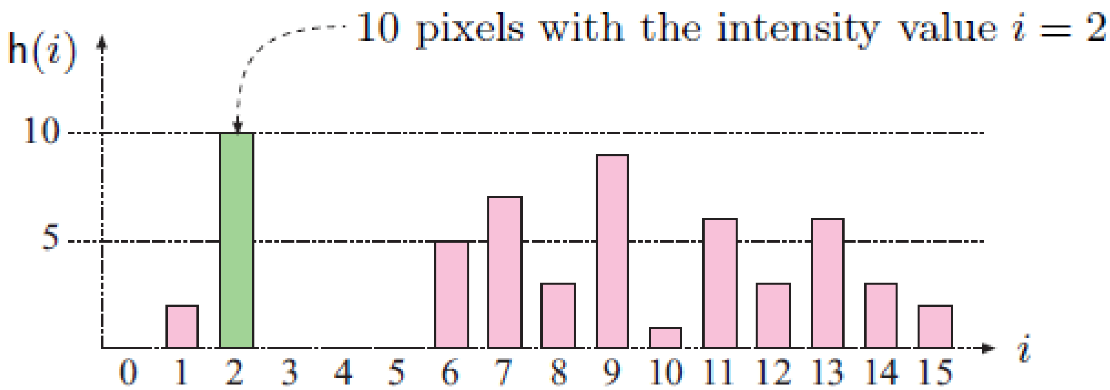
Histogram Processing

- 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

Example



$h(i)$	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- Histogram vector for an image with ***L*** **= 16** possible intensity values
- The **indices** of the vector element *i* = 0. . . 15 represent **intensity values**
- The value of 10 at index 2 means that the image contains 10 pixels of intensity value 2.

Normalized Histogram

- Dividing each of histogram value at gray level $\boldsymbol{r_k}$ by the total number of pixels in the image, ***MN***

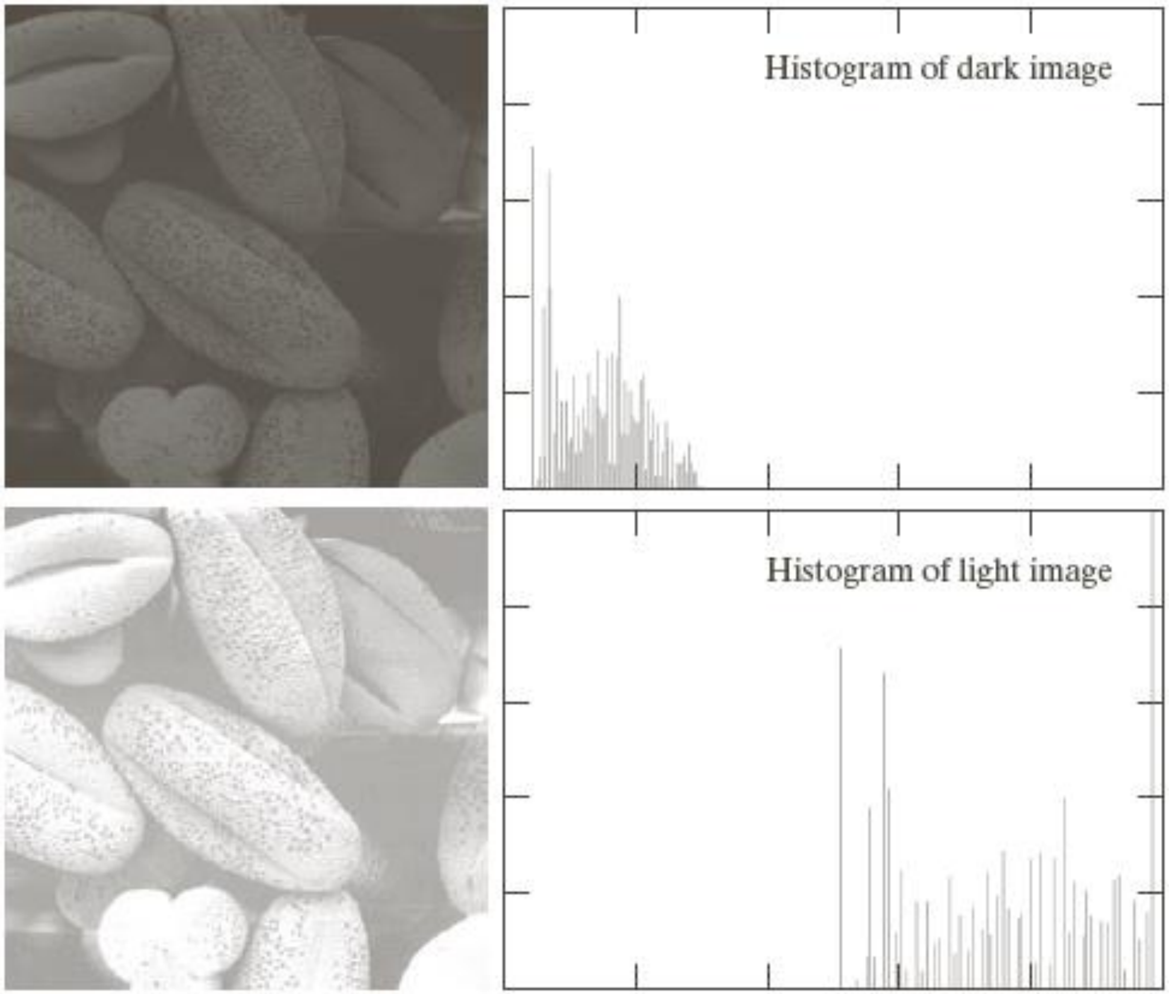
$$\boldsymbol{p(r_k) = n_k / MN}$$

  - For k = 0,1,…,L-1
  - $\boldsymbol{p(r_k)}$ gives an estimate of the probability of occurrence of gray level $\boldsymbol{r_k}$
  - The sum of all components of a normalized histogram is equal to 1
  - **M** and **N** are the row and column dimensions of the image.

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**
- Data-dependent pixel-based image enhancement method.

