Image Processing

Chapter(3)

Part 3:Intensity Transformation and spatial filters

Prepared by: Hanan Hardan

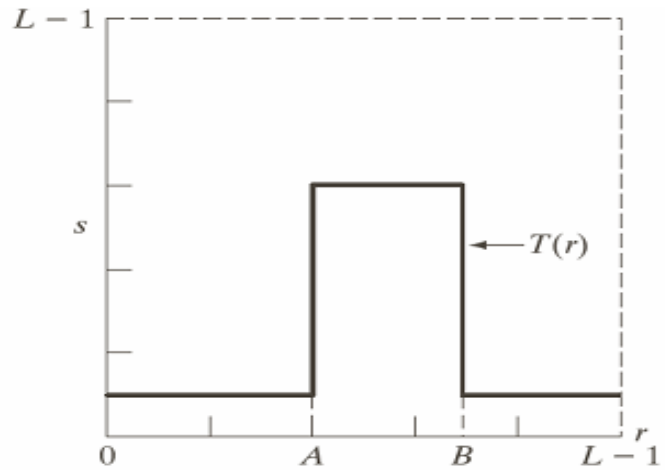
Gray-level Slicing

- This technique is used to highlight a specific range of gray levels in a given image.
  - Similar to thresholding
  - Other levels can be suppressed or maintained
  - Useful for highlighting features in an image
- It can be implemented in several ways, but the two basic themes are:
  - One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels.
  - The second approach, based on the transformation brightens the desired range of gray levels but preserves gray levels unchanged.

Gray-level Slicing

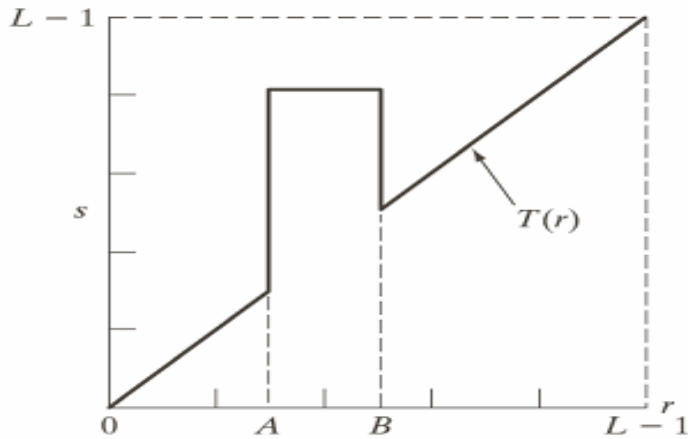
Highlighting a specific range of intensities in an image.

Approach 1



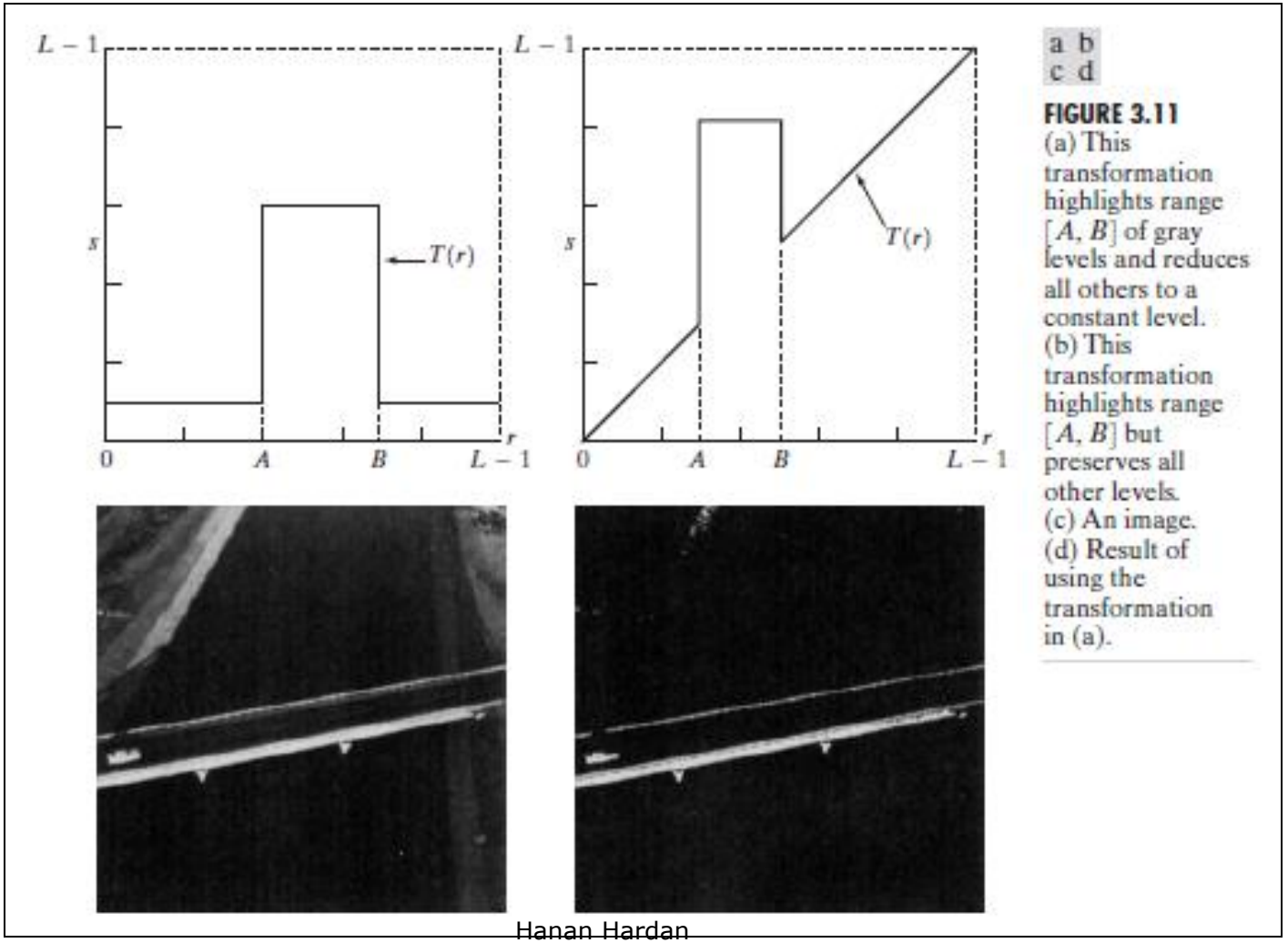
display in one value(e.g white) all the values in the range of interest , and in another (e.g black) all other intensities

Approach 2



Brightens or darkens the desired range of intensities but leaves all other intensity levels in the image unchanged

