Digital Image Processing

Lecture # 3
Image Enhancement

Image Enhancement

Image Enhancement

Image Enhancement

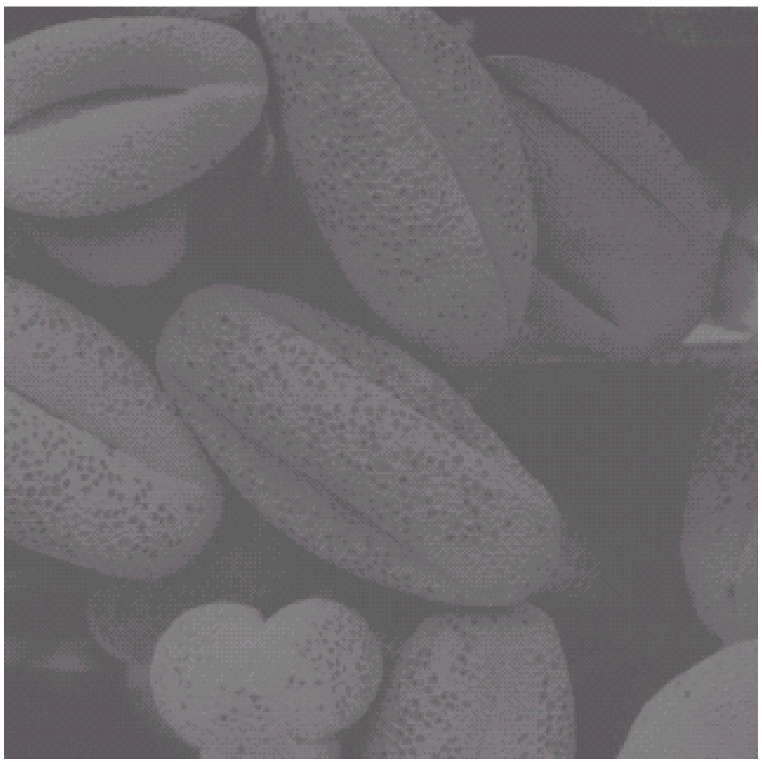


Image Enhancement

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

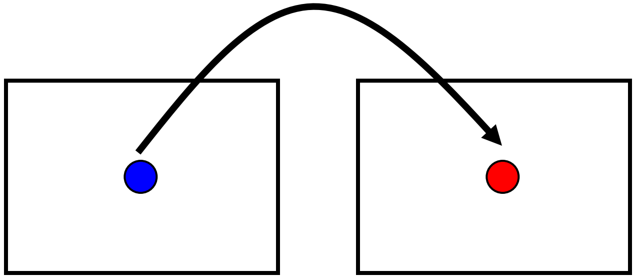
- Image Enhancement Methods
  - Spatial Domain: Direct manipulation of pixels in an image
  - Frequency Domain: Process the image by modifying the Fourier transform of an image

This Chapter – Spatial Domain

Types of image enhancement operations

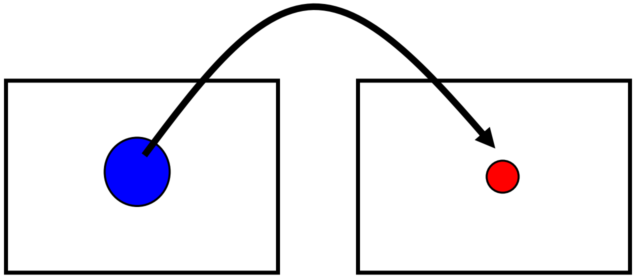
- Point/Pixel operations

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



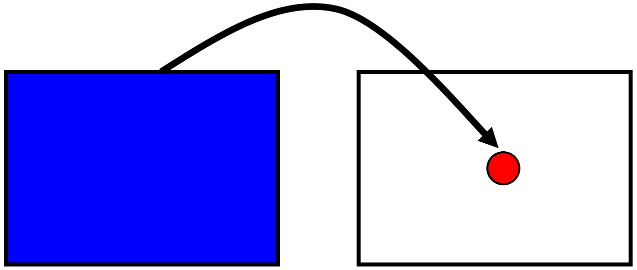
- Local operations

  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:

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

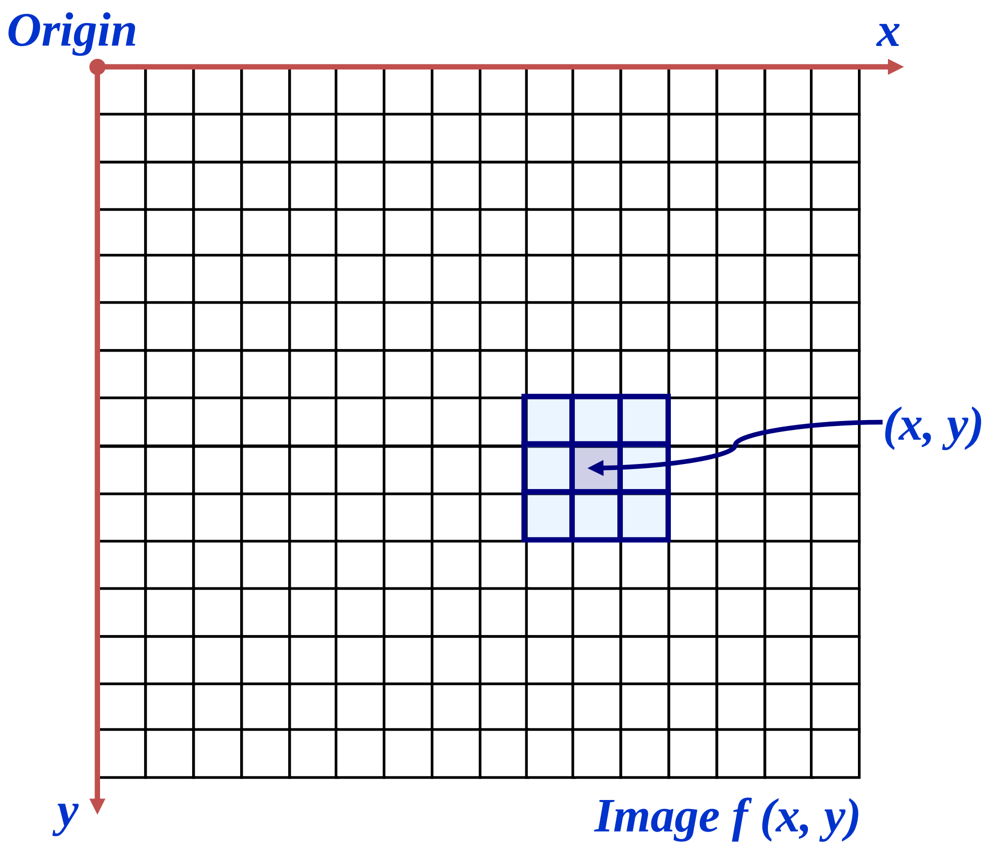
$f(x, y)$ = the input image

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

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

Basic Concepts

A square or rectangular sub-image area centered at (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

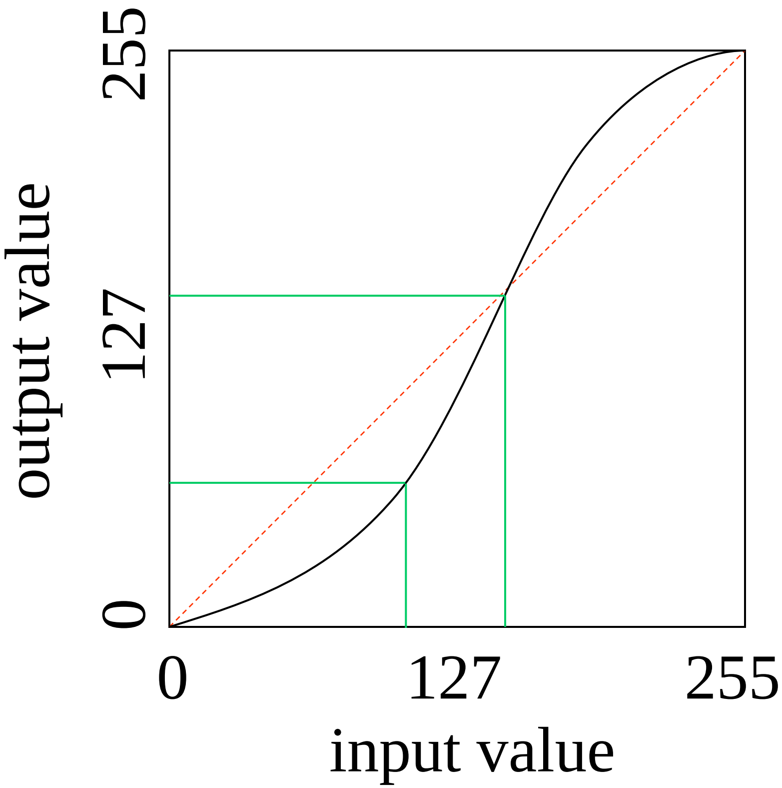
- Neighborhood of size 1x1:
- *g* depends only on *f* at *(x,y)*
- *T*: Gray-level/intensity transformation/ mapping function

$$s = T(r)$$

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

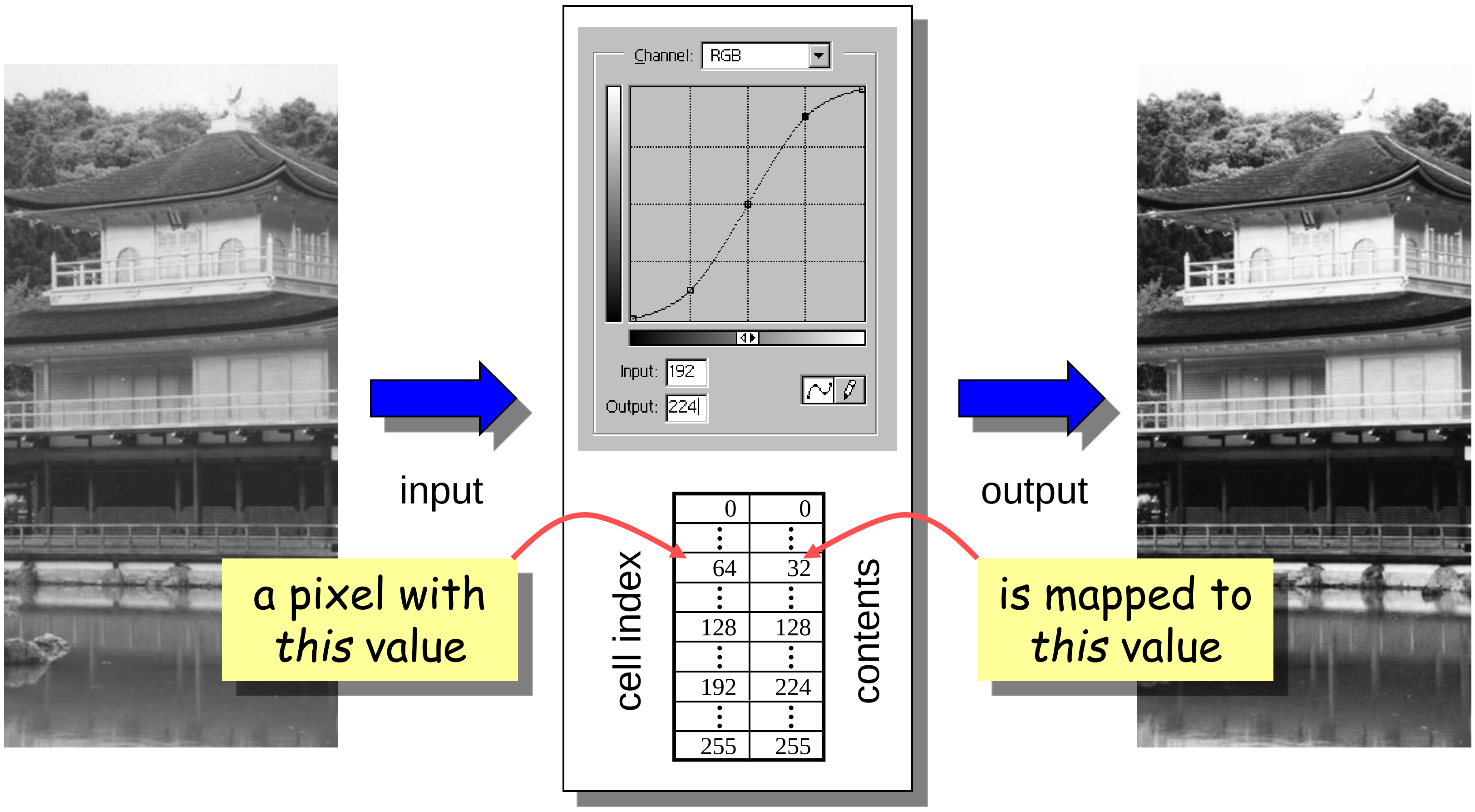
Point Processing using Look-up Tables

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

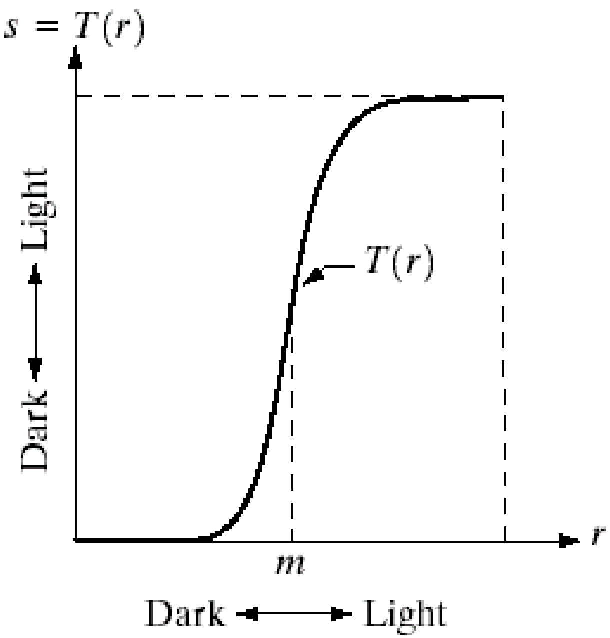


E.g.:	index	value
	...	...
	101	64
	102	68
	103	69
	104	70
	105	70
	106	71
	...	...
	input	output

