Digital Signal and Image Processing
CSC 701

Subject In-charge

Ms Sweedle M

Assistant Professor

email: sweedlemachado@sfit.ac.in

Module V

Image Enhancement in Spatial Domain

Image Enhancement

Image Enhancement

- Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further image analysis.
- Image Enhancement can be done in two domains:
  - Spatial Domain
  - Frequency Domain.
- Spatial domain enhancement involves direct manipulation of pixels of the image.
- Frequency domain enhancement involves modifying the frequency domain transform of an image

Reasons for doing Image Enhancement?

Image enhancement is the process of making images more useful

*The **reasons** for doing this include:*

- ***Highlighting** interesting detail in images*
- ***Removing noise** from images*
- *Making images more **visually** appealing*

Image Enhancement Examples

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Enhancement Examples (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

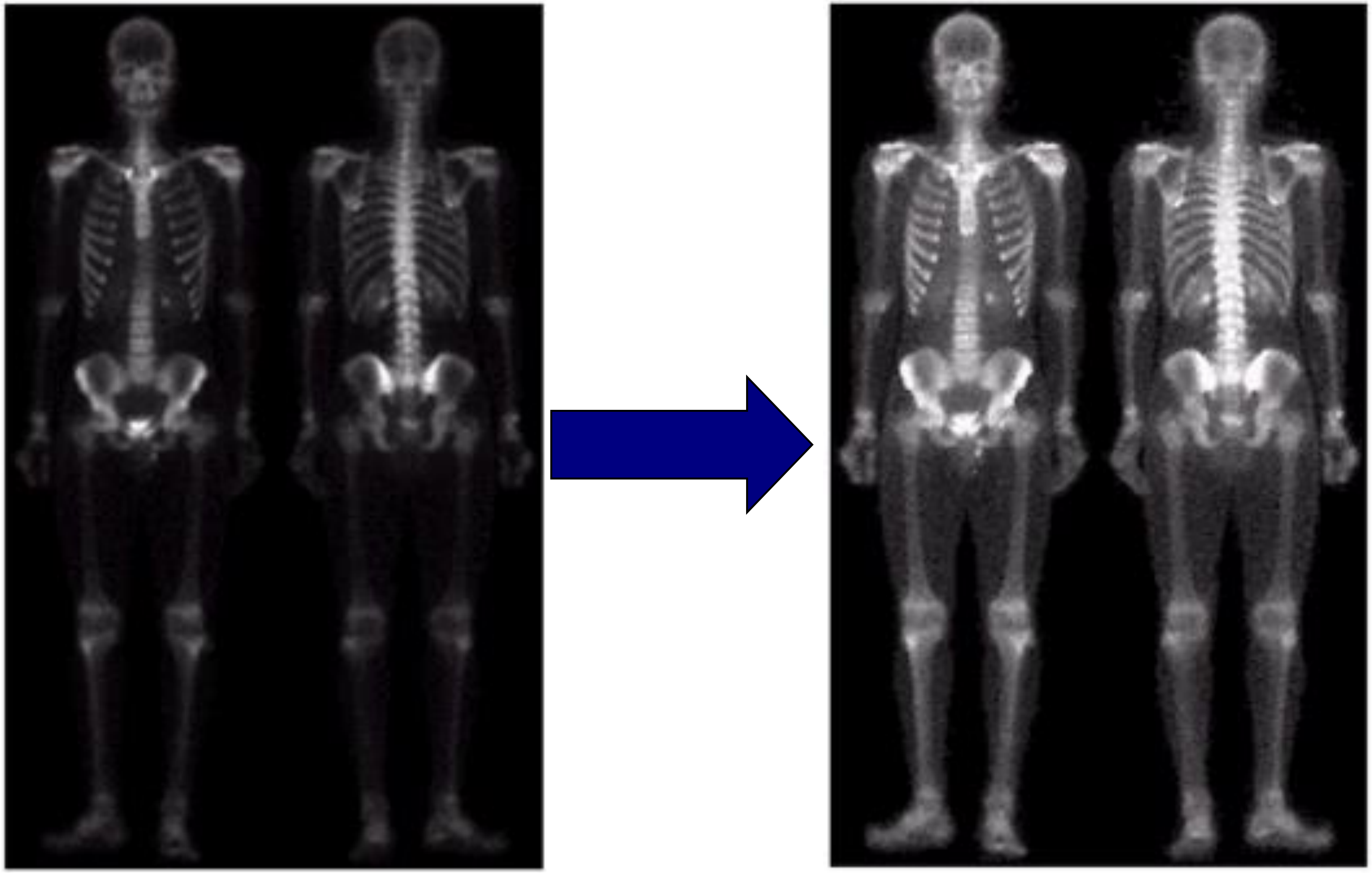
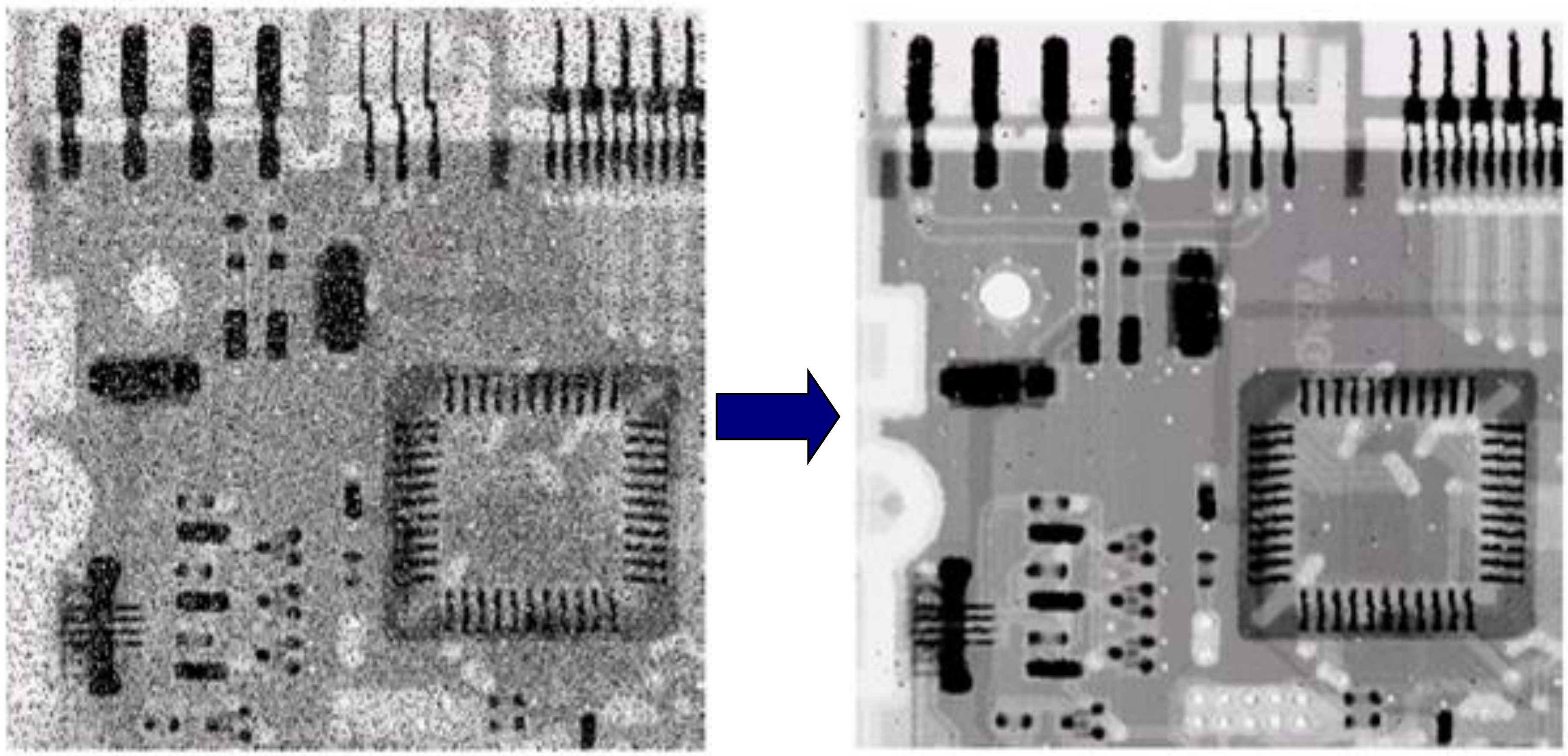


Image Enhancement Examples (cont…)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Enhancement Examples (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Spatial Domain Enhancement

Spatial Domain Enhancement

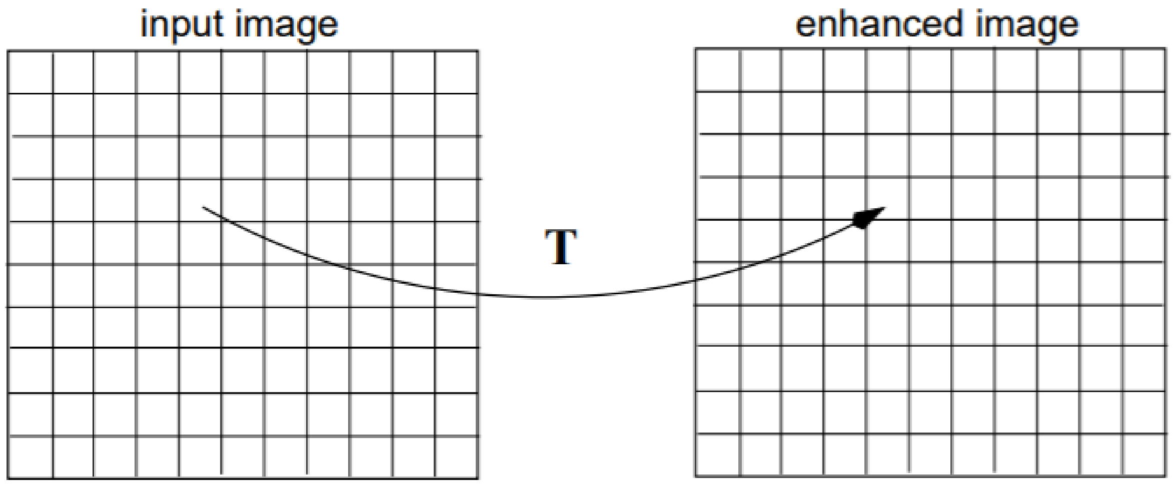
- Let **f(x,y)** be the original image where **f** is the grey level value and **(x,y)** be the image coordinates.
- Then the modified image can be represented as **g(x,y) = T{f(x,y)}** where **T** is the transformation applied to the image to get g(x,y).
- Spatial domain enhancement can be carried out in two different ways:
  - point processing and
  - neighborhood processing

Point Processing

Point Processing

- Point processing uses only the information in individual pixels to produce new images.
- A transform may be computed on the basis of regional or global information and then applied to the individual points. Such transforms are scalar functions.

Point Processing Methods



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

T operates on 1 pixel

Point Processing

- It means that the new value g(x,y) depends on the operator T and the present f(x,y).
- Some of the point processing techniques are
  1. Digital Negative
  2. Thresholding
  3. Gray Level Slicing
  4. Contrast Stretching
  5. Bit plane Slicing
  6. Dynamic Range Compression
  7. Power Law Transformation

Digital Negative

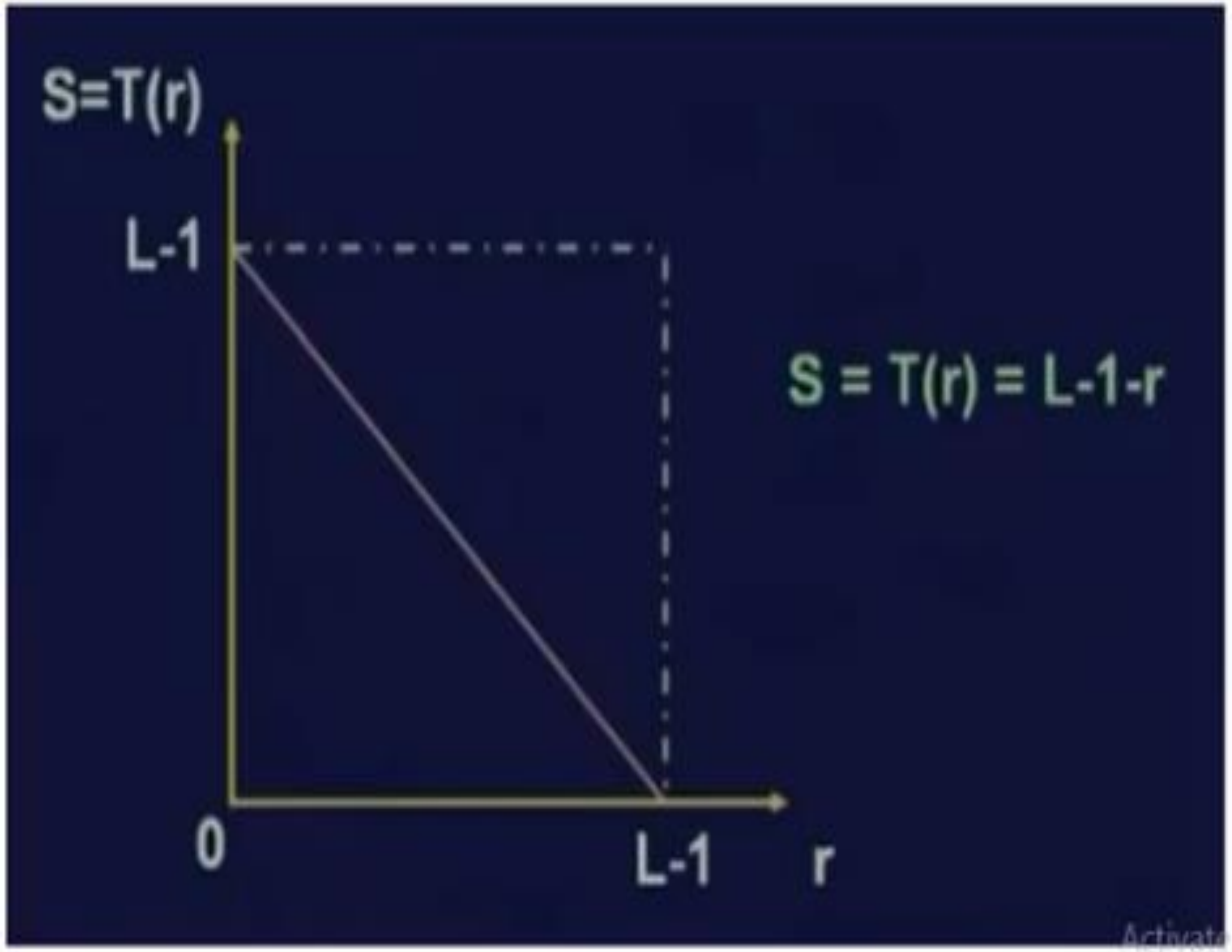
- Image negative is produced by subtracting each pixel from the maximum intensity value.
- Thus, the transformation function used in image negative is

$$s = T(r) = L - 1 - r$$

- where L-1 is the max intensity value and s, and r are the output and input pixel values respectively.
- For an 8-bit gray level image, the negative transformation is:

$$s = L - 1 - r = 2^8 - 1 - r = 256 - 1 - r = 255 - r$$

Digital Negative



Example

Original Image

0	255	127
10	250	50
200	100	50

8-bit & (3 X 3) Image

Negative Image

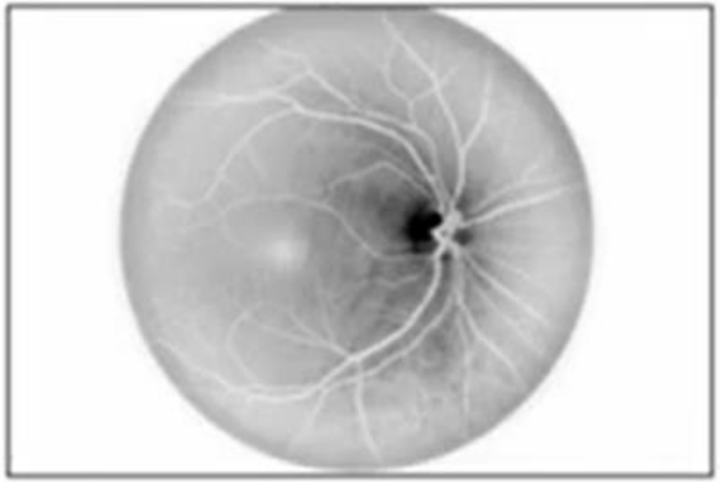
255	0	128
245	5	205
55	155	205

Digital Negative

- For gray scale images, light areas appear dark and vice versa.
- This kind of transformation is suited for enhancing white or gray detail embedded in dark areas of an image.

img

Negative



Example of an Digital Negative Image

Fig.-a) Original Image

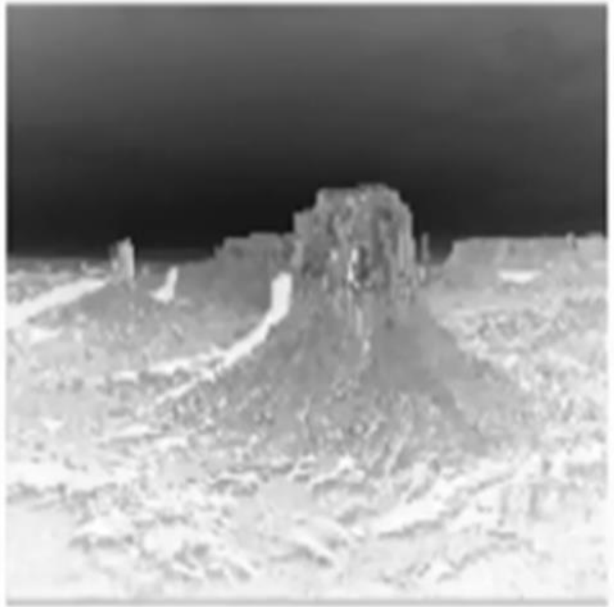


Fig.-b) Negative image

Digital Negative

- For color image, colors are replaced by their complementary colors.
- Thus, red areas appear cyan, greens appears magenta, and blues appears yellow, and vice versa.

Original image

Negative Image

Problem

- Obtain the digital negative of the 3-bit image shown below:

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Solution

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Solution

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

6	5	5	5	5
4	5	3	2	5
5	1	1	0	7
5	1	1	2	6
7	5	4	5	6

Problem

eg Obtain digital negative of foll. 8-bits per pixel- BPP image

121	205	217	156	151
139	127	157	117	125
252	117	236	138	142
227	182	178	197	242
201	106	119	251	240

→ Its a 8-bit image So no. of grey levels that this image can hold is $2^8 = 256 \quad \therefore L = 256$

Hence min^m grey level is 0. while max^m grey level is 255.

$$s(x,y) = (L-1) - r(x,y)$$

$$= (256-1) - r(x,y)$$

$$S(x,y) = 255 - r(x,y)$$

134	50	38	99	104
116	128	98	138	130
3	138	19	117	113
28	73	77	58	13
54	149	136	4	15

Problem

eg: Digital negative of an image.

4	3	2	1
3	1	2	4
5	1	6	2
2	3	5	6

→ $6 \Rightarrow$ neares $2^n \Rightarrow n = 3$

$L = 2^3 = 8$ — 0 to 7

$\therefore \cancel{s = r - 7}$

$s = (L-1) - f(x,y)$

$\therefore s = 7 - r$

$\therefore$ digital negative of image is,

3	4	5	6
4	6	5	3
2	6	1	5
5	4	2	1

Thresholding

- In some images, it is useful to be able to separate out the regions of the images in which we are interested, from the regions of the image that correspond to background.
- Thresholding often provides an easy and convenient way to perform this segmentation on the basis of the different intensities of colors in the foreground and background regions of an image.
- In addition, it is often useful to be able to see what areas of an image consist of pixels whose values lie within a specified range, or band of intensities (or colors).

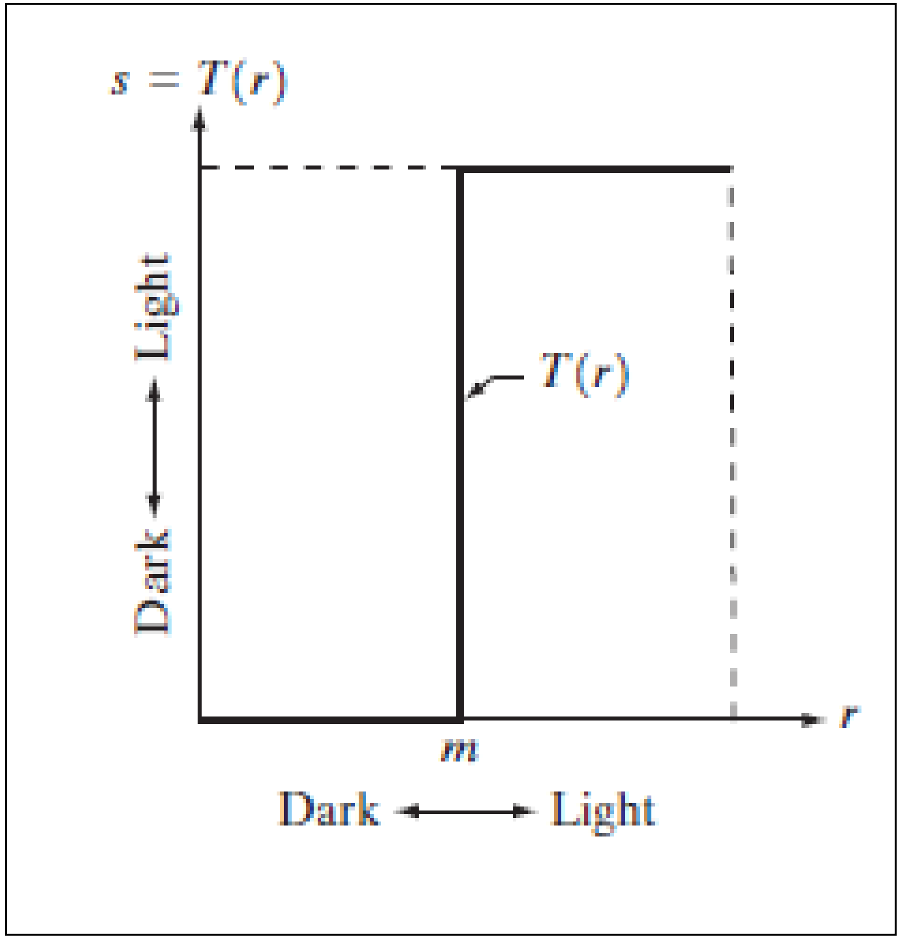
Thresholding

- In a single pass, each pixel in the image is compared with this threshold.
- The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity I{i,j} is less than some fixed constant T or a white pixel if the image intensity is greater than that constant.
- Single value thresholding can be represented mathematically as:

$$\boxed{g(x,y)=\begin{cases}1\ if\ f(x,y)>T\\0\ if\ f(x,y)\le T\end{cases}}$$

Thresholding

- It is represented as follows:



Example:

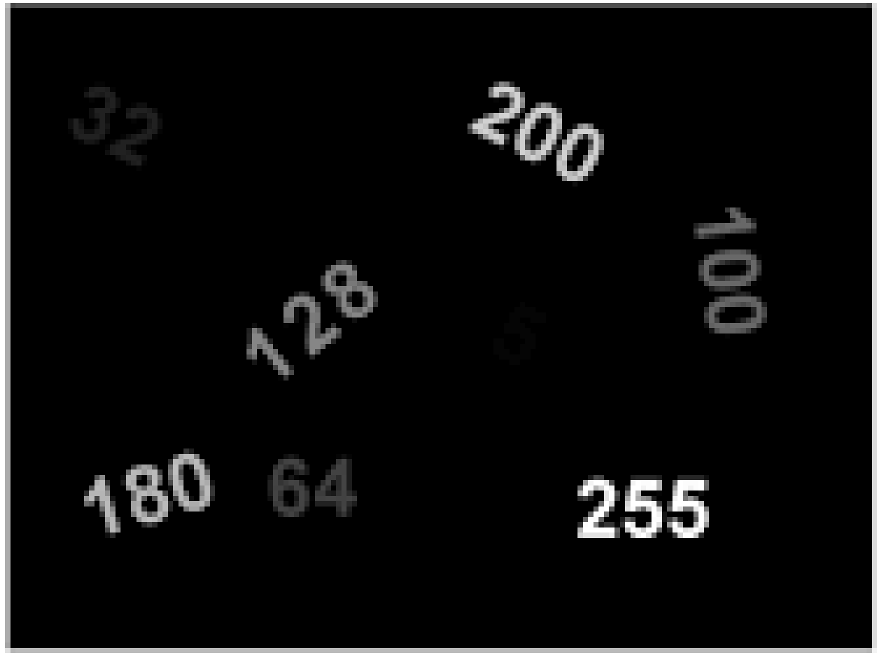
123	56	78
255	82	25
93	85	62

→

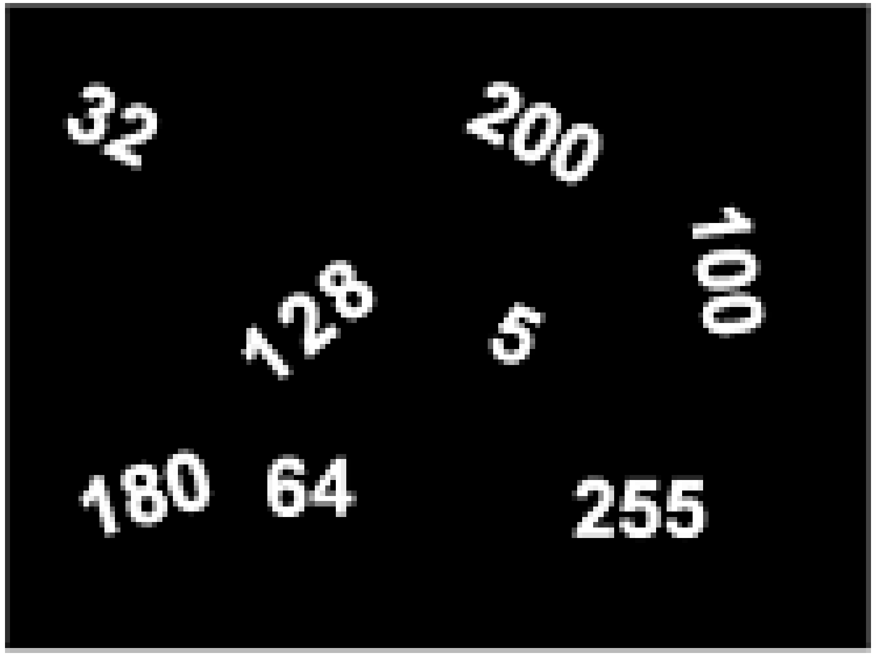
1	0	0
1	1	0
1	1	0

Threshold = 80

Examples:

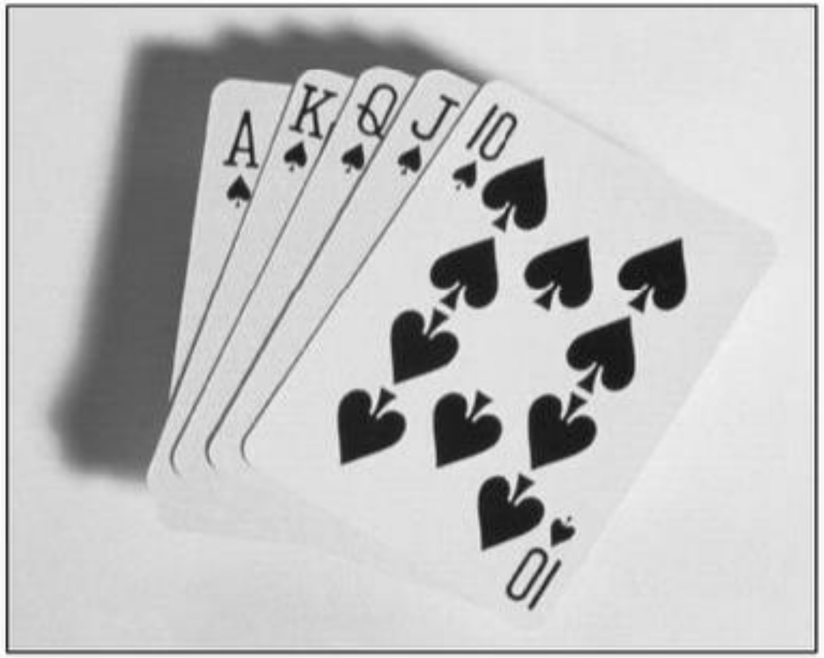


Original Image

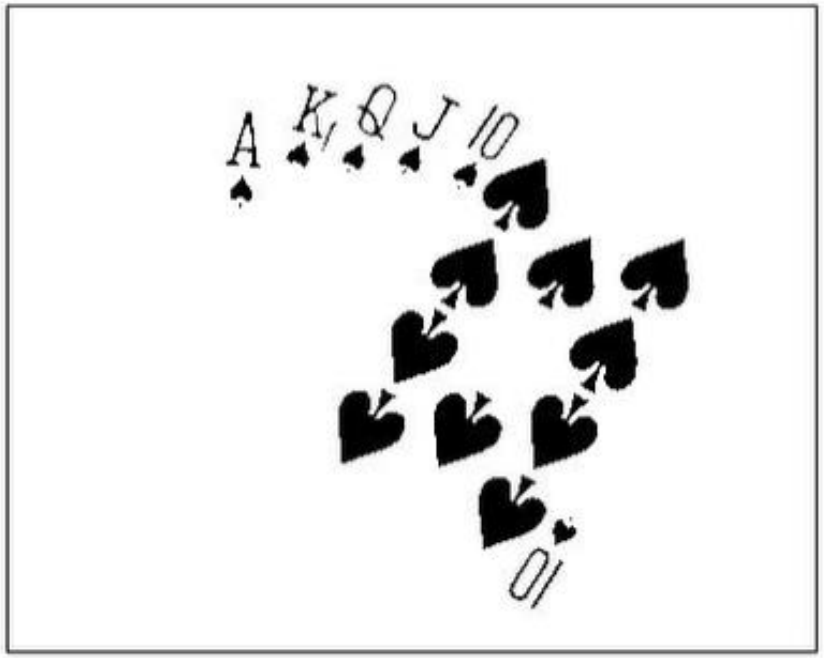


Thresholded Image

Examples:

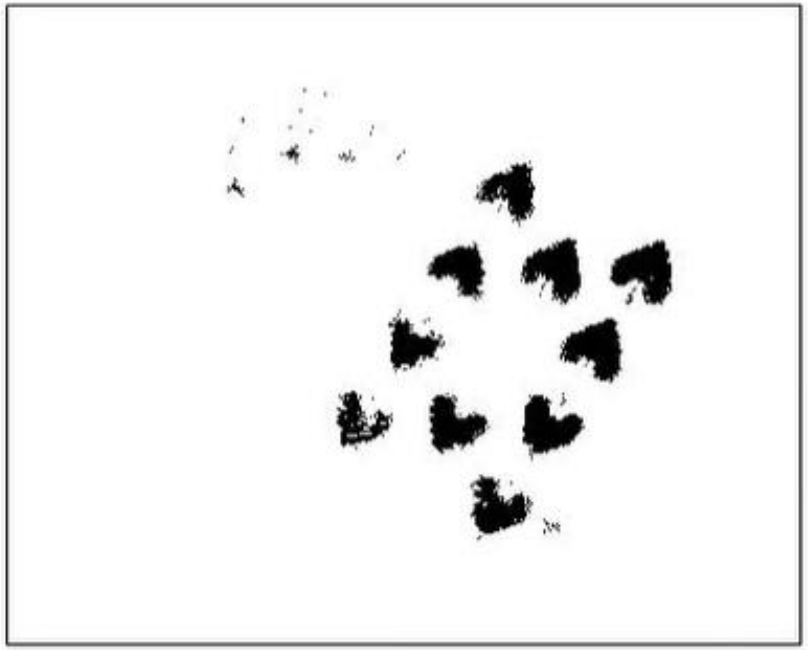


Original Image

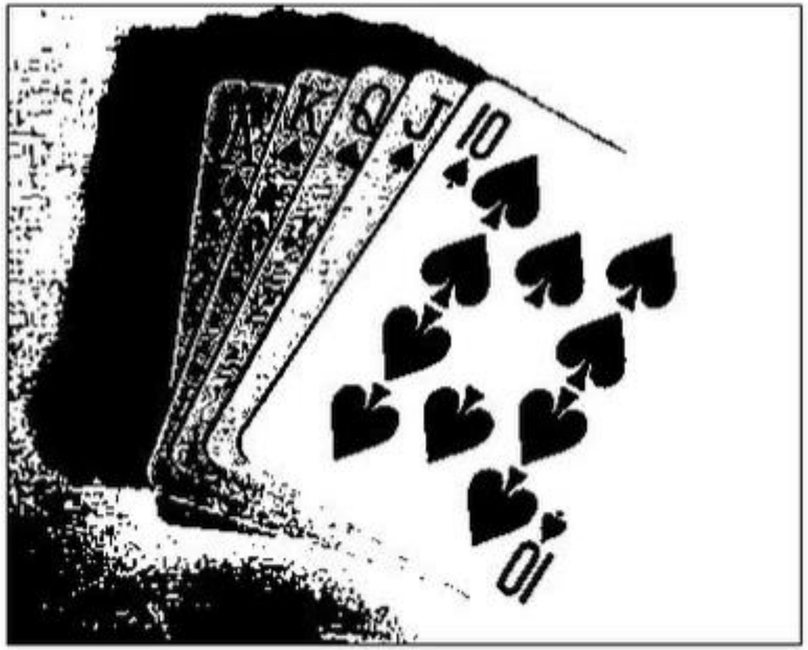


Thresholded Image

Effect of wrong values of threshold



Threshold Too Low



Threshold Too High

Thresholding

- In more sophisticated implementations, multiple thresholds can be specified, so that a band of intensity values can be set to white while everything else is set to black.
- For color or multi-spectral images, it may be possible to set different thresholds for each color channel, and so select just those pixels within a specified cuboid in RGB space.

Problem

- Obtain the Threshold image for the 3 Bit image shown below. Choose threshold value T = 4.

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Solution

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

0	0	0	0	0
0	0	0	7	0
0	7	7	7	0
0	7	7	7	0
0	0	0	0	0

- Problem: Obtain the Threshold image for the 3 Bit image shown below. Choose threshold value T = 4.

1	2	3	0
2	4	6	7
5	2	4	3
3	2	6	1

0	0	0	0
0	7	7	7
7	0	7	0
0	0	7	0

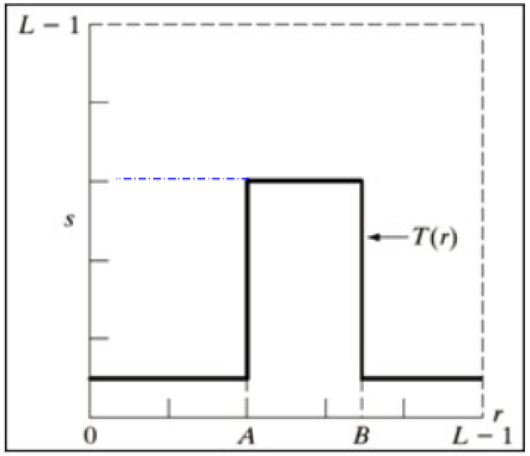
Gray Level Slicing

- This technique is used to highlight a specific range of gray levels in a given image.
- It can be implemented in several ways, but the two basic approaches are:
- Approach 1: First 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.

Approach 1 : Grey Level slicing without background

- Approach 1: First 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.
- This transformation is given as follows:

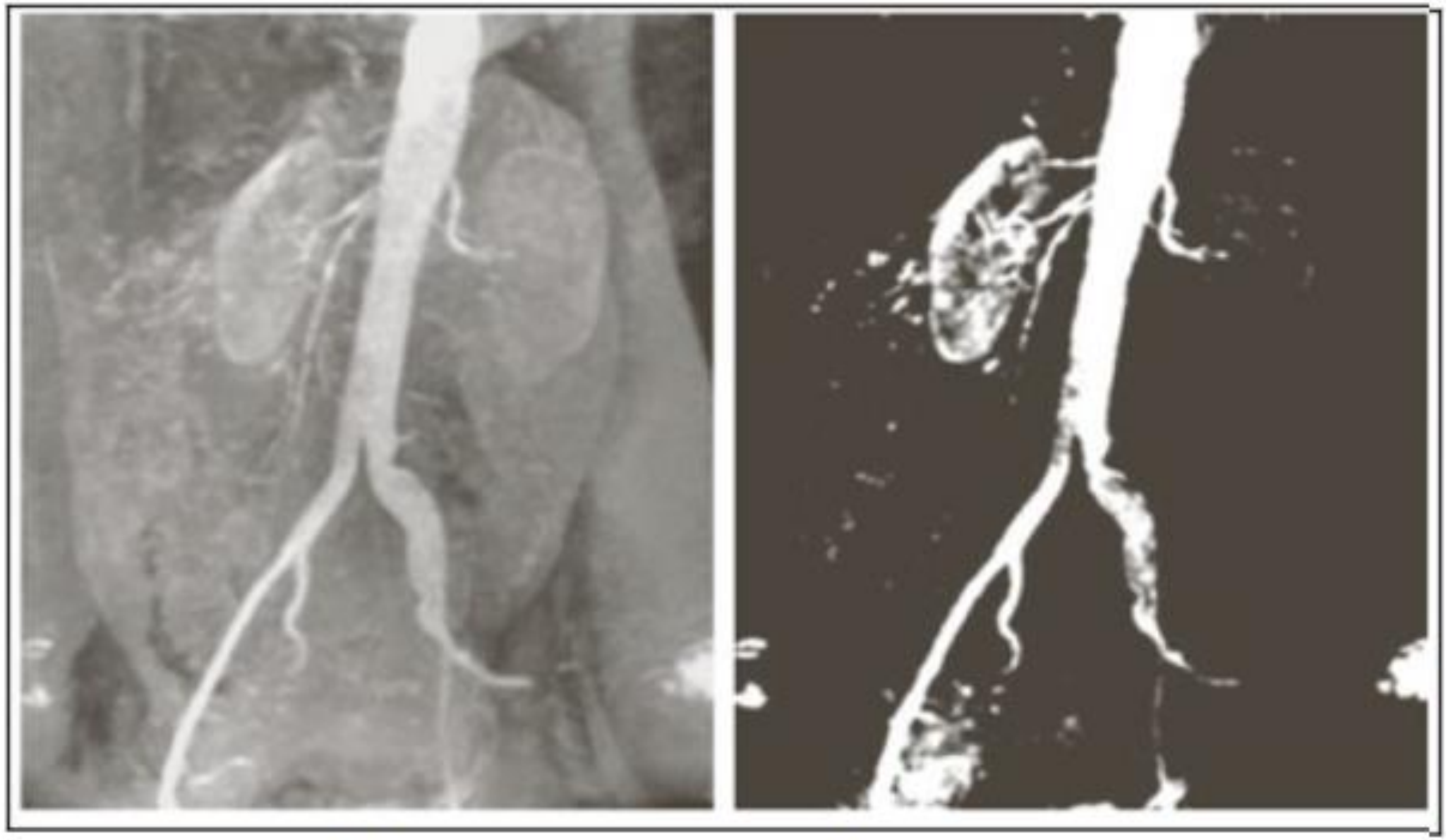
Without background



$$s = \begin{cases} L; A \le r \le B \\ 0; otherwise \end{cases}$$

- It produces a binary image.

Example:

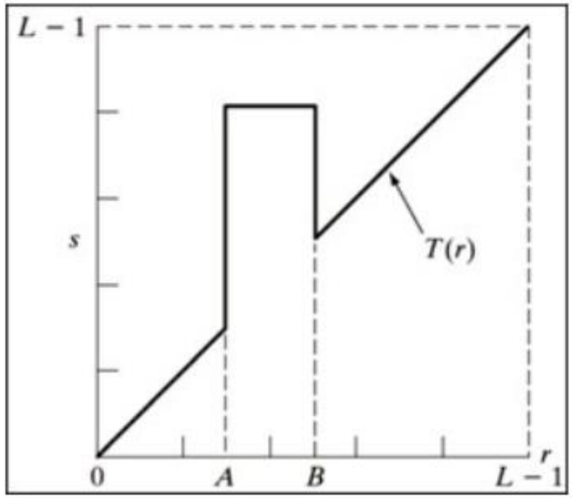


Input image Without background

Approach 2: Grey Level Slicing With Background

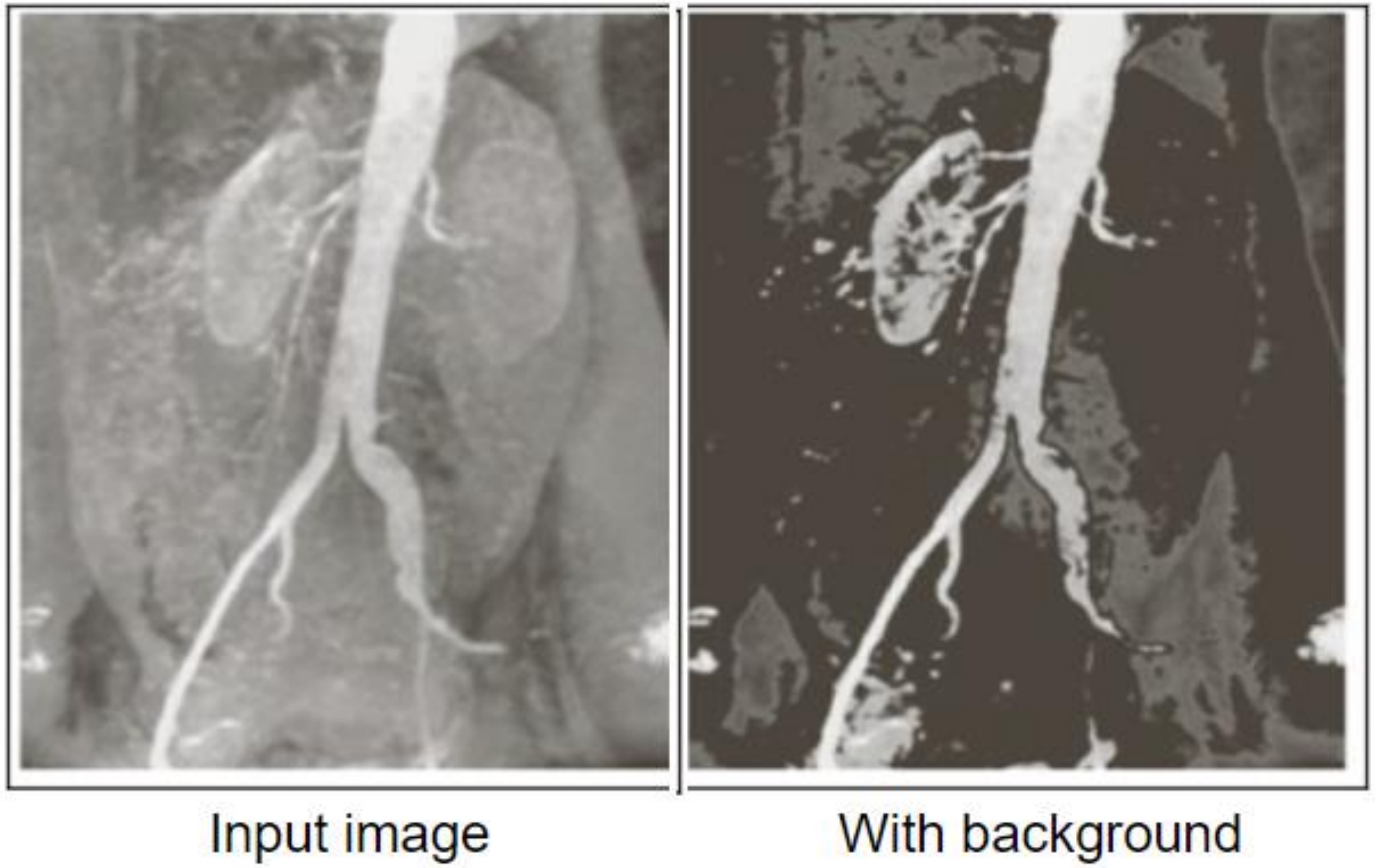
- The second approach is based on the transformation which brightens the desired range of gray levels but preserves gray levels unchanged.
- This transformation is given as follows:

With background



$$s = \begin{cases} L; A \le r \le B \\ r; otherwise \end{cases}$$

Example:



Gray level slicing/Intensity Slicing

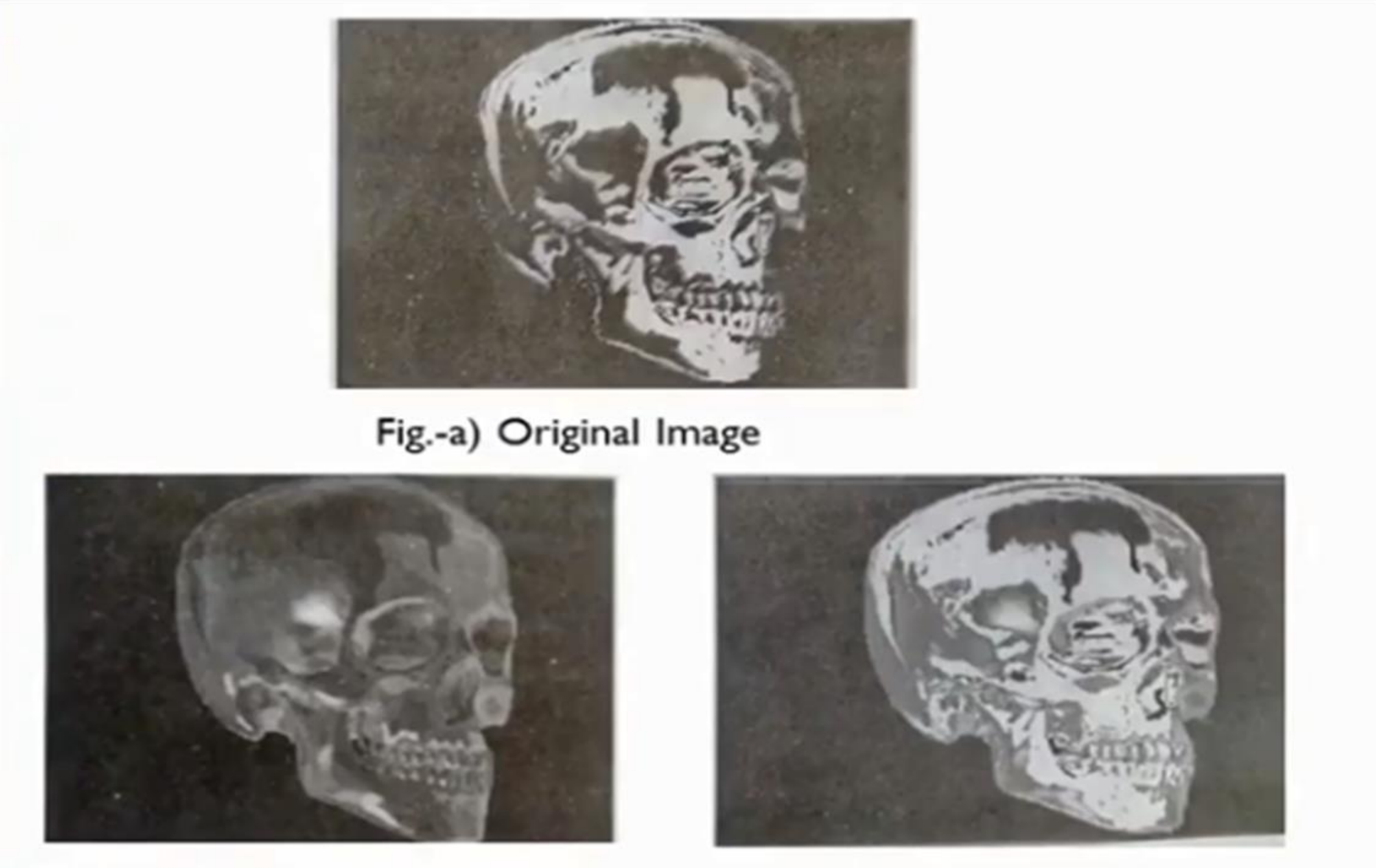


Fig.-a) Original Image

a. Gray level slicing w/o background   b. Gray level slicing with background

Problem

- Perform gray level slicing with Background and Without Background on the 3 bit image shown below. (Assume A =2 and B= 4).

