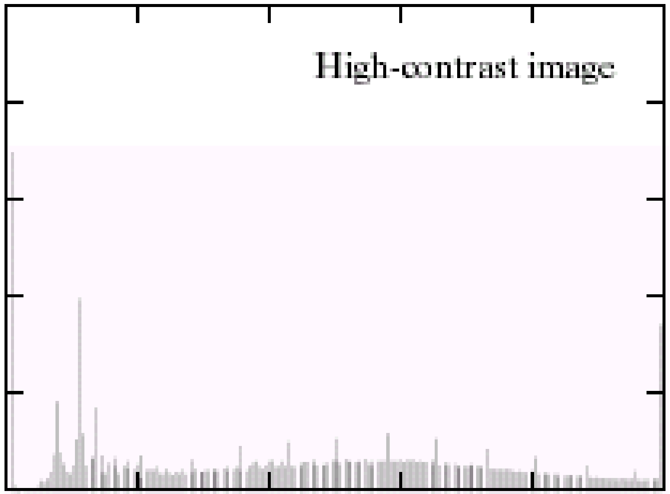


Example : Histogram Equalization Results

4







Thanks for your attention