EUROPEAN
SPALLATION
SOURCE

DMSC Update

PRESENTED BY SOEREN SCHMIDT

2021-04-15

Update on activities regarding ODIN

- IDS deliverables for ODIN in order to coordinate among the first instruments across DMSC
  - Documentation of data reduction workflow
  - Commissioning plans (data analysis)
  - Milestones for DMSC deliverables
- Weekly meetings with ODIN team and ECDC
- Nexus imaging format: Task Force just had first meeting, next one in approx. three weeks.
- YMIR, Light Tomography with ECDC and LU.

PAUL SCHERRER INSTITUT

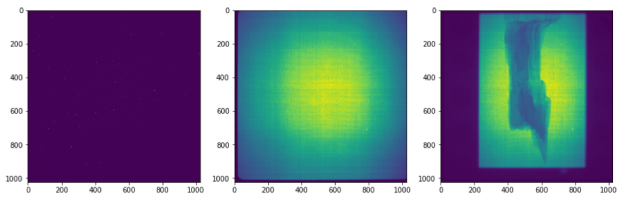
MuhRec meets python

The wood data

The wood data is the data I use a lot for demos, tests, and tutorials. It a piece of petrified wood DOI.

Load projection data

```
[39]: ob = io.readImages('/data/P08062_wood/raw_CCD/ob_{0:04d}.fits',1,5,averageStack=True).mean(axis=0)
      dc = io.readImages('/data/P08062_wood/raw_CCD/dark_{0:04d}.fits',1,5,averageStack=True).mean(axis=0)

[40]: proj = io.readImages('/data/P08062_wood/raw_CCD/wood_{0:04d}.fits',1,626) # This takes a while

[41]: fig,ax = plt.subplots(1,3,figsize=[15,7])
      ax[0].imshow(dc)
      ax[1].imshow(ob)
      ax[2].imshow(proj[0])

[41]: <matplotlib.image.AxesImage at 0x7fa218ee6e90>
```

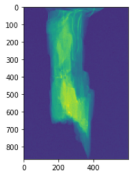


Preprocessing

Here the projections needs to be prepared for reconstruction. Typical operations are

- Cropping
- Normalization (possibly with scattering correction)
- Spot cleaning
- Ring cleaning

```
[42]: nproj = amg.normalizeImage(img=proj, ob=ob, dc=dc, doseROI=[100,250,200,300])

[49]: cproj = nproj[:,50:920,250:850]
      plt.imshow(cproj[0])
      del nproj
```



Algorithms included in pyMuhRec

Preprocessing:

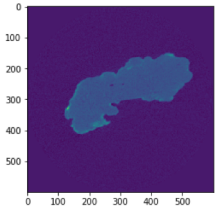
- *Normalization*
- *Spot cleanning*
- *Ring cleaning*
- *Basic beam hardening correction*

Reconstruction:

- *Center estimate*
- *Parallel CPU based FBP*
- *Cone beam FDK (not tested)*

Prepare and run the back-projection

```
[46]: Nproj = cproj.shape[0]
      # Information per projection
      args = {"angles"  : np.linspace(0,360,num=Nproj),
              "weights" : np.ones(Nproj)/Nproj}

      # Geometry information
      recon.configure({   "center" : 295,
                          "resolution" : 0.05
                      })

[47]: recon.process(cproj[:,500:532,:],args) # Reconstruct a part of the slices (32 slices here)

      vol = recon.volume() # Retrieve the reconstructed volume

[48]: plt.imshow(vol[0])

[48]: <matplotlib.image.AxesImage at 0x7fa2196c6910>
```



Ongoing work

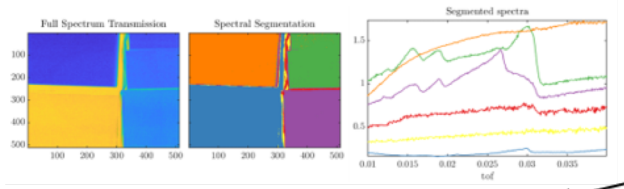
- Handling workflow differences between C++ and python
- Improving build and packaging to make module portable
- Scatter correction support

A. Kaestner

ToFlib

Purpose:

- Provide a list of tools to aid users in the data processing of ToF data

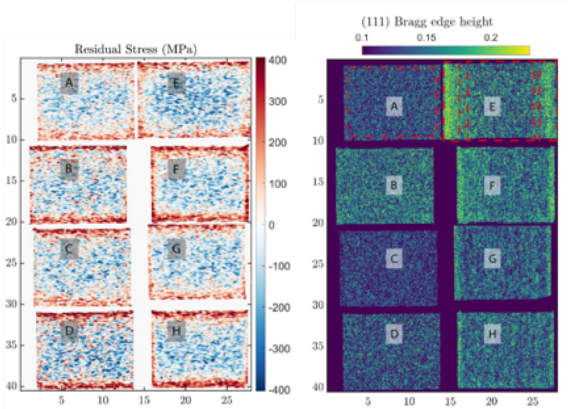


ToF library

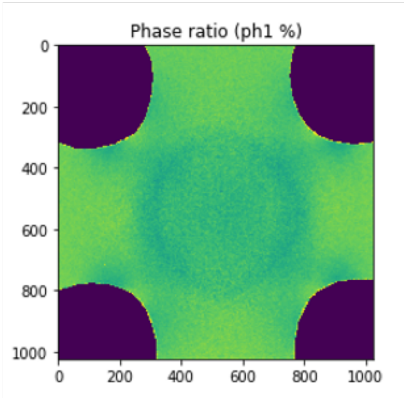
Reduction Tools

- Data processing and reduction
- Segmentation tool
- Image filtering

Single Edge Fitting



Phase Fraction Fitting



M. Busi

ToFlib status update: 1.0.0 release coming soon

PAUL SCHERRER INSTITUT

Features

Reduction tools:

i. Filtering tools to tackle low statistics regimes

ii. Segmentation tools

Edge fitting (for **residual stress mapping**):

i. *Gaussian method*: stable and tested on 3+ datasets

ii. *Advanced method*: the edge model fitting method has now a direct fit variant that works with a 5x gain in speed and nearly identical results (this is particularly useful for textured edge shapes for which the Gaussian method struggles)

Phase fraction fitting:

- *Linear combination method*: tested with nearly identical result to published data of TRIP steel.

Cross Section Library: the library has been furnished with some data library that can be expanded in the future

- Stainless steel 316L cross sections (FCC and BCC phases)

Development

Documentation:

- Online browser wiki https://neutronimaging.github.io/ToFImaging/
- Python source code is well documented
- Jupyter notebooks with detailed examples for strain mapping and phase fraction fitting
  - Requires data server to store example data (about 200Mb each)

Installation:

- Pip install now available
- Manually cloning source code

M. Busi

Finish presentation