# Image enhancement

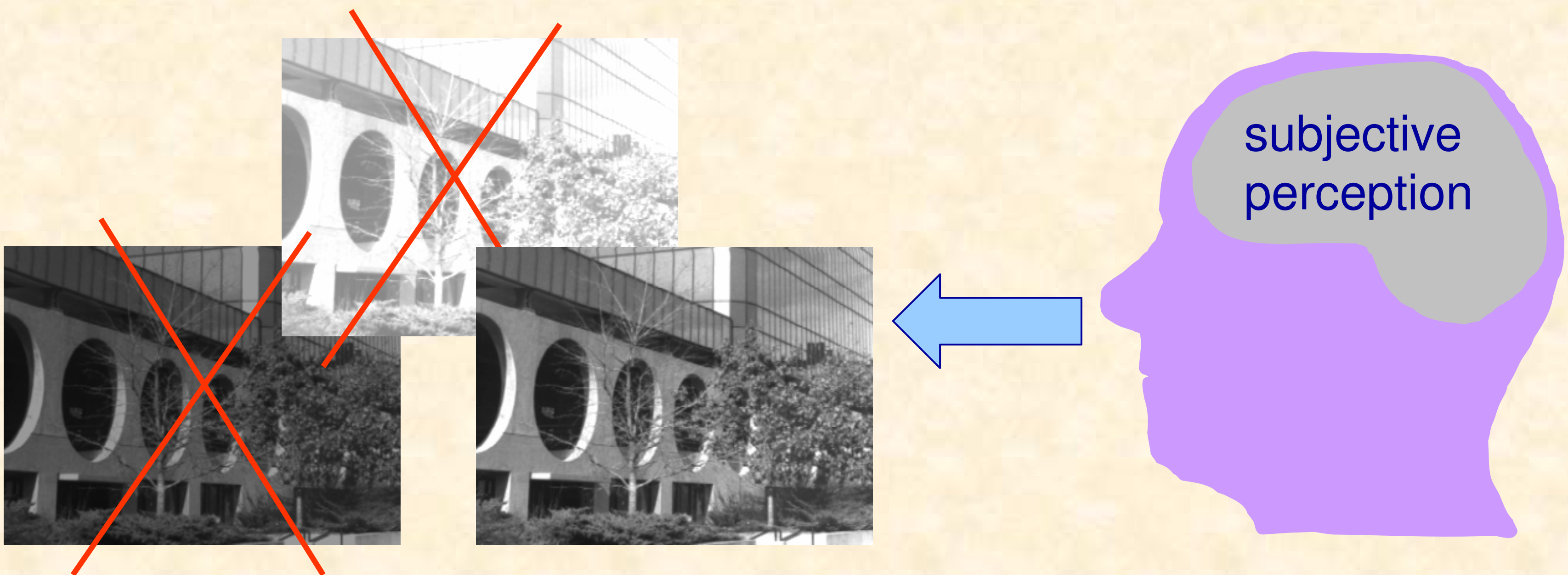
**Image enhancement** belongs to image pre-processing methods.

Objective of image enhancement – process the image (e.g. contrast improvement, image sharpening ,…) so that it is better suited for further processing or analysis

# Image enhancement

Image enhancement methods are based on subjective image quality criteria.

No objective mathematical criteria are used for optimizing processing results.



# Image enhancemet methods

**Point processing**

- Contrast enhancement
- Histogram modelling
- Image averaging

**Spatial filtering**

- Linear filters
- Nonlinear filters
- Edge detection
- Zooming

**Image colouring**

- Pseudo colouring
- False colouring

# Image enhancement

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

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

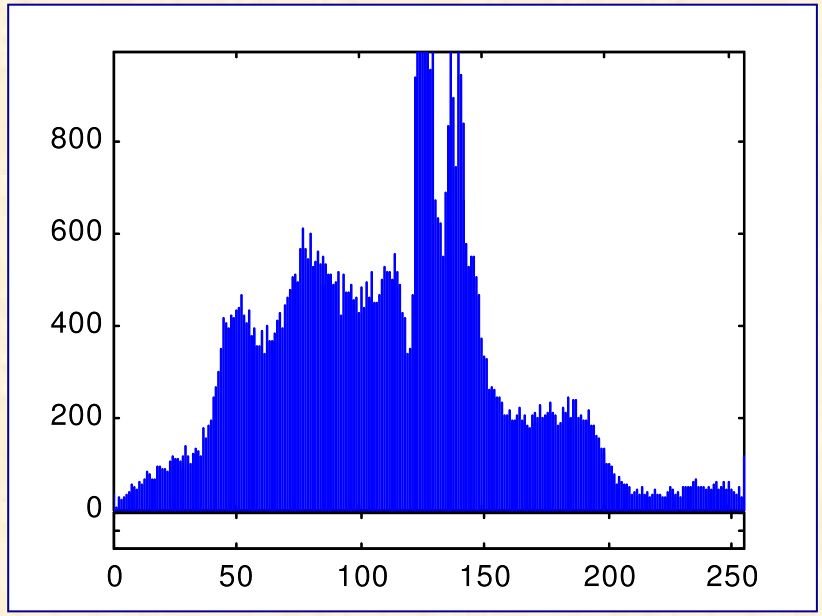
*M, N* – image dimensions

*f*(*i,j*) – gray level value at (*i,j*)


# Image histogram

Image **brightness** and **contrast** influence image subjective quality perception


# Image histogram




```
Image : array[1..M,1..N] of byte;
Hist : array[0..L-1] of longint;
...
Hist:=0;
for i:=1 to M do for j:=1 to N do
        Inc( Hist[ Image[i, j] ] );
…
```

imhist(**I**)

# Image histogram

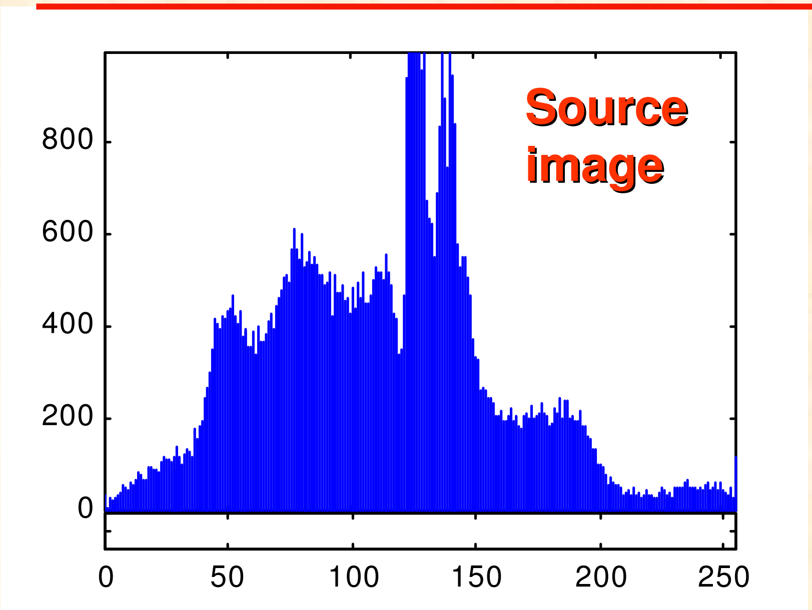
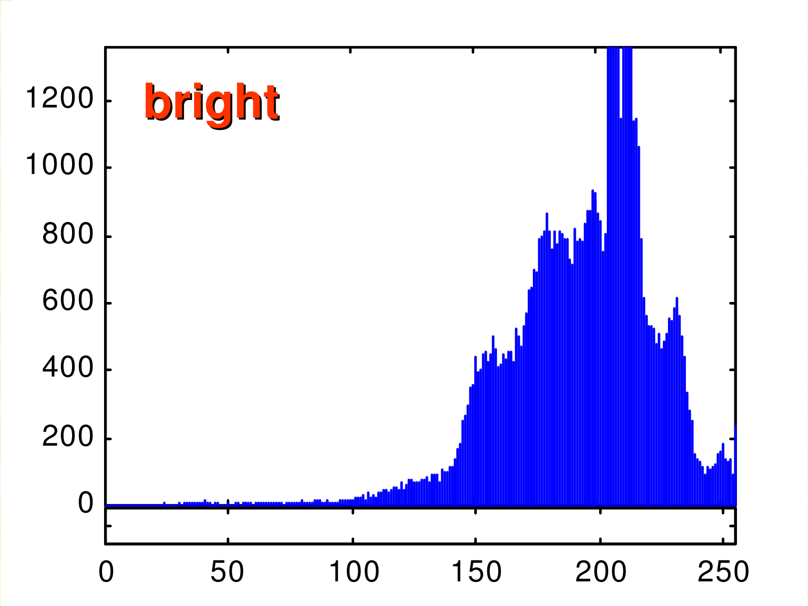
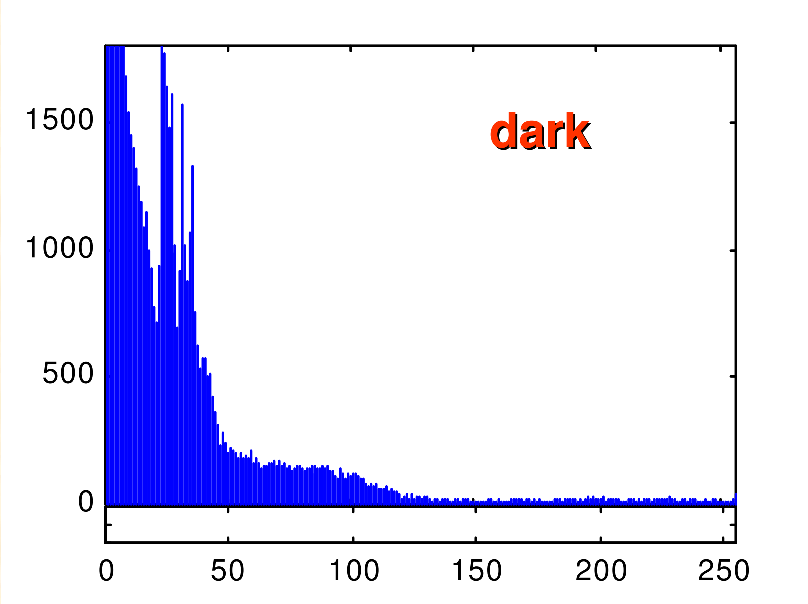
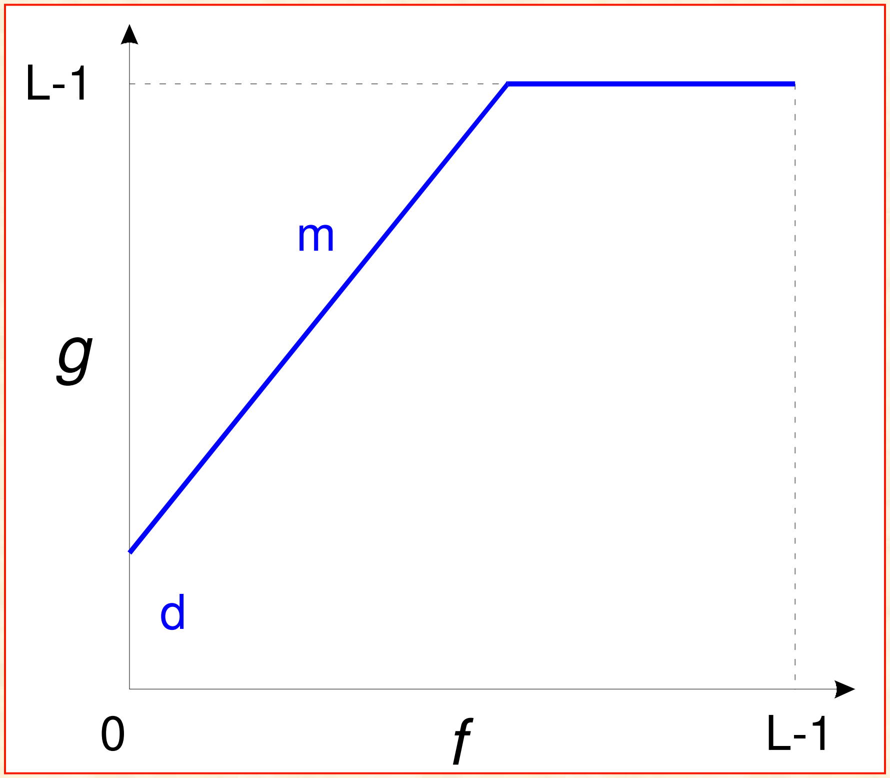



