Coordinate Systems

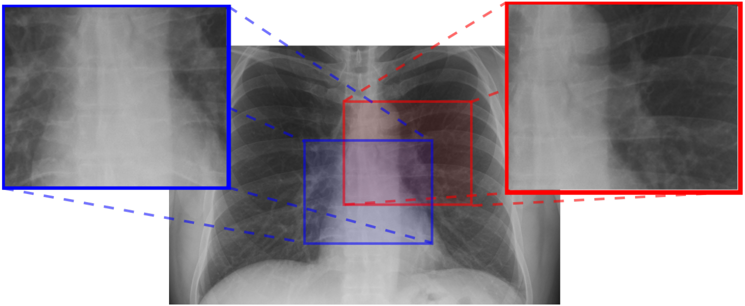
Mapping from pixel to world coordinates

Y. Huang, F. Wagner, A. Maier | Flat-Panel CT Reconstruction, FAU Erlangen, SS 2022

Pixel coordinates

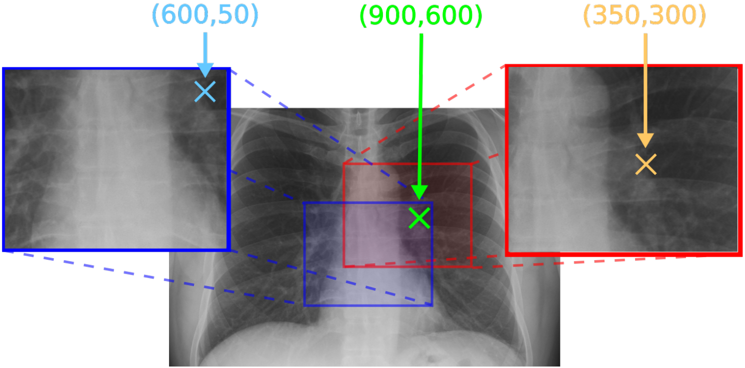
- Consider an X-ray image of arbitrary size

Pixel coordinates



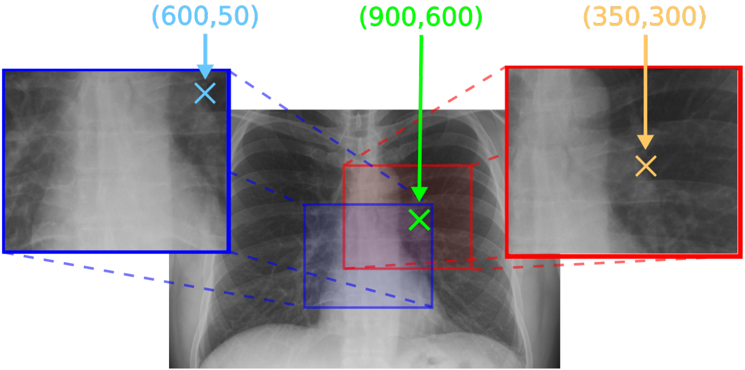
- Consider an X-ray image of arbitrary size
- ROIs drawn by Physician A and B

Pixel coordinates



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

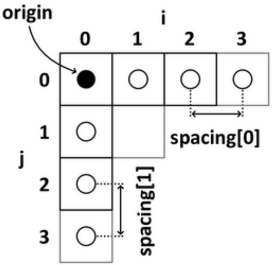
Pixel coordinates



- Pixel coordinates are all different

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

Image properties

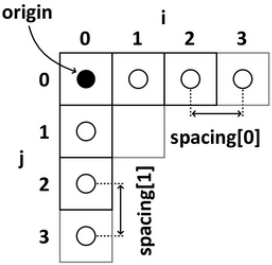


- Origin $\mathbf{o} \in \mathbb{R}^2$
- Basis vectors $\mathbf{e}_0, \mathbf{e}_1 \in \mathbb{R}^2$
- $\mathbf{e}_0^\top \cdot \mathbf{e}_1 = 0$
- $\|\mathbf{e}_0\| = s_0$ is the spacing in x-direction
- $\|\mathbf{e}_1\| = s_1$ is the spacing in y-direction

