

NikonDataReader_tmp

August 15, 2025

```
[1]: # -*- coding: utf-8 -*-  
# Copyright 2024 United Kingdom Research and Innovation  
# Copyright 2024 The University of Manchester  
#  
# Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.  
# You may obtain a copy of the License at  
#  
#     http://www.apache.org/licenses/LICENSE-2.0  
#  
# Unless required by applicable law or agreed to in writing, software  
# distributed under the License is distributed on an "AS IS" BASIS,  
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
# See the License for the specific language governing permissions and  
# limitations under the License.  
#  
# Authored by: Hannah Roberts (UKRI-STFC)
```

1 Load and visualise data with NikonDataReader

This how-to shows how to use the NikonDataReader to load data from Nikon .xtekct files and quickly visualise the data and geometry

Get the example dataset dataexample.KORN using download_data().

```
[2]: from cil.utilities import dataexample  
dataexample.KORN.download_data(data_dir='.', prompt=False)
```

Dataset folder already exists in .

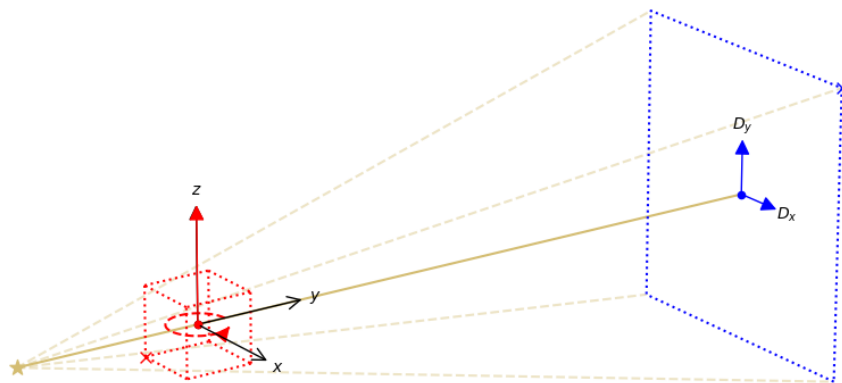
Now set up the NikonDataReader() to read an .xtekct file.

```
[3]: from cil.io import NikonDataReader  
file_name = 'korn/Korn i kasse/47209 testscan korn01_recon.xtekct'  
data_reader = NikonDataReader(file_name=file_name)  
data = data_reader.read()
```

Check the data has been loaded correctly by viewing the geometry with show_geometry() to display information about the source and detector setup.

```
[4]: from cil.utilities.display import show_geometry
show_geometry(data.geometry)
```

