Histograms, Profiles, Projections

Image Processing

Histograms

Pixel

- Pixel (picture cell) – is the smallest controllable element of a picture represented on the screen.
- Pixel parameters:
    - a pair of integer values (x, y) describes the pixel geometric position in the image plane,
    - I value characterizes pixel brightness (intensity) at a point on the plane.

Definitions

- ***Histogram*** is the occurrence frequency distribution of the same brightness pixels in the image.
- ***Brightness*** is the average signal intensity.
- ***Contrast*** is the values interval between the minimum and maximum image brightness.

Histogram

- For an 8-bit grayscale image the **histogram** is a one-dimensional integer array *hist* of 256 elements [0 … 255].
- The histogram element *hist[i]* is the image pixels sum with the brightness (intensity) *i*.
- For a RGB color image it is need to create three histograms for each color.

Histogram Equalization

- If the histogram is uneven for the image improving it can be equalized.
- Histogram equalization depending on the problem being solved can be performed in different ways.

Image

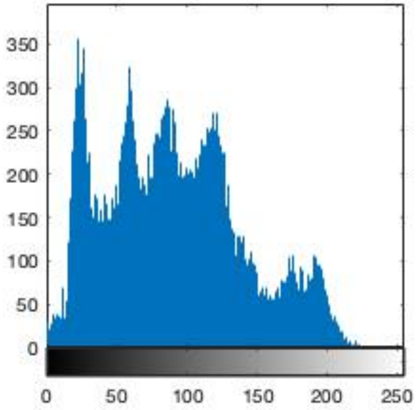
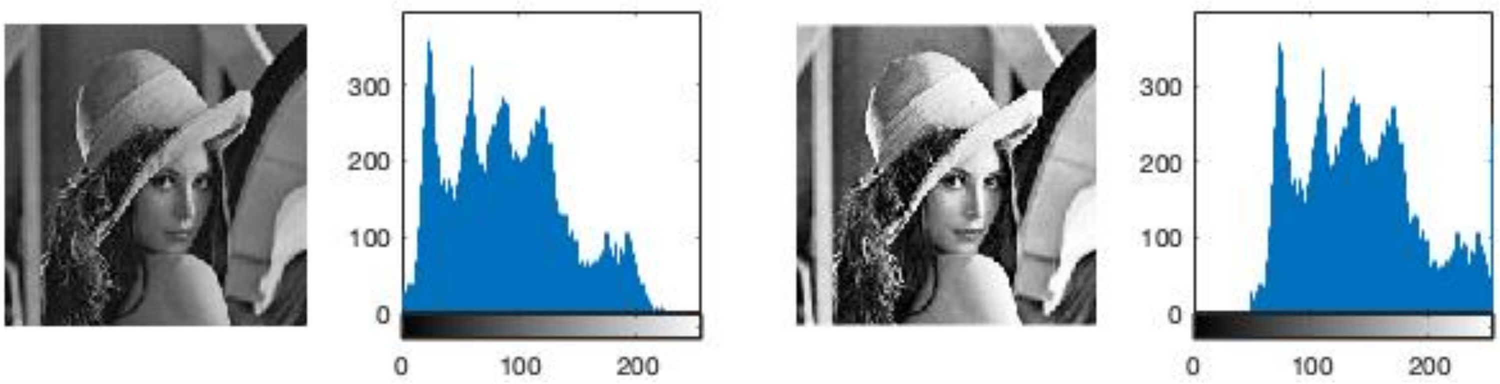


Image Histogram

Histograms: Arithmetic Operations

- If most of the histogram values are on the left, then the image is dark.
- To increase the detail in dark areas, it is need to shift the histogram to the right lighter area,
    - for example, by 50 gradations for each color:



Histograms: Linear Equalization

- Calculate the histogram H of the original image $f(x, y)$. Calculate the number of pixels N.
- Normalize the H array, so that the sum of all elements becomes equal to the maximum intensity value $L = 255$:

$$H(j) = \frac{L}{N} H(j)$$

- Calculate the cumulative histogram summing the intensities distribution from 0 to i:

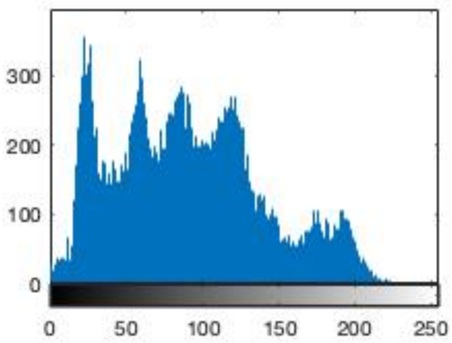
$$\mathrm{Sum}(i) = \sum_{0}^{i} H(j)$$

- Calculate the new intensity values of each pixel with coordinates (x, y) using the formula:

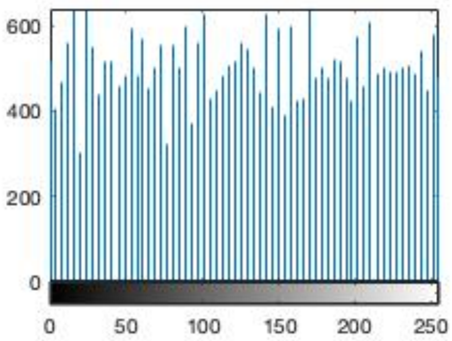
$$g(x, y) = \mathrm{Sum}\big(f(x, y)\big)$$

Histograms: Linear Equalization

Original image



Equalized image



Histograms: Dynamic Range Stretching

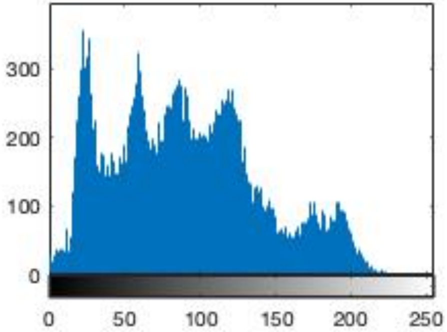
- If most of pixel intensities are in a narrow dynamic range, we can stretch it.
- This transformation is performed according to the following expression:

$$I_{new} = \left(\frac{I - I_{min}}{I_{max} - I_{min}}\right)^{\alpha},$$

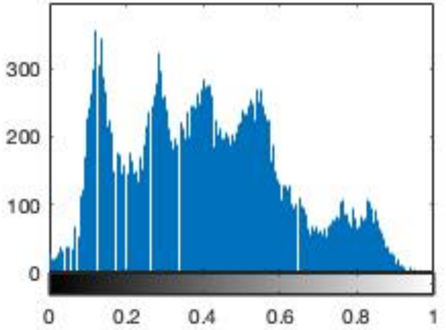
- where:
  - I and I_{new} are intensity values arrays of the original and new images correspondingly;
  - I_{min} and I_{max} are minimum and maximum intensity values arrays of the original image correspondingly;
  - α is a nonlinearity coefficient.

Histograms: Dynamic Range Stretching

Original image



Stretched image



Histograms: Uniform Transformation

- This transformation is carried out according to the following formula:
$$I_{new} = (I_{max} - I_{min}) \cdot P(I) + I_{min},$$
- where:
  - I and I_{new} are intensity values arrays of the original and new images correspondingly;
  - I_{min} and I_{max} are minimum and maximum intensity values arrays of the original image correspondingly;
  - $P(I)$ is original image probability distribution function which is approximated by *cumulative histogram*.

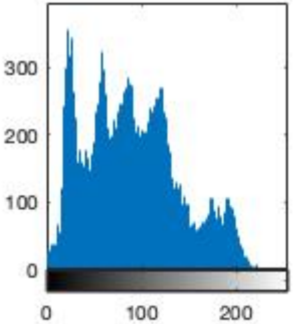
Histograms: Uniform Transformation

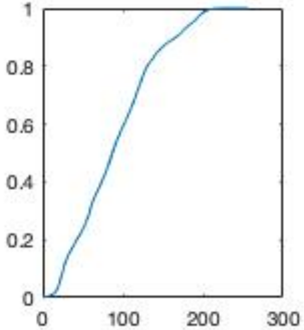
- **Cumulative histogram** is a histogram in which the vertical axis gives not just the counts for a single bin, but rather gives the counts for that **bin plus all bins for smaller values** of the response variable:

