

# Brightness and geometric transformations

Václav Hlaváč

Czech Technical University in Prague

Faculty of Electrical Engineering, Department of Cybernetics

Center for Machine Perception

<http://cmp.felk.cvut.cz/~hlavac>, [hlavac@fel.cvut.cz](mailto:hlavac@fel.cvut.cz)

## Outline of the talk:

- ◆ Brightness scale transformation.
- ◆ Brightness corrections.
- ◆ Geometric transformations.

# Image preprocessing, the intro

The input is an image, the output is an image too.

The image is not interpreted.

---

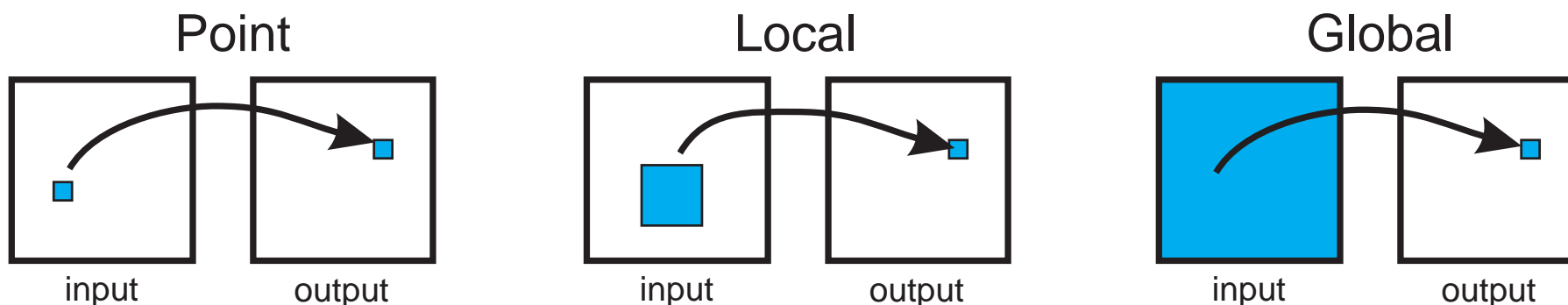
## The aim

- ◆ To suppress the **distortion** (e.g., correction of the geometric distortion caused by spherical shape of the Earth taken from a satellite).
- ◆ **Contrast** enhancement (which is useful only if the human looks at the image).
- ◆ **Noise** suppression.
- ◆ **Enhancement of some image features** needed for further image processing, e.g., edge finding.

# Taxonomy of the image preprocessing methods

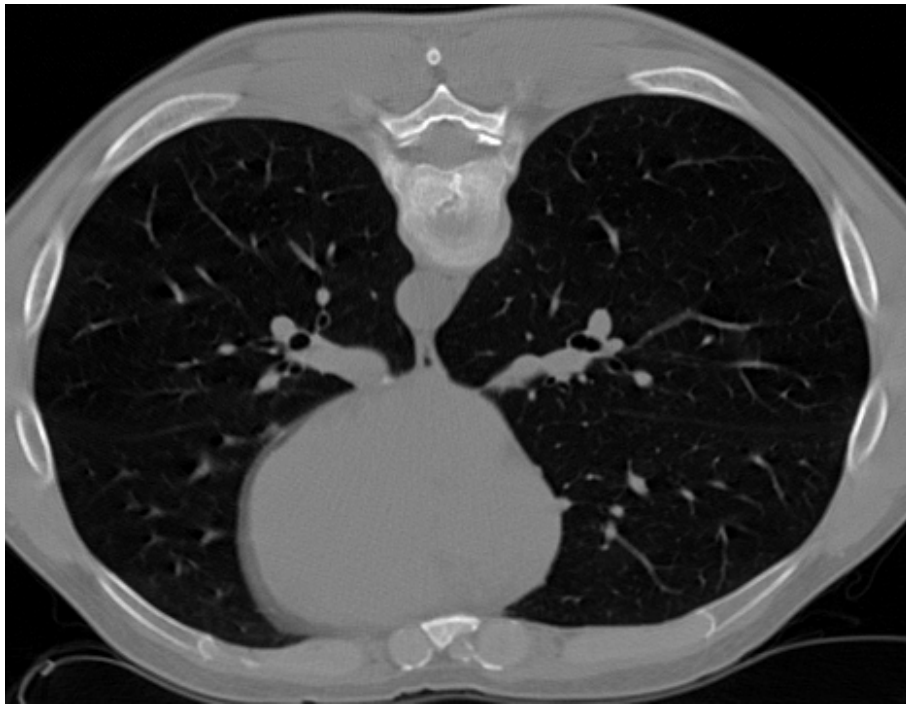
Taxonomy according to the neighborhood size of the current pixel.

<i>Neighborhood</i>	<i>Example of the operation</i>	<i>Actually processed neighborhood</i>
Single pixel	Brightness scale transform	The same for all pixels
	Brightness corrections	The current pixel
	Geometric transforms	A current pixel (theoretically), a small neighborhood (practically)
Local	Local filtration	A small neighborhood
Global	Fourier transform	The whole image
	Image restoration	