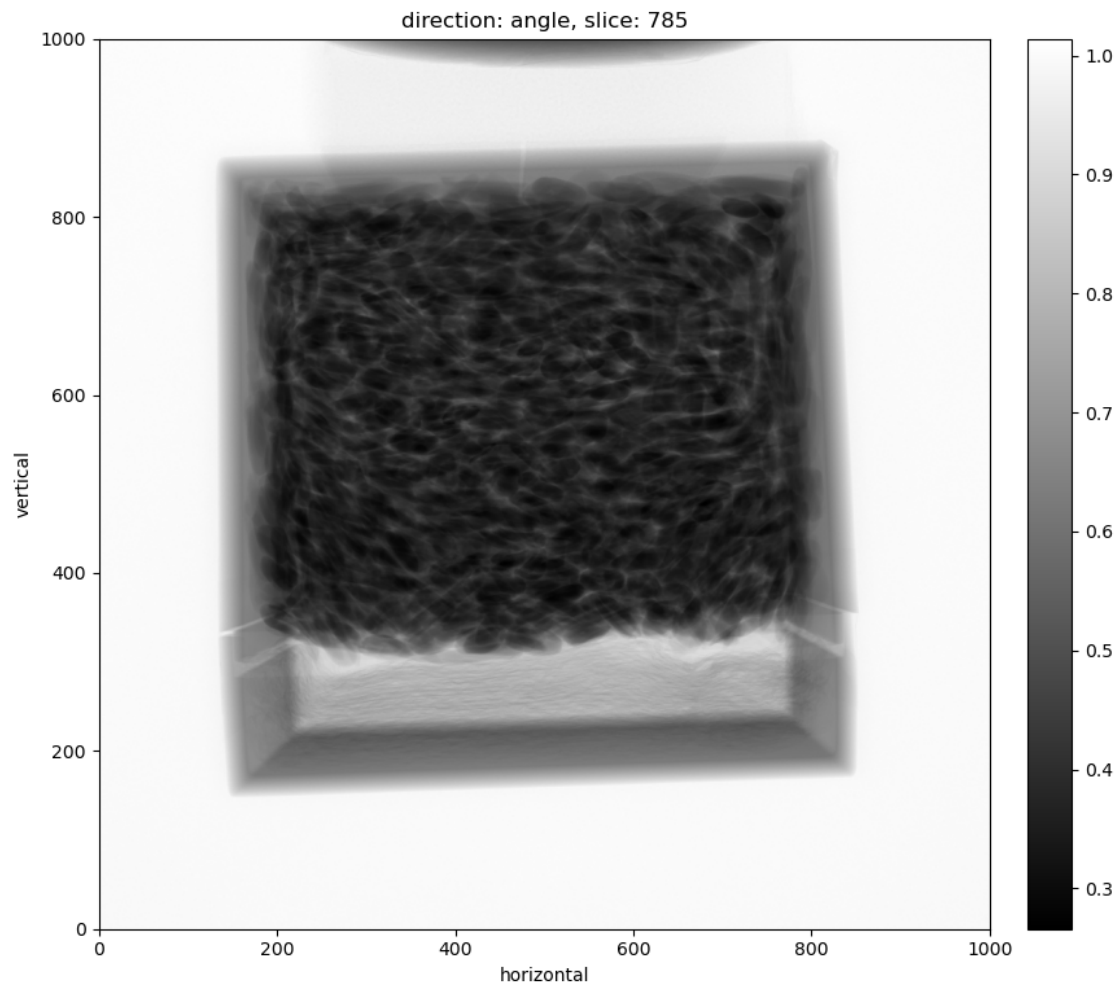
— world coordinate system	• rotation axis position	• detector position
★ source position	— rotation axis direction	— detector direction
	... image geometry	... detector
	× data origin (voxel 0)	× data origin (pixel 0)
	- - - rotation direction θ	



```
[4]: <cil.utilities.display.show_geometry at 0x7f72e5a6f470>
```

And view a central projection of the data with show2D()

```
[5]: from cil.utilities.display import show2D
show2D(data)
```