$$P(I) \approx \sum_{m=0}^{i} \mathrm{Hist}(m)$$

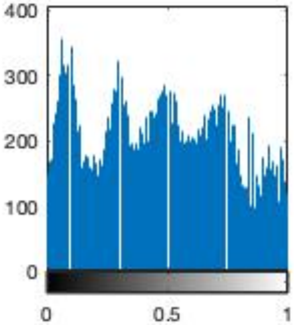
Histograms: Uniform Transformation

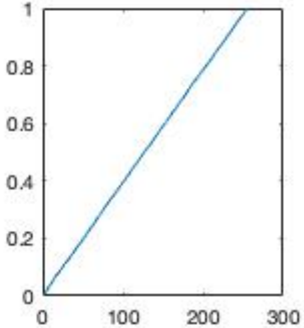
Original image





Transformed image





Histograms: Exponential Transformation

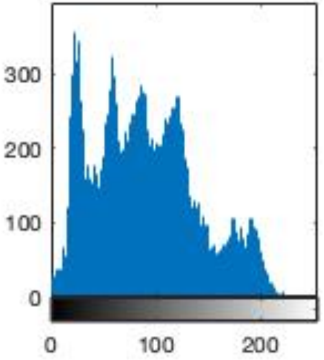
- This transformation is carried out according to the following formula:

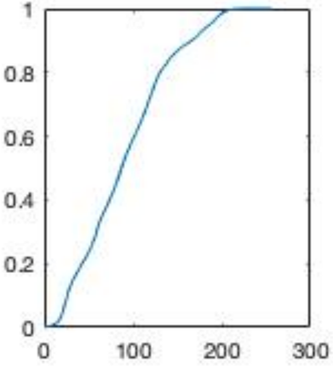
$$I_{new} = I_{min} - \frac{1}{\alpha} \cdot \ln\big(1 - P(I)\big),$$

- where:
  - I and I_{new} are intensity values arrays of the original and new images correspondingly;
  - I_{min} is a minimum intensity value array of the original image;
  - $P(I)$ is an original image probability distribution function which is approximated by *cumulative histogram*;
  - α is a constant characterizes transformation slope.

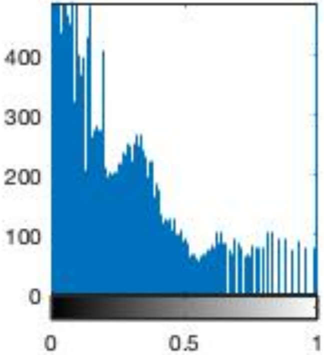
Histograms: Exponential Transformation

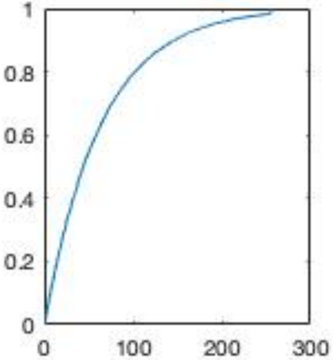
Original image





Transformed image





Histograms: Rayleigh Transformation

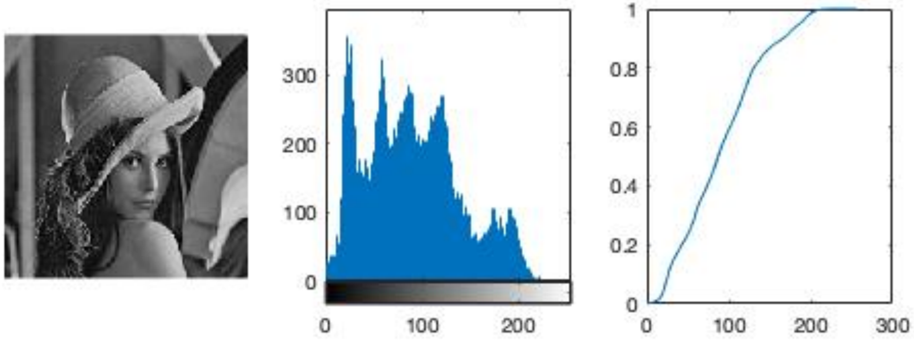
- This transformation is carried out according to the following formula:

$$I_{new} = I_{min} + \left(2\alpha^2 \cdot \ln\left(\frac{1}{1 - P(I)}\right)\right)^{1/2},$$

