**FILE STRUCTURE**

Create a folder for the case you are working on, housed within the date and condition the SEM images were taken.

Ex. 2023-07-18 / 30Pa / case 1.0 (calibration)

case 1.1 (roughness)

This folder should contain:

* SEM images in which the .bmp and .txt files for each detector for that case are found
* GNBF\_1\_makextlvecs
* GNBF\_2\_calibrate\_from\_Xtlvecs
* GNBF\_3\_retrieve.ipynb
* GNBF\_4\_flattenandfilter.ipynb
* GNBF\_5\_analyzeroughness\_of\_flattened\_npz.ipynb
* GNBF\_6\_grid2stlwskirt.ipynb if you want to create a .stl file
* Boxes.nml
* Segments1.nml

and more that will be created/overwritten as you execute code

**CALIBRATION**

**GNBF\_1\_makextlvecs**

Inputs: Images of a smooth ice crystal Output: Xtlvecs.nml (nameless with vectors)

1. **Input dataset**

Set the file being read in by changing the name after imageroot.

ims.getc2 reads in .bmp and .txt file and records dx,dy,cA,cB,cC,cD,Filename

dx = dy = pixel size (recorded in micrometers)

To change the detector image being viewed, add the preferred detector letter after imageroot. Detectors B and C tend to have better contrast making it easier to see the distinct edges between planes but this varies depending on crystal orientation.

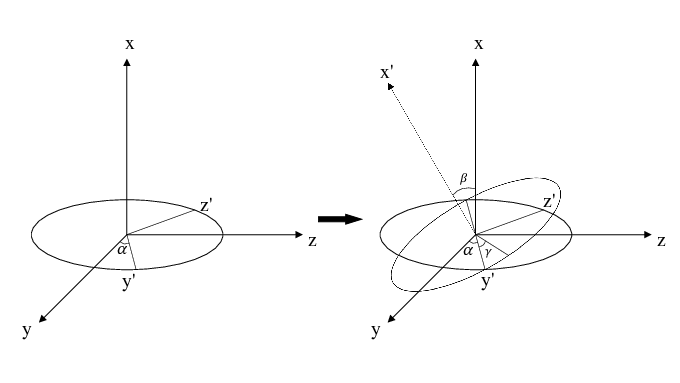
Ex. (for detector C):

dx,dy,cA,cB,cC,cD,Filename = ims.getc2('', 'SEMimages/',

imageroot,'C')

1. **Adjust Euler angles**

Euler angles create a new right-handed coordinate system (x’, y’, z’) describing the crystal’s position. In the original coordinate system (x, y, z), +x is to the right +y is down +z is into the plane xyz form a right-handed coordinate system.



**x’: bvec**

**y’: cvec** perpendicular to basal plane

**z’: avec**

Using the sliders, we want to vary xpos, ypos, alpha, beta, and gamma, such that

* The origin (xpos & ypos) put the center on a corner of the crystal
* **cvec** points along the c-axis of the crystal (perpendicular to basal plane)
* **bvec** points along the boundary between the basal and one pyramidal/prismatic
* **avec** points along the boundary between the basal and another pyramidal/prismatic



Order of moving sliders:

* 1. Move xpos & ypos bars into position
  2. Adjust gamma until blue lines up with a c-axis
  3. Adjust alpha and beta until red and green line up

1. **Run all**

**GNBF\_2\_calibrate\_from\_Xtlvecs**

Input: Xtlvecs.nml Output: calibration.nml

1. **Input dataset** (repeat step 1)
2. **Draw boxes on each plane**

It’s often helpful to open the images on your computer so you can see a clearer view and switch between detectors as you are deciding to draw the boxes.

The location and size of the boxes is specified in Boxes.nml

* The coordinates give the location of the upper left corner of the boxes. They are listed in the order a, b, c, where the last box in the list is for the basal plane (box c).
* Check that the drawn boxes are fully on the plane surfaces, including in the view of detector A and D, especially if there was some growth between images.

1. **Comment out and keep the code appropriate for the crystal planes being used**

* The colors are shaded such that a is the lightest and c is the darkest, with normal vectors color coded to correspond to the correct box
* Set cvecdir. This is usually pointing down, such as in the screenshot above
  + The direction of cvec also determines whether the pyramidal is rotated +28 or -28 degrees, as seen in printed comment.
* Set boxa to prismatic or pyramidal, corresponding to where the box is drawn
* Set boxb to prismatic or pyramidal

The prismatic normal is calculated by taking the cross product of cvec and either avec or bvec. The pyramidal normal uses the prismatic normal and rotates it 28 degrees. Make sure that drawn vectors appear to be normal to the chosen facets. If they are not try troubleshooting:

* + Check the sign of the 28 degree pyramidal rotation
  + Check the vector orientation for every detector image. Sometimes growth means that the xtlvecs get off slightly
  + Return to the makextlvecs code and adjust accordingly. Small changes can go a long way so returning to adjust these as you are drawing the normals is often necessary. (ex. If the prismatic normal should be angled more to the left, eyeball the cross product of cvec and a/bvec to angle more in this direction as you adjust)

Note: Box c should ideally go on the basal. If this is impossible, it is possible to calibrate with 3 different surfaces (such as 3 prismatic surfaces). This will require different code however, see 2023-07-18 / 30Pa / case1.0 (calibration) for example.

1. **Check calibration plot**

If the best fit lines are well lined up with each detector, calibration has been done well. A perfect fit is rare, but if there are big divets/outliers in the fit it’s worth repeating troubleshooting steps.

1. **Run all**

* Close and halt any code still running

**GNBF\_3\_retrieve.ipynb**

Note: If local imports fail, you may need to modify sys.path.append('../../../../') with the correct number of dashes depending on file structure

1. **Input dataset**

Set imageroot to correct case

Set Segmentnamelistroot to desired segments file (ex. Segments1)

1. **Draw segments**

Open Segments1.nml (or another iteration of this file) and draw segments such that they pass over a defined edge. Check that at least an entire box is in both of the chosen facets in every detector, you may need to slightly adjust box size (usually use about 40x40) or number of boxes (usually 1x4). This will be important for confirming angles.

1. **Run all**

* As the code loops to retrieve the segment, check the last number under di2. The code is set up to reduce this error by performing multiple iterations. It will stop and move on to the next segment when it has reached the max number (maxiter = 5) or the error has dropped below the tolerance (tolerance = 10).
* Close and halt

**GNBF\_4\_flattenandfilter.ipynb**

1. **Run all**
2. **Check angles to validate calibration**

The printout after the last cell gives the angles between boxes.

Ex. Box 0 on pyramidal, box 3 on prismatic.

0 1 0.9712001262087455 13.784174809233185

0 2 0.904515937231182 25.24183334751339

0 3 0.8852479914776505 27.717945324477146

1 2 0.9797758224061974 11.542708599512823

1 3 0.9704160043032377 13.97148539069795

2 3 0.9990146932992978 2.5436597286111464

Look at the second number between the boxes located squarely on the surfaces you are comparing, here it is highlighted, very close to 28 degrees making this an excellent calibration. (You don’t always get this close but within 8 degrees or so is reasonable).

Expected angles:

* + 28 pyramidal to prismatic
  + 62 pyramidal to basal
  + 60 prismatic to prismatic

That’s it for calibration! The calibration.nml file will be used for subsequent roughness analysis