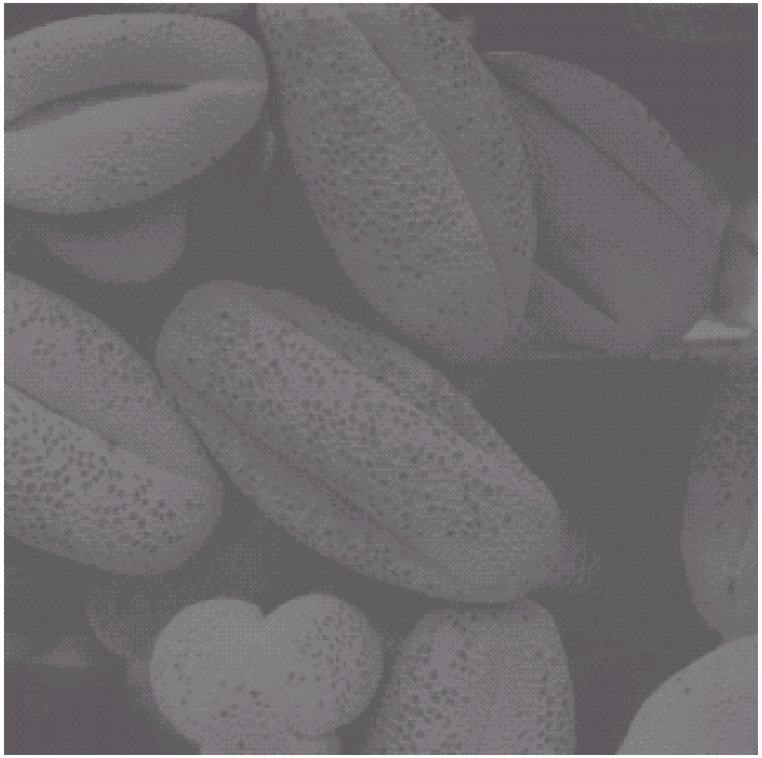
Digital Image Processing

Week-03

Image Enhancement


Image Enhancement



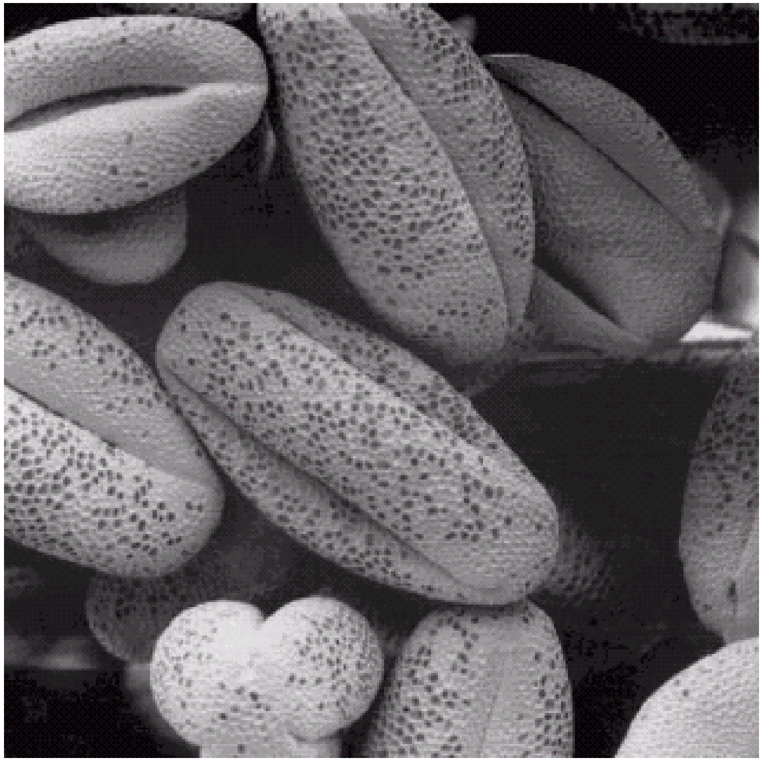


Image Enhancement

Process an image so that the result is more suitable than the original image for a specific application

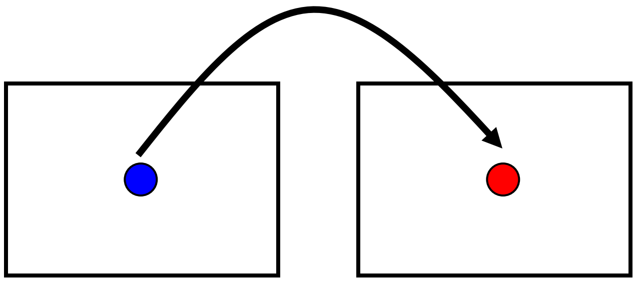
- Image Enhancement Methods
  - Spatial Domain: Direct manipulation of pixels in an image
  - Frequency Domain: Process the image by modifying the Fourier transform of an image

This Chapter – Spatial Domain

Types of image enhancement operations

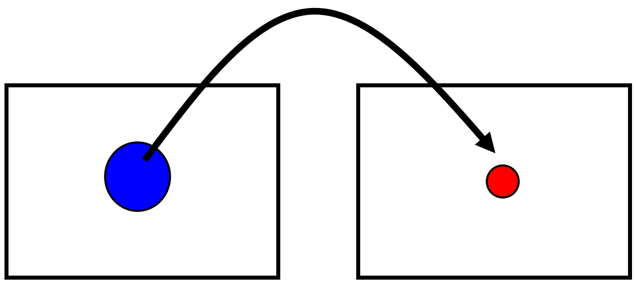
- Point/Pixel operations

  Output value at specific coordinates *(x,y)* is dependent only on the input value at *(x,y)*



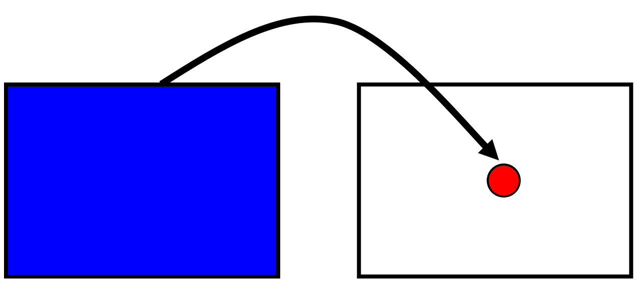
- Local operations

  The output value at *(x,y)* is dependent on the input values in the neighborhood of *(x,y)*



- Global operations

  The output value at *(x,y)* is dependent on all the values in the input image



Basic Concepts

- Most spatial domain enhancement operations can be generalized as:

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

f (x, y) = the input image

g (x, y) = the processed/output image

T = some operator defined over some neighbourhood of *(x, y)*

Point Processing

- In a digital image, point = pixel
- Point processing transforms a pixel's value as function of its value alone;
- It does not depend on the values of the pixel's neighbors.

Point Processing

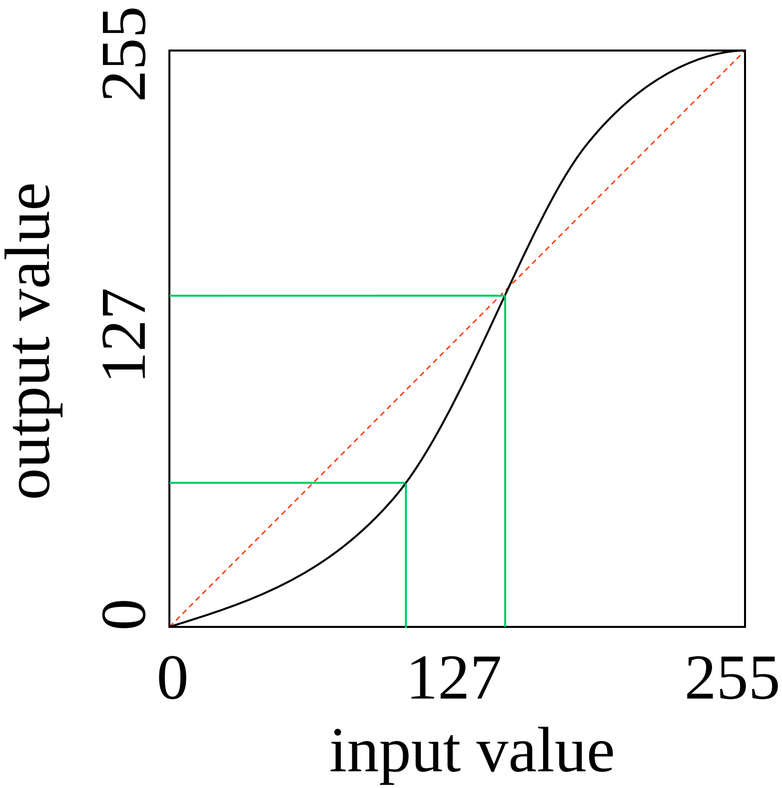
- Neighborhood of size 1x1:
- g depends only on f at (x,y)
- T: Gray-level/intensity transformation/ mapping function

$$s = T(r)$$

- r = gray level of f at (x,y)
- s = gray level of g at (x,y)

