**Installing Software to Process Tomography Data from 13-BM-D**

**Mark Rivers**

**November 30, 2021**

This document describes how to install the required software to reconstruct and visualize tomography data from the 13-BM-D beamline. This includes data collected both on the upstream tomography table, and that collected in the 250-ton press. This software can be installed on Linux and Windows. I recommend a computer with at least 32 GB of RAM, and at least 8 cores. A fast solid-state hard disk for the data is also recommended.

The tomography data processing is done in 2 steps:

1. Preprocessing: Subtraction of the dark current, normalization to the flat fields, and removal of hot pixels (zingers) that typically result from an x-ray directly striking the camera sensor.
2. Reconstruction of the preprocessed X-Y-Theta data into X-Y-Z data.

For data collected prior to July 2020, the preprocessing and reconstruction are both done in an IDL GUI program, or using the IDL command line.

For data collected in July 2020 and later, the preprocessing is done using the Python command line. The reconstruction is done using the IDL GUI program, or using the IDL command line.

The ultimate goal is to replace the IDL software with the Python software, which is based on the TomoPy package. We plan to add a GUI to TomoPy that is similar to the current IDL GUI.

However, we are currently using Python for preprocessing and IDL for reconstruction as an interim solution for two reasons:

* IDL 8.8 and earlier have terrible performance reading HDF5 files. This means that using IDL for the preprocessing step would be prohibitively slow. This read performance was very recently fixed in IDL 8.8.1, but the IDL code has not yet been modified to handle the HDF5 files.
* TomoPy reconstructions are significantly slower than IDL, and are poorer quality. This needs study to be understood and fixed, since both IDL and TomoPy are using the same underlying C++ Gridrec code to do the reconstructions, so they should nominally produce the same result.

**File Formats**

Prior to July 2020 the raw data is stored in netCDF files with the extension “.nc”. There are typically three netCDF files per dataset. Two are the flat field images collected at the beginning and end of the scan, the third contains all of the projections. An ASCII text file with the extension “.setup” contains the metadata, including the sample information, x-ray energy, and pixel size.

Beginning in July 2020 the raw data is stored in HDF5 files with the extension “.h5”. There is one HDF5 file per dataset, and it contains the flat field images and the projections. It also contains metadata including the positions of many motors on the beamline, the ring current, the tomography data collection parameters, the sample and optics information, and more. An ASCII text file with the extension “.config” contains the tomography data collection parameters, the sample and optics information, and more.

The preprocessed files are stored in netCDF format. They have the extension “.volume”. Ultimately, we plan to eliminate the use of the intermediate preprocessed files completely, and store the preprocessed results in memory for immediate reconstruction.

The reconstructed files are also stored in netCDF format. These file names end in “recon.volume”. Ultimately, we plan to store the reconstructed files in HDF5 format, because they can be read significantly faster than netCDF files into IDL, Python, and ImageJ. They can also be written significantly faster than netCDF files in Python. However, IDL 8.8.1 is 2-7 slower to write HDF5 files than netCDF files, and is 10 times slower than Python writing HDF5 files.

**Python Preprocessing Software**

The following software must be installed to preprocess HDF5 raw data files.

* Install the Anaconda Python distribution.

This is the page for Windows:

<https://docs.anaconda.com/anaconda/install/windows/>

This is the page for Linux:

<https://docs.anaconda.com/anaconda/install/linux/>

* Once Anaconda is installed, open the Anaconda command prompt on Windows or a new bash shell on Linux.
* Install tomopy, creating a new Conda environment. I used the name tomo\_test for it in the example below, but you use any name, e.g. ‘tomopy’, ‘tomo’, etc.

>conda create --name tomopy --channel conda-forge tomopy

* Activate this new Conda environment.

