Journal of Synchrotron Radiation

ISSN 0909-0495

# TomoPy: A framework for the analysis of synchrotron tomographic data

**Doğa Gürsoy,* Francesco De Carlo, Xianghui Xiao and Chris Jacobsen**

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

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. 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 proposed Python/C++ based framework is open-source, platform and data format independent, having multiprocessing capability and supports functional programming that many researchers prefer. This collaborative platform could affect 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.



## 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 are expected to boost the current data rates by several orders of magnitude, thereby stressing the need for the development and integration of automated and efficient analysis tools more than ever.

Several synchrotron users already collect data at different facilities to take advantage of specific instrument characteristics and specifications, but they are often left alone integrating this data with the available or home-grown software tools. However, 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 serve the needs of different facilities and beamlines performing similar tasks. The basic principles of this Python-based open-source framework, called *TomoPy* (Gürsoy, 2013), include ease of collaborative development of scripts, platform and data format independence, modularity, and support for a functional programming style that many researchers prefer. In the following sections, we will provide a brief background on the analysis of tomographic data at synchrotrons, and then shortly introduce the framework by describing the model structure together with the currently available methods.

## 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 (just like Babel—see Genesis 11:1–9 in the judeochristian bible). As different research teams and instruments have grown at various facilities, they have often developed local data and analysis models based on the 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 (MathWorks, 2014) and IDL (Exelis, 2014), while others rely on specific (and often complex to maintain) computing infrastructure like MPI-based CPU or GPU clusters. This diversity and specialization hampers widespread delivery of an integrated development environment and limits its long term use. On the other hand, modern chip design shows a clear trend toward integration of multiple processor cores, thus permitting concurrency without a need for a dedicated cluster. The availability of inexpensive multi-core CPU workstations and the reduced cost of computer memory allow to considerably simplify the computing infrastructure required to process tomographic data (Rivers, 2012), and make the development of single workstation tools able to perform most computing tasks at reasonable times. Besides, it allows the

analysis to be performed on an off-site location, freeing on-site computational resources for other tasks.

This transition is affecting all major synchrotron facilities where new effort is now dedicated into developing tools that can be deployed at the facility for real time processing as well as distributed to users for off-site data processing. We believe this will pave the way for an efficient use of the existing knowledge and methods base, thereby fostering collaborations and development of novel methods.

## 3. TomoPy: A python based framework

There is no free lunch when it comes to the integration of analysis methods. For example, Matlab and IDL use coding practices which are based on matrix and linear algebra operations that many researchers find useful, but different laboratories choose to purchase one package over another so that inevitably some researchers are left out when a choice to support one of these packages is made. The statistical programming language R (Foundation, 2014) is an open-source alternative, but it is generally considered as having a steep learning curve and is usually hard to maintain for large projects. Low-level programming languages like C or Fortran provide the desired computational speed but lack in readability and hinder a collaborative code development. Python plus standard packages like NumPy and SciPy offer a free, open-source, modular, readable and manageable framework that researchers can use and contribute to easily. Python also offers easy integration to C or Fortran codes through shared libraries. In addition, the native control software running at several synchrotron facilities (EPICS, 2014) is accessible via Python (Newville, 2014), allowing simultaneous data analysis and real-time feedback on the instrumentation status. These features make Python the tool of choice for development framework.

TomoPy relies on the Scientific Data Exchange, an HDF-5 based schema to store raw, analyzed and provenance data providing support for all synchrotron based tomography system raw data formats (Argonne, 2013). Intermediate, analyzed, or final reconstructed data can be stored used the same HDF-5 schema or distributed to the final user in any on the most common formats.

The data-driven pipeline of tomographic reconstruction allows for an easy kind of concurrency which is simply based on data parallelism. The tasks can be distributed through a queue into available processors and executed in parallel, and in many cases, there is either zero or minimal cross-talk between the distributed tasks which simplifies the implementation and exploitation of parallelism.

Digestion of synchrotron tomographic data is essentially modular in nature such that the processes can be divided into a number of independent, and in many situations serialized, steps. A natural way of segmenting these processes is to group tasks according to the similarities in transformations applied on data, that is distributing tasks into modules like pre-processing (data-to-data transformations), reconstruction (data-to-image transformations) and post-processing (image-to-image transformations). This modular pipeline design allows a degree of control for the entire process by checking the quality of individual steps taken and at the same time improves the readability of the code. The following section describes the journey of data, through multiple transformations, from acquisition to image display as presented in Figure 1.

## 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 sets of data taken in the absence of the sample which is usually refereed to as the white-field measurements ($I_{white}$) and in the absence of X-ray exposure which is called the dark-field measurements ($I_{dark}$). 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 three sets 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. Generally the pipeline of functions start with the normalization which include a dark-field (offset) and gain correction of raw intensity data ($I_{raw}$) of

$$I_{normalized} = \frac{I_{\text{raw}} - I_{\text{dark}}}{I_{\text{white}} - I_{\text{dark}}}. \quad (1)$$

This step is essential for almost all the time not only to compensate different sensitivities and responses in each detector pixel but also to scale the images between 0 and 1 to obtain a reliable attenuation information about the sample according to the Beer-Lambert's law.

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 when defects are present on the scintillator screen or on the detector (bad pixel or non–linearity of the pixel response). To correct for these ring artifacts, a combined wavelet-Fourier filtering (Münch et al., 2009) is adopted. The filtering 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 and is superior to many other filtering approaches in terms of robustness and conservation of image features (see Fig. 2).