**Image histogram** represents statistical distribution of image pixel brightnesses



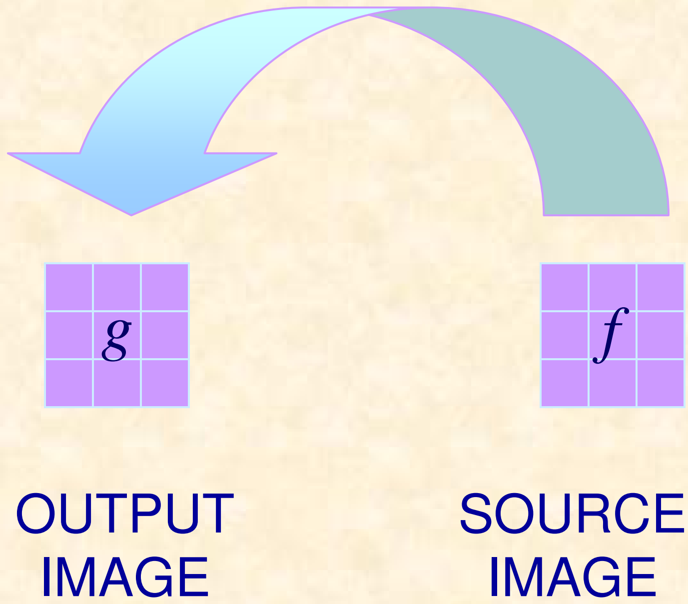
# Linear gray scale transformation




$m$ ~ contrast

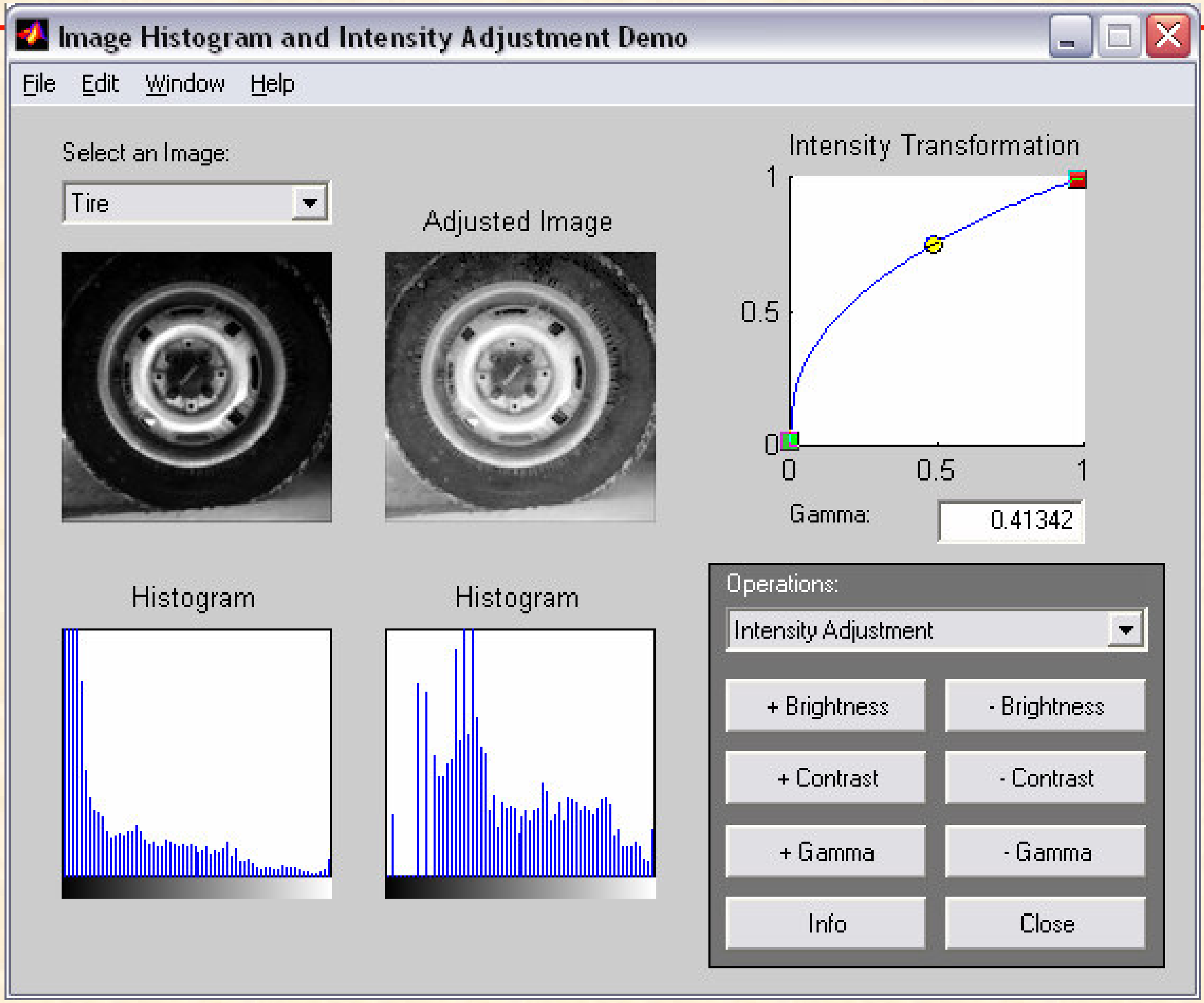
$d$ ~ brightness

$$g(i,j) = \mathrm{m}\, f(i,j) + \mathrm{d}$$

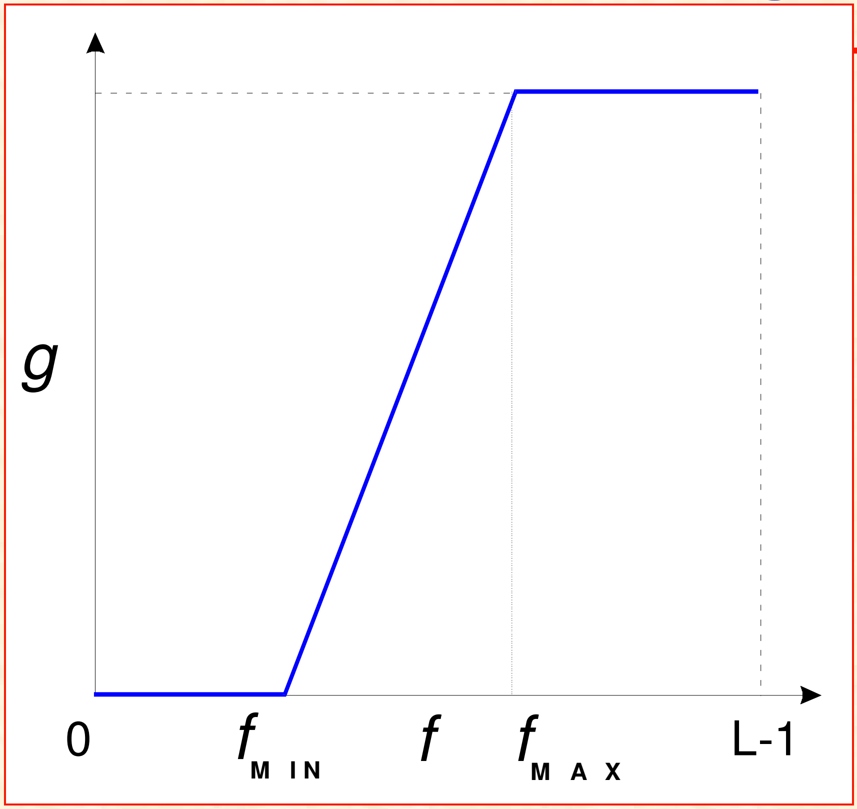



**POINT OPERATION**

# MATLAB Demo – image histogram



# Histogram „stretching"

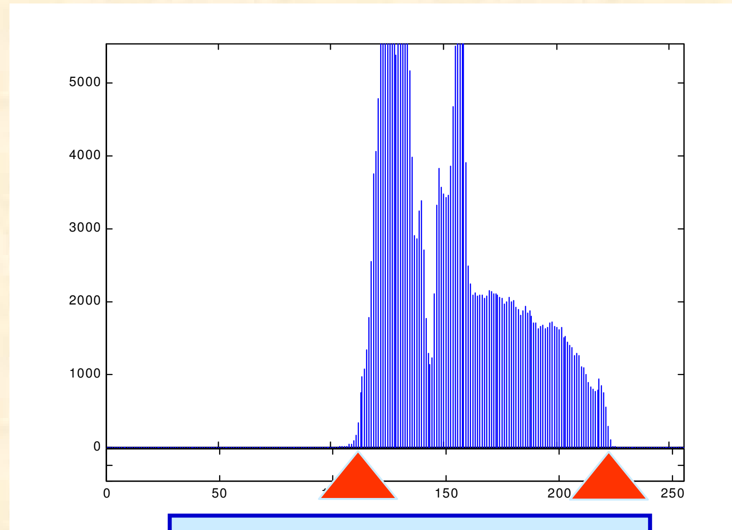



**POINT OPERATION?**

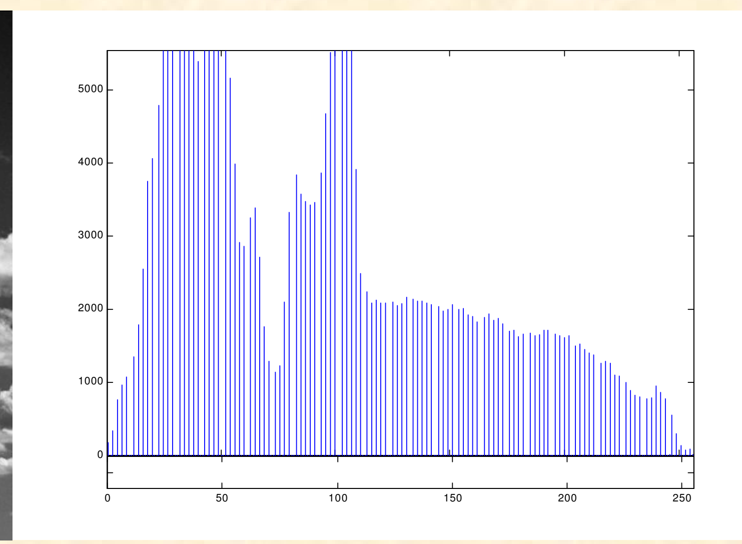
**G**=imadjust(**F**, [$\mathbf{f_{MIN}}$ $\mathbf{f_{MAX}}$], [$\mathbf{g_{MIN}}$ $\mathbf{g_{MAX}}$])

$$g(i,j) = \begin{cases} 0 & f(i,j) < f_{MIN} \\ \dfrac{L-1}{f_{MAX} - f_{MIN}}(f(i,j) - f_{MIN}), & f_{MIN} \leq f(i,j) \leq f_{MAX} \\ L-1 & f(i,j) > f_{MAX} \end{cases}$$

