Indian Institute of Information Technology, Surat

भारतीय सूचना प्रौद्योगिकी संस्थान ,सूरत

(Institute of National Importance under Act of Parliament)

EC 503:
Image Processing and Computer Vision

By:
Dr. Hemant S. Goklani
ECE Department
IIIT, Surat

UNIT 2
Image Enhancement in Spatial Domain and Frequency Domain

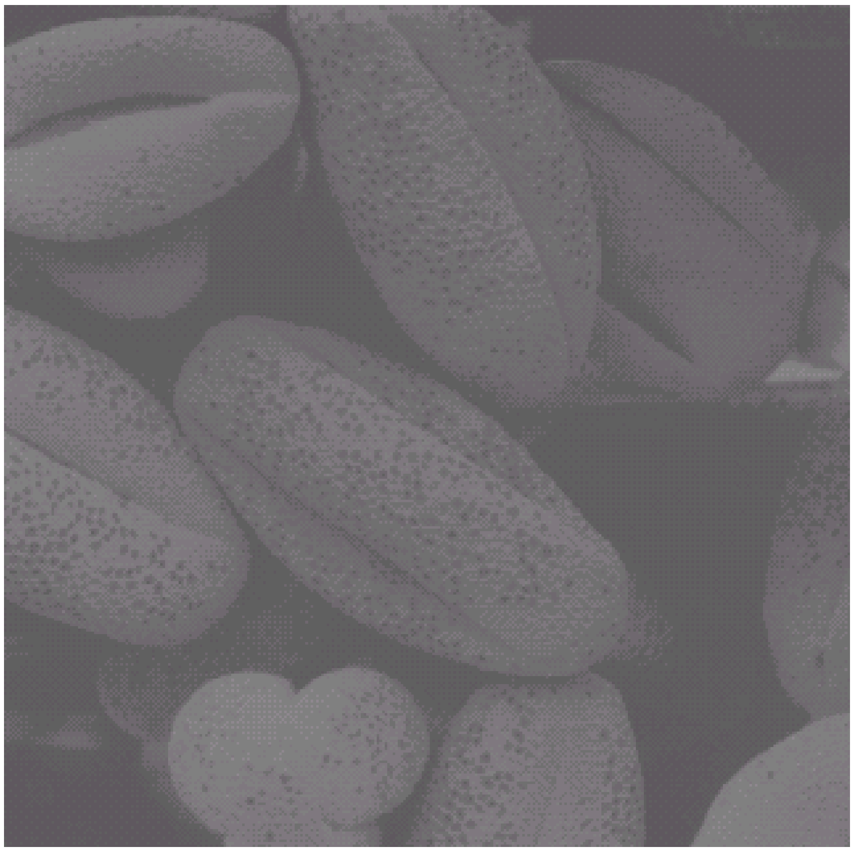
UNIT2_1:

Image Enhancement in the Spatial Domain

Image Enhancement

- **Goal**: process an image so that the result is more suitable than the original image for a specific application
- Visual interpretation
- Problem oriented

Image enhancement example



Two categories

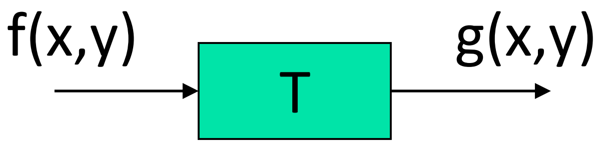
- There is no general theory of image enhancement
- Spatial domain
  - Direct manipulation of pixels
    - Point processing
    - Neighborhood processing
- Frequency domain
  - Modify the Fourier transform of an image

Outline: spatial domain operations

- Background
- Gray level transformations
- Arithmetic/logic operations

Background

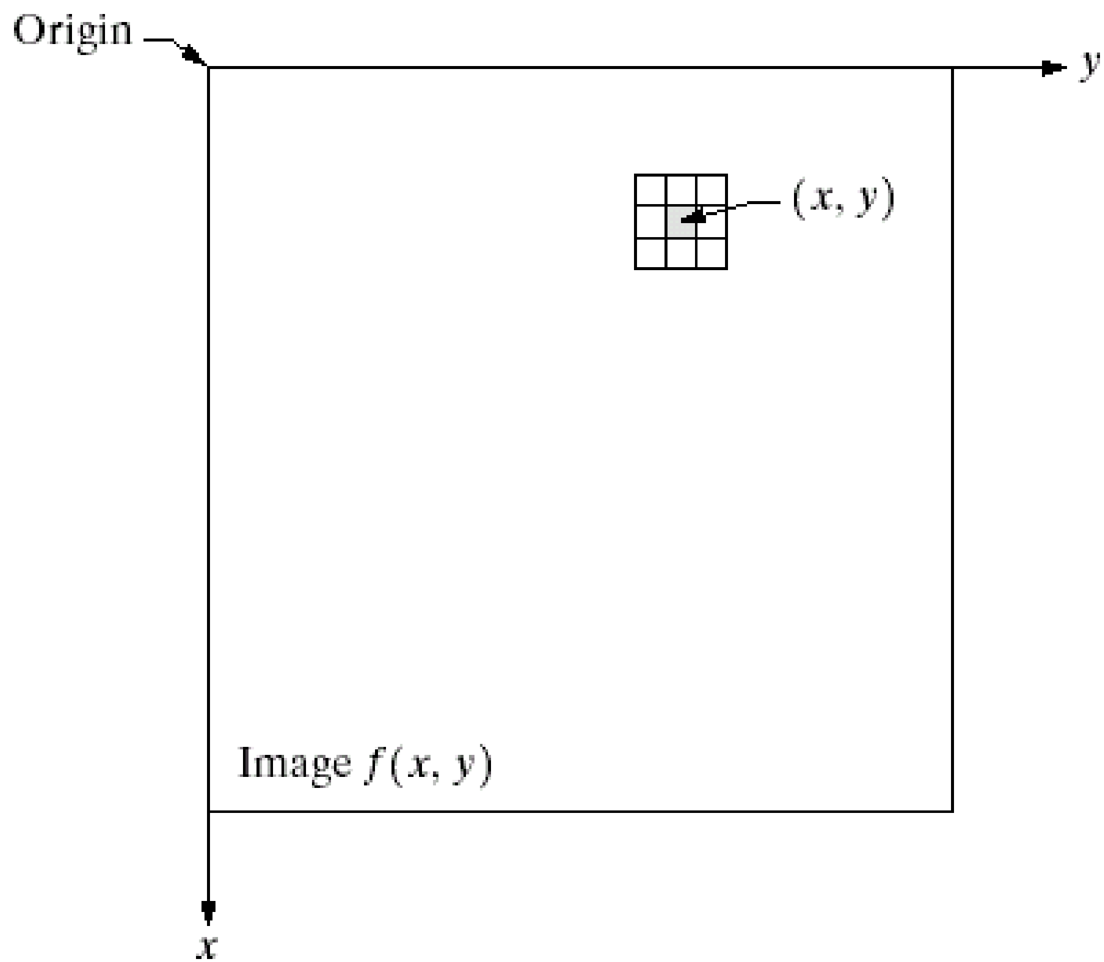
- Spatial domain processing
  - $g(x,y)=T[\,f(x,y)\,]$
  - $f(x,y)$: input image
  - $g(x,y)$: output image
  - T: operator
    - Defined over some neighborhood of (x,y)



Background (cont.)

* T applies to each pixel in the input image

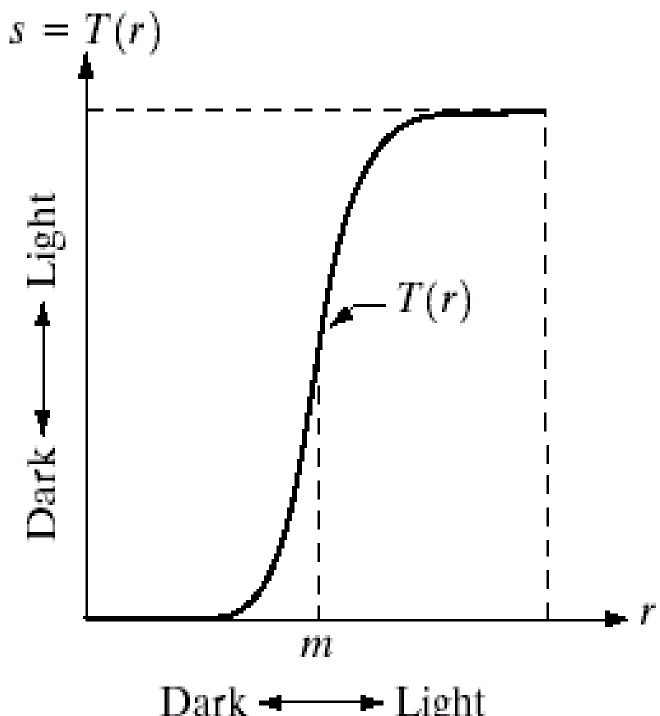
* T operates over neighborhood of (x,y)

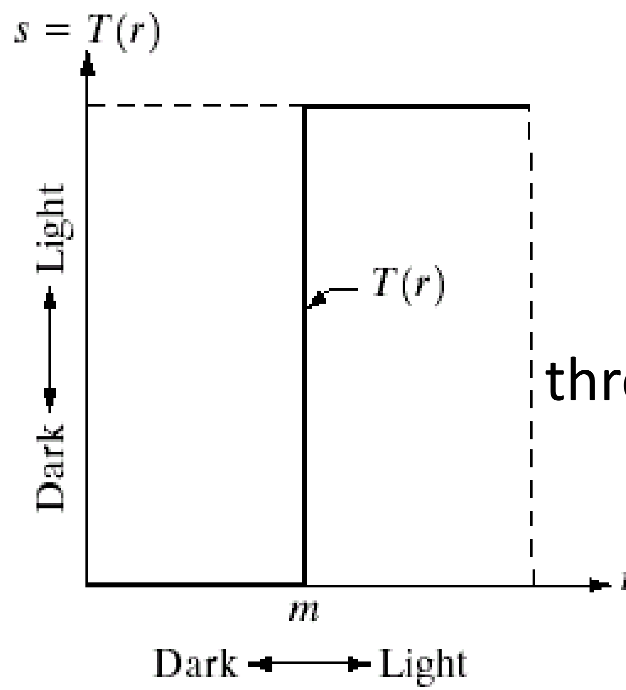


Point processing

- 1x1 neighborhood
  - Gray level transformation, or point processing
- s = T(r)   Where r is values of pixel before process
- s is values of pixel after process
- T is a transformation that maps a pixel value r into a pixel value s.

contrast stretching

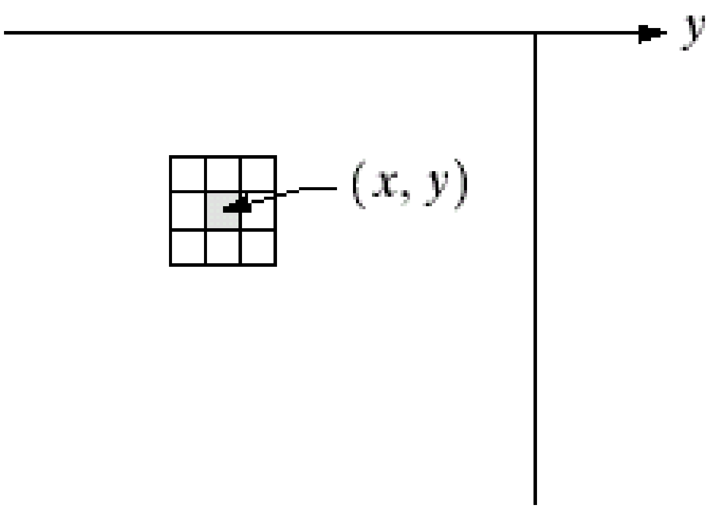




thresholding

Neighborhood processing

- A larger predefined neighborhood
  - Ex. 3x3 neighborhood
  - mask, filters, kernels, templates, windows
  - Mask processing or filtering



Some Basic Gray Level Transformations

- Image negatives (complement)
- Log transformation
- Power-law transform
- Piece-wise linear transform
- Gray level slicing
- Bit plane slicing

Some gray level transformations

- Lookup table
- Functional form

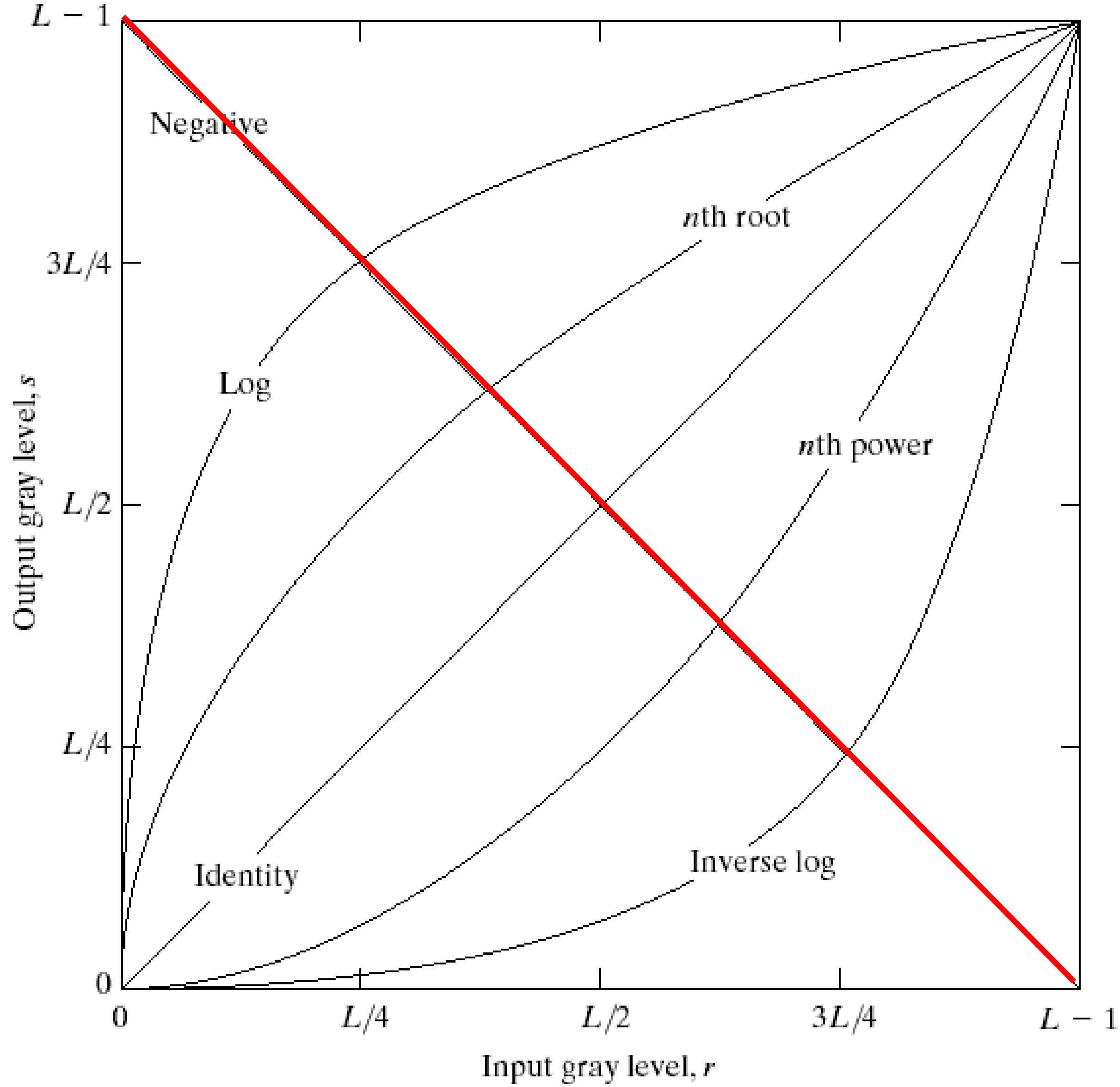
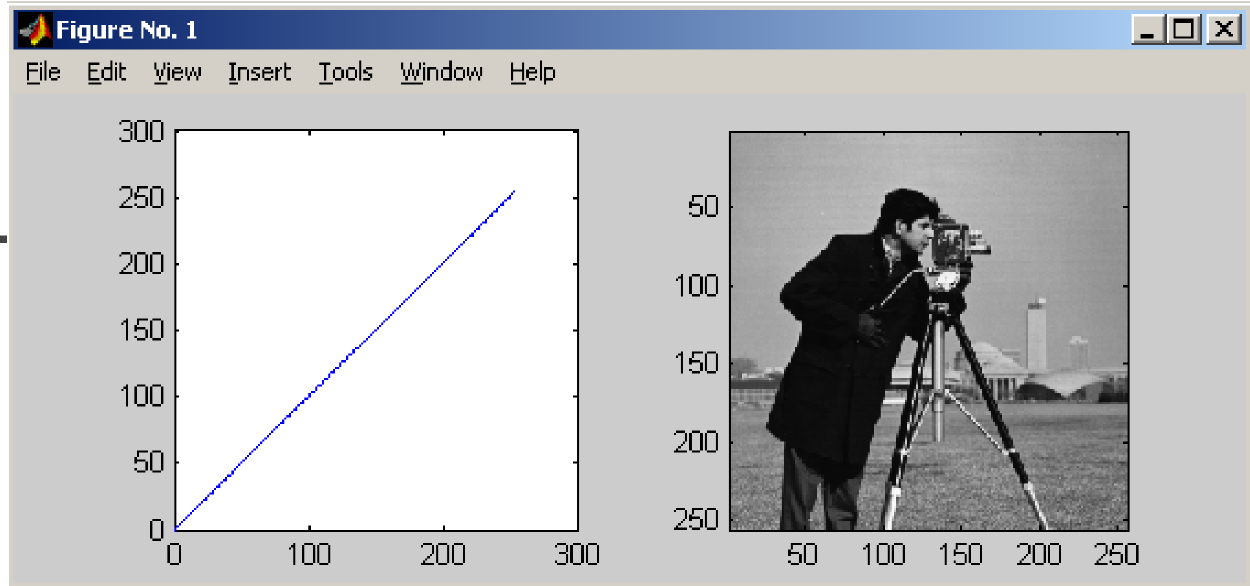


