Image Enhancement (Spatial Domain Methods)

What Is Image Enhancement?

- Image enhancement is the process of making images more useful
- The reasons for doing this include:
  - Highlighting interesting detail in images
  - Emphasize, sharpen or smoothen image features
  - Removing noise from images
  - Making images more visually appealing
  - Enhance otherwise hidden information

Classification of Image enhancement

- Spatial Domain
  - Process intensity of pixels
  - Two types- intensity transformation and spatial filtering
- Transform Domain
  - Compute transform of image
  - process transformed image
  - then find inverse transform to get image in spatial domain

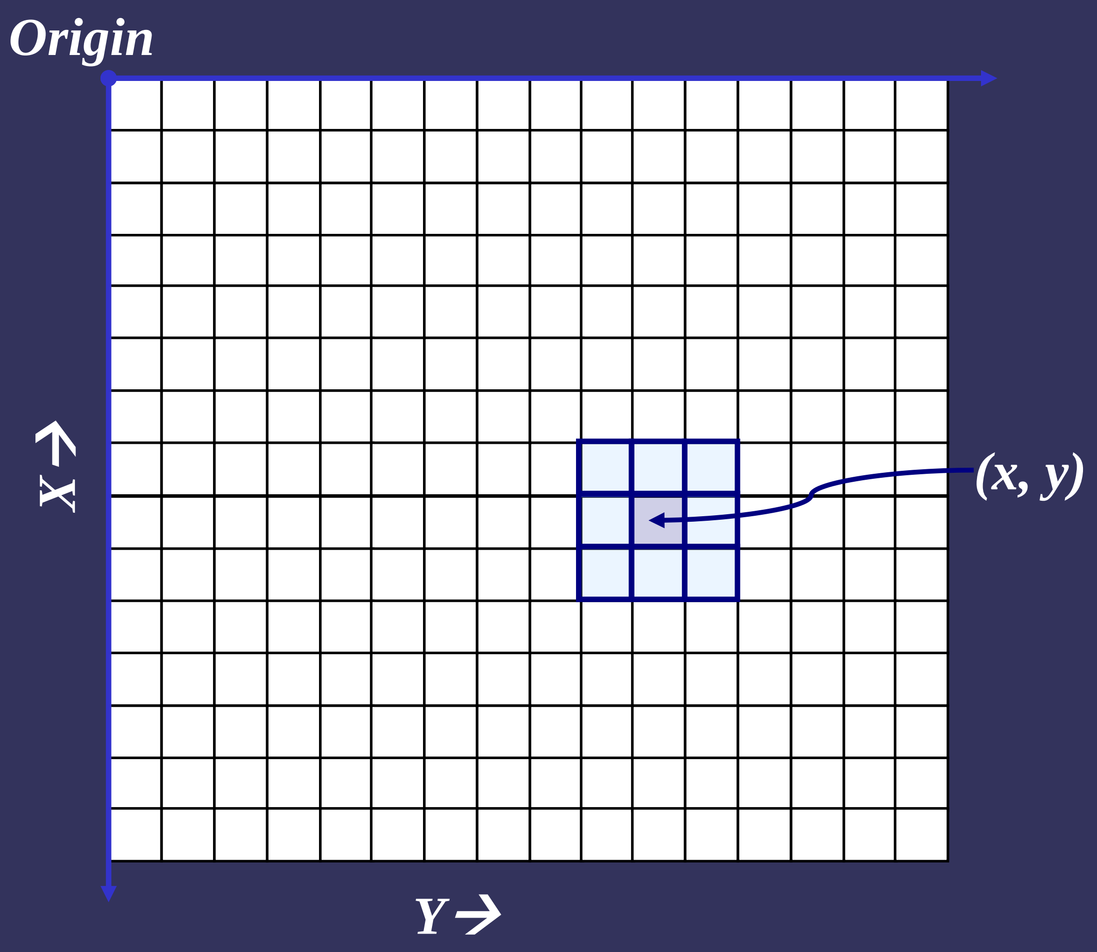
Enhancement in Spatial Domain

- *g (x, y) = T[f (x, y)]*

f (x, y) is input image

g (x, y) is processed image
T is operator defined over some neighbourhood of *(x, y)*

Image f (x, y)



Classification of spatial domain

- Point operation
- Mask operation
- Global operation

Point Processing

$$s = T(r)$$

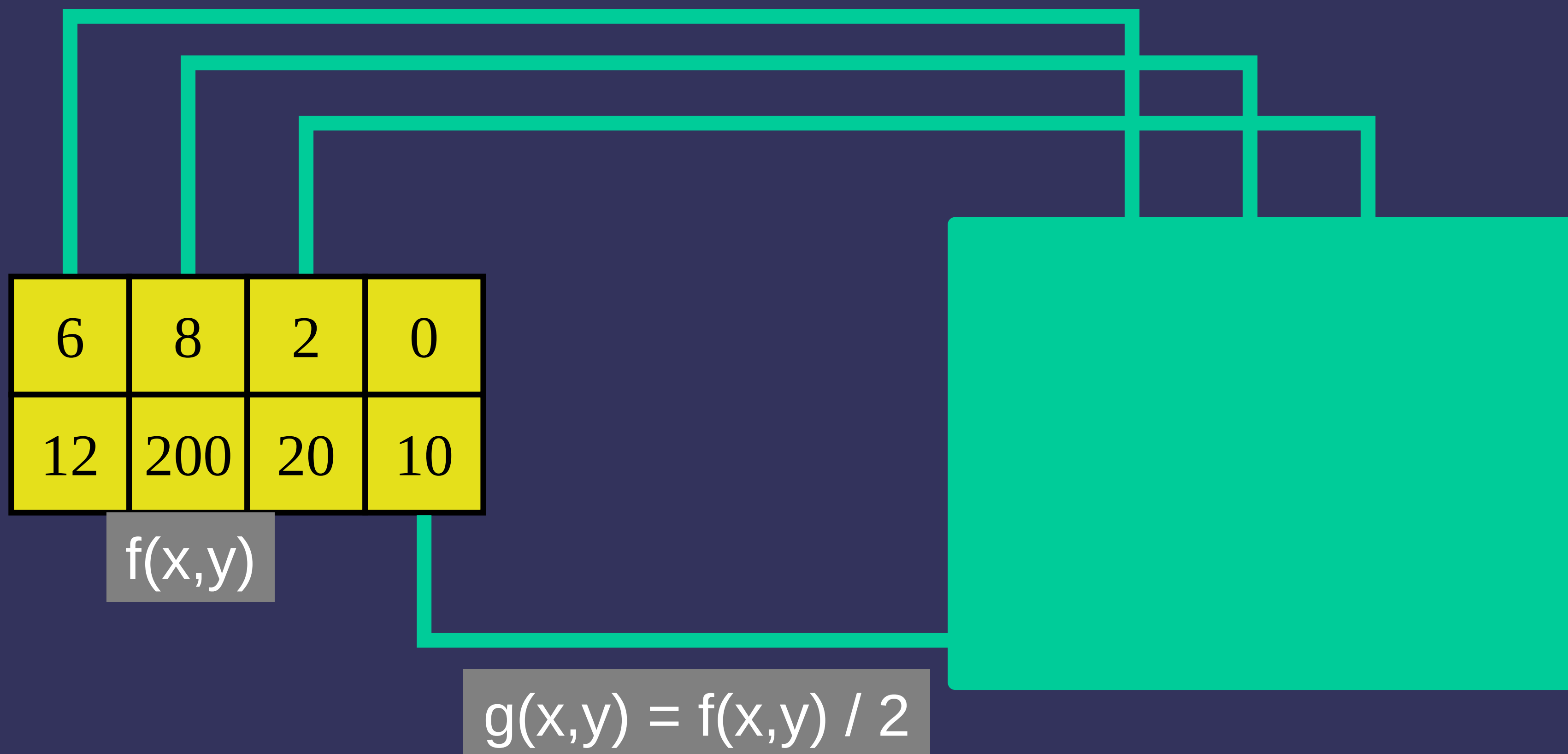
r is pixel value of the original image at (x,y)

s is pixel value of the processed image at (x,y)

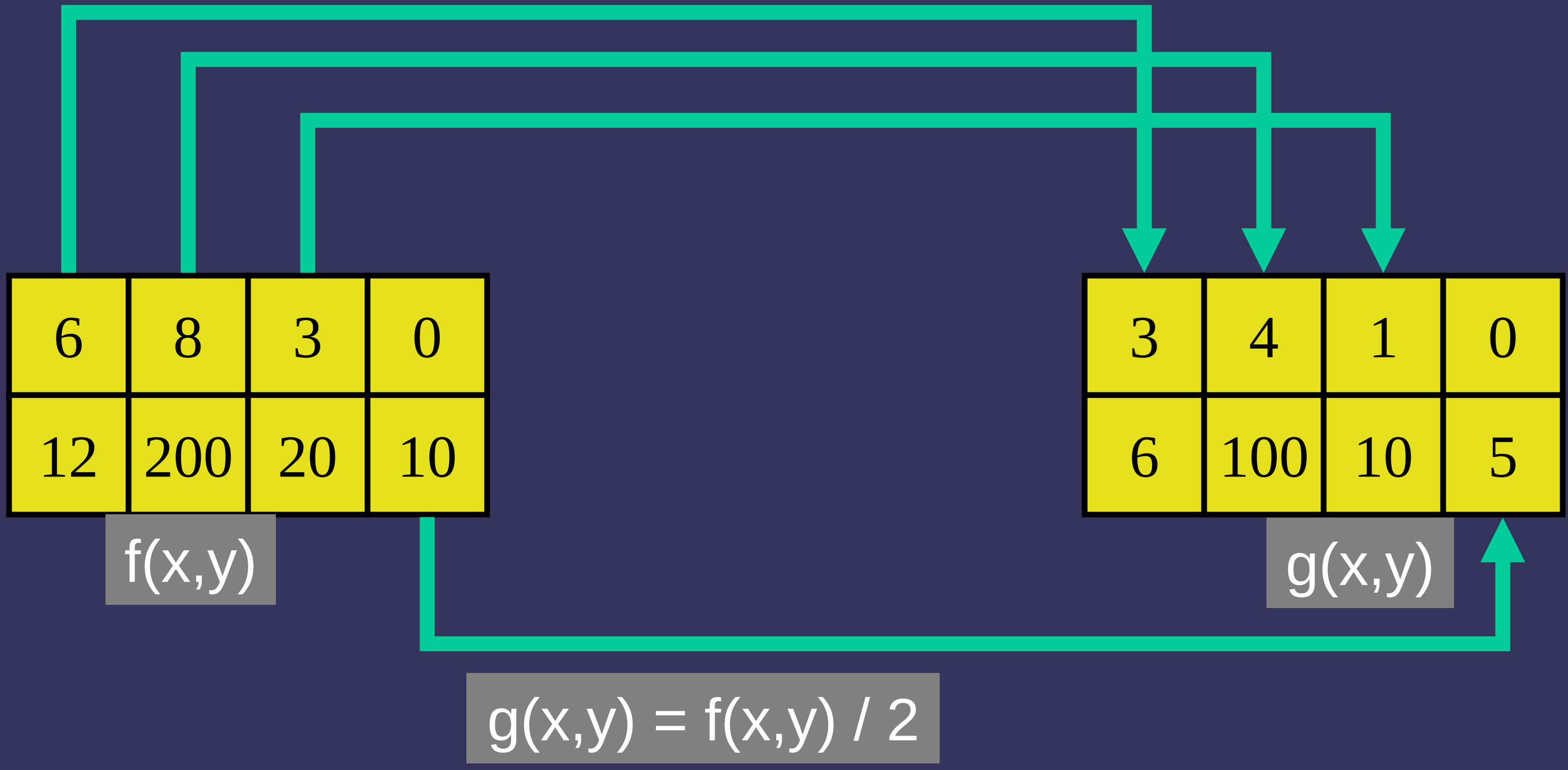
T is a grey level transformation function for a point

$$f(x,y) \rightarrow g(x,y)$$

Point Operation



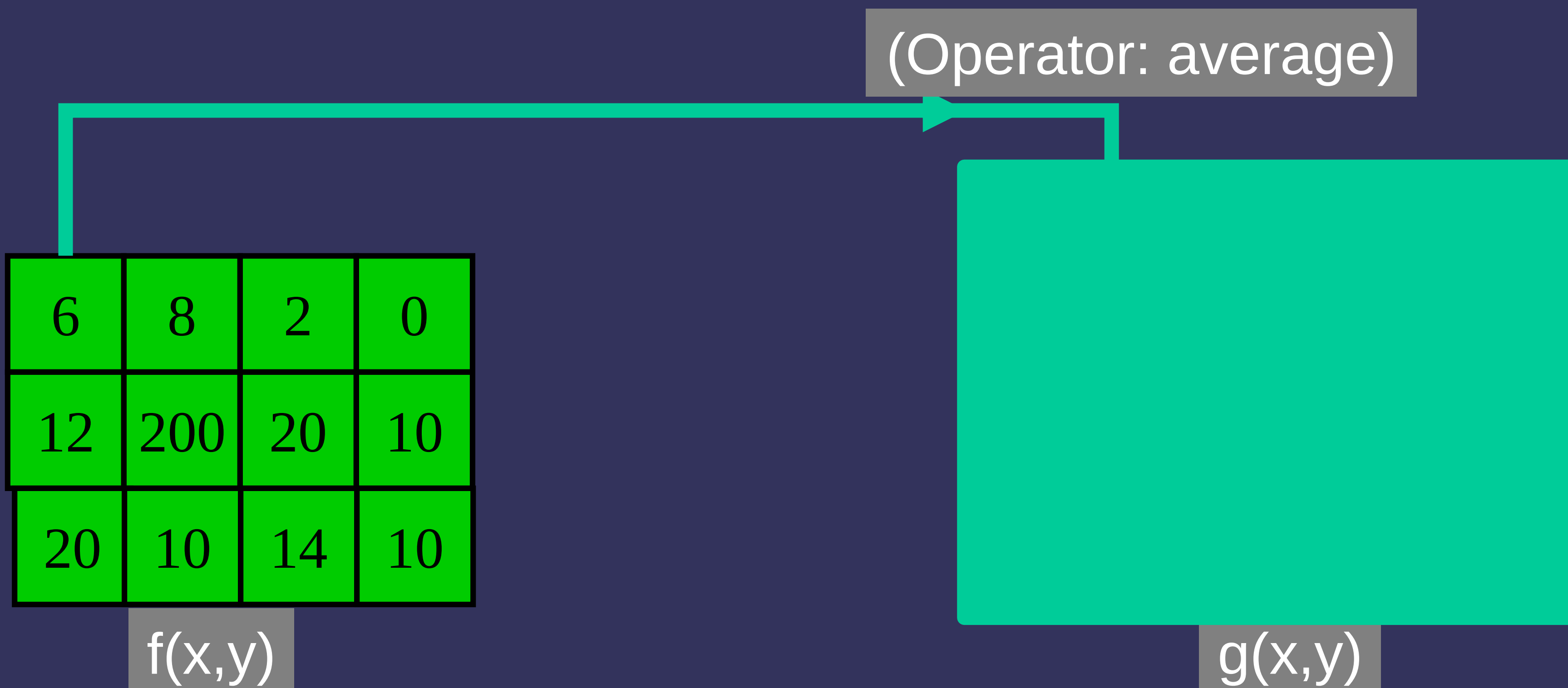
Point Operation



Neighbourhood Processing

The operator T can be defined over the set of ‘neighbourhood’, N(x, y) of each pixel

Neighbourhood Operation



Neighbourhood Operation

(Operator: average)

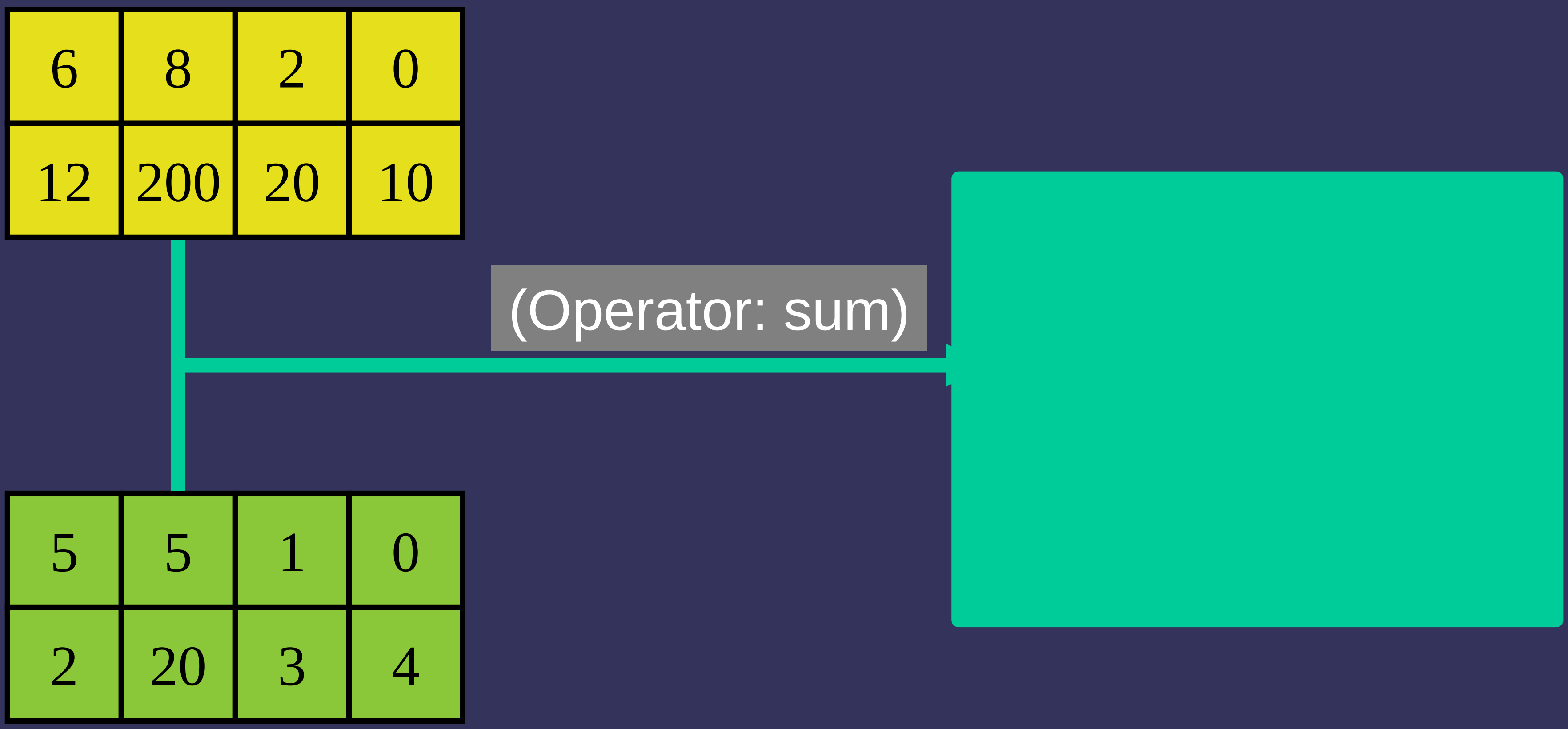
6	8	2	0
12	200	20	10
20	10	14	10

f(x,y)

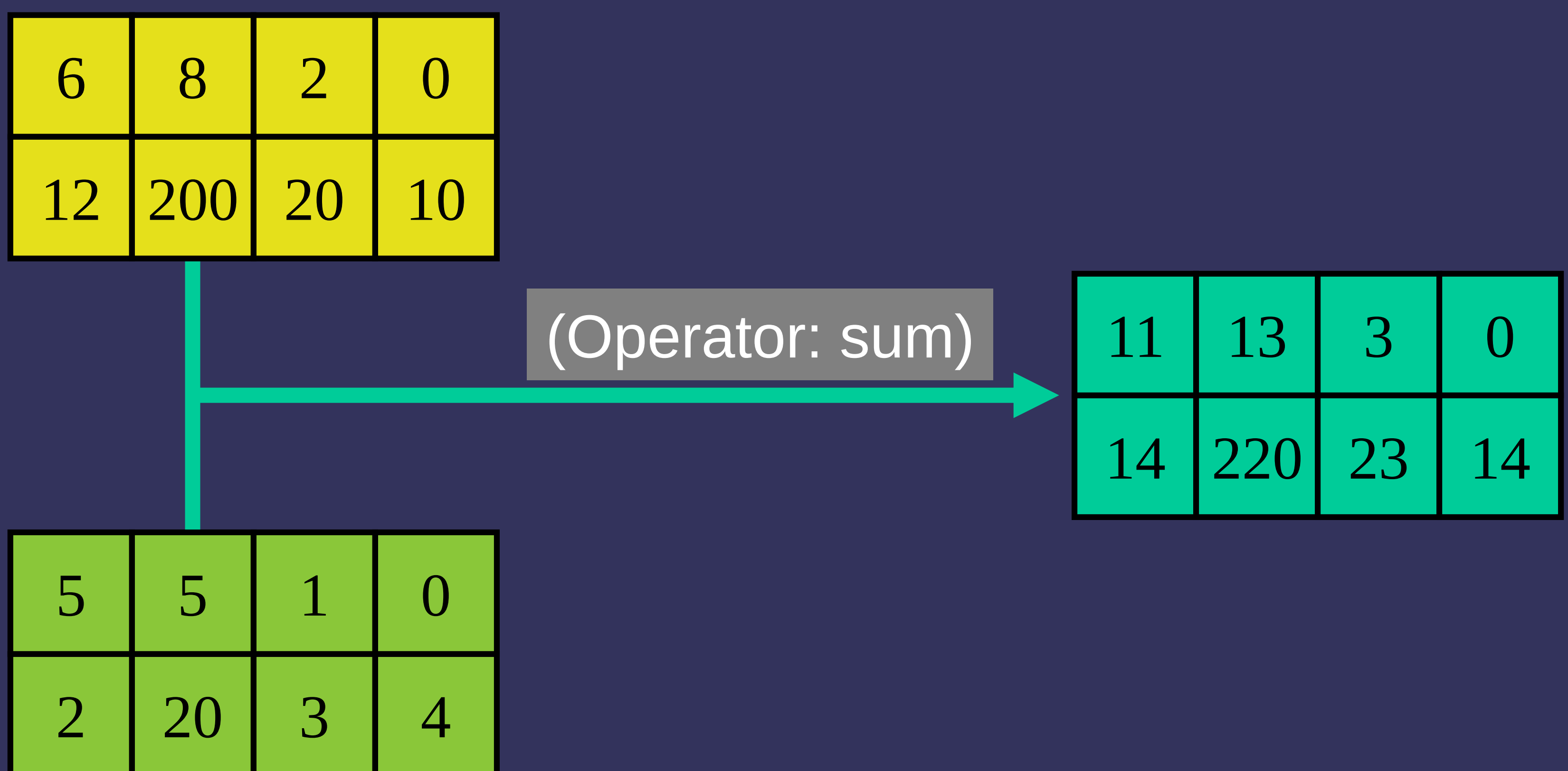
6	8	2	0
12	32	30	10
20	10	14	10

g(x,y)

Global Operation



Global Operation



Point operation

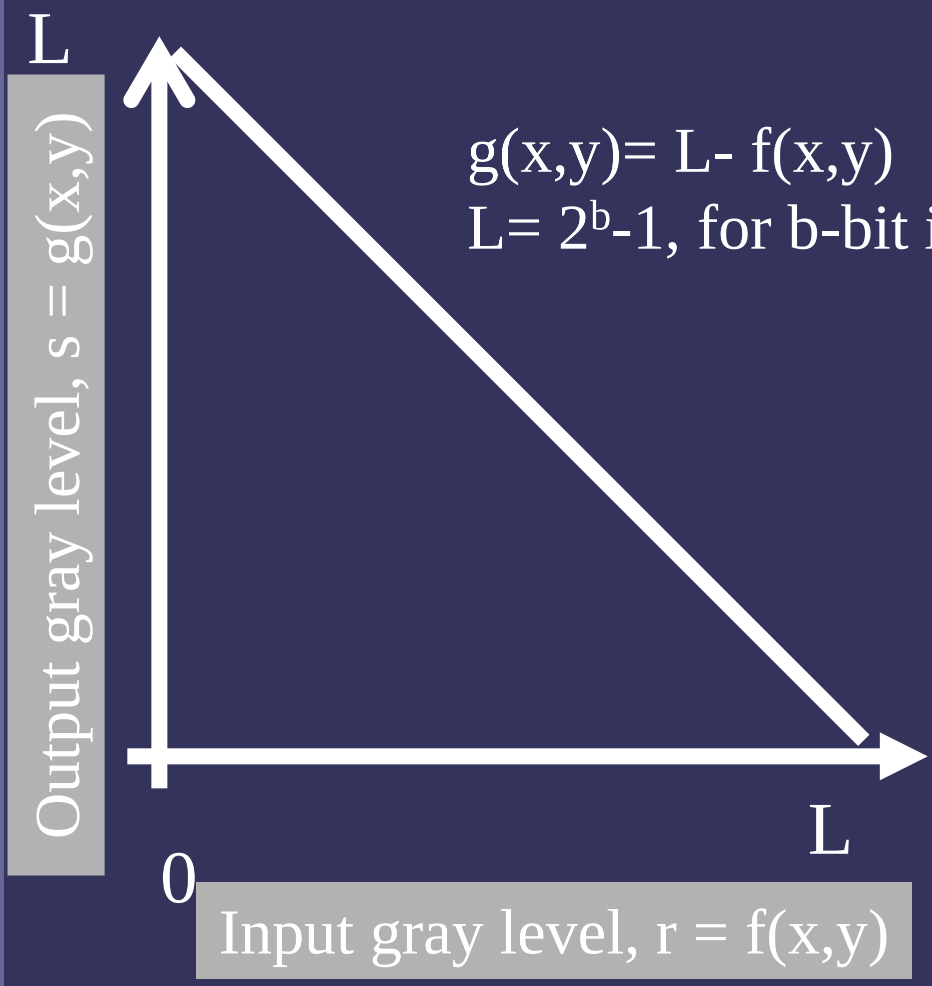
- Brightness modification
- Contrast manipulation
- Histogram manipulation

