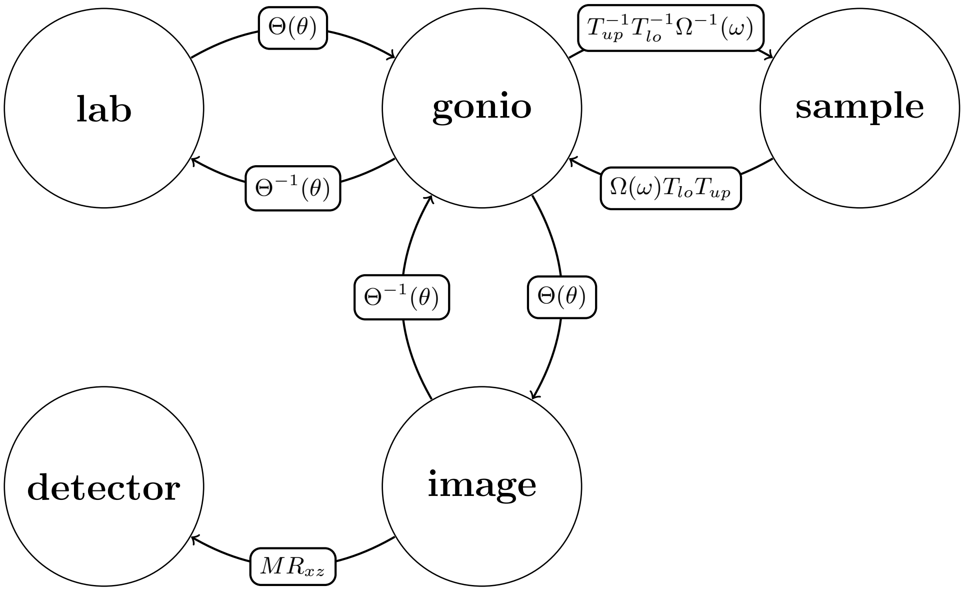
1 Coordinate systems for DFXRM

In the following document the coordinate transformations needed to reconstruct DFXRM data are presented. **The coordinate systems are chosen to all refer to the ID06 lab coordinate system, (x_l, y_l, z_l), defined with y_l along the incoming beam, z_l up and x_l to make right-handed.** The relationship between the lab system and the remaining coordinate systems is illustrated in Figure 1, where the transformation matrices are defined differently for the horizontal and vertical scattering geometries as given explicitly in the following sections.

The gonio system, (x_g, y_g, z_g), is related to the lab system by the base tilt, $\Theta(\theta)$, which makes the G-vector of a reflections with scattering angle 2θ parallel to the ω-rotation. Likewise, the image system, (x_i, y_i, z_i), is related to the gonio system by the same base tilt, $\Theta(\theta)$, making y_i parallel to the diffracted beam. The sample system, (x_s, y_s, z_s), is defined to coincide with the gonio system when the upper and lower tilt angles[1] and the ω-rotation around the G-vector are all zero. For lab, gonio, image and sample $(0, 0, 0)$ is defined where the rotation and tilt axes coincide (with the center of the incoming and diffracted beams).

Figure 1: Flow chart of coordinate transformations



[1] Could also be where the tilts maximise the scattered intensity, depends on whether we want to step relative to this or in absolute angles.

Regarding the detector system defined in Figure 1 we distinguish between the ideal detector system, $(x_d, 0, z_d)$, defined with the detector plane perpendicular to the diffracted beam with $(0, 0, 0)$ where the optical axis hits the detector, and the real detector system, $(x_r, 0, z_r)$ which has the same origo as the ideal detector system, but whose coordinate axes coincide with the lab system except for tilts t_x and t_z as defined in the following sections. Here we assume no rotation of the detector around the diffracted beam, $t_y = 0$.

In particular $y_d = 0$ and $y_r = 0$ everywhere on the detector. With a perfect lens system only the $(x_i, 0, z_i)$ plane can be imaged onto the detector, but in reality the opening angle of the lenses is such that all depths y_i will be projected onto $y_d = 0$ and $y_r = 0$.

