Image enhancement

Image enhancement belongs to image pre-processing methods.

<u>Objective</u> <u>of image enhancement</u> – 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.

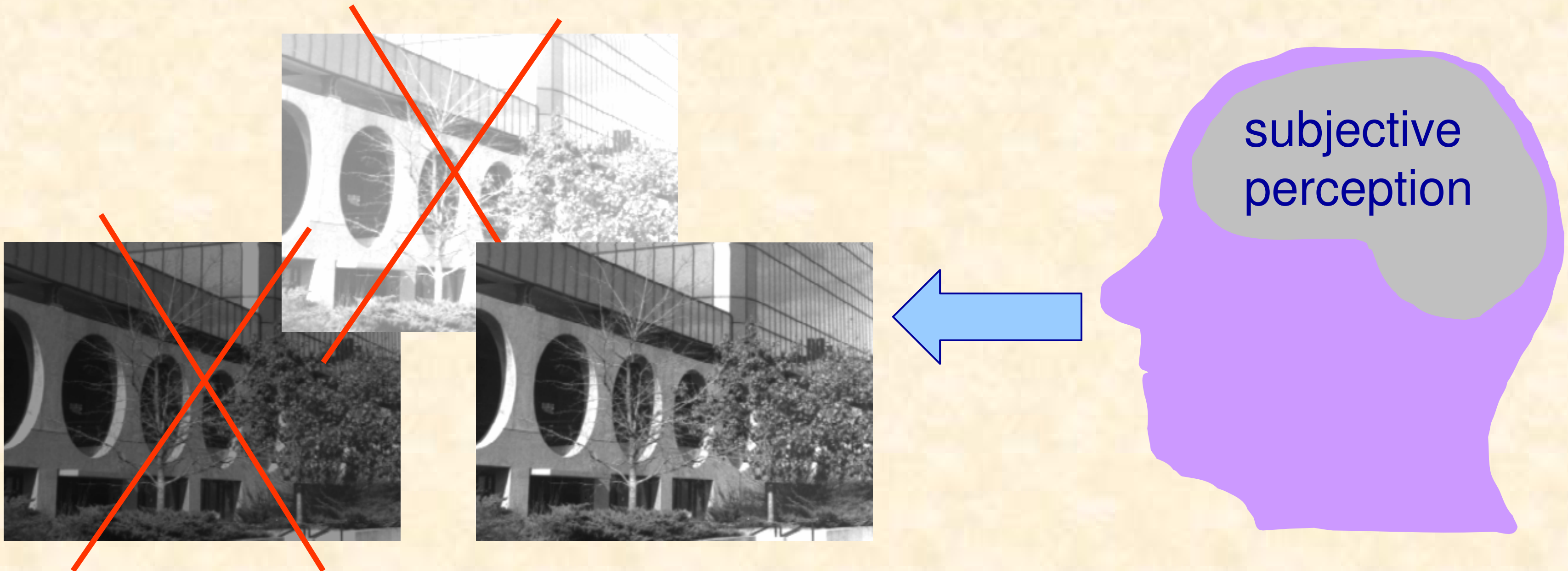


Image enhancemet methods

Point processing

- Contrast enhancement
- Histogram modelling
- Image averaging

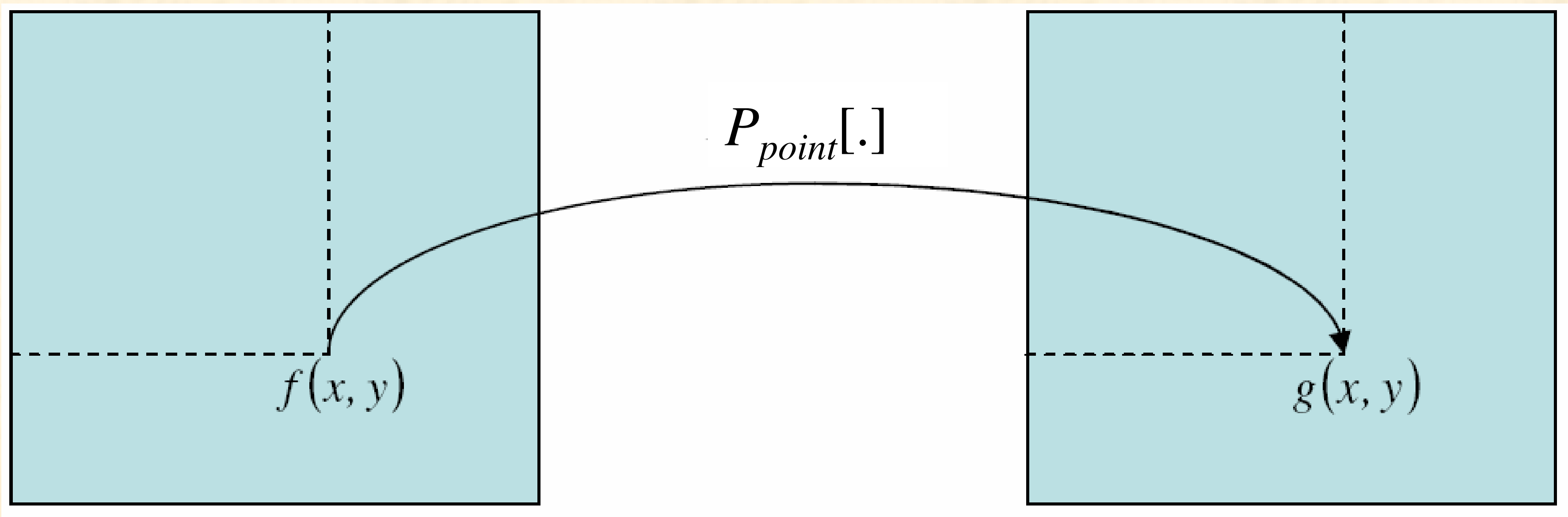
Spatial filtering

- Linear filters
- Nonlinear filters
- Edge detection
- Zooming

Image colouring

- Pseudo colouring
- False colouring

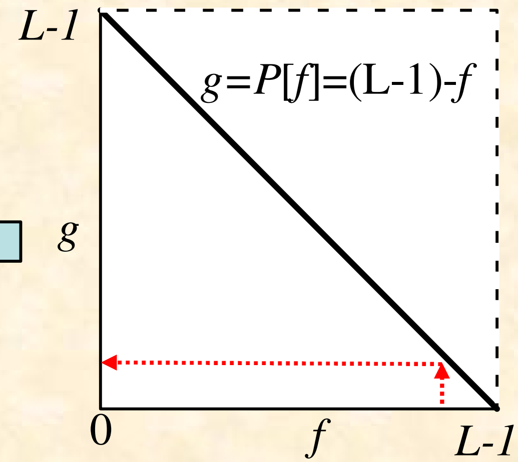
Point-to-point image transformation



Point-to-point transformation example

The operator's independent variable is just pixel brightness (not its coordinate):

$$g = P[f]$$



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

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 – definition 1

Image histogram *h* - is a vector of length *L* (*L* – is the number of image gray levels); ***h***(*q*) – i.e. *g*-th vector element, contains the number of image pixels equal to gray level *q*

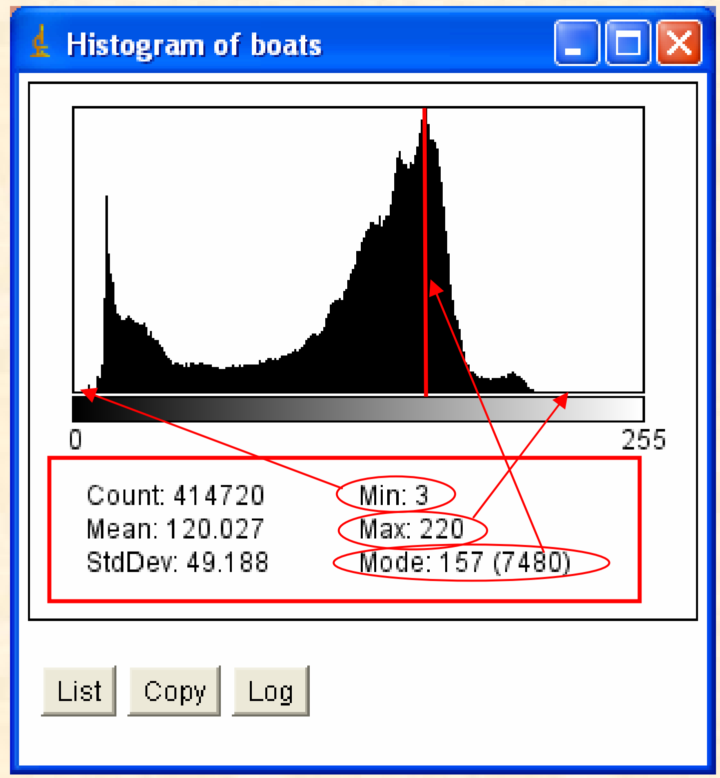


Image histogram – definition 2

$$h(q) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \delta(f(i,j), q)$$

Kronecker deta function:

