Brightness and geometric transformations

Václav Hlaváč

Czech Technical University in Prague

Czech Institute of Informatics, Robotics and Cybernetics

160 00 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic

http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz

also Center for Machine Perception, http://cmp.felk.cvut.cz

Outline of the lecture:

- Brightness scale transformation.
- γ-correction. Histogram equalization.
- Pseudocolor.
- Brightness corrections.
- Geometric transformations.
- Interpolation of image function values.

Image preprocessing, the intro

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

The image is not interpreted.

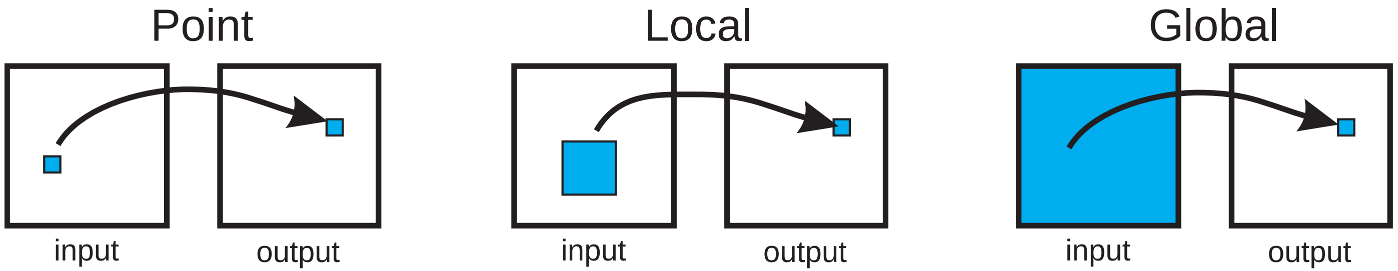
The aim

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

Taxonomy of the image preprocessing methods

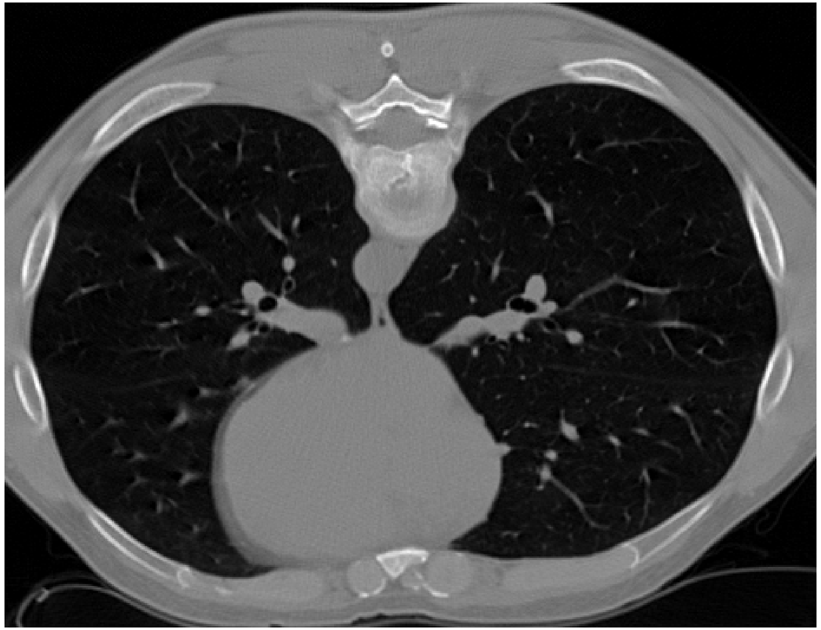
Taxonomy according to the neighborhood size of the current pixel.

Neighborhood	*Example of the operation*	*Actually processed neighborhood*
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 restauration	