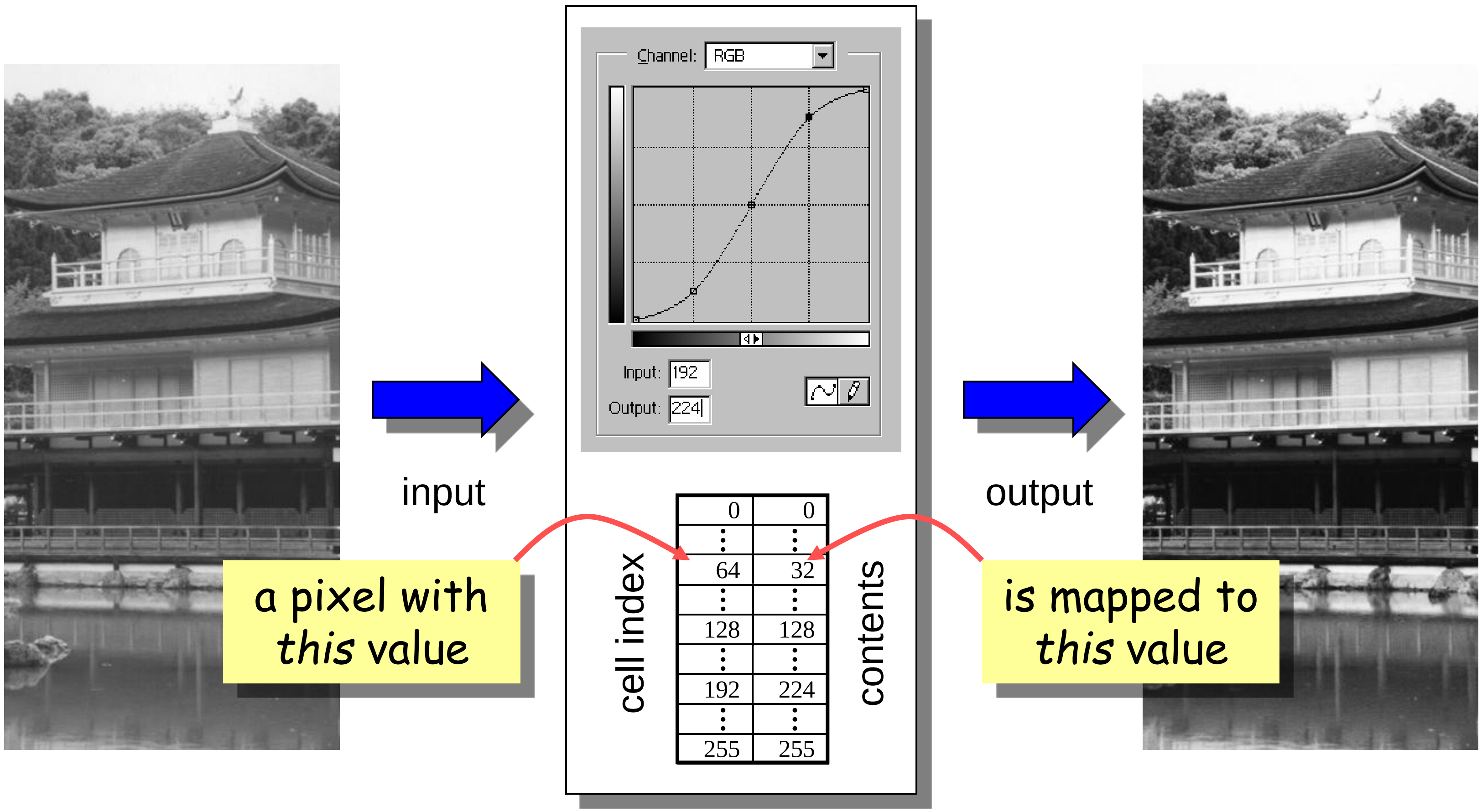
Point Processing using Look-up Tables

A look-up table (LUT) implements a functional mapping.

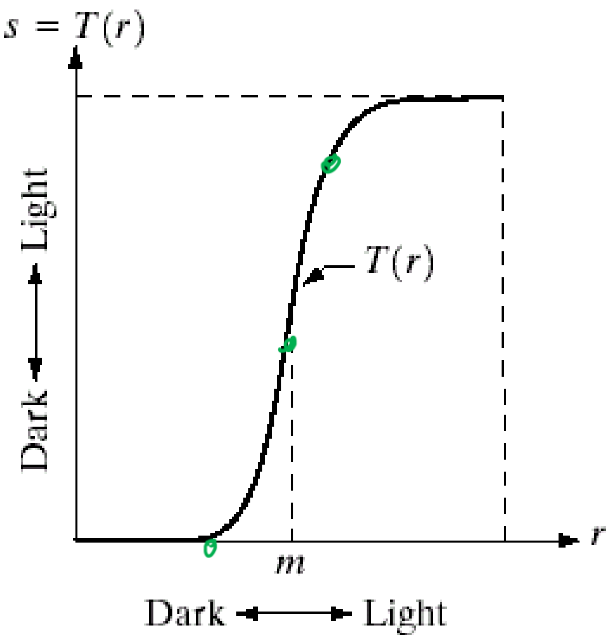


E.g.:	index	value
	...	...
	101	64
	102	68
	103	69
	104	70
	105	70
	106	71
	...	...
	input	output

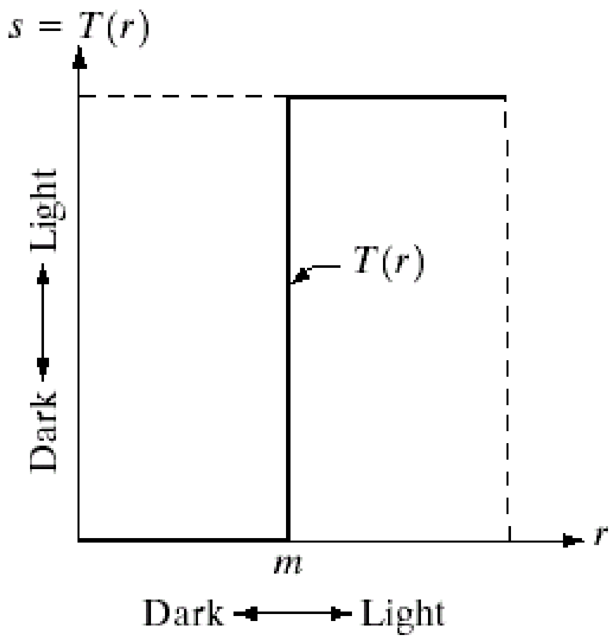
Point Processing using Look-up Tables



POINT PROCESSING



Contrast Stretching



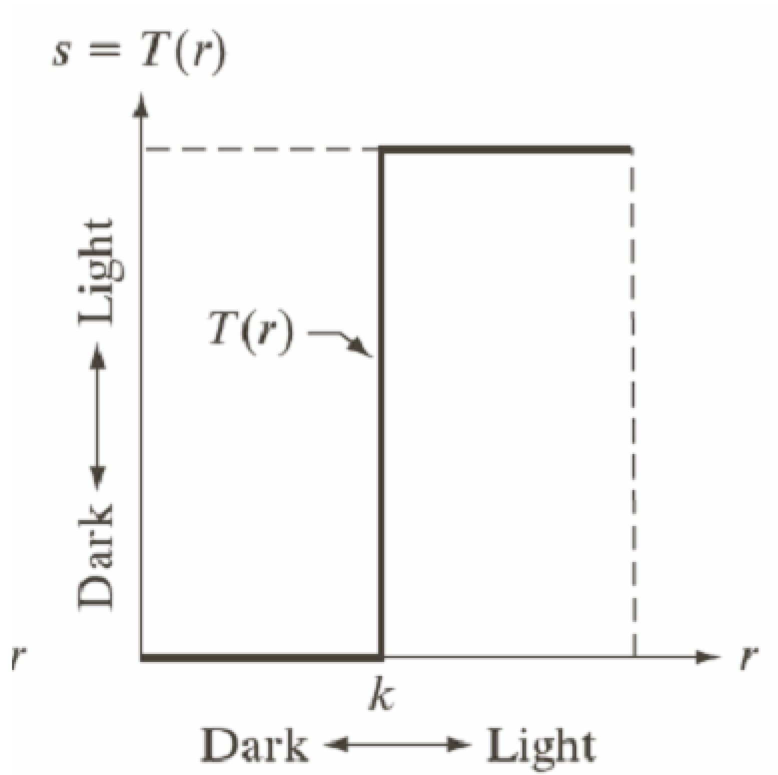
Thresholding

a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

Point Processing Example: Thresholding

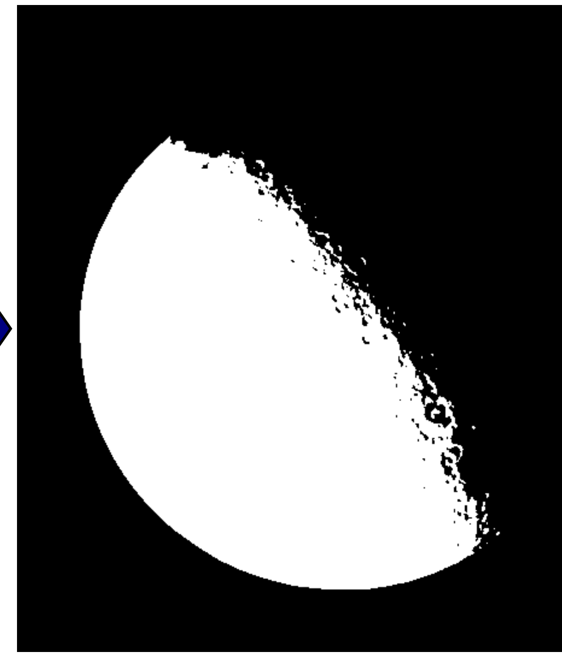
$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$



Point Processing Example: Thresholding

- Segmentation of an object of interest from a background

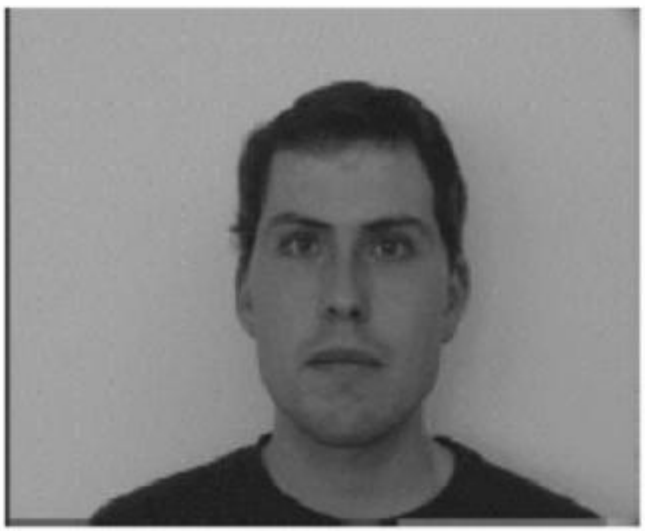
$$s = \begin{cases} 1.0 & r > \text{threshold} \\ 0.0 & r <= \text{threshold} \end{cases}$$



