Histogram Processing

What is a Histogram?

- In Statistics, **Histogram** is a graphical representation showing a visual impression of the distribution of data.

- An **Image Histogram** is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image. It plots the number of pixels for each value.

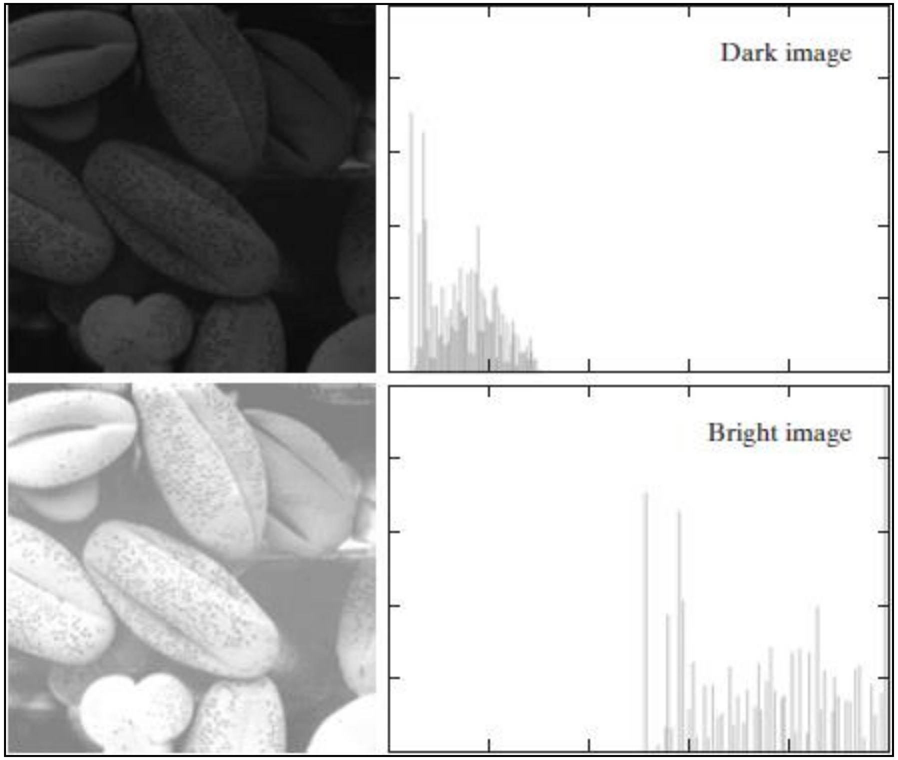
Why Histogram?

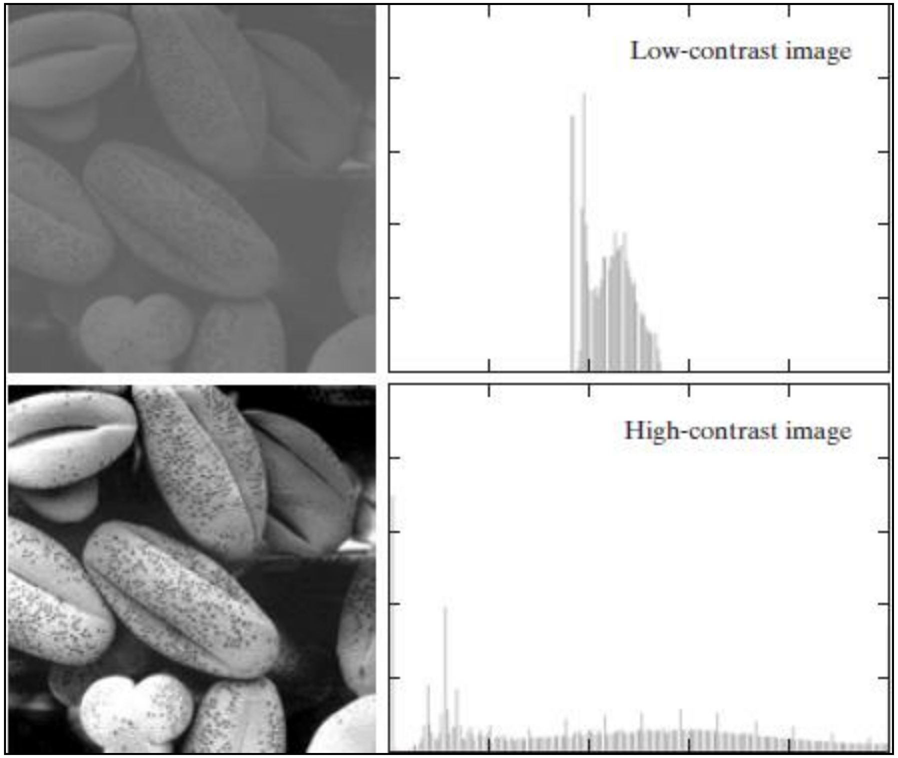
- Histograms are the basis for numerous spatial domain processing techniques
- Histogram manipulation can be used effectively for image enhancement
- Histograms can be used to provide useful image statistics
- Information derived from histograms are quite useful in other image processing applications, such as image compression and segmentation.

Introductory Example of Histograms

- As an introduction to the role of histogram processing in image enhancement, consider Fig. 3.15 shown in four basic gray-level characteristics: dark, light, low contrast, and high contrast.
- The right side of the figure shows the histograms corresponding to these images.
- The horizontal axis of each histogram plot corresponds to gray level values, r_k.
- The vertical axis corresponds to values of $h(r_k)=n_k$ or $p(r_k)=n_k/n$ if the values are normalized.
- Thus, as indicated previously, these histogram plots are simply plots of $h(r_k)=n_k$ versus r_k or $p(r_k)=n_k/n$ versus r_k.

Introductory Example of Histograms… Cont.