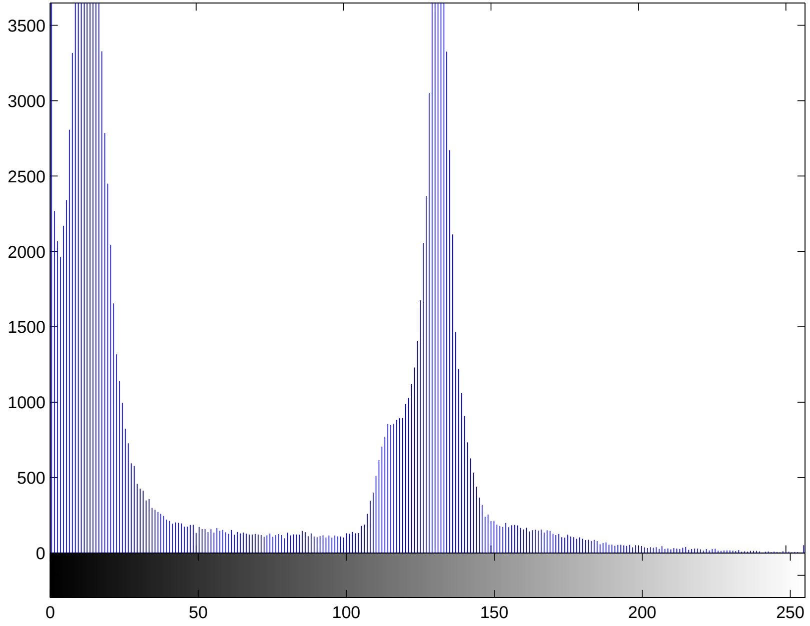


Brightness histogram

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



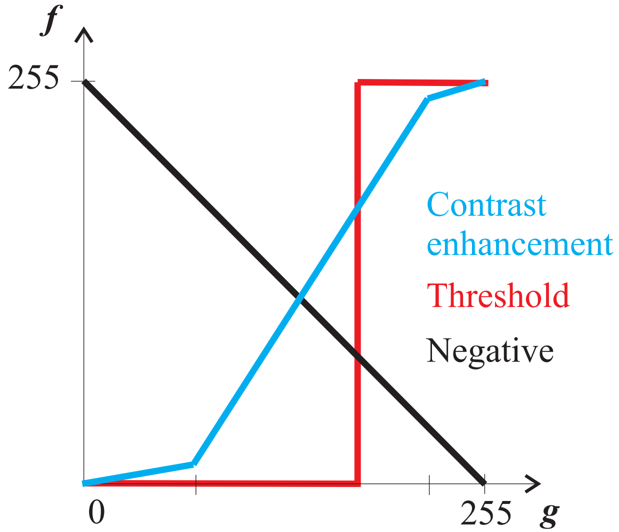
input image



brightness histogram

Transformation of the gray scale

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

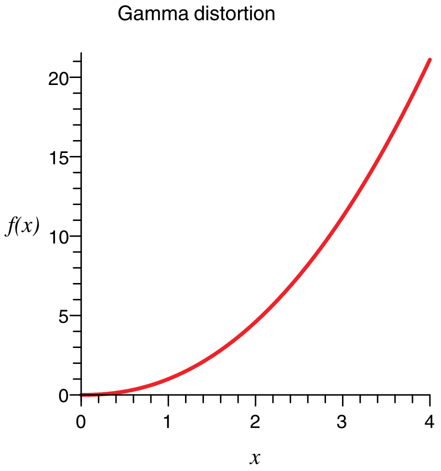


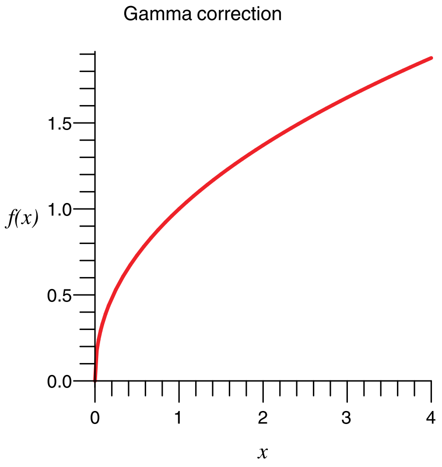
Notes:

- 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?

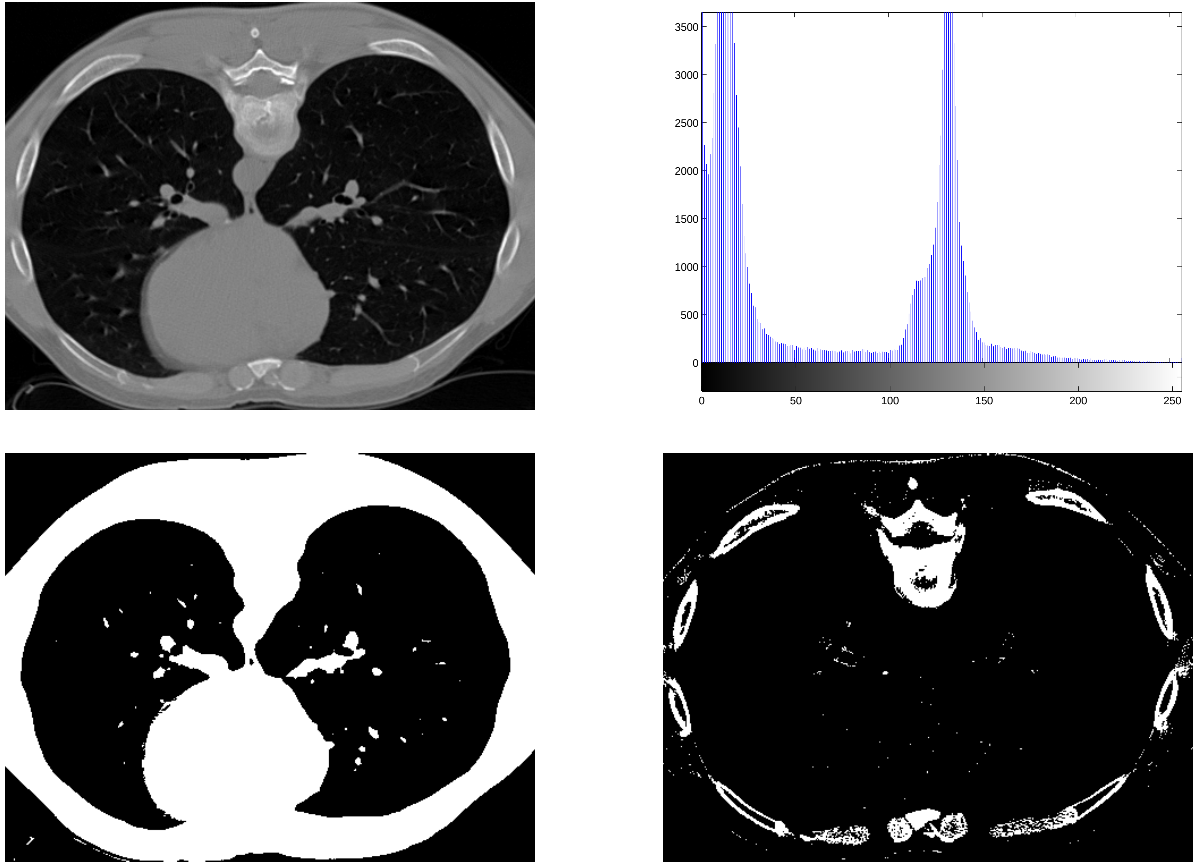
Nonlinearity in the intensity, γ correction

- Why cameras perform nonlinear brightness transformation, the γ-correction?
- In the cathode-ray tube (vacuum) displays, brightness $\approx U^{\gamma}$, where U is the grid voltage and γ is usually 2.2.
- The effect is compensated in cameras by the inverse function to preserve the linear transfer characteristic in the whole transmission chain. Having input irradiance E and the input voltage of the camera U, the inverse function is $U \approx E^{(1/\gamma)}$, usually $U \approx E^{0.45}$.
- Although the brightness of LCD monitors depends on the input voltage linearly, the γ-correction is still in use to secure the backward compatibility.





