Point Processing

Image Processing

Dr. Zein Al Abidin IBRAHIM

ibrahim.zein@gmail.com

Zein.Ibrahim@ul.edu.lb

IN433
Multimedia Processing

Histogram

- What is an histogram ?
- Interpreting histogram
- Computing histogram
- Grayscale image histogram
- Color image histogram
- Cumulative histogram

Point operations?

- Modifying image intensity
- Point operations and histograms
- Contrast adjustment
- Histogram normalization
- Histogram equalization
- Gamma correction
- ……

Outline

- 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 of Images

- Brightness and contrast adjustment
- Thresholding
- Histogram equalization
- Histogram matching
- ……..

Point Processing of Images

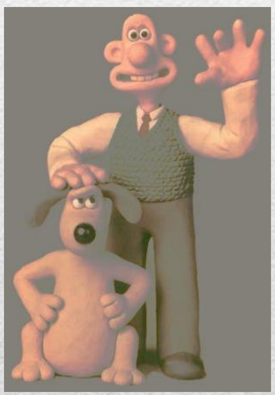
- gamma

- brightness

original

+ brightness

+ gamma



histogram mod

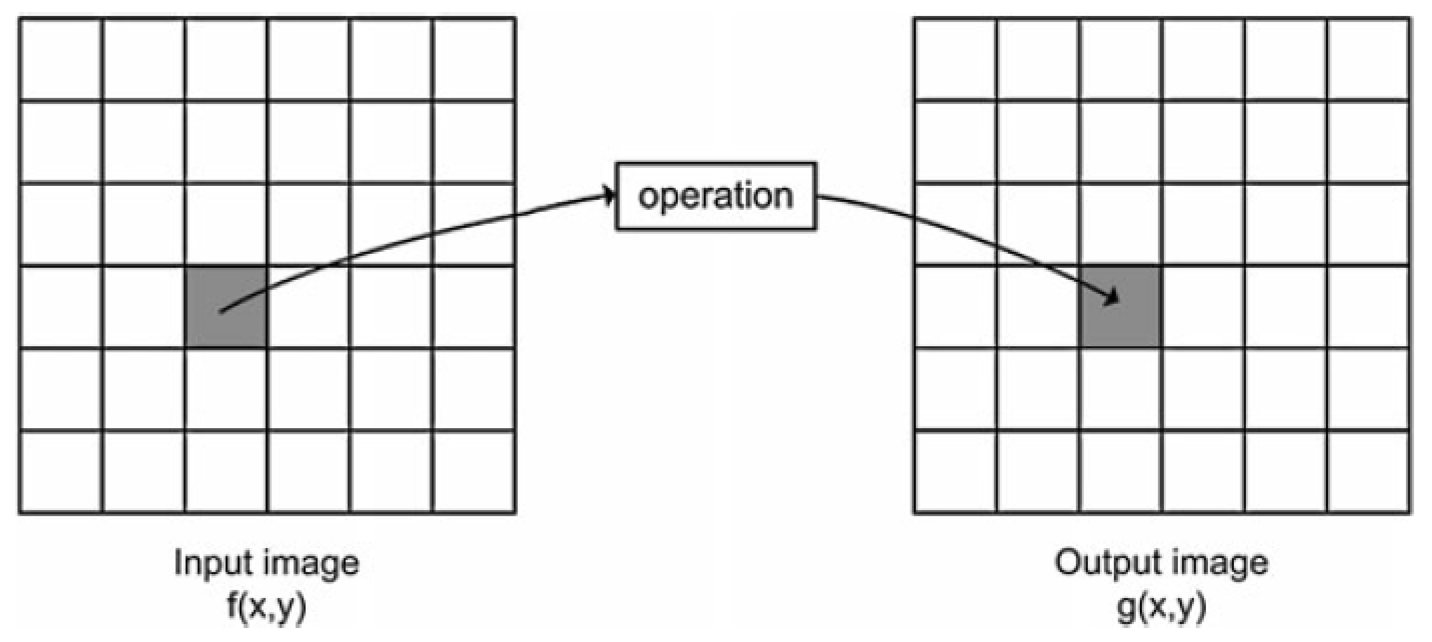
- contrast

original

+ contrast

histogram EQ

Point Processing



$$g(x, y) = f(x, y) + b$$

$$g(x, y) = a \cdot f(x, y)$$

$$g(x, y) = a \cdot f(x, y) + b$$

Point operations

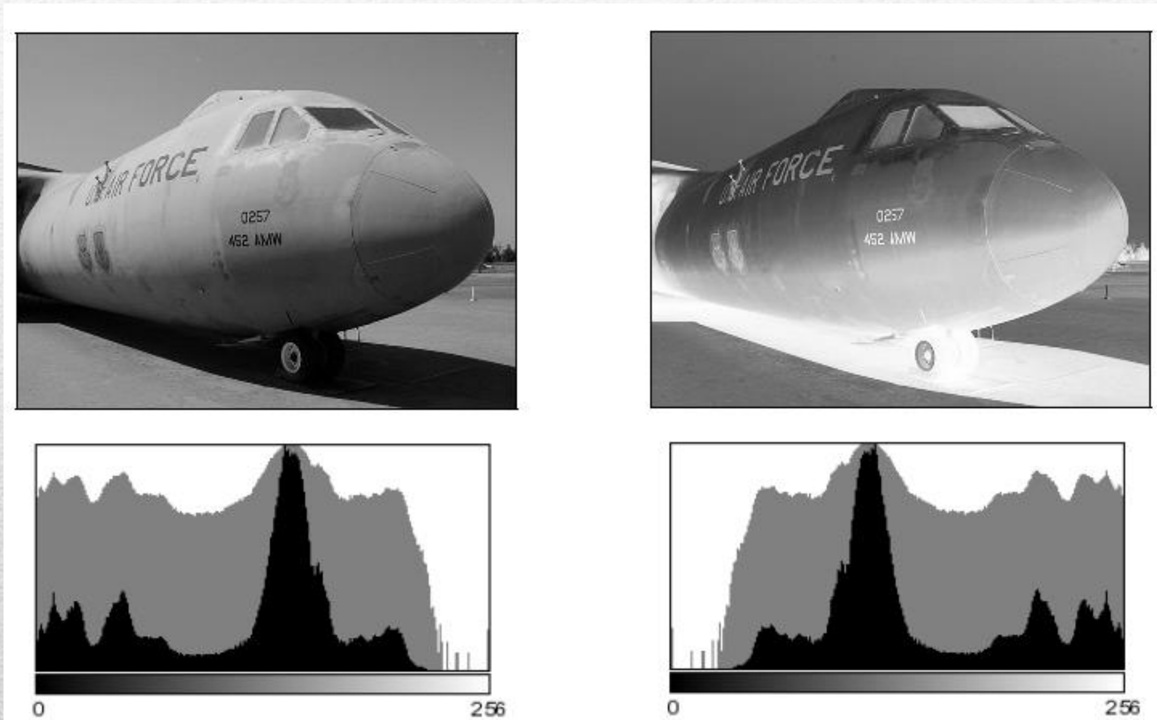
- Examples of point operations
  - modifying image brightness or contrast,
  - applying arbitrary intensity transformations (“curves”),
  - quantizing (or “posterizing”) images,
  - global thresholding,
  - gamma correction,
  - color transformations.

Point Operations

Image inversion (negative)

- Inverting an intensity image is a simple point operation that reverses the ordering of pixel values

$f(a) = a_{max} - a$ for grayscale images: $a_{max} = 255$



- Special type of quantization

$$f_{\text{threshold}}(a) = \begin{cases} a_0 & \text{for } a < a_{\text{th}} \\ a_1 & \text{for } a \geq a_{\text{th}} \end{cases}$$

(a) $a_{\text{th}} = 128$ (b)

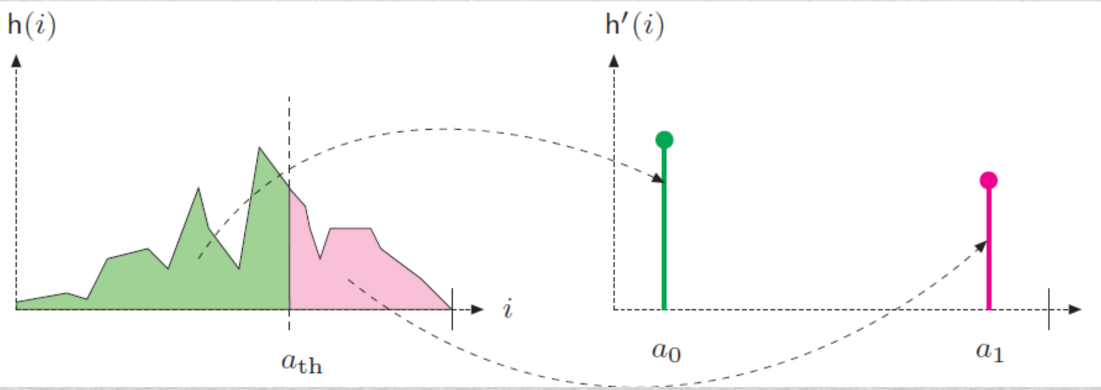


Image thresholding

- One of the application → segmentation of the foreground from the background (objects extraction)

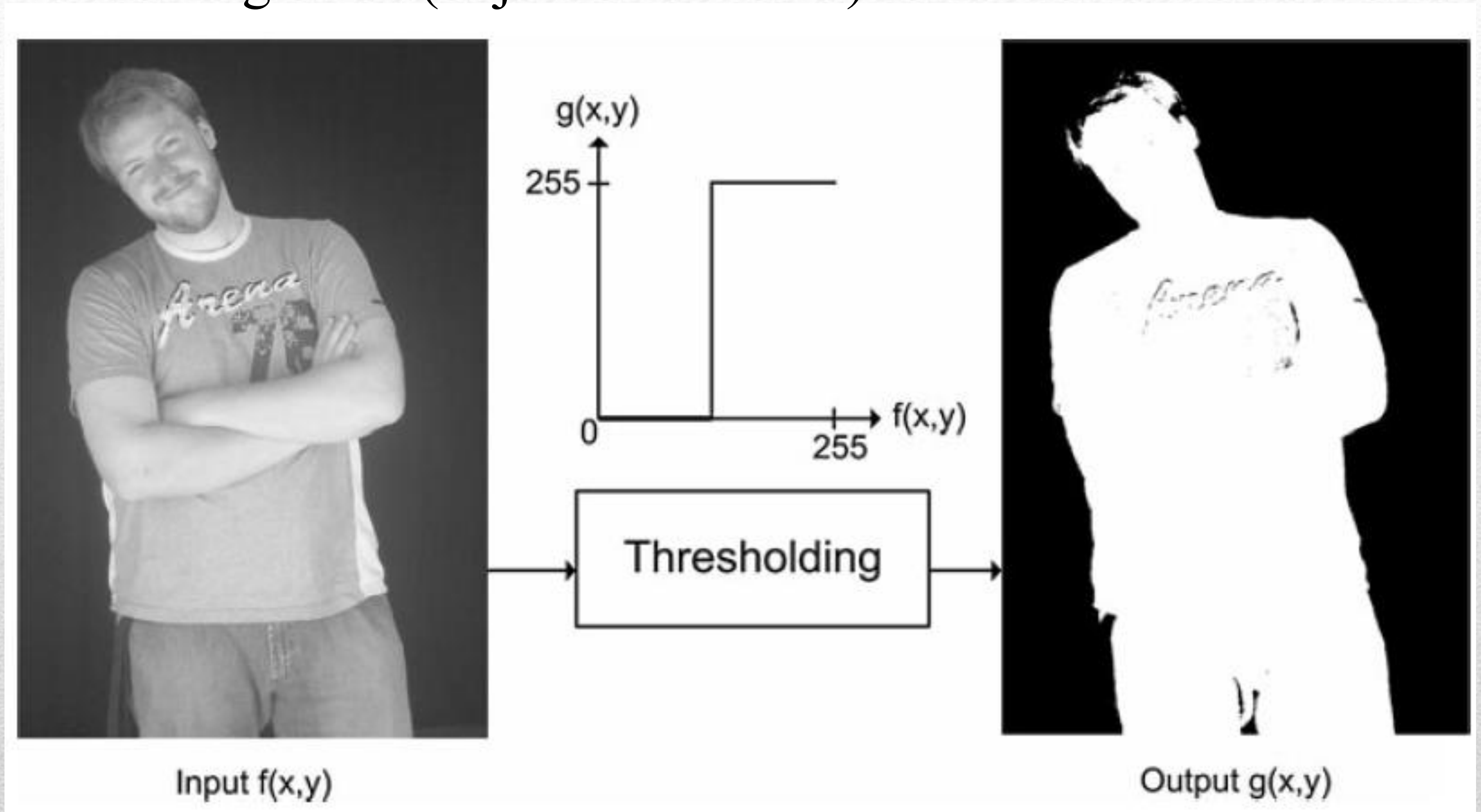
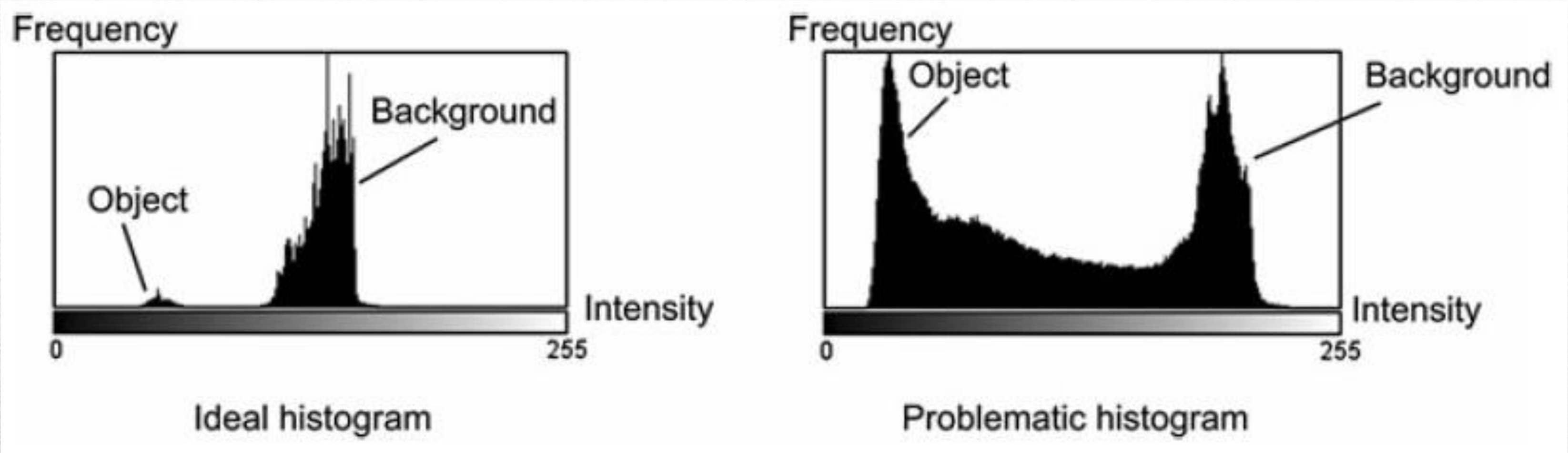


Image thresholding

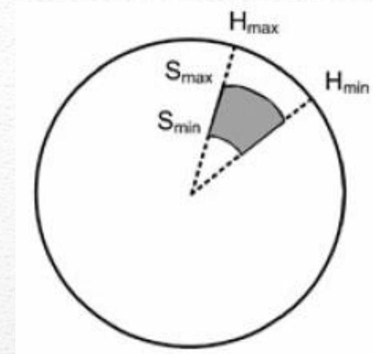
➢ Problematic situation



- Perfect situations → histogram is composed of two mountains
  - one for foreground pixels
  - one for background pixels
- How to fix the threshold
  - fix threshold → not practical
  - adaptive threshold → based on the distribution of pixel values in the histogram
- In the problematic situations → becomes hard to make the separation

Image thresholding

- Can be done in different ways
  - Threshold each band separately
  - Threshold all at the same time



- The following thresholding equivalent to defining a box in the RGB space

If

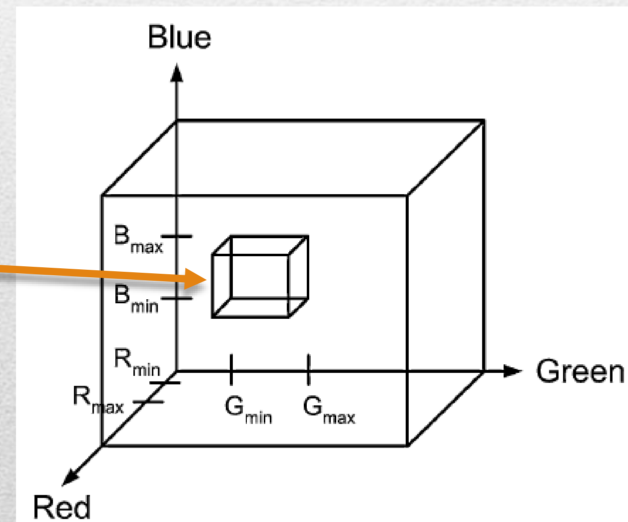
$R > R_{\min}$ and $R < R_{\max}$ and

$G > G_{\min}$ and $G < G_{\max}$ and

$B > B_{\min}$ and $B < B_{\max}$

Then $g(x, y) = 255$

Else $g(x, y) = 0$



Colored image thresholding

- Example of application: face detection
  - Suppose you know the color skin
  - Suppose that no other parts of the image have the same color as the skin
  - Define the thresholds to detect the skin pixels

- Problem: Sensitive to illumination which increases the intensity value of the pixels
- Solution 1: make the cube (thresholds) bigger but risk of including non-skin pixels
- Solution2: Use a color space that separates the color from the intensity such as HSV and then threshold the HS part.

Colored image thresholding

- In general is a frequency distribution
- Describes the frequency of the intensity values that occur in an image.
- Depicts image statistics in an easily interpreted visual format.
- Used to process an image

What is an histogram ?