$$\delta(m.n) = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases}$$

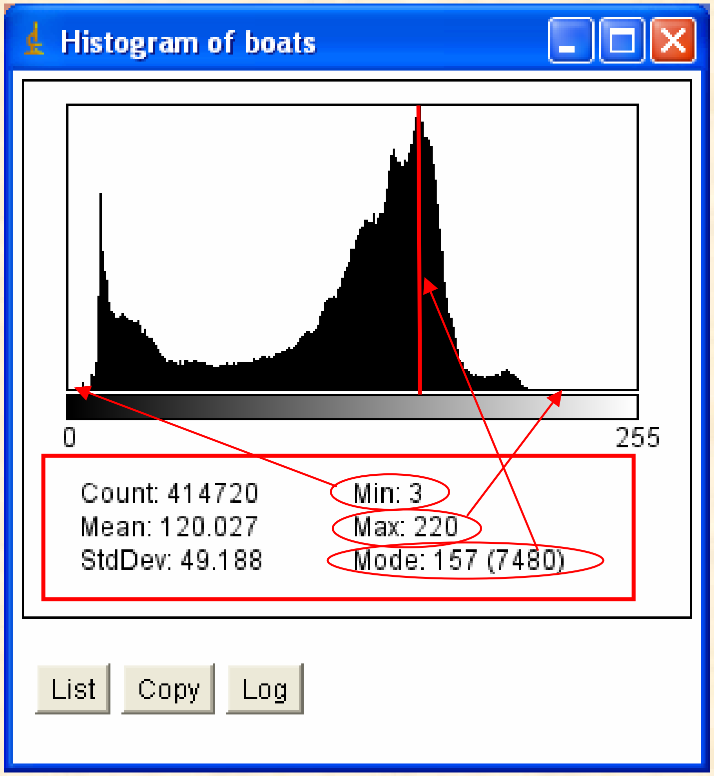
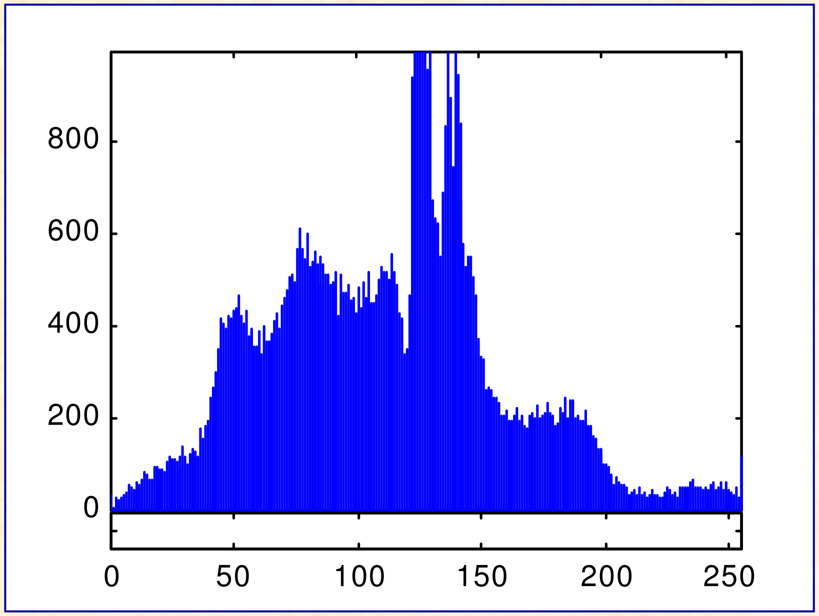


Image histogram - computation




```
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] ] );
…
```

Image histogram

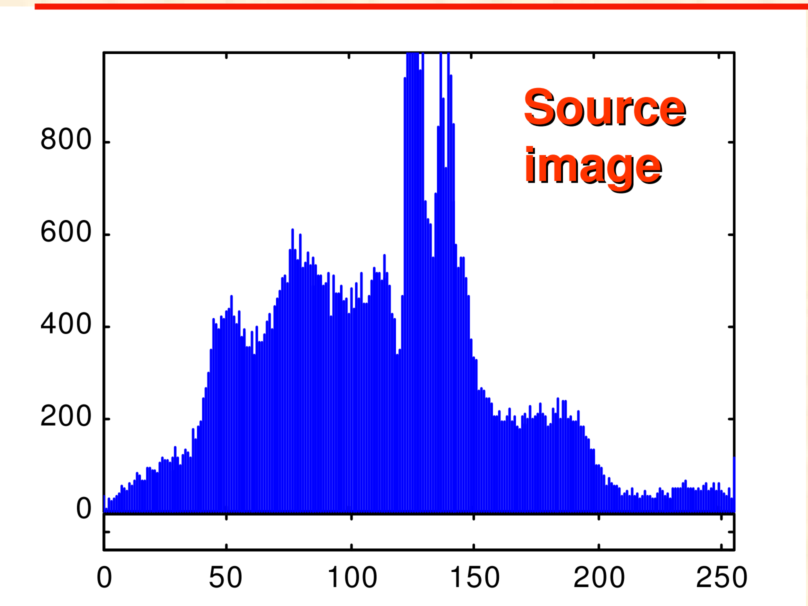
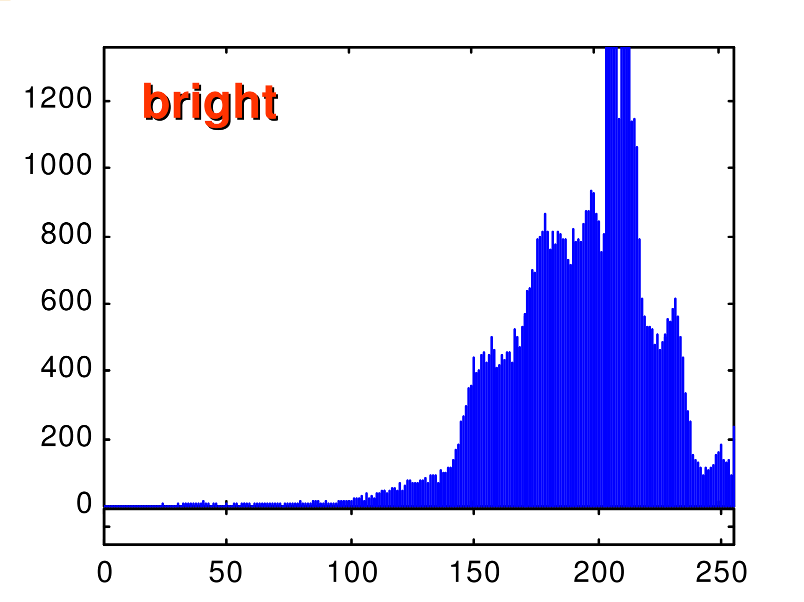


Image histogram represents statistical distribution of image pixel brightnesses ...



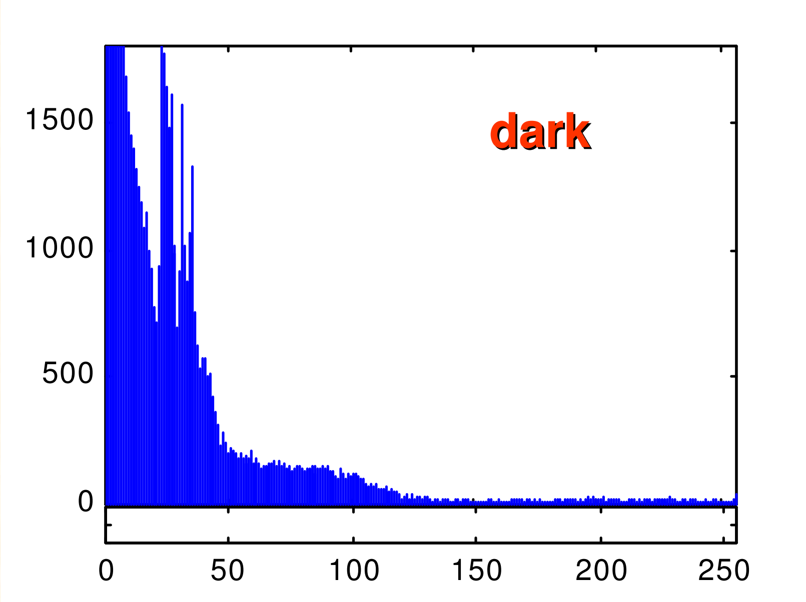
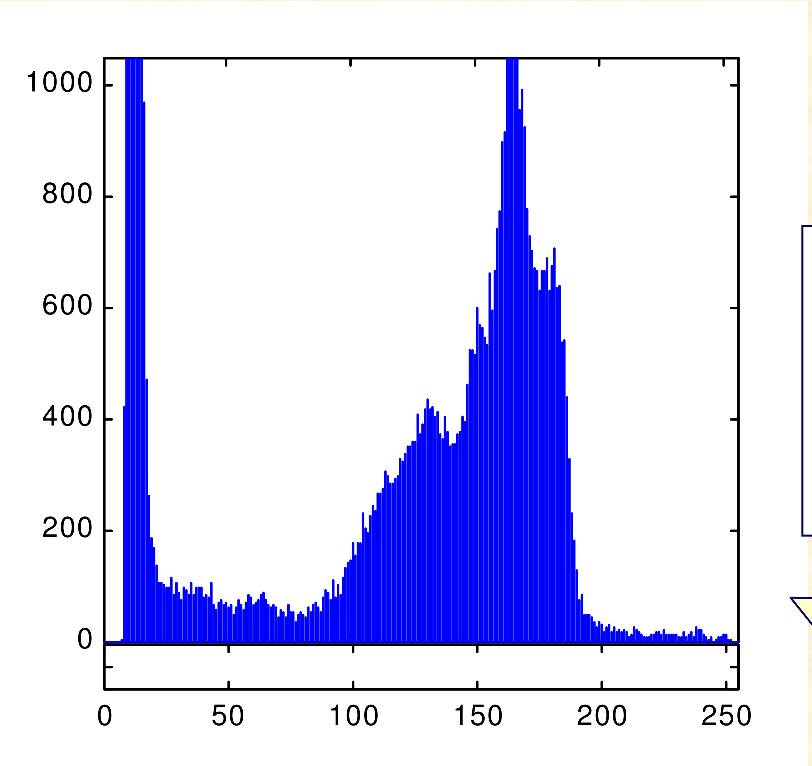