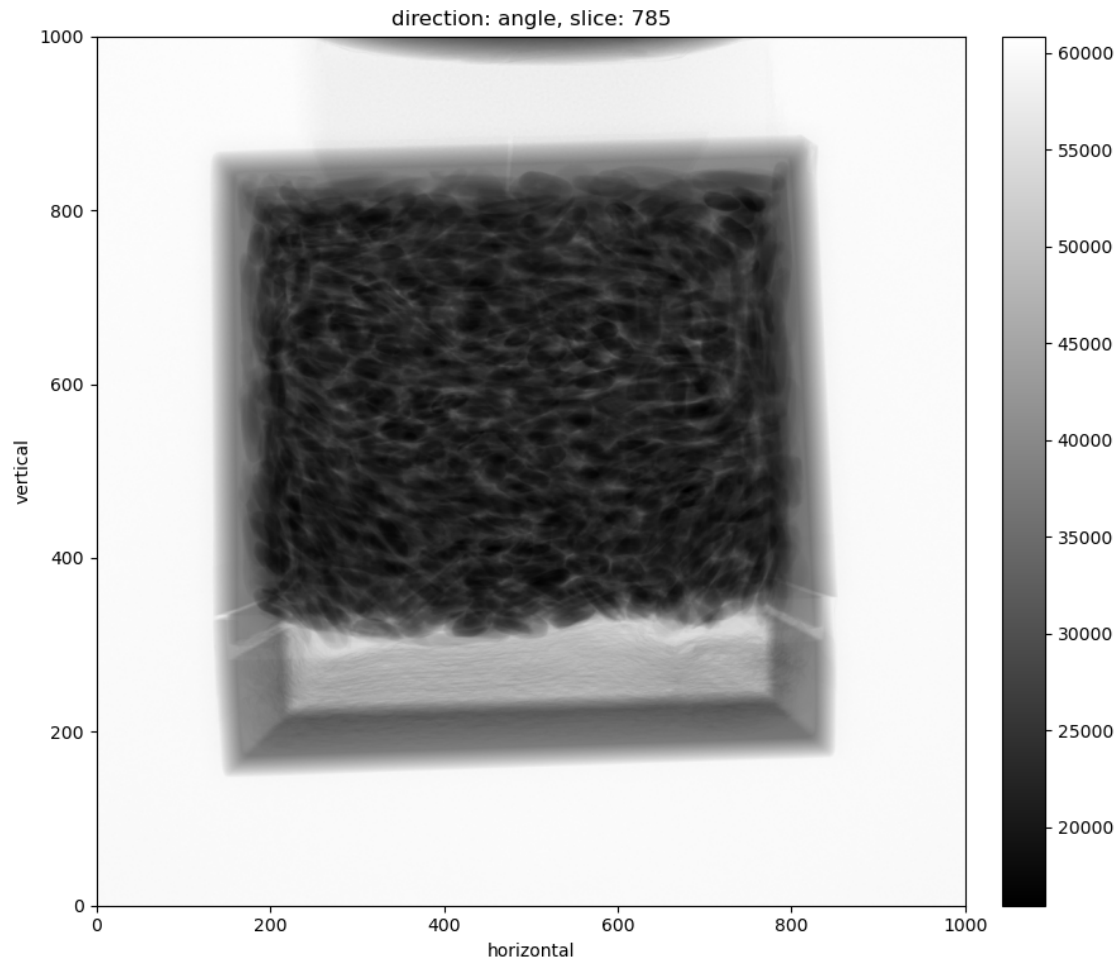


[5]: <cil.utilities.display.show2D at 0x7f71657a0d40>

By default the `NikonDataReader` argument `normalise` is `True`, which means all projections are loaded and normalised by the detector white level, which is stored in the `.xtekct` file as `WhiteLevel`. If you want to load the data without normalisation, specify `normalise=False`

```
[6]: file_name = 'korn/Korn i kasse/47209 testscan korn01_recon.xtekct'
      data_reader = NikonDataReader(file_name=file_name, normalise=False)
      data = data_reader.read()
```

```
[7]: show2D(data)
```