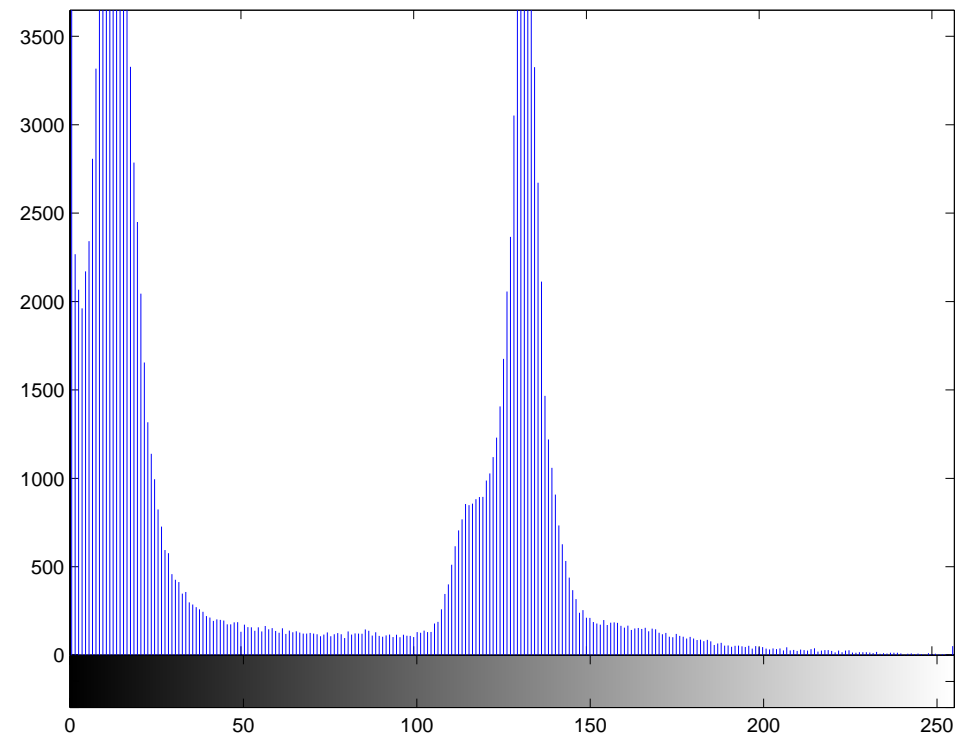


# Brightness histogram

The brightness **histogram** is the estimate of the **probability distribution** of the phenomenon that a pixel has a certain brightness.



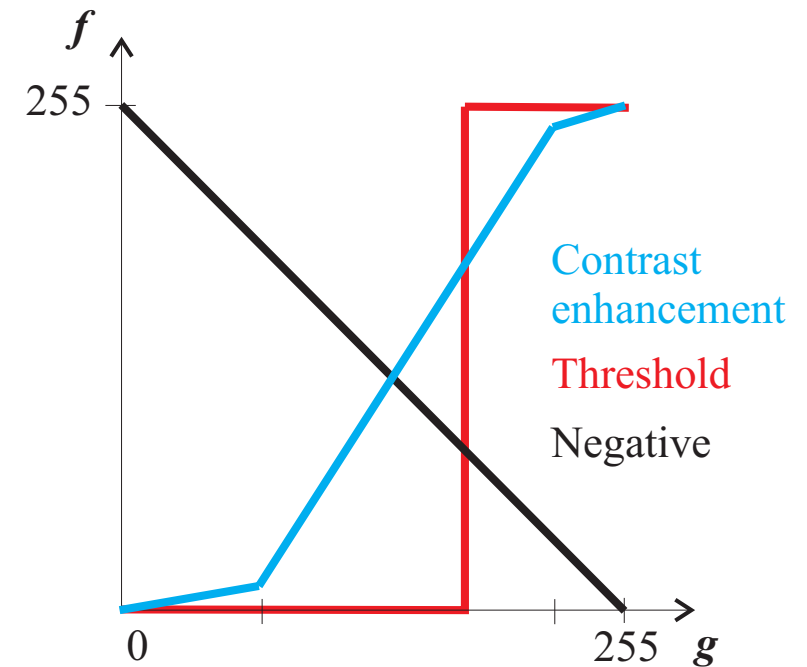
input image



brightness histogram

# Transformation of the gray scale

- ◆  $g(i)$  – input grey scale,
- ◆  $f(i)$  – output grey scale,
- ◆  $i = 0, \dots, i_{\max}$  (maximal brightness).

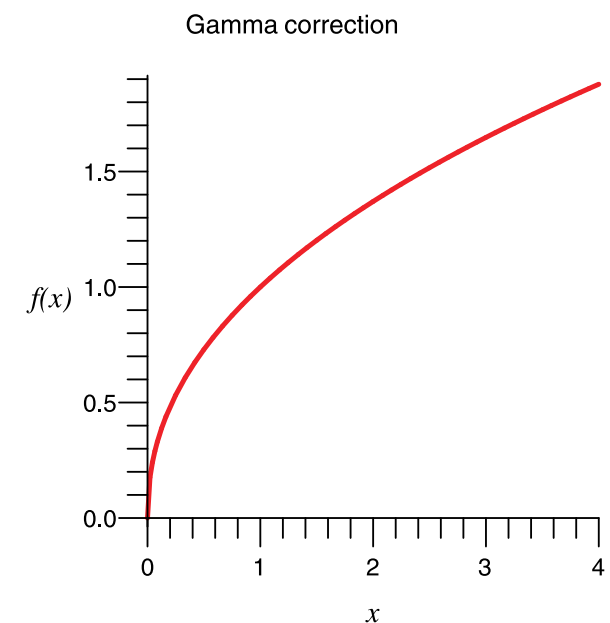
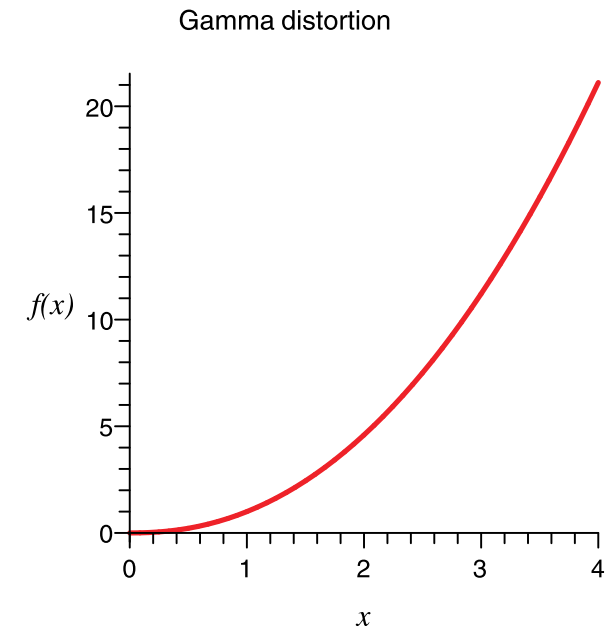


