SpectrumImageAnalysisPy

Isobel Bicket

December 7, 2017

1 Revision History

Date	Version	Notes
October 4, 2017	1.0	First draft
December 7, 2017	1.1	Edits to first draft

Contents

2 Reference Material

This section contains reference information to enable the reader to better understand the document.

2.1 Table of Units

Throughout this document the system of units standard to the field of electron microscopy is used as the unit system, for consistency with relevant literature and practices. For each unit, the symbol is given followed by a description of the quantity described by the symbol and the unit name.

symbol	quantity	unit name
counts	intensity	number of particles
nm	length	nanometre
eV	energy	electron volt

[Is "number of particles" a standard measure of intensity? I did a quick google search and I did not see any examples of intensity measured by the number of particles. —SS][If you think about the physical process of detecting a signal in the ideal case, you get one incoming photon and you get one count on a spectrum (or image). So in this sense, you are counting the number of photons (particles) which hit the detector at each pixel, and this builds up the intensity of that pixel. More particles hitting one pixel than another means that the first pixel has more intensity, so will be brighter/have higher counts. It's usually not explicitly stated that the counts refers to the number of particles, but I wanted to make this connection to the physical detector. In reality, you have additional problems like dark current in the detector electronics which add intensity to the spectrum/image which don't correspond to actual particles, but in the ideal case, this would not be a problem. (I think of it like 'counts'...what are you counting? ...things that hit the detector at that pixel) —Author]

2.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The symbols are listed in alphabetical order.

symbol	unit	description
A	nm^2	Area of interest on the sample
$b(k)$	counts	Background signal on a detector
c	–	Index of the current iteration in deconvolution

E	nm or eV	Energy of the collected particle, represented by wavelength (CL) or energy loss (EELS)
$f(\rho')$	–	Probability of an electron landing on the detector at position ρ'
$f(k')$	–	Probability of an electron landing on the detector at pixel k'
$g(k)$	–	Gain value at each pixel
$h(\rho, \rho')$	–	Probability of detecting an electron at position ρ if it hits position ρ'
$h(k, k')$	–	Probability of detecting an electron at pixel k if it hits pixel k'
i	–	Index for pixel number in a spectrum
$I(E)$	counts	Intensity of a spectrum as a function of energy
$I_{background}(E)$	counts	Intensity of a background contribution to the spectrum (from the spectrometer), as a function of energy
$I_{collected}(E)$	counts	Intensity of a spectrum as received from the spectrometer, as a function of energy
$I_{corrected}(E)$	counts	Intensity of a spectrum from the sample with gain artifacts removed as a function of energy
$I_{deconvolved}(E)$	counts	Intensity of a deconvolved spectrum as a function of energy
$I_{measured}(E)$	counts	Intensity of a measured spectrum as a function of energy
$I_{norm}(E)$	counts	Intensity of a normalized spectrum as a function of energy
$I_{PSF}(E)$	counts	Intensity of a reference point spread function (spectrum) as a function of energy
$I^0_{real}(E)$	counts	Initial guess for a "real" spectrum for deconvolution, as a function of energy
$I^R_{real(j)}(E)$	counts	Intensity for a deconvolved spectrum after R deconvolution iterations
$I_{sample}(E)$	counts	Intensity of a spectrum from the sample with background artifacts removed as a function of energy
j	–	Index for pixel number in a spectrum
k	–	Pixel (index) on a spectrometer
k_1	–	First pixel (index) on a spectrometer (user selected)
k_2	–	Second pixel (index) on a spectrometer (user selected)
K	–	Maximum pixel (index) on a spectrometer
l	–	Index for pixel number in a spectrum
$N(k)$	counts	Noise signal from the detector
$P(x, y)$	nm	Position coordinates describing possible locations of the electron beam on the sample

R	–	Integer number of deconvolution iterations the user desires the software to perform
$S(k)$	counts	Signal entering the detector
$S'(k)$	counts	Signal entering the detector, resulting from a beam which has not interacted with the sample
x	nm	Displacement of the beam in one spatial dimension
X	nm	Maximum displacement of the beam in one spatial dimension
y	nm	Displacement of the beam in one spatial dimension, perpendicular to x
Y	nm	Maximum displacement of the beam in one spatial dimension, perpendicular to x
ρ	–	Location of detection of an electron hitting a detector
ρ'	–	Location of an electron hitting a detector
$\zeta(k)$	counts	Signal at each pixel as read out by a detector