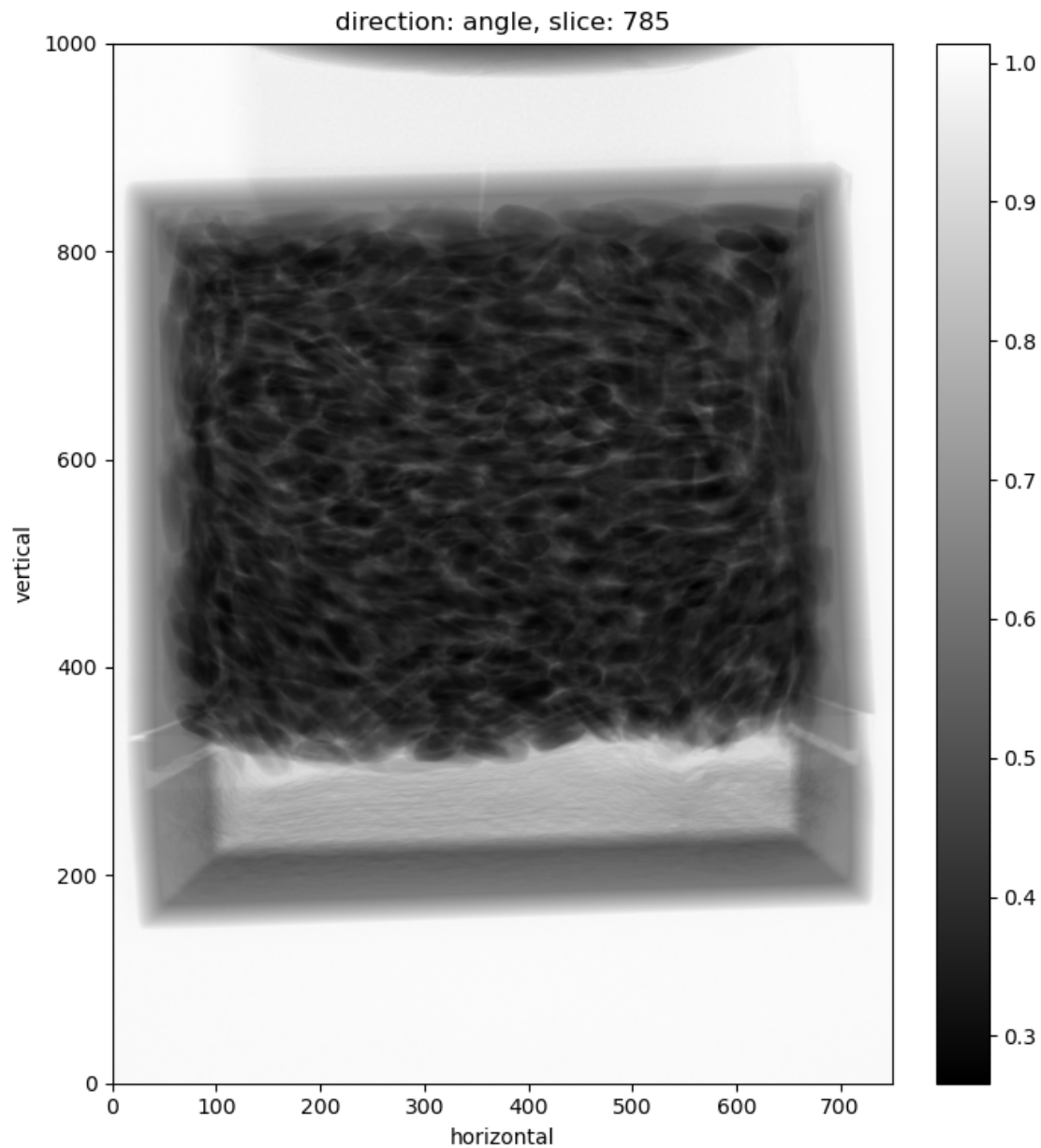


```
[7]: <cil.utilities.display.show2D at 0x7f7165006f00>
```

Use the `roi` argument to load a subset of the data. `roi` should be passed as a dictionary e.g. `{'axis_labels_1': (start, end, step), 'axis_labels_2': (start, end, step)}` with axis labels that describe the data dimension labels

To load a cropped subset of the data, change the start and end values. `'axis_label': -1` is a shortcut to load all elements along the axis.