For some samples that are either beam sensitive or have low absorption contrast like biological specimens, an in-line Phase-Contrast Imaging (PCI) mode, which exploits Fresnel diffraction, is applied. In PCI, to retrieve phase out of the projected phase-contrast data, single-step methods are of

primary importance because they allow a much faster scanning without a need to alter the detector position multiple times during data acquisition (Burvall et al., 2011). PCI also may be utilized to separate the phase and the attenuation contrast, and ultimately to reconstruct the distribution of the real and imaginary part of the refractive index separately. Figure 3 shows the phase-contrast data and the corresponding retrieved phase using the well-known Paganin filter (Paganin et al., 2002).

### 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, we provide Gridrec (Donath et al., 2006), which is a direct Fourier 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.

The success of the reconstructions usually require a good estimate of several geometrical parameters such as the rotation center. This problem is unique to synchrotron radiation where the detector is fixed and the sample is rotating. One common way to estimate it is to calculate the distance of sinogram's center of mass to the mid-point (Azevedo et al., 1990). 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 general use. Another approach is to exploit the systematic artifacts in reconstructed images due to shifts in the rotation center (Donath et al., 2006) (see Figure 4) as errors to be reduced within an optimization framework. To this end, we utilize Shannon entropy as a measure to evaluate the image quality with the corresponding cost function

$$\arg\min_{c} \left\{ \sum_{i} \sum_{j} H_{ij}\left[r(c)\right] \log_2 H_{ij}\left[r(c)\right] \right\} \qquad (2)$$

where $c$ is 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 rapidly solved using any suitable optimization method. As an example, in Fig. 5 we show 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. This reconstruction based approach to correct for unknown geometrical parameters can as well be utilized for example to determine the tilt angle of the rotation axis.

One of the key goals in reconstruction is to render the integration of nonlinear inversion 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, all require an accurate forward model which mainly relies on an efficient ray-tracing implementation (i. e., computation of the transmission matrix coefficients). To serve on this purpose, we build 2-dimensional and 3-dimensional ray-tracing algorithms that one can use to construct any nonlinear reconstruction method without much time and trouble. 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. More specific models or methods can be added on top of the existing methods as desired.

### 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 software packages like Amira or Avizo (FEI, 2014) can provide comprehensive visualization and analysis tools, but for several simpler tasks simpler, open-source tools might suffice. In this framework we intend to provide a number of post-processing methods like segmentation of regions and the quantitative analysis about the reconstructions and the sample. The success of an automated segmentation is highly sensitive to the preceding transformations applied on the data, but we believe that novel preprocessing methods like combined wavelet-Fourier filtering, model-based inverse models and improved regularization methods will render an easier segmentation possible.

The output data can be exported to any format that the users desire. Although the majority of users prefer TIFF as data format, TIFF files make it hard to keep track of provenance information about metadata and attributes. In these cases where the metadata can not 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 share data and more importantly software tools.

Acknowledgements

Work supported by the Argonne National Laboratory Laboratory Directed Research and Development (LDRD) "The Tao of Fusion: Pathways for Big-data Analysis of Energy Materials at Work" project 2013-168-N0. 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.

## References

Argonne, (2013). The data exchange file format.
**URL:** *http://www.aps.anl.gov/DataExchange/*

Azevedo, S. G., Schneberk, D., Fitch, J. & Martz, H. (1990). Nuclear Science, IEEE Transactions on, **37**(4), 1525–1540.

Burvall, A., Lundström, U., Takman, P. A. C., Larsson, D. H. & Hertz, H. M. (2011). Opt. Express, **19**(11), 10359–10376.
**URL:** *http://www.opticsexpress.org/abstract.cfm?URI=oe-19-11-10359*

Donath, T., Beckmann, F. & Schreyer, A. (2006). In Developments in X-Ray Tomography V, vol. 6318. SPIE.

EPICS, (2014). The experimental physics and industrial control system.
**URL:** *http://www.aps.anl.gov/epics/*

Exelis, (2014). Exelis visual information soultion.
**URL:** *http://www.exelisvis.com*

FEI, (2014). Visualization sciences group.
**URL:** *http://www.vsg3d.com/*

Foundation, R., (2014). The r project for statistical computing.
**URL:** *http://www.r-project.org/*

Gürsoy, D., (2013). The x-ray tomography toolbox.
**URL:** *http://www.aps.anl.gov/tomopy/*

MathWorks, (2014). Matlab the language of technical computing.
**URL:** *http://www.mathworks.com*

Münch, B., Trtik, P., Marone, F. & Stampanoni, M. (2009). Opt. Express, **17**(10), 8567–8591.
**URL:** *http://www.opticsexpress.org/abstract.cfm?URI=oe-17-10-8567*

Newville, M., (2014). Pyepics: Epics channel access for python.
**URL:** *http://cars.uchicago.edu/software/python/pyepics3/*

Paganin, D., Mayo, S. C., Gureyev, T. E., Miller, P. R. & Wilkins, S. W. (2002). Journal of Microscopy, **206**(1), 33–40.
**URL:** *http://dx.doi.org/10.1046/j.1365-2818.2002.01010.x*

Rivers, M. (2012). In Developments in X-Ray Tomography VIII, vol. 8506. SPIE.

**Figure 1**
TomoPy framework. The analysis chain is divided into pre-processing, reconstruction and post-processing modules. Any method or a collection of methods in any language can be hooked to one of these modules without compromising the modularity. A Python front-end is used to interface these modules and interact with the user. A customizable Python based multiprocessing interface is provided for time consuming computations.

**Figure 2**
Reconstructed images before (a) and after (b) the combined wavelet-Fourier method for stripe removal is applied on normalized data.

**Figure 3**
The phase-contrast projection data (a). Recovered phase image obtained with the single-step phase retrieval method (b). The corresponding reconstructed images with and without phase retrieval is applied (c), (d).

**Figure 4**
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 5**
Plot of Shannon entropy as a function of different rotation centers.