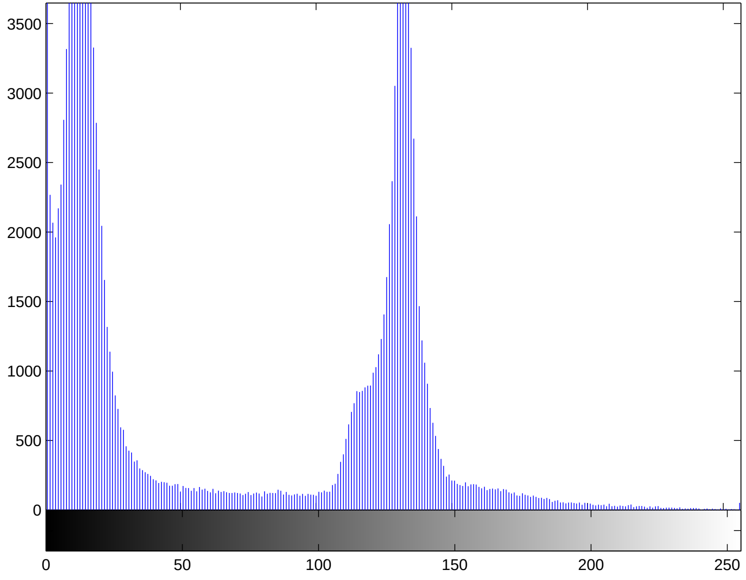
Example of two different thresholds



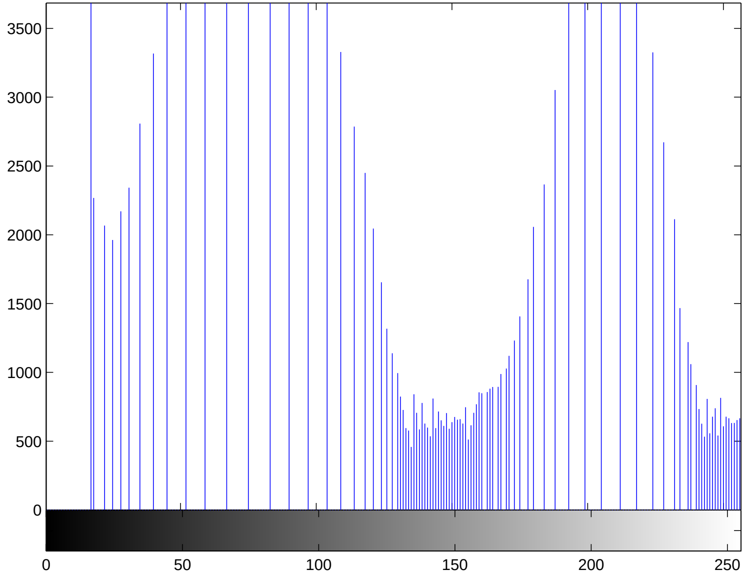
Histogram equalization

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 the automatic brightness comparison.

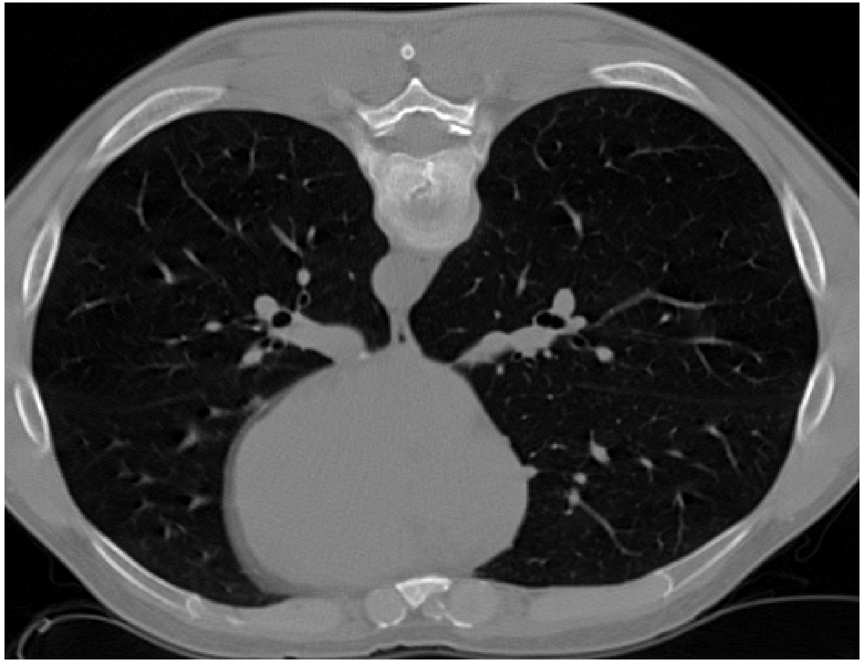


histogram of the input image

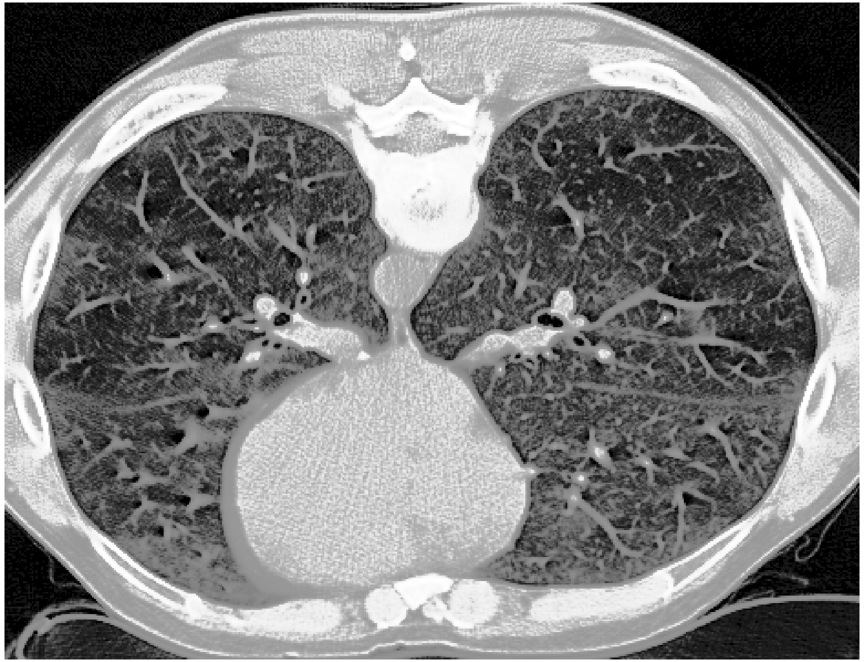


histogram after equalization

Increased contrast after the histogram equalization



original image



increased contrast

Derivation of the 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 brightness domain $q = \langle q_0, q_k \rangle$.

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

The equalized histogram $\approx$ the uniform distribution.

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

Derivation of the histogram equalization (2)

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

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

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

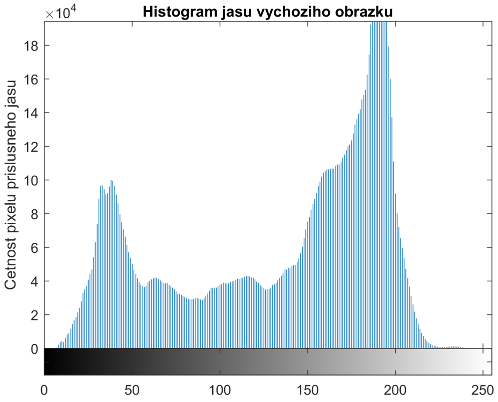
$$\frac{N^2(q - q_0)}{q_k - q_0} = \int_{p_0}^{p} H(s)\,\mathrm{d}s\,.$$

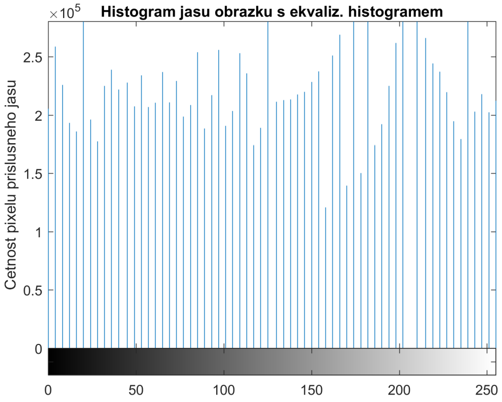
$$q = \mathcal{T}(p) = \frac{q_k - q_0}{N^2} \int_{p_0}^{p} H(s)\,\mathrm{d}s + q_0\,.$$