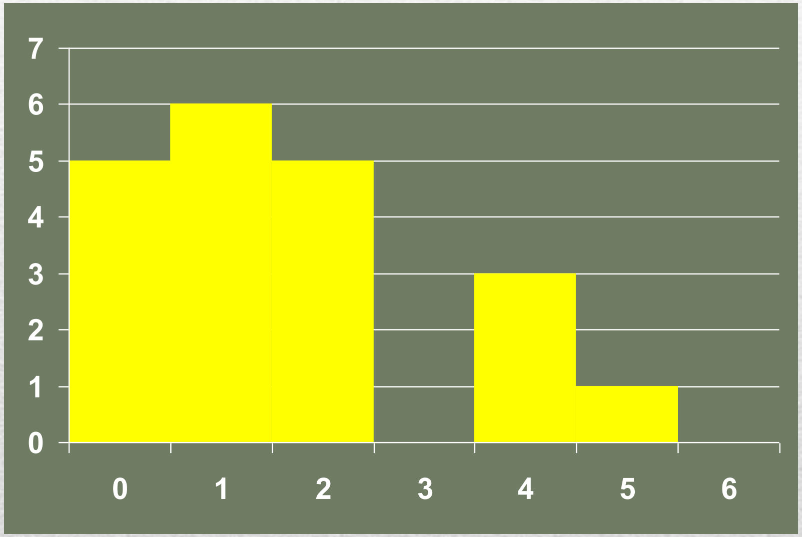
- Let **I** be a 1-band (grayscale) image.
- **I**(r,c) is an 8-bit integer between 0 and 255.
- Histogram, $h_{\mathbf{I}}$, of **I**:
  - a 256-element array, $h_{\mathbf{I}}$
  - $h_{\mathbf{I}}(i)$, for $i = 1, 2, 3, \ldots, 256$, is an integer
  - $h_{\mathbf{I}}(i)$ = number of pixels in **I** that have value i-1.

The Histogram of a Grayscale Image

- 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



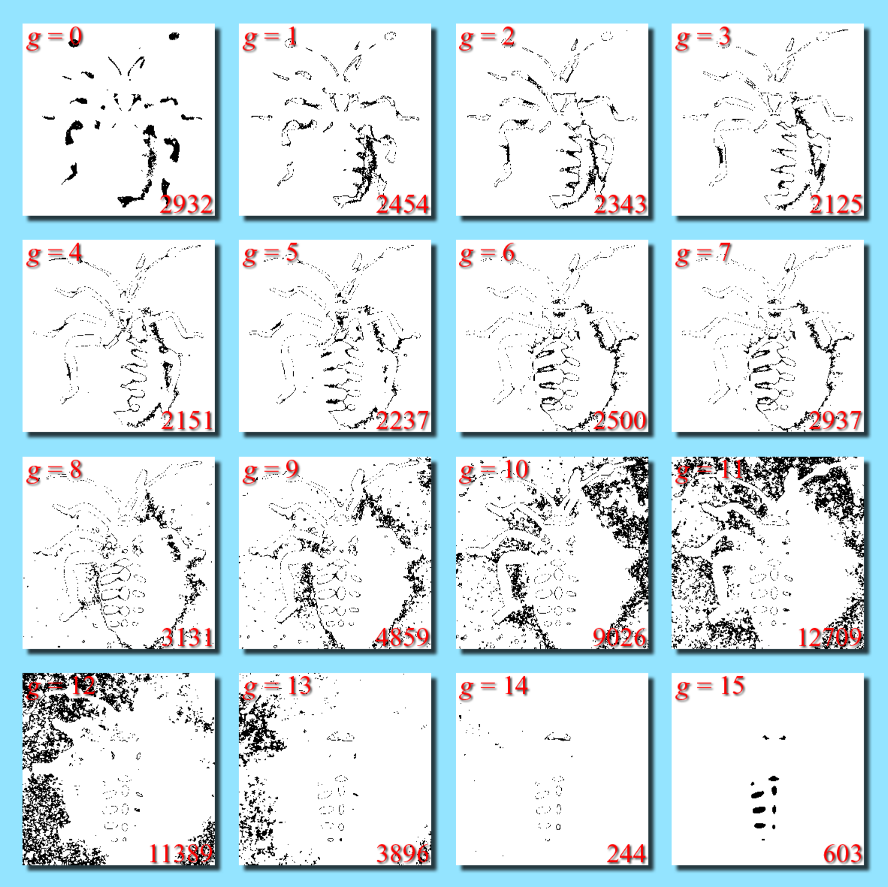
Histogram

histogram

- 4-bit grayscale image → 16 levels (0, 1, …, 15)

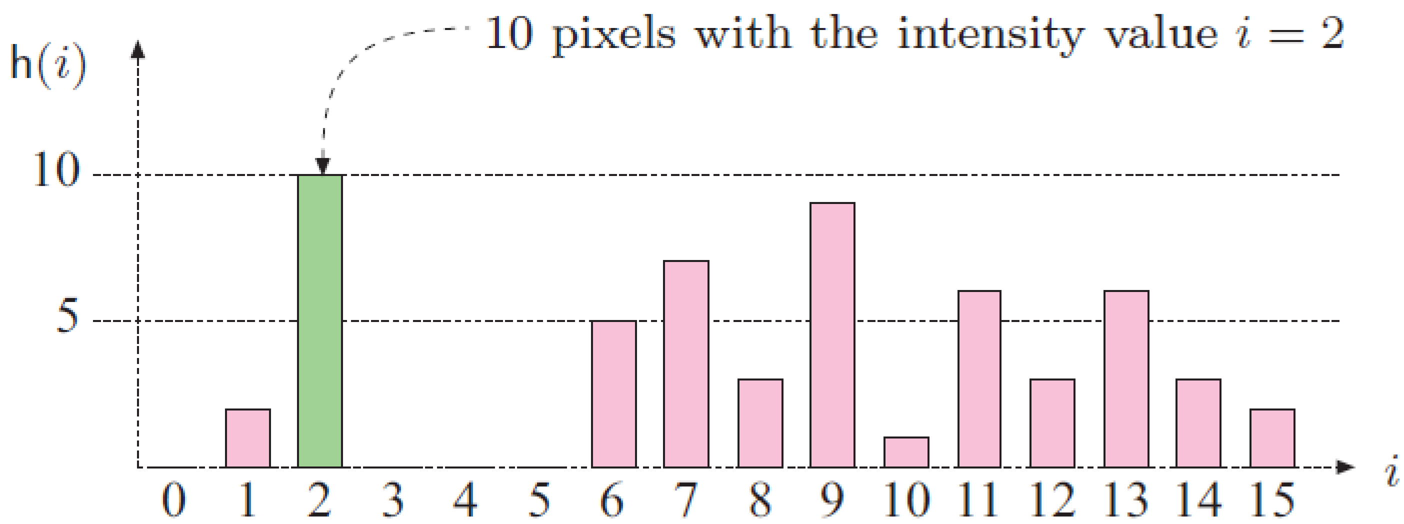
16-level (4-bit) image

lower RHC: number of pixels with intensity i



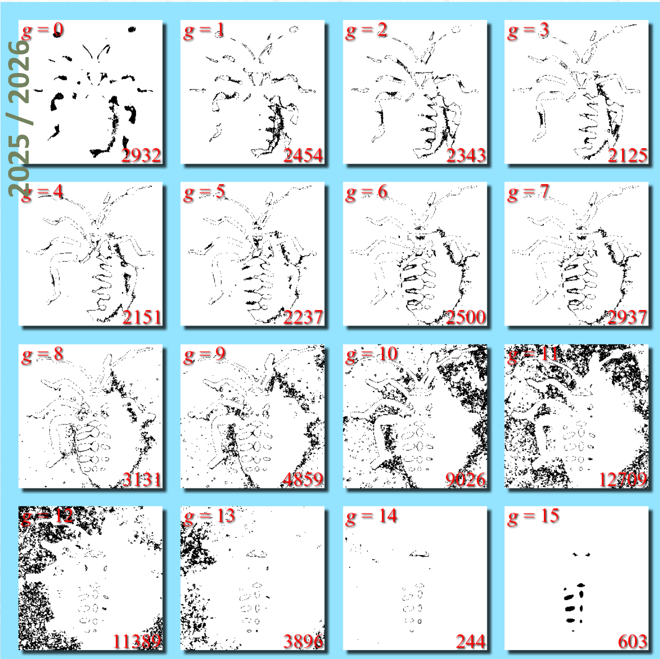
black marks pixels with intensity i

The Histogram of a Grayscale Image



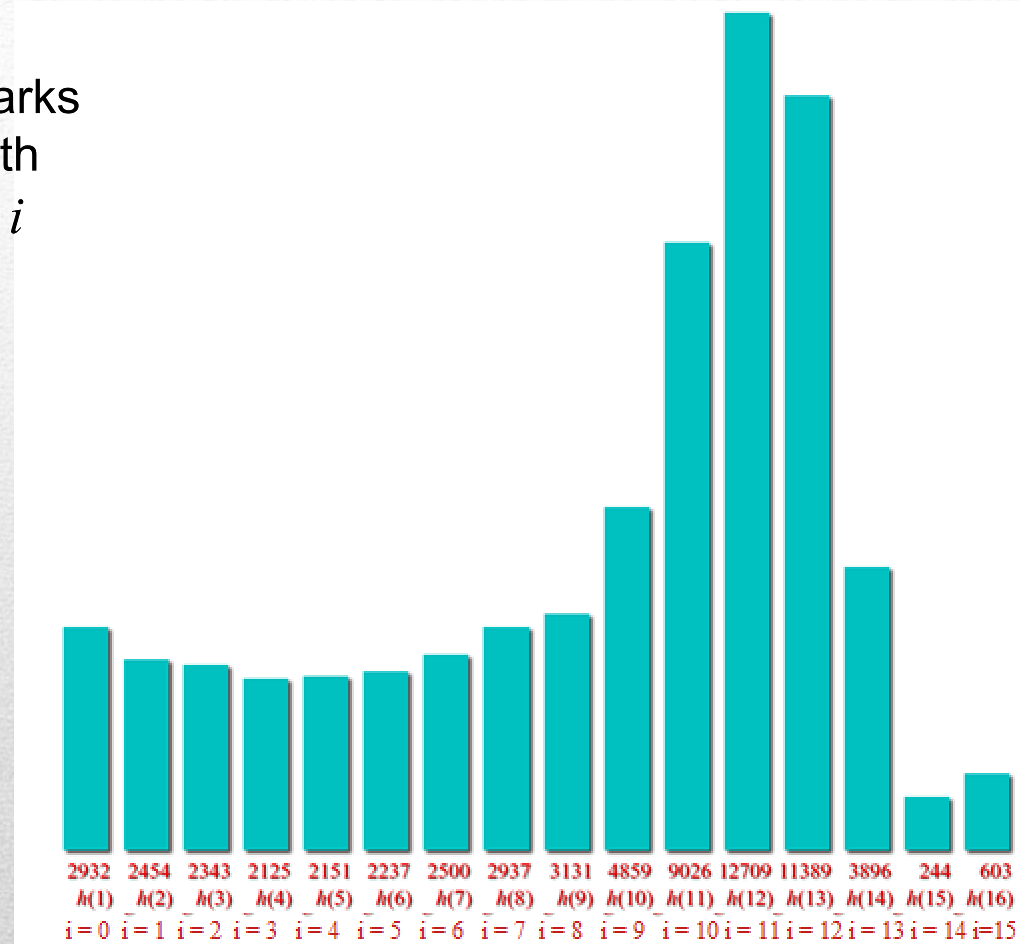
h(i)	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Grayscale image histogram

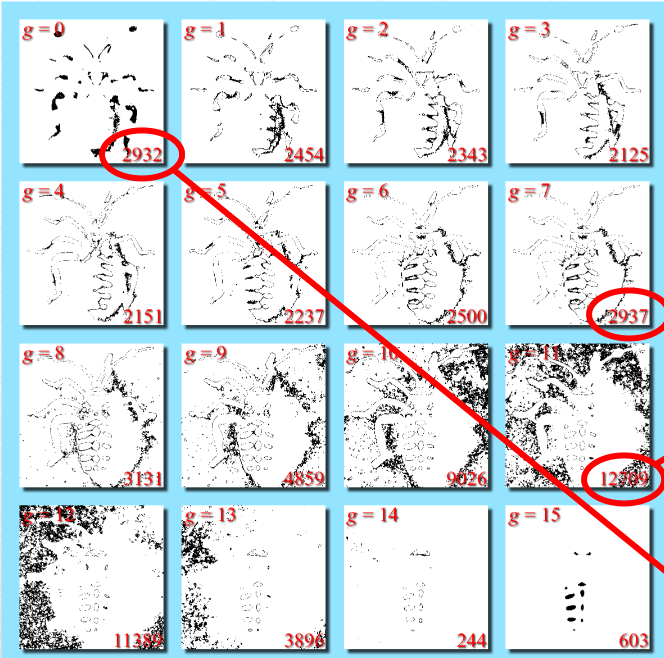


Black marks pixels with intensity i

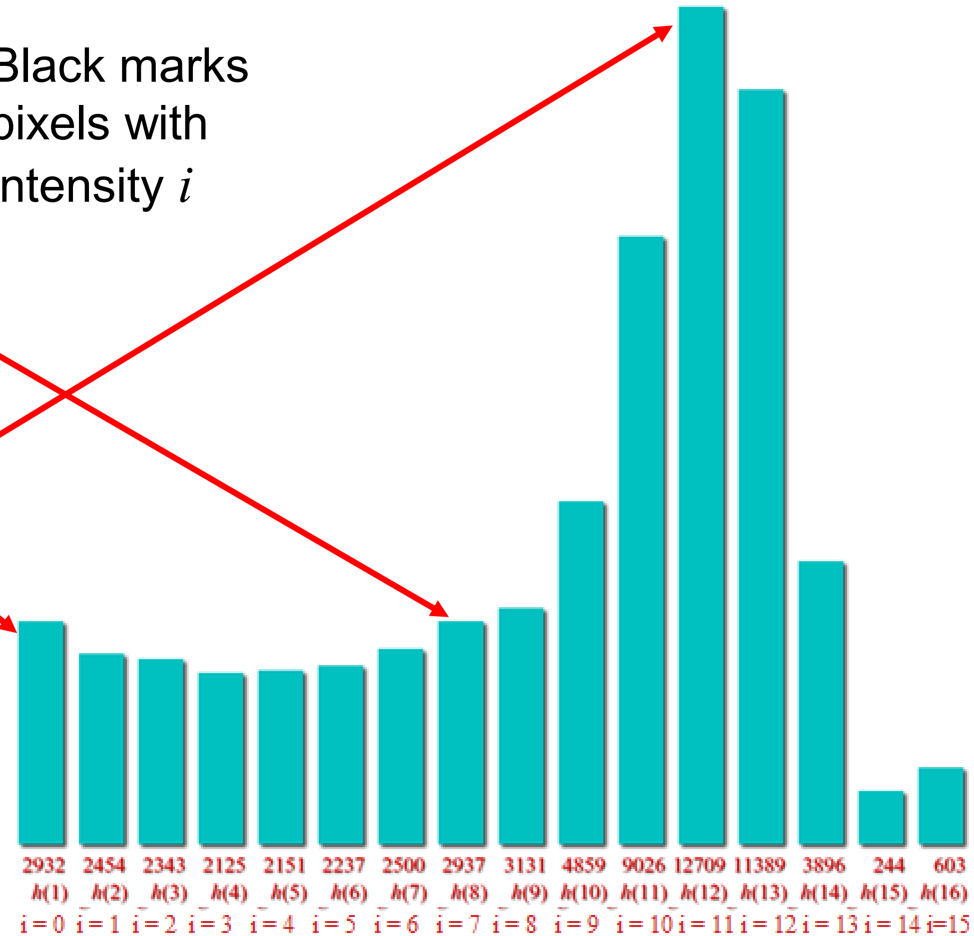
Plot of histogram:
number of pixels with intensity i



The Histogram of a Grayscale Image

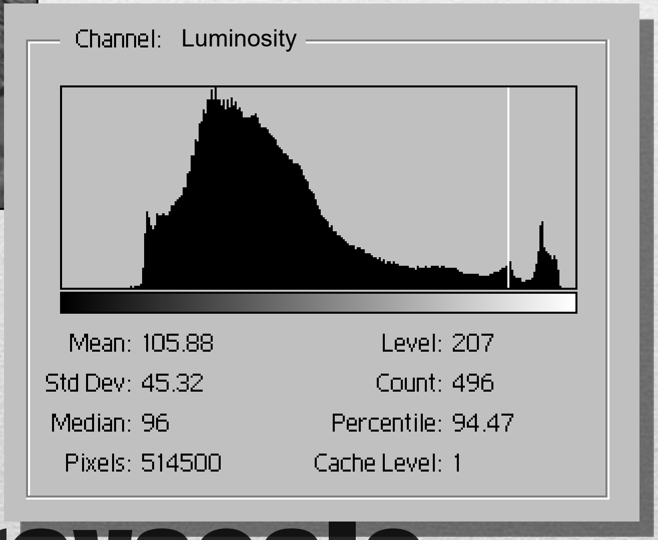


Black marks pixels with intensity i



Plot of histogram:
number of pixels with intensity i

The Histogram of a Grayscale Image



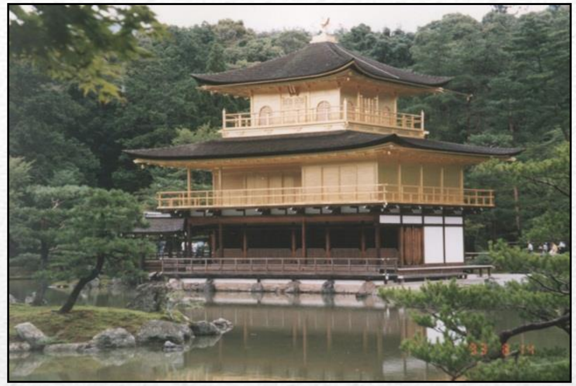
$h_I(i)$ = the number of pixels in I with intensity i