## Notes:

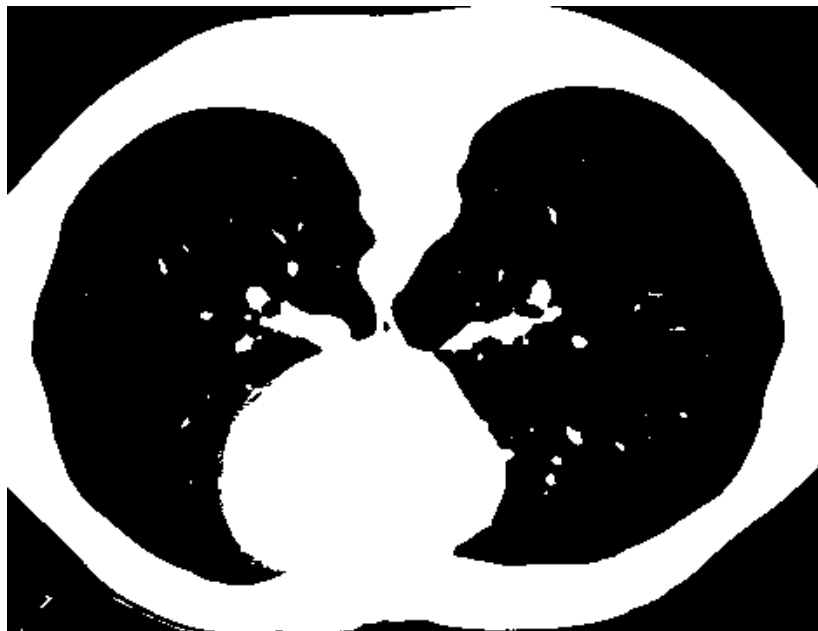
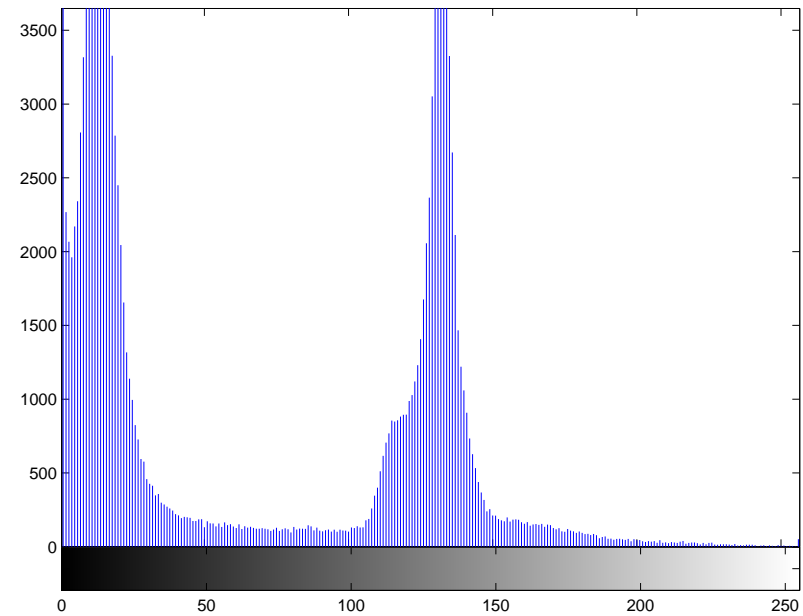
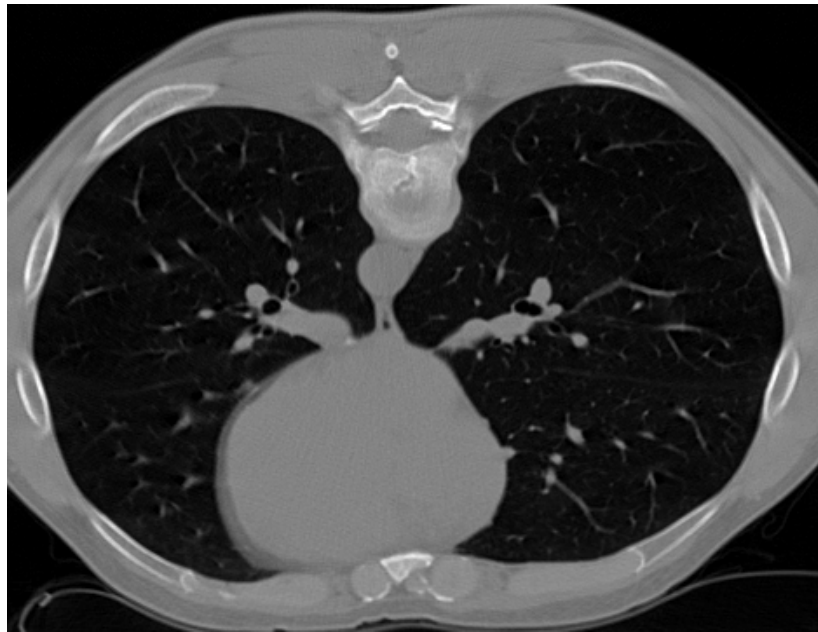
- ◆ Many cameras (and displays) transform light intensities  $g$  into values proportional to  $g^\gamma$ . Correction by  $g_{\max} \left( \frac{g}{g_{\max}} \right)^{\frac{1}{\gamma}}$ .
- ◆ Some transformations are implemented in HW, e.g., in the display card (e.g. in the VGA mode).
- ◆ Thresholding: converts a gray scale image into a binary image. It yields two regions. How to chose an appropriate threshold?