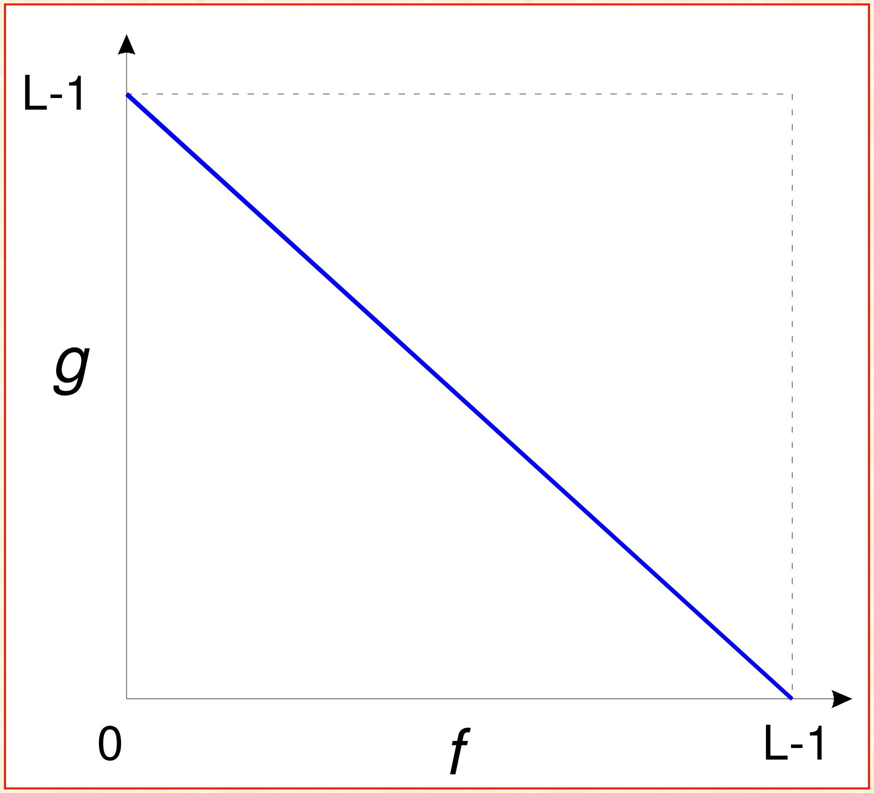
# Histogram „stretching” - example




$f_{MIN}$=110, $f_{MAX}$=225

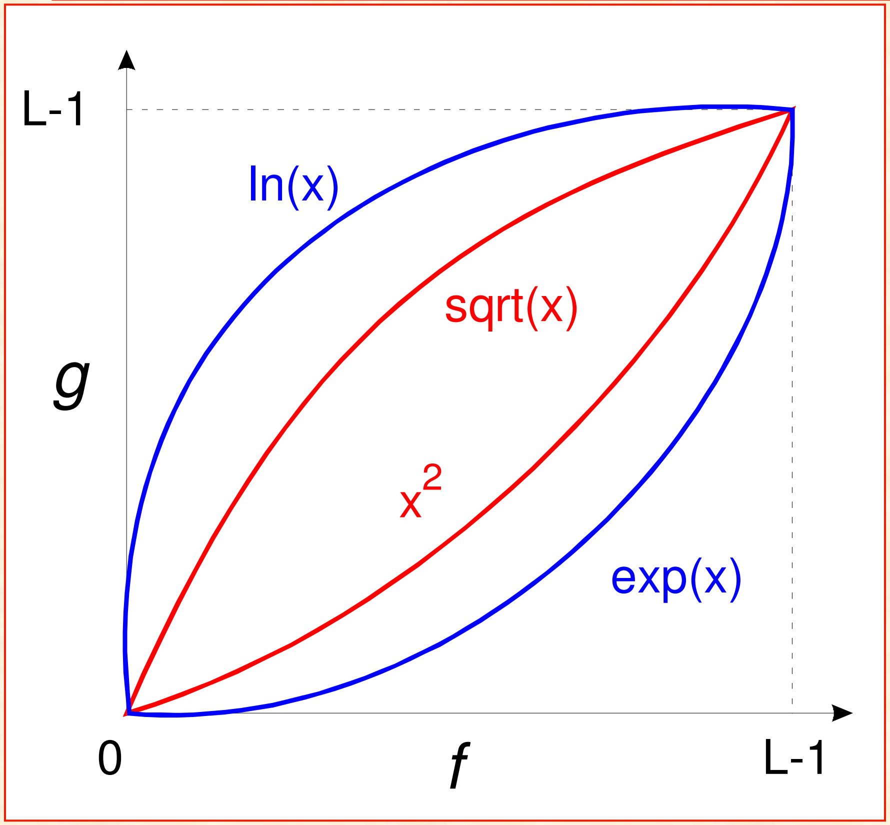


# Grayscale inversion

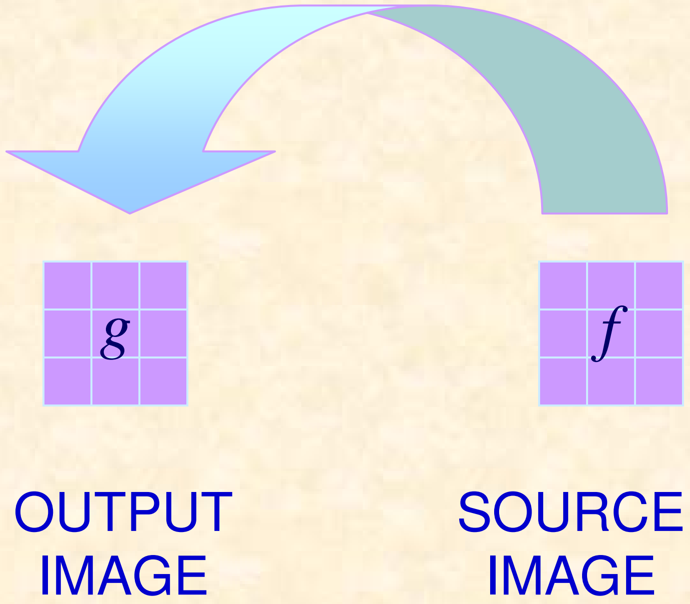



**We can use look-up table to implement image point operations**

# Nonlinear grayscale transformation



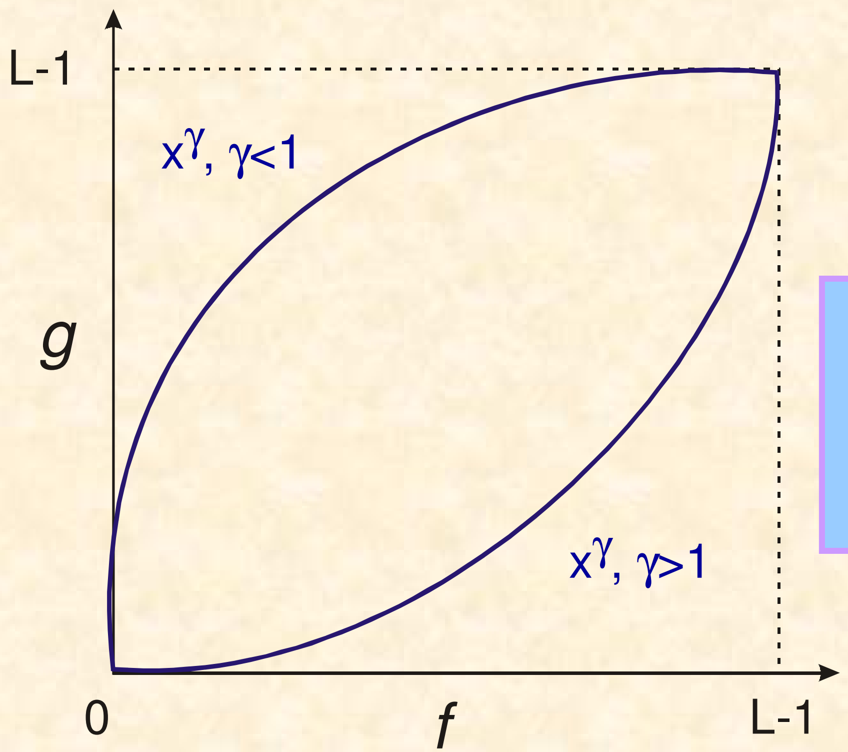

$$g(i,j) = \mathrm{T}(\,f(i,j))$$