2.3 Abbreviations and Acronyms

symbol	description
3D	Three-dimensions
A	Assumption
CCD	Charge-Coupled Device
CL	Cathodoluminescence Spectroscopy
DD	Data Definition
EELS	Electron Energy Loss Spectroscopy
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
PSF	Point Spread Function
R	Requirement
SI	Spectrum Image
SRS	Software Requirements Specification
STEM	Scanning Transmission Electron Microscope/Microscopy
SEM	Scanning Electron Microscope/Microscopy
TEM	Transmission Electron Microscope/Microscopy
T	Theoretical Model

3 Introduction

Three-dimensional datasets are produced by many spectro-microscopy techniques: techniques in which the response of a sample is mapped in two spatial domains and one spectral dimension. The visualization of 3D spectro-microscopy datasets is a challenging and often non-intuitive task. The motivation behind this software is to enable easy visualization and navigation through a 3D dataset to discover the features of the sample in both spatial and spectral domains. The data processing methods of 3D datasets are not trivial to perform and require a dedicated software designed to handle 3D data. This document provides the Software Requirements Specification (SRS) for a software designed to visualize and process 3D spectro-microscopy datasets derived from electron energy loss spectroscopy or cathodoluminescence.

3.1 Purpose of Document

This document details the requirements of the software SpectrumImageAnalysisPy. The responsibilities of the user and software are laid out and the requirements that the software must satisfy are explicitly detailed.

3.2 Scope of Requirements

The scope of the requirements for the software SpectrumImageAnalysisPy is limited to the import of; removal of instrumentation artifacts from; visualization and navigation through [Do you want the word through here? —SS][yes, though I have changed the rest of the clauses to match the same structure —Author]; and export of EELS and CL spectrum imaging data.

3.3 Characteristics of Intended Reader

The reader of this document should have an understanding of spectrum imaging techniques, particularly EELS and CL, and the data processing methods used to remove effects of the acquisition system from the data acquired. A basic knowledge of convolution theory, as might be obtained from an introductory post-secondary calculus or image processing course, will be helpful to the reader, as will an understanding of the characteristics of 3D datasets. The reader should understand EELS and CL, from either field experience or from an upper year post-secondary course on electron microscopy, as relevant to the sections of the document. [Can you also relate these characteristics to a standard degree and level? For instance, are you assuming an undergraduate degree in physics? —SS][To be honest, I don't think there is a standard education for understanding microscopy data or techniques. I suppose the lower limit would be secondary school physics, but beyond that I can picture university students/graduates, college students/graduates, employees in the field/lab...anyone with formal or informal education in the field should be able to read this. I have tried to clarify this more in the text. —Author]

3.4 Organization of Document

The document follows the organizational scheme laid out by Smith *et al.* [1, 2], starting with general reference material for the reader. The introduction of the document details the purpose of the document and intended reader, and is followed by the General System Description. The Specific System Description reduces the problem under consideration to a set of clearly defined definitions and models. The Requirements section of the document clearly outlines both the Functional and Non-functional Requirements that the software must comply with. Finally, the document contains a list of Likely Changes to the software and Traceability matrices and graphs.

4 General System Description

This section identifies the interfaces between the system and its environment and describes the user characteristics and lists the system constraints.

4.1 System Context

Use of the software SpectrumImageAnalysisPy confers the following responsibilities to the part of the user and to the program.

4.1.1 User Responsibilities

The user must

- provide the correct input data to the program;
- be capable of interacting with the software via a mouse, keyboard, and screen;
- have the necessary dependencies for the program installed; and
- possess sufficient knowledge on what processing steps need to be performed and be able to judge that the quality of the output data is sufficient for the application, or change the processing steps performed.

4.1.2 SpectrumImageAnalysisPy Responsibilities

The program must

- read inputs (either files or data arrays) and check the inputs for the correct data type and size;
- display the data and graphical user interface for the user to interact with;
- respond to user commands and accept user input as appropriate, including visualization and data processing commands; and

- export data in the correct file format(s).

4.2 User Characteristics

The end user of SpectrumImageAnalysisPy should be familiar with the concept of a spectrum image and an understanding of what the data represents and the appropriate actions needed to process spectrum images and extract useful information. A basic familiarity with programming is expected. An understanding of the spectro-microscopy technique (CL or EELS) used to generate the data will be beneficial, but is not strictly required.