# Nonlinearity in intensity, $\gamma$ correction

- ◆ Cameras contain a module performing nonlinear brightness transformation, the  $\gamma$ -correction.
- ◆ In the cathode-ray tube (vacuum) displays, the brightness depends exponentially on the grid voltage  $U$ , i.e., brightness =  $U^\gamma$ , where usually  $\gamma = 2, 2$ .
- ◆ In order to preserve linear transfer characteristic in the whole transmission chain, is the effect compensated in cameras by the inverse function. Having input irradiance  $E$  and the input voltage of the camera  $U$ , the inverse function writes  $U = E^{(1/\gamma)}$ , usually  $U = E^{0,45}$ .
- ◆ Although the brightness of LCD monitors depends on the input voltage linearly, is the  $\gamma$ -correction still in use to secure backward compatibility.



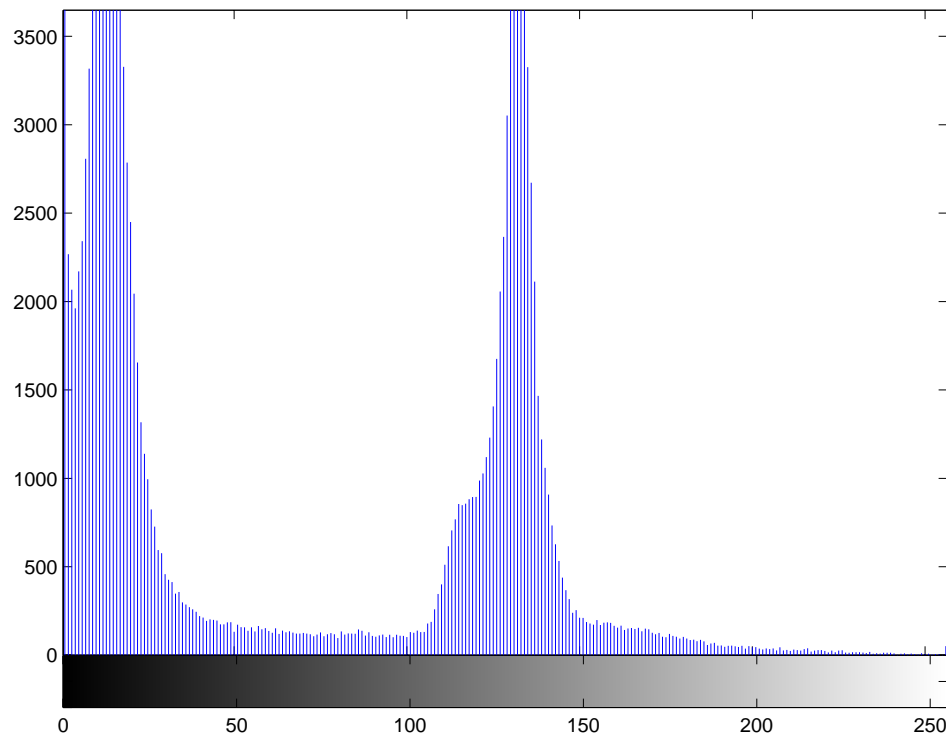
# Example of two thresholds



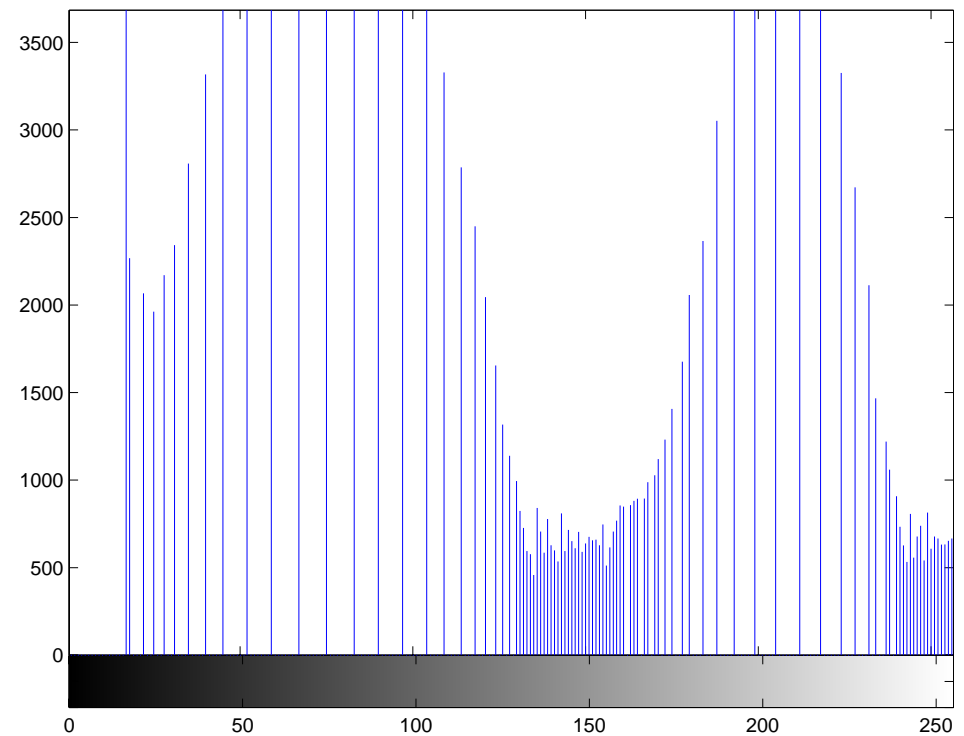
# Histogram equalization

The aim is:

- ◆ To enhance the contrast for a human observer by utilizing the gray scale fully,
- ◆ To normalize the brightness of the image for automatické brightness comparison.



