Test Report: Project Title

Isobel Bicket

December 17, 2017

1 Revision History

Date	Version	Notes
December 18, 2017	1.0	Initial draft

2 Symbols, Abbreviations and Acronyms

See SRS document for further definitions.

symbol	**description**
RMS	Root Mean Square
SNR	Signal-to-Noise Ratio
T	Test

Contents

List of Tables

List of Figures

This document details the results of the testing detailed in the Test Plan, for the code SpectrumImageAnalysisPy, which can be found, along with the other documentation, on `https://github.com/icbicket/SpectrumImageAnalysisPy/tree/SpectrumImageAnalysisPy_dev`. Data and results from testing can be found under `/home/isobel/Documents/McMaster/PythonCodes/DataAnalysis/src/TestCase`.

3 Functional Requirements Evaluation

3.1 R1: SI Inputs

3.2 R2: Spectrum Inputs

3.3 R3: Input Verification

3.4 R4: Slice (1D) SI and extract Image

3.5 R5: Mask (2D) SI and extract Spectrum

3.6 R6: Richardson-Lucy Deconvolution

The test data for testing the RMS error and SNR of the deconvolution algorithm were obtained from Dr. E.P.Bellido and are the same datasets used in [1].

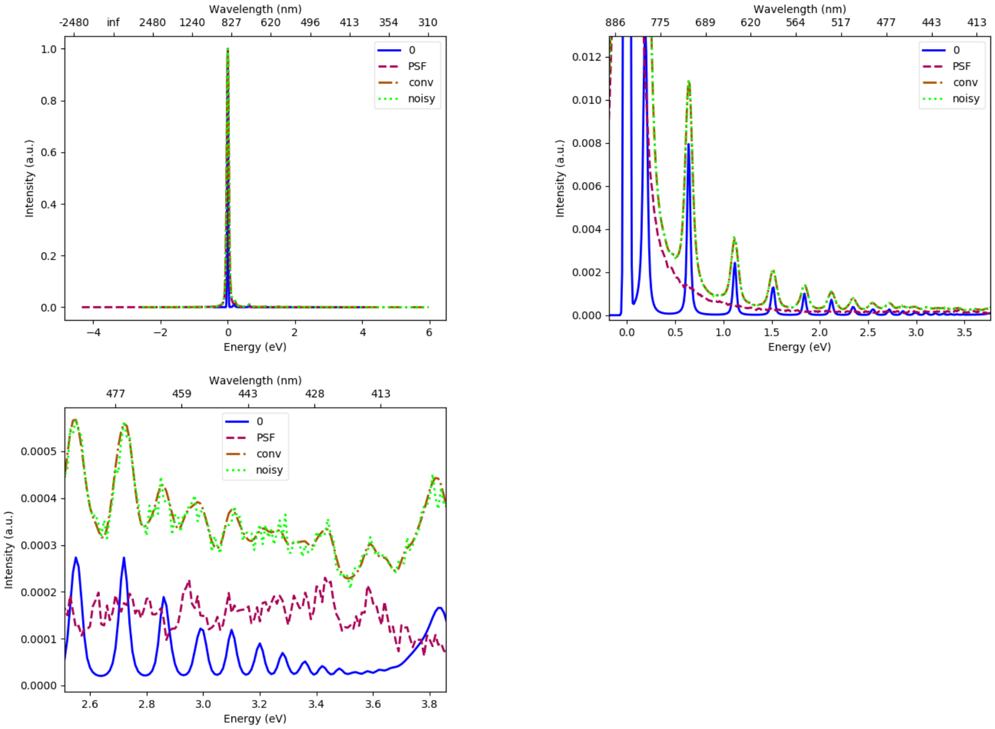

Figure 1: Original spectrum (0), Point spread function (PSF), the original spectrum convolved with the PSF (conv), and the original spectrum convolved with the PSF with Poisson noise added (noisy). Three images show three different regions of the spectrum.