Histogram in MATLAB

h = imhist (f, b)

Where f, is the input image, h is the histogram, b is number of bins (tick marks) used in forming the histogram (b = 255 is the default)

A bin, is simply, a subdivision of the intensity scale. For example, if we are working with uint8 images and we let b = 2, then the intensity scale is subdivided into two ranges: 0 – 127 and 128 – 255. the resulting histograms will have two values: h(1) equals to the number of pixels in the image with values in the interval [0,127], and h(2) equal to the number of pixels with values in the interval [128 255].

Image Enhancement: Histogram Based Methods

What is the histogram of a digital image?

- The histogram of a digital image with gray values $r_0, r_1, \cdots, r_{L-1}$ is the discrete function

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

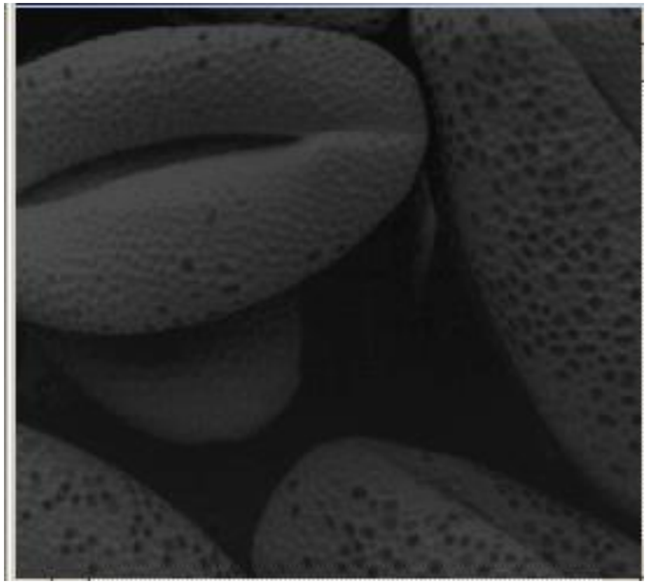
n_k: Number of pixels with gray value r_k

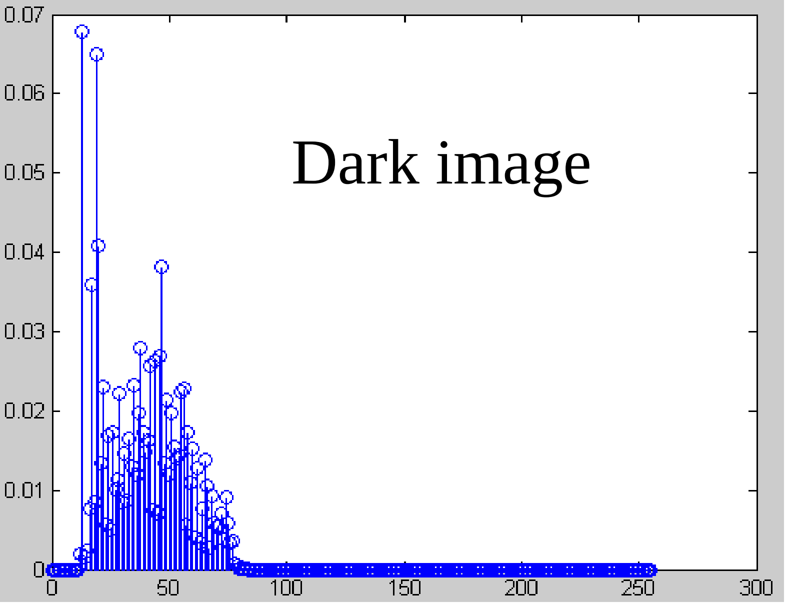
n: total Number of pixels in the image

- The function $p(r_k)$ represents the fraction of the total number of pixels with gray value r_k.