While M is the magnification determined by the lens distance and detector pixel size, the flipping and tilting of the detector is contained within the matrix R_{xz}. For the ideal detector case where only the image flipping caused by the lens needs to be taken into account we have:

$$R_{xz} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

Thus following Figure 1 the entire forward projection from a point in the sample (x_s, y_s, z_s) to a point on the real detector $(x_r, 0, z_r)$ can be calculated:

$$\begin{pmatrix} x_r \\ 0 \\ z_r \end{pmatrix} = M \; R_{xz} \; \Theta(\theta) \; \Omega(\omega) \; T_{lo}(\phi_{lo}) \; T_{up}(\phi_{up}) \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix}$$

1.1 Horizontal scattering geometry

The horizontal scattering geometry and the associated coordinate systems are sketched in Figure 2. In the horizontal scattering geometry the detector is moved towards $-x_l$ and the base tilt is θ right-handed around z_l:

$$\Theta(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

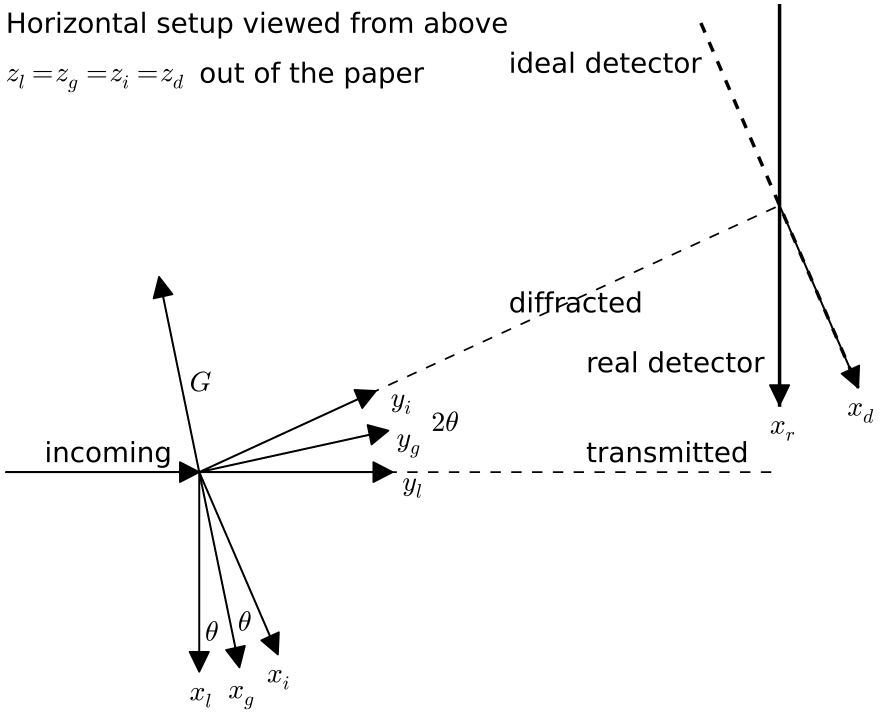
This means that the gonio system, (x_g, y_g, z_g), is defined with z_g up, x_g antiparallel to the G-vector, and y_g to make right-handed. Likewise, the image system, (x_i, y_i, z_i), is defined with y_i along the diffracted beam, z_i up and x_i to make right-handed.

Converting from the sample to the gonio system is done by applying first the upper tilt around z_s:

$$T_{up}(\phi_{up}) = \begin{pmatrix} \cos\phi_{up} & -\sin\phi_{up} & 0 \\ \sin\phi_{up} & \cos\phi_{up} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Then by applying the lower tilt around the y-axis of the tilted sample system:

Figure 2: Horizontal scattering geometry, viewed from above



$$T_{lo}(\phi_{lo}) = \begin{pmatrix} \cos\phi_{lo} & 0 & \sin\phi_{lo} \\ 0 & 1 & 0 \\ -\sin\phi_{lo} & 0 & \cos\phi_{lo} \end{pmatrix}$$

