Tomography at the Advanced Photon Source

Mark L. Rivers,
Department of the Geophysical Sciences and
Center for Advanced Radiation Sources (CARS),
University of Chicago
Francesco De Carlo, Viktor Nikitin
Advanced Photon Source,
Argonne National Laboratory

February 2020

1 Abstract

We will present the tools that are currently used in production at the Advanced Photon Source to handle all operational aspects to perform a tomography experiment including data collection (tomoscan tomoDisplay), reconstruction (tomostream) and automated data and instrument management (dmagic, globus, adjust, metah5).

tomoscan, tomoDisplay: Python Tomography Data Acquisition using EPICS

Mark Rivers

tomoscan is a Python module for collecting computed tomography data at the APS using the EPICS control system. It implements a base class (TomoScan) with the code that should be beamline-independent. Beamline-dependent code is implemented in derived classes that inherit from tomoscan. It is currently in production use on the 2-BM, 7-BM, and 13-BM beamlines, and is planned for use on 32-ID in the near future.

Some of the features of tomoscan include:

- Compact code. The base class is fewer than 900 lines of Python, and each of the beamline specific derived classes is 250-500 lines.
- tomoscan does not contain any PV prefixes in the Python code. It reads the prefixes from a configuration file, which makes porting to a new beamline very easy. The configuration file itself does not even need to be created, it is simply the existing EPICS autosave request file for the tomography database.

- tomoscan acts as a "tomography scan server" which only implements the code to collect a single tomography dataset, including dark-fields, flat-fields, and projections. Writing to the EPICS PV "StartScan" tells tomoscan to collect a new dataset in a separate Python thread. "StartScan" is an EPICS busy record, so a client can do a ca_put_callback to wait for the scan to complete. There are other PVs that to configure all of the scan parameters, and to indicate the status and progress of the scan. Thus, any EPICS client can be used to create complex scans of things such as sample position, x-ray energy, sample temperature, etc., and this code does not need to be in the same Python process that is running tomoscan. Such clients include OPI displays such as medm and CSS, Python, IDL, the EPICS sscan record, SPEC, Bluesky, etc. tomoDiplay

tomoDisplay

tomoDisplay is a tomography data processing software written in IDL. It performs:

- Preprocessing (dark field correction, flat field corraction, zinger removal) using IDL itself. IDL inherently implements multi-threaded operations for most of these operations, so it is quite efficient.

- Reconstruction using the tomoRecon multi-threaded implementation of the Gridrec algorithm. Reconstruction occurs in threads that are separate from the IDL thread. This allows IDL to be performing file I/O at the same time that the reconstruction is being done. This results in the elapsed time being almost completely controlled by the I/O, rather than the reconstruction. The reconstruction allows very convenient optimization of the centering, with optional rotation of each projection in the case where the rotation axis is not perfectly misaligned with the detector columns.

- Visualization of slices in any of the 3 principle directions, and ability to make movies to the screen, to JPEG or TIFF files, or to MP4 files.

The plan is to move tomoDisplay from IDL to Python. However, currently tomoDisplay out-performs TomoPy by a factor of 3 on Linux and a factor of 7 on Windows. These peformance issues need to be understood and fixed.

Tomostream, orthorec: Python Tomography Streaming Reconstruction

Viktor Nikitin

Tomostream

Tomostream is python command-line-interface for supporting streaming analysis of tomographic data where all processing/reconstruction procedures are performed in real time while the stage is rotating. TomoStream relies on Tomoscan for tomography instrument control, and EPICS AreaDetector plugins for real-time array data preprocessing.

Tomostream provides 3 main functionalities:

- Dark-flat field broadcasting PV server (running on the computer controlling the detector).
  The PV server performs binning of dark/flat fields and broadcasting them in a PV variable that can be read by reconstruction engine clients. The server also dumps dark and flat fields into a temporarily hdf5 file and rewrite this file whenever new flat/dark fields are acquired. Acquisition of dark and flat fields is performed without stopping rotation of the stage.