**Grayscale normalization!**

**POINT OPERATION**

# Nonlinear grayscale transformation

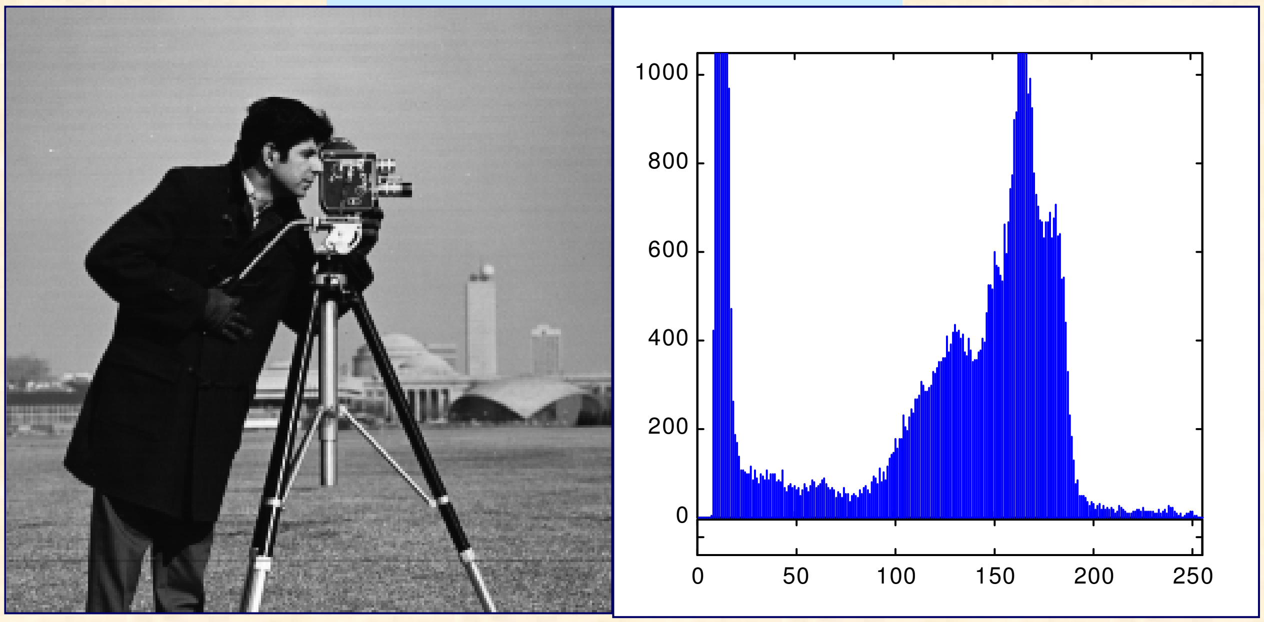



**G**=imadjust(**F**, [$\mathbf{f_{MIN}}$ $\mathbf{f_{MAX}}$], [$\mathbf{g_{MIN}}$ $\mathbf{g_{MAX}}$], $\gamma$)

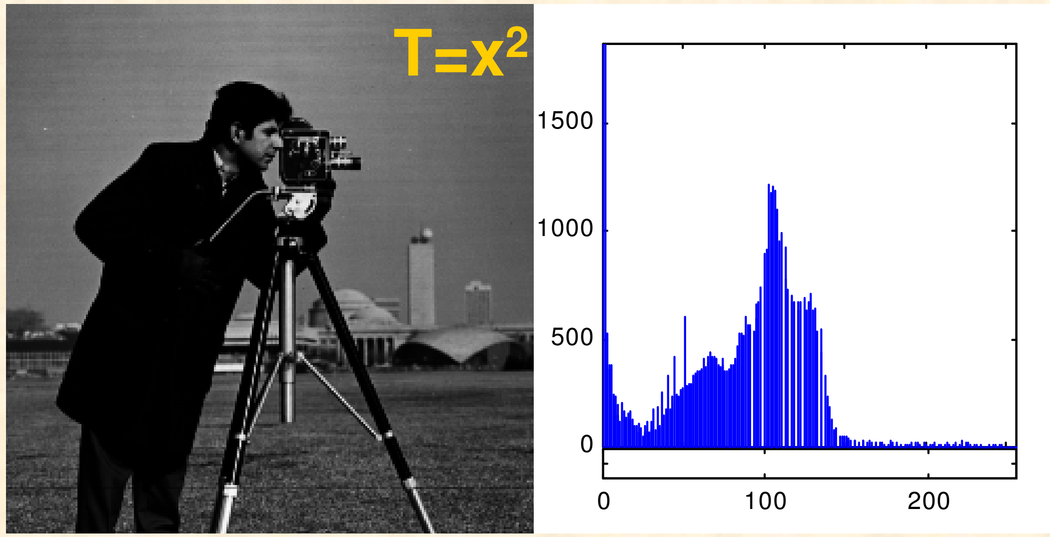
**$\gamma$ correction**

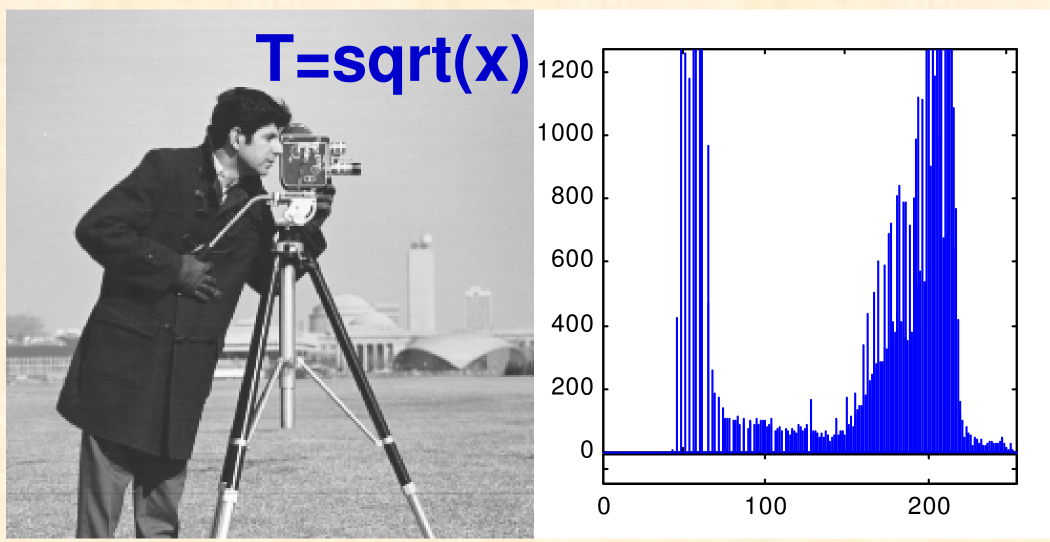
# Nonlinear grayscale transformation - example