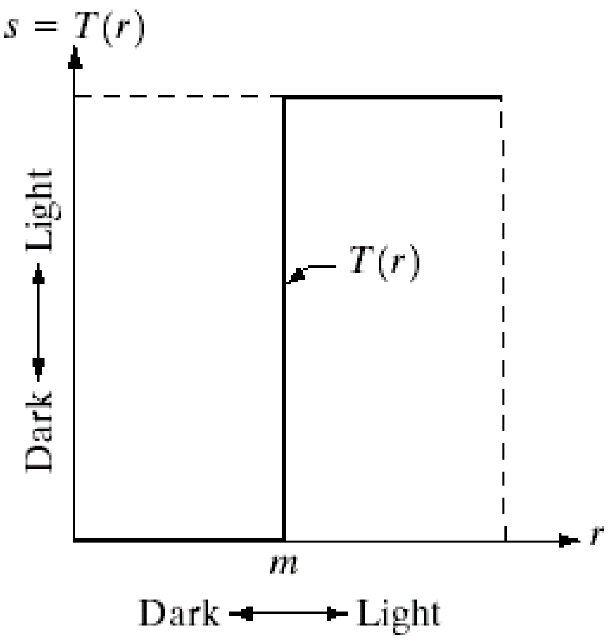
Point Processing using Look-up Tables



POINT PROCESSING



Contrast Stretching



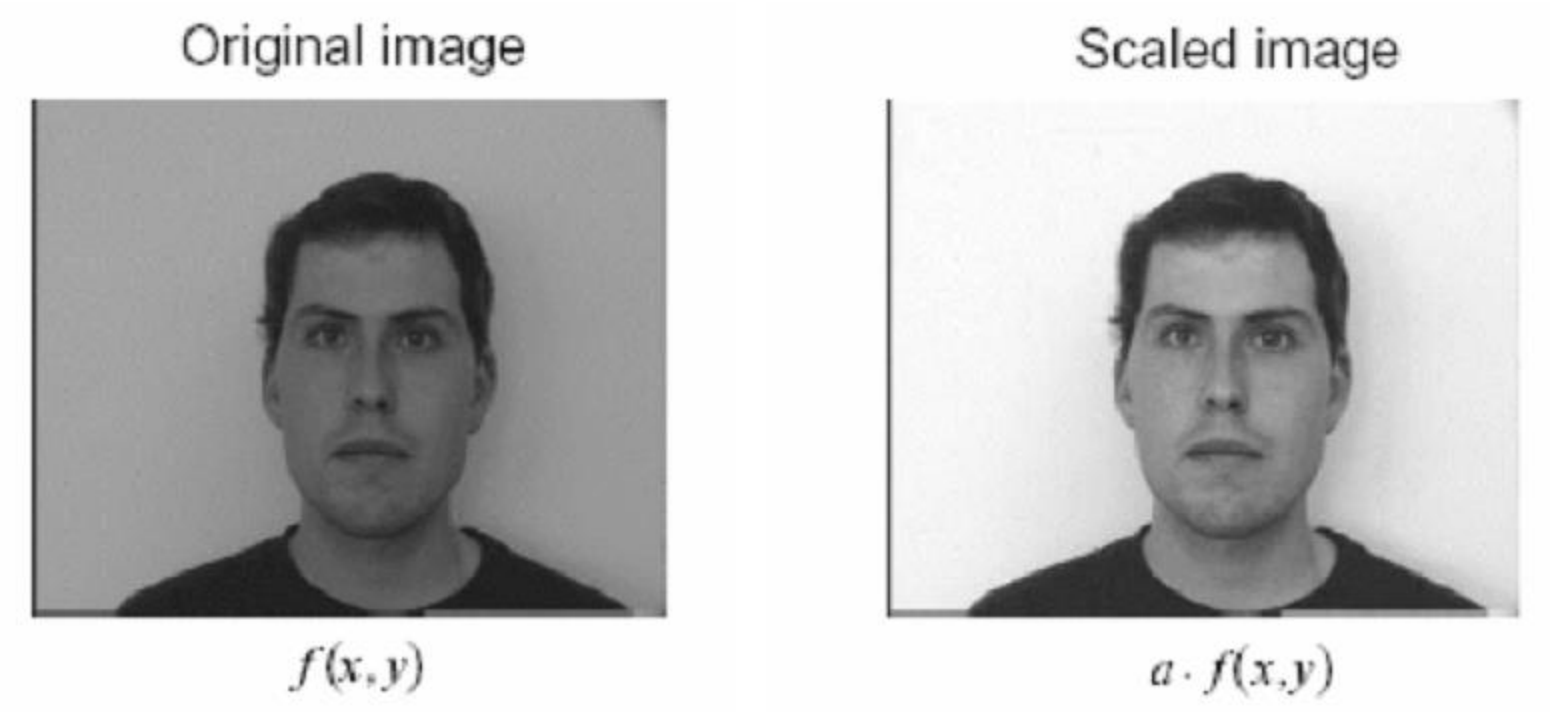
Thresholding

a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

Point Processing Example: Intensity Scaling

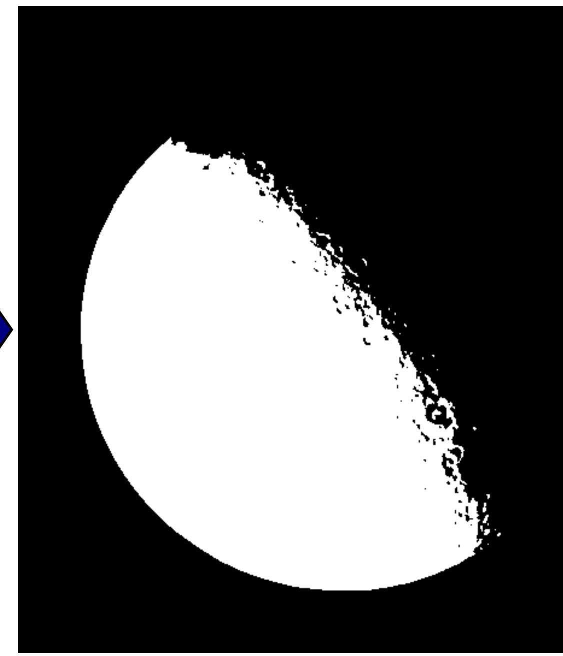
$$s = T(r) = a.r$$



Point Processing Example: Thresholding

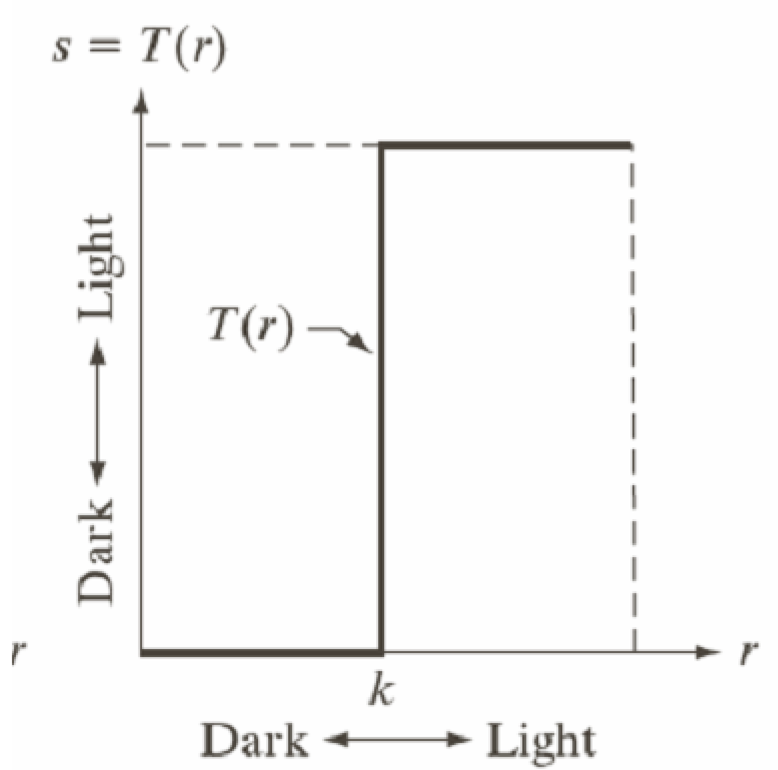
- Segmentation of an object of interest from a background

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



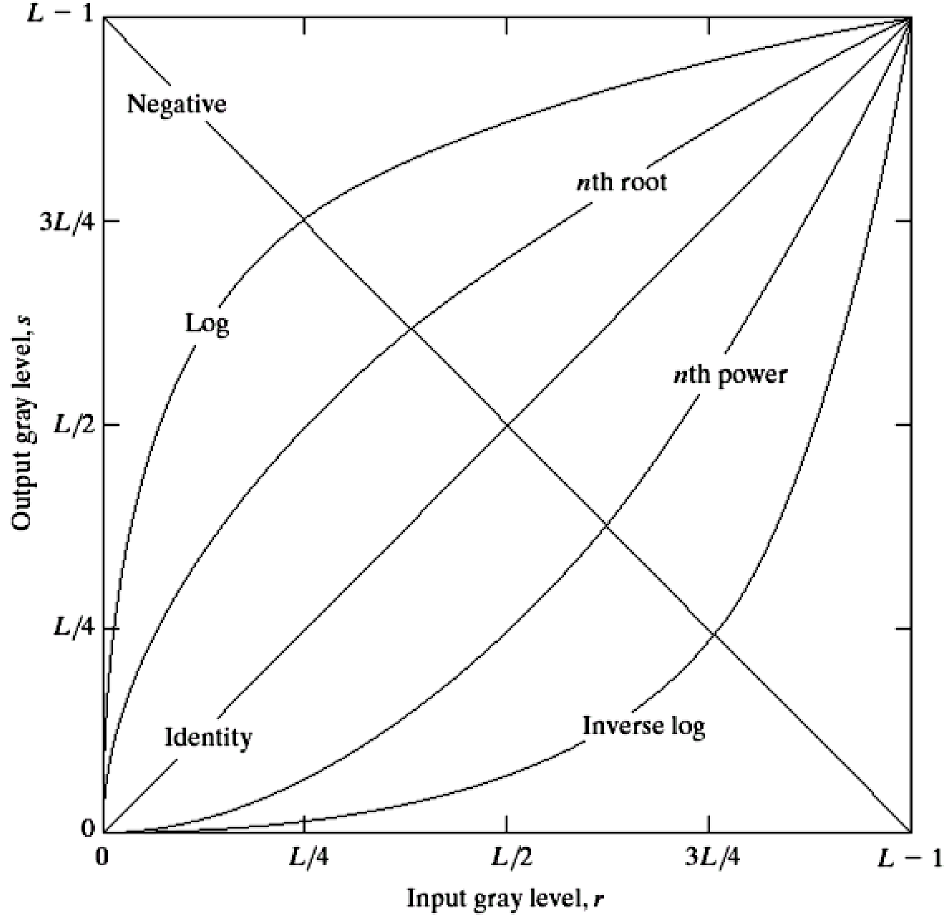
Point Processing Example: Thresholding

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



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)

Logarithmic Transformations

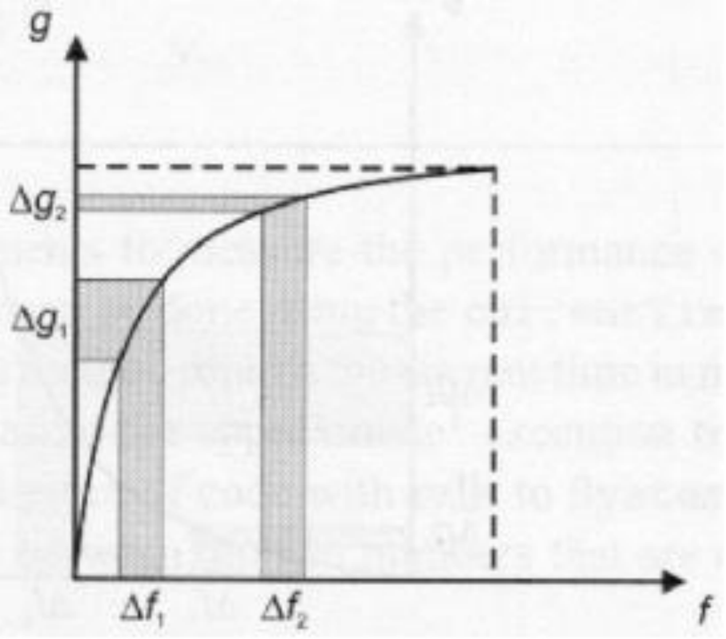
- The general form of the log transformation is

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

- 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

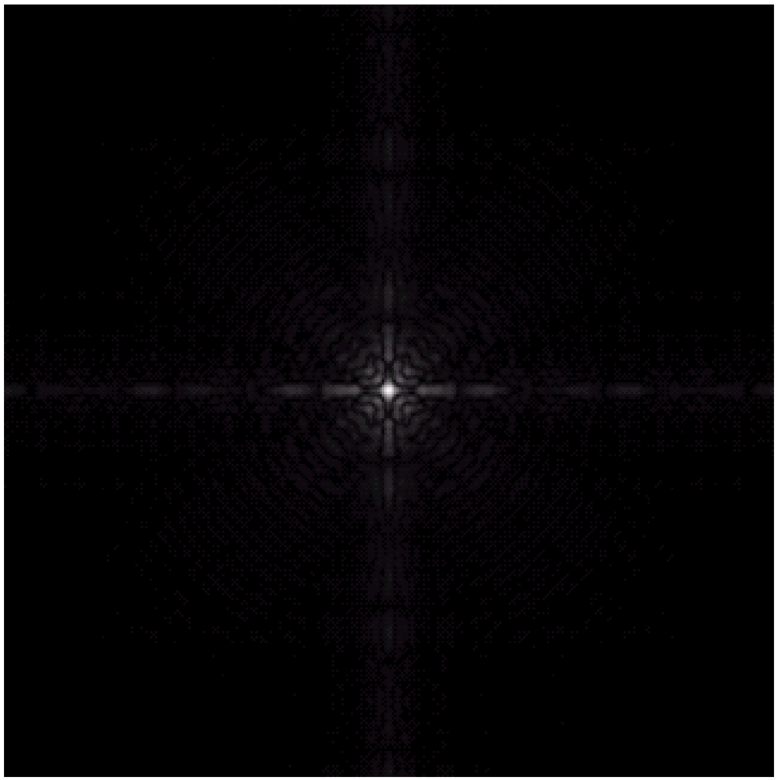
- **Properties**
  - For lower amplitudes of input image the range of gray levels is expanded
  - For higher amplitudes of input image the range of gray levels is compressed



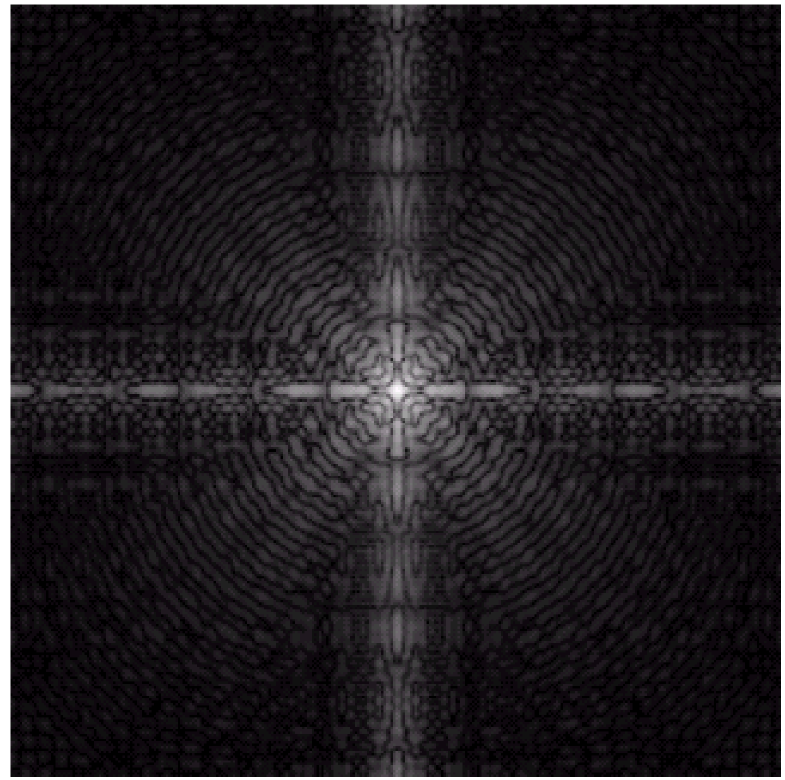
Logarithmic Transformations

- **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



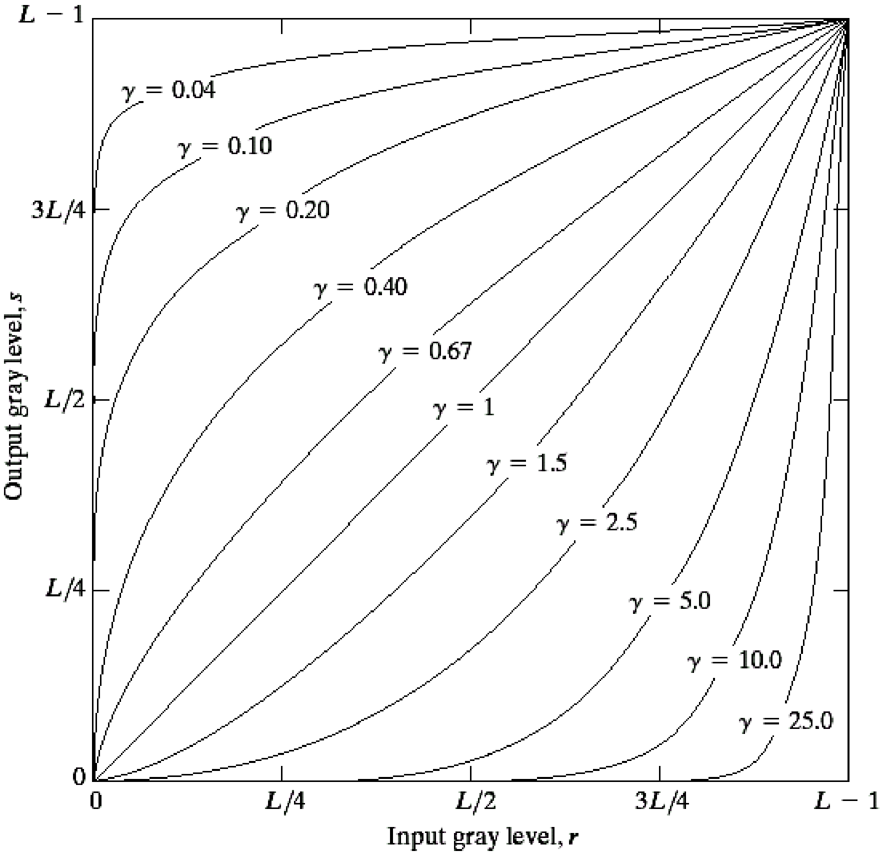
The result of log transformation with c = 1

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

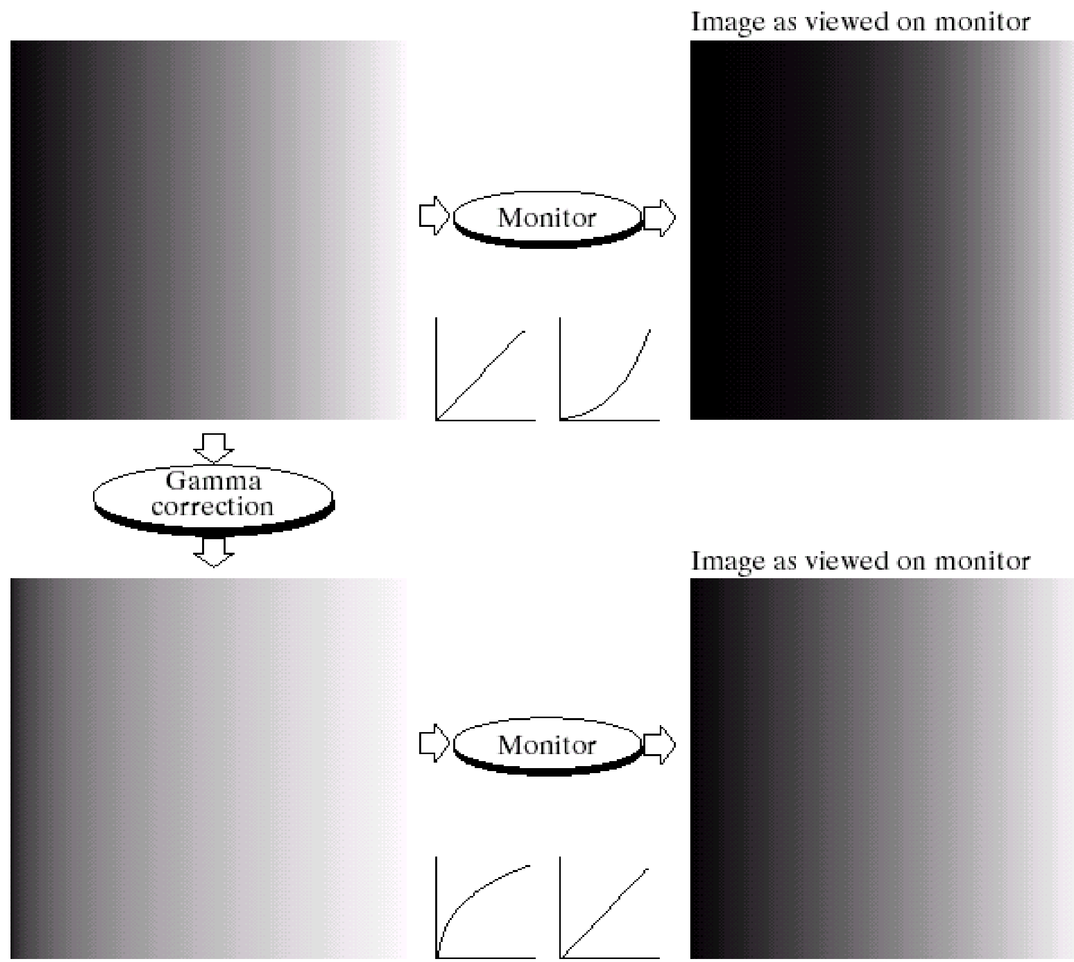
- For $\gamma < 1$: Expands values of dark pixels, compress values of brighter pixels
- For $\gamma > 1$: Compresses values of dark pixels, expand values of brighter pixels
- If $\gamma=1$ & $c=1$: Identity transformation ($s = r$)

- 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



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 Transformations
Contrast Enhancement

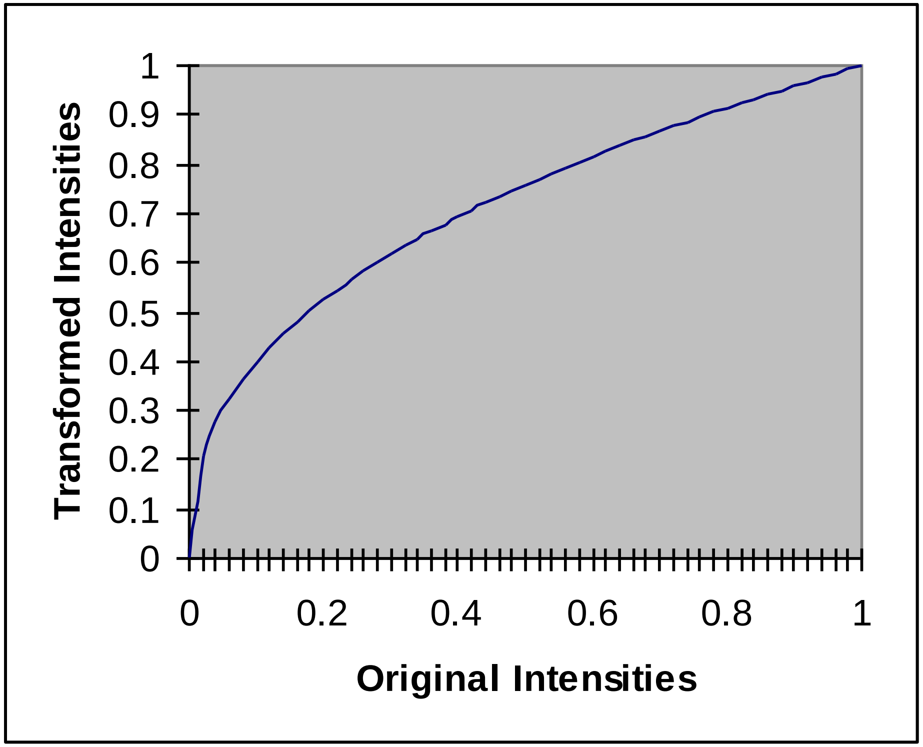
The images to the right show a magnetic resonance (MR) image of a fractured human spine