Gray Level/Intensity Transformations

- Brightness modification
- Log transformations
- Power Law transformations
- Piecewise-Linear transformation Functions

Brightness modification: Image Negative

Suited for enhancing white or grey detail embedded in dark region i.e. black area predominates

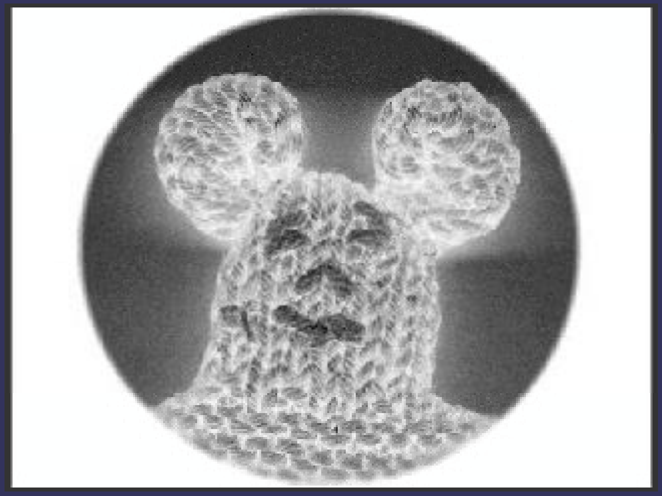


$g(x,y) = L - f(x,y)$

$L = 2^b - 1$, for b-bit image

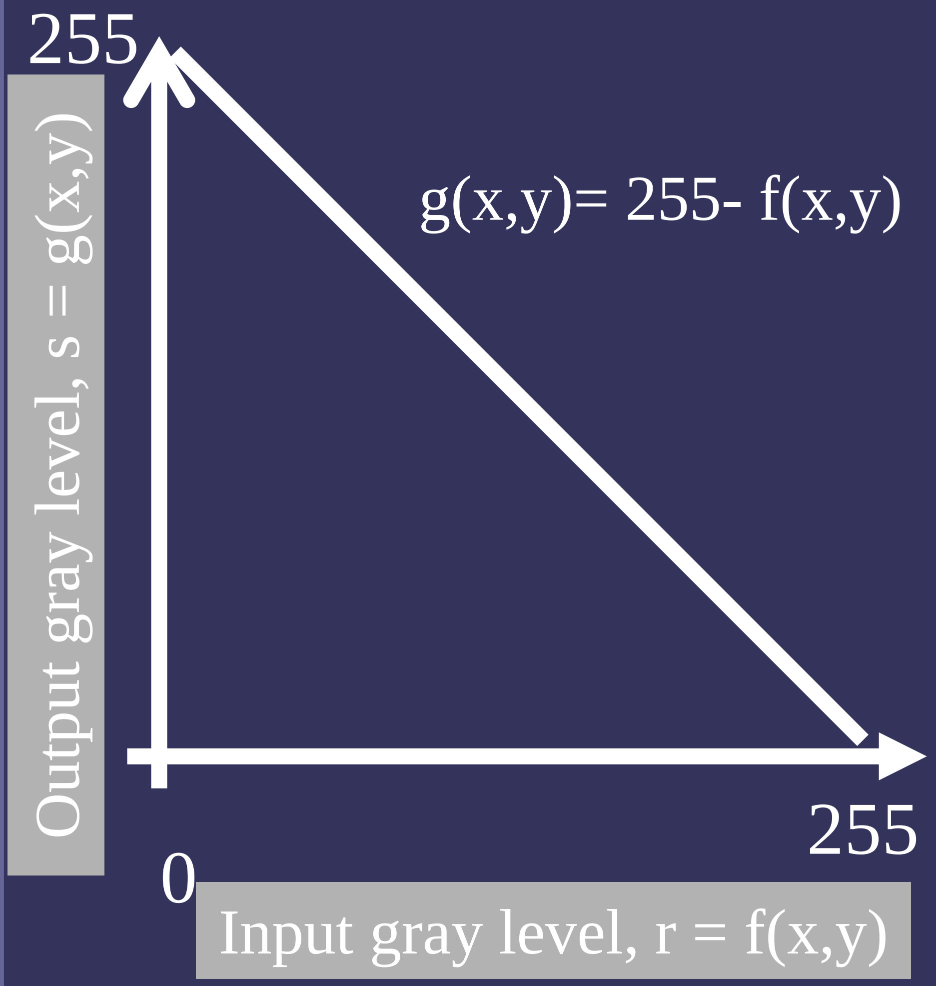
f(x,y)

g(x,y)



Brightness modification: Image Negative

For 8-bit image, L=255



f(x,y)

20	0	100	15
10	25	255	30
0	10	55	10
15	0	200	100

g(x,y)

235	255	155	240
245	230	0	225
255	245	200	245
240	255	55	155

example, 3-bit image negative

Image matrix is given by

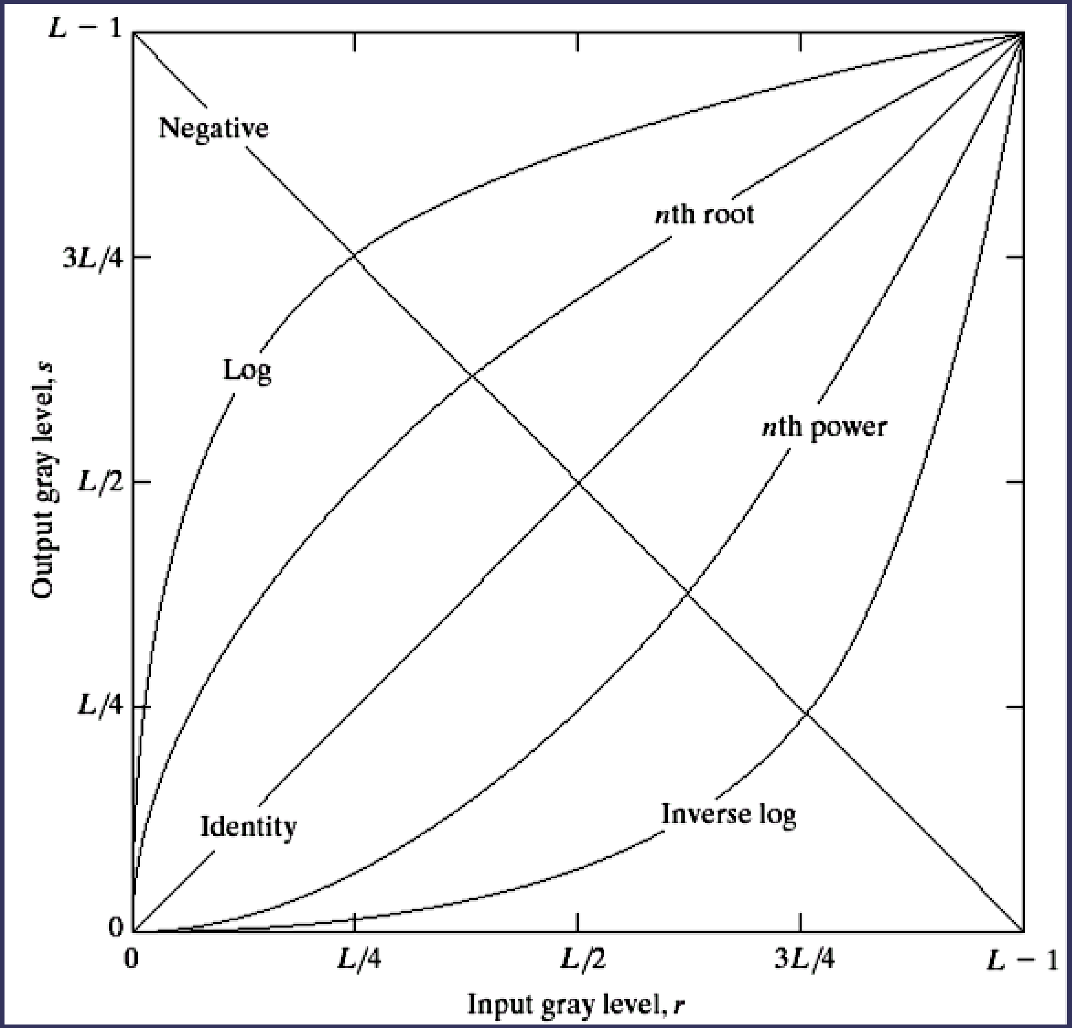
$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

Compute image negative

$$A1=7\text{-}A = \begin{bmatrix} 5 & 4 & 7 & 1 & 0 \\ 7 & 4 & 0 & 2 & 5 \\ 2 & 4 & 5 & 3 & 7 \\ 3 & 5 & 5 & 6 & 7 \\ 6 & 0 & 1 & 3 & 2 \end{bmatrix}$$

Intensity Level Transformations

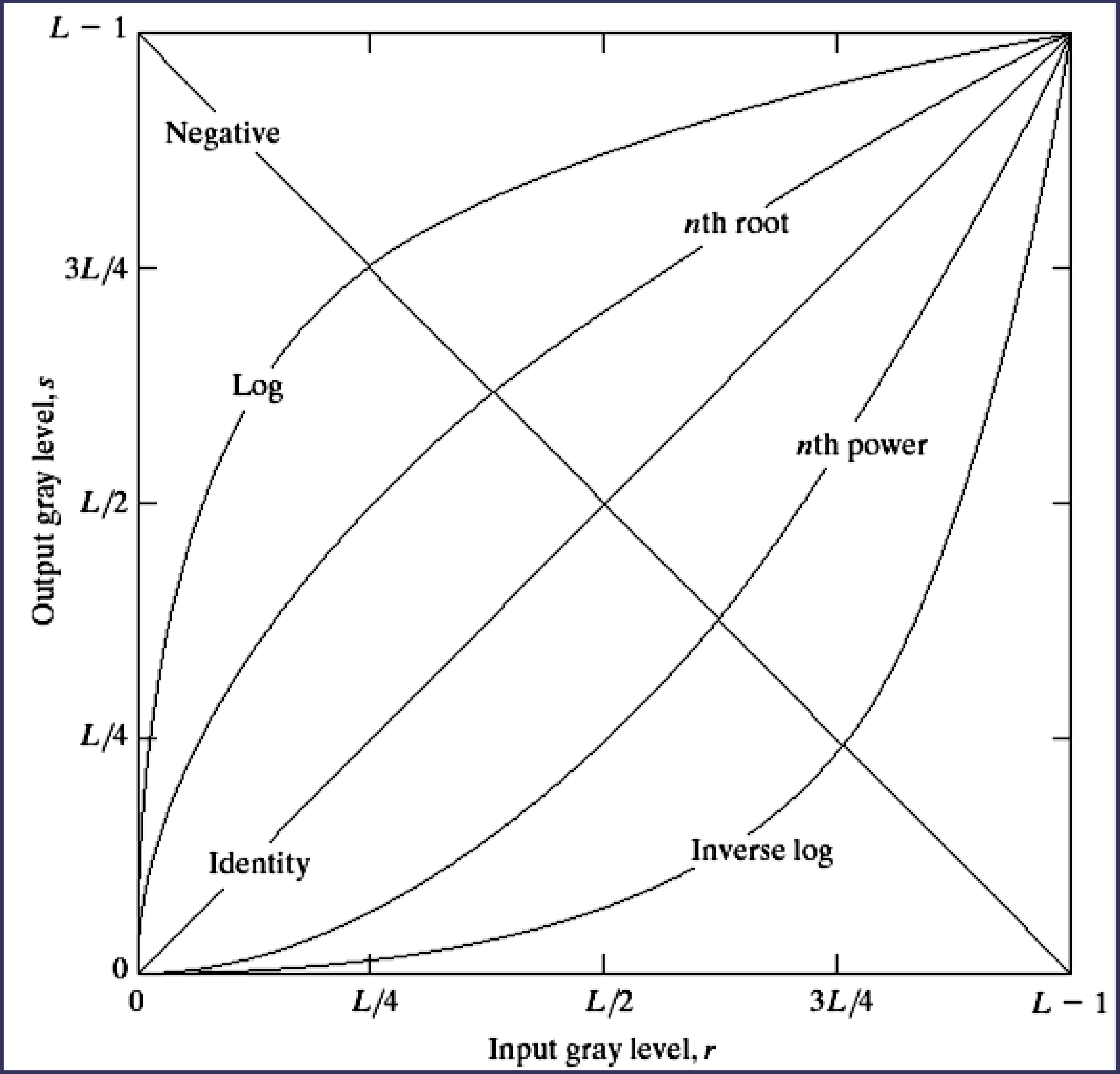
- Linear
  - Negative/Identity
- Logarithmic
  - Log/Inverse log
- Power law
  - n^{th} power/n^{th} root



maximum amplitude, L=255

Logarithmic Transformations

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



- The log transformation maps a narrow range of low input values into a wider range of output values
- The inverse log maps a wide range of input values into a narrow range of output values

Log Transformations

Input grey level values a has large range of values

Inverse log transformation maps a wide range of input values into a narrow range of output values

log transformation maps a narrow range of low input values into a wider range of output values

InvLog

Log

example, intensity change(3-bit image)

Image matrix is given by

$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

Use

1. Log Transformation (multiplier, c = 8)

example

3-bit Image matrix is given by

$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

1. Log of the image

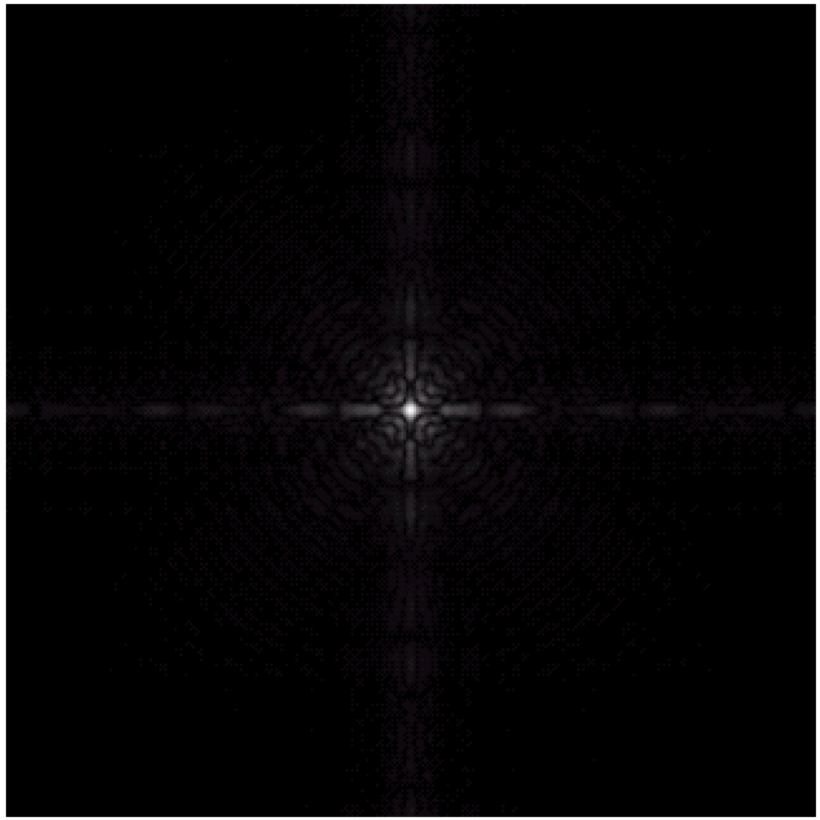
$$A2 = 8\log_{10}(1+A) = \begin{bmatrix} 3.81 & 4.81 & 0 & 6.76 & 7.22 \\ 0 & 4.81 & 7.22 & 6.22 & 3.81 \\ 6.22 & 4.81 & 3.81 & 5.59 & 0 \\ 5.59 & 3.81 & 3.81 & 2.40 & 0 \\ 2.40 & 7.22 & 6.76 & 5.59 & 6.22 \end{bmatrix}$$

$$A2\sim = \begin{bmatrix} 4 & 5 & 0 & 7 & 7 \\ 0 & 5 & 7 & 6 & 4 \\ 6 & 5 & 4 & 6 & 0 \\ 6 & 4 & 4 & 2 & 0 \\ 2 & 7 & 7 & 6 & 6 \end{bmatrix}$$

Log Transformations

a b

(a) Fourier spectrum.
(b) Result of applying the log transformation

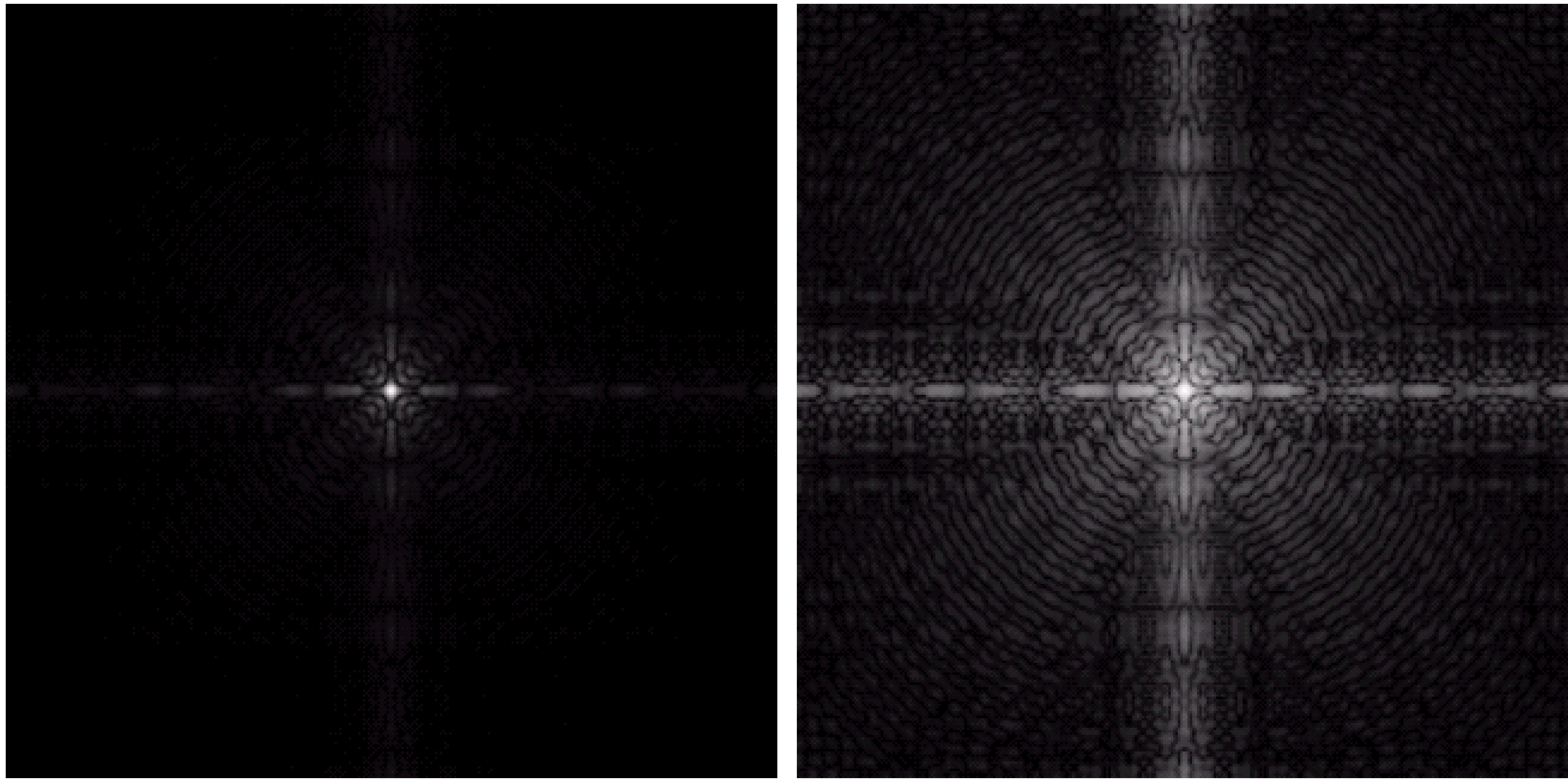


(a)

Log Transformations

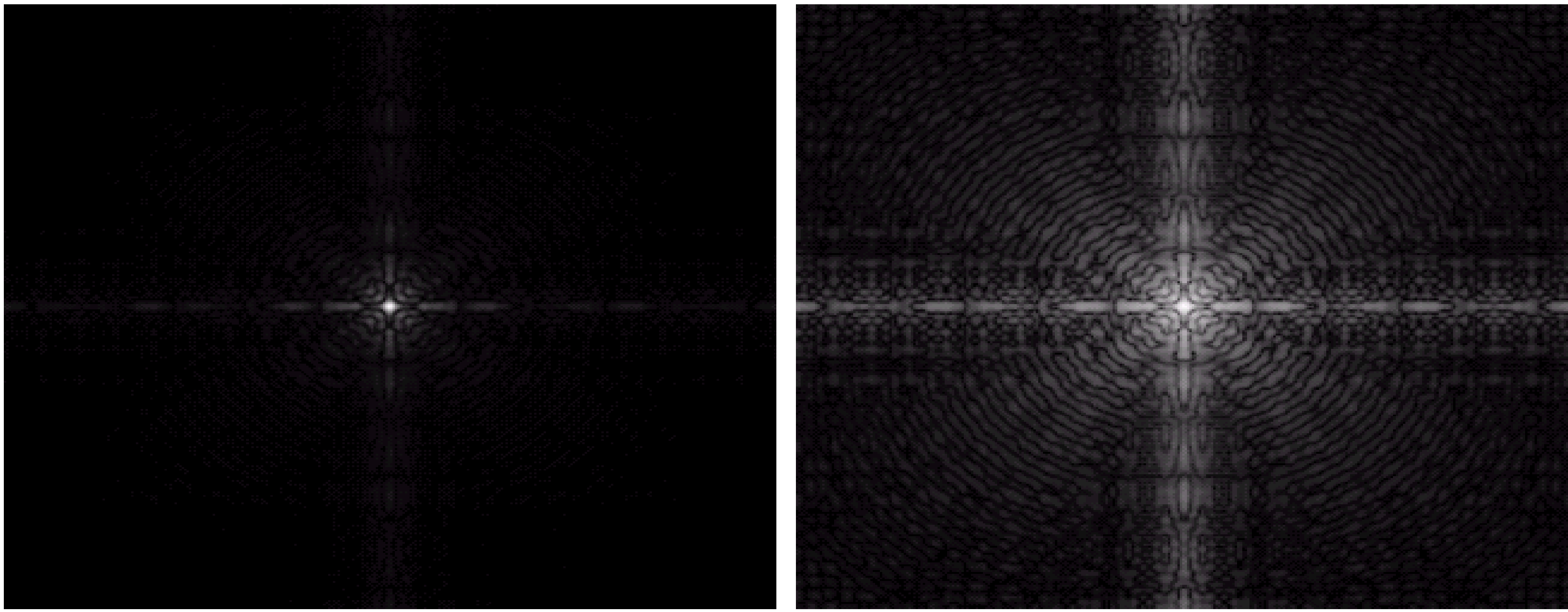
a b

(a) Fourier spectrum.
(b) Result of applying the log transformation



(a) (b)