Source image

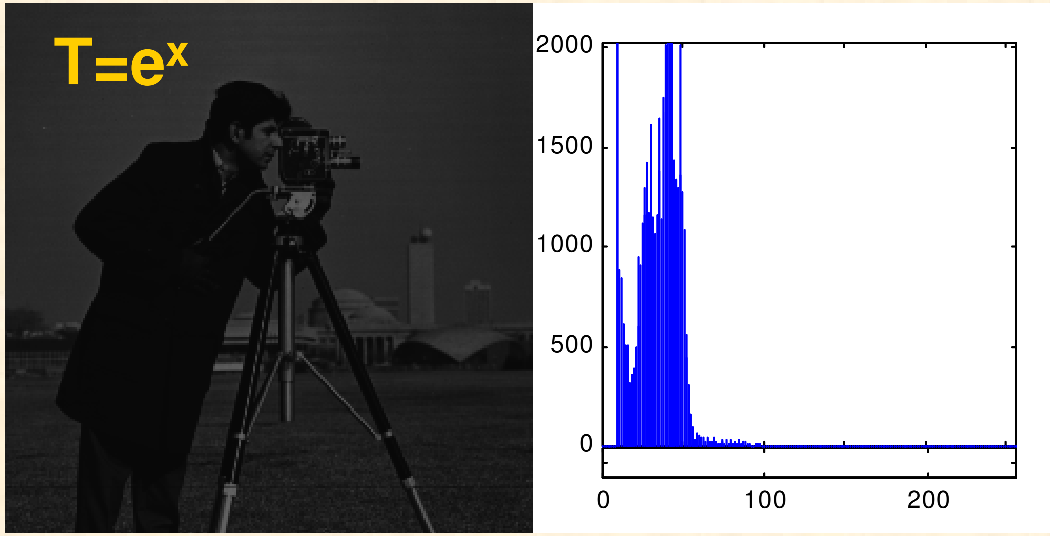


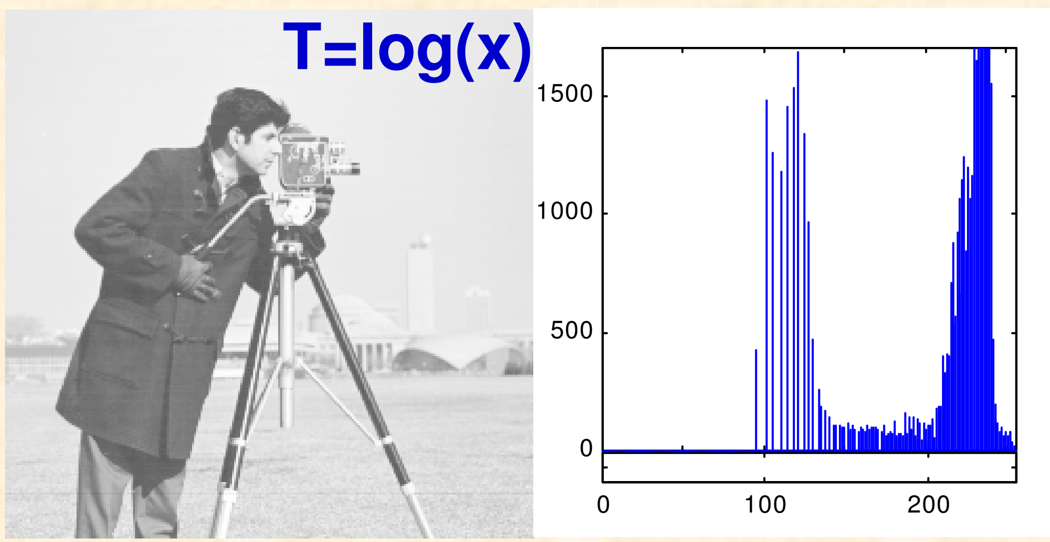
# Nonlinear grayscale transformation - example



# Nonlinear grayscale transformation - example



# Nonlinear grayscale transformation - algorithm

Example: square function

normalization: minimum value - 0 -> 0

maximum value - 255 -> $255^2$

Normalization coefficient: norm=1/255

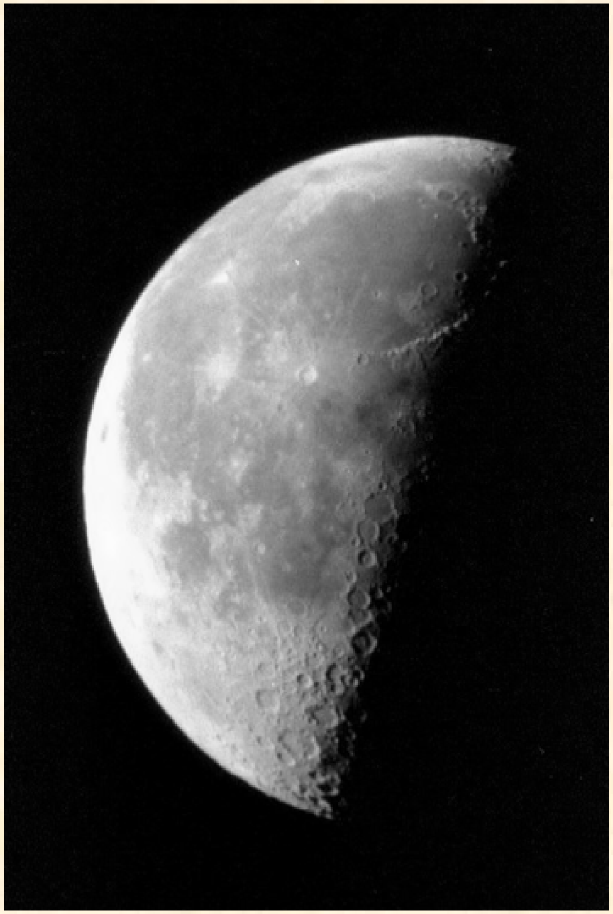
```
...
for i:=1 to M do for j:=1 to N do
        g[i,j]:=round(sqr(f[i,j])*norm);
...
```

# Nonlinear grayscale transformation - algorithm

Example: square function (using look-up-table)

```
lut : array[0..255]of byte;
...
for k:=0 to 255 do lut[k]:=round(k*k*norm)
for i:=1 to M do for j:=1 to N do
        g[i,j]:=lut[(f[i,j])];
...
```

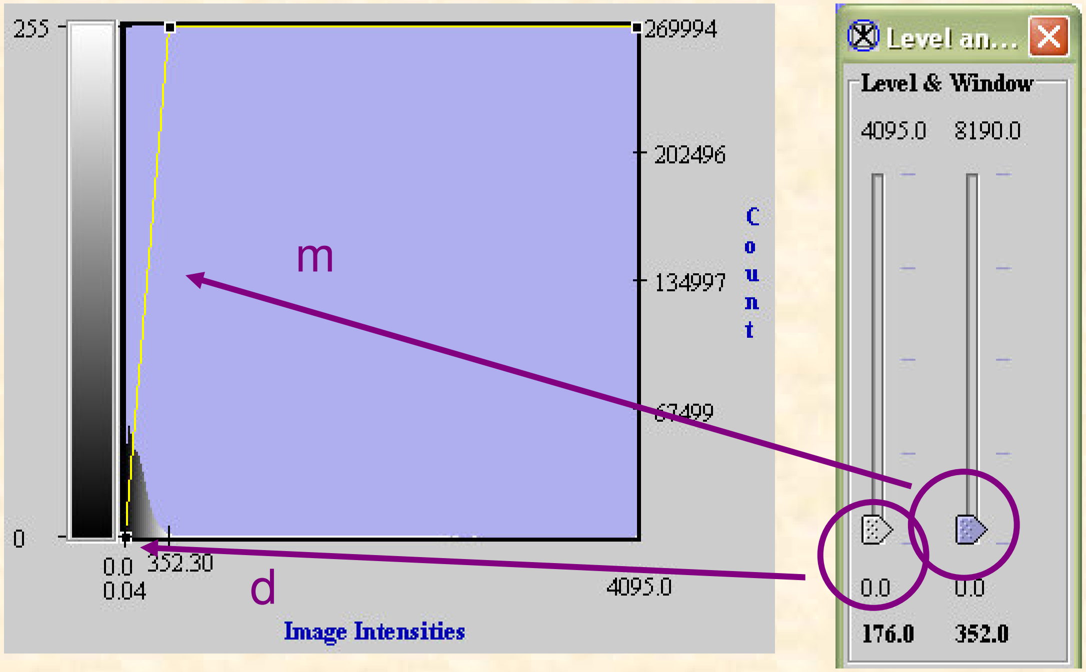
# Enhacement of a telescope moon image



$T = b \log(ax)$

# Linear gray scale transformation


MR 12 bit image




histogram

Brightness/Contrast
adjustment window

Medical Image Processing, Analysis and Visualization (MIPAV)
by Center for Information Technology, ver. 1.29

# Image ehancement by image averaging

Consider a noisy image:

$$\boxed{g(i,j) = f(i,j) + \eta(i,j)}$$

contaminated by additive noise $\eta(i,j)$ of zero average an variance $\sigma_\eta^2$ that is not correlated to the image.