Gray-level Slicing

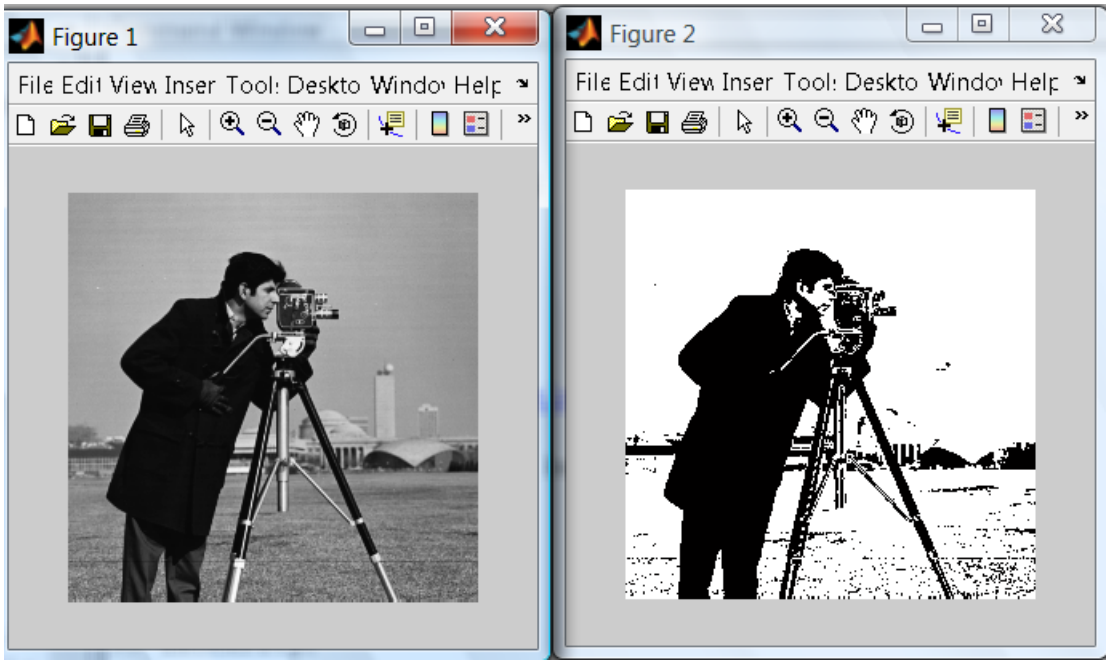


a b
c d

FIGURE 3.11
(a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level. (b) This transformation highlights range $[A, B]$ but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

Gray-level Slicing: approach 1

example: apply intensity level slicing in Matlab to read cameraman image , then If the pixel intensity in the old image is between (100 → 200) convert it in the new image into 255 (white). Otherwise convert it to 0 (black).



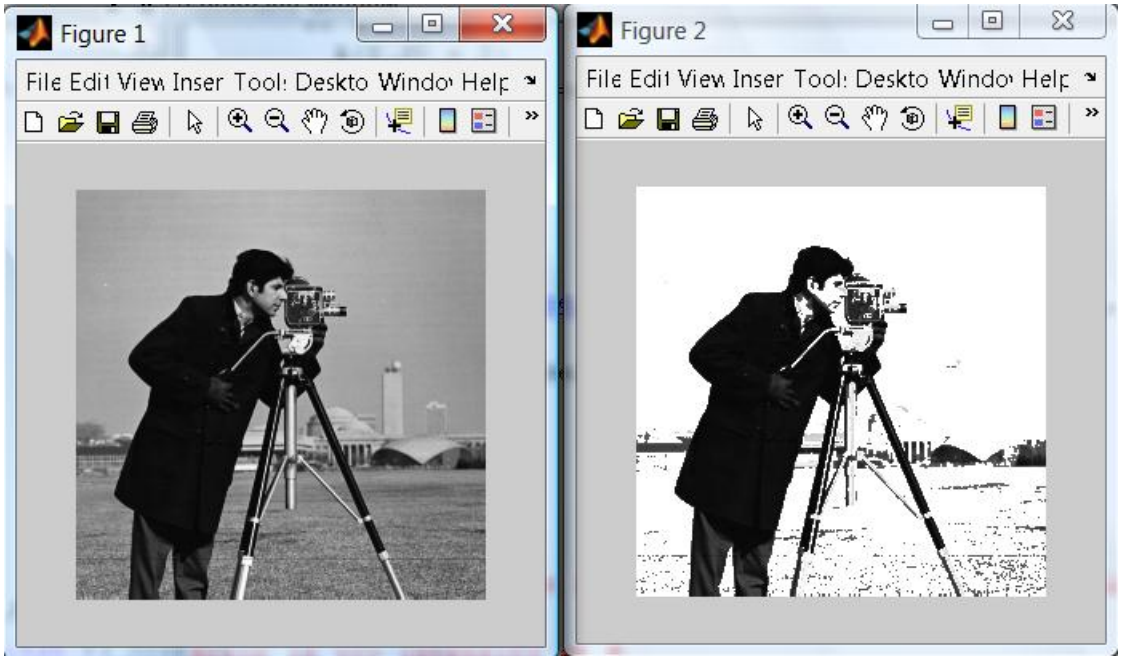
Gray-level Slicing: approach 1

Solution:

```
x=imread('cameraman.tif');
y=x;
[w h]=size(x);
for i=1:w
    for j=1:h
        if x(i,j)>=100 && x(i,j)<=200 y(i,j)=255;
        else y(i,j)=0;
        end
    end
end
figure, imshow(x);
figure, imshow(y);
```

Gray-level Slicing: approach 2

example: apply intensity level slicing in Matlab to read cameraman image , then If the pixel intensity in the old image is between (100 → 200) convert it in the new image into 255 (white). Otherwise it leaves it the same.



Gray-level Slicing: approach 2

Solution:

```
x=imread('cameraman.tif');
y=x;
[w h]=size(x);
for i=1:w
    for j=1:h
        if x(i,j)>=100 && x(i,j)<=200 y(i,j)=255;
        else y(i,j)=x(i,j);
        end
    end
end
figure, imshow(x);
figure, imshow(y);
```

Gray-level Slicing

Homework

example: apply intensity level slicing (approch2) in Matlab to read moon image , then If the pixel intensity in the old image is between (0 → 20) convert it in the new image into 130.

Bit-plane Slicing

- Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit.
- This method is useful and used in image compression.