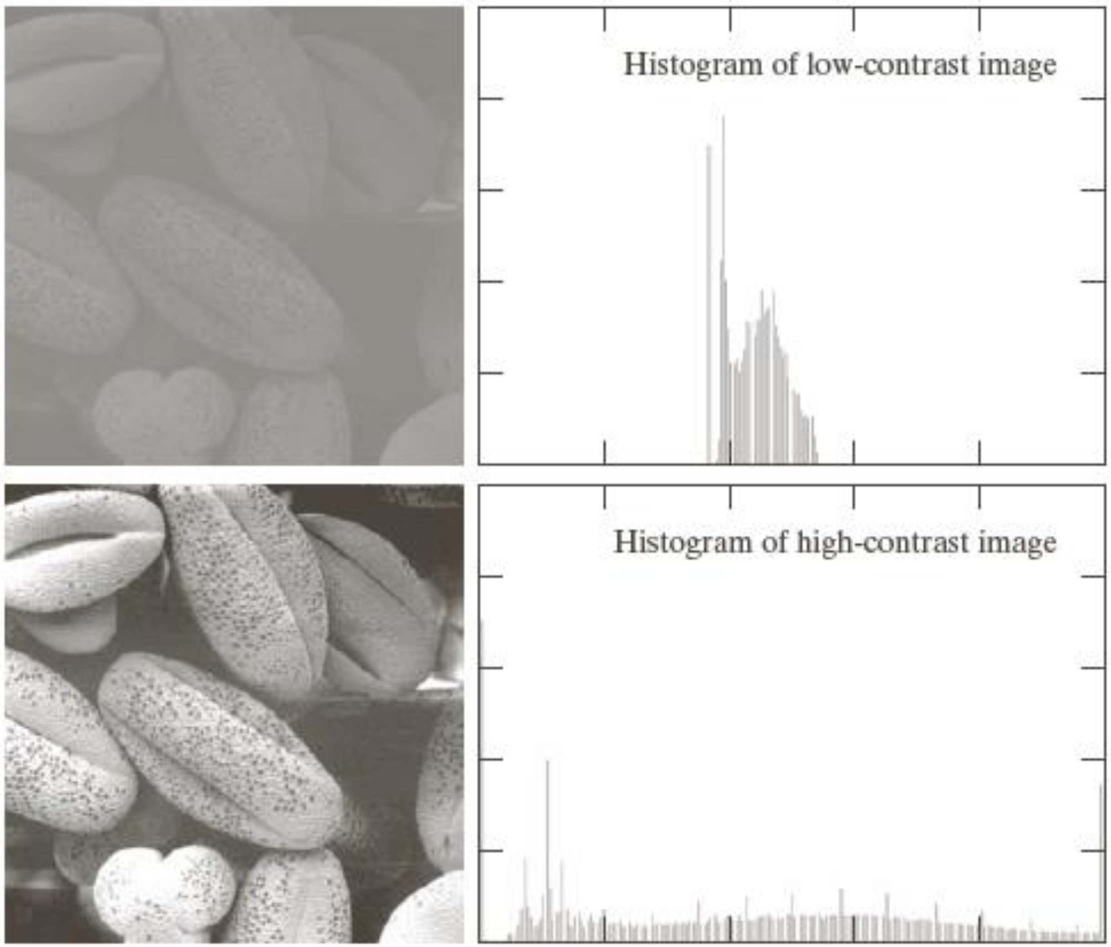
Example



- **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

Images are from the Digital Image Processing textbook, see slide 7

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 (PDF)** of the original histogram of the image so that the **probability spreads equally**

Discrete intensities and histogram equalization

- For discrete values, we deal with probabilities (histogram values) instead of PDF. So $p_r(r_k)=n_k/MN$, k=0,1,2,…,L-1
- Summations instead of integrals
- So $s_k=T(r_k)=(L\text{-}1)\sum_{j=0}^{k} p_r(rj) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$
- Thus, a processed (output) image is obtained by mapping each pixel in the **input image with intensity r_k into a corresponding pixel with level s_k in the output image**, using the above equation.

Implementation

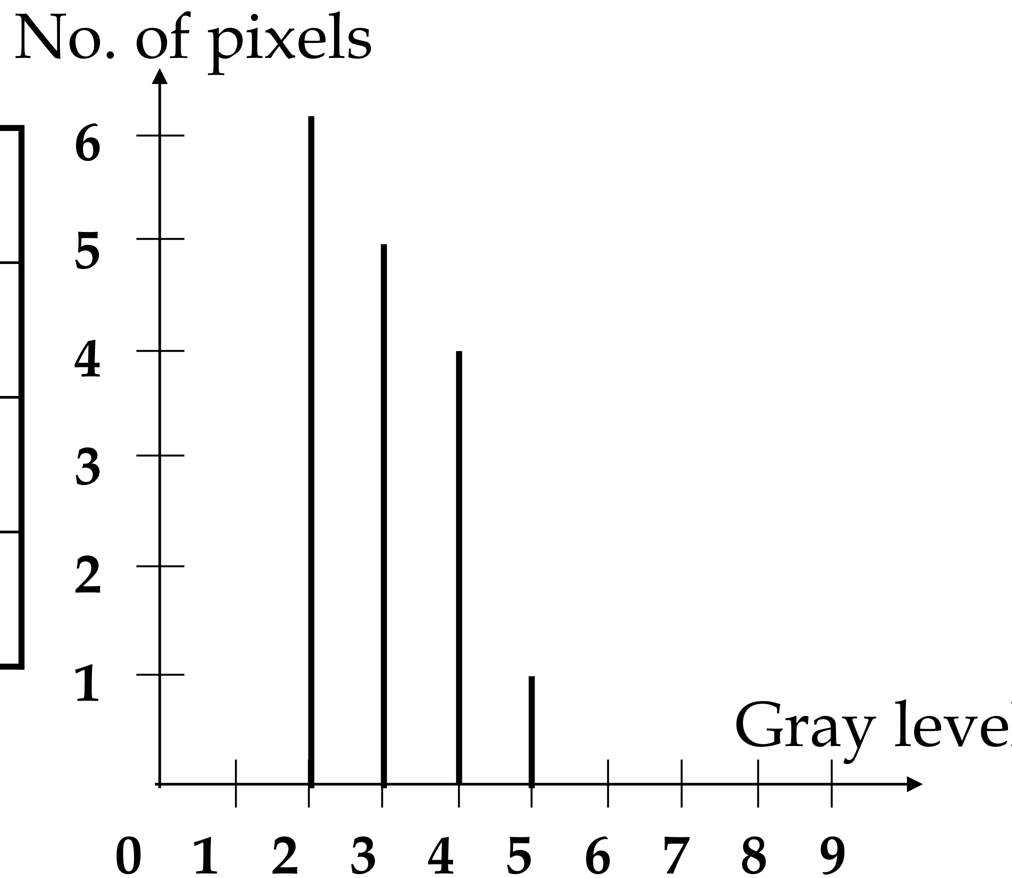
1. Obtain the histogram of the input image.
2. For each input gray level k, compute the cumulative sum.
3. For each gray level k, scale the sum by (max gray level)/(number of pixels).
4. Discretize the result obtained in 3.
5. Replace each gray level k in the input image by the corresponding level obtained in 4.