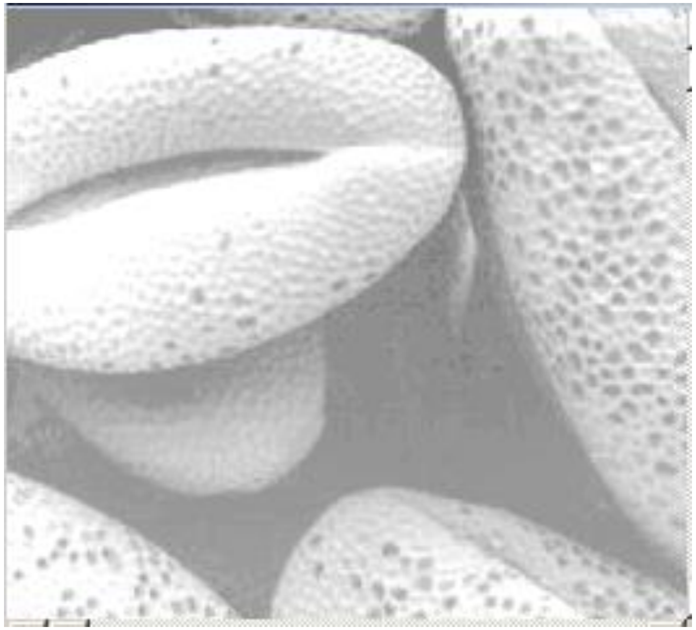
Some Typical Histograms

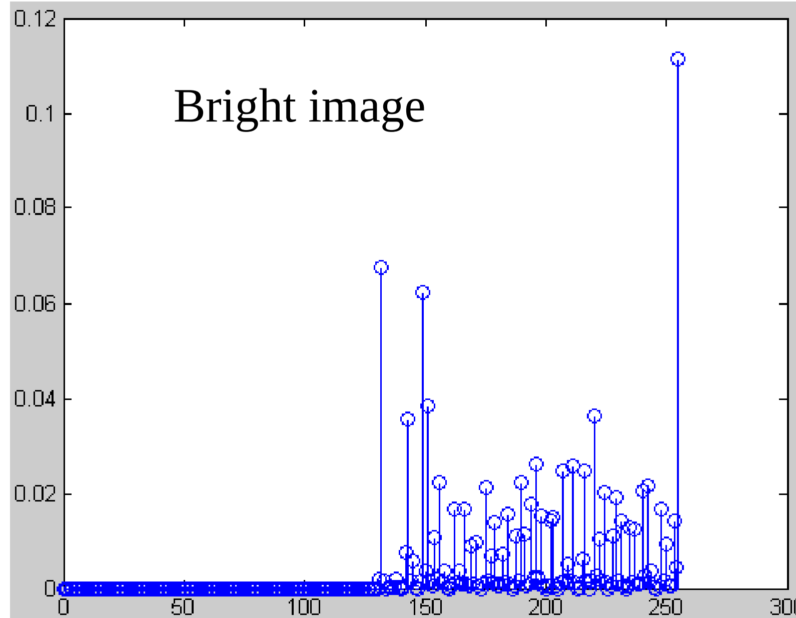
- The shape of a histogram provides useful information for contrast enhancement.

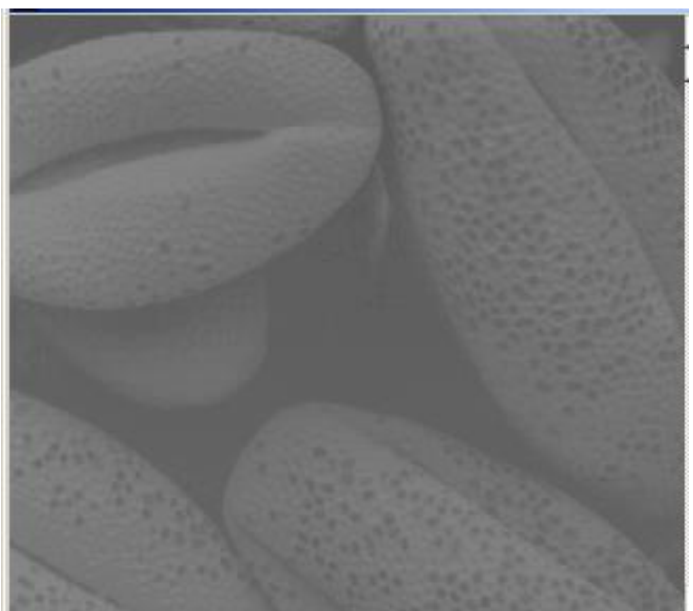


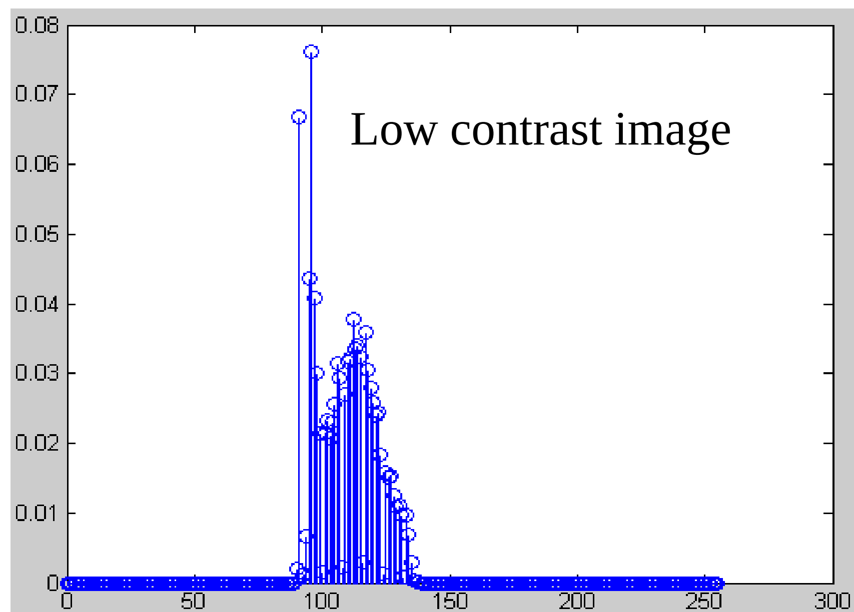


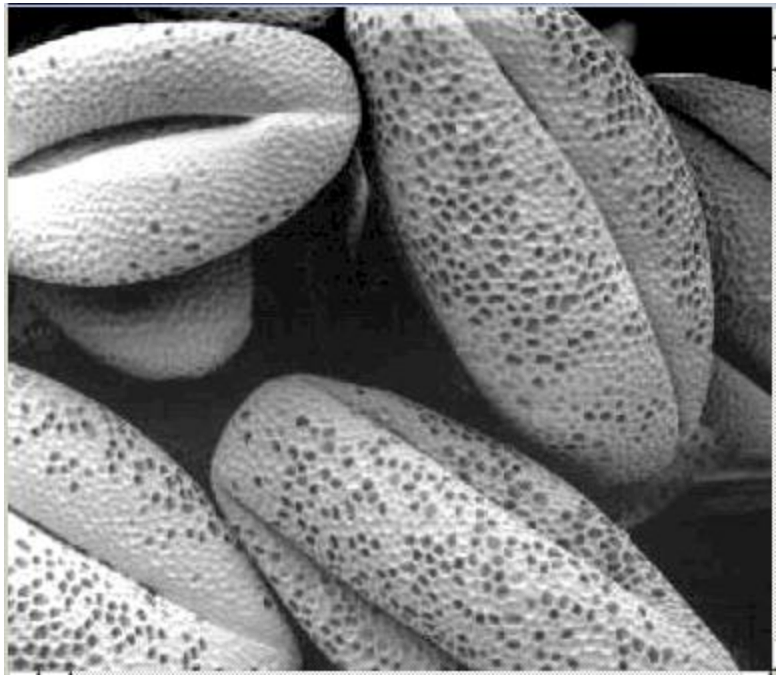

```
>> clear;
[ix,map]=imread('Fig3_15a.jpg');
imshow(ix)
figure;
ix=double(ix);
h=histogram(ix);
figure
stem(0:255,h);
```