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Solution : Without Background

- s = 7 if 2 ≤ r ≤ 4

  = 0 otherwise

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Solution : Without Background

- $s = 7$ if $2 \leq r \leq 4$

  $= 0$ otherwise

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Solution : Without Background

- s = 7 if 2 ≤ r ≤ 4

  = 0 otherwise

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

0	7	7	7	7
7	7	7	0	7
7	0	0	0	0
7	0	0	0	0
0	7	7	7	0

Solution : With Background

- $s = 7$ if $2 \leq r \leq 4$

  $= r$ otherwise

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Solution : With Background

- $s = 7$ if $2 \le r \le 4$

  $= r$ otherwise

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Solution : With Background

- s = 7 if 2 ≤ r ≤ 4

  = r otherwise

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

1	7	7	7	7
7	7	7	5	7
7	6	6	7	0
7	6	6	5	1
0	7	7	7	1

Additional Problems:

eg Intensity level slicing / Gray level slicing

$r_1 = 2 \qquad r_2 = 5$

4	2	3	0
1	3	5	7
5	3	2	1
2	4	6	7

a) With background

7	7	7	0
1	7	7	7
7	7	7	1
7	7	6	7

$s = L-1 \qquad r_1 \le r \le r_2$

$= 7$

$s = r$ other.

b) W/o background

7	7	7	0
0	7	7	0
7	7	7	0
7	7	0	0

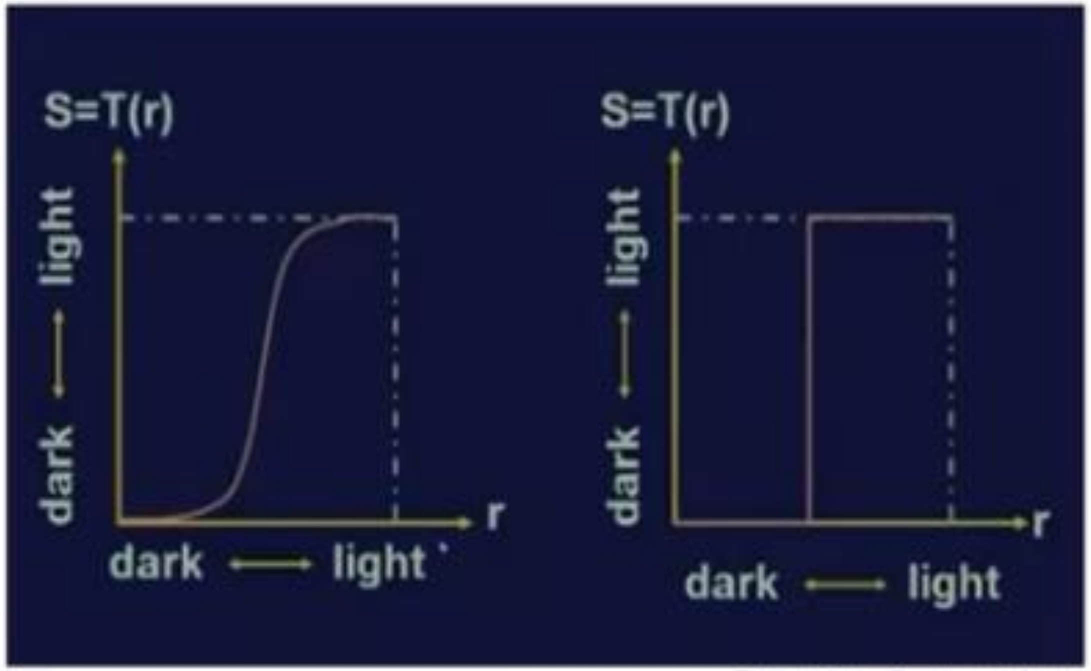
$s = L-1 \qquad r_1 \le r \le r_2$

$= 7$

$s = 0$ otherwise

Contrast Stretching

Also known by many other names such as *contrast stretching*, *gray-level stretching*, *contrast adjustment*, and *amplitude scaling*)

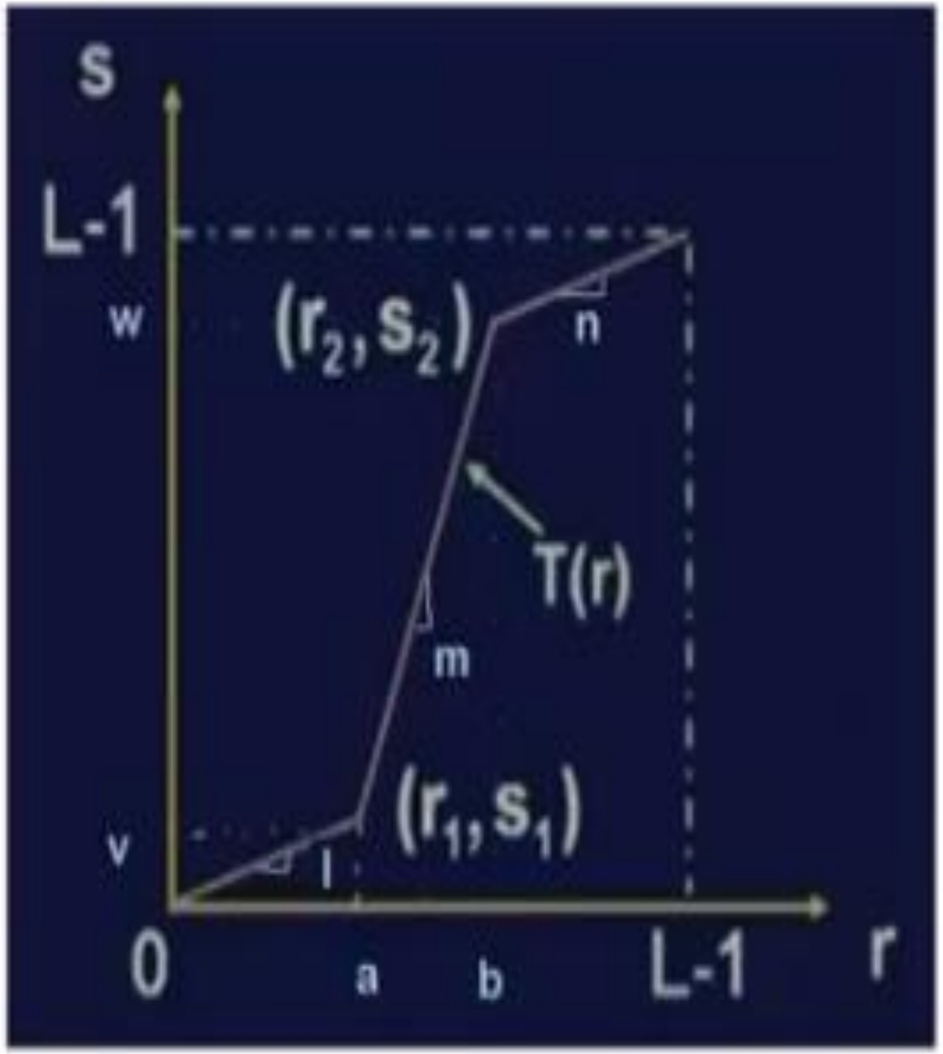


Contrast Stretching

S=l.r 0<=r<a

S=m.(r-a)+v a<=r<b

S=n.(r-b)+w b<=r<L-1



Contrast Stretching

(a) Example of an image whose original gray-level range was [90, 162];

(b) The result of applying the auto contrast transformation

(a) (b)

Contrast Stretching

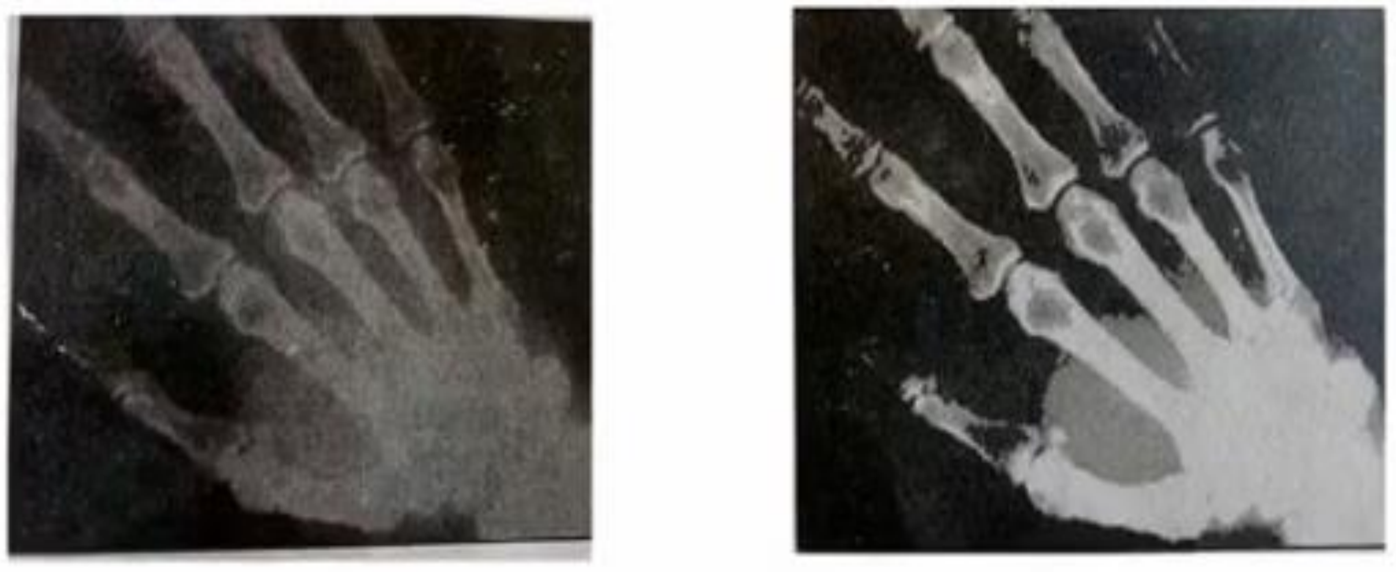


Fig.-a) Original Image Fig.-b) Contrast Stretched image

Contrast stretching ⇒

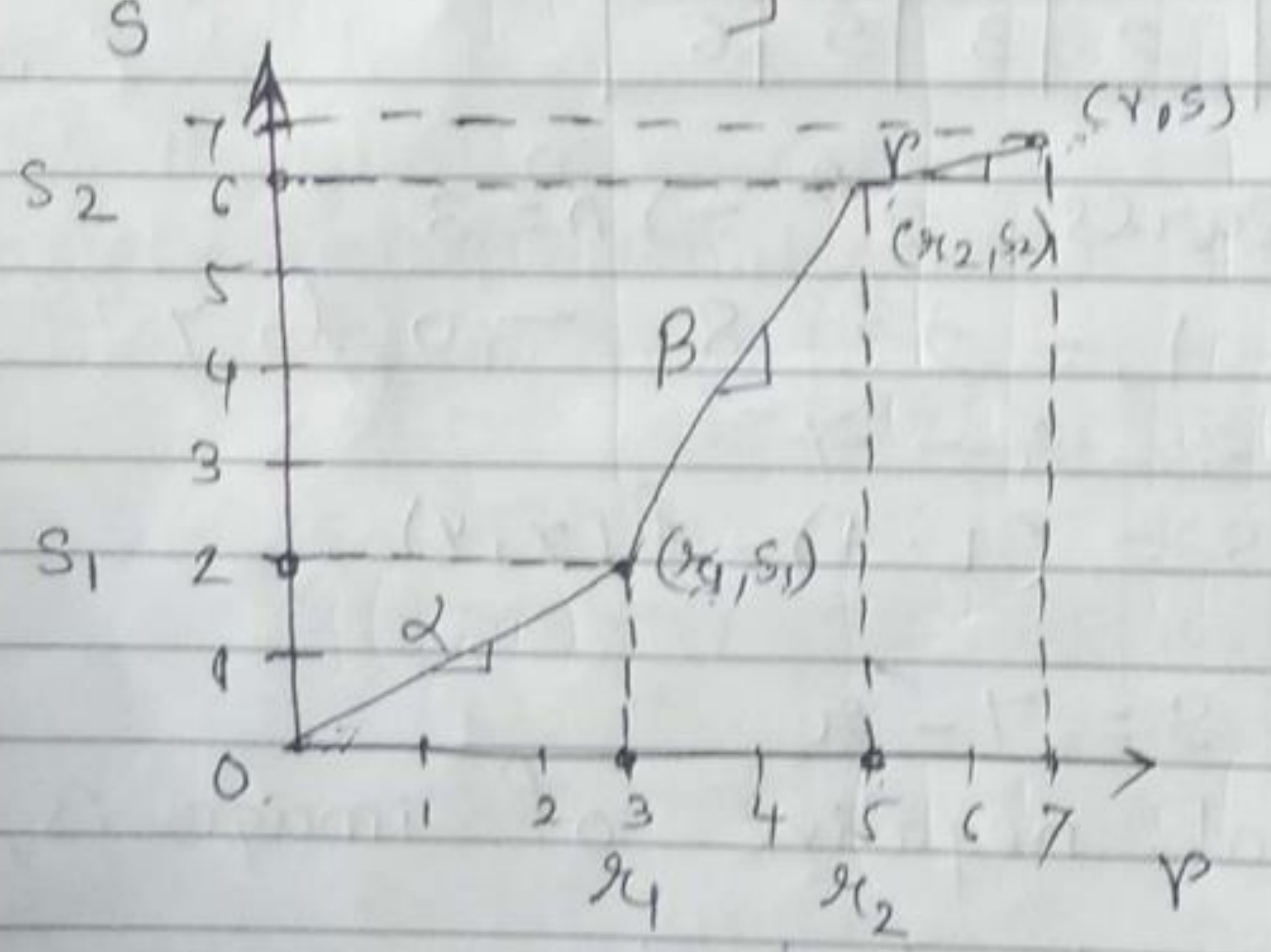
eg. ~~r_2/t_B~~ For the following image find contrast stretching.

$r_2 = 5, \; r_1 = 3, \; S_2 = 6, \; S_1 = 2$

$f(x, y) =$

4	3	2	1
3	1	2	4
5	1	6	2
2	3	5	6.

Contrast stretching transformation is as shown in fig.



$$\alpha = \frac{y_2 - y_1}{x_2 - x_1} = \frac{S1}{94} = 2/3 = 0.66$$

$$\beta = \frac{y_2 - y_1}{x_2 - x_1} = \frac{6-2}{5-3} = 4/2 = 2$$

$$\gamma = \frac{y_2 - y_1}{x_2 - x_1} = \frac{7-6}{7-5} = 1/2 = 0.5$$

Contrast stretching formula as below

$$s = \alpha \cdot r \qquad 0 \le r < 3$$

$$s = \beta \cdot (r - r_1) + s_1 \qquad 3 \le r < 5$$

$$s = \gamma (r - r_2) + s_2 \qquad 5 \le r \le 7$$

$r_1 = 3$, $r_2 = 5$, $s_1 = 2$, $s_2 = 6$

$\alpha = 0.66$, $\beta = 2$, $\gamma = 0.5$

r	S	
0	$S = \alpha . r$	$= 0.66 \times 0 = 0$
1.	$S = \alpha . r$	$= 0.66 \times 1 = 0.66$
2	$S = \alpha . r$	$= 0.66 \times 2 = 1.32$
3	$S = \beta (r - r_1) + S_1$	$= 2(3-3)+2 = 2$
4	$S = \beta (r - r_1) + S_1$	$= 2(4-3)+2 = 4$
5	$S = \gamma (r - r_2) + S_2$	$= 0.5(5-5)+6 = 6$
6.	$S = \gamma (r - r_2) + S_2$	$= 0.5(6-5)+6 = 6.5$
7.	$S = \gamma (r - r_2) + S_2$	$= 0.5(7-5)+6 = 7$

Original Image:

$f(x,y) =$

4	3	2	1
3	1	2	4
5	1	6	2
2	3	5	6

Hence contrast stretched image is,

4	2	1.32	0.66
2	0.66	1.32	4
6	0.66	6.5	1.32
1.32	2	6	6.5

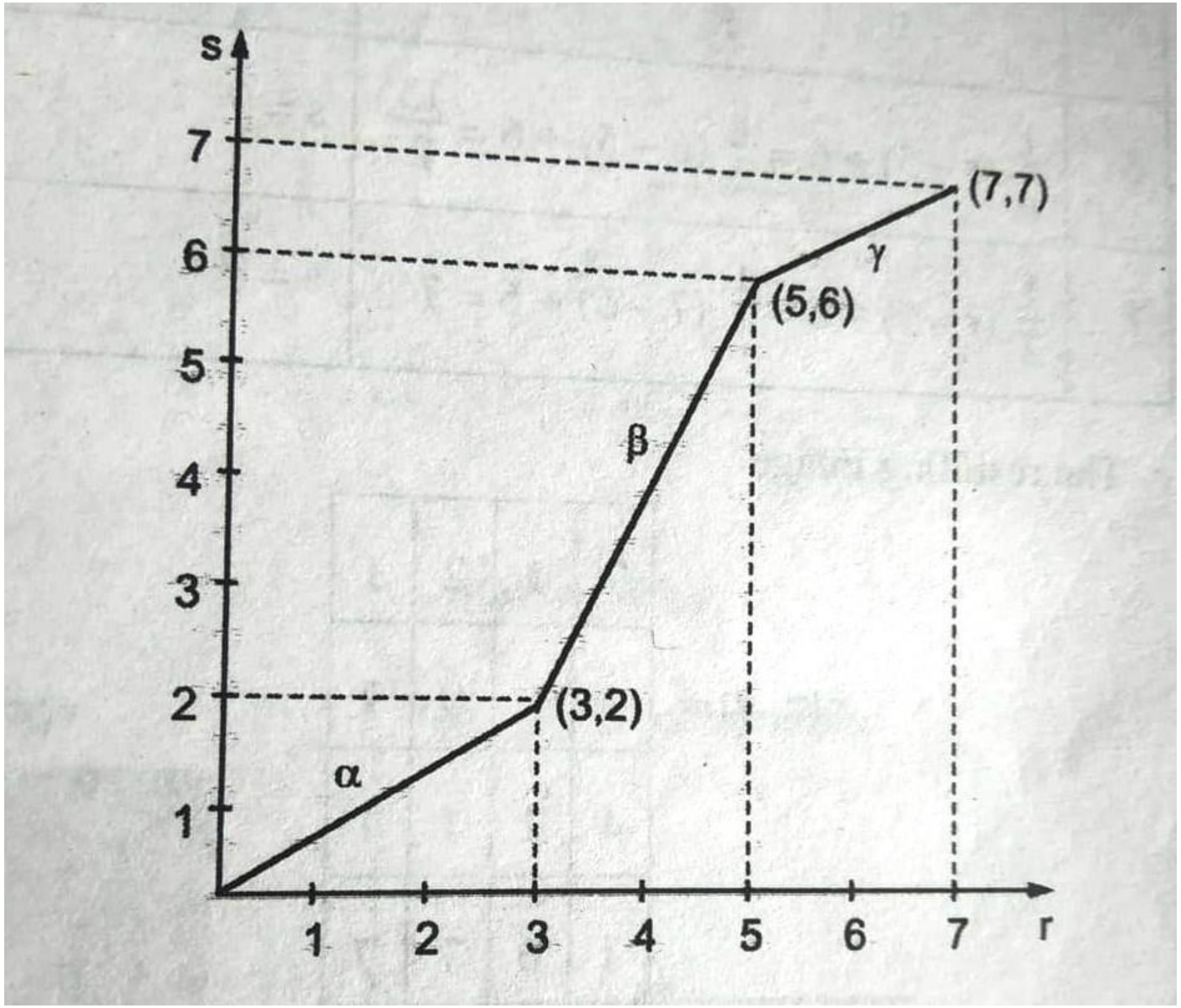
$\xrightarrow{\text{roundoff}}$

4	2	1	1
2	1	1	4
6	1	7	1
1	2	6	7

Problem:

For the 3BPP 4×4 size image perform contrast stretching $r_1 = 3_1$, $s_1 = 2$ and $r_2 = 5$, $s_2 = 6$

6	2	3	2
1	5	0	7
4	3	2	1
2	5	7	6


s
7
6
5
4
3
2
1
(7,7)
γ
(5,6)
β
(3,2)
α
1
2
3
4
5
6
7
r

$$\text{Slope} = \frac{y_1 - y_2}{x_1 - x_2}$$

$$\alpha = \frac{2-0}{3-0} = \frac{2}{3}$$

$$\beta = \frac{6-2}{5-3} = 2$$

$$\gamma = \frac{7-6}{7-5} = \frac{1}{2}$$

$$S = \begin{cases} \frac{2}{3}r & 0 \le r < 3 \\ 2(r-3)+2 & 3 \le r < 5 \\ \frac{1}{2}(r-5)+6 & 5 \le r \le 7 \end{cases}$$

Using the above transformation function

r	s	Rounding off
0	$\frac{2}{3}\ r = \frac{2}{3}(0) = 0$	s = 0
1	$\frac{2}{3}\ r = \frac{2}{3}(1) = \frac{2}{3}$	s = 1
2	$\frac{2}{3}\ r = \frac{2}{3}(2) = \frac{4}{3}$	s = 1
3	$2(r - 3) + 2 = 2(3 - 3) + 2 = 2$	s = 2
4	$2(r - 3) + 2 = 2(4 - 3) + 2 = 4$	s = 4
5	$\frac{1}{2}\ (r - 5) + 6 = \frac{1}{2}\ (5 - 5) + 6 = 6$	s = 6

r	S	Rounding off
6	$\frac{1}{2}(r-5)+6=\frac{1}{2}(6-5)+6=\frac{13}{2}$	s = 7
7	$\frac{1}{2}(r-5)+6=\frac{1}{2}(7-5)+6=7$	s = 7

∴ The resulting image

$$g(x, y) = \begin{array}{|c|c|c|c|} \hline 7 & 1 & 2 & 1 \\ \hline 1 & 6 & 0 & 7 \\ \hline 4 & 2 & 1 & 1 \\ \hline 1 & 6 & 7 & 7 \\ \hline \end{array}$$

Bit plane slicing

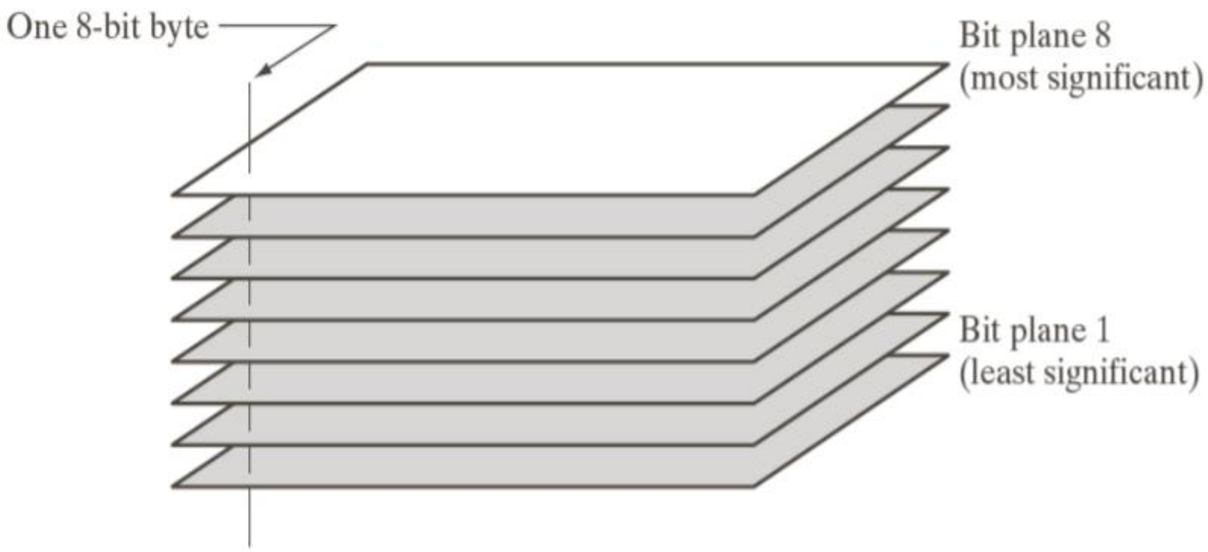
- Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image
- Higher-order bits usually contain most of the significant visual information
- Lower-order bits contain subtle details

Bit plane Slicing

- Pixels are digital numbers, each one composed of bits.
- This method is useful in image compression.
- Most significant bits contain the majority of visually significant data.