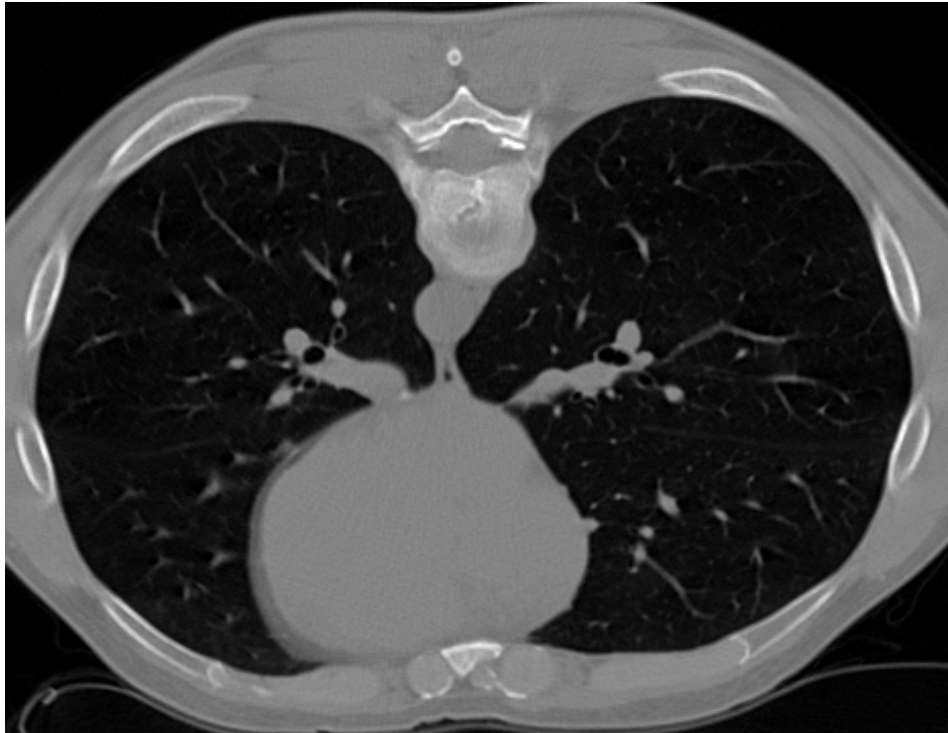
histogram of the input image



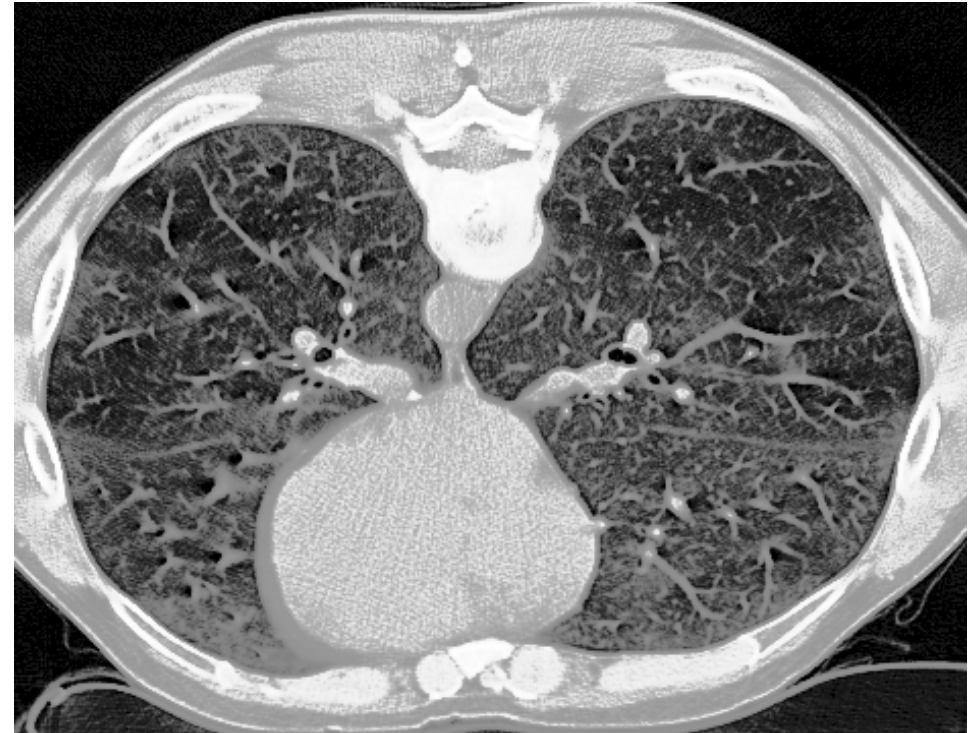
histogram after ekvalization



# Increased contrast after histogram equalization



original image



increased contrast

# Derivation of histogram equalization

**Input:** The histogram  $H(p)$  of the input image with the gray scale  $p = \langle p_0, p_k \rangle$ .

**Aim:** to find a monotonic transform of the gray scale  $q = \mathcal{T}(p)$ , for which the output histogram  $G(q)$  will be uniform for the entire input interval of brightnesses  $q = \langle q_0, q_k \rangle$ .

$$\sum_{i=0}^k G(q_i) = \sum_{i=0}^k H(p_i) .$$

The equalized histogram  $\approx$  the uniform distribution.

$$f = \frac{N^2}{q_k - q_0} .$$

## Derivation of histogram equalization (2)

The ideal continuous histogram is available only in the **continuous case**.