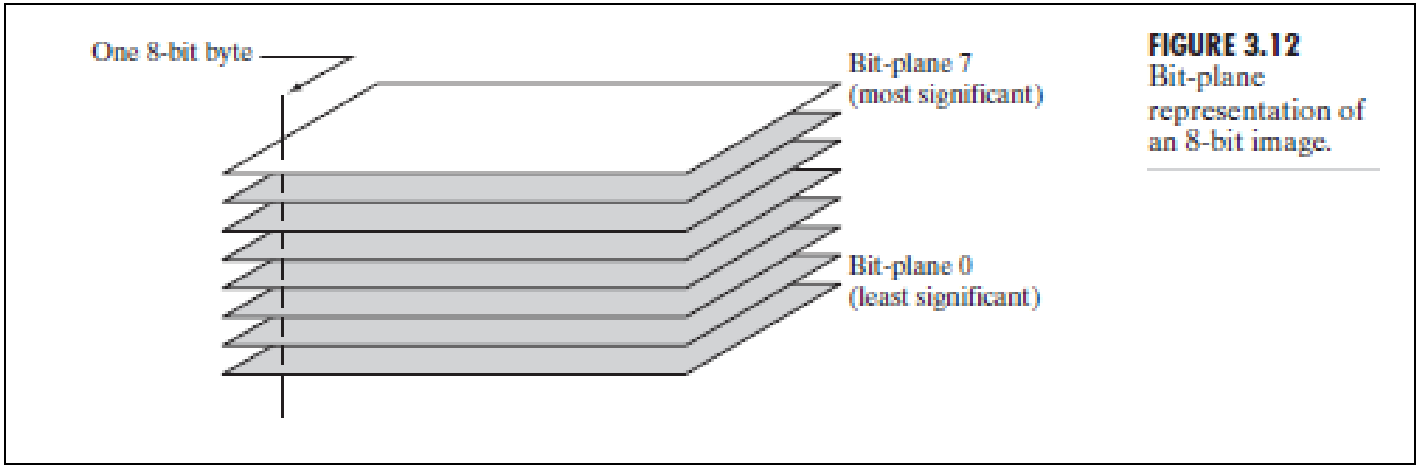
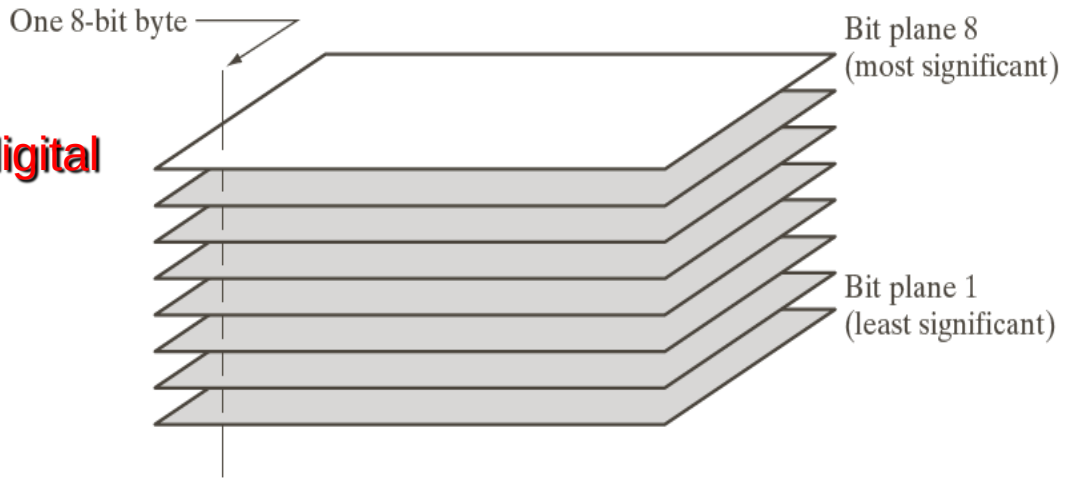


FIGURE 3.12 Bit-plane representation of an 8-bit image.

- Most significant bits contain the majority of visually significant data.

Bit-plane Slicing

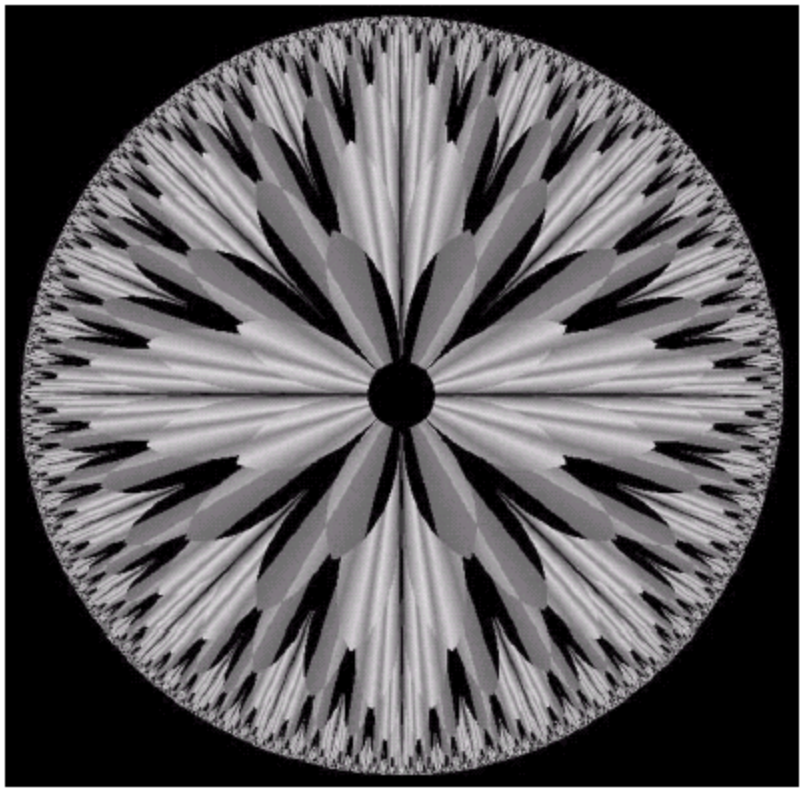
Remember that pixels are digital numbers composed of bits.