Image histogram

.. but no information about image structure



These two images have identical histograms

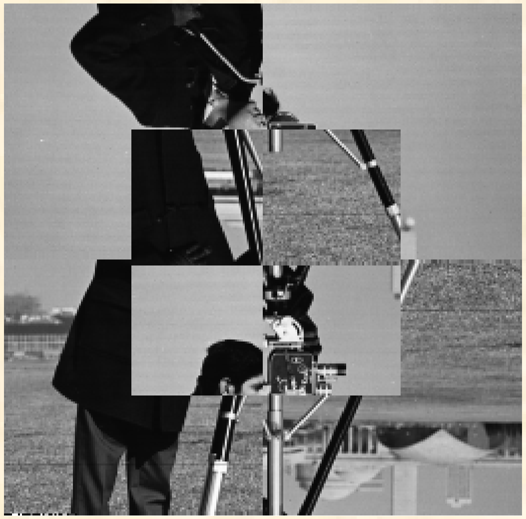
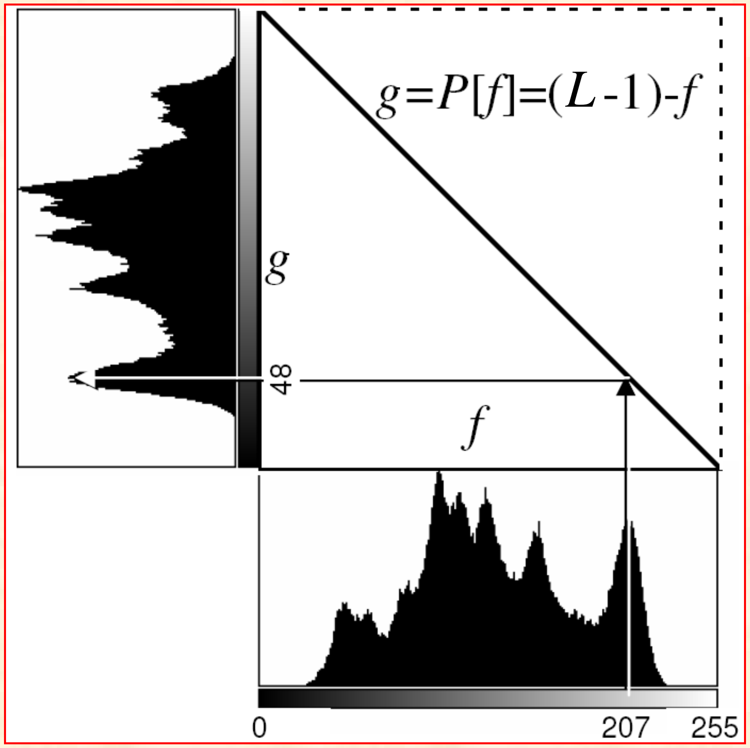
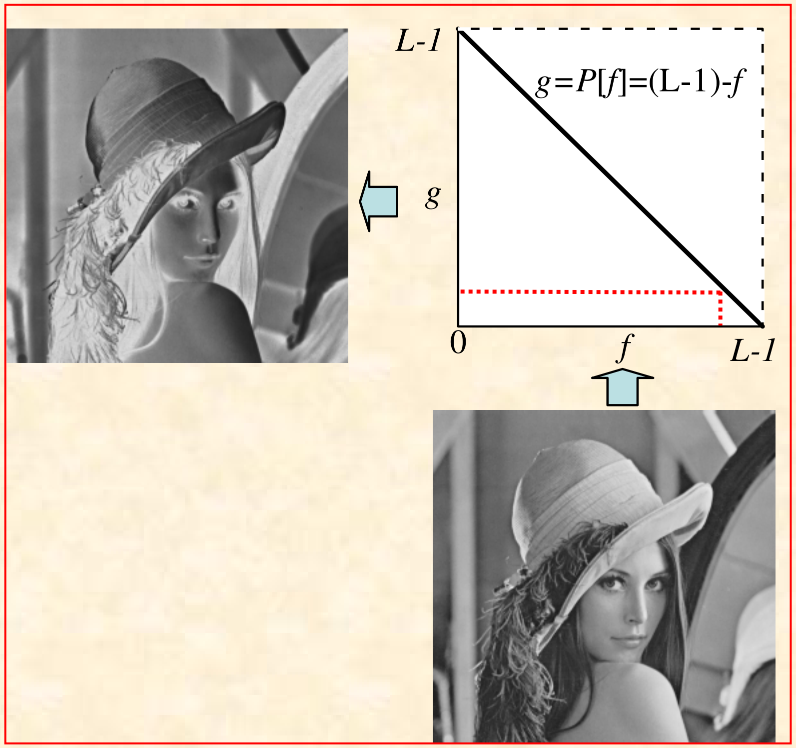