And finally rotating around the G-vector:

$$\Omega(\omega) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{pmatrix}$$

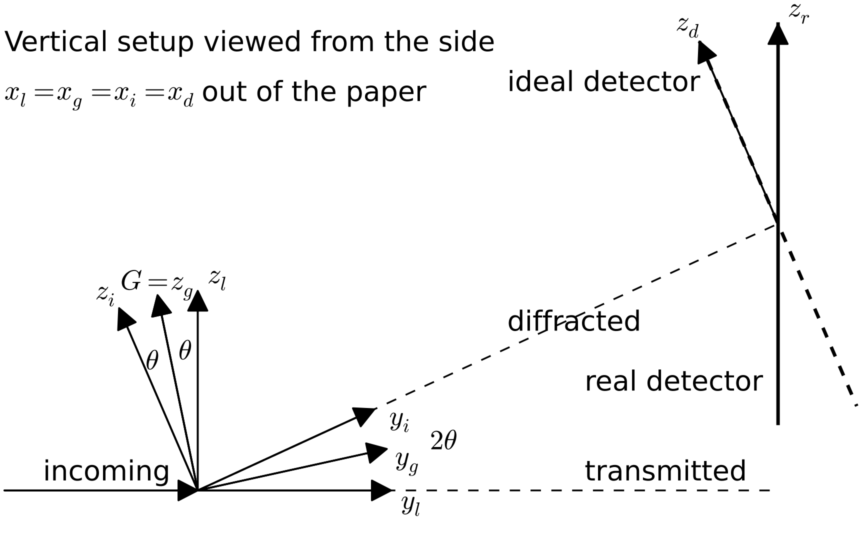
The detector tilt matrix for a non-ideal detector geometry is given by:

$$R_{xz} = \begin{pmatrix} -\frac{1}{\cos(t_z - 2\theta)} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

where t_z is the angular tilt of the detector around the z-axis (bringing x_l to coincide with x_r, positive clockwise around z). The detector tilts around the x- and y-axes are assumed to be insignificant.

1.2 Vertical scattering geometry

Figure 3: Vertical scattering geometry, viewed from the side



The vertical scattering geometry and the associated coordinate systems are sketched in Figure 3. In the vertical scattering geometry the detector is moved towards z_l and the base tilt is θ right-handed around x_l:

$$\Theta(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

This means that the gonio system, (x_g, y_g, z_g), is defined with z_g along to the G-vector, x_g along x_l, and y_g to make right-handed. Likewise, the image system, (x_i, y_i, z_i), is defined with x_i along x_l, y_i along the diffracted beam and z_i to make right-handed.

Converting from the sample to the gonio system is done by applying first the upper tilt around ??:

$$T_{up}(\phi_{up}) = ??$$

Then by applying the lower tilt around the ??-axis of the tilted sample system:

$$T_{lo}(\phi_{lo}) = ??$$

And finally rotating around the G-vector:

$$\Omega(\omega) = \begin{pmatrix} \cos\omega & -\sin\omega & 0 \\ \sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The detector tilt matrix for a non-ideal detector geometry is given by:

$$R_{xz} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\cos(t_x - 2\theta)} \end{pmatrix}$$

where t_x is the angular tilt of the detector around the x-axis (bringing z_l to coincide with z_r, positive clockwise around x). The detector tilts around the y- and z-axes are assumed to be insignificant.

1.3 Non-intersecting rotation axes

The above definitions of the transformation matrices builds on the assumption that the tilt and rotation axes are perpendicular to one another and that they all have a common focus point where the incoming beam intersects the ω-axis. However, from the setup at ESRF ID06 it is well-know that the focus points of the ϕ_{up} and ϕ_{lo} tilts are offset along the G-vector, hence we need to take this into account. The following generalisation still assumes the axes to be perpendicular, but allows the focus points of the three tilts ϕ_{up}, ϕ_{lo} and θ to be offset to the points (x_{up}, y_{up}, z_{up}), (x_{lo}, y_{lo}, z_{lo}) and $(x_\theta, y_\theta, z_\theta)$, respectively. The entire forward projection then becomes:

$$\begin{pmatrix} x_r \\ 0 \\ z_r \\ 1 \end{pmatrix} = M\, R_{xz}^{4\times4}\, \Theta^{4\times4}(\theta)\, \Omega^{4\times4}(\omega)\, T_{lo}^{4\times4}(\phi_{lo})\, T_{up}^{4\times4}(\phi_{up}) \begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix}$$