Power Law Transformations
Contrast Enhancement

γ = 0.6

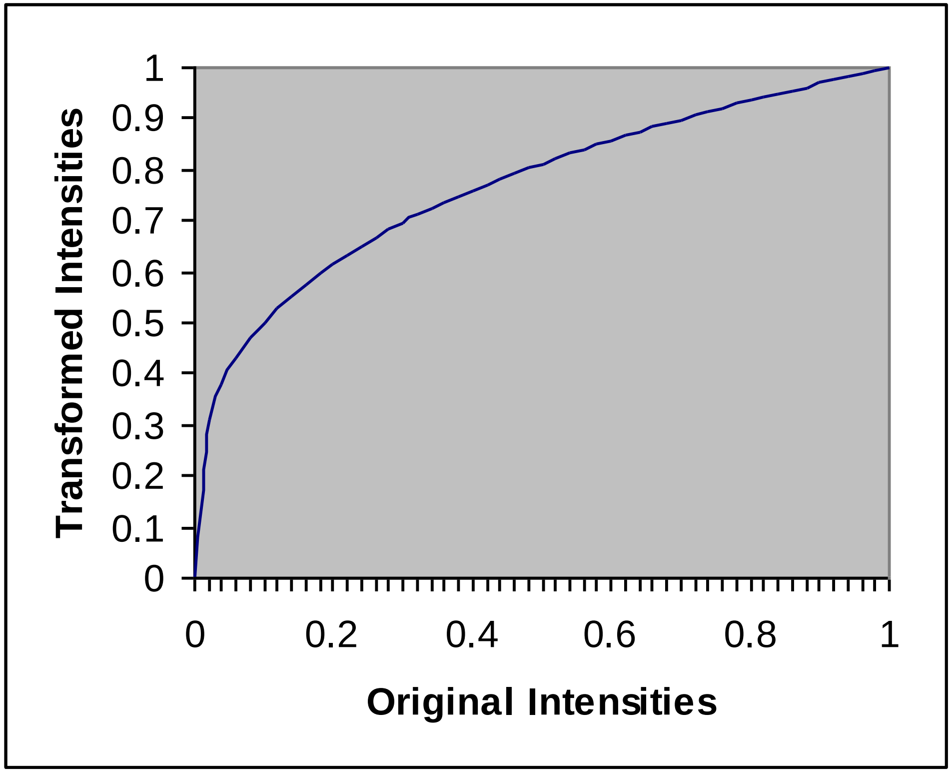
Power Law Transformations
Contrast Enhancement

$\gamma = 0.4$



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

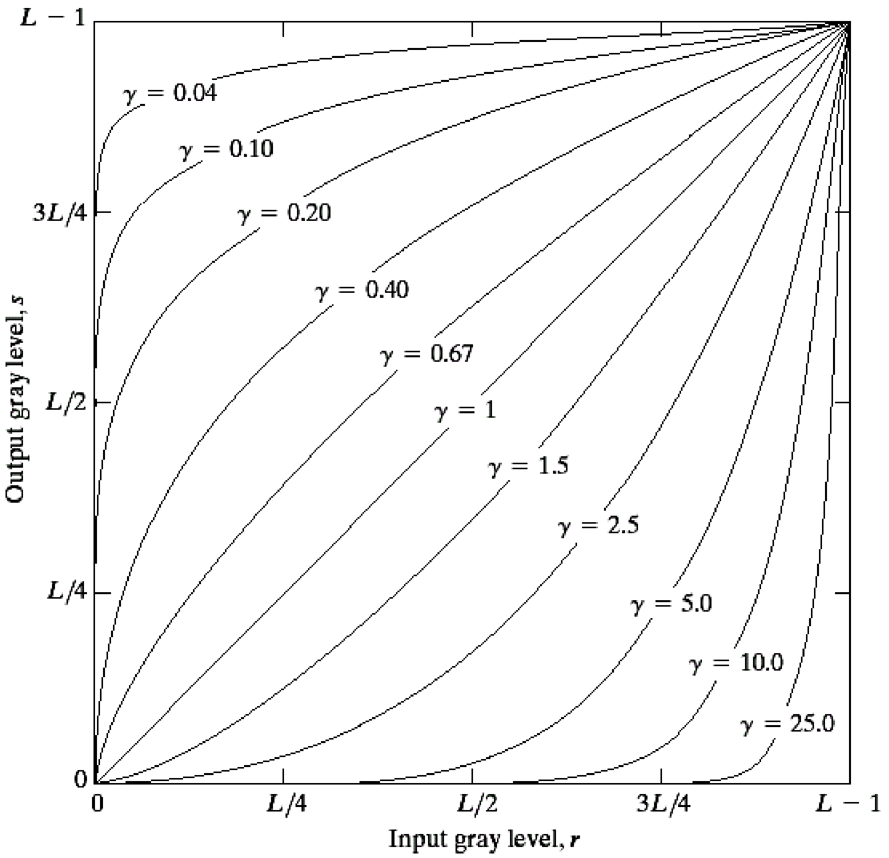
Result after

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, $\gamma = 3.0$ (suitable)

Result of Power law transformation c = 1, $\gamma = 4.0$ (suitable)

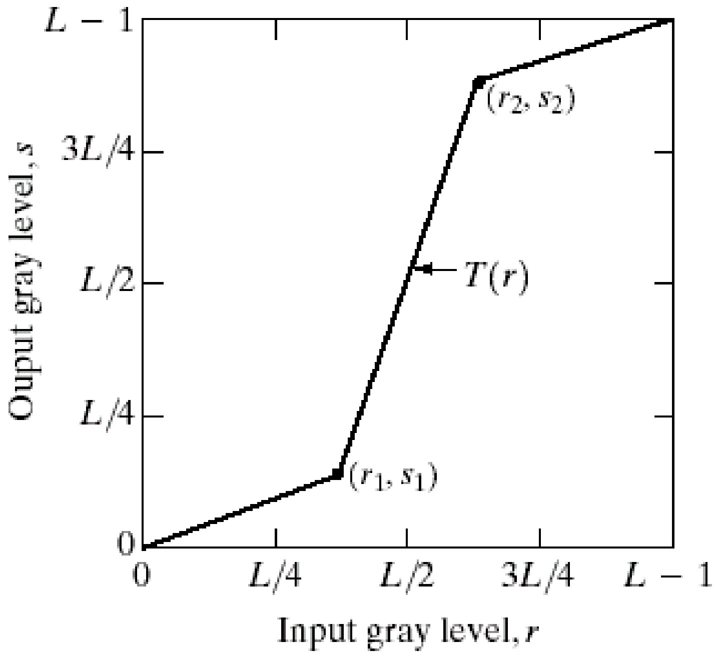
Result of Power law transformation c = 1, $\gamma = 5.0$ (high contrast, some regions are too dark)

Piecewise Linear Transformation Functions

- Contrast stretching
- Intensity level slicing

Contrast Stretching

- Objective
  - Increase the dynamic range of the gray levels for low contrast images
- Rather than using a well defined mathematical function we can use arbitrary user-defined transforms

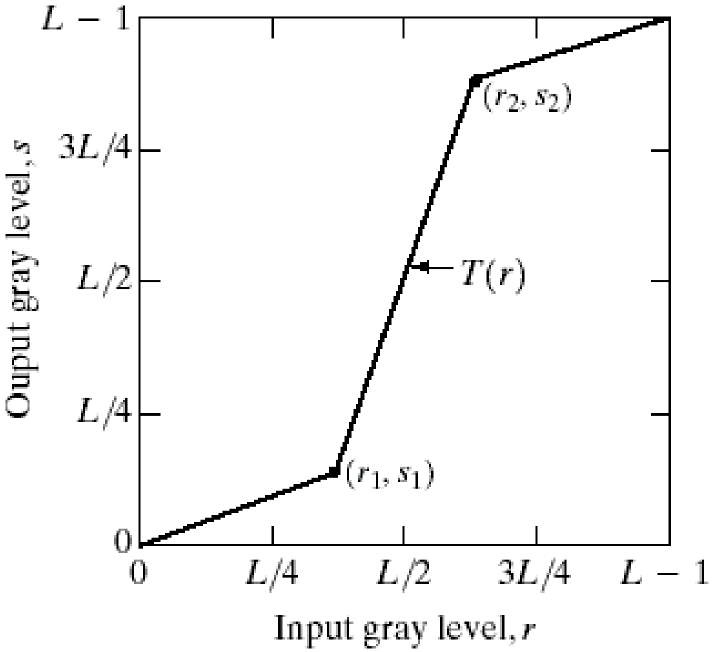


- If $r_1 = s_1$ & $r_2 = s_2$, no change in gray levels
- If $r_1 = r_2$, $s_1 = 0$ & $s_2 = L-1$, then it is a threshold function. The resulting image is binary

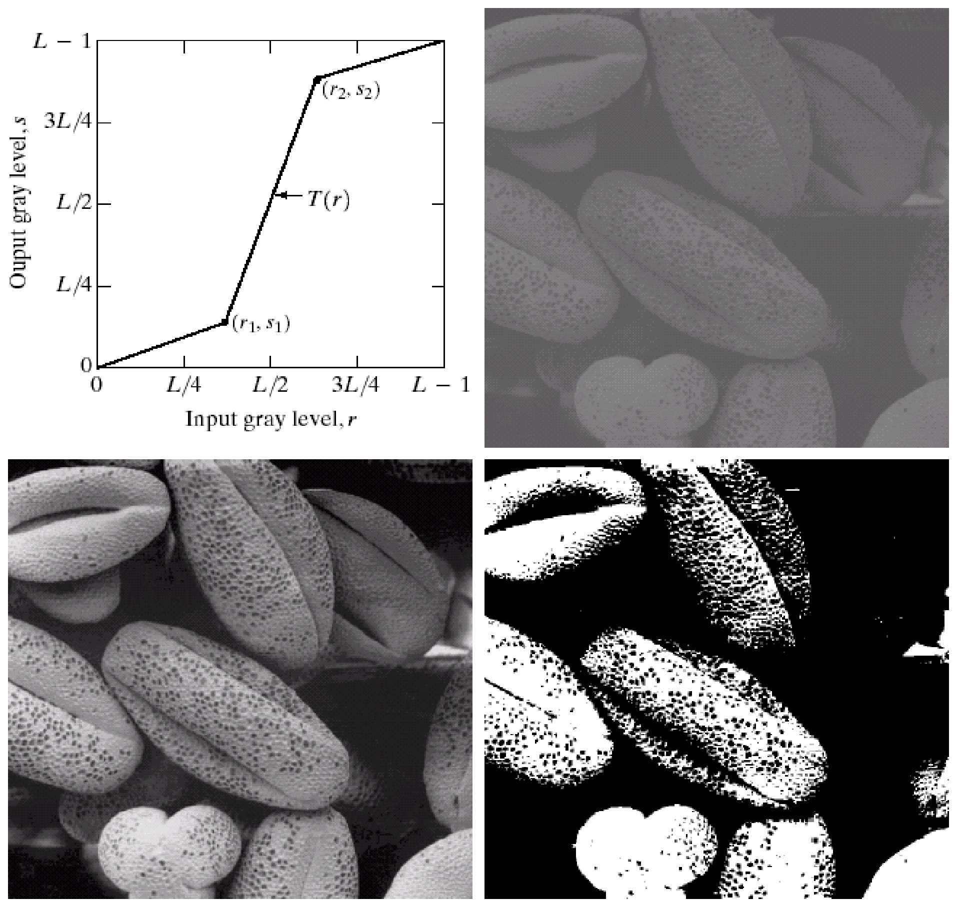
Contrast Stretching

$r_1 = r_{min}$ & $s_1 = 0$

$r_2 = r_{max}$ & $s_2 = L-1$



Contrast Stretching



a b
c d

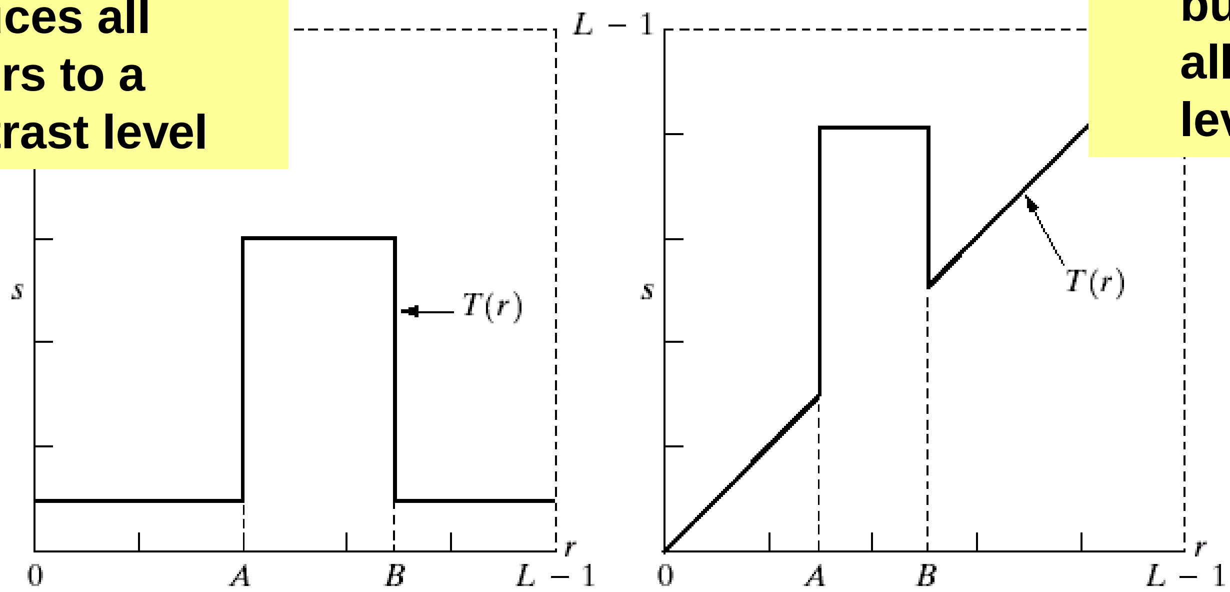
FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Grey Level Slicing

- Highlights a specific range of gray levels in an image
  - Similar to thresholding
  - Other levels can be suppressed or maintained
  - Useful for highlighting features in an image

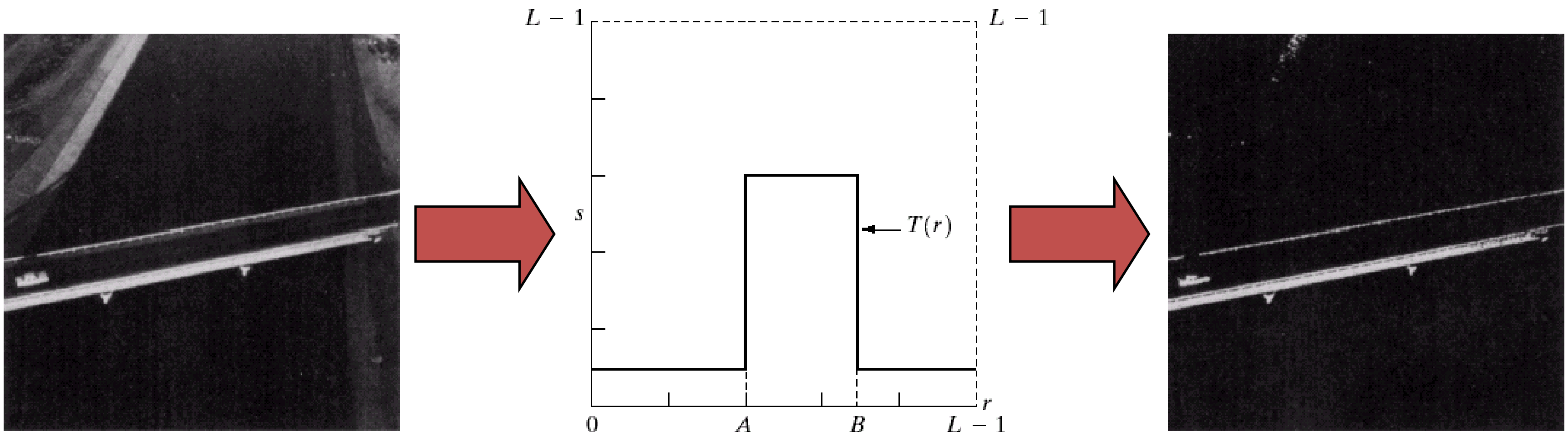
Grey Level Slicing

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



Histogram of a Grayscale Image

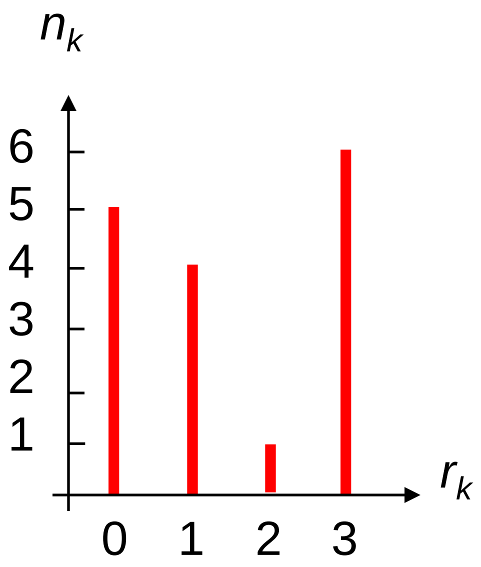
- Let *I* be a 1-band (grayscale) image.
- $I(r,c)$ is an 8-bit integer between 0 and 255.
- Histogram, h_I, of I:
  - a 256-element array, h_I
  - $h_I(g)$ = number of pixels in I that have value g.