Log Transformations



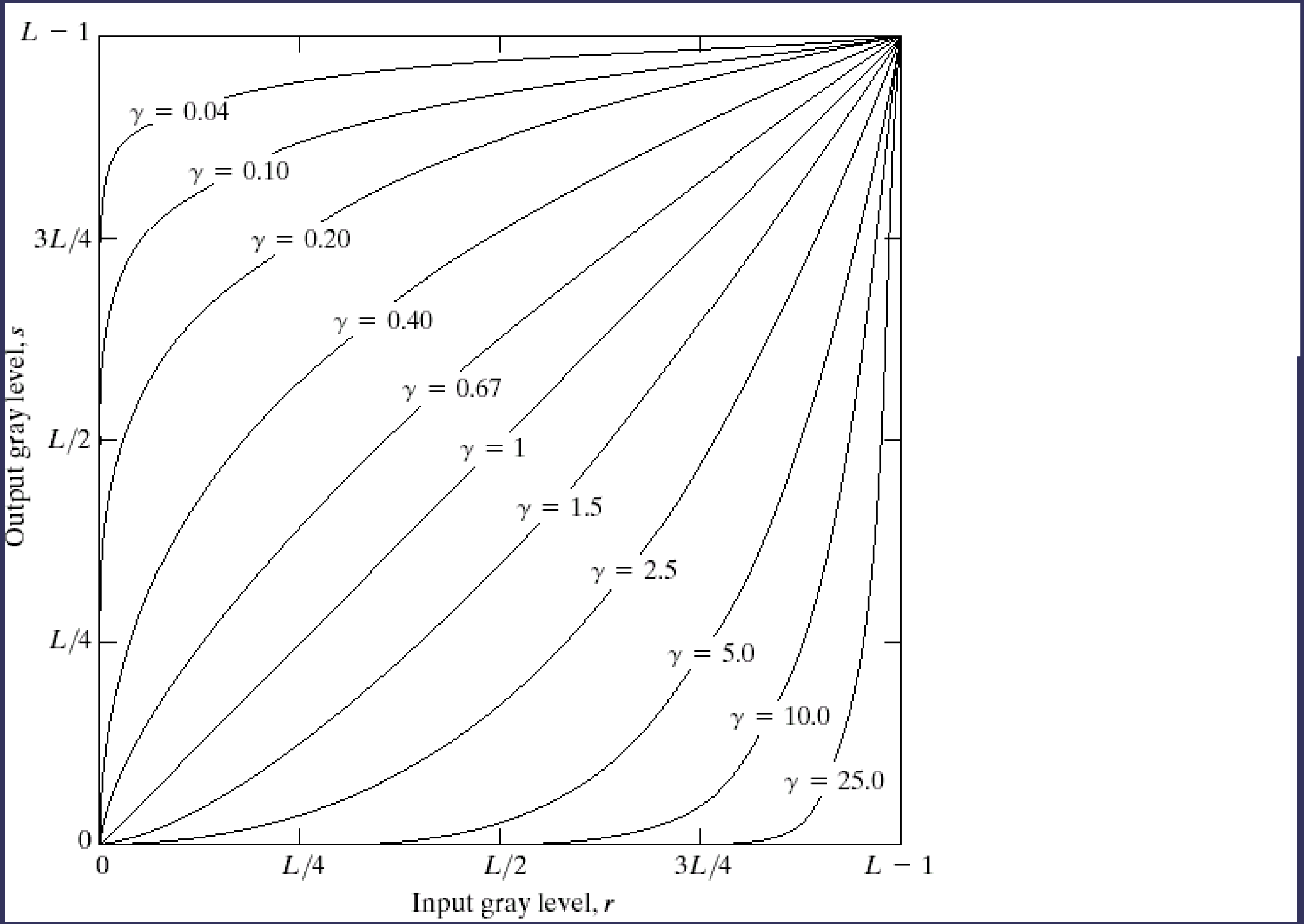
(a) (b)

- $s = c * \log(1+ r)$
- For $r = 0$, $s=0$
- For $r = 10^6$

  $s = \log_{10}(1+ 10^6) = 6$
- Range of 0 to 10^6 becomes 0 to 6.2 on log scale
- Therefore Logarithm of FT reveals more details

Power Law Transformations

$$s = T(r) = c \times r^{\gamma}$$



Power Law Transformations

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

- For γ < 1, map a narrow range of dark input values into a wider range of output values
- For γ > 1, map a narrow range of light input values into a wider range of output values
- Varying γ gives a family of curves

example, 3-bit image intensity change

Image matrix is given by

$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

Use

1. Power law Transformation using n^{th} root

Given multiplier, c=3 and root, n=2

examples

3-bit Image matrix is given by

$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

1. n=2 for n^{th} root of the image

$I = 3 \times A^{1/2} =$

4.24	5.19	0	7.34	7.93
0	5.19	7.93	6.70	4.24
6.7	5.19	4.24	6	0
6	4.24	4.24	3	0
3	7.93	7.34	6	6.7

$\approx$

4	5	0	7	8
0	5	8	7	4
7	5	4	6	0
6	4	4	3	0
3	8	7	6	7

Application of gamma-correction

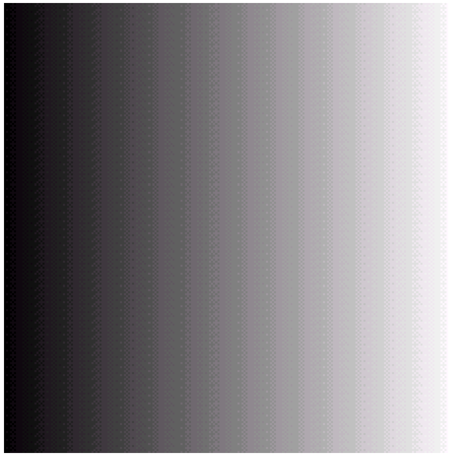
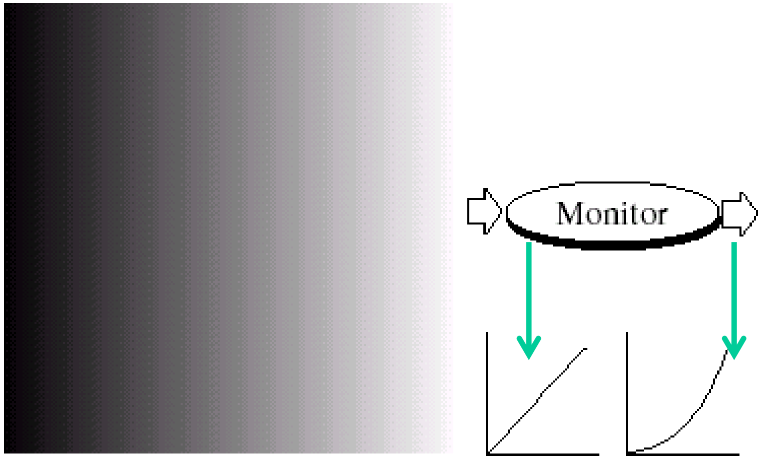


FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

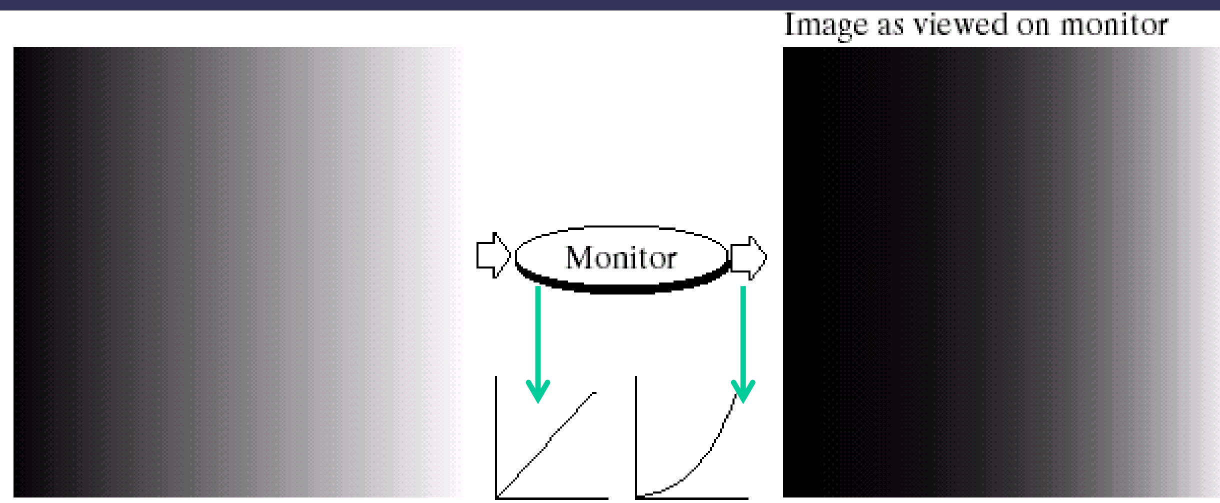
Application of gamma-correction

a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.



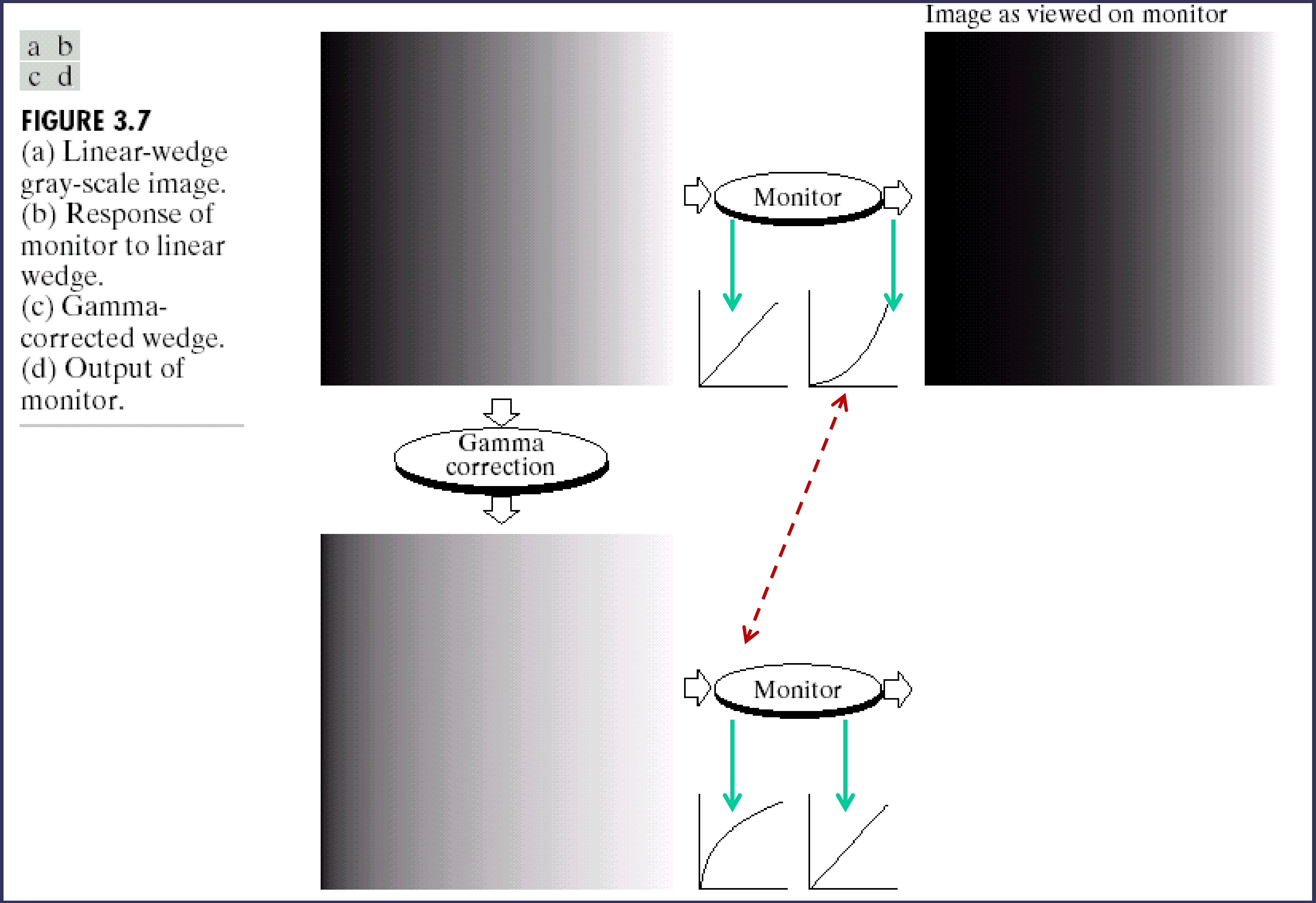
Application of gamma-correction



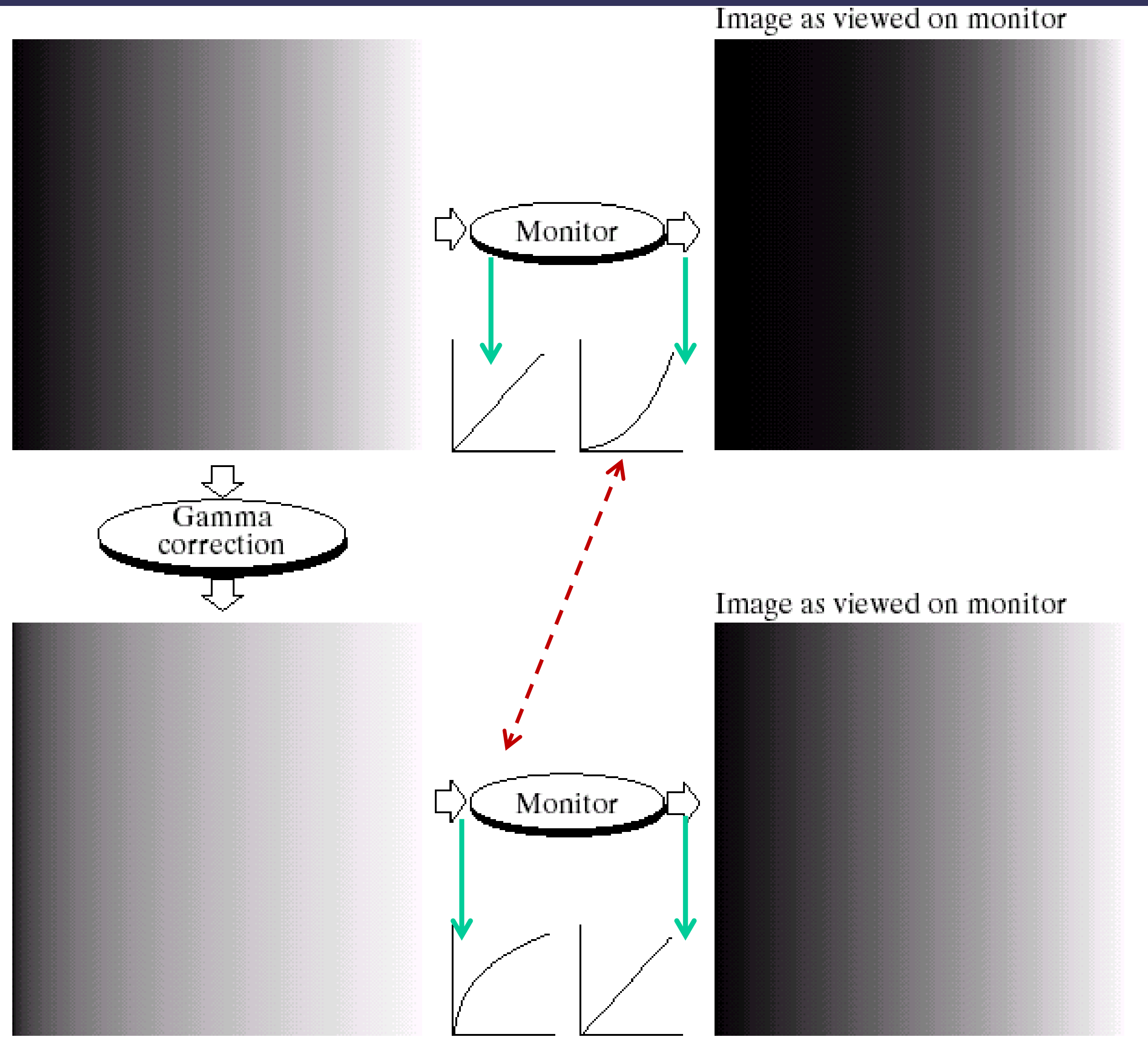
a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

Application of gamma-correction



Application of gamma-correction



a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

Power Law Example

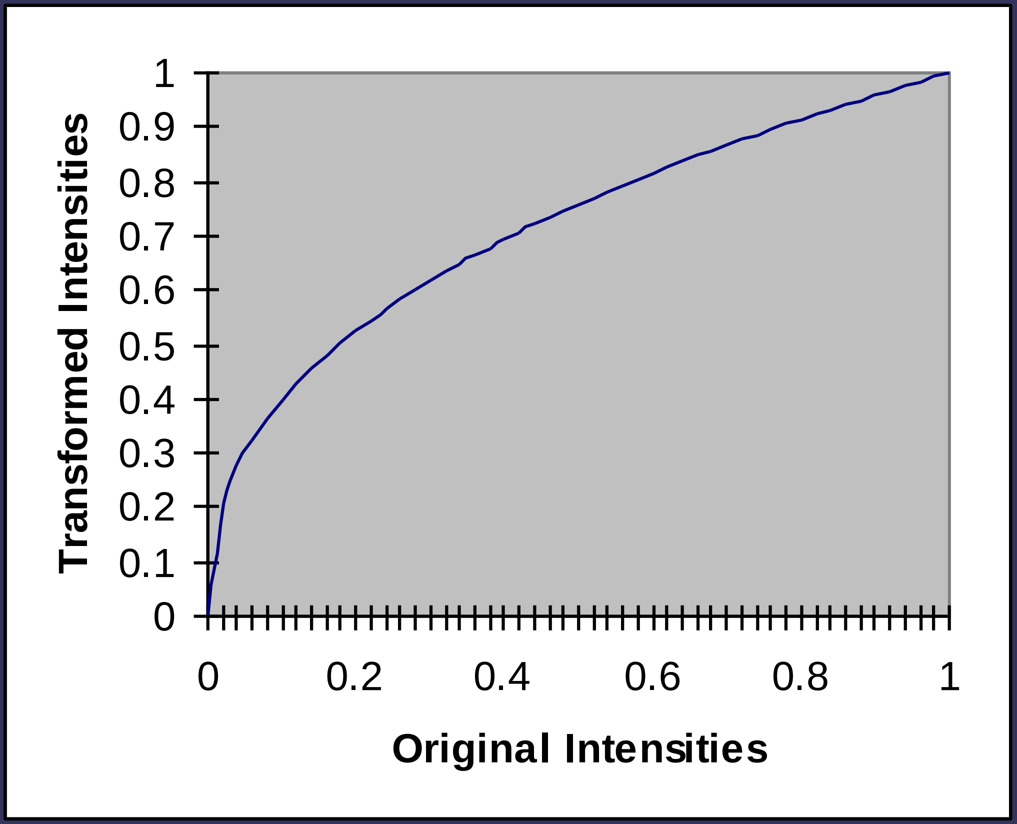
Magnetic Resonance (MR) image of a fractured human spine

Power Law Example

$\gamma = 0.6$

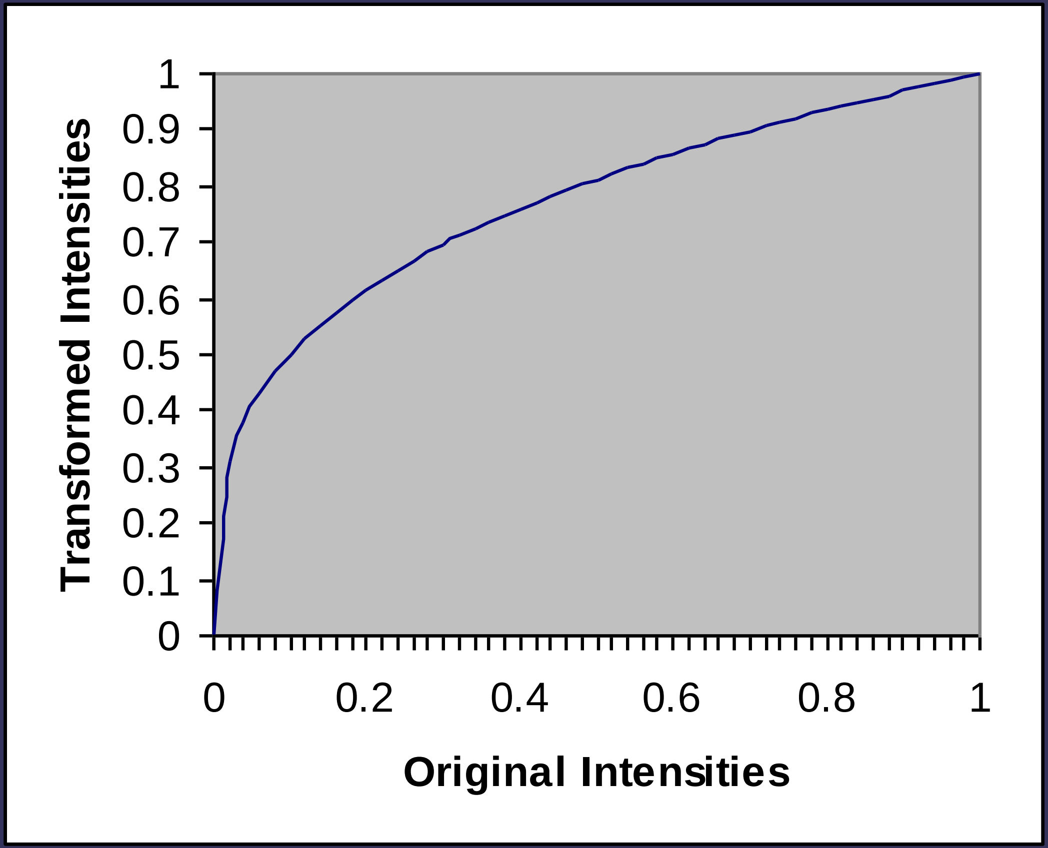
Power Law Example ($\gamma = 0.4$)

$\gamma = 0.4$



Power Law Example ($\gamma = 0.3$)

$\gamma = 0.3$



Power Law Example for power law

Power Law Example
(Image with washed out appearance)

An aerial view
of a runway

Image after gamma correction ($\gamma > 1$)

$\gamma = 5.0$

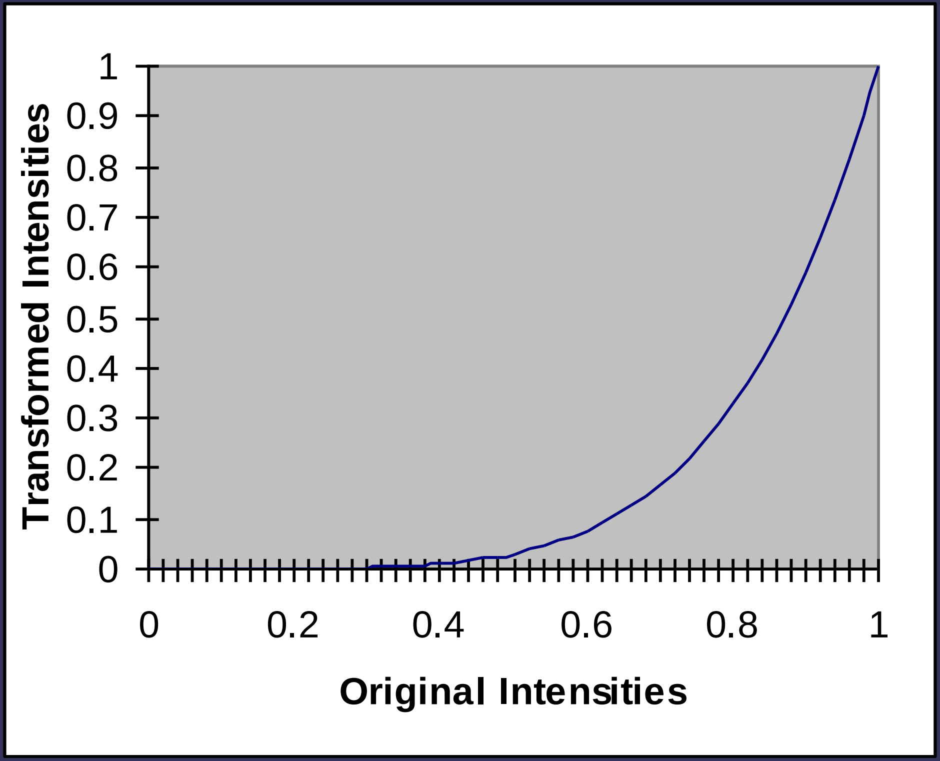


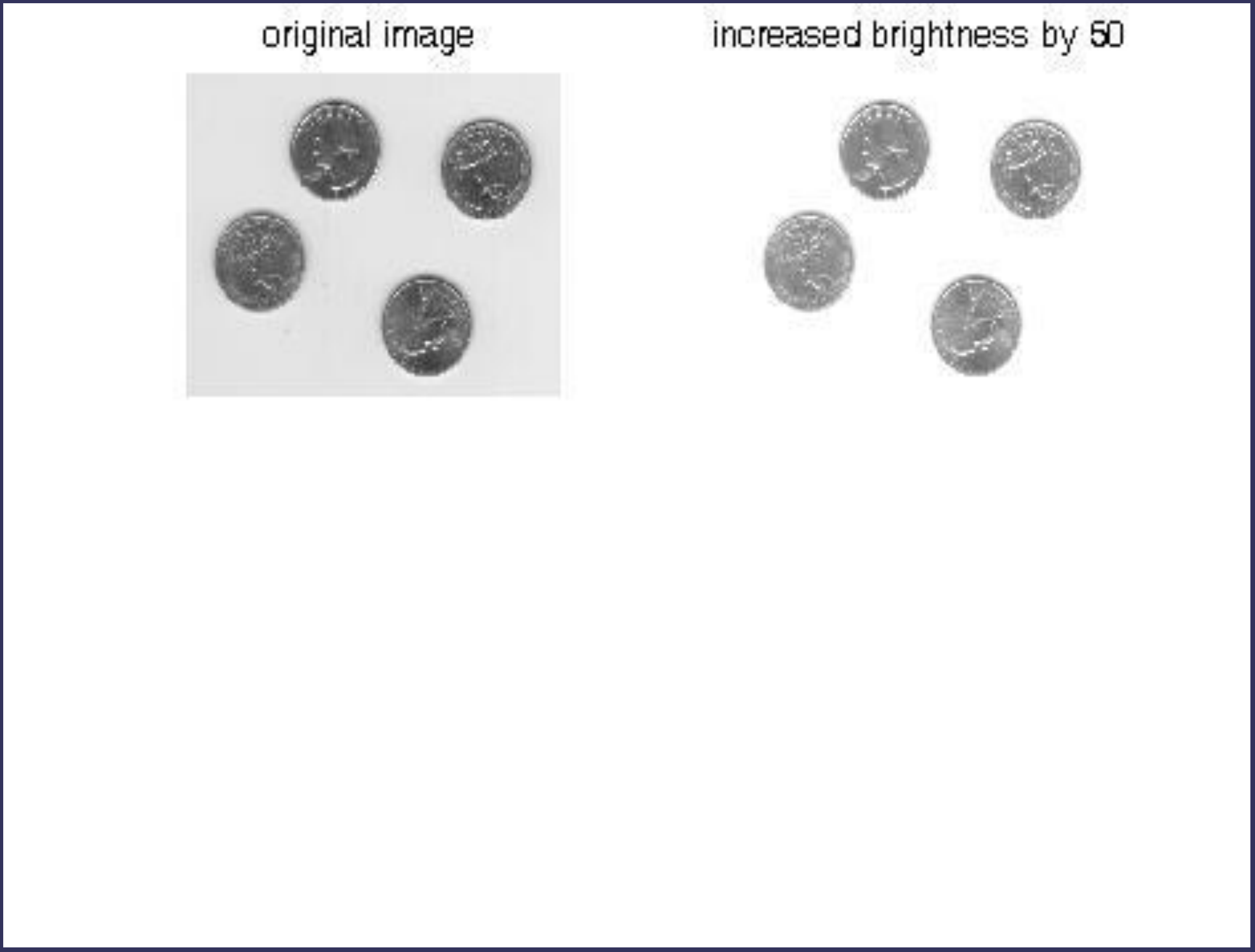
Image after gamma correction

Brightness/ contrast modification

- $g(m,n) = f(m,n) + k$ (increase brightness)
- $g(m,n) = f(m,n) - k$ (decrease brightness)
- $g(m,n) = k \times f(m,n)$