We will show that after $N$ averagings (acquisitions) of the noisy image $g(i,j)$ the variance of noise component will be reduced to:

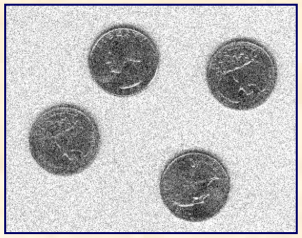
$$\boxed{\overline{\sigma_\eta^2} = \frac{\sigma_\eta^2}{N}}$$

# Image ehancement by image averaging

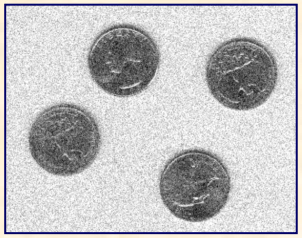
$$g(i,j)=\frac{1}{N}\sum_{k=1}^{N}[f(i,j)+n_k(i,j)]=f(i,j)+\frac{1}{N}\sum_{k=1}^{N}n_k(i,j)$$

**WARNING ! – grayscale range**

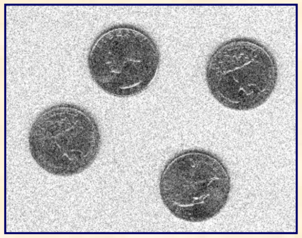
$\frac{1}{N}\Big($



+



+...+



$\Big)$ =

# Image ehancement by image averaging

Noise variance in the averaged image:

$$\sigma_\eta^2 = E\left\{\left(\frac{1}{N}\sum_{k=1}^{N}\eta_k\right)^2\right\} = \frac{1}{N^2}\cdot E\left\{\left(\sum_{k=1}^{N}\eta_k\right)^2\right\} =$$

$$= \frac{1}{N^2}\cdot E\left\{(\eta_1 + \eta_2 + \ldots + \eta_N)^2\right\} = \frac{1}{N^2}\cdot E\left\{\sum_{k=1}^{N}\eta_k^2 + \underbrace{2\left(\sum_{k \neq p}^{N}\eta_k\eta_p\right)}_{=0}\right\} =$$

$$= \frac{1}{N^2} E\left\{\sum_{k=1}^{N}\eta_k^2\right\} = \frac{1}{N^2} N\sigma_\eta^2 = \frac{1}{N}\sigma_\eta^2$$

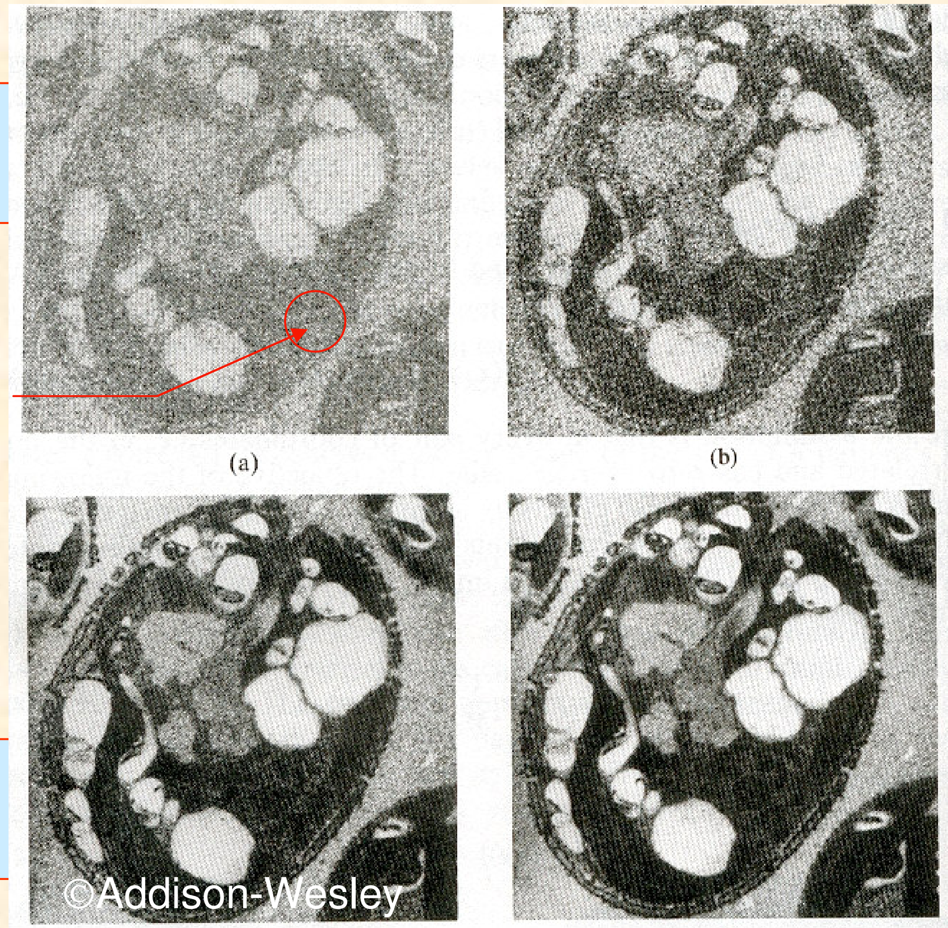
One can also show that the pick value of noise $\{n\}$ is reduced by a factor of $\sqrt{N}$ after $N$ image averagings

# Image averaging – example

N=1

N=2

Additive Gaussian noise




N=8

N=16

Microscope image of a cell

# Cumulative histogram

$$hist - image\ histogram, \quad histc - cumulative\ histogram$$

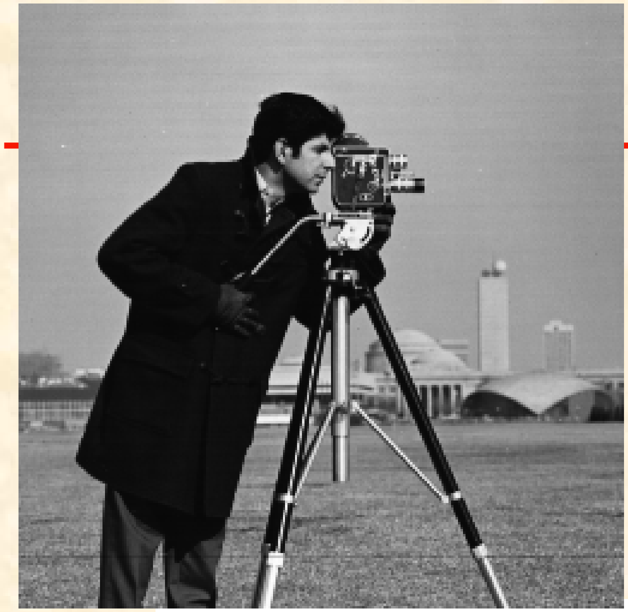
$$hist : array[0..255]\ \ of\ \ \mathbf{longint;}$$

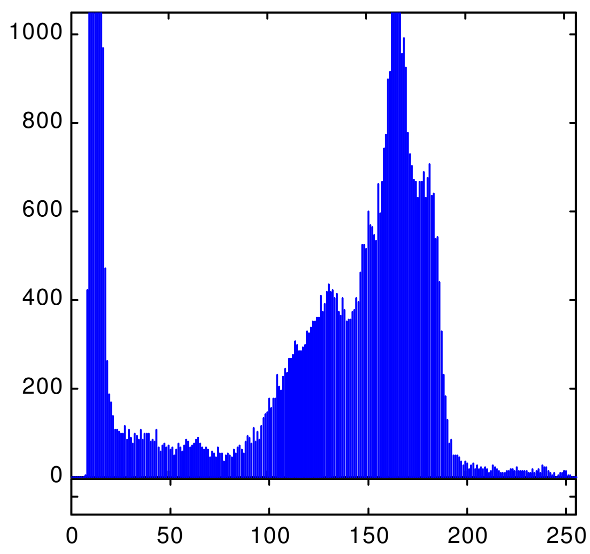
$$hists : array[0..255]\ \ of\ \ \mathbf{single};$$