Bit plane Slicing

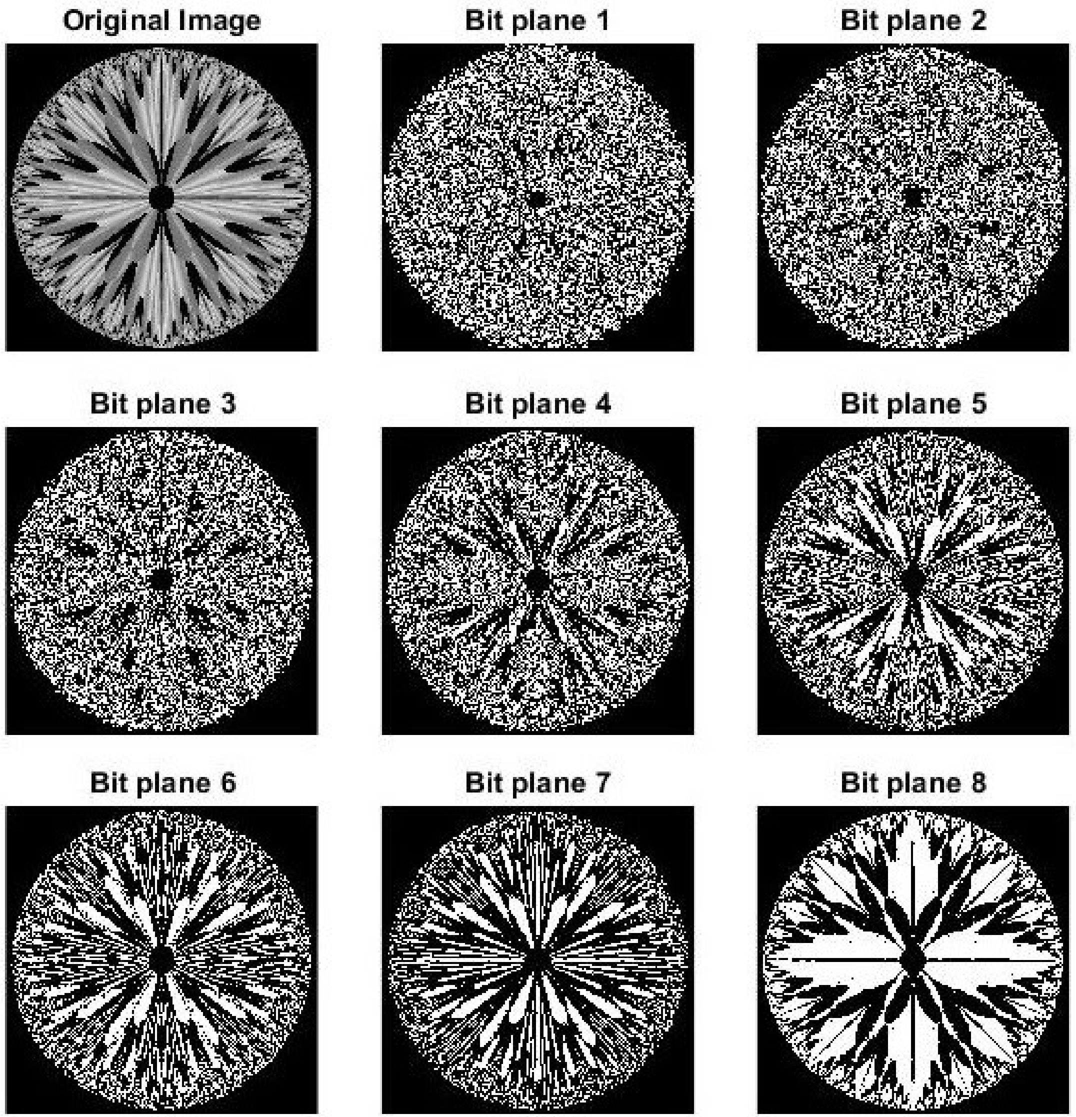
- For an 8-bit gray level image, the image is composed of 8 1-bit planes as shown below:



Bit plane Slicing

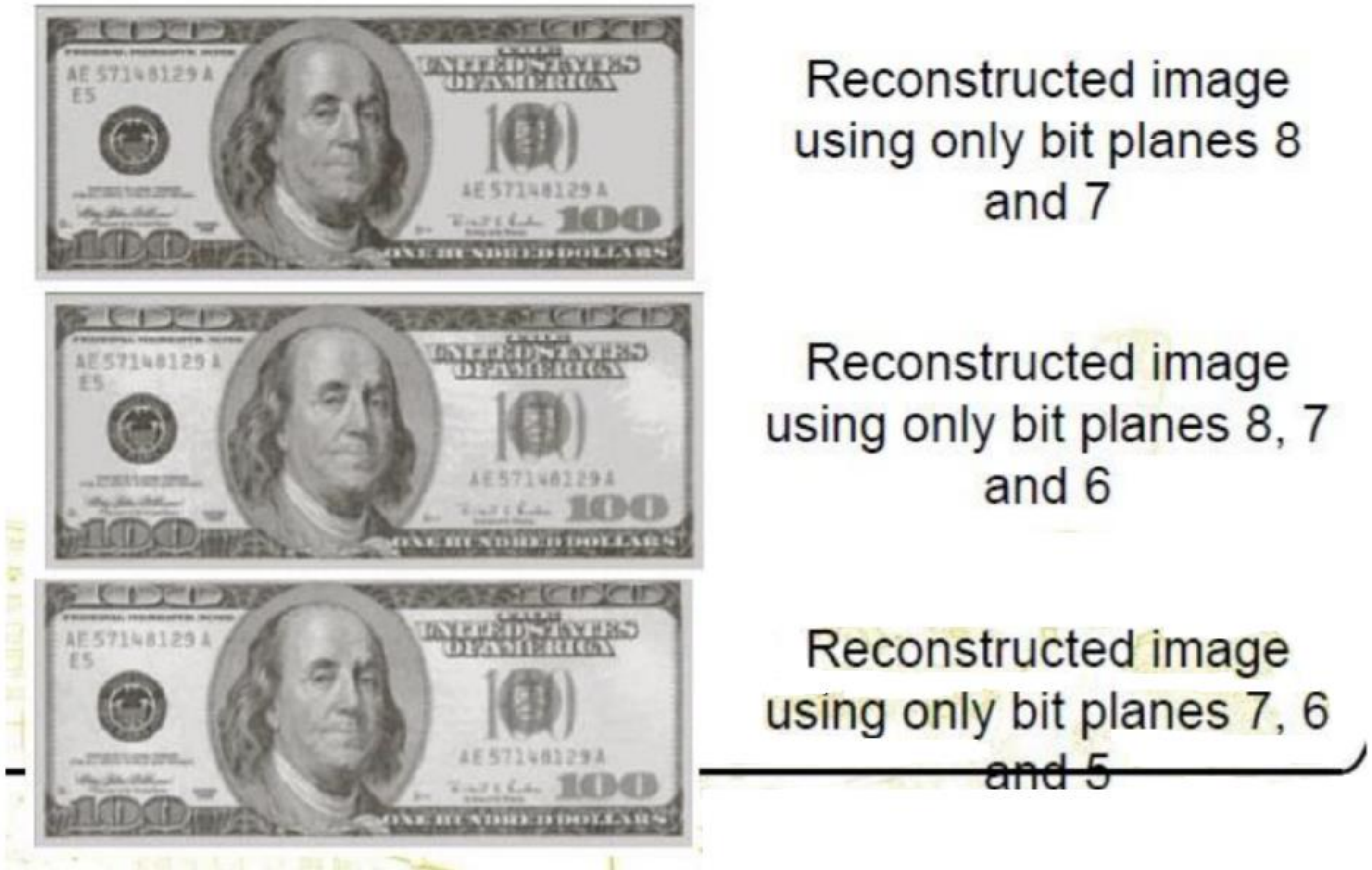
- Plane 0 contains the least significant bit and plane 7 contains the most significant bit.
- In terms of 8-bits bytes, plane 0 contains all lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all high order bits.

Bit plane Slicing : Example

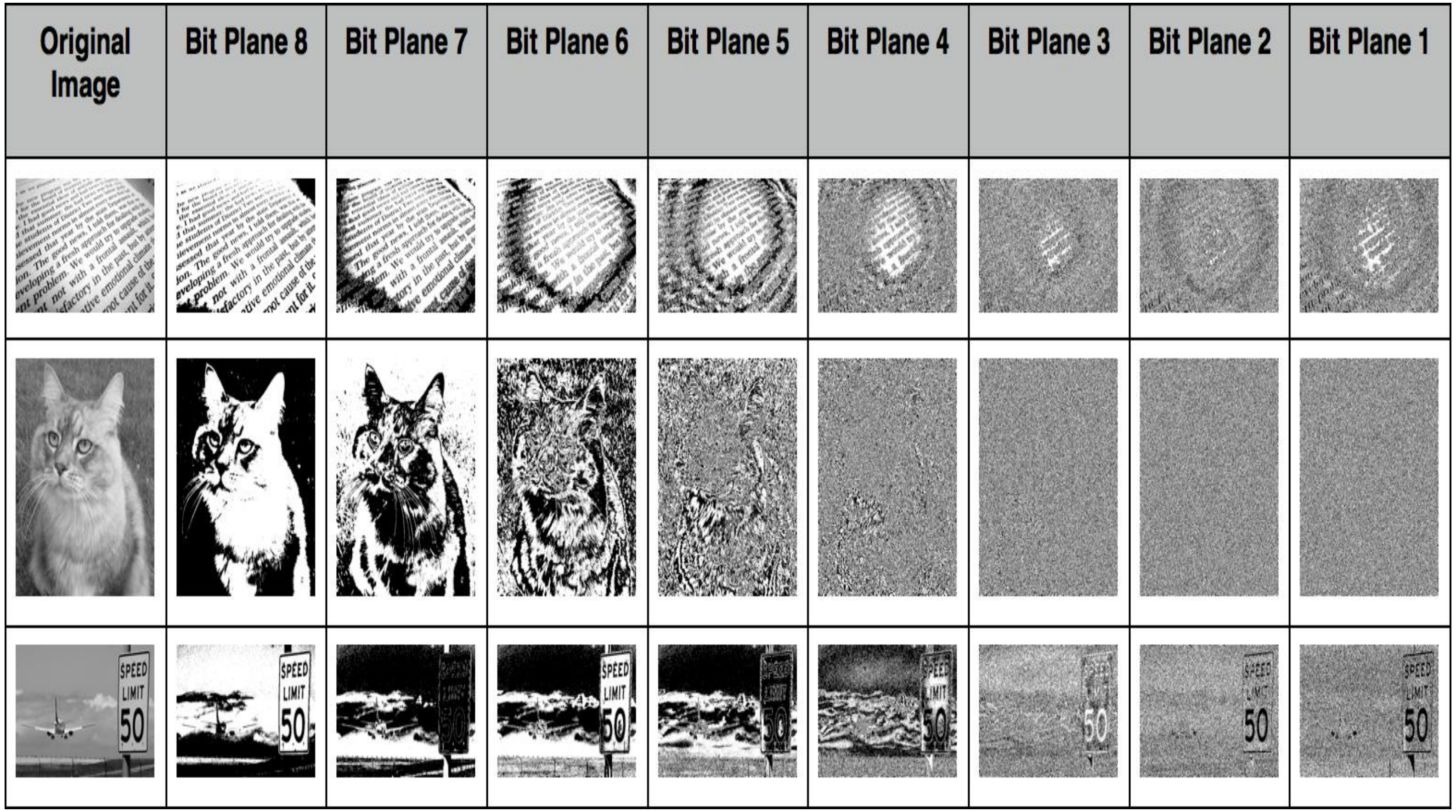


Bit plane Slicing : Example

Reconstruction using Most Significant Bits



Bit plane slicing



Problem:

- Perform bit plane slicing on the 3PP image

0	4	5
0	5	6
7	7	6

000	100	101
000	101	110
111	111	110

0	1	1
0	1	1
1	1	1

LSB

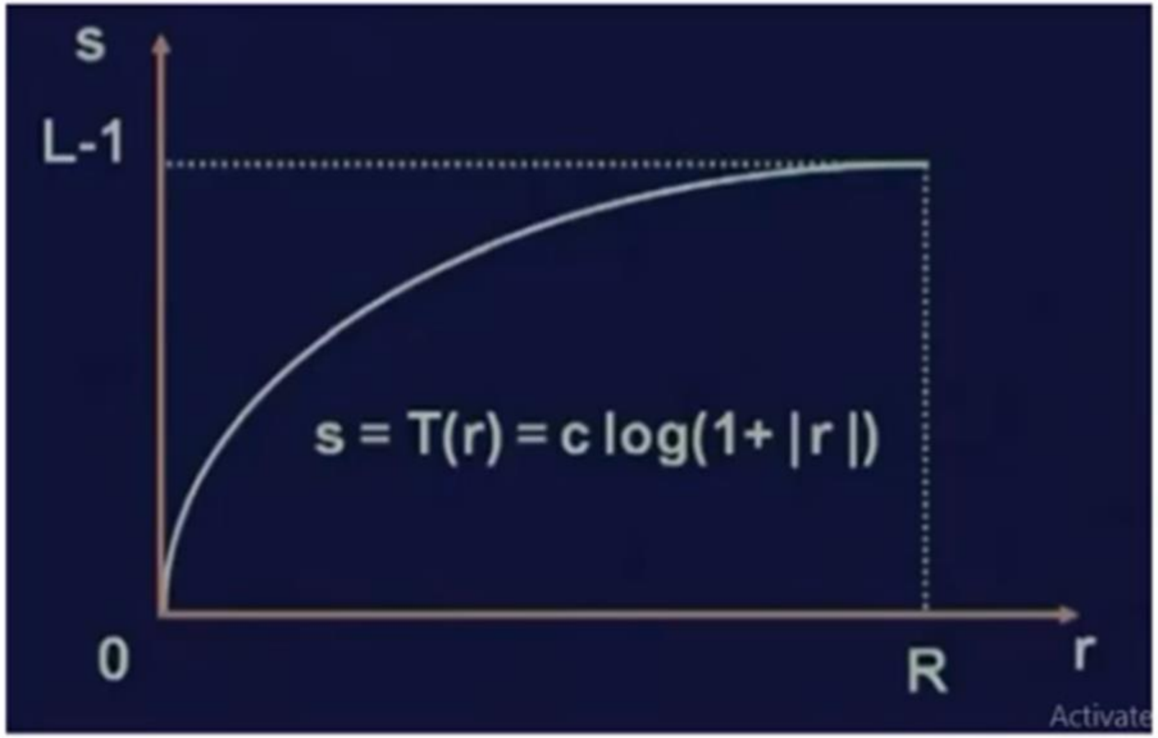
0	0	0
0	0	1
1	1	1

0	0	1
0	1	0
1	1	0

MSB

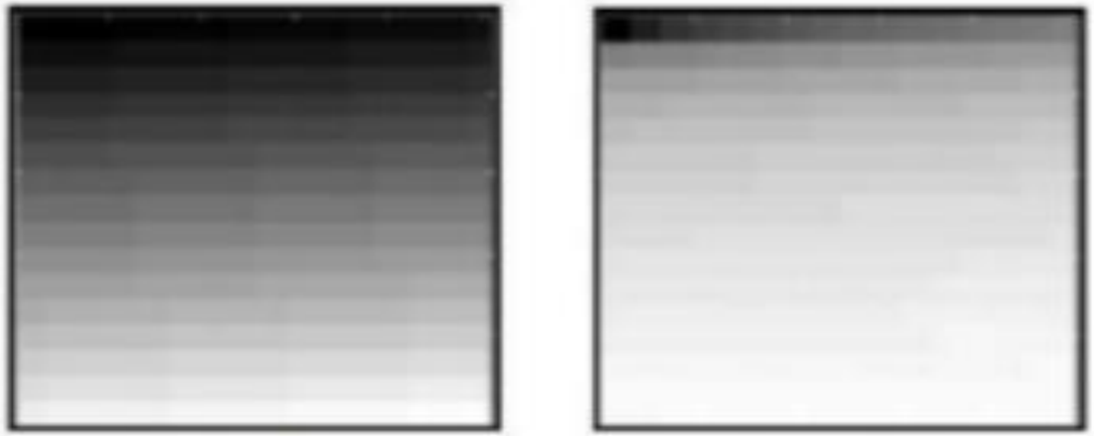
Dynamic range compression (Log Transformation)

- The purpose of dynamic range compression is **to map the natural dynamic range of a signal to a smaller range**.
- It is difference between darkest and brightest tone (pure black and pure white)



6-Range compression

- Sometimes the dynamic range of a processed image far exceeds the capability of the display device, in which case only the brightest parts of the images are visible on the display screen.



Original   Processed output

- An effective way to compress the dynamic range of pixel values is to perform the following intensity transformation function:

$$s = c \log(1+|u|)$$

where c is a scaling constant, and the logarithm function performs the desired compression.

Problem:

$$s = c \log(1 + |u|)$$

128	212	233
54	62	124
140	152	156

Original Image

C= 1

S= 1. log (1+ 128)= log (129)
S= 1. log (1+ 212)= log (213)
S= 1. log (1+ 233)= log (234)

.
.
.

2.11	2.32	2.36
1.73	1.79	2.09
2.14	2.18	2.19

Round →

2	2	2
2	2	2
2	2	2

Power-Law Transformations

- Power-law transformations have the basic form of:

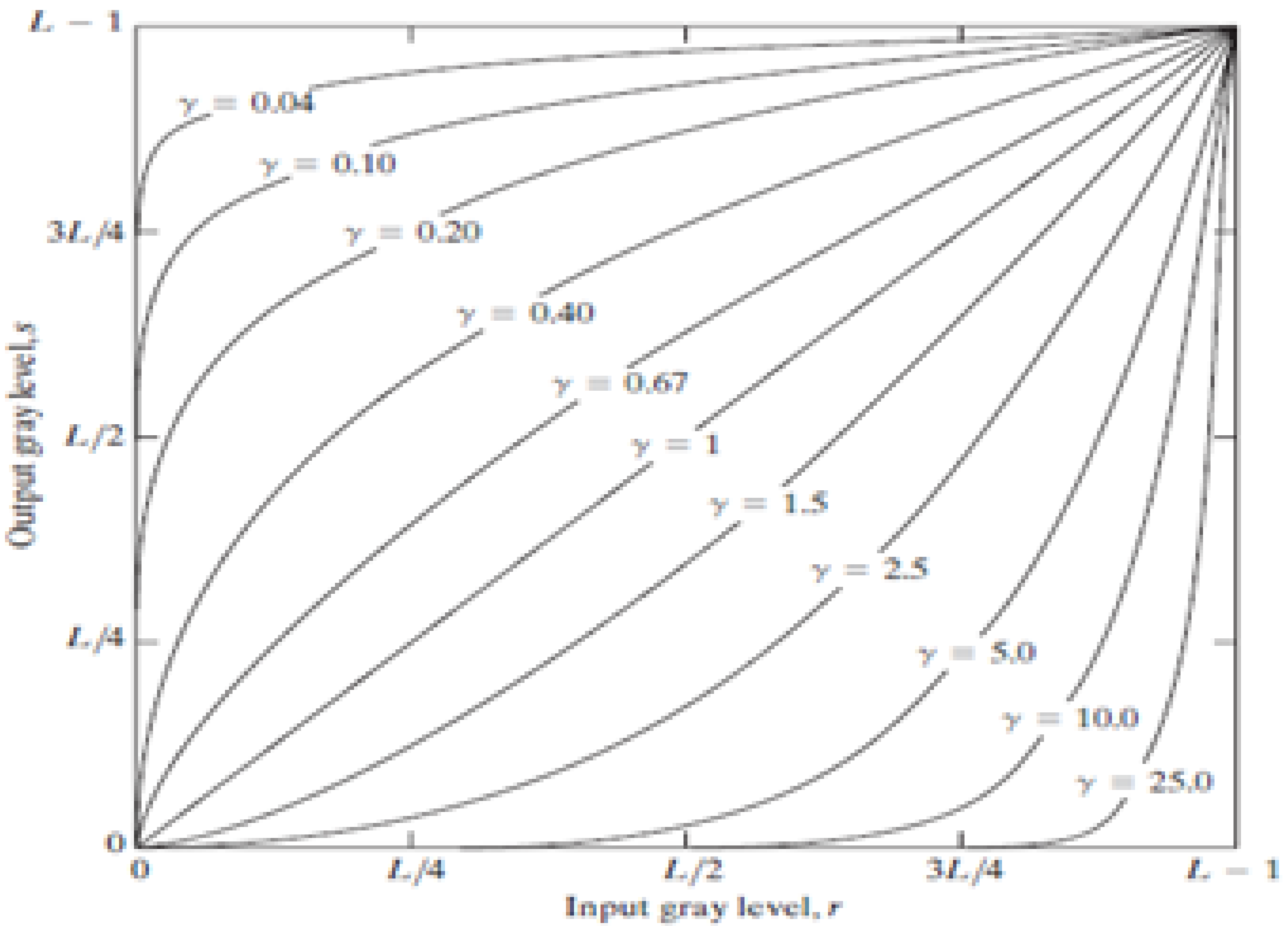
$$s = c.r^{\gamma}$$

Where c and γ are positive constants

γ is gamma correction factor.

Power-Law Transformations

- Different transformation curves are obtained by varying γ (gamma)



Power-Law Transformations

- Non-linearities encountered during image capturing, printing and displaying can be corrected using gamma correction.
- Improves dynamic range of an image.

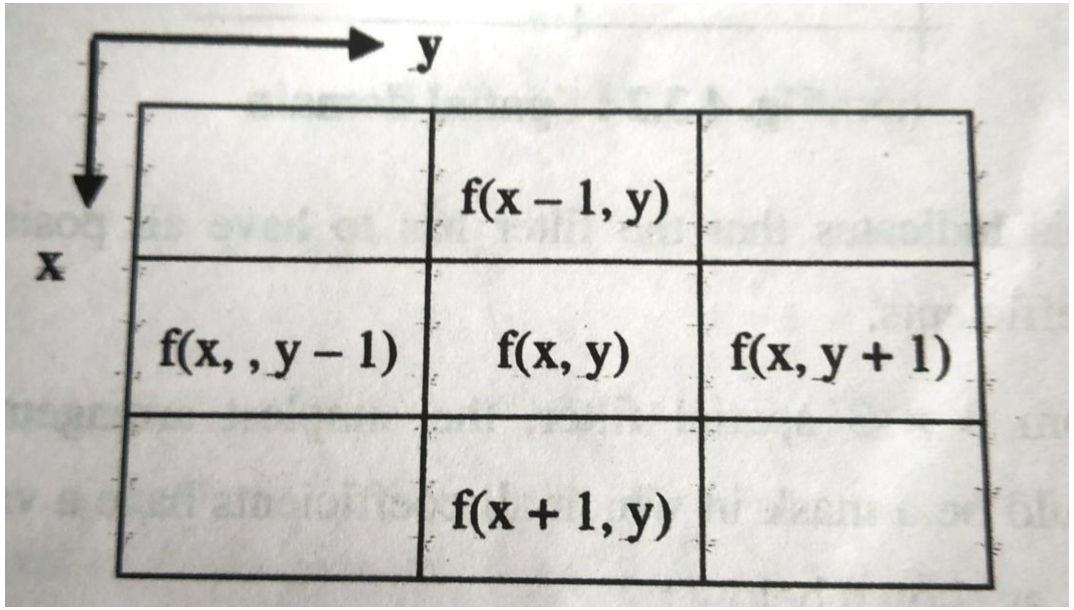
Power-Law Transformations

Examples of gamma correction for two different values of γ: 0.5 (left) and 2.2 (right)

Neighborhood Processing

Neighborhood Processing

- In point processing, a pixel's gray value is changed without any knowledge of its surrounding pixel values.
- In neighborhood processing, the intensity of the neighboring pixels are taken into account while performing an operation on the pixel of interest.

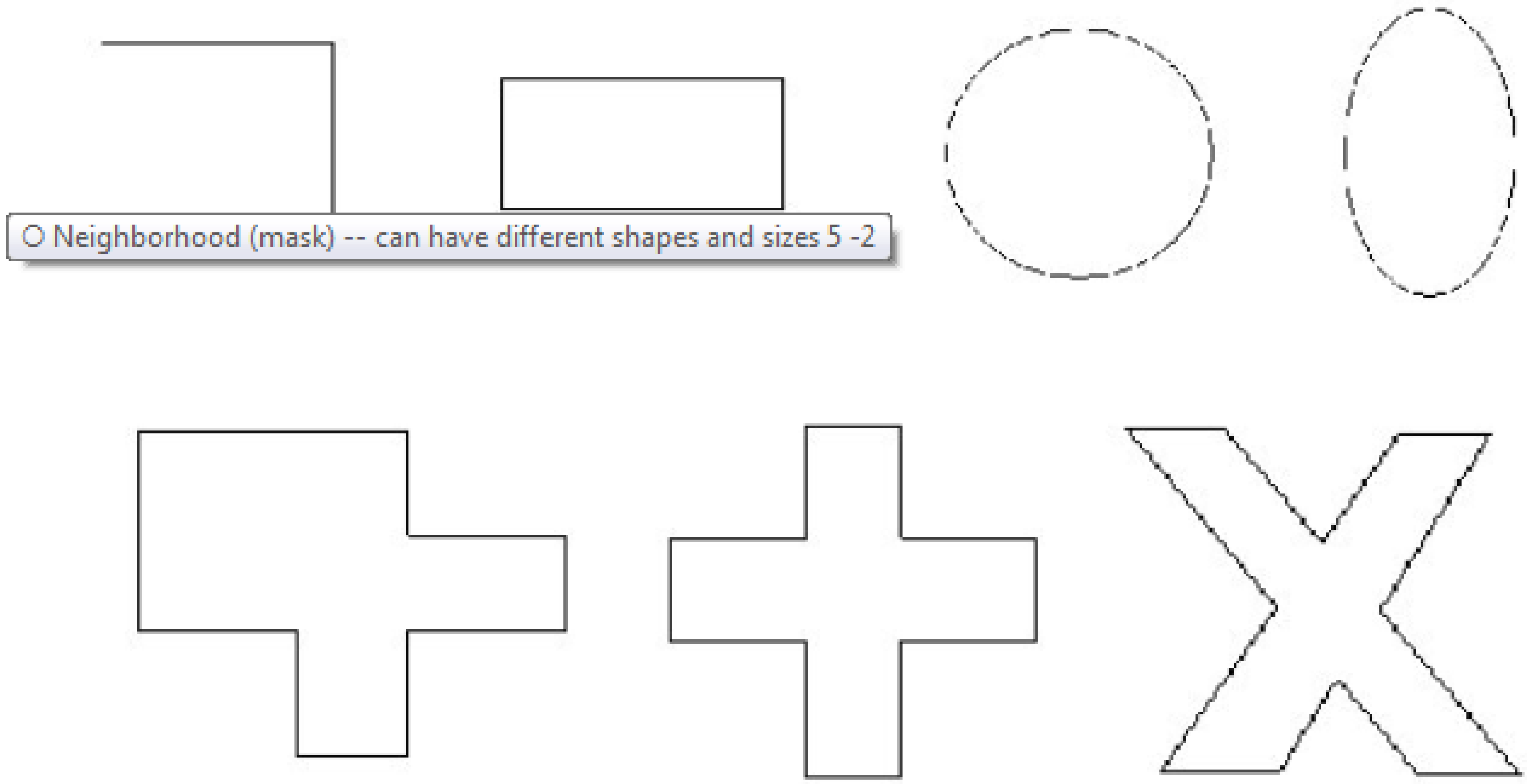


Neighborhood Processing

- **Neighborhood Processing are used in filtering.**
- The idea is to move a “mask” over the given image.
- As we do this, we create a new image whose pixels have grey values calculated from the grey values under the mask.
- The combination of mask and function is called a **filter.**
- If the function by which the new grey value is calculated is a linear function of all the grey values in the mask, then the filter is called a linear filter.