Image histogram – back to point processing

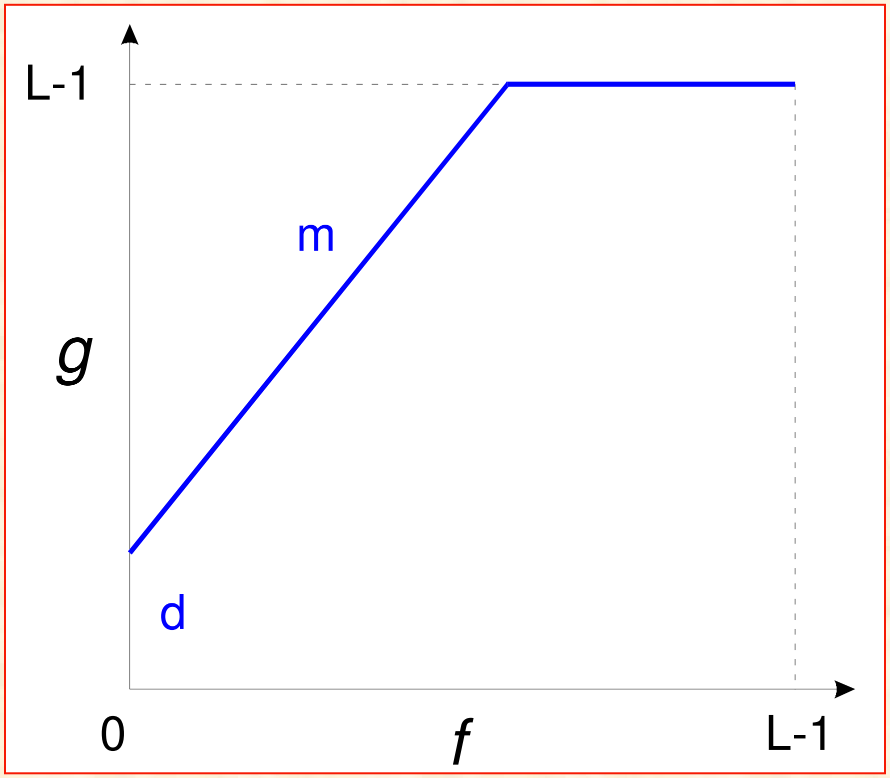


≡

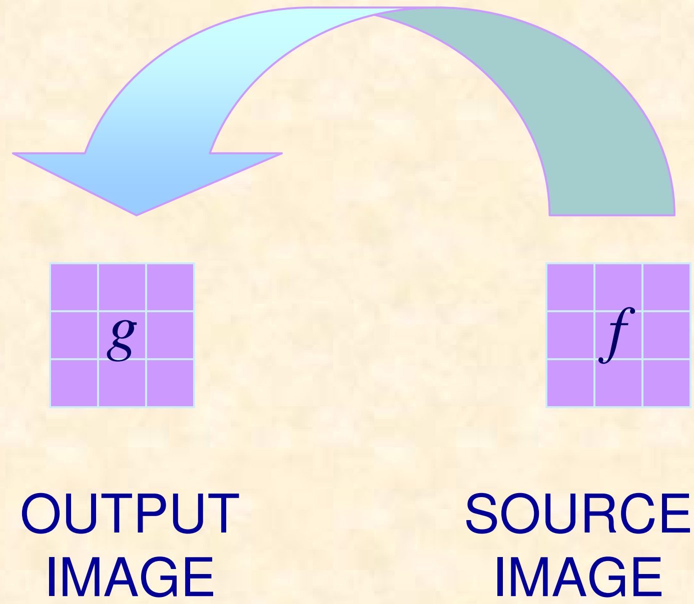


We can use look-up table to implement image point operations

Linear gray scale transformation



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

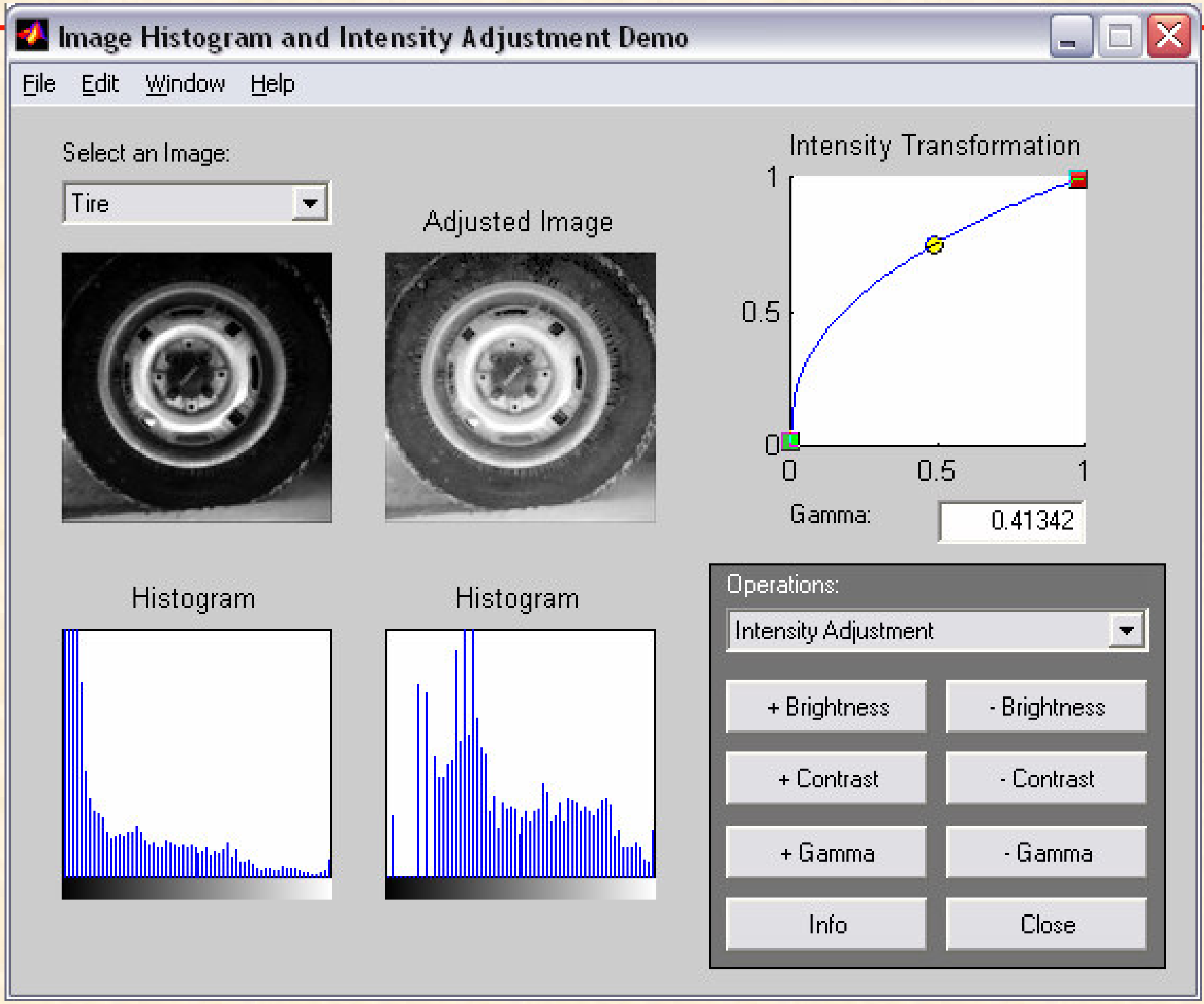


m ~ contrast

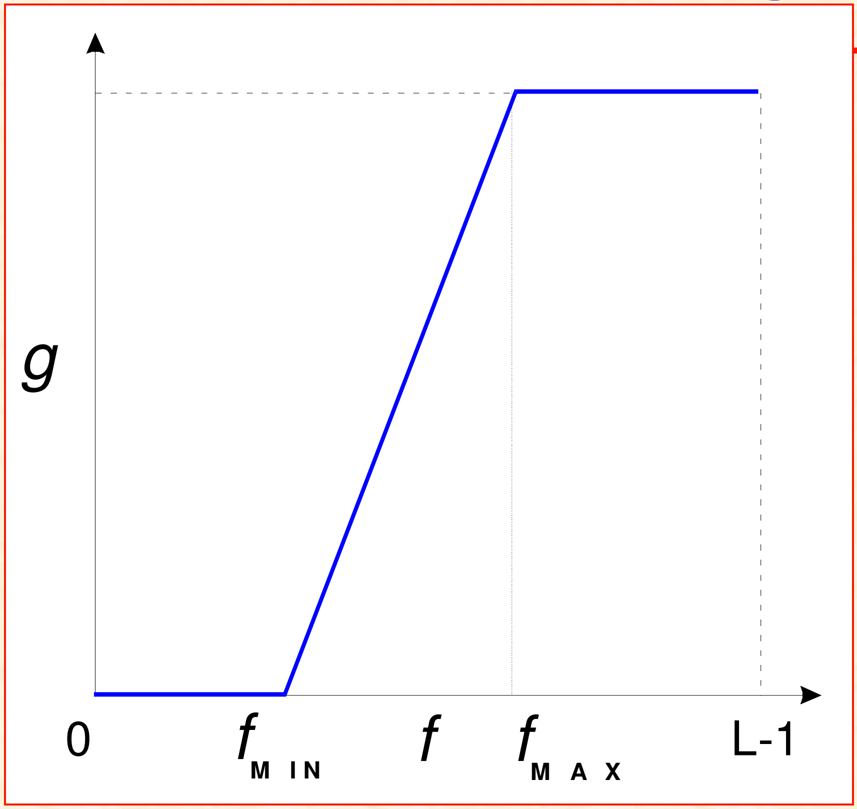
d ~ brightness

POINT OPERATION

MATLAB Demo – image histogram



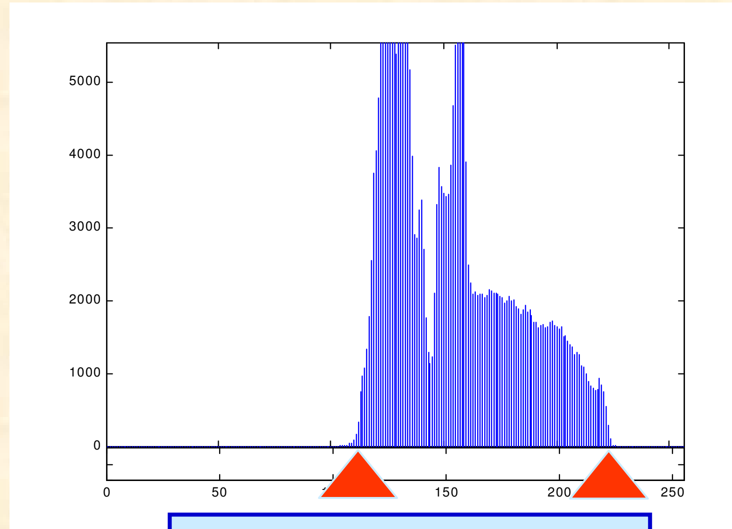
Histogram „stretching”



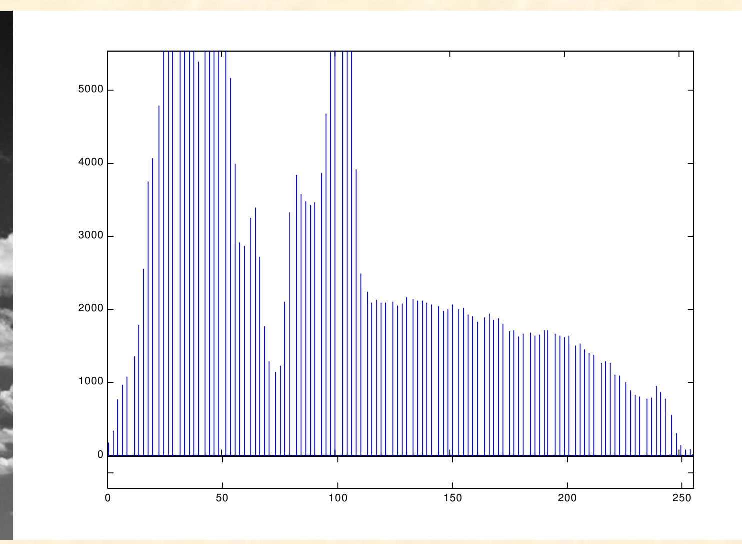
POINT OPERATION?

