Journal of Synchrotron Radiation

ISSN 0909-0495

TomoPy: A framework for the analysis of synchrotron tomographic data

Doğa Gürsoy,* Francesco De Carlo and ? ?

Advanced Photon Source, Argonne National Laboratory, 9700 S. Cass Ave., Argonne IL 60439-4837 USA.
Correspondence e-mail: dgursoy@aps.anl.gov

Abstract goes here.

1. Introduction

Analysis of large tomographic datasets at synchrotron light sources is becoming progressively more challenging due to the increasing data acquisition rates that new technologies in X-ray sources and detectors enable. The next generation of synchrotron facilities that are currently under design or construction throughout the world will provide diffraction limited X-ray sources and is expected to boost the current data rates by several orders of magnitude and stressing the need for the development and integration of efficient analysis tools more than ever. Several synchrotron users, to take advantage of specific instrument characteristics and specifications, already collect data at different facilities, but they are often left alone integrating this data with the available or home-grown software tools.

There are often strong similarities within the same class of instruments that are not systematically utilized to build a coordinated software development platform or framework where each facility and user group can take advantage from or contribute in, ultimately saving on-site resources by sharing the computing tasks with the user community. Here we describe in detail an attempt to provide such a collaborative framework for the analysis of synchrotron tomographic data that has the potential to unify the effort of different facilities and beamlines performing similar tasks. The basic principles of this Python/C++ based open-source framework, called *TomoPy*, are: OS and data format independent, parallelizable, modular and supporting a functional programming style that many researchers in Academia prefer.

2. Background

When it comes to the digital storage of tomographic experimental data and the development of analysis tools at synchrotron light sources around the world, the situation is very heterogeneous. As different research teams and instruments have grown at various facilities, they have often developed local data and analysis models based on instrument hardware specificity and often drawing upon the particular preferences of a scientist writing software.

Many tomographic analysis tools utilize licensed, and often expensive, software packages like Matlab (**?**) and IDL (**?**), others rely on specific, and often complex to maintain, computing infrastructure like MPI-based CPU or GPU clusters. The availability of inexpensive multi-core CPU workstations and the reduced cost of computer memory allowed to simplify the computing infrastructure required to process tomographic data (**?**), and started the development of single workstation tools able to perform most computing tasks.

This transition is affecting all major synchrotron facilities where new effort is now dedicated into developing new tools that can be deployed at the facility for real time processing as well as distributed to users for off site data processing.

3. TomoPy: A python based framework

There is no free lunch when it comes to the integration of analysis methods. For example, Matlab's or IDL's coding practice which is based on matrix and linear algebra operations that many researchers get accustomed to come with pricy licensing fees compared to open source alternatives. R, (**?**), as an open-source alternative, offers a wide variety of statistical data analysis tools but has a steep learning curve and is usually hard to maintain for large projects. Python with its standard library on the other hand provides a modular, readable and manageable framework that researchers can use freely or contribute easily without facing any hassles. Besides, the native control software running at several synchrotron facilities, EPICS, (**?**), is interfaced to Python (**?**), allowing data analysis and real-time feedback on the instrumentation status. These features make Python the method of choice for development framework.

Tomographic data processing at synchrotrons is essentially modular in nature such that the processes can be divided into a number of independent, and in many situations serialized, steps. This allows a degree of control for the entire process by checking the quality of the steps taken. A natural way to segment the processes is to group functions according to the similarities in transformations applied on data as depicted in figure 1. The first step is to convert data into a compatible HDF5 dataset. The data conversion and the assignment of metadata is carried out by the Data Exchange module which is currently supporting a wide variety of data formats. Once the HDF5 dataset is ready, a set of modules is available to preprocess, reconstruct and analyze data. The following section describe the analysis steps in the order they are performed.

4. Methods

In synchrotron tomography, the imaging sample is placed in the beam path and the x-ray intensity profiles on the detector (also called tomograms or projections) are acquired by rotating the sample during x-ray exposure. Almost all data acquisition

protocols require another set of data taken in the absence of the sample which is usually refereed to as the white-field (or flat-field) measurements. The order of the multidimensional arrays is, by default, assumed as the fastest changing dimension being the last dimension, and the slowest changing dimension being the first dimension. These two set of measurement together with the rotation angle information are the entry point of the analysis pipeline.

4.1. Pre-processing

Once the data and the corresponding projection angles are imported from the HDF5 dataset, a set of default methods is available under the preprocessing module simply to prepare data for reconstruction. The common denominator of these set of functions is that the range of the domain of the transformations lie in the same data space. Generally normalization is the initial step which is carried out by dividing the raw data to the averaged white field data not only to compensate different sensitivities and responses in each detector pixel but also to scale the images between 0 and 1 to obtain correct attenuation information about the sample.

For most datasets, a simple normalization can not satisfactorily correct for the detection artifacts, which are usually caused by the drifts in measurements or by the non-linearity in the detector response. These imperfections appear as stripes in sinogram and transform to image domain as ring shaped artifacts. This is particularly as issue for .... (Francesco: fly scan? slow data acquisition?) data acquisition where ..... . To correct for these ring artifacts, TomoPy employs the combined wavelet-Fourier filtering (**?**) which is superior to many other filtering approaches in terms of robustness and conservation of image features. The method is essentially based on a set of transformations to condense the artifacts into a tiny region so that removal of them would not cause any deformations to actual features of the sample.

consecutively applying the Wavelet transform to each sinogram image and normalizing each column of the vertical images with the average of the column. The normalization step is carried out by removing the off-set value of the 1-D Fourier transform of the columns. Sometimes the suppression of the nearby low-frequency components using a weighted band-pass filter may help to increase the accuracy of the filtering.

