

Nabu progress report as of 13/03/2020

Nabu is the tomography reconstruction software aiming at replacing and extending PyHST2. Primarily tailored for ESRF needs, it is also designed to be versatile to be used in other projects.

This document summarises the current state and lays out a tentative roadmap for future developments.

1. Nabu current state

1.1 - March Milestone and ID19 commissioning

It was agreed with DAU members to deliver a first working prototype of Nabu for early March. The aim was to use the ID19 commissioning for testing the software, notably the coupling between tomwer (high-level workflow) and Nabu (reconstruction).

The tests are still ongoing.

In its current state, Nabu provides a simple reconstruction pipeline.

A configuration file is used to describe this pipeline (Nabu provides a helper to generate it from scratch with pre-filled values and help).. The user provides the dataset to process, and possibly modifies some parameters in the configuration file to fits his needs.

From the configuration file, the following processing steps are performed:

- Flat-fielding
- Phase retrieval (Paganin)
- Absorption normalization
- Reconstruction (FBP)
- Saving in HDF5

The primary objective was to test that this processing pipeline works from data reading to final saving of the results. Therefore, it is relatively limited in terms of features. A notable amount of efforts is also being spent to ensure that Nabu and Tomwer work together.

1.2 - Short term works

The first tests suggest that the above pipeline works. However, as the priority was to "plug" everything together as soon as possible, many issues were left untreated. Therefore, **once the testing campaign is done, some work has to be carried out to "polish" the existing basic pipeline** to improve its robustness. **The current pipeline cannot be taken as a finished product.**

2. Future developments

In the frame of tomo-bridge project, the document "Work packages and milestones in detail" (October 2019) describes the features needed and their prioritisation. This document assumes that all the beamlines scientists agree with the features and their prioritisation.

2.1 - Features and schedule til September ("M 1.1")

Once these needs were defined, a refinement step was done to provide more precise dates. In the DAU tomo-bridge minutes of 30/01/2020 (available at <http://gitlab.esrf.fr/tomotools/minutes/-/blob/master/minutes-20200130.rst>), the following features were defined for **September** :

- Simple flat-fielding and absorption linearization (log) [F02]
- Projection-based ring correction (e.g. double flat-fielding) [F06]
- Paganin filtering with modifications (unsharp mask, deconvolution later) [F09]
- Simple rotation-axis finding algorithms (global, accurate, near, manual) [F12]
- Sample shifts for correction sample motion, reducing ring artifacts (aka. correct motion) [F13]
- Half-acquisition [F20]
- Data conversion (8/16 TIFF) [F18]

Warning: these features do not account for necessary internal works inside Nabu ; for example implement the computations distribution across the computing cluster (necessary for full-volume reconstruction in a reasonable time).

2.2 - Features after September ("M 1.2")

The features for "Milestone 1.2" are:

- Phase retrieval with Contrast Transfer Function [F10]
- Handling of XRD-CT, notably interface with pyFAI [F14]

3 - Time estimates

This section is an attempt at estimating how much time it will take to deliver the next features in the short term.

Before the time estimates, I kindly remind the following:

- Implementing an isolated feature is generally simple. In fact, there are chances that an implementation is already available in another software. What takes time is to (1) verify that it works *well* (unit tests), (2) provide a high performance implementation, (3) integrate this feature in a wider processing pipeline, and (4) document this feature for the final user. In particular, it takes time to ensure high performance when assembling components: "the whole takes more time than the sum of its parts".
- The features listed in the tomo-bridge document do not represent 100% of the time needed for the developments. Internal works (software architecture, computations distributions, tests) are needed.
- We can always choose only two out of these three : (1) software quality/robustness ; (2) number of features ; (3) fast delivery. In my opinion, it makes no sense to choose (2) and (3) at the expense of (1), since we then fall-back on the previous legacy software.

Simple Flat-fielding and absorption linearization [F02] Implemented, unit tested. Cuda and Numpy backends. Room for improvements:

- Use the "raw darks/refs" (for now done by tomwer)
- OpenCL backend
- Support more than one "processed dark" (?)

Double flat-fielding for rings correction [F06] Not done. Time estimate : 1 week.

Paganin phase retrieval and Unsharp mask [F09] Implemented and unit-tested against tomopy. NB: results are different from PyHST2. Cuda and Numpy backends. Room for improvement: OpenCL backend.

Simple rotation-axis finding algorithms [F12] Not done. Although probably important even for acquisition, **it is the feature bearing most incertitude**. It shares the principle of alignment routines ("correlate.m" in the legacy code). The principle is simple (correlation with FFT + fit a parabola), but "the devil is in the details", and existing code is thick and quite abstruse.

Time estimates:

- Sinogram-based CoR estimation (won't work in local tomography, except maybe in half acquisition): 1 week
- Projections-based CoR with "FFT + fit a parabola" : 1 week

More complicated algorithms will require more time.

Sample shifts [F13] Not done. Time estimate: 1-2 week.

Half-acquisition [F20] Not done Time estimate: 1 weeks

Data conversion [F18] Not done. Time estimate: 1 week (tiff, raw and jpeg2000 through Python)

Computations distributions on SLURM (internal work) Not done. Time estimate: 1-2 weeks

Test campaign (in September ?) Test the features for "Milestone 1.1". Detect and correct bugs, get users feed-back. Time estimate : at least 2 weeks

Phase retrieval with Contrast Transfer Function [F10] Not done. Time estimate: 1 week

Handling of XRD-CT, notably interface with pyFAI [F14] Not done. This feature will take more time, as it is about designing a complete pipeline for XRD-CT and not adding features to the "full-field" pipeline.

From a meeting we had with Marco and Gavin, the features needed are the following:

- Interface with pyFAI
- Interpolation / re-gridding methods (ex. non-equispaced points)
- Background removal (with darks/flats or with sinogram borders or multi-resolution scan)
- Reconstruction and first-order absorption correction

Time estimate: 1-2 months