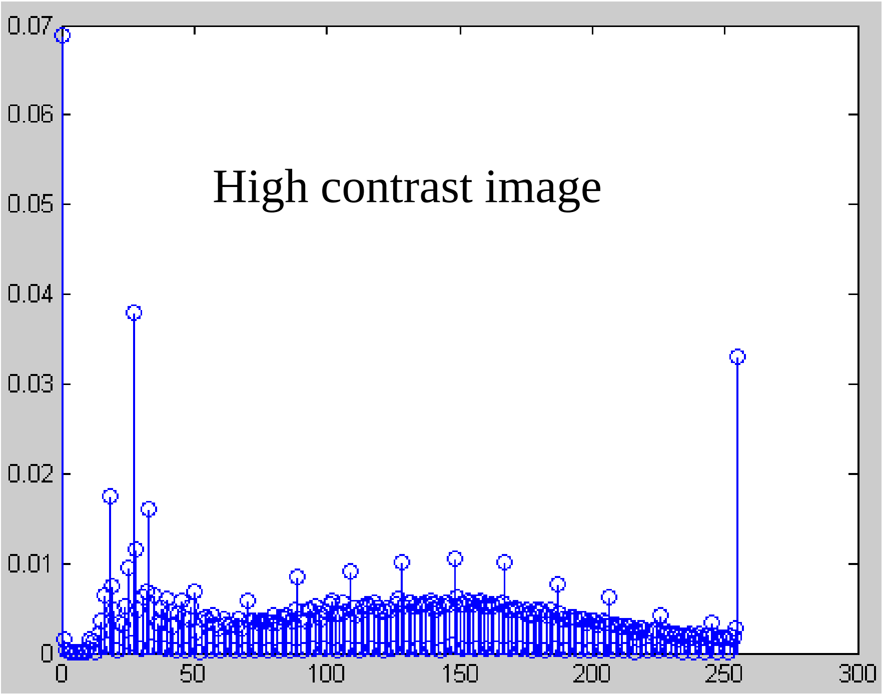



Bright image
0.12
0.1
0.08
0.06
0.04
0.02
0
0 50 100 150 200 250 300




Low contrast image
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 50 100 150 200 250 300





High contrast image
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
50
100
150
200
250
300

Histogram Equalization

- What is the histogram equalization?

- The histogram equalization is an approach to enhance a given image. The approach is to design a transformation *T(.)* such that the gray values in the output is uniformly distributed in [0, 1].

- Let us assume for the moment that the input image to be enhanced has continuous gray values, with $r = 0$ representing black and $r = 1$ representing white.

- We need to design a gray value transformation $s = T(r)$, based on the histogram of the input image, which will enhance the image.

How to implement histogram equalization?

Step 1:For images with discrete gray values, compute:

$$p_{in}(r_k) = \frac{n_k}{n} \qquad 0 \le r_k \le 1 \qquad 0 \le k \le L-1$$

L: Total number of gray levels

n_k: Number of pixels with gray value r_k

n: Total number of pixels in the image

Step 2: Based on CDF, compute the discrete version of the previous transformation :

$$s_k = T(r_k) = \sum_{j=0}^{k} p_{in}(r_j) \qquad 0 \le k \le L-1$$

Example:

- Consider an 8-level 64 x 64 image with gray values (0, 1, …, 7). The normalized gray values are (0, 1/7, 2/7, …, 1). The normalized histogram is given below:

k	r_k	n_k	$p(r_k) = n_k/n$
0	0	790	0.19
1	1/7	1023	0.25
2	2/7	850	0.21
3	3/7	656	0.16
4	4/7	329	0.08
5	5/7	245	0.06
6	6/7	122	0.03
7	1	81	0.02

NB: The gray values in output are also (0, 1/7, 2/7, …, 1).

Applying the transformation, $s_k = T(r_k) = \sum_{j=0}^{k} p_{in}(r_j)$ we have

$$s_0 = T(r_0) = \sum_{j=0} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) = 0.19 \rightarrow 1/7$$

$$s_1 = T(r_1) = \sum_{j=0}^{1} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) = 0.44 \rightarrow 3/7$$

$$s_2 = T(r_2) = \sum_{j=0}^{2} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + p_{\text{in}}(r_2) = 0.65 \rightarrow 5/7$$

$$s_3 = T(r_3) = \sum_{j=0}^{3} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_3) = 0.81 \rightarrow 6/7$$

$$s_4 = T(r_4) = \sum_{j=0}^{4} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_4) = 0.89 \rightarrow 6/7$$

$$s_5 = T(r_5) = \sum_{j=0}^{5} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_5) = 0.95 \rightarrow 1$$