Image width and height:

- Width = $N_0 \times s_0$
- Height = $N_1 \times s_1$

Image properties



- Origin $\mathbf{o} \in \mathbb{R}^2$
- Basis vectors $\mathbf{e}_0, \mathbf{e}_1 \in \mathbb{R}^2$
- $\mathbf{e}_0^\top \cdot \mathbf{e}_1 = 0$
- $||\mathbf{e}_0|| = s_0$ is the spacing in x-direction
- $||\mathbf{e}_1|| = s_1$ is the spacing in y-direction

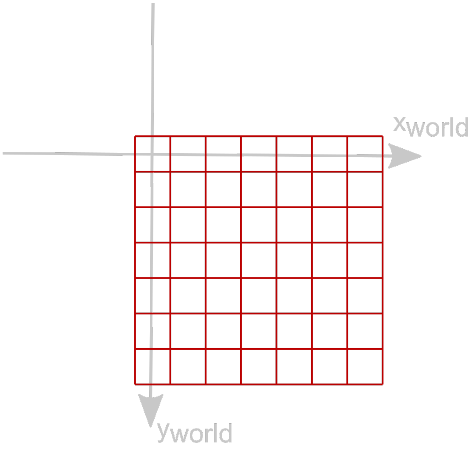
Image width and height:

- Width = $N_0 \times s_0$
- Height = $N_1 \times s_1$

Pixel and world coordinate conversion:

- Pixel coordinates: (i, j)
- What are its world coordinates?

How to set the origin



How to set the origin

$o_{pixel} = [0, 0]$

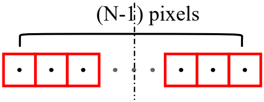
$o_{world} = [o_x, o_y] = [0, 0]$

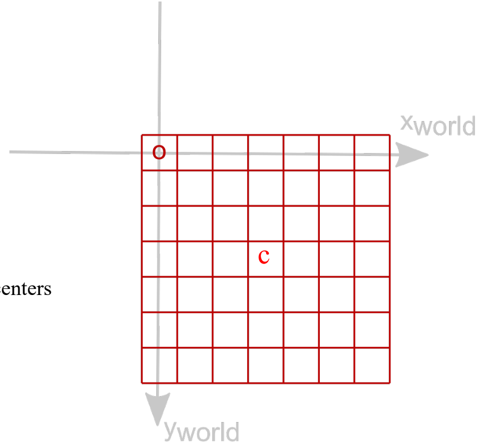
$c_{pixel} = [0.5(N_0\text{-}1), 0.5(N_1\text{-}1)]$

$c_{world}=[0.5(N_0\text{-}1)s_0, 0.5(N_1\text{-}1)s_1]$

Pixel indices stand for the pixel centers

Half a pixel shift



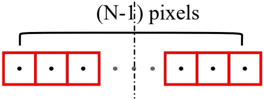


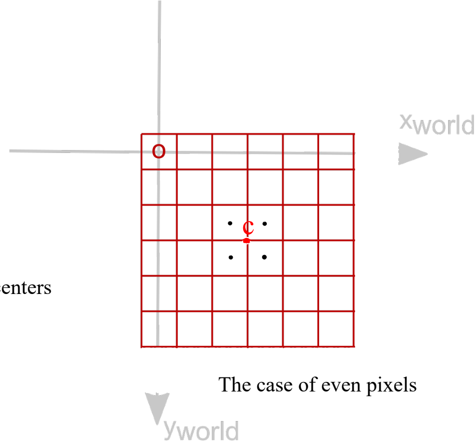
How to set the origin

$o_{pixel} = [0, 0]$
$o_{world} = [o_x, o_y] = [0, 0]$

$c_{pixel} = [0.5(N_0\text{-}1), 0.5(N_1\text{-}1)]$
$c_{world}=[0.5(N_0\text{-}1)s_0, 0.5(N_1\text{-}1)s_1]$