where

$$T_{up}^{4\times4}(\phi_{up}) = \begin{pmatrix} 1 & 0 & 0 & x_{up} \\ 0 & 1 & 0 & y_{up} \\ 0 & 0 & 1 & z_{up} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} & & & 0 \\ & T_{up}(\phi_{up}) & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -x_{up} \\ 0 & 1 & 0 & -y_{up} \\ 0 & 0 & 1 & -z_{up} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{lo}^{4\times4}(\phi_{lo}) = \begin{pmatrix} 1 & 0 & 0 & x_{lo} \\ 0 & 1 & 0 & y_{lo} \\ 0 & 0 & 1 & z_{lo} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} & & & 0 \\ & T_{lo}(\phi_{lo}) & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -x_{lo} \\ 0 & 1 & 0 & -y_{lo} \\ 0 & 0 & 1 & -z_{lo} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Omega^{4\times4}(\omega) = \begin{pmatrix} & & & 0 \\ & \Omega(\omega) & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Theta^{4\times4}(\theta) = \begin{pmatrix} 1 & 0 & 0 & x_\theta \\ 0 & 1 & 0 & y_\theta \\ 0 & 0 & 1 & z_\theta \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} & & & 0 \\ & \Theta(\theta) & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -x_\theta \\ 0 & 1 & 0 & -y_\theta \\ 0 & 0 & 1 & -z_\theta \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_{xz}^{4\times4} = \begin{pmatrix} & & & 0 \\ & R_{xz} & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2 Implementation

A python implementation has been made to simulate dark field images from a grain structure and reversely reconstruct a grain structure from a series of images. Both of the algorithms builds a forward projection routine relating the sample and detector coordinate systems as described in the previous section.

2.1 Forward projection lookup table

The fastest way perform the calculations behind the forward projection is to do this once for each combination of the angles θ, ω, ϕ_{lo} and ϕ_{up} and then build the resulting transformation matrix for each of these into a lookup table $T(\theta, \omega, \phi_{lo}, \phi_{up}) = MR_{xz}\Theta(\theta)\Omega(\omega)T_{lo}(\phi_{lo})T_{up}(\phi_{up})$ that can be used to directly go from (x_s, y_s, z_s) to $(x_r, 0, z_r)$. This functionality is implemented in forward_projection.build_rotation_lookup for both horizontal and vertical scattering geometry.

2.2 Grain map defined in terms of tilts above rotation

In this geometry the image data structure is defined as I(ϕ_{up},ϕ_{lo},ω,detx_size,detz_size), while the grainmap is saved in two variables, grain_xyz for positions (x,y,z) given in **NB! Need to define units for xyz** and grain_ang for angles (ϕ_{up},ϕ_{lo}) given in degrees.

The philosophy of image simulation and reconstruction is that the pair (ϕ_{up},ϕ_{lo}) is used to define a direction of the scattering vector, which we then rotate around while scanning ω. Thus the pair (ϕ_{up},ϕ_{lo}) that optimises the diffracted intensity from a given voxel in the sample for all values of ω will be the orientation assigned to that voxel.