Example

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

4x4 image

Gray scale = [0,9]



histogram

Example

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

Example

Gray Level(k)	0	1	2	3	4	5	6	7	8	9
No. of pixels (n_k)	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$										
$s = \sum_{j=0}^{k} \frac{n_j}{n}$										
s x 9										

Example

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

Example

Gray Level(k)	0	1	2	3	4	5	6	7	8	9
No. of pixels (n_k)	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	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
s x 9										

Example

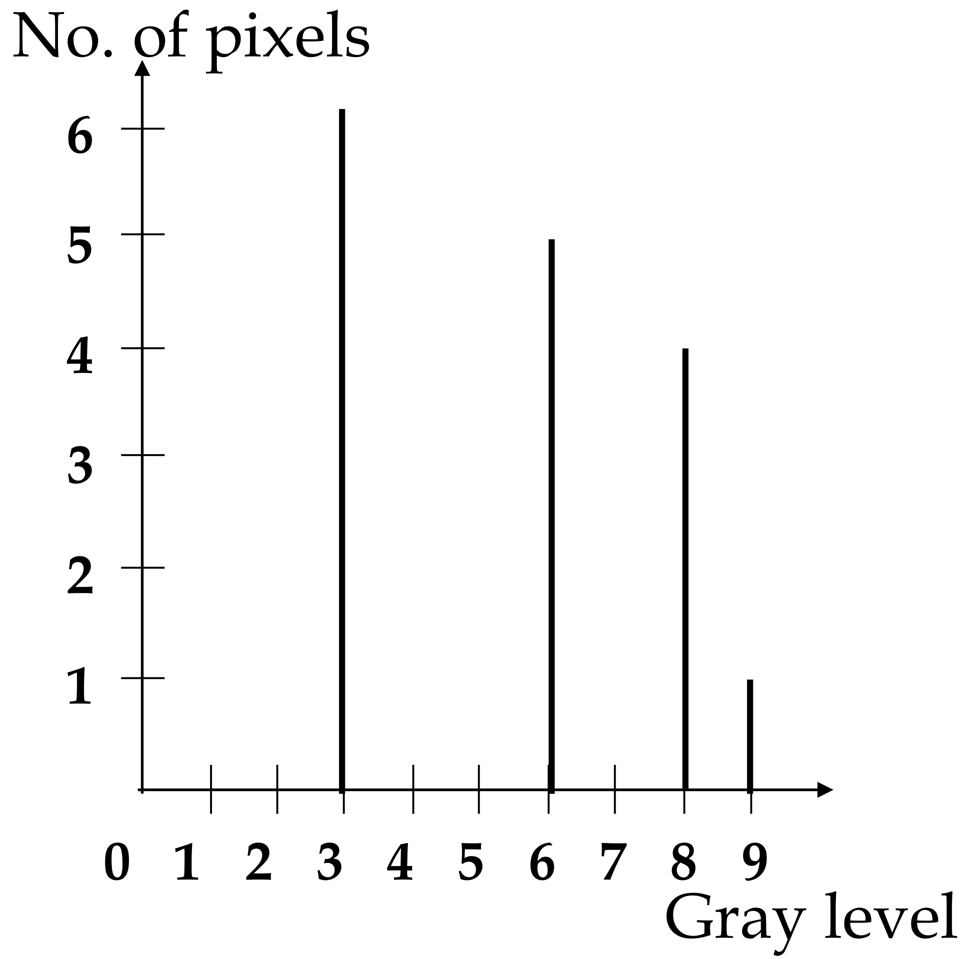
Gray Level(k)	0	1	2	3	4	5	6	7	8	9
No. of pixels (n_k)	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
s x 9	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

Example

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

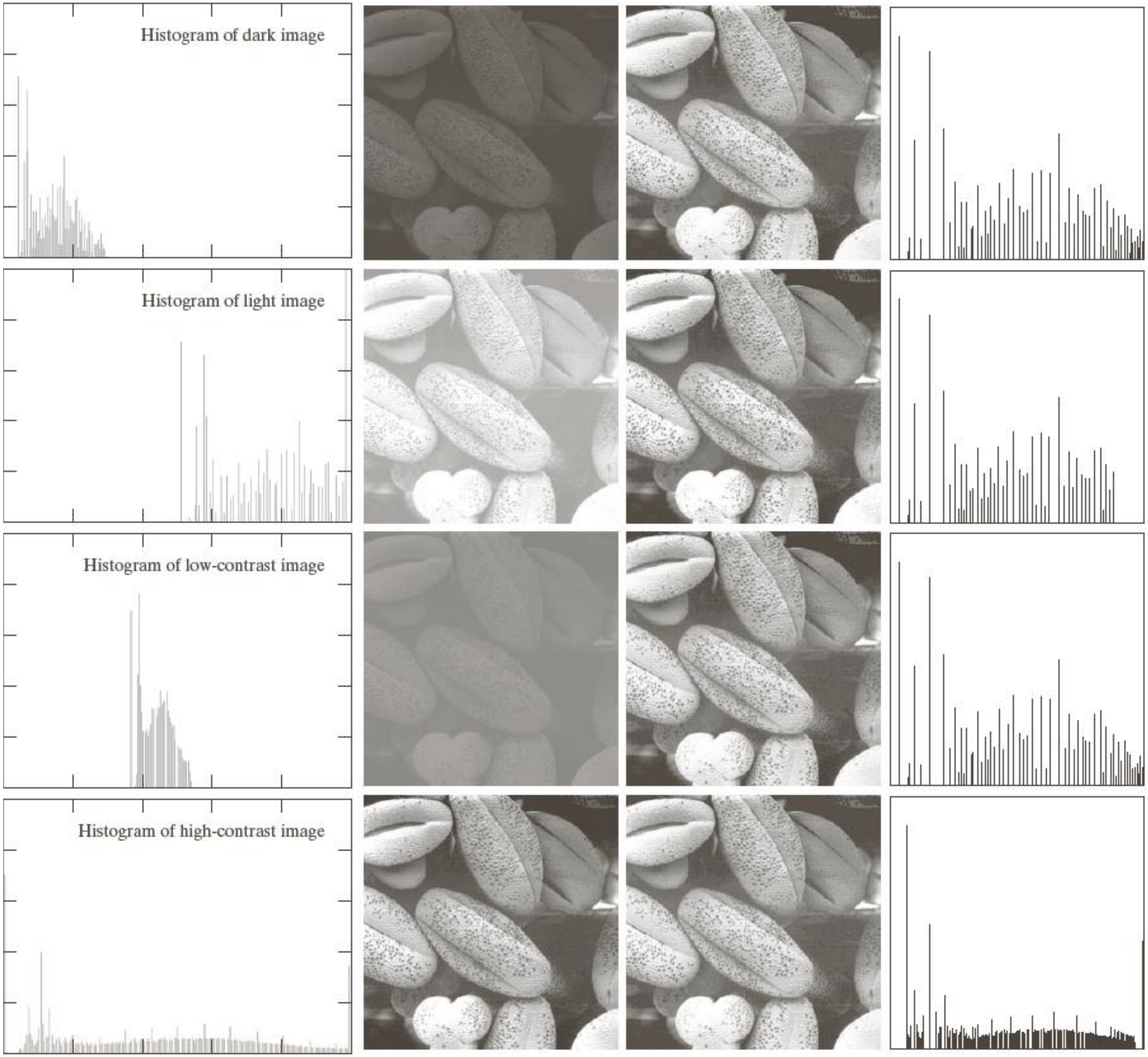
Output image

Gray scale = [0,9]