4.3 System Constraints

The software must be able to read .dm3 files for EELS data import, and .h5 files for CL data import to be able to interface with output from the data acquisition software. [Great example of a constraint. —SS]

5 Specific System Description

This section first presents the problem description, which gives a high-level view of what the program is expected to do. The solution characteristics specification follows, detailing

- the assumptions that the software will make;
- the theoretical models the data analysis is based on;
- definitions for the types of data that the software must be able to process; and
- the instance models, which detail the data processing algorithms that the software will implement.

5.1 Problem Description

SpectrumImageAnalysisPy is a software to allow users to import, process, navigate, and export 3D spectrum image data.

5.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

Terminology	Definitions
Beam	The probe used by an electron microscope, composed of electrons accelerated to high speeds
Beam current	the number of electrons travelling in the beam per unit time
Cathodoluminescence	A technique in a STEM or SEM in which light emitted from the sample is collected by a mirror and directed to a spectrometer or other analysis system
Charge-Coupled Device	The camera used in the spectrometer to collect signal and output it to the microscope acquisition software
Electron energy loss spectroscopy	a technique performed in a TEM or STEM which detects the energy lost by an electron beam after passing through a sample
Point Spread Function (PSF)	A function representing the spreading in the spectrometer, as defined in DD3
Sample	A physical sample of scientific interest that has been probed by the beam in the EELS or CL instrument
Scanning transmission electron microscope	A TEM with the capability of scanning the electron beam in a raster scan over the sample
Slice	A subset of the dataset
Spectrometer	An instrument attached to a microscope and equipped with a CCD or equivalent camera, used for collecting signal as a function of energy
Scanning electron microscope	A microscope which uses an electron beam to probe the sample and acquires data based on scattering off the surface or near the surface of the sample, operating using a raster scan of the electron beam
Spectrum Image	A 3D dataset with two spatial dimensions and one spectral dimension, obtained using spectro-microscopy techniques (*i.e.*, EELS or CL) (DD2)
Transmission electron microscope	A microscope which uses an electron beam to probe a sample and collects the signal of the transmitted beam.

5.1.2 Physical System Description

The physical system that the program is concerned with the analysis of includes the following elements:

PS1: Sample under study.

PS2: STEM equipped with EEL spectrometer.

PS3: SEM or STEM equipped with CL collection system.

5.1.3 Goal Statements

Given a spectrum image dataset collected by either EELS or CL and a desire to visualize and process this dataset, the goal statements for SpectrumImageAnalysisPy are

GS1: Import a 3D dataset and display it such that the user can interact with it and navigate all three dimensions.

GS2: Provide processing options, including normalization and deconvolution for EELS SI; background subtraction and gain correction for a CL SI.

GS3: Extract slices and areas from the dataset as desired by the user and communicated through the user interface.

GS4: Export spectrum, image, or spectrum image data as desired by the user.

[Sounds good! —SS]

5.2 Solution Characteristics Specification

The assumptions used in consideration of spectrum imaging datasets are given in this section, followed by the models and definitions used to clearly define the data processing steps.

5.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: The EELS data can be described as the convolution of the "real" spectrum of the sample with the response of the microscope and spectrometer system function. The microscope and spectrometer system typically causes broadening in the peaks of the "real" spectrum.

A2: The same PSF is valid for all pixels in the EELS spectrum image.

A3: Fluctuations in the intensity of the EELS signal are due to changes in the beam current as collected by the spectrometer.

A4: A subtraction is sufficient to [to? —SS][yes! —Author] remove the contributions of the dark signal and the substrate signal from the CL sample signal.

A5: The CL spectrometer wavelength sensitivity can be accurately modelled by experimental reference data.

5.2.2 Theoretical Models

This section focuses on the general equations and laws underlying the data acquisition process on a camera, providing the motivation for the data processing models that SpectrumImage-AnalysisPy will implement.

