Coordinate Systems

Mapping from pixel to world coordinates

A. Maier, J. Maier, B. Bier, A. Preuhs, C. Syben

WS 2017/18

Pattern Recognition Lab (CS 5)
Friedrich-Alexander University Erlangen-Nuremberg

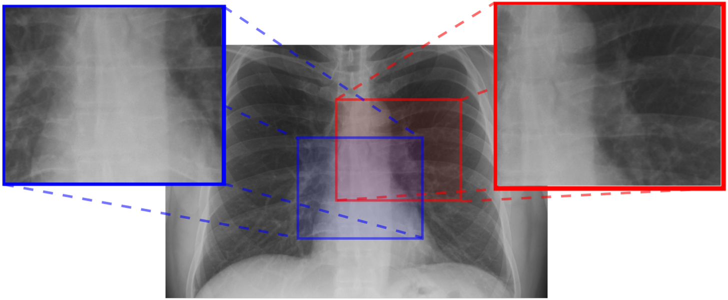
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Pixel coordinates

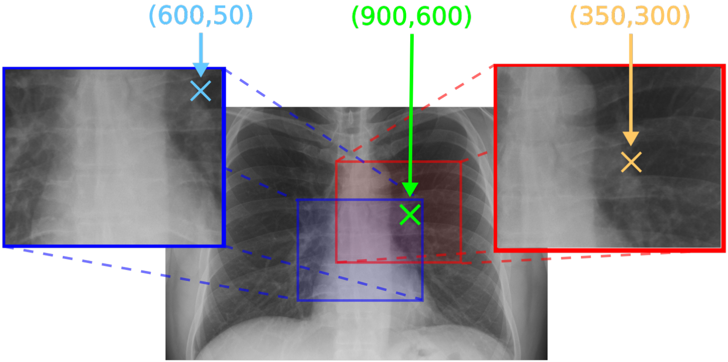
- Consider an X-ray image of arbitrary size

Pixel coordinates



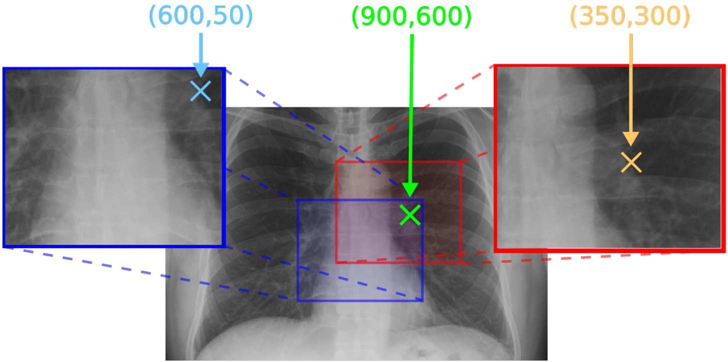
- Consider an X-ray image of arbitrary size
- ROIs drawn by Physician A and B

Pixel coordinates



- Consider an X-ray image of arbitrary size
- ROIs drawn by Physician A and B

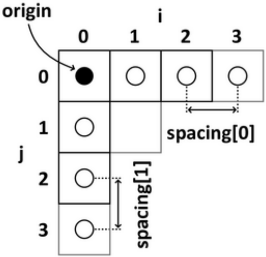
Pixel coordinates



- Pixel coordinates are all different

► **Yet, they refer to the exact same point!**

Image properties



- Origin $\mathbf{o} \in \mathbb{R}^2$
- Basis vectors $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{R}^2$

For the basis vectors it holds:

- $\mathbf{e}_1^\top \cdot \mathbf{e}_2 = 0$
- $\mathbf{e}_1^\top \cdot \mathbf{e}_1 = s_1$ is the spacing in e_1-direction
- $\mathbf{e}_2^\top \cdot \mathbf{e}_2 = s_2$ is the spacing in e_2-direction

From image to world coordinates

Map pixel coordinates $\mathbf{p} \in \mathbb{N}^2$ to world coordinates $\mathbf{x} \in \mathbb{R}^2$

Image to world

$$\mathbf{x} = (\mathbf{e_1}\ ,\ \mathbf{e_2}\ ,\ \mathbf{o}) \cdot \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix}$$

Often $\mathbf{e}_1 = \begin{pmatrix} s_1 \\ 0 \end{pmatrix}$, $\mathbf{e}_2 = \begin{pmatrix} 0 \\ s_2 \end{pmatrix}$, $\mathbf{o} = \begin{pmatrix} -(N_1 - 1.0)\frac{s_1}{2} \\ -(N_2 - 1.0)\frac{s_2}{2} \end{pmatrix}$,

where $\mathbf{N} \in \mathbb{N}^2$ is the image dimension (number of pixels).

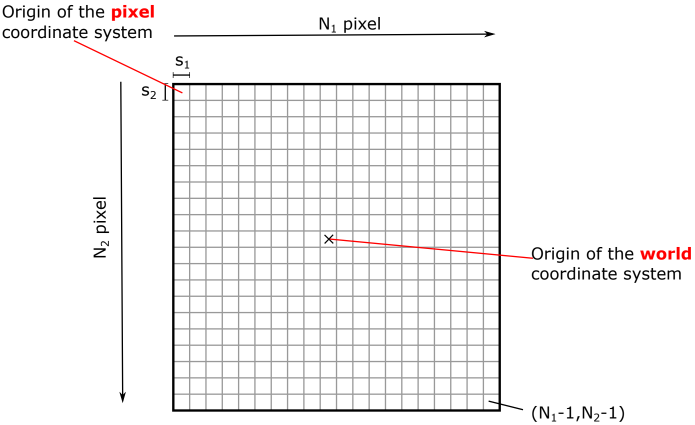
From world to image coordinates

If $\mathbf{e}_1 = \begin{pmatrix} s_1 \\ 0 \end{pmatrix}$, $\mathbf{e}_2 = \begin{pmatrix} 0 \\ s_2 \end{pmatrix}$ the inversion is easy:

World to image

$$\mathbf{p} = \begin{pmatrix} {}^1\!/_{s_1} & 0 \\ 0 & {}^1\!/_{s_2} \end{pmatrix} \cdot (\mathbf{x} - \mathbf{o})$$

Overview


Origin of the **pixel** coordinate system
N_1 pixel
s_1
s_2
N_2 pixel
Origin of the **world** coordinate system
$(N_1\text{-}1,N_2\text{-}1)$

Warning Warning Warning Warning Warning

80% of the "hard to find mistakes" in the end are due to ignoring the correct handling of the two coordinate systems.
In your own interest, please consider the following advice:

- Do not test with a spacing of 1.0 only
- Use the member functions of the class Grid2D to set the origin and spacing: **setOrigin**(...) and **setSpacing**(...)
- Once correctly set, use the member functions of the Grid2D to convert from the two coordinate systems into each other: **indexToPhysical**(...) and **physicalToIndex**(...)