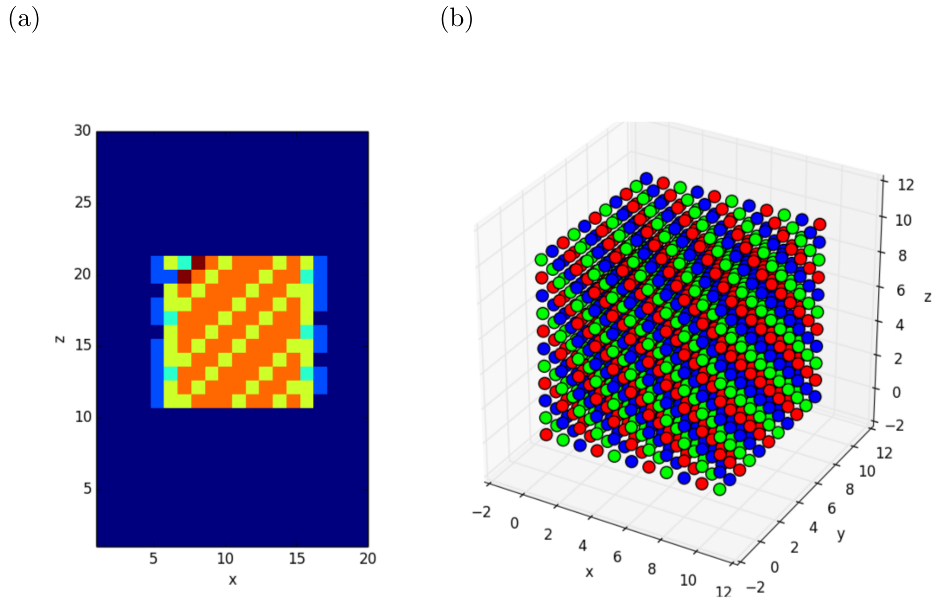
2.2.1 Simulating images

Simulating a set of images in the tilt above rotation geometry is straightforward:

```
for each (x_s,y_s,z_s,phi_up,phi_lo) in grainmap:
    for each (omega) measured:
        (x_r,0,z_r) = T(theta,omega,phi_lo,phi_up) (x_s,y_s,z_s)
        I(phi_up,phi_lo,omega,x_r+detx_cen,z_r+detz_cen) += 1
```

As seen above we simply assume that each voxel, when aligned correctly to fulfil the diffraction condition, adds the same intensity to the corresponding pixel on the detector. This is to some extent a relatively crude discretisation given (among other things) by the chosen voxel size in the sample geometry and the neglect of detector point spread. An example of a structure and the corresponding simulated image can be seen in Figure 4.

Figure 4: Simulated image (a) from a striped grain structure (b) in horizontal scattering geometry (rotation around x). The simulated image (a) corresponds to the signal from the blue orientation in (b).



2.2.2 Reconstructing sample

As can be seen from Figure 4 the diffracted intensity hitting the individual pixels of the detector carry important information about the structure of the sample, and in this particular case it is relatively difficult to set a universal intensity threshold and only work with binary detector images, 0=background, 1=diffraction signal. Hence, for the most general version of the reconstruction code it was decided to use the intensity of the detector pixels hit by forward projecting the diffracted signal from the individual voxels in the sample as a probability indicator, $p(\phi_{up}, \phi_{lo})$, of the fit quality in the following way:

```
for each (x_s,y_s,z_s) in grainmap:
    p(phi_up,phi_lo) = 0
    for each (phi_up,phi_lo,omega) measured:
        (x_r,0,z_r) = T(theta,omega,phi_lo,phi_up) (x_s,y_s,z_s)
        p(phi_up,phi_lo) += I(phi_up,phi_lo,omega,x_r+detx_cen,z_r+detz_cen)
    grain_ang(x_s,y_s,z_s) = center of mass of (phi_up,phi_lo)
```

Doing many projections (steps in ω) helps on the fact that each detector pixel is not only hit by the diffracted signal from one voxel in the sample, but from all voxels lying along y_i, which vary depending on ω. The summation over ω of diffracted intensities in each detector pixel hit by the forward projection from a given sample voxel reflects exactly the assumption that we are only located at the correct orientation (ϕ_{up}, ϕ_{lo}) if the corresponding scattered intensity is high for all values of ω, and if we have many ω-values a high probability $p(\phi_{up}, \phi_{lo})$ is not likely to occur based on coincidental overlap with other parts of the sample along y_i. Furthermore the error bars on the orientations can be estimated as the variance of $p(\phi_{up}, \phi_{lo})$.

2.2.3 Code optimisation

A number of tricks can be played with the code compared to the above-mentioned overview in order to limit the memory usage and speed up the reconstruction process. The main thing is to avoid looping by vectorizing the calculations as much as possible, and to make simplifying assumptions whenever they make sense. How to choose between the different simplification as well as some of the implications for the output are summarised in the following.