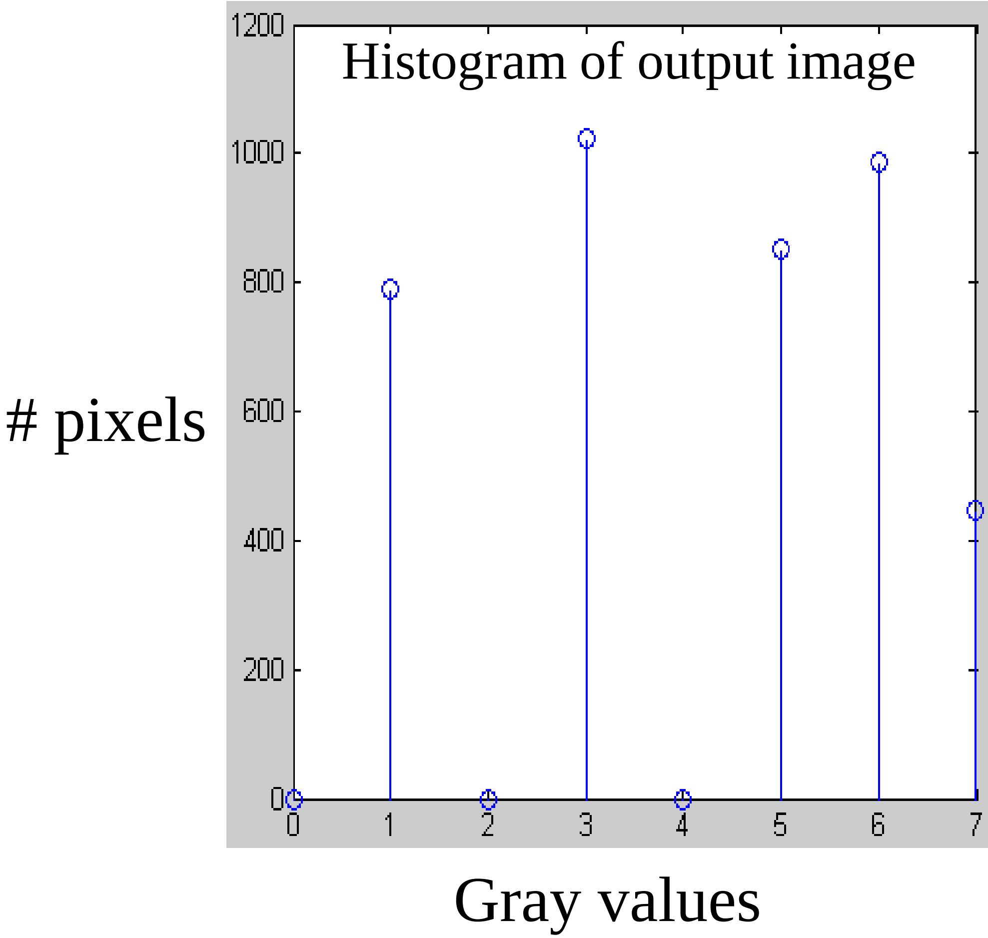
$$s_6 = T(r_6) = \sum_{j=0}^{6} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_6) = 0.98 \rightarrow 1$$

$$s_7 = T(r_7) = \sum_{j=0}^{7} p_{\text{in}}(r_j) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_7) = 1.00 \rightarrow 1$$

- Notice that there are only five distinct gray levels --- (1/7, 3/7, 5/7, 6/7, 1) in the output image. We will relabel them as (s_0, s_1, …, s_4).

- With this transformation, the output image will have histogram

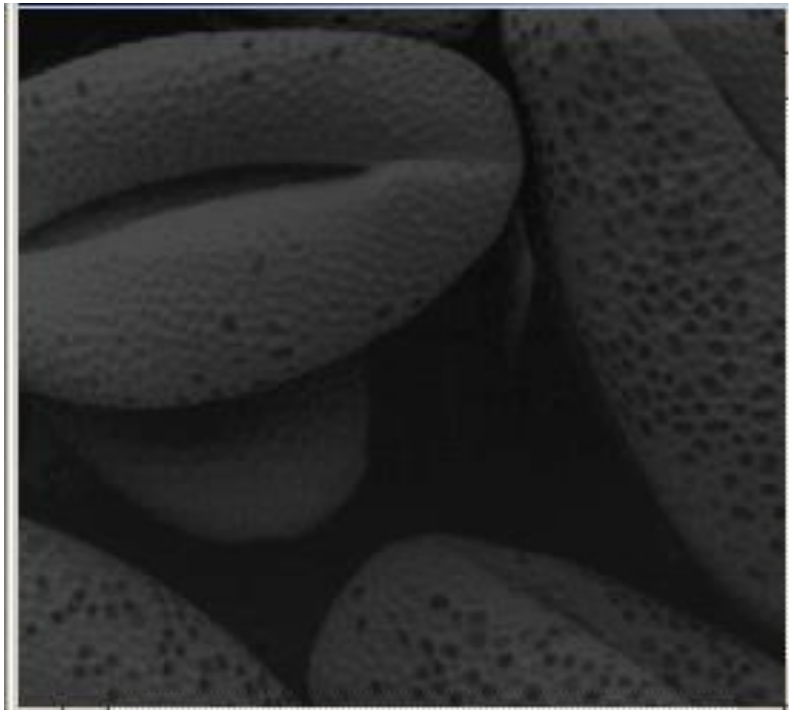
k	s_k	n_k	$p(s_k) = n_k/n$
0	1/7	790	0.19
1	3/7	1023	0.25
2	5/7	850	0.21
3	6/7	985	0.24
4	1	448	0.11

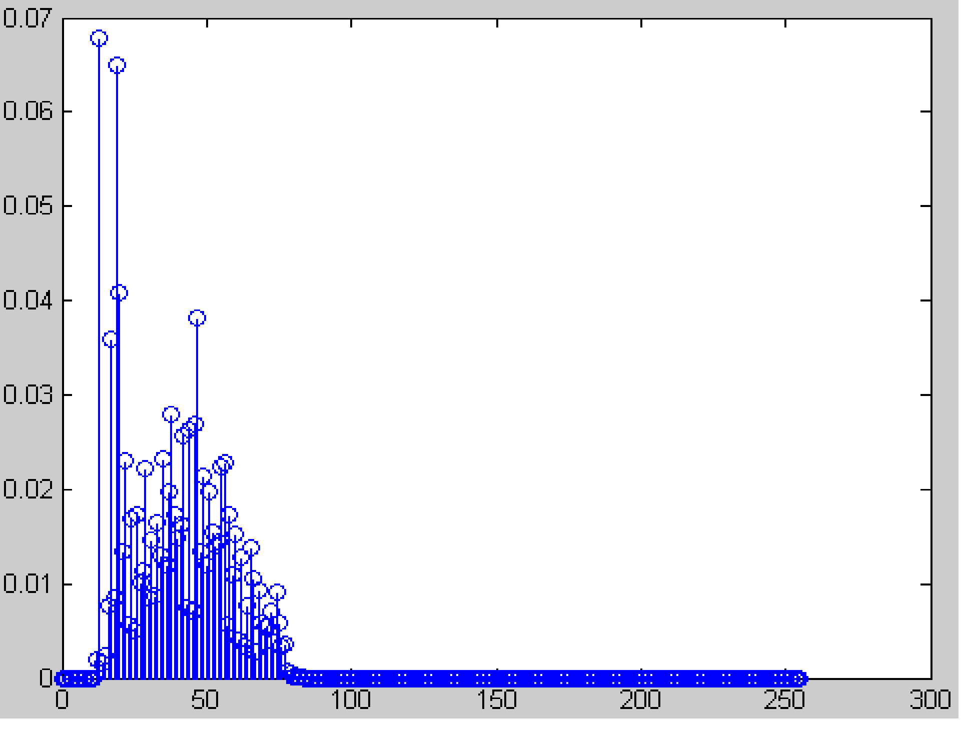



```
>> clear
>> h=[0 790 0 1023 0 850 985 448];
>> stem(0:7,h)
```

