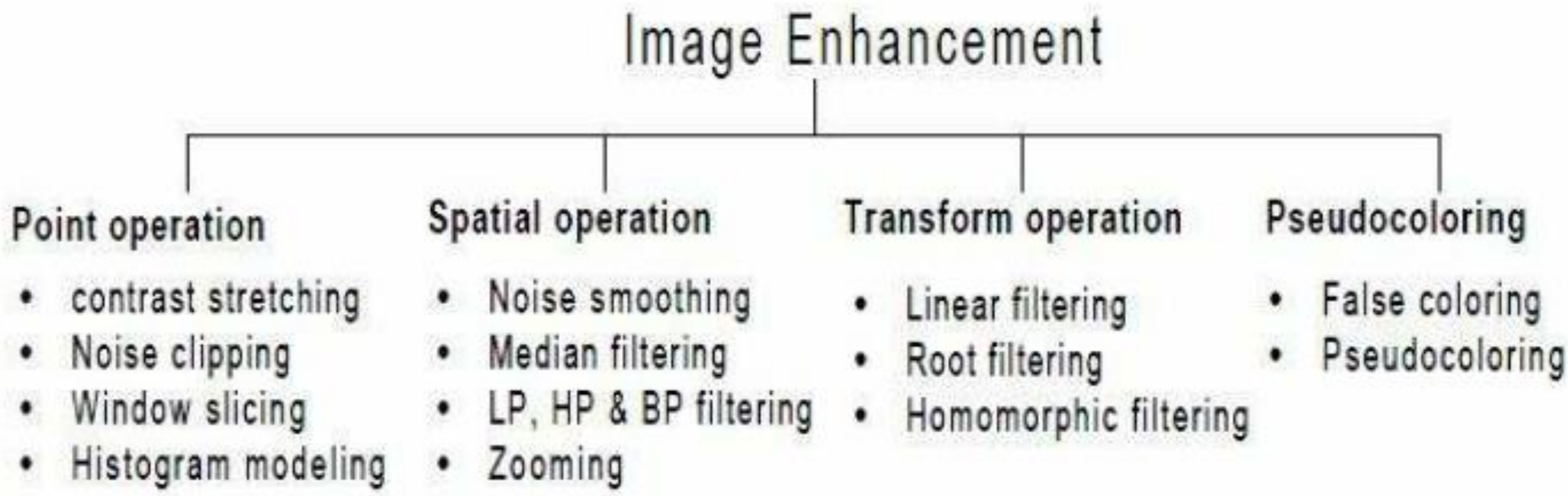
Image Enhancement

Introduction

- It highlights or sharpens image features such as edges, boundaries, or contrast to make a graphic display more helpful for display and analysis

- The enhancement doesn't increase the inherent information content of the data, but it increases the dynamic range of the chosen features so that they can be detected easily



- The greatest difficulty in image enhancement is quantifying the criterion for enhancement and, therefore, a large number of image enhancement techniques are empirical and require interactive procedures to obtain satisfactory results

- Image enhancement methods can be based on either spatial or frequency domain techniques

Spatial-Frequency domain enhancement methods

Spatial domain enhancement methods:

- Spatial domain techniques are performed to the **image plane** itself and they are based on direct manipulation of pixels in an image.
- The operation can be formulated as $g(x,y) = T\,[f(x,y)]$,

  where g is the output, f is the input image and T is an operation on *f defined over some* neighborhood of (x,y).

  According to the operations on the image pixels, it can be divided into 2 categories: *Point operations and spatial operations*.

Frequency domain enhancement methods:

- These methods enhance an image *f(x,y) by* **convoluting** the image with a linear, position invariant operator
- The 2D convolution is performed in frequency domain with **DFT**

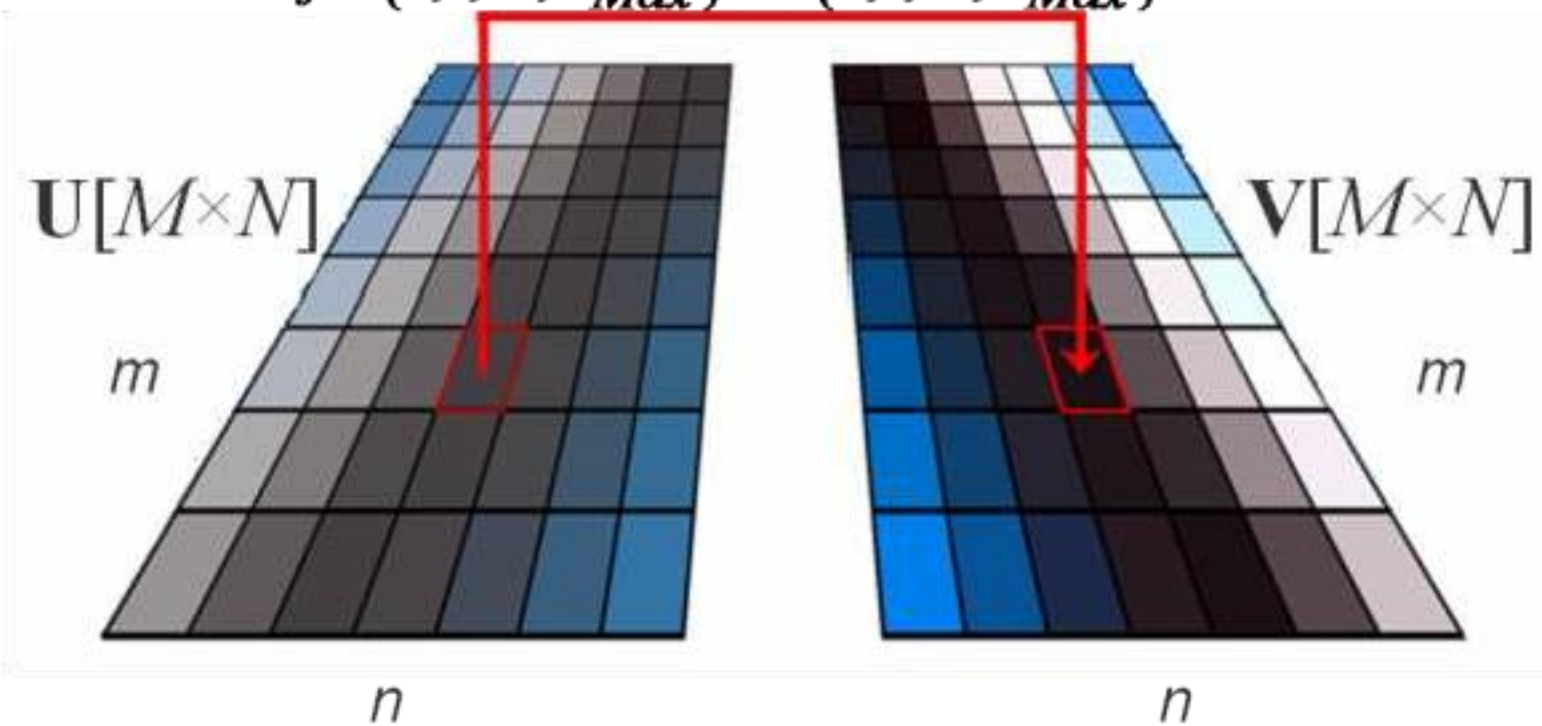
Point operations

-Zero-memory operations where a given gray level u∈[0,L] is mapped into a gray level v∈[0,L] according to a transformation.

$$v(m,n)=f(u(m,n))$$

$$v(m,n)=f\big(u(m,n)\big),\ \forall m=0,1,\ldots,M-1;\ n=0,1,\ldots,N-1;$$

$$f:\{0,1,\ldots,L_{Max}\}\rightarrow\{0,1,\ldots,L_{Max}\}$$

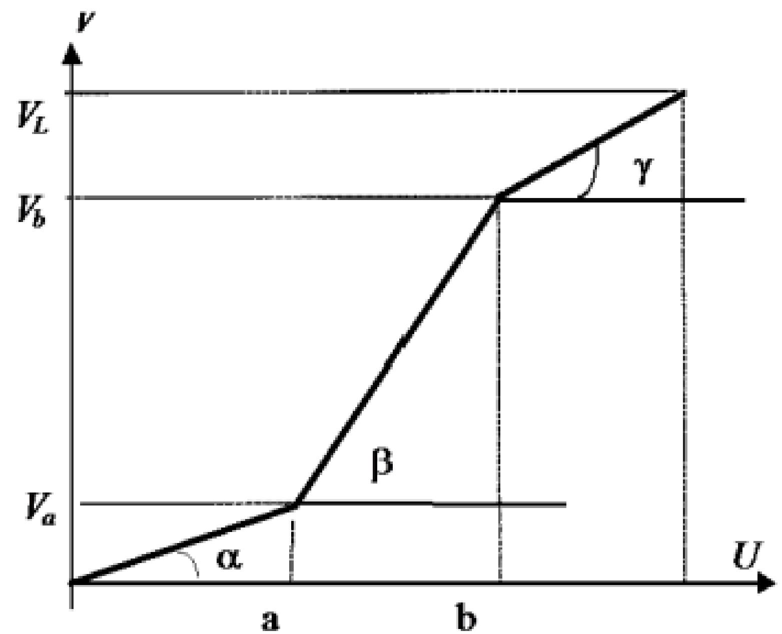


1-contrast stretching

- The idea behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.

Expressed as :

$$v = \begin{cases} \alpha u, & 0 \le u < a \\ \beta(u-a) + v_a, & a \le u < b \\ \gamma(u-b) + v_b, & b \le u < L \end{cases}$$

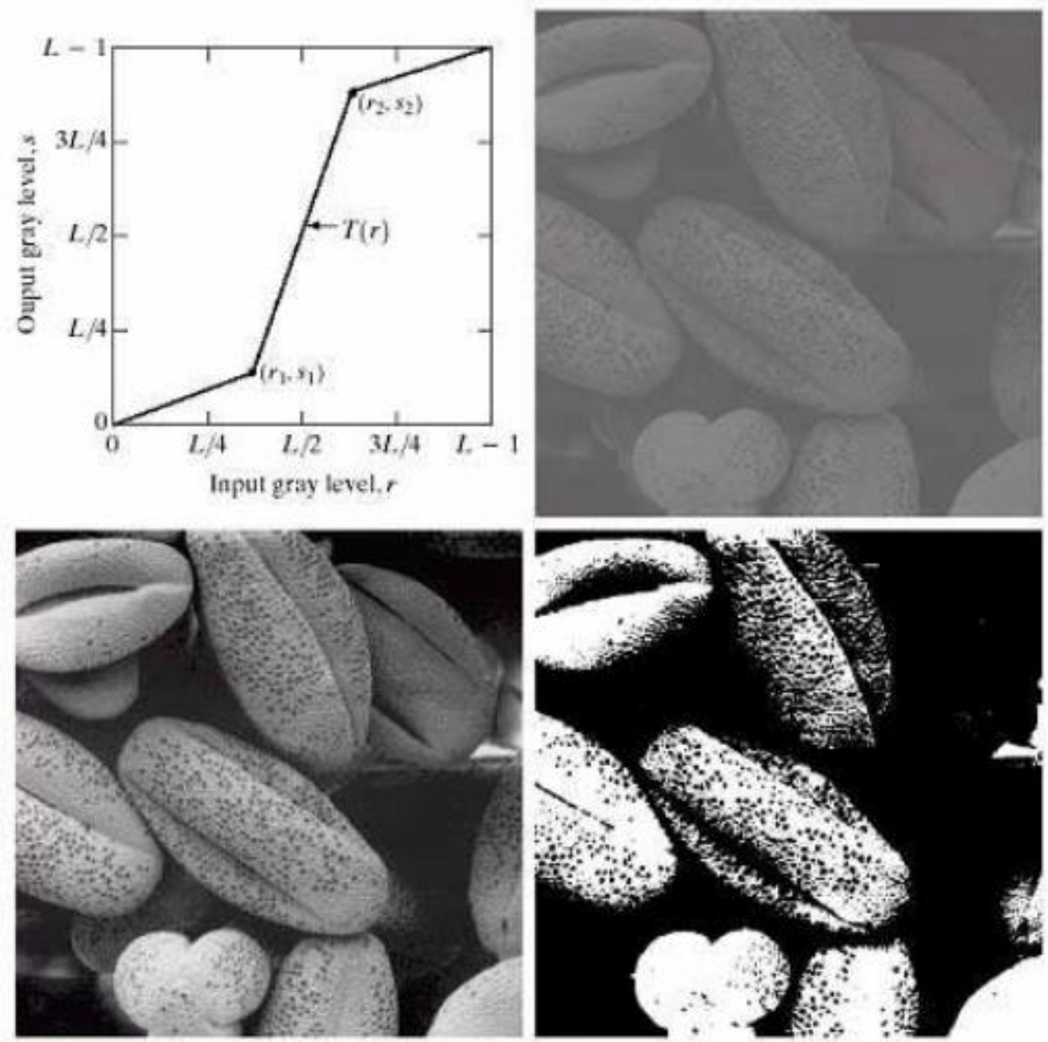


-For dark region stretch α>1 ,a=L/3

-For mid region stretch β>1 ,b=2/3L

-For bright region stretch γ>1

Example 1

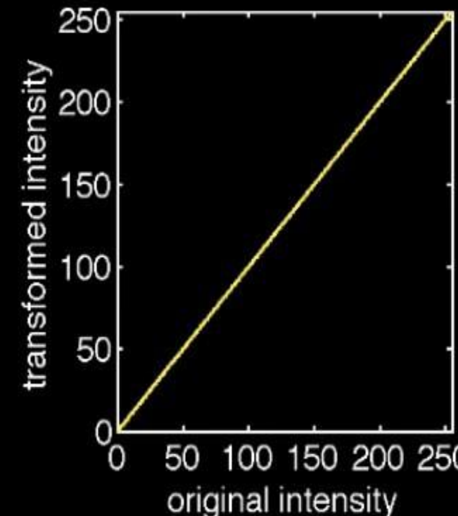


(b) a low-contrast image : results From i) poor illumination, ii)lack of dynamic range in the imaging sensor, or iii)even wrong setting of a lens aperture of image acquisition

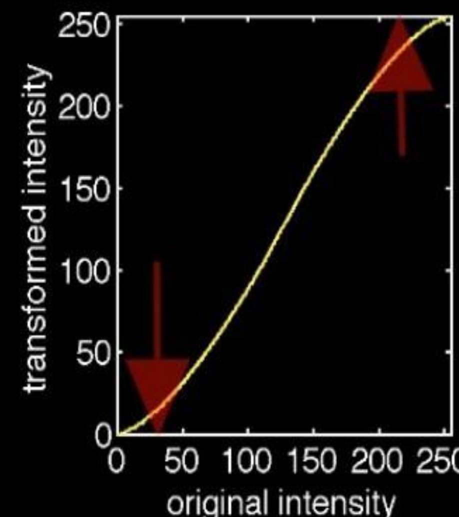
(c) result of contrast stretching :
(r1,s1) = (rmin,0) and
(r2,s2) = (rmax,L-1)

(d)result of thresholding

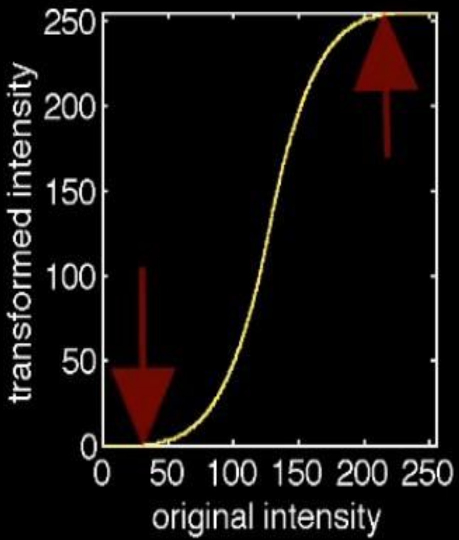
Contrast


transformed intensity
original intensity
0
50
100
150
200
250

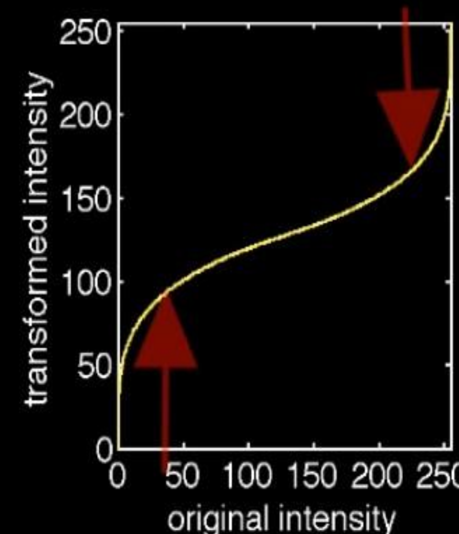
Contrast Increasing: Decreasing low intensity values and increasing high intensity values.