Histogram equalization

Histogram equalization Example



- First column from left is histogram for each picture in the secod column.
- Third column for left is the images after applying histogram equalization to each correspondent image in the second column
- Fourth column is shows histogram of each image in the third column

Properties of Histogram Equalization

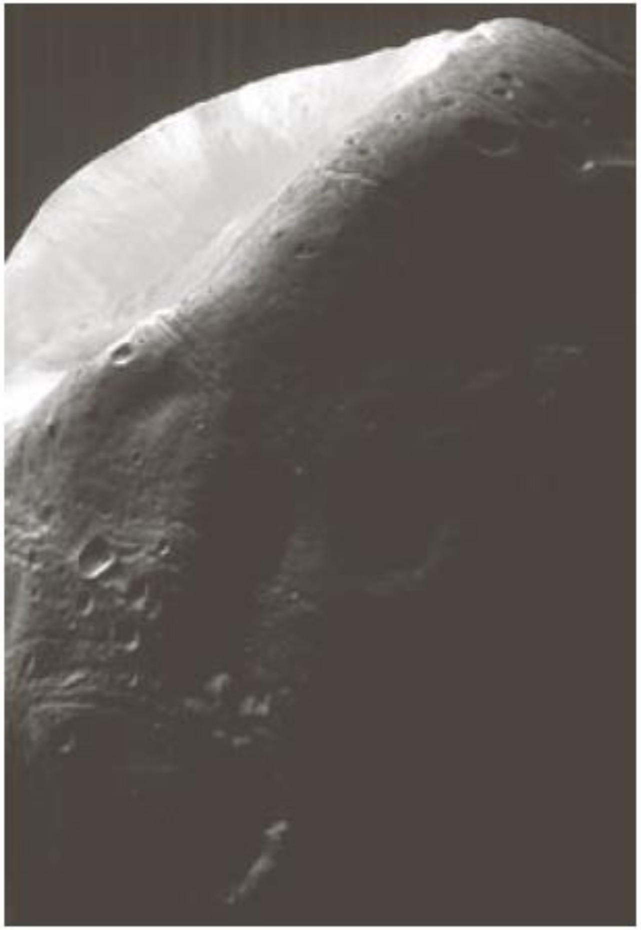
- The gray levels are spread over the entire intensity reange
  - Fully automatic
  - Data dependent
  - contrast enhancement

- Perfectly flat histograms are rare in practical applications of histogram equalization.

Histogram Matching (Specification)

- Histogram equalization has a disadvantage which is, it can generate only **one type of output image**.
- This is **not always disired**, and **may not work** for some applications
- It doesn't have to be a uniform histogram
- It is useful sometimes to be able to specify the **shape of the histogram that we wish the processed image to have**.
- To generate a processed image that has a specified histogram is called *histogram matching* or *histogram specification.*

Example



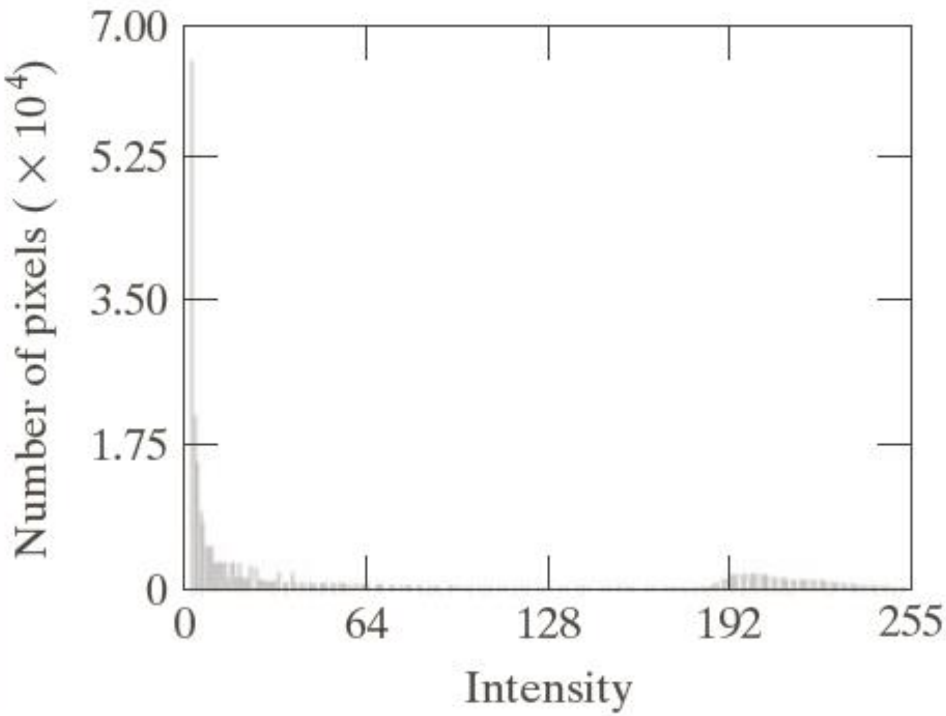
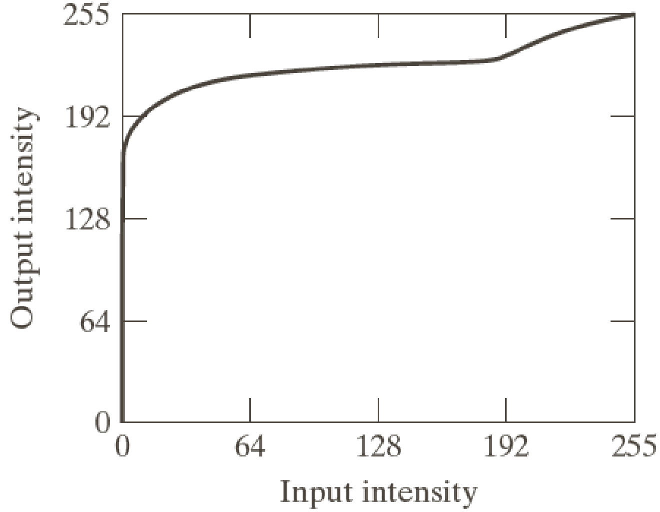
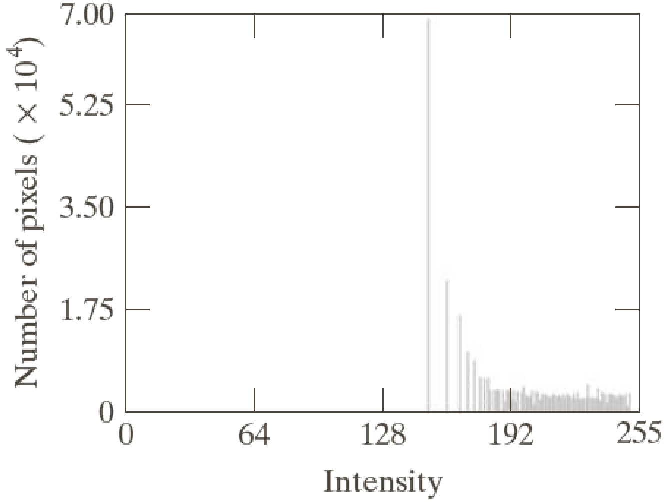