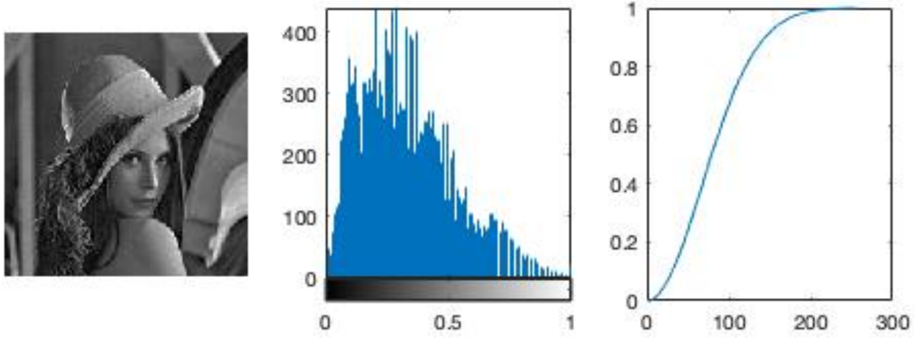
- where:
  - I and I_{new} are intensity values arrays of the original and new images correspondingly;
  - I_{min} is a minimum intensity value array of the original image;
  - $P(I)$ is an original image probability distribution function which is approximated by *cumulative histogram*;
  - α is a constant characterizing the histogram of the resulting image elements intensity distribution.

Histograms: Rayleigh Transformation

Original image



Transformed image



Histograms: Transformation of 2/3-degree

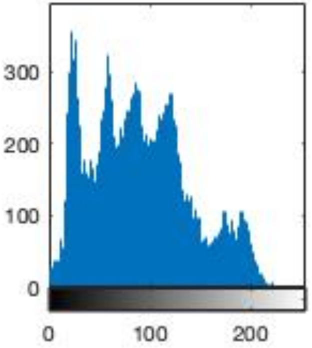
- This transformation is carried out according to the following formula:

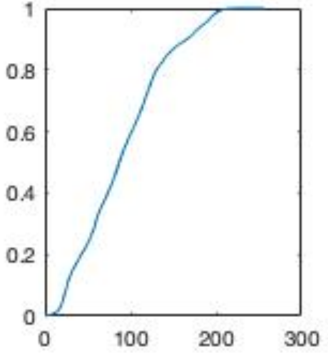
$$I_{new} = P(I)^{2/3},$$

- where:
  - $P(I)$ is an original image probability distribution function which is approximated by *cumulative histogram*.

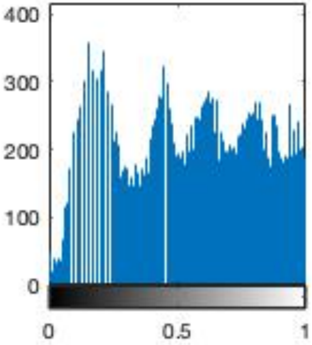
Histograms: Transformation of 2/3-degree

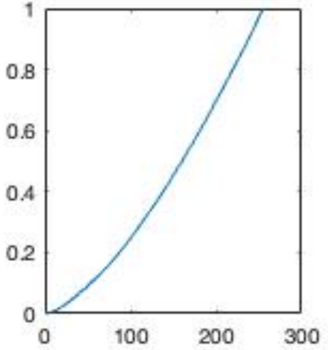
Original image





Transformed image





Histograms: Hyperbolic Transformation

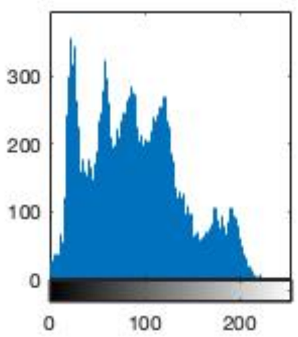
- This transformation is carried out according to the following formula:

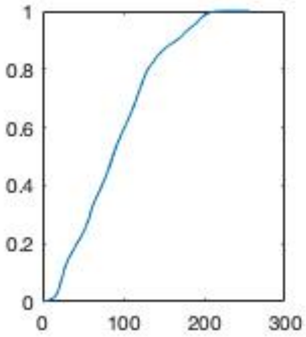
$$I_{new} = \alpha^{P(I)},$$

- where:
  - α is a constant relative to which the transformation is carried out. As a rule, this constant is equal to the original image elements intensity minimum value $\alpha = I_{min}$.