Point Processing Example: Intensity Scaling

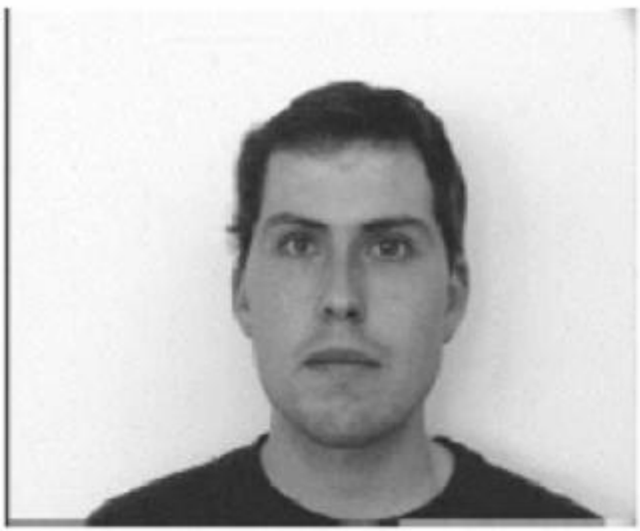
$$s = T(r) = a.r$$

Original image



$f(x,y)$

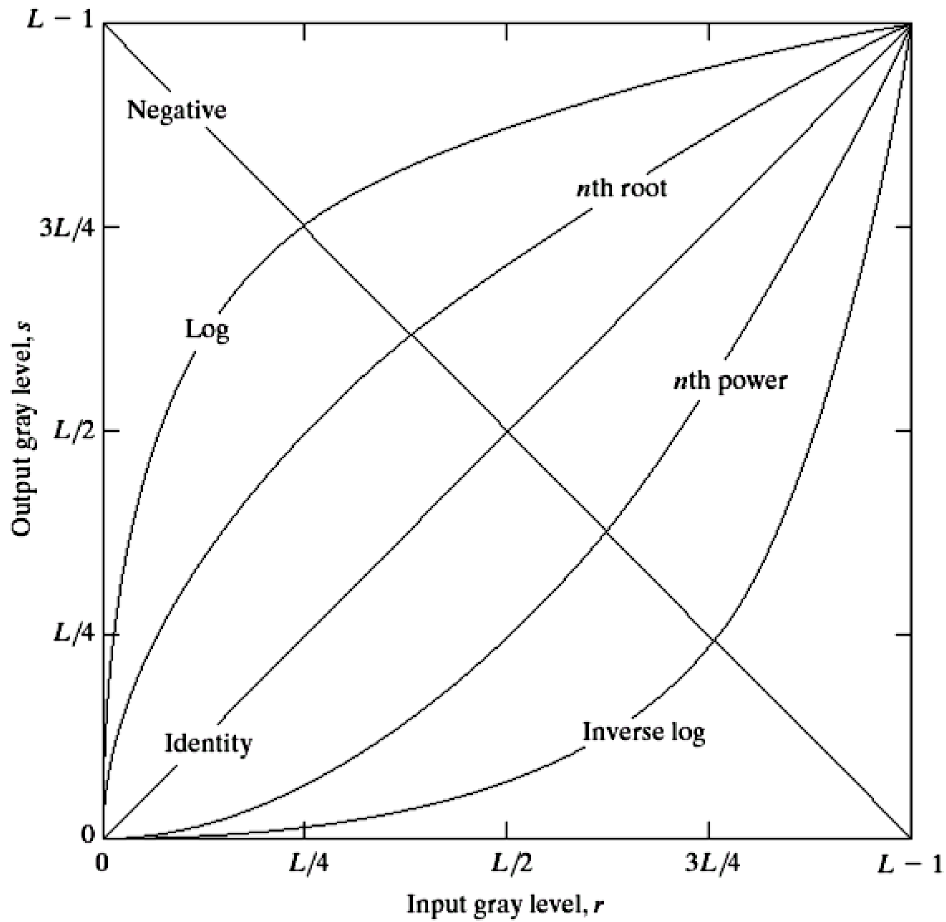
Scaled image



$a \cdot f(x,y)$

Point Processing Transformations

- There are many different kinds of grey level transformations
- Three of the most common are shown here
  - Linear
    - Negative/Identity
  - Logarithmic
    - Log/Inverse log
  - Power law
    - n^{th} power/n^{th} root



Point Processing Example: Negative Images

- Reverses the gray level order
- For L gray levels, the transformation has the form:

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

- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image

Point Processing Example: Negative Images

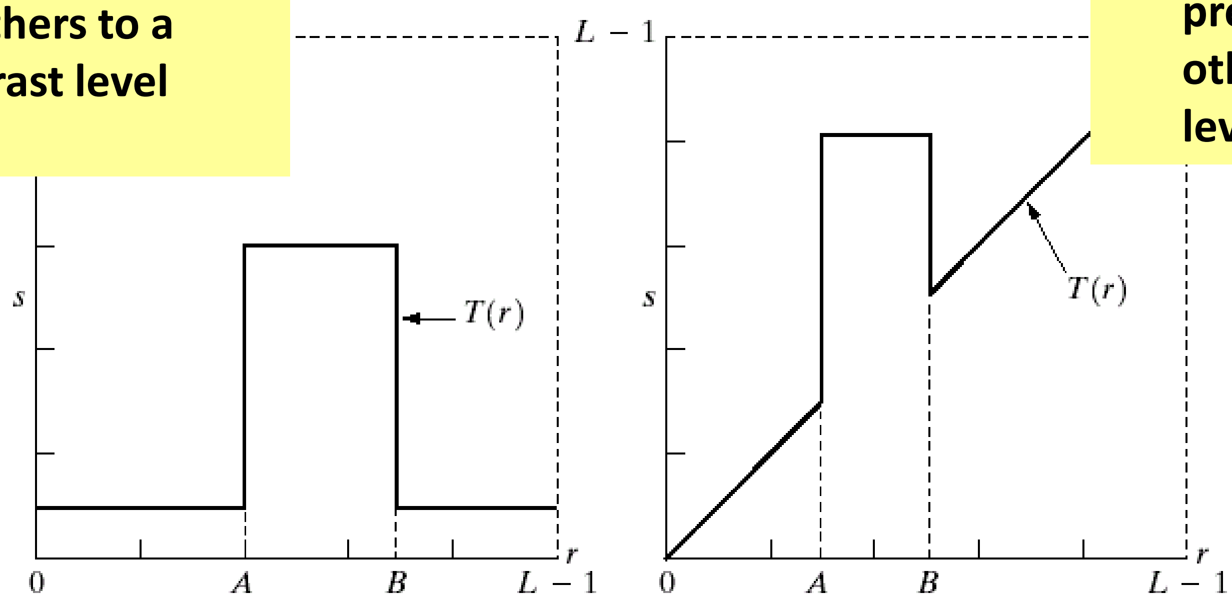
Input image (X-ray image)

Output image (negative)

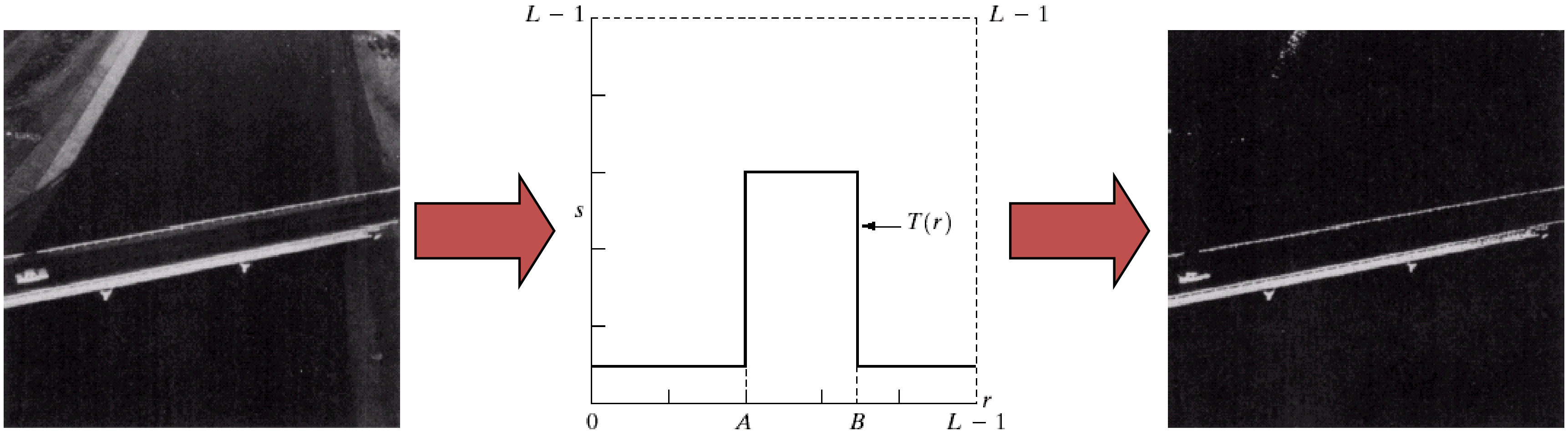
Grey Level Slicing

Highlights range [A,B] of gray levels and reduces all others to a contrast level

Highlights range [A,B] but preserves all other gray levels

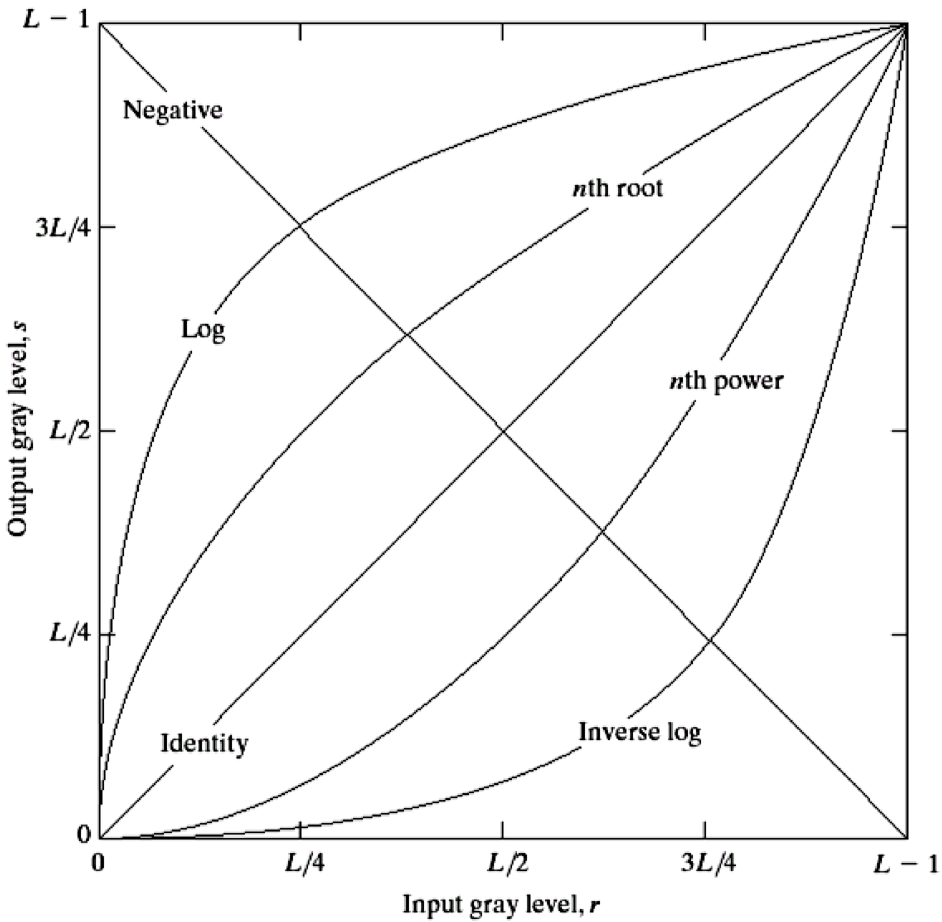


Grey Level Slicing



Point Processing Transformations

- There are many different kinds of grey level transformations
- Three of the most common are shown here
  - Linear
    - Negative/Identity
  - Logarithmic
    - Log/Inverse log
  - Power law
    - n^{th} power/n^{th} root