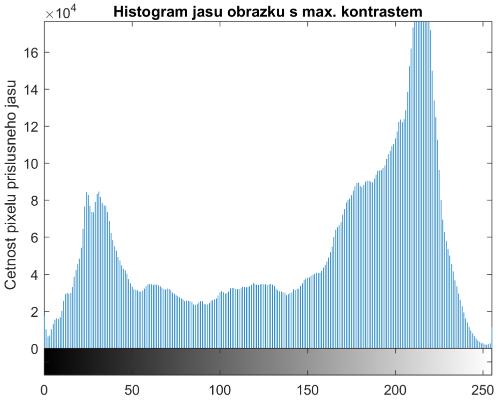
Discrete case, a cumulative histogram

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

Example: Contrast enhancement by a histogram equalization





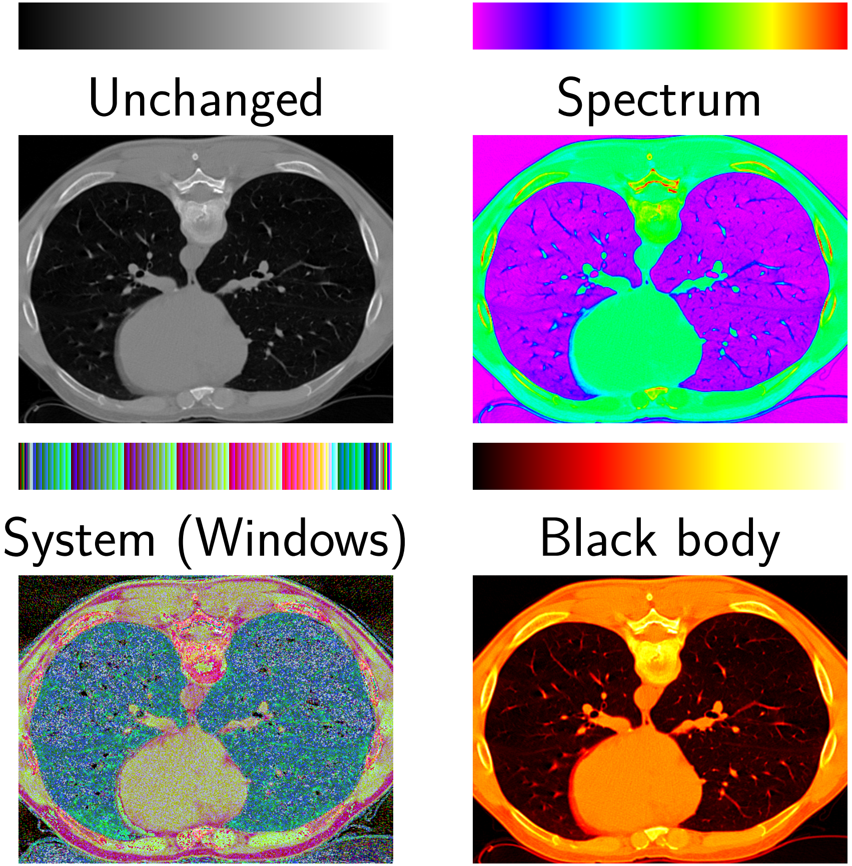


original

equalized histogram

max. contrast (Photoshop)

Pseudocolor



- Used when a grayscale image is shown to a human who recognizes $\approx$ 50 gray levels.
- Pseudocolor displays a grayscale image as a color image by mapping each intensity value to a color according to a lookup table or a function.
- The mapping table (function) is called the palette.
- Pseudocolor is often used to display a single channel data as temperature, distance, etc.

Brightness corrections

- The new brightness $f(i,j)$ depends on the position in the input image i,j of the input image $g(i,j)$.
- The multiplicative model of the disturbance is often considered: $f(i,j) = e(i,j)\, g(i,j)$.

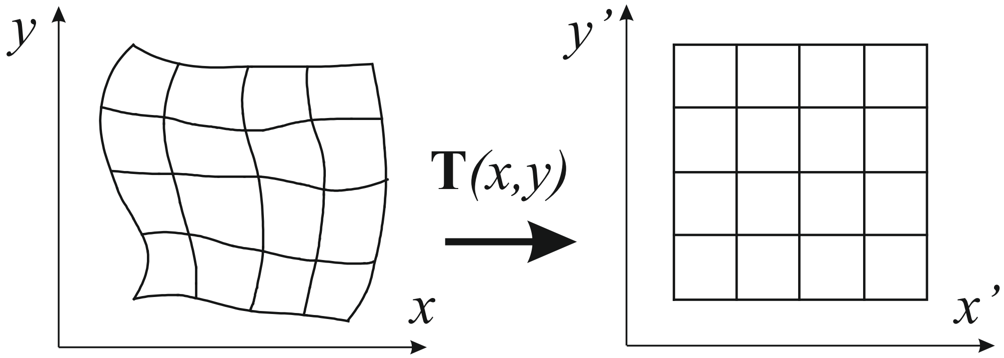
Two approaches:

1. The corrective coefficients are obtained by capturing some etalon area with the known brightness c. This is used to compensate the irregular illumination. (The AGC in the 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 transform, a motivation in 2D



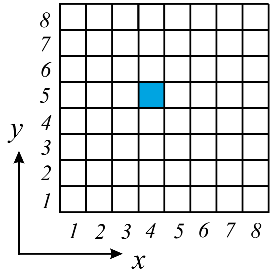
The vector transform $\mathbf{T}$ can be split into two components