Normally it doesn't require a parameter tuning which makes it the method of choice.

Unless corrected, these artifacts transform into the image space as ring shaped artifacts. The easiest and computationally most effective way of removing these effect is to apply median filtering along the sinogram, preferably with a small window size not to cause excessive blurring, but large artifacts need a more careful attention.

TODO: wrap pre-processing methods more generally than explaining each in detail. Emphasize only the core points.

4.2. Reconstruction

The next module in the pipeline is the reconstruction module which contains functions that maps data from data space into image space. There are numerous methods suitable for this task each having both strengths and weaknesses depending on the applied data and timing constraints. As the default method, TomoPy provides Gridrec, which is a linear method that relies on discrete Fourier transforms of data similar to the filtered-backprojection method. The main difference is that it samples the Fourier domain on a regular grid before transforming back to the spatial domain. Utilizing fast Fourier transforms on regular grids outperforms other methods in terms of computational speed and is usually desired for quick reconstructions.

One of the key goals of the framework is to render the integration of nonlinear reconstruction methods possible in this framework. Nonlinear methods, also referred to as the model-based methods, try to minimize for data fidelity based on a system model, and in principle, they all require an accurate forward model which mainly relies on an efficient ray-tracing implementation. To serve on this purpose, TomoPy provide 2-dimensional and 3-dimensional ray-tracing algorithms that one can use to construct any nonlinear reconstruction method without much programming hurdle. We also provide models for various imaging components such as the stages, detectors and source characteristics, to be available in case an accurate forward model is required.

The success of the reconstructions usually require a good estimate of several geometrical parameters such as the rotation center. One common way to estimate it is to calculate the distance of sinogram's center of mass to the mid-point (**?**). Although, this method is computationally cheaper than the alternatives, its applicability is only limited to uniformly sampled 360 degree datasets which makes it impractical for many situations. Another approach is to exploit the systematic artifacts in reconstructed images due to shifts in the rotation center (**?**), see Figure 3, favorably within an optimization framework. TomoPy utilizes Shannon entropy as a measure to evaluate the image quality with the corresponding cost function:

$$\arg\min_{c} \left\{ \sum_{i} \sum_{j} H_{ij}\left[r(c)\right] \log_2 H_{ij}\left[r(c)\right] \right\} \qquad (1)$$

with c being the unknown rotation center, $r(c)$ is the reconstructed image which is a function of c, and H is the 2-D histogram of $r(c)$. The optimization problem is usually well-behaved and can be solved using any suitable optimization method in short times. For example, figure 3 plots the cost function for different rotation centers. A simplex search algorithm converges to the correct rotation center usually in less than twenty iterations for a wide range of initial points which makes it a method of choice if robustness is desired. The central point of the image as an initial guess is generally more than enough for many datasets to find an accurate center in an automated fashion.

4.3. Post-processing and data storage

For some applications further processing steps such as segmentation of regions or a quantification analysis may be

desired. Several commercial softwares like Amira or Avizo can provide comprehensive visualization and analysis tools, but for some simpler tasks they may be a little overkill. In TomoPy we intend to provide a number of post-processing functions, the most noteworthy of which is the segmentation of regions to acquire quantitative information about the sample based on image histograms. The success of this step is highly dependent on the previous transformations applied on the data, but modern methods like Wavelet-Fourier filtering to remove ring artifacts in reconstructed images together with the total variation type regularization methods renders an easy segmentation possible.

The output data can be exported to any format that the users desire. Although the majority of users prefer TIFF as a means of data format, TIFF files makes it hard to keep track of provenance information about metadata and attributes. In these cases where the metadata can't be written as a header or inside the file, all the provenance information including acquisition, analysis, storage and deployment processing steps are written on the original HDF5 data file. This allows reconstruction and analysis steps to be easily reproduced by reading information in the provenance group of the HDF5 file.

5. Discussions

In this paper, we tried to give a brief glimpse of the methods in the framework, but more importantly to emphasize the importance of unifying the efforts towards the development of data analysis and reconstruction tools targeting synchrotron tomography applications. The success of such initiative, of course, depends on many factors the most critical of which is to ... , but we believe that ...

Wrap the important points, vision and aims and conclude.

Figure 1
The scotch of the data analysis pipeline.

Figure 2
The reconstructed images obtained with different centers of rotations: (a) Correct center, (b) 16 pixels off-center horizontally, (c) 32 pixels off-center horizontally.

Figure 3
Plot of Shannon entropy as a function of different rotation centers.

Appendix A
Appendix title

Appendix text.

Acknowledgements

Use of the Advanced Photon Source, an Office of Science User Facility operated for the U.S. Department of Energy (DOE) Office of Science by Argonne National Laboratory, was supported by the U.S. DOE under Contract No. DE-AC02-06CH11357