Image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in pixels in the dark end of the gray scale

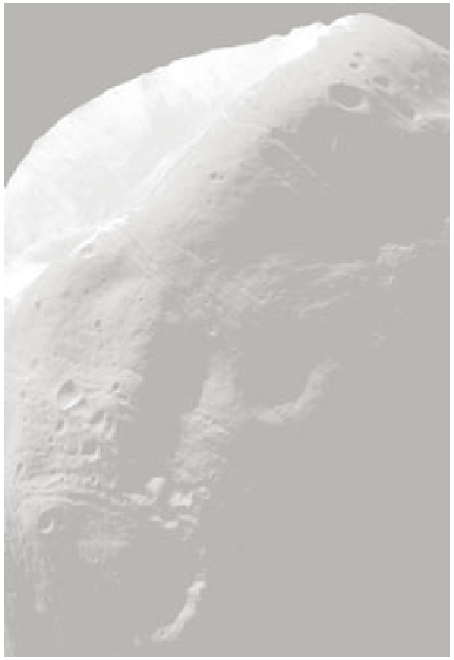
Histogram Equalization

Transformation function for histogram equalization





Transformation function for histogram equalization



Result image after histogram equalization

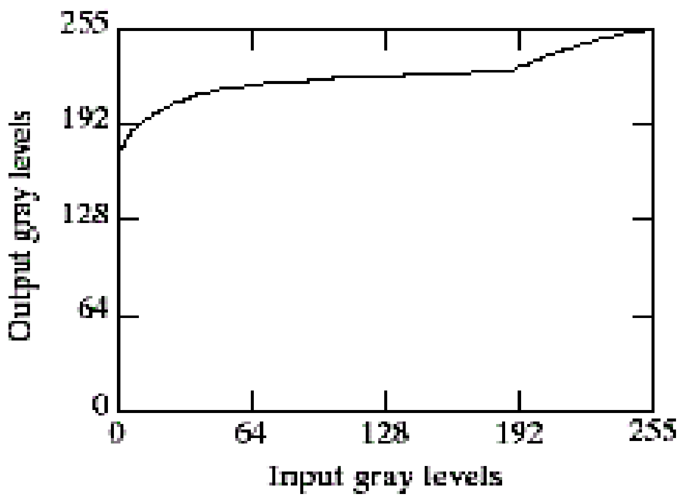
- The histogram equalization doesn't make the result image look better than the original image.
- Consider the histogram of the result image, the net effect of this method is to map a very narrow interval of dark pixels into the upper end of the gray scale of the output image.
- As a consequence, the output image is light and has a washed-out appearance.

Solve the Problem

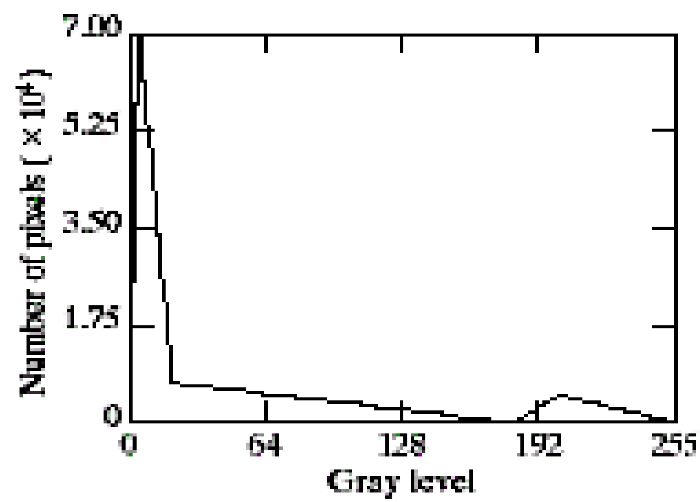
- Since the problem with the transformation function of the histogram equalization was caused by a **large concentration of pixels in the original image with levels near 0**

- A reasonable approach is to modify the histogram of that image so that it does not have this property

Histogram Equalization



Histogram Specification



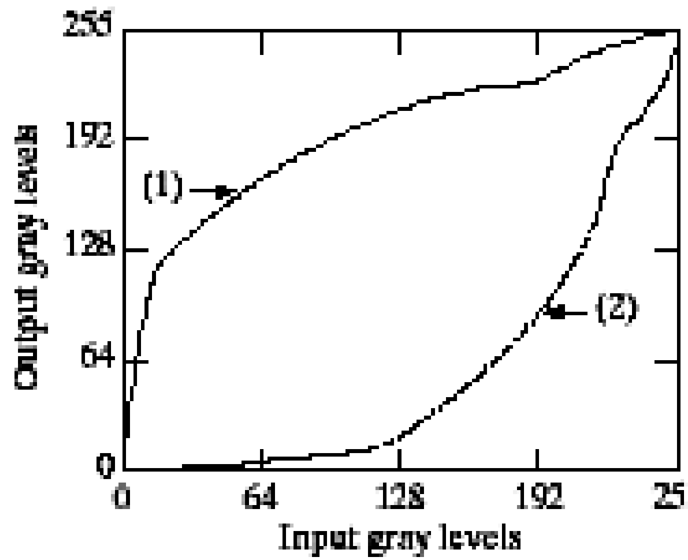
Histogram Specification

- (1) The transformation function G(z) obtained from

$$G(z_k) = \sum_{i=0}^{k} p_z(z_i) = s_k$$

$$k = 0,1,2,\dots, L-1$$

- (2) The inverse transformation $G^{-1}(s)$



Histogram Specification

(1) Obtain the transformation function $T(r)$ by calculating the histogram equalization of the input image