Number	T1
Label	**Intensity**
Equation	$\zeta(k) = g(k)S(k) + b(k) + N(k)$
Description	The signal, $\zeta(k)$ (T2), obtained from each pixel, k, on the CCD detector can be considered as a sum of several different contributions: • the signal obtained from the microscope $S(k)$, which is multiplied by the gain factor $g(k)$ of the pixel; • the background signal of each pixel $b(k)$; and • the noise present at each pixel $N(k)$.
Source	[3]
Ref. By	DD3, IM3, IM4

Number	T2
Label	**Signal**
Equation	$S(k) = \int_{pixel} d\rho \lfloor \int f(\rho')h(\rho,\rho')d\rho' \rfloor$ (continuous) $S(k) = \sum_{k'} f(k')h(k,k')$ (discrete)[Maybe in brackets after each option for $S(k)$ you should mention which version it is (continuous versus discrete). Otherwise it is confusing why you define a function twice. —SS][Ok, that makes sense. —Author]
Description	The signal $S(k)$ (continuous) from the microscope, as read at one pixel on the CCD (k), is an integral over the area of one pixel, and becomes $S(k)$ (discrete). Inside the integral of the continuous function is the probability ($f(\rho')$) of an electron landing at position ρ' on the detector given the interaction of the beam with the sample, multiplied by the probability ($h(\rho,\rho')$) of detecting an electron at location ρ if the electron arrives at position ρ' on the detector. The probability $h(\rho,\rho')$ of detecting an electron at one location after it hits the detector at another location is the point spread function (PSF: DD3). Discretization of the Signal is done by the camera; each pixel in the camera integrates signal over its area, discretizing a continuous Signal into discrete Signal at each pixel. [When do you use the continuous versus the discrete version? —SS][I have tried to clarify this in the text. —Author] The second equation is the discretized version of the first. The signal $S(k)$ is the sum over all pixels k' of the probability ($f(k')$) of an electron hitting pixel k' multiplied by the probability ($h(k,k')$) of detecting an electron at k after it hits pixel k'.
Source	[3]
Ref. By	T1, DD1, DD3, IM2

5.2.3 General Definitions

This section provides a definition beneficial to understanding the data definitions.

Number	GD1
Label	**Scanning grid**
Units	nm
Equation	$P(x, y) = \{\langle x, y\rangle \in A \mid x \in \{0..X\}, y \in \{0..Y\}\}$
Description	$P(x, y)$ is a set of position coordinates describing the possible locations of the beam in two spatial coordinates (x, y) during a raster scan across the sample in a rectangular grid pattern across the area on the sample, A^2. The limits on (x, y) are set by the operator on the instrument, ranging from 0 to the maximum (X, Y). [Do you really mean A^2? I would have thought you just meant A. —SS] [I had a reason for doing that, but now I don't remember why...I have changed it —Author]
Ref. By	DD2

5.2.4 Data Definitions

This section collects and defines the data types needed to build the instance models.

Number	DD1
Label	**Spectrum**
Symbol	$I(E)$
Units	Electron counts (EELS) Photon counts (CL) or arbitrary units (a.u.) for EELS or CL
Equation	$I(E) = \zeta(k = 0..K)$
Description	Each pixel (k) on the spectrometer CCD has an energy loss (EELS) or wavelength (CL) value associated with it (E, where E is also function of k), obtained from the spectrometer acquisition settings, and a collected signal ($\zeta(k)$, T2). The intensity of the spectrum ($I(E)$, in electron counts for EELS and photon counts for CL) is obtained from the collected signal over the whole CCD, from the 0th pixel to the final (Kth) pixel on the CCD. [Which count applies is not clear. Does the user select this, or does it change depending on the type of imaging, or something else? This might be something you can address under your requirements? —SS][I've tried to clarify this: when you are doing EELS, you are counting electrons at each pixel, when you are doing CL, you are counting photons at each channel, so it depends on the type of experiment you did. It's covered in the modules via an input argument. —Author]
Sources	[4]
Ref. By	T2, DD2, DD3, IM1, IM2, IM3, IM4

Number	DD2
Label	**Spectrum Image**
Symbol	$I(x, y, E)$
Units	Electron counts (EELS) Photons (CL) or arbitrary units (a.u.) for EELS or CL
Equation	$I(x, y, E) : \forall \langle x, y \rangle \in P(x, y), \exists I(x, y, E) \in \mathbb{R}$
Description	For all the pixels ((x, y) pairs) within the scanning grid acquired ($P(x, y)$, GD1), there is a Spectrum, ($I(x, y, E)$, DD1), collected by the spectrometer. [I suggest you restate the math from the above equation in natural language. —SS][I have tried to rephrase this, though I think it is very similar to what I said before and I'm not sure what you want me to change... —Author]
Sources	[5]
Ref. By	DD3, IM1, IM2

