Simple features of digital image

...

Histogram modelling

...

Segemntation

Paweł Forczmański
Chair of Multimedia Systems, Faculty of Computer Science and Information Technology

Agenda

1. luminance and contrast
2. intensity histogram
3. histogram modifications
4. coocurrence matrix

DIP scheme

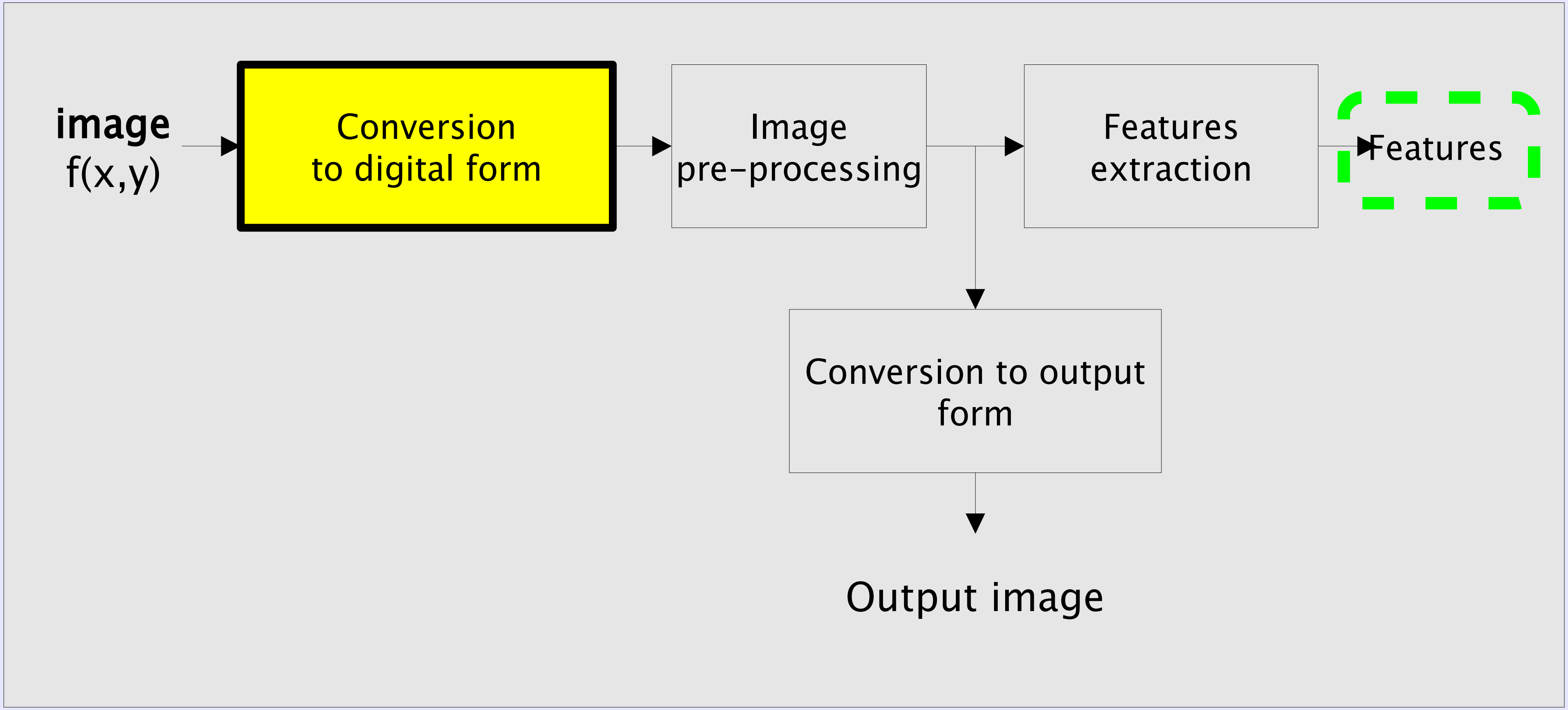


image
f(x,y)
Conversion
to digital form
Image
pre-processing
Features
extraction
Features
Conversion to output
form
Output image

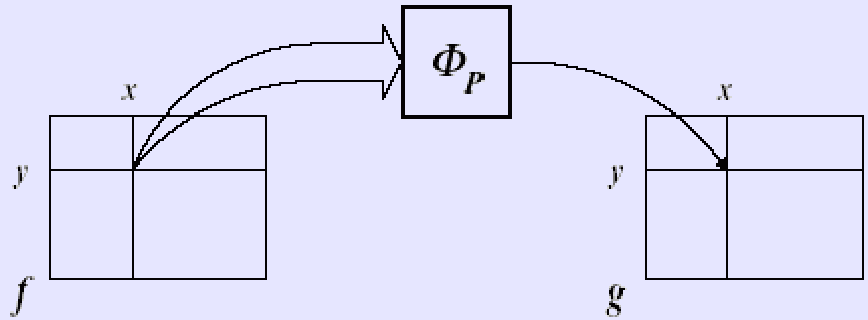
The influence of luminance and contrast on image's characteristics

$$J = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} f(i,j)$$

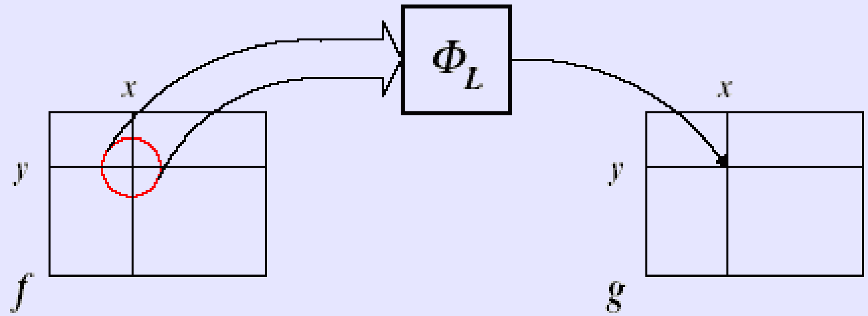
$$C = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[f(i,j) - J\right]^2}$$