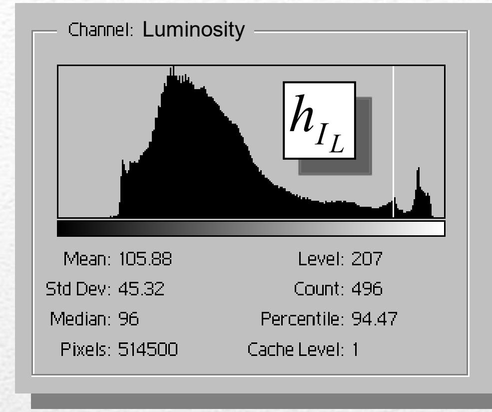
The Histogram of a Grayscale Image

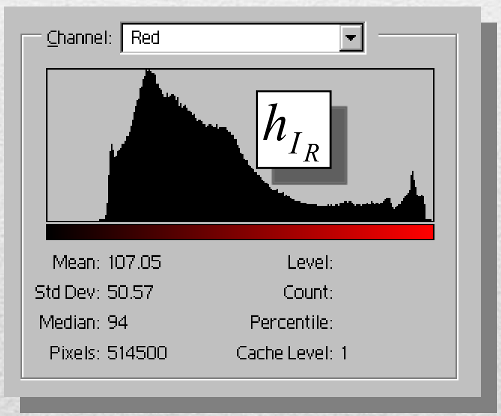
The Histogram of a Color Image

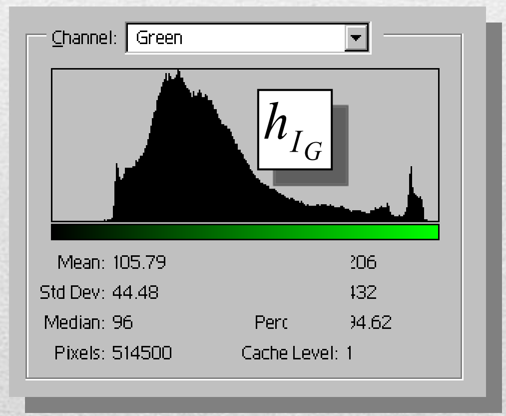
- If **I** is a 3-band image (truecolor, 24-bit)
- then **I**(*row*,*column*,*band*) is an integer between 0 and 255.
- Either **I** has 3 histograms:
  - $h_{\mathbf{R}}(i+1)$ = # of pixels in **I**(:,:,1) with intensity value i
  - $h_{\mathbf{G}}(i+1)$ = # of pixels in **I**(:,:,2) with intensity value i
  - $h_{\mathbf{B}}(i+1)$ = # of pixels in **I**(:,:,3) with intensity value i
- or 1 vector-valued histogram, $h(i,1,b)$ where
  - $h(i+1,1,1)$ = # of pixels in **I** with red intensity value i
  - $h(i+1,1,2)$ = # of pixels in **I** with green intensity value i
  - $h(i+1,1,3)$ = # of pixels in **I** with blue intensity value i

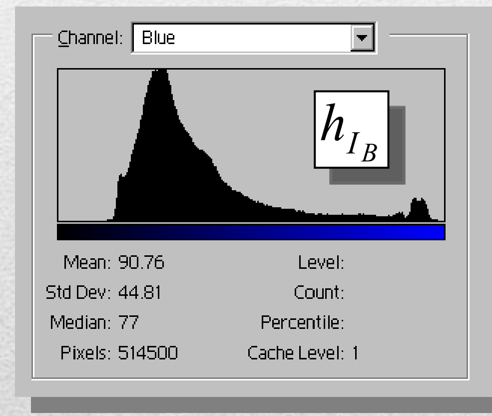
There is one histogram per color band R, G, & B. Luminosity histogram is from 1 band = (R+G+B)/3



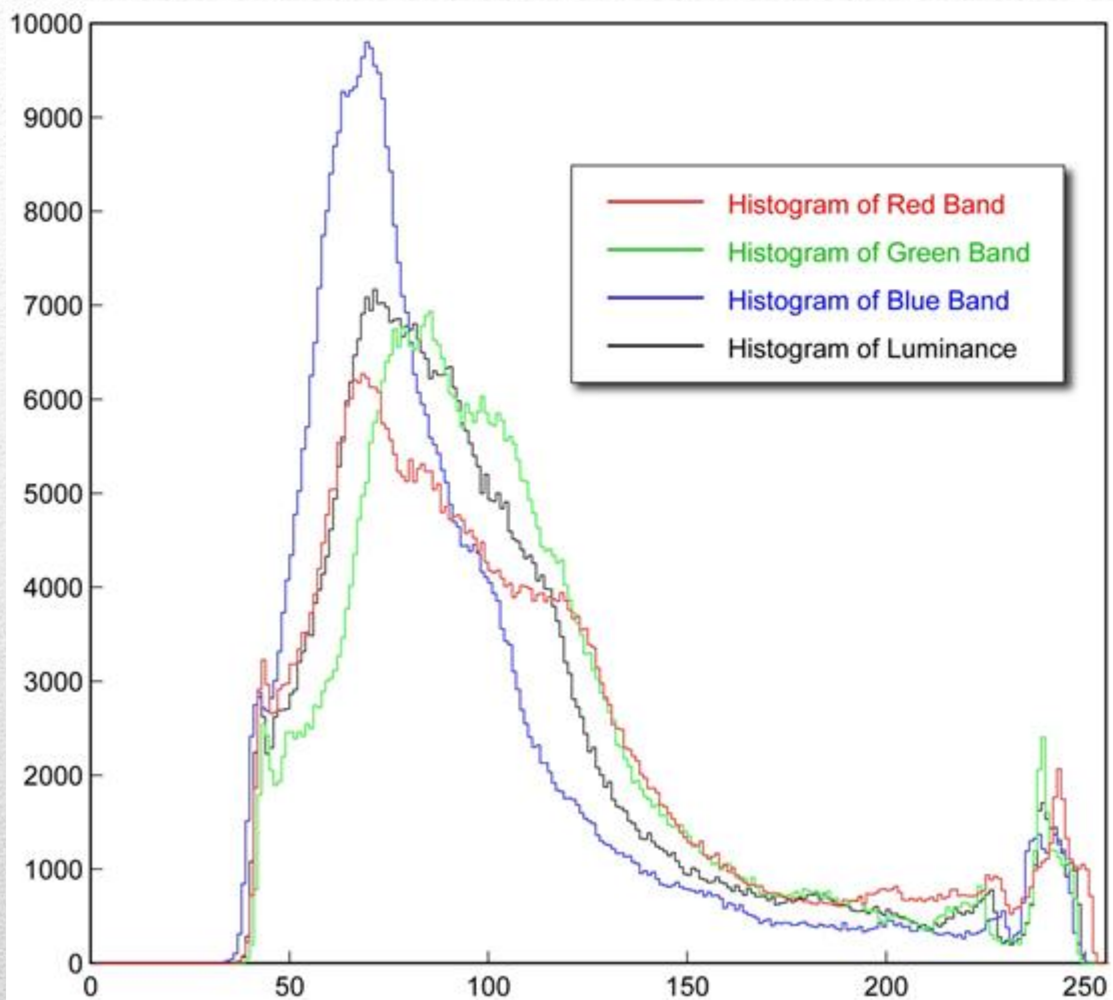








The Histogram of a Color Image



The Histogram of a Color Image

- The value histogram of a 3-band (true-color) image, **I**, is the histogram of the value image,

$$\mathbf{V}(r,c) = \frac{1}{3}\left[\boldsymbol{R}(r,c) + \boldsymbol{G}(r,c) + \boldsymbol{B}(r,c)\right]$$

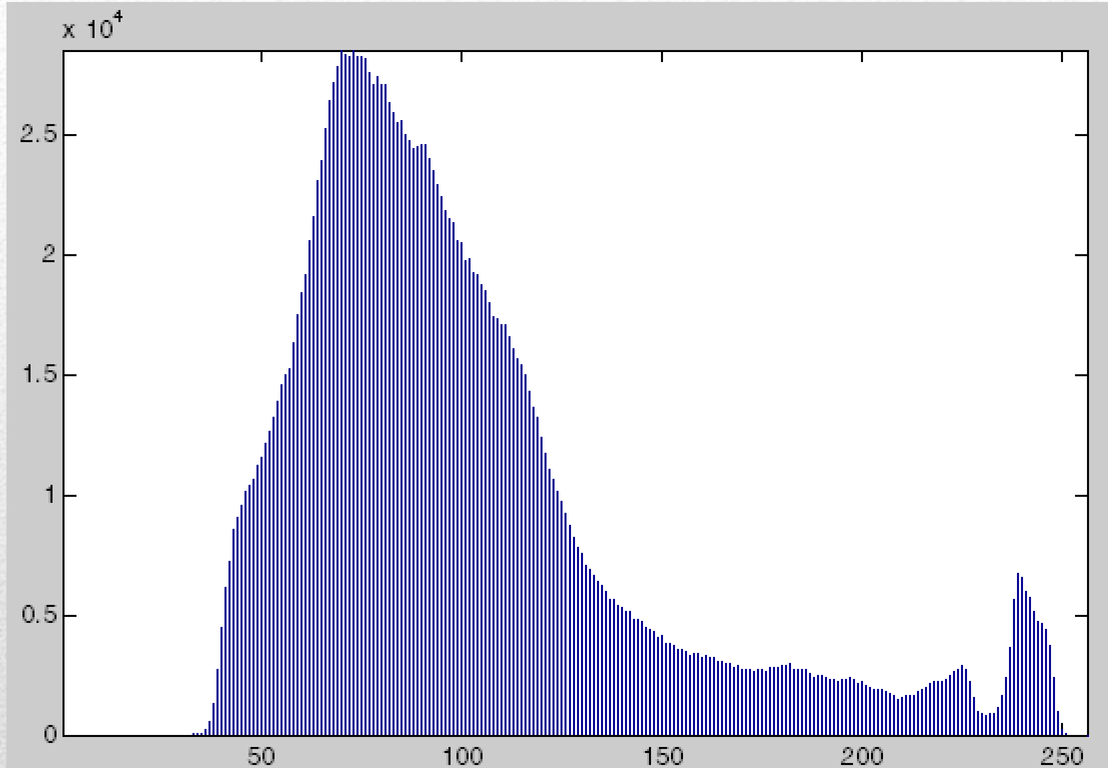
Where **R**, **G**, and **B** are the red, green, and blue bands of **I**.

- The luminance histogram of **I** is the histogram of the luminance image,

$$\mathbf{L}(r,c) = 0.299 \times \boldsymbol{R}(r,c) + 0.587 \times \boldsymbol{G}(r,c) + 0.114 \times \boldsymbol{B}(r,c)$$

Value or Luminance Histograms

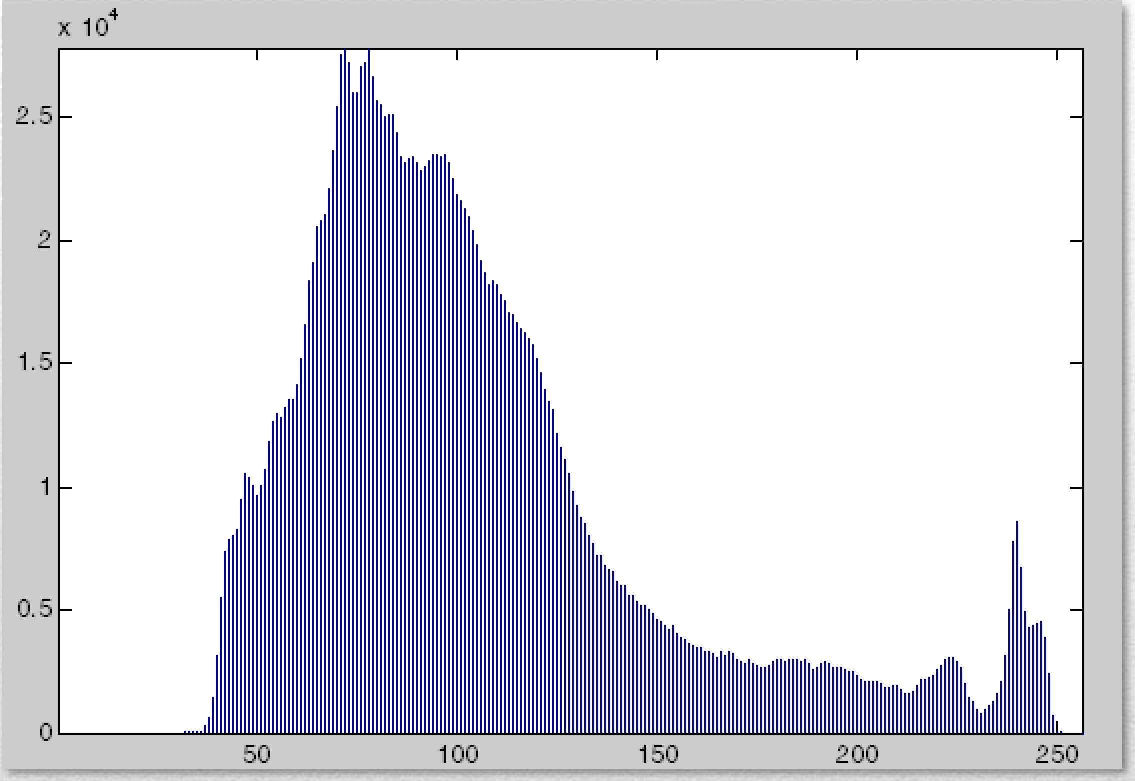
Value image, **V**.



Histogram of the value image.

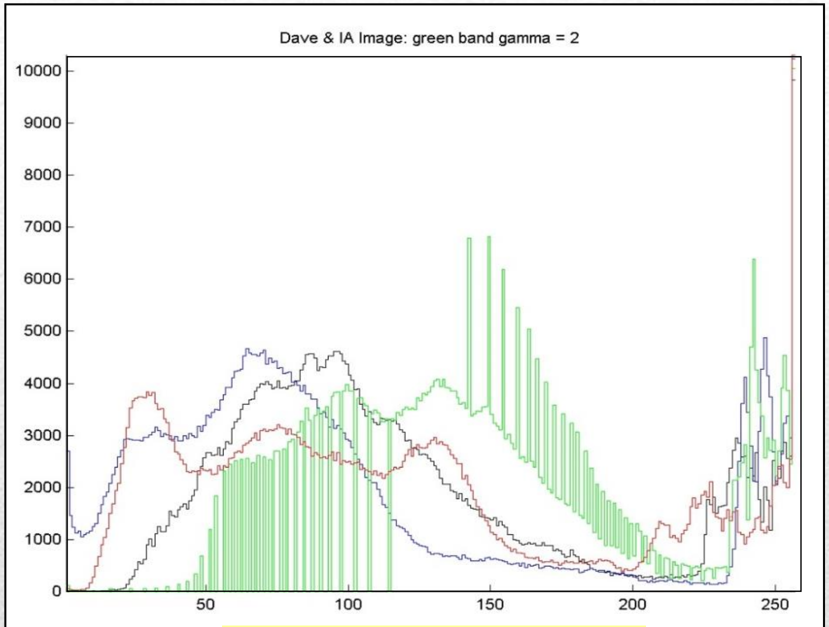
Value Histogram

Luminance image, L.

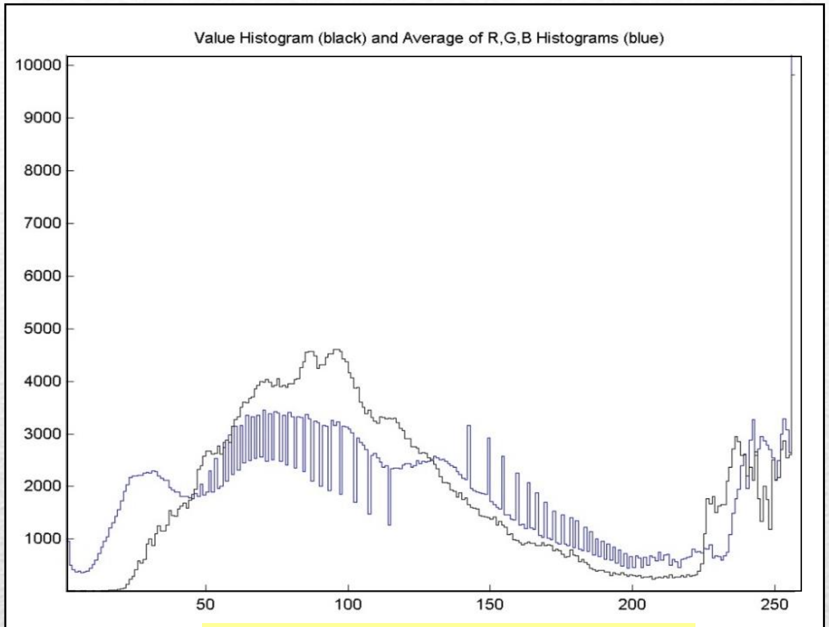


Histogram of the luminance image.