Number	DD3
Label	**Point Spread Function**
Symbol	$I_{PSF}(E)$
Units	Electron counts or arbitrary units (a.u.) (EELS)
Equation	$I_{PSF}(E) = \zeta(k) = g(k)S'(k) + b(k) + N(k),\ k \in [0..K]$ [defining the limits on k inside the function seems odd to me. Maybe you could just say $\zeta(k)$ and then follow the equation with a comma and $k \in [0..K]$? —SS][Sure, ok —Author]
Description	$I_{PSF}(E)$ is a reference spectrum ($I(E)$, DD1) representing the point spread function (see T2) on the collected spectra in the spectrum image (DD2). It is a spectrum collected, ideally, with no interaction with a sample; the variables in the equation are defined as in T1, with the exception of $S'(k)$, which is an example of $S(k)$ with no sample interaction with the beam.
Sources	[3, 5, 6]
Ref. By	T2, IM2

5.2.5 Instance Models

This section details the instance models, which are the algorithms that the code must perform to process spectrum imaging data, given the data types defined in Data Definitions above.

The data processing goal GS2 is solved by the following instance models (IM1, IM2, IM3 IM4).

Number	IM1
Label	**Normalization to the integral**
Input	k_1 is the index of beginning of the spectrum range (spectrum axis), user input k_2 is the index of end of the spectrum range (spectrum axis), user input $I(E)$ is the spectrum, user input The input is constrained such that $k_1 >= 0$, and $k_2 <= N$
Equation	$I_{norm}(E) = \frac{I(E)}{\sum_{E(k=k_1)}^{E(k=k_2)} I(E(k))}$ [Technically, LaTeX will interpret norm in the equation as n times o times r ... in terms of spacing. You probably actually want I_{norm}. Your reviewer (Keshav) noted that the denominator looks odd for this equation. I agree. :-) —SS][I have changed the subtext and moved the misplaced bracket! —Author]
Output	Normalized spectrum
Description	$I(E)$ is a spectrum (DD1), user input. k_1: Index of beginning of spectrum range (spectrum axis), user input. k_2: Index of end of spectrum range (spectrum axis), user input. K: Last index along the spectral axis. x: Location along first spatial axis. y: Location along second spatial axis. $I_{norm}(E)$: Spectrum normalized along the spectral axis to the integral defined by a user. Note that this algorithm can be applied to a 3D spectrum image (DD2) by applying the algorithm to each (x, y) pixel independently.
Sources	–
Ref. By	IM2

Number	IM2
Label	**Richardson-Lucy Deconvolution**
Input	$I_{PSF}(E)$ is the point spread function of the instrument (see DD3). $I^0_{real}(E)$ is the initial guess for the "real" spectrum (DD1). $I_{measured}(E)$ is the measured spectrum, as obtained from the spectrometer software (DD1). R is the total (non-negative integer) number of iterations of the deconvolution algorithm applied to the spectrum. c is the number of the current iteration (at iteration 0, $c = 0$). i, j, l are each indices of pixels in the spectrum. Three different variables are used for the pixel index to make the effect of the different summations clear.
Equation	$I^{c+1}_{real(j)}(E) = I^c_{real(j)}(E) \sum_i \frac{I_{PSF}(E) I_{measured}(E)}{\sum_l I_{PSF}(E) I^c_{real(l)}(E)}$, while $c < R$
Output	$I_{deconvolved}(E_k) = I^R_{real(j)}(E)$
Description	The purpose of the RL deconvolution algorithm is to separate the effects of the spectrometer point spread function from the "real" signal of the sample. *I.e.* in T2, the signal is described as the convolution of the point spread function with the signal from the microscope. Deconvolution is usually run after normalization of the spectra (IM1) to eliminate effects of changes in the beam current. This algorithm can also be expanded to apply to the 3D spectrum image (DD2)) by applying it at every (x, y) pixel in the spectrum image.
Sources	[6, 7]
Ref. By	–