- Neighborhood (mask)

  -- can have different shapes and sizes




ned Owl, ©Dan Sudia

Some filters are:

- Low –pass filter **(smoothing)**
  - ✓ Low-pass averaging filtering
  - ✓ Low-pass Median filtering

- High-pass filtering
- High-Boost filtering

Low Pass Filters (Smoothing)

- Low pass filtering is used to remove noise present in the image. Noise is a high frequency signal and hence low pass filters eliminate them.
- A low pass filter is the basis for most **smoothing** methods.
- An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels.
- Using a low pass filter tends to retain the low frequency information within an image while reducing the high frequency information.

Types of Low pass filter

- Low –pass filter **(smoothing)**
  - ✓ Low-pass averaging filtering
  - ✓ Low-pass Median filtering

Low-pass averaging filter (Mean Filtering)

- The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself.
- This has the effect of eliminating pixel values which are unrepresentative of their surroundings.

Low-pass averaging filtering

- Often a 3×3 square kernel is used, as shown below:

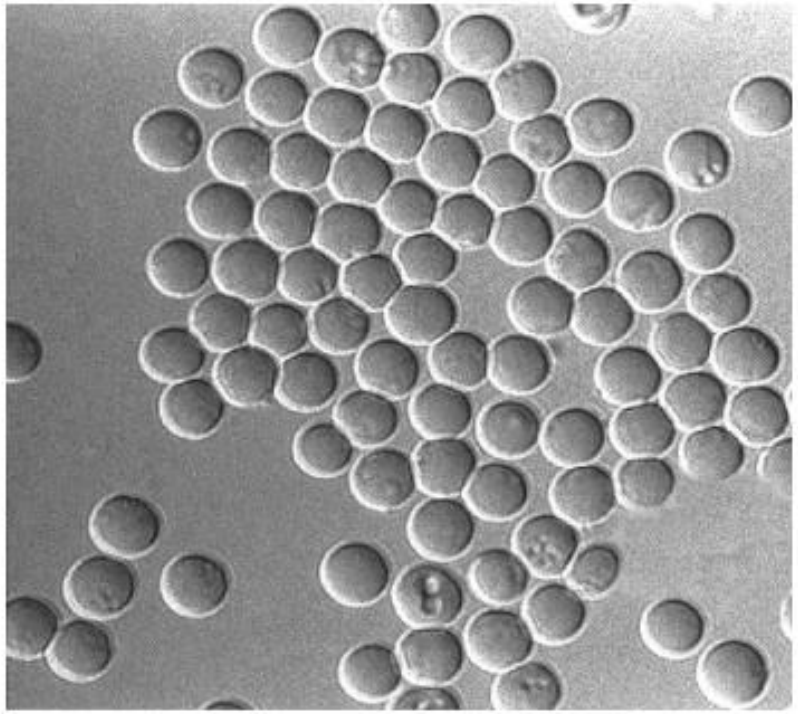
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$1/25 \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

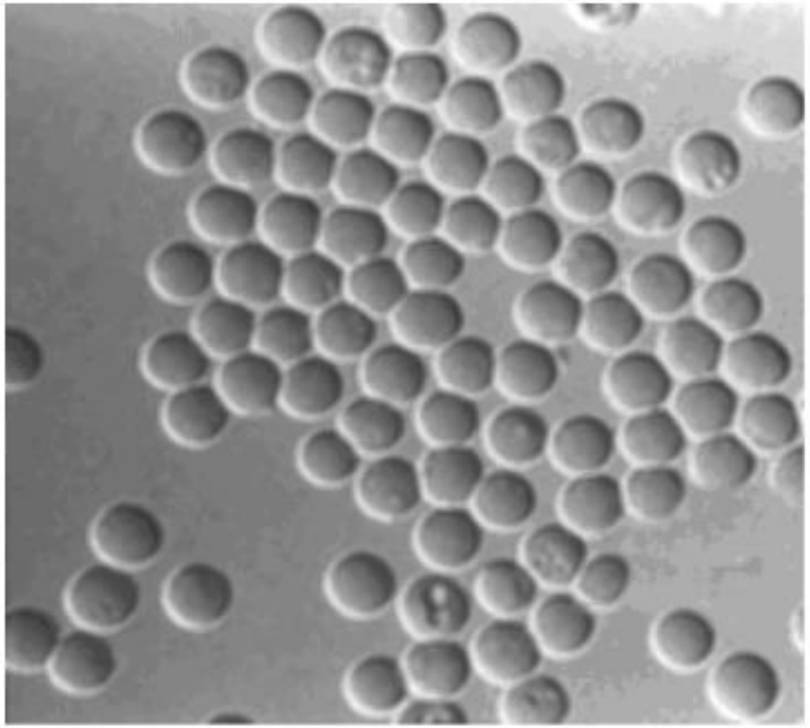
Input 3X3 5X5 7X7

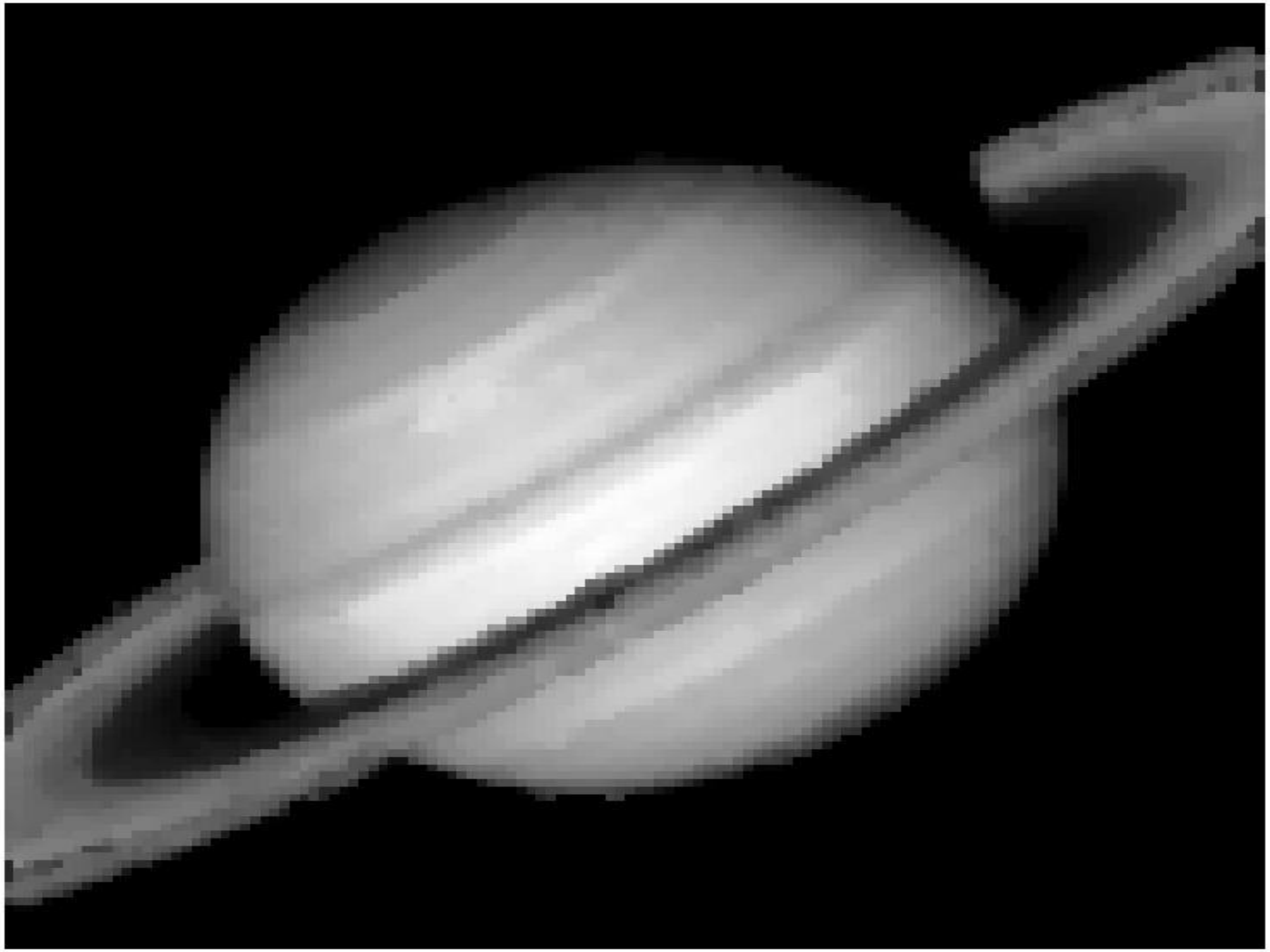
Example:

Original RBC Image

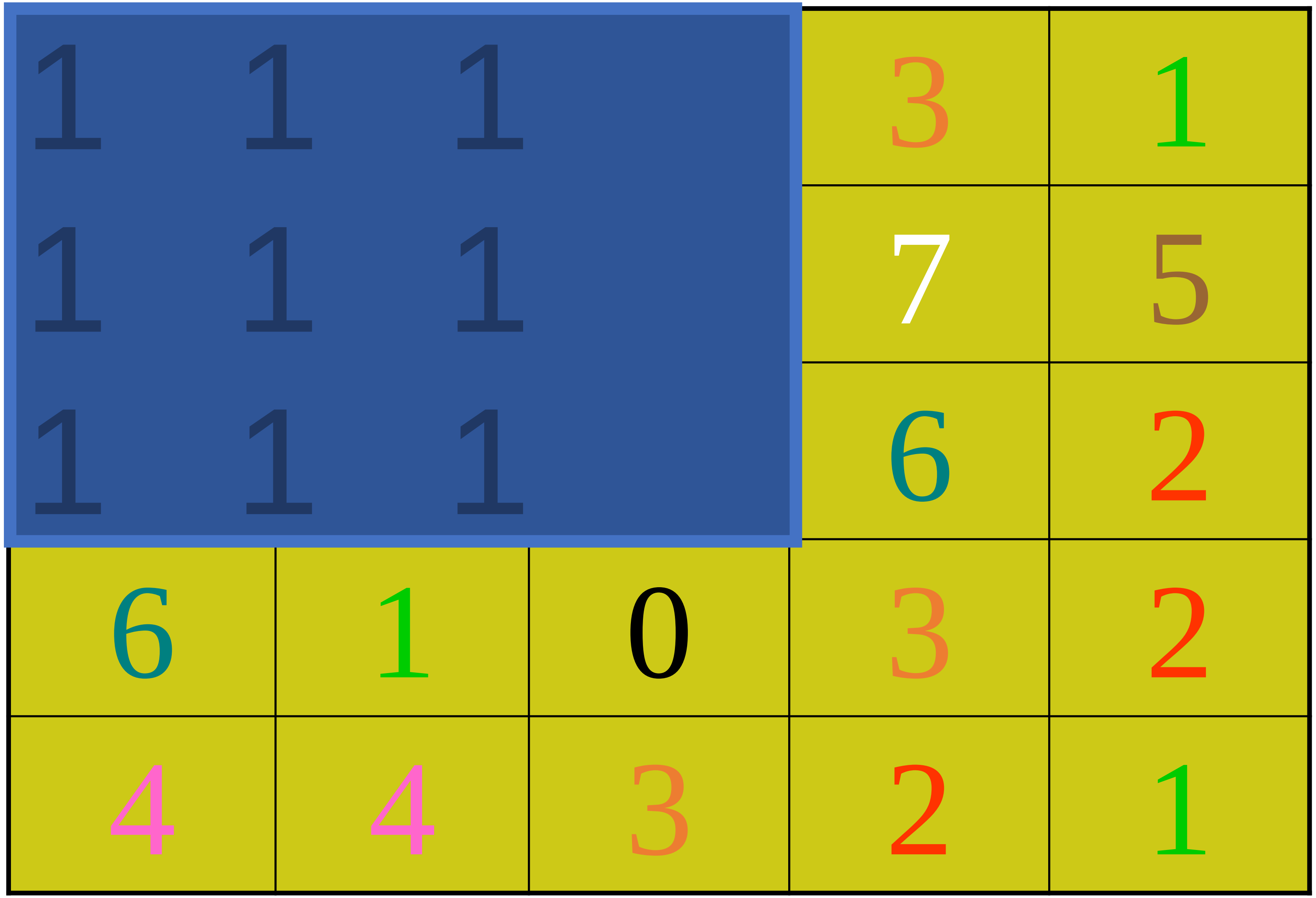


Average-Smoothed RBC Image





Problem:



3	7	1	3	1
2	5	1	7	5
5	5	2	6	2
6	1	0	3	2
4	4	3	2	1

3	7	1	3	1
2	3.44	1	7	5
5	5	2	6	2
6	1	0	3	2
4	4	3	2	1

3	7	1	3	1
2	3.44	3.93	7	5
5	5	2	6	2
6	1	0	3	2
4	4	3	2	1

3	7	1	3	1
2	3.44	3.93	3.34	5
5	5	2	6	2
6	1	0	3	2
4	4	3	2	1

3	7	1	3	1
2	3.44	3.93	3.34	5
5	3.15	2	6	2
6	1	0	3	2
4	4	3	2	1

3	7	1	3	1
2	3.44	3.93	3.34	5
5	3.15	2.87	6	2
6	1	0	3	2
4	4	3	2	1

3	7	1	3	1
2	3.44	3.93	3.34	5
5	3.15	2.87	3.12	2
6	1	0	3	2
4	4	3	2	1

3	7	1	3	1
2	3.44	3.93	3.34	5
5	3.15	2.87	3.12	2
6	3.22	0	3	2
4	4	3	2	1

3	7	1	3	1
2	3.44	3.93	3.34	5
5	3.15	2.87	3.12	2
6	3.22	2.7	3	2
4	4	3	2	1

3	7	1	3	1
2	3.44	3.93	3.34	5
5	3.15	2.87	3.12	2
6	3.22	2.7	2.41	2
4	4	3	2	1

3	7	1	3	1
2	3	4	3	5
5	3	3	3	2
6	3	3	2	2
4	4	3	2	1

- **Disadvantages :**

  In this technique, the border pixel are not considered at all.

  Solution:

  Use the following two techniques where border pixels are considered

  (i)Pixel Replication

  (ii) Zero padding

Problem

- Filter the following image using 3 x 3 neighborhood averaging by assuming (i)Pixel Replication (ii) Zero padding

10	20	30
10	120	30
10	20	30

(i)Pixel Replication

Solution:

- After pixel replication:

10	20	30
10	120	30
10	20	30

10	10	20	30	30
10	10	20	30	30
10	10	120	30	30
10	10	20	30	30
10	10	20	30	30

Solution:

- After applying mask

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

10	10	20	30	30
10	10	20	30	30
10	10	120	30	30
10	10	20	30	30
10	10	20	30	30

Solution:

- After applying mask

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

24.4	31.1	37.8
24.4	31.1	37.8
24.4	31.1	37.8

Roundoff →

24	31	38
24	31	38
24	31	38

Solution:

- Approximating:

24	31	38
24	31	38
24	31	38

(ii) Zero Padding

Solution:

- After zero padding:

0	0	0	0	0
0	10	20	30	0
0	10	120	30	0
0	10	20	30	0
0	0	0	0	0

Solution:

- After applying mask

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0	0	0	0	0
0	10	20	30	0
0	10	120	30	0
0	10	20	30	0
0	0	0	0	0

Solution:

- After applying mask

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

17.7	24.4	22.2
21.1	31.1	27.7
17.7	24.4	22.2

Roundoff

18	24	22
21	31	28
18	24	22

Solution:

- Approximating:

18	24	22
21	31	28
18	24	22

Example

Filter the following image using 3*3 neighborhood averaging by using

1) **Zero padding**
2) **Pixel replication**

1	2	3	2
4	2	5	1
1	2	6	3
2	4	6	7

Solution- using Zero padding

0	0	0	0	0	0
0	1	1.88	1.66	1.22	0
0	1.33	2.88	2.88	2.22	0
0	1.66	3.55	4	3.11	0
0	1	2.33	3.11	2.44	0
0	0	0	0	0	0

Solution- using pixel replication

1	1	2	3	2	2
1	2	2.55	2.44	2.33	2
4	2	2.88	2.88	2.88	1
1	2.44	3.55	4	4.33	3
2	2.22	3.66	5	5.77	7
2	2	4	6	7	7

Low-pass Median filtering

Low-pass Median filtering

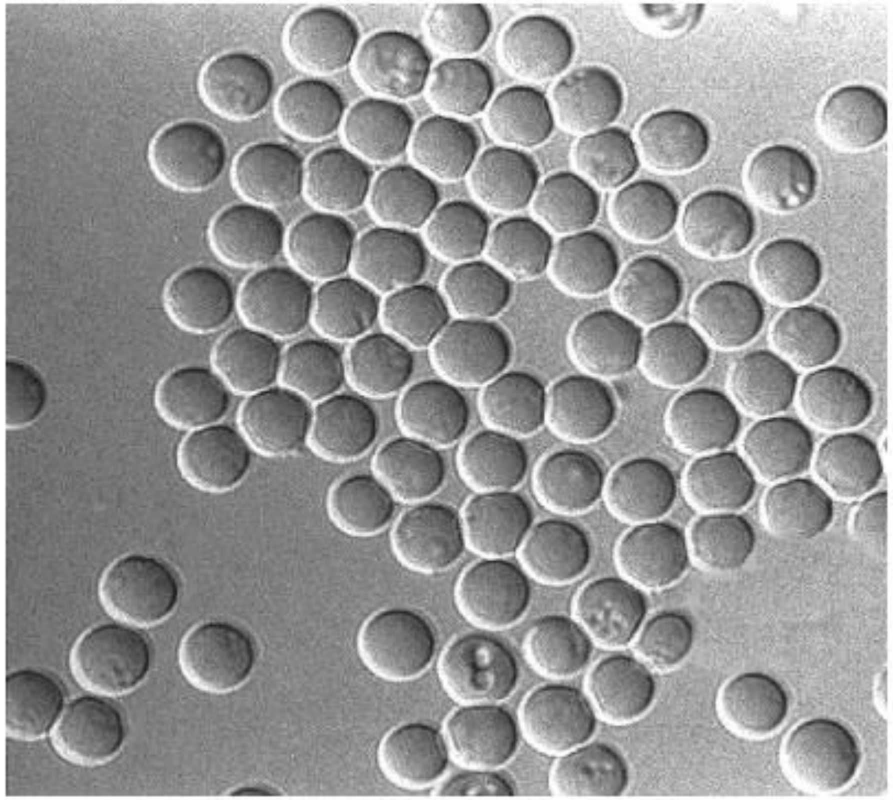
- The median filter is normally used to reduce noise in an image, somewhat like the mean filter.
- However, it often does a better job than the mean filter of preserving useful detail in the image.
- Like the mean filter, the median filter looks at its nearby neighbors to decide whether or not it is representative of its surroundings.
- Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values.

Smoothing with Median Values

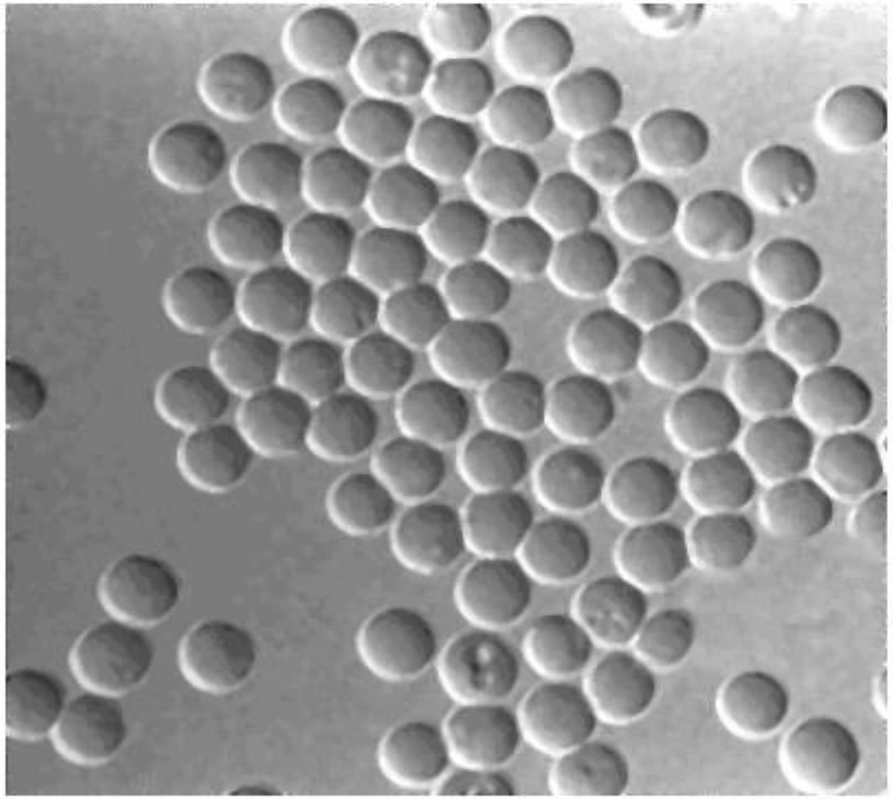
- The median is calculated by first sorting all pixel values from the neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value.
- If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.
- It is best suited for removing salt and pepper noise.

Example:

Original RBC Image



Median-Smoothed RBC Image



Problem:

- Filter the following image using 3 x 3 neighborhood median filter by assuming pixel replication

10	20	30
10	120	30
10	20	30

Solution:

10	20	30
10	120	30
10	20	30

- After pixel replication:

10	10	20	30	30
10	10	20	30	30
10	10	120	30	30
10	10	20	30	30
10	10	20	30	30

Solution:

10	10	20	30	30
10	10	20	30	30
10	10	120	30	30
10	10	20	30	30
10	10	20	30	30

- Applying median filter

Arrange numbers in ascending/descending order:

(10, 10, 10, 10, 10, 10, 20, 20, 120)

Median:

(10, 10, 10, 10, 10, 10, 10, 10, 250)

Solution

10	20	30
10	20	30
10	20	30

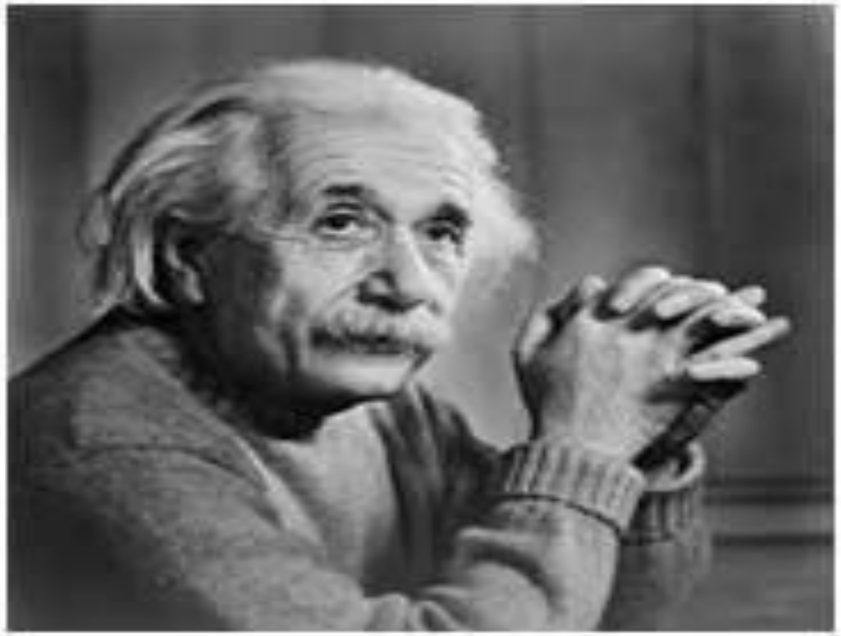
Additional Sums

Problem: Apply Low pass averaging filter on below image

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

Solution:

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	23.3	23.3	23.3	23.3	23.3	23.3	10
50	23.3	23.3	23.3	23.3	23.3	23.3	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50



Problem: Apply Low-pass Median Filtering on below image:

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	**250**	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	**250**	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

Low-pass Median Filtering

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	**250**	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	**250**	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

Low-pass Median Filtering

Arrange numbers in ascending/descending order:
(10, 10, 10, 10, 10, 10, 10, 10, 250)

Median:
(10, 10, 10, 10, 10, 10, 10, 10, 250)

Low-pass Median Filtering

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	**250**	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	**250**	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

Low-pass Median Filtering

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	**10**	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	**50**	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

High Pass Filters

- A high pass filter tends to retain the high frequency information within an image while reducing the low frequency information.
- A high pass filter is used for sharpening the image.
- An image is sharpened when contrast is enhanced between adjoining areas with little variation in brightness or darkness.
- High-pass filtering can also cause small, faint details to be greatly exaggerated.
- An over-processed image will look grainy and unnatural, and point sources will have dark donuts around them.
- So high-pass filtering can improve an image by sharpening detail, overdoing it can actually degrade the image quality significantly.

