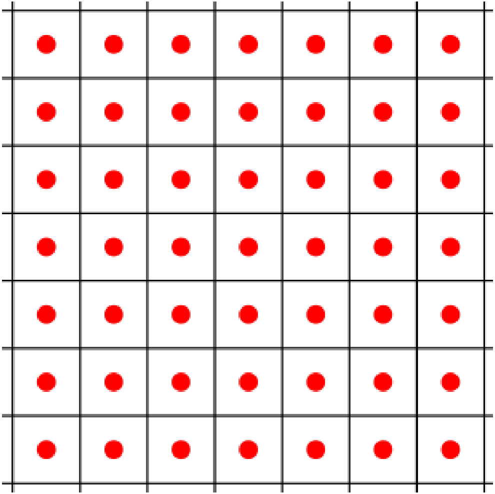
Advanced Image Processing - Image Transformations

Ing. Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

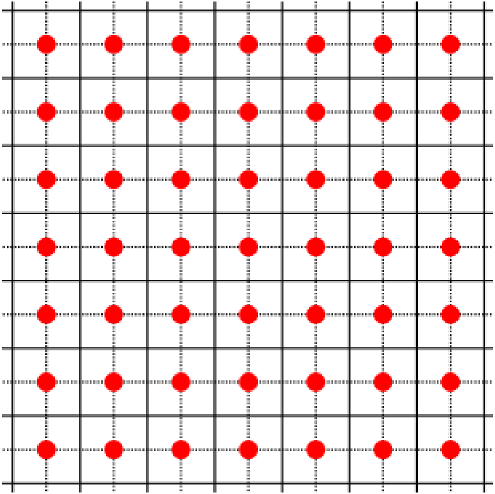
4.12.2019

Information in images



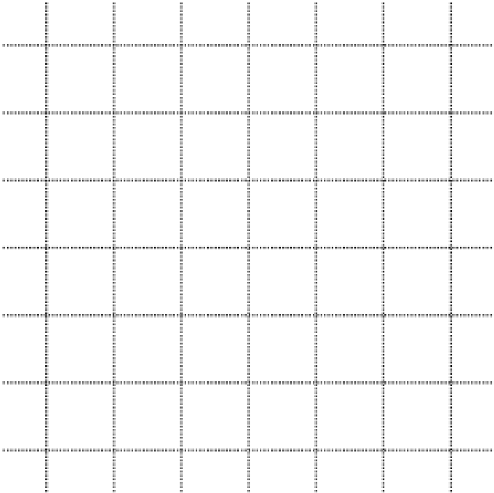
We consider the intensity of a pixel to be in its center.

Informácia v obraze

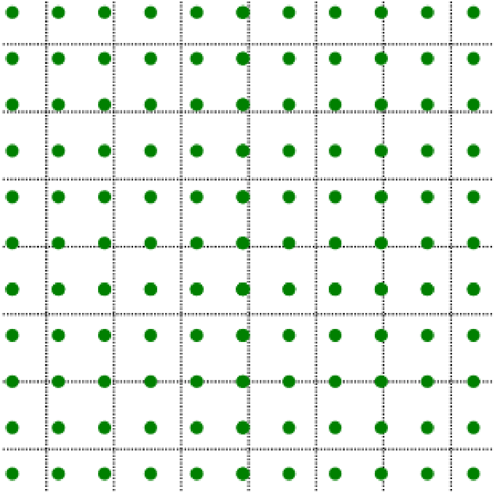


Dashed grid therefore shows the centers of pixels not their boundaries.

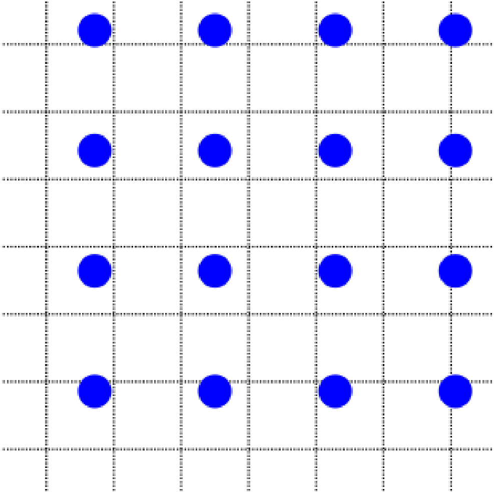
Information in images



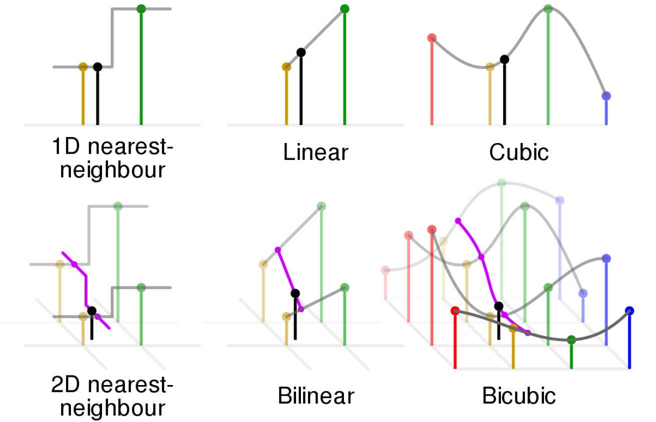
Resizing - Enlargement



Resizing - Reduction



Interpolácia



1D nearest-neighbour

Linear

Cubic

2D nearest-neighbour

Bilinear

Bicubic

How the value of the new pixel is calculated is given by interpolation.

