Title: LANL12-RS-107J PYTHON Radiography Analysis Tool (PyRAT) Mid-Year Deliverable Report for FY15

Author(s): Temple, Brian Allen
Armstrong, Jerawan Chudoung

Intended for: Report

Issued: 2015-04-14

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer,is operated by the Los Alamos National Security, LLC for the National NuclearSecurity Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Departmentof Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

LA-UR-15- March 31th, 2015

LANL12-RS-107J
PYTHON Radiography Analysis Tool (PyRAT)
Mid-Year Deliverable Report for FY15

Brian Temple and Jerawan Armstrong

This document is a mid-year report on a deliverable for the PYTHON Radiography Analysis Tool (PyRAT) for project LANL12-RS-107J in FY15. The deliverable is deliverable number 2 in the work package and is titled "Add the ability to read in more types of image file formats in PyRAT". The details behind the deliverable is as follows: Currently PyRAT can only read in uncompressed TIF (or tiff) files. We would like to expand the image file formats that can be read by PyRAT so that we have the ability to read and process more types of image files. Many different formats of image files exist (jpg, png, and tiff for example). Adding more readable formats to PyRAT will make it easier to use in more situations. A summary of file formats added will be uploaded into webPMIS.

The deliverable has been completed in March 2015. Subroutine(s) have been developed for PyRAT to read in several different image file formats. Previously PyRAT could only read in uncompressed *tiff files. PyRAT can now read in *jpeg, *jpg, *png files, and formatted ASCII files in addition to *tiff files. The *png files and the uncompressed *tiff files are 16-bit files that have 65536 levels of data. The *jpeg/jpg and compressed *tiff files are 8-bit files that have 256 levels of data. The ASCII files are just matrices of the image data expressed as numbers in the rows and columns. Since XTK is used extensively by the program, we have tested the ability to read in files from XTK. Currently XTK 2.3 has the jpeg, *jpg, and *tif file formats available from the list of formats we have investigated. XTK has other formats available, but these formats are either more WINDOWS-based formats that we believe would have issues on the LINUX platforms used for PyRAT or are non-standard experimental formats. We felt it was not worthwhile to further investigate the use of these other formats because the existing formats in XTK are sufficient options for saving the image data. There are also multiple ways to save the *tiff and jpeg/jpg in XTK that were investigated in addition.

All of these formats can be read into and viewed in PyRAT. We have made additional changes to the code to allow the user to do lineout and take Abel inversions of the various formats. These capabilities are useful to perform qualitative analyses of the image which are very useful. These capabilities are sufficient to meet the deliverable requirements.

We still believe it is necessary to determine which formats can be used to perform material identification in PyRAT. We know that we can use the uncompressed *tiff files (16-bit) from XTK for material identification in PyRAT. We have determined that the use of the uncompressed 16-bit *png files could also be performed in the same manner as the uncompressed *tiff files. It is important to note that these *png files are about the same size as the uncompressed *tiff files, so no real benefit is obtained by using this format over the uncompressed *tiff files. In addition, XTK 2.3 does not have this file format as an option. Therefore we will not use uncompressed 16-bit *png files for analyses in

PyRAT, but will allow the user to load in the images in order to perform lineouts and Abel inversions.

We have determined that with the current form of our forward model, any data compression of the image data (going from 16-bit data to 8-bit data) is counter-productive to the material identification analyses in PyRAT. The compressed image files make the data range between similar areal density layers too close in magnitude to allow material identification of the layers. We are exploring whether we can fold in the data compression used for storing 8-bit image data into our forward model to see if it is possible to optimize compressed files. We are testing this theory on jpeg files stored from XTK. If this is successful on this format, we shall further explore other formats in XTK. If this is not successful, we shall consider it impossible or impractical to use 8-bit image data for quantitative analyses. Any further work on using other formats will be performed in future development efforts beyond FY15.

If we are limited to use only uncompressed 16-bit image files, we have developed a subroutine that can divide the uncompressed *tif files into multiple smaller ASCII files that can be more easily transferred over ER communications. Currently the capability to subdivide and read the *tiff files is in a standalone Python script. It is in this form because we are not sure how NA-42 wants to implement this capability. It should be possible to develop the script to be incorporated into XTK, maintained as a standalone code, implemented in PyRAT, or be implemented in any combination of choices mentioned. We will have the ability to subdivide the *tif files and read the ASCII files implemented in PyRAT by default, but await NA-42's input for implementing this capability outside of PyRAT.

All the code modifications we have made and will be making will be implemented in a beta version of the code. The production version of PyRAT containing all the improvements will not be released until all of the code changes for FY15 have been implemented and further tested. We expect this to be done the last quarter of FY15.