Pixel indices stand for the pixel centers
Half a pixel shift





The case of even pixels

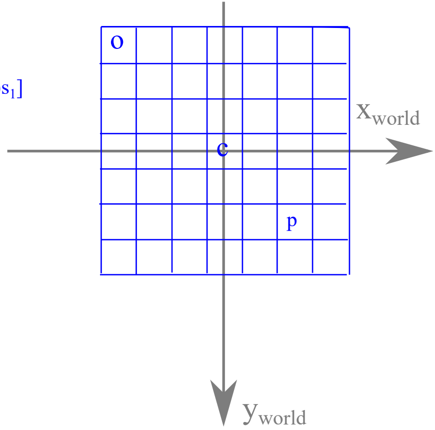
setting the origin = setting the world-coordinate of the upper left pixel

How to set the origin

$o_{pixel} = [0, 0]$
$o_{world} = [o_x, o_y] = [-0.5(N_0-1)s_0, -0.5(N_1-1)s_1]$

$c_{pixel} = [0.5(N_0-1), 0.5(N_1-1)]$
$c_{world}=[0, 0]$

$p_{pixel} = [i, j]$
$p_{world} = [o_x + i*s_0, o_y + j*s_1]$



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

$$\mathbf{x} = (\mathbf{e}_0\,,\, \mathbf{e}_1\,,\, \mathbf{o}) \cdot \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix}$$

Often $\mathbf{e}_0 = \begin{pmatrix} s_0 \\ 0 \end{pmatrix}$, $\mathbf{e}_1 = \begin{pmatrix} 0 \\ s_1 \end{pmatrix}$, $\mathbf{o} = \begin{pmatrix} -0.5 \cdot (N_0 - 1.0)s_0 \\ -0.5 \cdot (N_1 - 1.0)s_1 \end{pmatrix}$,

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

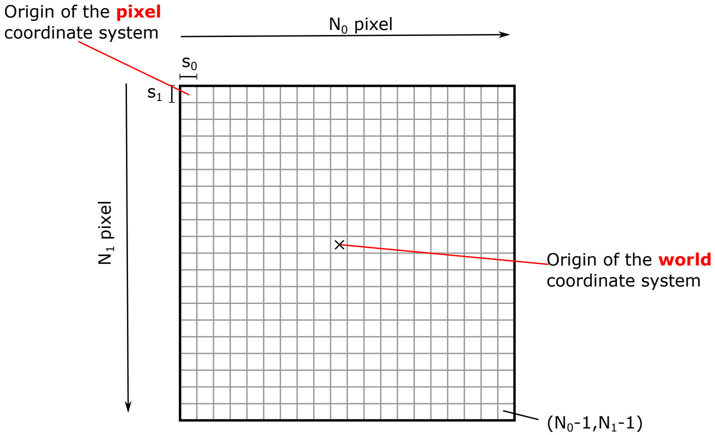
From world to image coordinates

If $\mathbf{e}_0 = \begin{pmatrix} s_0 \\ 0 \end{pmatrix}$, $\mathbf{e}_1 = \begin{pmatrix} 0 \\ s_1 \end{pmatrix}$ the inversion is easy:

World to image

$$\mathbf{p} = \begin{pmatrix} 1/s_0 & 0 \\ 0 & 1/s_1 \end{pmatrix} \cdot (\mathbf{x} - \mathbf{o})$$

Overview


Origin of the **pixel** coordinate system
N_0 pixel
s_0
s_1
N_1 pixel
Origin of the **world** coordinate system
(N_0-1,N_1-1)

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.
- Set the origin and spacing correctly when initializing your Grid.
- Do not forget the half-pixel shift.
- Once correctly set, use the member functions of your Grid to convert from the two coordinate systems into each other: **index_to_physical**(...) and **physical_to_index**(...).