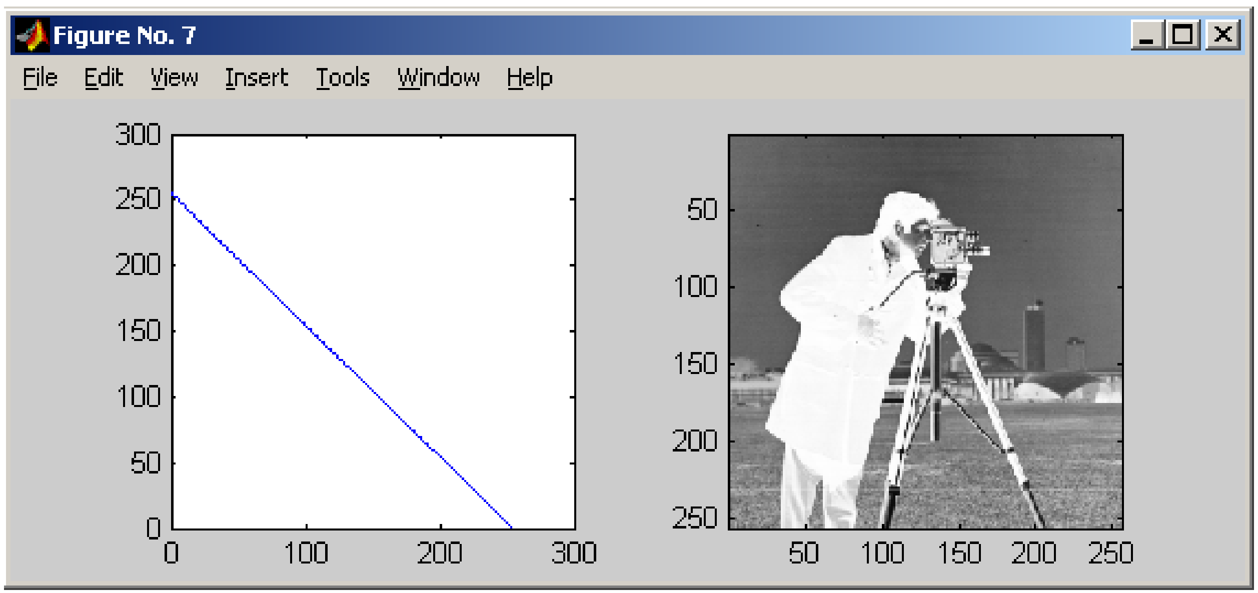
Image Negatives

- The negative of an image with gray L levels is given by the expression

$$s = L - 1 - r$$

Where r is value of input pixel, and
s is value of processed pixel





Suitable for images with dominant black areas

Log Transformations

- The general form of the log transformation shown in

$$s = c \log(1+r)$$

Where c is constant, and it is assumed that $r \geq 0$

- Compress the dynamic range of images with large variation in pixel values

C = 1.0

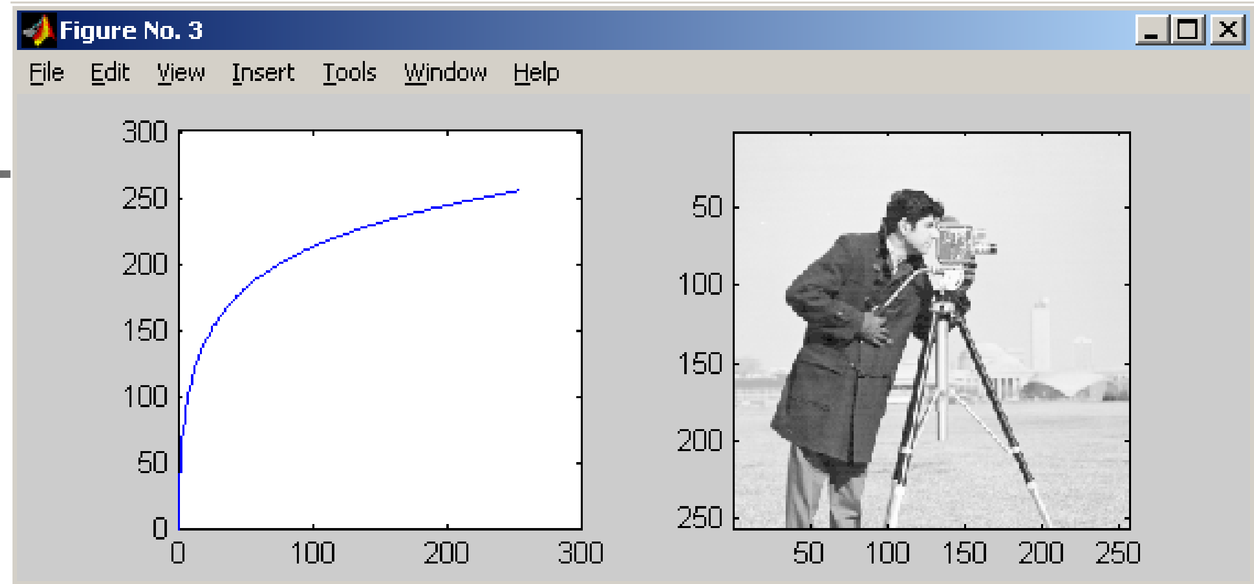


Figure No. 3
File Edit View Insert Tools Window Help
300
250
200
150
100
50
0
0
100
200
300
50
100
150
200
250
50
100
150
200
250

C = 0.8

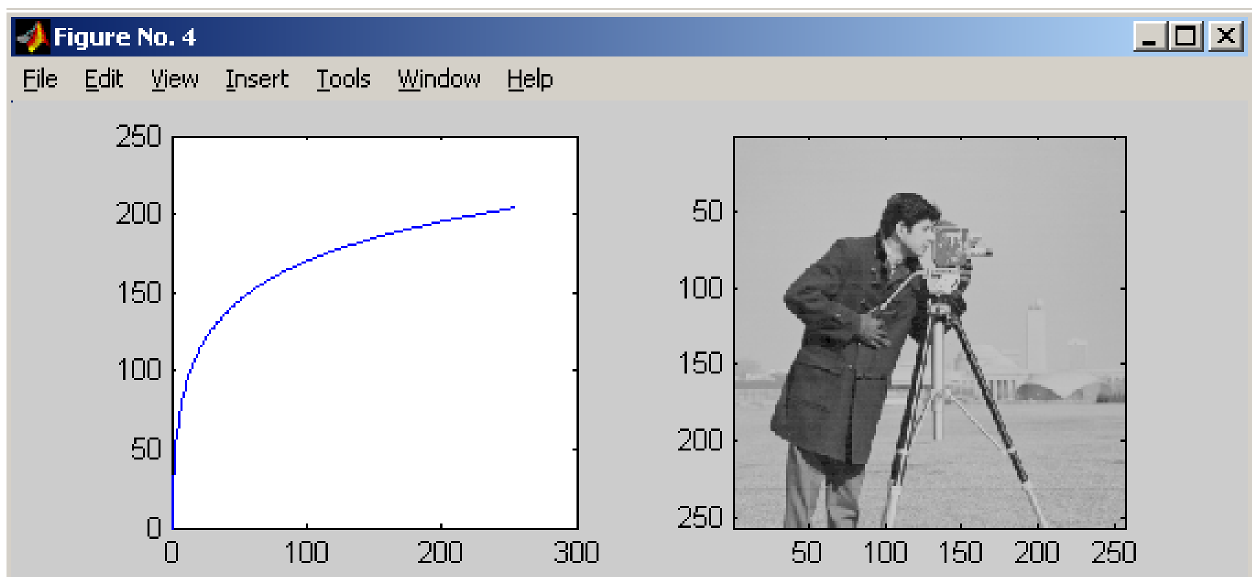
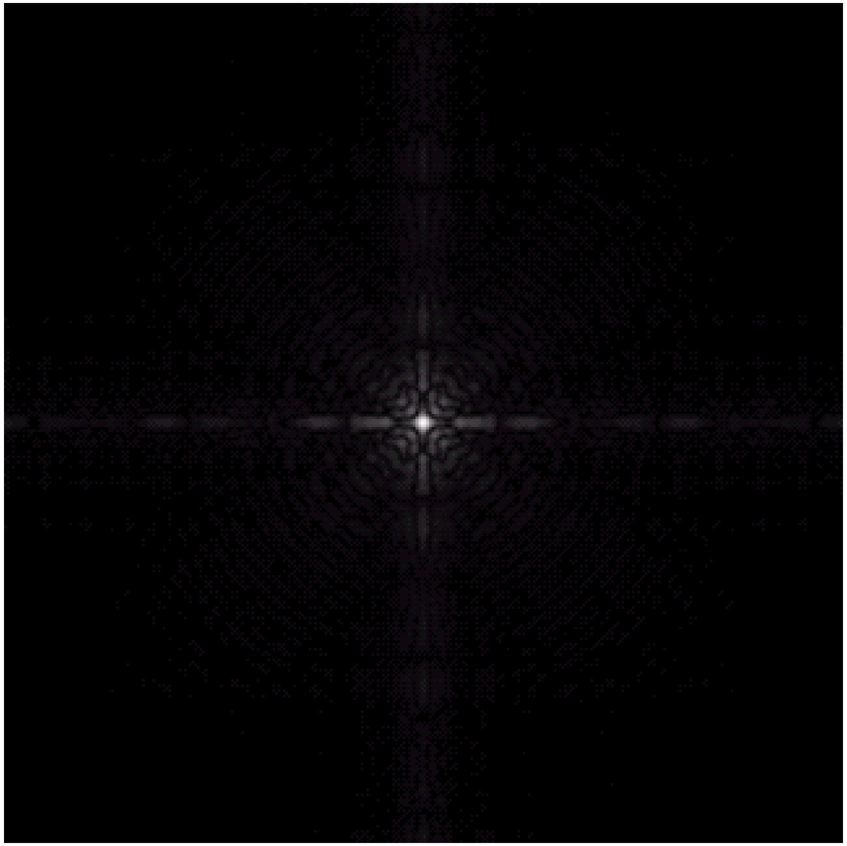


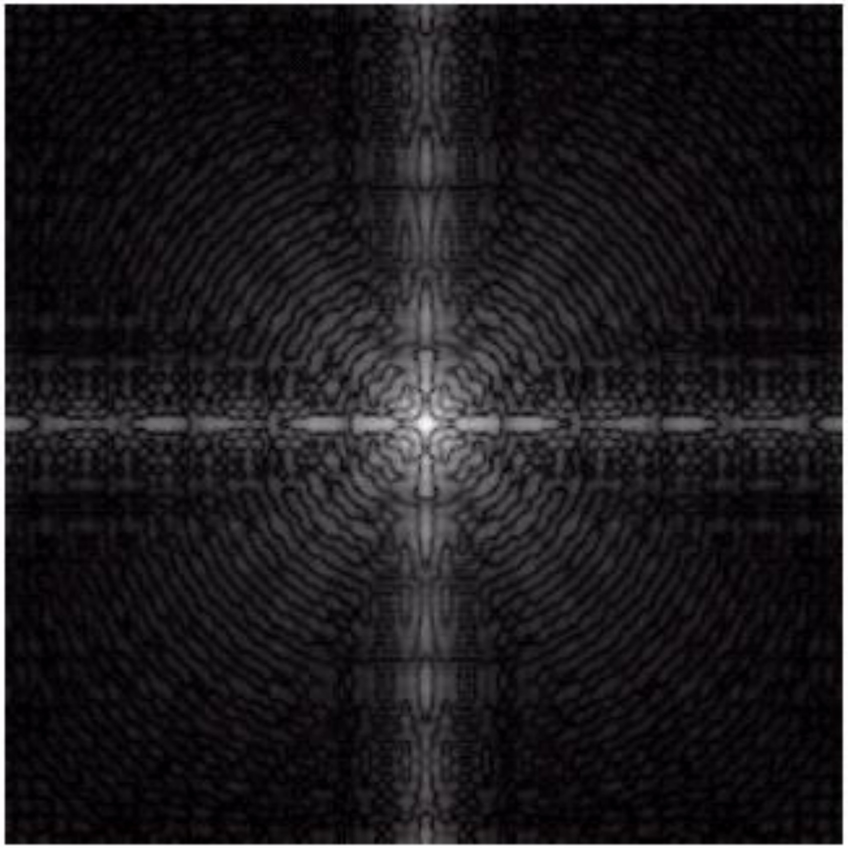
Figure No. 4
File Edit View Insert Tools Window Help
250
200
150
100
50
0
0
100
200
300
50
100
150
200
250
50
100
150
200
250

Example: Log transformations

- log(fft2(I)) : log of Fourier transform



log



2d Fourier transform

Exponential Transformations

- The general form of the log transformation shown in

$$s = c \exp(r)$$

Where c is constant, and it is assumed that $r \geq 0$

$c = 1.0$

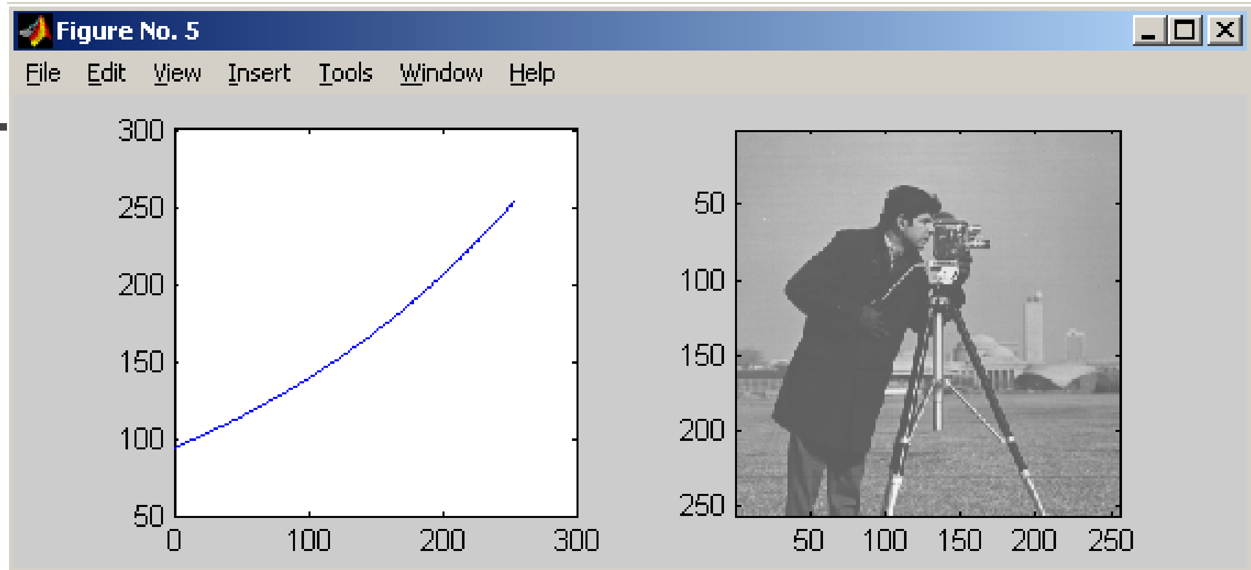


Figure No. 5
File Edit View Insert Tools Window Help
300
250
200
150
100
50
0
100
200
300
50
100
150
200
250
50
100
150
200
250

$c = 0.8$

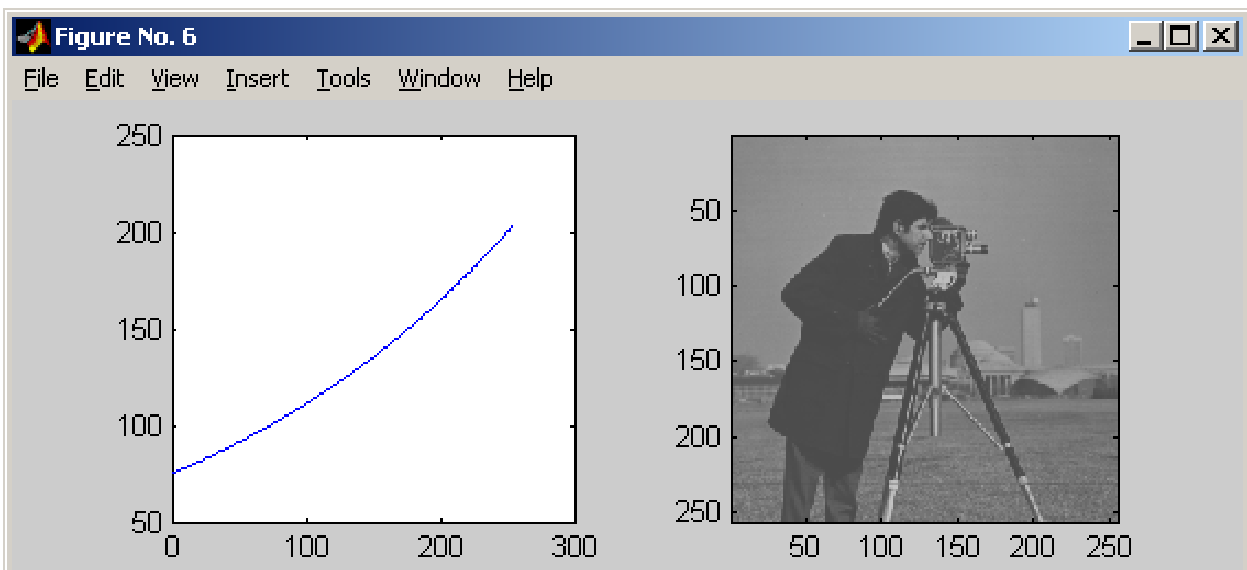


Figure No. 6
File Edit View Insert Tools Window Help
250
200
150
100
50
0
100
200
300
50
100
150
200
250
50
100
150
200
250

Power-Law Transformations

- Power-law transformations have the basic form

$$s = cr^{\gamma}$$

Where c and γ are positive constant.

- Sometime above Equation is written as

$$s = c(r + \varepsilon)^{\gamma}$$

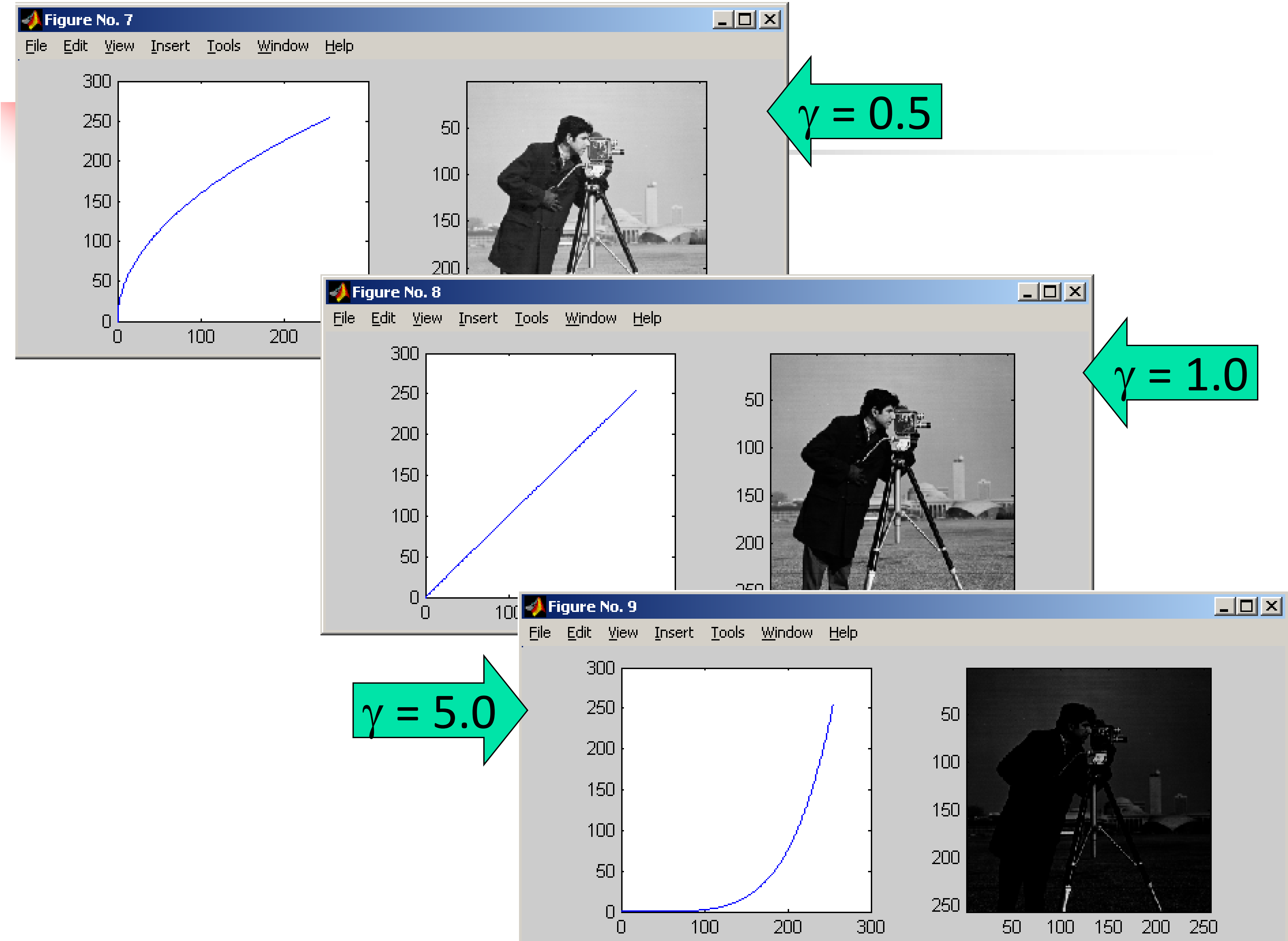


Figure No. 7
File Edit View Insert Tools Window Help
γ = 0.5
Figure No. 8
File Edit View Insert Tools Window Help
γ = 1.0
Figure No. 9
File Edit View Insert Tools Window Help
γ = 5.0