$$S = T(r)$$

(2) Obtain the transformation function $G(z)$ by calculating histogram equalization of the desired density function

$$v = G(z)$$

Histogram Specification

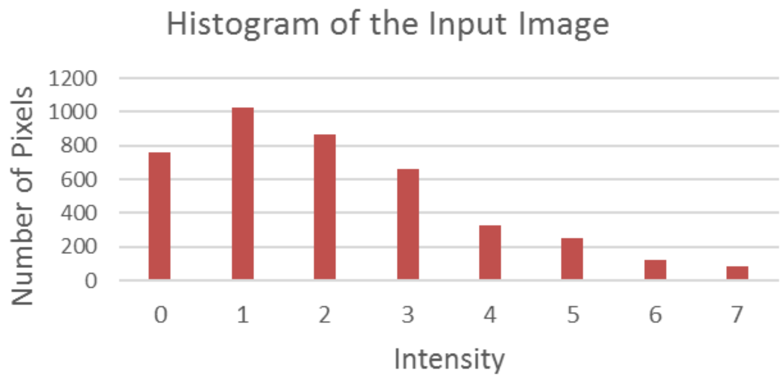
(3) Set v = s to obtain the inversed transformation function G^{-1}

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

(4) Obtain the output image by applying the processed gray-level from the inversed transformation function to all the pixels in the input image

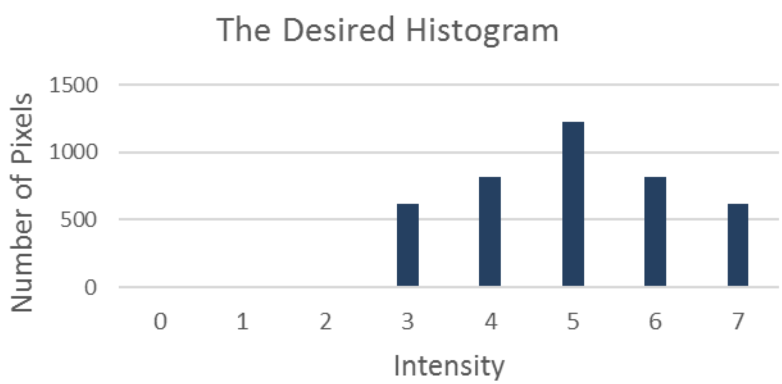
Histogram Specification - Example

- Given a histogram for 8-level image



0	**760**
1	**1023**
2	**870**
3	**660**
4	**331**
5	**249**
6	**122**
7	**81**

- Match it to this histogram



0	**0**
1	**0**
2	**0**
3	**615**
4	**819**
5	**1229**
6	**819**
7	**614**

Histogram Specification - Example

(1) Equalize the histogram of the input image using transform $s = T(r)$

r	n_r	$\sum$	s	$s*7$	$s=T(r)$
0	760	760	0.185547	1.298828	1
1	1023	1783	0.435303	3.047119	3
2	870	2653	0.647705	4.533936	5
3	660	3313	0.808838	5.661865	6
4	331	3644	0.889648	6.227539	6
5	249	3893	0.950439	6.653076	7
6	122	4015	0.980225	6.861572	7
7	81	4096	1	7	7

Histogram Specification - Example

(2) Equalize the desired histogram $v = G(z)$

z	n_z	$\sum$	v	$v*7$	$v=T(z)$
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	615	615	0.150146	1.051025	1
4	819	1434	0.350098	2.450684	2
5	1229	2663	0.650146	4.551025	5
6	819	3482	0.850098	5.950684	6
7	614	4096	1	7	7

Histogram Specification - Example

- (3) Set $v = s$ to obtain the composite transform $z = G^{-1}(s) = G^{-1}[T(r)]$

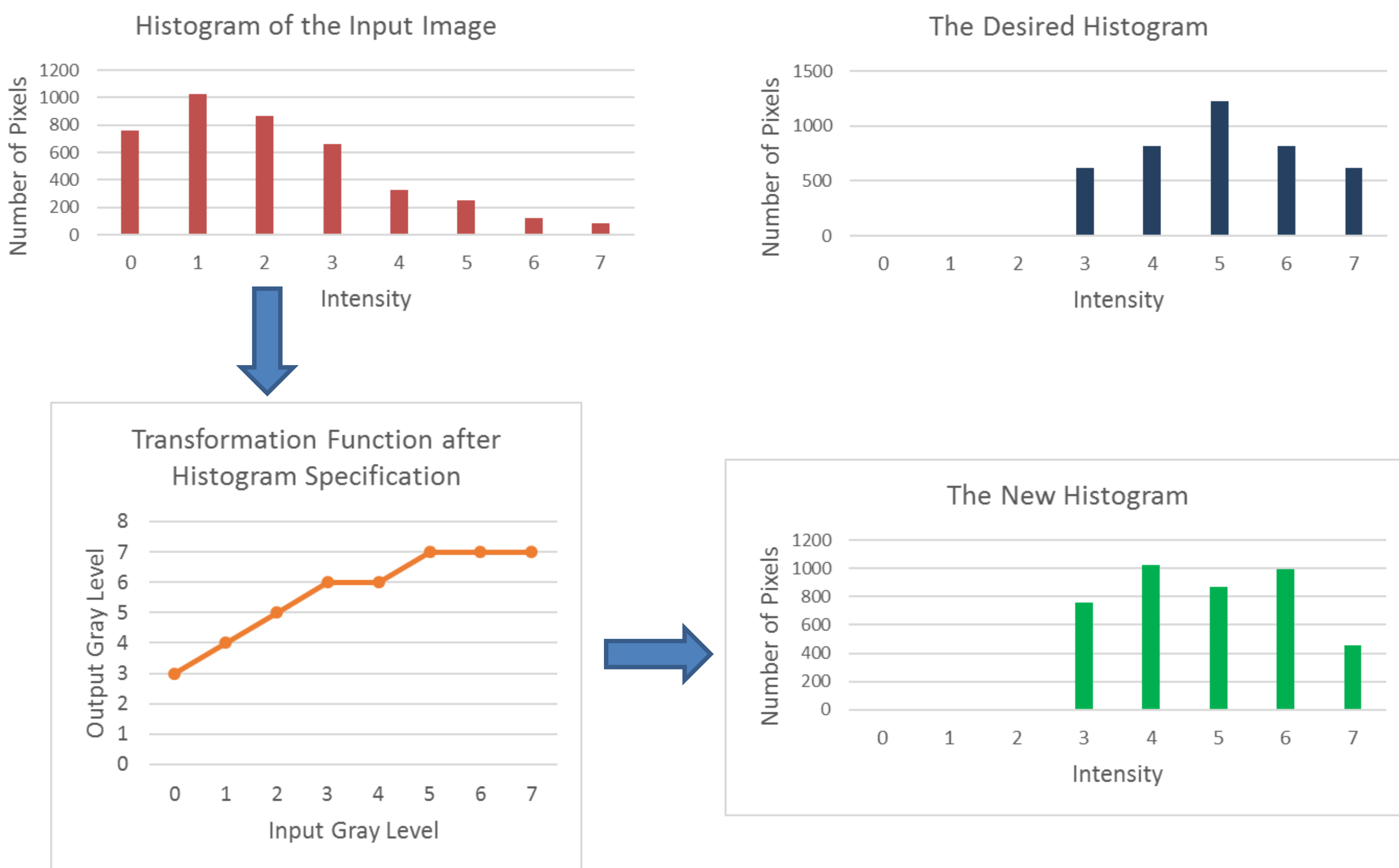
r	$s=T(r)$	$z = G^{-1}[T(r)]$
0	1	3
1	3	4
2	5	5
3	6	6
4	6	6
5	7	7
6	7	7
7	7	7