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

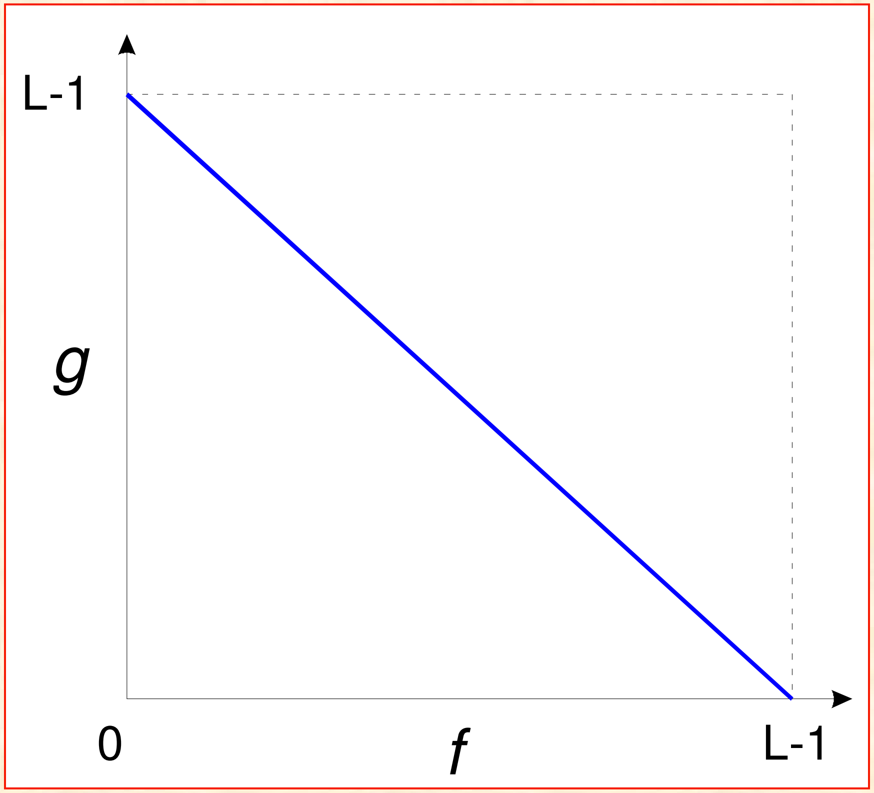
Histogram „stretching” - example



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

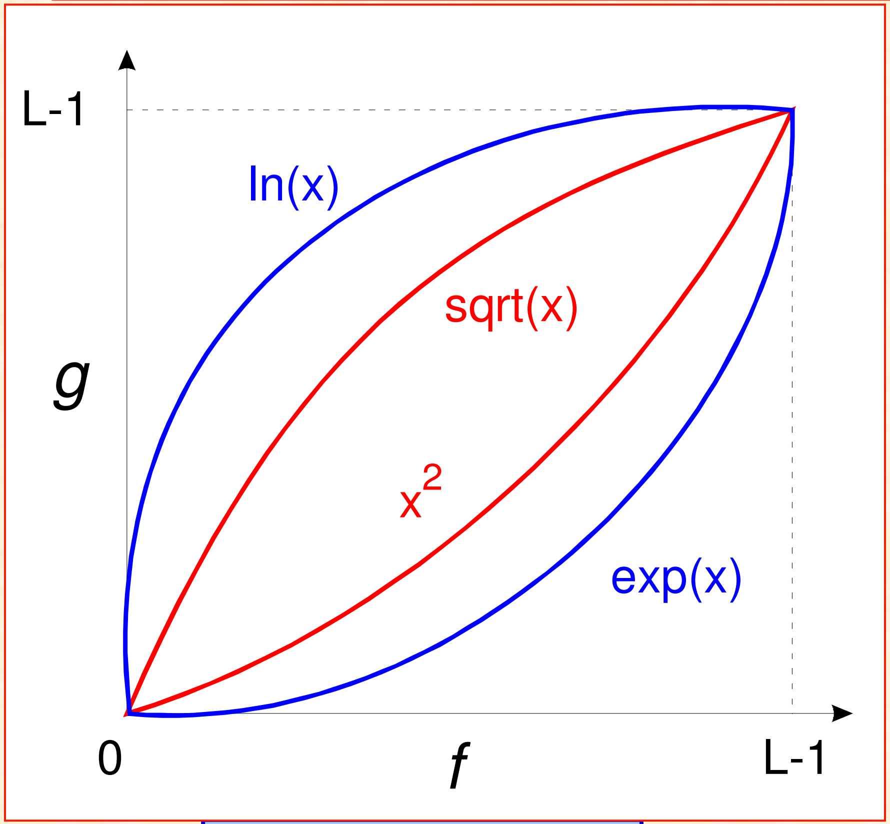


Grayscale inversion



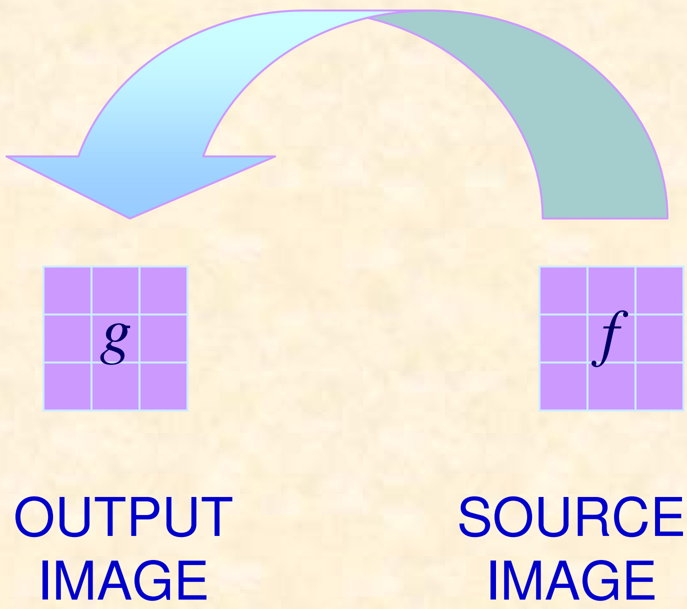
We can use look-up table to implement image point operations

Nonlinear grayscale transformation



γ correction

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

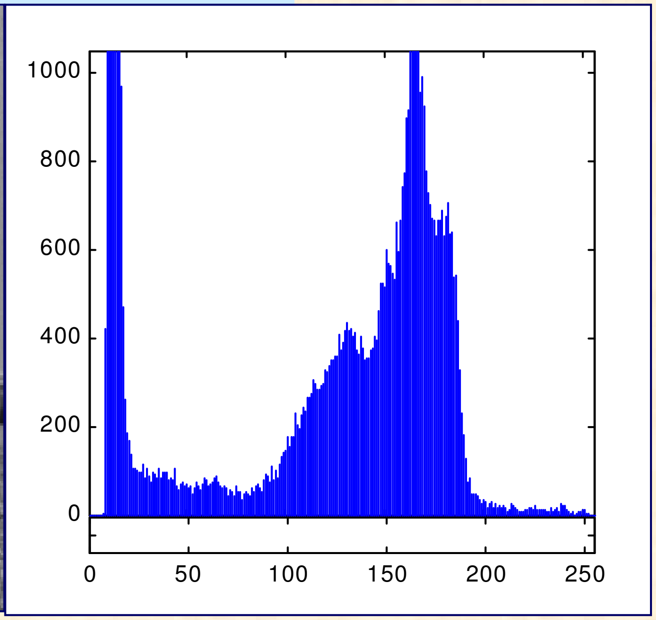


Grayscale normalization!

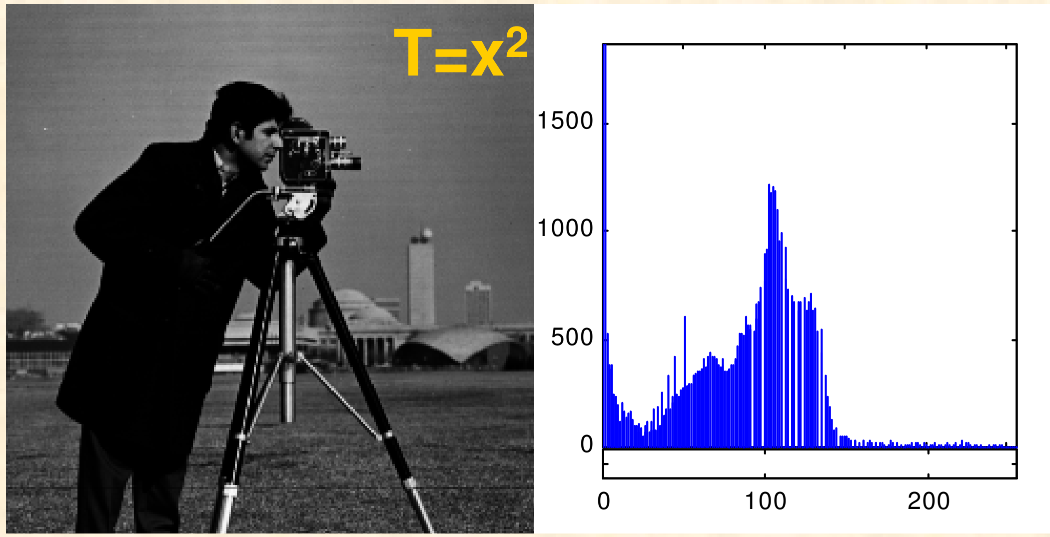
POINT OPERATION

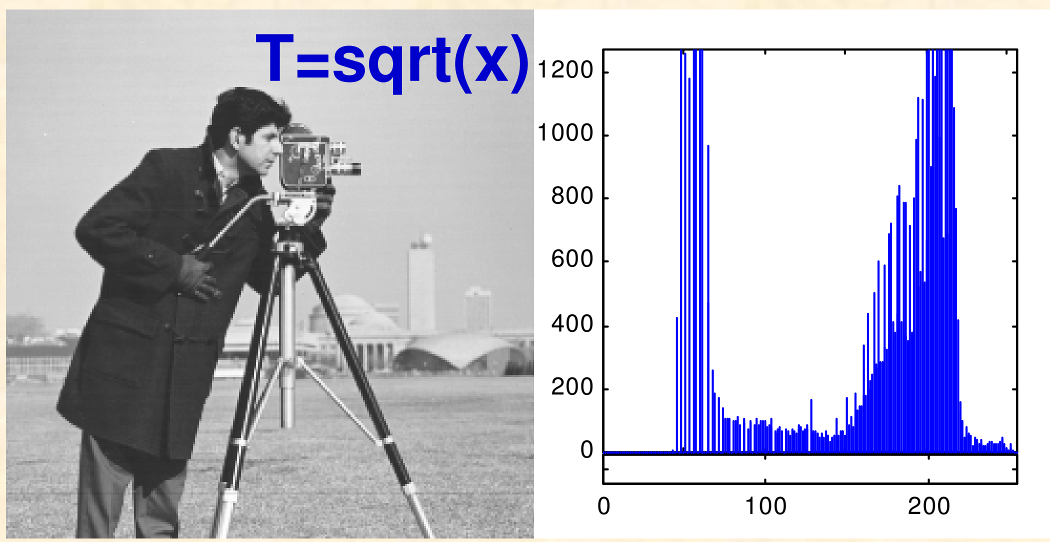
Nonlinear grayscale transformation - example

Source image

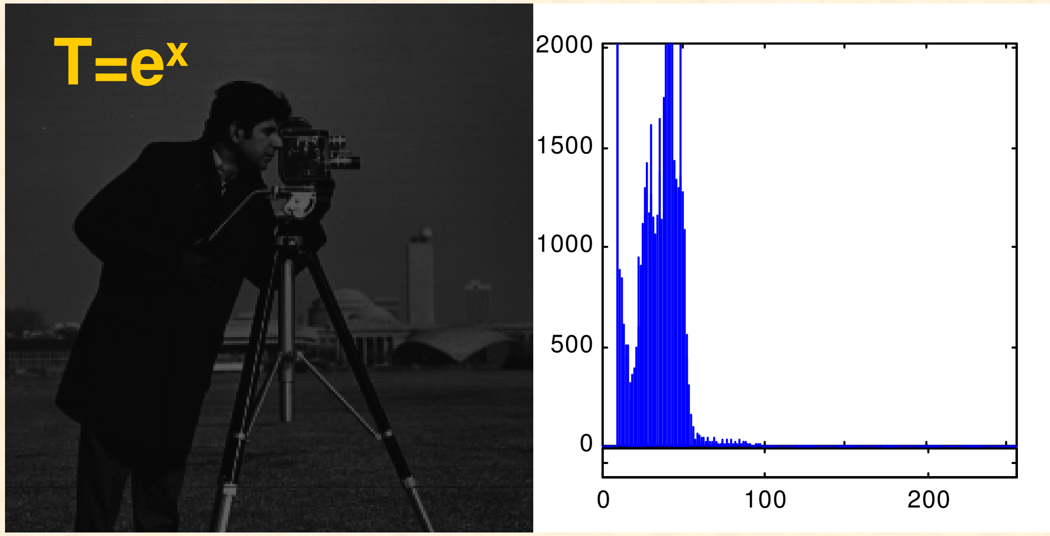


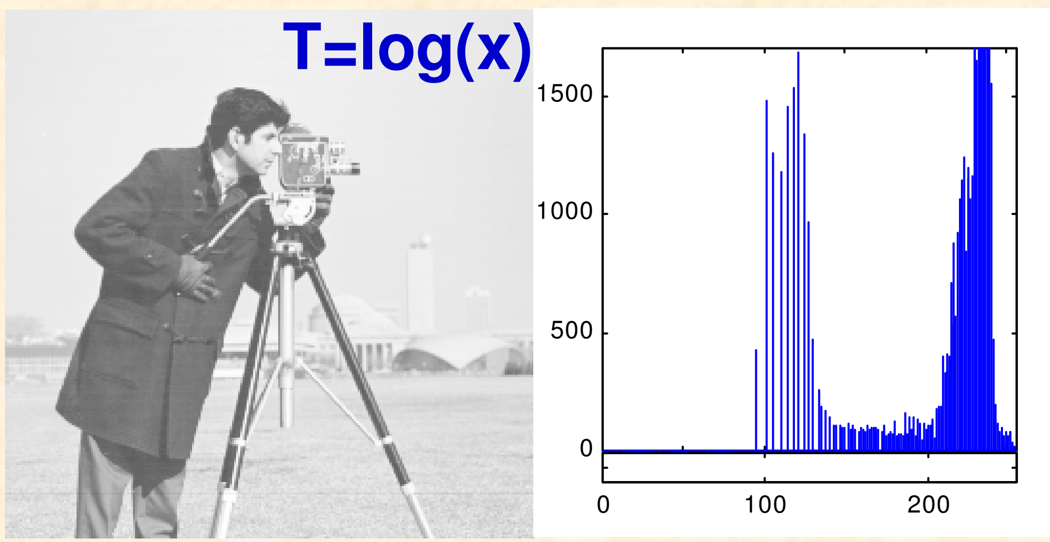
Nonlinear grayscale transformation - example