$$\int_{q_0}^q G(s) ds = \int_{p_0}^p H(s) ds .$$

$$N^2 \int_{q_0}^q \frac{1}{q_k - q_0} ds = \int_{p_0}^p H(s) ds .$$

$$\frac{N^2(q - q_0)}{q_k - q_0} = \int_{p_0}^p H(s) ds .$$

$$q = \mathcal{T}(p) = \frac{q_k - q_0}{N^2} \int_{p_0}^p H(s) ds + q_0 .$$

**Discrete case, cumulative histogram**

$$q = \mathcal{T}(p) = \frac{q_k - q_0}{N^2} \sum_{i=p_0}^p H(i) + q_0 .$$

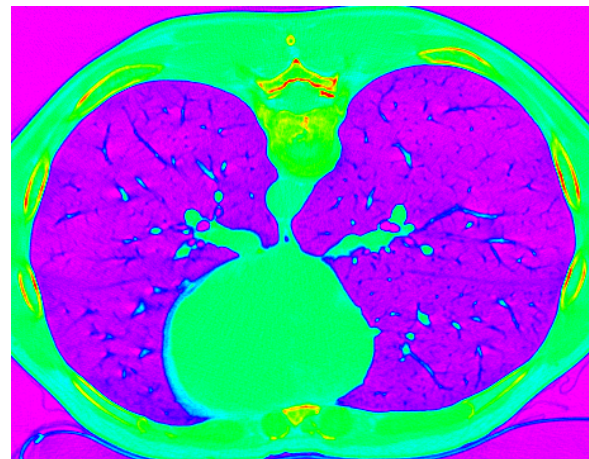
# Pseudocolor



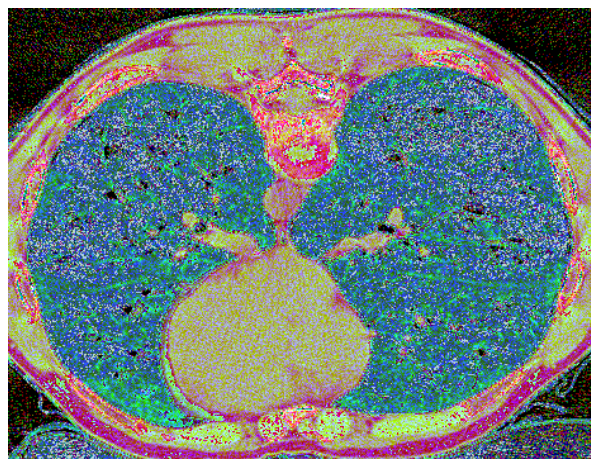
Unchanged změny



Spectrum



System (Windows)



Black body



# Brightness corrections

The new brightness  $f(i, j)$  depends on the position in the input image  $i, j$  of the input image  $g(i, j)$ .

Often the multiplicative model of the disturbance is considered:

$$f(i, j) = e(i, j) g(i, j).$$

Two approaches:

1. The corrective coefficients are obtained by capturing some **etalon area** with known brightness  $c$ . This is used to compensate irregular illumination. (The AGC in camera has to be switched off when capturing the etalon area). The captured image is

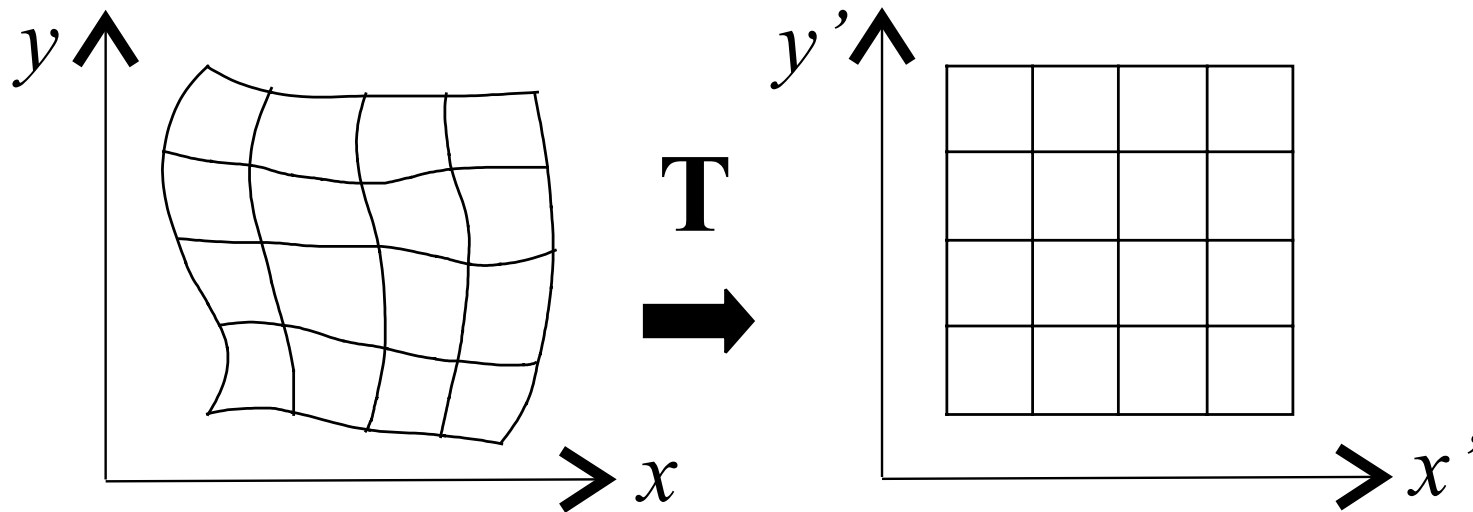
$$f_c(i, j) = e(i, j) c \quad \Rightarrow \quad e(i, j) = \frac{f_c(i, j)}{c}$$

2. The approximation of the background by some analytic surface and its subtraction from the original image.

# Geometric transforms

- ◆ Used for
  - size change, shift, rotation, slanting according to the transformation known in advance.
  - Undoing geometric distortions. The geometric distortions are often given by examples, images..
- ◆ Same techniques are used in theoretical mechanics, robotics, and computer graphics.
- ◆ Representation using homogeneous coordinates  $\implies$  a linear transformation expressed in the matrix form.

# Geometric transforms in 2D



Let us break down the vector transform  $T$  into two components

$$x' = T_x(x, y), \quad y' = T_y(x, y).$$

## Two needed steps due to discrete raster

*Transformation of pixel coordinates* provides new positions of pixel, which is calculated in continuous coordinates (real numbers) because the outcome can be off the original raster.