Transformations

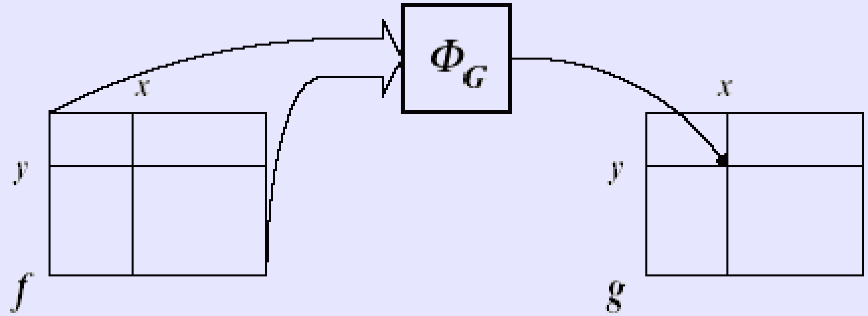
point transform



local transform



global transform



DIP scheme

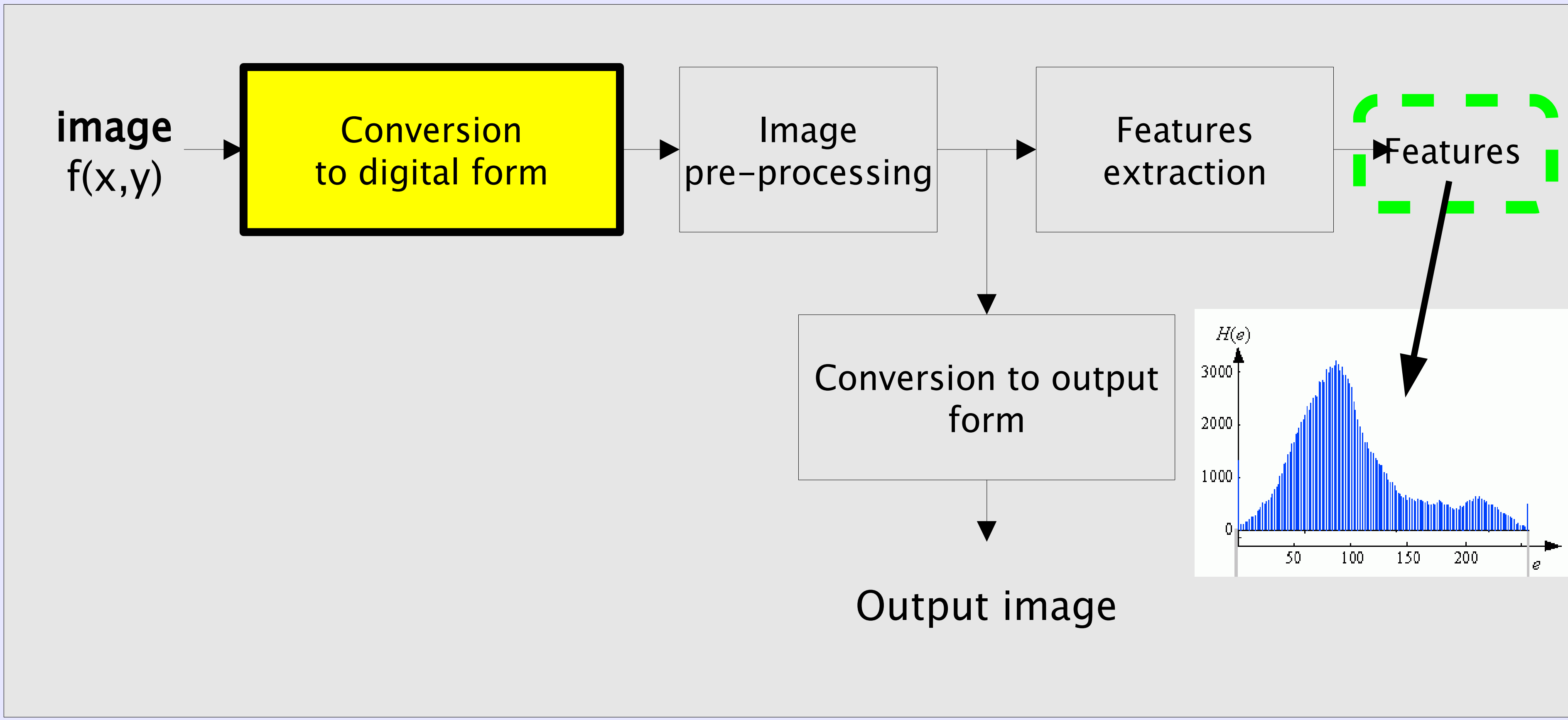
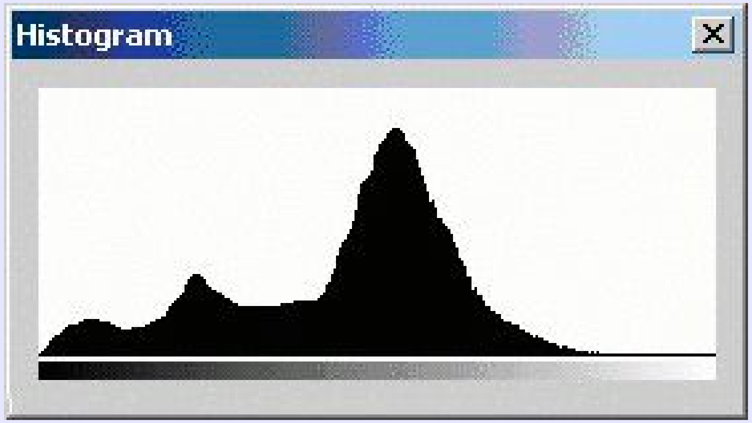
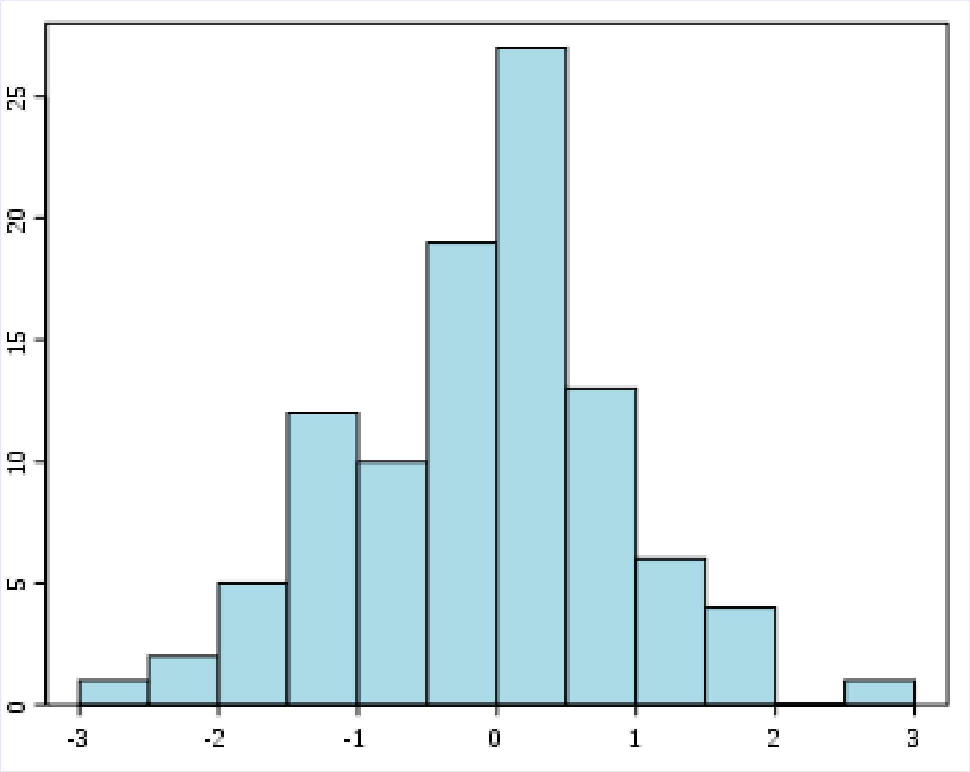


image
f(x,y)
Conversion
to digital form
Image
pre-processing
Features
extraction
Features
Conversion to output
form
Output image
H(e)
3000
2000
1000
0
50
100
150
200
e

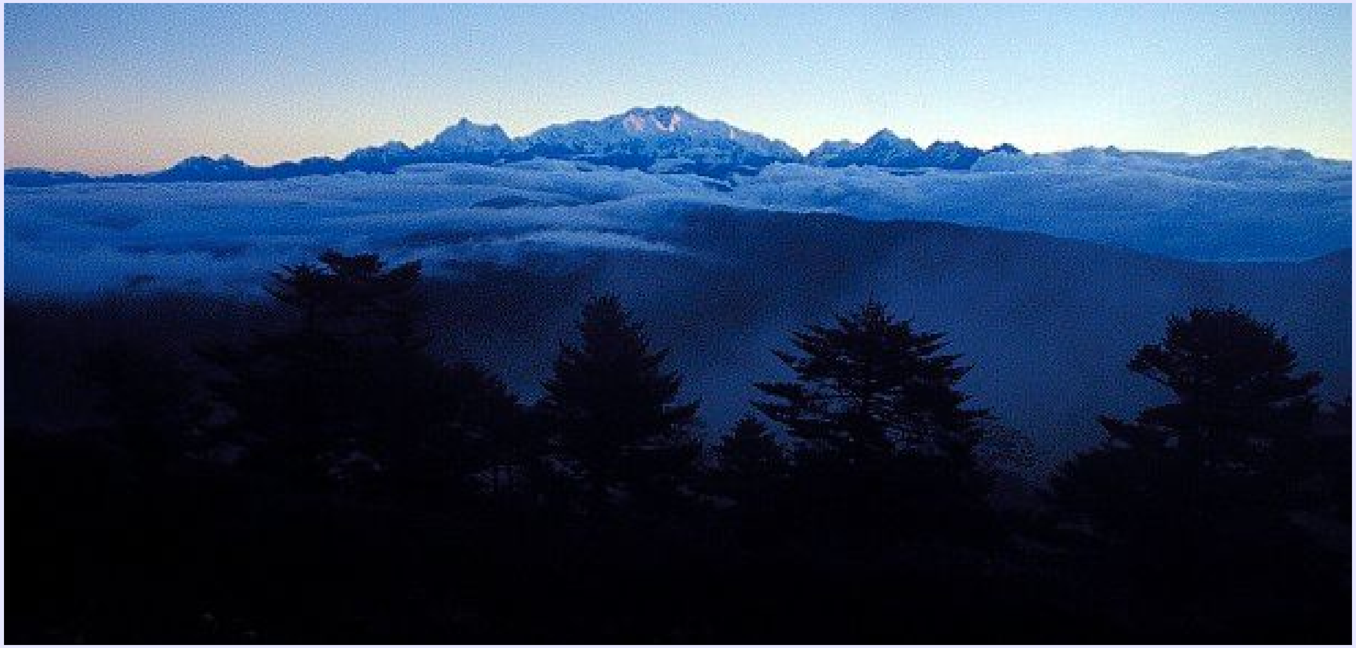
Histogram : 1D

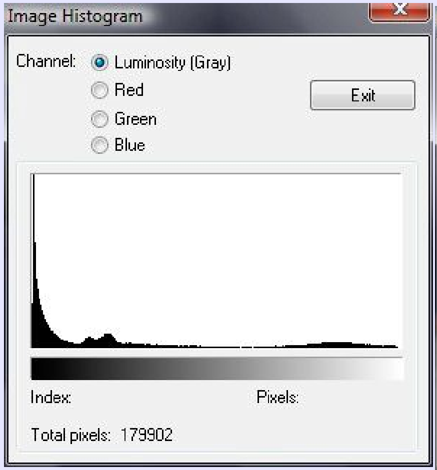


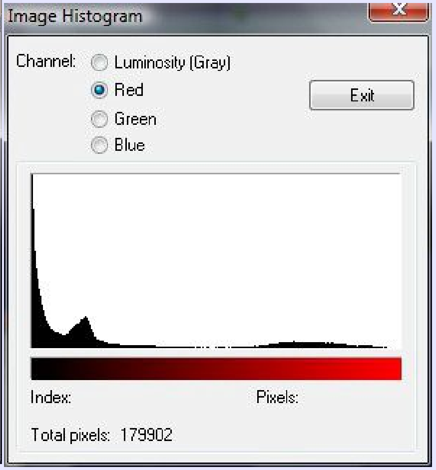


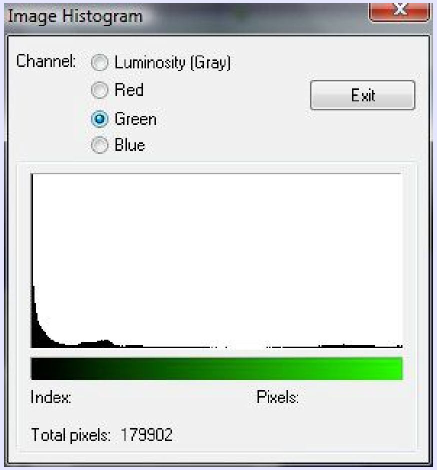
- The most simple case - *one-dimensional histogram*
- Representing the frequency of elements in some set;
  - in case of an image – the number of pixels with adequate in-tensity level