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

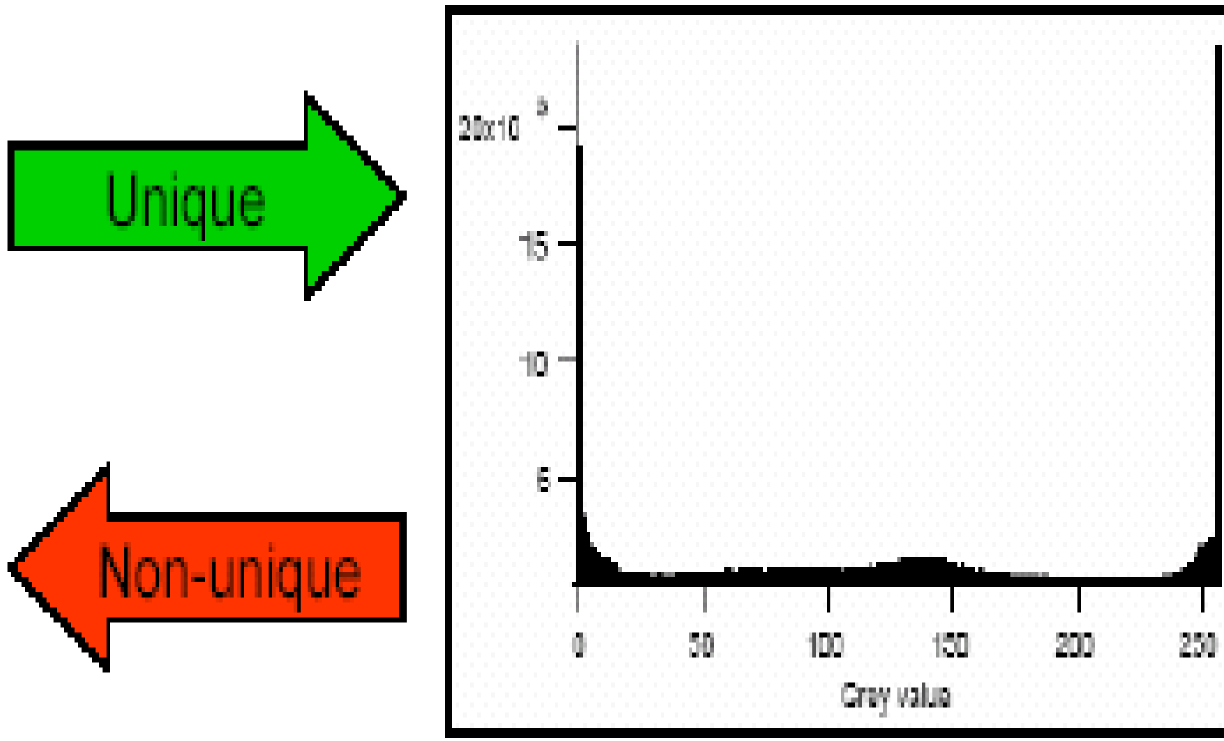
HISTOGRAM

- 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

- 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 = k^{th} gray level
- n_k = number of pixels in the image having gray level r_k
- $h(r_k)$ = histogram of an image having r_k gray levels

Normalized Histogram

- Dividing each of histogram at gray level r_k by the total number of pixels in the image, n

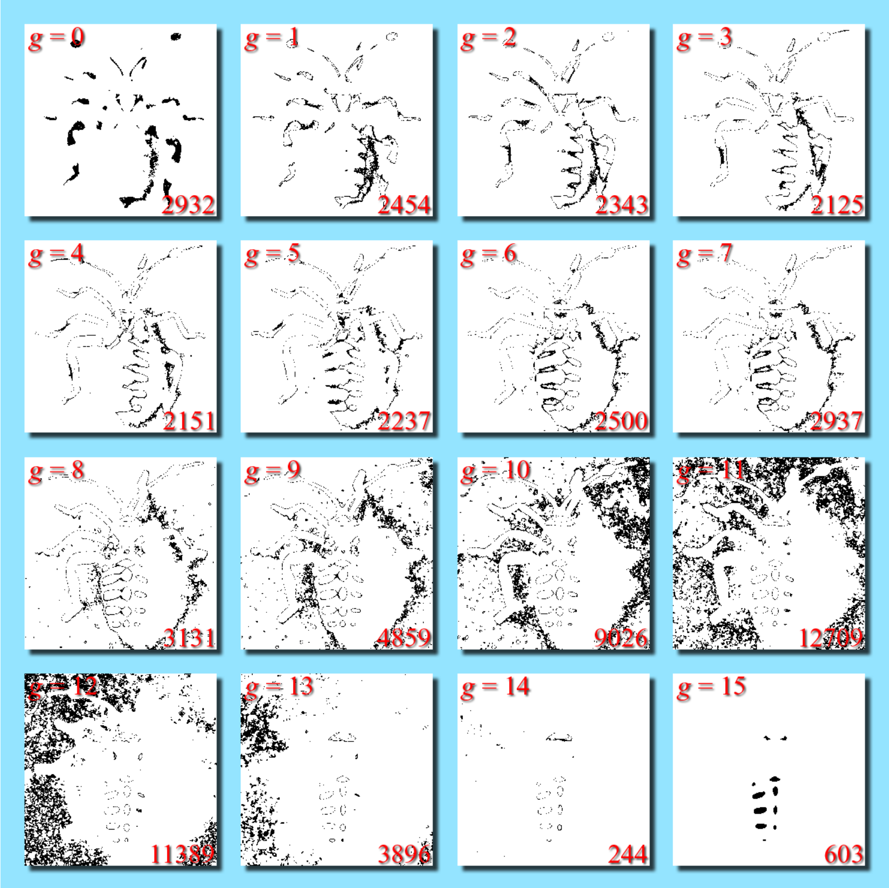
$$p(r_k) = n_k / n \quad for\ k = 0, 1, \ldots, L-1$$

- $p(r_k)$ gives an estimate of the probability of occurrence of gray level r_k
- The sum of all components of a normalized histogram is equal to 1

Histogram of a Grayscale Image

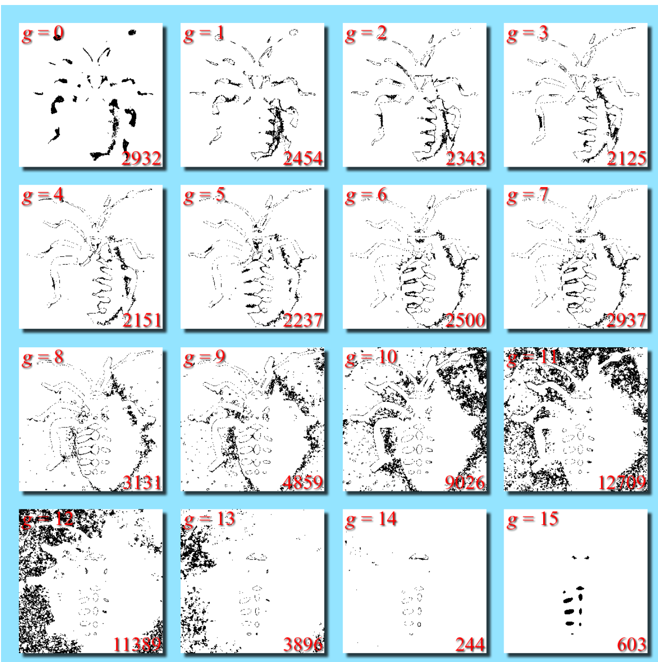
16-level (4-bit) image

lower RHC: number of pixels with intensity *g*

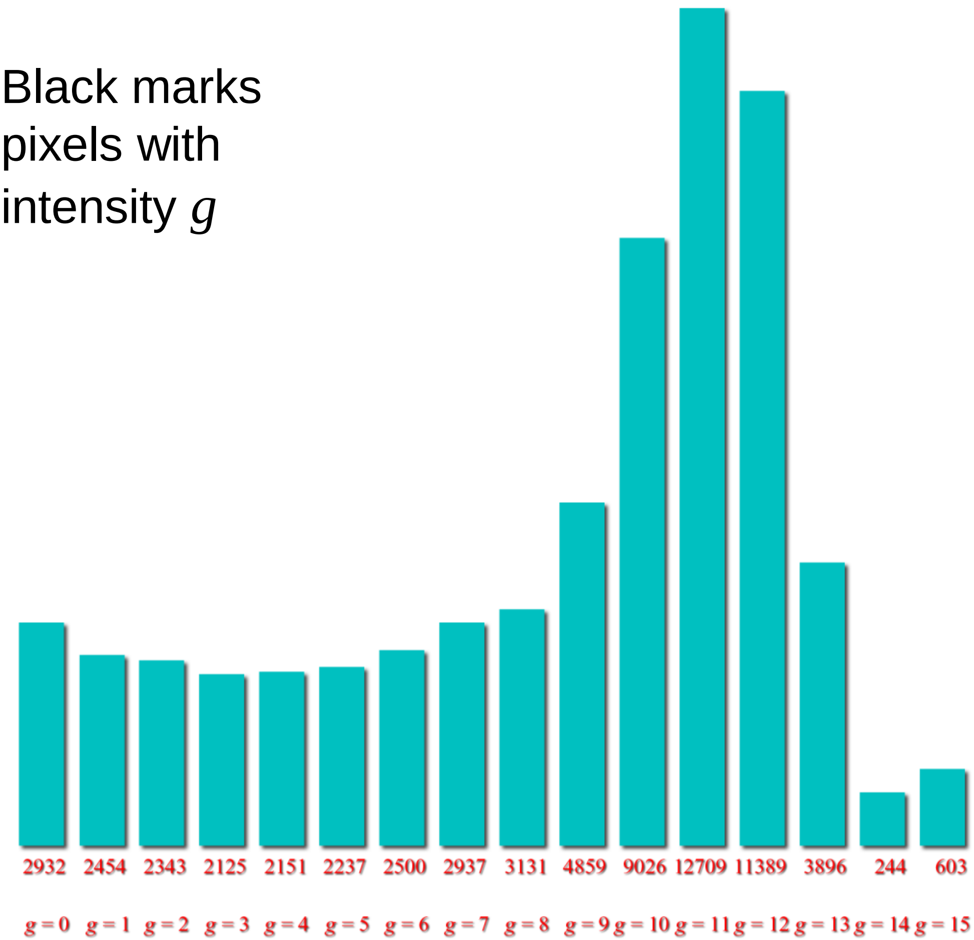


black marks pixels with intensity *g*

Histogram of a Grayscale Image

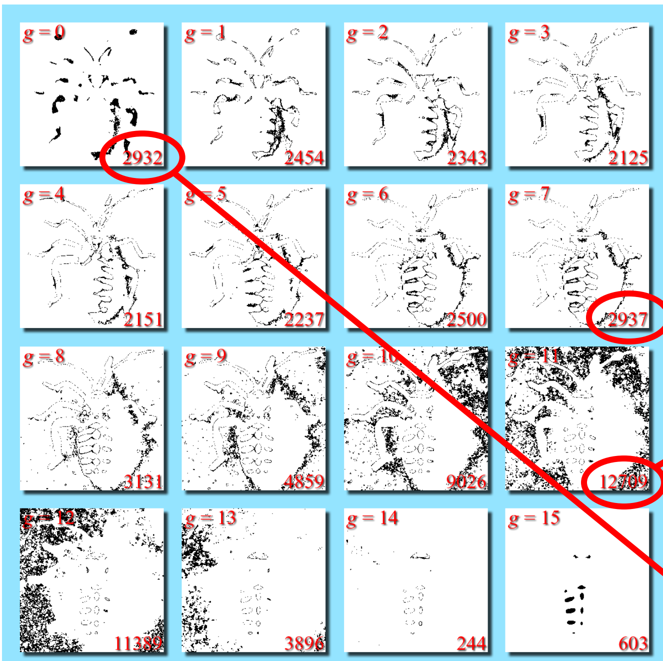


Black marks pixels with intensity g

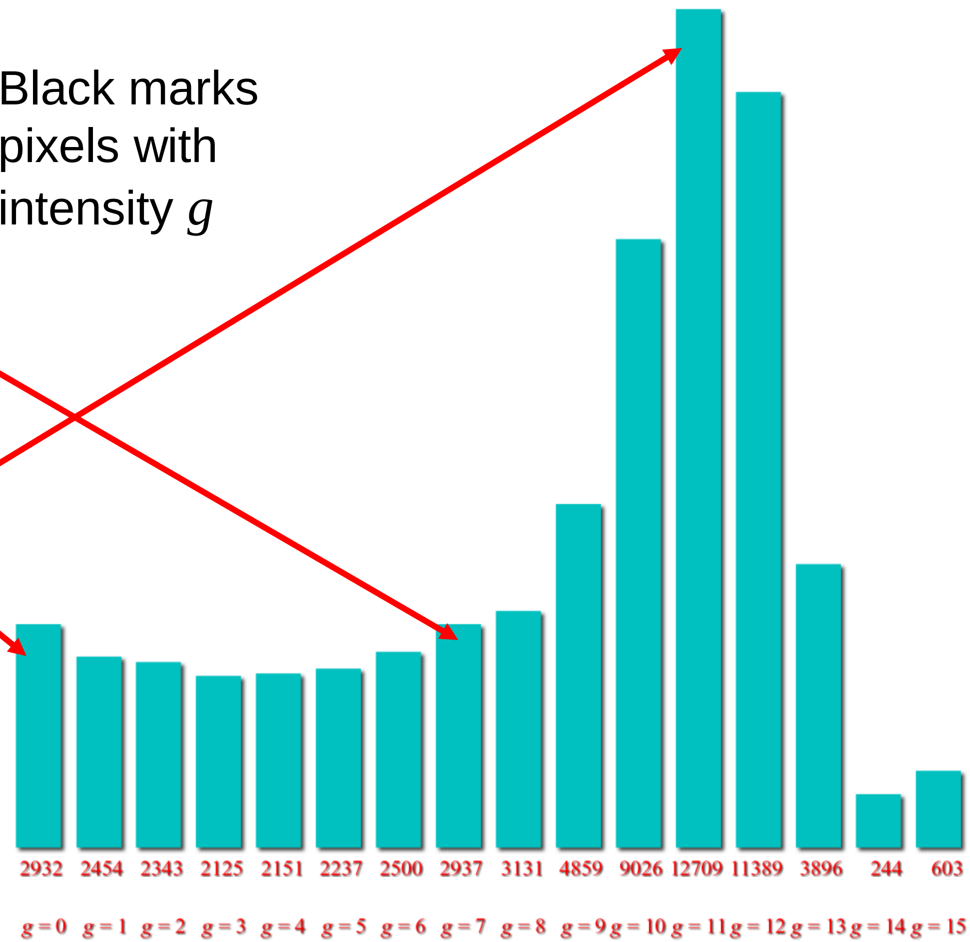


Plot of histogram:
number of pixels with intensity g

Histogram of a Grayscale Image



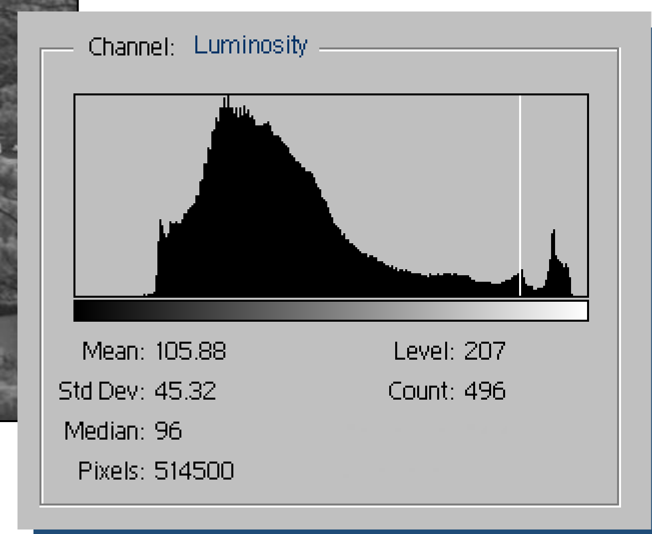
Black marks pixels with intensity g



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.

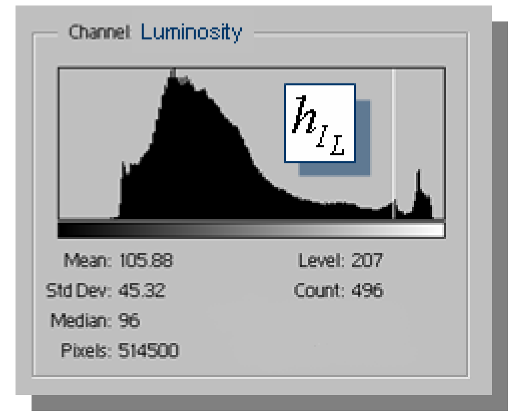


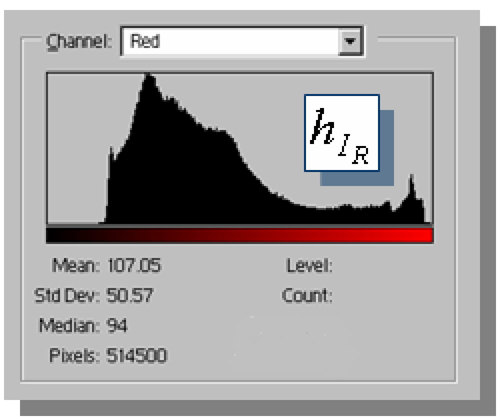
Histogram of a Color Image

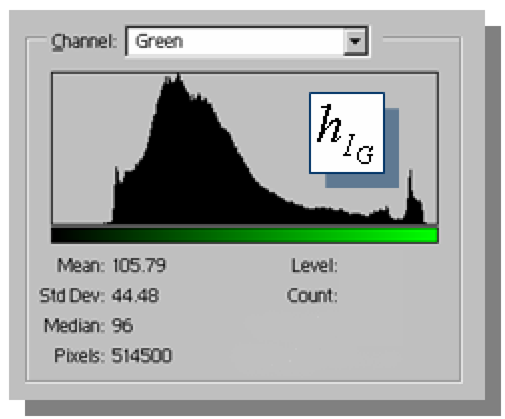
- If I is a 3-band image
- then I(r,c,b) is an integer between 0 and 255.
- I has 3 histograms:
  - $h_R(g)$ = # of pixels in I(:,:,1) with intensity value g
  - $h_G(g)$ = # of pixels in I(:,:,2) with intensity value g
  - $h_B(g)$ = # of pixels in I(:,:,3) with intensity value g