Luminance Histogram



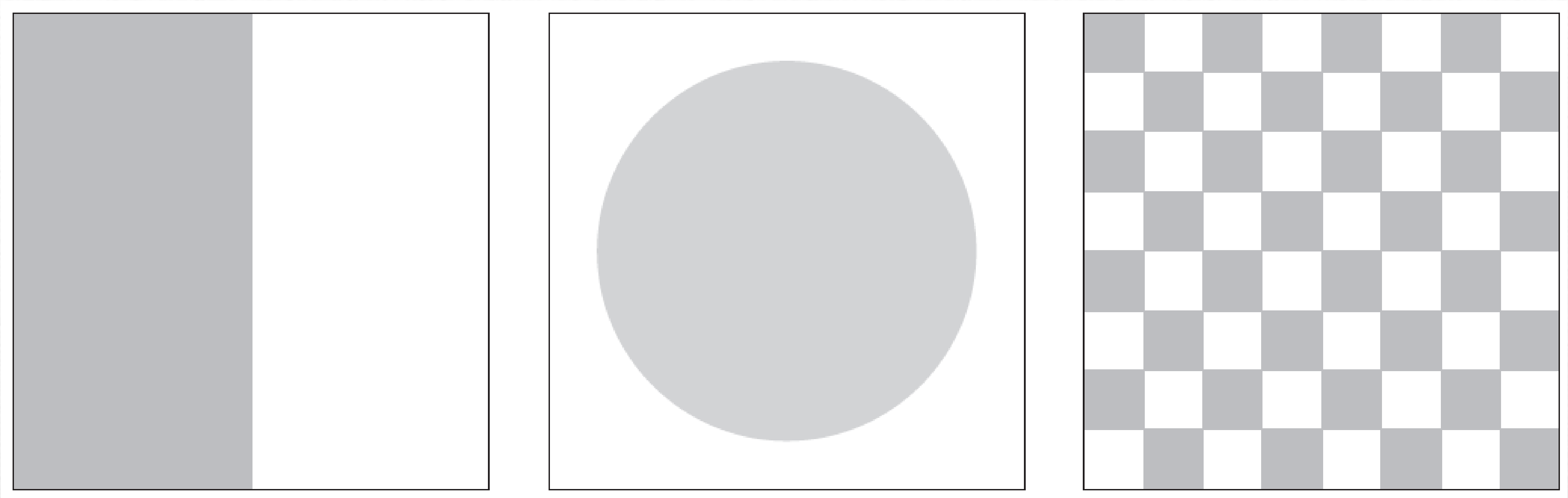
R,G,B,&V histograms



V & avg. of R,G,&B histograms

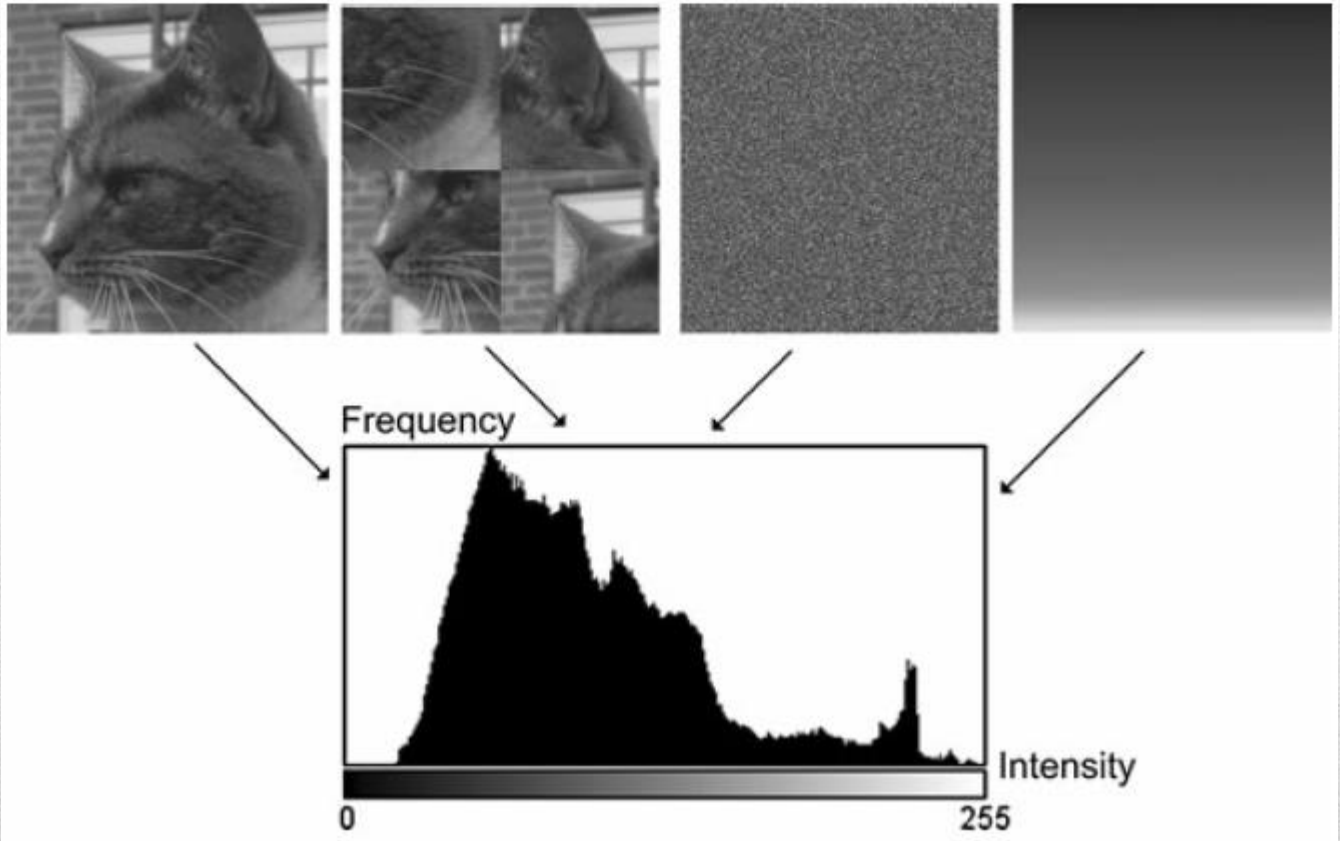
Value Histogram vs. Average of R,G,&B Histograms

➢Drawback → different images may have the same histogram as images below



Spatial distribution of colors

- Drawback → different images may have the same histogram as images below



Spatial distribution of colors

Q: What happens if I reshuffle all pixels within the image?

A: Its histogram won't change. No point processing will be affected.

Image shuffling

- Derived from the traditional histogram
- h(i) = sum of all the histogram entries h(0)…..h(i)
- Useful when performing some image processing using histograms such as image equalization

$$H(i) = \begin{cases} h(0) & \text{for } i = 0 \\ H(i-1) + h(i) & \text{for } 0 < i < K. \end{cases}$$

Cumulative Histogram

- Histograms used to evaluate image attributes
  - Brightness → quantity of luminosity perception
  - Contrast → difference in luminosity between darkest and brightest areas

a. Dark image → concentration of bars on the lower end

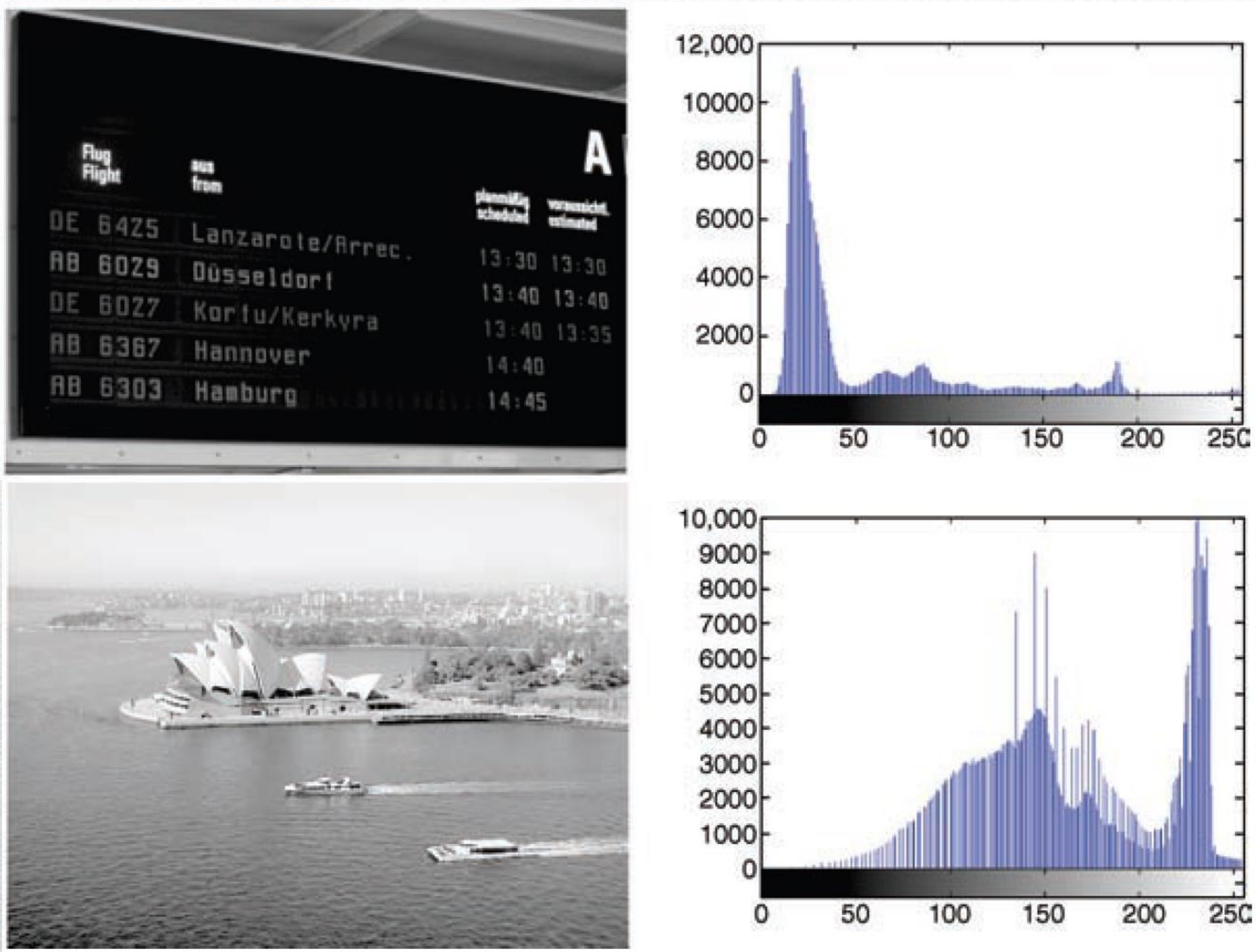
b. Bright image → concentration of bars on the upper end

c. Low contrast image → histogram is clustered within small range of levels

d. High contrast image → opposite

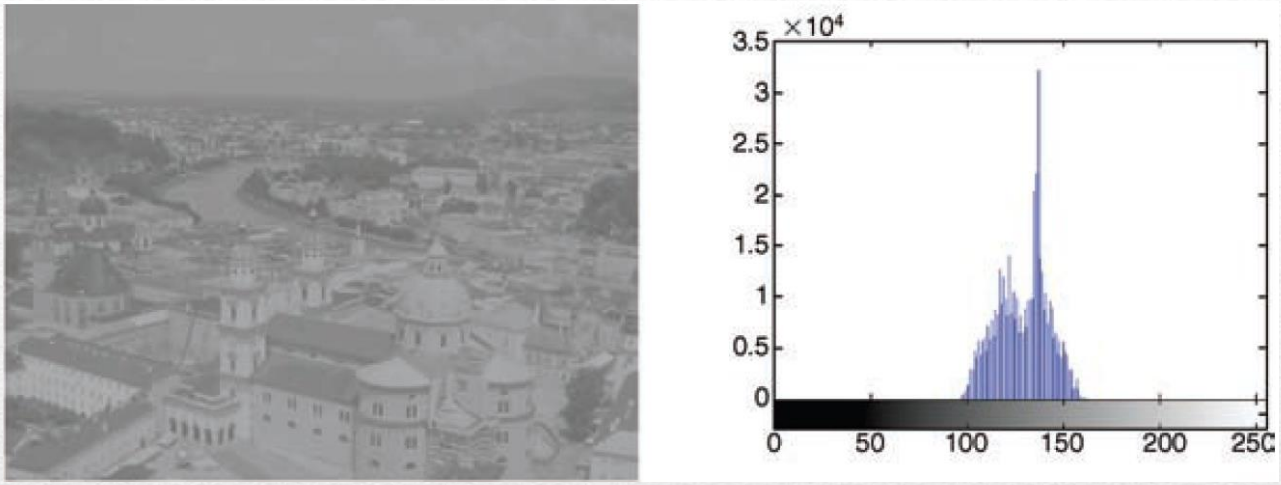
Interpreting Image Histograms

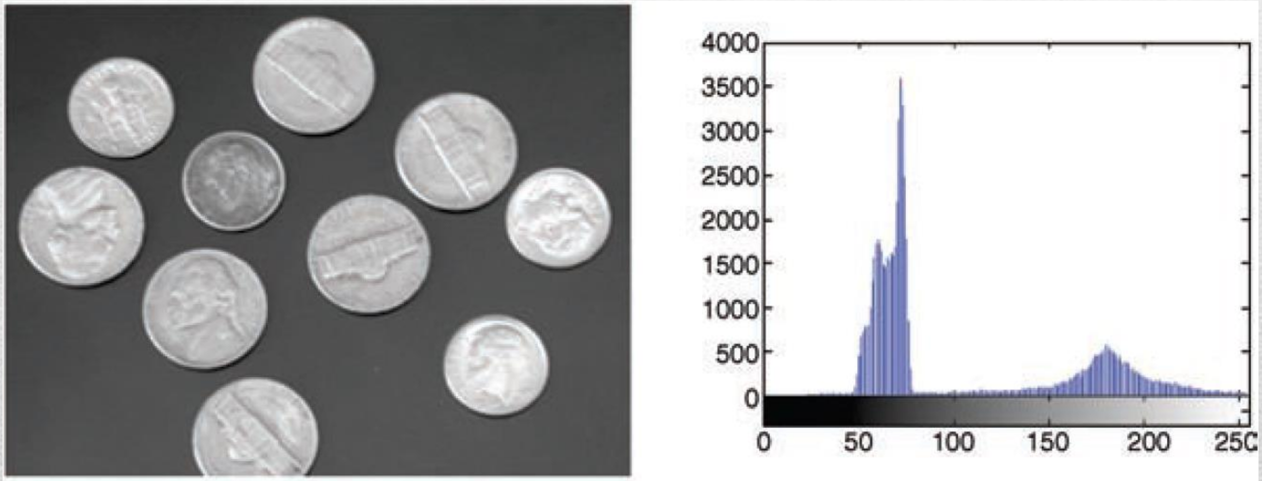
a. Dark image & Bright image



Interpreting Image Histograms

a. Low contrast image & High contrast image



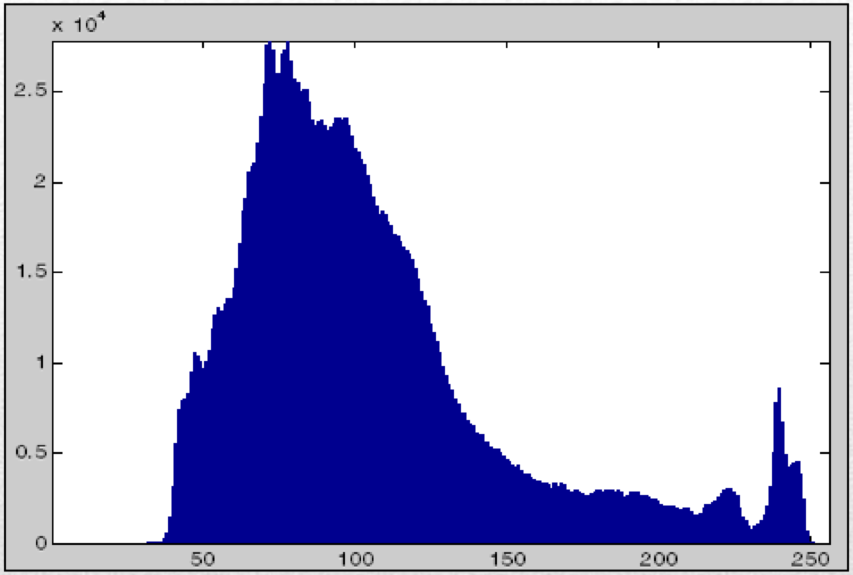


Interpreting Image Histograms

- perform a modification of the pixel values without changing:
  - size,
  - geometry, or
  - local structure of the image
- Each new pixel value $a' = I'(u, v) = f(\text{a})$
  - depends exclusively on the previous value of a
  - independent from any other pixel value, in particular from any of its neighboring pixels.

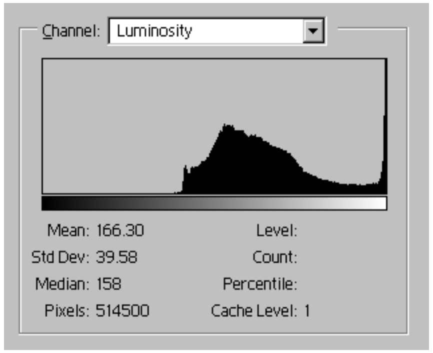
If f() is independent of image coordinates → global or homogeneous operation f(I(i,j)) = a
Otherwise → non-homogeneous operation f(I(i,j), i, j) = a

Point Operations



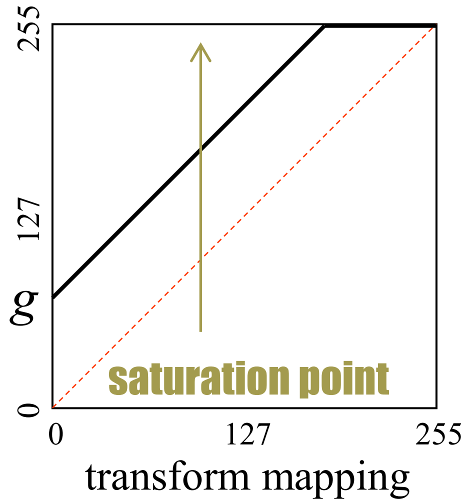
Luminance Histogram

Point Processes: Original Image



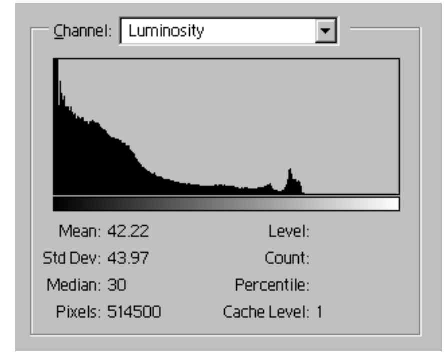
$$\mathbf{J}(r,c,b) = \begin{cases} \mathbf{I}(r,c,b) + g, & \text{if } \mathbf{I}(r,c,b) + g < 256 \\ 255, & \text{if } \mathbf{I}(r,c,b) + g > 255 \end{cases}$$

$g \geq 0$ and $b \in \{1,2,3\}$ is the band index.



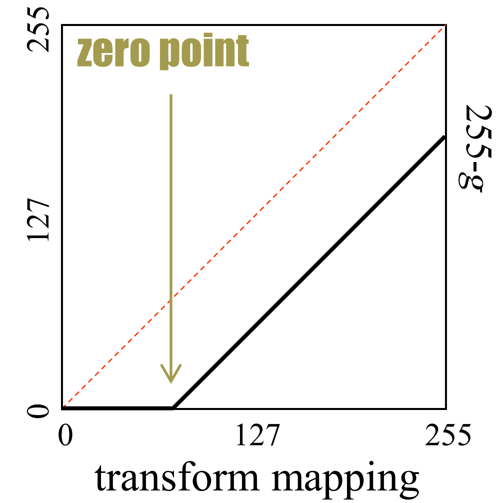
Point Processes: Increase Brightness

Point Processes: Decrease Brightness

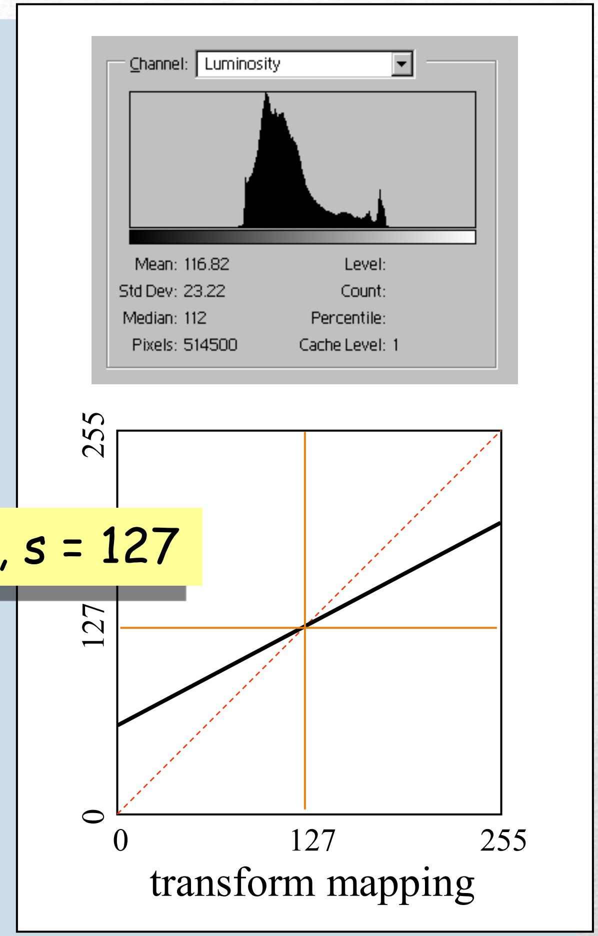


$$\mathbf{J}(r,c,b) = \begin{cases} 0, & \text{if } \mathbf{I}(r,c,b) - g < 0 \\ \mathbf{I}(r,c,b) - g, & \text{if } \mathbf{I}(r,c,b) - g > 0 \end{cases}$$

$g \geq 0$ and $b \in \{1,2,3\}$ is the band index.