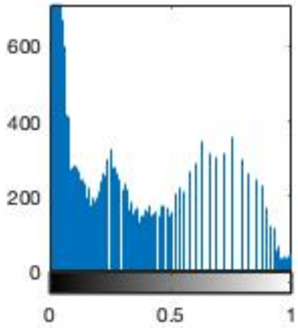
Histograms: Hyperbolic Transformation

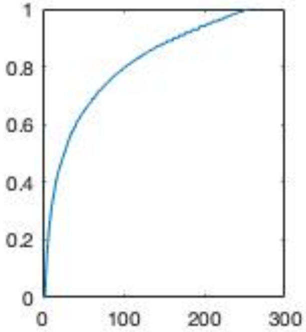
Original image





Transformed image





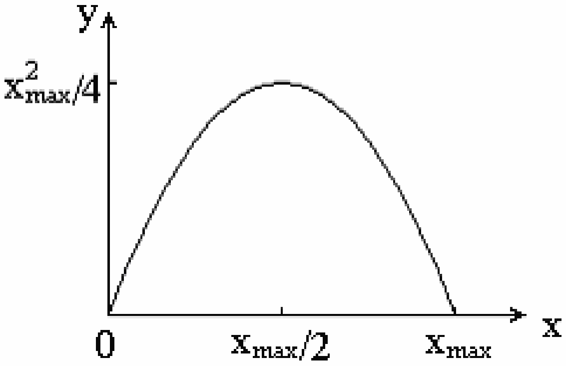
Histograms: Solarization

- This transformation is carried out according to the following formula:

$$y = kI(I_{max} - I)/I_{max},$$

- where:
  - I_{max} – maximum intensity value (typically is equal to 255),
  - k – is a constant that controls the dynamic range,
  - Could be $k = 1/64$ or $k = 4$, for example.

Histograms: Solarization



Transformation function

Histograms: Arbitrary Nonlinear Transformation

- Look-Up-Table (LUT):

$$y = f(x).$$

x	x_1	x_2	…	x_{n-1}	x_n
$y = f(x)$	y_1	y_2	…	y_{n-1}	y_n

- x – original intensity, used as an index in the table;
- y – the new intensity value.

Histograms: Arbitrary Nonlinear Transformation

- Look-Up-Table (LUT):
  - Example: **solarization** according to the formula:
  $$y = 4x(255 - x)/255.$$
  - LUT is described by a one-dimensional array:
  [0,4,8,12,16,20,23,27,…, 198,199,201,203,…, 254,255,255,255,…,12,8,4,0].
  - The new intensity values are calculated 256 times **only**.

Histograms: Arbitrary Nonlinear Transformation

- For image X calculate the histogram h_x of the original image and its cumulative histogram H_x:

$$H_x[j] = \sum_{i=0}^{j} h_x[i].$$

- Determine the **desired histogram** h_z *(from some reference image or set as a function)* and its **cumulative histogram** H_z*:*

$$H_z[j] = \sum_{i=0}^{j} h_z[i].$$

Histograms: Arbitrary Nonlinear Transformation

- Build a Look-Up-Table (LUT):

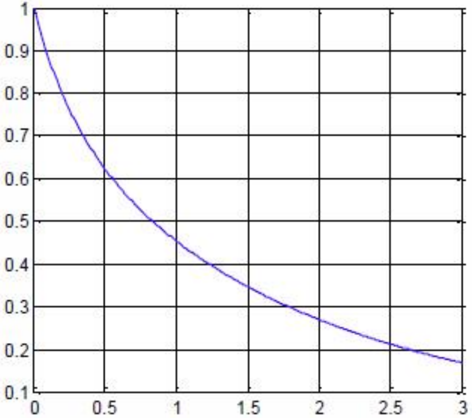
```
for(i = 0; i<= 255; i++) {
    j = i;
    if(Hx[i] ≤ Hz[j])
        LUT[i] = j;
    else
        j = j + 1;
    if(Hx[i]-Hz[j]> Hx[i]-Hz[j-1])
        j = j - 1;
    else
        LUT[i] = j;}
```

Histograms: average intensity to the given value

- Set the required value for the average intensity *K*.
- Calculate the minimum I_{min}, maximum I_{max} and arithmetic mean I_{av} of the original image intensity values.
- Choose parameters for intensity values transformation such that the image pixels average intensity becomes *K*.