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

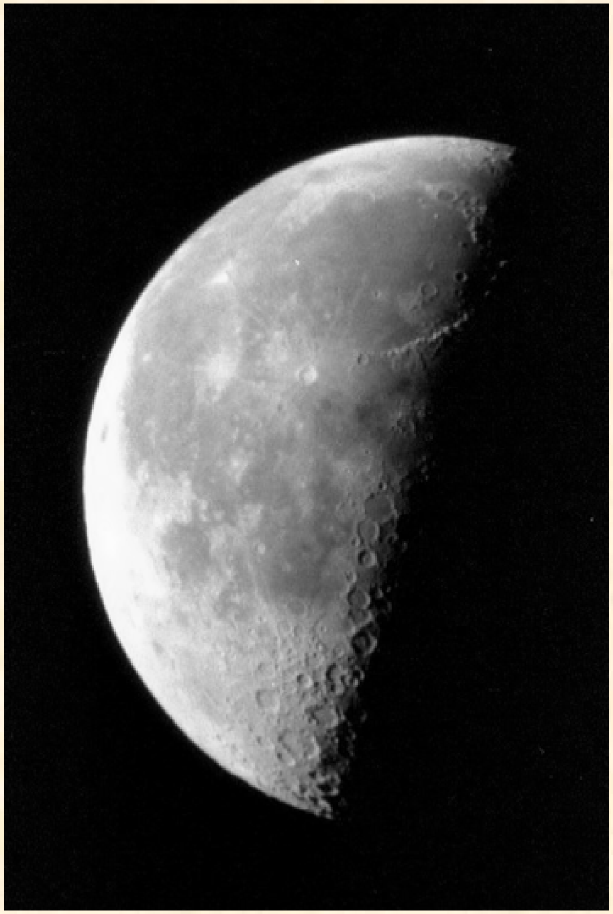
```
...
for i:=1 to M do for j:=1 to N do
        g[i,j]:=round(sqr(f[i,j])*norm);
...
```

Nonlinear grayscale transformation - algorithm

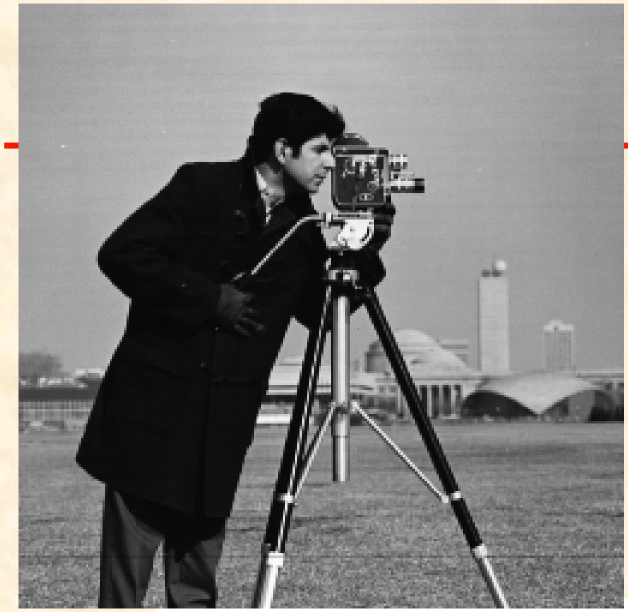
Example: square function (using look-up-table)

```
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])];
...
```