• Note that the histogram of output image is only approximately, and not exactly, uniform. This should not be surprising, since there is no result that claims uniformity in the **discrete** case.

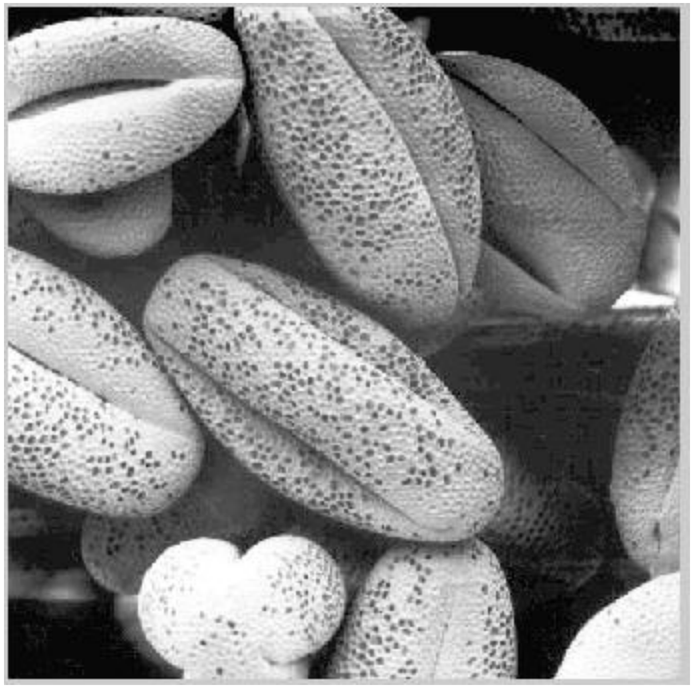
Example Original image and its histogram

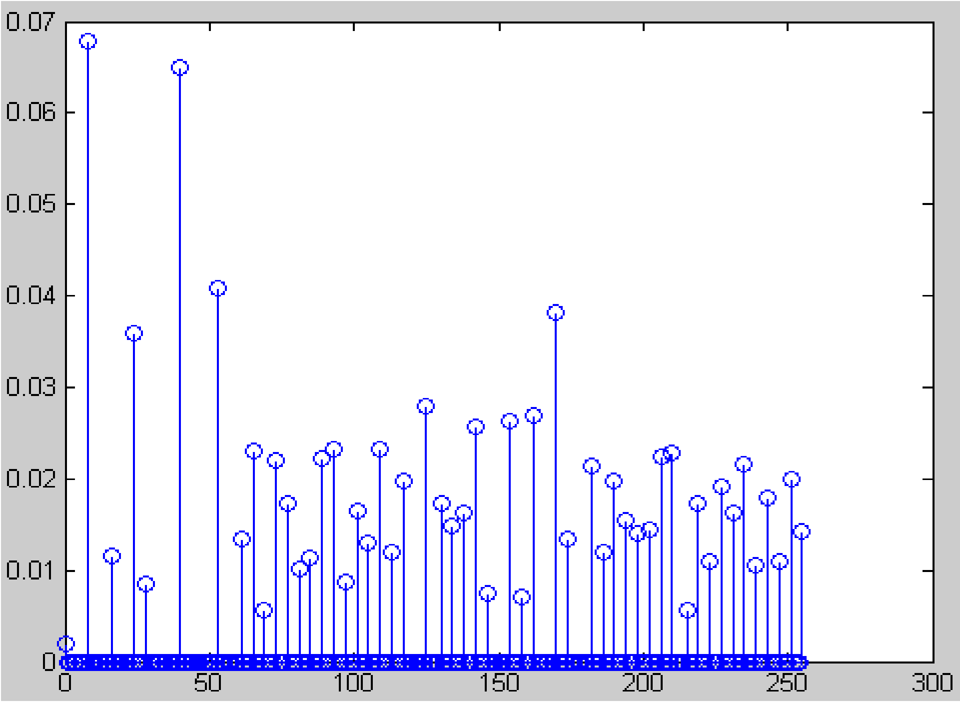





```
>> clear;
[ix,map]=imread('Fig3_15a.jpg');
imshow(ix)
figure;
ix=double(ix);
h=histogram(ix);
stem(0:255,h);
```

Histogram equalized image and its histogram



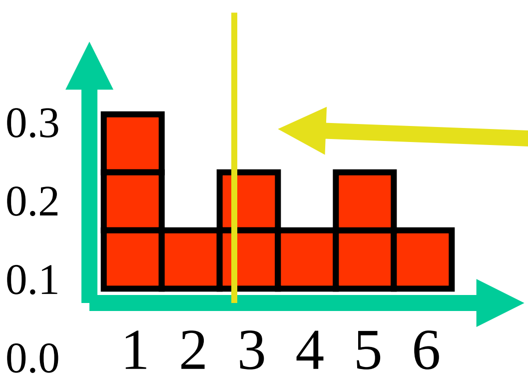



```
>> clear
>> [ix,map]=imread('Fig3_15a.jpg');
>> imshow(ix);
>> iy=histeq(ix);
>> figure
>> imshow(iy);
>> iy=double(iy);
>> hy=histogram(iy);
>> figure
>> stem(0:255,hy);
```

What can the (normalized) histogram tell about the image ?