$\gamma > 1$

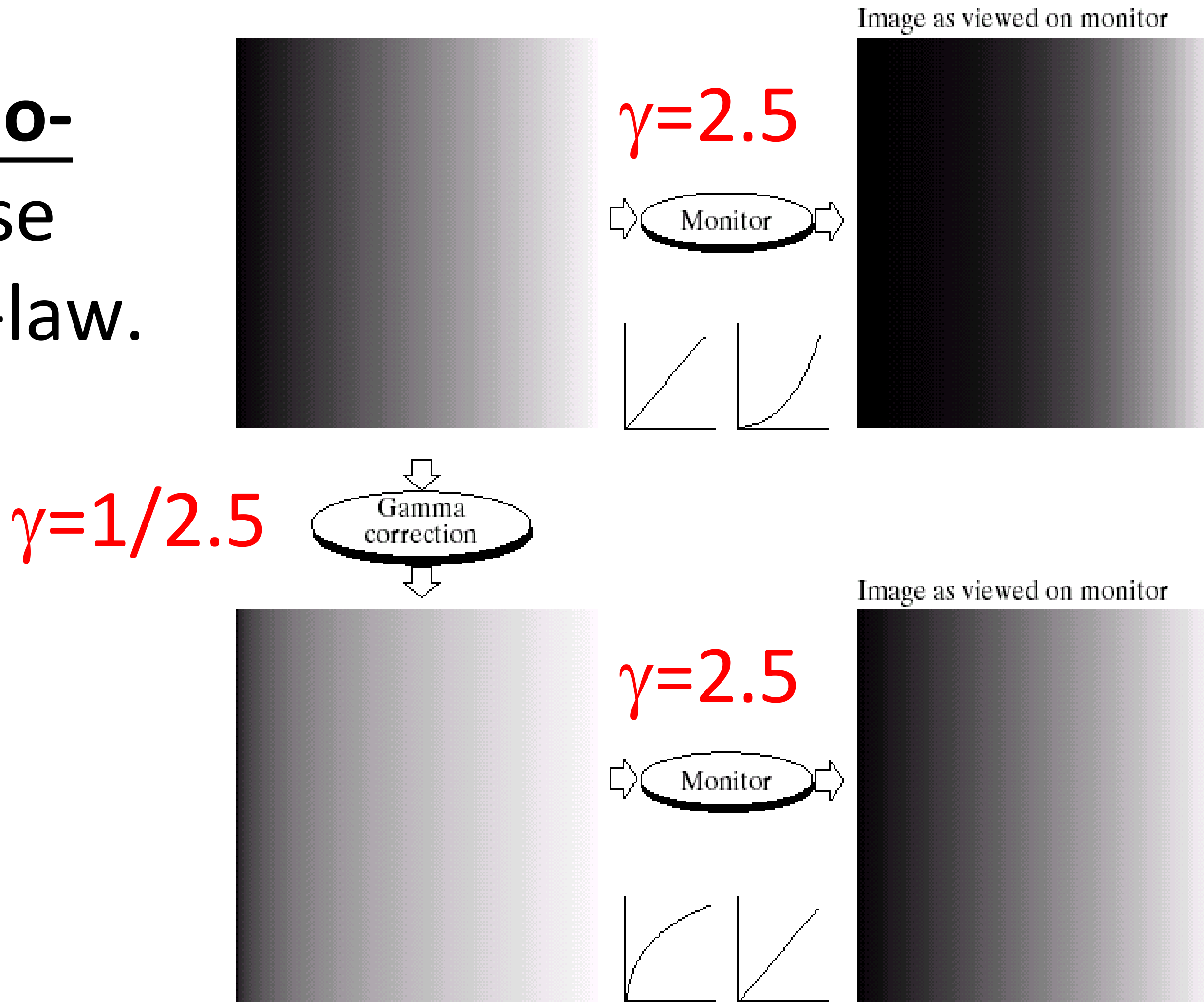
$\gamma < 1$

γ: gamma display, printers, scanners follow power-law

Gamma correction

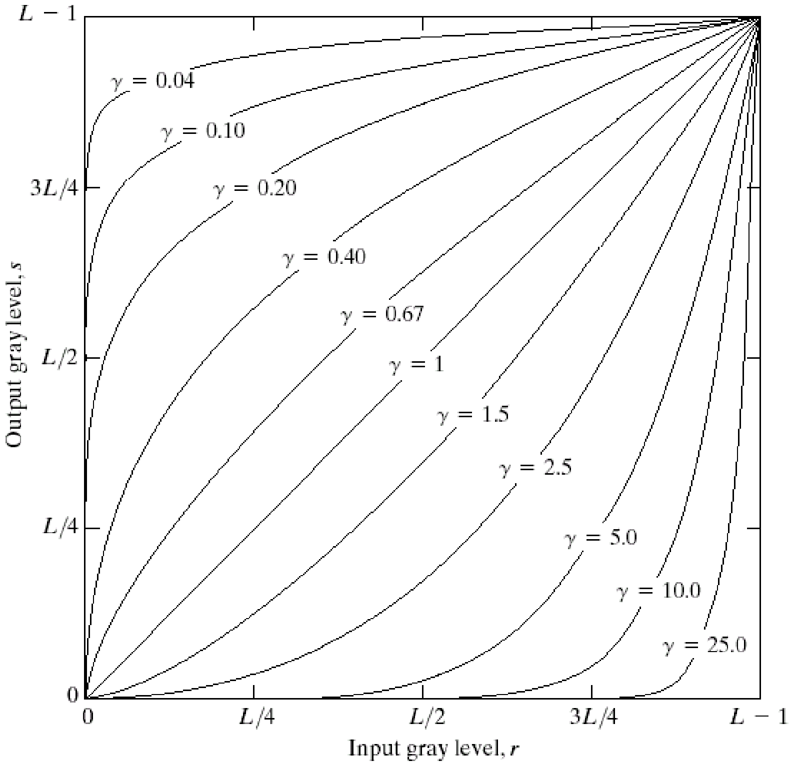
Example: Gamma correction

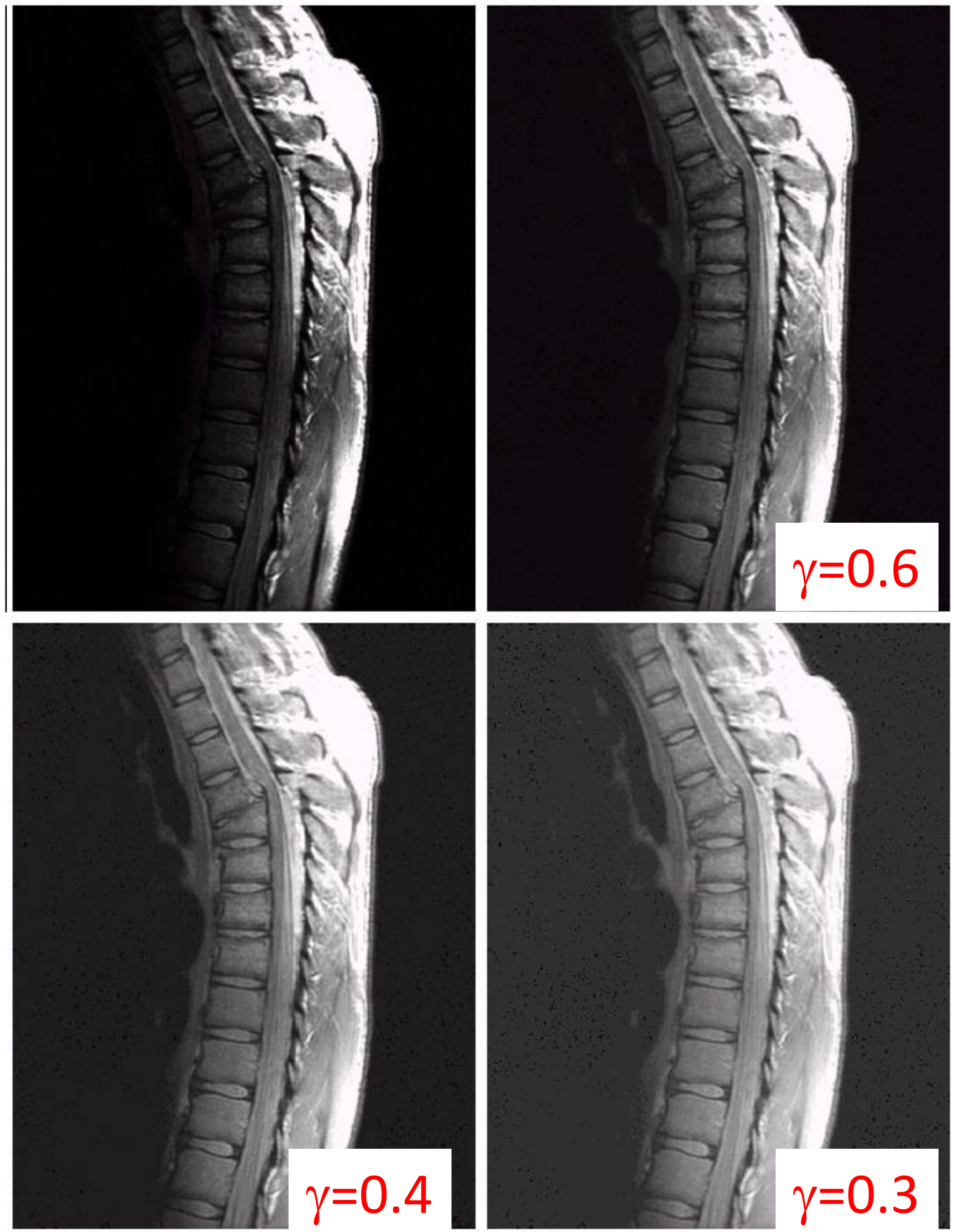
- CRT: **intensity-to-voltage** response follow a power-law. 1.8<γ<2.5



Power-law: $\gamma<1$

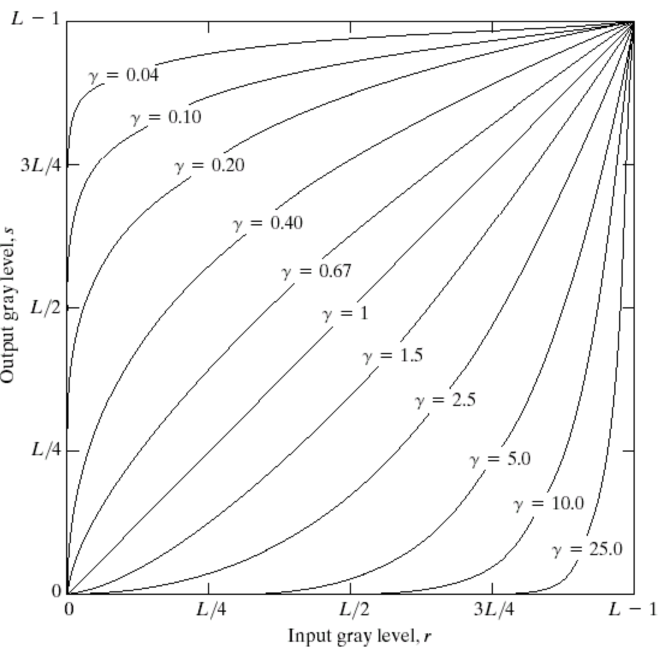
- Expand dark gray levels





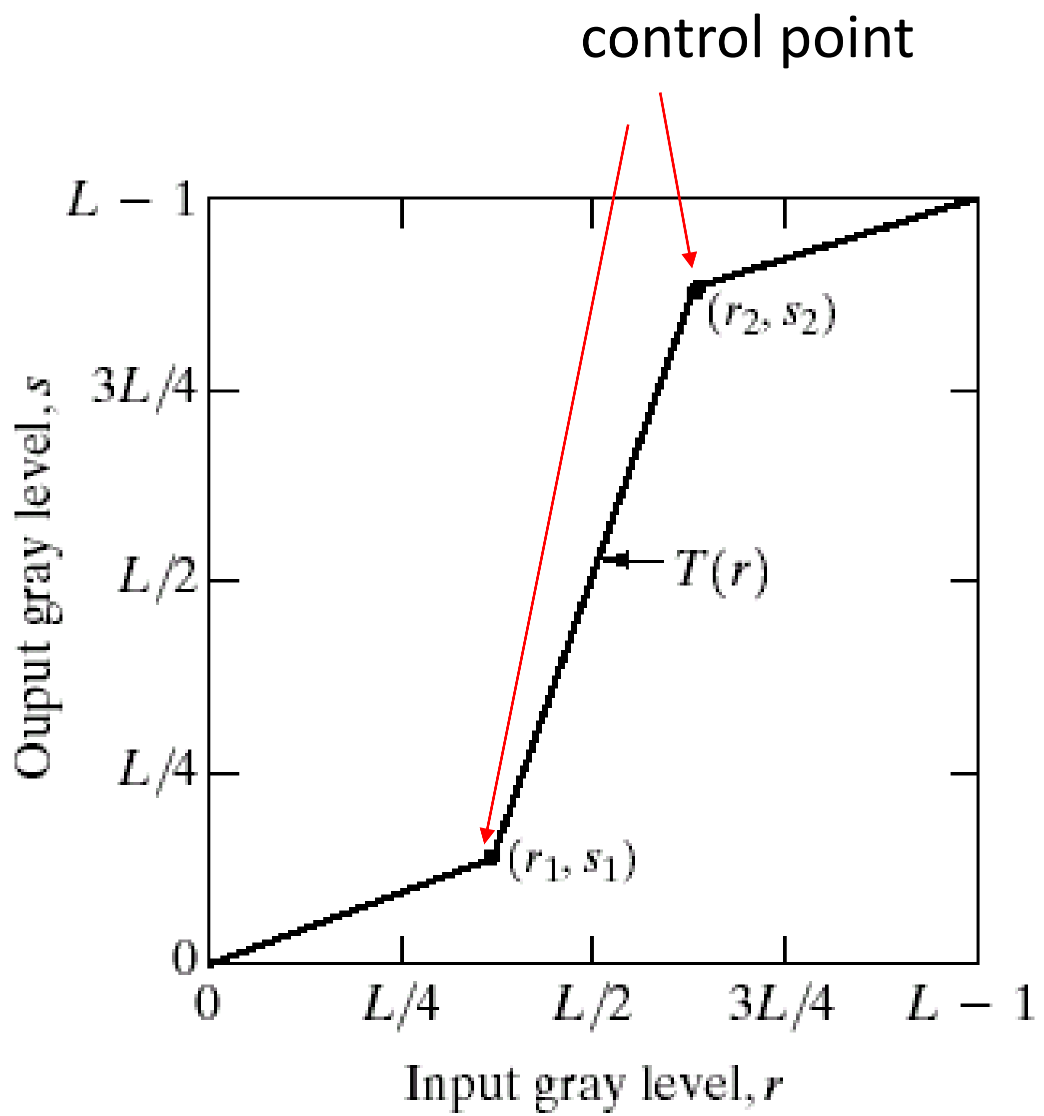
Power-law: γ>1

- Expand light gray levels



Piece-wise linear transformations

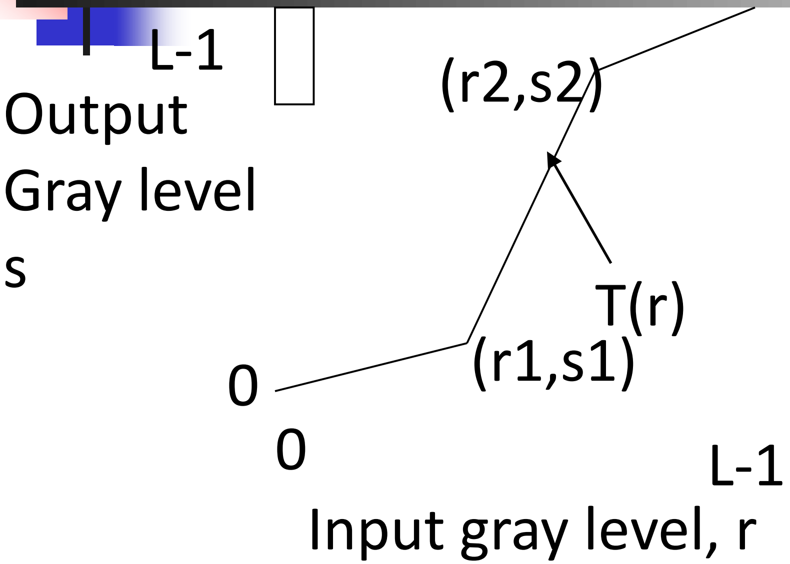
- Advantage: the piecewise function can be arbitrarily complex



Contrast Stretching

- Low contrast image can result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture during image acquisition.
- The idea behind the contrast stretching is to increase the dynamic range of the gray levels in the image being processed.

Contrast Stretching



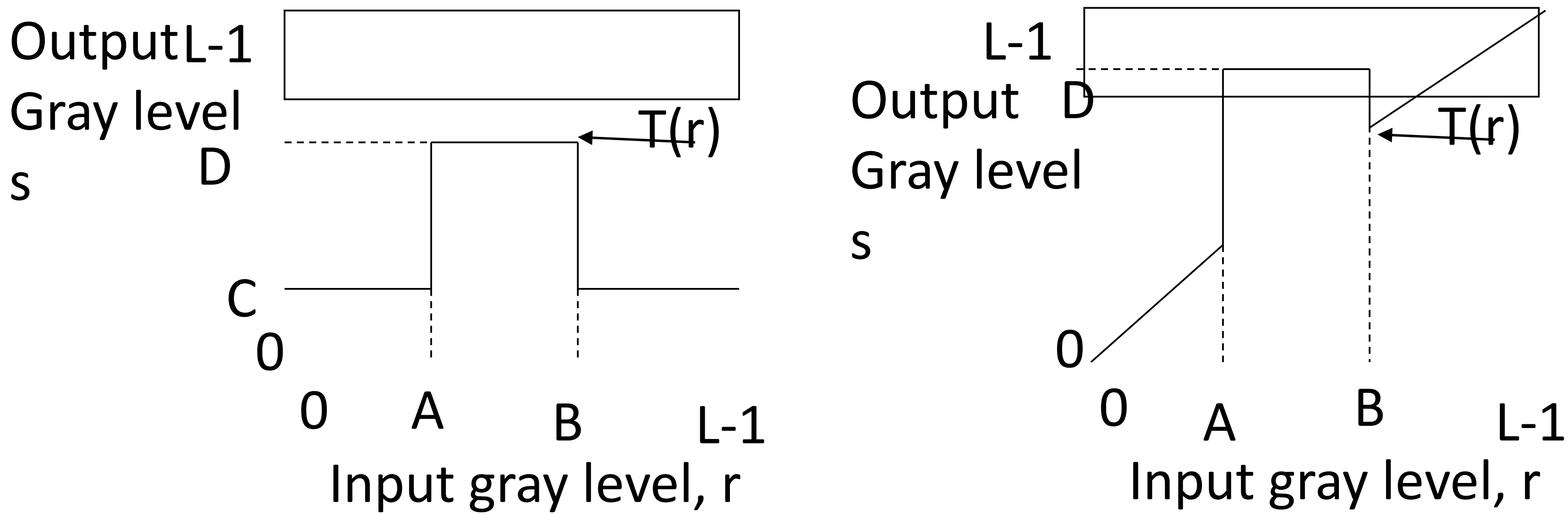
- The location points (r1,s1) and (r2,s2) control the shape of the transformation function

- If r1=s1 and r2=s2, the transformation is a linear function that produces no changes in gray levels

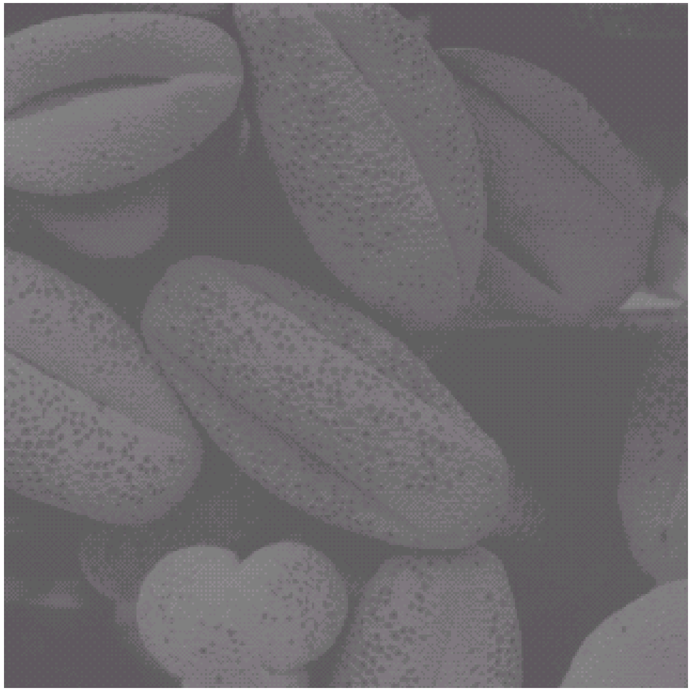
- If r1=r2, s1=0 and s2=L-1 the transformation becomes thresholding function that creates a binary image

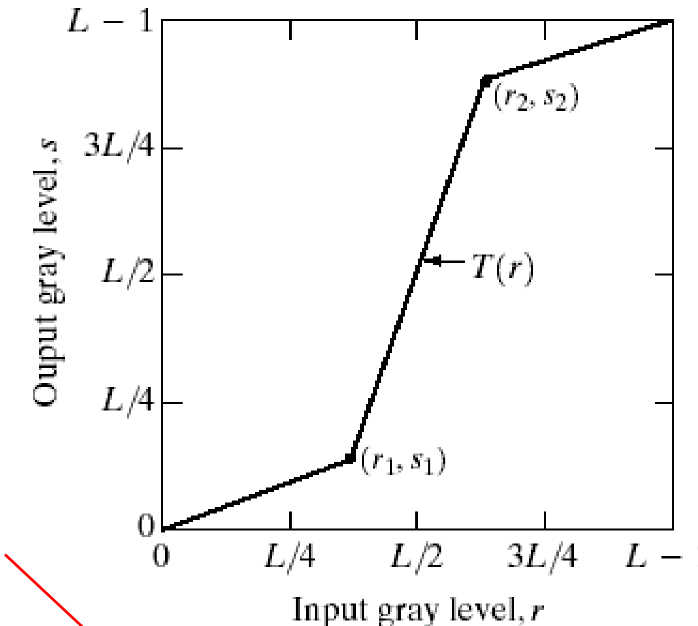
- Intermediate values of (r1,s1) and (r2,s2) produces various degrees of spread in the gray levels of the output image, thus affecting its contrast.