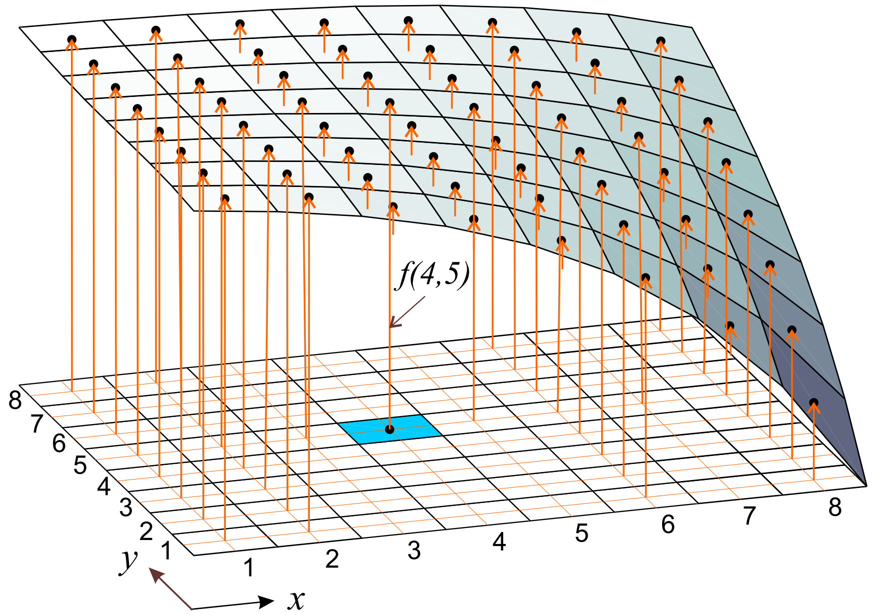
$$x' = T_x(x, y),\ y' = T_y(x, y).$$

- What are the conditions ensuring a one-to-one mapping?
- Geometric transforms are used for
  - Size change, shift, rotation, slanting according to the transformation known in advance.
  - Undoing geometric distortions. The geometric distortions are often estimated from examples, images.

Topographic view of the digital intensity image

- Recall that we considered the image $f(x, y)$ as a landscape in the introduction lecture.
- The image intensity (brightness) $f(x, y)$ corresponds to the height at the position x, y.
- A digital image provides only samples on a discrete grid illustrated as the orange arrows.

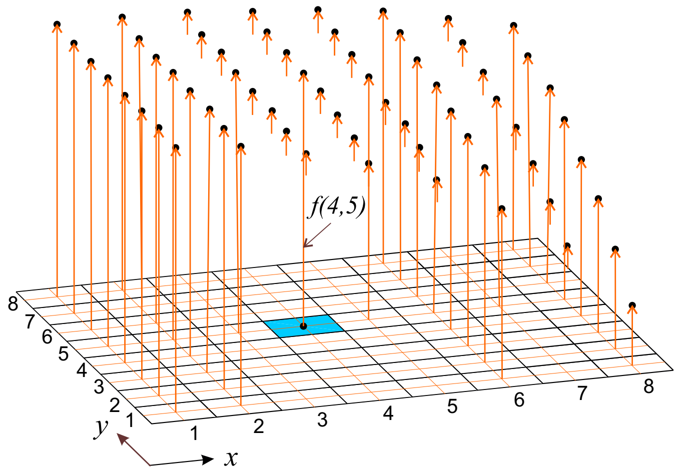


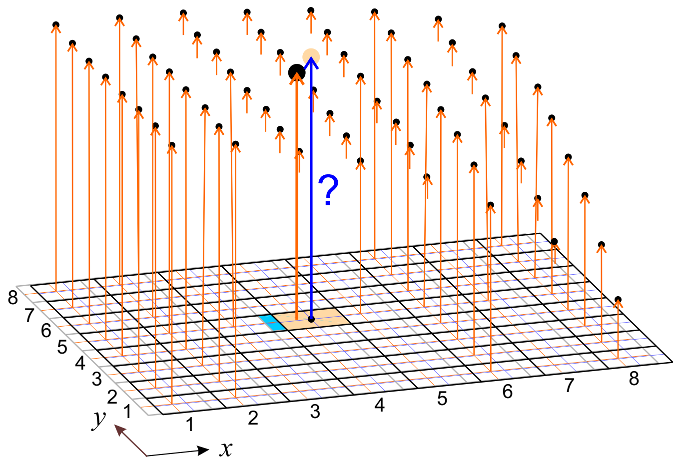


Topographic view, the geom. transformed image

- The digital image provides samples in the discrete grid only. See the top right picture.
- Consider the example: The image is shifted half a pixel in the x-axis direction. See the bottom right picture.
- The issue: There is no image function $f(x + 0.5, y)$ value available.
- The solution: Interpolate the missing value from the neighboring image function values.

  Q: Why neighboring values only?





Two needed steps due to the discrete raster

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

   Issues:

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

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

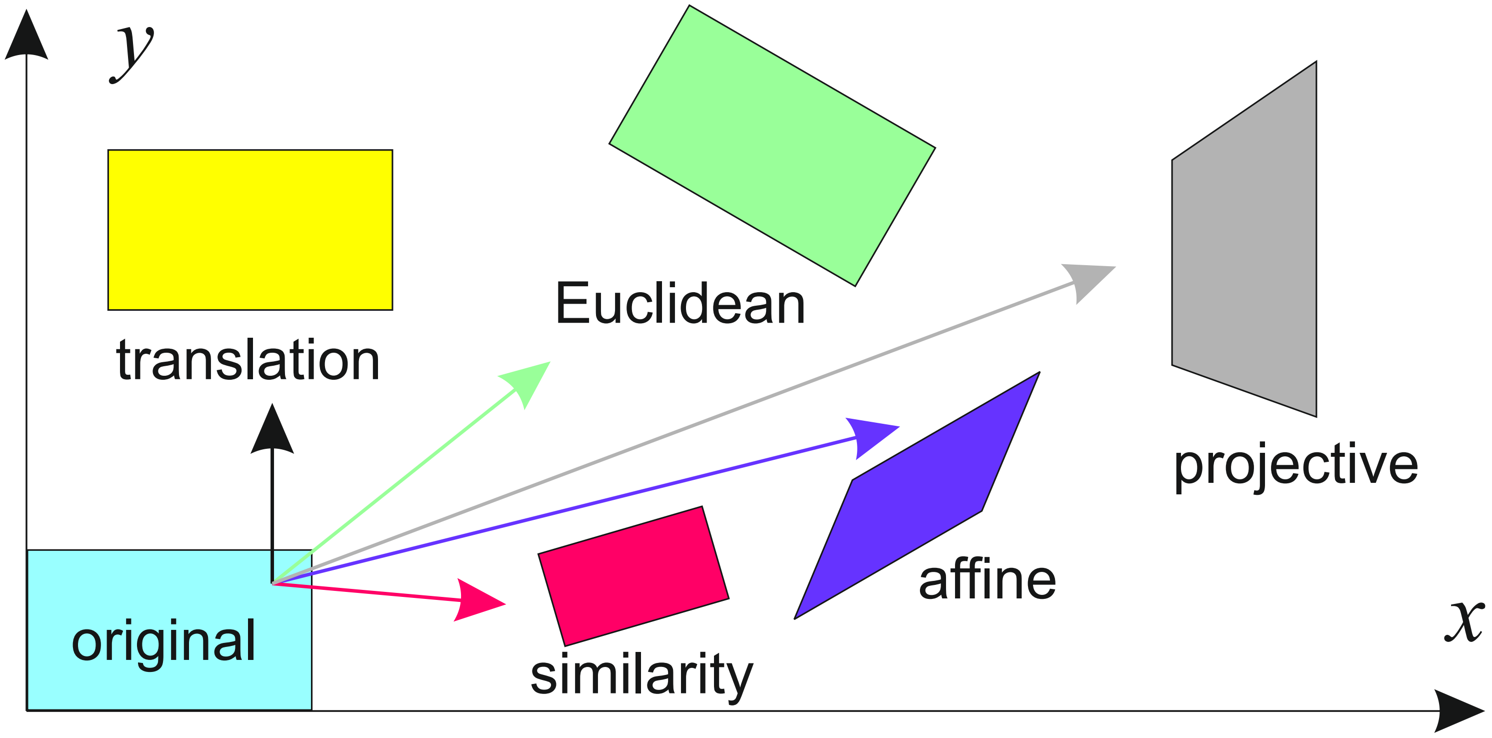
Transformation of pixel coordinates

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

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

- The formula is linear with respect to the coefficients a_{rk}, b_{rk}.
- The unknown coefficients can be obtained by solving a system of linear equations, in which pairs of corresponding control points (x, y) and (x', y') are known.
- If the system of linear equations were solved deterministically, only a few pairs of control points corresponding to the number of unknown parameters would suffice.
- However, very many control point pairs are used in practice, which leads to an overdetermined system of linear equations. The unknown coefficients a_{rk}, b_{rk} are estimated using the least-squares method.

A basic set of 2D planar transformations



Inspiration R. Szeliski.

Bilinear, affine transformation of coordinates

If the geometric transformation changes slowly with respect to the position then approximating polynomials of lower order $m = 2$ or $m = 3$ suffice.

- Bilinear transformation

$$
\begin{aligned}
x' &= a_0 + a_1 x + a_2 y + a_3 xy \,, \\
y' &= b_0 + b_1 x + b_2 y + b_3 xy \,.
\end{aligned}
$$

- Affine transformation, which is even more special, comprises practically needed rotation, translation and slant.

$$
\begin{aligned}
x' &= a_0 + a_1 x + a_2 y \,, \\
y' &= b_0 + b_1 x + b_2 y \,.
\end{aligned}
$$

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 $[x, y]^\top$ is expressed in the 3D vector space as $[\lambda x, \lambda y, \lambda]^\top$, where $\lambda \neq 0$.
- To simplify expressions, often only one of the infinitely many forms is used $[x, y, 1]^\top$.

Affine transformation expressed in a matrix form

The affine mapping is expressed after introducing homogeneous coordinates as

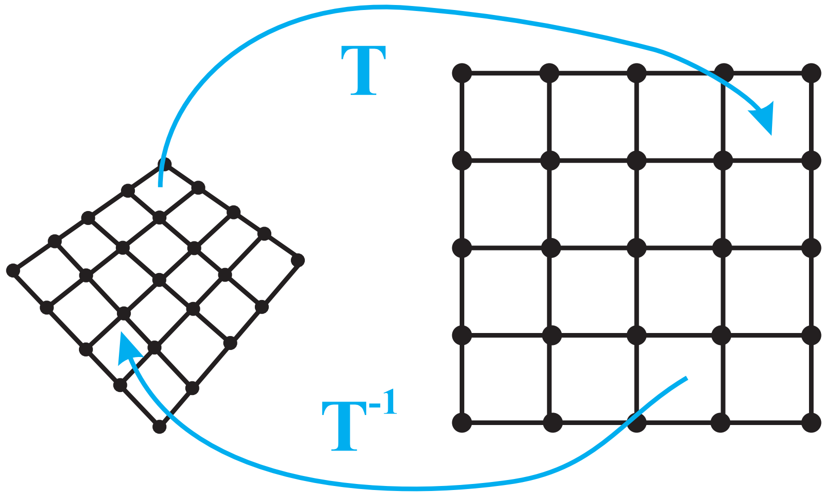
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_0 \\ b_1 & b_2 & b_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} .$$

Notes:

- Relation to the PostScript language (sold from 1984; Apple LaserWriter 1985).
- 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

- The input image has to be mapped by the transform $\mathbf{T}$ to 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 the needed brightness interpolation.



Forward, backward mapping – comparison

Forward mapping

- Output coordinates (x', y') can lie off the 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 gray value should be taken from is computed using $\mathbf{T}^{-1}$.
- No gaps will appear.
- Problem: $\mathbf{T}^{-1}$ might not always exist.

Image function approximation

- The principally correct answer provides the approximation theory. The continuous 2D function is estimated from available samples.
- Often, the estimated 2D function is approximated locally by a 2D polynomial.
- Interpolation is a special case of approximation, when the estimated value lies "within" known samples.

Q: Why is the approximation by polynomials popular?

A_1: A polynomial has a few parameters only. The best values of the parameters are sought using an optimization method. Polynomials have continuous derivatives. Consequently, the optimum can be derived analytically or can be computed numerically using the gradient method.

A_2: In the case that the approximation polynomial parameters are estimated experimentally, there are usually many more examples than parameters. Having in mind A_1, the needed derivatives can be expressed analytically.

Three used interpolation methods

Three interpolation methods are used to estimate the value of 2D image function, e.g. after geometric transformations:

1. Nearest neighbor.
2. Bilinear interpolation.
3. Bicubic interpolation.

Note: The prefix “bi” in bilinear/bicubic means that 1D interpolace is applied in one dimension first and in the second dimension next.

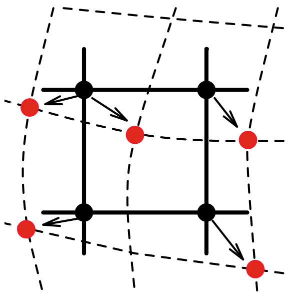
For each of three methods, we will explain the idea of the interpolation function in three figures. We will show how the interpolation looks in 1D case. Two steps of the interpretation in 2D will be explicated next.

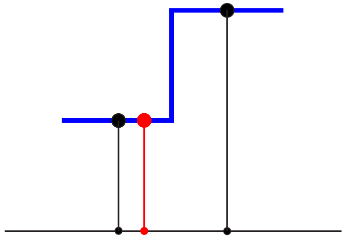
After three methods are explained, we will illustrate methods behavior on one practical example, enlarging the text 20 times.