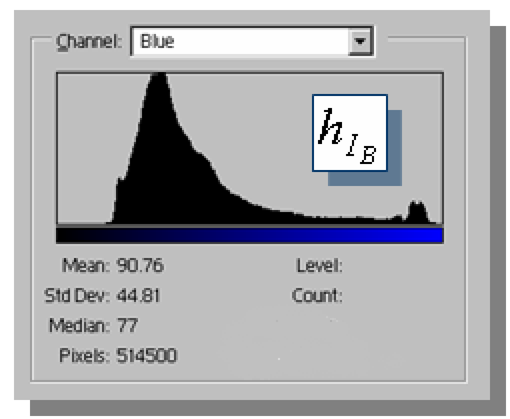
Histogram of a Color Image

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

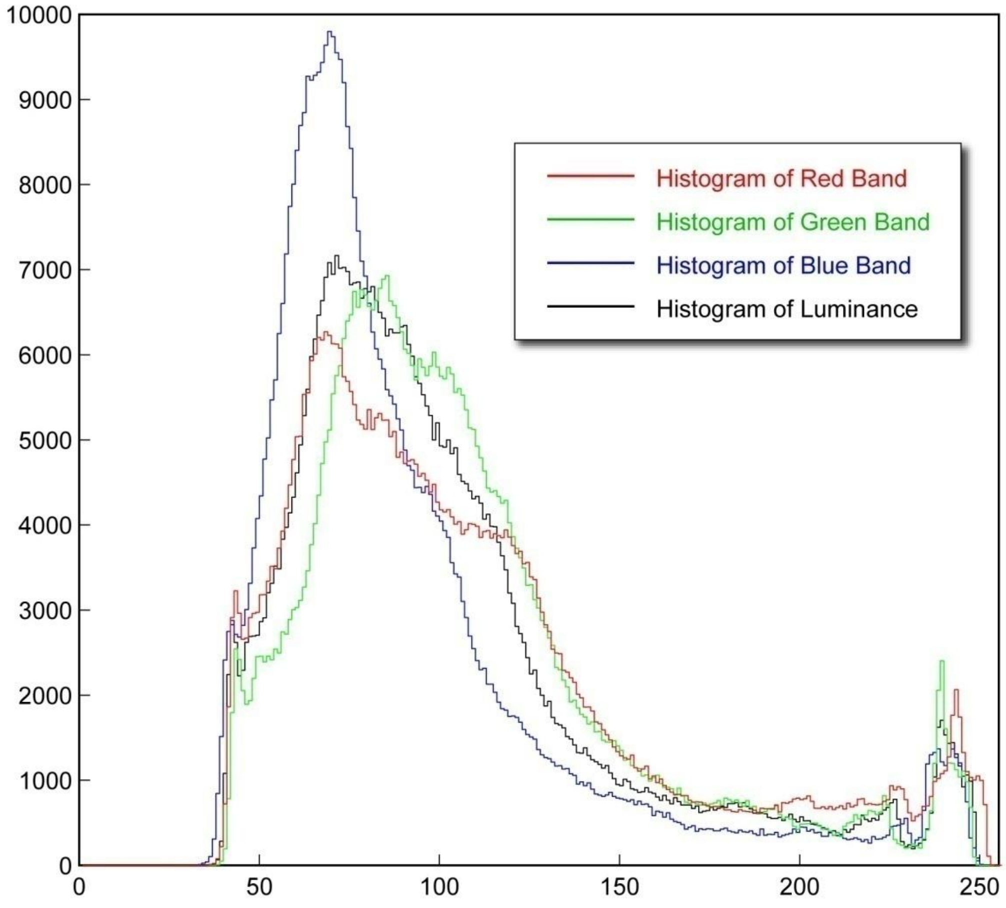




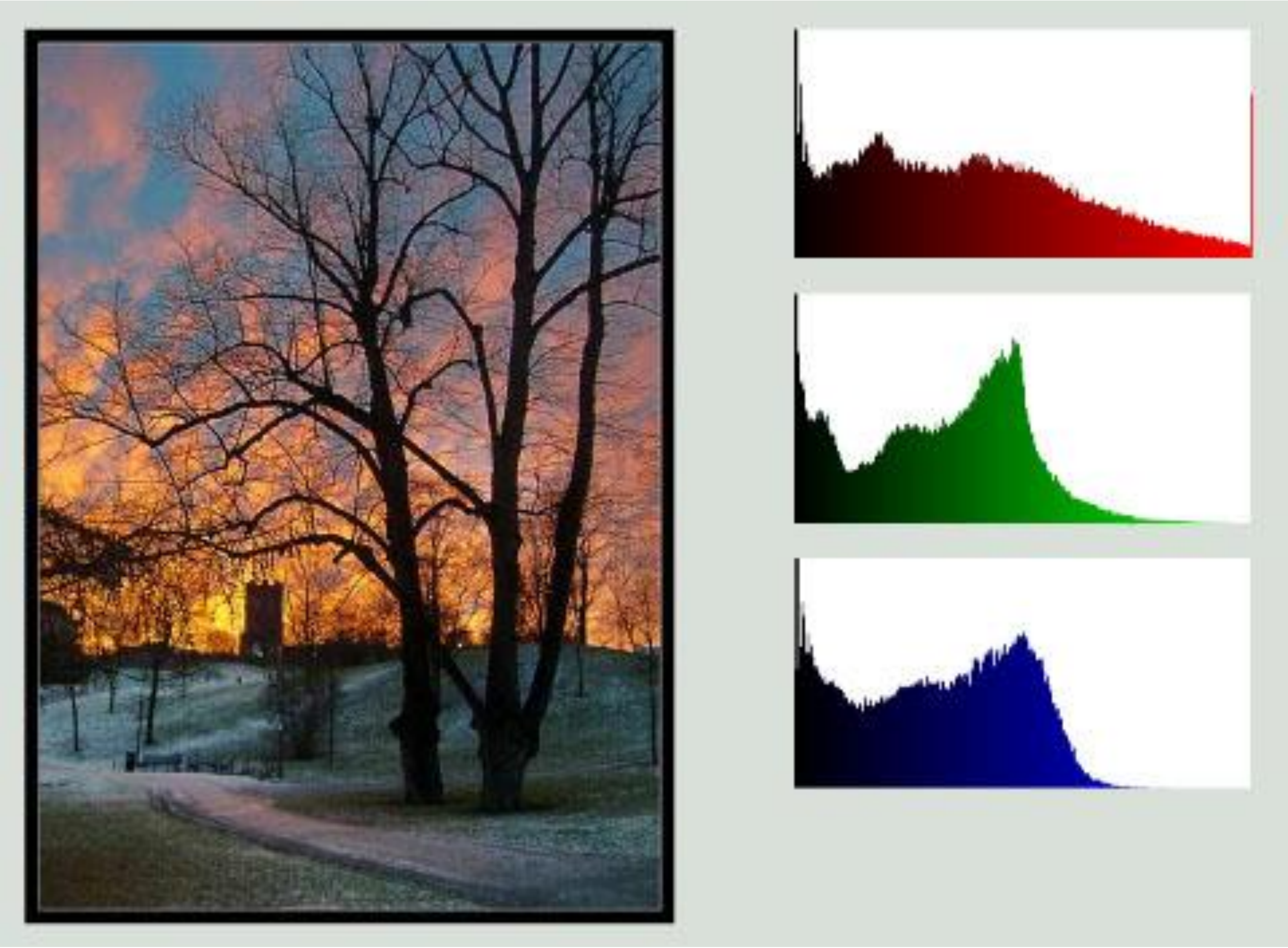




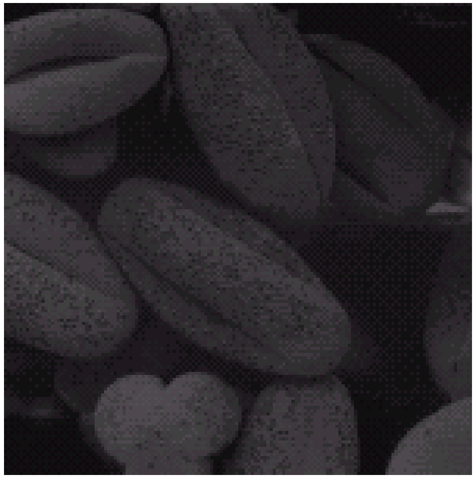
Histogram of a Color Image



Histogram of a Color Image



Histogram: Example

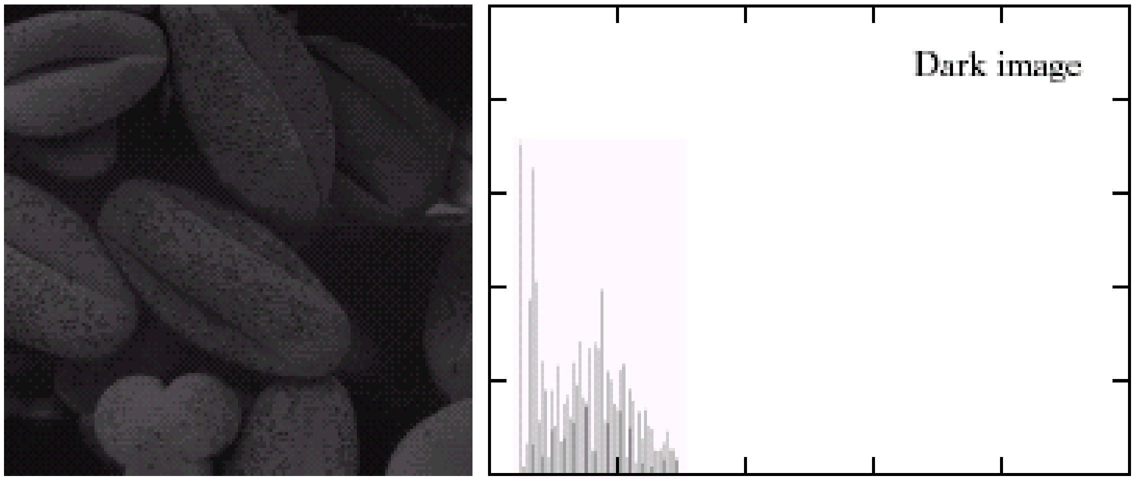


Dark Image

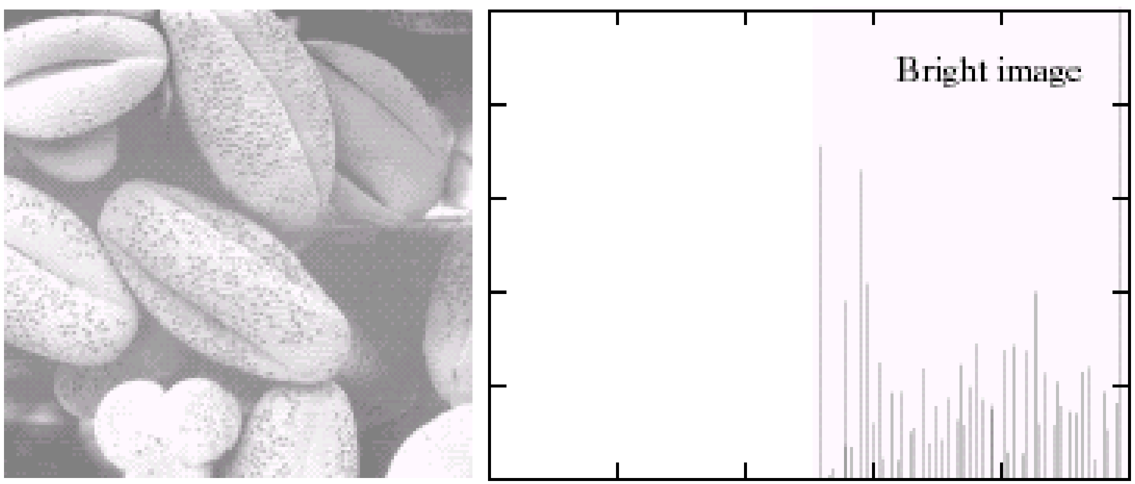
How would the histograms of these images look like?

Bright Image

Histogram: Example



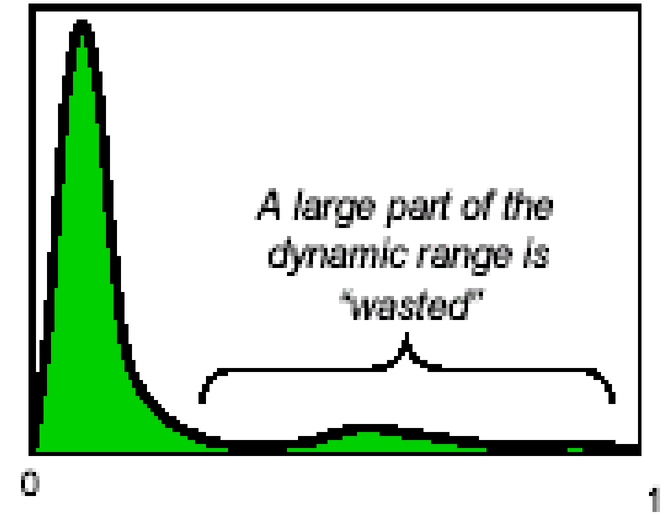
Dark image
Components of histogram are concentrated on the low side of the gray scale

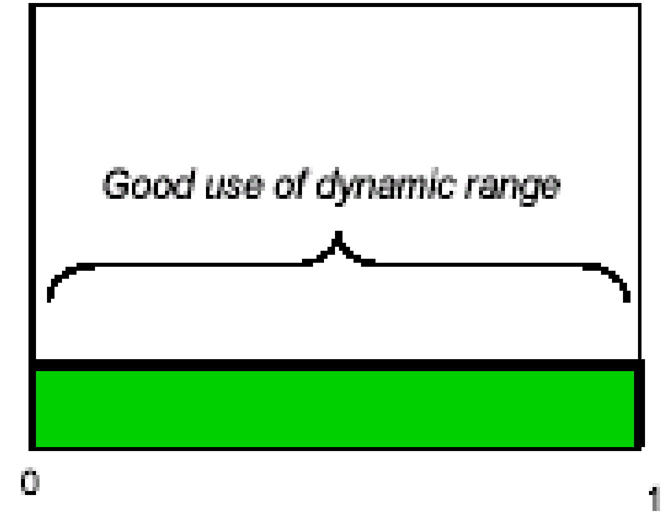


Bright image
Components of histogram are concentrated on the high side of the gray scale

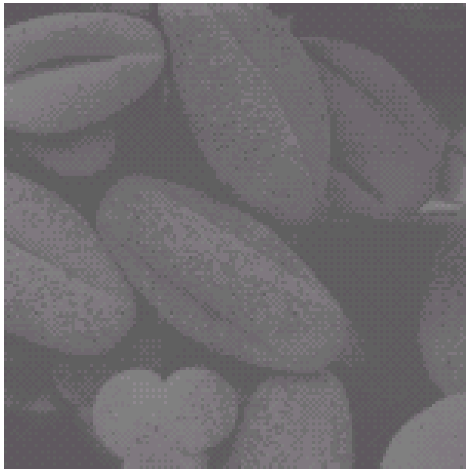
HISTOGRAM INSIGHT INTO CONTRAST

- 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

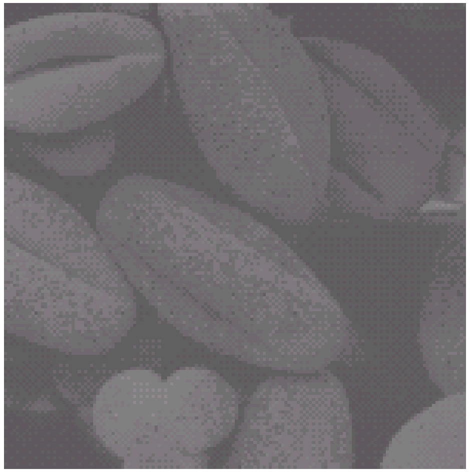


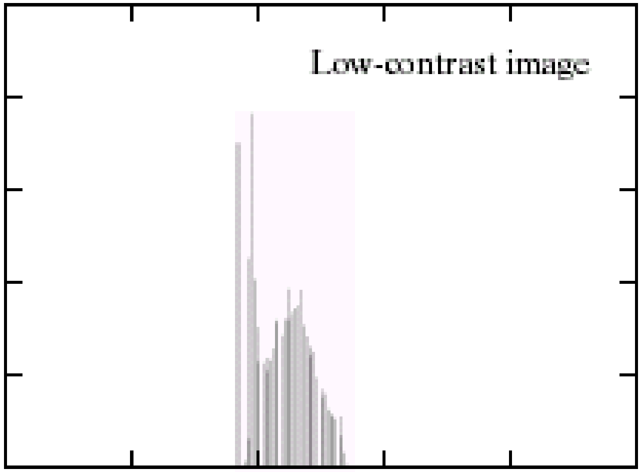
Low Contrast Image

How would the histograms of these images look like?

High Contrast Image

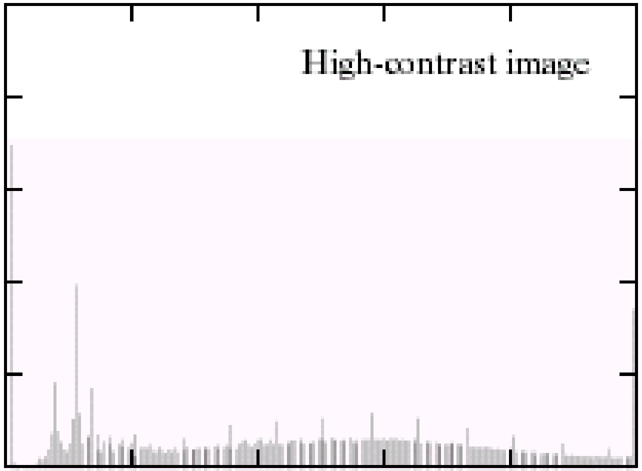
Histogram: Example





Low contrast image

Histogram is narrow and centered toward the middle of the gray scale

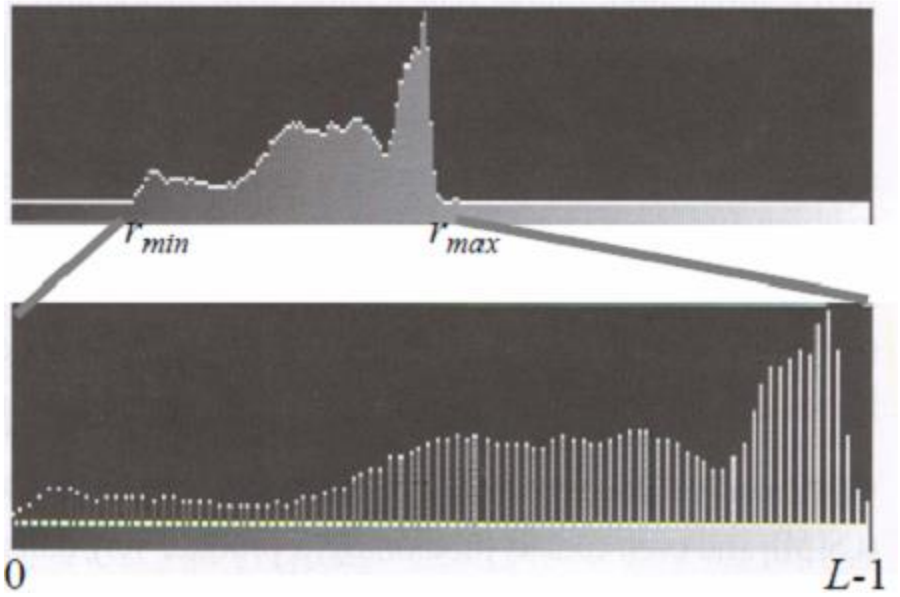


High contrast image

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

Contrast Stretching

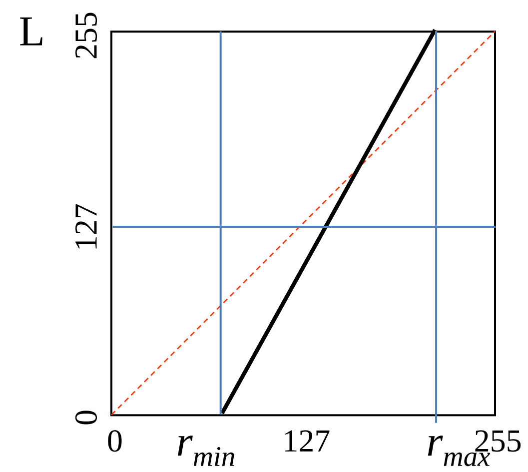
Improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values, *e.g.* the the full range of pixel values



