On specification of the FABLE orientation matrix

One of the reoccuring issues then analysing diffraction data is to read the diffraction image and determine (it not known *a priori*) to orientation of the image. The image orientation is dependent on a number of things – read out electronics of the specific detector, detector software and beamline setup.

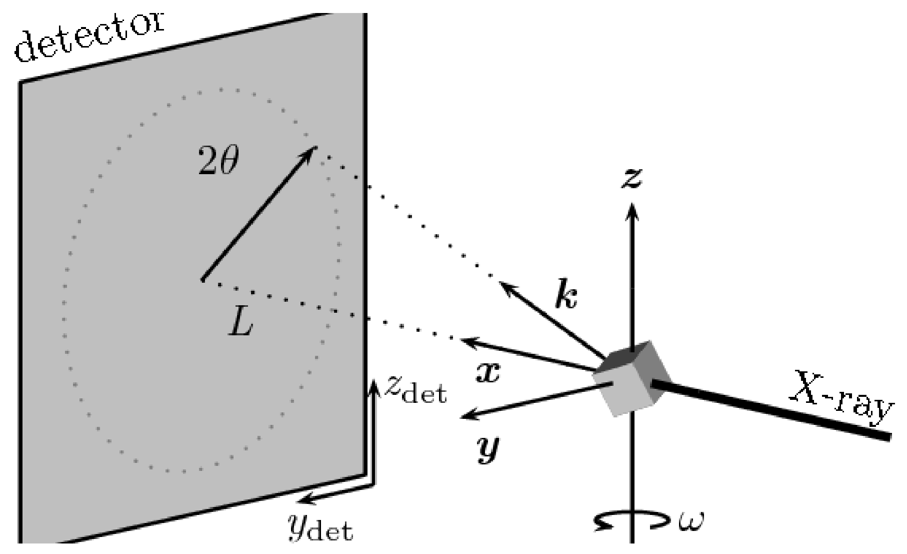
Using the FABLE ImageD11 peaksearch program the coordinates of the peaks are given as (f)ast,(s)low-coordinates. This is due to the way the pixel intensities are read from the diffraction image. The pixels are saved as one long string of binary numbers.

E.g. A 5 by 5 image

```
1,2,3,4,5,6,7,8,9,10,11,.....,25
```

this array will be made into a 5x5 image as

```
           FAST (f) ──▶
 SLOW   1,  2,  3,  4,  5
 (s)    6,  7,  8,  9, 10
  │    11, 12, 13, 14, 15
  ▼    16, 17, 18, 19, 20
       21, 22, 23, 24, 25
```



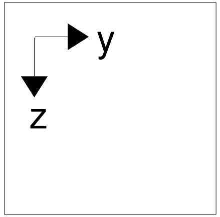
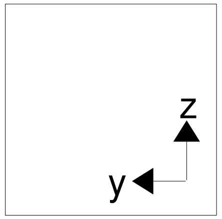
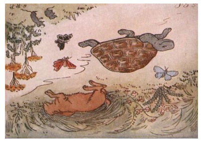
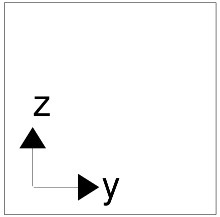
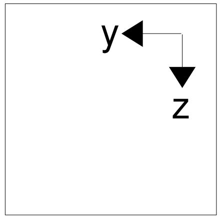
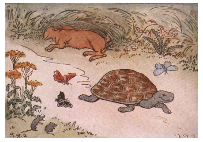
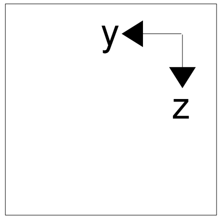
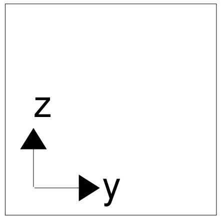
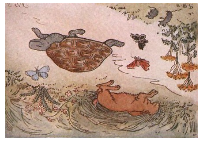
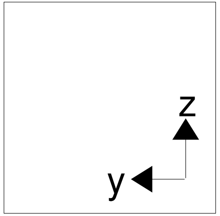
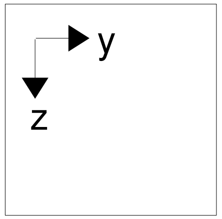
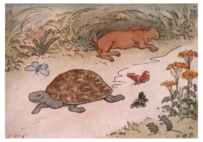
3DXRD geometry, below (y,z) = (y_{det}, z_{det})