Binarizing data. While it is difficult to set a a universal intensity threshold for simulated data like the limiting case depicted in Figure 4, where structural features extend over individual pixels, this is usually somewhat easier for typical experimental maps of grains with orientation gradients or internal domains, where the features extend over several pixels. In the code it is possible to set the intensity threshold and turn the binarization on or off. **input options, default is ??**

Completeness. With binarization of the input data, the maximum intensity in $p(\phi_{up}, \phi_{lo})$ is the number of ω steps, and the completeness of the solution can be directly given as the ratio between the actual value of $p(\phi_{up}, \phi_{lo})$ divided by the number of ω steps. The completeness measure can be used to cut off the reconstruction at a certain value (the default is intensity_cutoff 0.5) in order to define the grain boundaries. Giving a complimentary measure of the completeness with binarization off is difficult – suggestions? Here perhaps more about interpolation in $p(\phi_{up}, \phi_{lo})$ to determine the completeness

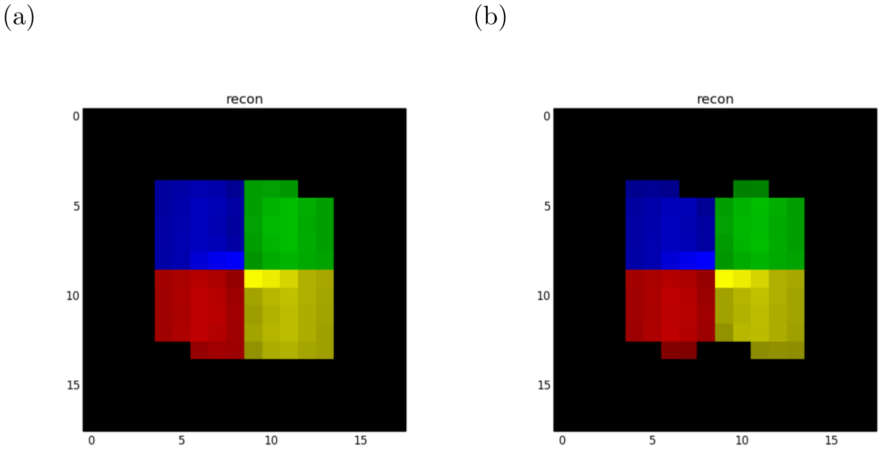
Distributing on multiple cores. Binarizing the data means that it fills up much less memory space, hence we can work with several identical copies and have different cores work on separate layers of the reconstruction simulateneously by accession their own separate copies of the binarized data. **input options, default is ??**

Limiting the number of coordinate transformations. The time limiting step in the calculation is the coordinate transformation from (x_s, y_s, z_s) to $(x_r, 0, z_r)$. Realising that for most samples the scanned ranges in ϕ_{up} and ϕ_{lo} are so narrow ($< 1^\circ$) that we can approximate $T_{up}(\phi_{up}) \approx T_{up}(\overline{\phi_{up}})$ and $T_{lo}(\phi_{lo}) \approx T_{lo}(\overline{\phi_{lo}})$, where $\overline{\phi_{up}}$ is the mean value of ϕ_{up} and $\overline{\phi_{lo}}$ the mean of ϕ_{lo}, we only need to perform the coordinate transformation for each value of ω at each reconstructed voxel. This reduces the number of coordinate transformations by a factor of the number of steps in ϕ_{up} times the number of steps in ϕ_{lo}, usually of the order 20×20. While this simplifications may shift the forward projection a few pixels on the detector, which is mostly irrelevant for real structures expanding over a substantial area on the detector, the reconstruction will not suffer since it build on the intensity of the given voxel in the image measure at the specific values of ϕ_{up} and ϕ_{lo}. **input options, default is ??**

2.2.4 Testing the effect of detector tilt

In order to test the effect of a wrong tilt of the detector a reconstruction of a simulated structure using the correct ideal detector position perpendicular to the diffracted beam, or a detector perpendicular to the incident beam (off-set of $2\theta = 21^\circ$). As can be seen this corresponds to the reconstruction becoming disproportionate in size by a factor of $\cos 2\theta = 0.93$. From this it seems fair to conclude that if the tilt is good within 2° (which should be achievable when setting up the equipment), the shape of the reconstruction is off by less than 0.1%, hence detector tilt issues it not a main concern for this type of investigations.

Figure 5: Reconstructed slice of a simulated structure consisting of 4 different orientations of $5 \times 5 \times 5$ voxels each. The reconstructions are shown with a minimum completeness of 50% and assuming either the correct ideal detector tilt (a) or an extreme tilt off-set of $2\theta = 21°$ (b) corresponding to a detector perpendicular to the incident beam.