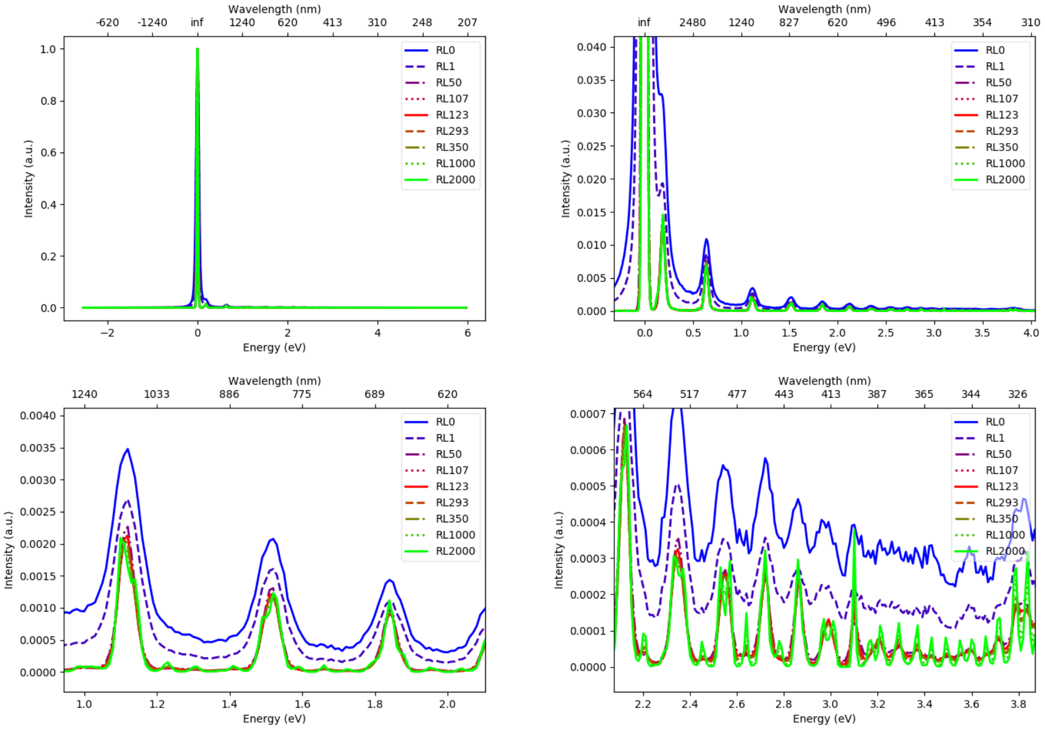

Figure 2: Spectra deconvolved using the Richardson-Lucy algorithm. Different numbers of iterations were used, as shown in the legend (RL2000 = 2000 iterations). Four images show different regions of the same plot.

3.7 R7: Normalization

3.8 R8: Background subtraction

3.9 R9: Gain Correction

4 Nonfunctional Requirements Evaluation

4.1 Usability

4.2 Performance

4.3 etc.

5 Comparison to Existing Implementation

This section will not be appropriate for every project.

6 Unit Testing

To date, partial unit testing has been performed on the following files:

- Filenamer.py, used to name the files to be exported and avoid conflicts with already existing files).
- Spectrum.py, used to implement the Data 1D Spectrum module.
- SpectrumImage.py, used to implement the Data 3D Spectrum module
- ImagePlotter.py, implementing the Display 2D Image module.

Coverage statistics for these files can be found in Statement Coverage and Branch Coverage.

7 Changes Due to Testing

- A zero in the denominator during RL deconvolution now raises an exception

8 Automated Testing

9 Trace to Requirements

10 Trace to Modules

11 Code Coverage Metrics

Code coverage metrics were obtained using the Python coverage library (coverage.py-4.4.2) and represent the statement and branch coverage of the unit tests, to date.

11.1 Statement Coverage

Statement coverage provides a value for the percentage of lines which were run during the unit tests.

Name	Statements	Missed	Coverage %
FileNamer.py	40	3	92%
ImagePlotter.py	136	97	29%
SpectrumImage.py	247	155	37%
Spectrum.py	136	70	49%

Table 1: Statement coverage

11.2 Branch Coverage

Branch coverage combines the number of statements run during the test with the number of branches covered during the test.

Branch Name	Statements	Missed	Branches	Partial Branches	Coverage %
FileNamer.py	40	3	14	1	93%
ImagePlotter.py	136	97	56	0	23%
SpectrumImage.py	247	155	60	1	37%
Spectrum.py	136	70	32	1	48%

Table 2: Branch coverage

References

[1] E. P. Bellido, D. Rossouw, and G. A. Botton, "Toward 10 meV Electron Energy-Loss Spectroscopy Resolution for Plasmonics," *Microscopy and Microanalysis*, vol. 20, pp. 767–778, June 2014.