Gray level slicing

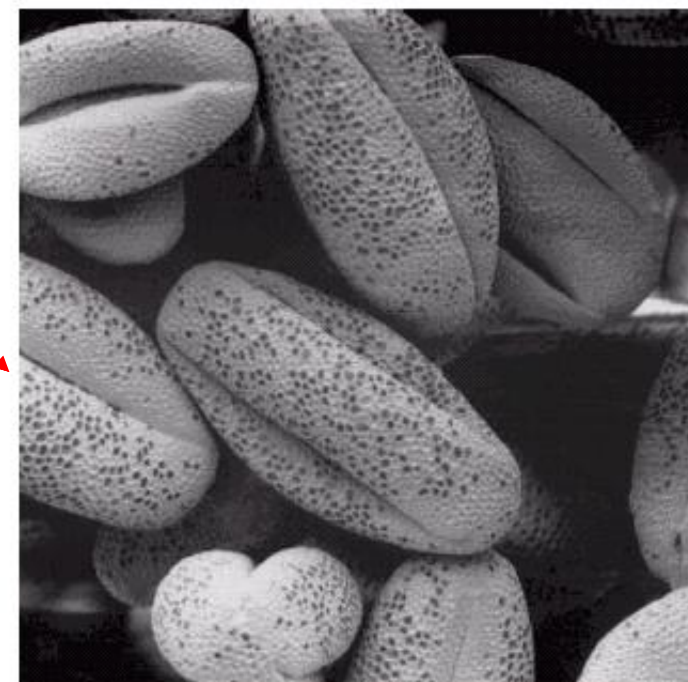


- Highlighting a specific range of gray levels in an image often is desired
- Applications include enhancing features such as masses of water in satellite imagery and enhancing flows in X-rays images.

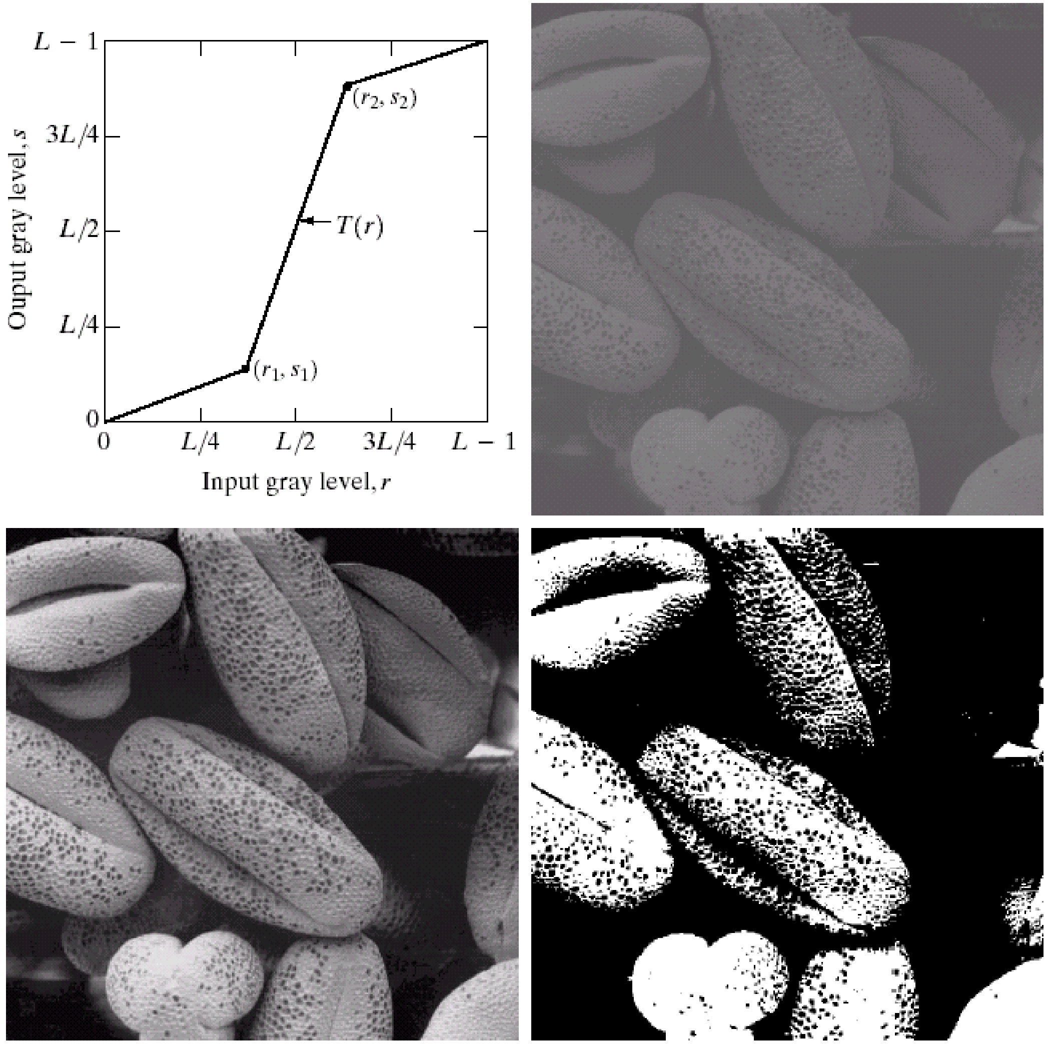




Contrast stretching



Contrast Streching

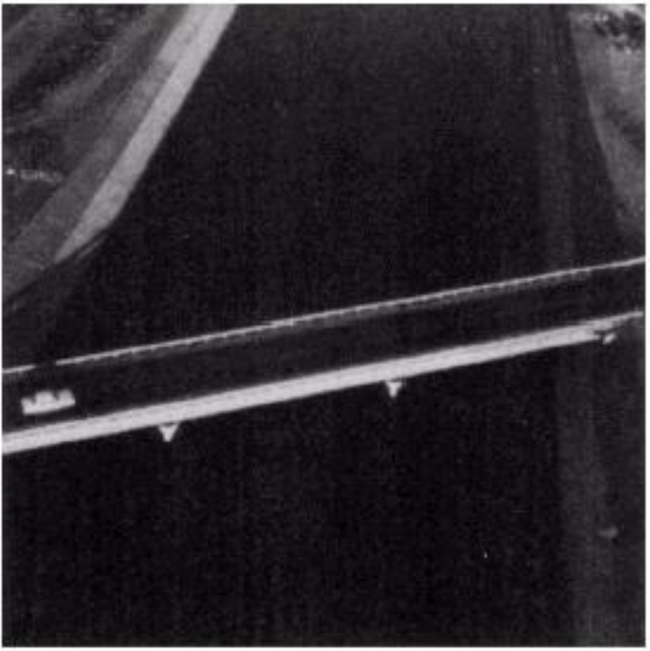


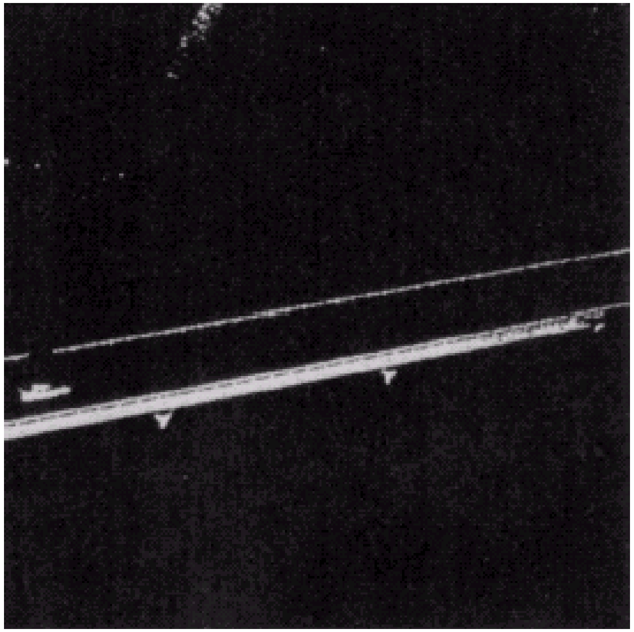
a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Gray-level slicing

- Highlighting a specific range of gray levels





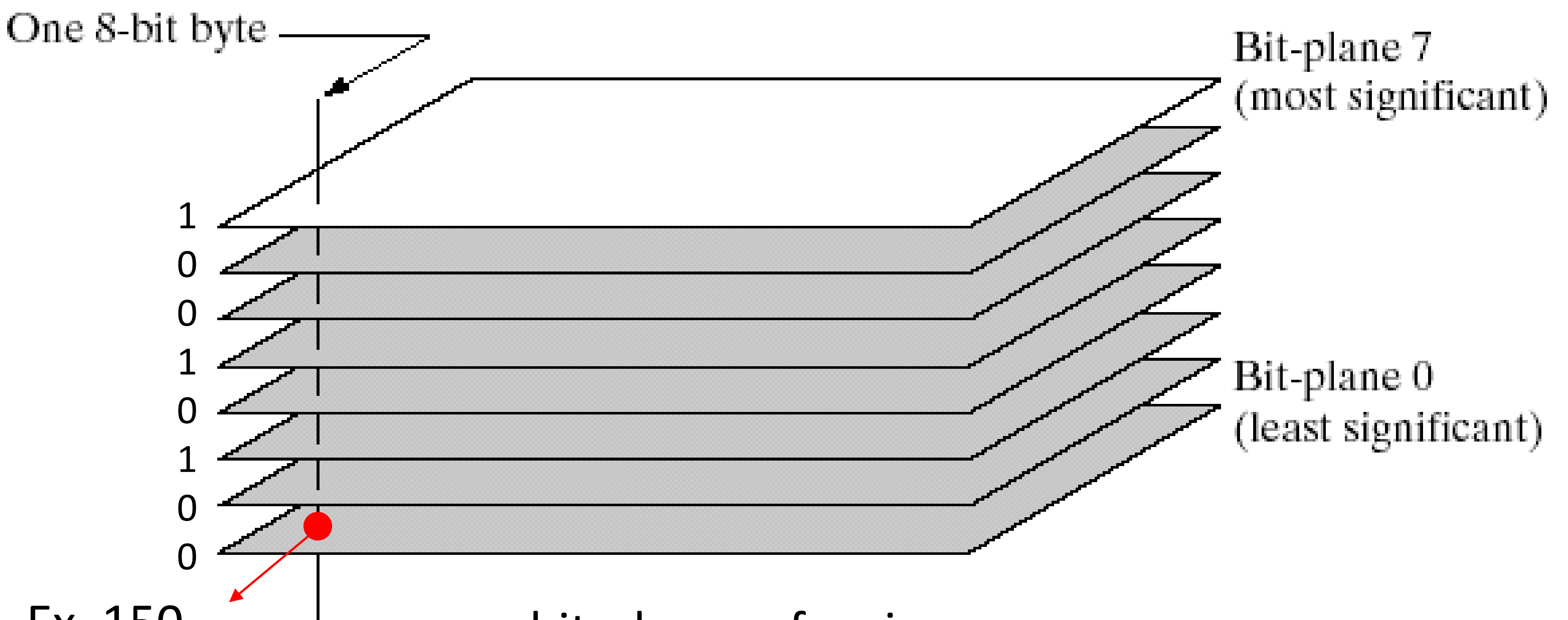
Bit-plane slicing

- Instead of highlighting gray level ranges, highlighting the contribution made to total image appearance by specific bits might be desired.

- Suppose that each pixel in the image is represented by 8-bits. Imagine that the image is composed off eight 1-bit planes, ranging from bit-plane 0 for the least significant bit to bit plane 7 for most significant bit

Bit-plane slicing

*** Highlight specific bits**



bit-planes of an image
(gray level 0~255)

Bit-plane slicing: example

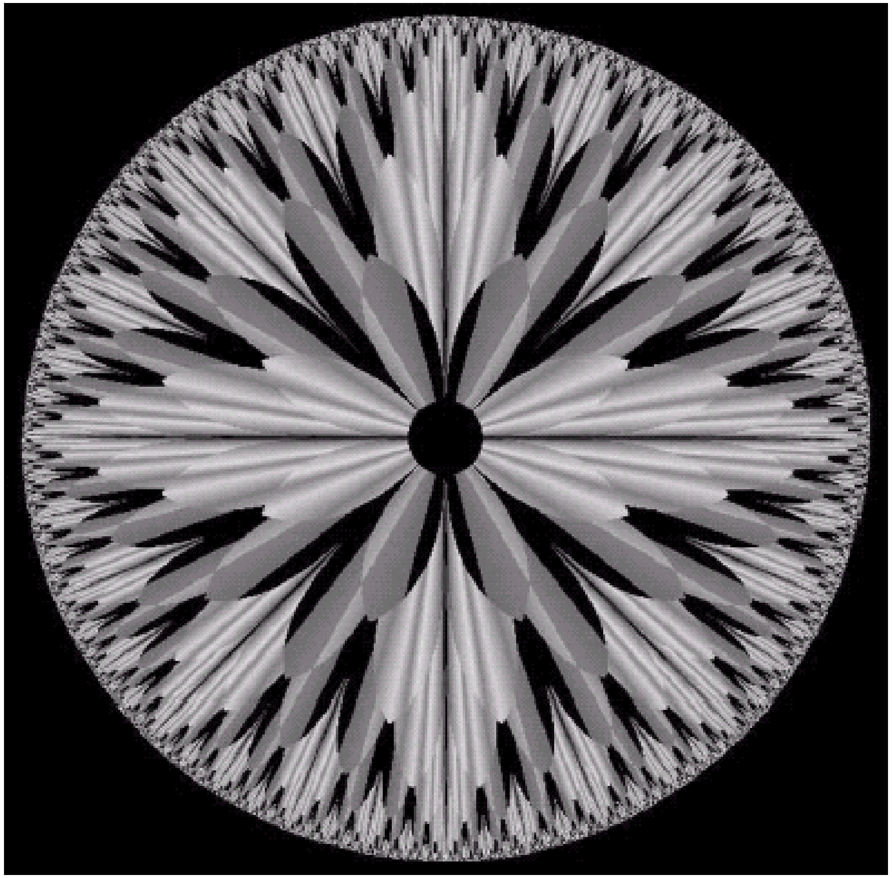
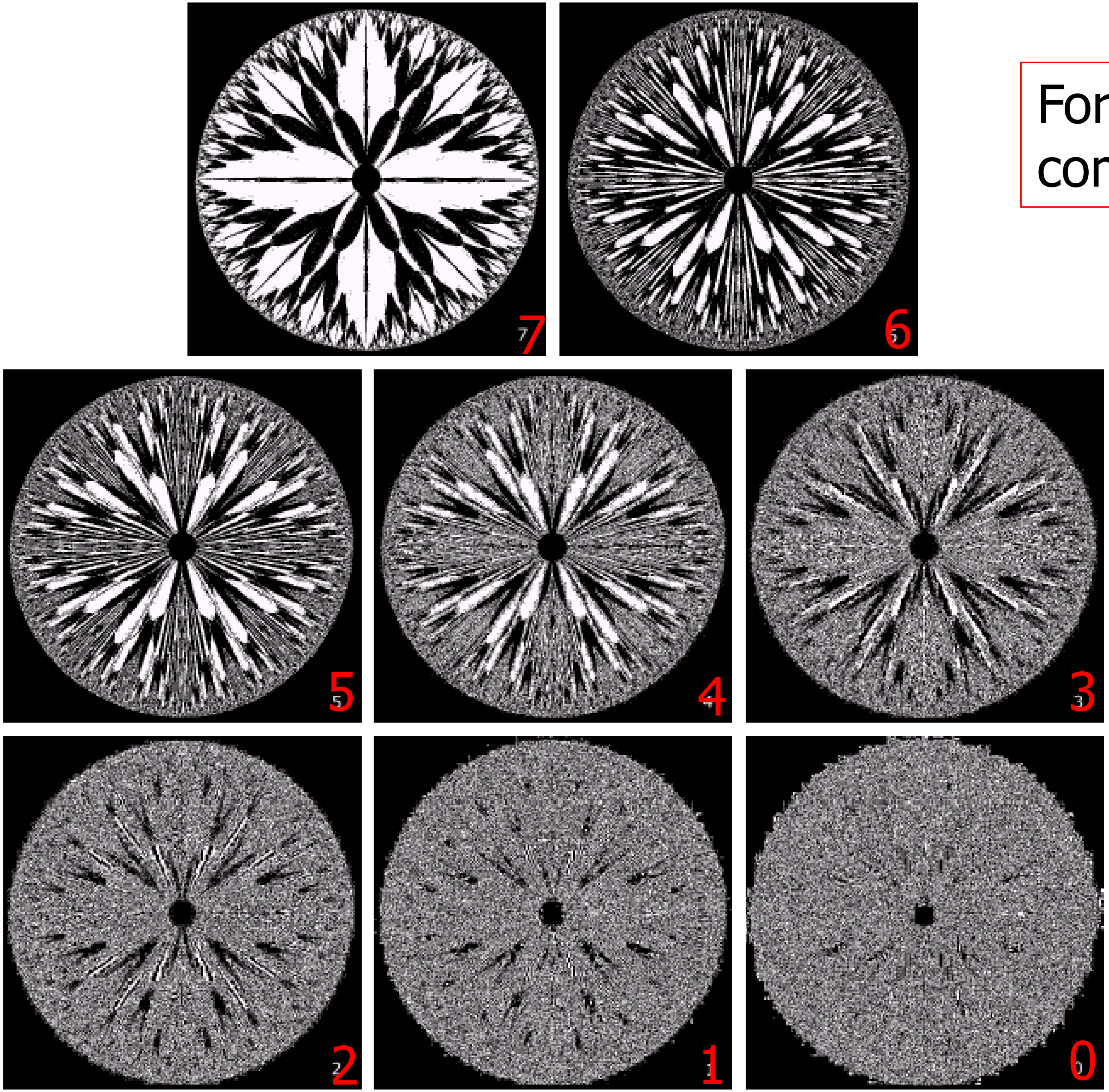


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)



For image compression

FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

Histogram Processing

- 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 is the k^{th} gray level and

n_k is the number of pixels in the image having gray level r_k

- In an image processing context, the histogram of an image normally refers to a **histogram of the pixel intensity values**
- This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image