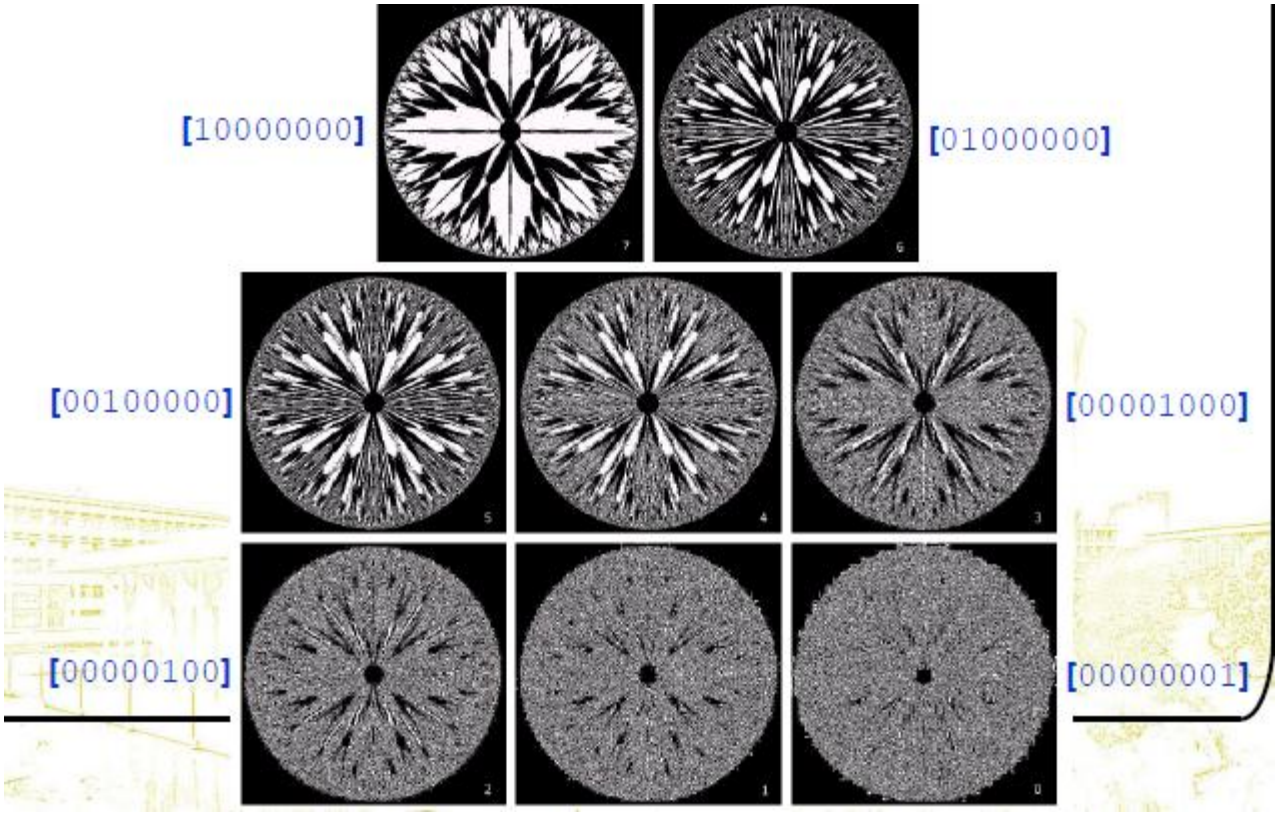
8-bit Image composed of 8 1-bit planes

Bit-plane Slicing

- Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image
- Higher-order bits usually contain most of the significant visual information
- Lower-order bits contain subtle details



Bit–Plane Slicing



Bit-plane Slicing

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

Reconstructed image using only bit planes 8 and 7

Reconstructed image using only bit planes 8, 7 and 6

Reconstructed image using only bit planes 7, 6 and 5

Bit-plane Slicing(example)

We have to use bit get and bit set to extract 8 images;

0 1 1 0 0 1 0 0

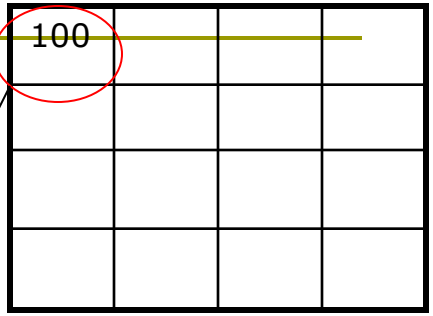


Image of bit1:
0000000**0**

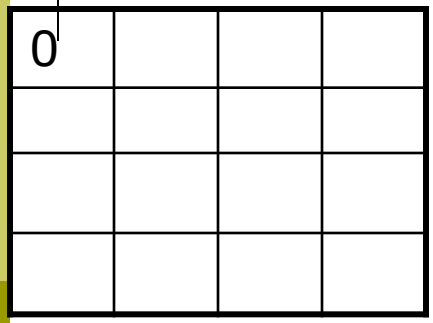


Image of bit2:
000000**0**0

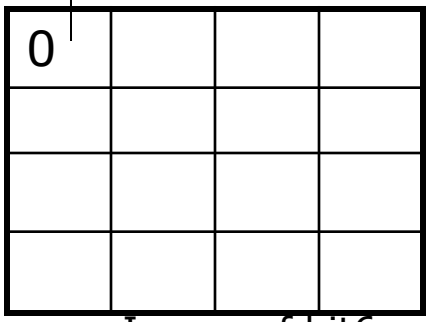


Image of bit3:
00000**1**00

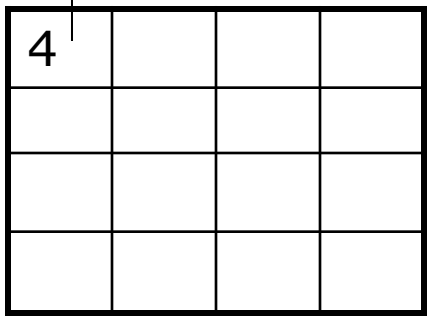


Image of bit4:
0000**0**000

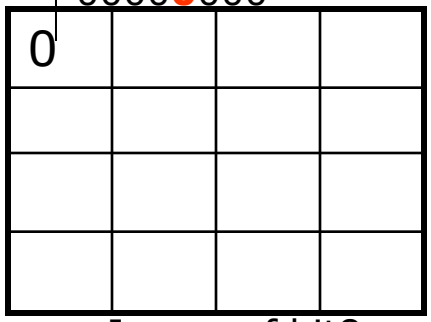


Image of bit5:
000**0**0000

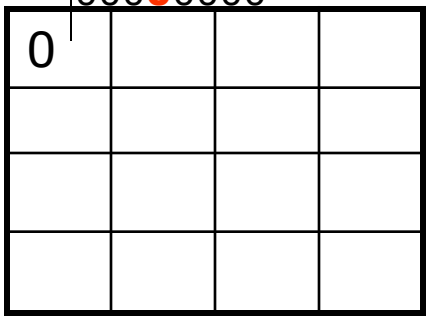


Image of bit6:
00**1**00000

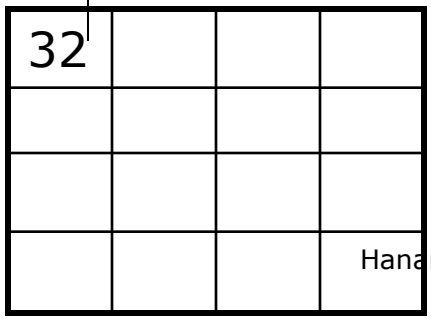


Image of bit7:
0**1**000000

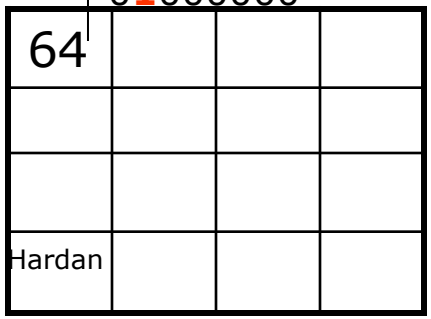
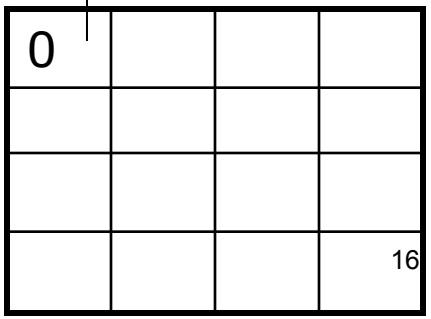


Image of bit8:
00000000



Bit-plane Slicing- programmed

example: apply bit-plane slicing in Matlab to read cameraman image , then extract the image of bit 6.