High Pass Filters

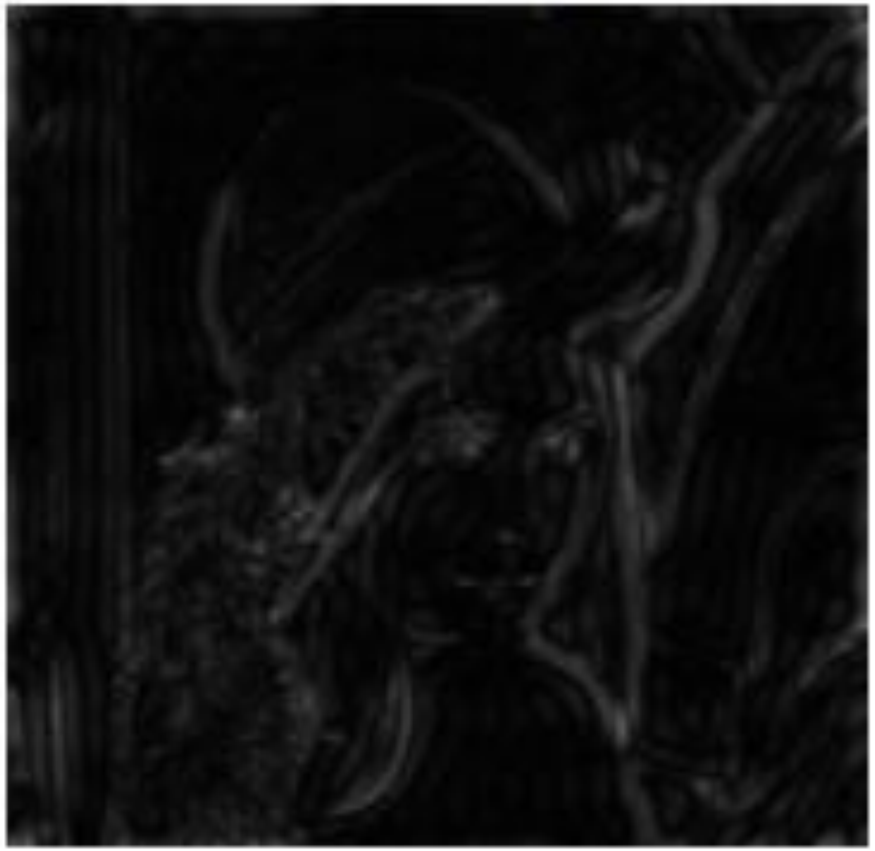
- The kernel of the high pass filter is designed to increase the brightness of the center pixel relative to neighboring pixels.
- The kernel array usually contains a single positive value at its center, which is completely surrounded by negative values.
- High-pass filtering works in exactly the same way as low-pass filtering; it just uses a different convolution kernel.
- The following array is an example of a 3 by 3 kernel for a high pass filter:

$$h = \frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Example:

Original Image

high-pass filter transform



Problem:

- Filter the following image using 3 x 3 neighborhood high pass filtering by assuming pixel replication

10	20	30
10	120	30
10	20	30

- After pixel replication:

10	10	20	30	30
10	10	20	30	30
10	10	120	30	30
10	10	20	30	30
10	10	20	30	30

Solution:

- After applying mask

$$h = \frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

10	10	20	30	30
10	10	20	30	30
10	10	120	30	30
10	10	20	30	30
10	10	20	30	30

-14.4	-11.1	-7.7
-14.4	88.9	-7.7
-14.4	-11.1	-7.7

Roundup →

0	0	0
0	89	0
0	0	0

Additional Problems: Apply High-pass filtering on the following image

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

High-pass Median filtering:

Mask=> 3×3

1/9

-1	-1	-1
-1	8	-1
-1	-1	-1

High-pass filtering

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-270	-270	-270	-270	-270	-270	-270	-270
270	270	270	270	270	270	270	270
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

High Boost Filtering

- Applying a high pass filter gets rid of the background, completely.
- There are times when we need to enhance the edges but also retain some background information.
- To do this, we use a modified version of high pass filter known as High Boost Filter.

Derivation

- High Pass = Original - Low Pass

- To pass some of the background information, we multiply the original image with a multiplicative factor A. This gives us high boost filtering

- High Boost = (A)Original - Low Pass
  = (A-1)Original + Original - Low Pass
  = (A-1)Original + High Pass

- If A =1, then High Boost = High Pass

High Boost Filtering

- If A > 1 , some of the original signal is added back which restores some background. This process is known as unsharp masking.
- The mask for high boost filter is given as follows:

A>=1
w = 9A-1

-1	-1	-1
-1	w	-1
-1	-1	-1

A=2
w = 17

-1	-1	-1
-1	17	-1
-1	-1	-1

Some other High boost filtering masks are:

0	−1	0
−1	$A + 4$	−1
0	−1	0

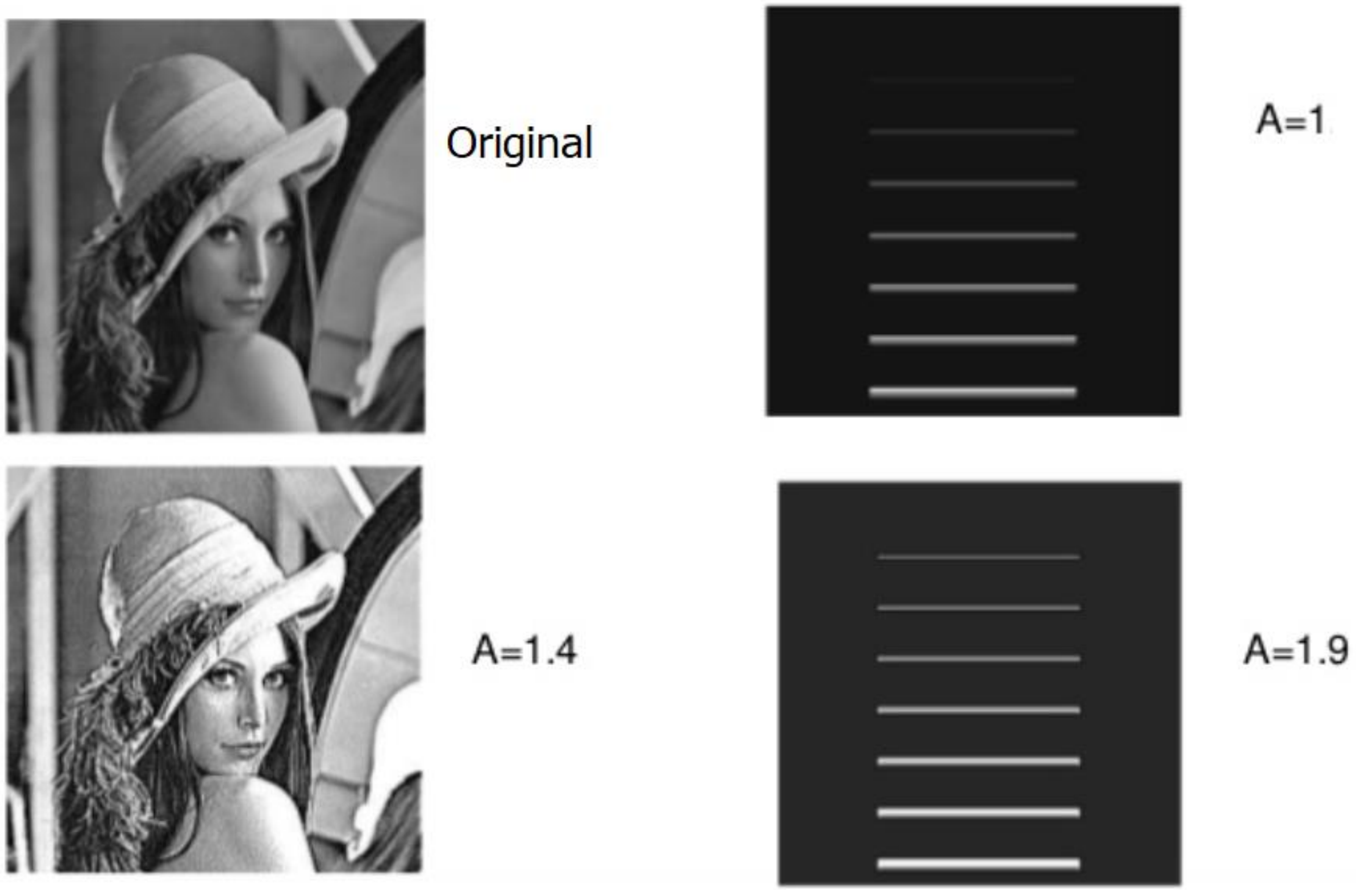
−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

Example:

Original Image

High Boost Filtered Image

Effect of different values of A



Problem:

- Filter the following image using 3 x 3 neighborhood high boost filtering (A=1.1) by assuming pixel replication

30	60	30
30	60	30
30	60	30

- After pixel replication:

30	30	60	30	30
30	30	60	30	30
30	30	60	30	30
30	30	60	30	30
30	30	60	30	30

A>=1

w = 9A-1

-1	-1	-1
-1	w	-1
-1	-1	-1

A=1.1

w= 9*1.1 – 1 = 8.9

w = 1/9

-1	-1	-1
-1	8.9	-1
-1	-1	-1

Solution:

- After applying mask

w = 1/9

-1	-1	-1
-1	8.9	-1
-1	-1	-1

- We get

30	30	60	30	30
30	30	60	30	30
30	30	60	30	30
30	30	60	30	30
30	30	60	30	30

Solution:

- We get

-7	26	-7
-7	26	-7
-7	26	-7

Solution:

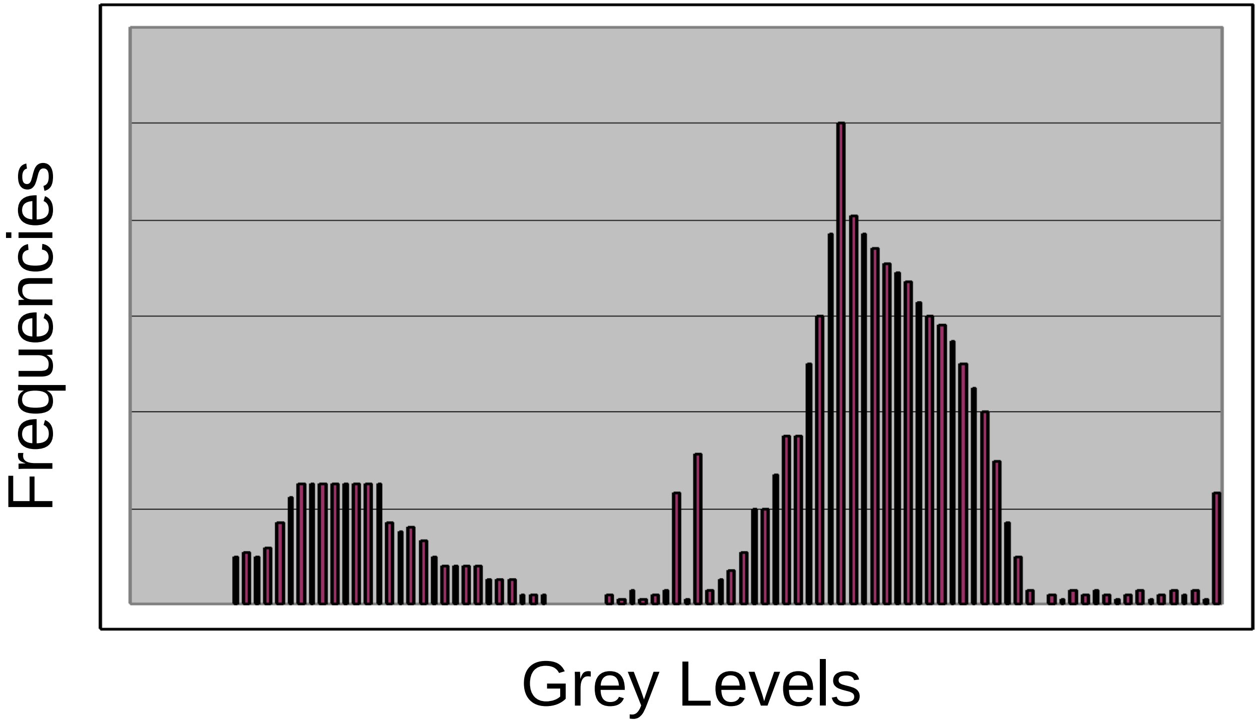
- Approximating

0	26	0
0	26	0
0	26	0

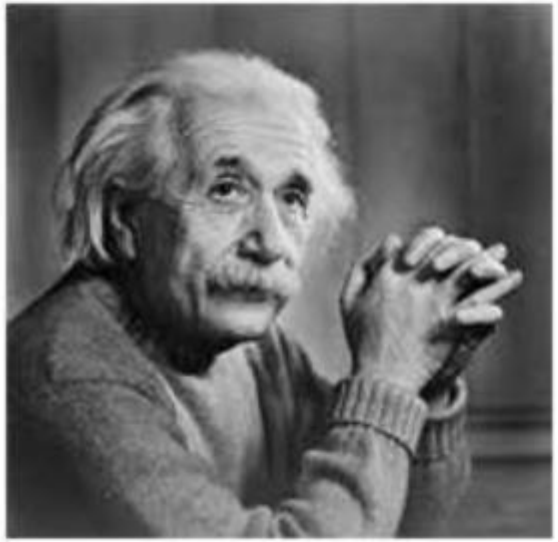
Histogram processing

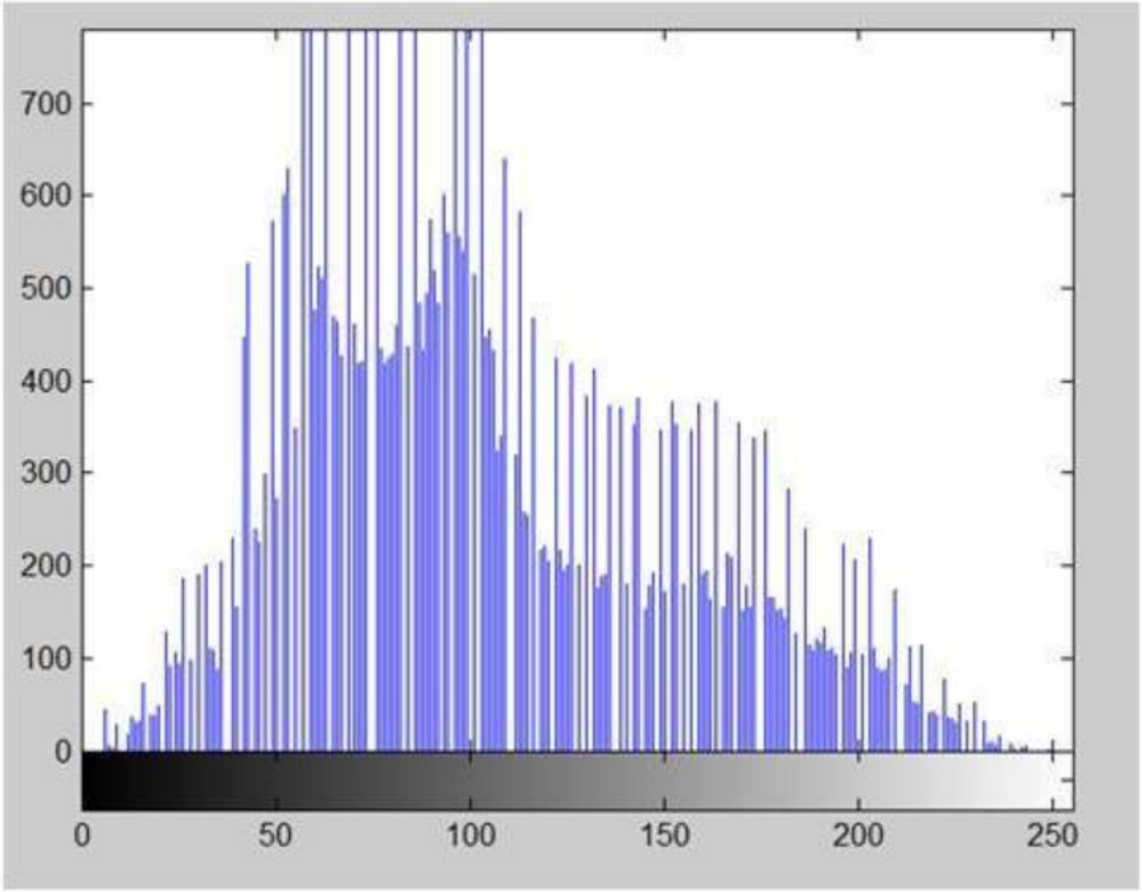
Image Histograms

- The histogram of an image shows us the distribution of grey levels in the image
- Massively useful in image processing, especially in segmentation.



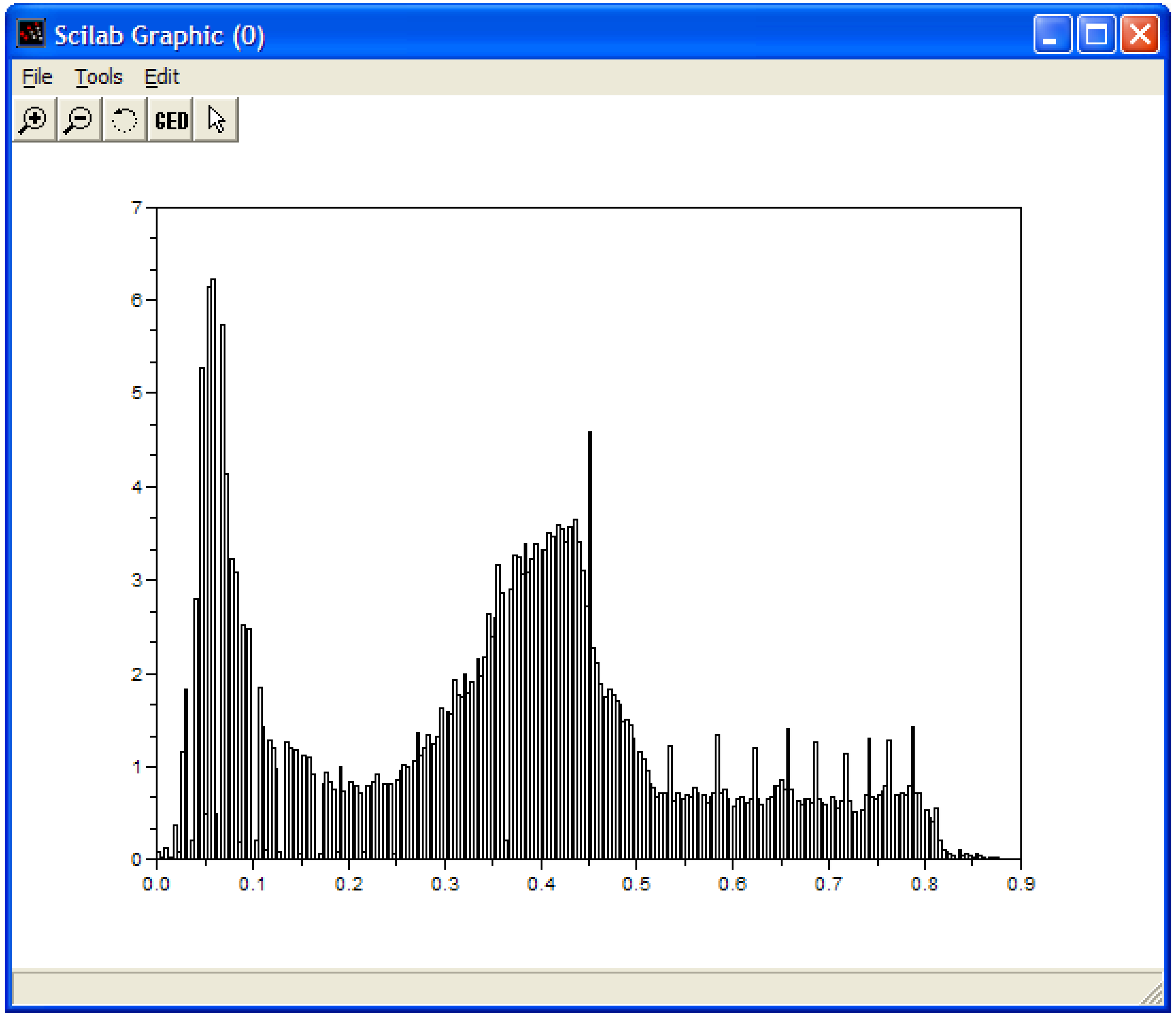
Its histogram is given as follows:





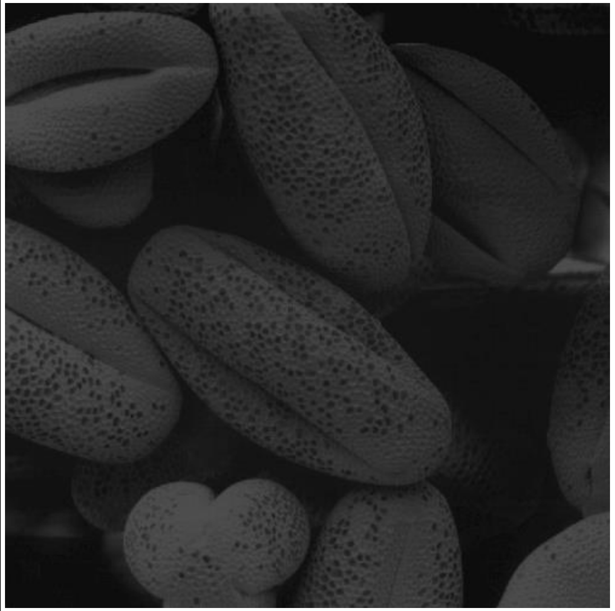
Histogram Examples (cont...)