Problems:

- ◆ Part of the new transformed image lies off the image.
- ◆ Transformations are not invertible because some approximations were needed.

*Approximation of the brightness function* seeks the integer brightness value for the integer coordinate which best matches the newly calculated position  $x'$ ,  $y'$  given as real numbers.



# Transformation of pixel coordinates

The geometric transformation  $x' = T_x(x, y)$ ,  $y' = T_y(x, y)$  is often approximated by a polynomial of  $m$ -th order.

$$x' = \sum_{r=0}^m \sum_{k=0}^{m-r} a_{rk} x^r y^k, \quad y' = \sum_{r=0}^m \sum_{k=0}^{m-r} b_{rk} x^r y^k.$$

The formula is linear with respect to the coefficients  $a_{rk}$ ,  $b_{rk} \Rightarrow$  estimate using least-squares method.

The needed pairs of corresponding control points are  $(x, y)$  and  $(x', y')$ .

# Bilinear, affine transformation of coordinates

If the geometric transformation changes slowly in dependence on the position then approximating polynomials of lower order  $m = 2$  or  $m = 3$  suffice.

## Bilinear transformation

$$\begin{aligned}x' &= a_0 + a_1x + a_2y + a_3xy , \\y' &= b_0 + b_1x + b_2y + b_3xy .\end{aligned}$$

Even more special is the **affine transformation** which comprises practically needed rotation, translation and slant.

$$\begin{aligned}x' &= a_0 + a_1x + a_2y , \\y' &= b_0 + b_1x + b_2y .\end{aligned}$$

# Homogeneous coordinates $\Rightarrow$ matrix notation

- ◆ Homogeneous coordinates are common in theoretical mechanics, projective geometry, computer graphics, robotics, etc.
- ◆ The key idea is to represent the point in the space of the dimension increased by one.
- ◆ The point  $[x, y]^T$  is expressed in the 3D vector space as  $[\lambda x, \lambda y, \lambda]^T$ , where  $\lambda \neq 0$ .
- ◆ To simplify expressions, often only one of the infinitely many forms is used  $[x, y, 1]^T$ .

# Affine transformation expressed in matrix form

Affine mapping is expressed after introducing homogeneous coordinates as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_0 \\ b_1 & b_2 & b_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} .$$

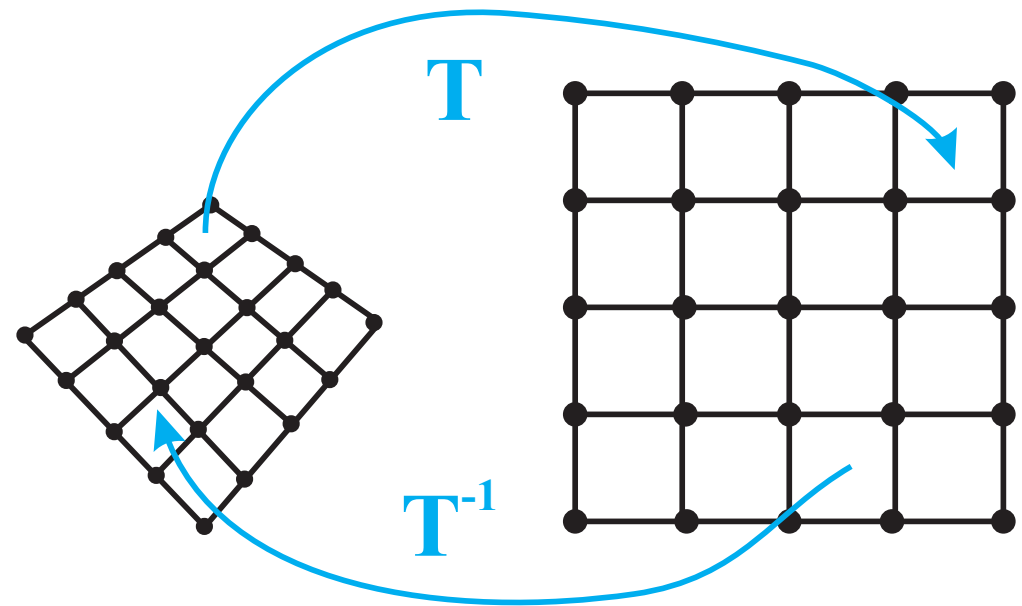
Notes:

- ◆ Relation to the PostScript language.
- ◆ The complicated geometric transforms (deformations) can be approximated by tiling the image into smaller rectangular areas. The simpler transform is used to express distortions in each tile, maybe estimated from the control points.

# Applying the geometric transformation

## Dual expression

- ◆ Input image has to be mapped by the transform  $\mathbf{T}$  into the output image.
- ◆ Two dual approaches:
  - Forward mapping:  
 $(x', y') = \mathbf{T} (x, y)$ .
  - Backward mapping:  
 $(x, y) = \mathbf{T}^{-1} (x', y')$ .
- ◆ The difference between the two expressions is due to needed brightness interpolation.



# Forward, backward mapping – comparison

## Forward mapping

- ◆ Output coordinates  $(x', y')$  can lie off raster.
- ◆ Two or more input pixels can be mapped to the same output pixel.
- ◆ Some output pixels might not get a value assigned. Gaps occur.

## Backward mapping

- ◆ For each pixel in the output image  $(x', y')$  the position in the input image  $(x, y)$  where the grayvalue should be taken from is computed using  $\mathbf{T}^{-1}$ .
- ◆ No gaps will appear.
- ◆ Problem:  $\mathbf{T}^{-1}$  might not always exist.

# Approximation of the brightness function

- ◆ Transformed coordinates  $(x', y')$  lie off the raster. Only the information about the input image  $f(x, y)$  in integer coordinates is available.
- ◆ The transformed image has to be expressed as a matrix too. That is the reason why the rectangular sampling grid is prescribed to us here too.
- ◆ Example: topographic illustration. Look at  $f(x, y)$  as at a landscape.
- ◆ The principally correct answer provides the [approximation theory](#). From available samples, the continuous 2D function is estimated.
- ◆ Often, a polynomials serve as this 2D function.