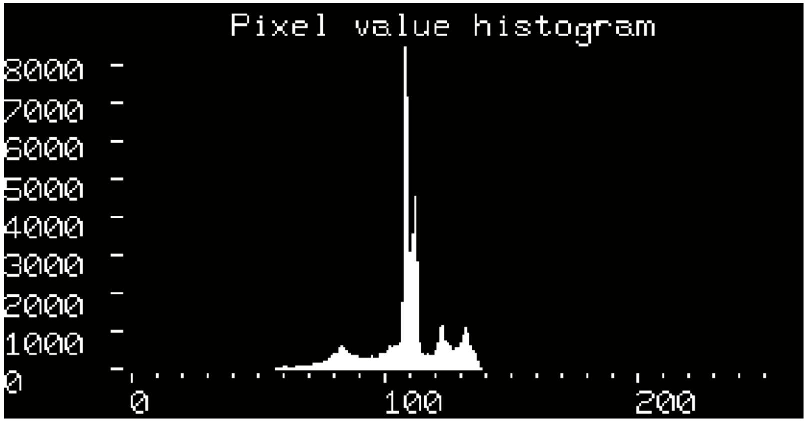
Contrast Stretching

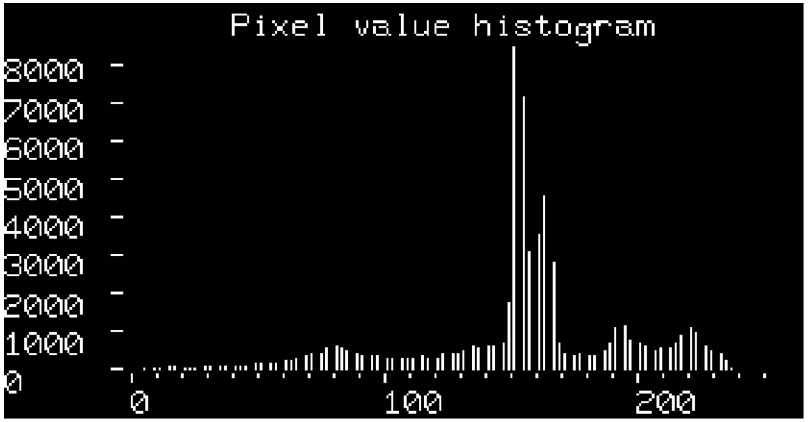
If r_{max} and r_{min} are the maximum and minimum gray level of the input image and L is the total gray levels of output image, the transformation function for contrast stretch will be

$$s = T(r) = (r - r_{\min})\left[\frac{L-1}{r_{\max} - r_{\min}}\right]$$



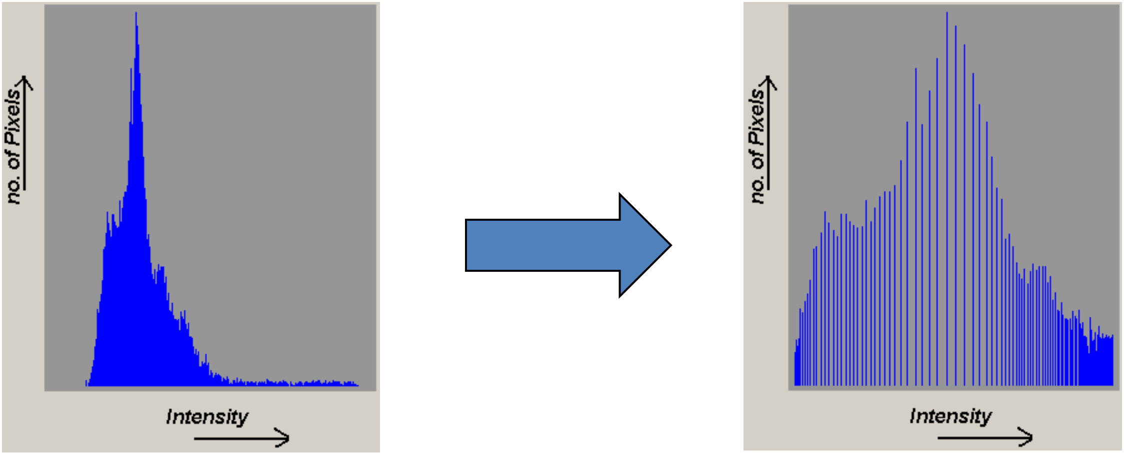
Contrast Stretching



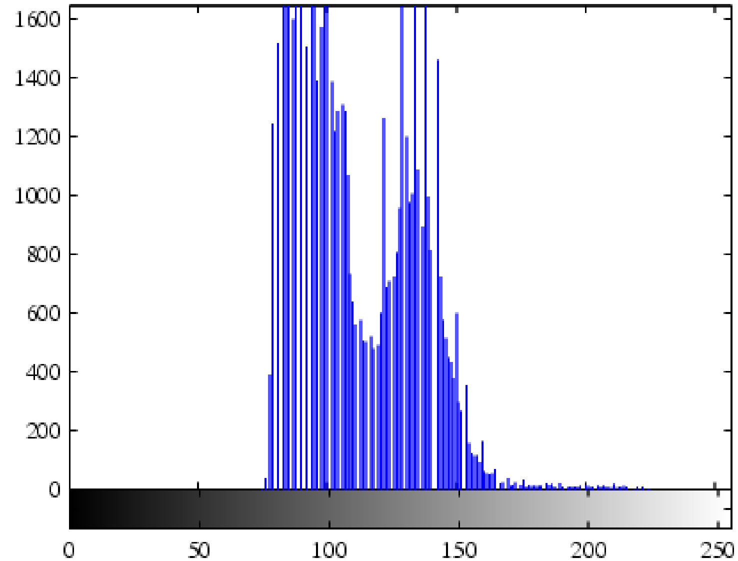


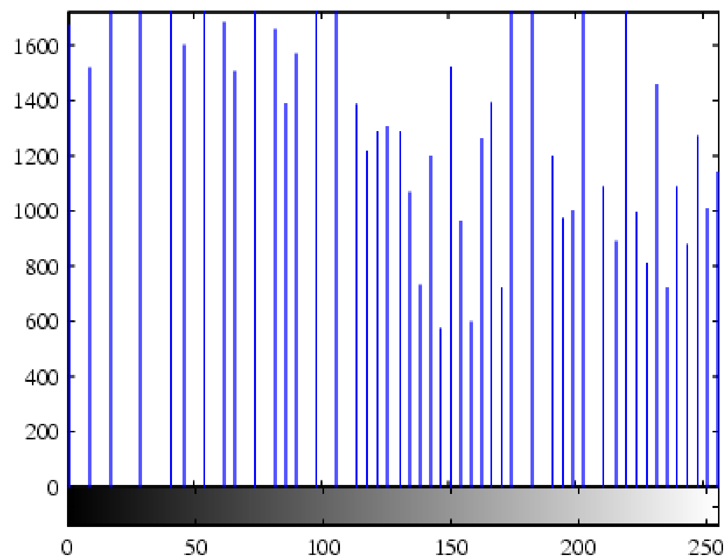
Histogram Equalization

Histogram equalization re-assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities

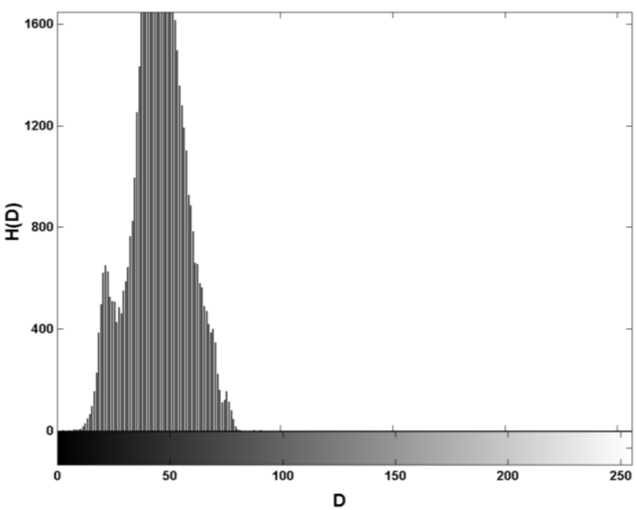


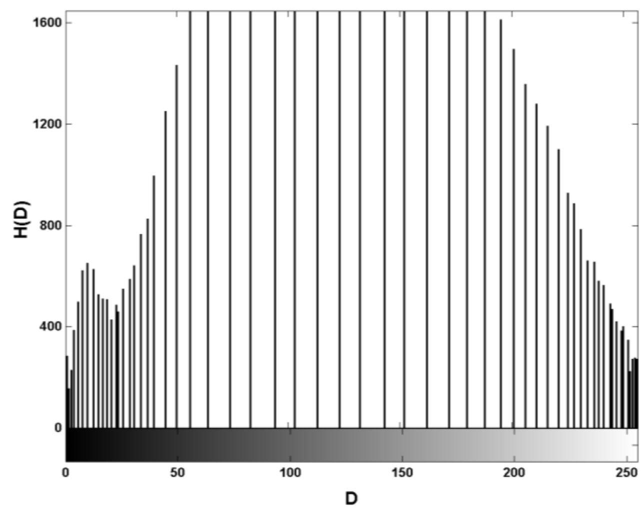
HISTOGRAM EQUALIZATION





AERIAL PHOTOGRAPH OF THE PENTAGON





Resulting image uses more of dynamic range.
Resulting histogram almost, but not completely, flat.

The Probability Distribution Function of an Image

Let $A = \sum_{g=0}^{255} h_I(g)$

Note that since $h_I(g)$ is the number of pixels in I with value g,

A is the number of pixels in I. That is if I is R rows by C columns then $A = R \times C$.

Then,

$$p_I(g) = \frac{1}{A} h_I(g)$$

This is the probability that an arbitrary pixel from *I* has value *g*.

The Probability Distribution Function of an Image

- $p(g)$ is the fraction of pixels in an image that have intensity value g.
- $p(g)$ is the probability that a pixel randomly selected from the given image has intensity value g.
- Whereas the sum of the histogram $h(g)$ over all g from 0 to 255 is equal to the number of pixels in the image, the sum of $p(g)$ over all g is 1.
- p is the normalized histogram of the image

The Cumulative Distribution Function of an Image

Let **q** = $I(r,c)$ be the value of a randomly selected pixel from I. Let g be a specific gray level. The probability that $q \leq g$ is given by

$$P_I(g) = \sum_{\gamma=0}^{g} p_I(\gamma) = \frac{1}{A}\sum_{\gamma=0}^{g} h_I(\gamma) = \frac{\sum_{\gamma=0}^{g} h_I(\gamma)}{\sum_{\gamma=0}^{255} h_I(\gamma)},$$

where $h_I(\gamma)$ is the histogram of image I.

This is the probability that any given pixel from I has value less than or equal to g.

The Cumulative Distribution Function of an Image

Let **q** = *I*(*r*,*c*) be the value of a randomly selected pixel from *I*. Let *g* be a specific gray level. The probability that q ≤ g is given by

$$P_I(g) = \sum_{\gamma=0}^{g} p_I(\gamma) = \frac{1}{A}\sum_{\gamma=0}^{g} h_I(\gamma) = \frac{\sum_{\gamma=0}^{g} h_I(\gamma)}{\sum_{\gamma=0}^{255} h_I(\gamma)},$$

where $h_I(\gamma)$ is the histogram of image *I*.

Also called CDF for "Cumulative Distribution Function".

This is the probability that any given pixel from *I* has value less than or equal to g.

The Cumulative Distribution Function of an Image

- $P(g)$ is the fraction of pixels in an image that have intensity values less than or equal to g.
- $P(g)$ is the probability that a pixel randomly selected from the given band has an intensity value less than or equal to g.
- $P(g)$ is the cumulative (or running) sum of $p(g)$ from 0 through g inclusive.
- $P(0) = p(0)$ and $P(255) = 1$;

Histogram Equalization

Task: remap image I so that its histogram is as close to constant as possible

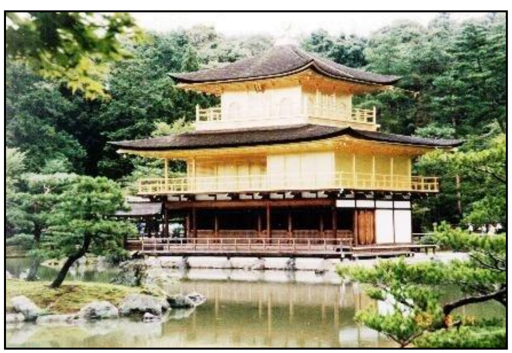
Let $P_I(\gamma)$

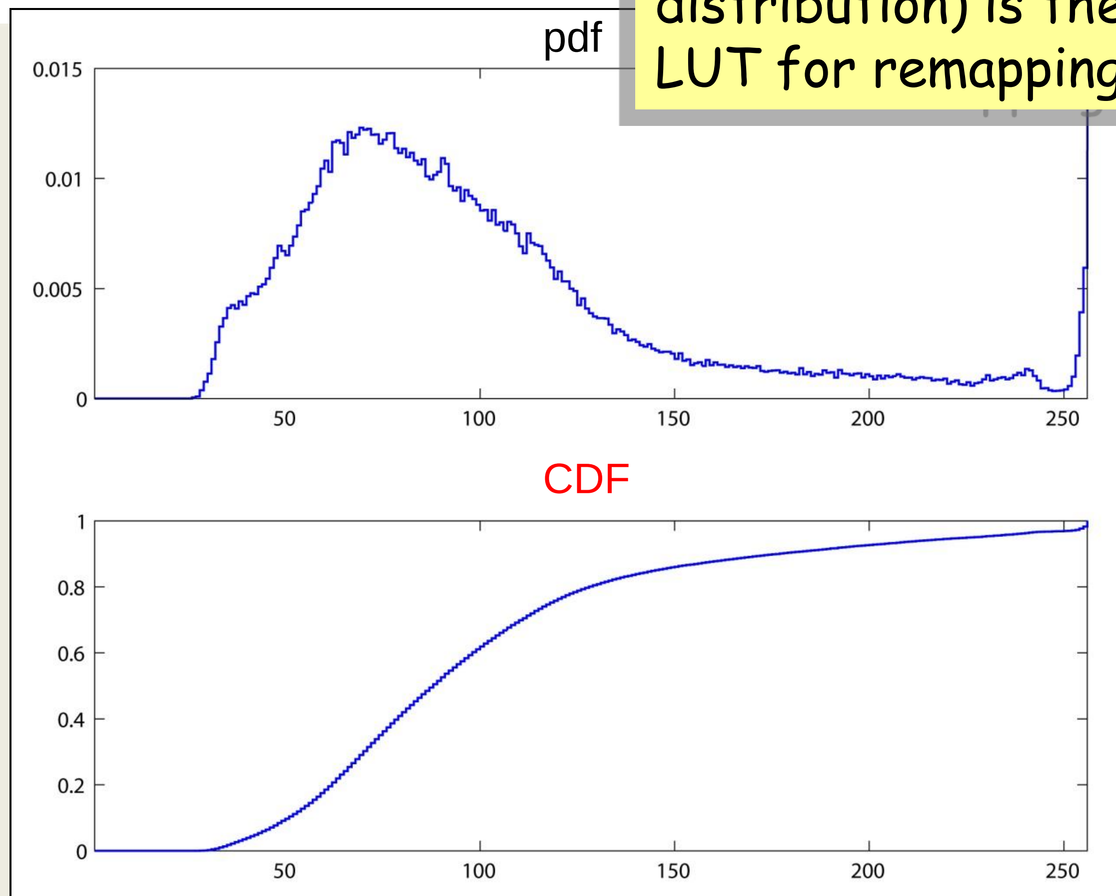
be the cumulative (probability) distribution function of I.

The CDF itself is used as the LUT.

Histogram Equalization

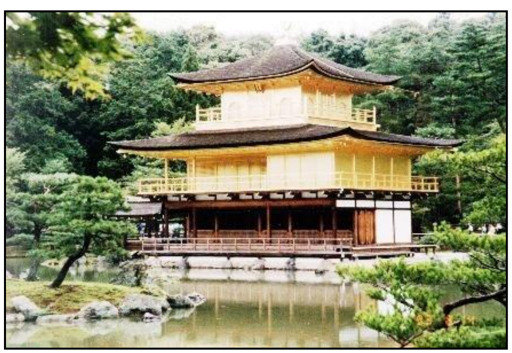
The CDF (cumulative distribution) is the LUT for remapping.

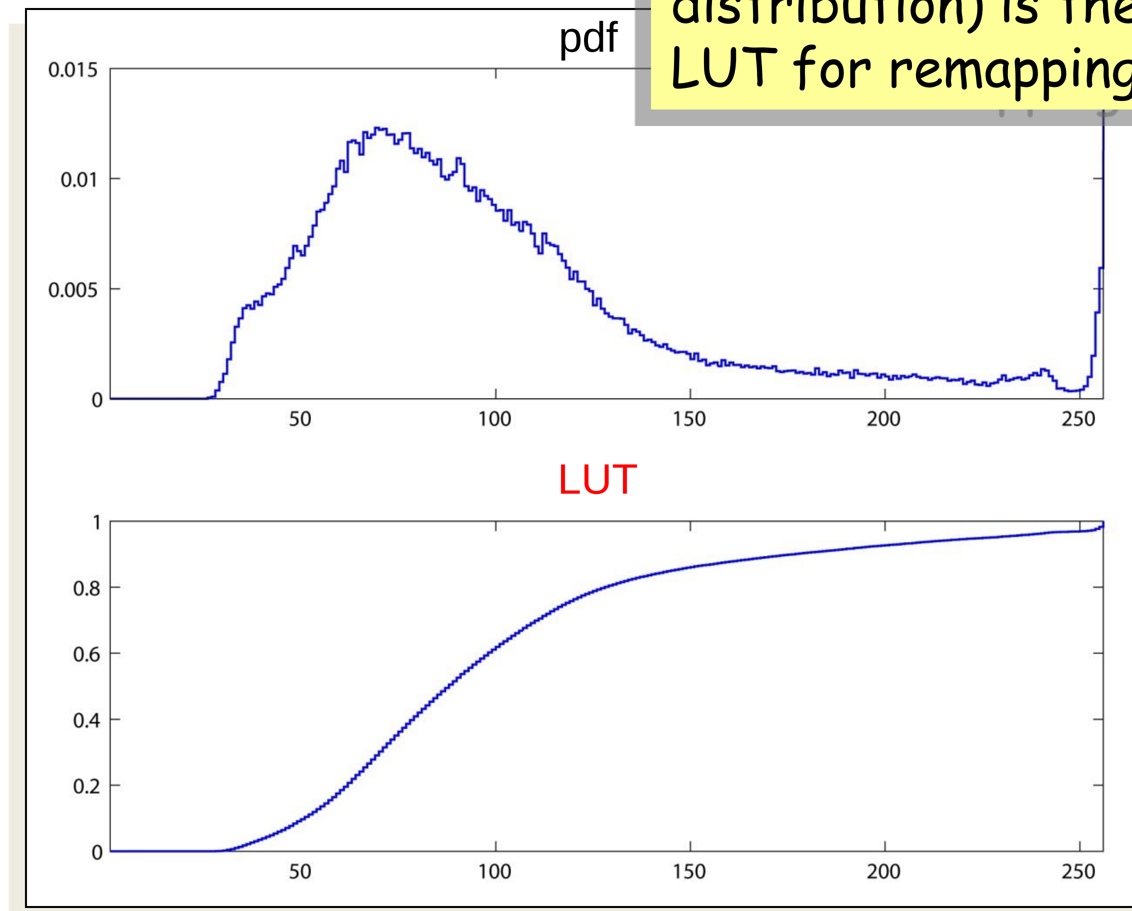




Histogram Equalization

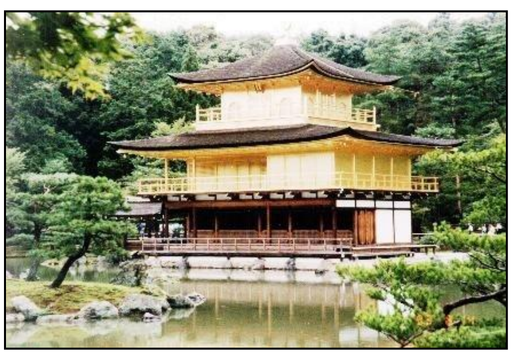
The CDF (cumulative distribution) is the LUT for remapping.