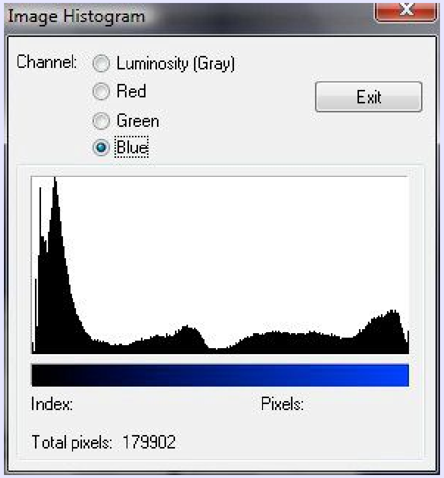
RGB Histogram



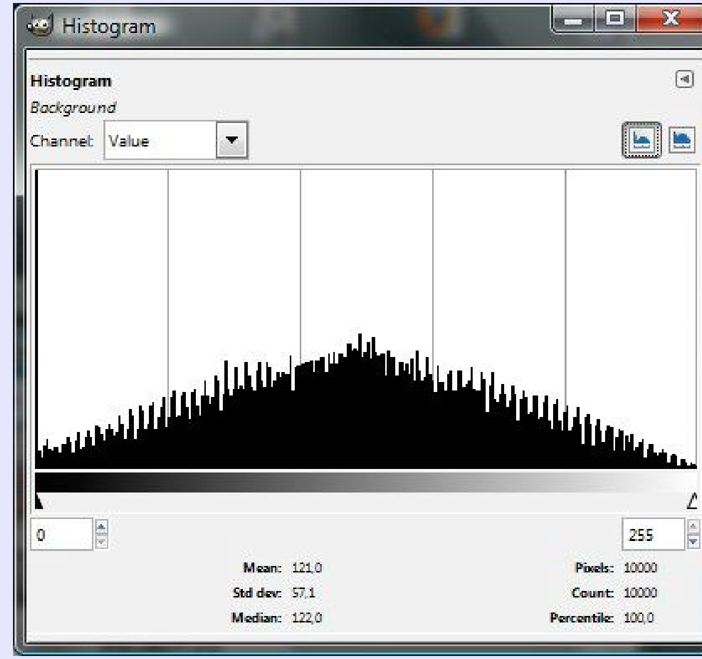


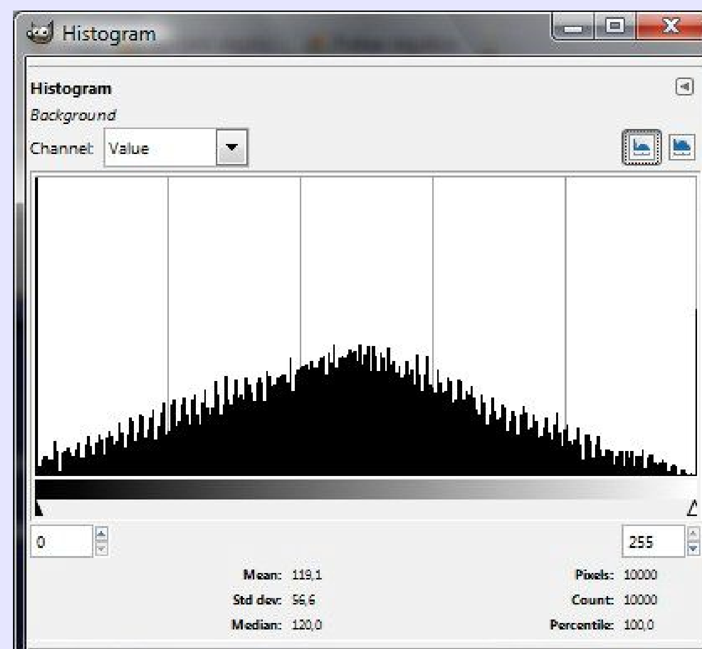




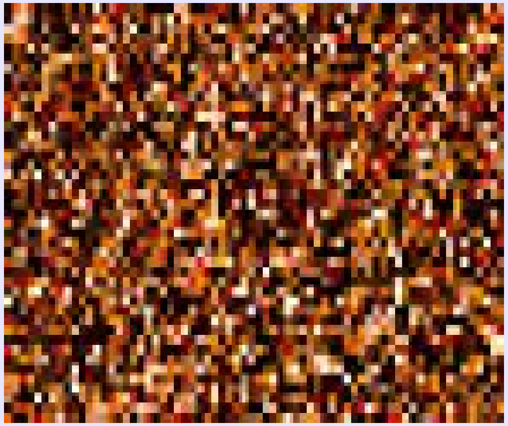


Histogram in not unique!





Examples of equal hitograms



Definition

From the purely statistical point of view, it is an estimate of the probability distribution of a continuous variable (first introduced by Karl Pearson)

The area under a histogram H(b). (cumulative histogram):

$$\int_0^b H(b)\,db$$

where:
H(b) – histogram function,
db - step,
H(b)db – height of each hitogram interval,

For digital images (db is limited), the histogram is calculated as a sum of all pixels with a certain value, often normalized to 1 (by dividing by the sum of all pixels)

The algorithm

The algorithm of calculating simple intensity histogram for 2D image is as follows:

➔Set (or assume) the range of intensity for input image (i.e. 0-1, 0-255, 1-256, etc.);

➔Assume the number of intervals;

➔Calculate the width of an interval (by dividing range by the number of intervals);

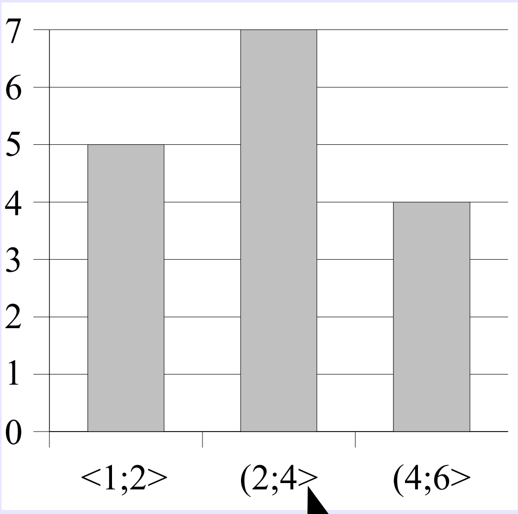
➔Count the number of pixels ineach interval.

The idea of calculations

image

1	4	1	3
5	6	2	6
4	4	1	3
4	1	5	4

histogram



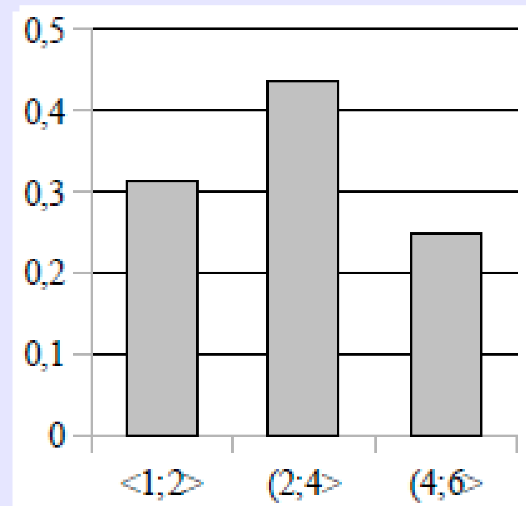
intervals								number of pixels
<1;2>	1	1	2	1	1			5
(2;4>	4	3	4	4	3	4	4	7
(4;6>	5	6	6	5				4

Normalization

image

1	4	1	3
5	6	2	6
4	4	1	3
4	1	5	4

histogram

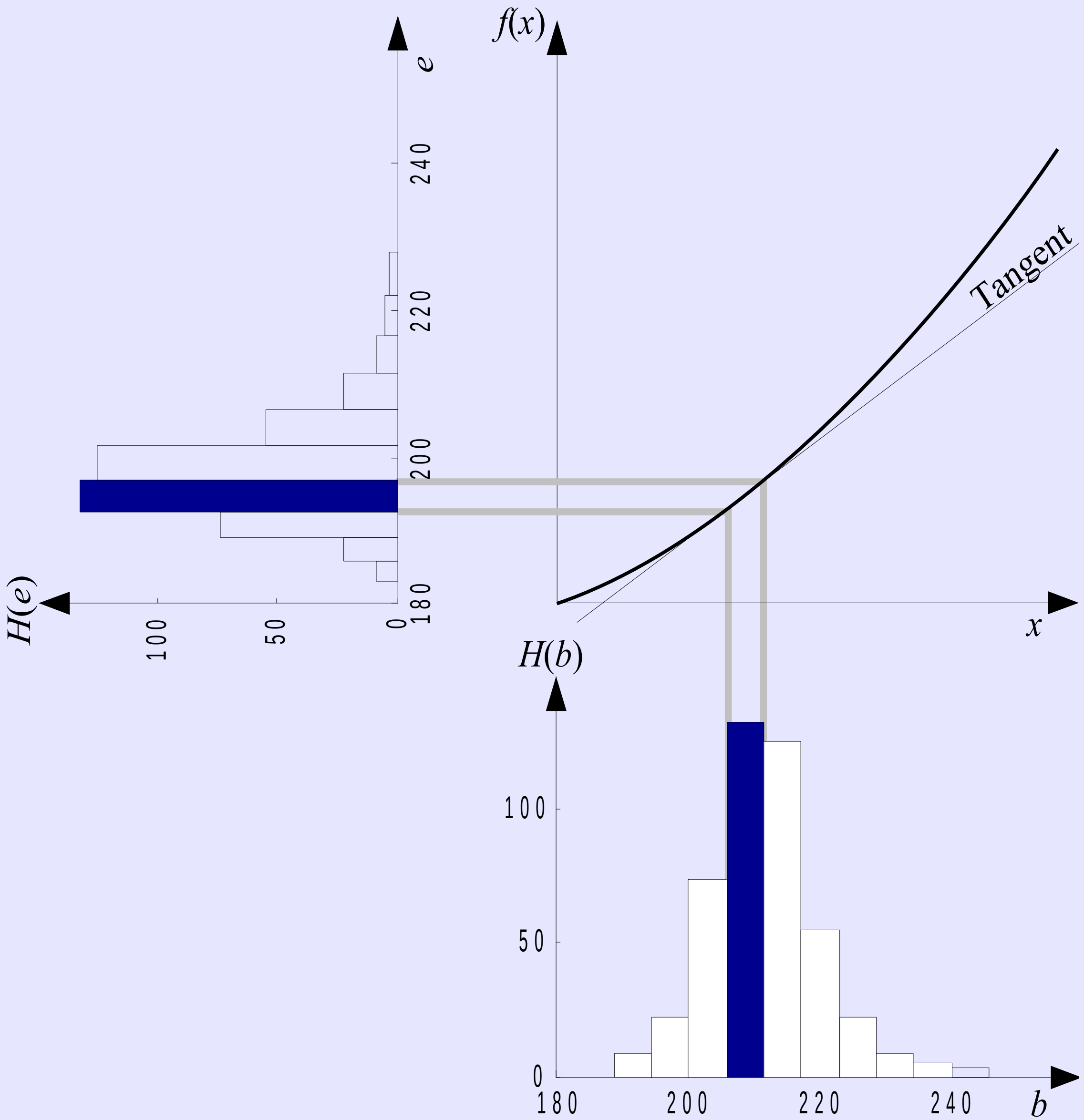


<1;2>	1	1	2	1	1			5 / 16
(2;4>	4	3	4	4	3	4	4	7 / 16
(4;6>	5	6	6	5				4 / 16

intervals

normalization

Histogram modelling



Hostogram stretching along a defined line changes the **distribution of intensities** in an image by the alternation of **intensity assignment** in each interval

Each interval changes its width:

$$d\,e = f\,'(b)\,d\,b$$

where

b –pixel intensity before:

e –pixel intensity after stretching;

f(b) –stretching function.