Point Processes: Decrease Contrast



Here, s = 127

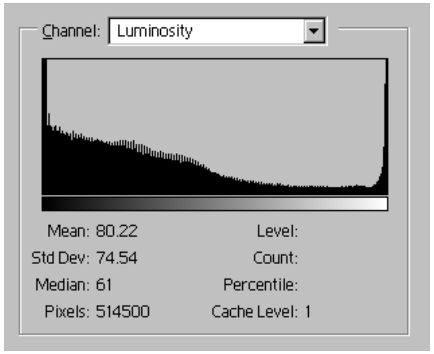
$$\mathbf{T}(r,c,b) = a\left[\mathbf{I}(r,c,b) - s\right] + s,$$

where $0 \le a < 1.0$,

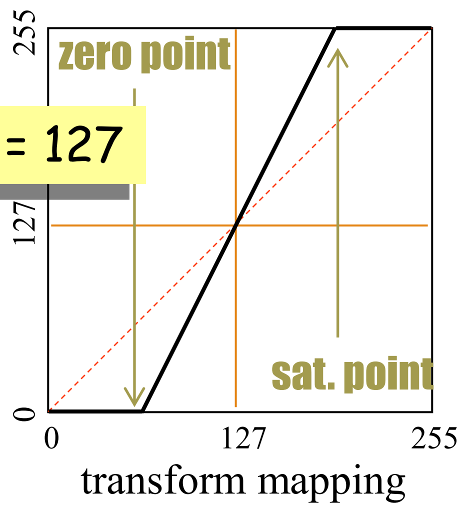
$s \in \{0,1,2,\ldots,255\}$, and

$b \in \{1,2,3\}$.

s is the center of the contrast function.



Here, s = 127



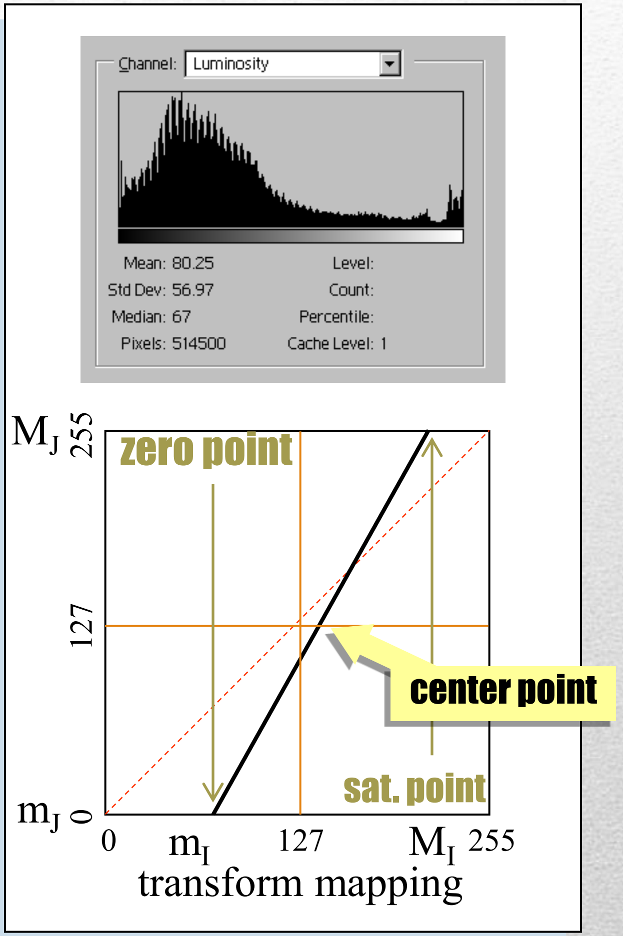
$$\mathbf{T}(r,c,b) = a\left[\mathbf{I}(r,c,b) - s\right] + s$$

$$\mathbf{J}(r,c,b) = \begin{cases} 0, & \text{if } \mathbf{T}(r,c,b) < 0, \\ \mathbf{T}(r,c,b), & \text{if } 0 \le \mathbf{T}(r,c,b) \le 255, \\ 255, & \text{if } \mathbf{T}(r,c,b) > 255. \end{cases}$$

$$a > 1, \quad s \in \{0,\ldots,255\}, \quad b \in \{1,2,3\}$$

Point Processes: Increase Contrast

- Modify the pixels such that the available range of values is fully covered (stretching)



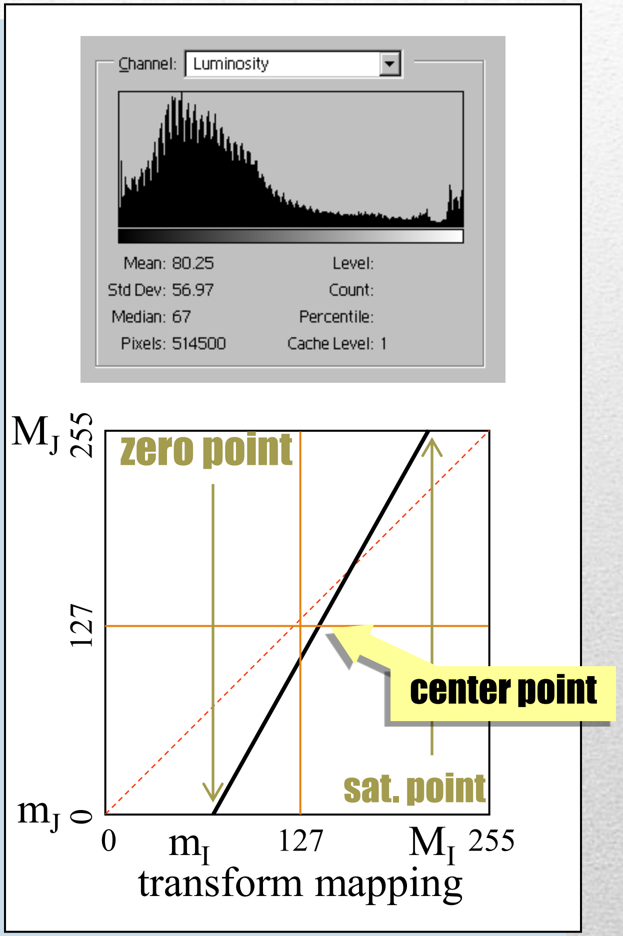
Let m_I = min[I(r,c)], M_I = max[I(r,c)]

$$J(r,c) = \frac{L-1}{M_i - m_i} \times (I(r,c) - m_i)$$

Range strectched to [0 L-1]
L=highest gray level you want to map to.
For full range use 255

Automatic Contrast Adjustment

- Auto-contrast based on another image or to a specific range not full range.



Let $m_{\mathbf{I}} = \min[\mathbf{I}(r,c)], \quad M_{\mathbf{I}} = \max[\mathbf{I}(r,c)],$

$m_{\mathbf{J}} = \min[\mathbf{J}(r,c)], \quad M_{\mathbf{J}} = \max[\mathbf{J}(r,c)].$

Then,

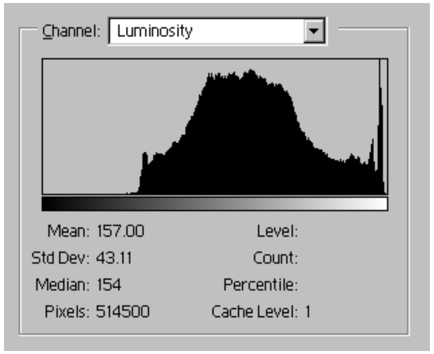
$$\mathbf{J}(r,c) = (M_{\mathbf{J}} - m_{\mathbf{J}})\frac{\mathbf{I}(r,c) - m_{\mathbf{I}}}{M_{\mathbf{I}} - m_{\mathbf{I}}} + m_{\mathbf{J}}.$$

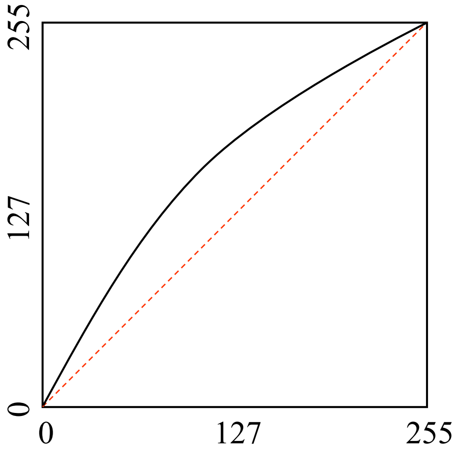
AutoContrast

- Camera, video recorders do not capture luminance correctly → they tend to be more darker and constrasted
  - If 1 volt generate 100% luminance value, 0.5 generate a luminance 18% → cathode tubes in display devices are not linear.

Gamma

$$\mathbf{J}(r,c) = 255 \cdot \left[\frac{\mathbf{I}(r,c)}{255}\right]^{\frac{1}{\gamma}} \quad \text{for} \quad \gamma > 1.0$$

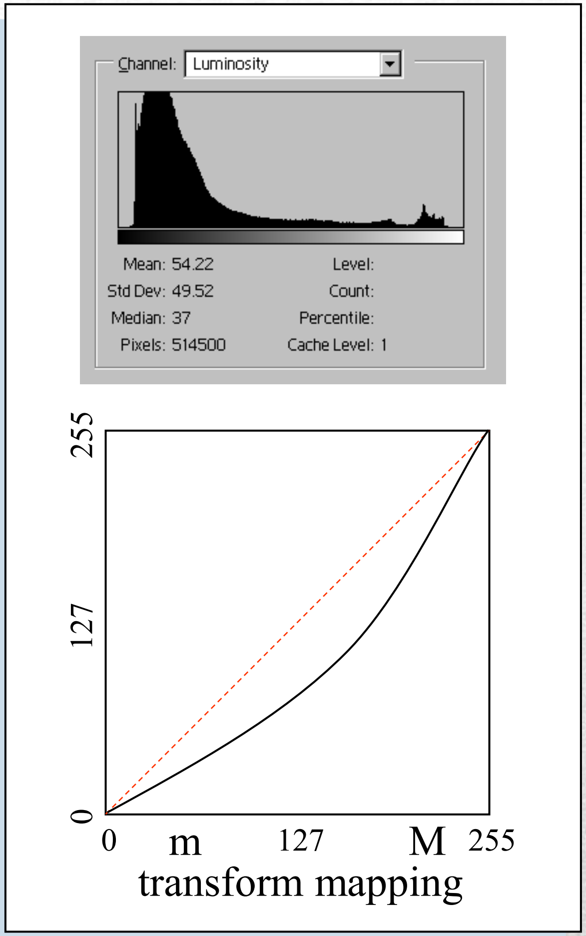




transform mapping

Point Processes: Increased Gamma

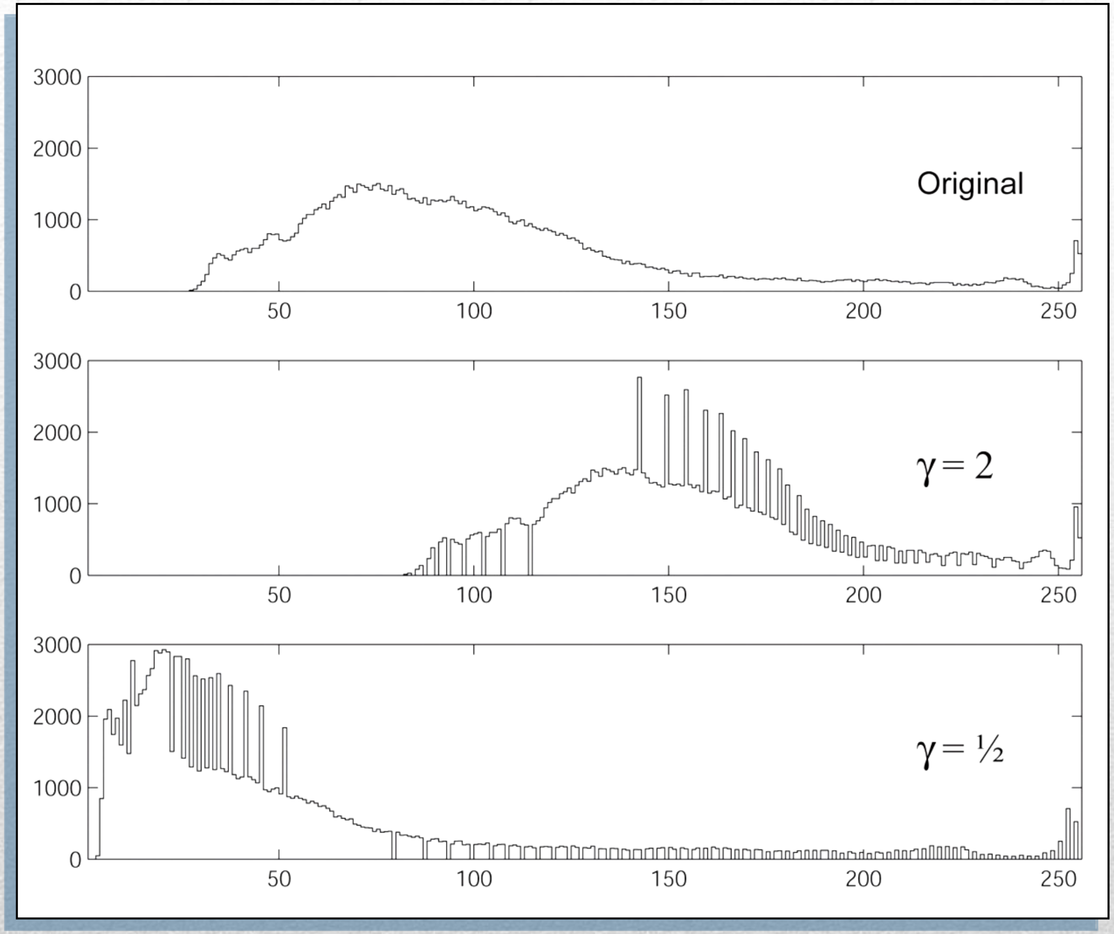
2025 / 2026



$$\mathbf{J}(r,c) = 255 \cdot \left[\frac{\mathbf{I}(r,c)}{255}\right]^{\frac{1}{\gamma}} \quad \text{for} \quad \gamma < 1.0$$

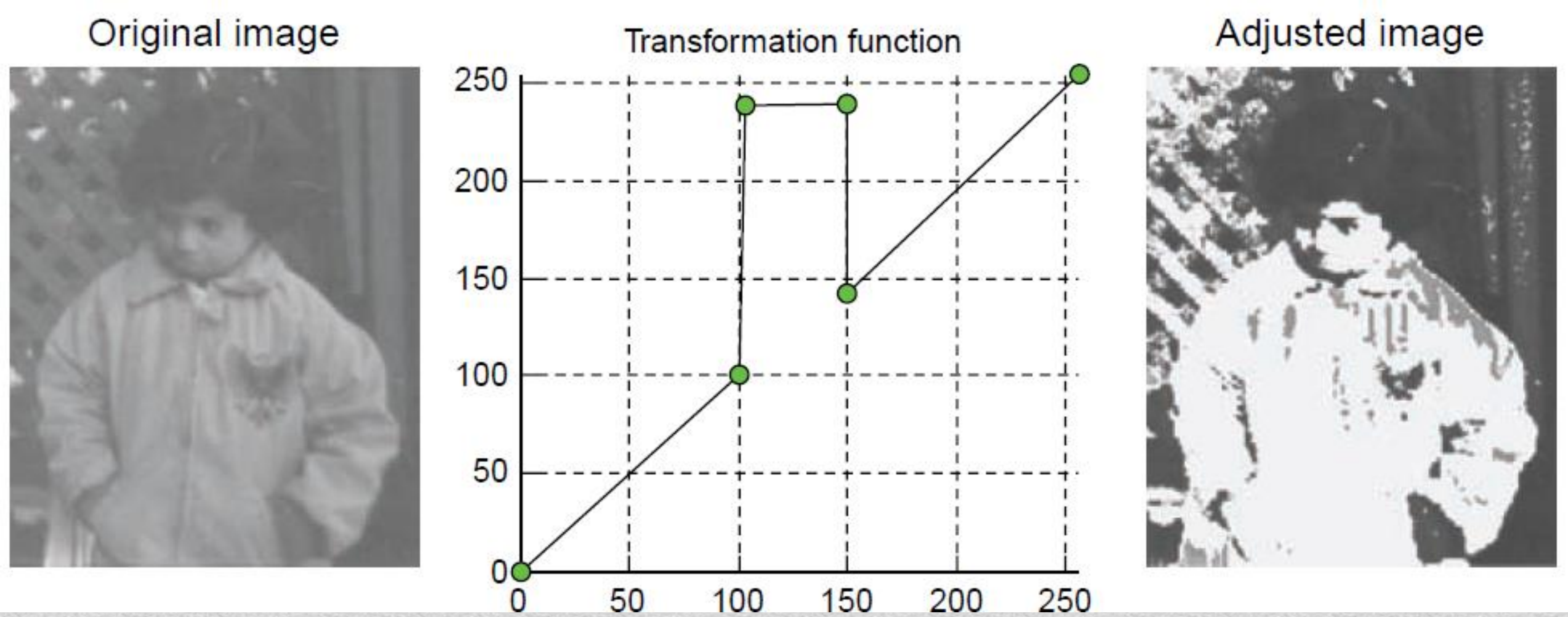
Point Processes: Decreased Gamma

Dr. Zein Ibrahim



Gamma Correction: Effect on Histogram

- Several linear equations, one for each interval of gray-level values
  - Used to improve the contrast