- Streaming reconstruction of 3 X-Y-Z ortho-slices through the sample (running on a machine with GPU).
  The streaming reconstruction engine generates 3 selectable X-Y-Z orthogonal planes and makes them available as an EPICS PV viewable in ImageJ using the EPICS_NTNDA_Viewer plugin. Reconstruction of the ortho-slices is rapidly done by direct discretization of line integrals in computing the backprojection operator (opposed to gridrec where the Fourier-slice theorem is used for evaluating backprojection). Projections for reconstruction are taken in real time from a PV access variable (pvapy) and stored in a synchronized queue. On each reconstruction call new data are taken from the queue, copied to a circular GPU buffer containing projections for a 180 degrees interval, and then reconstructed.

- On-demand capturing projections to an hdf5 file.
  Users have the ability to save raw tomographic data at any time while streaming. Capturing to an hdf5 file is done immediately when the capture button is pressed. Whenever capturing is done, dark/flat fields from the temporarily hdf5 file generated by the PV server are copied to the file with data. In addition, we employ the circular buffer plugin (CB1) of AreaDetector to store a set of projections acquired before capturing is started. This allows to dump projections containing information about the sample right before the structural changes occur. Data from the circular buffer is also added to the hdf5 after capturing is done.

The resulting hdf5 file has the same format as in regular scanning by tomoscan.

We plan to improve tomostream by adding automatic detection of the sample changes, focusing to a region with the changes (switch X-Y-Z ortho-slices to that region), and auto-dumping data to an hdf5 file. Automatic detection of the region can be done with performing additional real-time 3D reconstruction of binned data, or with ML procedures analyzing projections.

Orthorec

Orthorec provides the same functionalities of tomostream using an already collected raw data file. Reconstruction speed is limited by the disk I/O and CPU-GPU data transfers. 3 ortho-slices generation takes approximately 20 seconds for data with sizes (1500x2048x2448) when using a modern GPU and an SSD disk. We expect to optimize the procedure to have it in real time. On the current stage, we use orthorec for the center search procedure with generating reconstructions for a range of centers. Manual center searching is more efficient with 3 X-Y-Z ortho-slices than with only one Z slice.

dmagic, globus, adjust, metah5, dxfile Python tools for data management and instrument automation

Francesco De Carlo

Several tools are in use to automate data management (data transfer, distribution etc.) and tomography instrument optimization (alignment, focus, etc.). We will describe these tools and how they integrate with tomoscan and tomostream. DMagic and Globus are designed to help managing the massive amount of data generated at the Advanced Photon Source by providing an automatic way to tag, share, notify and distribute data to users by using the EPICS PV for year-month, pi_last_name and pi_email served by tomoscan. These PVs are automatically updated for the current user using dmagic tag direct access to the APS scheduling system.

- dmagic is a python tool that provides easy command-line interface to the APS scheduling system. Basic functionalities include the ability, for the current experiment, to retrieve the users' email addresses from the APS scheduling as well as experiment information. This infomation can be printed ("show" option) or used to update EPICS PVs ("tag" option)
- globus is a python tool that interfaces the APS Data Management System. It reads beamline PVs to set up a Data Management experiment, create directories on the data acquisition and analysis machines, manage users for the experiment, send e-mails to users with information on how to get their

data from the APS globus servers (one click download link), and manage automated data transfer (termed DAQs) from the analysis machine to APS globus servers. Other tools include:

- adjust is commad-line-interface to automatically determine the detector pixel size, to adjust focus, to align rotation axis tilt/pitch and to center the rotation axis in the middle of the detector field of view.
- metah5 is commad-line-interface for extracting meta data from data exchange tomographic data used at the Advanced Photon Source beamlines and publish them on the beamline operation web log.

Acknowledgment

This research used resources of the Advanced Photon Source, a U.S. Department of Energy (DOE) Office of Science User Facility, operated for the DOE Office of Science by Argonne National Laboratory under Contract No. DE-AC02-06CH11357.