## Approximation as 2D convolution

- ◆ Instead of the originally continuous image function  $f(x, y)$ , its sampled variant  $f_s(l, k)$  is available.
- ◆ The outcome of the approximation (interpolation) is the brightness  $f_n(x, y)$ , where index  $n$  denotes individual interpolation methods. The brightness can be expressed as a 2D convolution

$$f_n(x, y) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f_s(l, k) h_n(x - l, y - k) .$$

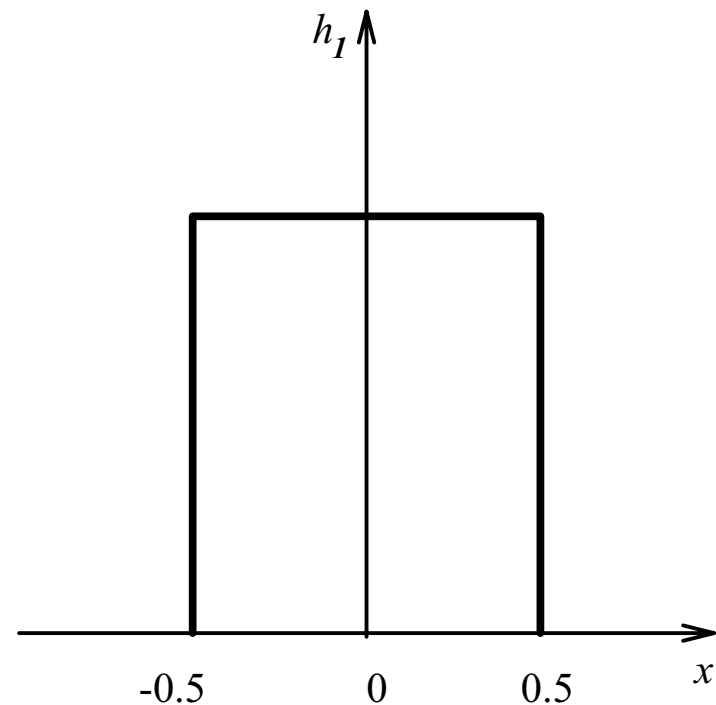
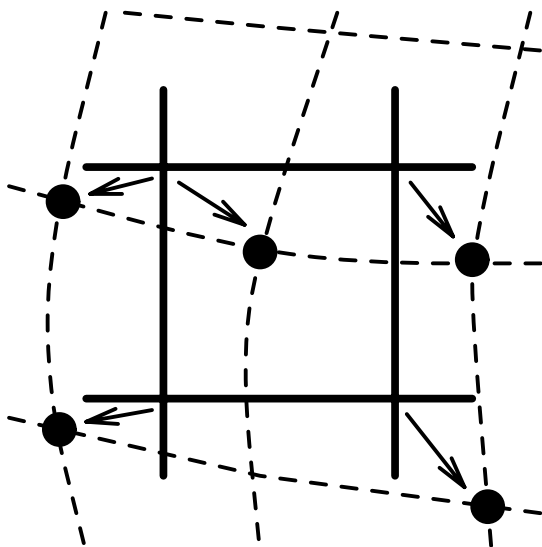
- ◆ Function  $h_n$  is the [interpolation kernel](#).
- ◆ Often, the local approximation kernel is used which covers only a small neighborhood of the current pixel to save computations. The values of the kernel  $h_n = 0$  outside the kernel domain.



# The nearest neighbor approximation

The point  $(x, y)$  is assigned the value of the nearest point  $g_s$  in the discrete raster.

$$h_1(x, y) = g_s(\text{round}(x), \text{round}(y)) .$$



# Linear interpolation

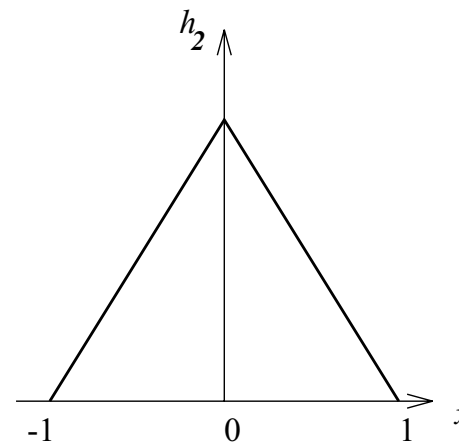
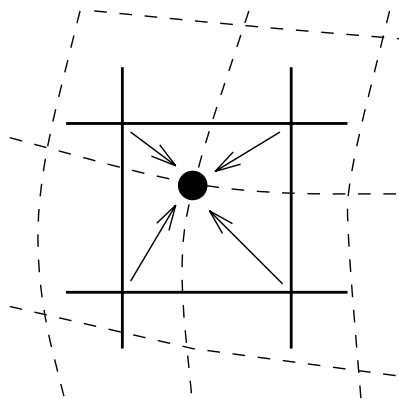
Explores 4 points in the neighborhood of  $(x, y)$  and combines them linearly. The influence of each point in the linear combination depends on the proximity to the current point.

$$f_2(x, y) = (1 - a)(1 - b) g_s(l, k) + a(1 - b) g_s(l + 1, k) + b(1 - a) g_s(l, k + 1) + ab g_s(l + 1, k + 1),$$

kde

$$l = \text{ceil}(x), \quad a = x - l,$$

$$k = \text{ceil}(y), \quad b = y - k.$$



# Bicubic interpolation

- ◆ interpolates locally the image function by the bicubic polynomial from sixteen points in the neighborhood.
- ◆ 1D interpolation kernel (for simplicity).

$$h_3 = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{pro } 0 \leq |x| < 1, \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{pro } 1 \leq |x| < 2, \\ 0 & \text{jinde.} \end{cases}$$