Solution:

```
x=imread('cameraman.tif');
y=x*0;
[w h]=size(x);
for i=1:w
    for j=1:h
        b=bitget(x(i,j),6);
        y(i,j)=bitset(y(i,j),6,b);

    end
end

figure, imshow(x);
figure, imshow(y);
```

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.

Histogram?

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

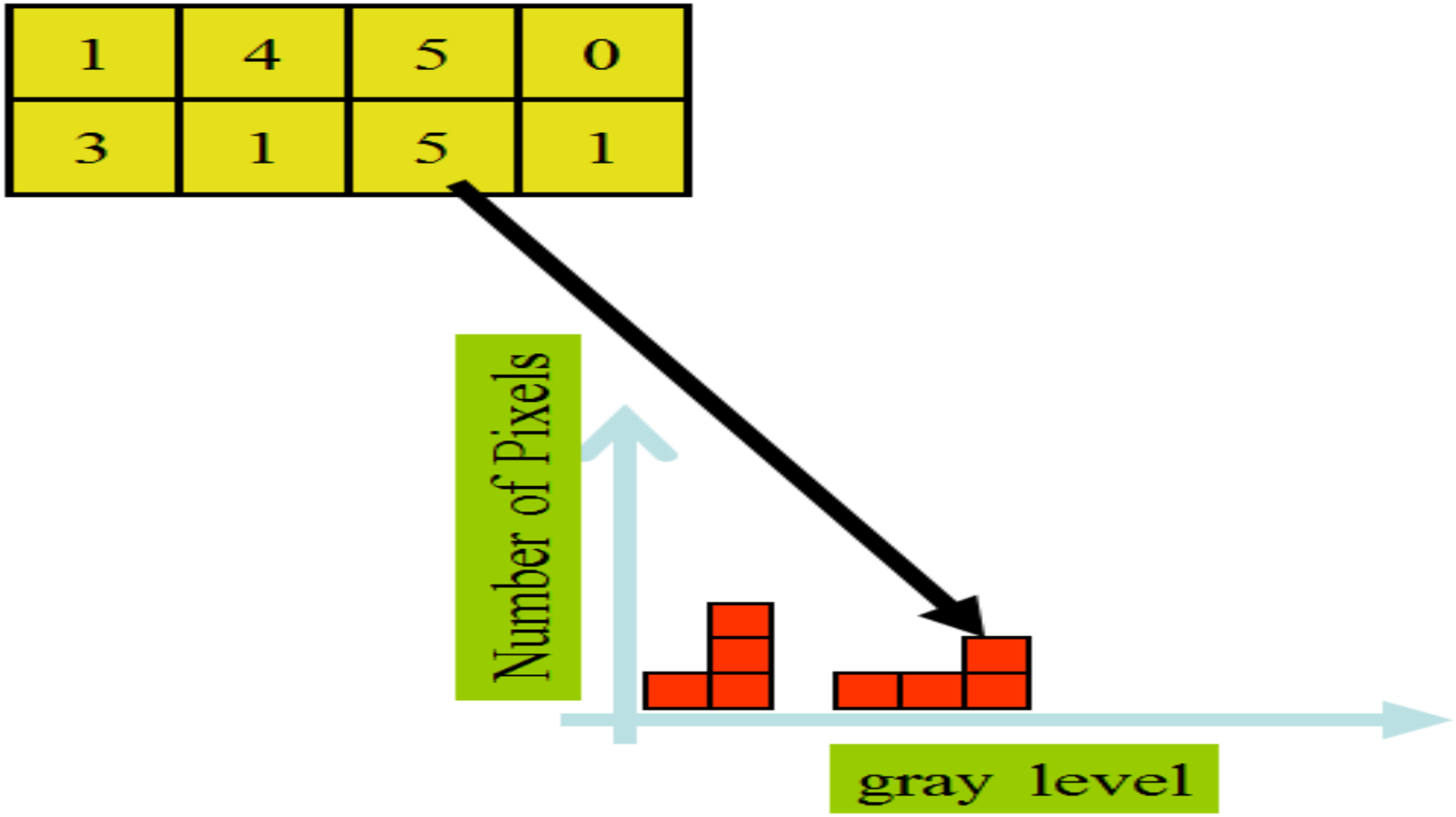
$$h(r_k) = n_k$$

Where:

r_k : kth gray level

n_k : # of pixels with having gray level r_k

Image Histogram



Histogram Processing

- It is common practice to normalize a histogram by dividing each of its values by the total number of pixels in the image, denoted by n. Thus, a normalized histogram is given by

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

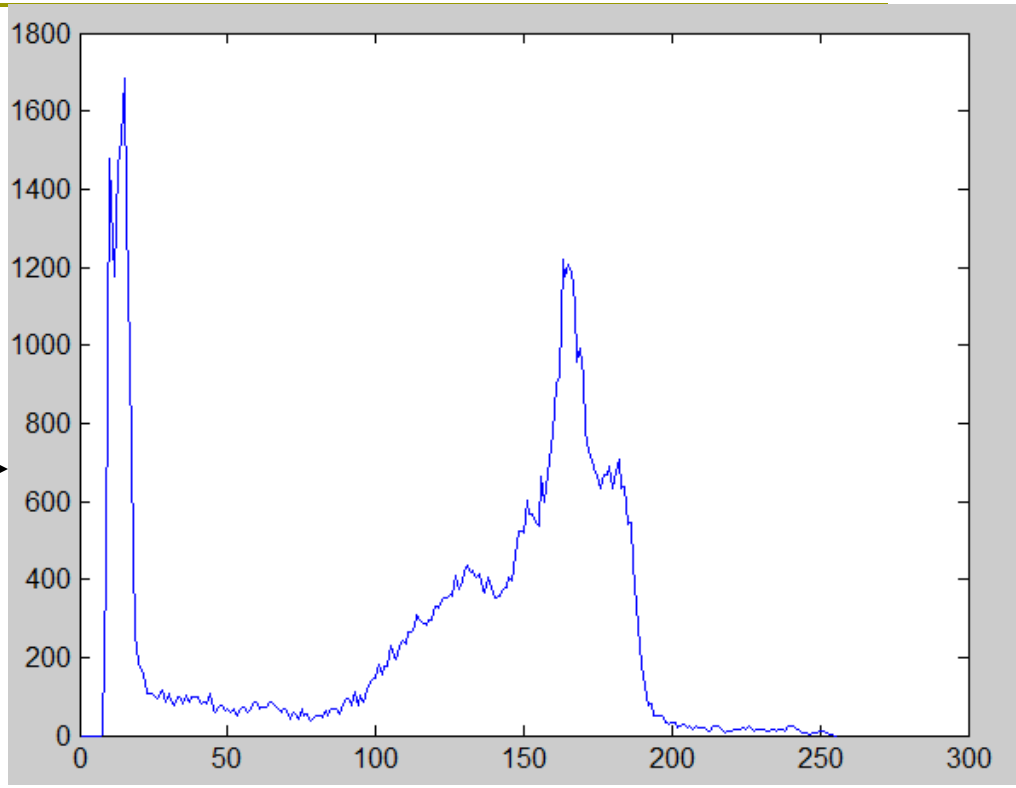
- Thus, $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k.
- Note that the sum of all components of a normalized histogram is equal to 1.