<u>Contrast Increasing:</u>



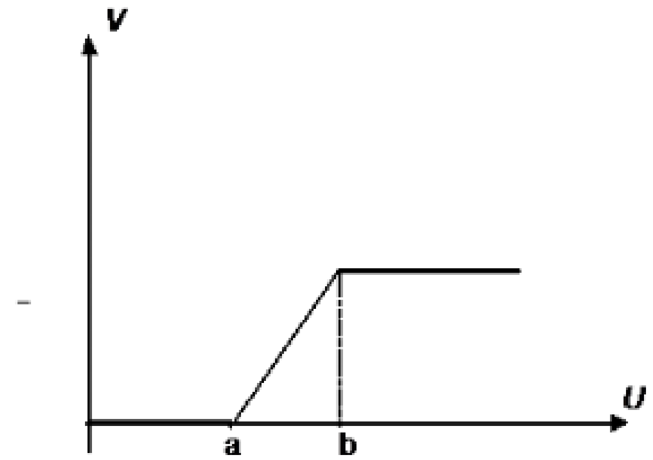
Contrast Decreasing: Increasing low intensity values and decreasing high intensity values.



2-Clipping and Thresholding

- Expressed as :

$$f(u)=\begin{cases}0, & 0\le u<a\\ \alpha u, & a\le u\le b\\ L, & u\ge b\end{cases}$$



- **Clipping:**

-Special case of contrast stretching ,where α= γ=0

-Useful for noise reduction when the input signal is known to lie in the range [a,b].

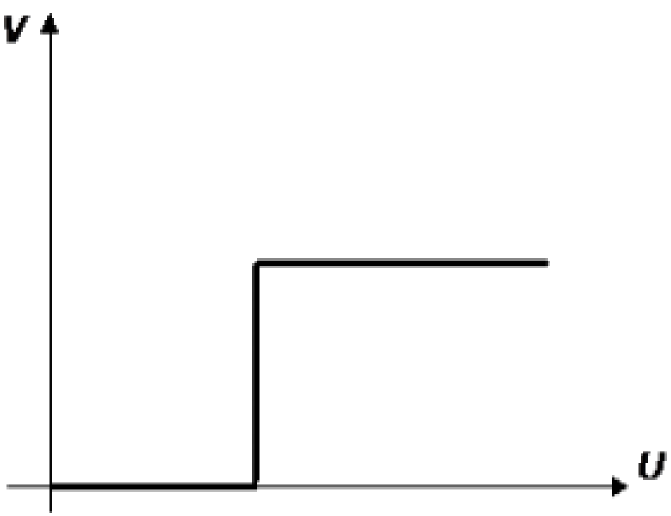
Thresholding:

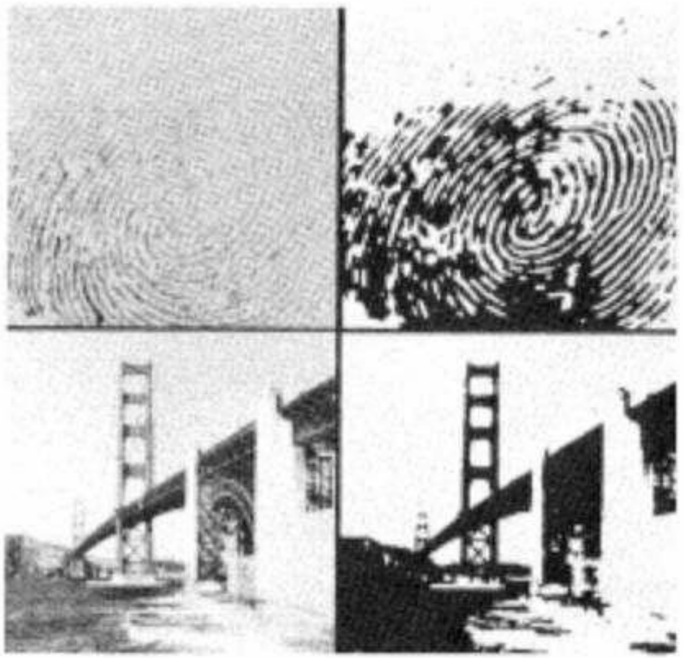
Cont.

-is a special case of case of clipping where a=b=t and the output comes binary.

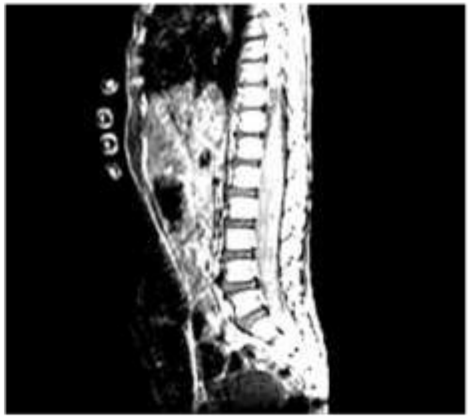
$$f(u)=\begin{cases}0, & 0\le u<a\\ \alpha u, & a\le u\le b\\ L, & u\ge b\end{cases}$$

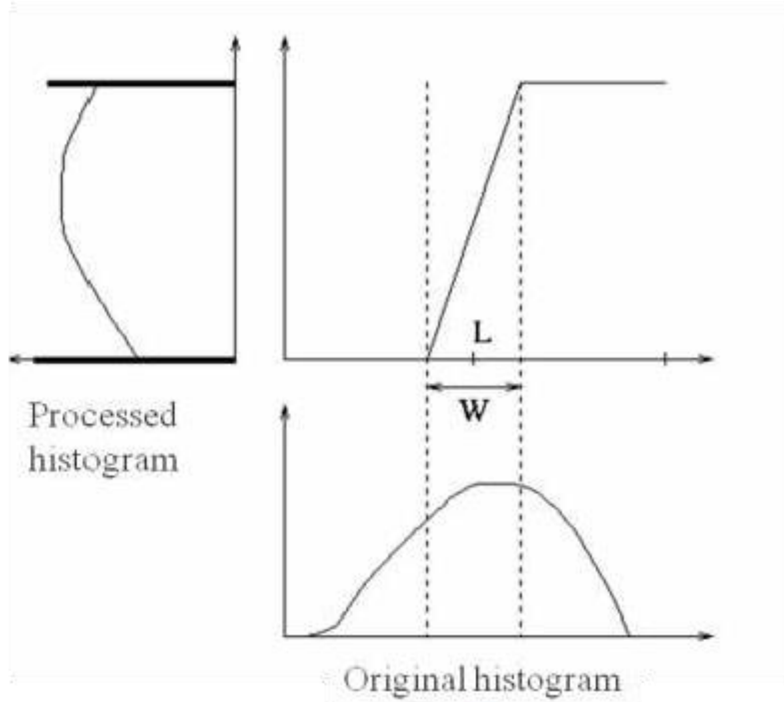
Useful for binary or other images that have bimodal distribution of gray levels. The a and b define the valley between the peaks of the histogram. For $a=b=t$, this is called *thresholding*.

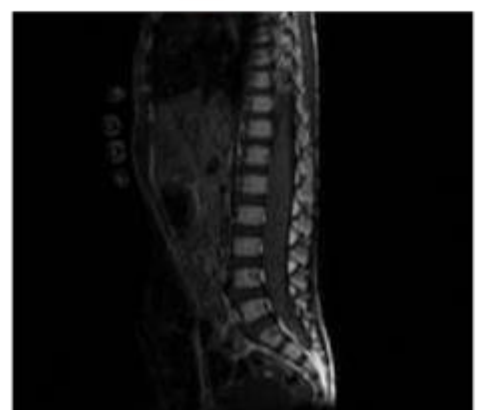




Example2

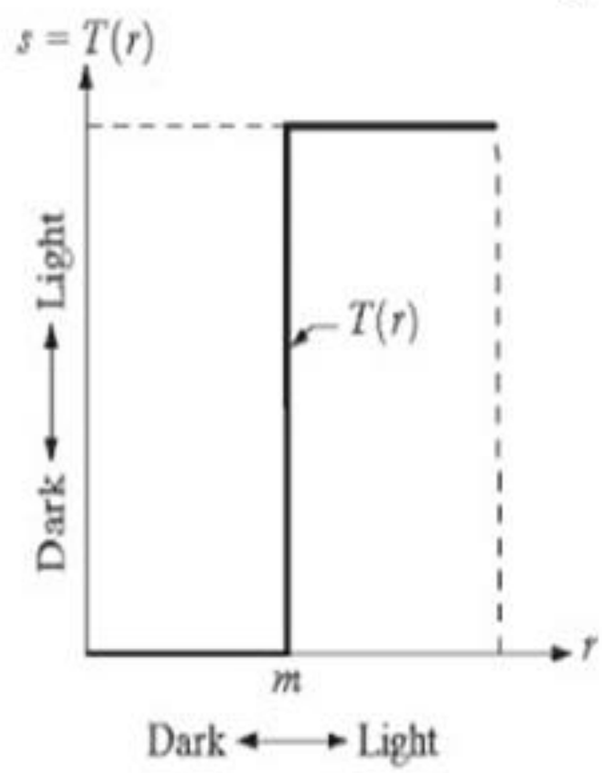


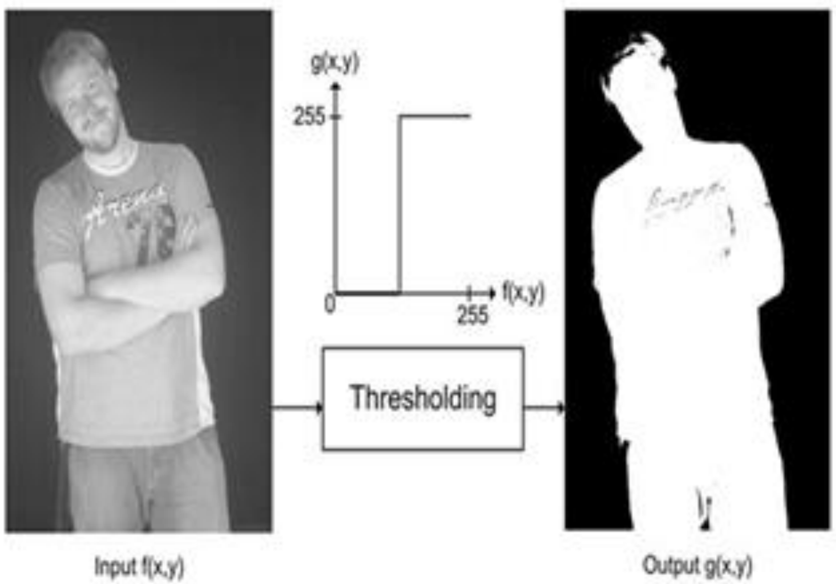




Example3

- Thresholding

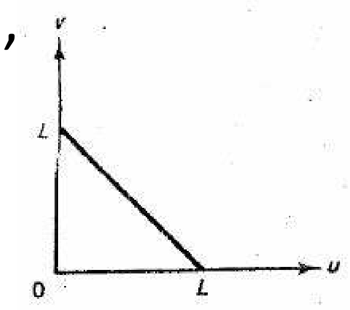


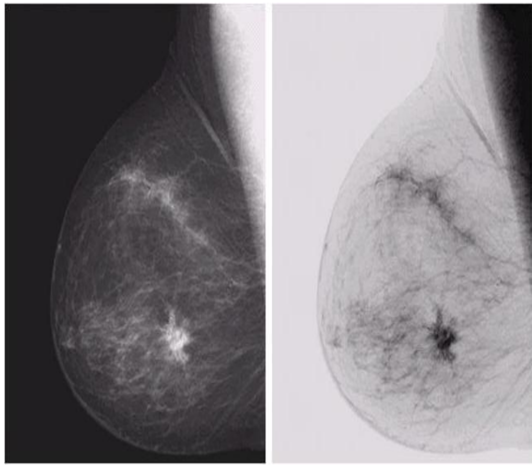


3-Digital negative

- Negative image can be obtained by reverse scaling of the gray levels according to the transformation,