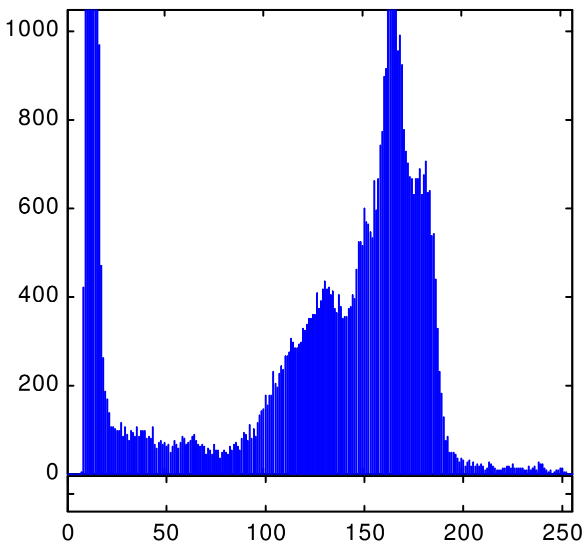
Enhacement of a telescope moon image

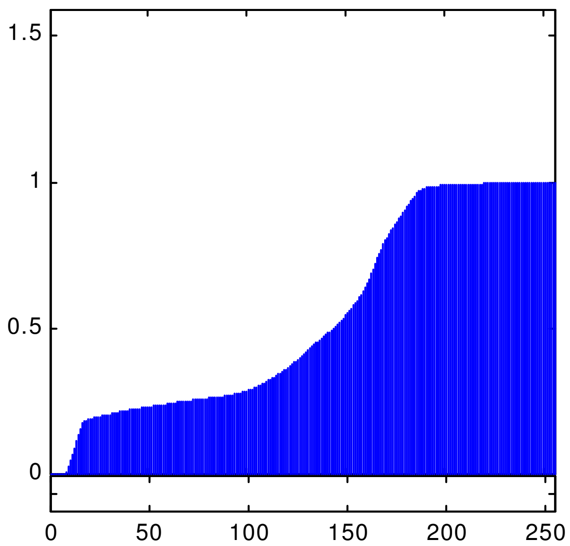


$T = b \log(ax)$



Cumulative histogram





Histogram **Cumulative histogram**

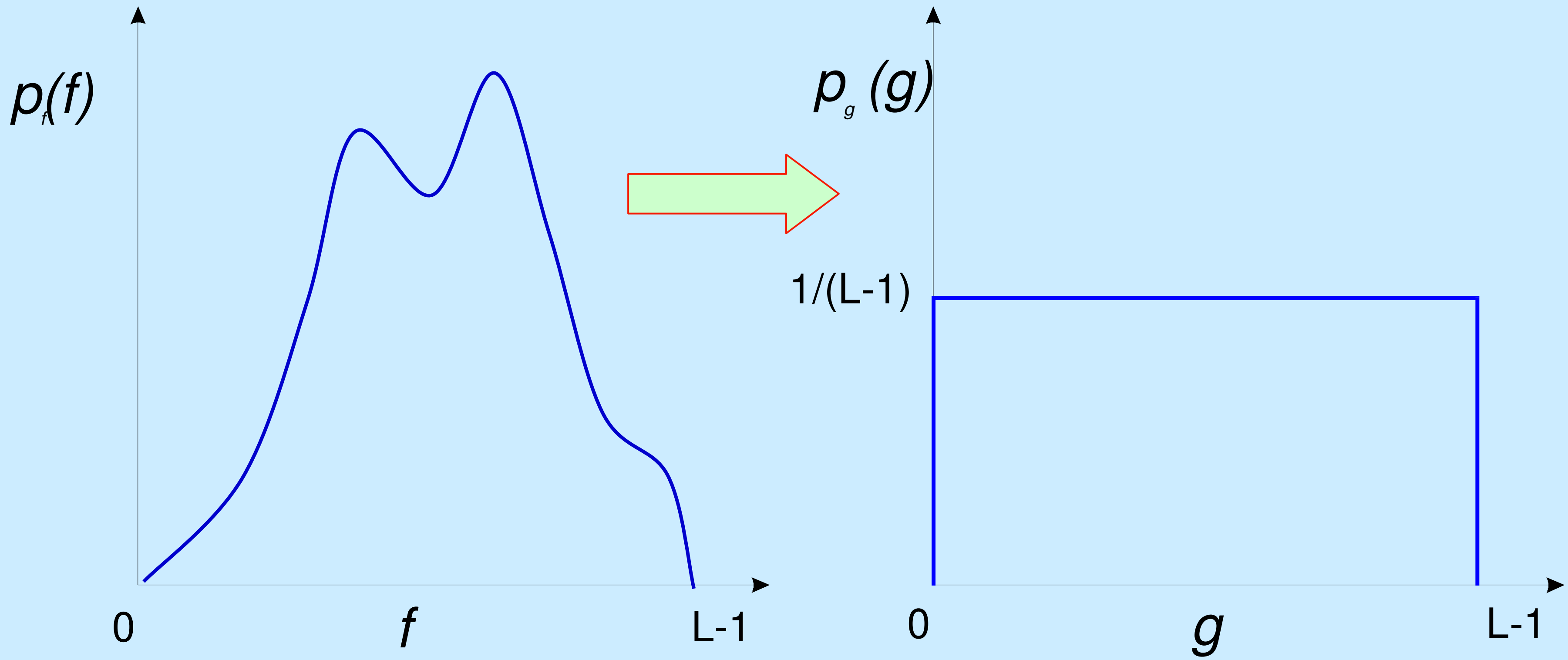
Histogram equalization

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

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

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

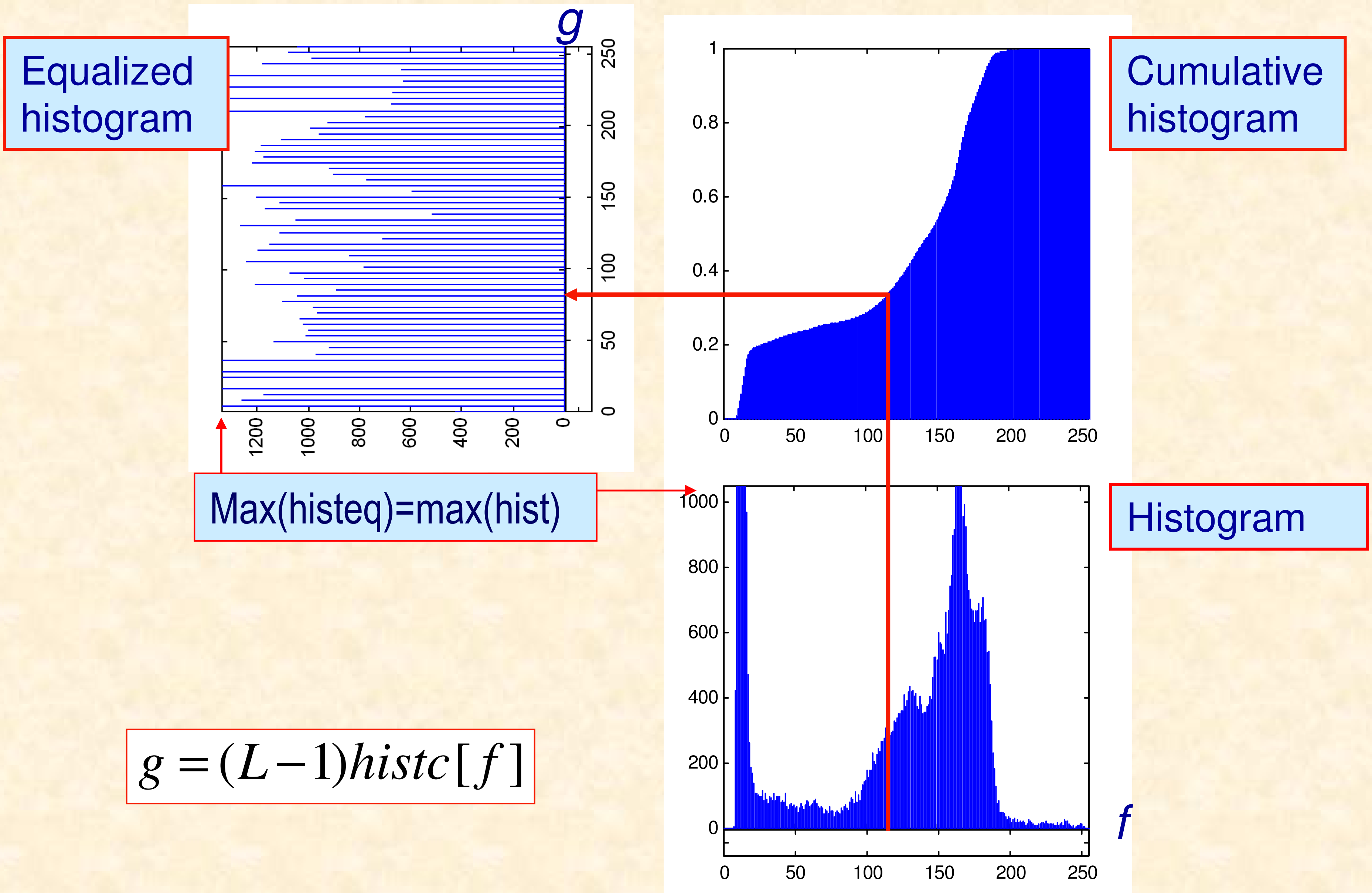
$$\sum_{i=0}^{f} p_f(i) = \frac{g}{L-1} \qquad f, g = 0, 1, 2, \ldots, L\text{-}1$$

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

Cumulative histogram - algorithm

```
%hist   – given image histogram
%hists – computed cumulative histogram
% M,N – number of image rows and columns
% L – numer of gray levels, eg. L=256

hists=zeros(1,256);
hist(1)=hists(1);
for i=2:256,
   hc(i)=hc(i-1)+h(i);
end;
hists=hists/(M*N);
```

Histogram equalization

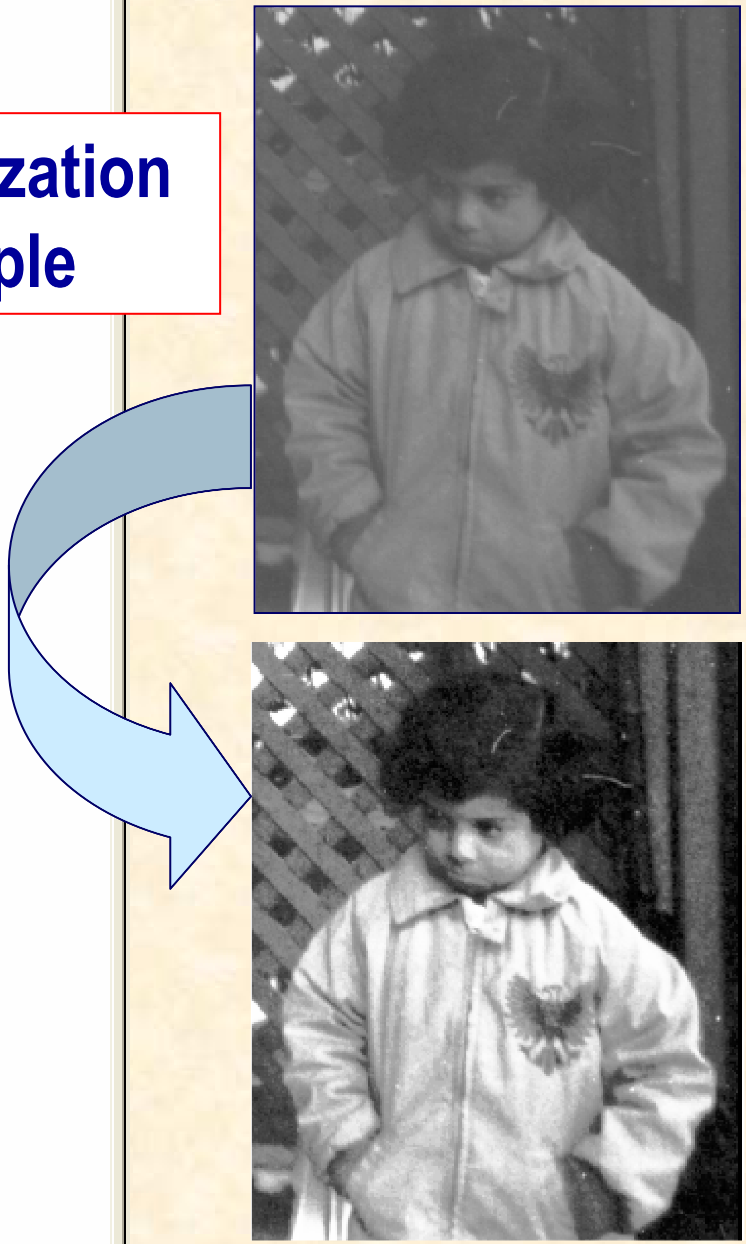


$$g = (L-1)histc[f]$$

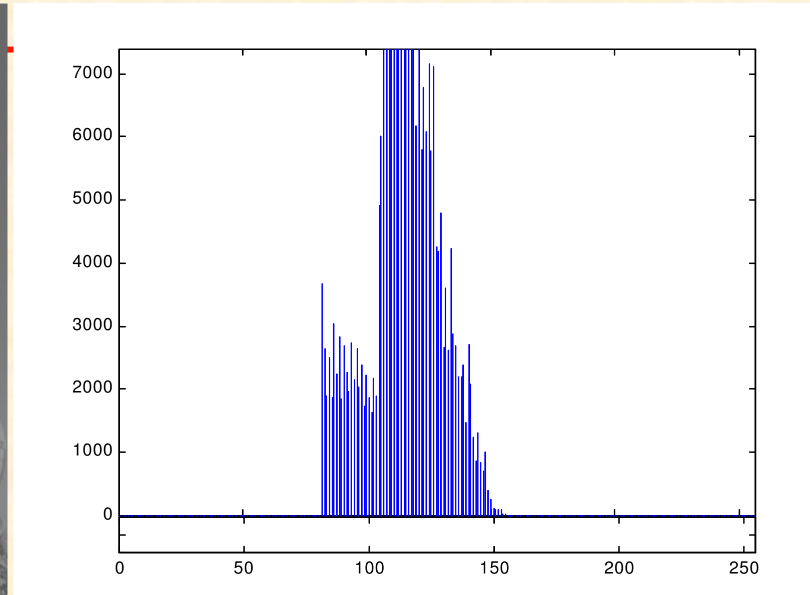
Histogram equalization algorithm - example

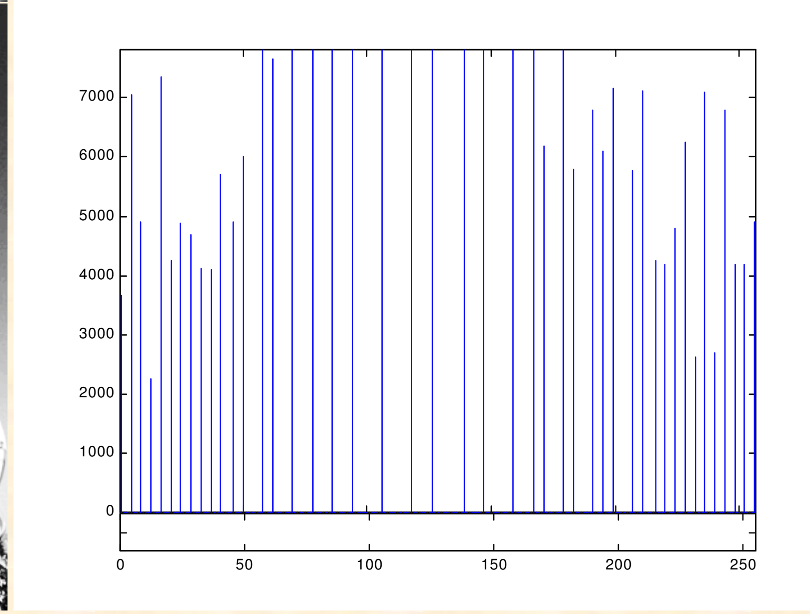
```
clear all; close all;
h=zeros(1,256); %histogram
he=zeros(1,256); %equalized histogram

x=imread('pout.tif');
imshow(x);
y=imcrop; y=double(y);
[W,K]=size(y);

%computation of the histogram
for w=1:W,
   for k=1:K,
      h(y(w,k)+1)=h(y(w,k)+1)+1;
   end;
end;
figure,bar(h);
pause;

%computation of the cumulative histogram
hcum=cumsum(h);
hcumn=round(hcum/(W*K)*255);
figure, bar(hcumn);

%gray-scale transformation of images
for w=1:W,
   for k=1:K,
      z(w,k)=hcumn(y(w,k)+1);
   end;
end;
figure,imshow(z,[])
pause;

%computation of the equalized histogram
for w=1:W,
   for k=1:K,
      he(z(w,k)+1)=he(z(w,k)+1)+1;
   end;
end;
figure,bar(he);
```

hist_eq_eng....