The MEAN VALUE (or average gray level)

$$M = \sum_g g\, h(g)$$



1*0.3+2*0.1+3*0.2+4*0.1+5*0.2+6*0.1=

2.6

The MEAN value is the average gray value of the image, the 'overall brightness appearance'.

2. The VARIANCE

$$V = \sum_g (g-M)^2 \; h(g)$$

(with M = mean)

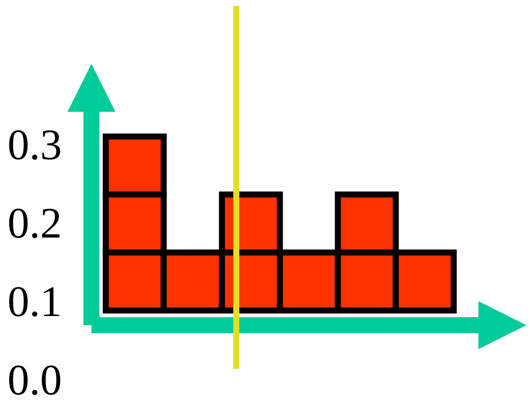
or similar:

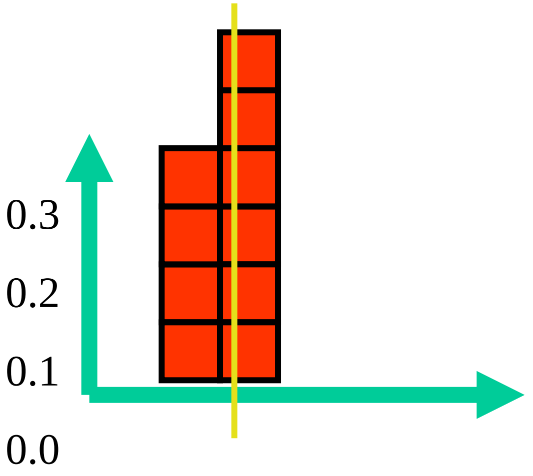
The STANDARD DEVIATION

$$D = \mathrm{sqrt}(V)$$

Histograms

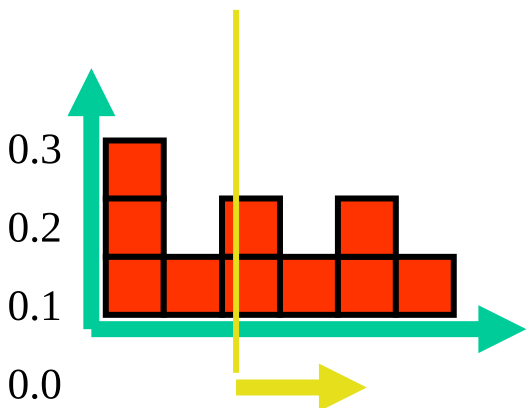
VARIANCE gives a measure about the distribution of the histogram values around the mean.

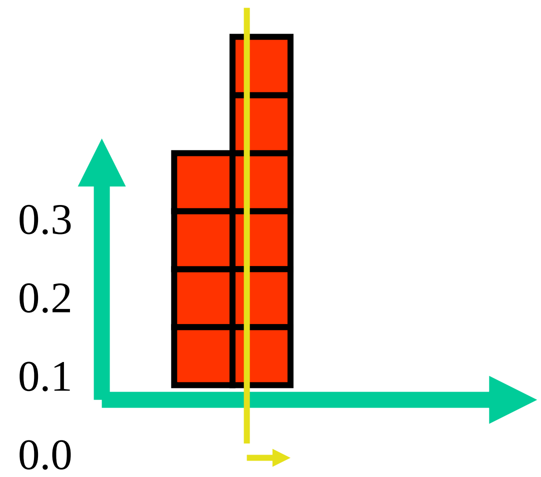




V1 > V2

The STANDARD DEVIATION is a value on the gray level axis, showing the average distance of all pixels to the mean