Number	IM3
Label	**Background subtraction**
Input	$I_{collected}$ is the spectrum as collected by the spectrometer. $I_{background}(E)$ is a spectrum representing the background to be subtracted (this could be associated with the dark signal or with the signal coming from unwanted sources: *e.g.* outside noise, substrate signal...).
Equation	$I_{sample} = I_{collected} - I_{background}$
Output	$I_{sample}(E)$
Description	This model addresses the removal of background contributions to the signal ($b(k)$ in T1), where $b(k)$ is transformed to a spectrum (DD1) as a function of energy. The background spectrum must be of the same dimensions as the spectrum it is to be removed from. For the spectrum image, the background spectrum may be applied independently to each pixel.
Sources	–
Ref. By	–

Number	IM4
Label	**Gain correction**
Input	$I_{collected}$ is the spectrum (DD1) of the signal as collected by the spectrometer (see T1). $g(E)$ is the gain "spectrum" of the spectrometer $g(k)$ transformed into a function of energy (DD1).
Equation	$I_{corrected}(E) = \frac{I_{collected}(E)}{g(E)}$
Output	$I_{corrected}(E)$ is the spectrum after the gain correction from the spectrometer CCD is applied.
Description	This model addresses the removal of the gain ($g(k)$ in T1) from the collected signal, where $g(k)$ is transformed to a "spectrum" (DD1) as a function of energy.
Sources	–
Ref. By	–

5.2.6 Data Constraints

This section details the constraints on the data that the software must process, including the input, processing steps, and output from the software. Physical constraints based on the expectations of the dataset are left out to allow the user more freedom in the input. The constraints on the data are defined as

- The input spectrum image (DD2) must be a 3D structure, or a 3D structure contained within a .dm3 or .h5 file.
- The user input number of iterations to use for the deconvolution algorithm must be an integer number greater than or equal to 0.
- For the normalization model, the user input of the number of bins to normalize must not be empty.
- The user input of the background signal (IM3) must be of the same length along the spectral axis as the spectrum image it will be applied to.
- The user input of the gain signal (IM4) must be of the same length along the spectral axis as the spectrum image it will be applied to.
- The normalization (IM1) and deconvolution (IM2) data processing algorithms both include a division by experimentally-derived values. The denominator in these steps must not be 0.

5.2.7 Properties of a Correct Solution

A correct solution must be acceptable to a user experienced with EELS and/or CL data processing, as is relevant to the EELS/CL data being processed.

6 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

6.1 Functional Requirements

R1: Accept and read the following as Spectrum Image inputs:

- 3D data array (*e.g.*, already stored in computer memory)
- File in format .dm3 containing 3D data array
- File in format .h5 containing 3D data array

R2: Accept and read the following as Spectrum inputs:

- 1D data array
- File containing 1D data array in .csv format

R3: Verify that all data inputs are of the correct dimensionality and composed of Real numbers

R4: Accept user input to select a spectral range and extract (x, y) slices averaged over the selected spectral range and display the (x, y) slice as an image. The software should also export this image to an image file, as desired by the user.

R5: Accept user input to define an area on an (x, y) image display and extract a spectrum averaged over selected pixels. The software should also export this spectrum as a *.csv file, as desired by the user.

R6: Given a 1D spectrum array and an integer number, the software should perform that number of deconvolution iterations on a 3D EELS dataset using the 1D spectrum array as a reference of the system response (IM2). The software should display the output of the deconvolution for navigation by the user. Output of the deconvolution algorithm will be validated by an expert user.

R7: The software should be able to normalize a 3D spectrum image (IM1) upon user command: either to an integrated portion of a chosen range along the spectral axis or to a single chosen channel.

R8: The software should be able to perform a background subtraction (IM3), given the appropriate inputs of a spectrum or spectrum image and a background reference spectrum.

R9: The software should be able to perform a gain correction (IM4), given the appropriate inputs of a spectrum or spectrum image and a gain reference spectrum.

[Your requirements should reference explicitly the instance models that apply. This information is in your traceability matrices, but it should also be explicit in the definition of the requirements themselves. —SS][Perhaps you can include a comment about this in the template, because this wasn't clear at all. For my type of software, it is also the case that not all requirements are based on an instance model. I have added the cross-reference for those which do. —Author]