$$v = L - u$$



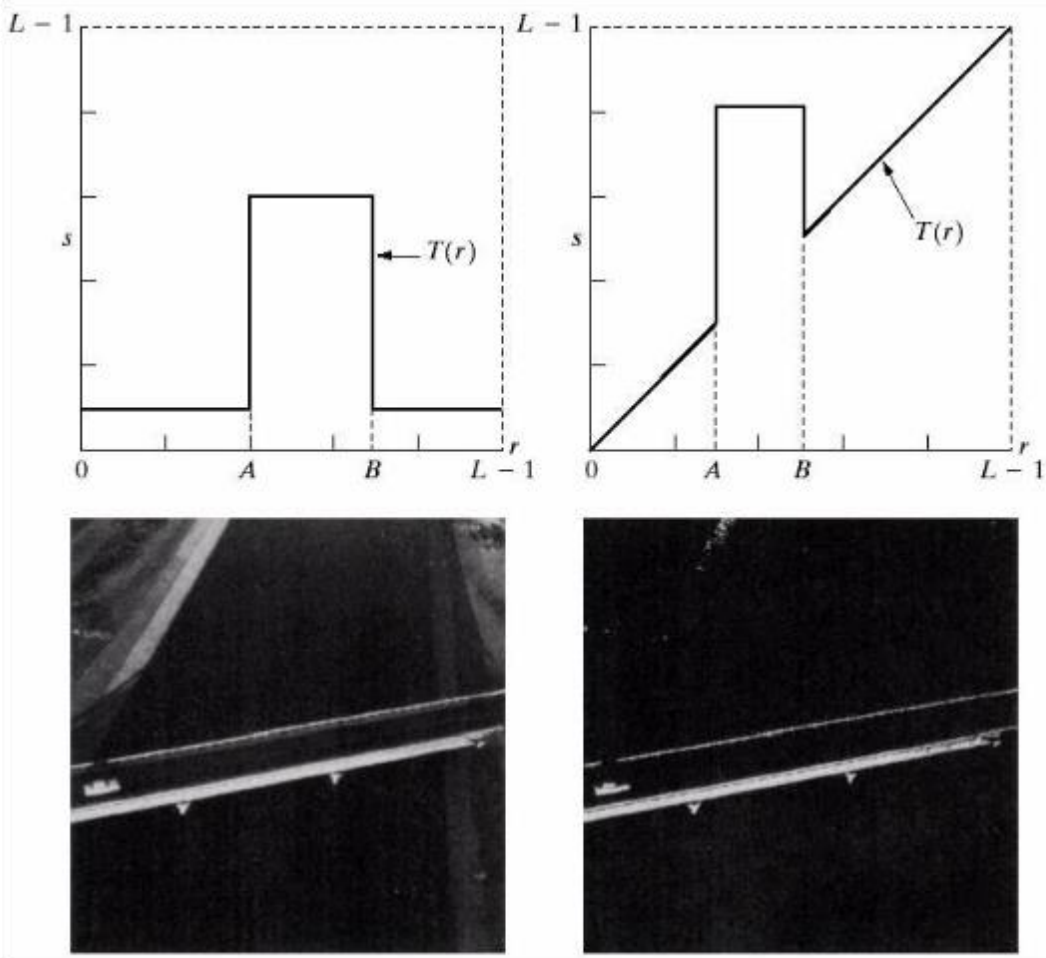


- Useful in the display of medical images.
- Example:

4-intensity level slicing

- Permit segmentation of certain gray level regions from the rest of the image.

$$v = \begin{cases} L, & a \le u \le b \\ 0, & otherwise \end{cases} \qquad v = \begin{cases} L, & a \le u \le b \\ u, & otherwise \end{cases}$$

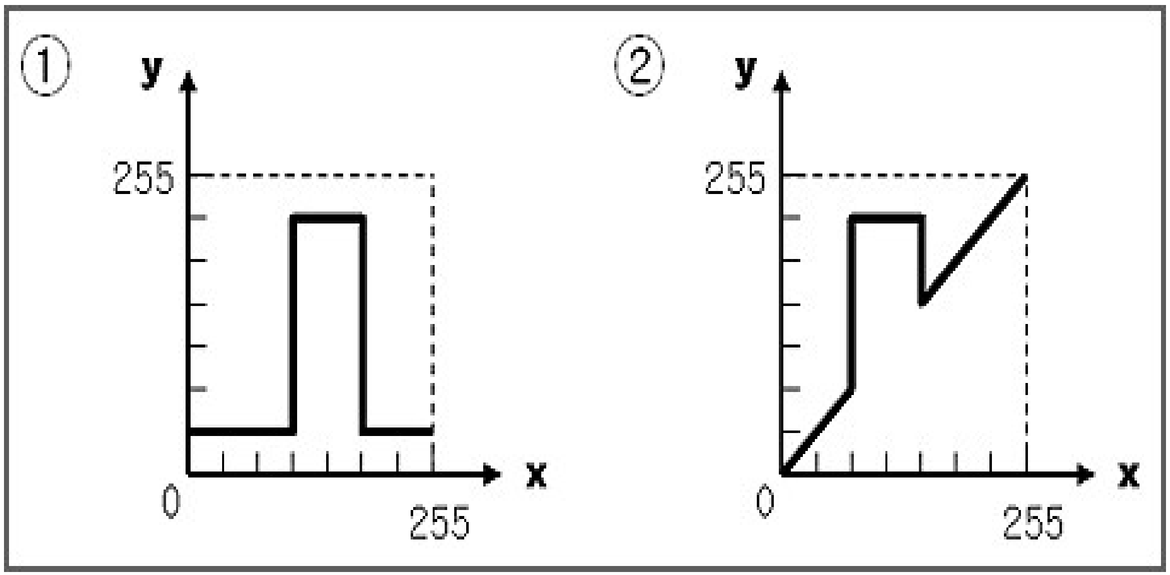


a b
c d

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).

e.g. Gray-level slicing:- Highlighting specific range of intensity values by

1. Non preserving background
2. Preserving background



Gray-level slicing:-
Highlighting specific range of intensity values by

1. Non preserving background
2. Preserving background

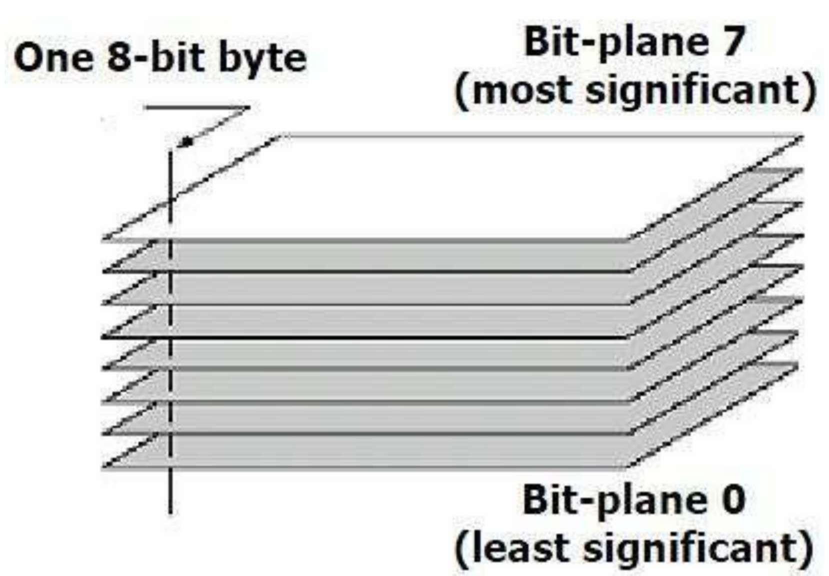
5-Bit extraction

- This transformation is useful

In determining the number of

Visually significant bits in an

Image.



- Suppose each pixel is

represented by 8 bits it is desired

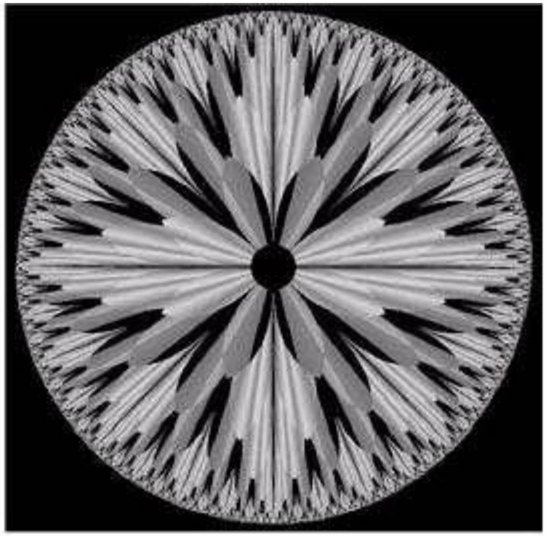
To extract the *n*th most significant bit

And display it .

- Higher-order bits contain the

majority of the visually significant data

Example



8-bit fractal image

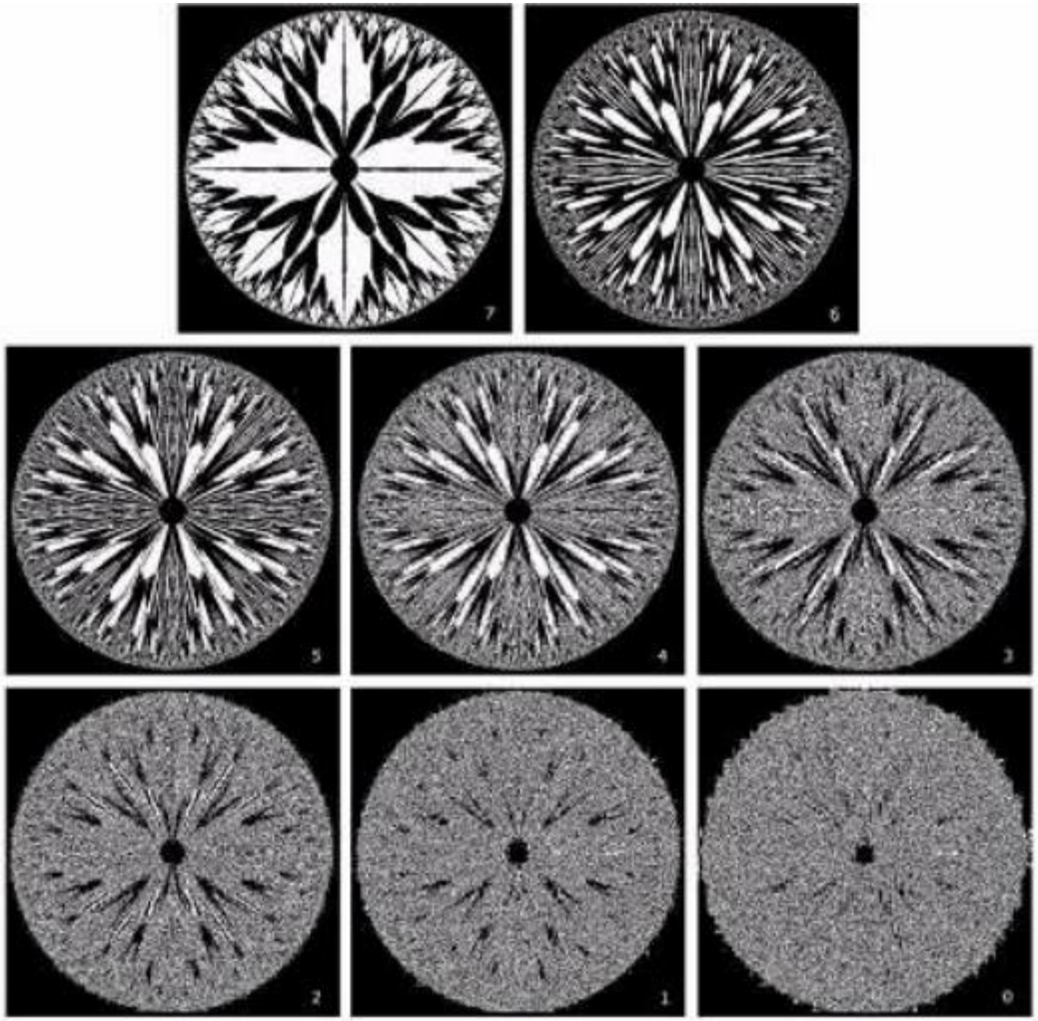
- The (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation.

  -Map all levels between 0 and 127 to 0

  -Map all levels between 128 and 255 to 255

Cont.

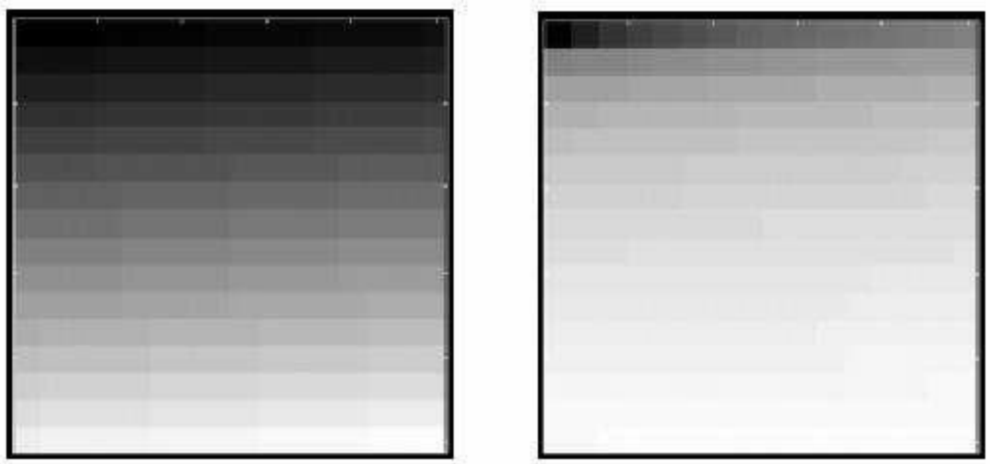
8-bit plane image



Bit-plane 7	Bit-plane 6	
Bit-plane 5	Bit-plane 4	Bit-plane 3
Bit-plane 2	Bit-plane 1	Bit-plane 0

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

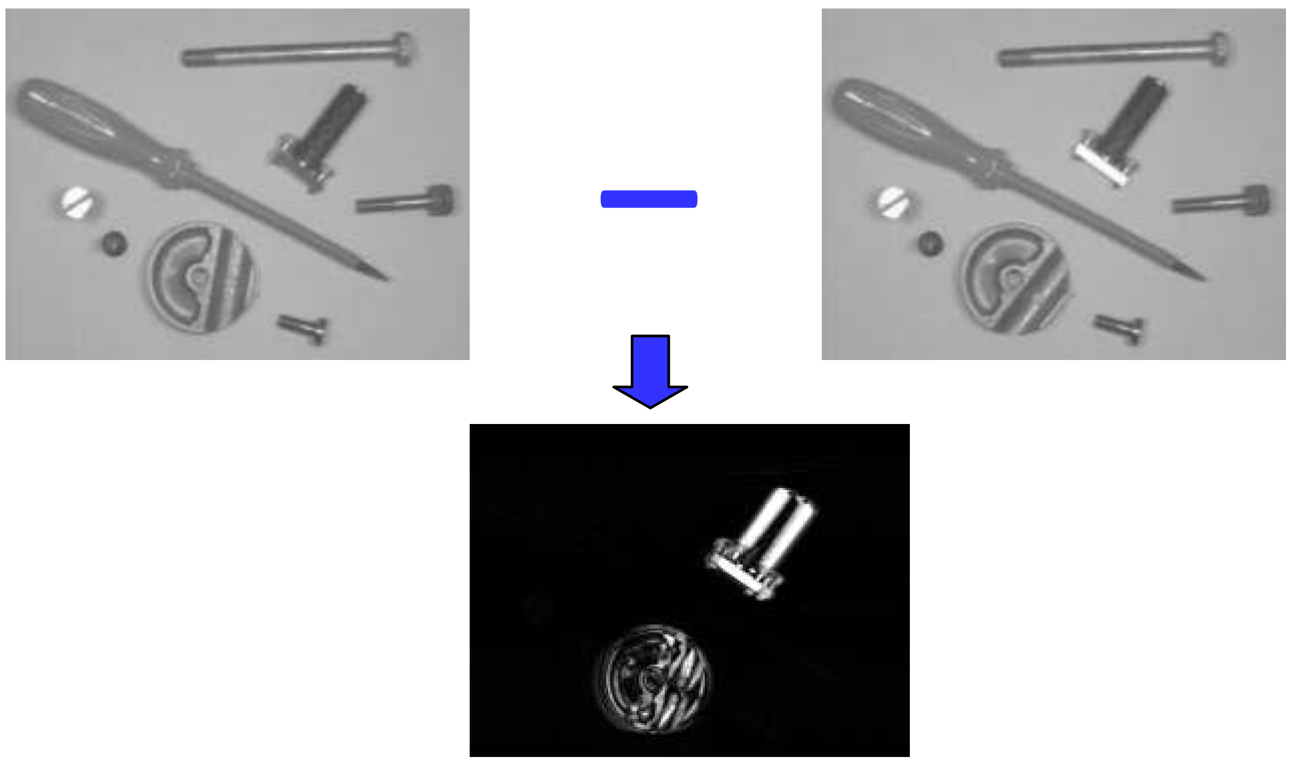
- An effective way to compress the dynamic range of pixel values is to perform the following intensity transformation function:

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

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

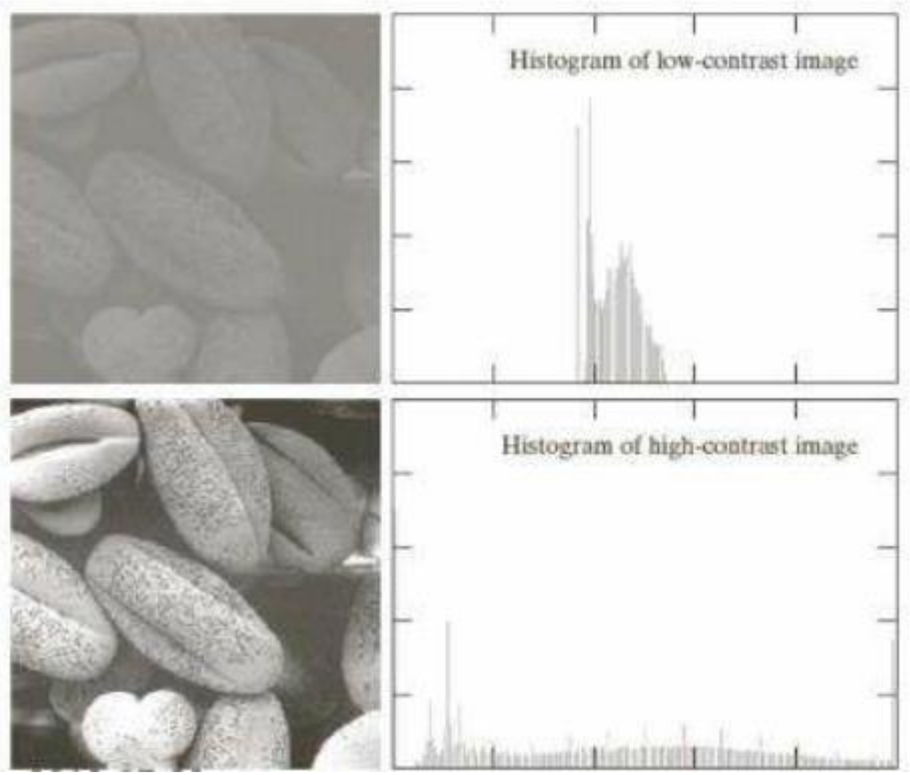
7-Image subtraction and change detection

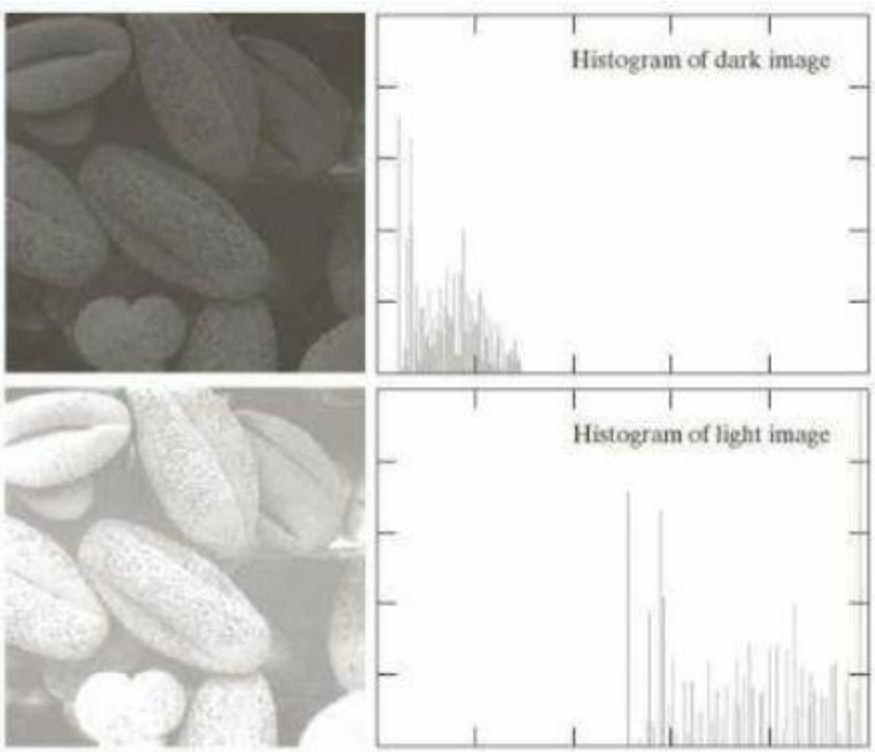
- In many imaging applications it is desired to compare two complicated or busy images .
- A simple ,but powerful method is to align the two images and subtract them. The difference image is then enhanced.
- Applications such as imaging of the blood vessils and arteries in a body , security monitoring systems .
- Example:



8. Histogram modeling

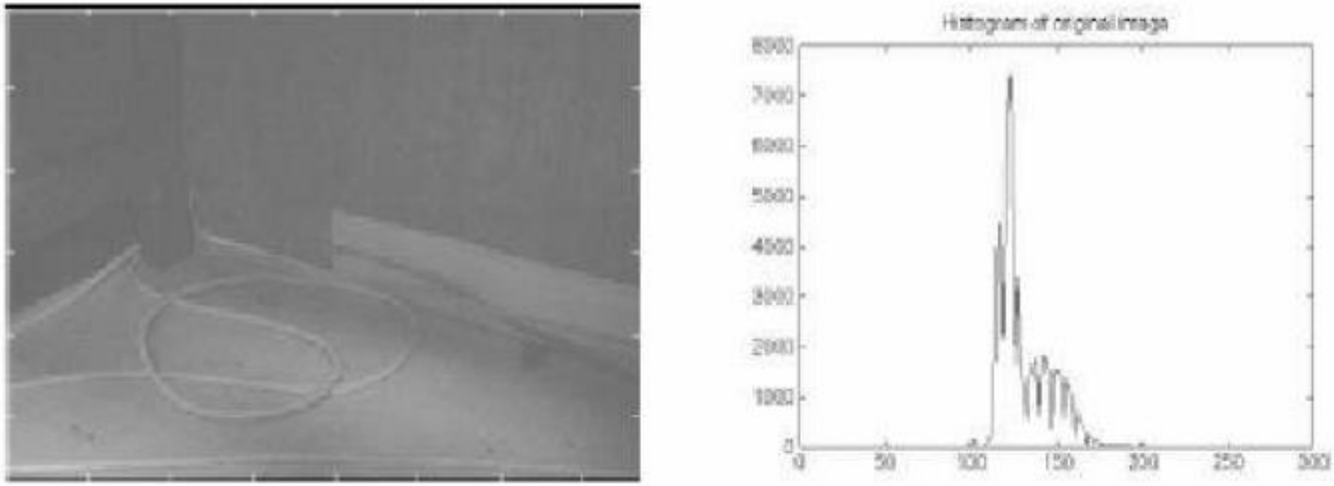
Histogram modeling techniques modify an image so that it's histogram has a desired shape. This is useful in stretching the low contrast levels with narrow histograms.



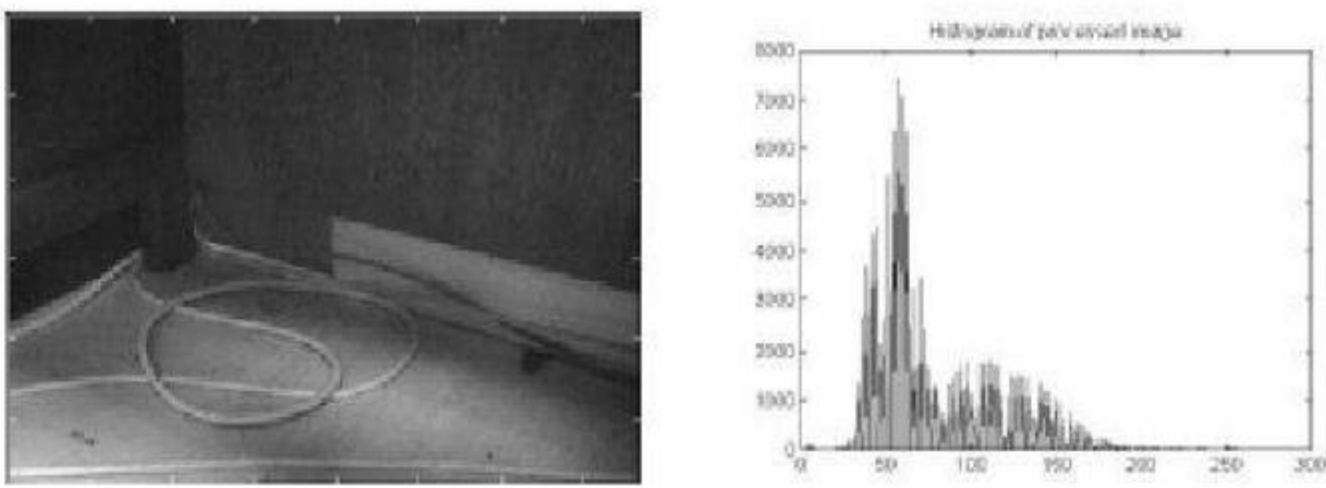


It is possible to develop a transformation function that can automatically achieve this effect ,based on histogram of input image .

Example2



Original



Processed image

Histogram modeling Cont.

Image Histogram

- Assume we have image with n_k pixels with intensity r_k, $k = 0, 1, \ldots, L-1$
- We define the image histogram as $h(r_k) = n_k$, and we define the normalized histogram as:

$$p(r_k) = h(r_k) / (M \cdot N)$$

- The normalized histogram is an estimate of the probability of occurrence of intensity level r_k in an image
- The normalized histogram sums to 1

8.1-Histogram equalization

- The objective is to map an input image to an output image such that its histogram is uniform after the mapping.
- Let *r* represent the gray levels in the image to be enhanced and s is the enhanced output with a transformation of the form ***s=T(r)***
- Assumptions

1. $T(r)$ is single-valued and monotonically increasing in the interval [0,1], which preserves the order from black to white in the gray scale.
2. $0 \le T(r) \le 1$ for $0 \le r \le 1$, which guarantees the mapping is consistent with the allowed range of pixel values.

- Possible for multiple values of ***r*** to map to a single value of ***s.***

Histogram Equalization cont.

Example 1:

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in following table.

Get the histogram equalization transformation function and give the $p_s(s_k)$ for each s_k.

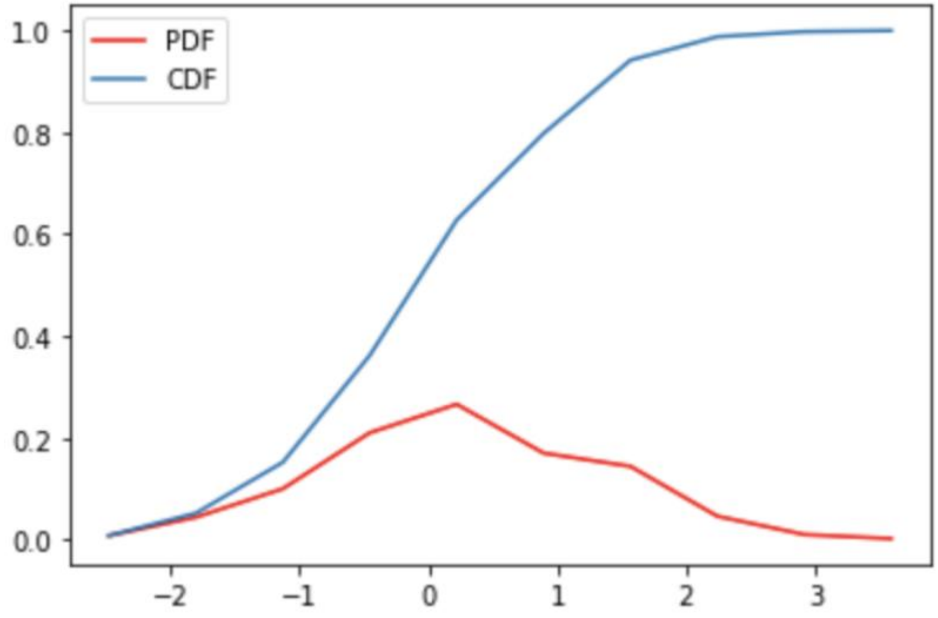
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

- **Histogram equalization** is achieved by having a transformation function ( ), which can be defined to be the **Cumulative Distribution Function (CDF) of a given Probability Density Function (PDF) of a gray-levels in a given image** *(histogram of an image can be considered as the approximation of the PDF of that image)*
- PDF represents the **probability with areas**
- The CDF represents probability with vertical distances

- **What is the relationship between CDF and PDF?**

- PDF is the derivative of a CDF
- pdf) f(x) of a continuous random variable x

  = derivative of the cdf F(x)
- The pdf **f(x)≥0, for all x**
- the area under the curve of a PDF between negative infinity and x is equal to the value of x on the CDF



Example on histogram equalization (Continued..)

(a) r_k	(b) n_k	(c) $p_r(r_k)$
0	790	0.19
1/7	1023	0.25
2/7	850	0.21
3/7	656	0.16
4/7	329	0.08
5/7	245	0.06
6/7	122	0.03
1	81	0.02
total	4096	1.00

(d) Cdf = s_k	(e) Quant. Values
0.19	1/7
0.44	3/7
0.65	5/7
0.81	6/7
0.89	6/7
0.95	1
0.98	1
1.00	1

(a) Quantized Gray levels; (b) a sample histogram; (c) its pdf; (d) Computed CDF and (e) approximated to the nearest gray level.