The nearest neighbor interpolation

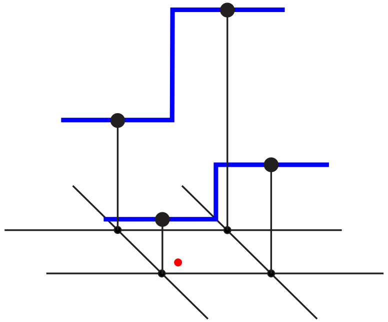
- The point (x, y) is displayed as a red circle in the left picture inside a square corresponding to a sampling raster.
- The nearest neighbor interpolation assigns the value of the nearest sample in sampled image g_s raster to the value to the image function $g(x, y)$.

$$f(x, y) = g_s(\mathrm{round}\,(x),\ \mathrm{round}\,(y))\ .$$

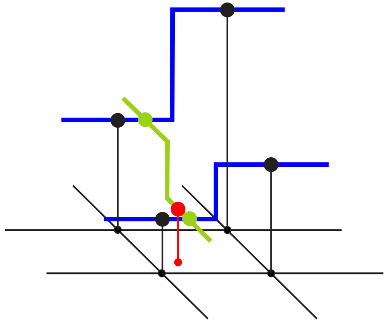




1D nearest neighbor



2D nearest neighbor step 1



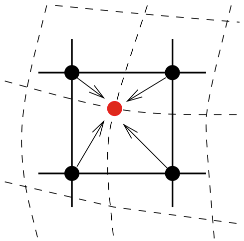
2D nearest neighbor step 2

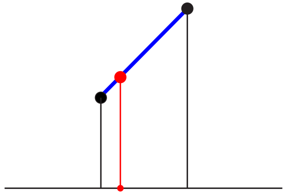
Bilinear interpolation

Consider one pixel (a square) in a discrete raster. We know four values of an image function f in its corner points (black points), left figure. We approximate the value of the continuous image function $f(x, y)$ for the point (x, y) (red point) lying outside known samples.

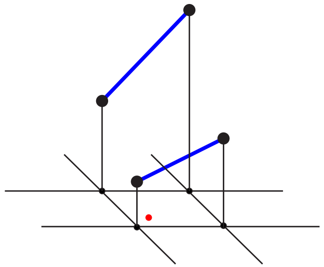
In the first step, the function is 1D approximated in one coordinate system direction by linearly fitting two pairs of image function extreme values by a (blue) straight line segments. Second figure from the left shows 1D situation; third figure from the left 2D situation.

In the second step, two locations (green points) are found on two blue line segments corresponding to point (x, y). These locations are fitted linearly again (green line segment). This yields the sought value $f(x, y)$ (red point).

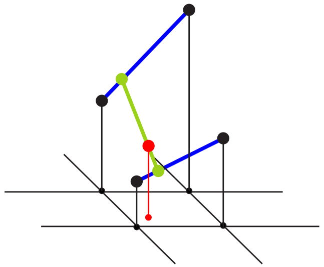




1D linear



2D bilinear
step 1



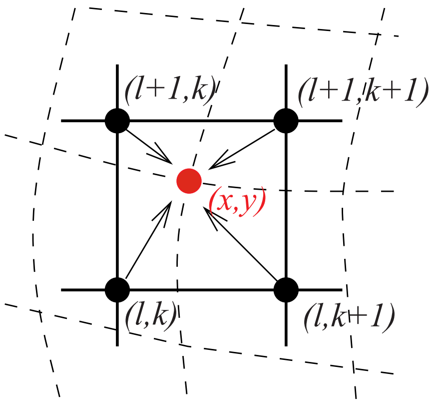
2D bilinear
step 2

Bilinear interpolation, calculation in 2D

The interpolated value $f(x,y)$ is calculated by a linear combination of four available samples $f_s(x,y)$. The influence of each samples depends on the proximity to the interpolated point in the linear combination.

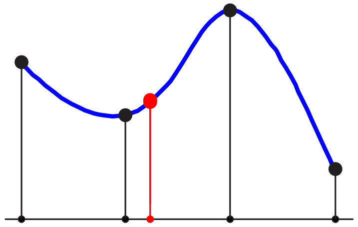
$$\begin{aligned} f(x,y) &= (1-a)(1-b)\,g_s(l,k) + a\,(1-b)\,g_s(l+1,k) \\ &+ b\,(1-a)\,g_s(l,k+1) + ab\,g_s(l+1,k+1)\,, \end{aligned}$$

where
$$\begin{aligned} l &= \mathrm{ceil}\,(x)\,, \quad a = x - l\,, \\ k &= \mathrm{ceil}\,(y)\,, \quad b = y - k\,. \end{aligned}$$

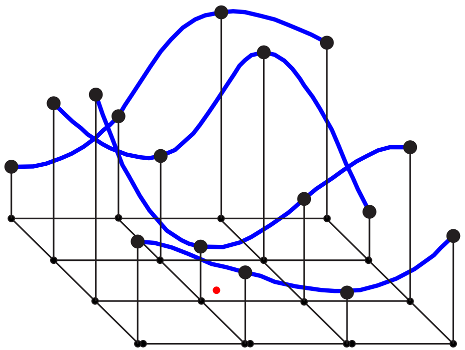


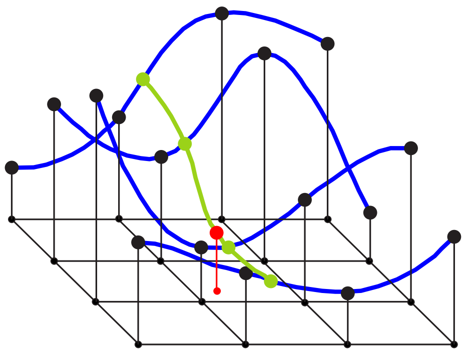
Bicubic interpolation

- The input consists of 16 samples f_s in the sampling grid.
- The output is the interpolated image function value $f(x, y)$ for the location (x, y) (red point) lying outside of the sampling grid.
- In the first step, the interpolation is in one coordinates direction by four 1D polynomials (four blue curves). In the second step, the four locations (green points) are found corresponding to the location (x, y). These points are interpolated by one 1D cubic polynomial yielding sought value (red point).