Histogram equalization - example





MATLAB Demo – intensity adjustment

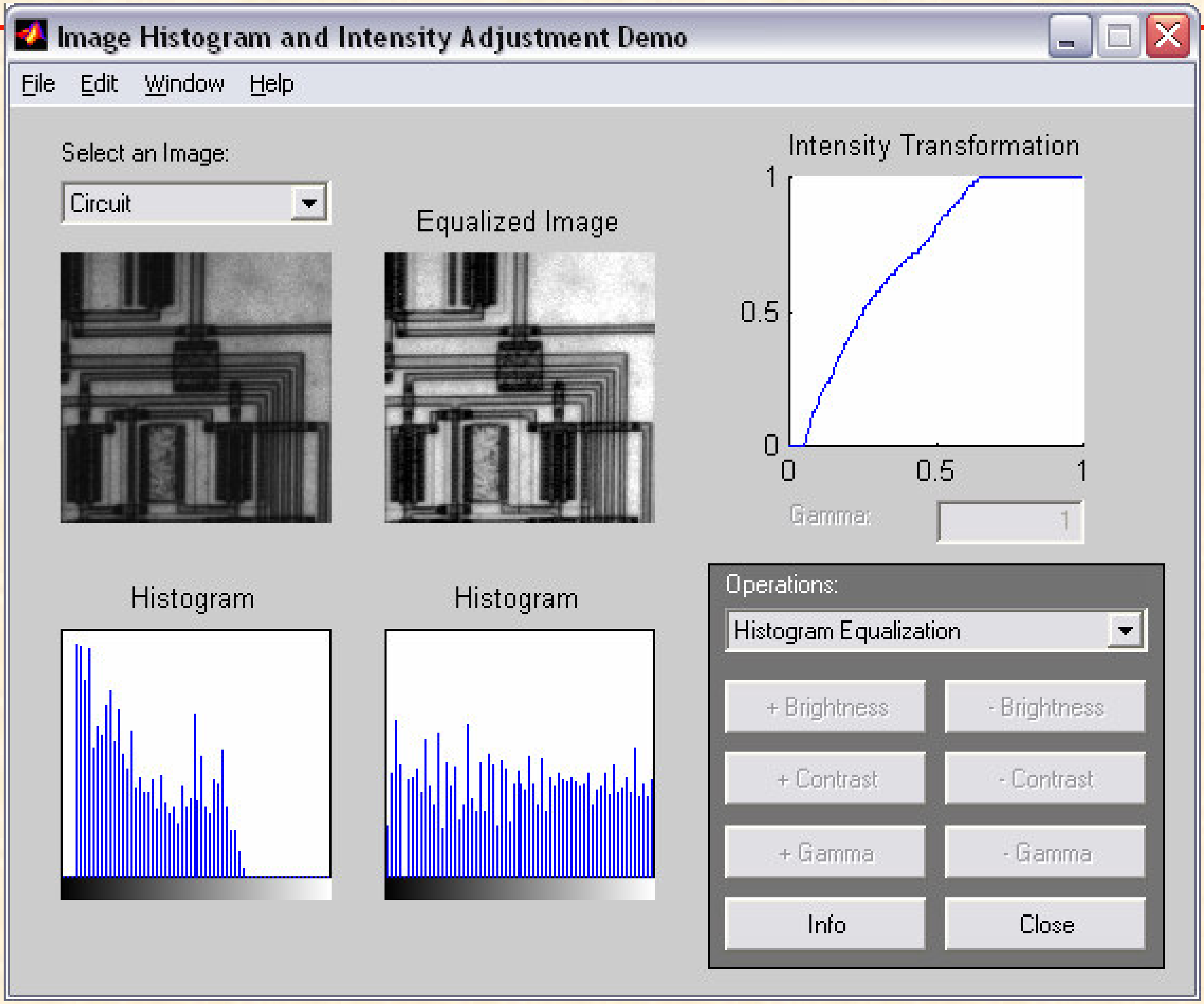


Image ehancement by image averaging

Consider a noisy image:

$$g(i,j) = f(i,j) + n(i,j)$$

contaminated by additive noise $n(i,j)$ of zero average and variance σ_n^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:

$$\overline{\sigma_n^2} = \frac{\sigma_n^2}{N}$$

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

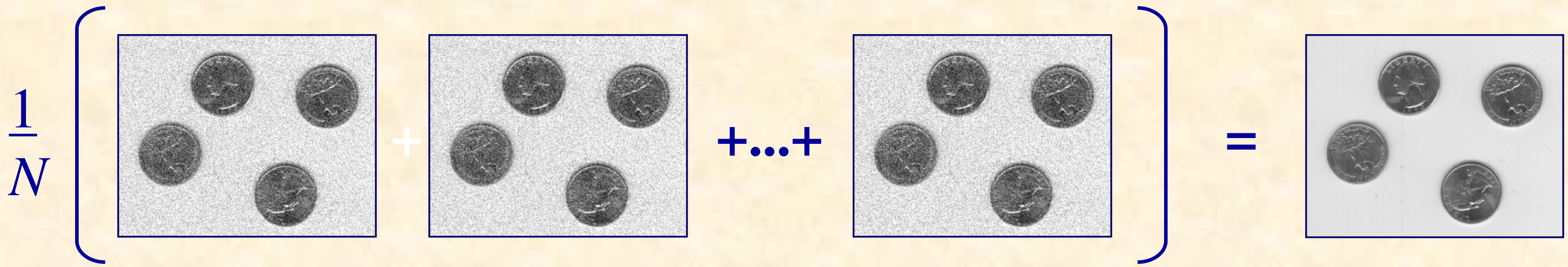


Image ehancement by image averaging

Noise variance in the averaged image:

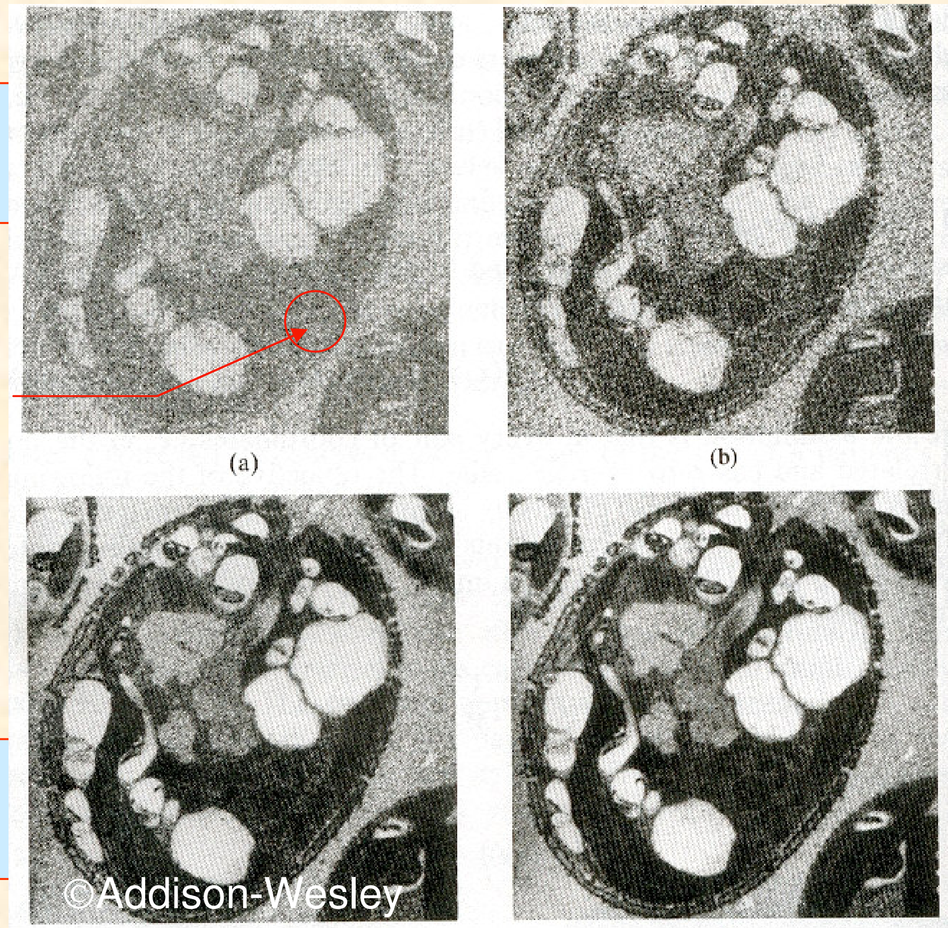
$$\overline{\sigma_n^2} = E\left\{\left(\frac{1}{N}\sum_{k=1}^{N} n_k\right)^2\right\} = \frac{1}{N^2}\cdot E\left\{\left(\sum_{k=1}^{N} n_k\right)^2\right\} =$$

$$= \frac{1}{N^2}\cdot E\left\{(n_1 + n_2 + \ldots + n_N)^2\right\} = \frac{1}{N^2}\cdot E\left\{\sum_{k=1}^{N} n_k^2 + 2\underbrace{\left(\sum_{k\neq p}^{N} n_k n_p\right)}_{=0}\right\} =$$

$$= \frac{1}{N^2} E\left\{\sum_{k=1}^{N} n_k^2\right\} = \frac{1}{N^2} N\sigma_n^2 = \frac{1}{N}\sigma_n^2$$

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



Microscope image of a cell