The tangent of an angle of function f(b) is the coeficient that changes the width of each histogram interval

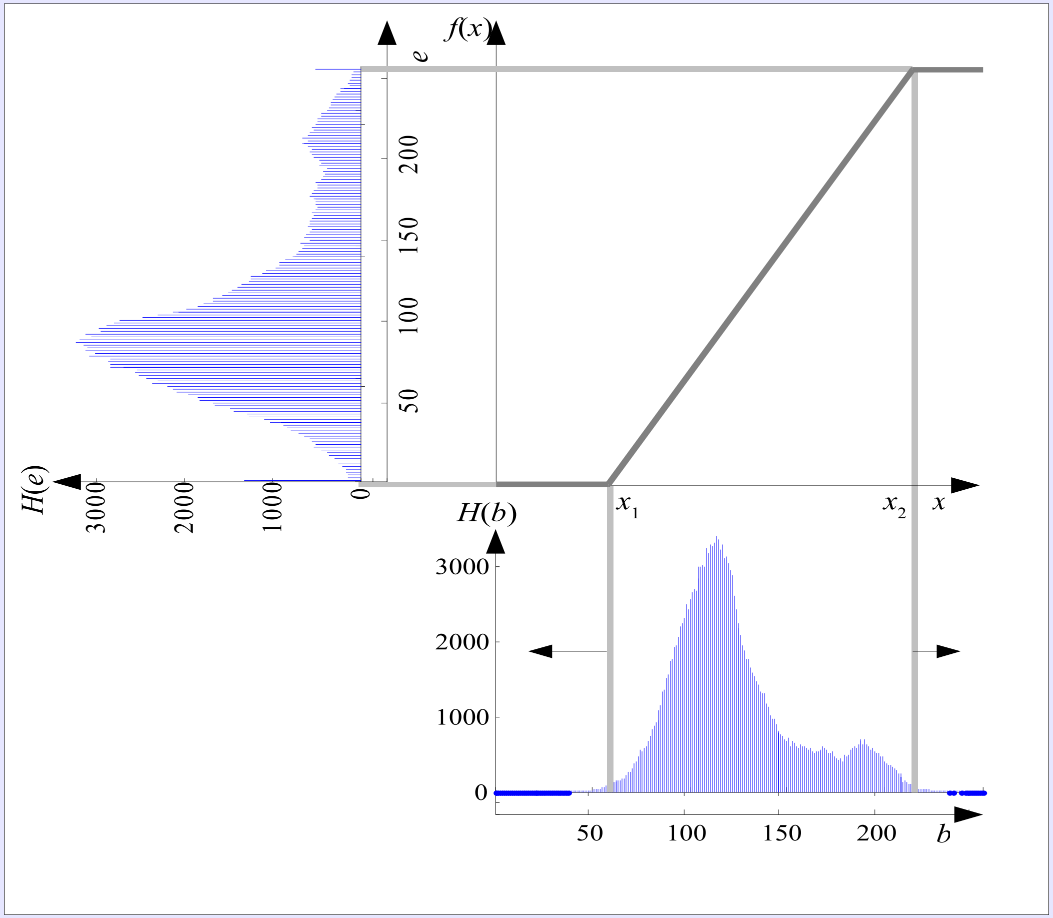
Simple linear case

The most simple is a linear stretching:

$$f(x)=\begin{cases}0 \ for \ x<0\\ ax\\ E \ for \ x>E\end{cases}$$

Where *a* can is equal to:

$$a=\frac{E}{x_2-x_1}$$

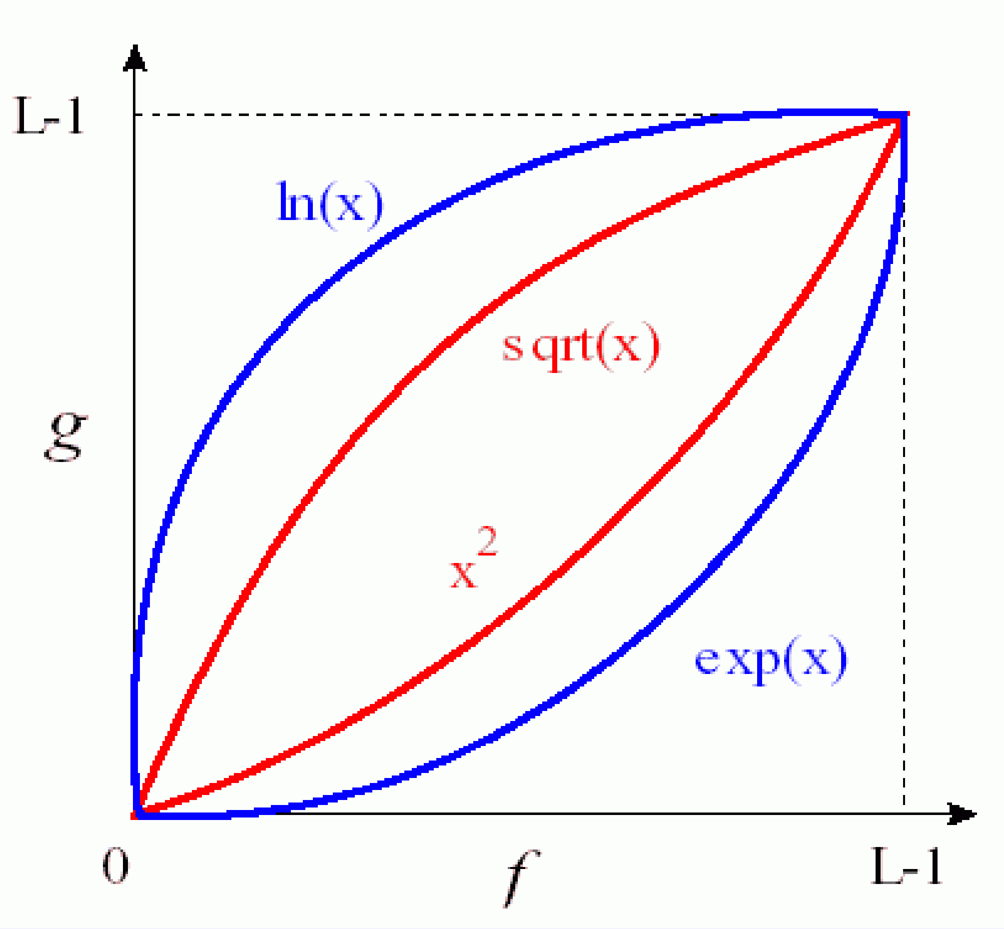


where

x_1, x_2 – boundaries of intensity.

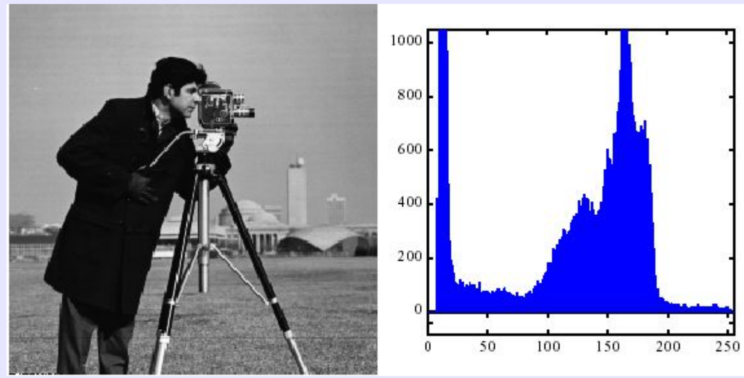
E – maximum possible intensity

Non-linear cases

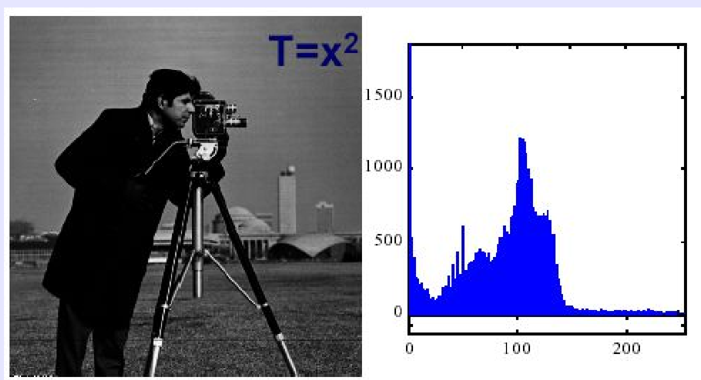


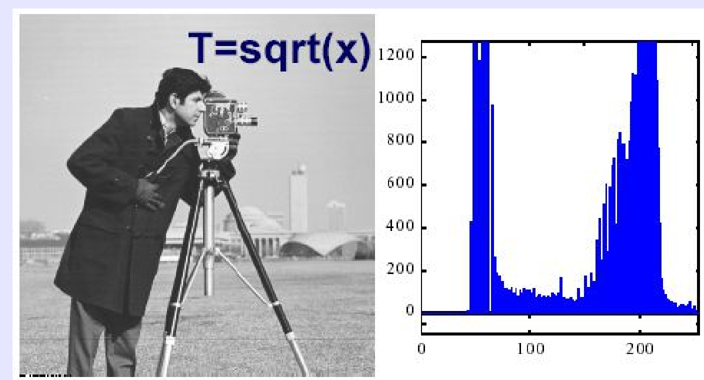
Non-linear cases (examples)