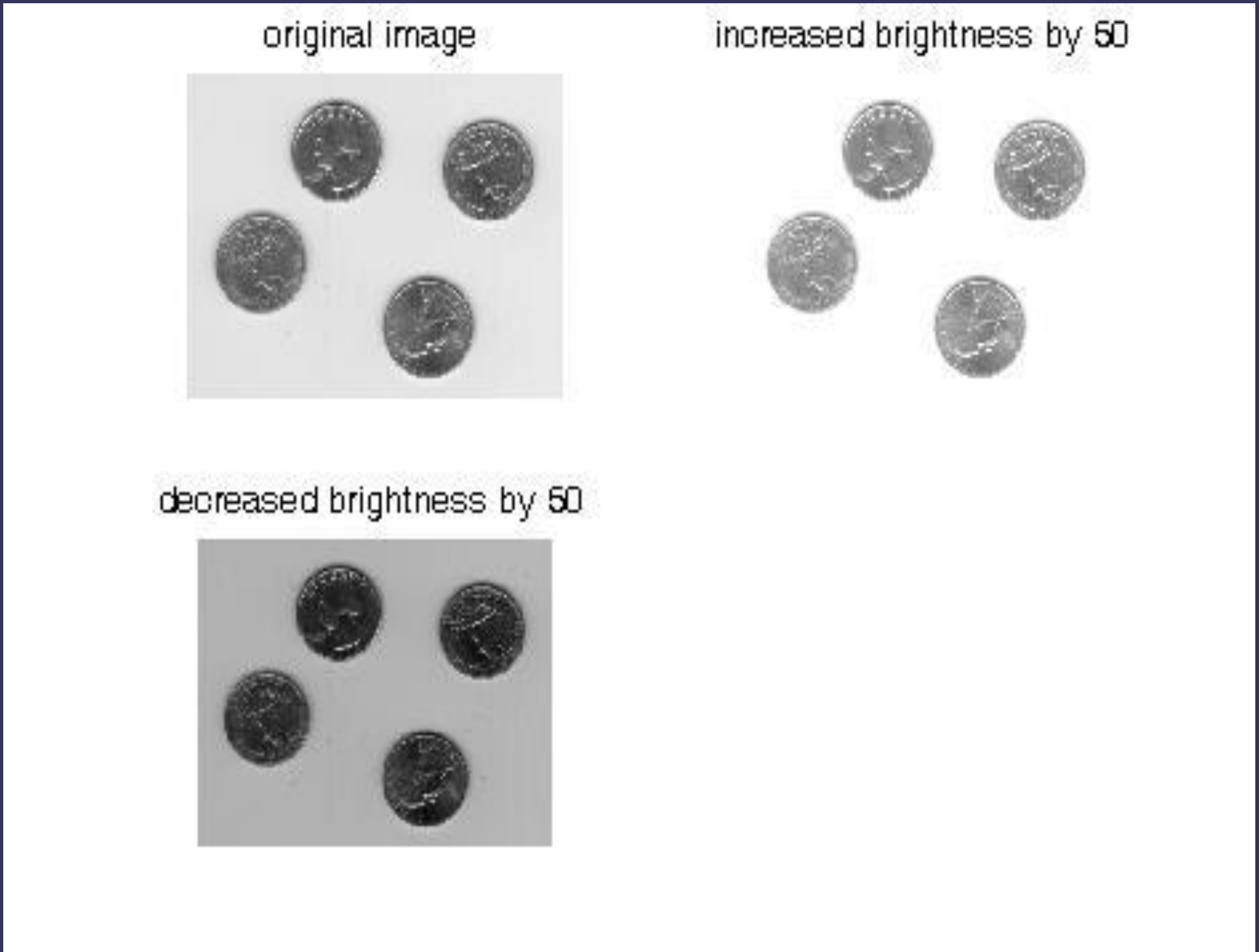
Brightness/contrast modification

- g(m,n) = f(m,n) + k (increase brightness)
- g(m,n) = f(m,n) – k (decrease brightness)
- g(m,n) = k× f(m,n)



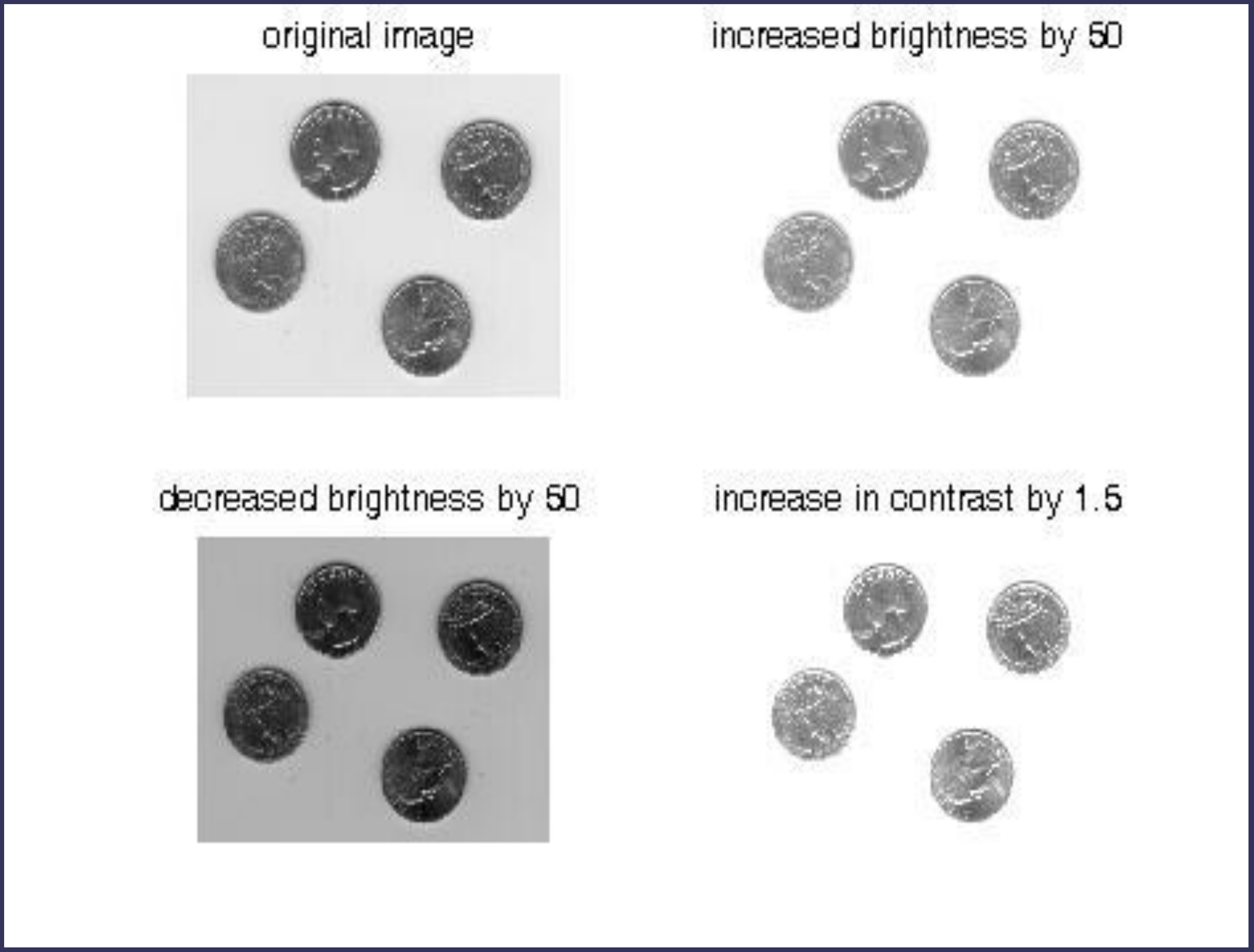
Brightness/contrast modification

- g(m,n) = f(m,n) + k (increase brightness)
- g(m,n) = f(m,n) – k (decrease brightness)
- g(m,n) = k× f(m,n)



Brightness/contrast modification

- $g(m,n) = f(m,n) + k$ (increase brightness)
- $g(m,n) = f(m,n) - k$ (decrease brightness)
- $g(m,n) = k\times f(m,n)$



Gray Level/Intensity Transformations

- Brightness modification
- Log transformations
- Power Law transformations
- Piecewise-Linear transformation Functions

Piecewise Linear Transformations

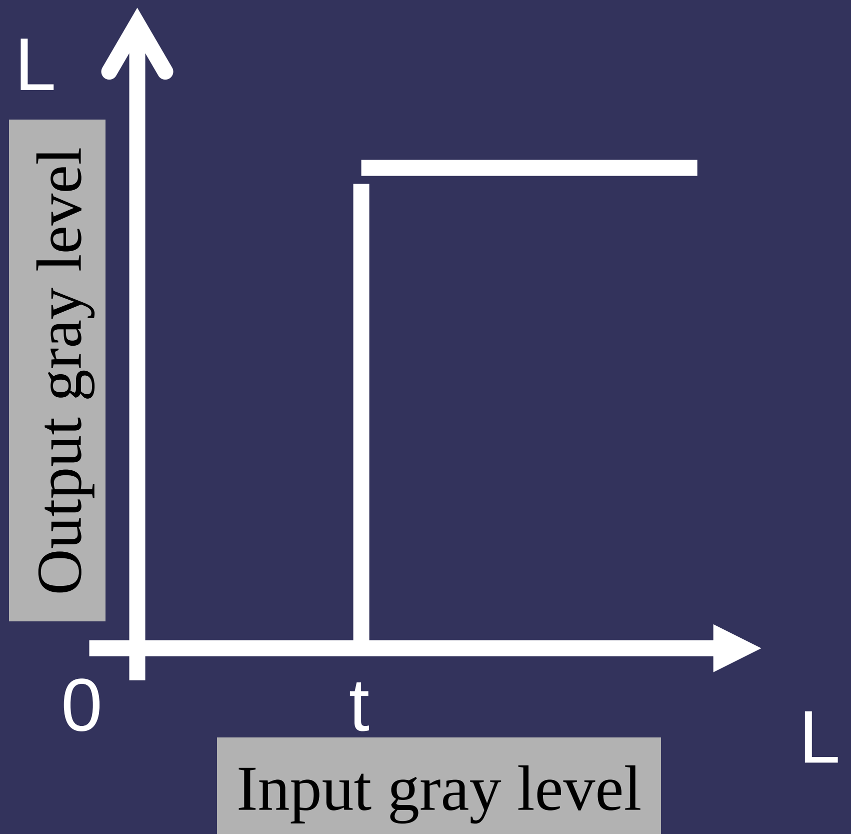
Thresholding Function

L=255 for 8-bit image

$g(x,y) = L, \quad \text{if } f(x,y) > t$

$= 0, \quad \text{if } f(x,y) < t$

t = 'threshold level'



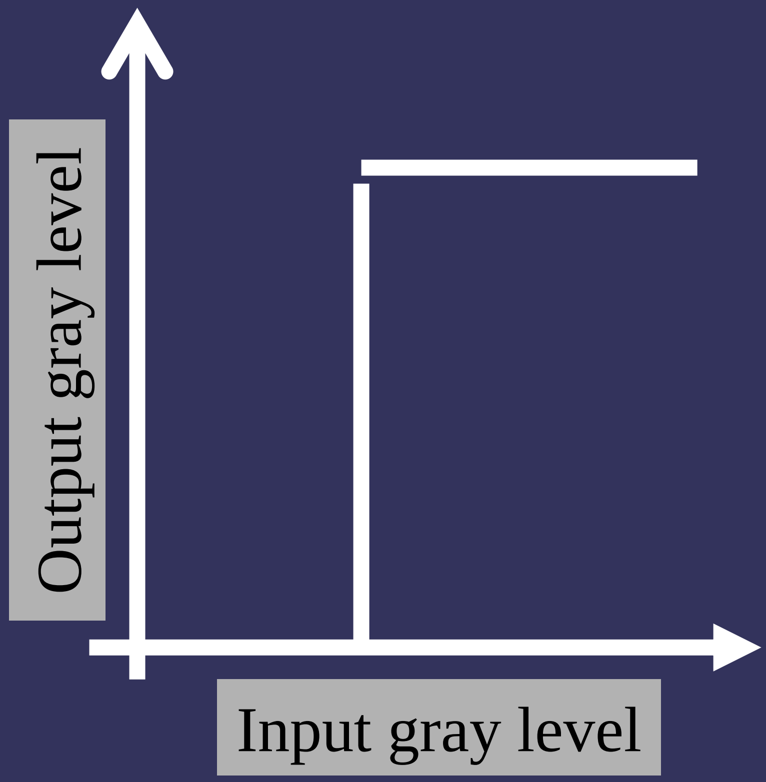
Piecewise Linear Transformations

Thresholding Function

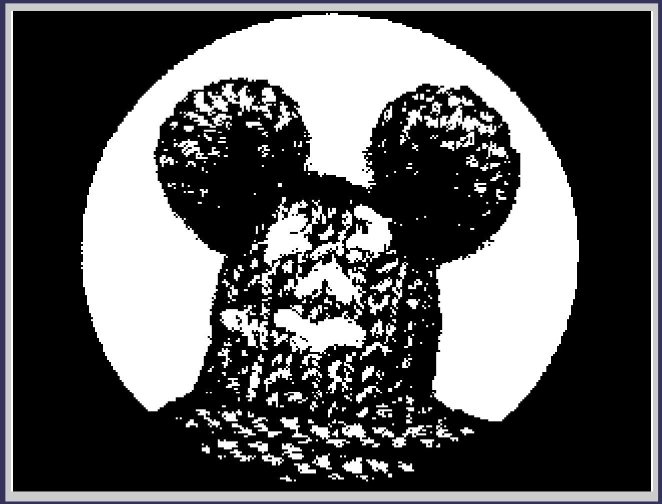
L=255 for 8-bit image

$$g(x,y) = \begin{cases} L, & f(x,y) > t \\ 0, & f(x,y) < t \end{cases}$$

t = ‘threshold level’



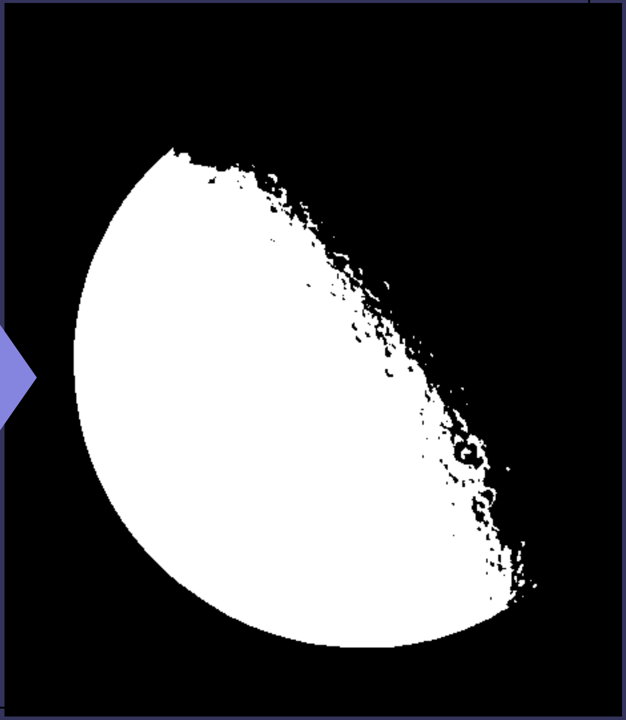
t=128



Thresholding

useful for segmentation in order to isolate an object of interest from a background

$$s = \begin{cases} 1.0 & r > \text{threshold} \\ 0.0 & r <= \text{threshold} \end{cases}$$



Examples, Thresholding for 3-bit image

Image matrix is given by

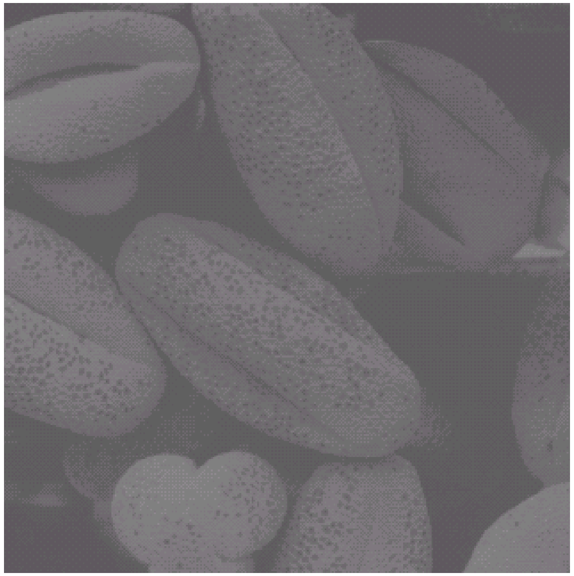
$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

1. Highlight intensity for r >30% of maximum
2. T=0.3*7 = 2.1 = 2 , r > 2, s = 7 else s = 0

$$A2 = \begin{bmatrix} 0 & 7 & 0 & 7 & 7 \\ 0 & 7 & 7 & 7 & 0 \\ 7 & 7 & 0 & 7 & 0 \\ 7 & 0 & 0 & 0 & 0 \\ 0 & 7 & 7 & 7 & 7 \end{bmatrix}$$