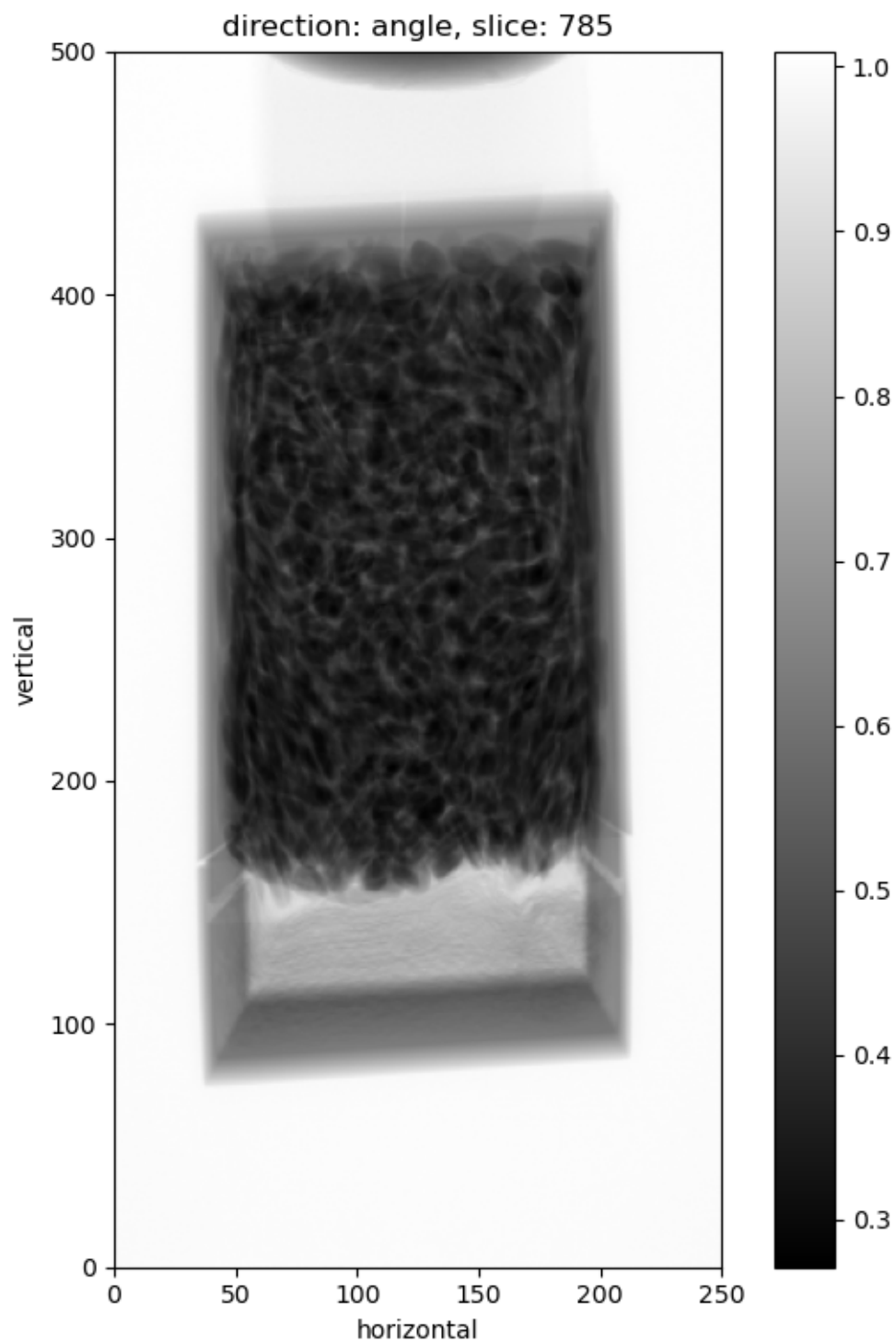
```
[8]: roi = {'horizontal': (120, 870, 1), 'vertical': -1}
data_reader = NikonDataReader(file_name=file_name, roi=roi)
data = data_reader.read()
show2D(data)
```



[8]: <cil.utilities.display.show2D at 0x7f71646e5910>

To load a binned subset of the data, change the step value. Here we use different binning for the horizontal and vertical dimensions which results in a different aspect ratio

```
[9]: roi = {'horizontal':(None, None, 4), 'vertical':(None, None, 2)}
data_reader = NikonDataReader(file_name=file_name, roi=roi)
data = data_reader.read()
show2D(data)
```

