# CONVENTION: Conversion between image and physical coordinates

Medical images, in addition to the 2D (3D) matrix of numbers, usually have a known pixel (voxel) size. The size is typically given in millimeters, and it might not be equal in both (all three) directions, i.e. pixels (voxels) might not be isotropic. The pixel (voxel) size allows one to determine the physical distance (in millimeters) between any two points in the image.

The indexes that determine the pixel (voxel) position in the image matrix are referred to as image coordinates. Being dimensionless integers, these coordinates do not provide information about physical sizes. For this reason one needs to define physical coordinates that determine point locations in the image in millimeters. In order to relate the image and physical coordinates, a convention that defines the conversion between the two is needed. Presented here is the convention adopted by the Biomedical Imaging and Modeling Laboratory (BIML). Identical convention is used in the Visualization Toolkit (VTK) for the image class (vtkImageData).

The conversion is done independently for each dimension. For a given dimension, let $i$ denote the image coordinate, $x$ the physical coordinate, $N$ the number of pixels, and $\Delta$ the pixel size. The physical coordinate and pixel size are expressed in millimeters, while the image coordinate and number of pixels are dimensionless. Although the image coordinate is an integer, here it is extended to real numbers, where strictly integer values correspond to pixel centers, and non-integer values to locations in between pixel centers. Note that this extended image coordinate still lacks information about physical sizes. The convention is that the image coordinate of the (center of the) first pixel is 1. This implies that the image coordinate of the (center of the) last pixel is $N$. The conversion between the image and physical coordinates is based on the convention that the center of the first pixel ($i = 1$) coincides with the origin of the physical coordinate axis ($x = 0$), that the image and physical coordinate axis are oriented in the same direction, and that the "pixel center to pixel center distance" is $\Delta$. This leads to a simple conversion formula

$$x = (i - 1)\Delta \tag{1}$$

Note that $x$, $i$, and $\Delta$ are real numbers. Thus, the center of the first pixel has coordinates $i = 1$ and $x = 0$, the center of the second pixel has coordinates

$i = 2$ and $x = \Delta$, ..., and the center of the $N$-th pixel has coordinates $i = N$ and $x = (N-1)\Delta$. This conversion can directly be implemented in a language which assumes that indexes start from 1 (e.g. Matlab). If the conversion is implemented in a computer language which assumes that indexes start from 0 (e.g. C/C++, Python), than the conversion formula needs to be modified to $x = i\Delta$. In any case, the center of the first pixel needs to be aligned with the origin of the physical coordinate axis.

Given here are conversion formulas for the 3D case. The first two equations from the 3D case define the conversion for the 2D case. Let $(i, j, k)$ denote the three image coordinates, $(x, y, z)$ the three physical coordinates, and $\Delta x$, $\Delta y$, and $\Delta z$ the voxel size in the three directions. Again, it is assumed that the image coordinates are real numbers, have integer values for voxel centers and non-integer values in between voxels, and are equal to 1 for the first voxel in their respective direction. Eq. 1 applied to the 3D case yields

$$x = (i-1)\Delta x, \qquad y = (j-1)\Delta y, \qquad z = (k-1)\Delta z.$$

Figure 1 illustrates the conversion in the 2D case. Note that for the pixel $(i, j)$, in physical coordinates, the center is at $x = (i-1)\Delta x$, $y = (j-1)\Delta y$, the left border is at $x = (i - \frac{3}{2})\Delta x$, the right border is at $x = (i - \frac{1}{2})\Delta x$, the bottom border is at $y = (j - \frac{3}{2})\Delta y$, and the top border is at $y = (j - \frac{1}{2})\Delta y$.

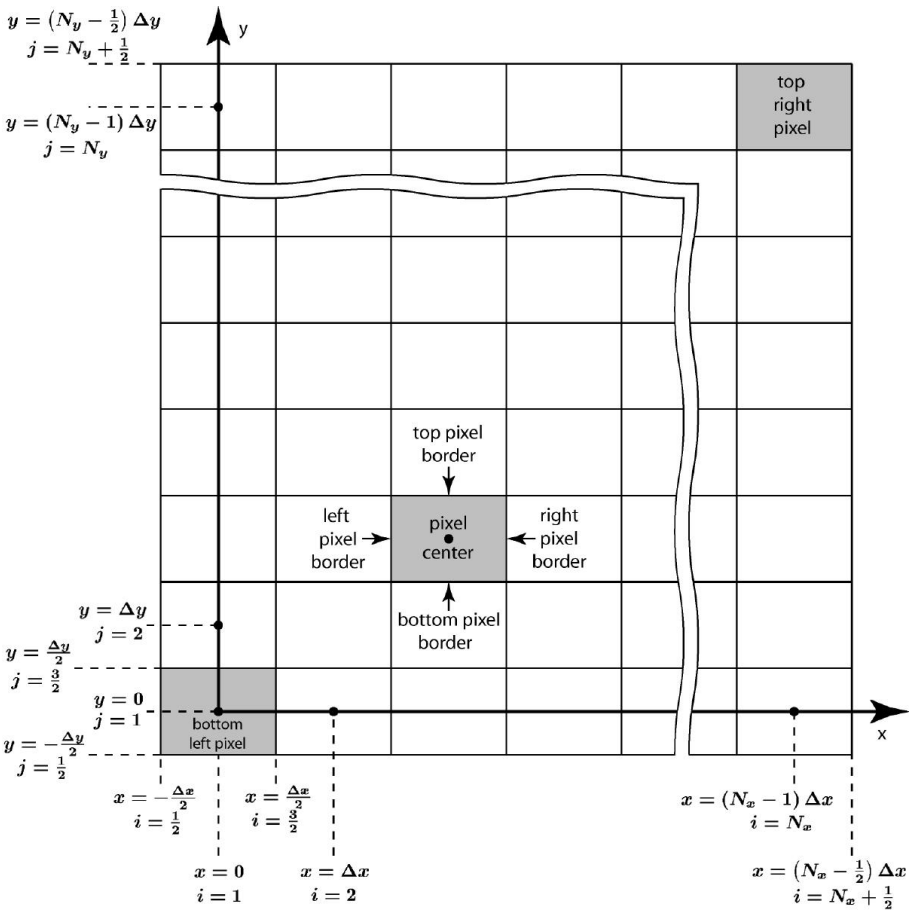


Figure 1: Conversion between image and physical coordinates in 2D: $(i, j)$ are image coordinates, $(x, y)$ are physical coordinates [mm], $N_x$ is the number of pixels in the x direction, $N_y$ is the number of pixels in the y direction, $\Delta x$ is the pixel size in the x direction [mm], and $\Delta y$ is the pixel size in the y direction [mm]. It is a BIML convention that the pixel (1,1) of a 2D image is displayed in the lower left corner, x axis is horizontal, and y axis is vertical.