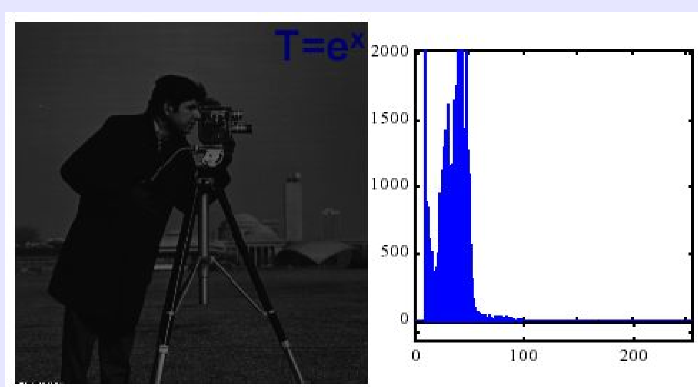
source →

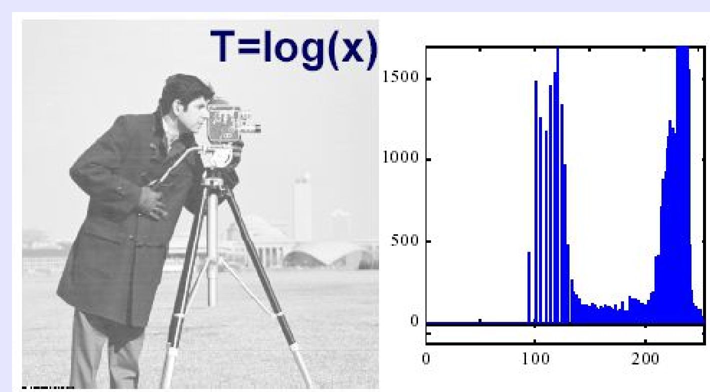


←histogram







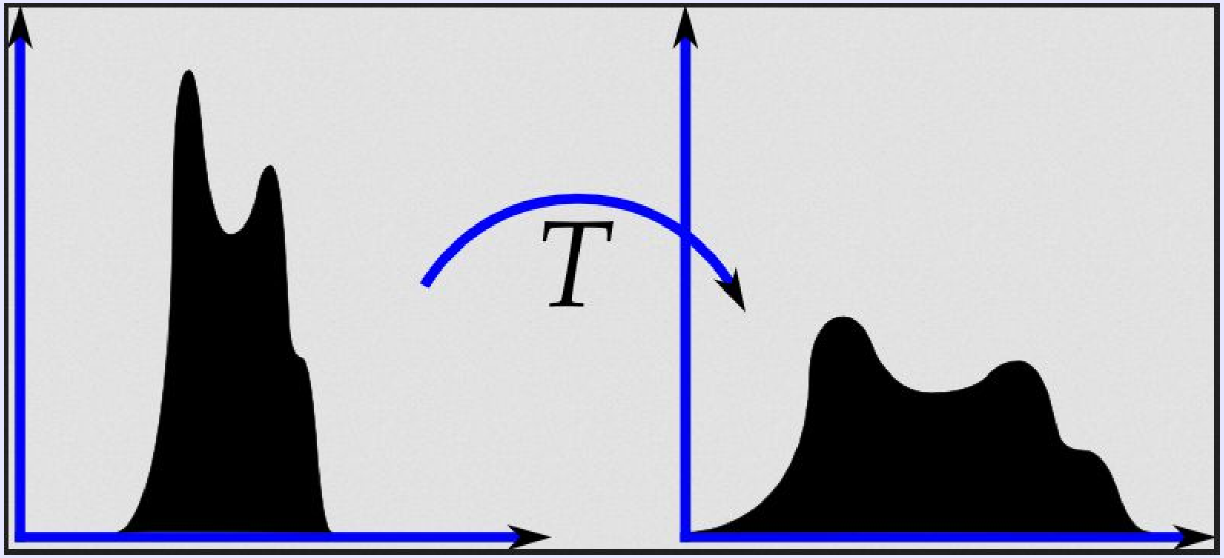


Histogram equalization

It usually increases the global contrast of images, especially when the usable data of the image is represented by close contrast values.

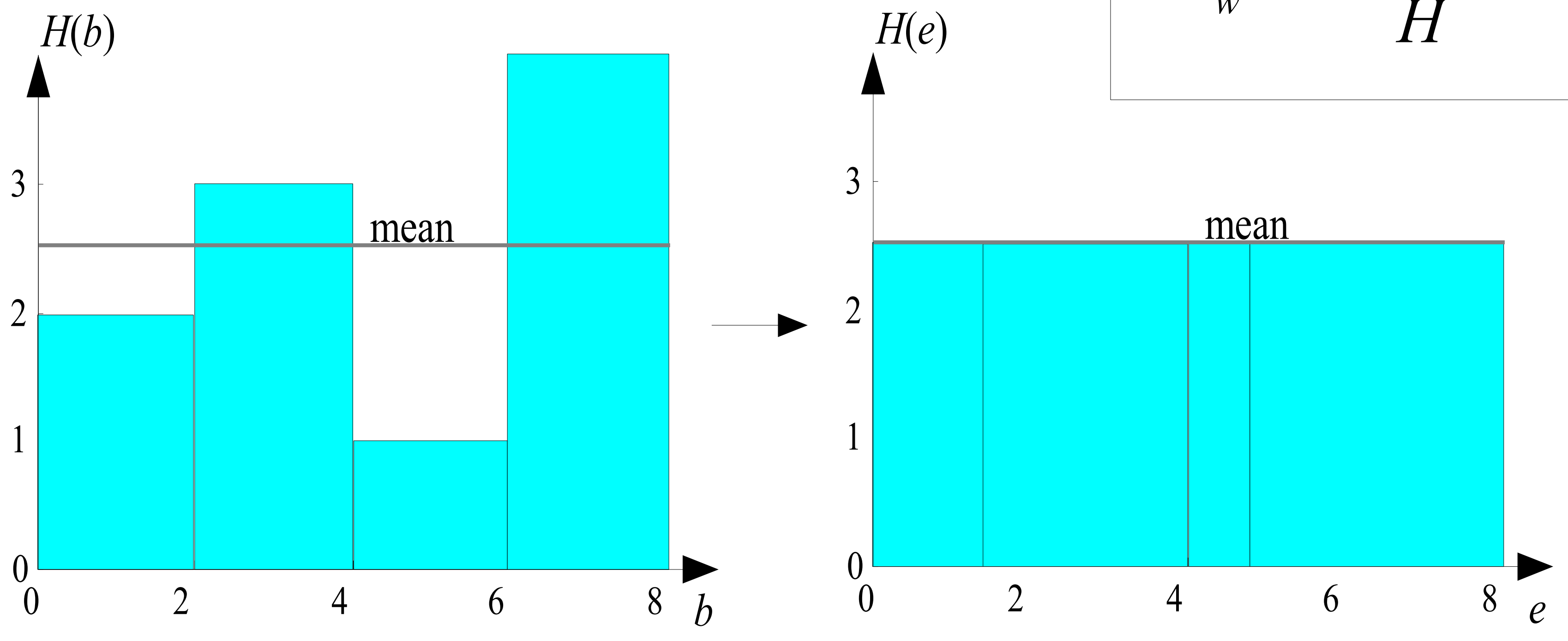
Through this adjustment, the intensities can be better distributed on the histogram.

Areas of lower local contrast gain a higher contrast.



New pixel values can be calculated as follows:

$$o_w = \frac{oH\left[b(o)\right]}{\overline{H}}$$

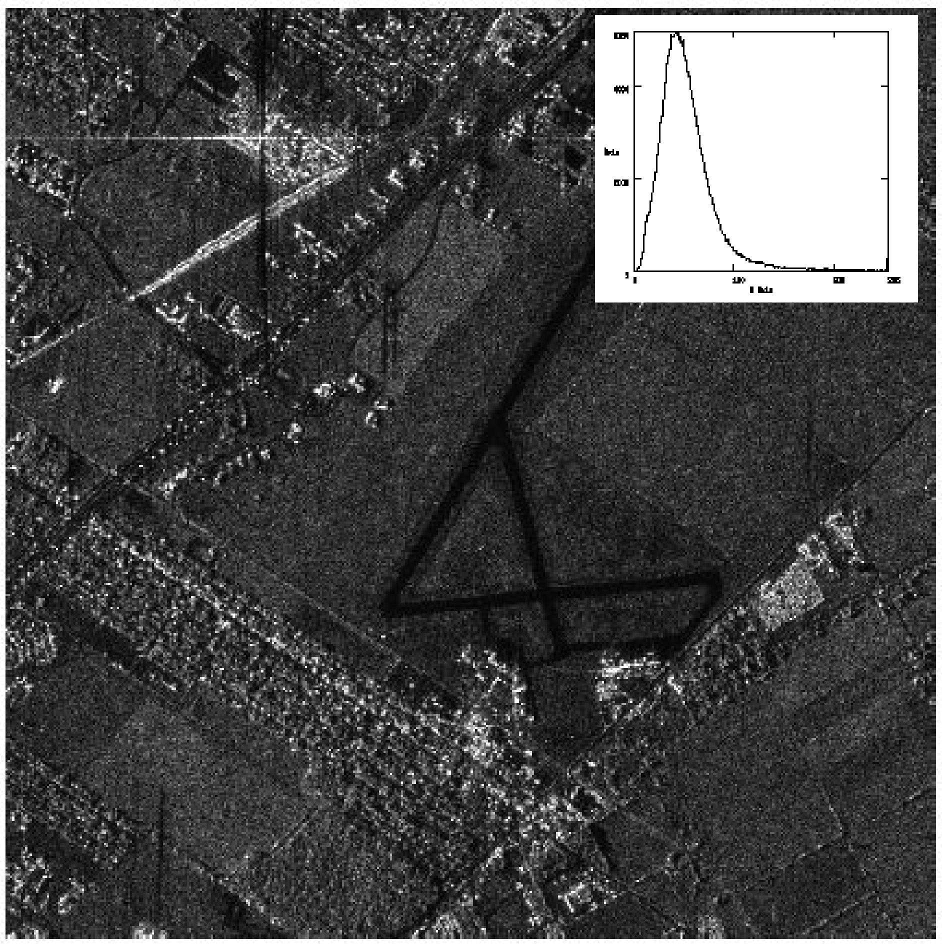


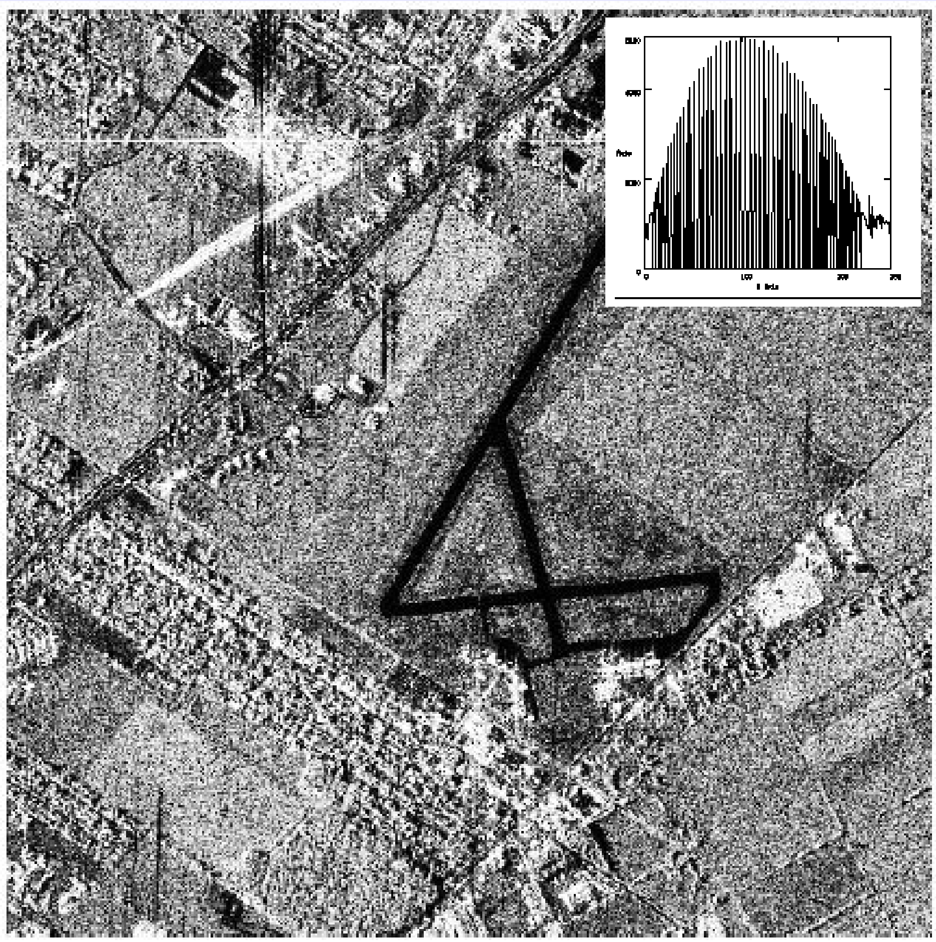
Hist. equalization : example

Let us assume an image wits 64 x 64 pixels with one of eight intensities: *K*=***8***, *n=64*64=* ***4096***

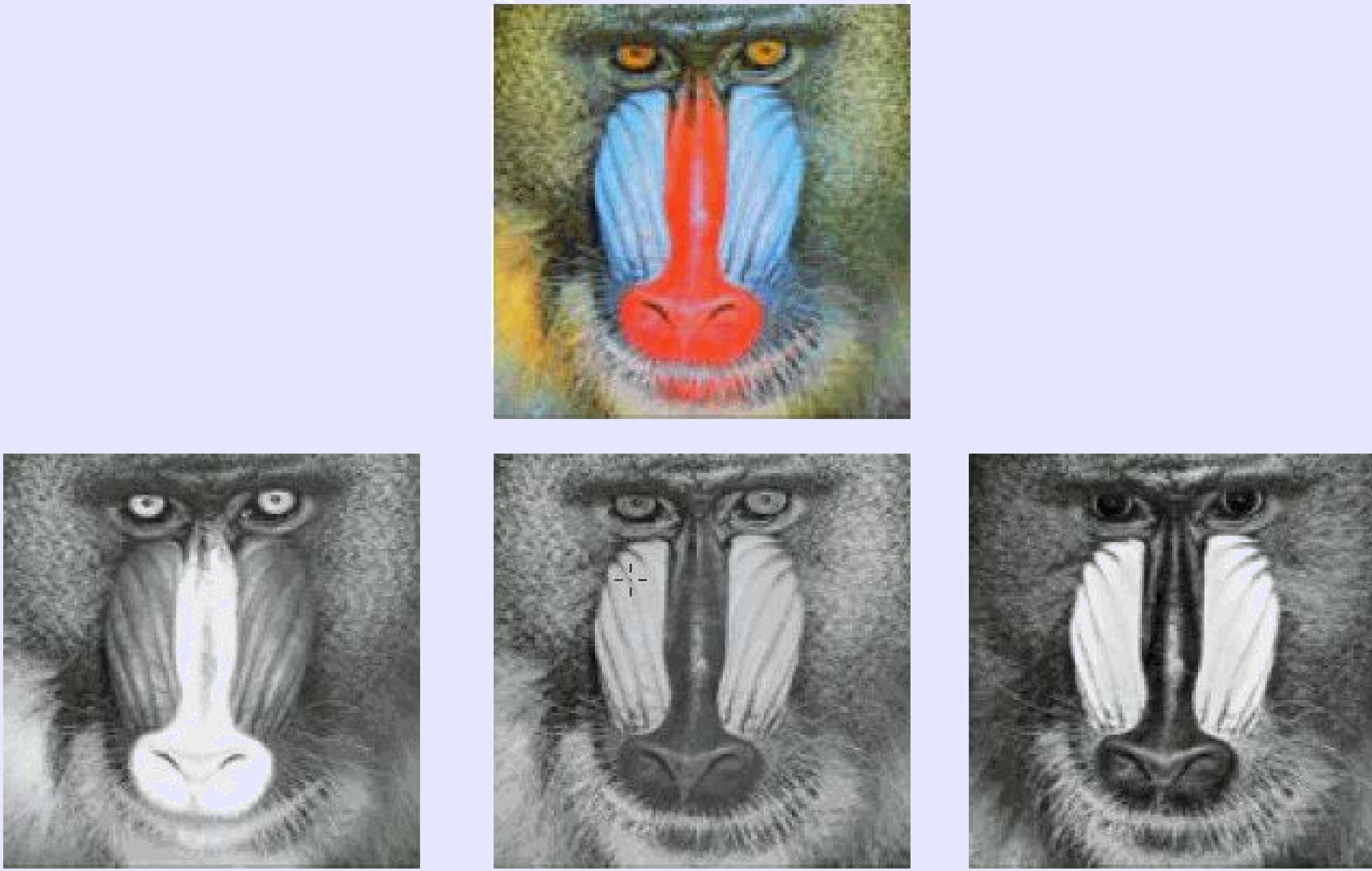
q_k	n_k	$H(q_k)=\frac{n_k}{n}$	$w_k=\sum_{j=0}^{k}H(q_k)$	$v_k=w_k$
0	790	0.19	0.19	$\frac{1}{7}$
$\frac{1}{7}$	1023	0.25	0.44	$\frac{3}{7}$
$\frac{2}{7}$	850	0.21	0.65	$\frac{5}{7}$
$\frac{3}{7}$	656	0.16	0.81	$\frac{6}{7}$
$\frac{4}{7}$	329	0.08	0.89	$\frac{6}{7}$
$\frac{5}{7}$	245	0.06	0.95	1
$\frac{6}{7}$	122	0.03	0.98	1
1	81	0.02	1.00	1