$$histc[i] = (\sum_{k=0}^{i} hist[k]) / MN, \qquad i = 0,\ldots, L-1$$

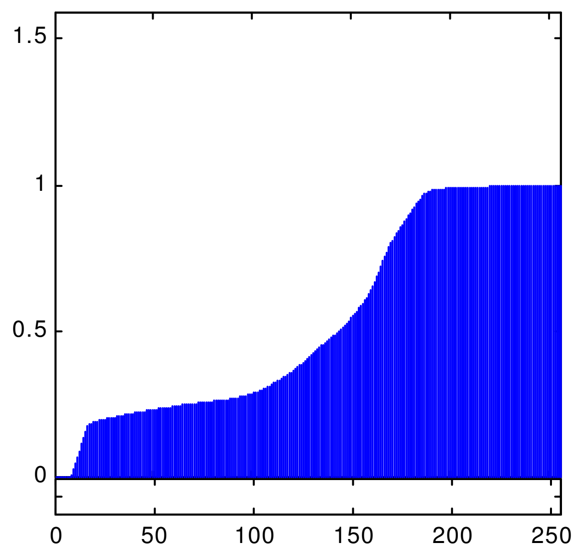
$$M, N - image\ \mathrm{dim}\,ensions$$

# Cumulative histogram






Histogram




Cumulative histogram

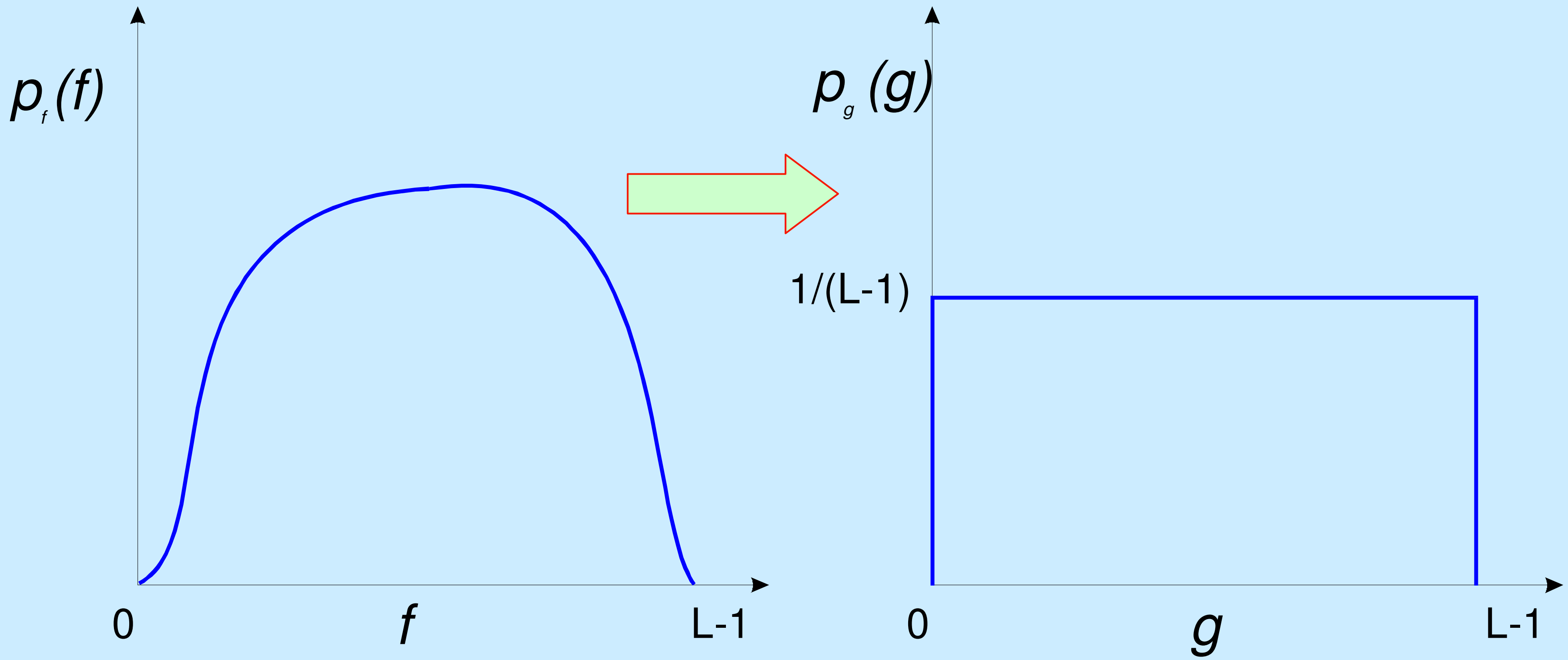
# Histogram equalization

Histogram equalization aims at obtaining uniform statistical distribution of image gray levels (uniform probability density function)

By histogram equalization one gets:

- contrast enhancement
- image normalization

# Histogram equalization




$p_f(f)$=hist[f]/MN

$p_g(g)=1/(L-1)$

# Histogram equalization

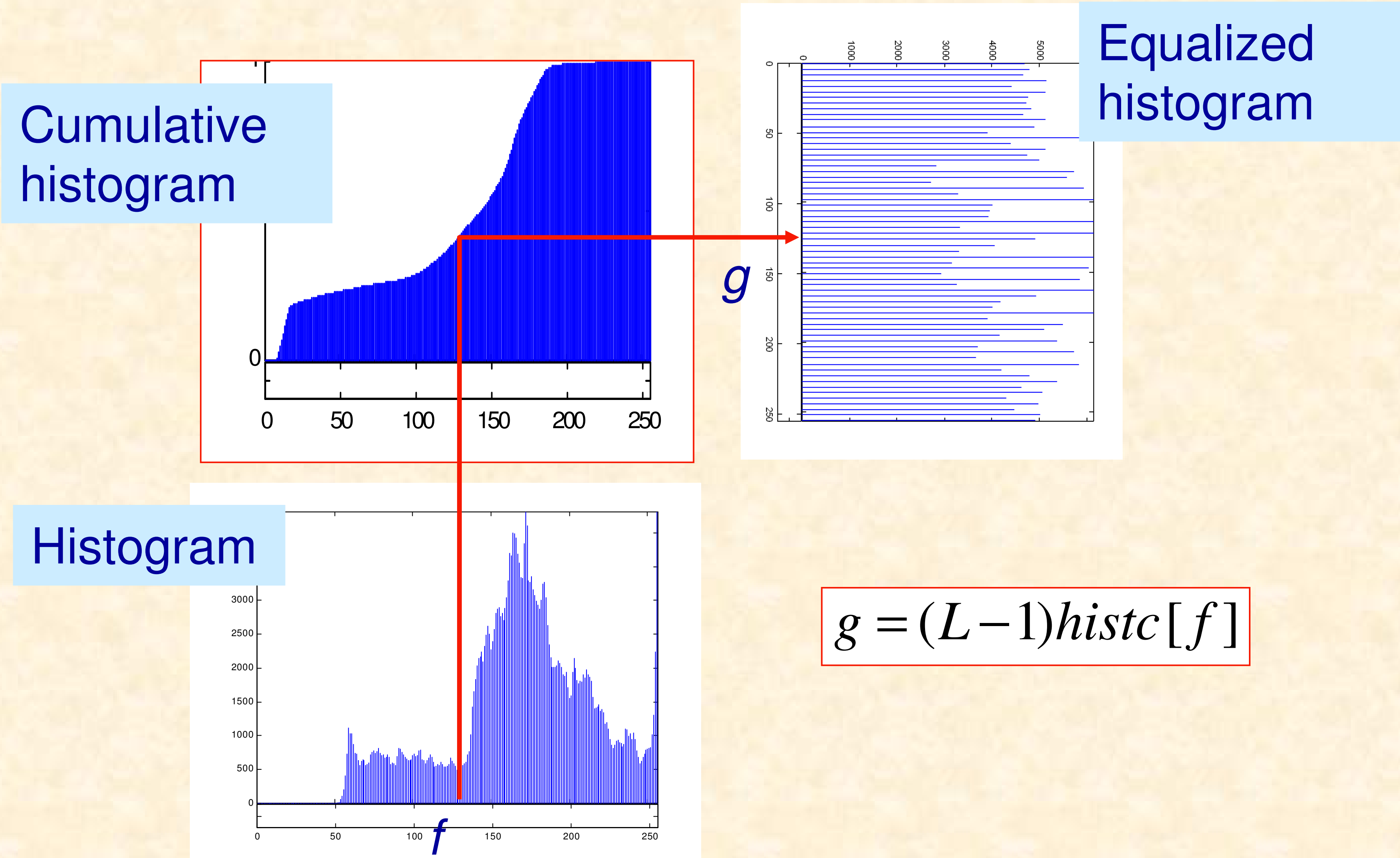
$$\int p_f(h)dh = \int p_g(u)du$$

$$\int_0^f p_f(h)dh = \int_0^g \frac{1}{L-1}du = \frac{1}{L-1}u\Big|_0^g = \frac{g}{L-1} \qquad 0 \le f, g \le L-1$$

$$\sum_{i=0}^{f} p_f(i) = \frac{g}{L-1} \qquad f, g = 0, 1, 2, \ldots, L\text{-}1$$

$$g = (L-1)\sum_{i=0}^{f} p_f(i) = (L-1)\sum_{i=0}^{f} \frac{hist[i]}{MN} = (L-1)histc[f]$$