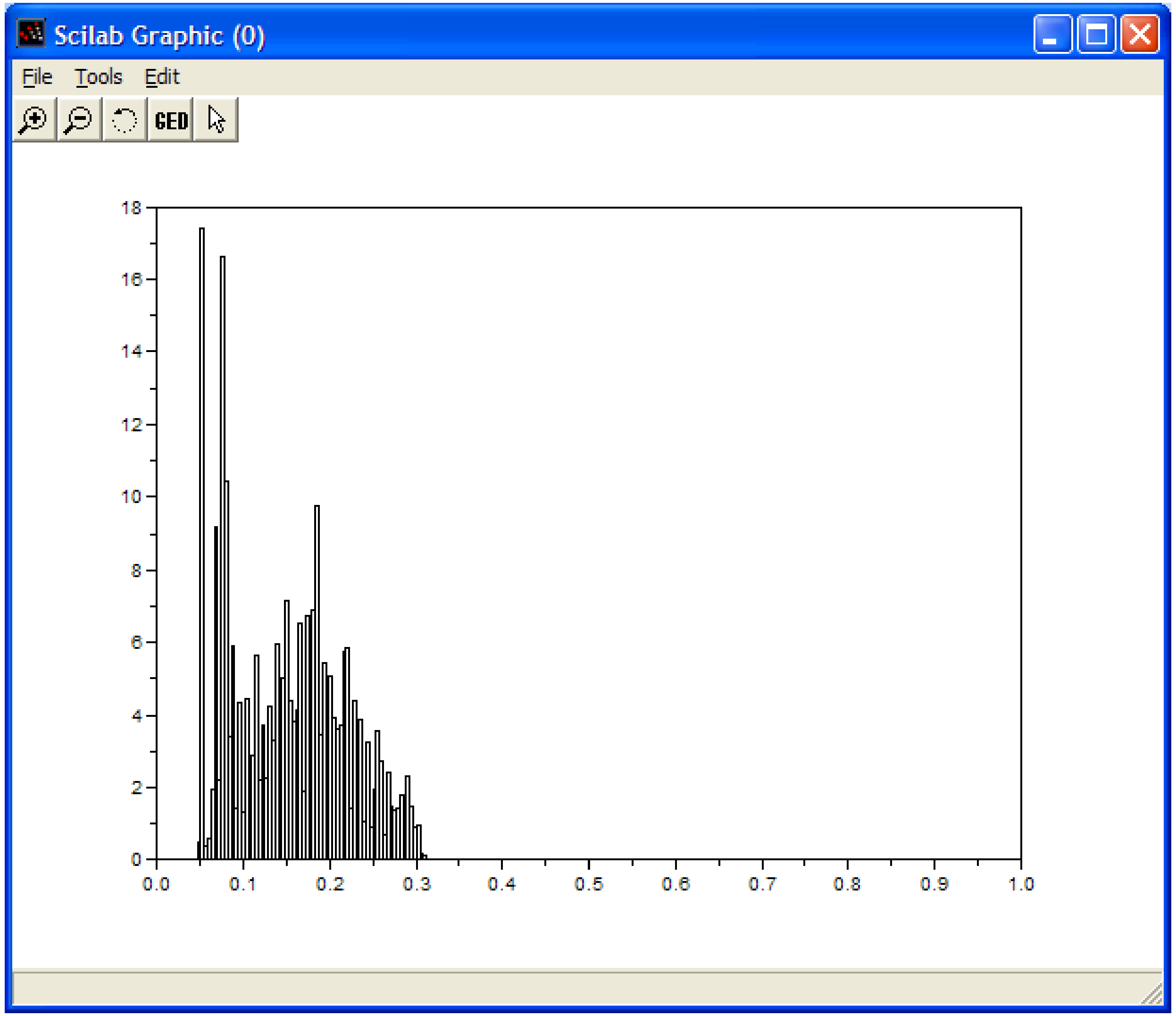
Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Histogram Examples (cont…)

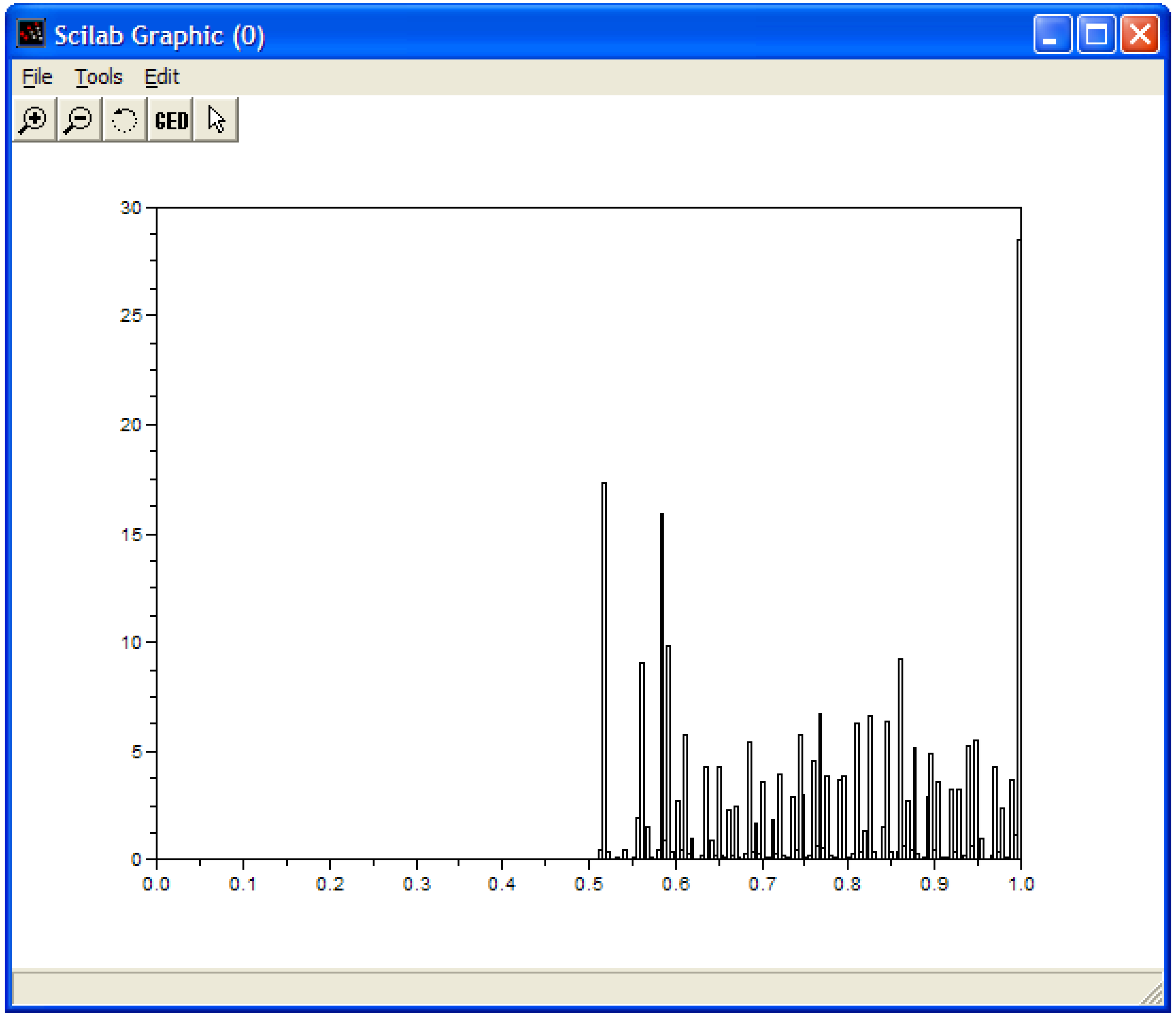
Images taken from Gonzalez & Woods, Digital Image Processing (2002)





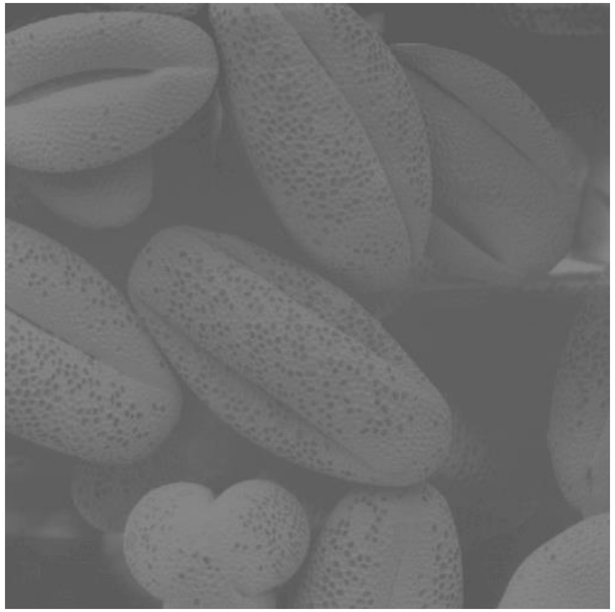
Histogram Examples (cont…)

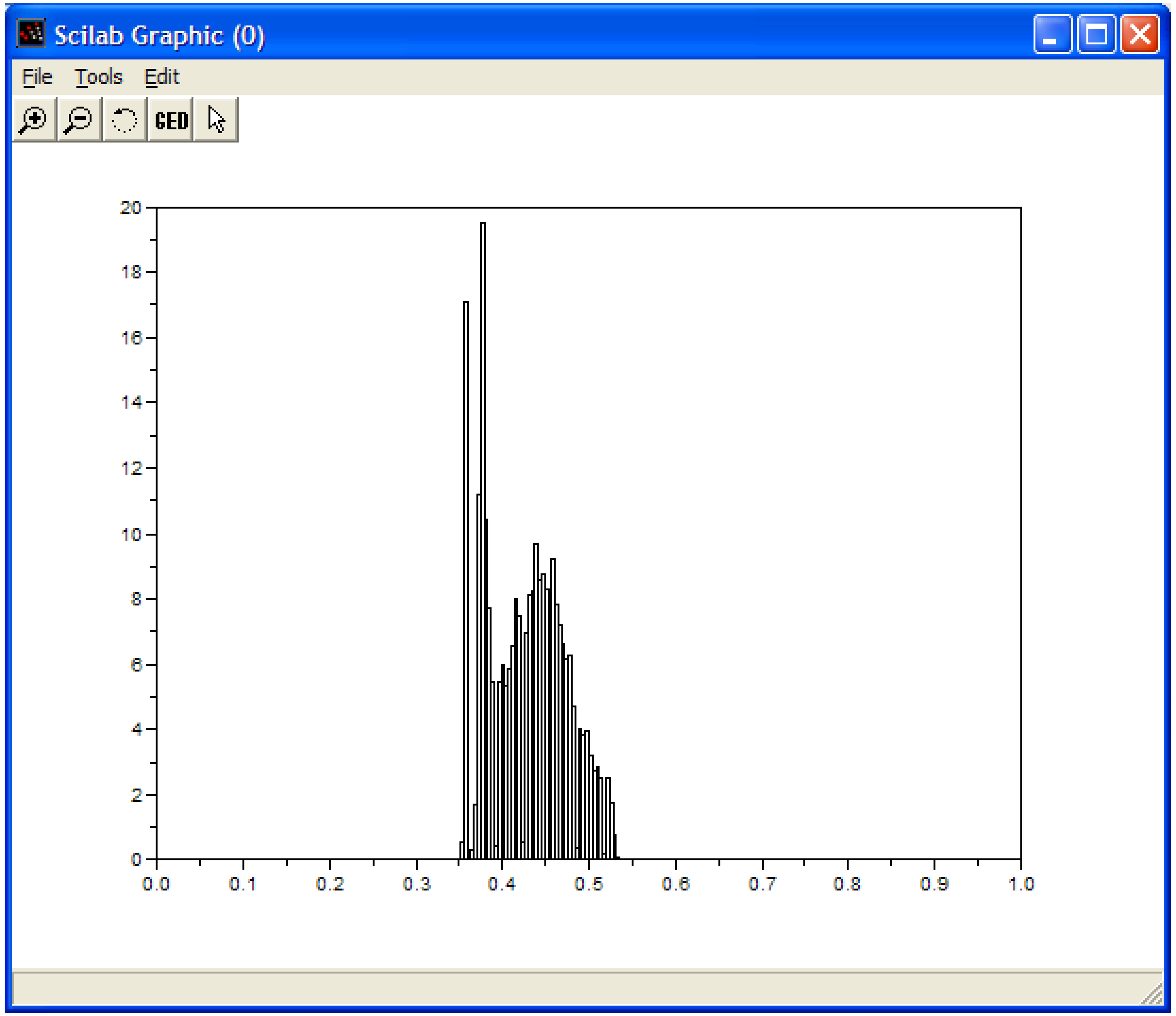
Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Histogram Examples (cont…)

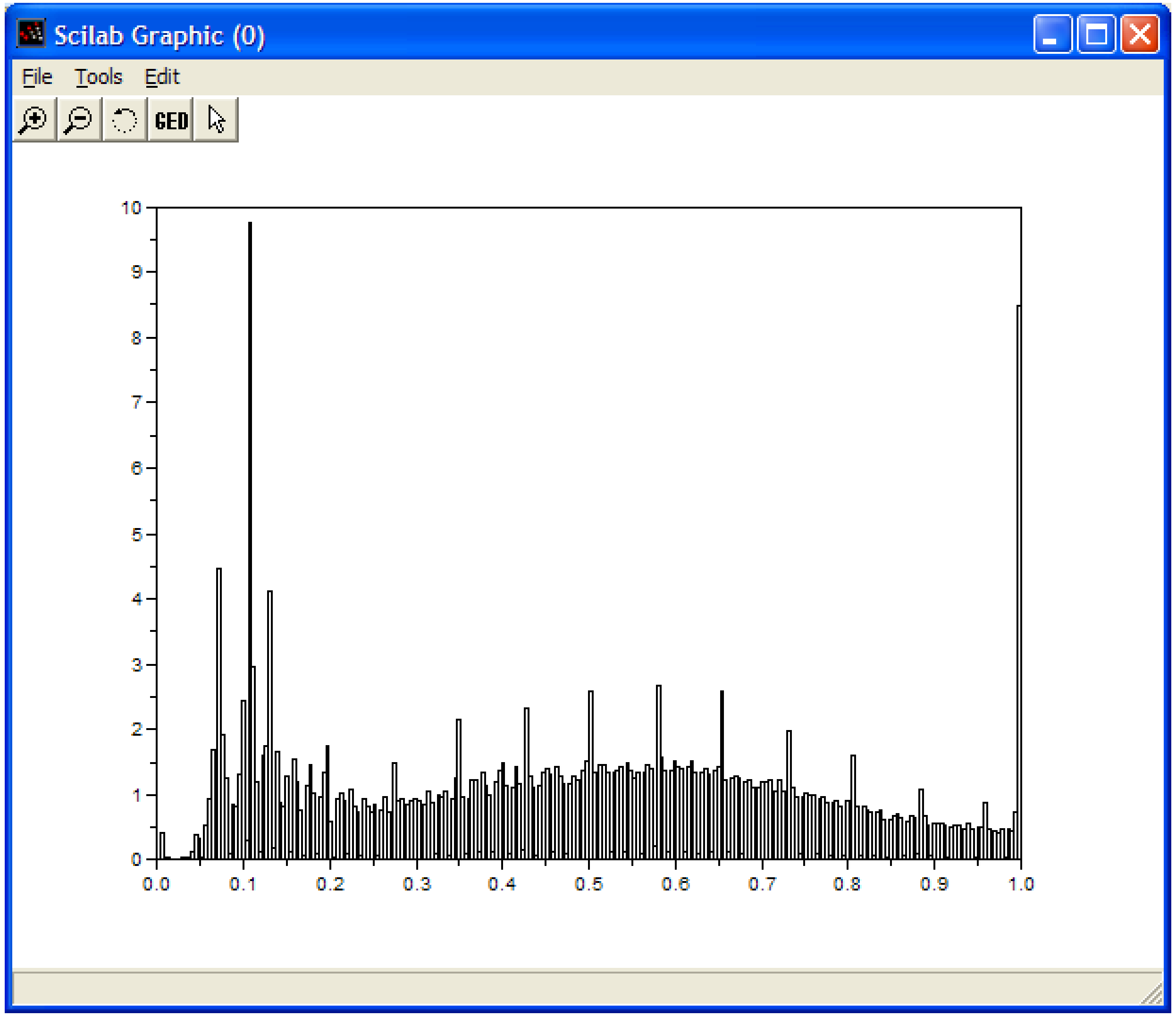
Images taken from Gonzalez & Woods, Digital Image Processing (2002)





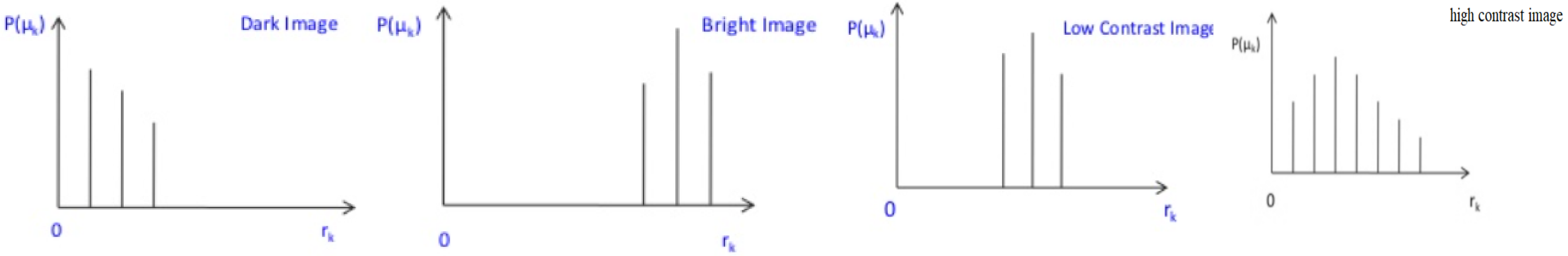
Histogram Examples (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)



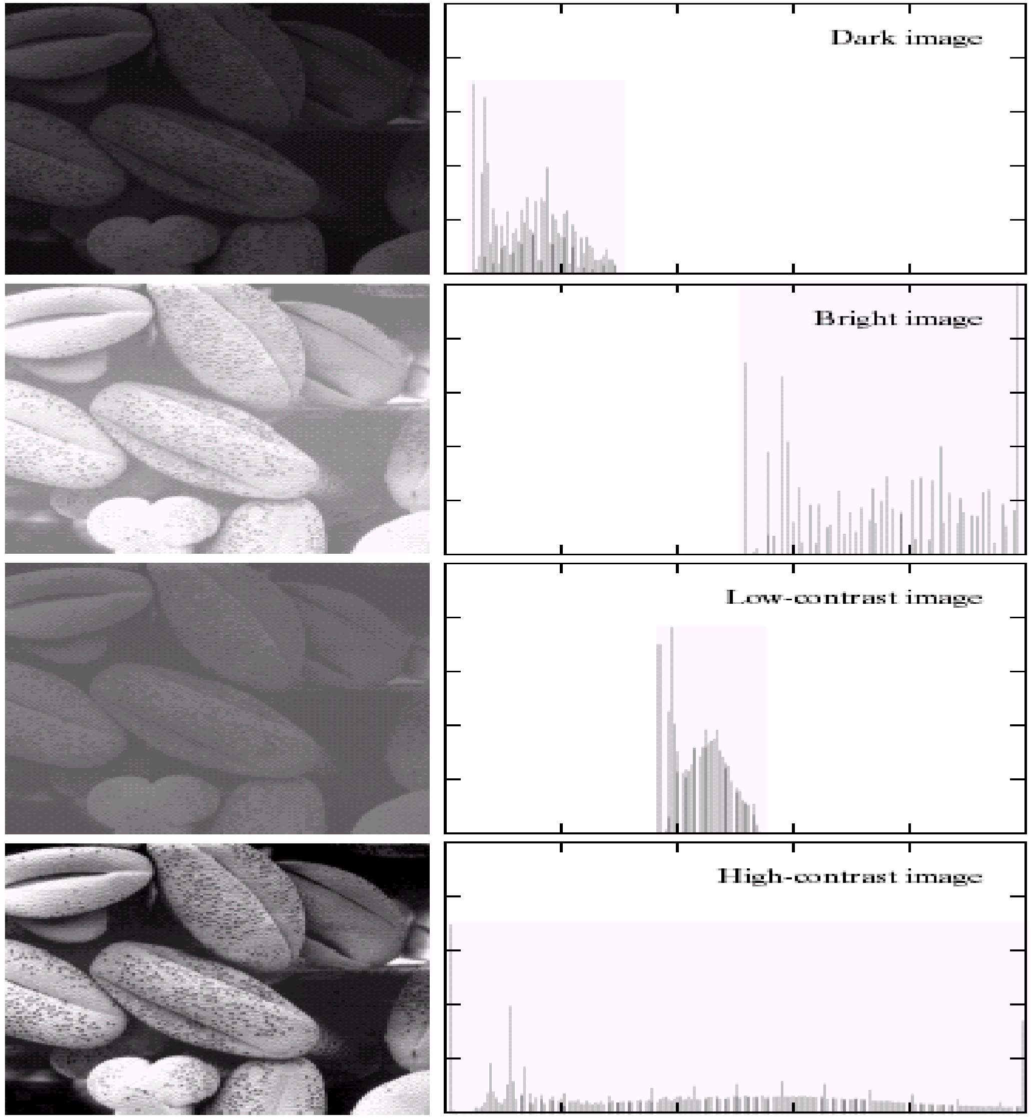
Histogram of an Image

- Histograms have many uses in image processing.
- It can be used to analyze an image, to adjust the contrast of the image and to equalize the image.
- A great deal of information can be obtained just by looking at the histogram.

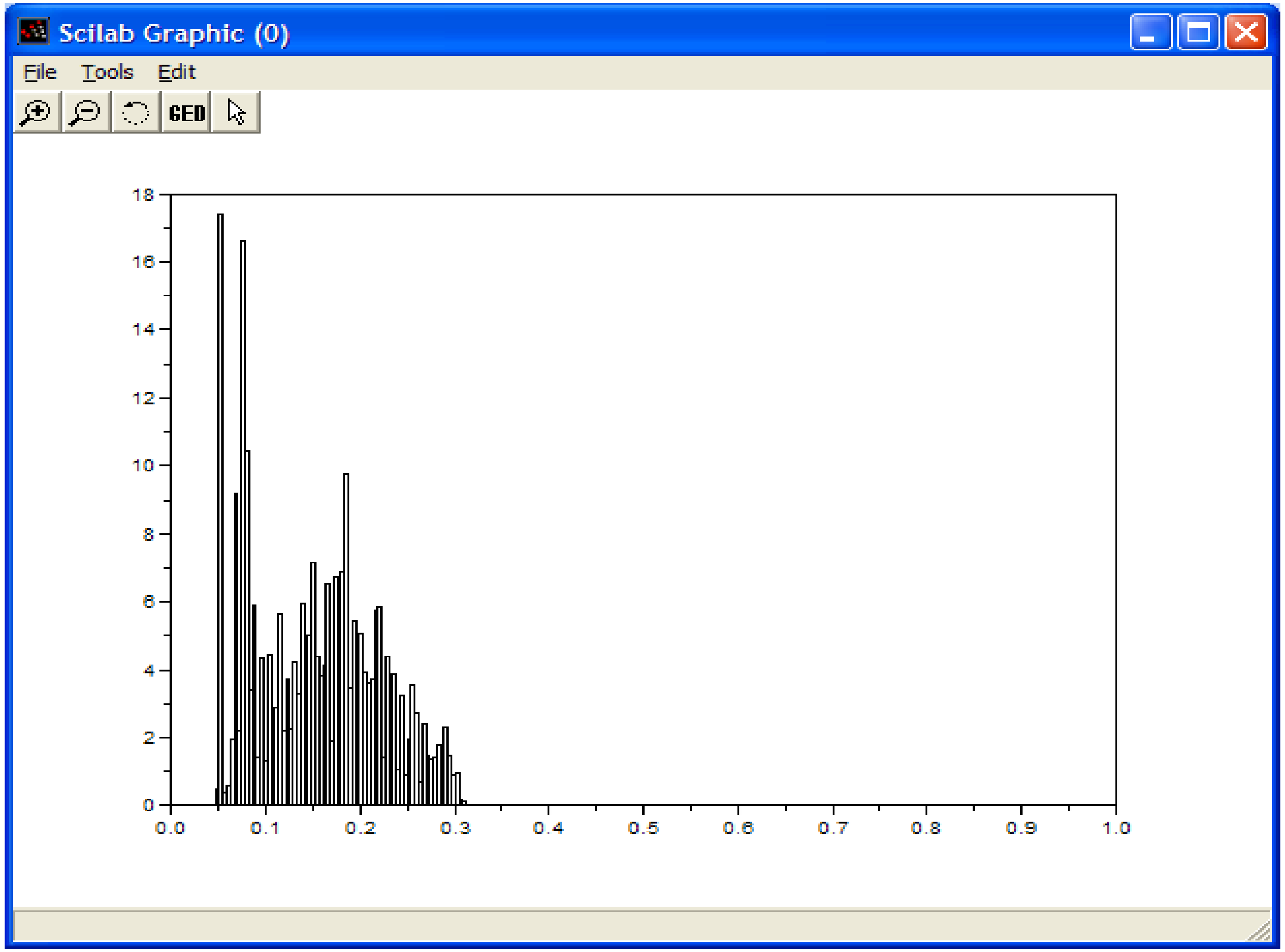


Histogram Examples (cont…)

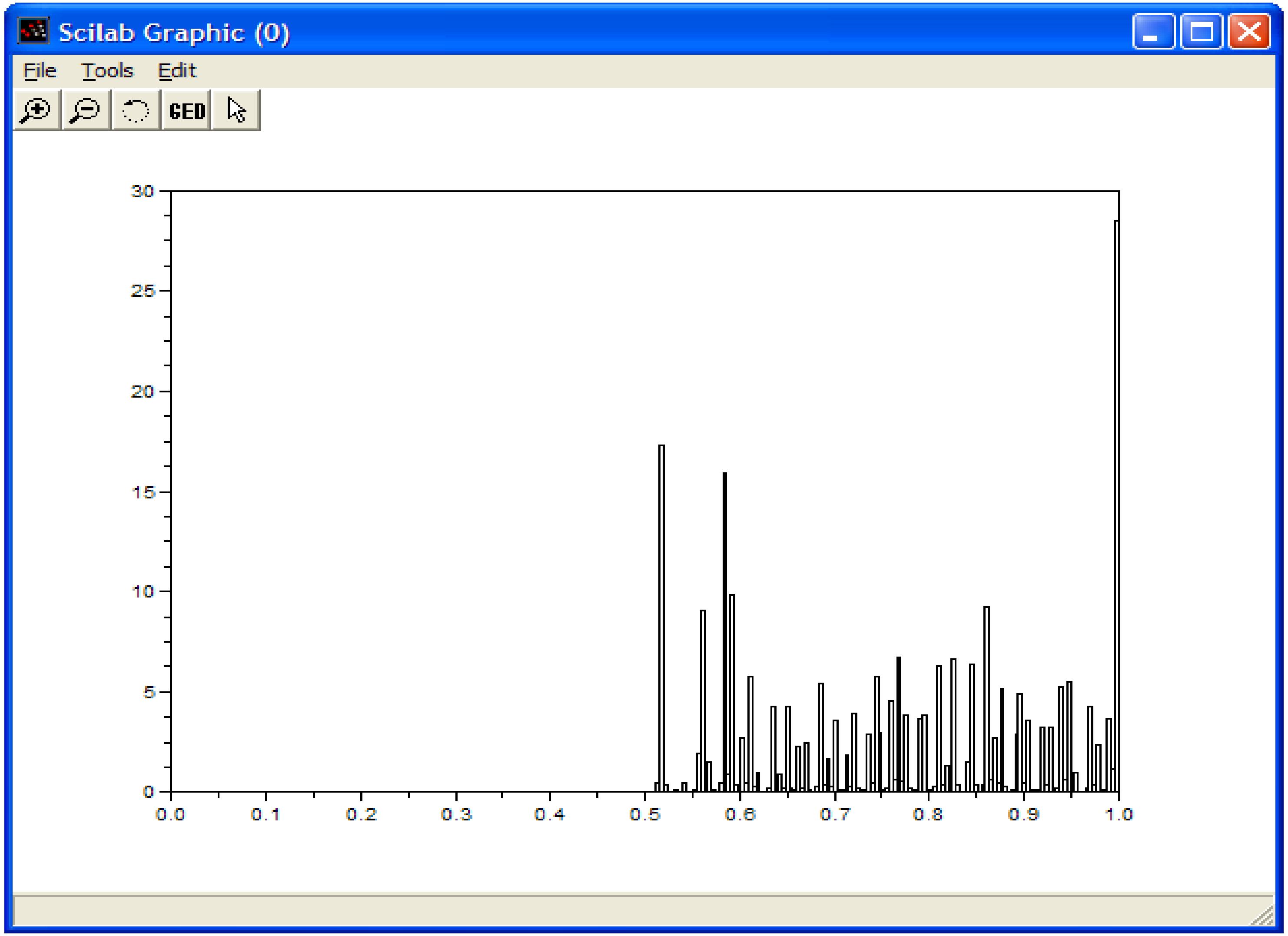
- A selection of **images and their histograms.**
- Notice the relationships between the images and their histograms.
- Note that the **high contrast image has the most evenly spaced histogram.**