Point Processing Example: Negative Images

- Reverses the gray level order
- For L gray levels, the transformation has the form:

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

- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image

Point Processing Example: Negative Images

Input image (X-ray image)

Output image (negative)

Test Yourself

A 5x4 image is given below. Use 8 connectivity based connected component analysis to find number of objects and size of each object

Even

0	1	0	0
1	1	0	0
0	0	0	1
0	1	1	0
0	0	1	0

Odd

0	0	0	1
1	1	0	0
0	0	1	1
0	1	1	0
1	0	0	0

Logarithmic Transformations

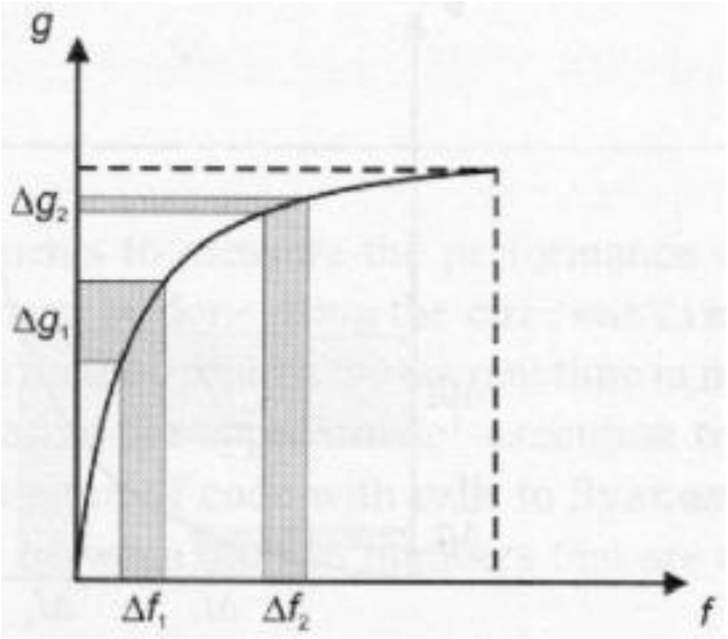
- The general form of the log transformation is

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

- The log transformation maps a narrow range of low input grey level values into a wider range of output values
- The inverse log transformation performs the opposite transformation

Logarithmic Transformations

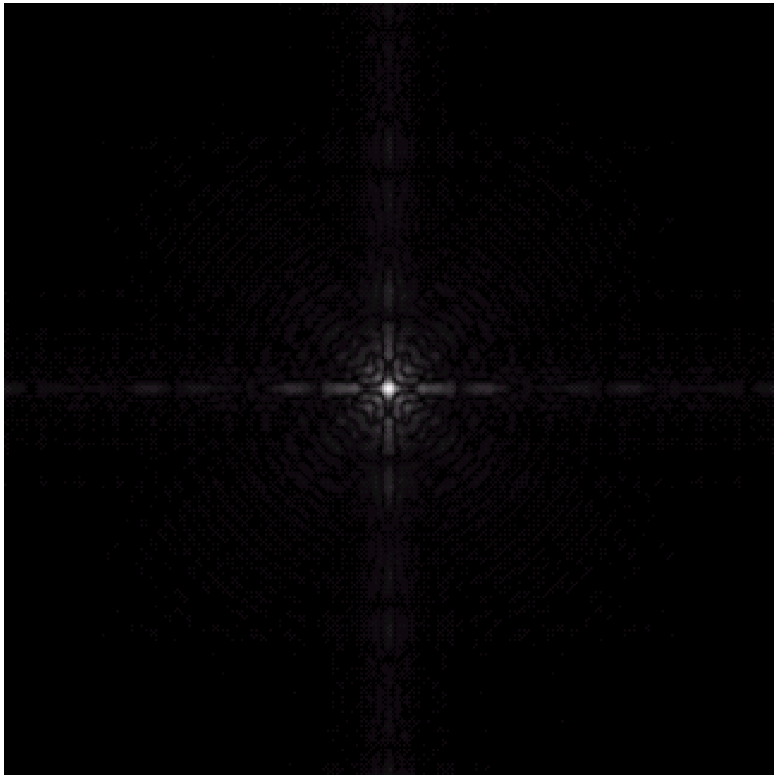
- **Properties**
  - For lower amplitudes of input image the range of gray levels is expanded
  - For higher amplitudes of input image the range of gray levels is compressed



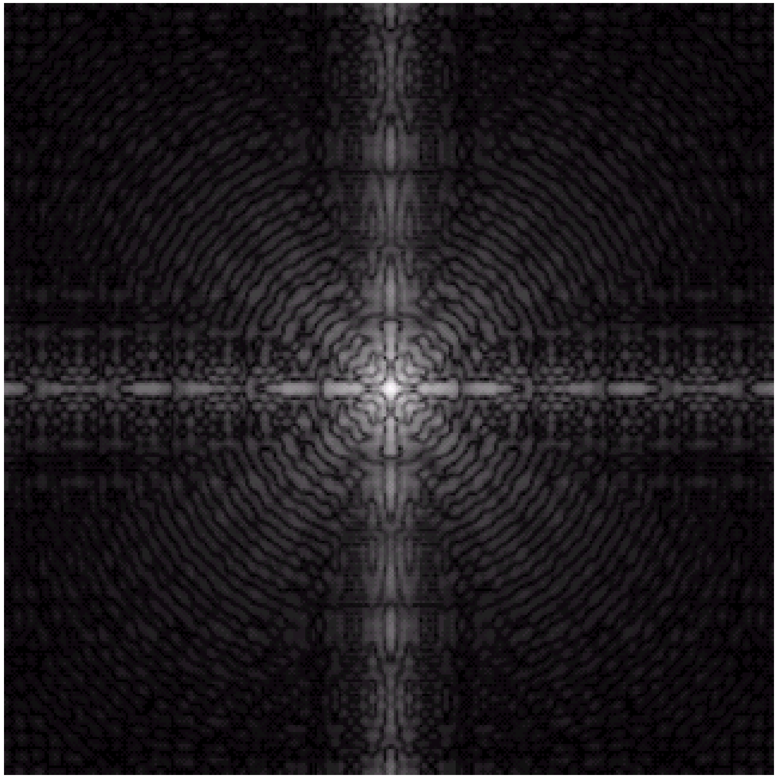
Logarithmic Transformations

- **Application**
  - This transformation is suitable for the case when the dynamic range of a processed image far exceeds the capability of the display device (e.g. display of the Fourier spectrum of an image)
  - Also called "dynamic-range compression / expansion"

Logarithmic Transformations



Fourier spectrum: image values ranging from 0 to 1.5×10^6



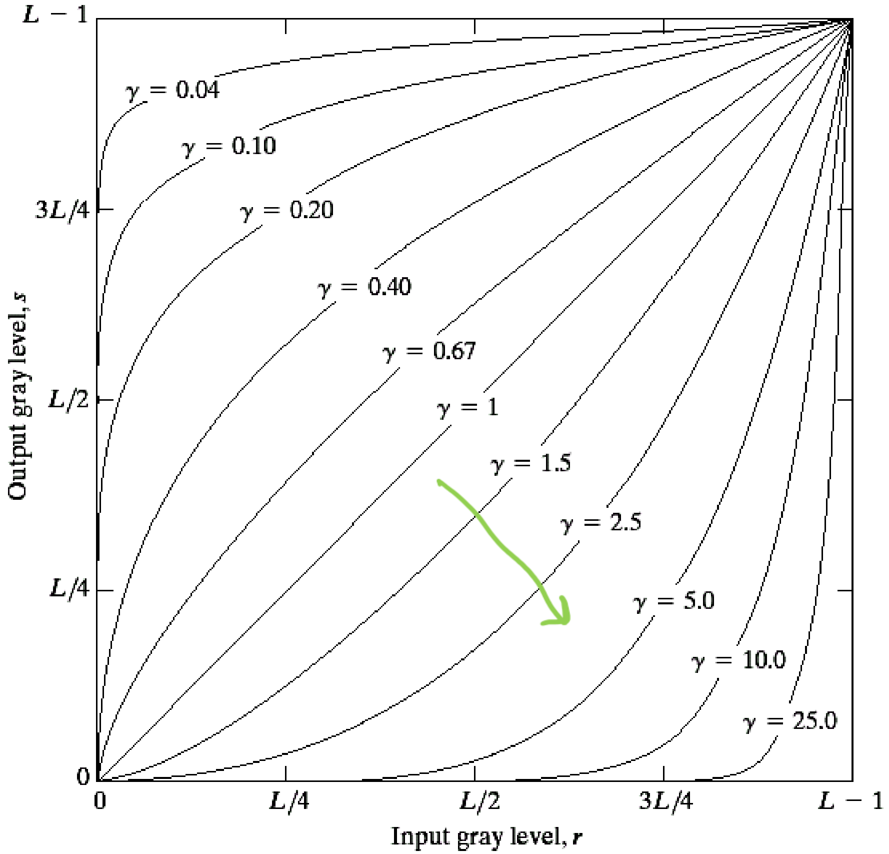
The result of log transformation with c = 1

Power Law Transformations

- Power law transformations have the following form

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

- Map a narrow range of dark input values into a wider range of output values or vice versa
- Varying γ gives a whole family of curves



Power Law Transformations

- For $\gamma < 1$: Expands values of dark pixels, compress values of brighter pixels
- For $\gamma > 1$: Compresses values of dark pixels, expand values of brighter pixels
- If $\gamma=1$ & $c=1$: Identity transformation ($s = r$)

- A variety of devices (image capture, printing, display) respond according to a power law and need to be corrected