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.

Histogram of the image:

histogram



gray levels هو تمثيل لعدد البكسل في كل قيمة لونية من درجات

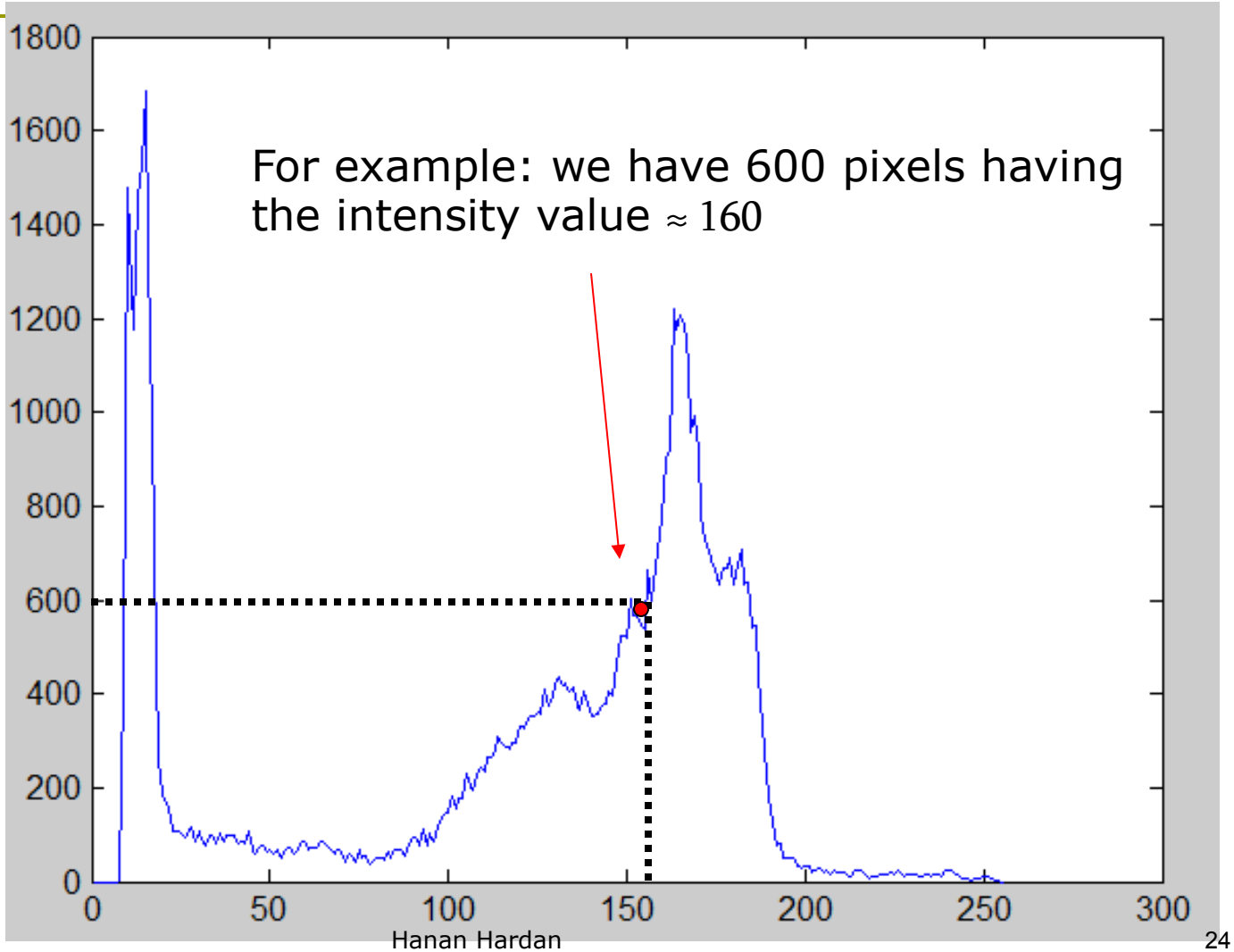
$h(r_k) = n_k$

Where:

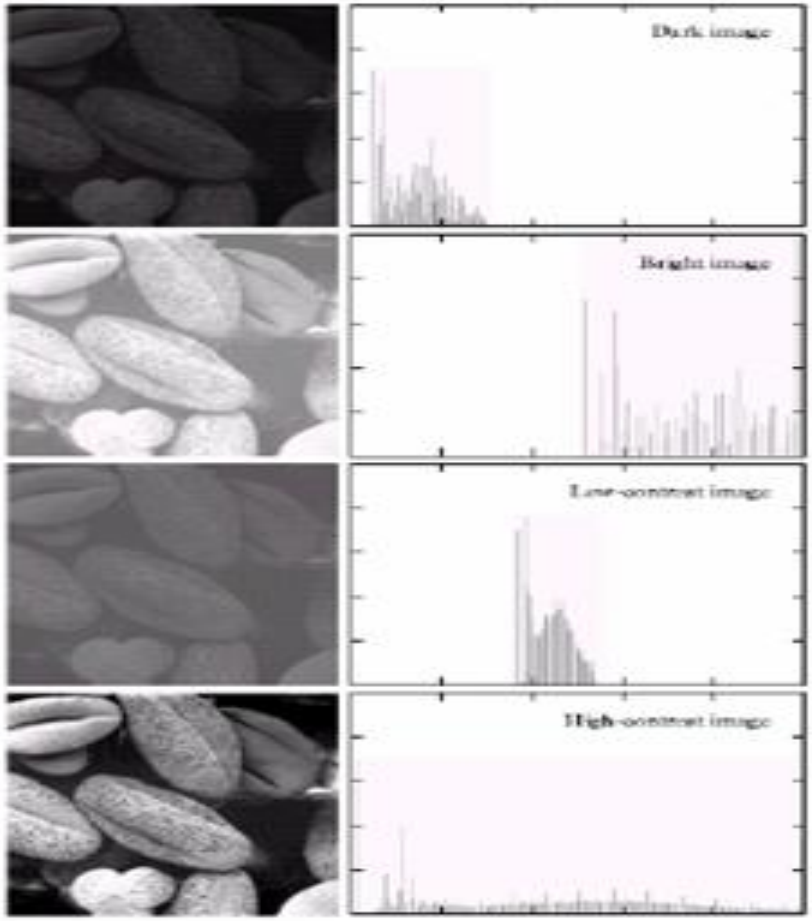
r_k : kth gray level

n_k : # of pixels with having gray level rk

Histogram of the image:



Histogram of the image:



→ low contrast image

→ low contrast image

→ low contrast image

→ high-contrast image

An image whose pixels tend to occupy the entire range of possible gray levels and, in addition, tend to be distributed uniformly, will have an appearance of high contrast and will exhibit a large variety of gray tones.