HISTOGRAM

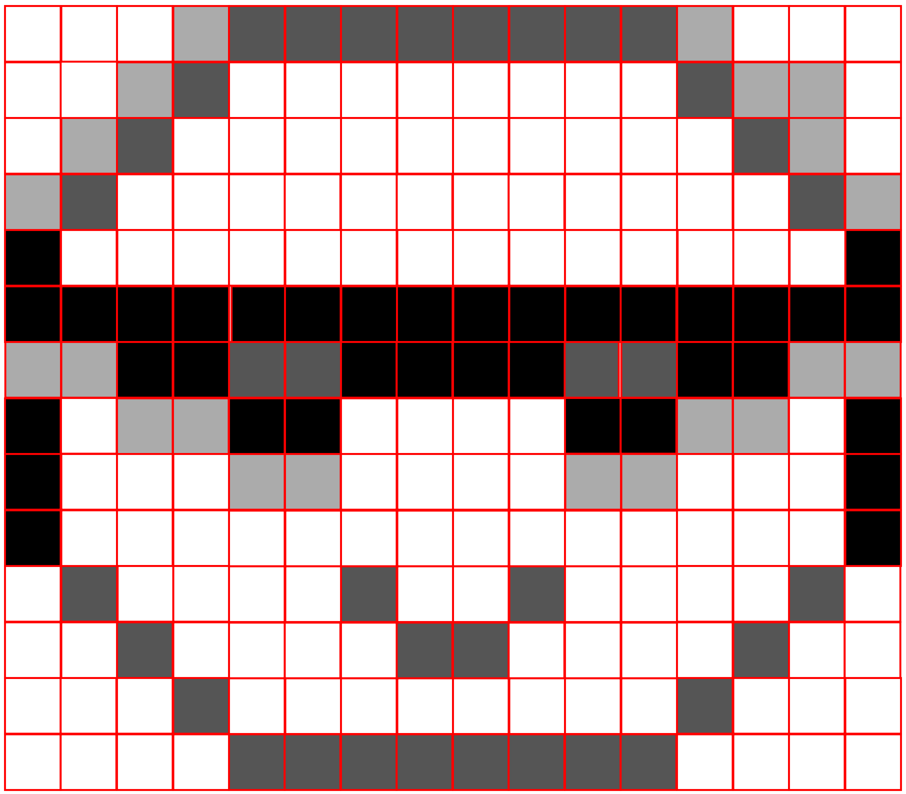
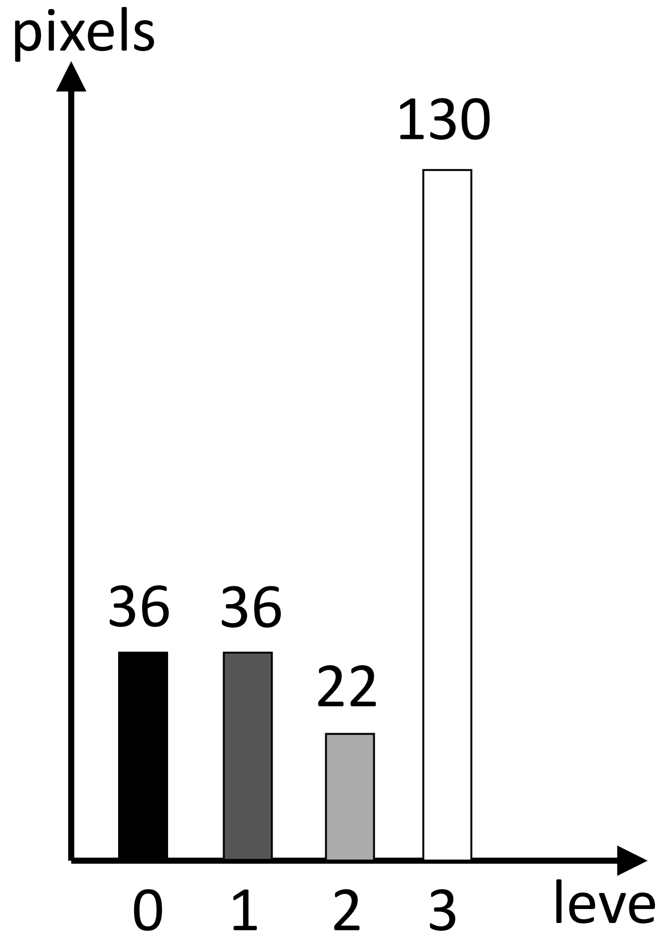


Image 16 x 14 = 224 pixels

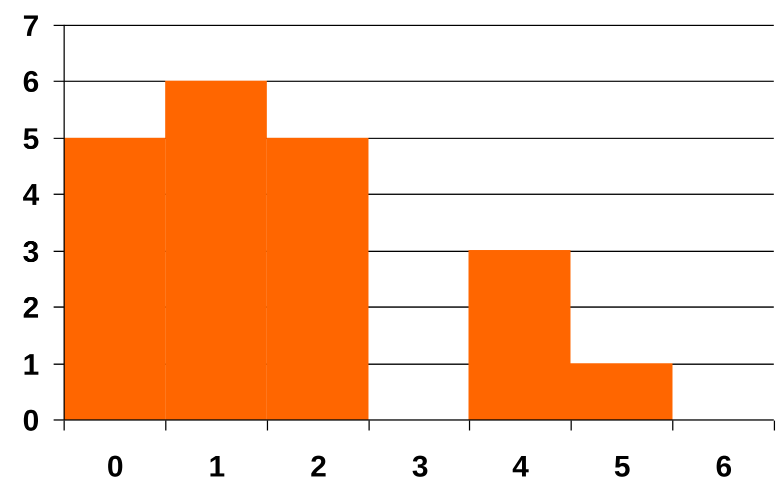


Histogram

The (intensity or brightness) histogram shows how many times a particular grey level (intensity) appears in an image.
For example, 0 - black, 255 – white

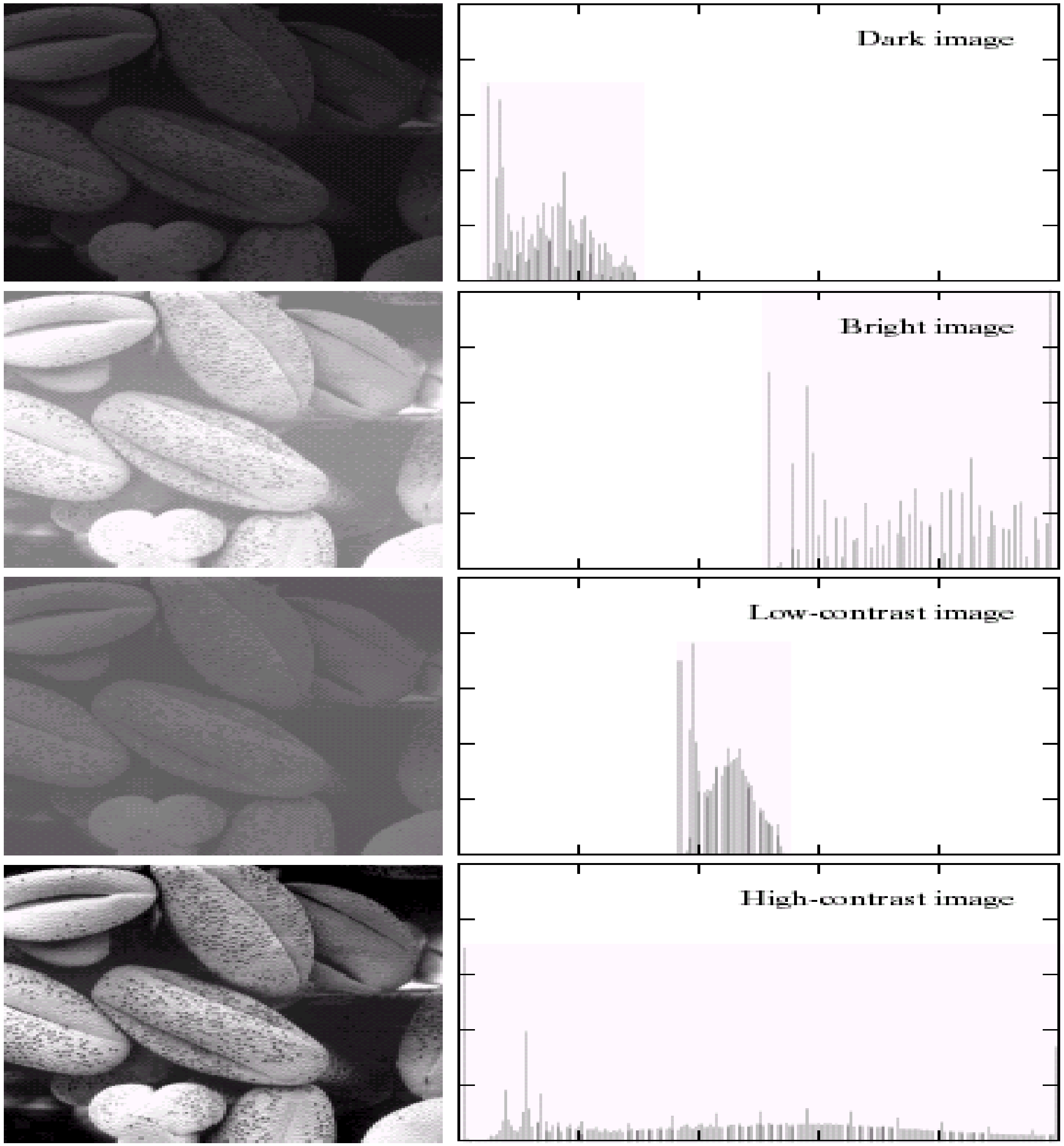
0	1	1	2	4
2	1	0	0	2
5	2	0	0	4
1	1	2	4	1

image



histogram

Image Histograms



x-axis – values of intensities

y-axis – their frequencies

Histogram Processing

Histogram $h(r_k) = n_k$

r_k is the k^{th} intensity value

n_k is the number of pixels in the image with intensity r_k

Normalized histogram $p(r_k) = \frac{n_k}{MN}$

n_k: the number of pixels in the image of

size $\mathrm{M} \times \mathrm{N}$ with intensity r_k

Histogram Processing

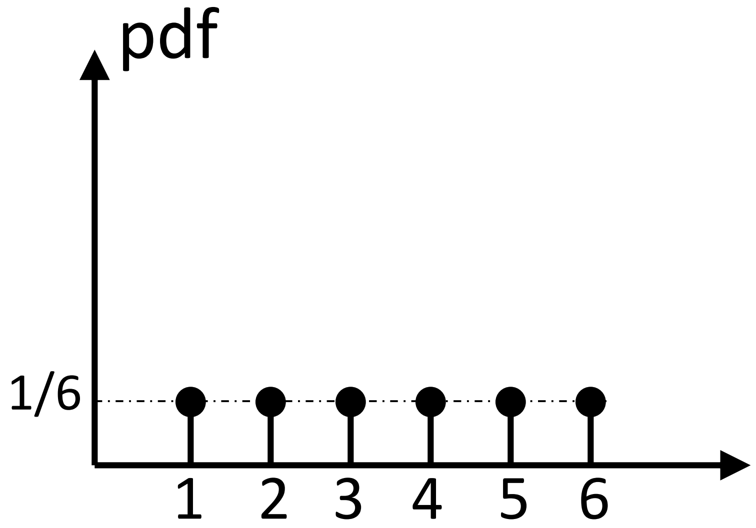
- Histogram Equalization
- Histogram Matching(Specification)
- Local Enhancement

Fundamental of random variable

- PDF (Probability Density Function) is the probability of each element
- CDF (Cumulative Distribution Function) is summation of the probability of the element that value less than or equal this element

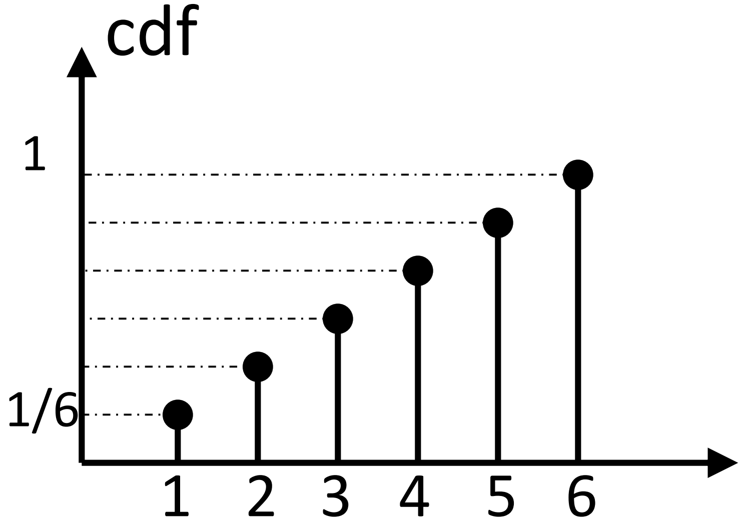
PDF

- The PDF (probability density function) is denoted by p(x)



CDF

- The CDF (cumulative density function) is denoted by P(x)



Relation between PDF and CDF

- PDF can find from this equation

$$p(x) = \frac{d[P(x)]}{dx} = P'(x)$$

- CDF can find from this equation

$$P(x = x_n) = \int_{x=x_0}^{x_n} p(x)dx$$

Histogram Equalization

- We focus attention on transformations of the form

$$s = T(r) \quad where\, 0 \le r \le 1$$

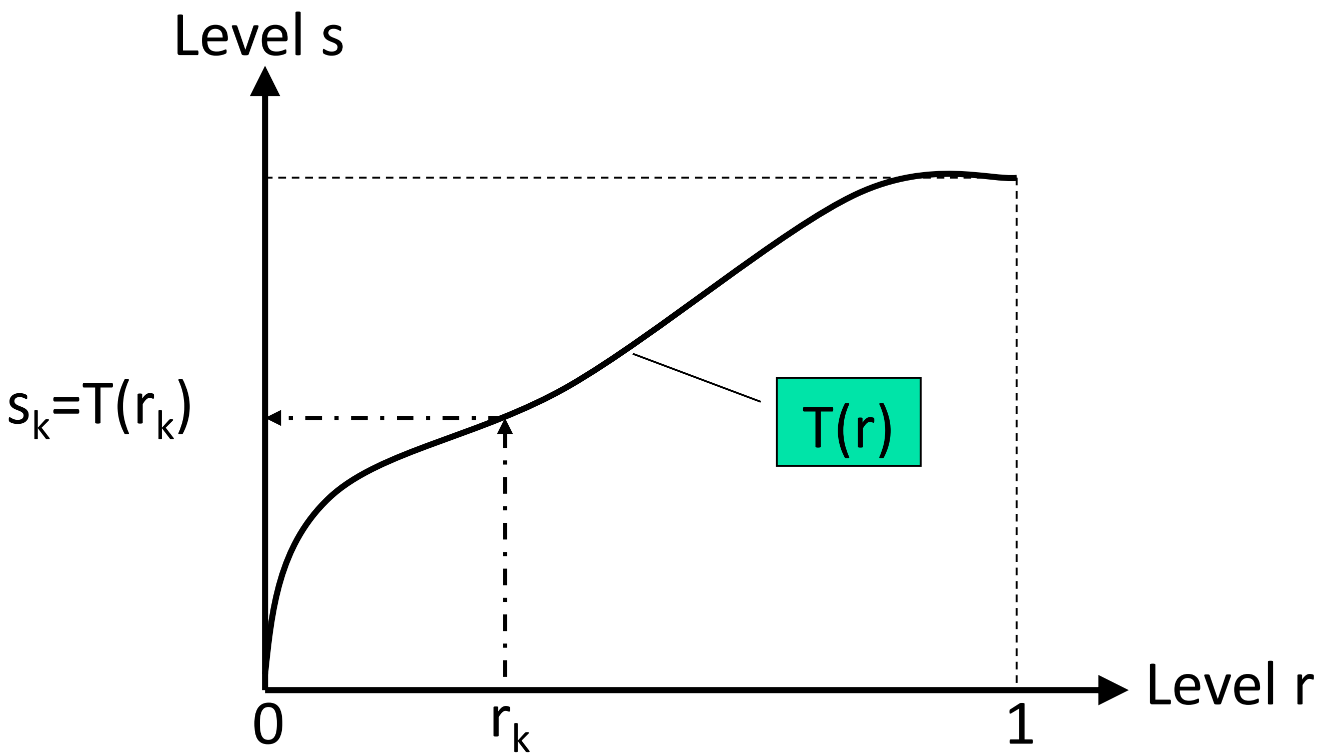
That produce a level s for every pixel value r in the original image.

And we assume that the transformation function T(r) satisfies the following conditions:
(a) T(r) is single-valued and monotonically increasing in the interval $0 \le r \le 1$; and
(b) $0 \le T(r) \le 1$ for $0 \le r \le 1$

Reason of Condition

- The requirement in (a) that T(r) be single valued is needed to guarantee that the **inverse transformation** will exist, and the monotonicity condition preserves the **increasing order from black to white in the output image**
- Condition (b) guarantees that the output gray levels will be in the **same range** as the input levels

Transformation function



Note

- The inverse transformation from s back to r is denoted

$$s = T^{-1}(r) \quad where\, 0 \le r \le 1$$

There are some cases that even if T(r) satisfies conditions (a) and (b), it is possible that the corresponding inverse $T^{-1}(s)$ may fail to be single valued.

Idea of Histogram Equalization

- The gray levels in an image may be viewed as random variables in the interval [0,1]

Let $p_r(r)$ denote the pdf of random variable r and $p_s(s)$ denote the pdf of random variable s; if $p_r(r)$ and $T(r)$ are known and $T^{-1}(s)$ satisfies condition (a), the formula should be

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

$$s = T(r) = \int_0^r p_r(w)dw$$

Idea of Histogram Equalization

- By Leibniz's rule that the derivative of a definite integral with respect to its upper limit is simply the integrand evaluated at that limit

$$
\begin{aligned}
\frac{ds}{dr} &= \frac{dT(r)}{dr} \\
&= \frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right] \\
&= p_r(r)
\end{aligned}
$$

Idea of Histogram Equalization

- Substituting into the first equation

$$
\begin{aligned}
p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| \\
&= p_r(r)\left|\frac{1}{p_r(r)}\right| \\
&= 1, \quad 0 \le s \le 1
\end{aligned}
$$

Idea of Histogram Equalization

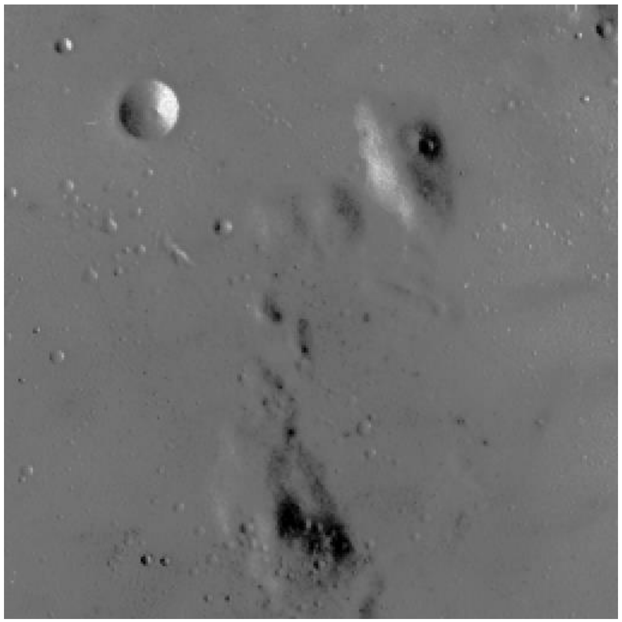
- The probability of occurrence of gray level r_k in an image is approximated by

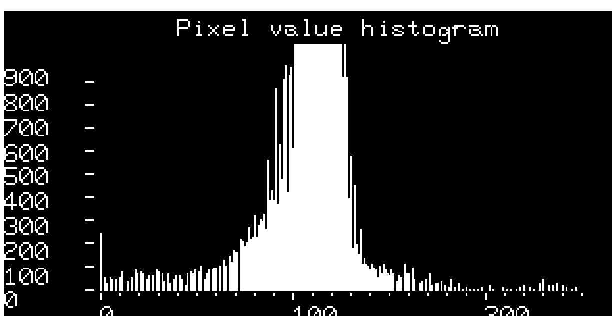
$$p_k(r_k) = \frac{n_k}{n} \quad k = 0,1,2,\ldots, L-1$$

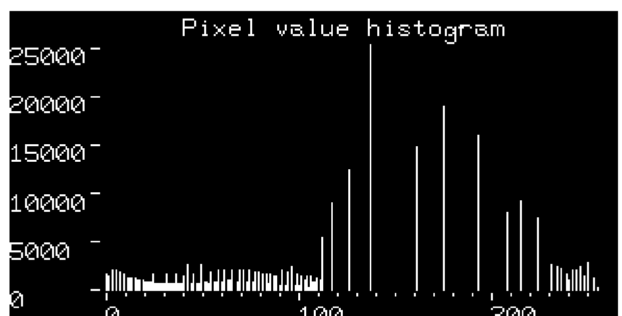
$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n} \quad k = 0,1,2,\ldots, L-1$$

Histogram Equalization







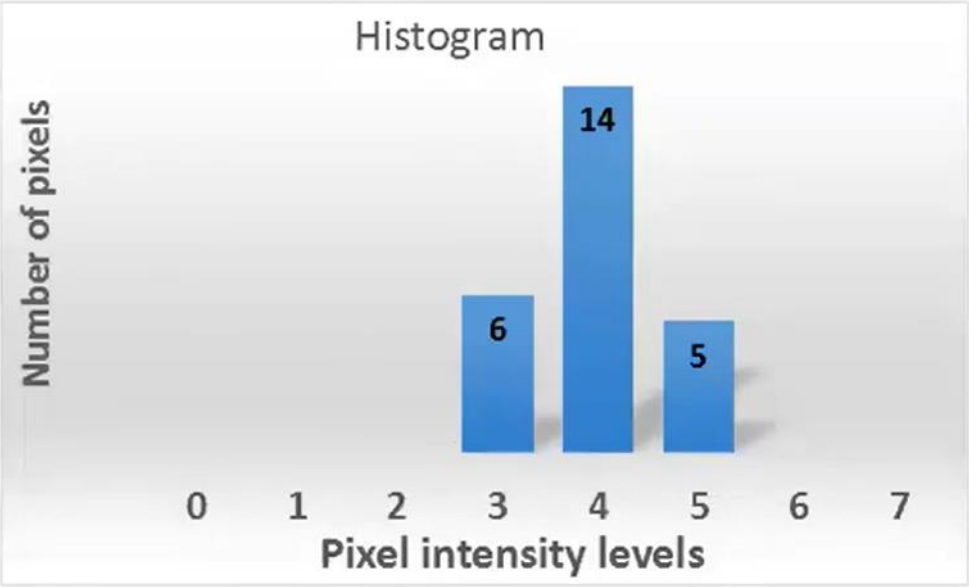
Example: Perform Histogram equalization of the image