- **Gamma (γ) correction**

  The process used to correct the power-law response phenomena

Power Law Transformations: Gamma Correction

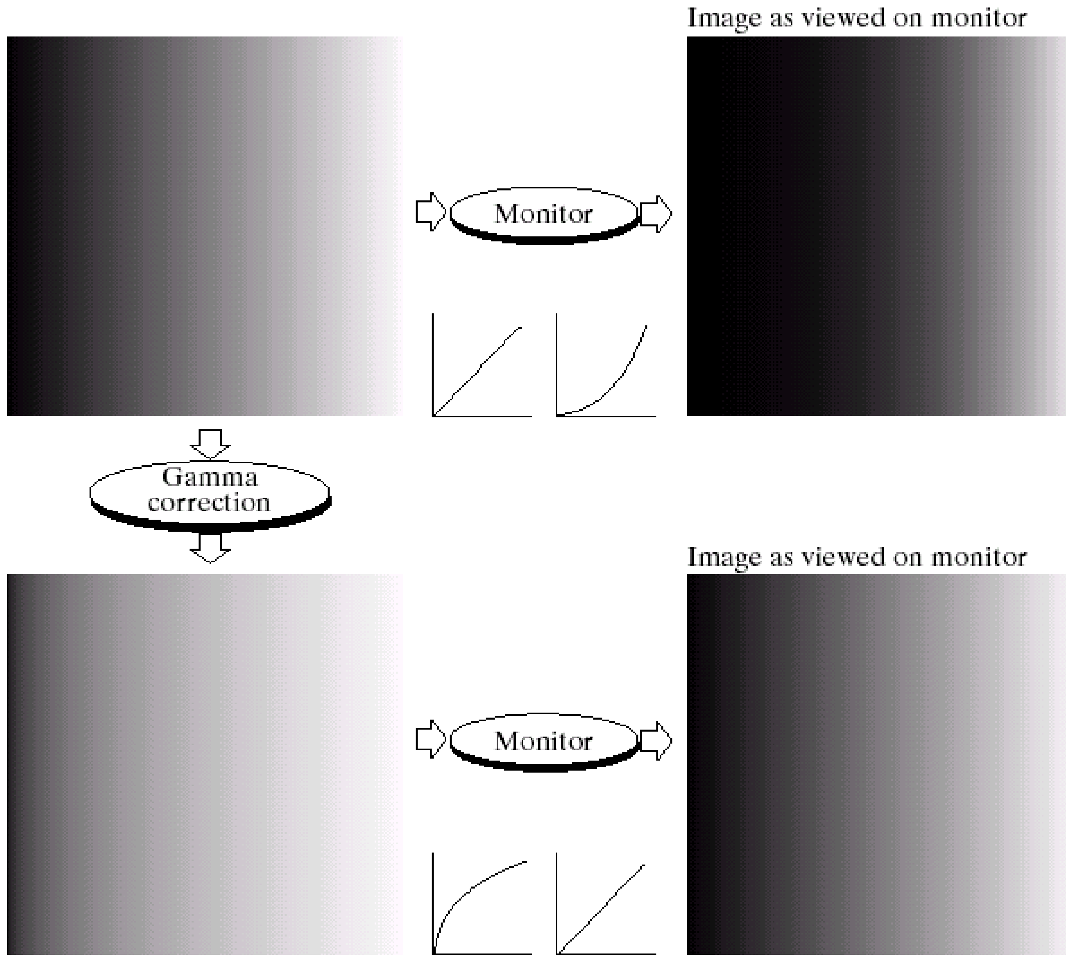


FIGURE 3.7 (a) Linear-wedge gray-scale image. (b) Response of monitor to linear wedge. (c) Gamma-corrected wedge. (d) Output of monitor.

Power Law Transformations
Contrast Enhancement

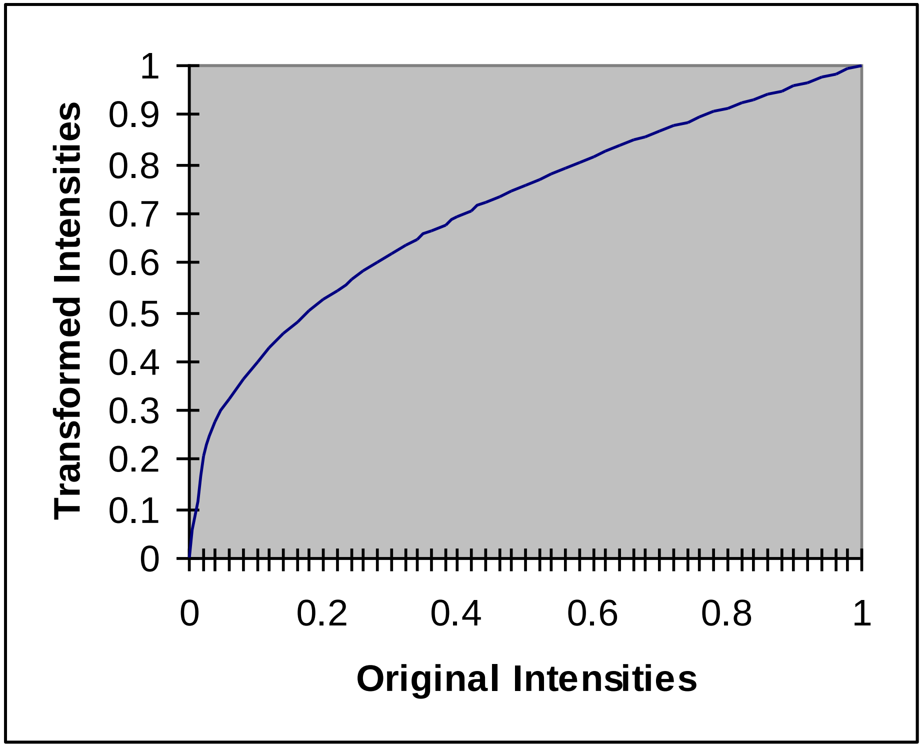
The images to the right show a magnetic resonance (MR) image of a fractured human spine

Power Law Transformations
Contrast Enhancement

γ = 0.6

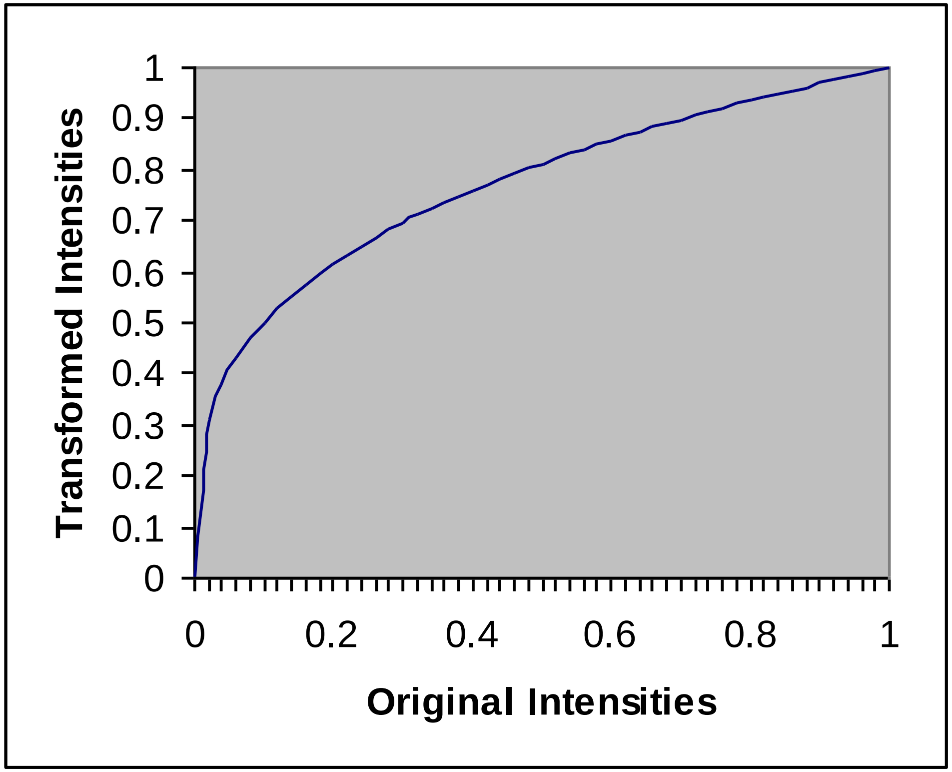
Power Law Transformations
Contrast Enhancement

$\gamma = 0.4$



Power Law Transformations
Contrast Enhancement

γ = 0.3



Power Law Transformations
Contrast Enhancement

MR image of

fractured human spine

Result after

Power law transformation

c = 1, γ = 0.6

Result after

Power law transformation

c = 1, γ = 0.4

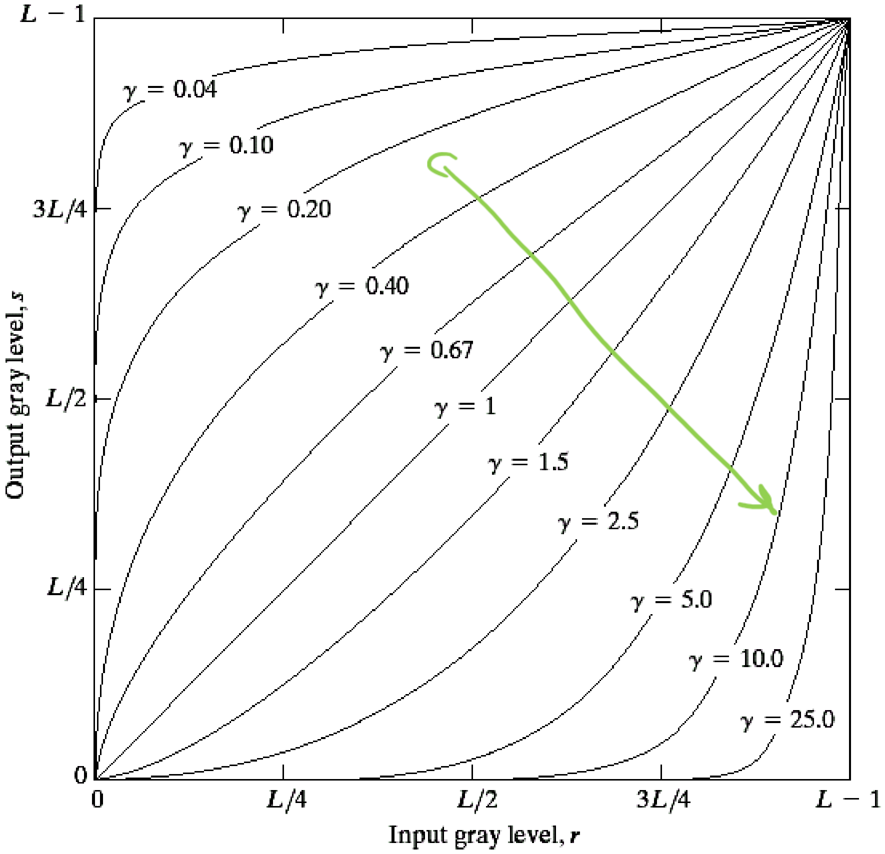
Result after

Power law transformation

c = 1, γ = 0.3

Power Law Transformations
Contrast Enhancement

When the γ is reduced too much, the image begins to reduce contrast to the point where the image started to have very slight "wash-out" look.