Graphical example



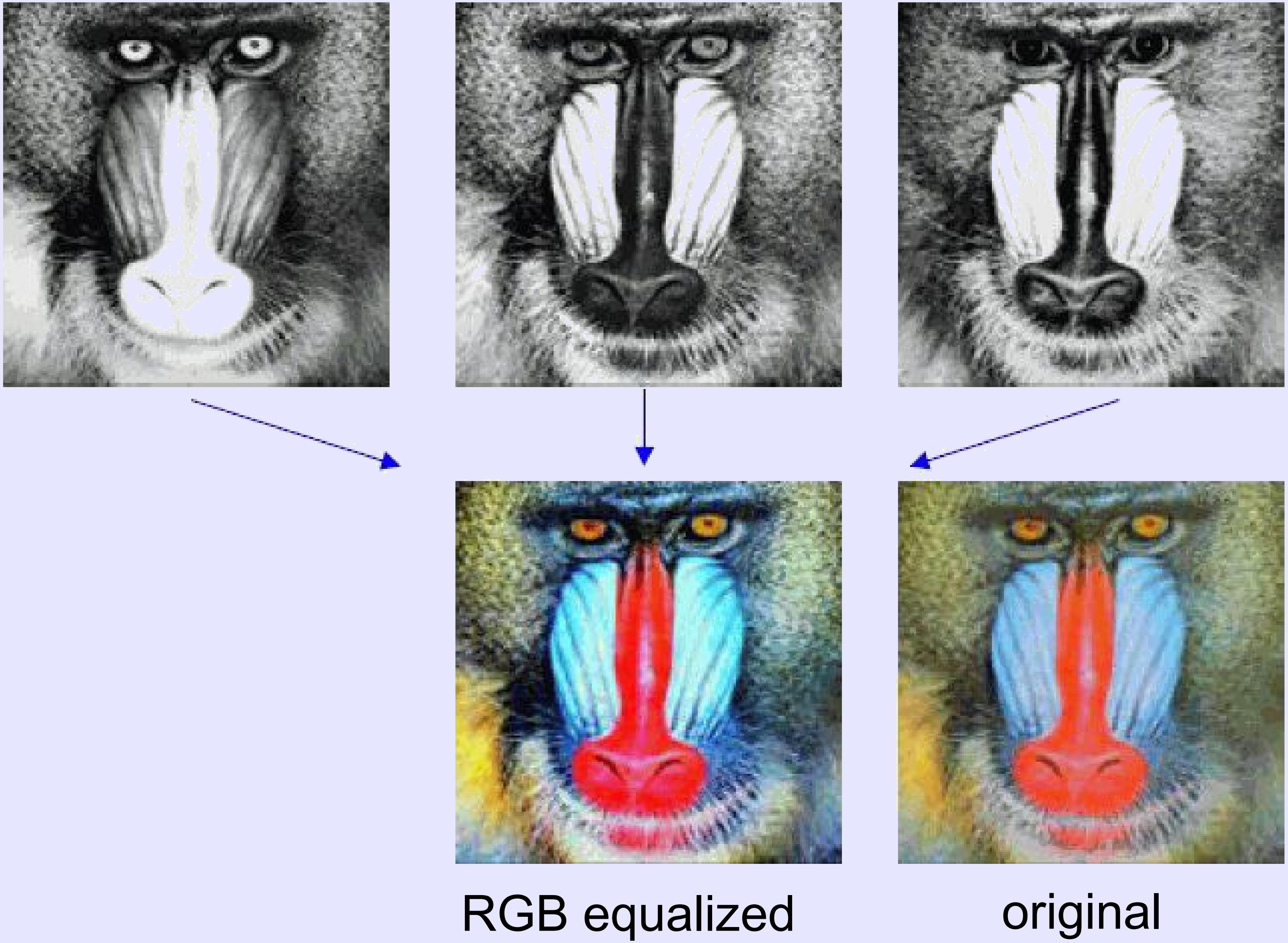


Work in RGB space

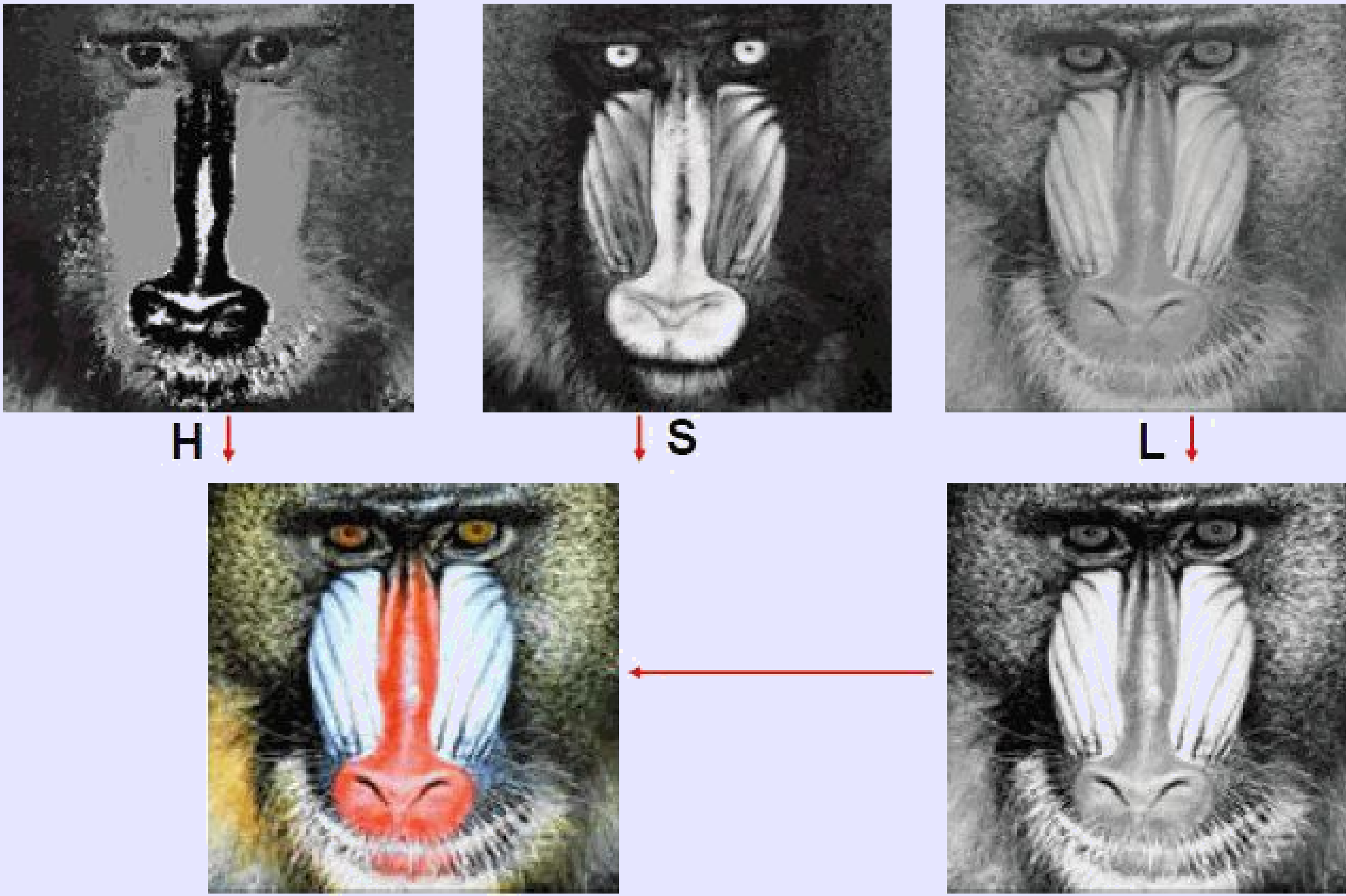


R G B

Work in RGB space



Work in HSL space



HSL equalized

RGB and HSL comparison

original

RGB equalized

HSL equalized

Another example (RGB vs HSL)

oroginal

RGB

HSL

1D vs 2D histogram

One-dimensional histogram if defined by function *f* :

$$f \ : \ X \times Y \rightarrow Z$$
$$f^{-1} \ : \ Z \rightarrow \mathbf{2}^{X \times Y}$$
$$f^{-1} \ : \ \{(x, y) | f(x, y) = z\}$$

1D vs 2D histogram

Two-dimensional histogram if defined by functions *f* and *g* :

$$
\begin{aligned}
f &: X \times Y \to Z \\
g &: X \times Y \to V \\
f^{-1} &: Z \to \mathbf{2}^{X \times Y} \\
g^{-1} &: V \to \mathbf{2}^{X \times Y} \\
f^{-1} &: \{(x, y) | f(x, y) = z\} \\
g^{-1} &: \{(x, y) | g(x, y) = v\}
\end{aligned}
$$

Co-occurrence matrix (1)

There are many 2D histograms! One of the most useful is coocurrence *matrix*

$$M_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}; \quad M_2 = \begin{bmatrix} 1 & 3 & 2 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix};$$

$$z = [0\ 1\ 2\ 3\ ]; \qquad z = [0\ 1\ 2\ 3\ ];$$

$$H_1(z) = [7\ 5\ 3\ 1]; \qquad H_2(z) = [7\ 5\ 3\ 1];$$

Co-occurrence matrix (2)

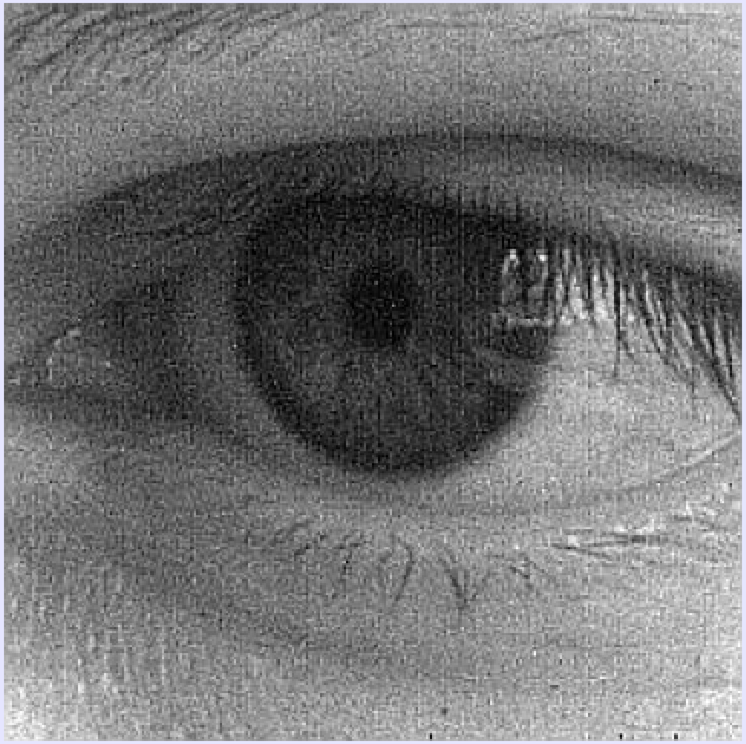
Co- occurrence matrix relies on the condition (relation) between functions *f* and *g* (pixels), i.e.:

$$r = \{((x,y),(x,y+1))\};$$
$$C_r = H_{fg}(z,v);$$
$$f(x,y) = g(x,y+1);$$

$$C_{r_1} = \begin{bmatrix} 3 & 3 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \qquad C_{r_2} = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix};$$