4	4	4	4	4
3	4	5	4	3
3	5	5	5	3
3	4	5	4	3
4	4	4	4	4

Solution

- Maximum gray value = 5
- No. of bits required to represent each intensity = 3 bits
- No. of possible gray levels = 8 (varies from 0 to 7)

Gray level	0	1	2	3	4	5	6	7
No. of pixel nk	0	0	0	6	14	5	0	0



Solution

Gray level	No. of pixel = nk	PDF = $\frac{\text{No.of pixel}}{sum}$ Pk = $\frac{nk}{N}$	(Running sum) CDF Sk	Running sum * maximum gray level 7 x sk	Histogran equalizat level
0	0	0/25 = 0	0	7 X 0 = 0	0
1	0	0/25 = 0	0	7 X 0 = 0	0
2	0	0/25 = 0	0	7 X 0 = 0	0
3	6	6/25 = 0.24	0.24	7 X 0.24 = 1.68	2
4	14	14/25 = 0.56	0.8	7 X 0.8 = 5.6	6
5	5	5/25 = 0.2	1	7 X 1 = 7	7
6	0	0/25 = 0	1	7 X 1 = 7	7
7	0	0/25 = 0	1	7 X 1 = 7	7
	N= 25				

Solution

Input Image

4	4	4	4	4
3	4	5	4	3
3	5	5	5	3
3	4	5	4	3
4	4	4	4	4

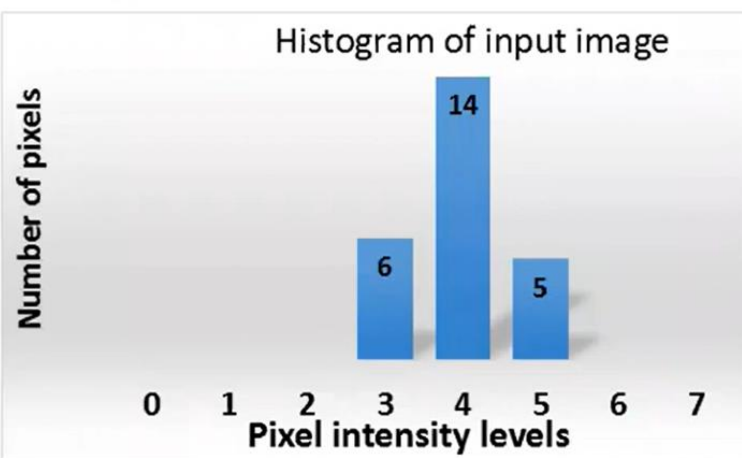
After equalization

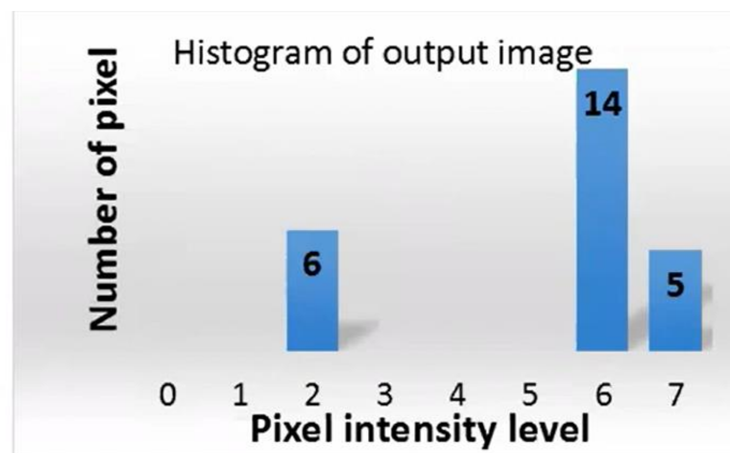
Output Image

6	6	6	6	6
2	6	7	6	2
2	7	7	7	2
2	6	7	6	2
6	6	6	6	6

Gray level	0	1	2	3	4	5	6	7
No. of pixel nk	0	0	0	6	14	5	0	0

Gray level	0	1	2	3	4	5	6	7
No. of pixel nk	0	0	6	0	0	0	14	5





Question

Is histogram equalization always good?

No

Histogram Specification (Histogram Matching)

- Histogram equalization yields an image whose pixels are (in theory) uniformly distributed among all gray levels

- Sometimes, this may not be desirable

- Instead, we may want a transformation that yields an output image with a pre-specified histogram

- This technique is called **histogram specificatio**n

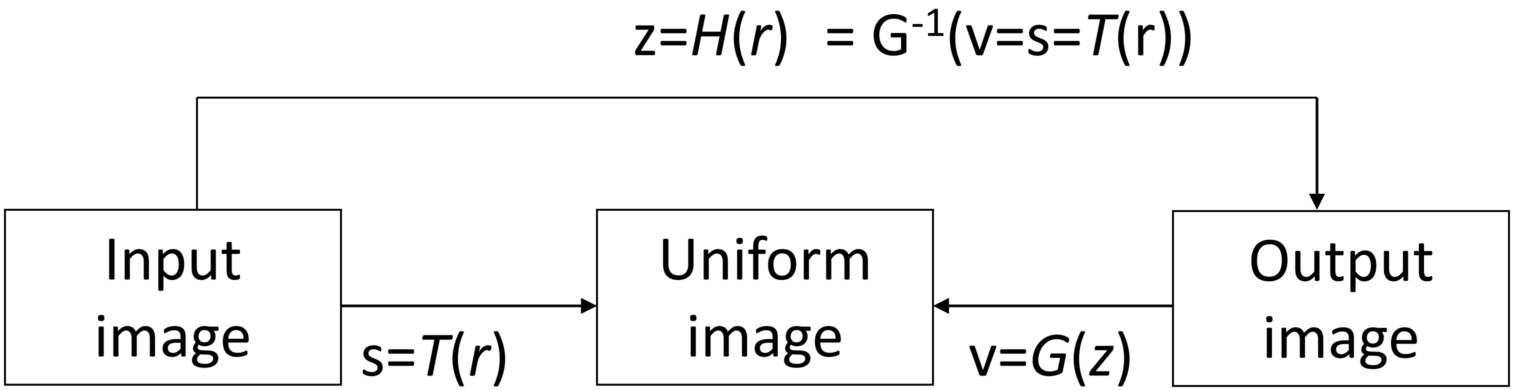
- Given Information

(1) Input image from which we can compute its histogram .
(2) Desired histogram.

- Goal
- Derive a point operation, $H(r)$, that maps the input image into an output image that has the user-specified histogram

- Again, we will assume, for the moment, continuous-gray values

Approach of derivation



· Suppose, the input image has probability density in *p(r)* . We want to find a transformation *z = H (r)* , such that the probability density of the new image obtained by this transformation is $p_{out}(z)$, which is not necessarily uniform.

· First apply the transformation

$$s = T(r) = \int_0^r p_{in}(w)dw, \qquad 0 \le r \le 1$$

This gives an image with a uniform probability density.

· If the desired output image were available, then the following transformation would generate an image with uniform density:

$$V = G(z) = \int_0^z p_{out}(w)dw, \qquad 0 \le z \le 1$$

· From the gray values n we can obtain the gray values *z* by using the inverse transformation, $z = G^{-1}(v)$

· If instead of using the gray values n obtained from (eq-2), we use the gray values *s* obtained from (eq-1) above (both are uniformly distributed !), then the point transformation

$$Z=H(r)= G^{-1}[\, v=s = T(r)]$$

will generate an image with the specified density out *p(z)* , from an input image with density in *p(r)*

· For discrete gray levels, we have

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

$$v_k = G(z_k) = \sum_{j=0}^{k} p_{out}(z_j) = s_k \qquad 0 \le k \le L-1$$

· If the transformation z_k ® $G(z_k)$ is one-to-one, the inverse transformation s_k ® $G^{-1}(s_k)$, can be easily determined, since we are dealing with a small set of discrete gray values.

· In practice, this is not usually the case (i.e.,) z_k ® $G(z_k)$ is not one-to-one) and we assign gray values to match the given histogram, as closely as possible.

Algorithm for histogram specification:

(1) Equalize input image to get an image with uniform gray values, using the discrete equation:

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

(2) Based on desired histogram to get an image with uniform gray values, using the discrete equation:

$$v_k = G(z_k) = \sum_{j=0}^{k} p_{out}(z_j) = s_k \qquad 0 \le k \le L-1$$

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

Modify Histogram (a) as given by (b)

(a)

Gray level.	0	1	2	3	4	5	6	7
No. of pixels	790	1023	850	656	329	245	122	81

(b)

Gray level.	0	1	2	3	4	5	6	7
No. of pixels	0	0	0	614	819	1230	819	614

Equalize Histogram (a)

Gray level	nk	PDF	CDF	Sk x 7	Round off	New nk.
0	790	0.19	0.19	1.33	1	790
1	1023	0.25	0.44	3.08	3	1023
2	850	0.21	0.65	4.55	5	850
3	656	0.16	0.81	5.67	6 656+329	985
4	329	0.08	0.89	6.23	6	
5	245	0.06	0.95	6.65	7	
6	122	0.03	0.98	6.86	7 245 +122+81	448
7	81	0.02	1	7	7	
	N=4096					

Equalize Histogram (b)

Gray level	nk	PDF	CDF	Sk x 7	Round off
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	614	0.149	0.149	1.05	1
4	819	0.20	0.35	2.50	3
5	1230	0.30	0.65	4.55	5
6	819	0.20	0.85	5.97	6
7	614	0.15	1	7	7
	N=4096				

Mapping

First and last columns of histogram (b)

Gray level	Round off
0	0
1	0
2	0
3	1
4	3
5	5
6	6
7	7

Last two columns of histogram (a)

Round off	New nk.
1	790
3	1023
5	850
6	
6	985
7	
7	448
7	

First and last columns of histogram (b)

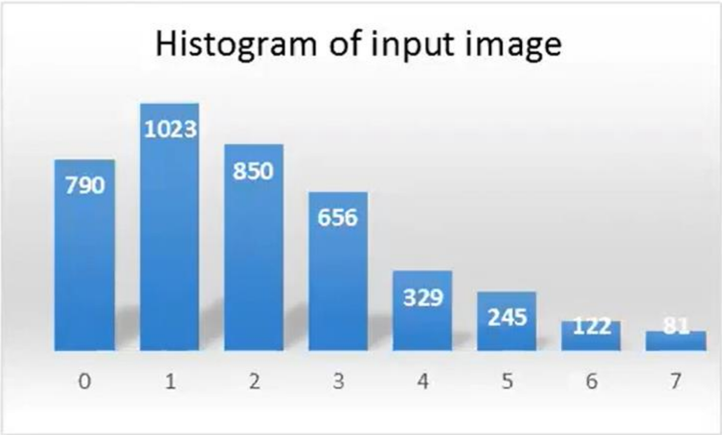
Gray level	Round off
0	0
1	0
2	0
3	1
4	3
5	5
6	6
7	7

Last two columns of histogram (a)

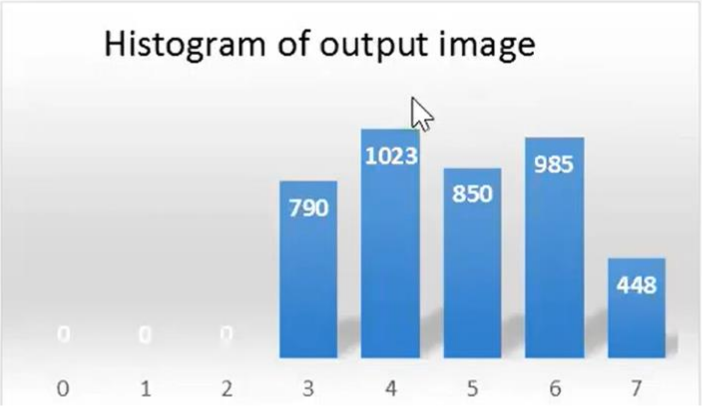
Round off	New nk.
1	790
3	1023
5	850
6	
6	985
7	
7	448
7	

Gray level.	0	1	2	3	4	5	6	7
No. of pixels	0	0	0	790	1023	850	985	448

Mapping



Plot histogram for modified image.



Example: Histogram Matching

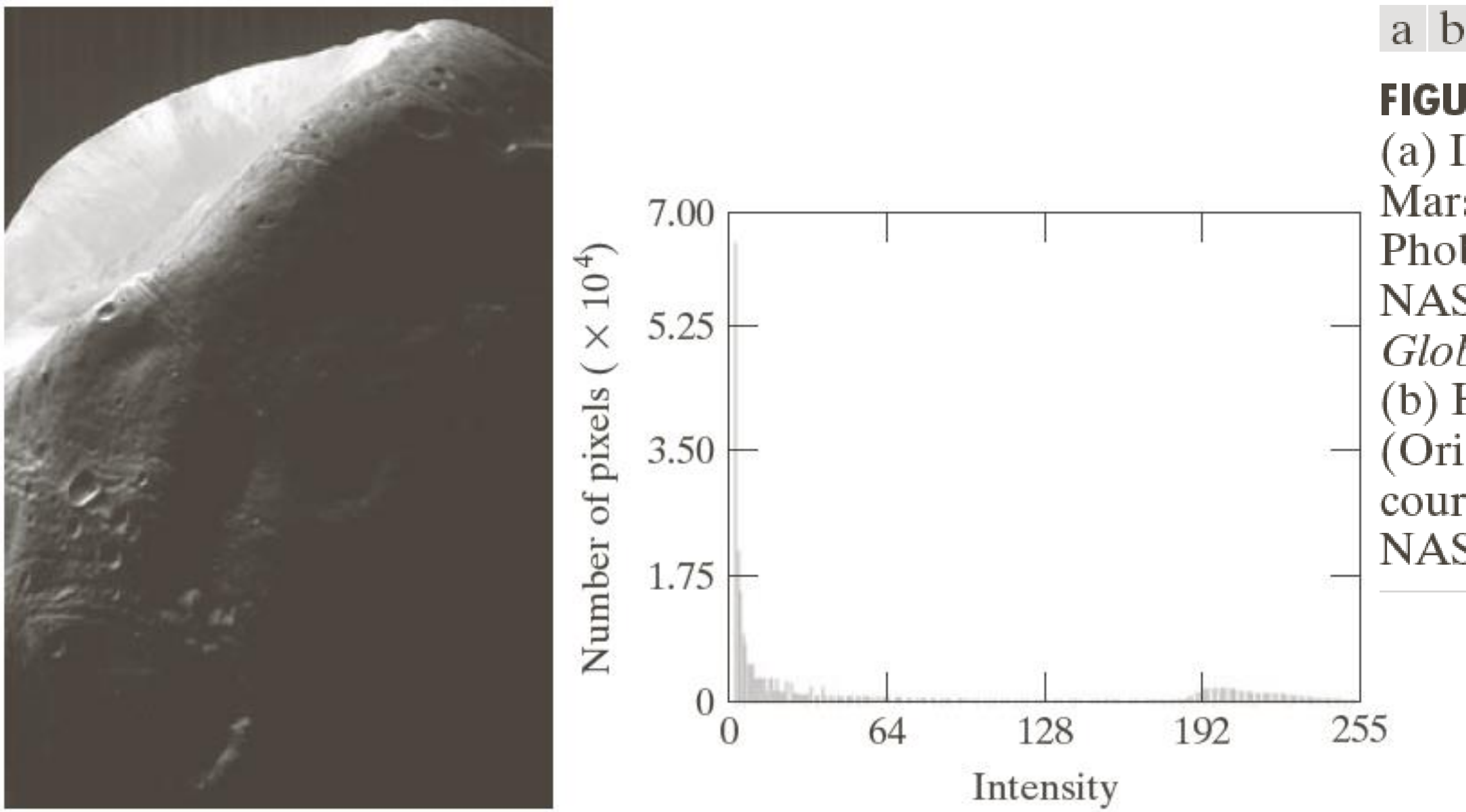
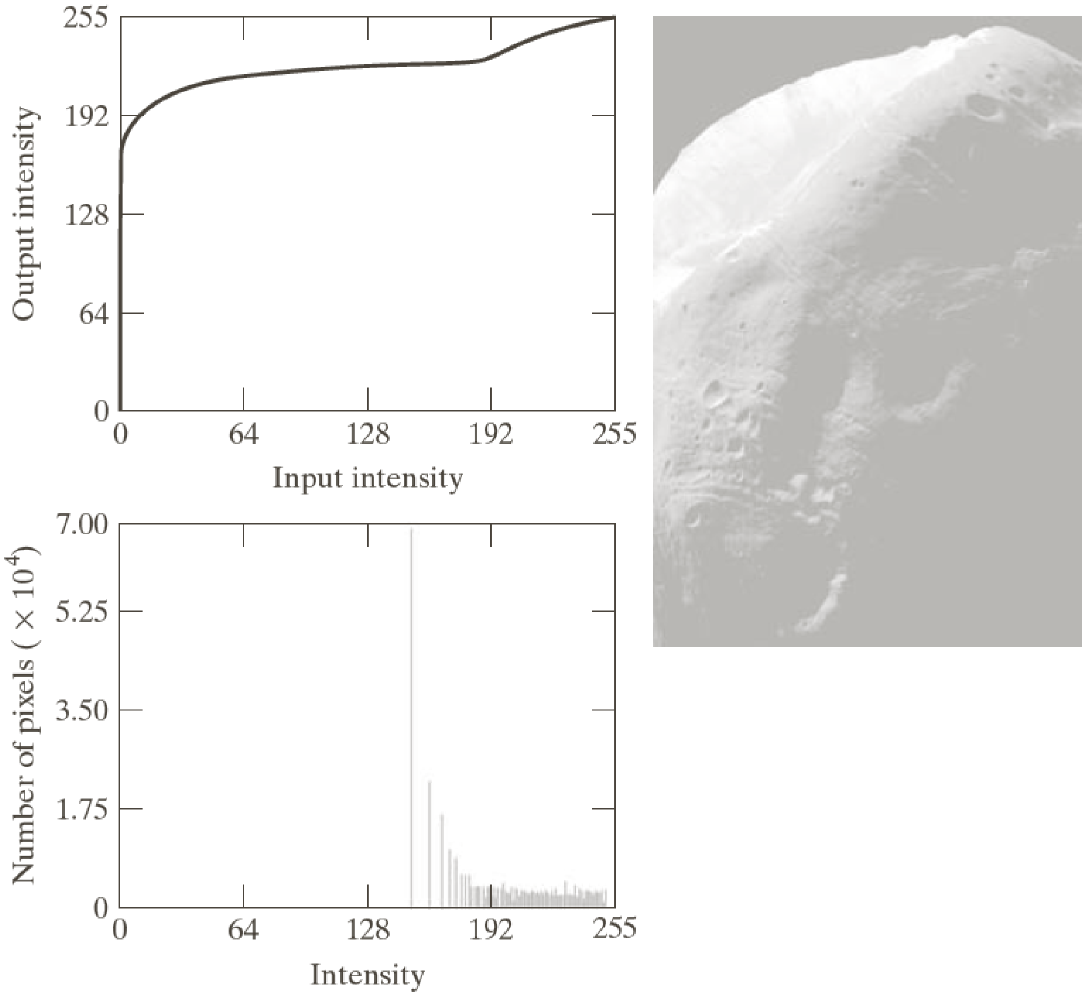