Contrast stretching to enhance beans

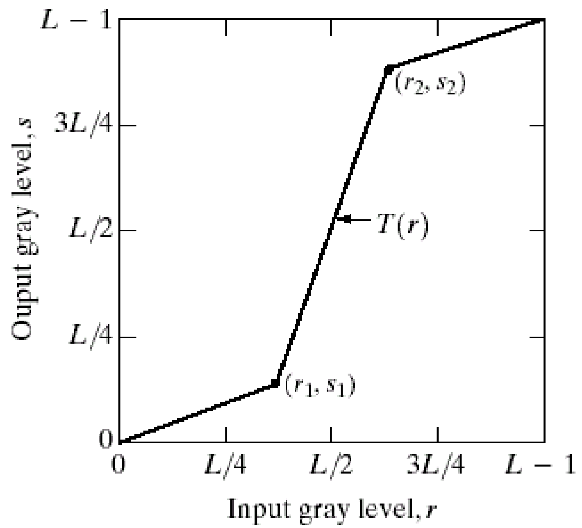
Low contrast image

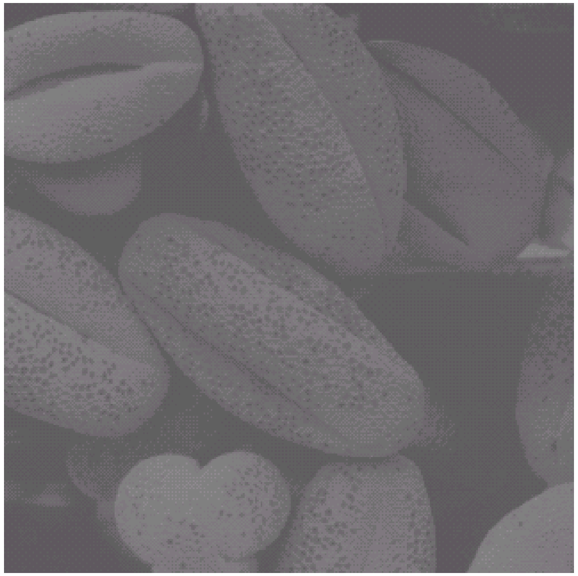


Contrast stretching to enhance beans

Low contrast image

Transformation function

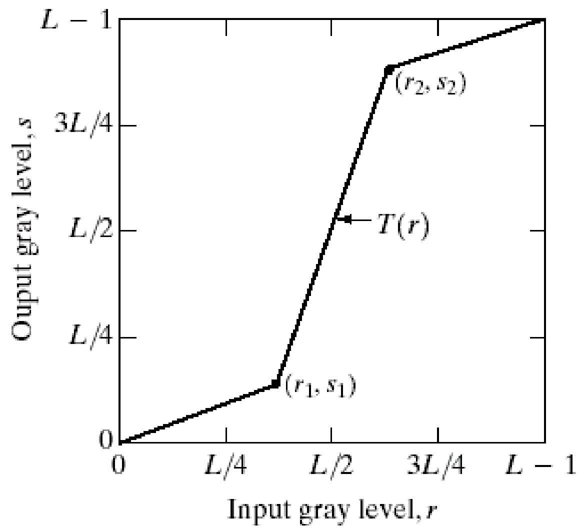


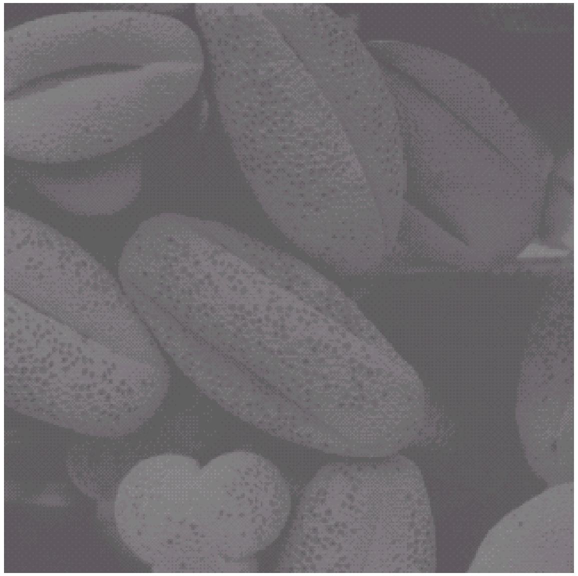


Contrast stretching to enhance beans

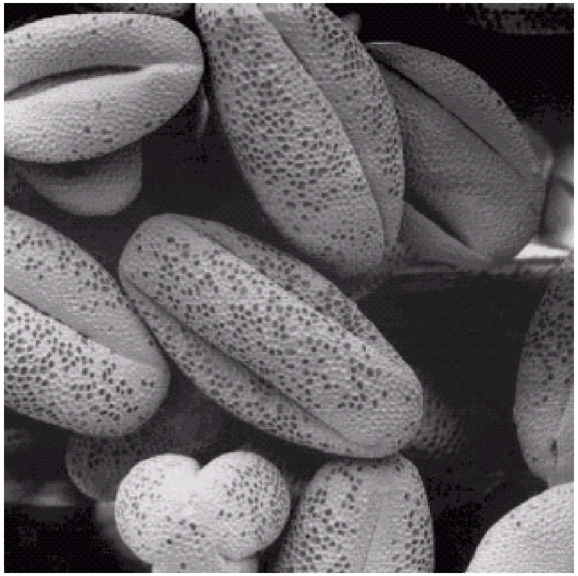
Low contrast image

Transformation function





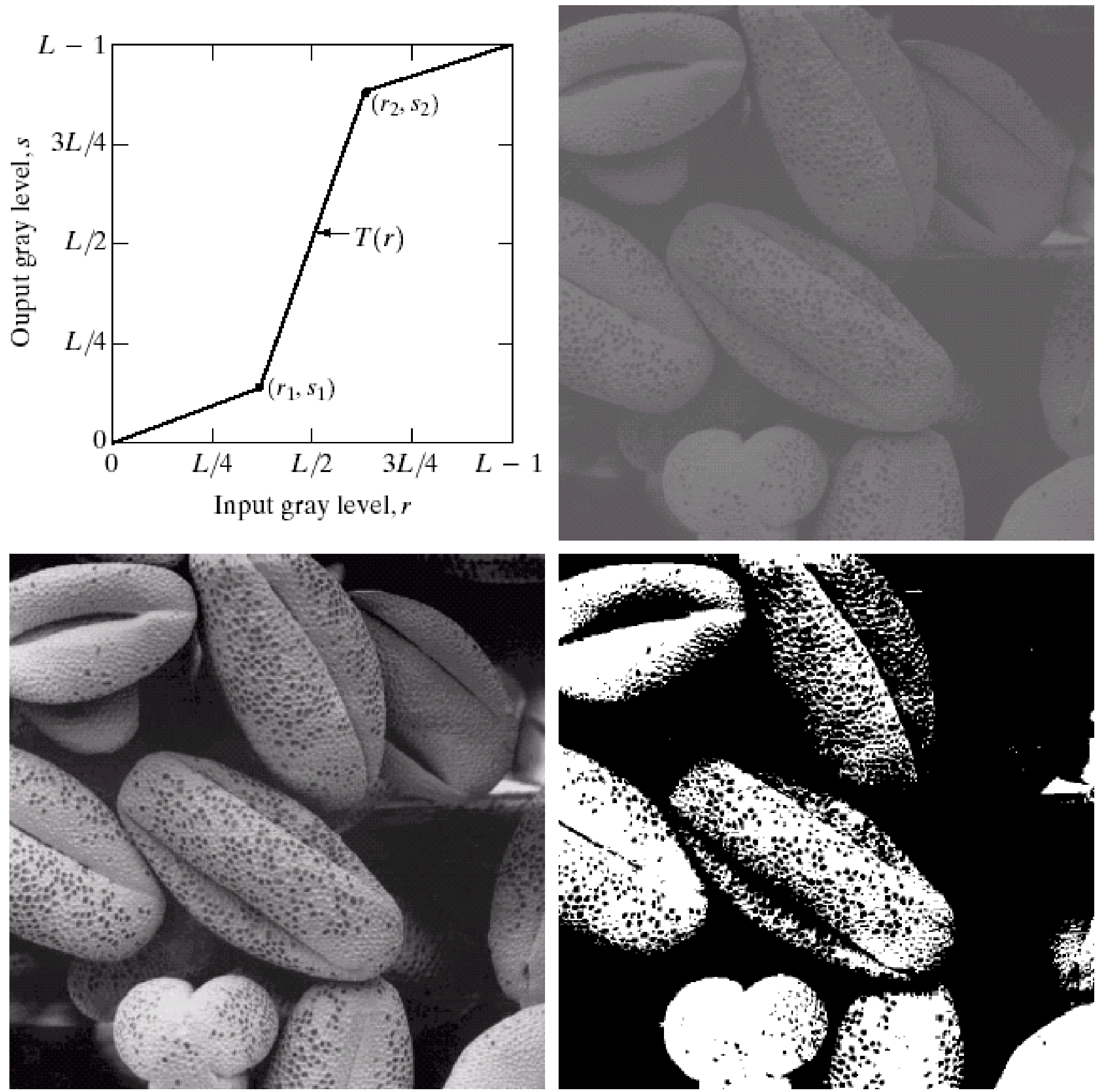
After contrast stretching



Contrast stretching to enhance beans

Low contrast image

Transformation function



After contrast stretching

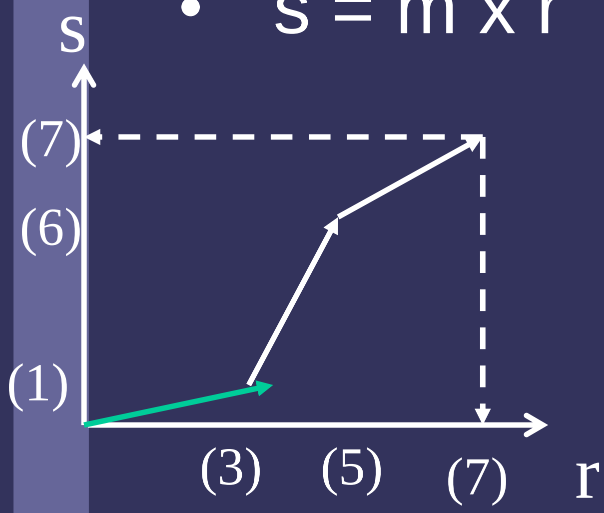
After thresholding

examples, contrast stretching

Image matrix is given by

$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

- Perform contrast stretching using two location points, (3,1) and (5,6)
- For the first segment, slope, m = (1-0)/(3-0)= 0.3
- s = m x r



r	**0**	**1**	**2**	**3**				
s	0	0	1	1				

examples, contrast stretching

Image matrix is given by

$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

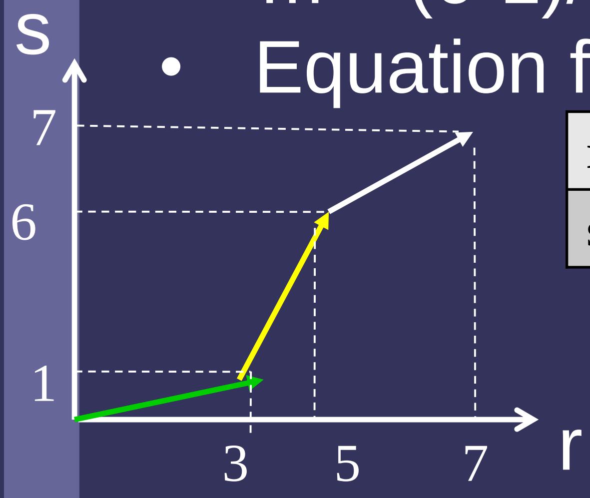
- Perform contrast stretching using two location points, (3,1) and (5,6)

examples, contrast stretching

Image matrix is given by

$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

- Perform contrast stretching using two location points, (3,1) and (5,6)
- For the middle segment, slope,
  m = (6-1)/(5-3)= 2.5
- Equation for middle segment, s - 1 = m (r - 3)



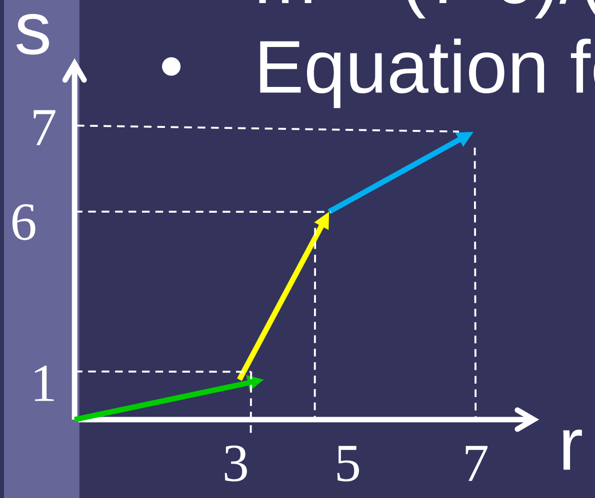
r	0	1	2	3	4	5	6	7
s	0	0	1	1	3	6		

examples, contrast stretching

3-bit Image matrix is given by

$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

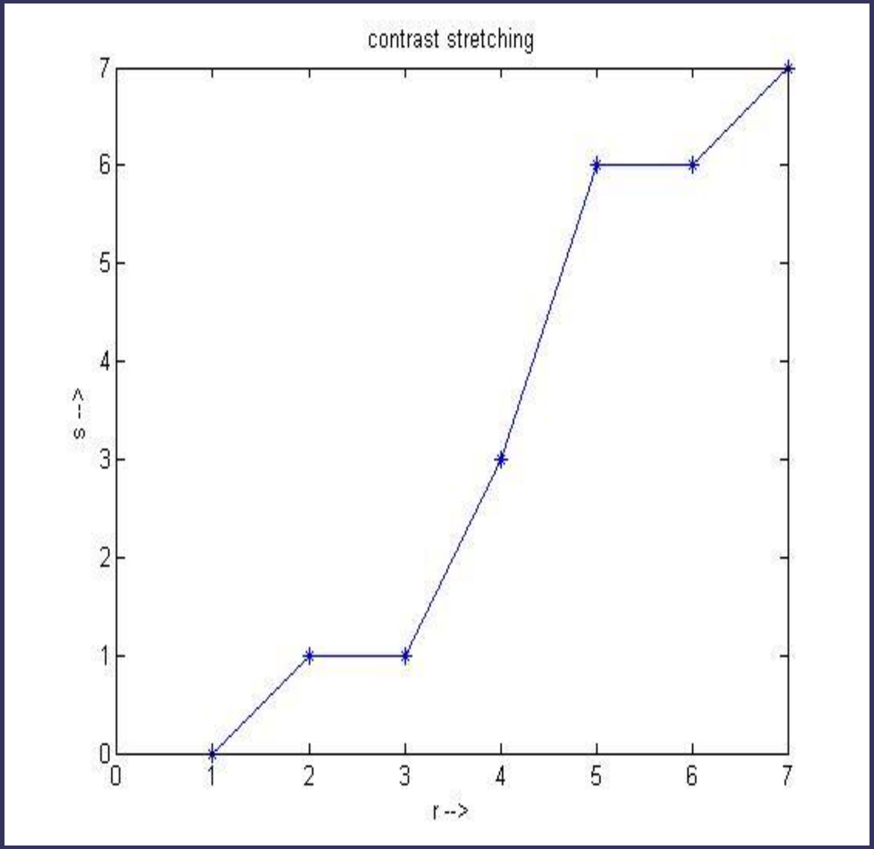
- Perform contrast stretching using two location points, (3,1) and (5,6)
- For the third segment, slope,
- m = (7-6)/(7-5)= 0.5
- Equation for the third segment, S - 6 = 0.5 (r - 5)



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

examples, contrast stretching

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



$$A = \begin{bmatrix} 2 & 3 & 0 & 6 & 7 \\ 0 & 3 & 7 & 5 & 2 \\ 5 & 3 & 2 & 4 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 1 & 7 & 6 & 4 & 5 \end{bmatrix}$$

$$B \text{ (enhanced A)} = \begin{bmatrix} 1 & 1 & 0 & 6 & 7 \\ 0 & 1 & 7 & 6 & 1 \\ 6 & 1 & 1 & 3 & 0 \\ 3 & 1 & 1 & 0 & 0 \\ 0 & 7 & 6 & 3 & 6 \end{bmatrix}$$

examples, contrast stretching

3-bit Image matrix is given by

$$A = \begin{bmatrix} 2 & 3 & 2 & 6 & 4 \\ 6 & 3 & 4 & 5 & 2 \\ 5 & 3 & 2 & 4 & 2 \\ 4 & 2 & 3 & 6 & 5 \\ 5 & 3 & 6 & 4 & 5 \end{bmatrix}$$

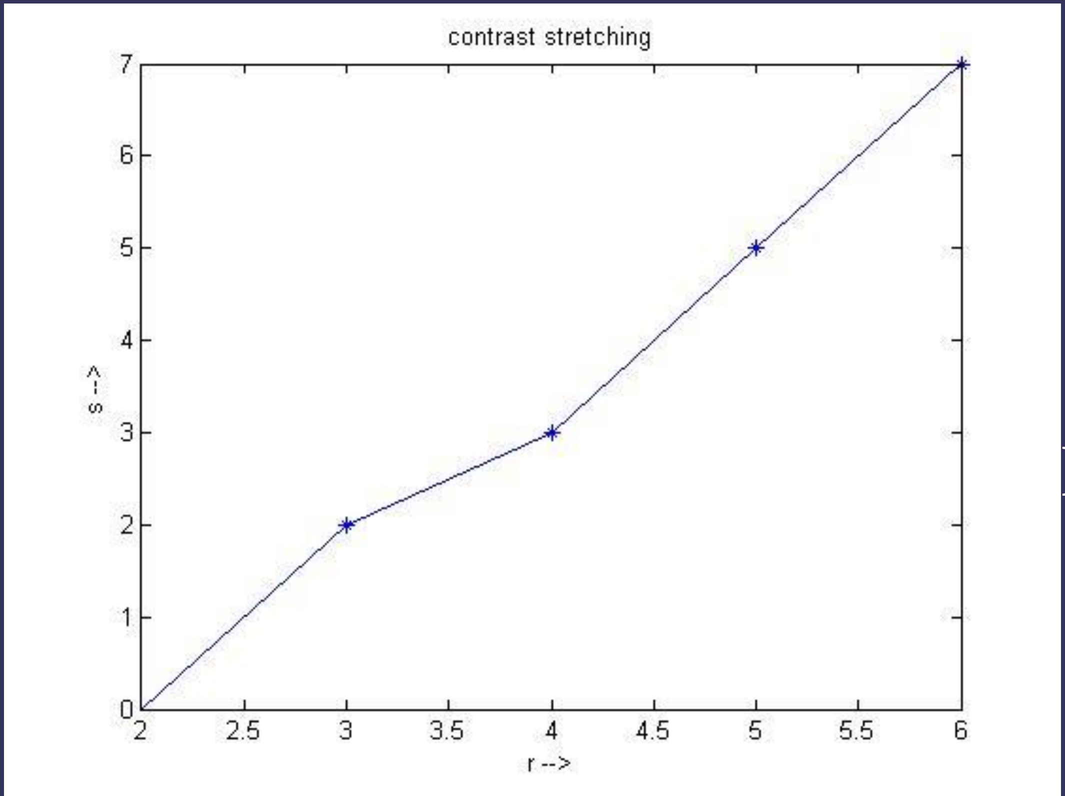
- Apply contrast stretching to cover the entire range of the given image.
- location points, $r_{min} = 2$, $r_{max} = 6$
- To stretch the image $r_{min} \rightarrow 0$ $r_{max} \rightarrow 7$
- Step size for the original image, st = (6-2)/7 = 0.57
- $s = (r - r_{min})/st = (r - 2)/0.57$

r	**2**	**3**	**4**	**5**	**6**
s	0	2	3	5	7

Some examples (4), contrast stretching

r	2	3	4	5	6
s	0	2	3	5	7

$$A = \begin{bmatrix} 2 & 3 & 2 & 6 & 4 \\ 6 & 3 & 4 & 5 & 2 \\ 5 & 3 & 2 & 4 & 2 \\ 4 & 2 & 3 & 6 & 5 \\ 5 & 3 & 6 & 4 & 5 \end{bmatrix}$$

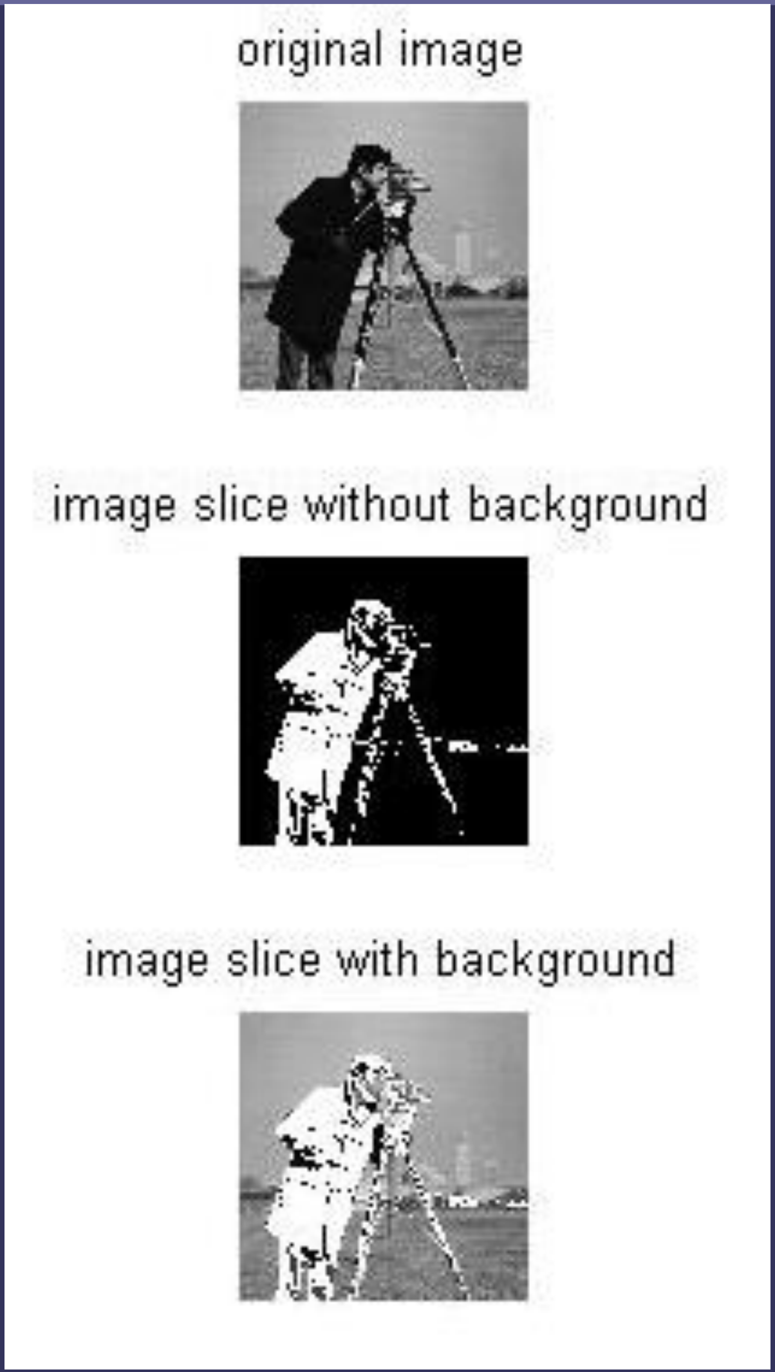


$$\text{Enhanced } A = \begin{bmatrix} 0 & 2 & 0 & 7 & 3 \\ 7 & 2 & 3 & 5 & 0 \\ 5 & 2 & 0 & 3 & 0 \\ 3 & 0 & 2 & 7 & 5 \\ 5 & 2 & 7 & 3 & 5 \end{bmatrix}$$

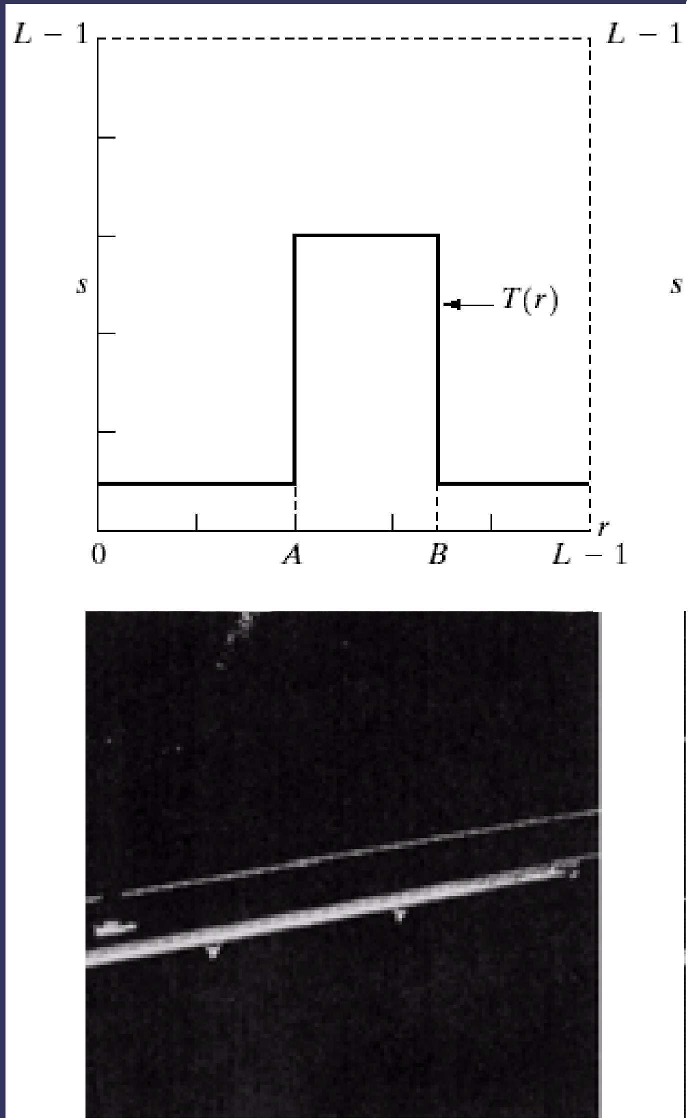
Gray/Intensity Level Slicing

- Highlight a specific range of gray values
- Two approaches:
  - Display high value for range of interest and discard background
  - Display high value for range of interest, and preserve background