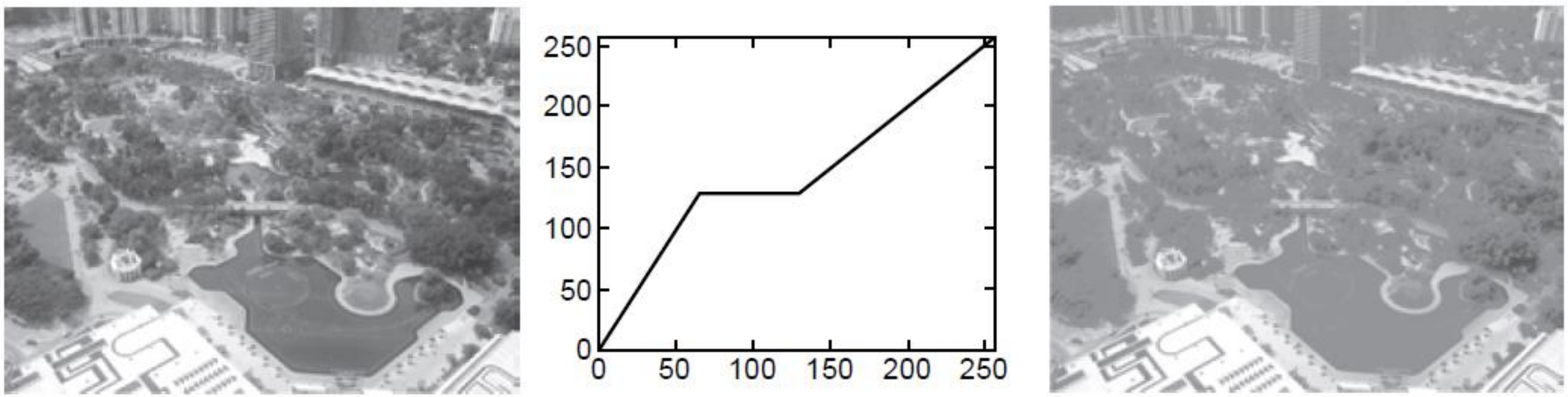


Piecewise Linear Transformation

$$s = \begin{cases} 2 \cdot f & \text{for } 0 < r \le 64 \\ 128 & \text{for } 64 < r \le 128 \\ f & \text{for } r > 128 \end{cases}$$

input array is $\begin{bmatrix} 20 & 40 & 0 \\ 178 & 198 & 64 \\ 77 & 128 & 1 \end{bmatrix}$

resulting array will be $\begin{bmatrix} 40 & 80 & 0 \\ 178 & 198 & 128 \\ 128 & 128 & 2 \end{bmatrix}$



(a) input image (b) transformation function (c) output image

Piecewise Linear Transformation - Example

PDF and CDF

- h(i) → number of pixels in I having the value I
- If I is an M.N image $\sum_{i=0}^{K-1} \mathsf{h}(i) = M \cdot N$
- Normalized histogram $\mathsf{p}(i) = \frac{\mathsf{h}(i)}{MN}, \quad \text{for } 0 \le i < K$
  - Called also the probability distribution (function) (pdf)
- CDF is the cumulative distribution function

$$\mathsf{P}(i) = \frac{\mathsf{H}(i)}{\mathsf{H}(K-1)} = \frac{\mathsf{H}(i)}{MN} = \sum_{j=0}^{i} \frac{\mathsf{h}(j)}{MN} = \sum_{j=0}^{i} \mathsf{p}(j), \quad \text{for } 0 \le i < K$$

$$\mathsf{P}(0) = \mathsf{p}(0) \qquad \text{and} \qquad \mathsf{P}(K-1) = \sum_{i=0}^{K-1} \mathsf{p}(i) = 1$$

CDF(h)

Returns the cumulative distribution function $\mathsf{P}(i) \in [0,1]$ for a given histogram $\mathsf{h}(i)$, with $i = 0, \ldots K-1$.

Let $K \leftarrow \text{Size}(\mathsf{h})$

Let $n \leftarrow \sum_{i=0}^{K-1} \mathsf{h}(i)$

Create table P of size K

Let $c \leftarrow 0$

for $i \leftarrow 0 \ldots (K-1)$ **do**

  $c \leftarrow c + \mathsf{h}(i)$ ▷ cumulate histogram values

  $\mathsf{P}(i) \leftarrow c/n$

return P.

CDF Computation Algorithm

- Task: remap image **I** so that its histogram is as close to obtain a uniform (flat) resulting histogram.
  - Percentage of pixels of every level is the same.
  - Is essential for enhancing details and improving visual quality in low-contrast images.
- Should use a transformation function T(r):
  - $T(r)$ must be a monotonically increasing function in the interval $0 \le r \le L-1$.
  - $0 \le T(r) \le L-1$ for $0 \le r \le L-1$.
- Most used T(r) is the cdf (cumulative distribution function)

all bands processed similarly

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p(r_j)$$

Multiply the values with nb of gray levels

$$\mathbf{J}(r,c,b) = 255 \times P_{\mathbf{I}}[\mathbf{I}(r,c,b)+1]$$

Point Processes: Histogram Equalization

Using the *cdf* as the transformation function, we can calculate

$$s_0 = T(r_0) = \sum_{j=0}^{0} p(r_j) = p(r_0) = 0.068$$

$$s_1 = T(r_1) = \sum_{j=0}^{1} p(r_j) = p(r_0) + p(r_1) = 0.264$$

and $s_2 = 0.560$, $s_3 = 0.769$, $s_4 = 0.891$, $s_5 = 0.939$, $s_6 = 0.972$, and $s_7 = 1$.

Since the image was quantized with only eight gray levels, each value of s_k must be rounded to the closest valid (multiple of 1/7) value.

Thus, $s_0 \simeq 0$, $s_1 \simeq 2$, $s_2 \simeq 4$, $s_3 \simeq 5$, $s_4 \simeq 6$, $s_5 \simeq 7$, $s_6 \simeq 7$, and $s_7 \simeq 7$.

Example of a Histogram

Gray Level (r_k)	n_k	$p(r_k)$
0	1120	0.068
1	3214	0.196
2	4850	0.296
3	3425	0.209
4	1995	0.122
5	784	0.048
6	541	0.033
7	455	0.028
Total	16,384	1.000

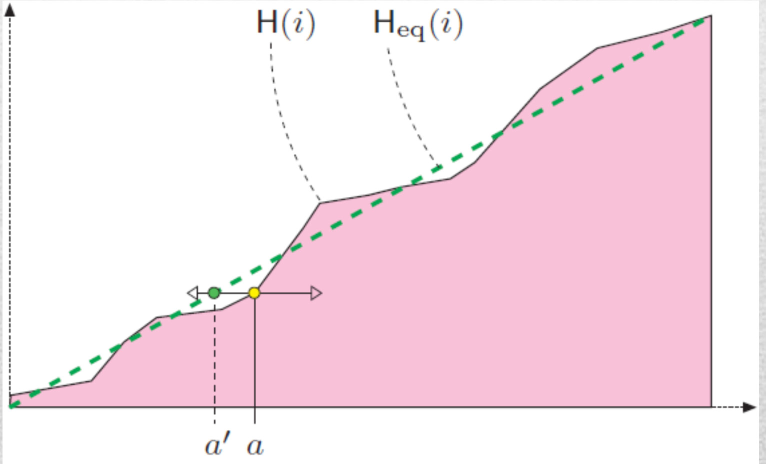
Gray Level (s_k)	n_k	$p(s_k)$
0	1120	0.068
1	0	0.000
2	3214	0.196
3	0	0.000
4	4850	0.296
5	3425	0.209
6	1995	0.122
7	1780	0.109
Total	16,384	1.000

Example

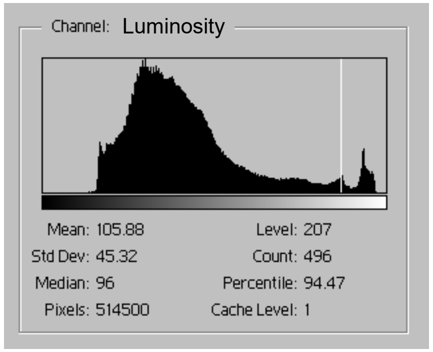
- Other formulation: $$f_{eq}(a) = \left\lfloor \mathsf{H}(a) \cdot \frac{K-1}{MN} \right\rfloor$$

  H is the cumulative histogram for an image where pixel values is in [0, k-1]

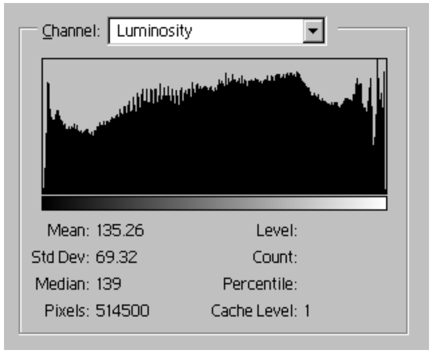
- The idea is to obtain as possible a linear cumulative histogram



Point Processes: Histogram Equalization

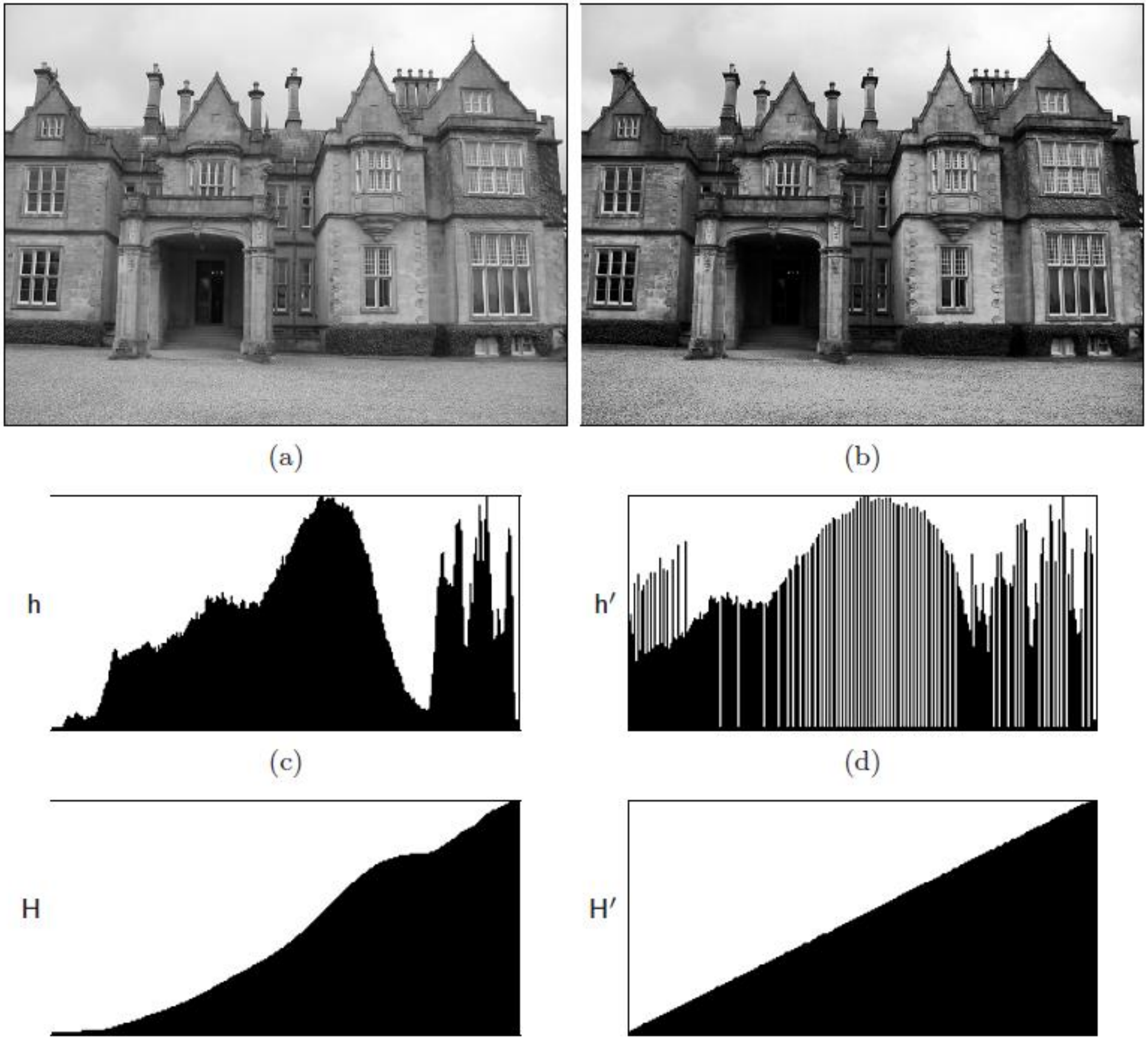


before



after

Point Processes: Histogram Equalization

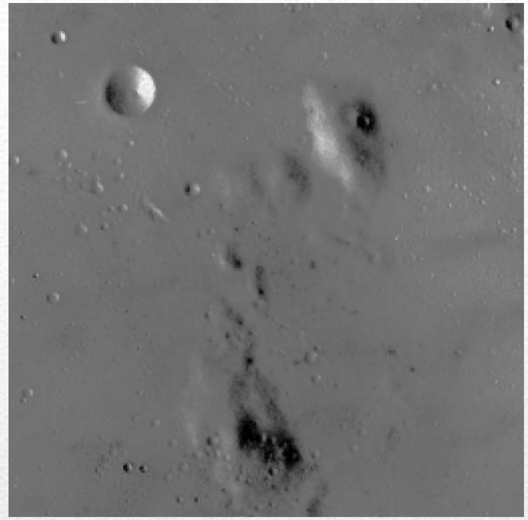


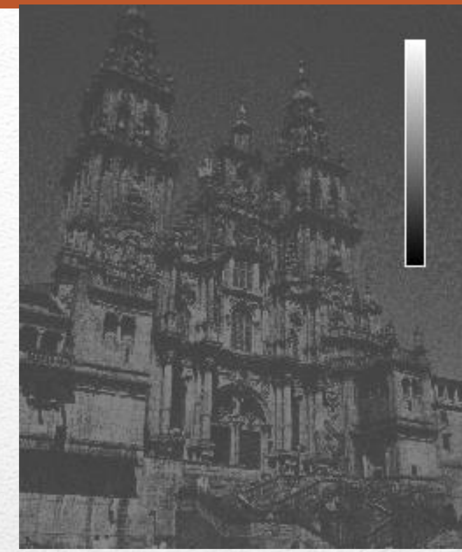
(a) (b)

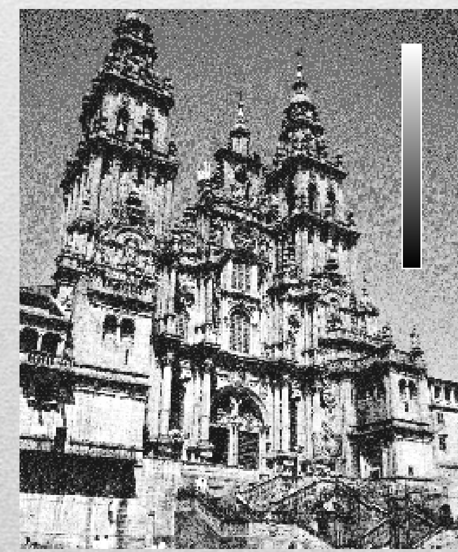
(c) (d)

(e) (f)

Point Processes: Histogram Equalization







Equalized images

- Task: remap image **I** with min = $m_{\mathbf{I}}$ and max = $M_{\mathbf{I}}$ so that its histogram is as close to constant as possible and has min = $m_{\mathbf{J}}$ and max = $M_{\mathbf{J}}$.
- Let $P_I(i+1)$ be the cumulative (probability) distribution function of **I**.
- Then **J** has, as closely as possible, the correct histogram if

$$\mathbf{J}(r,c) = (M_{\mathbf{J}} - m_{\mathbf{J}}) \frac{P_{\mathbf{I}}[\mathbf{I}(r,c)+1] - P_{\mathbf{I}}(m_{\mathbf{I}}+1)}{P_{\mathbf{I}}(M_{\mathbf{I}}+1) - P_{\mathbf{I}}(m_{\mathbf{I}}+1)} + m_{\mathbf{J}}.$$

Point Processes: Histogram Equalization

- Good images never have uniform distribution
  - Perhaps similar to a Gaussian distribution
- Why: images acquired with the same local illumination, over the same location, but with different sensors, atmospheric conditions or global illumination → **Color Adjustment**
- Task: remap image **I** so that it has, as closely as possible, the same histogram as image **J**.
- Because images are digital it is not, in general, possible to make $h_{\mathbf{I}} \equiv h_{\mathbf{J}}$.

Q: How, then, can the matching be done?
A: By matching percentiles.

Histogram Matching (Specification)

Recall:

- CDF of image $\mathbf{I}$ is such that $0 \leq P_{\mathbf{I}}(g_{\mathbf{I}}) \leq 1$.
- $P_{\mathbf{I}}(g_{\mathbf{I}}+1) = c$ means that c is the fraction of pixels in $\mathbf{I}$ that have a value less than or equal to $g_{\mathbf{I}}$.
- $100c$ is the *percentile* of pixels in $\mathbf{I}$ that are less than or equal to $g_{\mathbf{I}}$.

… assuming a 1-band image or a single band of a color image.

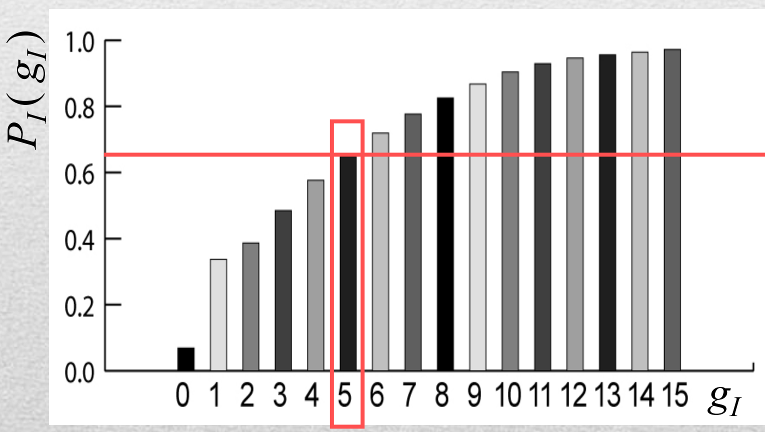
To match percentiles, replace all occurrences of value g_I in image I with the value, g_J, from image J whose percentile in J most closely matches the percentile of g_I in image I.