# Histogram equalization



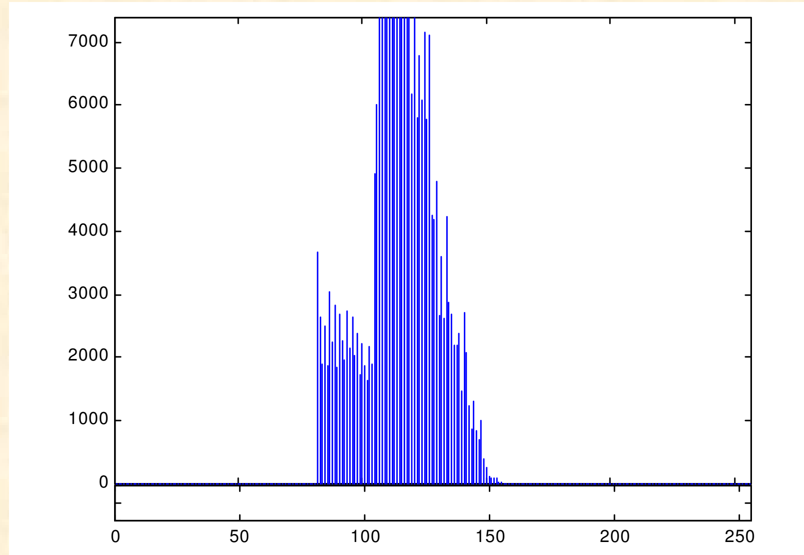

$$g = (L-1)histc[f]$$

# Cumulative histogram - algorithm

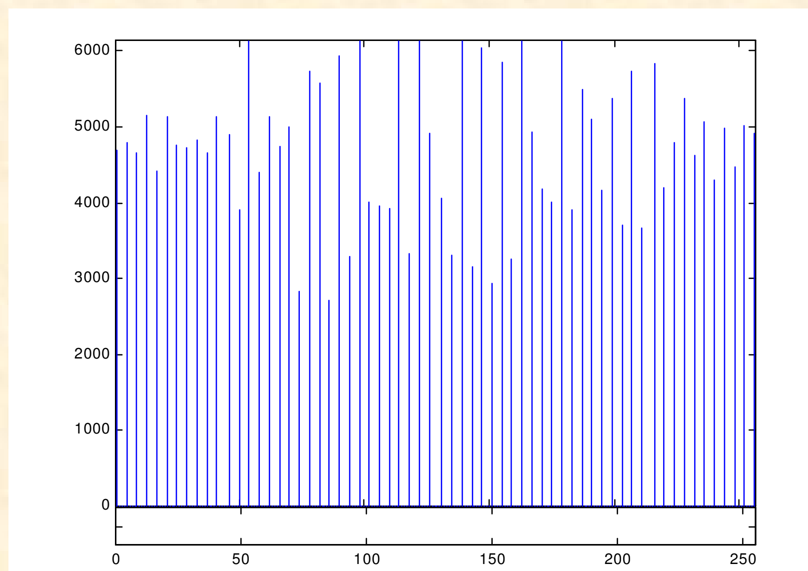
```
hist : array[0..255] of longint;
histc : array[0..255] of single;

...
histc[0]:=hist[0];
for k:=1 to 255 do
    histc[k]:=histc[k-1]+hist[k];
...
```

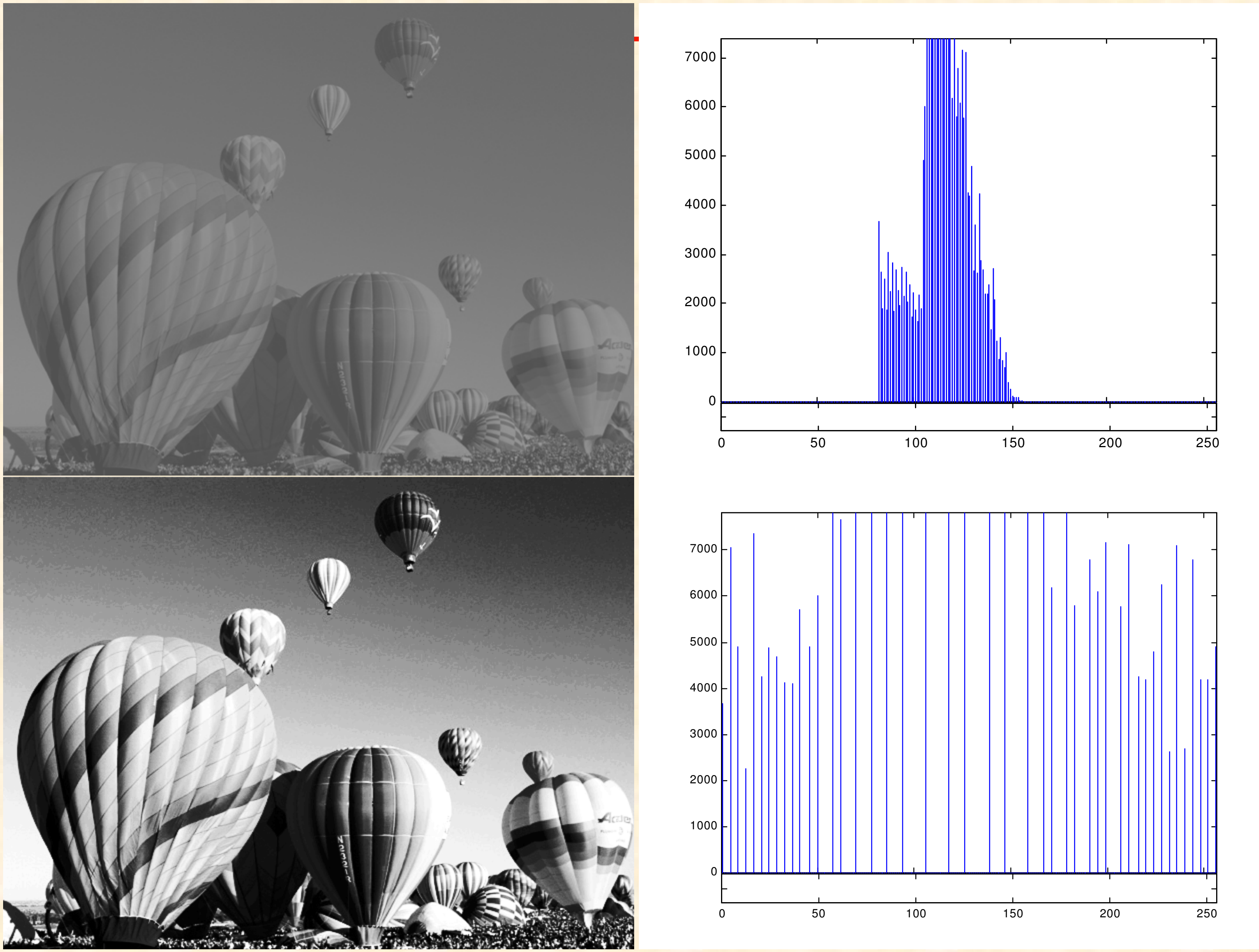
# Histogram equalization



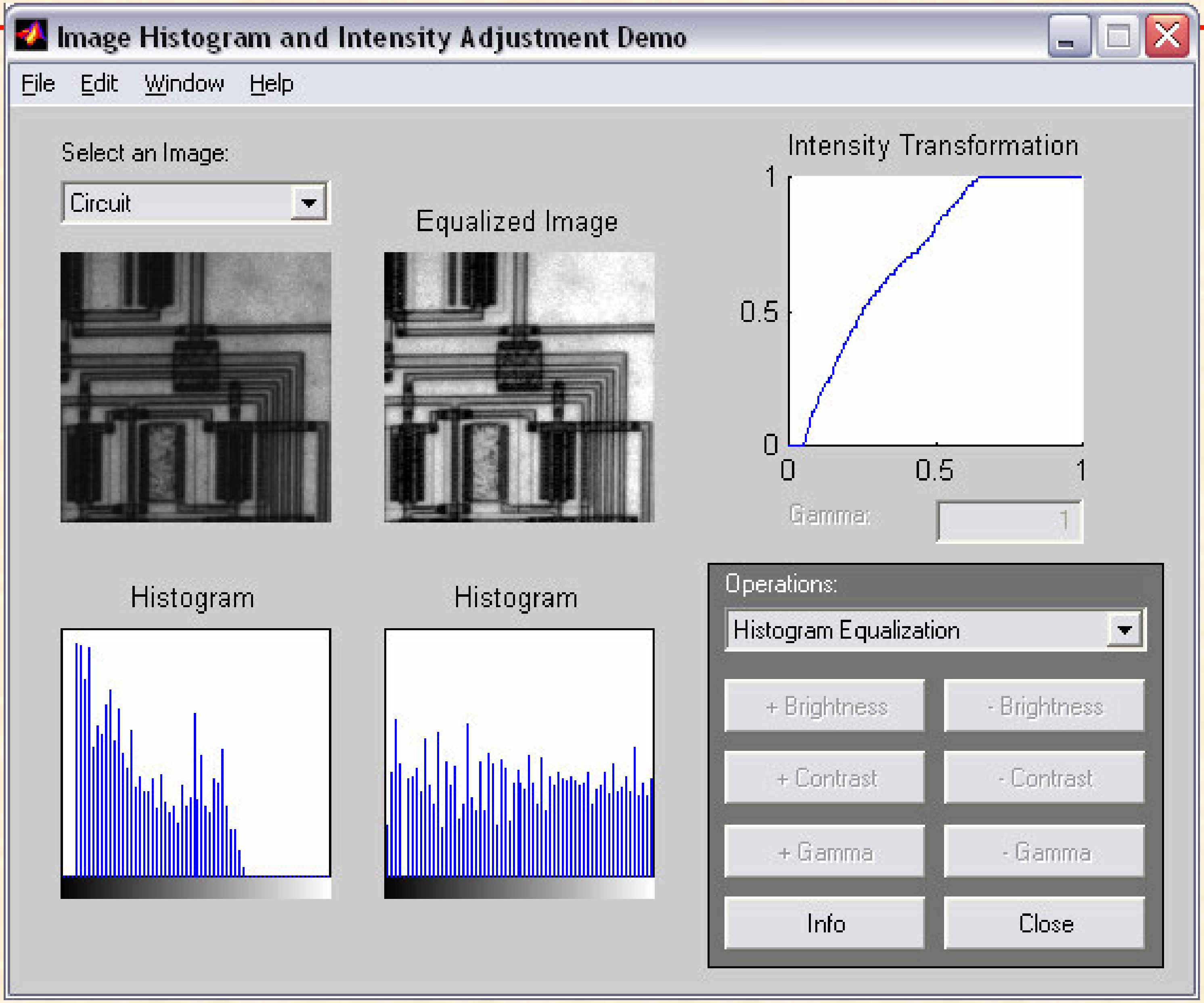

**J**=histeq(**I**)



# Histogram equalization - example



# MATLAB Demo – intensity adjustment



# Correction of nonuniform illumination