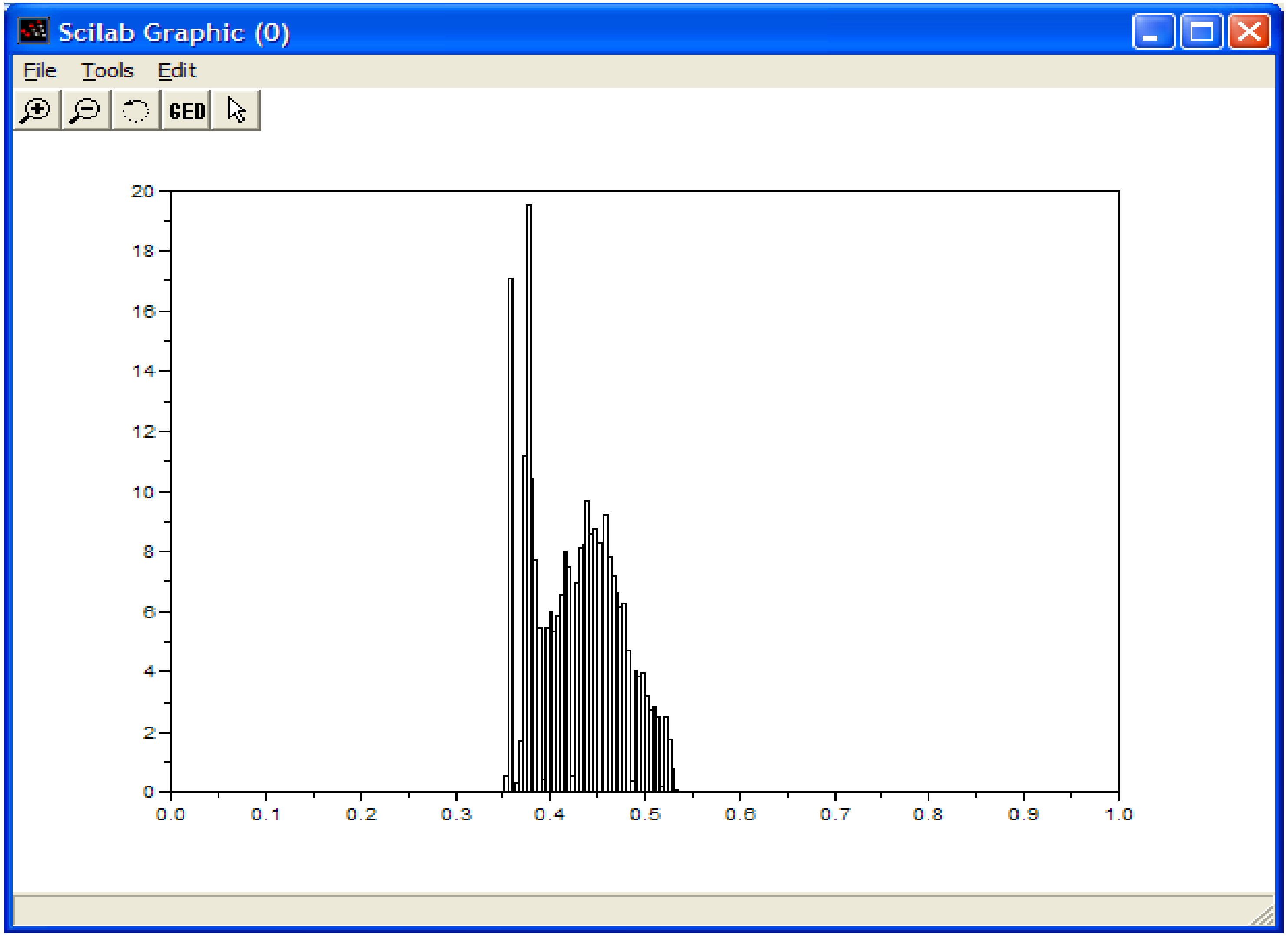
Dark image


Scilab Graphic (0)
File Tools Edit
GED
18
16
14
12
10
8
6
4
2
0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

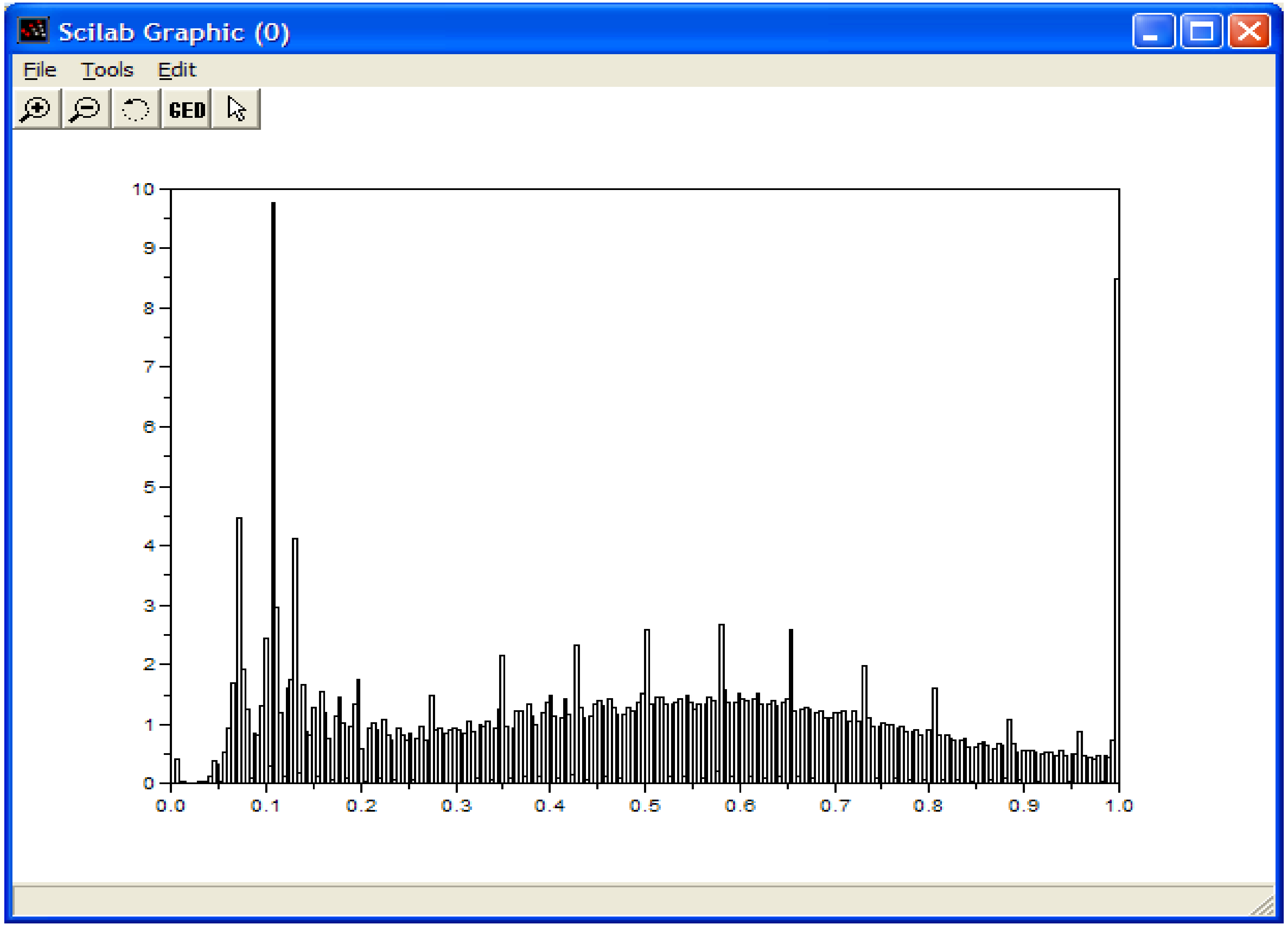
Bright image



Low contrast image



High contrast image



Histogram

- Out of above 4 **, last histogram** represents the **best image**.
- So to get the best image we would **transform** the **first three histogram** into the **fourth histogram**.
- We try to **increase dynamic range** of the image.
- Methods to increase dynamic range :

    1)**Linear Stretching**

    2)**Histogram equalization**

    3)**Histogram Specification**

Linear Stretching

- Way to increase dynamic range also known as Histogram stretching.
- We do not alter shape of the histogram, but we spread it so as to cover the entire dynamic range.
- We do this by using a straight line equation having slope:

  $(s_{max}-s_{min})/(r_{max}-r_{min})$

Where,

s_{max}=>Maximum gray level of output image

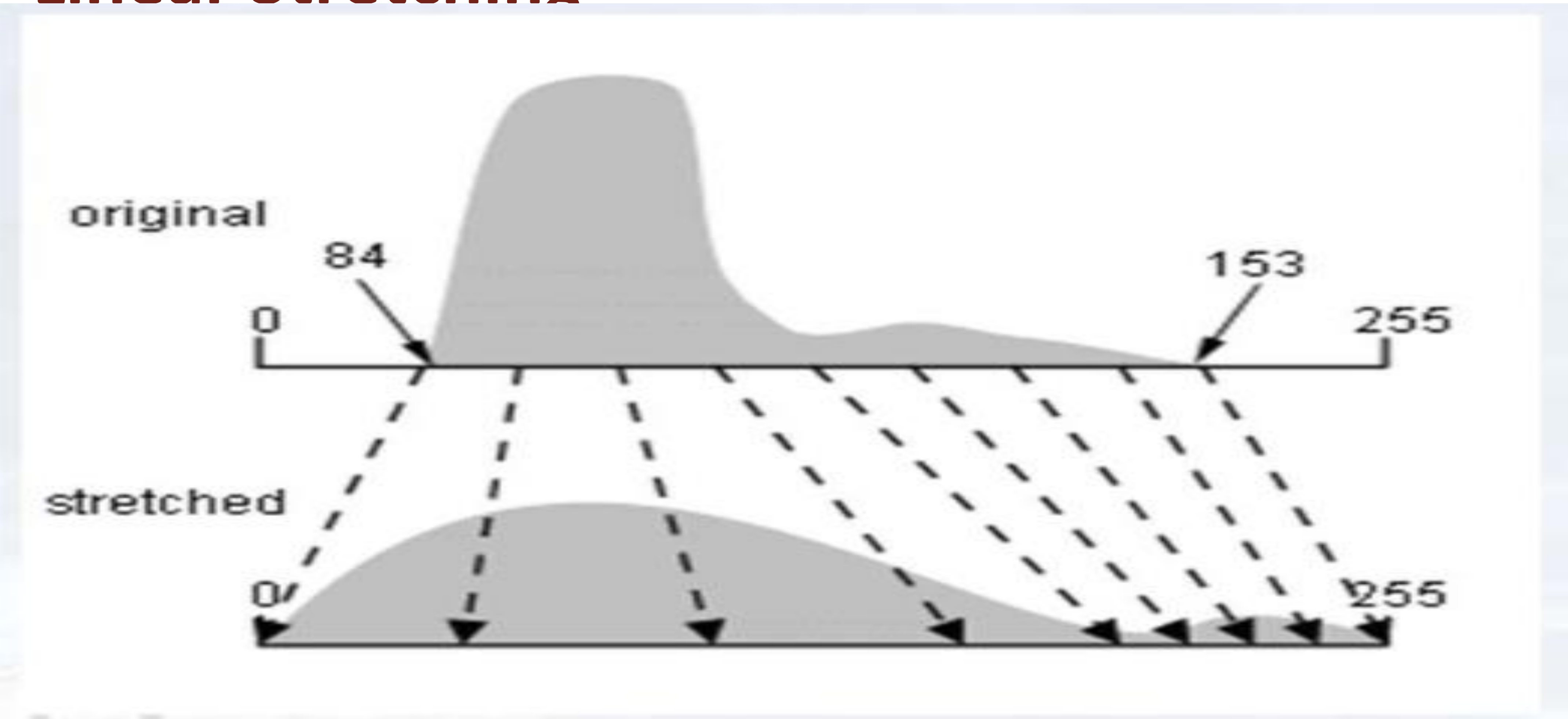
s_{min}=>Minimum gray level of output image

r_{max}=>Maximum gray level of input image

r_{min}=>Minimum gray level of input image

$s=T(r)=((s_{max}-s_{min})/(r_{max}-r_{min}))*(r-r_{min})+s_{min}$

Linear Stretching



The transformation function shifts and stretches the gray level range of input image to occupy the entire dynamic range: (s_{min} , s_{max})

Example-1 using Linear stretching

Perform histogram stretching so that the new image has a dynamic range of [0,7].

Grey Level	0	1	2	3	4	5	6	7
Number of pixels	0	0	50	60	50	20	10	0

$$s=T(r)=((s_{max}-s_{min})/(r_{max}-r_{min}))*(r-r_{min})+s_{min}$$

Solution

$s_{max} = 7$

$s_{min} = 0$

$r_{max} = 6$

$r_{min} = 2$

$$s = T(r) = \frac{s_{max} - s_{min}}{r_{max} - r_{min}}(r - r_{min}) + s_{min}$$

Solution

- Substituting the values
- s = (7-0)/(6-2) (r-2)+0

  = (7/4)(r-2)

  = 1.75(r-2)

Solution

- Substituting the values
- s = (7-0)/(6-2) (r-2)+0

  = (7/4)(r-2)

  = 1.75(r-2)

r	s
2	0
3	1.75≅ 2
4	3.5≅ 4
5	5.25≅5
6	7

Modified histogram

Grey Level	0	1	2	3	4	5	6	7
Number of pixels	50	0	60	0	50	20	0	10

Example-2 using Linear stretching

Perform histogram stretching so that the new image has a dynamic range of [0,7].

Grey Level	0	1	2	3	4	5	6	7
Number of pixels	100	90	85	70	0	0	0	0

$$s=T(r)=((s_{max}-s_{min})/(r_{max}-r_{min}))*(r-r_{min})+s_{min}$$

$$s = T(r) = \frac{(s_{max} - s_{min})}{(r_{max} - r_{min})} (r - r_{min}) + s_{min}$$

$s_{max} = 7$ $\quad$ $s_{min} = 0$

$r_{max} = 3$ $\quad$ $r_{min} = 0$

$$s = \left(\frac{7 - 0}{3 - 0}\right)(r - 0) + 0$$

$$s = \frac{7}{3} \cdot r.$$

$$r = 0 \longrightarrow 3.$$

$$r = 0, \quad s = \frac{7}{3} \cdot 0 = 0$$

$$r = 1, \quad s = \frac{7}{3} \cdot 1 = \frac{7}{3} = 2.33 \approx 2$$

$$r = 2, \quad s = \frac{7}{3} \cdot 2 = \frac{14}{3} = 4.66 = 5$$

$$r = 3, \quad s = \frac{7}{3} \cdot 3 = 7$$

Modified histogram

Grey Level	0	1	2	3	4	5	6	7
Number of pixels	100	0	90	0	0	85	0	70

Histogram Stretching

- The transformation of the image is given by

$$s = T(r) = \frac{s_{max} - s_{min}}{r_{max} - r_{min}}(r - r_{min}) + s_{min}$$

- where
  - s_{max} → Maximum gray level of the output image
  - s_{min} → Minimum gray level of the output image
  - r_{max} → Maximum gray level of the input image
  - r_{min} → Minimum gray level of the input image

Histogram Equalization

- A good histogram is one in which almost all of the pixel intensities are used.
- Thus the whole available range of intensities contributes to the image.
- The contrast of the image is low if we use only a portion of the available pixel values.
- Thus a low contrast image can be altered so that all the available pixel intensities are used. This is known as histogram equalization.

Procedure

- The intensity of a pixel is represented by i and the available intensities are $0 \le i \le L-1$ where L for an 8-bit image would be 256, the available intensities are in that case 0 to 255.
- An intensity (i) value of L-1 is considered as white and an i value of 0 is considered black.
- The intensity values in an image can be regarded as random variables that can have any value between 0 and L-1.
- This random event has a cumulative distribution function (CDF) associated to itself.
- This function describes the likelihood that the random variable will be assigned a value less or equal to a specific value.

Procedure

- The cumulative distribution function (CDF) is given by

$$CDF(i \leq t) = \sum_{k=0}^{t} p_k$$

- The probability that a random pixel has the intensity value k is pk given as

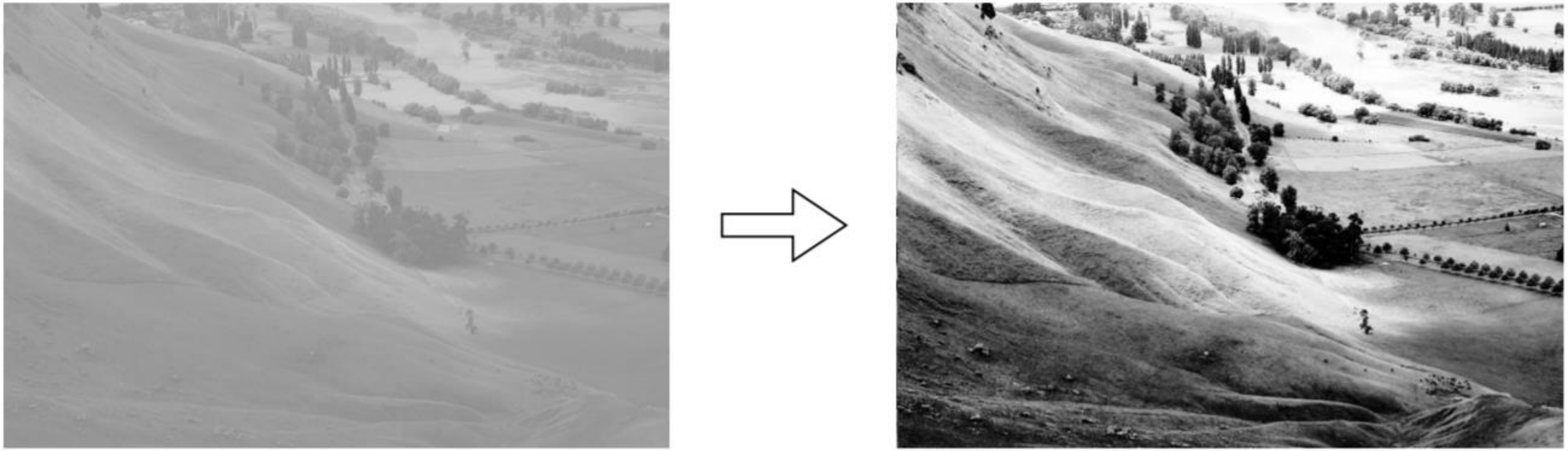
$$P_k = \frac{amount\ of\ pixels\ with\ intensity\ k}{total\ number\ of\ pixels}$$

Procedure

- The values are often normalized, so that $p_0 + p_1 + \ldots + p_{L-1} = 1$.
- The transformation T(i) is given by

$$s = T(i) = \text{round}\left((L-1) * \sum_{k=0}^{i} p_k\right)$$

Example



input output

Problem

- Equalize the histogram given below:

Gray Level	0	1	2	3	4	5	6	7
n_k	790	1023	850	656	329	245	122	81

Solution

Gray Level	n_k	$p_k = n_k/n$	p_k	CDF	(L-1)*CDF	Round
0	790	790/4096	0.19	0.19	1.33	1
1	1023	1023/4096	0.25	0.44	3.08	3
2	850	850/4096	0.21	0.65	4.55	5
3	656	656/4096	0.16	0.81	5.67	6
4	329	329/4096	0.08	0.89	6.23	6
5	245	245/4096	0.06	0.95	6.65	7
6	122	122/4096	0.03	0.98	6.86	7
7	81	81/4096	0.02	1	7	7
	4096					

Solution

- Equalized histogram is given below:

Gray Level	0	1	2	3	4	5	6	7
n_k	0	790	0	1023	0	850	985	448

Example-2 using Histogram Equalization

Equalize the histogram.

Grey Level	0	1	2	3	4	5	6	7
Number of pixels	100	90	50	20	0	0	0	0

Histogram ...

o)

Gray level	0	1	2	3	4	5	6	7
nk	100	90	50	20	0	0	0	0

Intensity → levels

Gr lvl	nk	pk nk/n	pk	CDF	(L-1)* CDF	Round off
0	100	100/260	0.384	0.384	2.688	3
1	90	90/260	0.346	0.73	5.11	5
2	50	50/260	0.192	0.922	6.454	6
3	20	20/260	0.0769	1	7	7
4	0	0/260	0	1	7	7
5	0	0/260	0	1	7	7
6	0	0/260	0	1	7	7
7	0	0/260	0	1	7	7
	$\Sigma nk = 260$					

Gray level	0	1	2	3	4	5	6	7
nk	0	0	0	100	0	90	50	20

Histogram Specification

- In image processing, histogram matching or histogram specification is the transformation of an image so that its histogram matches a specified histogram.

Steps in Histogram Specification

1. Find histogram of input image h_x, and find its cumulative H_x, the histogram equalization mapping function:

$$H_x[j] = \sum_{i=0}^{j} h_x[i]$$

2. Specify the desired histogram h_z, and find its cumulative H_z, the histogram equalization mapping function:

$$H_z[j] = \sum_{i=0}^{j} h_z[i]$$

Steps in Histogram Specification

3. Relate the two mapping above to build a lookup table for the overall mapping. Specifically, for each input level i, find an output level j so that $H_z[j]$ best matches $H_x[i]$:

$$|H_x[i] - H_z[j]| = min_k |H_x[i] - H_z[k]|$$

4. Then we set-up a lookup entry lookup[i]=j

5. Transform the histogram of the input image

Problem

- Given histograms (a) and (b) , modify histogram (a) as given by histogram (b)

Histogram (a)

Gray Level	0	1	2	3	4	5	6	7
n_k	790	1023	850	656	329	245	122	81

Histogram(b)

Gray Level	0	1	2	3	4	5	6	7
n_k	0	0	0	614	819	1230	819	614

Solution

For histogram (a) we get :

Gray Level	n_k	$p_k = n_k/n$	p_k	CDF
0	790	790/4096	0.19	0.19
1	1023	1023/4096	0.25	0.44
2	850	850/4096	0.21	0.65
3	656	656/4096	0.16	0.81
4	329	329/4096	0.08	0.89
5	245	245/4096	0.06	0.95
6	122	122/4096	0.03	0.98
7	81	81/4096	0.02	1

For histogram (b) we get :

Gray Level	n_k	$p_k = n_k/n$	p_k	CDF
0	0	0/4096	0	0
1	0	0/4096	0	0
2	0	0/4096	0	0
3	614	614/4096	0.15	0.15
4	819	819/4096	0.20	0.35
5	1230	1230/4096	0.30	0.65
6	819	819/4096	0.20	0.85
7	614	614/4096	0.15	1

To make lookup table

Gray Level	CDF1	CDF2	$Min_{graylevel2}$ (CDF1-CDF2)
0	0.19	0	3
1	0.44	0	4
2	0.65	0	5
3	0.81	0.15	6
4	0.89	0.35	6
5	0.95	0.65	7
6	0.98	0.85	7
7	1	1	7

Solution

- The lookup table is

i	0	1	2	3	4	5	6	7
j	3	4	5	6	6	7	7	7

Using the histogram and lookup table

Gray Level	0	1	2	3	4	5	6	7
n_k	790	1023	850	656	329	245	122	81

The new histogram is

Gray Level	0	1	2	3	4	5	6	7
n_k	0	0	0	790	1023	850	985	448