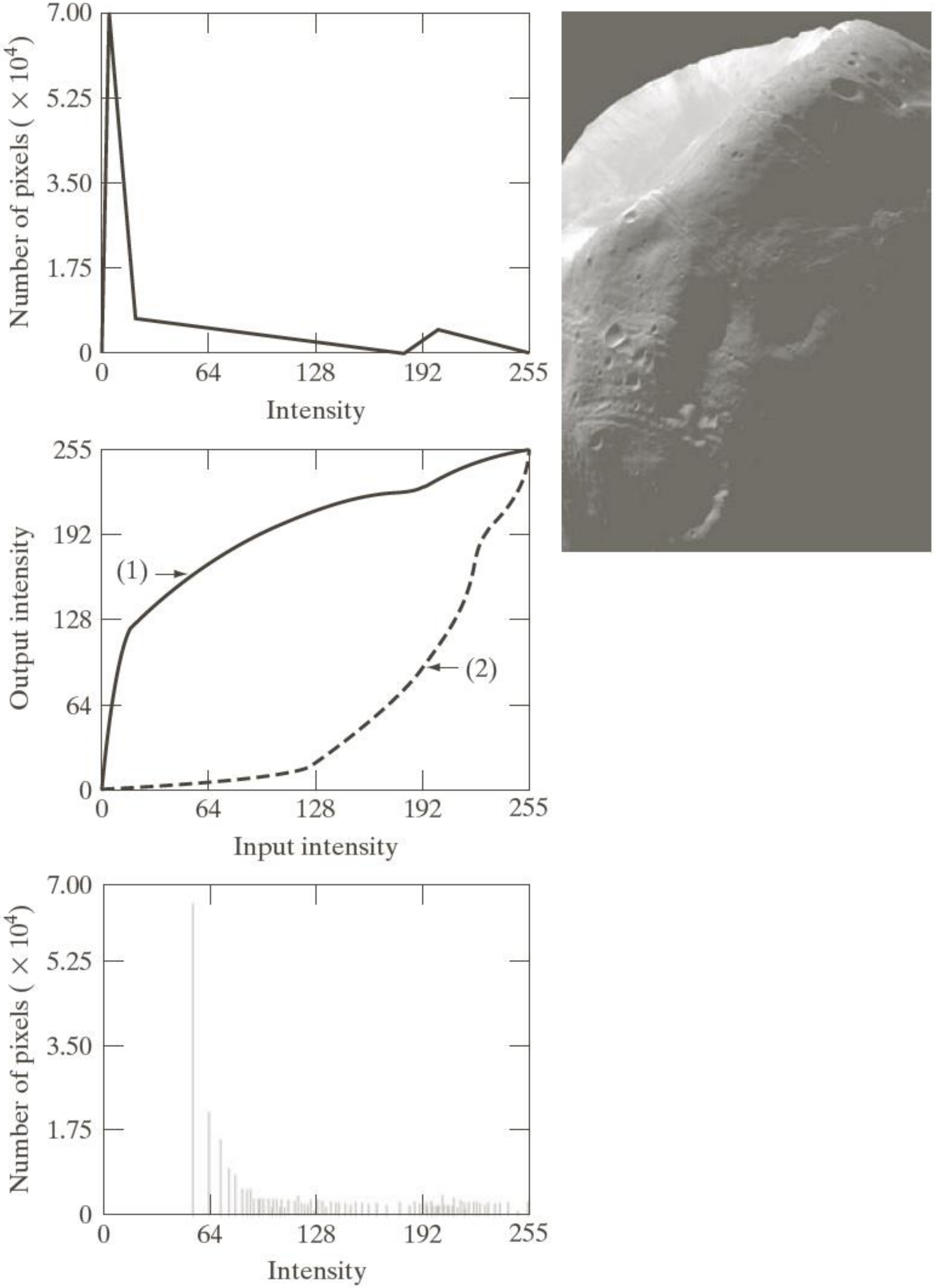
FIGURE 3.23 (a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor.* (b) Histogram. (Original image courtesy of NASA.)

Example: Histogram Matching



a b
c

FIGURE 3.24
(a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).



a c
b
d

FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

Arithmetic/logic operations

- Logic operations
- Image subtraction
- Image averaging

Logic operations

- Logic operations: pixel-wise AND, OR, NOT
- The pixel gray level values are taken as string of binary numbers

  Ex. 193 => 11000001

- Use the binary mask to take out the region of interest(ROI) from an image

Logic operations: example

A | B | A and B

AND

A or B

OR

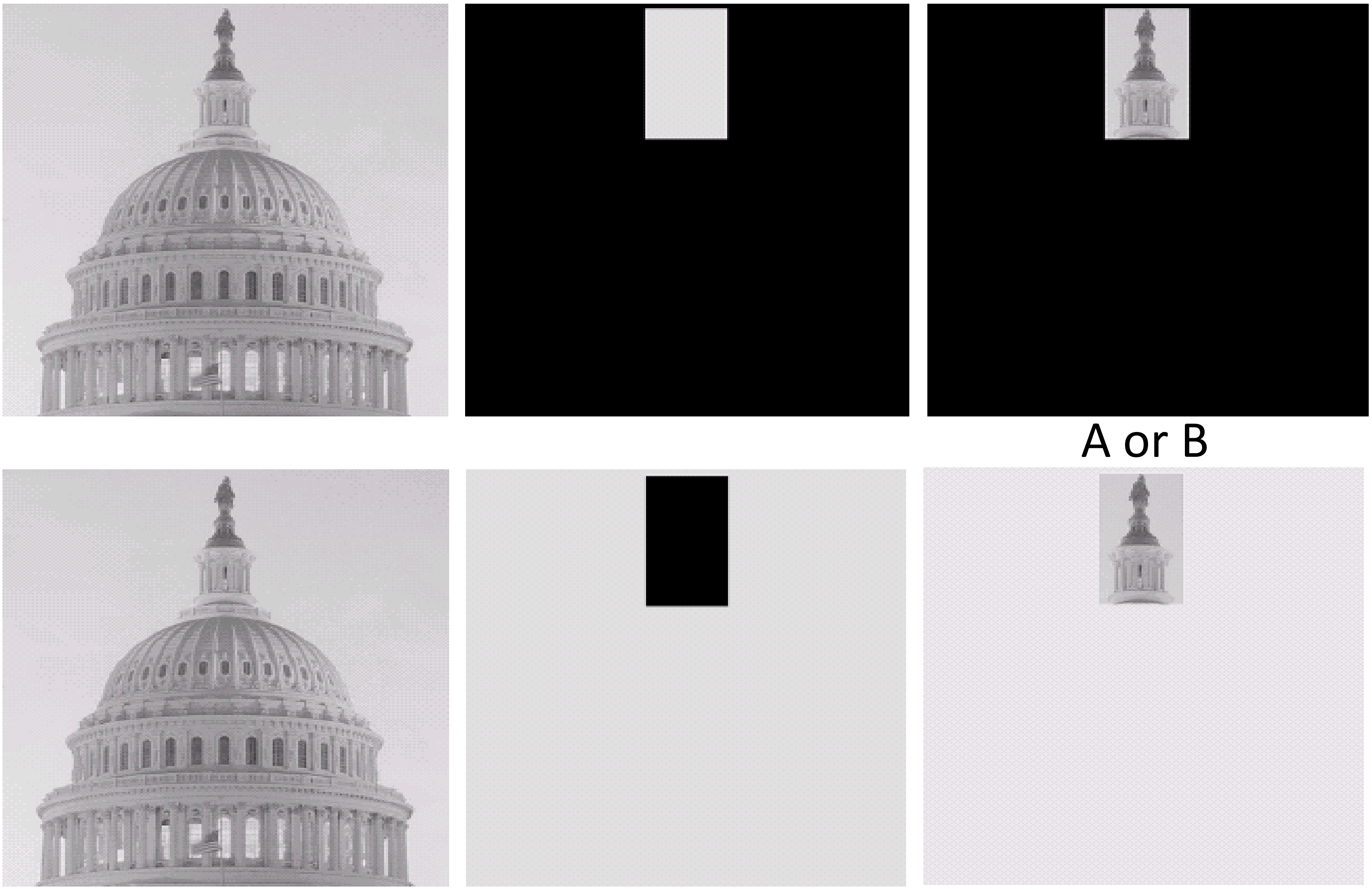
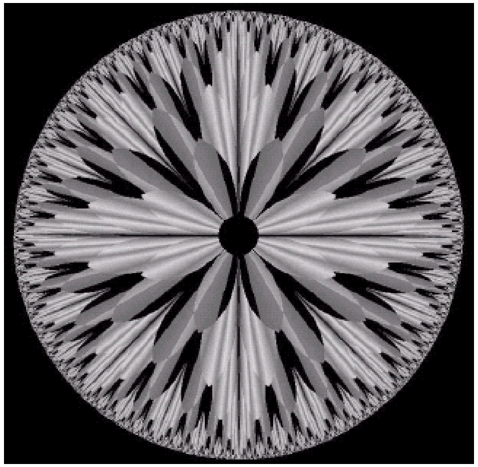


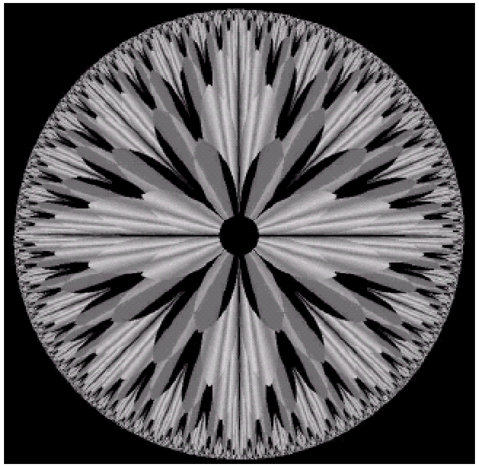
Image subtraction

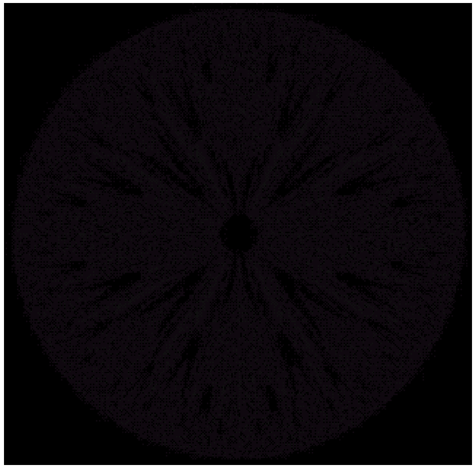
- Difference image
  g(x,y)=f(x,y)-h(x,y)

f:original(8 bits)



h:4 sig. bits





difference image

scaling

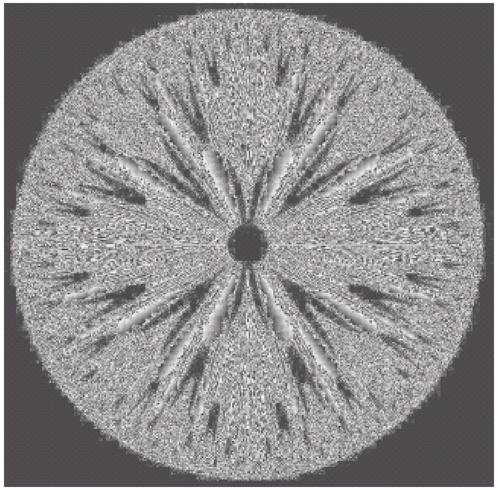


Image subtraction: scaling the difference image

- $g(x,y)=f(x,y)-h(x,y)$
  - f and h are 8-bit => $g(x,y) \in [-255, 255]$

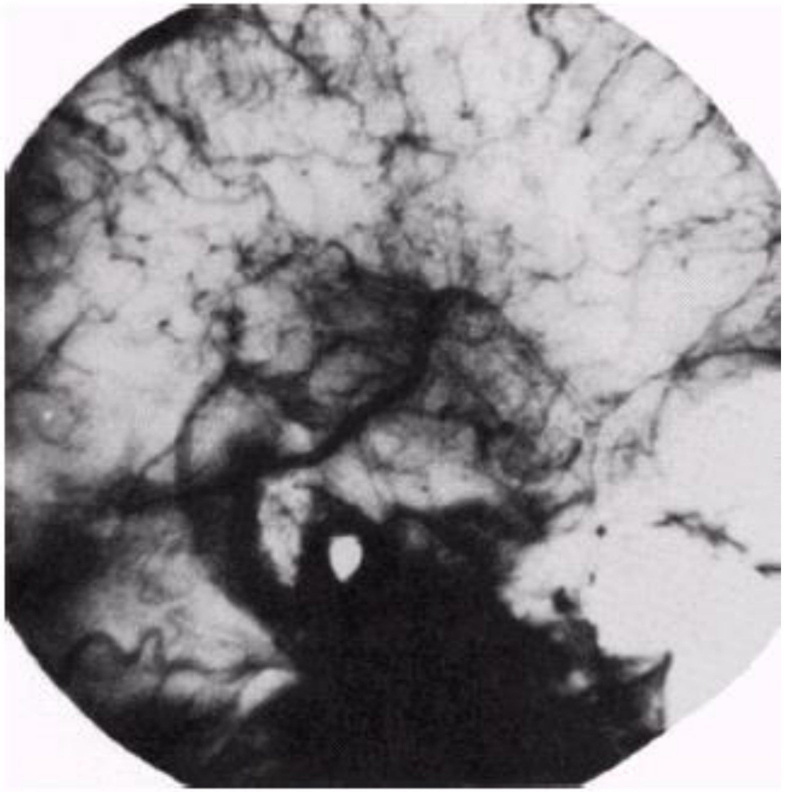
1. (1)+255 (2) divide by 2
   - The result won't cover [0,255]
2. (1)-min(g) (2) *255/max(g)

Be careful of the dynamic range after the image is processed.

Image subtraction example: mask mode radiography

- Inject contrast medium into bloodstream

original (head)



difference image

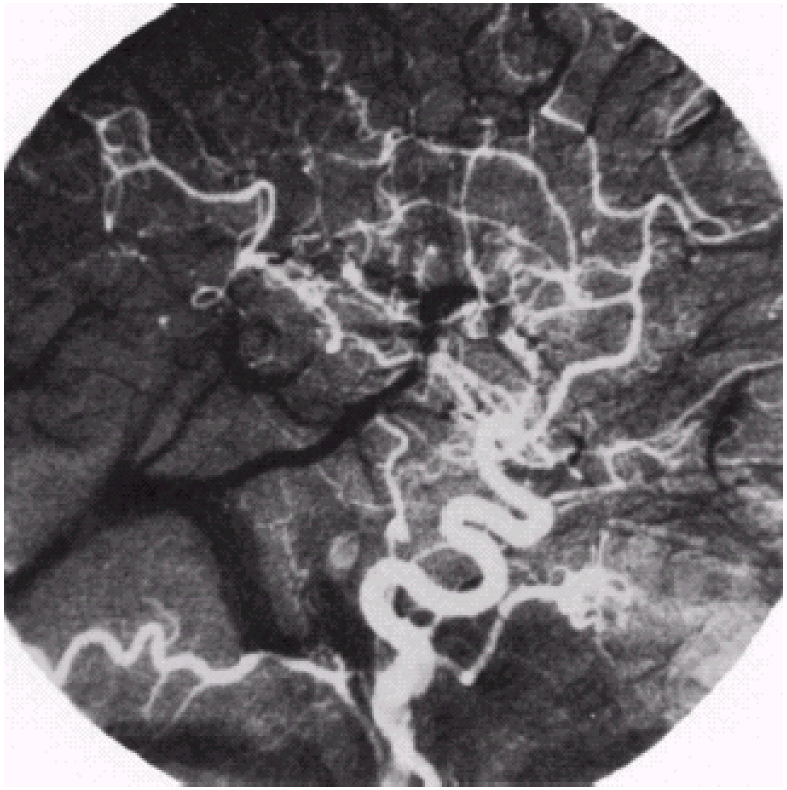
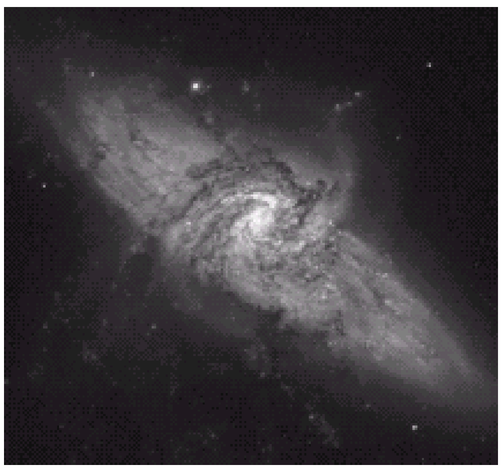


Image averaging

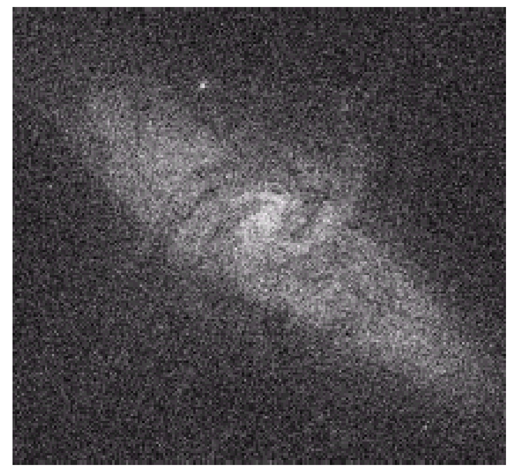
- Noisy image $g(x,y)=\mathbf{f(x,y)}+\eta(x,y)$

original

noise



Clear image



Noisy image

- Suppose $\eta(x,y)$ is uncorrelated and has zero mean

Image averaging: noise reduction

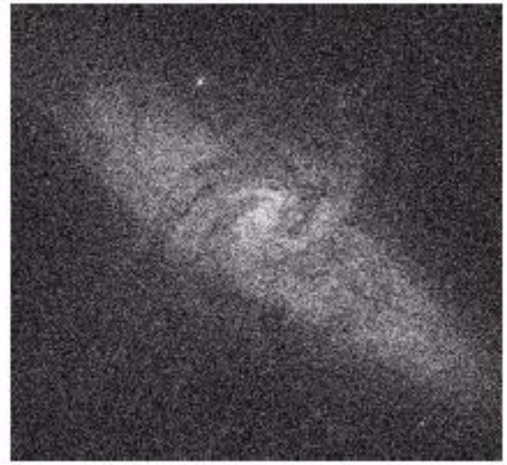
- Averaging over K noisy images $g_i(x,y)$

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

$$\begin{cases} E\{\bar{g}(x,y)\} = f(x,y) \\ \sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)} \end{cases}$$

$$\boxed{K \uparrow \Rightarrow \sigma^2 \downarrow}$$

original



Gaussian
noise

averaging
K=8

averaging
K=16

averaging
K=64

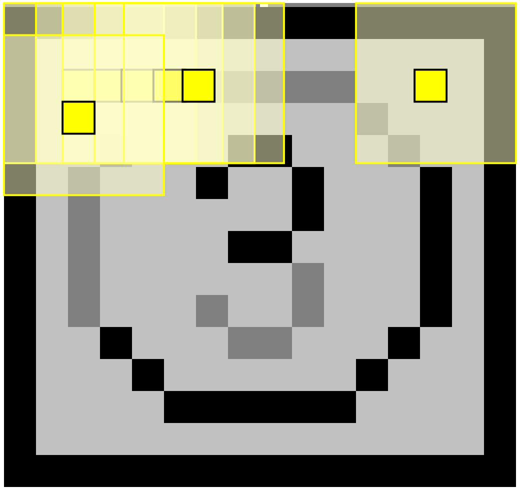
averaging
K=128

Local Enhancement

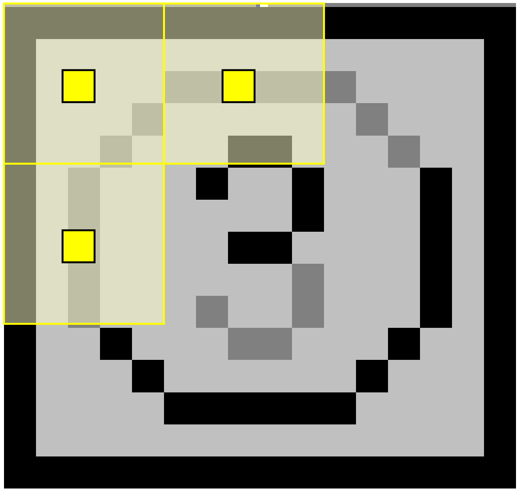
- Normally, Transformation function based on the content of an entire image
- Some cases it is necessary to enhance details over small areas in an image
- The histogram processing techniques are easily adaptable to local enhancement