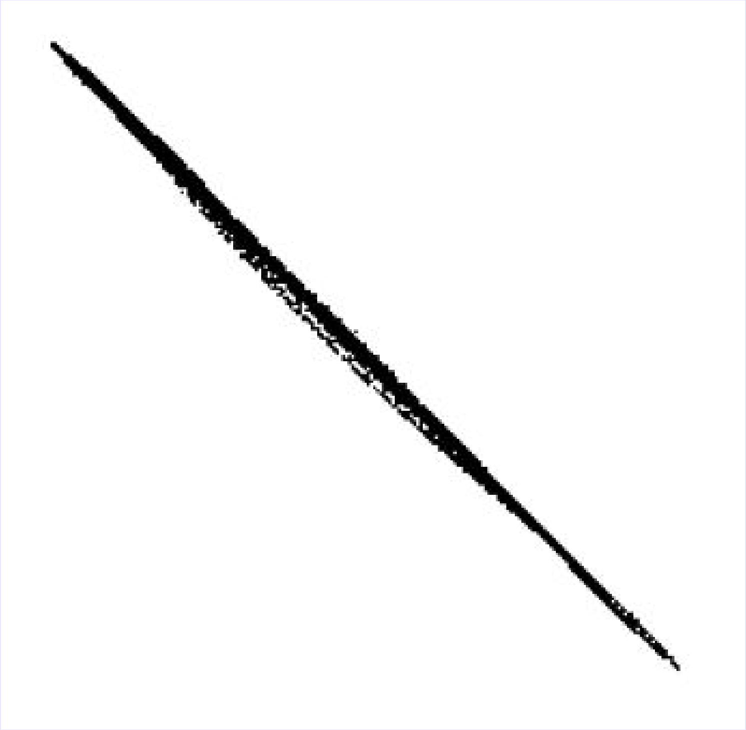
Example

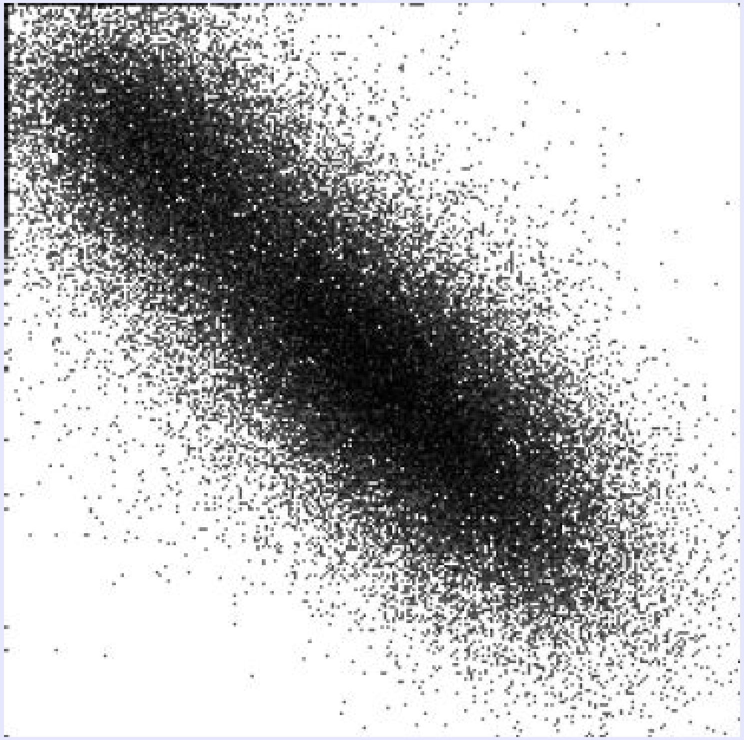
Example of calculation on real image – it helps when we want to tell if the image is crisp or blurred



Example

Example of calculation on real image – it helps when we want to tell if the image is crisp or blurred





Segmentation

Image segmentation is the process of partitioning a digital image into **multiple segments** (sets of pixels, also known as superpixels).

The goal of segmentation is to **simplify** and/or change the representation of an image into something that is more meaningful and easier to analyze.

It is typically used to locate objects and boundaries (lines, curves, etc.) in images.

It is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

Segmentation algorithms

Point-based methods

- thresholding,
- clusterization (grouping)

Edge-based methods

- based on edge detection

Area-based methods

- region growing,
- region merging,
- region splitting,
- split & merge,
- watershed segmentation,

Hybrid methods

- using more than single approach, i.e. region growing with egde detection

Thresholding

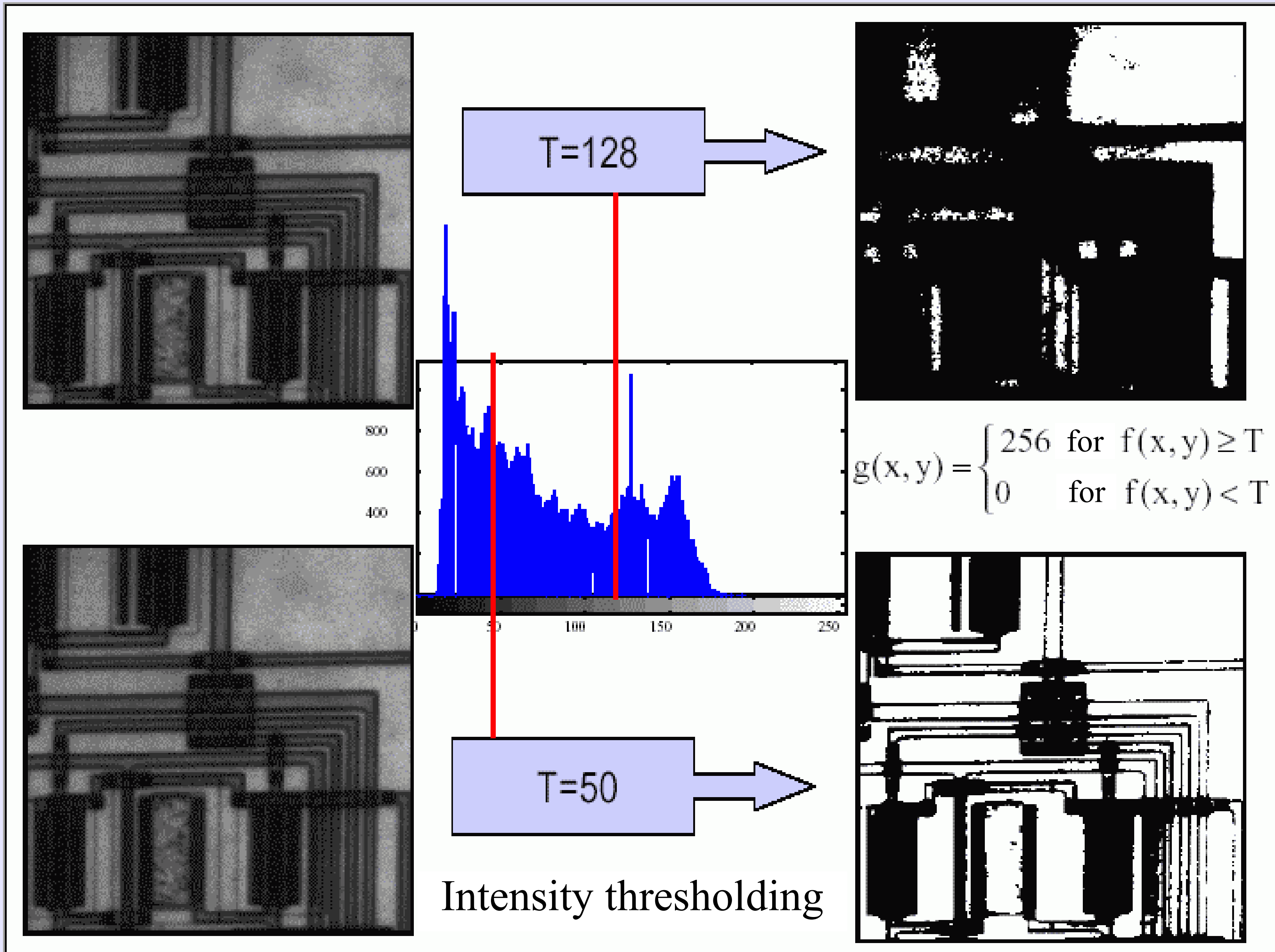
Thresholding is the simplest method of image segmentation.

From a grayscale image, thresholding can be used to create binary images

T is the threshold.

$$g(x,y)=\begin{cases}1 & \text{for} \quad f(x,y)\geq T\\ 0 & \text{for} \quad f(x,y)<T\end{cases}$$

example


T=128
T=50
800
600
400
100
150
200
250
$g(x,y)=\begin{cases}256 & \text{for } f(x,y)\geq T\\ 0 & \text{for } f(x,y)<T\end{cases}$
Intensity thresholding

Multi-thresholding

We define *m* thresholds, i.e. *m=4*, {T1,T2,T3,T4}

After that we get 5 level image:

$$g(x,y)=\begin{cases} 0 & \text{for } f(x,y)\leqslant T_1 \\ 1 & \text{for } T_1< f(x,y)\leqslant T_2 \\ 2 & \text{for } T_2< f(x,y)\leqslant T_3 \\ 3 & \text{for } T_3< f(x,y)\leqslant T_4 \\ 4 & \text{for } f(x,y)> T_4 \end{cases}$$

Semi-thresholding

It changes only a limited group of pixels:

$$g(x,y)=\begin{cases}0 & \text{for } f(x,y)\leqslant T\\ f(x,y) & \text{for } f(x,y)>T\end{cases}$$