Solution

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

$$\boxed{s_k = \frac{L-1}{MN} \sum_{j=0}^{k} n_j}$$

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7 \times 0.19 = 1.33 \quad \rightarrow 1$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7 \times (0.19 + 0.25) = 3.08 \quad \rightarrow 3$$

$$s_2 = 4.55 \quad \rightarrow 5 \qquad s_3 = 5.67 \quad \rightarrow 6$$

$$s_4 = 6.23 \quad \rightarrow 6 \qquad s_5 = 6.65 \quad \rightarrow 7$$

$$s_6 = 6.86 \quad \rightarrow 7 \qquad s_7 = 7.00 \quad \rightarrow 7$$

Solution cont.

(a) r_k	(b) n_k	(c) $p_r(r_k)$
0	790	0.19
1/7	1023	0.25
2/7	850	0.21
3/7	656	0.16
4/7	329	0.08
5/7	245	0.06
6/7	122	0.03
1	81	0.02
total	4096	1.00

(d) Cdf $= s_k$	(e) Quant. Values
0.19	1/7
0.44	3/7
0.65	5/7
0.81	6/7
0.89	6/7
0.95	1
0.98	1
1.00	1

final transform:

$$r_0 \rightarrow s_0 = 1 \Rightarrow 790 \text{ pixels map to } 1$$

$$r_1 \rightarrow s_1 = 3 \Rightarrow 1023 \text{ pixels map to } 3$$

$$r_2 \rightarrow s_2 = 5 \Rightarrow 850 \text{ pixels map to } 5$$

$$r_3 \rightarrow s_3 = 6 \Rightarrow 656 + 329 = 985 \text{ pixels map to } 6$$

$$r_4 \rightarrow s_4 = 6 \Rightarrow 656 + 329 = 985 \text{ pixels map to } 6$$

$$r_5 \rightarrow s_5 = 7 \Rightarrow 245 + 122 + 81 = 458 \text{ pixels map to } 7$$

$$r_6 \rightarrow s_6 = 7 \Rightarrow 245 + 122 + 81 = 458 \text{ pixels map to } 7$$

$$r_7 \rightarrow s_7 = 7 \Rightarrow 245 + 122 + 81 = 458 \text{ pixels map to } 7$$

Solution cont.

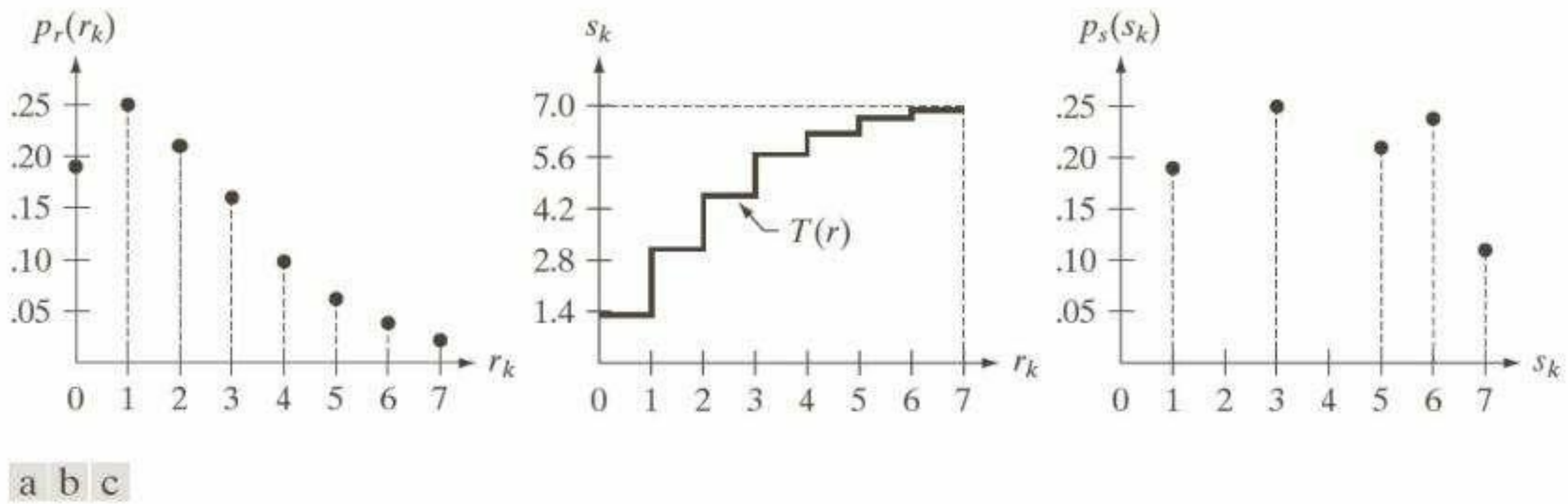
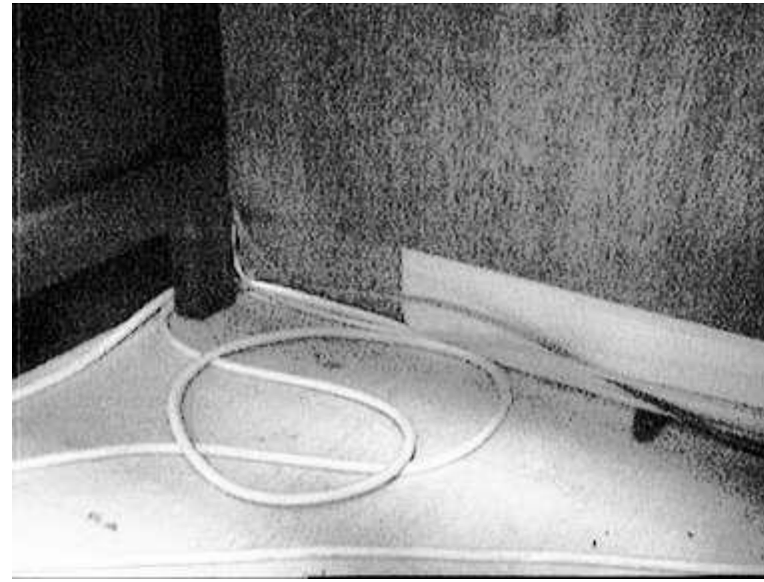
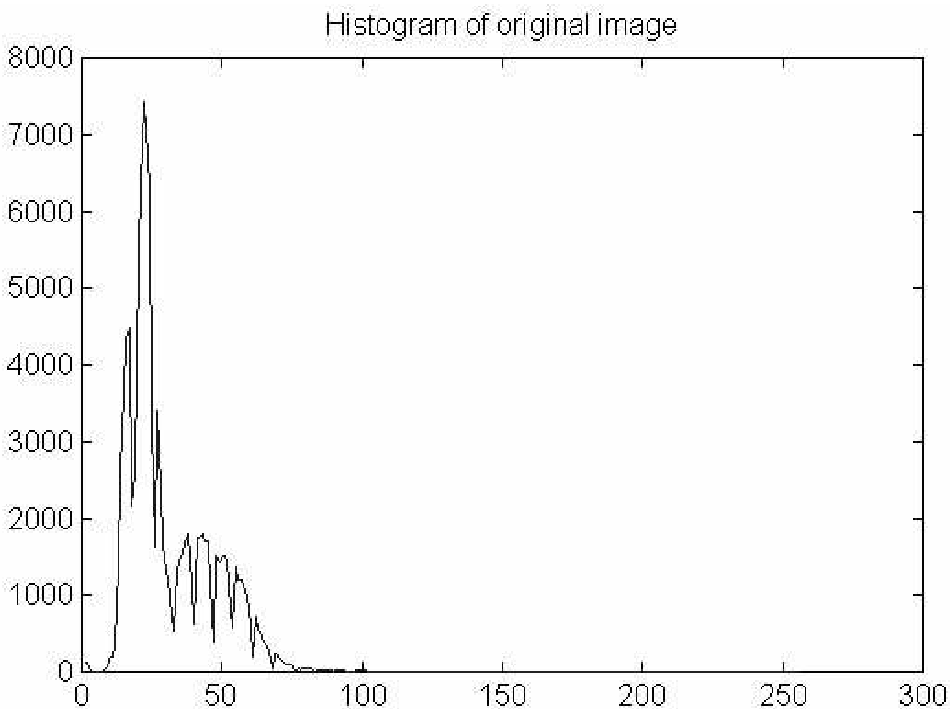
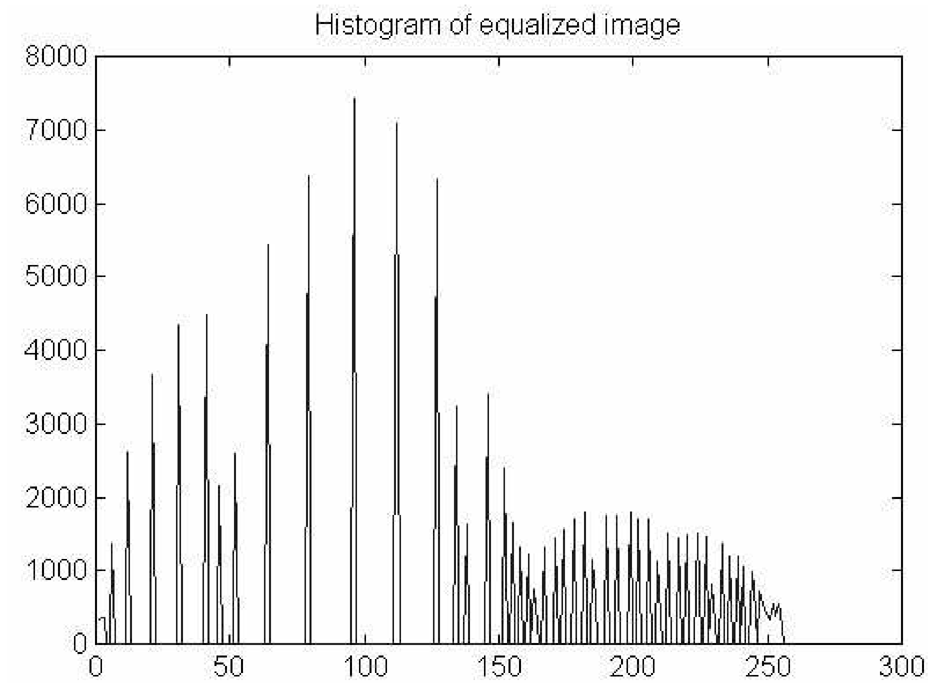


FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

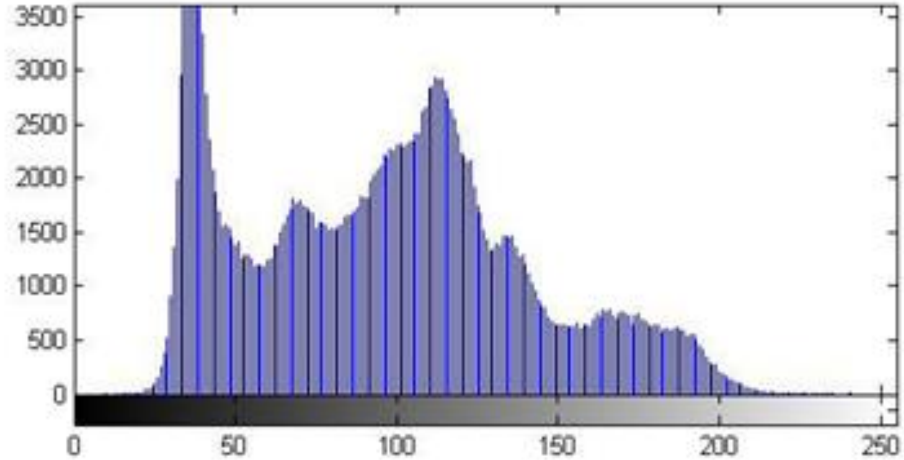
c) 1= 790/4096 = 0.19
2= 0
3= 1023/4096= 0.24
4= 0
5= 850/4096= 0.21
6= 985/4096 = 0.24
7= 458/4096 = 0.11

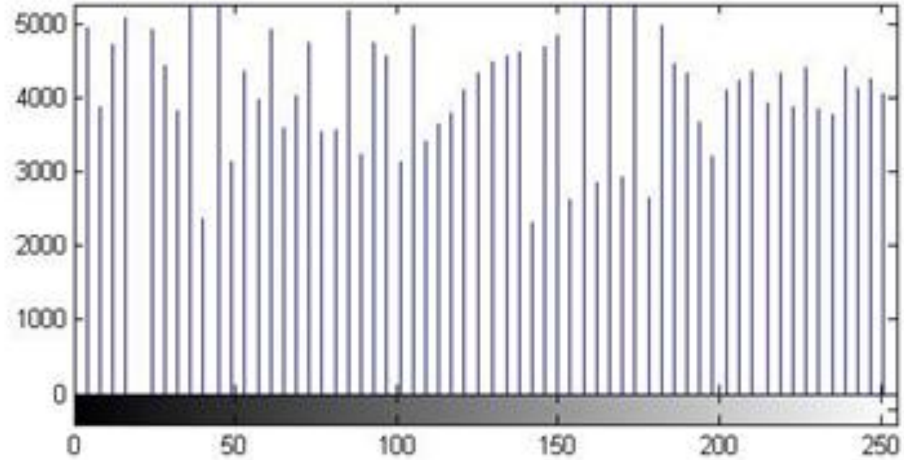



Histogram of original image
8000
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300


Histogram of equalized image
8000
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300