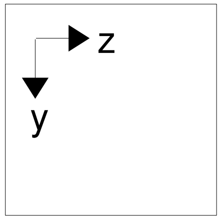
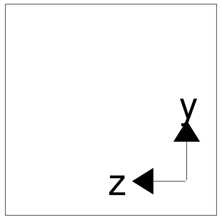
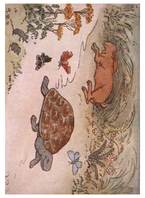
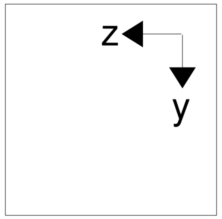
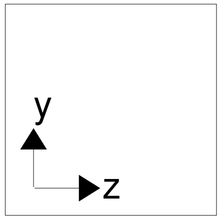
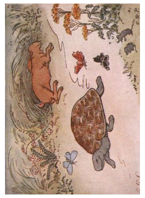
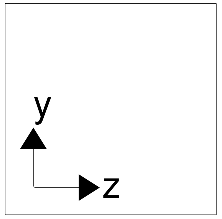
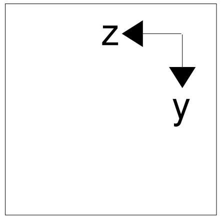
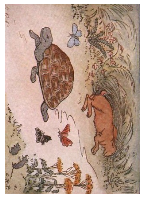
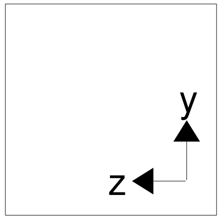
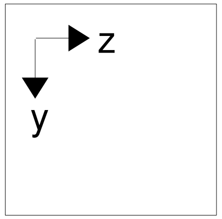
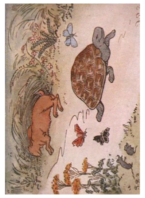
Orientations matrix to take f,s coordinates into the 3DXRD coordinates

$$\begin{bmatrix} o11 & o12 \\ o21 & o22 \end{bmatrix}\begin{bmatrix} s \\ f \end{bmatrix}=\begin{bmatrix} z \\ y \end{bmatrix}$$

Since the s,f and $z_{det,}$,y_{det} are defined between -0.5 and the maximum number of pixels along each direction. Negative values are added to total number of pixels

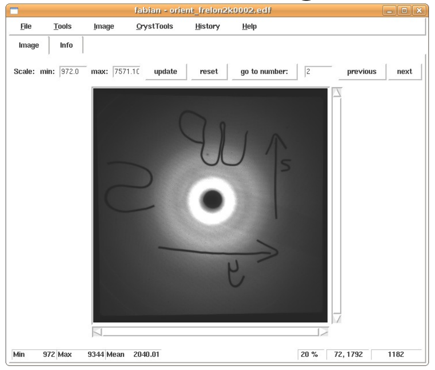
We have tried to make a scheme over the possible orientations and operations of transformation back to a standard setting can be used to determine the orientation matrix, o, used by the FABLE programs to transform peak positions etc.

Orientation matrix $\begin{bmatrix} o11 & o12 \\ o21 & o22 \end{bmatrix}$	Standard Image setting (0,0) top left	3DXRD setting (0,0) lower right (180° rotation)	Example picture after operating by O in the 3DXRD setting shown in the former column	Operation on coordinates to determine coordinates in 3DXRD setting (y,z) $\begin{bmatrix} o11 & o12 \\ o21 & o22 \end{bmatrix}\begin{bmatrix} s \\ f \end{bmatrix}=\begin{bmatrix} z \\ y \end{bmatrix}$	Transformation to make image in standard orientation – to read the image array as (z,y).	Transformation to make image in 3DXRD orientation. Image viewed as looking onto the detector along the beam
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$				$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} z \\ y \end{bmatrix}=\begin{bmatrix} z \\ y \end{bmatrix}$	none	fliplr+flipud
$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$				$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} -z \\ y \end{bmatrix}=\begin{bmatrix} z \\ y \end{bmatrix}$	flipud	fliplr
$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$				$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} z \\ -y \end{bmatrix}=\begin{bmatrix} z \\ y \end{bmatrix}$	fliplr	flipud
$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$				$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} -z \\ -y \end{bmatrix}=\begin{bmatrix} z \\ y \end{bmatrix}$	fliplr+flipud	none