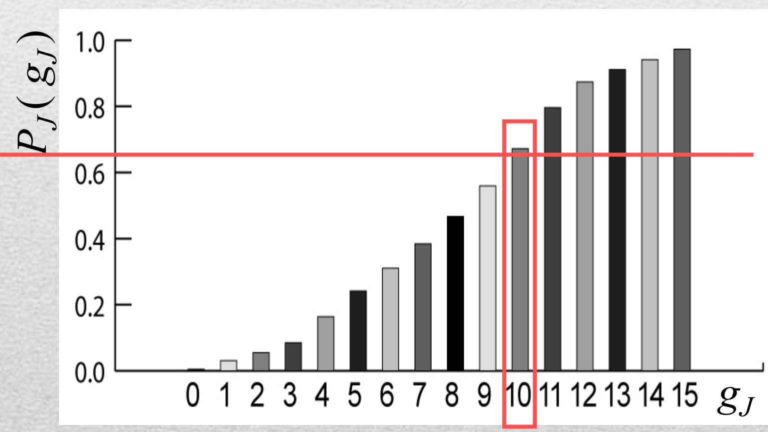
Matching Percentiles

- To create an image, **K**, from image **I** such that **K** has nearly the same CDF as image **J** do the following:
- If $\mathbf{I}(r,c) = g_{\mathbf{I}}$ then let $K(r,c) = g_{\mathbf{J}}$ where $g_{\mathbf{J}}$ is such that

$$P_{\mathbf{I}}(g_{\mathbf{I}}) > P_{\mathbf{J}}(g_{\mathbf{J}} - 1) \ \ \textbf{AND} \ \ P_{\mathbf{I}}(g_{\mathbf{I}}) \le P_{\mathbf{J}}(g_{\mathbf{J}})$$

… assuming a 1-band image or a single band of a color image.



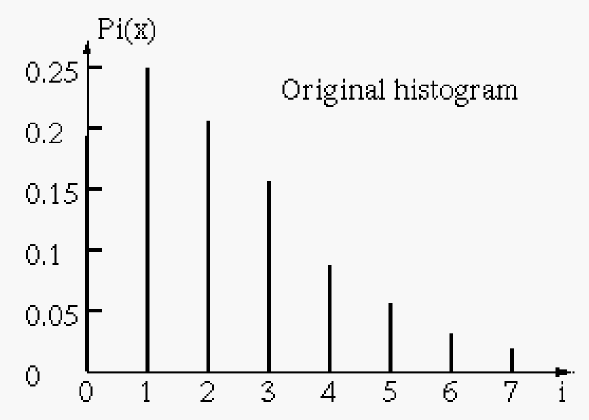


Example:
$\mathbf{I}(r,c) = 5$
$P_{\mathbf{I}}(5) = 0.65$
$P_{\mathbf{J}}(9) = 0.56$
$P_{\mathbf{J}}(10) = 0.67$
$\mathbf{K}(r,c) = 10$

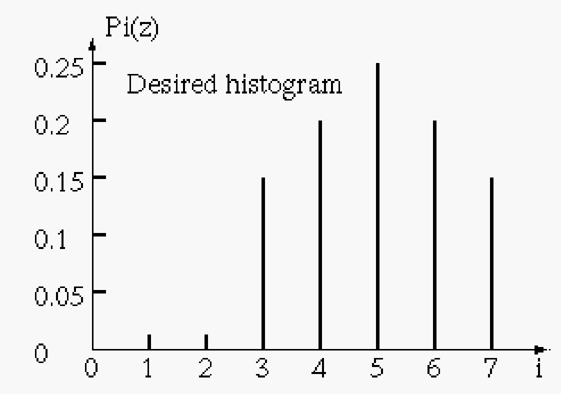
Matching Percentiles

- Suppose having the following image information

x_i	n_j	h_x	$y = H_x$
0	790	0.19	0.19
1	1023	0.25	0.44
2	850	0.21	0.65
3	656	0.16	0.81
4	329	0.08	0.89
5	245	0.06	0.95
6	122	0.03	0.98
7	81	0.02	1.00



z_i	p_z	$y' = H_z$
0	0.0	0.0
1	0.0	0.0
2	0.0	0.0
3	0.15	0.15
4	0.20	0.35
5	0.30	0.65
6	0.20	0.85
7	0.15	1.0



Example: Matching histograms

- Suppose having the following image information

$x_i = i$	$y_j = H_x$
0	0.19
1	0.44
2	0.65
3	0.81
4	0.89
5	0.95
6	0.98
7	1.0

The closest value

$y'_j = H_z$
0.0
0.0
0.0
0.15
0.35
0.65
0.85
1.0

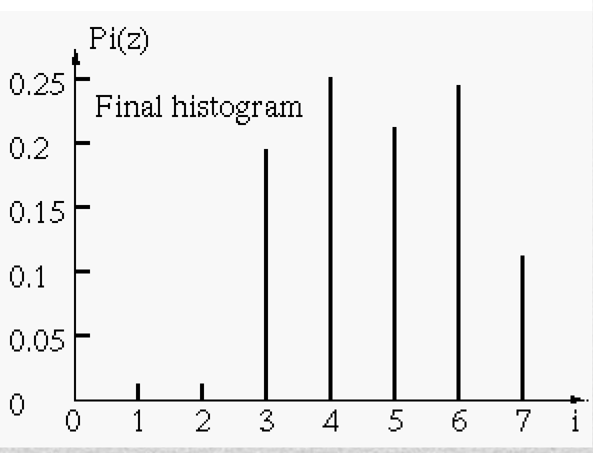
Gray level 0 is closest to the level 3 in the target histogram so replace 0 by 3

Example: Matching histograms

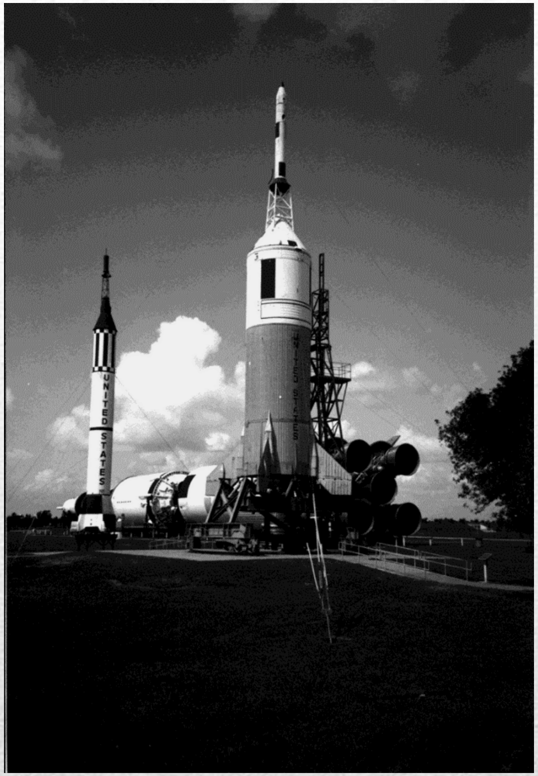
- Gray level 0 in original is replaced by 3
- Gray level 1 in original is replaced by 4
- Gray level 2 in original is replaced by 5
- Gray level 3 in original is replaced by 6
- Gray level 4 in original is replaced by 6
- Gray level 5 in original is replaced by 7
- Gray level 6 in original is replaced by 7
- Gray level 7 in original is replaced by 7

$x_i = i$	$y_j = H_x$	$y'_j = H_z$	$z_j = j$
0	0.19	0.0	3
1	0.44	0.0	4
2	0.65	0.0	5
3	0.81	0.15	6
4	0.89	0.35	6
5	0.95	0.65	7
6	0.98	0.85	7
7	1.0	1.0	7

$P_{new}(0) = 0$
$P_{new}(1) = 0$
$P_{new}(2) = 0$
$P_{new}(3) = P_{old}(0) = 0.19$
$P_{new}(4) = P_{old}(1) = 0.25$
$P_{new}(5) = P_{old}(2) = 0.21$
$P_{new}(6) = P_{old}(3) + P_{old}(4) = 0.24$
$P_{new}(7) = P_{old}(5) + P_{old}(6) + P_{old}(7) = 0.11$

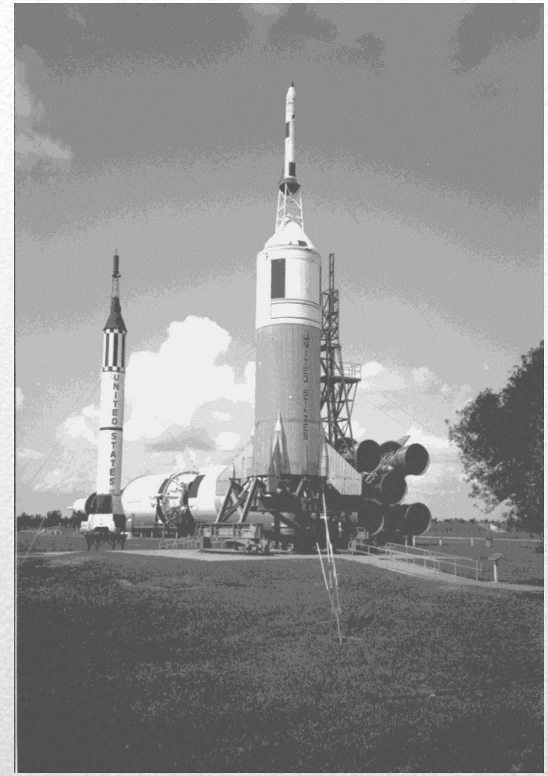


Example: Matching histograms



original

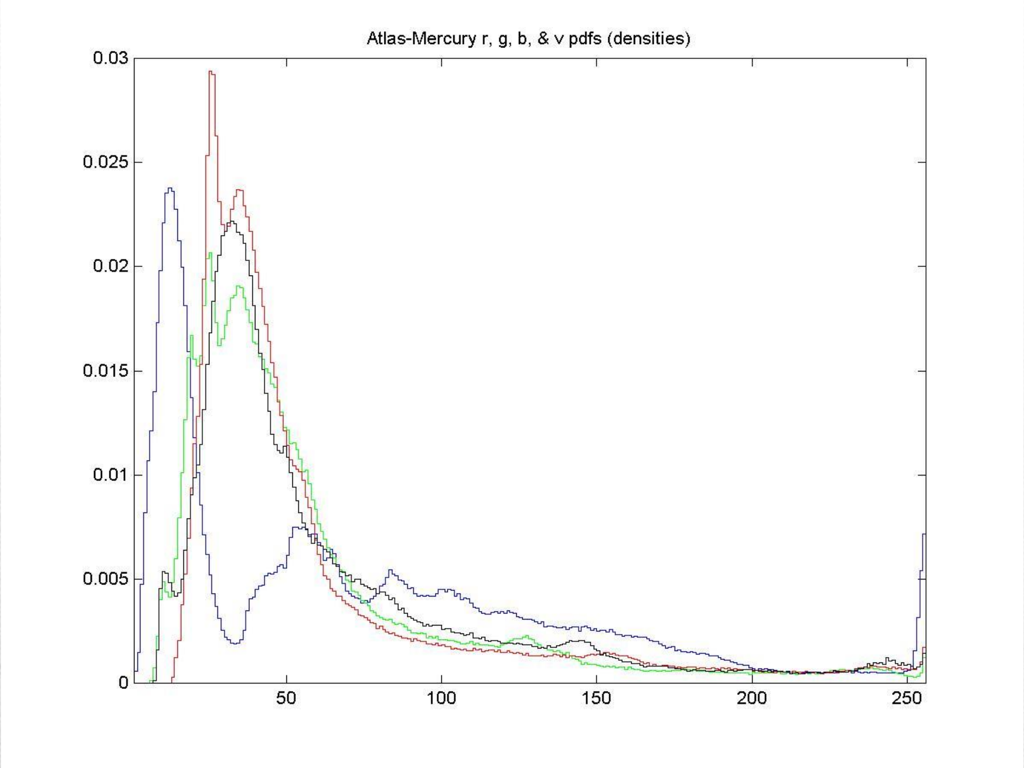
target



remapped

Example: Histogram Matching

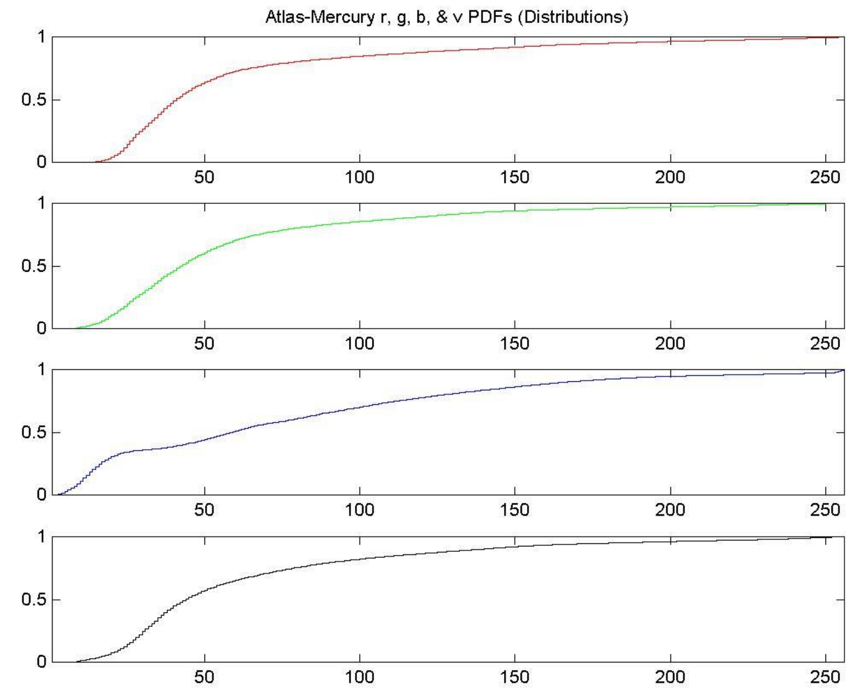
➢ Probability Density Functions of a Color Image



red pdf
green pdf
blue pdf
luminosity pdf

Coloured Images: PDF

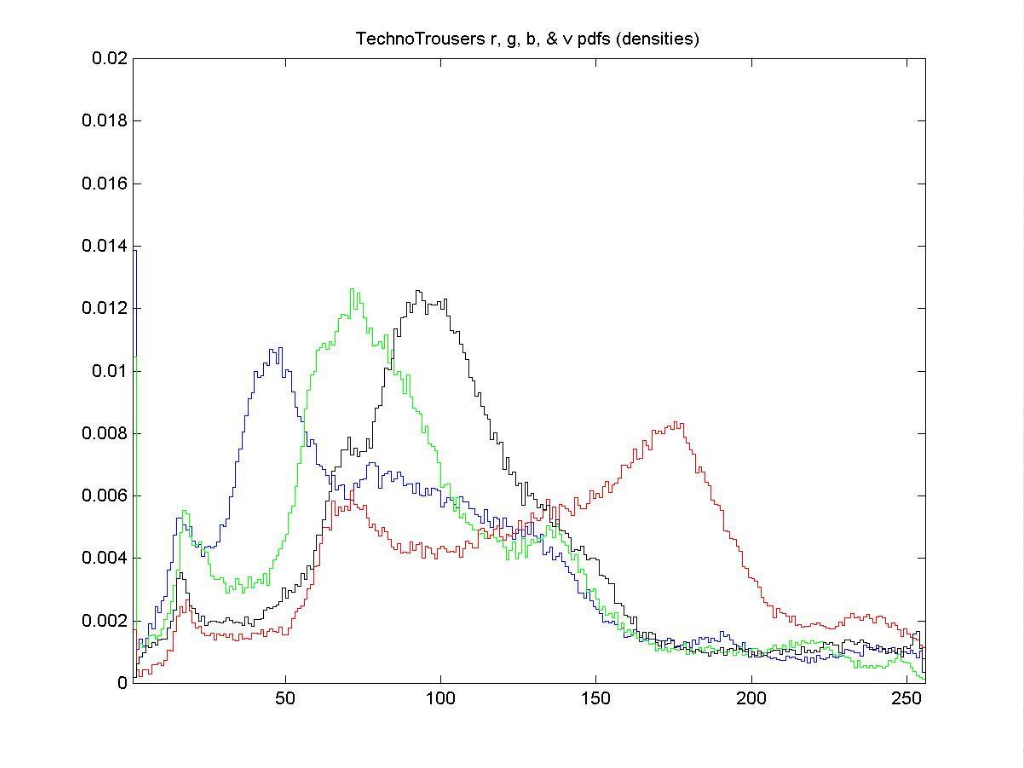
- Cumulative Distribution Functions (CDF)



red CDF
green CDF
blue CDF
luminosity CDF

Coloured Images: CDF

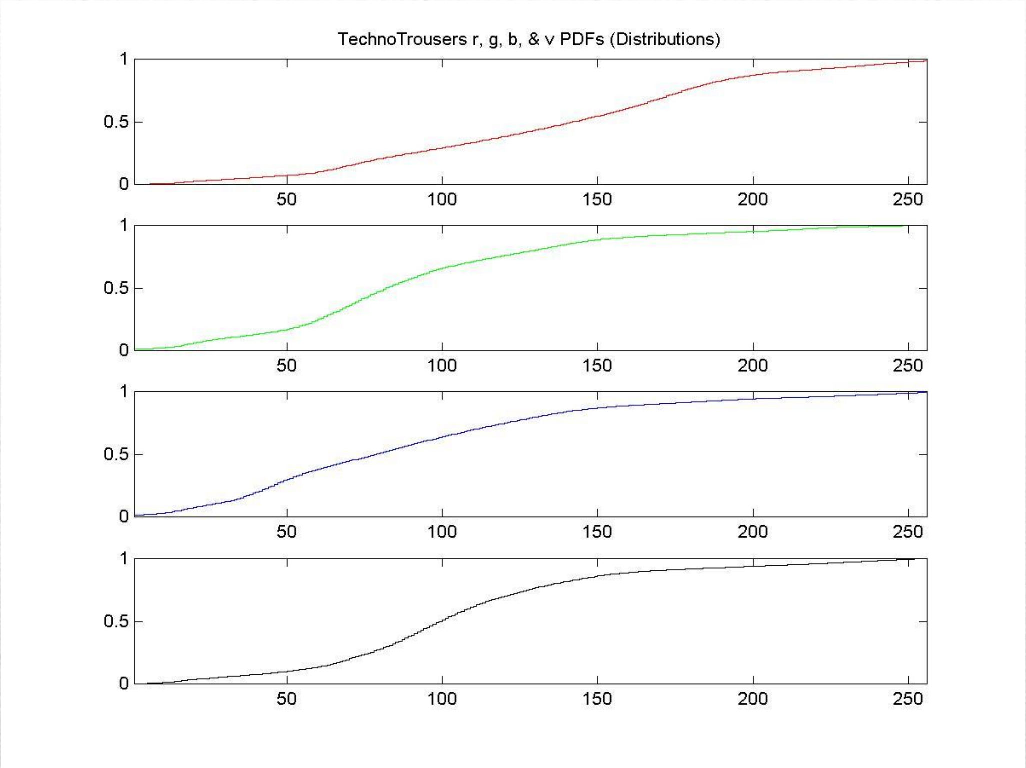
➢ Probability Density Functions of a Color Image



red pdf
green pdf
blue pdf
luminosity pdf

Coloured Images: PDF

➢ Cumulative Distribution Functions (CDF)



red CDF ——— blue CDF———
green CDF ——— luminosity CDF———

Coloured Images: CDF


original

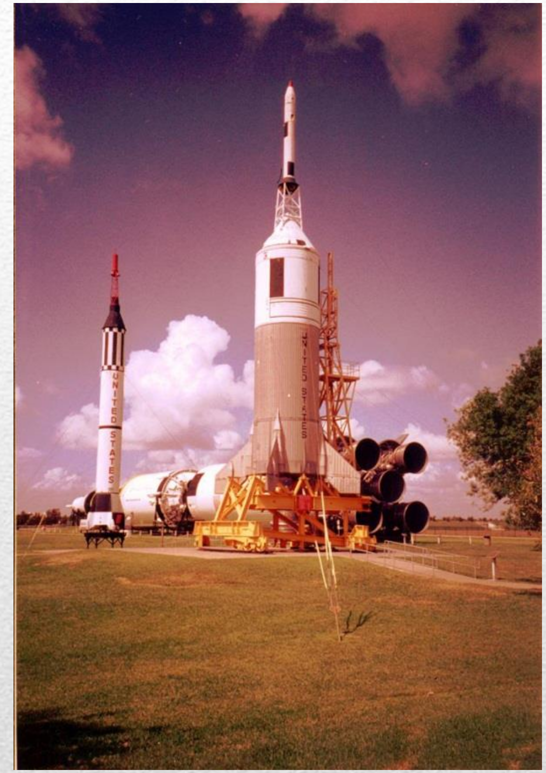

target

luminosity remapped

Remap an Image to have the Lum. CDF of Another

original

target



R, G, & B remapped

Remap an Image to have the rgb CDF of Another

- We can transform the values of an image as we want.
- There are some commonly used transformations:
  - Gamma mapping → increase/decrease the contrast in dark/light gray levels
  - Logarithmic mapping → low intensity pixels are enhanced (enhancing details in dark images)
  - Exponential mapping → enhance details in bright areas while decreasing details in dark areas