Power Law Transformations
Contrast Enhancement

Image has a washed-out appearance – needs $\gamma > 1$

Image Enhancement

Aerial

Image

Result of Power law transformation c = 1, γ = 3.0 (suitable)

Result of Power law transformation c = 1, γ = 4.0 (suitable)

Result of Power law transformation c = 1, γ = 5.0 (high contrast, some regions are too dark)

Histogram of a Grayscale Image

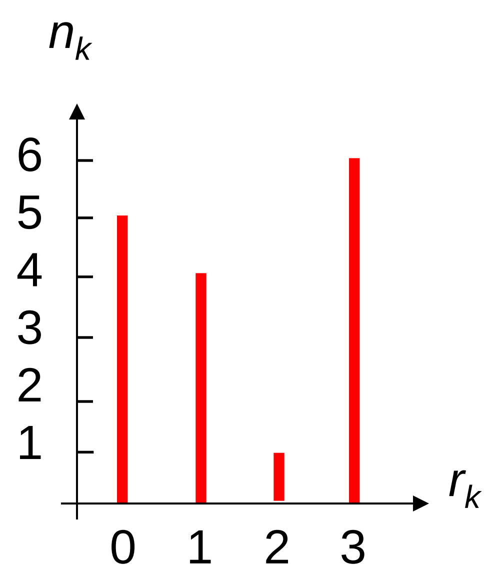
- Let I be a 1-band (grayscale) image.
- $I(r,c)$ is an 8-bit integer between 0 and 255.
- Histogram, h_I, of I:
  - a 256-element array, h_I
  - $h_I(g)$ = number of pixels in I that have value g.

    for $g = 0, 1, 2, 3, \ldots, 255$

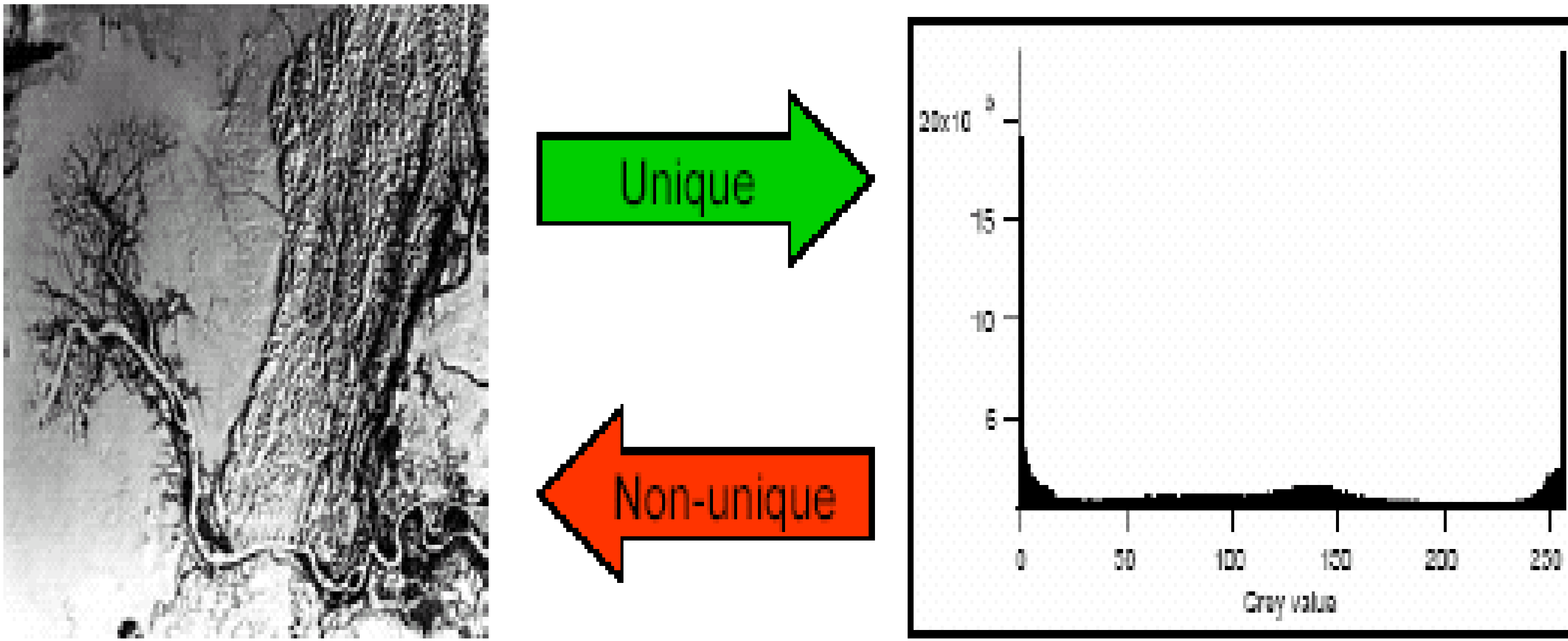
HISTOGRAM

- A discrete function $h(r_k)=n_k$
  - r_k is the kth gray level
  - n_k is the number of pixels having gray level r_k in the image
- Ex:

0	1	2	3
1	3	3	0
0	1	3	0
3	0	3	1



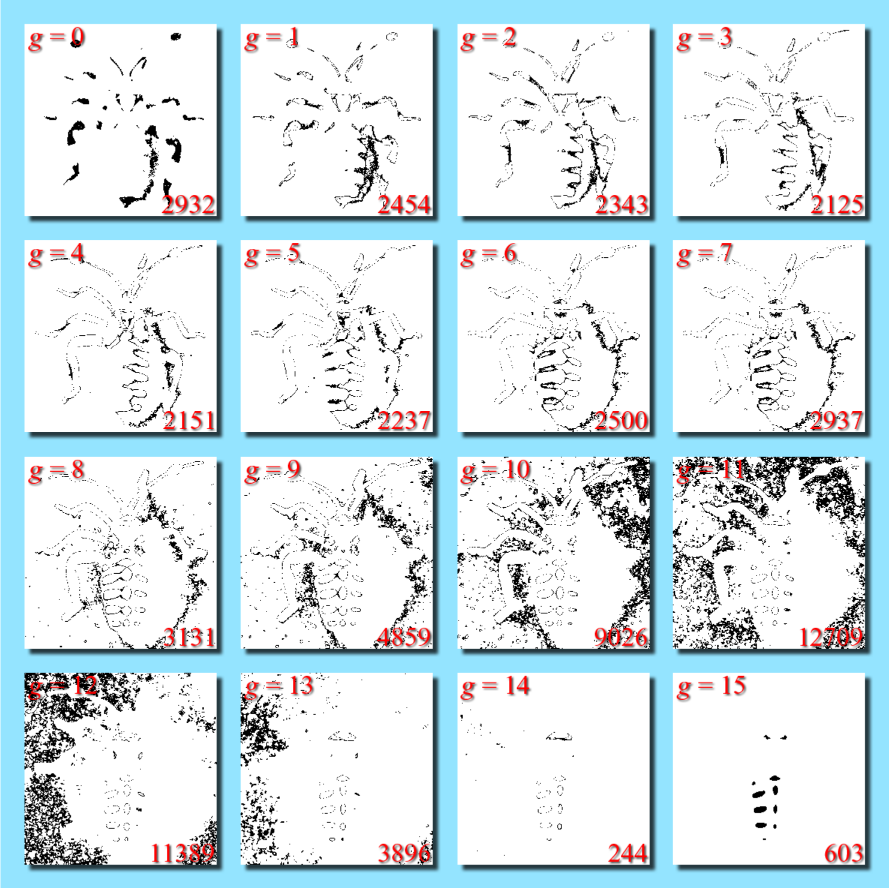
UNIQUENESS



Histogram of a Grayscale Image

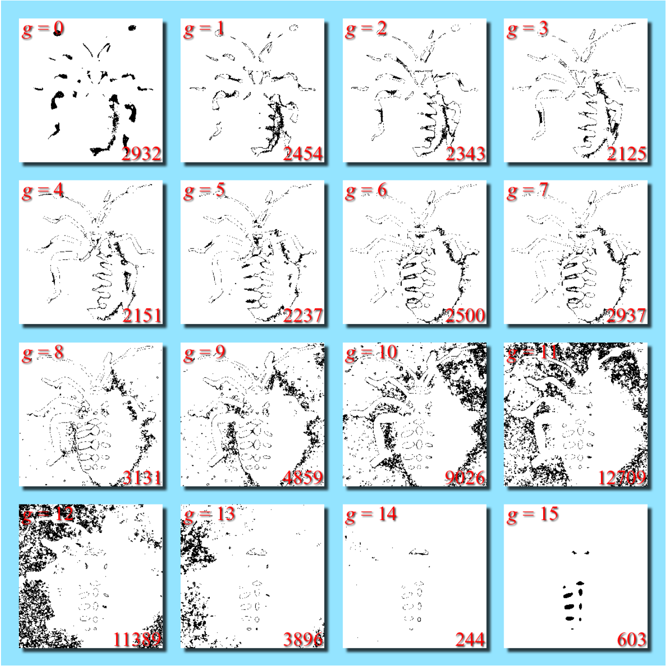
16-level (4-bit) image

lower RHC: number of pixels with intensity g

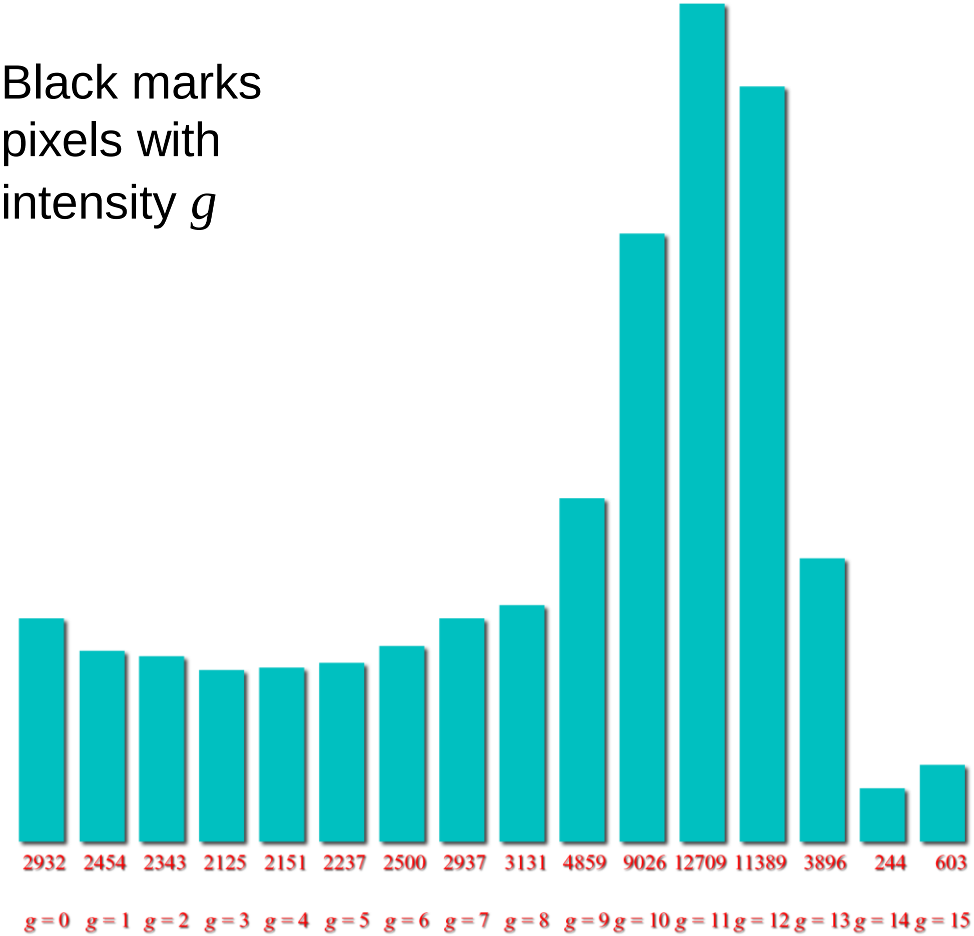


black marks pixels with intensity g

Histogram of a Grayscale Image

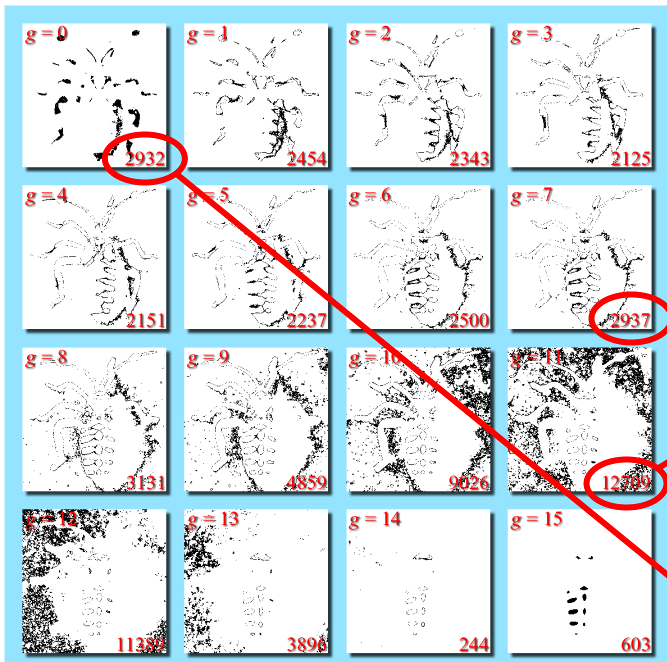


Black marks pixels with intensity g

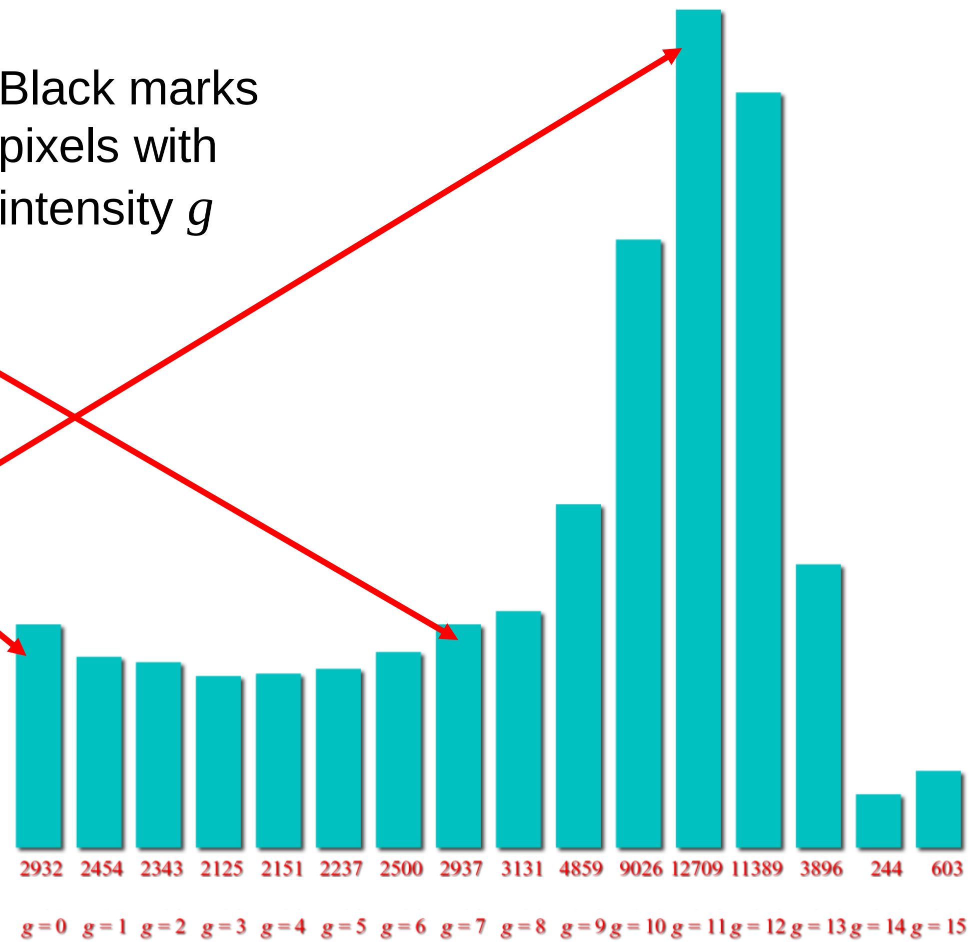


Plot of histogram:
number of pixels with intensity g

Histogram of a Grayscale Image



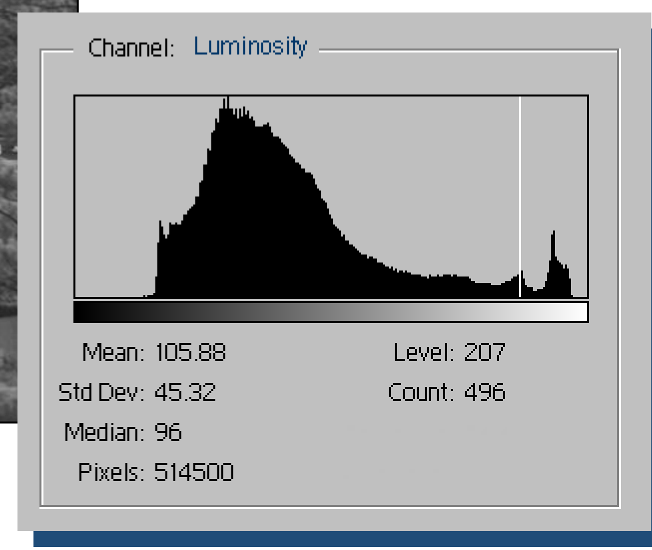
Black marks pixels with intensity g



Plot of histogram:
number of pixels with intensity g

Histogram of a Grayscale Image

$h_I(g) =$ the number of pixels in I with graylevel g.

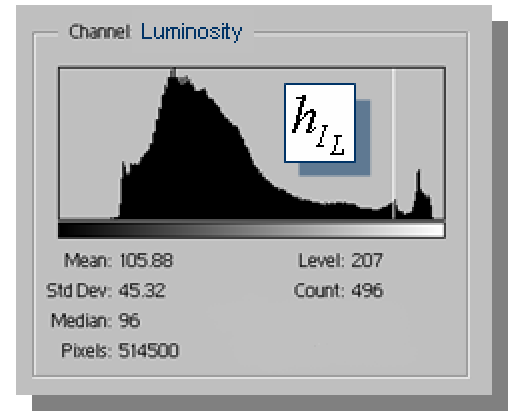


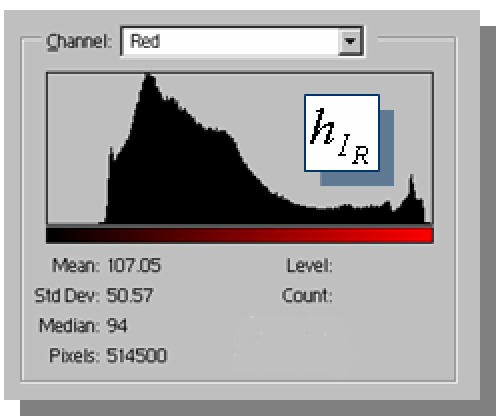
Histogram of a Color Image

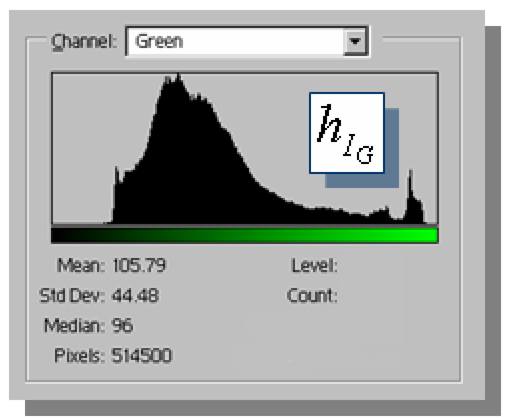
- If I is a 3-band image
- then I(*r*,*c*,*b*) is an integer between 0 and 255.
- I has 3 histograms:
  - $h_R(g)$ = # of pixels in I(:,:,1) with intensity value *g*
  - $h_G(g)$ = # of pixels in I(:,:,2) with intensity value *g*
  - $h_B(g)$ = # of pixels in I(:,:,3) with intensity value *g*

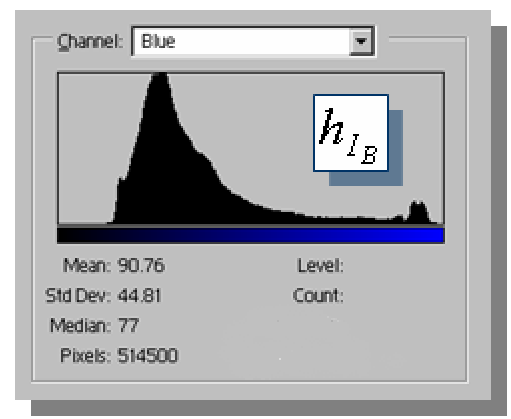
Histogram of a Color Image

There is one histogram per color band R, G, & B. Luminosity histogram is from 1 band = (R+G+B)/3