>conda activate tomo\_test

* Install dxchange, which provides utilities to read and write HDF5, netCDF, and other files for tomopy

>conda install -c conda-forge dxchange

* Download the most recent version Python code for preprocessing 13-BM-D data from <https://github.com/CARS-UChicago/Python_tomography/tags>. The zip file is more convenient to use for Windows, and either the zip file or the tar.gz file can be used on Linux. Unpack the file into a directory on your Windows or Linux machine. For example on Windows you might put it in C:\tomography\Python, and on Linux you might put it in /usr/local/tomography/Python.
* Create the environment variable PYTHONPATH to point to the directory where you put the files in the previous step. On Windows you do this with Control Panel/System/Advanced System Settings/Environment Variables. On Linux you do this by editing your .bashrc or other shell startup script.
* To do the preprocessing type the following commands, using the actual absolute or relative path to your data files. base-file-name is the file name before the “.” of the .config and .h5 files.

cd path-to-your-data-files

python –m preprocess\_13bm base-file-name

**IDL Reconstruction Software**

* To use the reconstruction software you must install IDL, at least version 8.5.1. If you do not have an IDL license you can run the reconstruct software for free using the IDL Virtual Machine, which is part of the IDL installation package.
* You can register to download IDL on the Harris Geospatial Website here: <https://www.l3harrisgeospatial.com/Company/Create-Account?returnurl=https%3a%2f%2fwww.l3harrisgeospatial.com%2fSoftware-Technology%2fIDL>
* Alternatively I have put the installation files for IDL 8.5.1 and 8.8.1 here: <https://cars.uchicago.edu/data/idl/>
* On Windows you must add the directory <PATH\_TO\_TOMO\_DISPLAY>\win32dll or <PATH\_TO\_TOMO\_DISPLAY>\win64dll to your Windows environment variable "Path". This will allow Windows to find the correct version of libfftw3f-3.dll. This is done using Control Panel/System/Advanced System Settings/Environment Variables.
* Download the tomo\_display reconstruction software written at GSECARS. Zip and tar files can be downloaded from here. These could installed in C:\tomography\idl on Windows or /usr/local/tomography/idl on Linux.

<https://cars.uchicago.edu/software/pub/idl_tomography.zip>

<https://cars.uchicago.edu/software/pub/idl_tomography.tar>

* If you have an IDL license
  + To use the IDL tomo\_display software with an IDL license simply add the directory <PATH\_TO\_TOMO\_DISPLAY> to your IDL\_PATH, using the IDL Window/Preferences/IDL/Paths menu. <PATH\_TO\_TOMO\_DISPLAY> is where you downloaded the software in the previous step.
  + You do *not* need to define the environment variable TOMO\_RECON\_SHARE discussed below.
  + Start tomo\_display by typing “tomo\_display” at the IDL prompt.
* If you do not have an IDL license
  + The reconstruction uses the tomoRecon shareable C/C++ library. To use this library to reconstruct in tomo\_display do the following:
    - * On Linux:
      * Define the environment variable TOMO\_RECON\_SHARE to point to the complete path to tomoRecon\_linux\_x86\_64.so, which is contained in the idl\_tomography software downloaded above. Example:

export TOMO\_RECON\_SHARE=/usr/local/tomography/idl/tomoRecon\_linux\_x86\_64.so

* + - On Windows:
    - Define the environment variable TOMO\_RECON\_SHARE to point to the complete path to tomoRecon\_Win32\_x86.dll. Example

set TOMO\_RECON\_SHARE=C:\tomography\idl\tomoRecon\_Win32\_x86.dll

* + - Add the win64dll directory (e.g. C:\tomography\idl\win64dll) to your PATH environment variable. This is needed because that directory contains the fftw library needed by tomoRecon.
  + The idl\_tomography software downloaded above contains a file called tomo\_display.sav, which is a package that runs tomo\_display in the IDL Virtual Machine.
  + On Windows you can run tomo\_display.sav in the Virtual Machine by double-clicking on its icon or by opening the IDL Virtual Machine from the Windows start menu.
  + On Linux type the command:

idl –vm=<PATH\_TO\_TOMO\_DISPLAY>/tomo\_display.sav