2.3 Grain map defined in terms of rocking curve scans

In this special case, the image data structure is defined as I(θ,ω,detx_size,detz_size), while the grainmap is saved in two variables, grain_xyz for positions (x,y,z) given in **NB! Need to define units for xyz** and grain_ang for angles (ω,θ) given in degrees.

Now assume that for the rotation angle ω_0 we observed the maximum diffracted intensity from a given voxels at a tilt of $\theta+\Delta\theta_0$. Then the maximum intensity at $\omega_0+180°$ will be observed at $\theta-\Delta\theta_0$, and the function describing the maximum intensity for any value of ω is defined as:

$$\Delta\theta(\omega) = \Delta\theta_0 \cos(\omega - \omega_0),$$

The pair $(\omega_0, \Delta\theta_0 > 0)$, corresponding to the largest $\Delta\theta$ over the entire ω-range, can now be used to uniquely define the orientation of each voxel in the reconstructed sample.

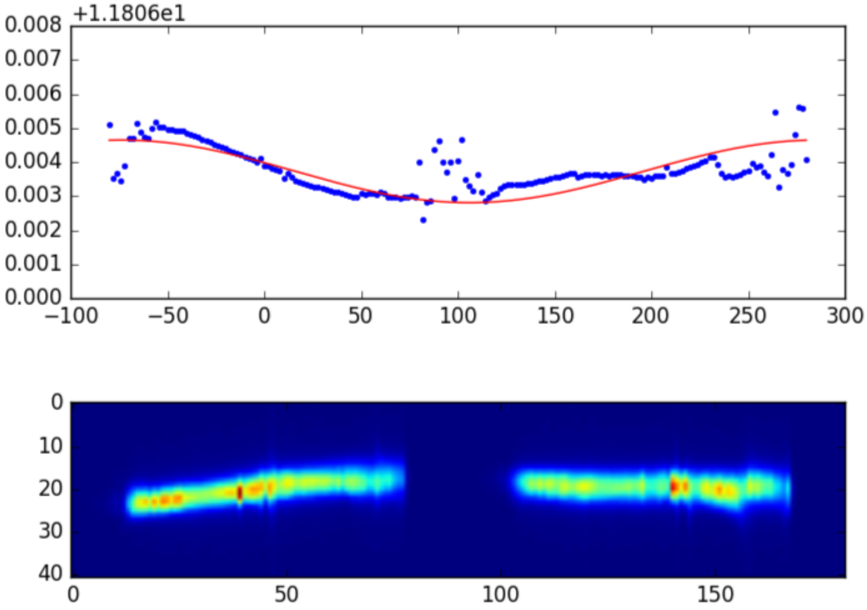
2.3.1 Reconstructing sample

As outlined above we are searching for the position ω_0 and amplitude $\Delta\theta_0$ of a cosine curve in (ω, θ)-space for each voxel in the sample. The code to do this has the following logic:

```
for each (x_s,y_s,z_s) in grainmap:
    p(omega,theta) = 0
    for each (omega,theta) measured:
        (x_r,0,z_r) = T(theta,omega,0,0) (x_s,y_s,z_s)
        p(omega,theta) += I(theta,omega,x_r+detx_cen,z_r+detz_cen)
    COM of p for each omega to find Delta_theta(omega)
    cosine fit Delta_theta(omega) to find omega_0 and Delta_theta_0
    grain_ang(x_s,y_s,z_s) = (omega_0,Delta_theta_0)
```

The above-mentioned cosine fit is weighted with the amplitudes of the individual gauss fits such that points with low diffracted intensities in the $p(\omega, \theta)$ probability space are down-weighted in the determination of $(\omega_0, \Delta\theta_0)$.

Figure 6: Cosine fit of maximum intensity for a given voxel as a function of ω, θ to determine $(\omega_0, \Delta\theta_0)$. NB bottom image flipped vertically compared to top curve fit. The fit of the cosine curve is weighted by the scattered intensity.



2.4 File formats

2.4.1 Input parameters

2.4.2 Image formats

2.4.3 Output grain map