1D cubic

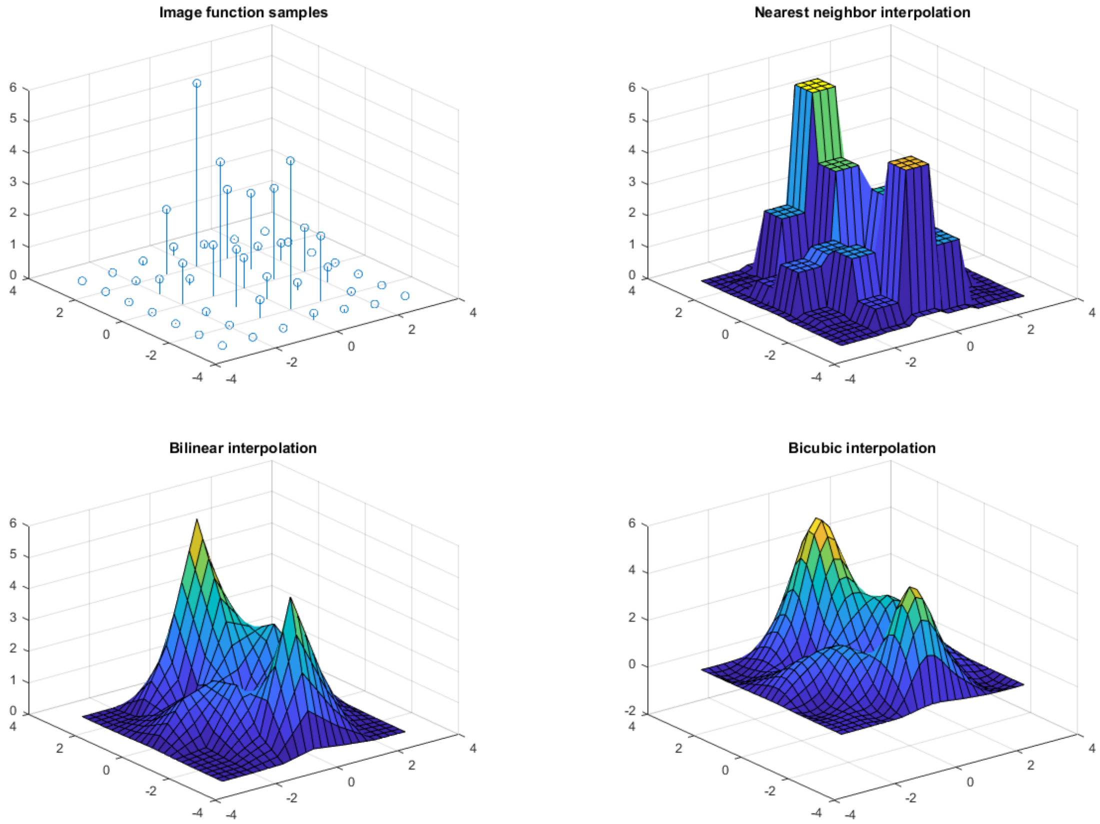




2D bicubic
step 2

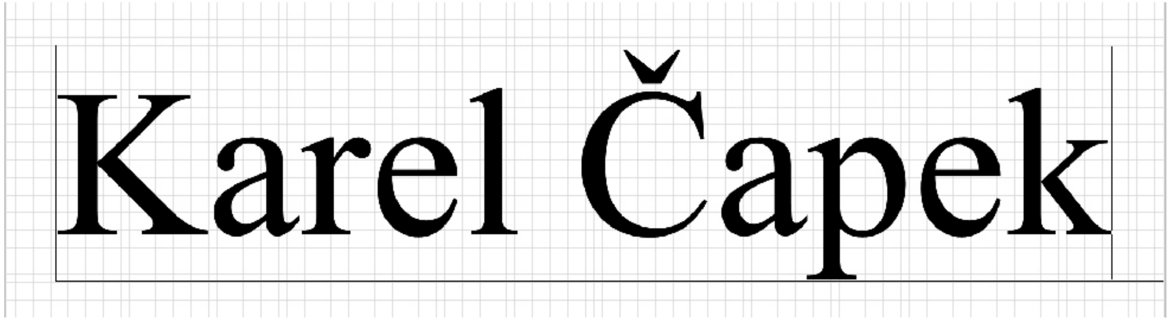
Three interpolation method illustrated as surfaces

Input data are deliberately in a low resolution 7 $\times$ 7 resolution.



Input image for comparing three interpolations

Times Roman
Ideal
36 x 132 px



Times Roman
Real
36 x 132 px

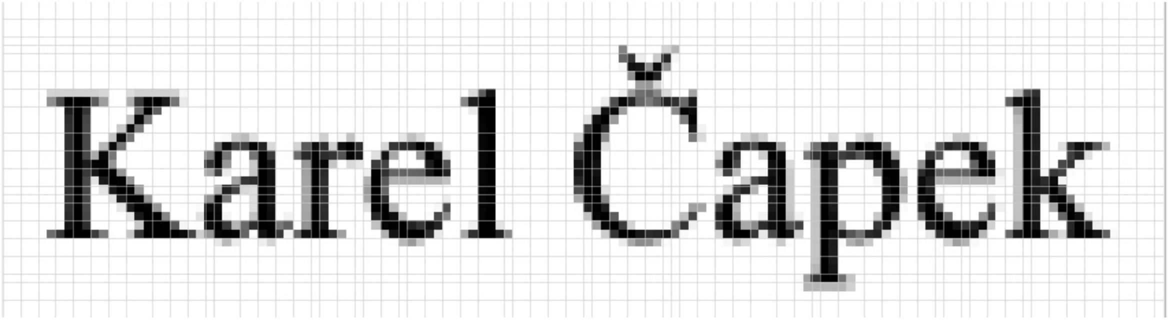


Image enlarging 20x, comparison, three methods

Ideal
Times Roman

Karel Čapek

Nearest neighbor

Bilinear

Bicubic

Approximation as 2D convolution

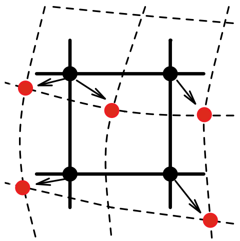
- Instead of the originally continuous image function $f(x, y)$, its sampled variant $f_s(l, k)$ is available.
- The outcome of the approximation (interpolation) is the brightness $f_n(x, y)$, where index n denotes individual interpolation methods. The brightness can be expressed as a 2D convolution

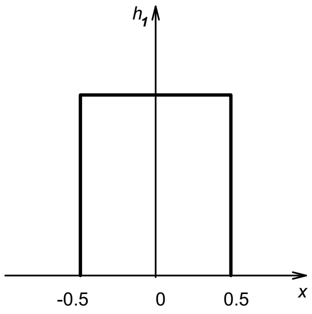
$$f_n(x, y) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f_s(l, k)\, h_n(x - l,\; y - k)\,.$$

- Function h_n is the interpolation kernel. h_1 for nearest neighbor, h_2 for bilinear interpolation, h_3 for bicubic interpolation.
- 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.

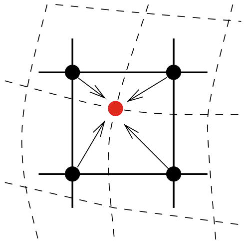
Three convolution kernels pictorially in 1D

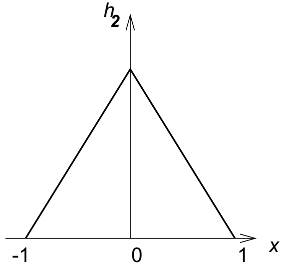
Nearest neighbor





Bilinear interpolation





Bicubic interpolation