Resizing in Matlab

imresize

imresize(I, scale) - returns the image I resized by the scale factor

imresize

imresize(I, [r, c]) - returns the image I resized to size $r \times c$

imresize

imresize(I, s, 'method') - returns the resized image I with the use of method: 'nearest', 'bilinear', 'bicubic'.

Exercise

Test resizing with different methods for the image shell.jpg and zatisie.jpg

Affine transformation

How is it calculated

The transform is given by the following equation where $\vec{y}$ is the new position of the pixel.

$$\vec{y} = \mathbb{A}\vec{x} + \vec{t}$$

Calculation for images

When considering images we do not calculate $\vec{y}$ based on pixel positions $\vec{x}$, but instead we first choose some regular grid of $\vec{y}$ vectors and then calculate their respective position in the image using the inverse transform $\vec{x} = \mathbb{A}^{-1}(\vec{y} - \vec{t})$. This allows us to use interpolation in a straightforward fashion.

Exercises

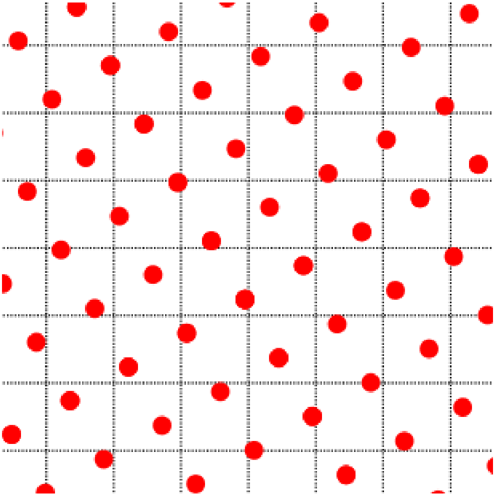
Rotation

$$\mathbb{A} = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix}$$

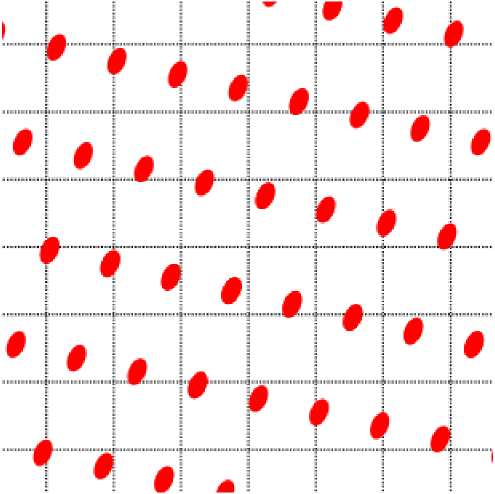
x-axis scaling

$$\mathbb{A} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

Rotation



Affine transform



Affine transformation in Matlab

imtransform

imtransform(I, tform, interp) - transforms the image I with a transformation object t from using interpolation method interp: 'nearest', 'bilinear', 'bicubic'.

maketform

maketfrom('affine', B) - retursn transformation object for affine transformation. The transformation is defined with matrix B, which in our definition is the matrix A with additional column containing the vector $\vec{t}$.

imrotate

imrotate(I, angle) - returns the image I rotated by the given angle.

Exercises

Exercises

Perform a rotation of an image using imrotate. Try to accomplish the same result with affine transformation.

Exercise

Construct and affine transformation which flips just the x or y axis.

Exercise

Test various matrices for affine transformation.

Perspective transformation

imtransform

imtransform(I, tform, interp) - transforms the image I with a transformation object t from using interpolation method interp: 'nearest', 'bilinear', 'bicubic'.

maketform

maketfrom('projective', U, X) - returns a transformation object for perspective transformation. The matrices U and X are of shape 4×2. Each row of U is transformed to the corresponding row in X.

Matrices U and X

We can create the U matrix by calling U $=$ ginput(4). We can use ginput to create X as well, or in case of rectification (making an object axis aligned) we can create a matrix in which each row is a different corner of a rectangle.

Exercises

Exercise

In the image qr.jpg use the perspective transformation in a way so that the QR code is rectified. Perform the same with the image book.jpg.

Exercise

In the image road.png use the perspective transformation so that the traffic lines are aligned with the y-axis. Is this task well-defined?