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].

Histogram in MATLAB

- We obtain the normalized histogram simply by using the expression.

  p = imhist (f, b) / numel(f)

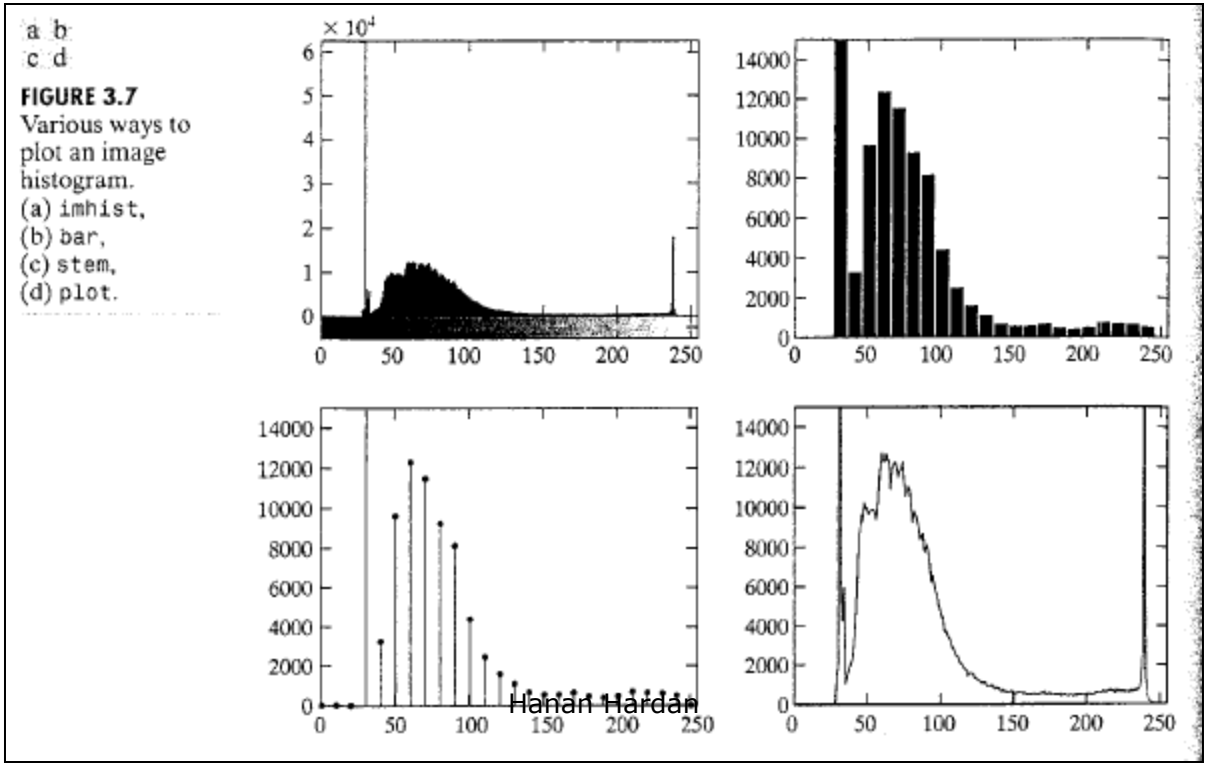
  numel (f): a MATLAB function that gives the number of elements in array f (i.e. the number of pixels in an image).

Other ways to display Histograms

- Consider an image f. The simplest way to plot its histogram is to use imhist with no output specified:

  >> imhist (f);

  Figure 3.7(a) shows the result.



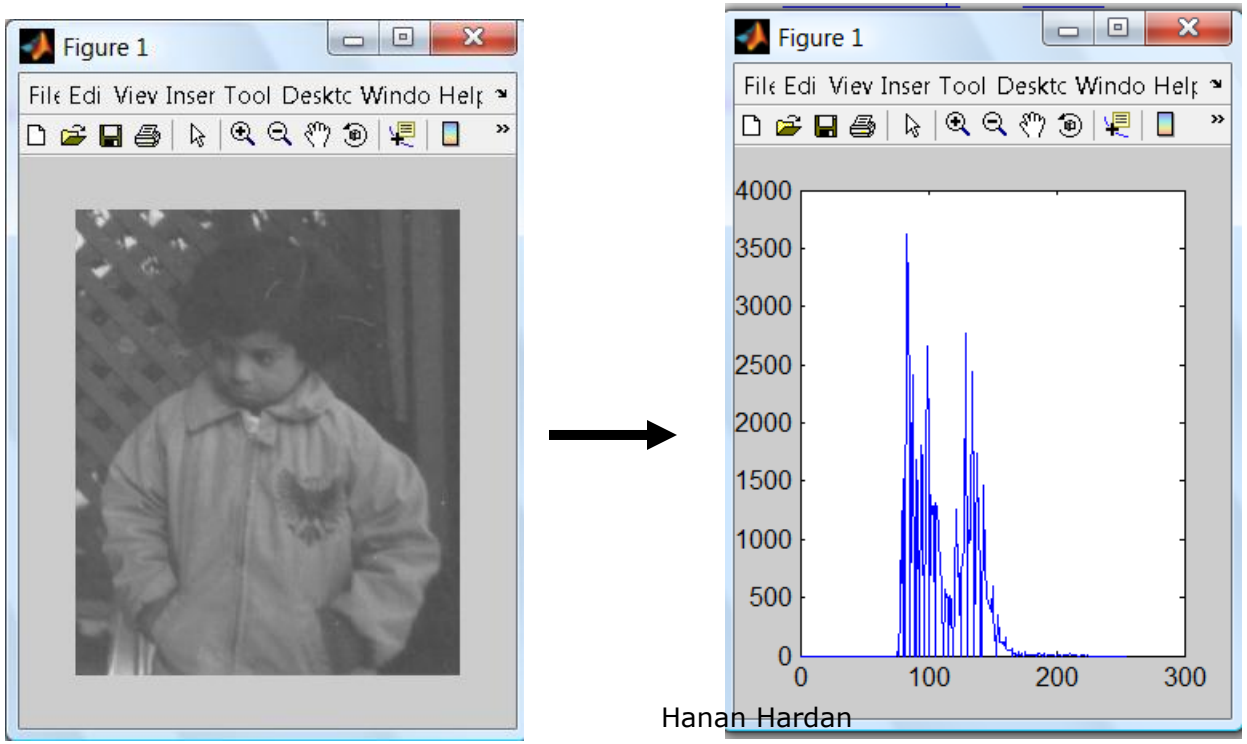
Other ways to display Histograms

- A **stem** graph
- A **bar** graph
- A **Plot** graph

```
>> h = imhist(f);
>> bar (h);
>> plot (h);
>> stem (h);
```