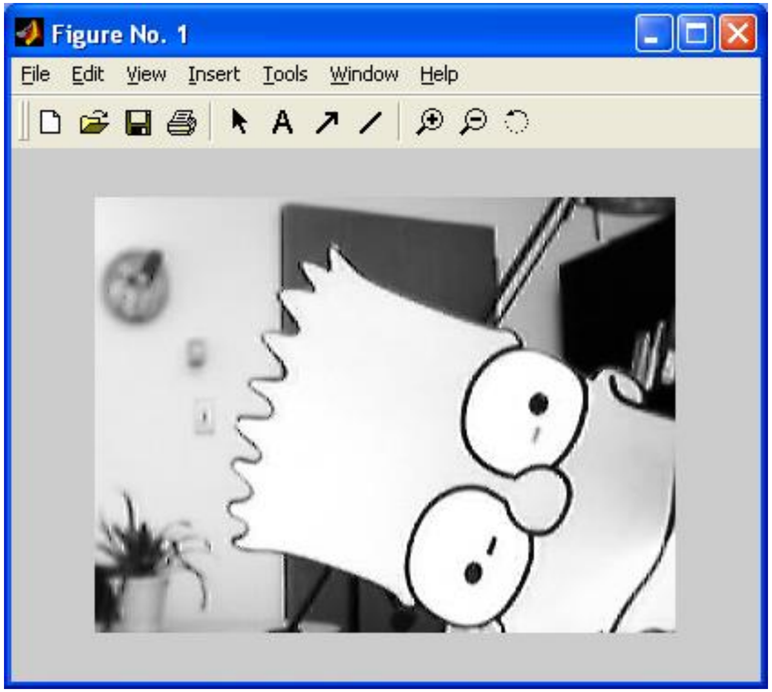
D1 > D2

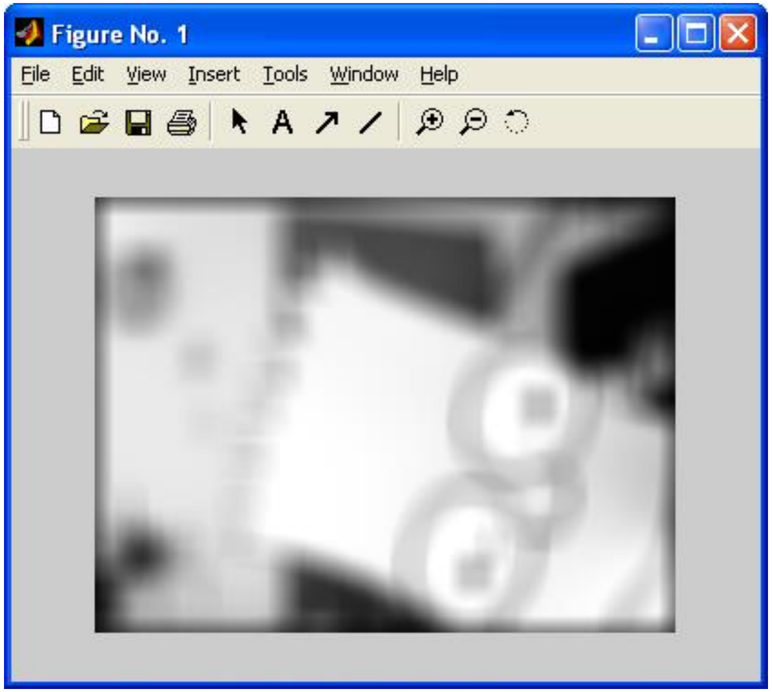
VARIANCE and STANDARD DEVIATION of the histogram tell us about the average contrast of the image !

The higher the VARIANCE (=the higher the STANDARD DEVIATION), the higher the image's contrast !

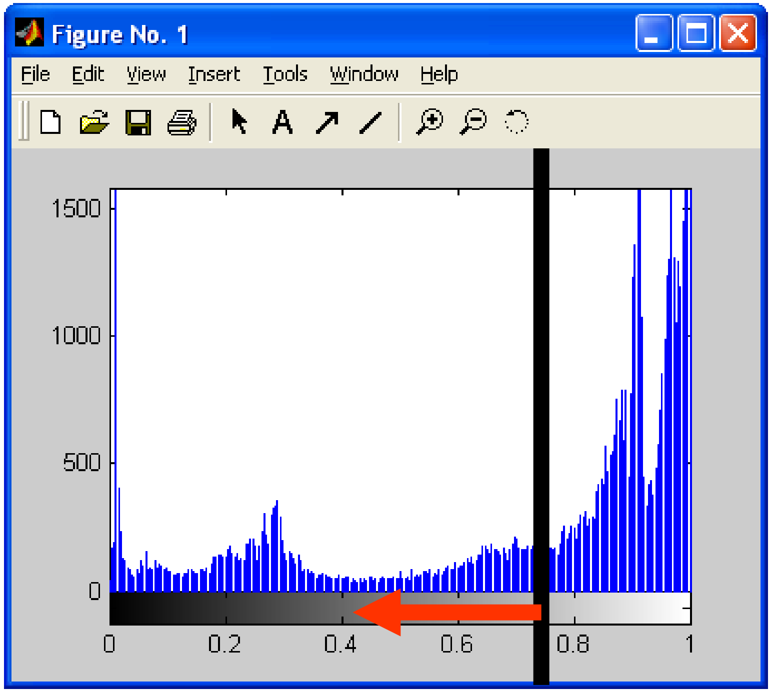
Example:

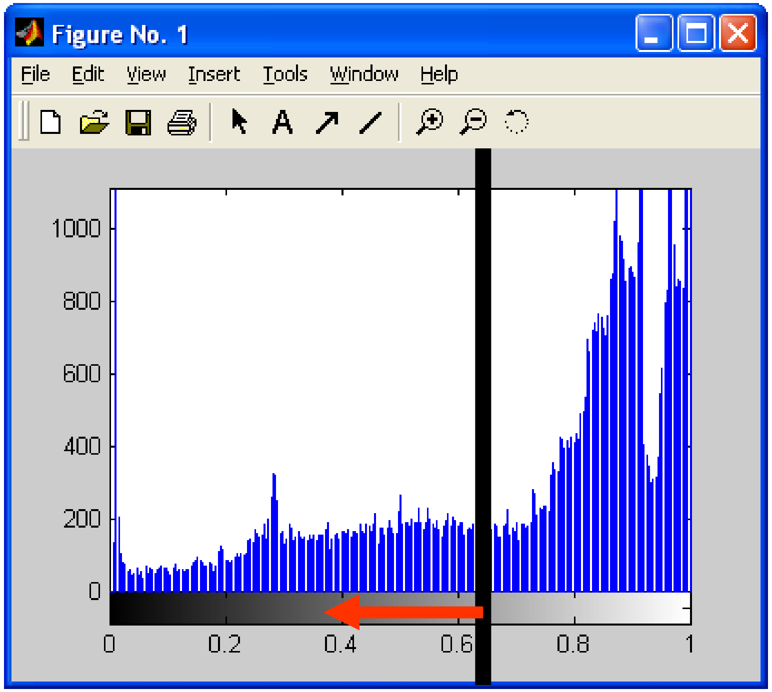
Image and blurred version





Histograms with MEAN and STANDARD DEVIATION





M=0.73 D=0.32

M=0.71 D=0.27