Histograms: average intensity to the given value

$I_{av} = 119$



$K = f(\alpha)$

$I_{av1} = 177$

$I_{av2} = 77$

Profiles

Profile

- To reduce the image geometric components to a one−dimensional data array $n = 1$ are used such characteristics as image “profiles” and “projections”.
- *Profile* along the line is an intensity function of the image, distributed along this line.

Profile

- The simplest case of image profile is a *row profile*:

$$\text{Profile } i(x) = I(x, i),$$

where i – is the image row number.

- *Column profile* is:

$$\text{Profile } j(y) = I(y, j),$$

where j – is the image column number.

Profile

- In MATLAB computed by the `improfile()` function:

Projections

Projection

- *Image Projection* onto a certain axis is the image pixels intensities sum in the direction perpendicular to this axis.
- The simplest case of two-dimensional image projection is the *vertical projection* on the axis *Ox*,
  - which is the image pixel intensities sum *by columns*:

$$\mathrm{Proj}X(y) = \sum_{y=0}^{\dim Y - 1} I(x, y)\,.$$

Projection

- Similarly, you can calculate the *horizontal projection* on the Oy axis,
  - which is the image pixel intensities sum *by rows*:

$$\mathrm{Proj}Y(x) = \sum_{x=0}^{\dim X - 1} I(x, y)\,.$$

Projection

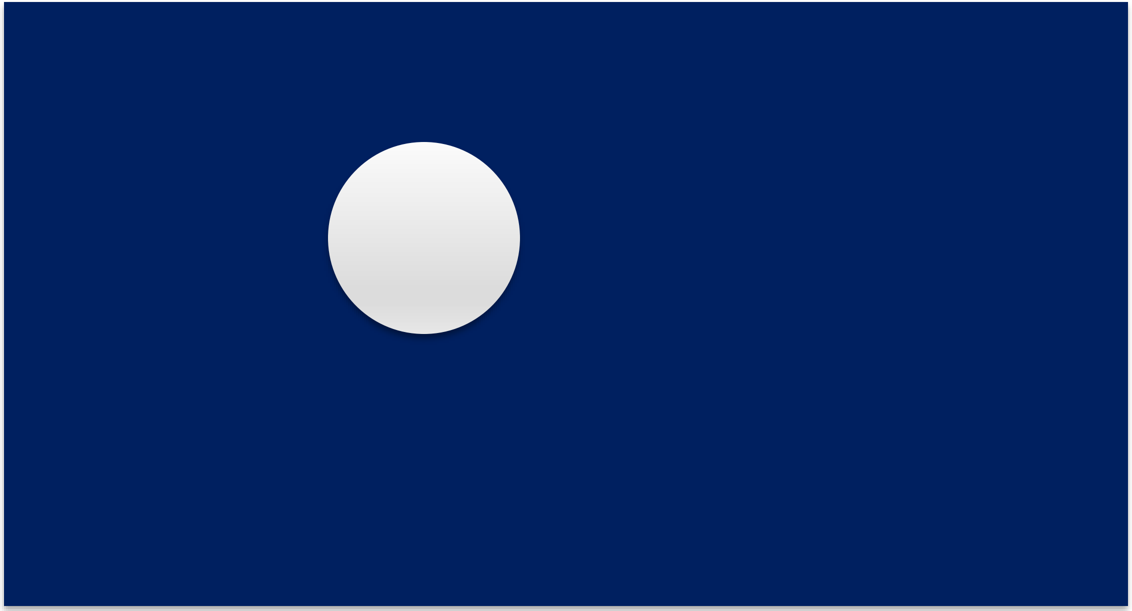
- Let suppose that the arbitrary *Oe* axis direction is given by a unit vector with coordinates (e_x, e_y).
- In this case the image projection onto the arbitrary *Oe* axis is determined by the following expression:

$$\mathrm{Proj}E(t) = \sum_{xe_x + ye_y = t} I(x, y).$$

Activity Time

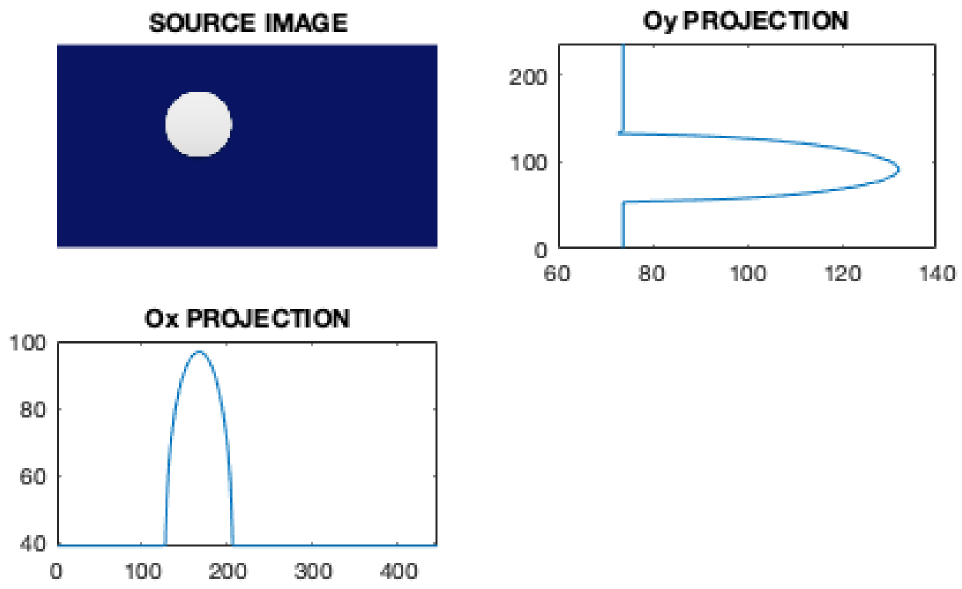
Activity Time

- How to calculate the monotone object coordinates on a uniform background?



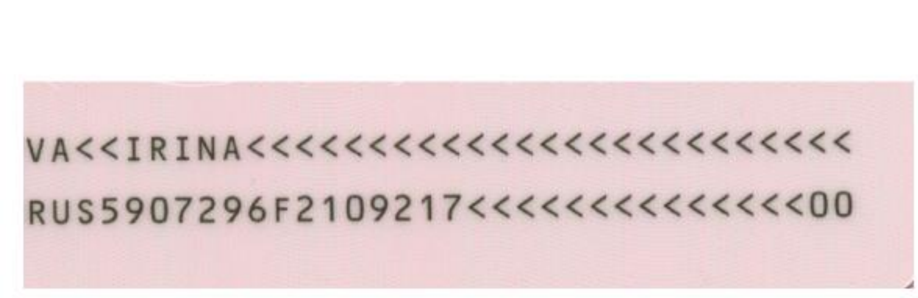
Activity Time

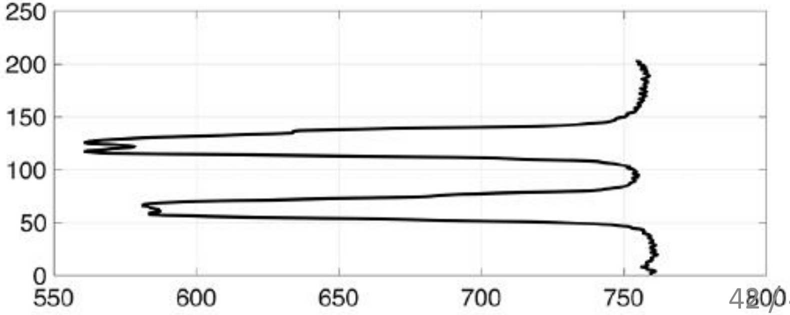
- Calculate projections!



Example

- On the figure is a projection example onto *Oy* axis of machine-readable document.
- Two machine-readable text lines generate two significant projection function extremums.
- Such projections can be used in algorithms for detection and segmentation of text strings in text recognition systems.





THANK YOU
FOR YOUR TIME!

s.shavetov@itmo.ru