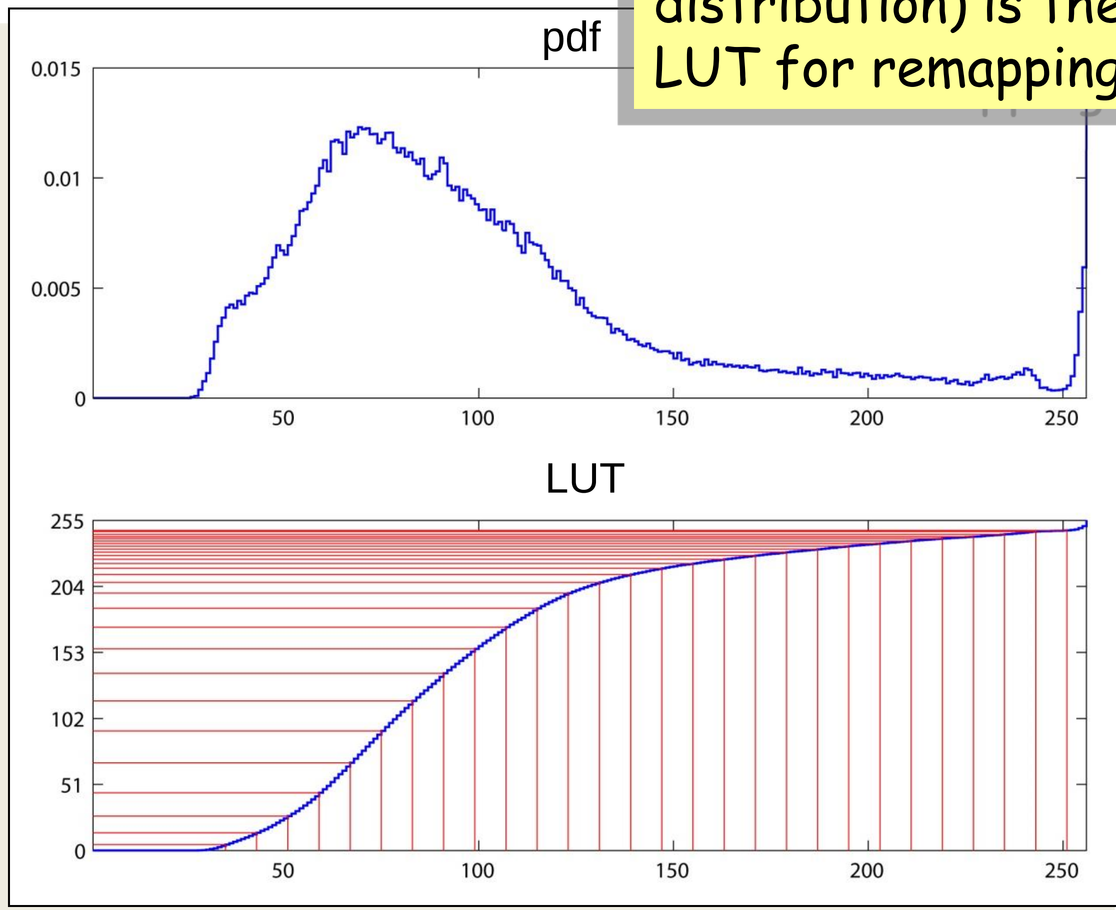




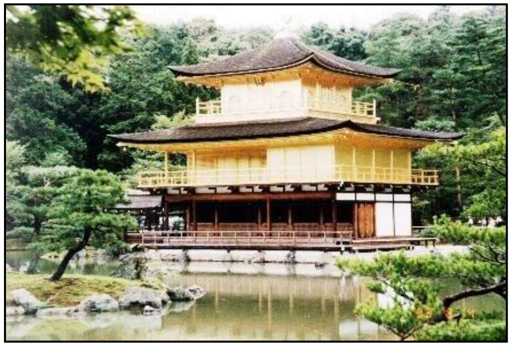
Histogram Equalization

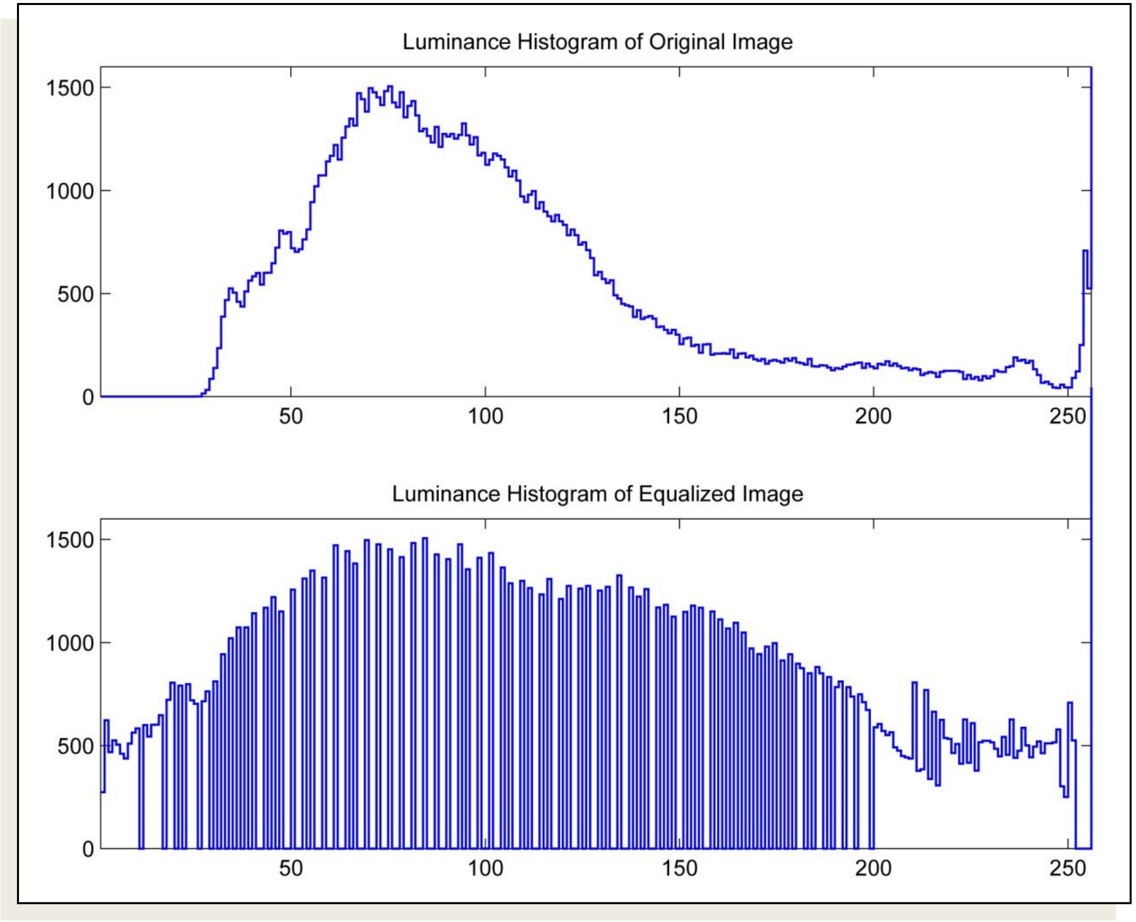
The CDF (cumulative distribution) is the LUT for remapping.





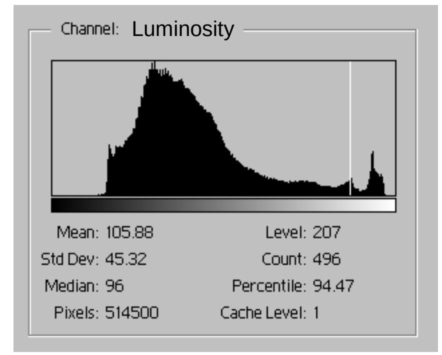
Histogram Equalization



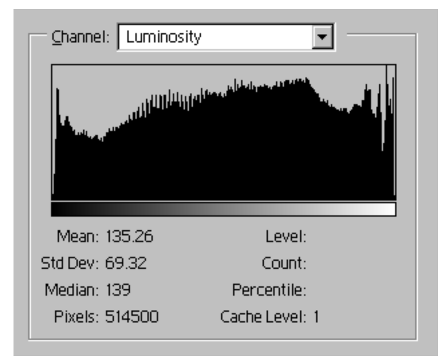


Histogram Equalization

$$J(r,c) = 255 \cdot P_I\left[I(r,c)\right].$$



before



after

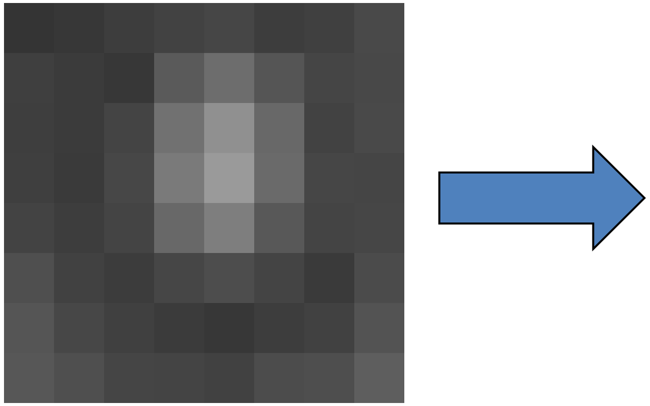
HISTOGRAM EQUALIZATION IMPLEMENTATION

0	0	0	0	0
1	1	1	1	4
4	5	6	6	6
8	8	8	8	9

2	2	2	2	2
4	4	4	4	5
5	5	7	7	7
9	9	9	9	9

Gray levels	0	1	2	3	4	5	6	7	8	9
Counts ($h(r_k)$)	5	4	0	0	2	1	3	0	4	1
	r_0	r_1			r_2	r_3	r_4		r_5	r_6
Normalized h ($P(r_k)$)	5/20	4/20	0	0	2/20	1/20	3/20	0	4/20	1/20
cdf $F(r_k)$	5/20	9/20			**11/20**	**12/20**	**15/20**		**19/20**	**20/20**
$s_k = \text{round}(9 \cdot F(r_k))$	2	4			5	5	7		9	9
	s_0	s_1			s_2	s_3	s_4		s_5	s_6

Histogram Equalization: Example



52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

An 8x8 image

Histogram Equalization: Example

Fill in the following table/histogram

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52		64		72		85		113	
55		65		73		87		122	
58		66		75		88		126	
59		67		76		90		144	
60		68		77		94		154	
61		69		78		104			
62		70		79		106			
63		71		83		109			

Image Histogram (Non-zero values)

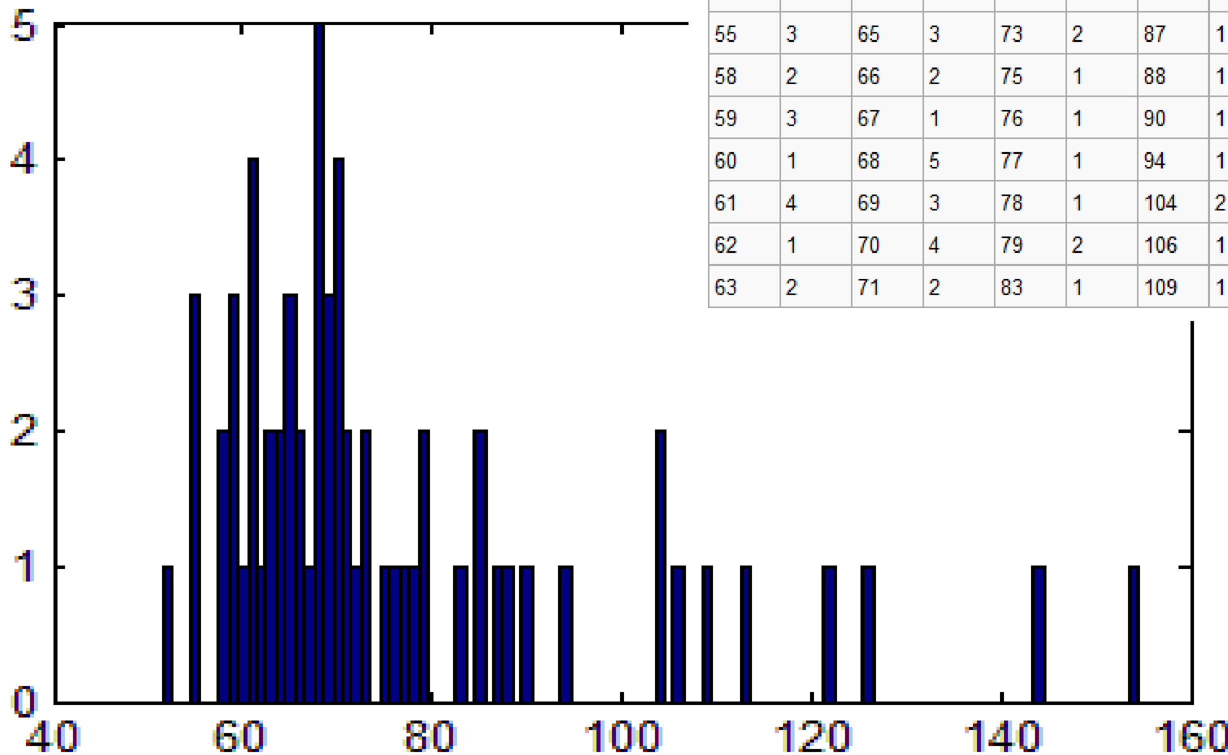
Histogram Equalization: Example

Image Histogram (Non-zero values shown)

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

Histogram Equalization: Example

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		



Histogram Equalization: Example

Image Histogram/Prob Mass Function

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

Cumulative Distribution Function (cdf)

Value	cdf	Value	cdf	Value	cdf	Value	cdf	Value	cdf
52		64		72		85		113	
55		65		73		87		122	
58		66		75		88		126	
59		67		76		90		144	
60		68		77		94		154	
61		69		78		104			
62		70		79		106			
63		71		83		109			

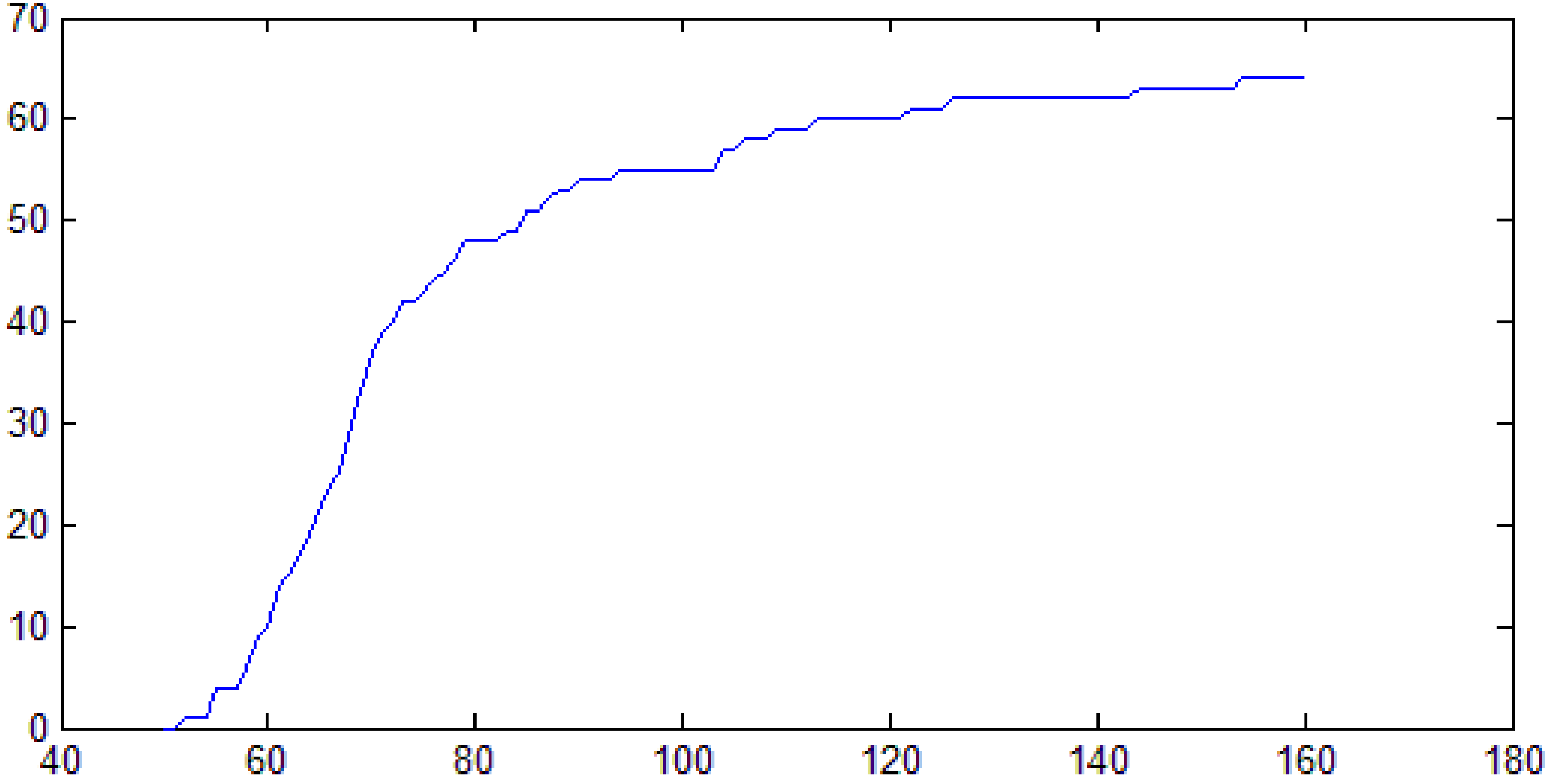
Histogram Equalization: Example

Cumulative Distribution Function (cdf)

Value	cdf	Value	cdf	Value	cdf	Value	cdf	Value	cdf
52	1	64	19	72	40	85	51	113	60
55	4	65	22	73	42	87	52	122	61
58	6	66	24	75	43	88	53	126	62
59	9	67	25	76	44	90	54	144	63
60	10	68	30	77	45	94	55	154	64
61	14	69	33	78	46	104	57		
62	15	70	37	79	48	106	58		
63	17	71	39	83	49	109	59		