6.2 Nonfunctional Requirements

The nonfunctional priorities for the software are the correctness, useability, speed, maintainability, expandability, and understandability. The user must be able to easily use the software to extract the data that they want and do the processing operations that they want,

given the user characteristics defined elsewhere. The speed of performance is of interest in the interactive display of the 3D datasets and in the data processing operations, which can be computationally heavy with large datasets. To improve the software over time and add new functionality as the data processing field expands, the software must be both maintainable and understandable by the original programmer and future programmers. In addition, a more advanced user may wish to add their own functions into the software data processing algorithms; the software should be written in such a way to make expanding its functionality simple and understandable. A user may also wish to perform their own tests of the software functions, emphasizing the need to have transparent and understandable software. To reach a wide user base, the program should be usable on many different operating systems. The nonfunctional requirements are listed below:

- The software should be maintainable by the original programmer or future programmers to keep up to date with advances in data processing practices.
- The software should allow the addition of more data processing functions as required by an advanced user experienced in programming.
- The software should not display significant lag when interacting with the user, except in the case of data processing steps, which may be computationally expensive.
- Use of the interactive display and of data processing functions should be intuitive for a user experienced with the processing techniques for 3D datasets.
- The coding practices for the software should be open and transparent to facilitate inspection by an experienced user to encourage trust in the data processing steps performed.
- The display for the user to interact with should be readable: *i.e.* the text should be large enough to read by a user with average eyesight, with high contrast against the background for visibility.
- The colours used in the display should be clearly distinguishable from each other.
- The software should function on multiple operating systems, including Windows, Linux, and MacOS, in a laptop or desktop environment.

7 Likely Changes

This section details some changes which are likely to be made to the software once the initial requirements are met, in the interests of continuous improvement of the software's capabilities.

LC1: A PSF unique to each pixel in the spectrum image (DD2) will be used instead of assuming one PSF is valid for all pixels within a spectrum image (A2)

LC2: Cosmic rays appear as high intensity single pixel spikes in a spectrum and can be removed via 3D image filtering techniques. A future change will implement this filtering option for the user.

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 2 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 3 shows the dependencies of instance models, requirements, and data constraints on each other. Table 1 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

	A1	A2	A3	A4	A5
T1				X	X
T2	X		X		
GD1		X			
DD1	X				
DD2	X	X			
DD3	X	X			
IM1			X		
IM2	X	X			
IM3				X	
IM4					X

Table 1: Traceability Matrix Showing the Connections Between Assumptions and Other Items

	T1	T2	GD1	DD1	DD2	DD3	IM1	IM2	IM3	IM4
T1		X							X	X
T2										
GD1										
DD1	X								X	
DD2			X	X						
DD3	X			X				X		
IM1		X								
IM2	X	X				X				
IM3	X									
IM4	X									

Table 2: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM2	IM3	IM4	5.2.6	R1	R2	R3	R4	R5	R6	R7	R8	R9
IM1						X						X		
IM2	X					X		X			X			
IM3						X	X	X					X	
IM4						X	X	X						X
R1					X	X		X		X				
R2					X			X			X			
R3					X	X	X		X	X			X	X
R4														
R5														
R6		X					X					X		
R7	X													
R8			X				X							
R9				X			X							

Table 3: Traceability Matrix Showing the Connections Between Requirements and Instance Models

References

[1] W. S. Smith and L. Lai, "A new requirements template for scientific computing," in *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05* (J. Ralyté, P. Ågerfalk, and N. Kraiem, eds.), (Paris, France), pp. 107–121, In conjunction with 13th IEEE International Requirements Engineering Conference, 2005.

[2] S. Smith, L. Lai, and R. Khedri, "Requirements Analysis for Engineering Computation: A Systematic Approach for Improving Reliability," *Reliable Computing*, vol. 13, pp. 83–107, Feb. 2007.

[3] J. M. Zuo, "Electron detection characteristics of a slowscan CCD camera, imaging plates and film, and electron image restoration," *Microscopy research and technique*, vol. 49, no. 3, pp. 245–68, 2000.

[4] R. F. Egerton, "An Introduction to EELS," in *Electron Energy-Loss Spectroscopy in the Electron Microscope*, pp. 1–28, Springer US, Jan. 2011.

[5] C. Jeanguillaume and C. Colliex, "Spectrum-image: The next step in EELS digital acquisition and processing," *Ultramicroscopy*, vol. 28, pp. 252–257, Apr. 1989.

[6] A. Gloter, A. Douiri, M. Tenc, and C. Colliex, "Improving energy resolution of EELS spectra: an alternative to the monochromator solution," *Ultramicroscopy*, vol. 96, pp. 385–400, Sept. 2003.

[7] 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.