Orientation matrix $\begin{bmatrix} o11 & o12 \\ o21 & o22 \end{bmatrix}$	Standard Image setting (0,0) top left	3DXRD setting (0,0) lower right (180° rotation)	Example image An image having the following orientations can be operated by O of that row	Operation on coordinates to determinecoordinates in 3DXRD setting (y,z) $\begin{bmatrix} o11 & o12 \\ o21 & o22 \end{bmatrix}\begin{bmatrix} s \\ f \end{bmatrix} = \begin{bmatrix} z \\ y \end{bmatrix}$	Transformation to make image in standard orientation – to read the image array as (z,y).	Transformation to make image in 3DXRD orientation. Image viewed as looking onto the detector along the beam
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$				$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} z \\ y \end{bmatrix}$	transpose	transpose + fliplr + flipud
$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$				$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} y \\ -z \end{bmatrix} = \begin{bmatrix} z \\ y \end{bmatrix}$	transpose+ flipud	transpose+ fliplr
$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$				$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} -y \\ z \end{bmatrix} = \begin{bmatrix} z \\ y \end{bmatrix}$	transpose+ fliplr	transpose+ flipud
$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$				$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} -y \\ -z \end{bmatrix} = \begin{bmatrix} z \\ y \end{bmatrix}$	transpose + fliplr + flipud	transpose

Example: If you have an image you can use fabian (or Fable ImageViewer) to display it. You can choose between the different possibilities for orientation matrix . When the image is transformed correctly it will look as if up is up left is left if you stand in front of the detector looking towards the detector (same direction as the beam). An example is shown below for the Frelon2k detector at ID11, there raw data has the orientaion matrix (1,0,0,-1). Some metal wire has been bent to show the y,z axes directions and put in front of the detector to determine its orientation.
OBS: If no transformation is made, i.e. orientation matrix is (1,0,0,1), the coordianates shown below the image is (f,s).

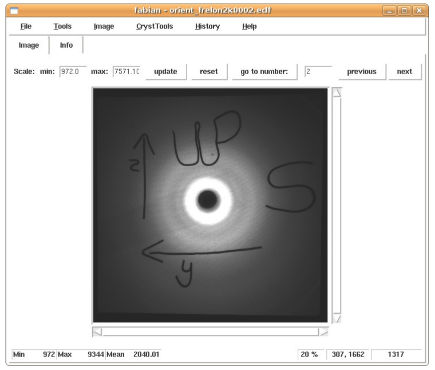
3DXRD setting



fabian - orient_frelon2k0002.edf
File Tools Image CrystTools History Help
Image Info
Scale: min: 972.0 max: 7571.1 update reset go to number: 2 previous next
Min 972 Max 9344 Mean 2040.01 20 % 72, 1792 1182

$$\xrightarrow{\hspace{3cm}}$$

Transform (1,0,0,-1)



fabian - orient_frelon2k0002.edf
File Tools Image CrystTools History Help
Image Info
Scale: min: 972.0 max: 7571.1 update reset go to number: 2 previous next
up
z
s
y
Min 972 Max 9344 Mean 2040.01 20 % 307, 1662 1317

Specific parameters used by ImageD11:

Regarding the beamcenter which is spefied in ImageD11, which is presently identified by the parameters *z_center*, *y_center.*
y_center is the beam center along the f coordinate (similarly y_size is the pixel size in along f)
z_center is the beam center along the s coordinate (similarly z_size is the pixel size in along s)

The 3DXRD detector coordinates defined in the figure above can then be calculated as follows.

$$\begin{bmatrix} z_{det} \\ y_{det} \end{bmatrix} = \begin{bmatrix} o11 & o12 \\ o21 & o22 \end{bmatrix} \begin{bmatrix} s - z_{center} \\ f - y_{center} \end{bmatrix}$$

Specific parameters used by PolyXSim/Fabric:

In these programs the beam center is defined using the 3DXRD standard parameters $y^0_{det,}z^0_{det}$ (keyword dety_center and detz_center, respectively). If you need to determine what these are – they can be calculated using the python function **xy2detyz** found in xfab.detector.

E.g. On the python command line

```
>from xfab import detector
>[dety_center,detz_center] = detector.xy2detyz([z_center,y_center],o11,o12,o21,o22,z_pixel,y_pixel)
```