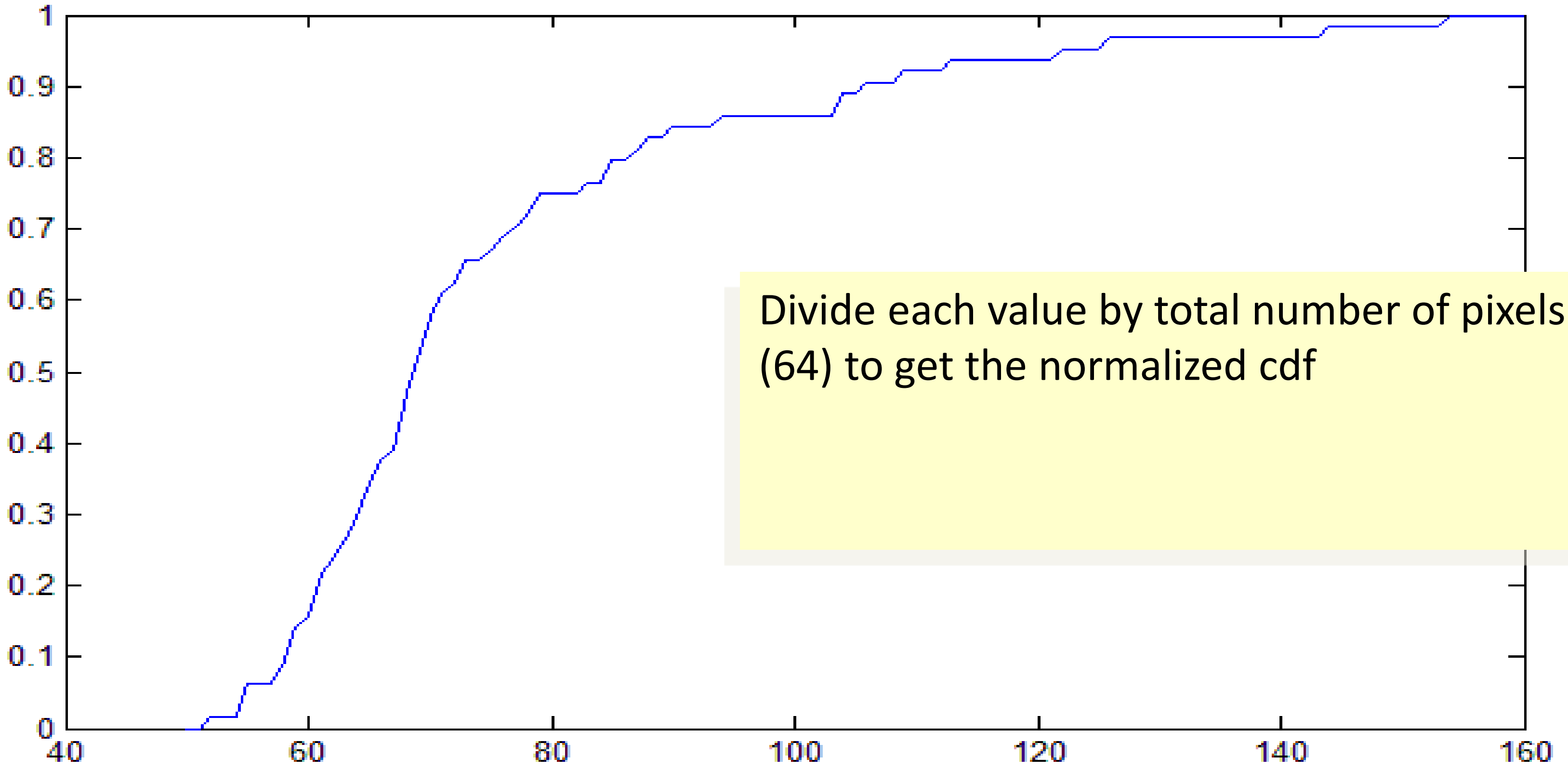
Histogram Equalization: Example

Cumulative Distribution Function (cdf)



Histogram Equalization: Example

Normalized Cumulative Distribution Function (cdf)



Divide each value by total number of pixels (64) to get the normalized cdf

Histogram Equalization: Example

Value	cdf	Value	cdf	Value	cdf	Value	cdf	Value	cdf
52	1	64	19	72	40	85	51	113	60
55	4	65	22	73	42	87	52	122	61
58	6	66	24	75	43	88	53	126	62
59	9	67	25	76	44	90	54	144	63
60	10	68	30	77	45	94	55	154	64
61	14	69	33	78	46	104	57		
62	15	70	37	79	48	106	58		
63	17	71	39	83	49	109	59		

$$J(r,c) = 255 \cdot P_I\left[I(r,c)\right].$$

If cdf is normalized

$$s = round(255.cdf(r))$$

If cdf is NOT normalized

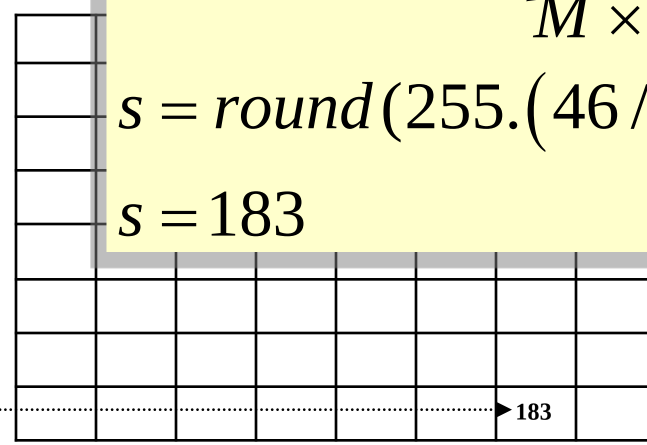
$$s = round(255.\frac{cdf(r)}{M \times N})$$

$$s = round(255.(46/64))$$

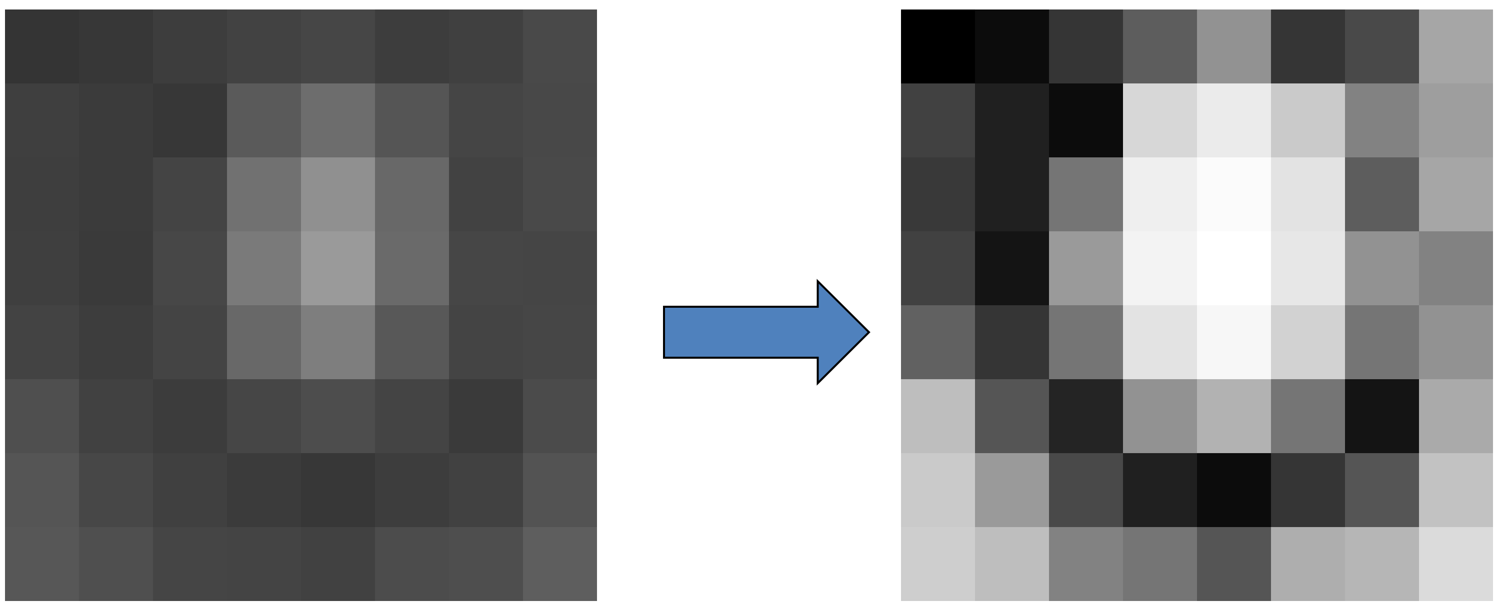
$$s = 183$$

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Original Image

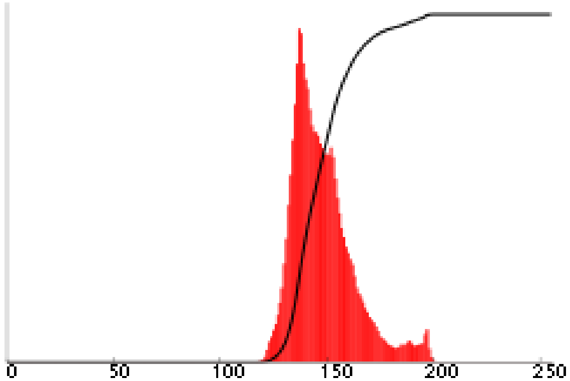


Histogram Equalization: Example



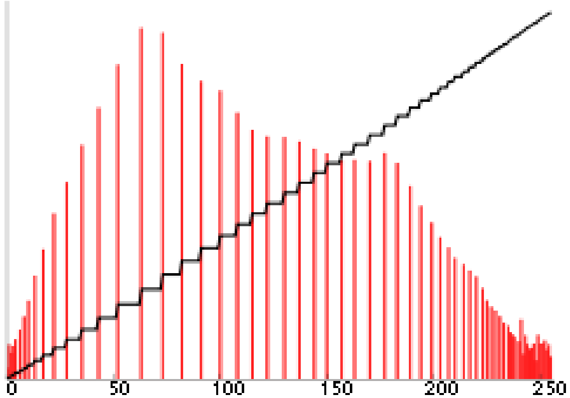
Histogram Equalization: Example

Original Image



Corresponding histogram (red) and cumulative histogram (black)

Image after histogram equalization

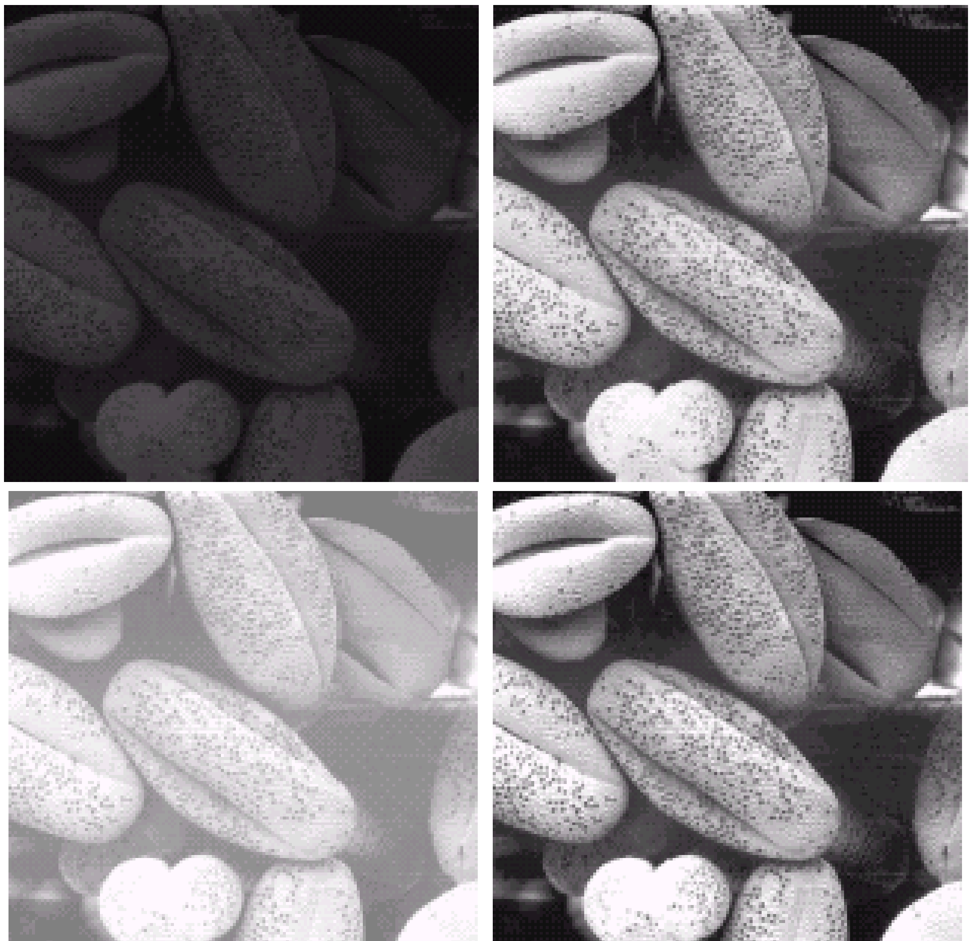


Corresponding histogram (red) and cumulative histogram (black)

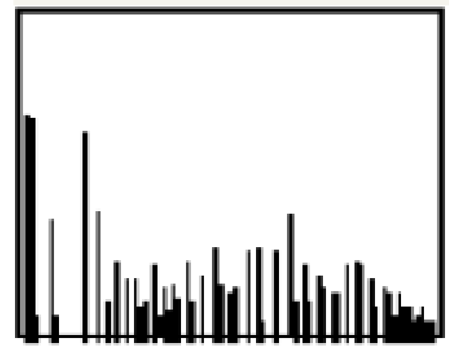
Histogram Equalization: Example

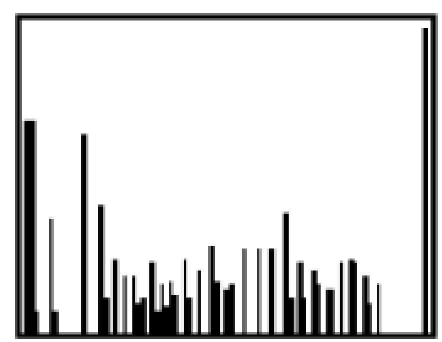
Dark image

Bright image



Equalized Histogram

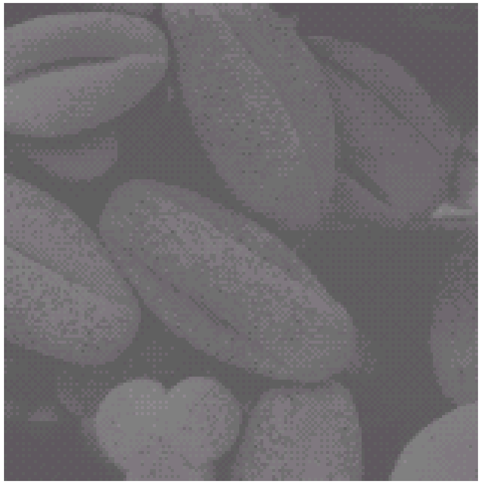




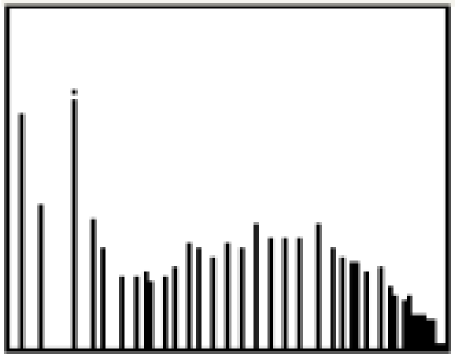
Equalized Histogram

Histogram Equalization: Example

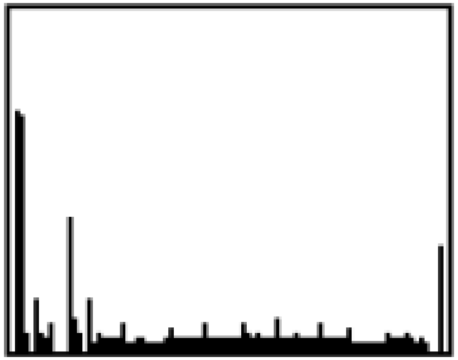
Low contrast



Equalized Histogram



High Contrast



Equalized Histogram

Histogram Equalization vs. Contrast Stretching

Histogram equalization is sophisticated version of contrast stretching

Contrast Stretching – Enhancement is less harsh

Histogram Equalization vs. Contrast Stretching

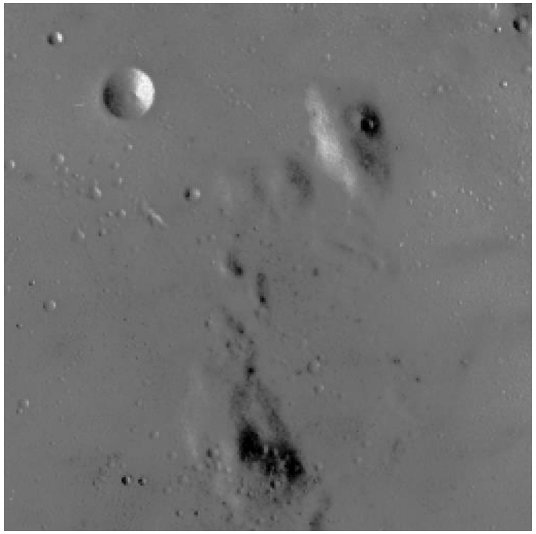
Original Image

Contrast Stretching

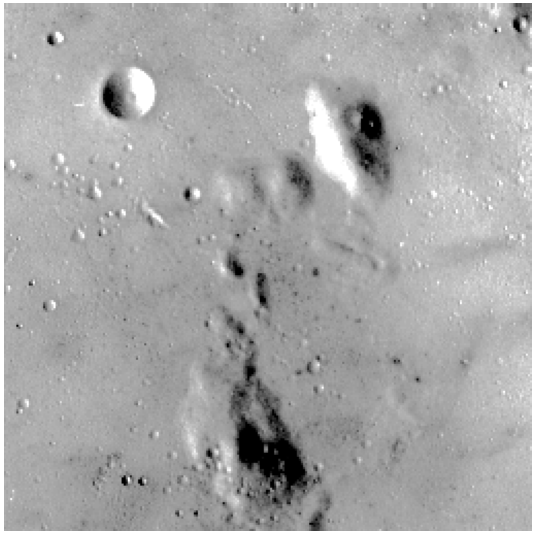
Histogram Equalization

Dramatic Enhancement - Artificial look

Histogram Equalization vs. Contrast Stretching



Original Image



Contrast Stretching

Histogram Equalization

Readings from Book (3^{rd} Edn.)

- Topics from 3^{rd} chapter which we covered during lecture
- Bit plane slicing
- Histogram Matching/specification
- Local/global histogram equalization

Acknowledgements

- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Peters, Richard Alan, II, Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008
- Brian Mac Namee, Digitial Image Processing, School of Computing, Dublin Institute of Technology
- Computer Vision for Computer Graphics, Mark Borg

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