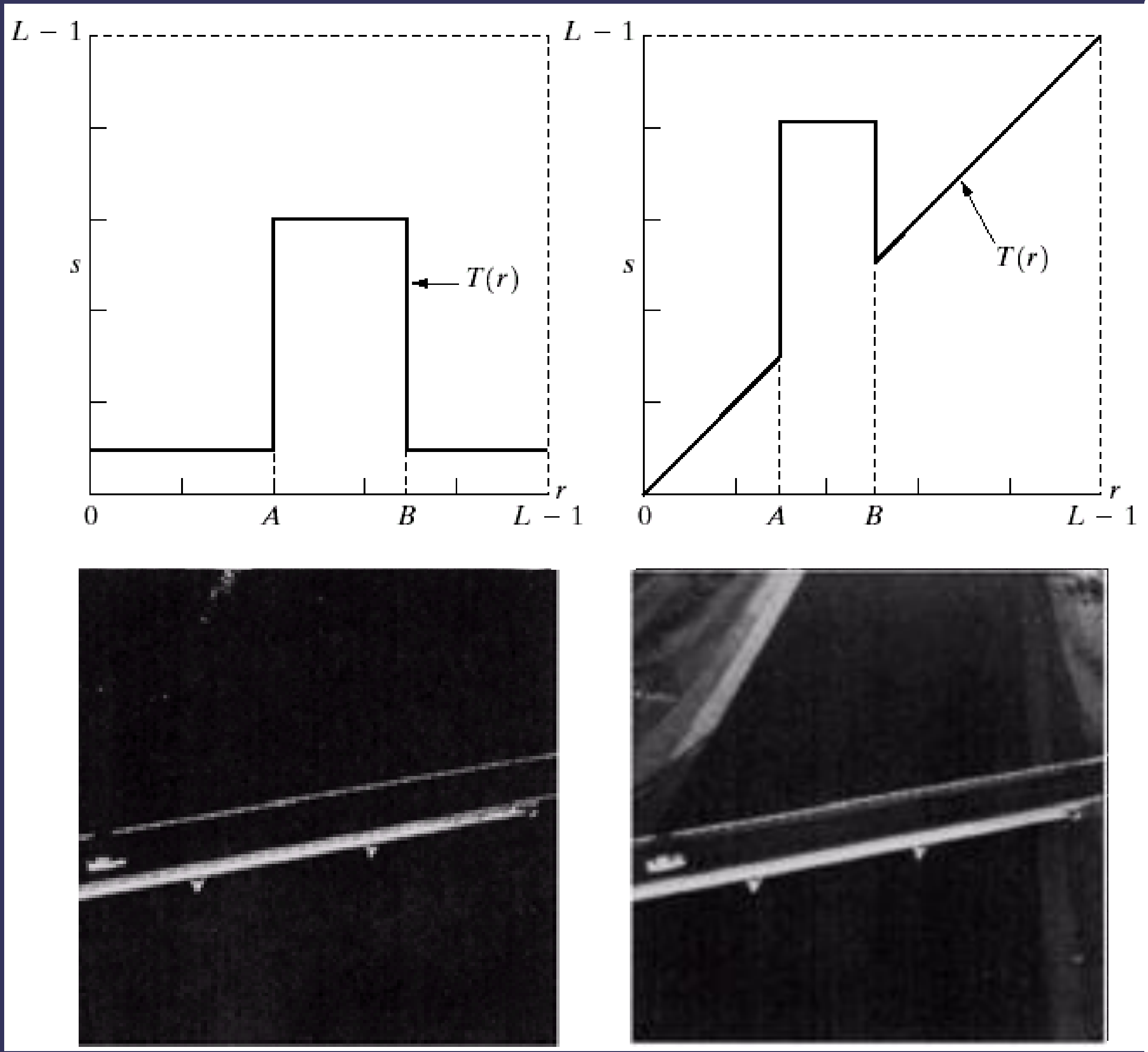
Gray/Intensity Level Slicing



Gray Level Slicing, example



Gray Level Slicing, example



example, intensity level slicing

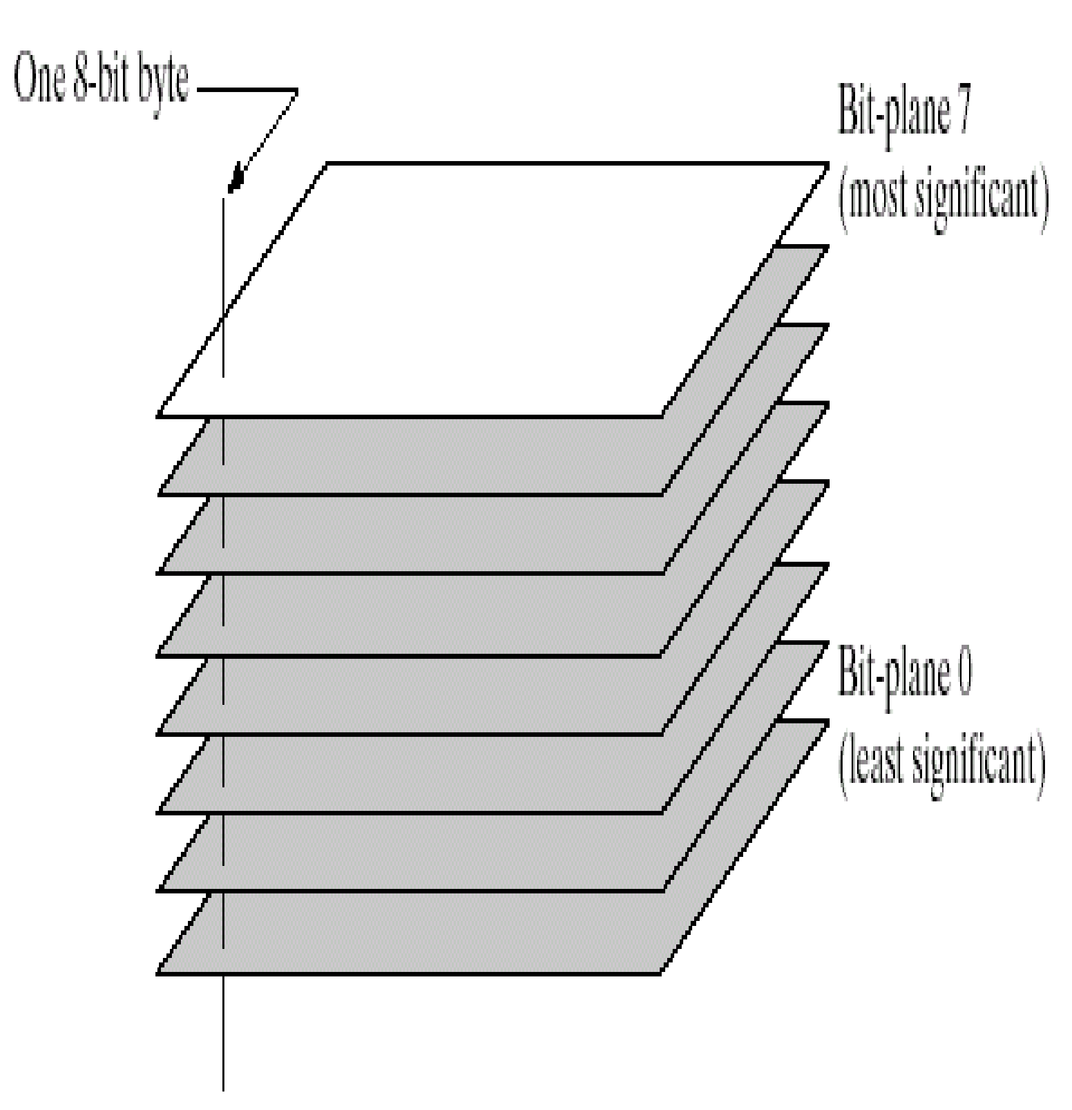
$$\text{3-bit Image matrix, } f(x,y) = A = \begin{bmatrix} 0 & 3 & 2 & 6 & 4 \\ 6 & 3 & 4 & 5 & 2 \\ 5 & 3 & 2 & 1 & 2 \\ 4 & 2 & 3 & 6 & 5 \\ 5 & 3 & 6 & 4 & 5 \end{bmatrix}$$

- Highlight pixel with intensity in the range 40-70 % of max possible intensity and keep other pixels unchanged
- Range is $.4\times7 \approx 3$ to $0.7\times7 \approx 5$
- $g(x,y) = 7, \; 3 <= f(x,y) < 5$

  $= f(x,y), \;$ otherwise

$$G(x,y) = B = \begin{bmatrix} 0 & 7 & 2 & 6 & 7 \\ 6 & 7 & 7 & 5 & 2 \\ 5 & 7 & 2 & 1 & 2 \\ 7 & 2 & 7 & 6 & 5 \\ 5 & 7 & 6 & 7 & 5 \end{bmatrix}$$

Bit Plane Slicing



example, bit plane slicing (3-bit image)

$$\text{Image matrix, } A = \begin{bmatrix} 0 & 3 & 2 & 6 & 4 \\ 6 & 3 & 4 & 5 & 2 \\ 5 & 3 & 2 & 1 & 2 \\ 4 & 2 & 3 & 6 & 5 \\ 5 & 3 & 6 & 4 & 5 \end{bmatrix}$$

- Determine bit planes

$$A = \begin{bmatrix} 000 & 011 & 010 & 110 & 100 \\ 110 & 011 & 100 & 101 & 010 \\ 101 & 011 & 010 & 001 & 010 \\ 100 & 010 & 011 & 110 & 101 \\ 101 & 011 & 110 & 100 & 101 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

example, bit plane slicing

Image matrix, $A = \begin{bmatrix} 0 & 3 & 2 & 6 & 4 \\ 6 & 3 & 4 & 5 & 2 \\ 5 & 3 & 2 & 1 & 2 \\ 4 & 2 & 3 & 6 & 5 \\ 5 & 3 & 6 & 4 & 5 \end{bmatrix}$ $B = \begin{bmatrix} 000 & 011 & 010 & 110 & 100 \\ 110 & 011 & 100 & 101 & 010 \\ 101 & 011 & 010 & 001 & 010 \\ 100 & 010 & 011 & 110 & 101 \\ 101 & 011 & 110 & 100 & 101 \end{bmatrix}$

- Compute rms error if two planes (B_2 and B_1) are retained.

$B_1 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$ $B_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$

For $B_{2,1} \begin{bmatrix} 000 & 010 & 010 & 110 & 100 \\ 110 & 010 & 100 & 100 & 010 \\ 100 & 010 & 010 & 000 & 010 \\ 100 & 010 & 010 & 110 & 100 \\ 100 & 010 & 110 & 100 & 100 \end{bmatrix} \approx \begin{bmatrix} 0 & 2 & 2 & 6 & 4 \\ 6 & 2 & 4 & 4 & 2 \\ 4 & 2 & 2 & 0 & 2 \\ 4 & 2 & 2 & 6 & 4 \\ 4 & 2 & 6 & 4 & 4 \end{bmatrix}$

examples, bit plane slicing

$$f(x,y) = A = \begin{bmatrix} 0 & 3 & 2 & 6 & 4 \\ 6 & 3 & 4 & 5 & 2 \\ 5 & 3 & 2 & 1 & 2 \\ 4 & 2 & 3 & 6 & 5 \\ 5 & 3 & 6 & 4 & 5 \end{bmatrix} \qquad g(x,y) = B_{2,1} \approx \begin{bmatrix} 0 & 2 & 2 & 6 & 4 \\ 6 & 2 & 4 & 4 & 2 \\ 4 & 2 & 2 & 0 & 2 \\ 4 & 2 & 2 & 6 & 4 \\ 4 & 2 & 6 & 4 & 4 \end{bmatrix}$$

$$\text{Error, } g(x,y)-f(x,y) = |A - B_{21}| = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}, M=5 \text{ and } N=5$$

examples, bit plane slicing

$$f(x,y) = A = \begin{bmatrix} 0 & 3 & 2 & 6 & 4 \\ 6 & 3 & 4 & 5 & 2 \\ 5 & 3 & 2 & 1 & 2 \\ 4 & 2 & 3 & 6 & 5 \\ 5 & 3 & 6 & 4 & 5 \end{bmatrix} \qquad g(x,y) = B_{2,1} \approx \begin{bmatrix} 0 & 2 & 2 & 6 & 4 \\ 6 & 2 & 4 & 4 & 2 \\ 4 & 2 & 2 & 0 & 2 \\ 4 & 2 & 2 & 6 & 4 \\ 4 & 2 & 6 & 4 & 4 \end{bmatrix}$$

$$\text{Error, } g(x,y)-f(x,y)=s-r = |A - B_{21}| = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}, M=5 \text{ and } N=5$$

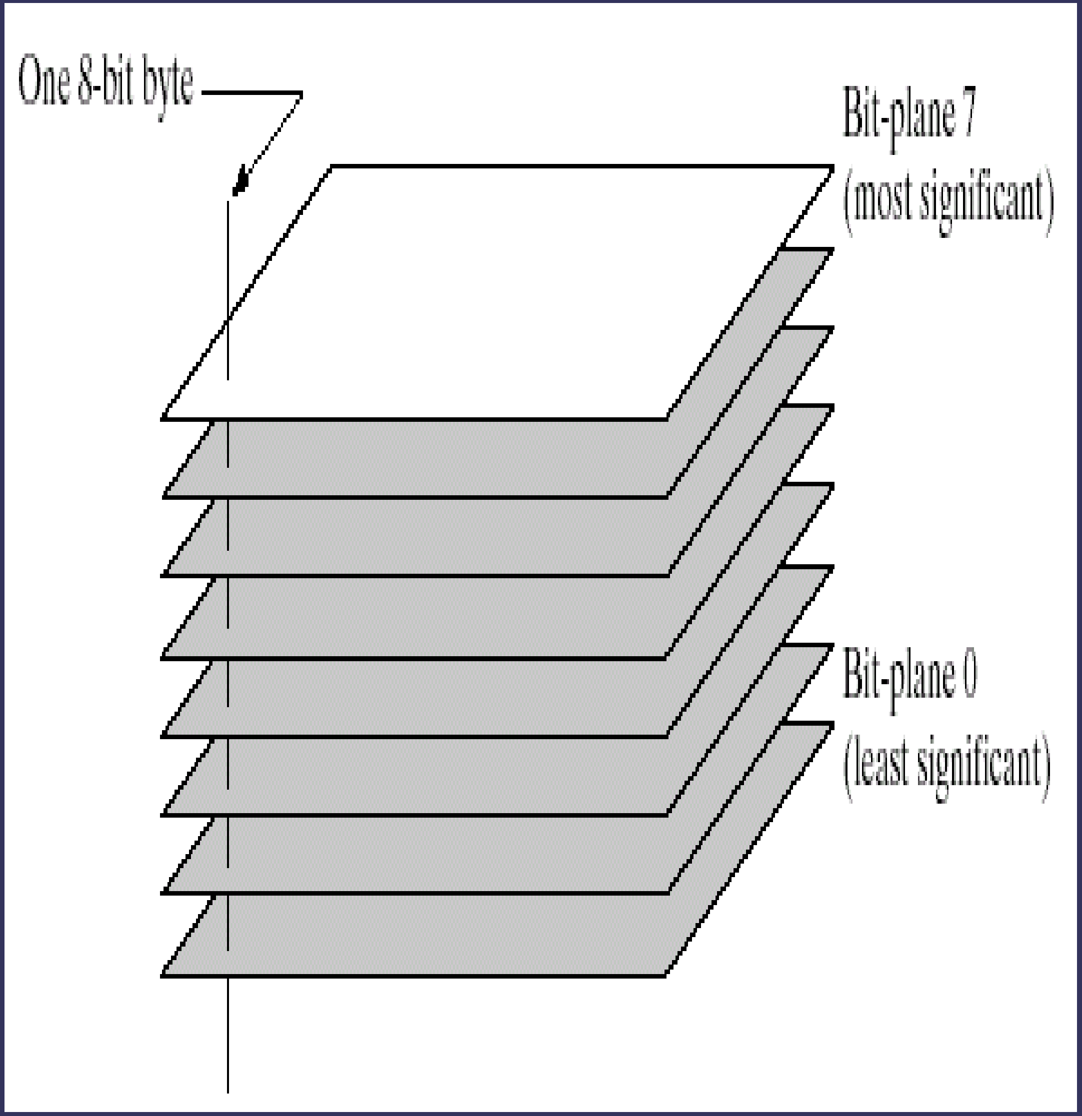
$$rms\ error = \sqrt{\sum_{i=0}^{M\times N} (s_i - r_i)^2} \ /(M \times N)$$

$$\text{Error} = (1+1+1+1+1+1+1+1+1+1+1)^{1/2}/25$$

$$= 0.13$$

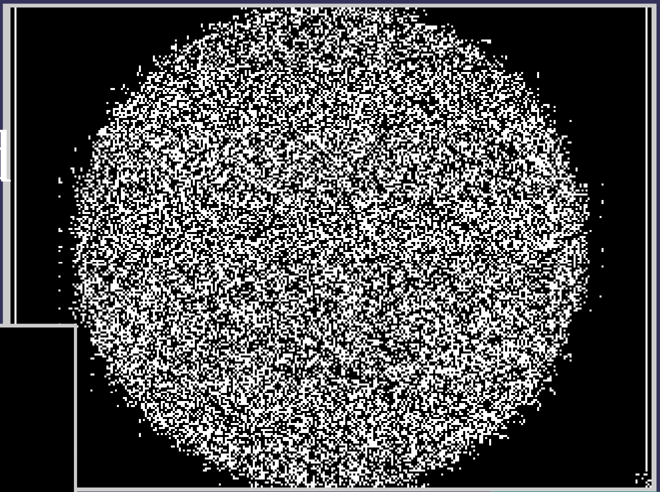
Bit Plane Slicing

- Isolate each bit of pixel intensity
- Higher-order bits usually contain most of the significant visual information
- Lower-order bits contain subtle details



Pixel intensity = (b_7 b_6 b_5 b_4 b_3 b_2 $b_1 b_0$)

b_0 of each pixel is considered and other bits are 0



BP 0

b_5 is considered and other bits are 0

BP 5

b_7 is considered and other bits are 0

BP 7

Bit Plane Slicing (all 8 bits are considered)

Pixel intensity = (b_7 b_6 b_5 b_4 b_3 b_2 b_1b_0)

Bit Plane Slicing, (plane 0)

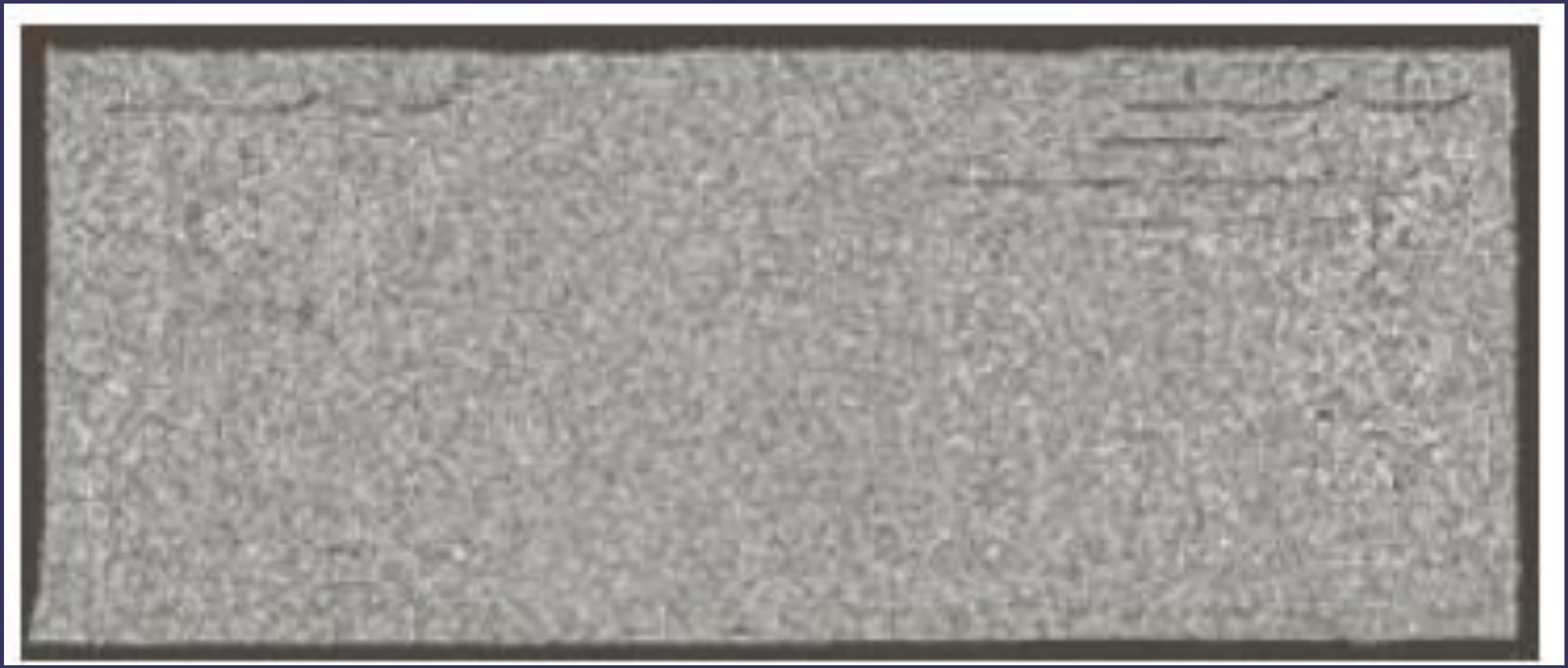
b_0 of each pixel is considered and other bits are 0

Bit Plane Slicing (plane 1)

b_1 of each pixel is considered and other bits are 0

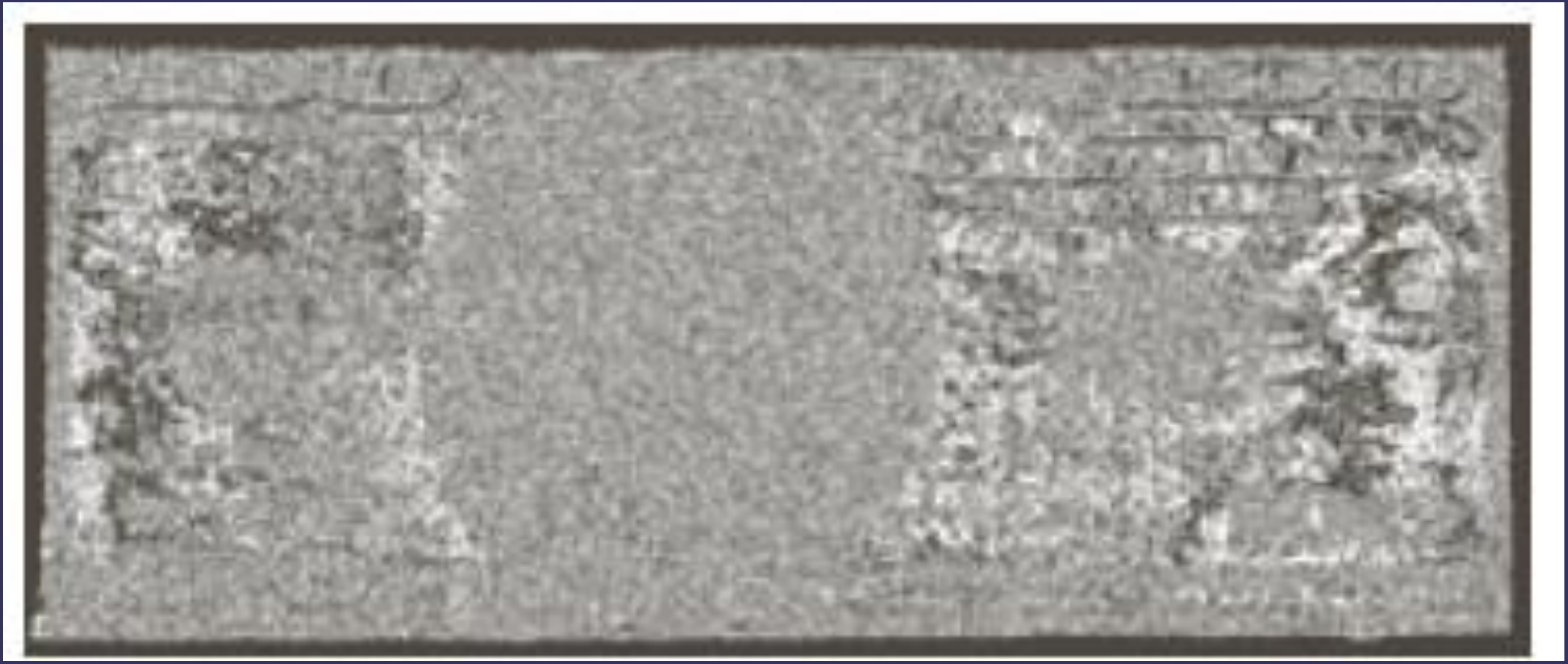
Bit Plane Slicing (plane 2)

b_2 of each pixel is considered and other bits are 0



Bit Plane Slicing (plane 3)

B_3 of each pixel is considered and other bits are 0



Bit Plane Slicing (plane 4)

B_4 of each pixel is considered and other bits are 0

Bit Plane Slicing (plane 5)

B_5 of each pixel is considered and other bits are 0

Bit Plane Slicing (plane 6)

B_6 of each pixel is considered and other bits are 0

Bit Plane Slicing (plane 7)

B_7 of each pixel is considered and other bits are 0

Bit Plane Slicing

Reconstructed image using bit planes 7 and 6

Reconstructed image using bit planes 7,6 and 5

Reconstructed image using bit planes 7, 6, 5 and 4

Histogram

Plot of number of occurrences of grey levels against each grey level value for the given image matrix

$$A=\begin{bmatrix} 1 & 3 & 0 \\ 0 & 1 & 2 \\ 1 & 2 & 4 \\ 3 & 5 & 0 \end{bmatrix}$$

Histogram

Plot of number of occurrences of grey levels against each grey level value for the given image matrix

0	1	2	3	4	5	Intensity, r
3	3	2	2	1	1	No of pixels, n
0.25	0.25	0.16	0.16	0.08	0.08	Probability of pixels, p = n/N

$$A=\begin{bmatrix}1 & 3 & 0\\ 0 & 1 & 2\\ 1 & 2 & 4\\ 3 & 5 & 0\end{bmatrix}$$

Histogram

Plot of number of occurrences of grey levels against each grey level value for the given image matrix

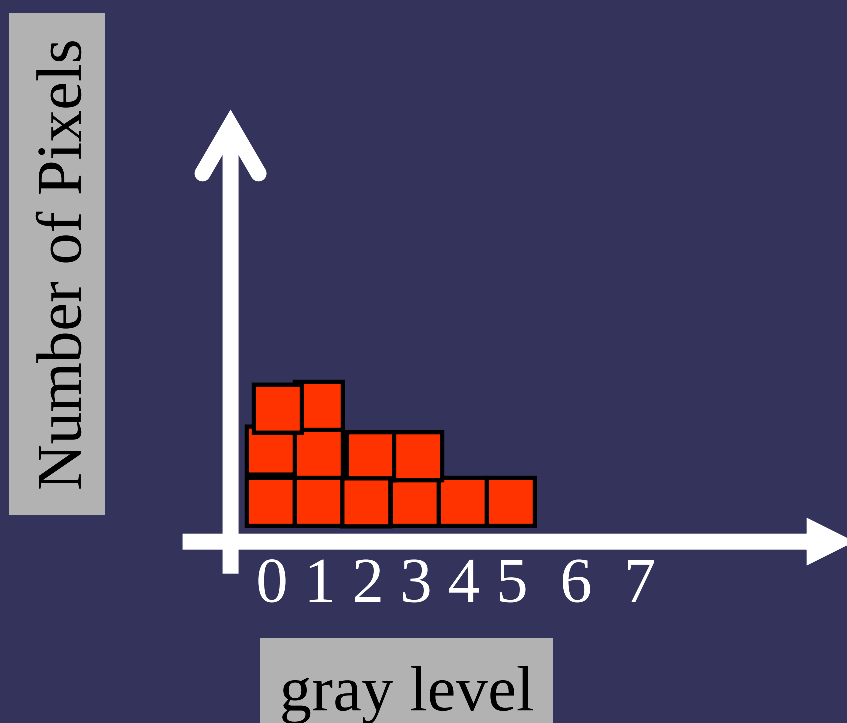
0	1	2	3	4	5	Intensity, r
3	3	2	2	1	1	No of pixels, n
0.25	0.25	0.16	0.16	0.08	0.08	Probability of pixels, p = n/N

$$A=\begin{bmatrix} 1 & 3 & 0 \\ 0 & 1 & 2 \\ 1 & 2 & 4 \\ 3 & 5 & 0 \end{bmatrix}$$

Histogram

Plot of number of occurrences of grey levels against grey level values for 4 by 3 image, N = 12