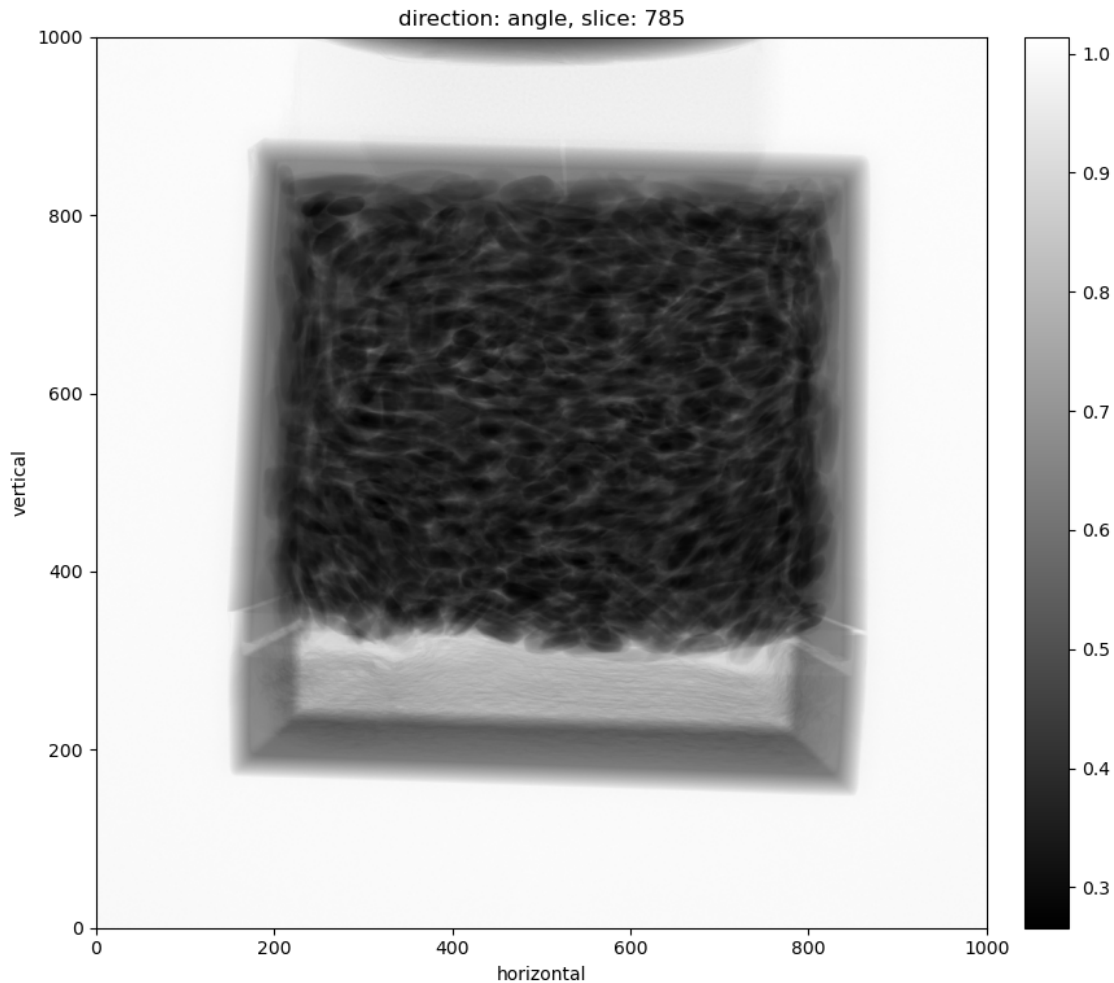


[9]: <cil.utilities.display.show2D at 0x7f71652fe900>

We can also use the argument `flipplr=True` to flip all projections in the vertical axis. If we enable

this option we see that the projection is flipped in the left-right direction/

```
[10]: data_reader = NikonDataReader(file_name=file_name, fliplr=True)
      data = data_reader.read()
      show2D(data)
```

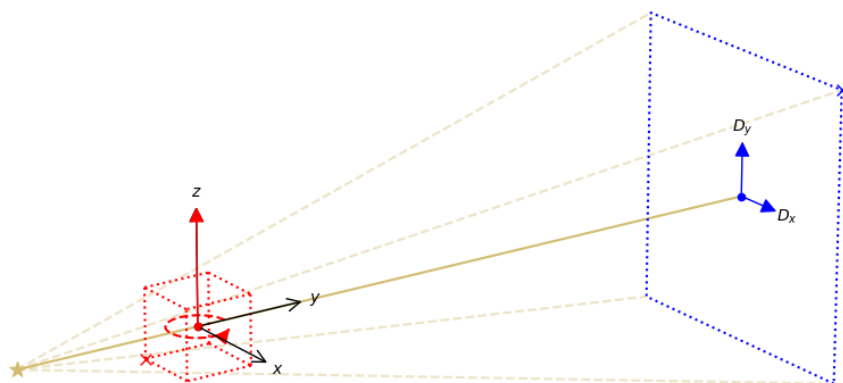


```
[10]: <cil.utilities.display.show2D at 0x7f71645b2db0>
```

Sometimes we might want to load the geometry from the Nikon file without loading the data, for example if we want to load the raw data files separately and use the geometry from the .xtekct file. We can do this by instantiating the data reader and calling `get_geometry()`

```
[11]: data_reader = NikonDataReader(file_name=file_name)
      ag = data_reader.get_geometry()
      show_geometry(ag)
```

— world coordinate system	• rotation axis position	• detector position
★ source position	— rotation axis direction	— detector direction
	⋯ image geometry	⋯ detector
	× data origin (voxel 0)	× data origin (pixel 0)
	- - - rotation direction θ	



```
[11]: <cil.utilities.display.show_geometry at 0x7f71c3a3f560>
```