Histogram Specification - Example

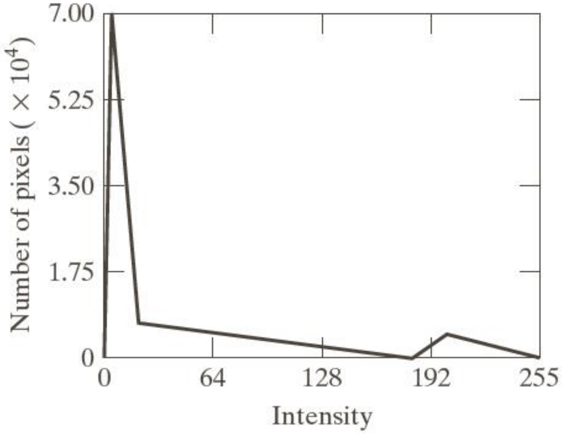
- (4) Obtain the output image (after histogram specification) according to the following table:

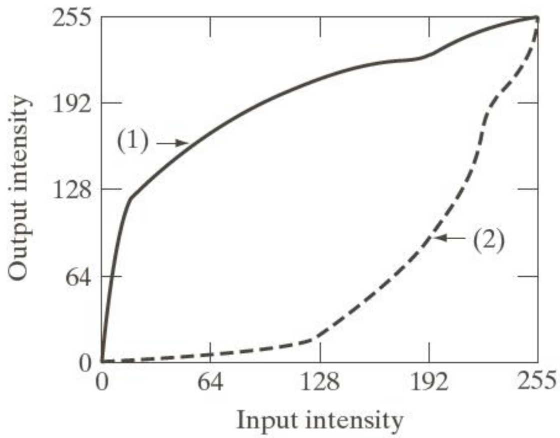
Input gray level	Output gray level
0	3
1	4
2	5
3	6
4	6
5	7
6	7
7	7

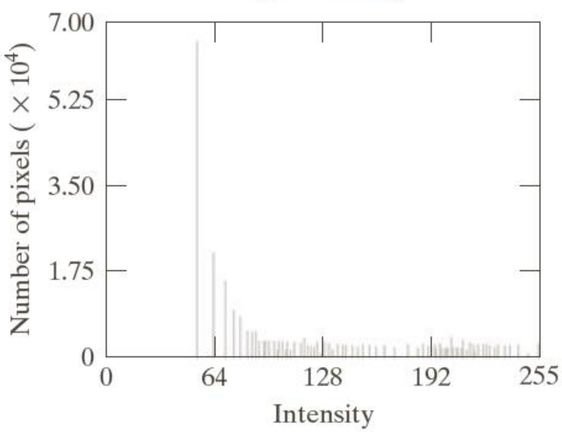
Histogram Specification - Summary



Result Image and its Histogram







After histogram Specification

Result image after histogram equalization

Original image

Notice that the output histogram's low end has shifted right toward the lighter region of the gray scale as desired.

Note

- Histogram specification is a **trial-and-error process**

- There are **no rules** for specifying histograms, and one must resort to analysis on a **case-by-case** basis for any given enhancement task.

Note

- The previously 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

- The procedure is to define a neighborhood and move its center from pixel to pixel.
- At each location, the histogram of the points in the neighborhood is computed and either a histogram equalization or histogram specification transformation function is obtained.
- The movement overlapping produce no problem. The nonoverlapping movement, however, produces an undesirable „blocky" effect

Local Enhancement

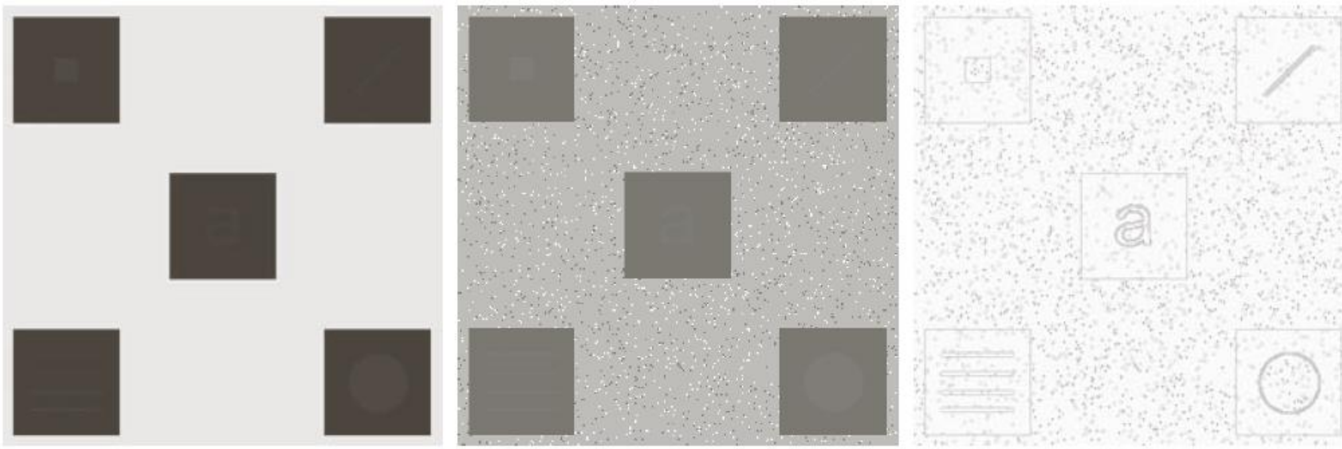


FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3 × 3.

a) Original image (slightly blurred to reduce noise)
b) global histogram equalization (enhance noise & slightly increase contrast but the construction is not changed)
c) local histogram equalization using 3x3 neighborhood (reveals the small squares inside larger ones of the original image.