0	1	2	3	4	5	Intensity, r
3	3	2	2	1	1	No of pixels, n
0.25	0.25	0.16	0.16	0.08	0.08	Probability of pixels, p = n/N

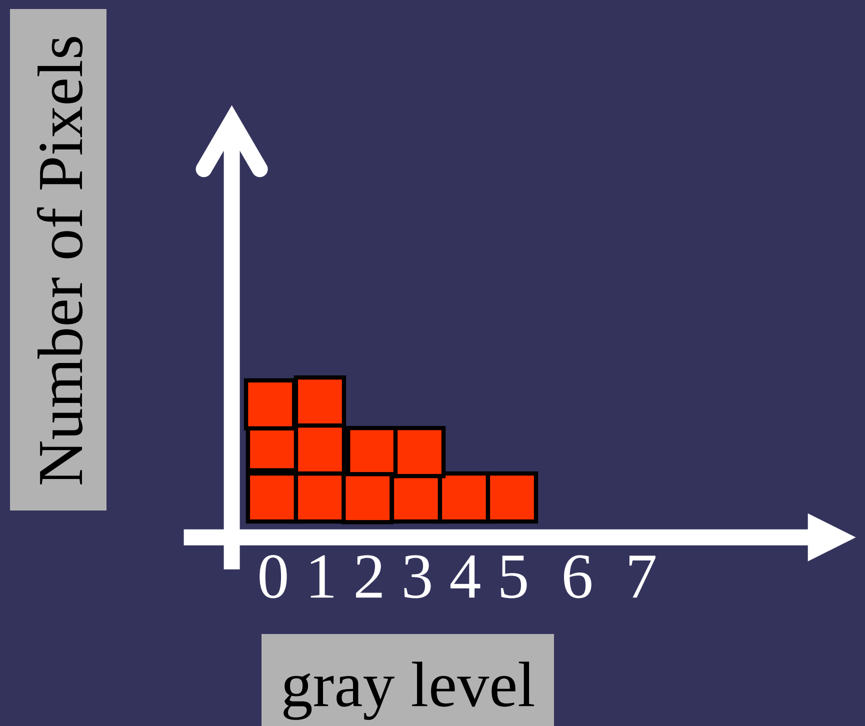


$$A=\begin{bmatrix}1 & 3 & 0\\ 0 & 1 & 2\\ 1 & 2 & 4\\ 3 & 5 & 0\end{bmatrix}$$

Histogram

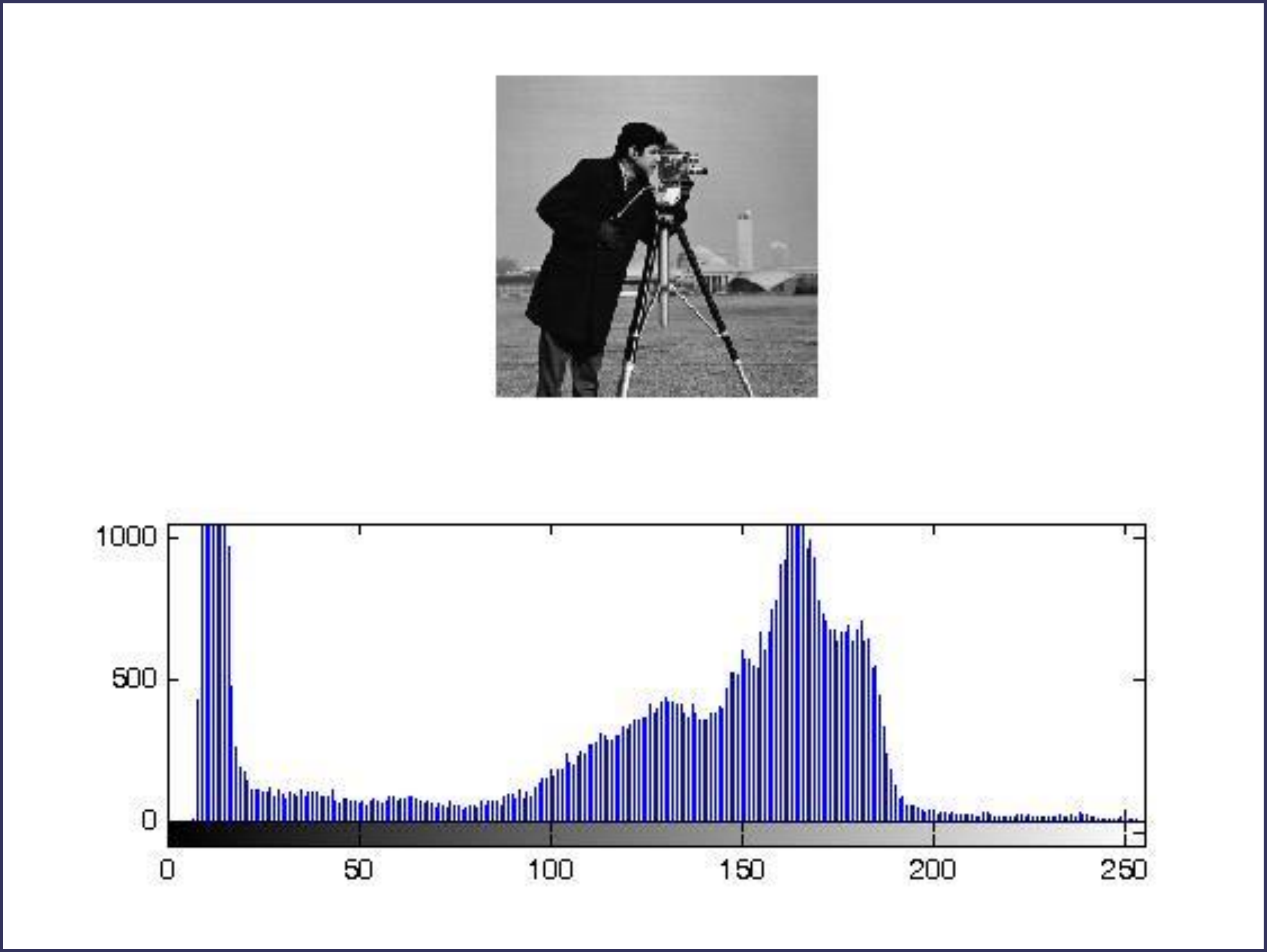
Plot of number of occurrences of grey levels against grey level values for 4 by 3 image, N = 12

0	1	2	3	4	5	Intensity, r
3	3	2	2	1	1	No of pixels, n
0.25	0.25	0.16	0.16	0.08	0.08	Probability of pixels, p = n/N



$$A=\begin{bmatrix}1 & 3 & 0\\ 0 & 1 & 2\\ 1 & 2 & 4\\ 3 & 5 & 1\end{bmatrix}$$

Histogram of an image

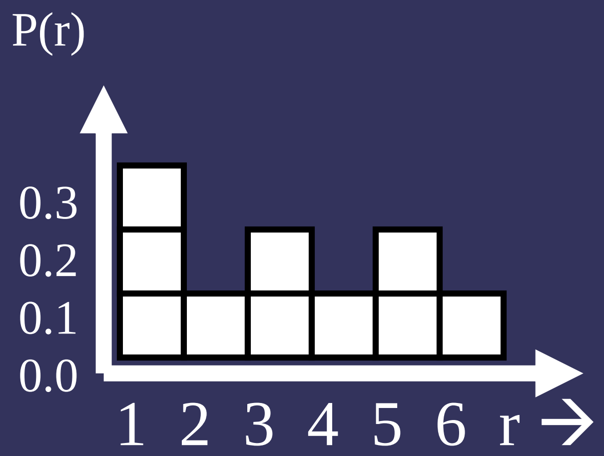


Mean value (or average gray level)

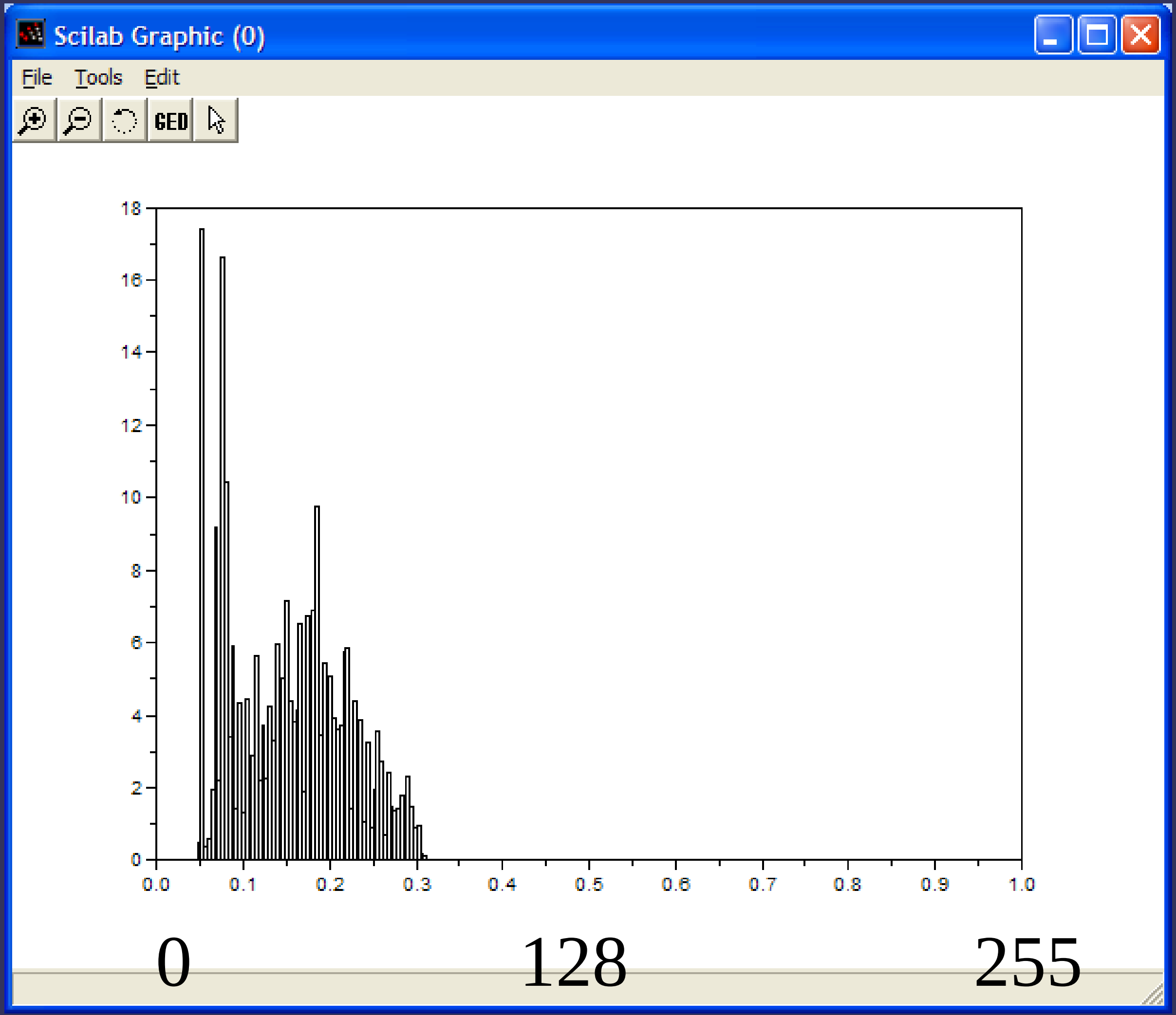
- Probability, $p(r_i) = n_i/(N)$
- Mean, $m = \Sigma_i\, r_i\, p(r_i)$

  $= 1*0.25+2*0.16+3*0.16+4*0.08+5*0.08$

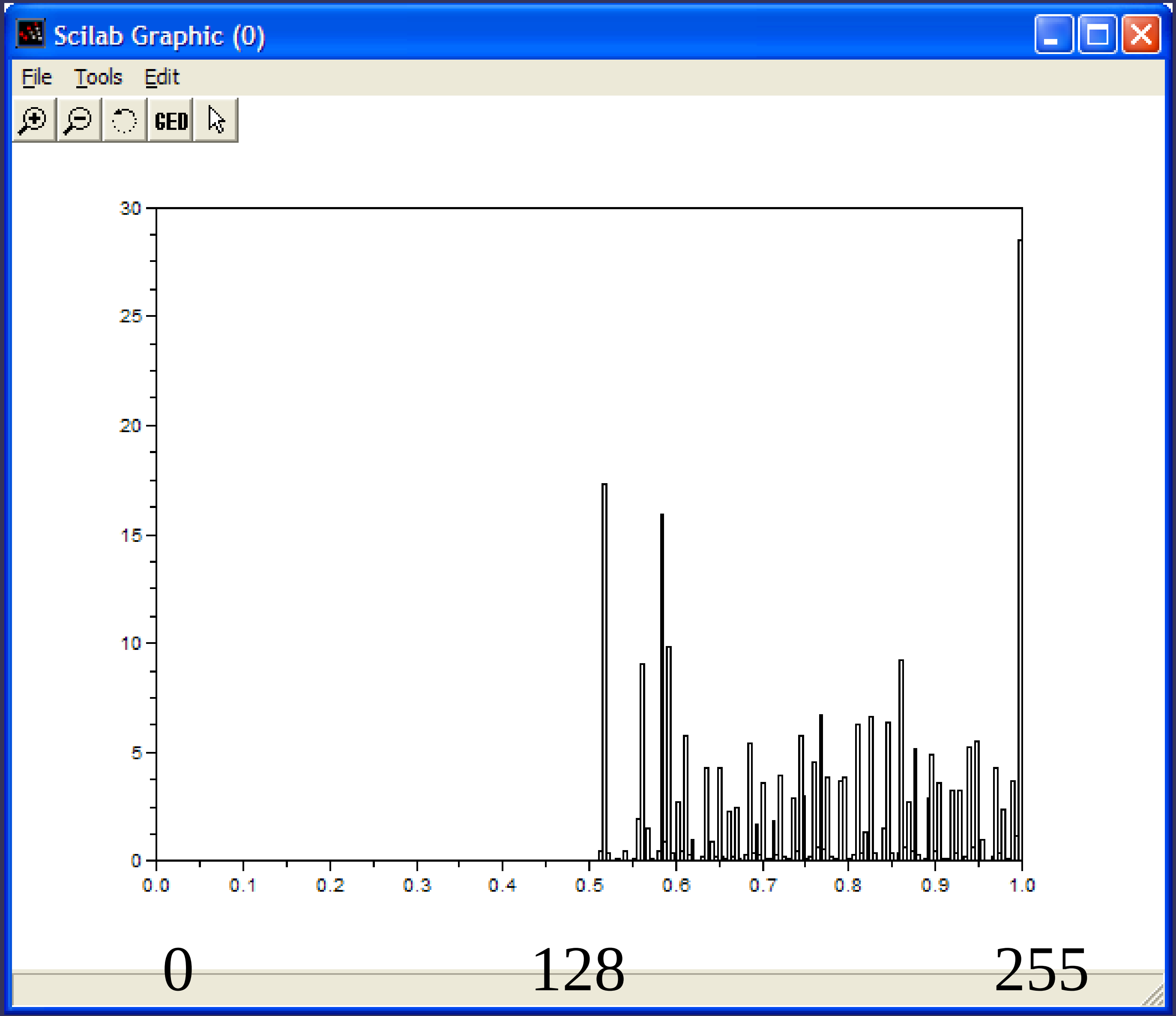
  $= 1.77$
- Mean value represents overall brightness