Histogram Equalization





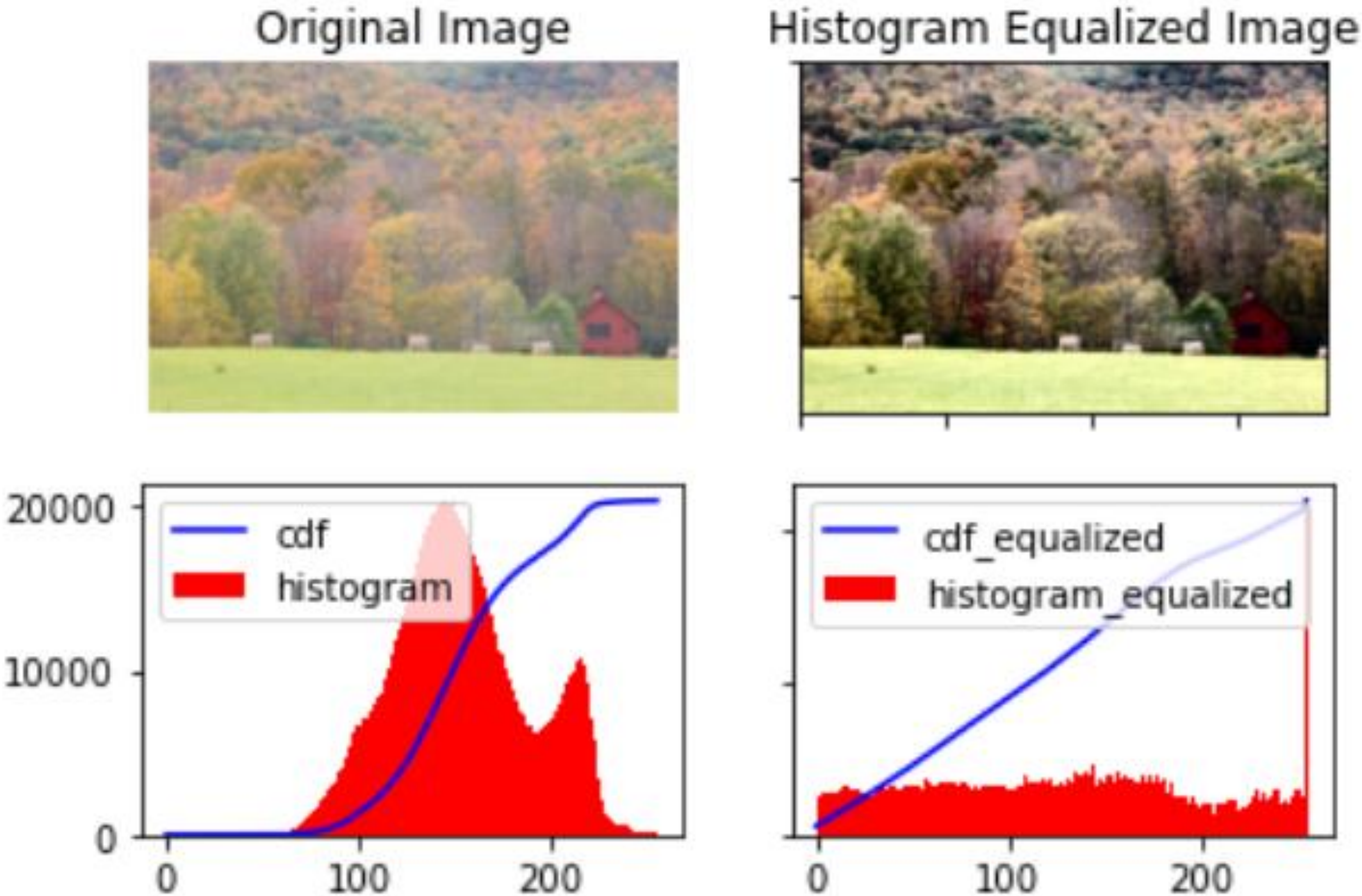

Original Image
Histogram Equalized Image
cdf
histogram
cdf_equalized
histogram_equalized
20000
10000
0
0
100
200
0
100
200

Table 1: Comparison of Spatial Domain Techniques

Techniques	Advantages	Disadvantages
Point Processing operation	Can only be used for linear stretching	Cannot produce much attractive results in many cases
Histogram equalization	This technique is best for visual perception especially when image have close contrast data, produces best result for radiographic and thermal images.	This technique results in noise amplification when the images has major low intensity area.

Sr.No.	Contrast Stretching	Histogram Equalization
1.	Contrast stretching is all about increasing the difference between the maximum and the minimum intensity value in an image	Histogram equalization is about modifying the intensity values of all the pixels in the image such that the histogram is "flattened"
2.	The transformation function used in contrast stretching is selected manually based on the requirement of the application.	Histogram equalization derives the transformation function automatically from probability density function (PDF) of the given image
3.	The transformation function has to be specified in order to do contrast stretching	There is no need to specify the transformation function in the case of Histogram equalization.
4.	Once an image undergoes contrast stretching, the original image can be obtained back.	Once histogram equalization is performed, the original image cannot be obtained back.
5.	Image enhancement is lesser in contrast stretching	Image enhancement in more in Histogram equalization
6.	Contrast stretching applies only a linear scaling function to the input image.	In histogram equalization, the input image is scaled such that a nearly equalized histogram is obtained.

Example 2: Equalizing an image of **6** gray levels

Index k	0	1	2	3	4	5
Normalized Input level, $r_k/5$	0.0	0.2	0.4	0.6	0.8	1.0
Freq. Count of r_k, n_k	**4**	**7**	**2**	**1**	**0**	**1**
Probability $P(r_k) = n_k/n$	4/15	7/15	2/15	1/15	0/15	1/15
$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n}$	4/15 = **0.27**	11/15 = **0.73**	13/15 = **0.87**	14/15 = **0.93**	14/15 = **0.93**	15/15 = **1.00**
Quantized s_k	0.2	0.8	0.8	1.0	1.0	1.0

n= 15
= 4+7+2+1+0+1

	freq
0.0:	4
0.2:	7
0.4:	2
0.6:	1
0.8:	0
1.0:	1

T

r_k		s_k	
0.0	$\rightarrow$	0.2	4
0.2	$\rightarrow$	0.8	7
0.4	$\rightarrow$	0.8	2
0.6	$\rightarrow$	1.0	1
0.8	$\rightarrow$	1.0	0
1.0	$\rightarrow$	1.0	1

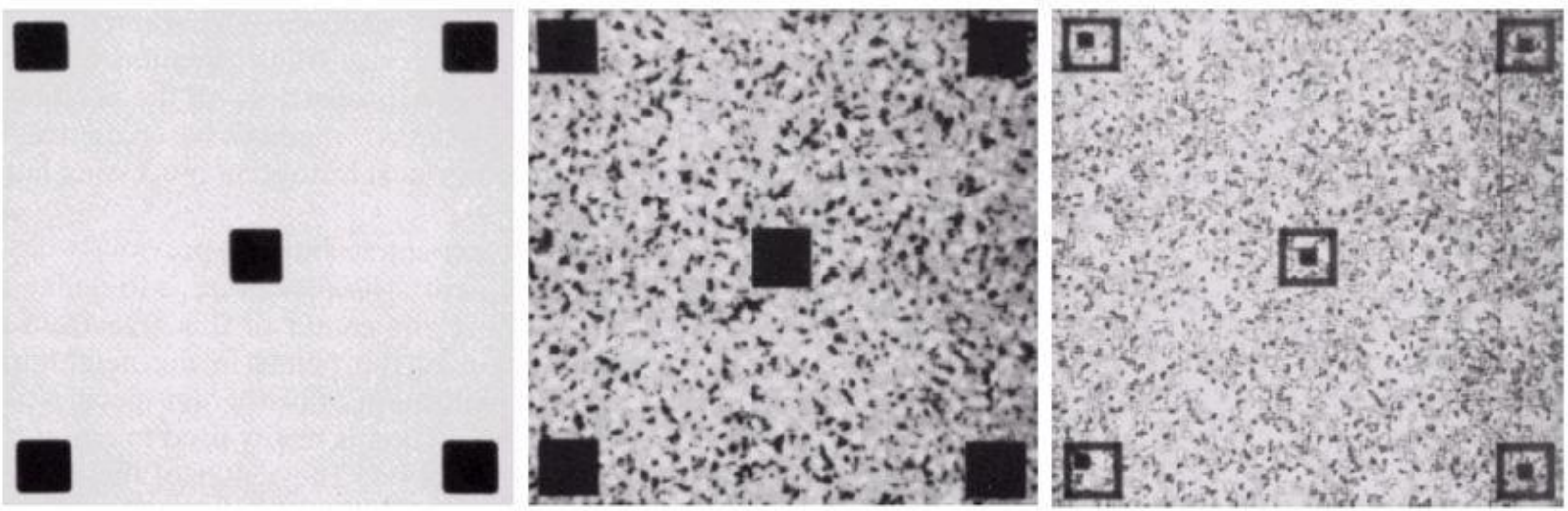
	freq	
0.0:	0	
0.2:	4	
0.4:	0	
0.6:	0	
0.8:	9	$\rightarrow$ 7+2
1.0:	2	$\rightarrow$ 1+1

Local Histogram Processing

- Entire process is same as global histogram processing, only difference is mask size.
- In case of global histogram processing, the mask size is M*N
- In case of local hidtogram processing mask size can specifed which is << M*N
- Neighborhood mask is moved over image with pixel by pixel at centre and corresponding histogram processing is carried out.
- Mask is generally not a nonoverlapping which produces a blocky effect

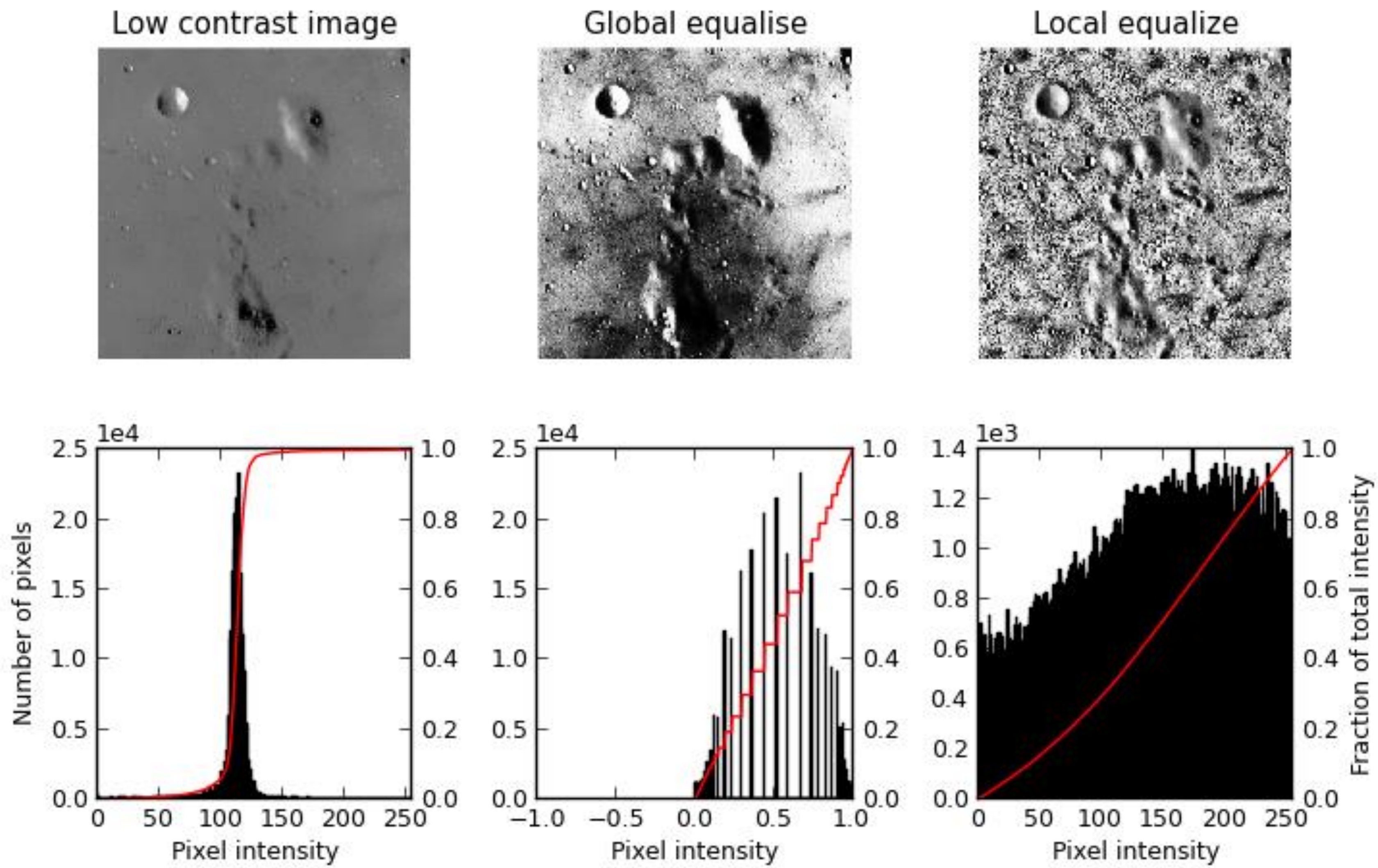
Enhancement using local statistics

- Histogram processing on a local neighborhood



a b c

FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.


Low contrast image
Global equalise
Local equalize
1e4
1e4
1e3
Number of pixels
Fraction of total intensity
Pixel intensity
Pixel intensity
Pixel intensity
0.0
0.5
1.0
1.5
2.0
2.5
0
50
100
150
200
250
−1.0
−0.5
0.0
0.5
1.0
0.2
0.4
0.6
0.8
1.2
1.4

Histogram Specification

- Histogram equalization only generates an approximation to a uniform histogram
- With Histogram specification, we can specify the shape of the histogram that we wish the output image to have
- It need not to be a uniform histogram
- The principal difficulty in applying the histogram specification method to image enhancement lies in being able to construct a meaningful histogram

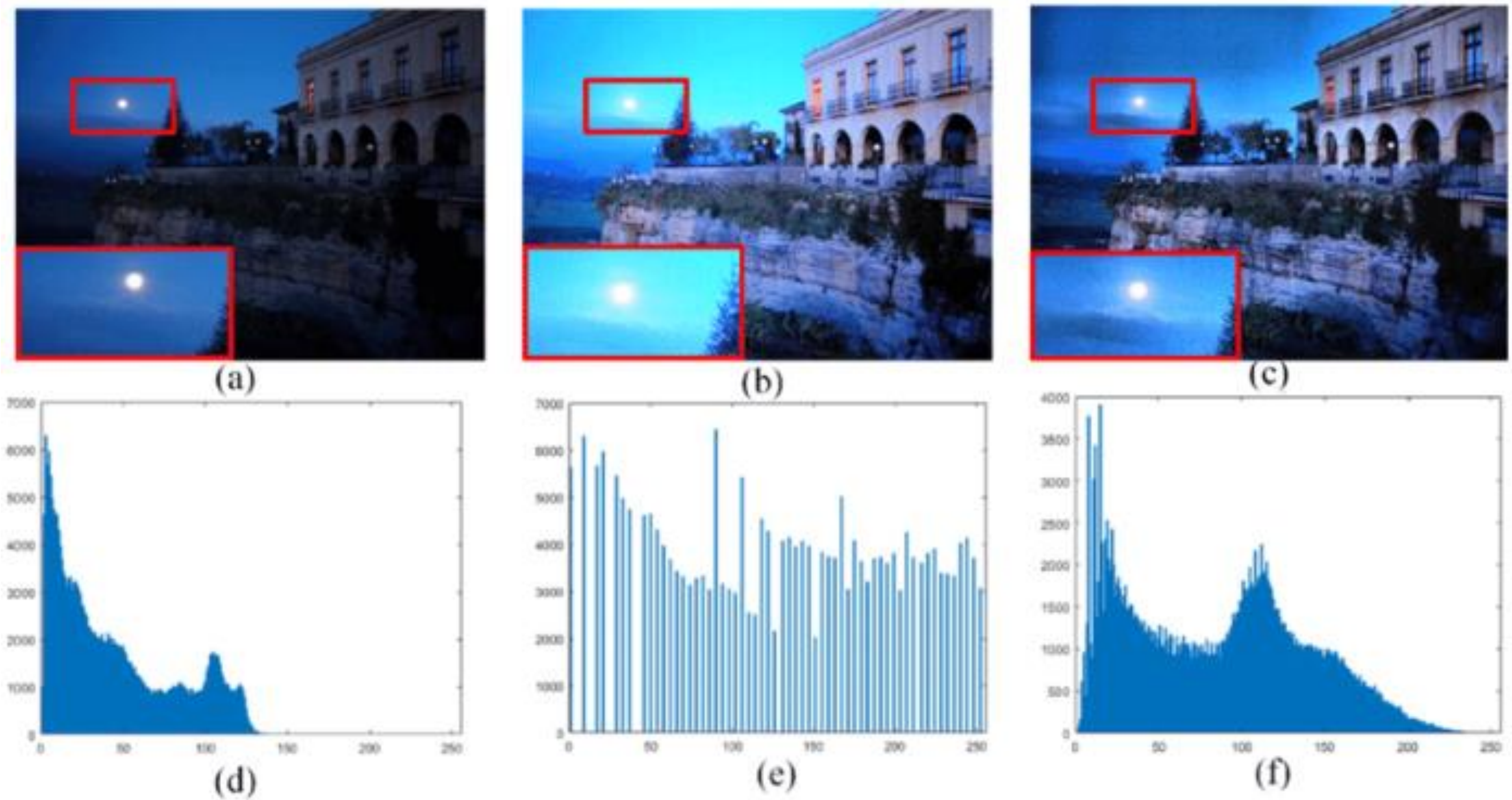


Image enhancement based on histogram specification. a is original image, b and c are the enhancement results of histogram equalization (HE) and adaptive histogram equalization (AHE), respectively. d-f are the histograms of (a-c), respectively

Histogram specification cont.