Explanation of the result in c)

- Basically, the original image consists of many small objects inside the larger dark squares.
- However, the small objects were too close in gray level to the larger squares, and their sizes were too small to influence global histogram equalization significantly.
- So, when we use the local enhancement technique, it reveals the small areas.
- Note also the finer noise texture is resulted by the local processing using relatively small neighborhoods.

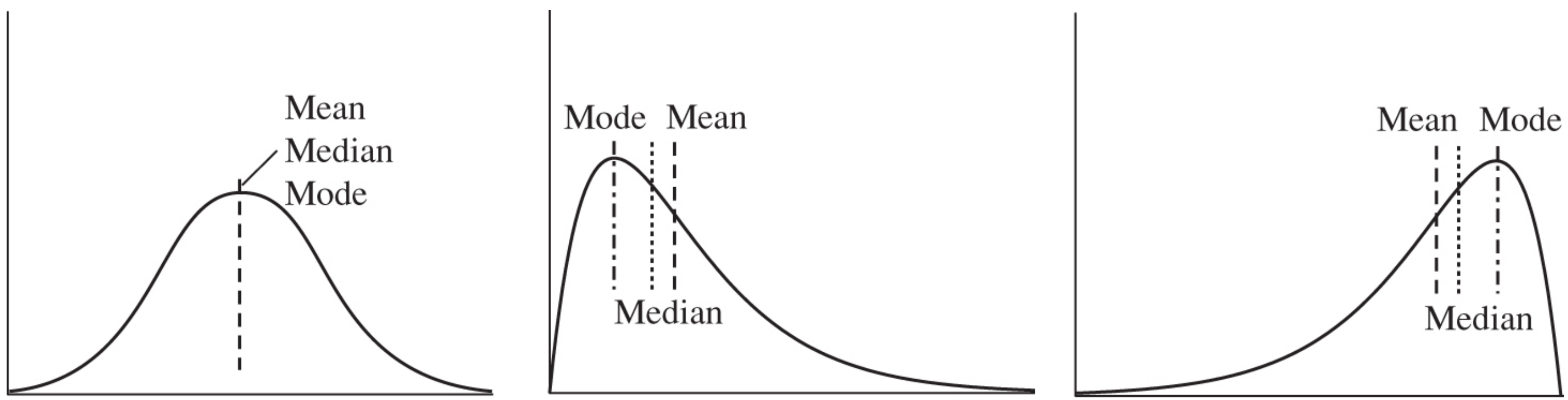
Using Histogram Statistics for Image Enhancement

- Mean and variance can be calculated using the n^{th}-momentum equation using histogram, but it also can be caluculated using the following equations:

- $m = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$
- $\sigma^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y) - m]^2$

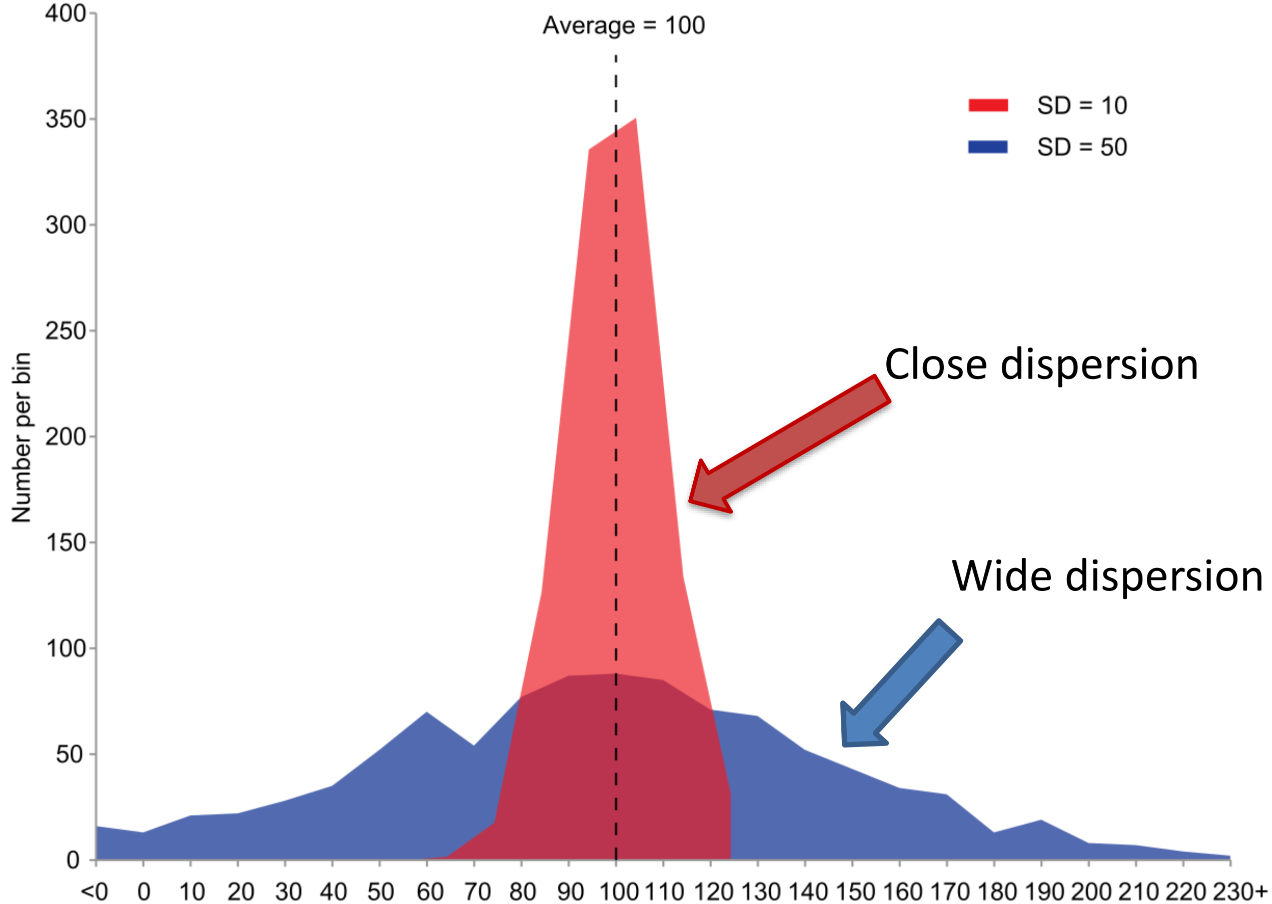
Symmetric vs. Skewed Data

- Median, mean and mode of symmetric, positively and negatively skewed data



(a) Symmetric data **(b)** Positively skewed data **(c)** Negatively skewed data

Measuring the Dispersion of Data



Images are from the Digital Image Processing textbook, see slide 7