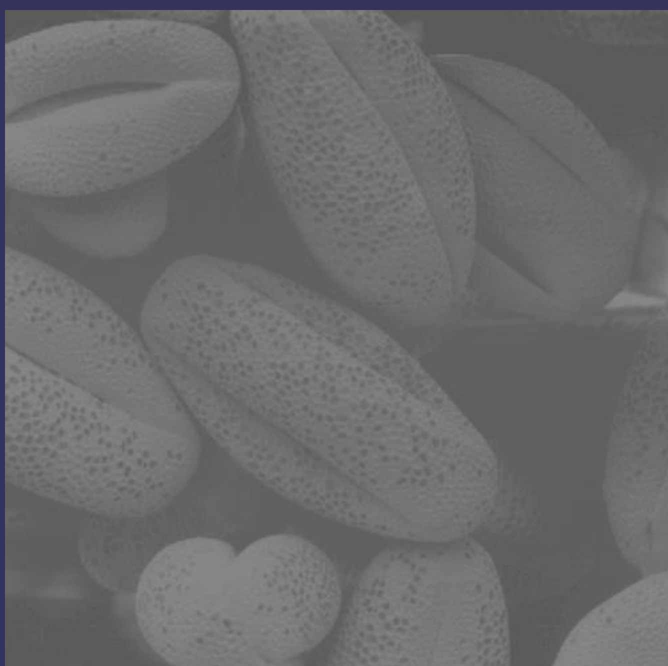
Histogram for contrast

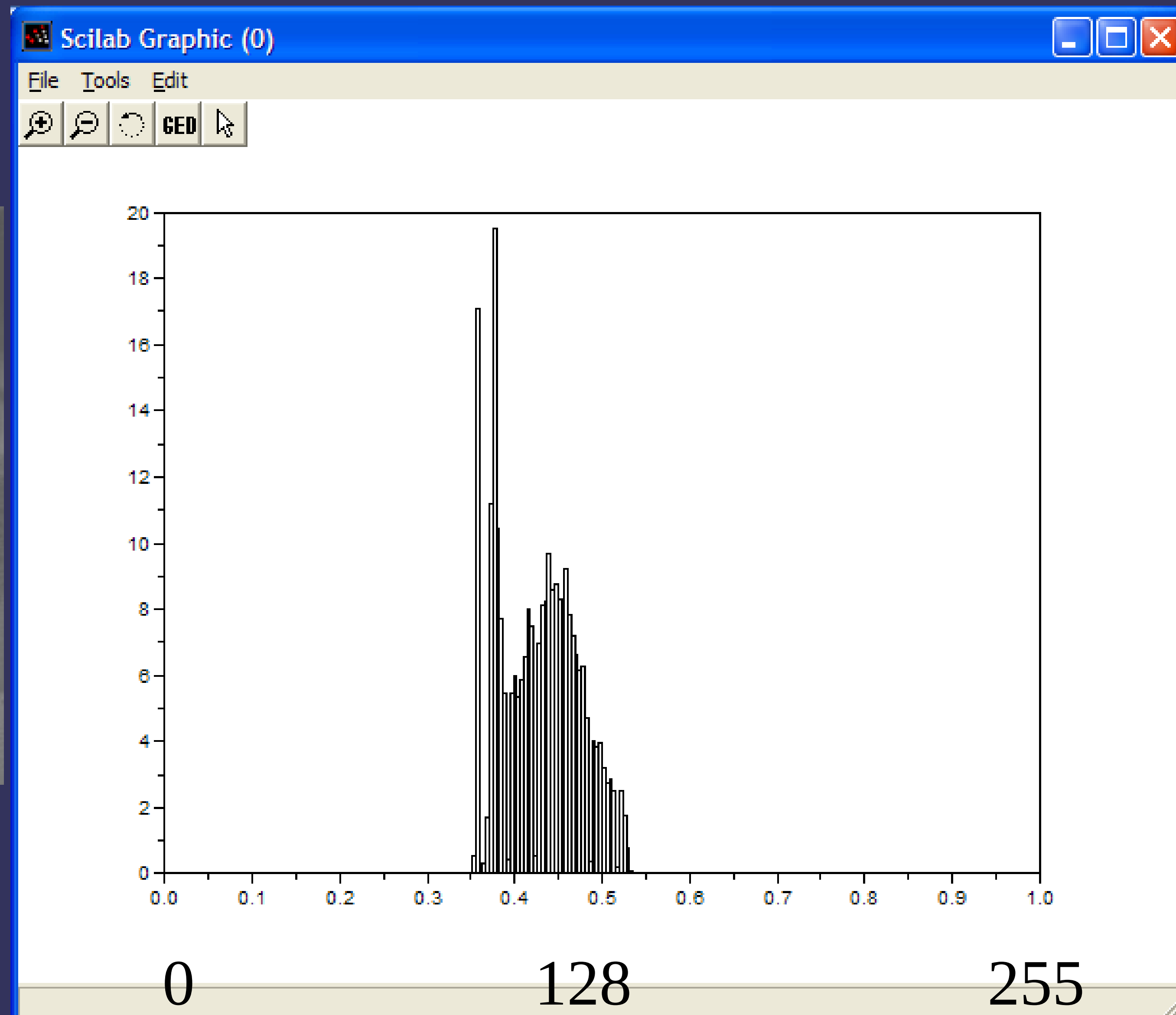


Histogram for contrast

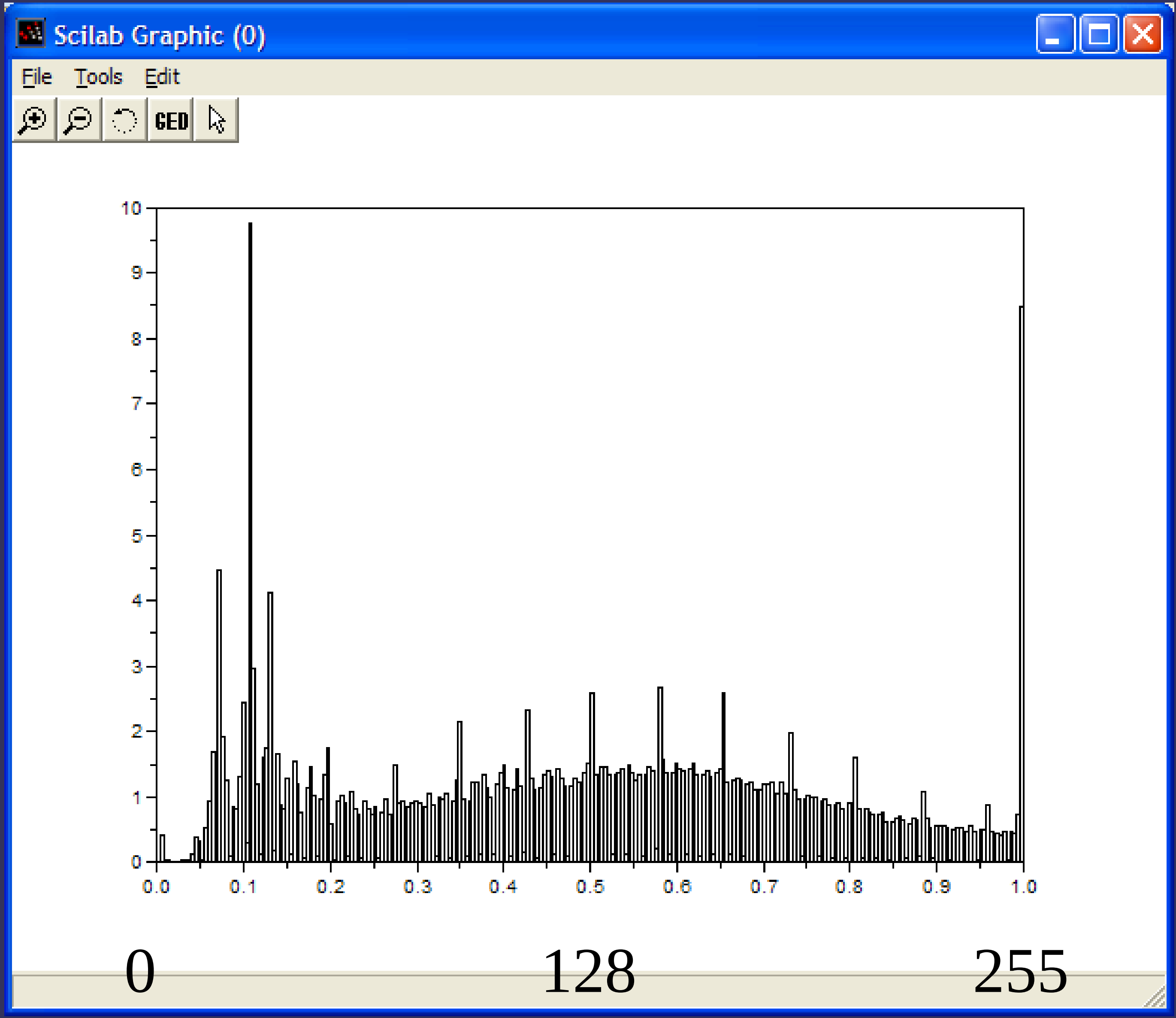


Histogram for contrast



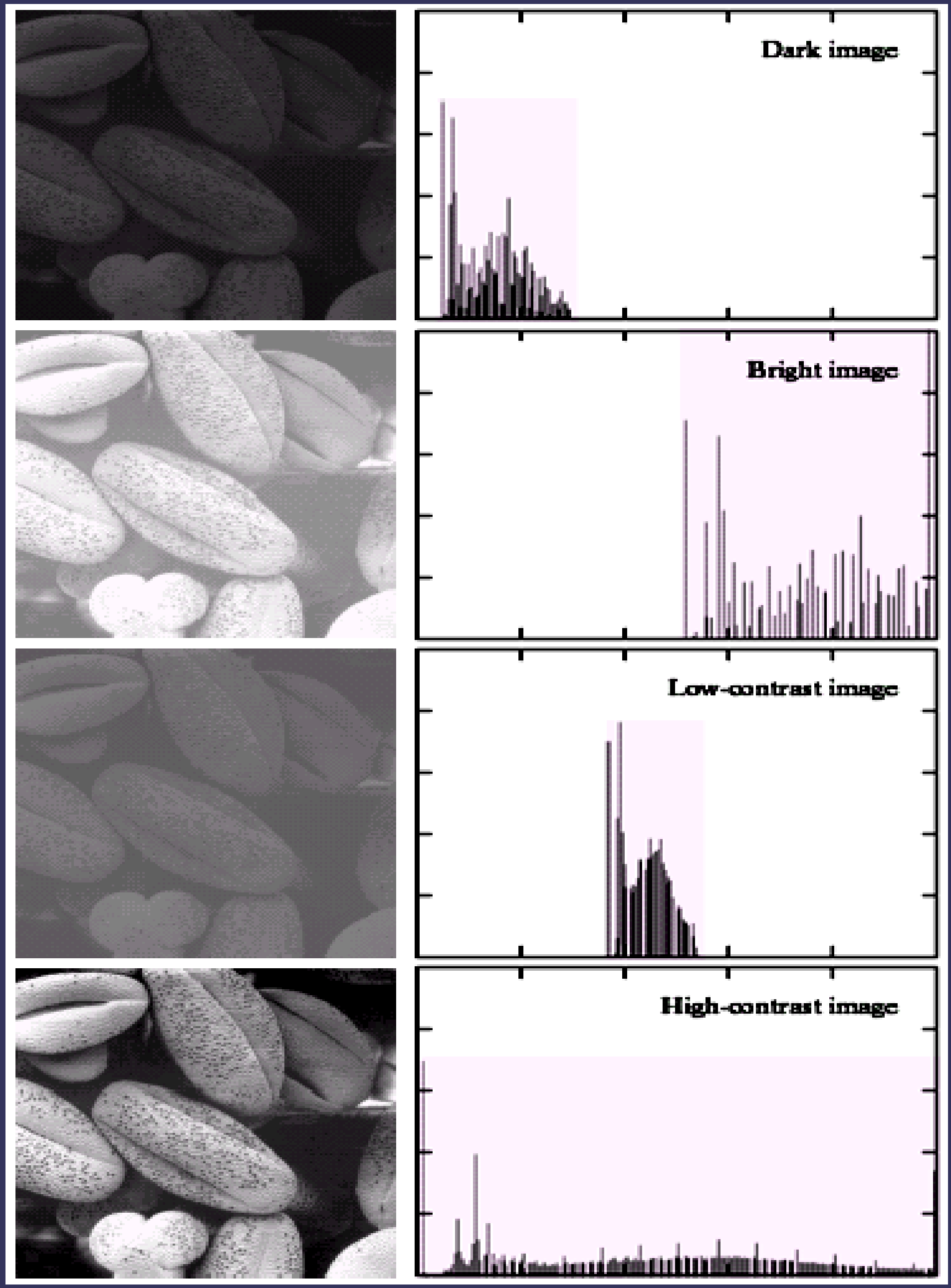


Histogram for contrast



Histogram for contrast

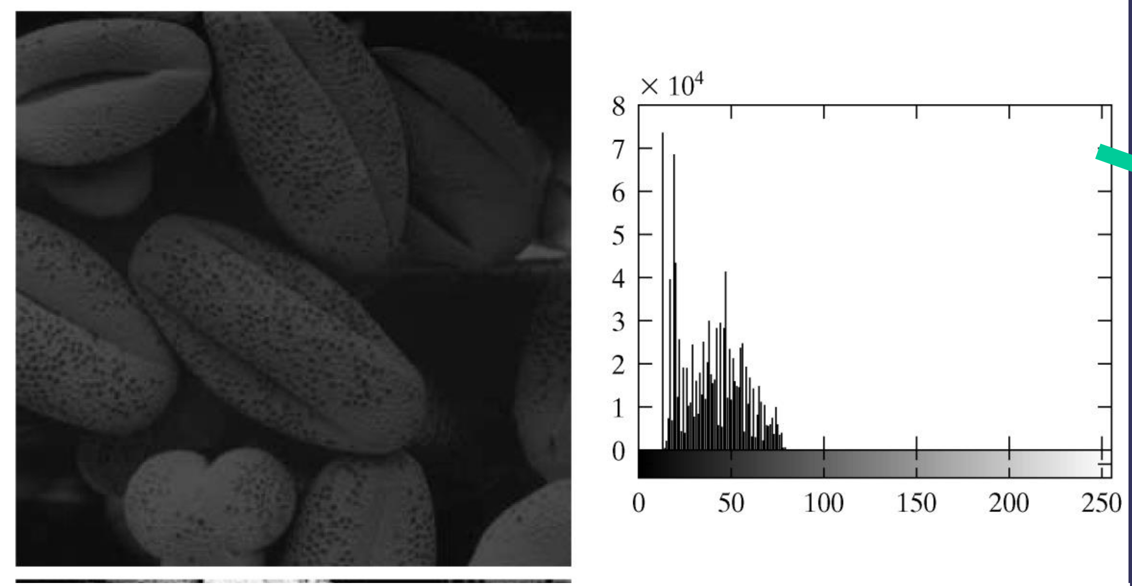
High contrast
image has the most
evenly spaced histogram

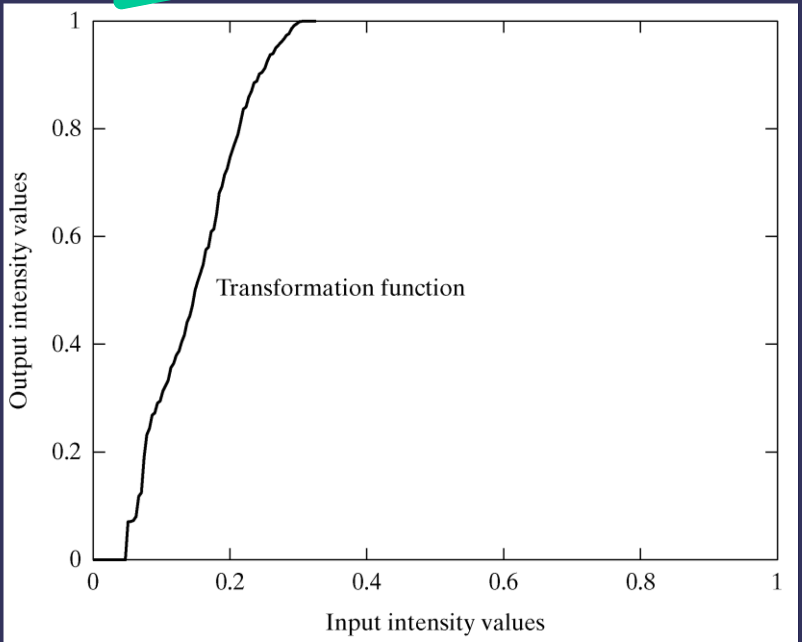


Histogram Equalization

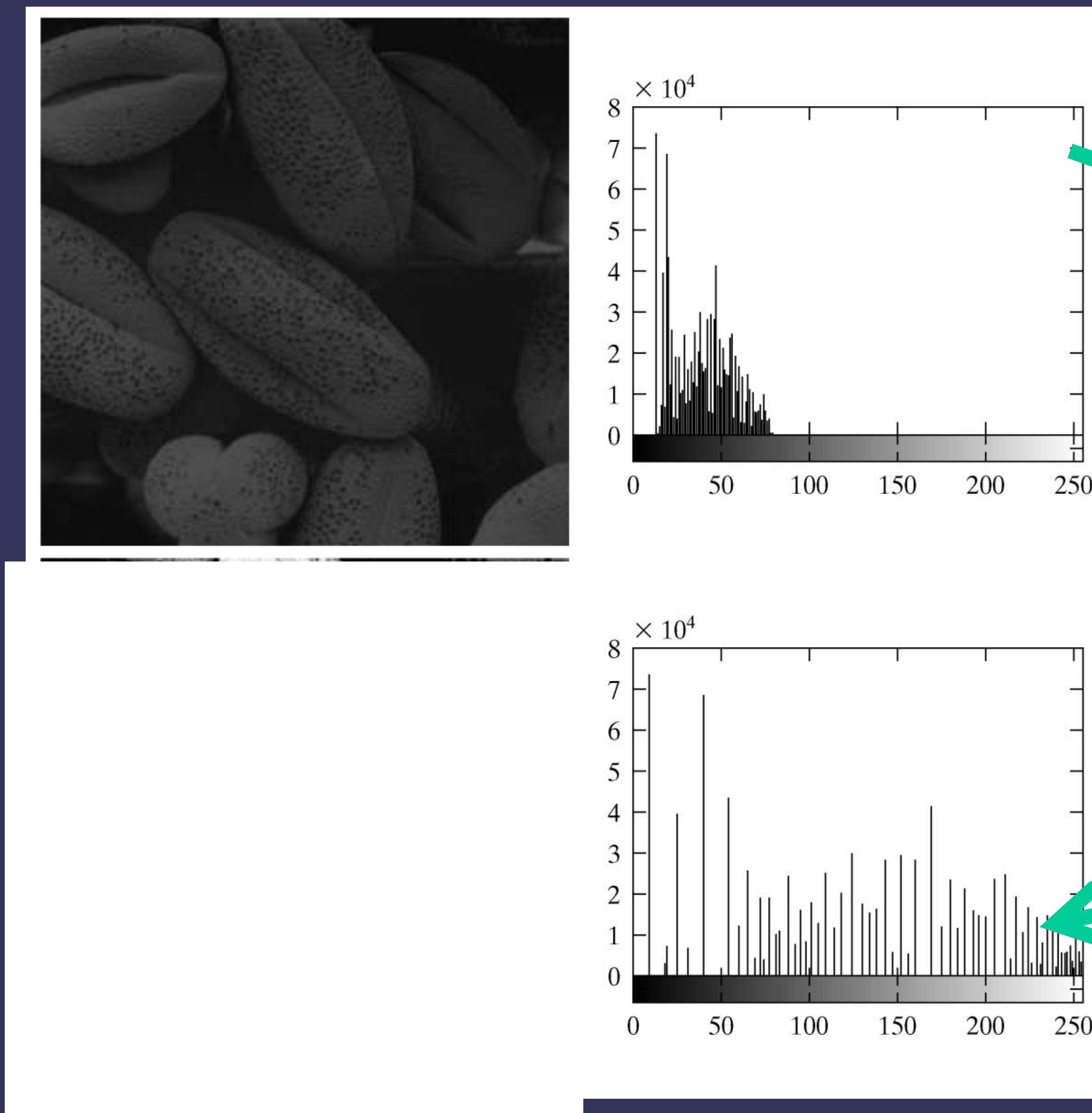
- Preprocessing technique to enhance contrast in ‘natural’ images
- Improves dark or washed out images
- Redistributes to generate equal number of pixels for every gray-value
- Spreads the frequencies of an image
- Therefore it is called as equalization
- Gray level transformation function T to transform image f such that the histogram of T(f) is ‘equalized’

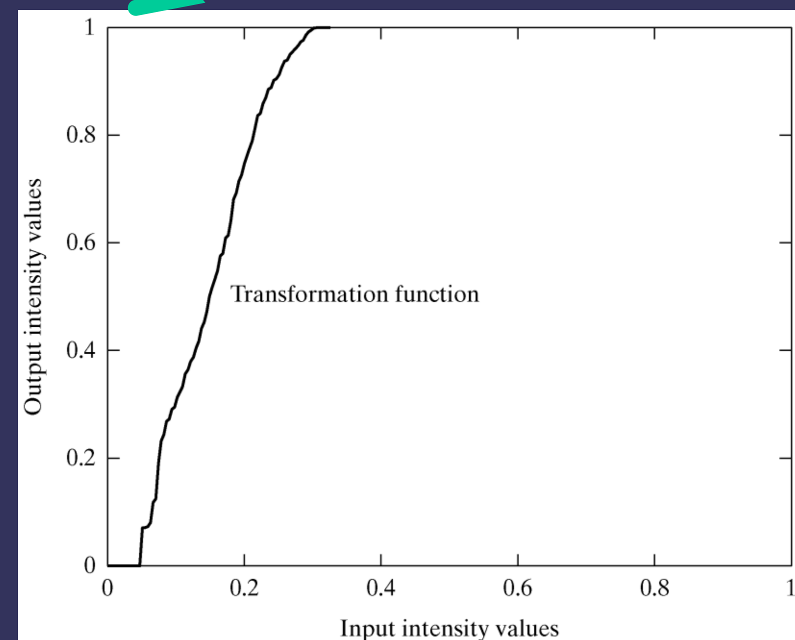
Equalisation Transformation Function



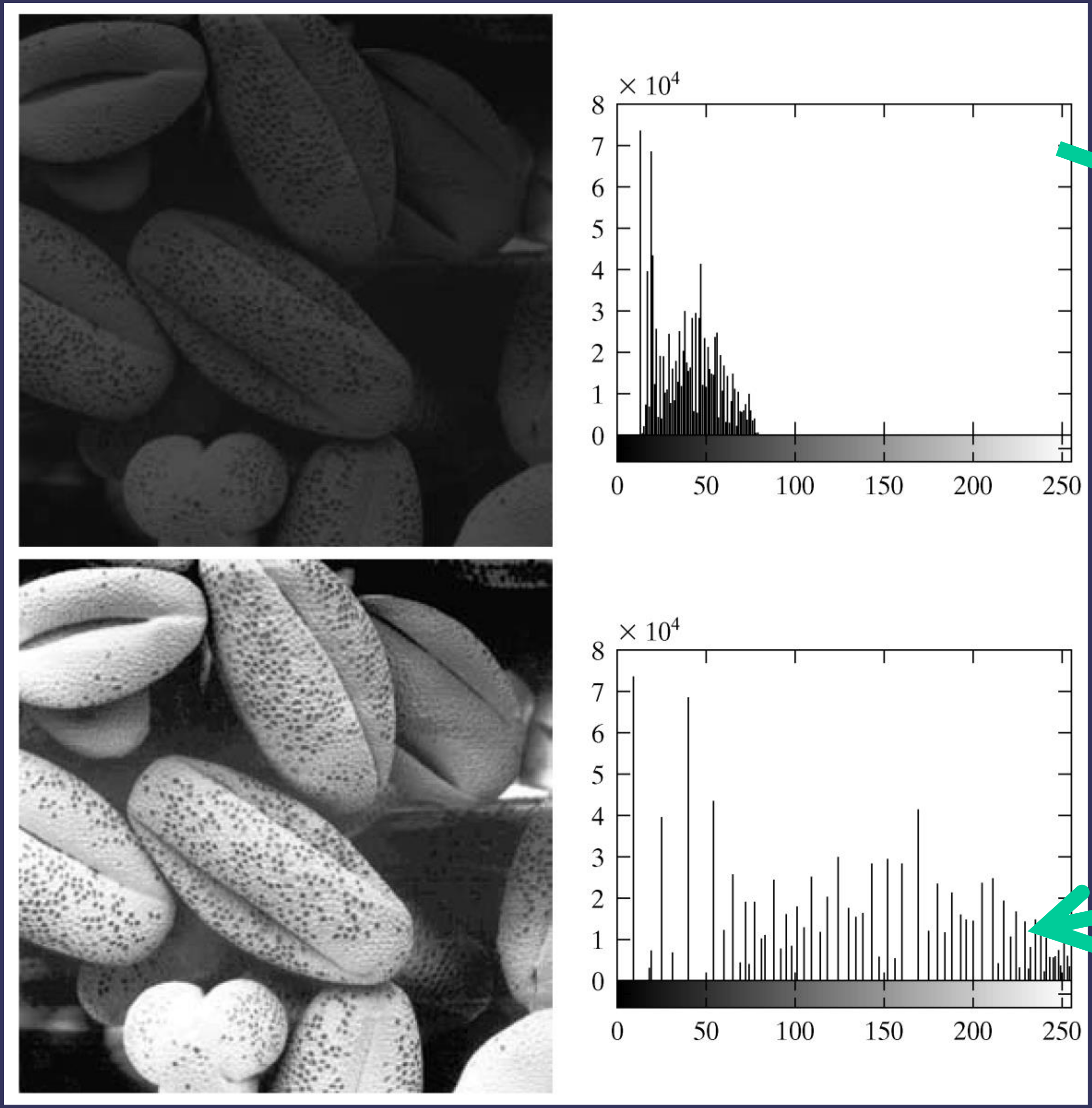


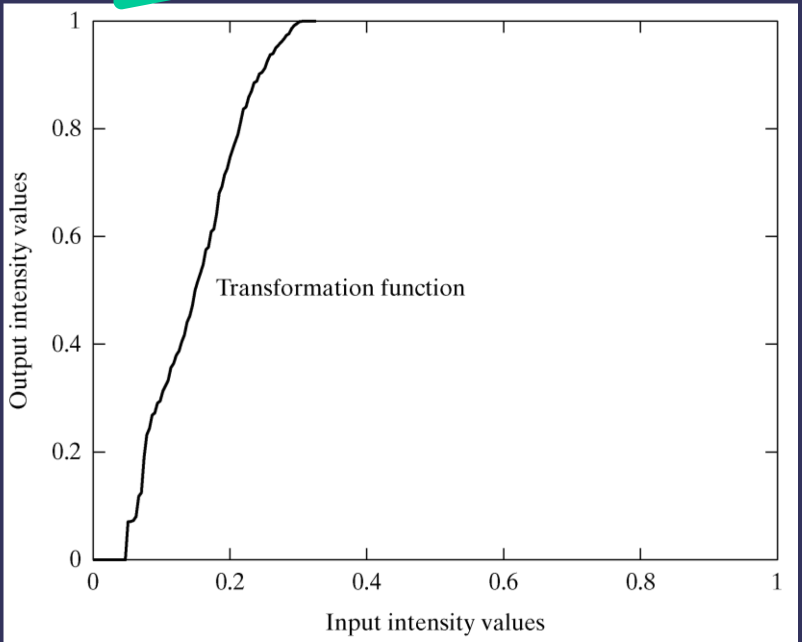
Equalisation Transformation Function



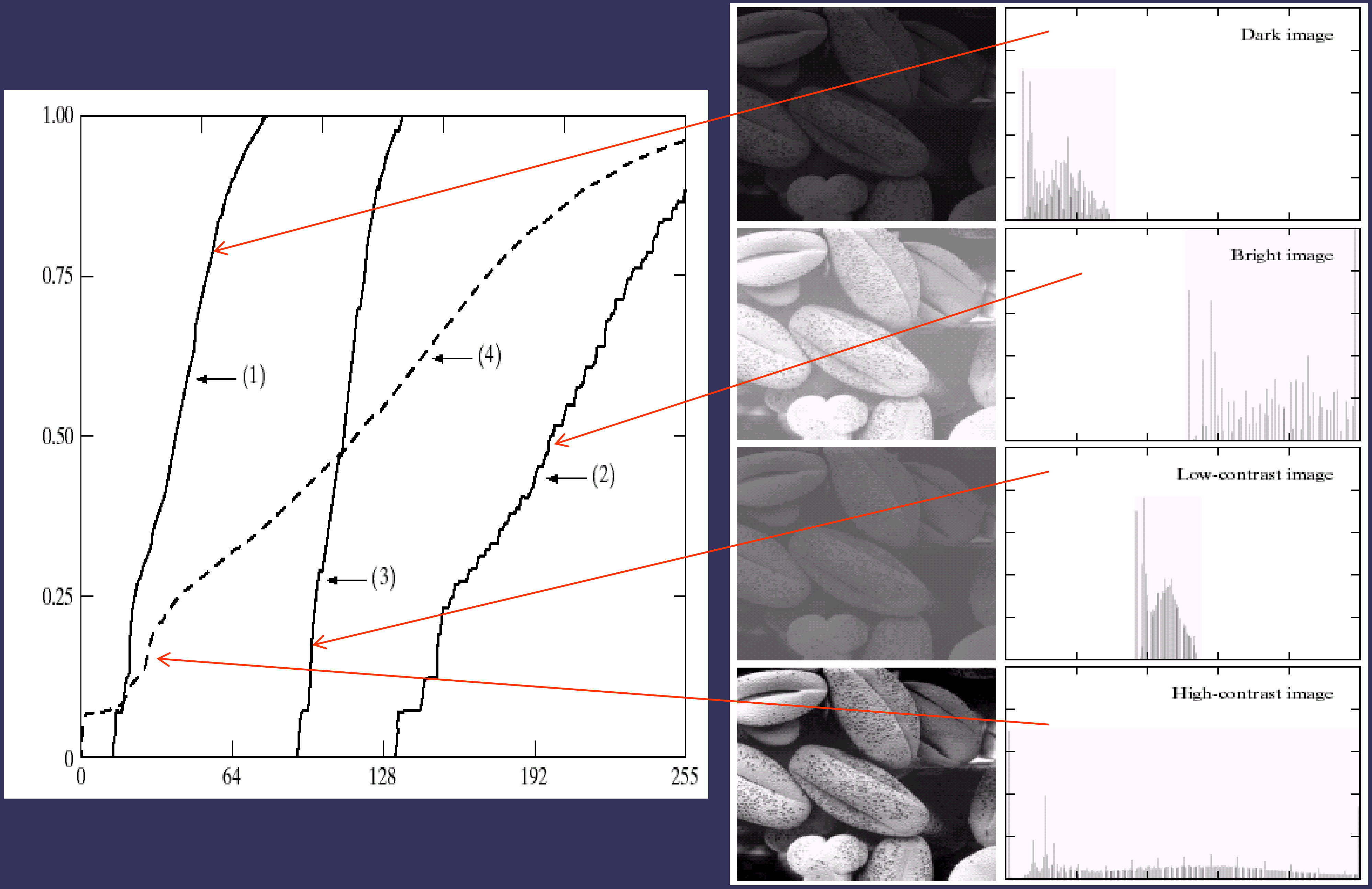


Equalisation Transformation Function

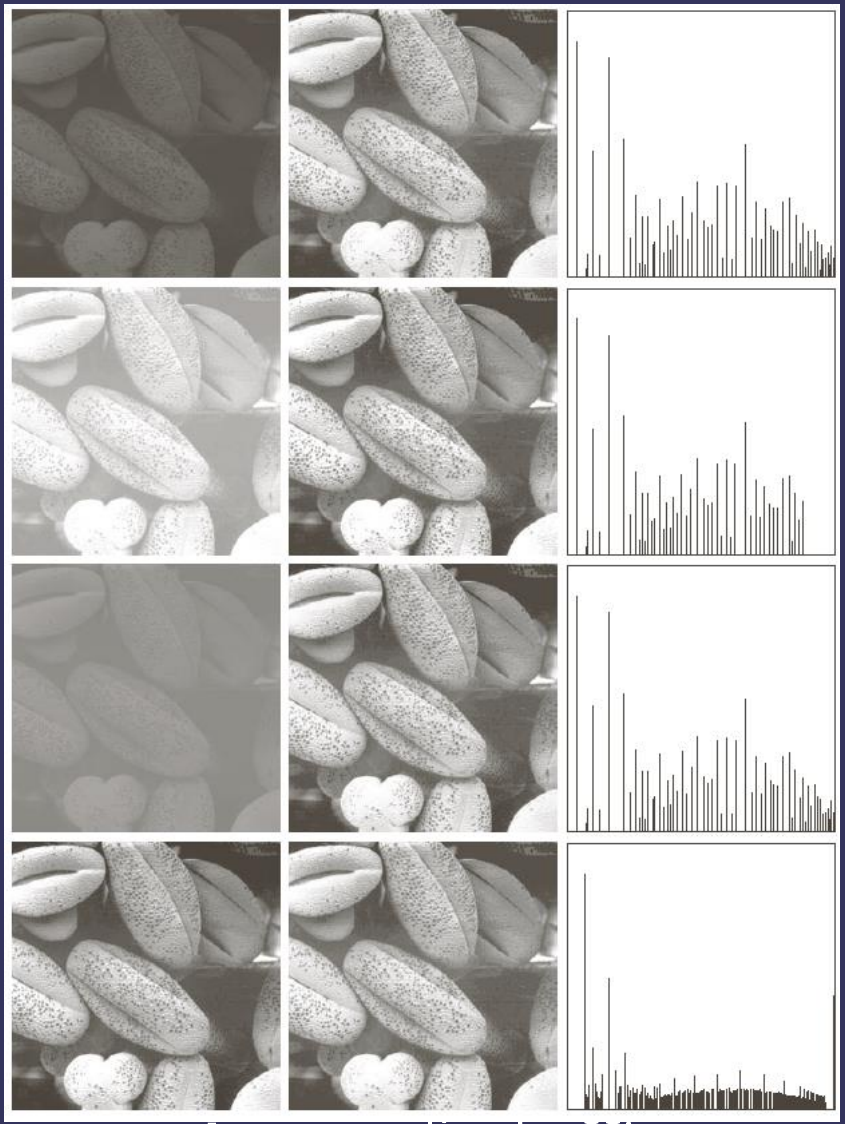




Equalisation Transformation Functions



Histogram Equalization



Original images

equalized images

Histogram of equalized images