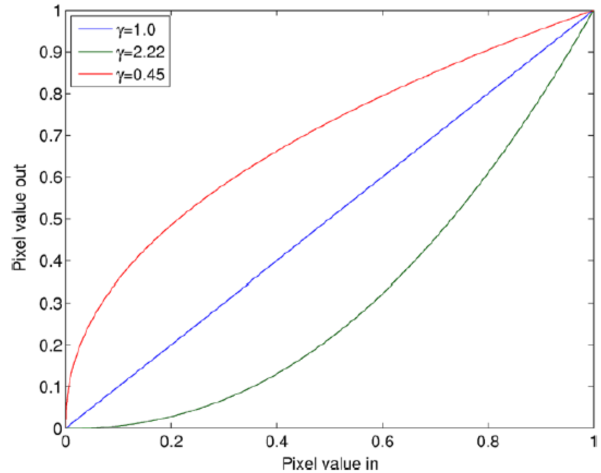
Non-linear gray-level mapping

Non-linear gray-level mapping

$$g(x, y) = f(x, y)^{\gamma}$$

Gamma value: 0.45 | No gamma correction | Gamma value: 2.22

- Gamma mapping:

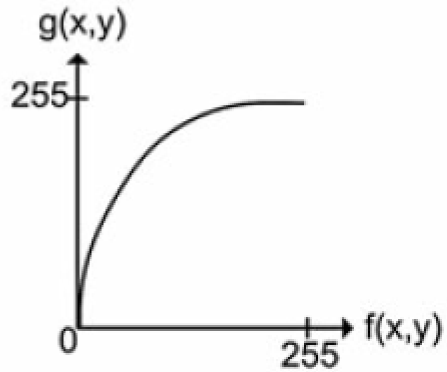


- Logarithmic mapping:

$$g(x, y) = c \cdot \log\big(1 + f(x, y)\big)$$

$$c = \frac{255}{\log(1 + v_{\max})}$$

where $v_{\max}$ is the maximum pixel value in the input image



Logarithmic greyscale mapping

g(x,y)

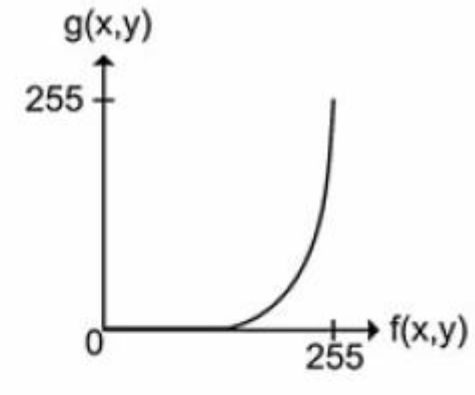
Non-linear gray-level mapping

- Exponential mapping: $g(x, y) = c \cdot \left(k^{f(x,y)} - 1\right)$

k is a parameter that can be used to change of shape of the transformation curve
c is a scaling constant that ensures that the maximum output value is 255

$$c = \frac{255}{k^{v_{\max}} - 1}$$

$v_{\max}$ is the maximum pixel value in the input image



Exponential greyscale mapping


g(x,y)

Non-linear gray-level mapping

IMAGE ARITHMETIC

Image arithmetic

- We can perform arithmetic operations between images
  - Add two images
  - Subtract an image from another one
  - Perform logic operations between binary images
  - Average, max, min, multiplication, …

- Image addition and subtraction
  - $g(x, y) = f_1(x, y) \pm f_2(x, y)$
  - if g(x,y) > 255 or < 0 → overflow and underflow
- When overflow or underflow:
  - g(x,y) = 255 (0 resp.) if g(x,y) > 255 (g(x,y) < 0 resp.)
  - Or leave the value as they are and then map them to the original interval as follow:

Image arithmetic

The algorithm is as follows:

1. Find the minimum number in the temporary image, f_1
2. Find the maximum number in the temporary image, f_2
3. Shift all pixels so that the minimum value is 0: $g_i(x, y) = g_i(x, y) - f_1$
4. Scale all pixels so that the maximum value is 255: $g(x, y) = g_i(x, y) \cdot \frac{255}{f_2 - f_1}$

where $g_i(x, y)$ is the temporary image.

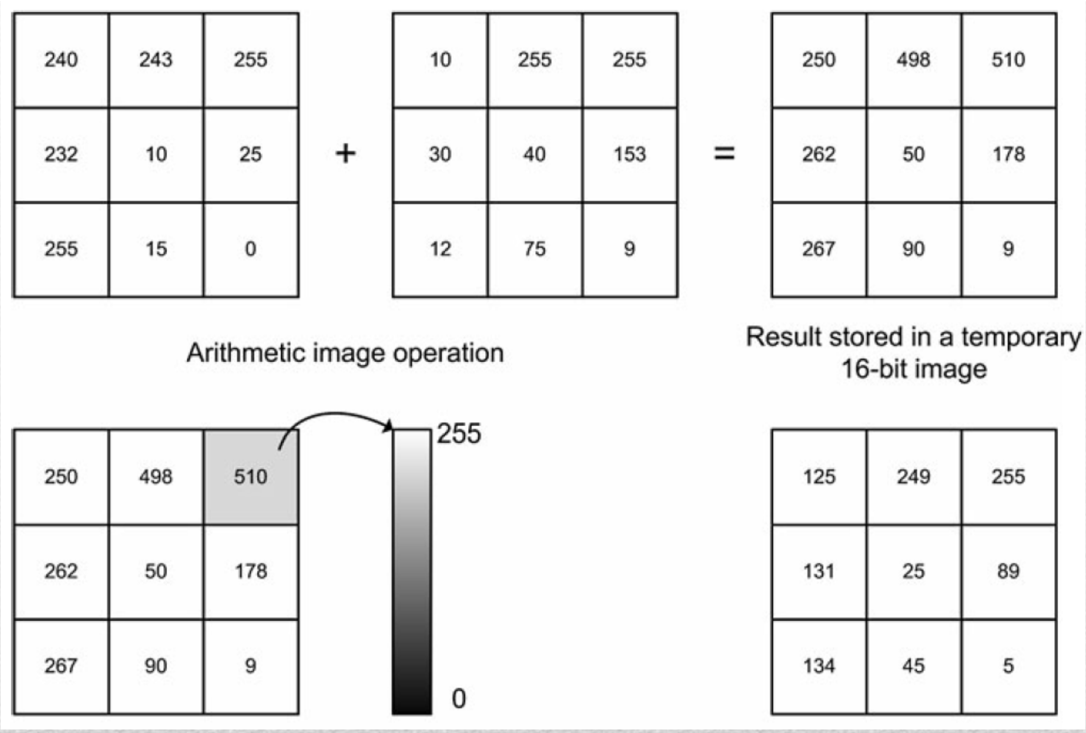


Image arithmetic

- One of the applications of image arithmetic
  - Image negative: $g(x,y) = 255 - f(x,y)$
  - Alpha blending → mixing two images especially for transitions from one image to another
- Alpha blending: Each image has an importance

$$g(x,y) = \alpha . f_1(x,y) + (1-\alpha) . f_2(x,y)$$

Image arithmetic

$$g(x, y) = \alpha . f_1(x, y) + (1 - \alpha) . f_2(x, y)$$

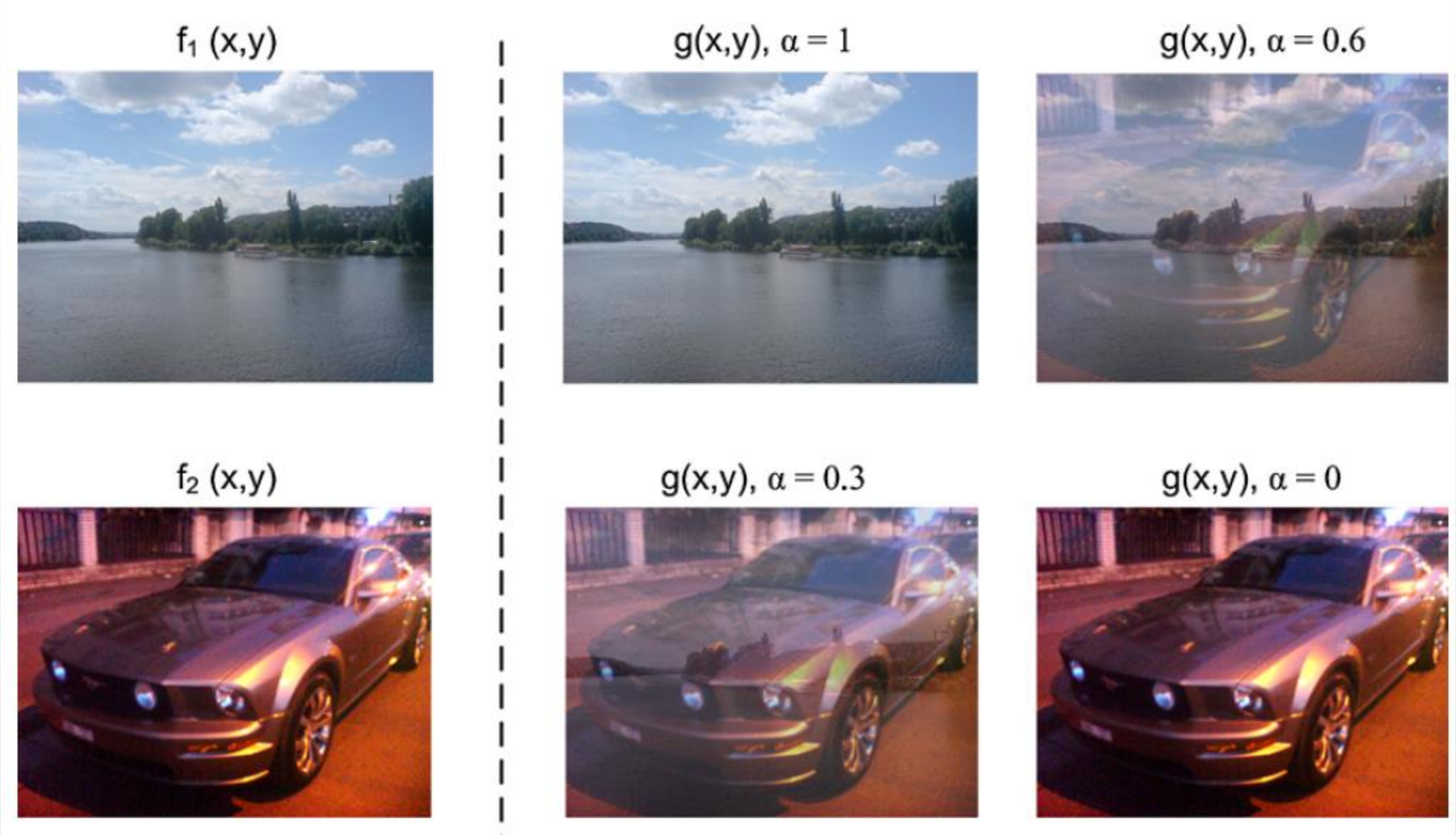


Image arithmetic

$\alpha = 0.25$

$\alpha = 0.50$

$\alpha = 0.75$

Image arithmetic

Logic operations

Truth table for AND

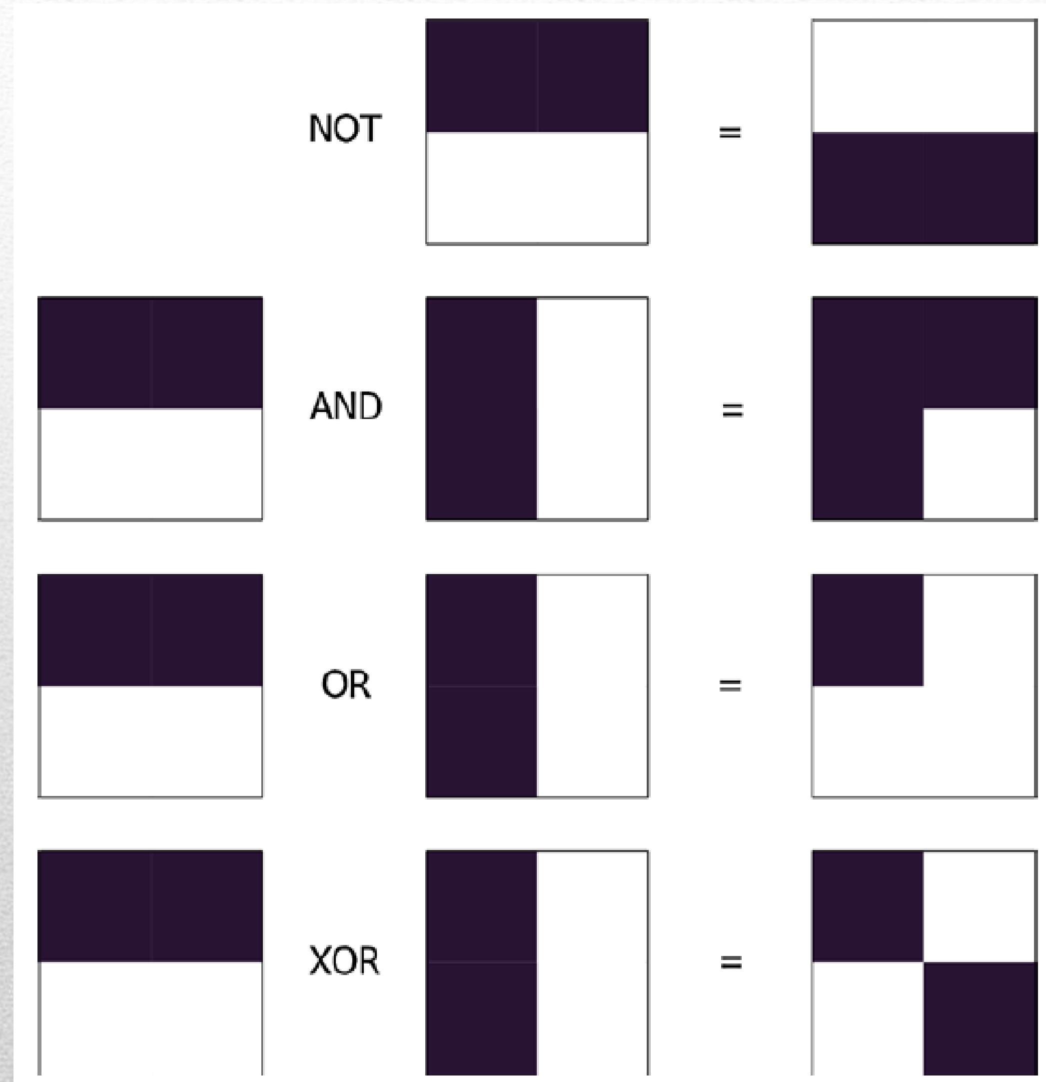
AND		Input 2	
		0	255
Input 1	0	0	0
	255	0	255

Truth table for OR

OR		Input 2	
		0	255
Input 1	0	0	255
	255	255	255

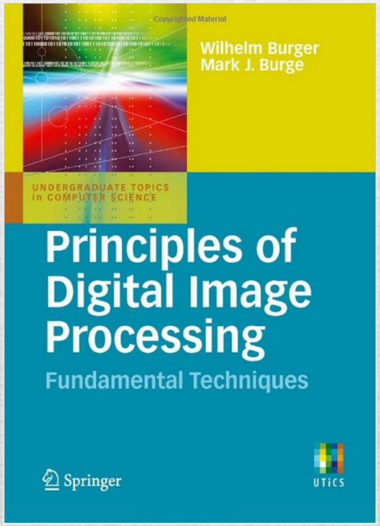
Truth table for XOR

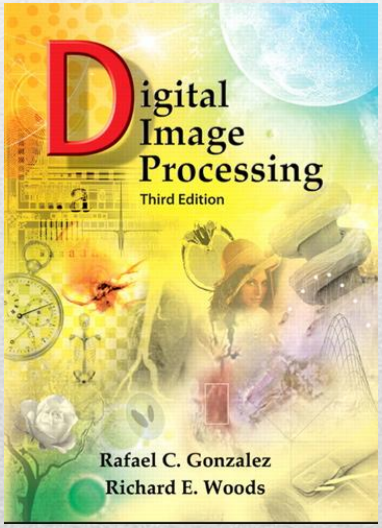
XOR		Input 2	
		0	255
Input 1	0	0	255
	255	255	0



This is the end of the lecture on **<u>Point Processing</u>** techniques !

Do you have any question ?





The end !

End of Lecture !