Local Enhancement

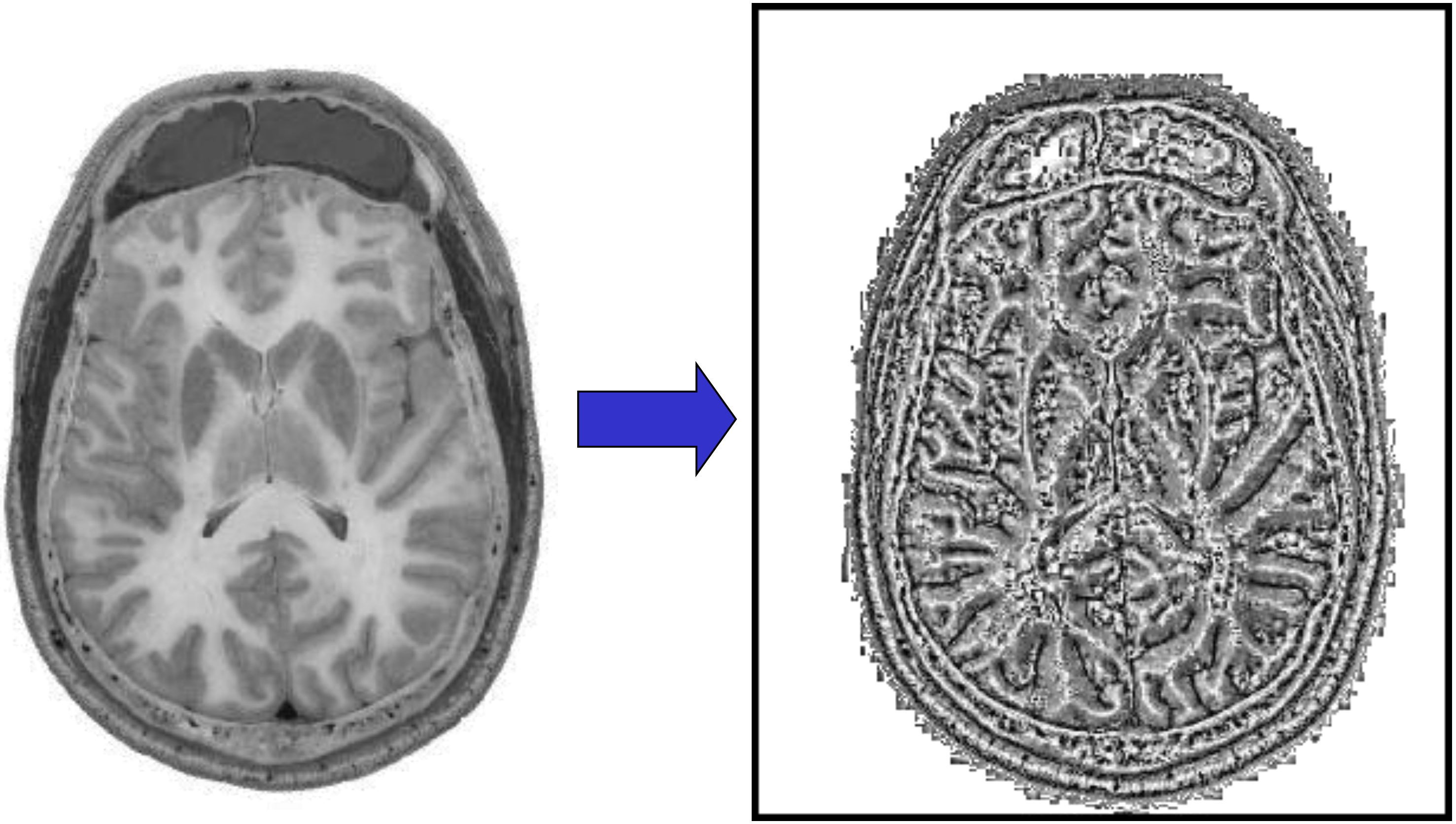
Pixel-to-pixel translation



Nonoverlapping region



Local Equalization



Enhancement Using Arithmetic/Logic Operations

- Are performed on a pixel-by-pixel basis between two or more images
- Logic operations are concerned with the ability to implement the AND, OR, and NOT logic operators because these three operators are functionally complete
- Arithmetic operations are concerned about +,-,*, / and so on (arithmetic operators)

Image Subtraction

- The difference between two images f(x,y) and h(x,y) expressed as

$$g(x,y) = f(x,y) - h(x,y)$$

Subtraction: MR-DSA

Unsubtracted

Subtracted

Image Averaging

- Consider a noisy image g(x,y) formed by the addition of noise to original image f(x,y)

$$g(x,y) = f(x,y) + \eta(x,y)$$ where $\eta(x,y)$ is noise

Where the assumption is that at every pair of coordinates(x,y) the noise is uncorrelated and has zero average value.

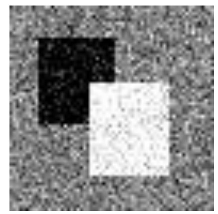
- The objective of this procedure is to reduce the noise content.

Image Averaging

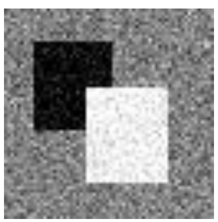
- Let there are K different noisy images
- If an image $\bar{g}(x, y)$ is formed by averaging K different noisy images

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^{K} g_i(x, y)$$

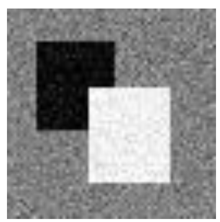
Image Averaging (Gray Scale)



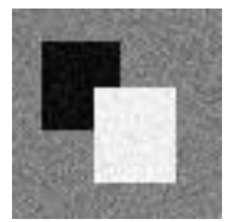
1 image



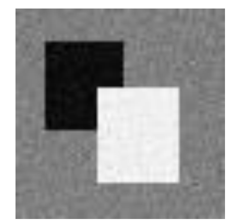
2 images



5 images



10 images



20 images

Image Averaging (Color Image)


(1)

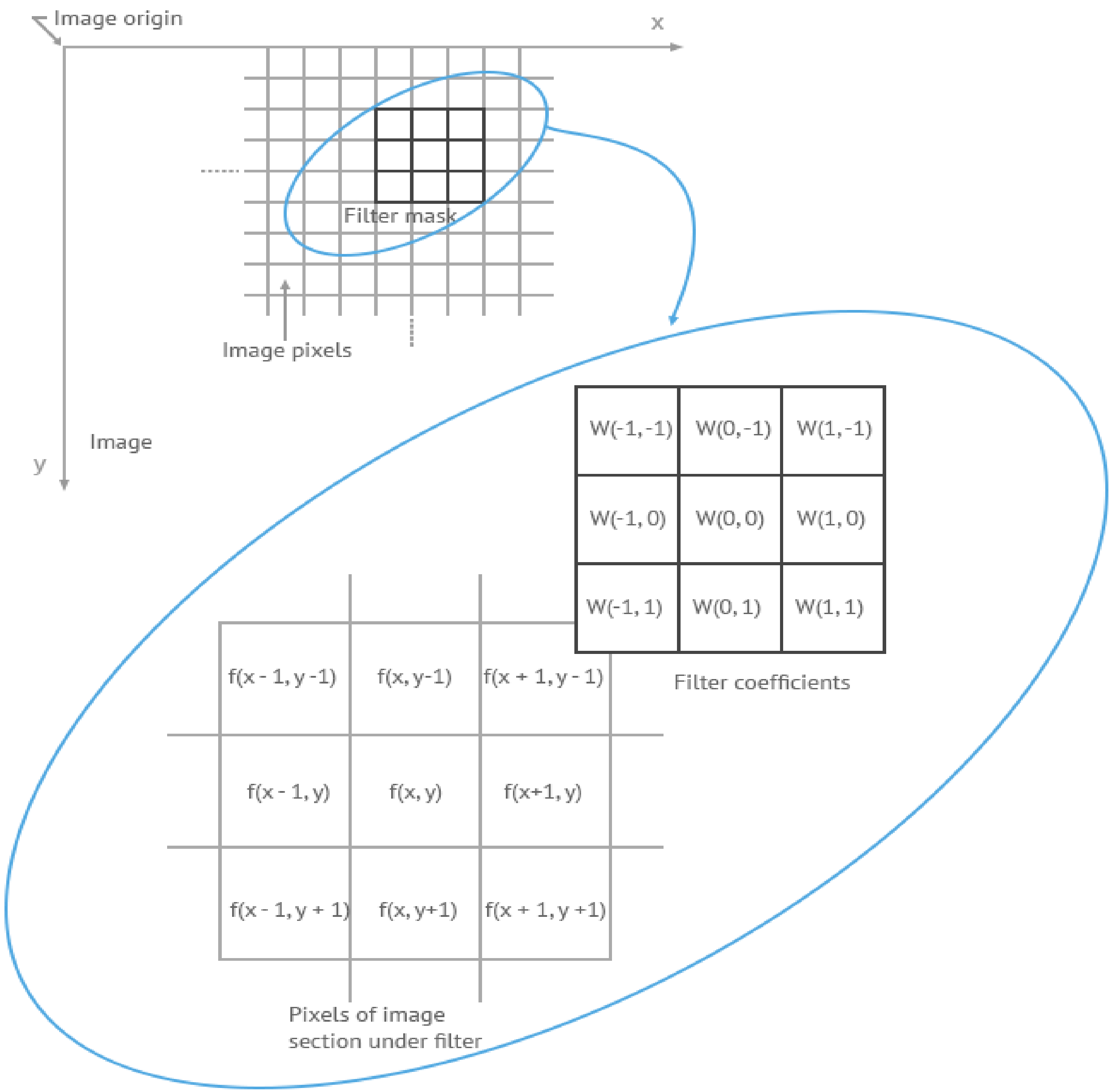

(2)


(3)

Average image

Spatial filtering

Basics of Spatial Filtering



- Mask, Convolution kernels
- Odd sizes

Spatial Filtering

- Some neighbourhood operation between neighbourhood of image pixel and corresponding value of mask/filter/kernel/window that has the same dimension as the neighbourhood
- Mask size is M x N and M=2a+1, N=2b+1
- Modification at border is handled by padding rows and col. of ‘0’s or by replicating some rows and columns

Spatial Filtering

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

a=(m-1)/2 and *b=(n-1)/2*,
m x *n* (odd numbers)

For *x*=0,1,…,M-1 and *y*=0,1,…,N-1

The basic approach is to sum products between the mask coefficients and the intensities of the pixels under the mask at a specific location in the image:

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9$$ (for a 3 x 3 filter)

Neighborhood Averaging

Each point in the smoothed image, $\hat{F}(x, y)$ is obtained from the average pixel value in a neighbourhood of (x, y) in the input image.

For example, if we use a 3×3 neighbourhood around each pixel we would use the mask

$$\begin{array}{ccc} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{array}$$

General Spatial Filter

FIGURE 3.33
Another representation of a general 3×3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

a b

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

FIGURE 3.34 Two 3×3 smoothing (averaging) filter masks. The constant multipli er in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

Smoothing Spatial Filters

- Smoothing linear filters: averaging filters, low-pass filters
  - Box filter
  - Weighted average
- Order-statistics filters:
  - Median-filter: removing salt-and-pepper noise
  - Max filter
  - Min filter

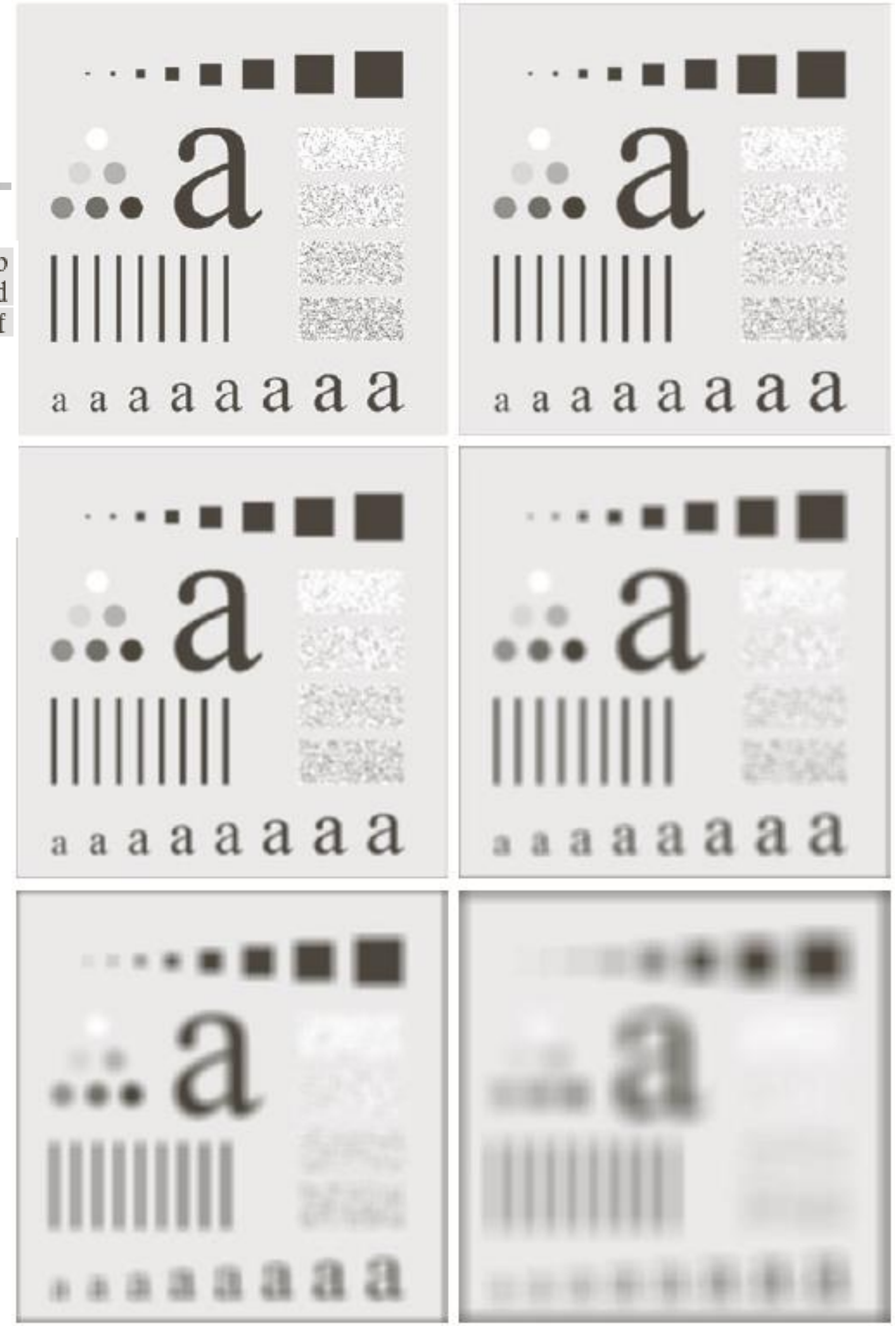
$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Smoothing Spatial Filters

- Average or Low pass filter
- Also called box filter
- Blurring of edges
- Smoothing of false contours
- Blurring can be reduced by using weighted average filters. That is assigning more weight to centre and less weight to neighbouring pixels
- Image details that are of same size as the filter mask gets affected considerably
- Jagged borders or edged appears pleasingly smooth
- Generally used to eliminate small objects. That is interest is gross representation of image. Size of the mask decides relative size of the object that will be blended with the background

FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

a b
c d
e f



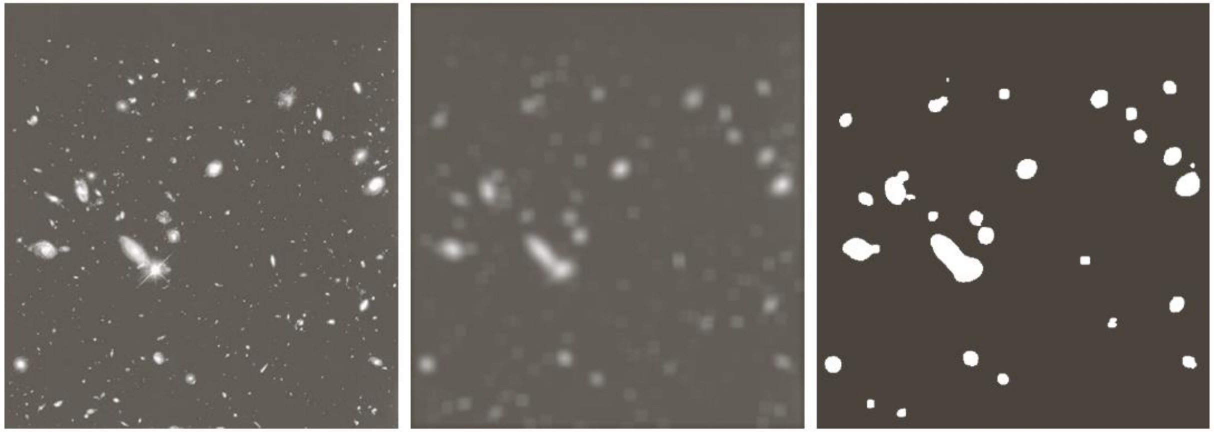
Smoothing linear filter

Original Image

Averaging by
1/9*ones(3,3)

Averaging by
1/36*ones(6,6)

A spatial averaging filter that all coefficients are equal is called box filter



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

- Mean Filter, Median Filter, Enhancement Filter.

1- These are implemented with convolution masks. Result is some weighted sum of the values of a pixel and its neighbours. Also called linear filter.

2- General Prediction.

- If mask coefficients sum to 1 – Average brightness of the image will be retained.
- If mask coefficients sum to 0 – Resulting image will return a dark image.
- If coefficients are alternating +ve and –ve – Filter will return edge information.
- If coefficients are all +ve – filter will blur the image.

Order Static's Filter

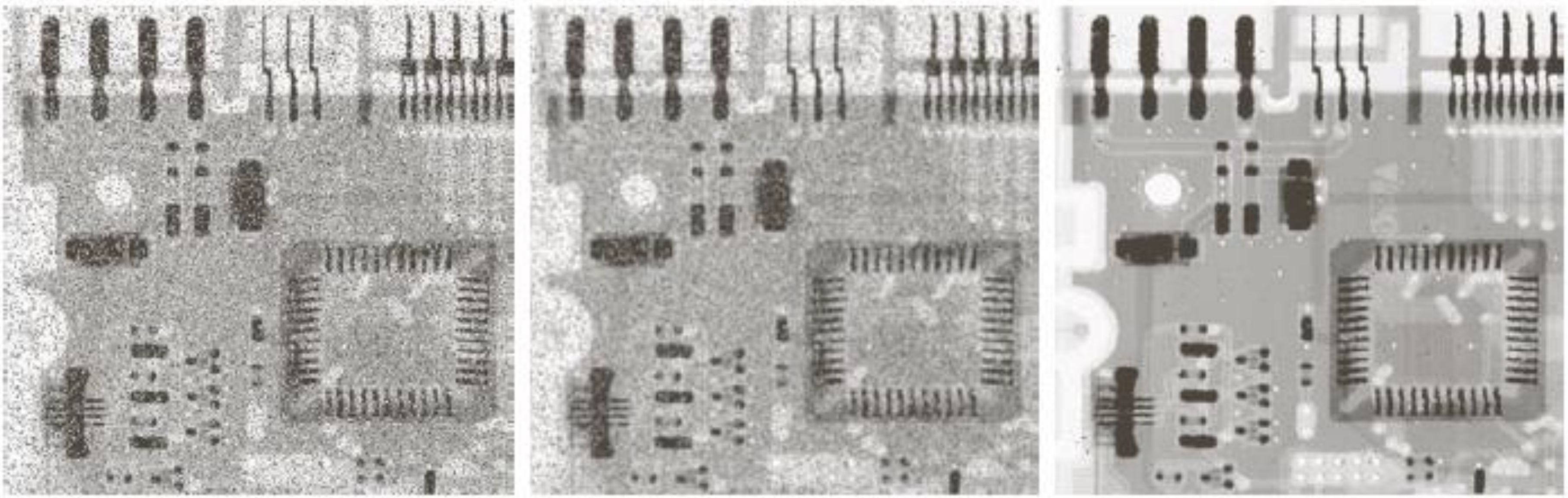
- Order statistics filters are nonlinear spatial filters whose response is based on ordering the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result

- The best known example in this category is median filter. Each output pixel contains the median value in the m-by-n neighborhood around the corresponding pixel in the input image

Median Filter

- Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper (impulse)" noise. Median filtering is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges.

- The principle function of the median filter is to force points with distinct gray levels to be more like their neighbors.
- For e.g.3*3 neighborhood has values [10 20 20 20 15 20 20 25 100]. These values are sorted as [10 15 20 20 20 20 20 25 100] which result in a median of 20.



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Corrupted by
salt and pepper noise
Noise density 2%

Averaging
Mask 3*3

Using median filter

original

added noise

average

median

Sharpening Spatial Filters

- To highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition

- **Blurring vs Sharpening**
  - Blurring/smooth is done in spatial domain by pixel averaging in a neighbors, it is a process of integration
  - Sharpening is an inverse process, to find the difference by the neighborhood, done by spatial differentiation. spatial differentiation

Sharpening Spatial filters

- **Applications**:

  - Electronic printing
  - Medical imaging
  - Industrial inspections and
  - Autonomous guidance in military systems

THANK YOU