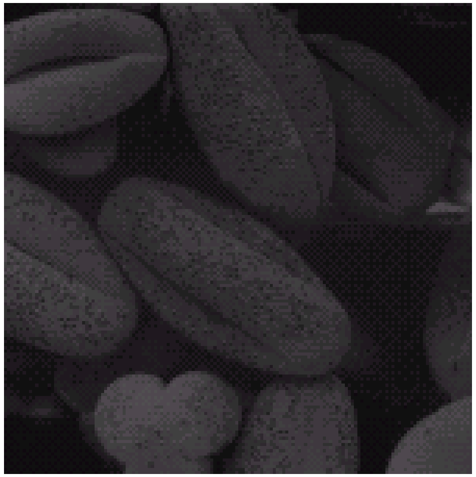








Histogram: Example

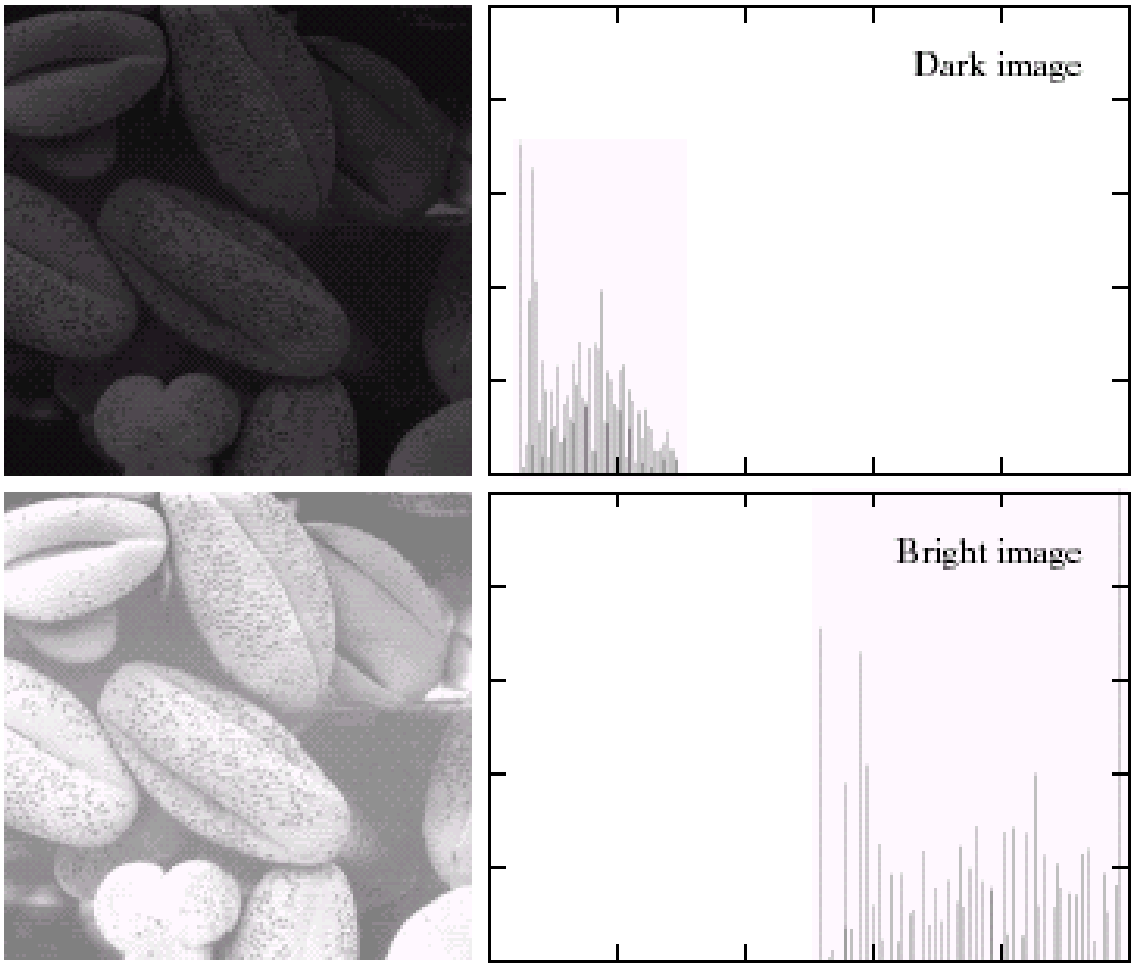


Dark Image

How would the histograms of these images look like?

Bright Image

Histogram: Example

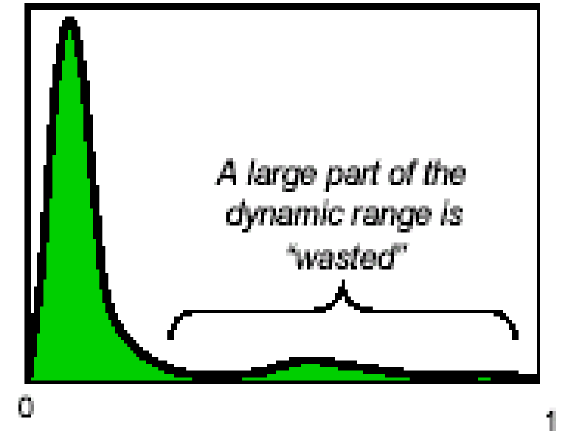


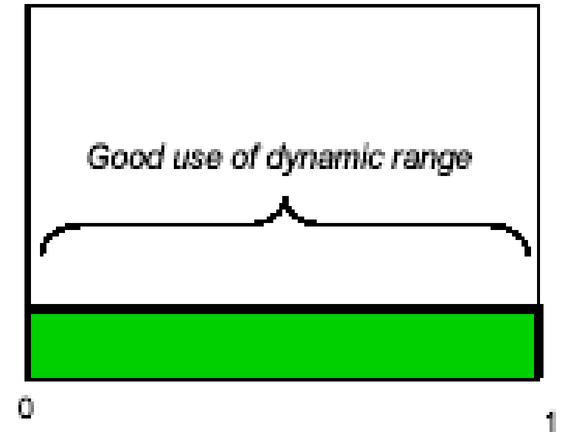
Dark image
Components of histogram are concentrated on the low side of the gray scale

Bright image
Components of histogram are concentrated on the high side of the gray scale

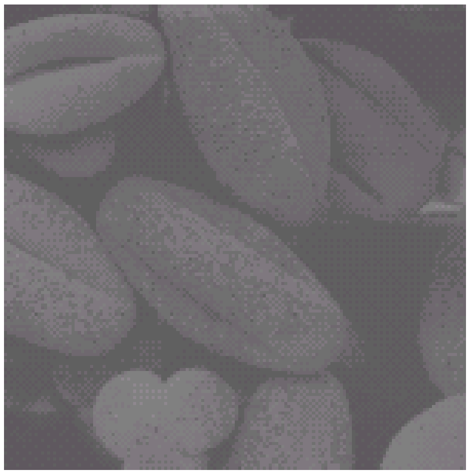
HISTOGRAM INSIGHT INTO CONTRAST

- A high contrast image makes good use of the full dynamic range available.
- Hence in some applications it may be desirable to make more optimal use of the full dynamic range.
- In some circumstances this results in a clearer image.





Histogram: Example

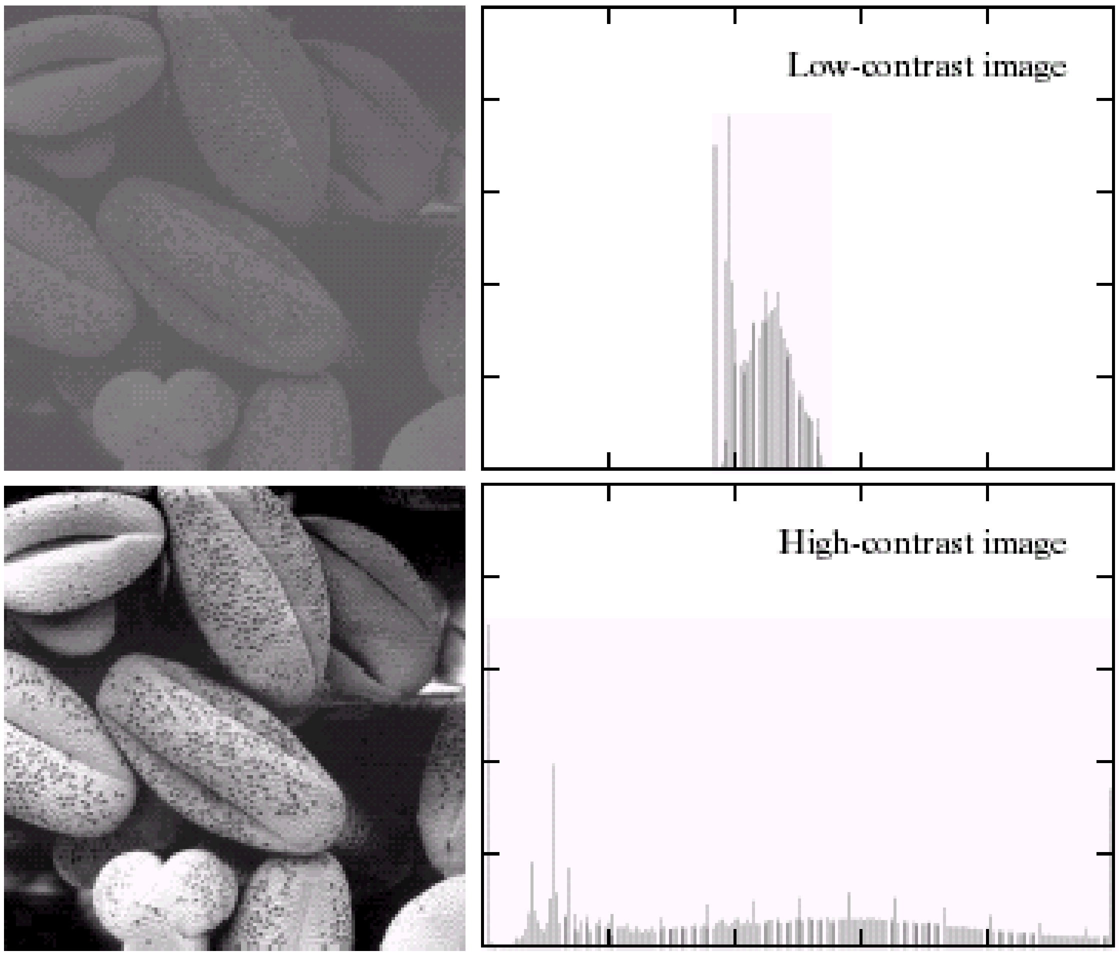


Low Contrast Image

How would the histograms of these images look like?

High Contrast Image

Histogram: Example



Low contrast image
Histogram is narrow and centered toward the middle of the gray scale

High contrast image
Histogram covers broad range of the gray scale and the distribution of pixels is not too far from uniform with very few vertical lines being much higher than the others

Readings from Book (4^{th} Edn.)

- Chapter – 3
  - 3.1
  - 3.2

Acknowledgements

Material in these slides has been taken from, the following resources

- Statistical Pattern Recognition: A Review – A.K Jain et al., PAMI (22) 2000
- Pattern Recognition and Analysis Course – A.K. Jain, MSU
- *Pattern Classification”* by Duda et al., John Wiley & Sons.
- Digital Image Processing”, Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Machine Vision: Automated Visual Inspection and Robot Vision”, David Vernon, Prentice Hall, 1991
- www.eu.aibo.com/
- Advances in Human Computer Interaction, Shane Pinder, InTech, Austria, October 2008