- histogram equalization gives:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad 0 \le k \le L-1 \quad (5)$$

- given the value of s_k, we can solve for z_q as:

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) \qquad (6)$$

- for a value of q, so that:

$$G(z_q) = s_k \qquad (7)$$

- we can find z_q as:

$$z_q = G^{-1}(s_k) \qquad (8)$$

- this is the mapping from s to z

Histogram specification cont.

Procedure :

1. compute histogram, $p_r(r)$, of image; find histogram equalization transformation:

$$s_k = T(r_k) = \frac{(L-1)}{M \cdot N} \sum_{j=0}^{k} n_j \quad 0 \le k \le L-1$$

round all values of s_k to nearest integer in $[0, L-1]$

2. compute all values of transformation G

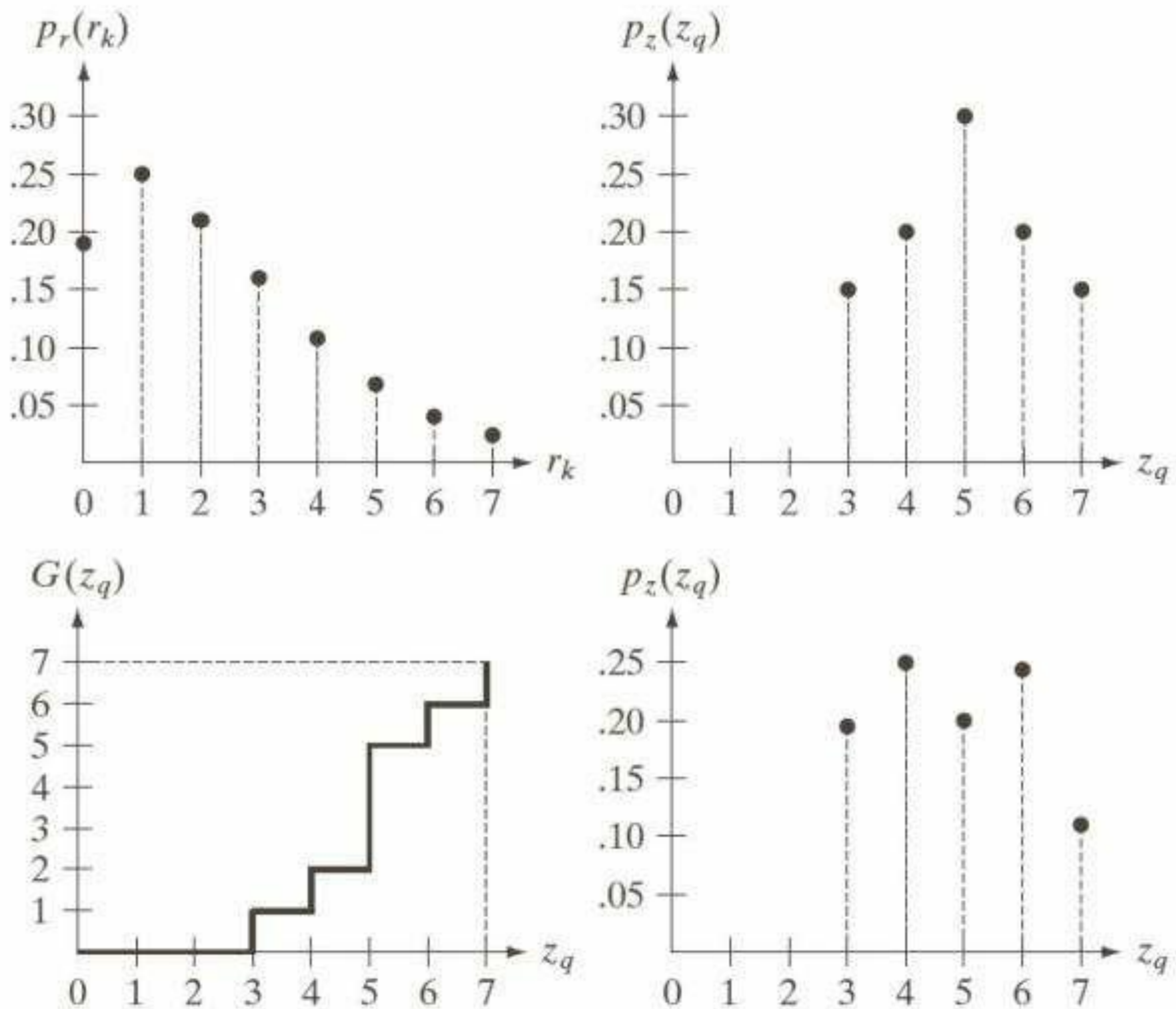
$$G(z_q) = (L-1) \sum_{i=0}^{q} p_z(z_i)$$

for $q = 0, 1, \ldots, L-1$ where $p_z(z_i)$ are values of specified histogram; round values of G to integers in range $[0, L-1]$; store values of G in table

3. for every value of s_k, $0 \le k \le L-1$, use stored values of G to find corresponding value of z_q so that $G(z_q)$ is closest to s_k, and store these mappings from s to z. For multiple z_q matches to s_k, choose smallest z_q.

4. form histogram matched image by first histogram equalizing input image and then mapping every equalized pixel value, s_k, to the corresponding value, z_q, using the mappings in Step 3.

Example



a	b
c	d

FIGURE 3.22
(a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

Example cont.

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

- first obtain scaled histogram equalized values:

  $s_0 = 1;\ s_1 = 3;\ s_2 = 5;\ s_3 = 6;\ s_4 = 6;\ s_{5,6,7} = 7$

- next compute and round all values of transformation G:

  $G(z_0) = 0 \rightarrow 0;\ G(z_1) = 0 \rightarrow 0;\ G(z_2) = 0 \rightarrow 0;$

  $G(z_3) = 1.05 \rightarrow 1;\ G(z_4) = 2.45 \rightarrow 2;\ G(z_5) = 4.55 \rightarrow 5;$

  $G(z_6) = 5.95 \rightarrow 6;\ G(z_7) = 7 \rightarrow 7$

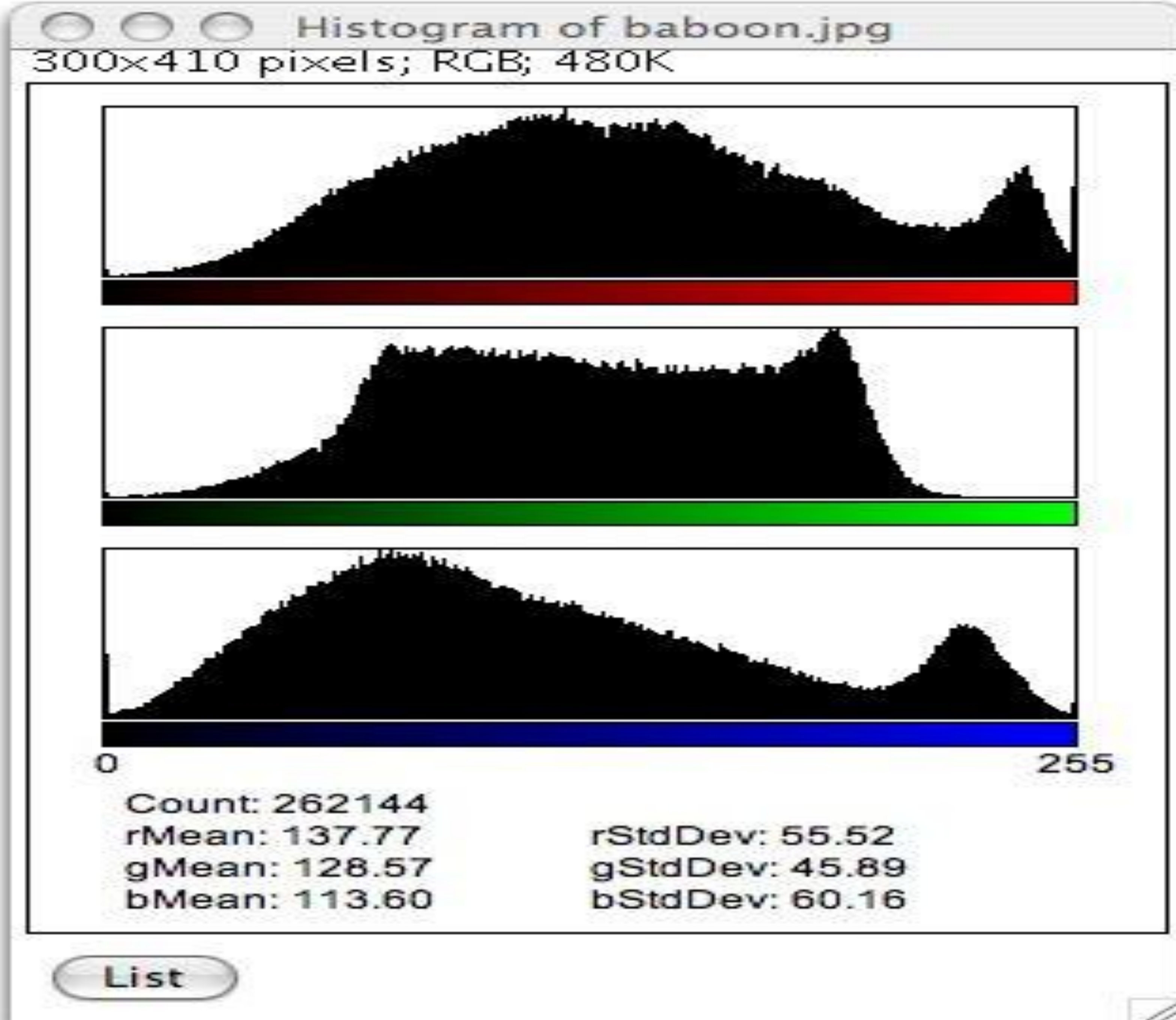
- need to find smallest value of z_q so that $G(z_q)$ is closest to s_k; do this for all s_k to create required mapping:

  $s_1 \rightarrow z_3;\ s_3 \rightarrow z_4;\ s_5 \rightarrow z_5;\ s_6 \rightarrow z_6;\ s_7 \rightarrow z_7$

k	s_k	$Q[s_k]$	z_q	$G(z_q)$	$Q[G(z_q)]$
0	1.33	1	0	0	0
1	3.08	3	1	0	0
2	4.55	5	2	0	0
3	5.67	6	3	1.05	1
4	6.23	6	4	2.45	2
5	6.65	7	5	4.55	5
6	6.86	7	6	5.95	6
7	7	7	7	7	7

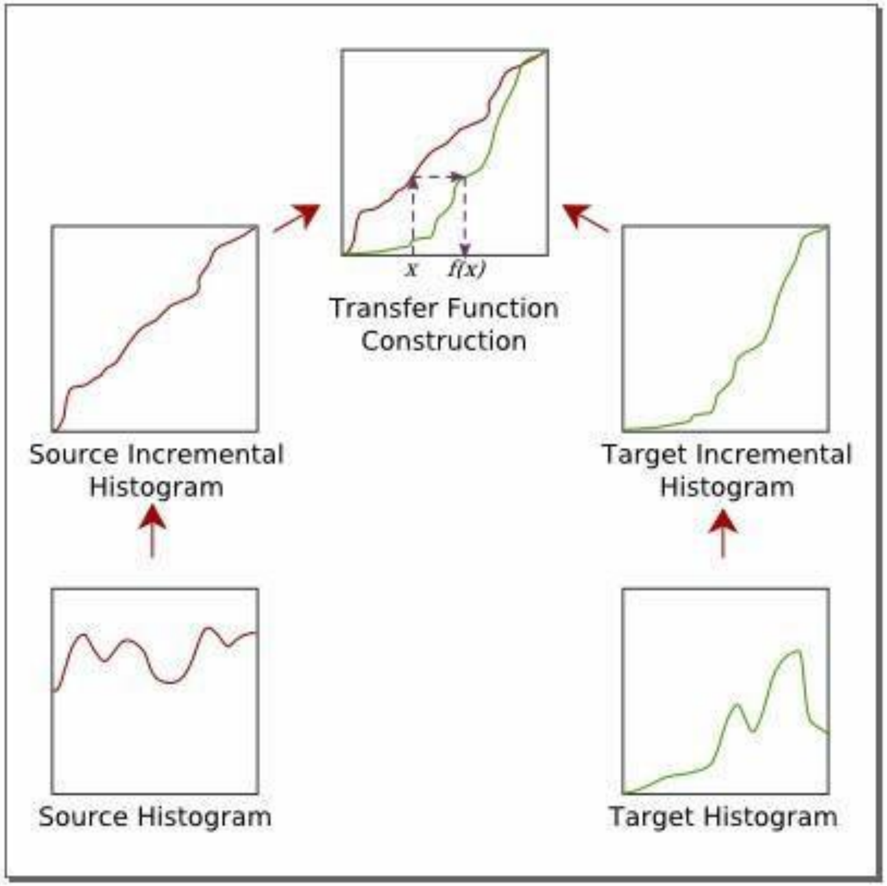
- Find smallest value of z_q so that $G(z_q)$ is closest to s_k

s_k	$G(z_q)$	z_q
1	1	3
3	2	4
5	5	5
6 6	6	6
7 7 7	7	7



Histogram of baboon.jpg
300x410 pixels; RGB; 480K
0
255
Count: 262144
rMean: 137.77
gMean: 128.57
bMean: 113.60
rStdDev: 55.52
gStdDev: 45.89
bStdDev: 60.16
List