Histogram equalization of the image:

We have this image in matlab called pout.tif, when we plot its histogram it is showed like this:



Notice that the pixels intensity values are concentrated on the middle

(low contrast)

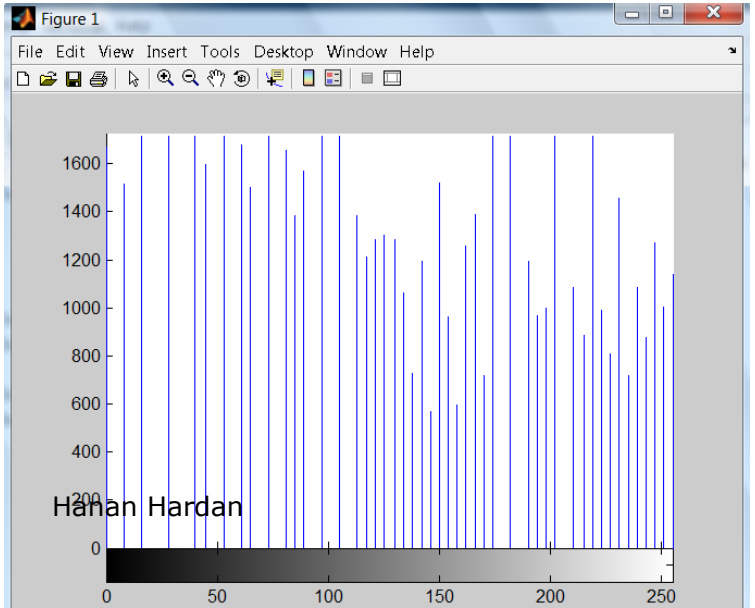
Histogram equalization of the image:

histogram equalization :

is the process of adjusting intensity values of pixels.

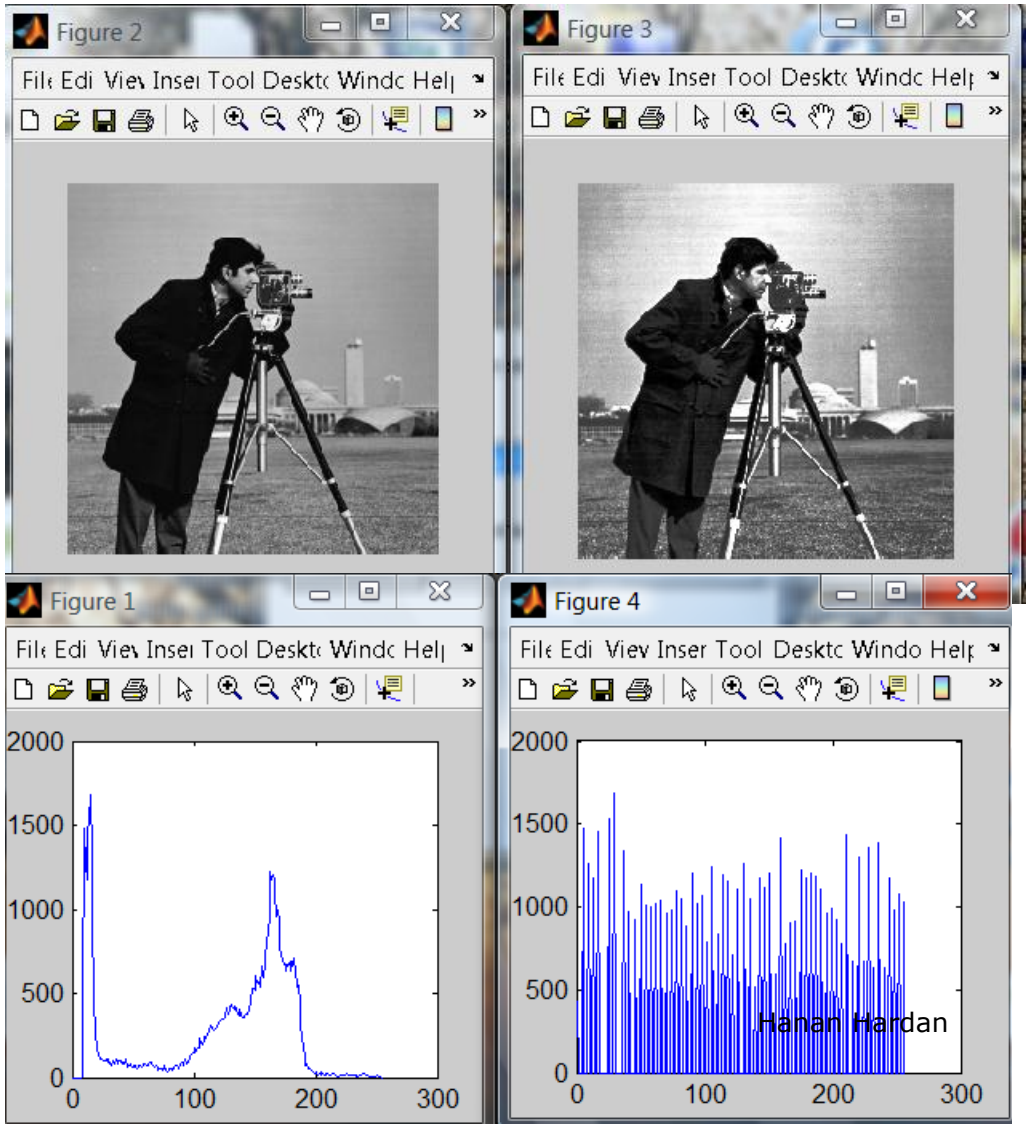
The process which increases the dynamic range of the gray level in a law contrast image to cover full range of gray levels.

Im matlab : we use **histeq** function



Histogram produces pixels having values that are distributed throughout the range

Histogram equalization of the image:



Notice that histogram equalization does not always produce a good result

Equalization (mathematically)

$g(x) = (L/n).\ T(X) - 1$

Where,

G(X) : the new image after equalization

L: No of gray levels 2^n

n: No of pixels

T(x): cumulative sum of each gray level

Equalization (mathematically)

L grayl evels	X عدد البكسل لكل Graylevel	T(X)مجموع تراكمي للبكسل	G(x)
0	1	1	0
1	3	4	0
2	5	9	1
3	6	15	2
4	6	21	4
5	6	27	5
6	2	29	6
7	3	32	7

Assume that we have (3bits per pixels) or 8 levels of grayscale, and we want to equalize the following image example.

$G(x)=(L/n).\ T(X) -1$

$=(8/32).\ T(x) -1$

8 عدد ال graylevel

عدد البكسلات الكلي No of pixels