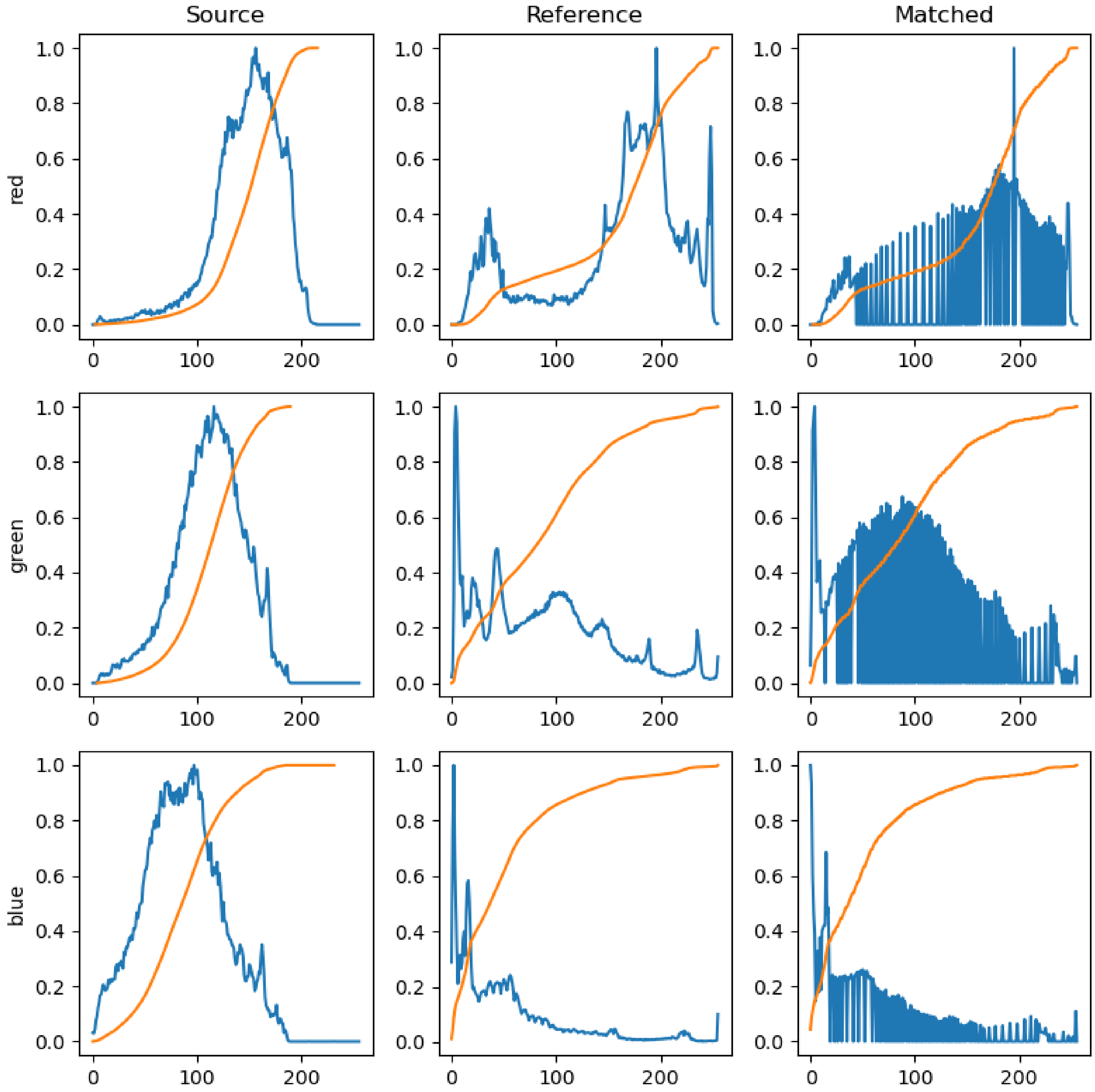
Histogram Matching



Histogram Matching or Histogram Specification is **the transformation of an image so that its histogram matches a specified histogram**.

-e.g. It manipulates the pixels of an input image so that its histogram matches the histogram of the reference image. If the images have multiple channels, the matching is done independently for each channel, as long as the number of channels is equal in the input image and the reference.

- Histogram matching can be used as a lightweight normalization for image processing, such as feature matching, especially in circumstances where the images have been taken from different sources or in different conditions


Source
Reference
Matched
red
green
blue
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200

Spatial and Frequency Domains

- Spatial domain
  - refers to planar region of intensity values at time t

- Frequency domain
  - think of each color plane as a sinusoidal function of changing intensity values
  - refers to organizing pixels according to their changing intensity (frequency)

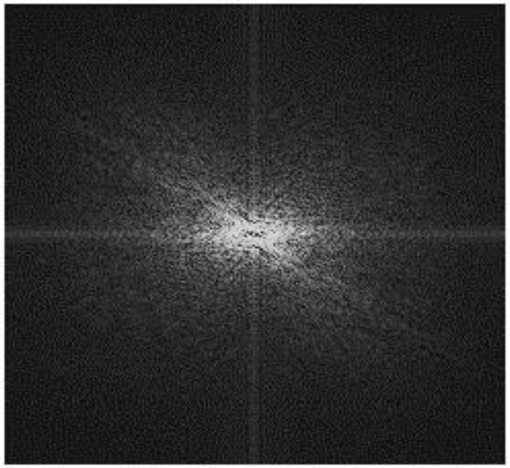
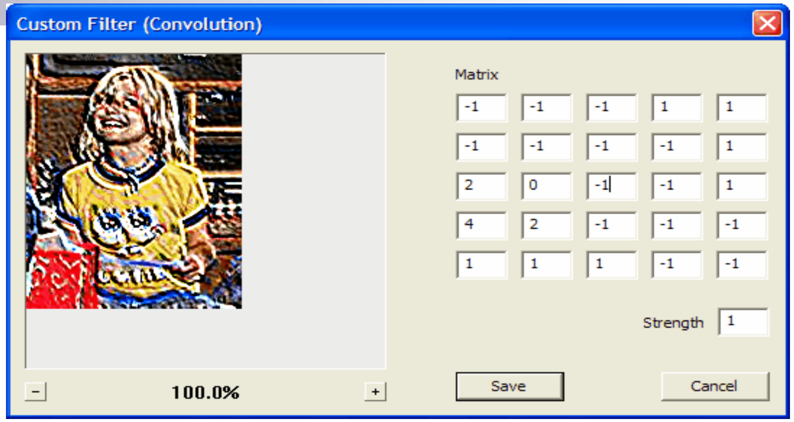


Image Processing Function: 1. Filtering

- Filter an image by replacing each pixel in the source with a weighted sum of its neighbors
- Define the filter using a *convolution mask*, also referred to as a *kernel*
  - non-zero values in small neighborhood, typically centered around a central pixel
  - generally have odd number of rows/columns

Convolution Filter



100	100	100	100	100
100	100	50	50	100
100	100	100	100	100
100	100	100	100	100
100	100	100	100	100

X

	0	1	0	
	0	0	0	
	0	0	0	

=

100	100	100	100	100
100	100	50	50	100
100	100	50	100	100
100	100	100	100	100
100	100	100	100	100

Convolution is the treatment of a matrix by another one which is called "kernel".

Mean Filter

20	12	14	23
45	15	19	33
55	34	81	22
8	64	49	95

Subset of image

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

Convolution filter

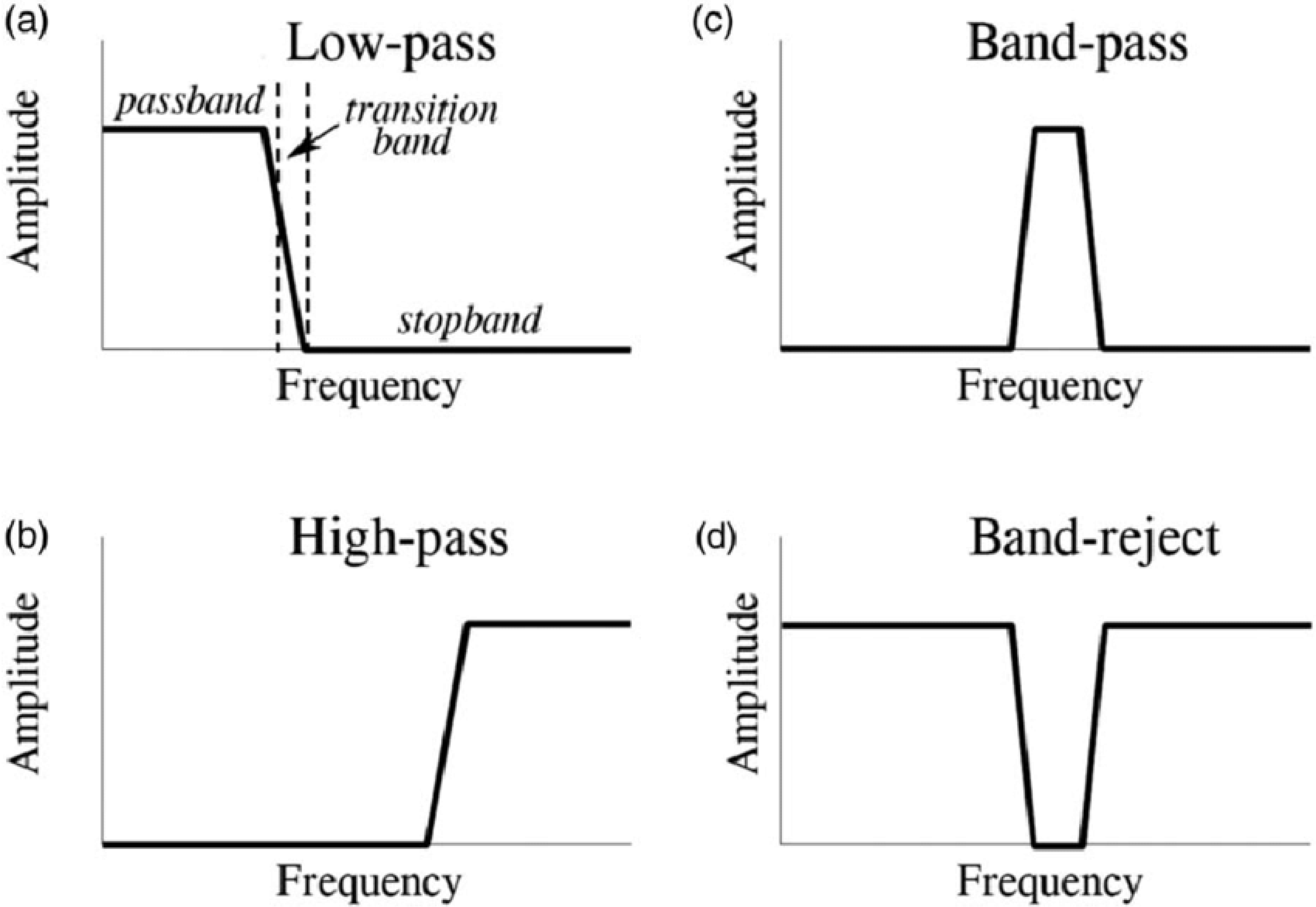
Common 3x3 Filters

- Low/High pass filter $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
- Blur operator $\frac{1}{13}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}$
- Edge detector $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$ $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

Table 2: Comparison Of Frequency Domain Techniques

Techniques	Advantages	Disadvantages
Low Pass Filter	The Low Pass Filter is good for removing a small amount of high frequency noise from an *N* dimensional signal.	It suffers from two problem : Blurring and Ringing caused due to undulation assciated with spatial domain filter.
High Pass Filter	The High Pass Filter is good for removing a small amount of low frequency noise from an *N* dimensional signal	This filter is only a first-order filter, it may not give you a step enough cutoff frequency for the application you need.
Homomorphic Filter	Used to remove multiplicative and additive noise	Illumination and reflectance are not separable.

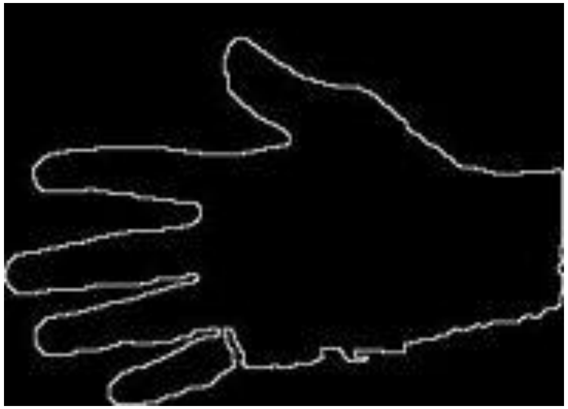
- A high pass filter tends to **retain the high frequency information within an image while reducing the low frequency information**. The kernel of the HPF is designed to increase the brightness of the center pixel relative to neighboring pixels.
- A low pass filter is the basis for most smoothing methods. An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels. Use of LPF tends to **retain the low frequency information within an image while reducing the high frequency information**.


(a)
Low-pass
passband
transition band
stopband
Amplitude
Frequency
(c)
Band-pass
Amplitude
Frequency
(b)
High-pass
Amplitude
Frequency
(d)
Band-reject
Amplitude
Frequency

Example

Image Function: 2. Edge Detection

- Identify areas of strong intensity contrast
  - filter useless data; preserve important properties
- Fundamental technique
  - e.g., use gestures as input
  - identify shapes, match to templates, invoke commands



Edge Detection

Simple Edge Detection

- Example: Let assume single line of pixels

5	7	6	4	152	148	149

- Calculate 1st derivative (gradient) of the intensity of the original data
  - **Using gradient, we can find peak pixels in image**
  - *I(x)* represents intensity of pixel *x* and
  - *I'(x)* represents gradient (in 1D),
  - Then the gradient can be calculated by convolving the original data with a mask (-1/2 0 +1/2)
  - $I'(x) = -\frac{1}{2} * I(x-1) + 0 * I(x) + \frac{1}{2} * I(x+1)$

Basic Method of Edge Detection

- Step 1: filter noise using mean filter
- Step 2: compute spatial gradient
- Step 3: mark points > *threshold* as edges

Mark Edge Points

- Given gradient at each pixel and threshold
  - mark pixels where *gradient* > *threshold* as edges