SpectrumImageAnalysisPy

Isobel Bicket

October 4, 2017

1 Revision History

Date	Version	Notes
October 4, 2017	1.0	First draft

Contents

2 Reference Material

This section records information for easy reference.

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**
nm	length	nanometre
eV	energy	electron volt

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**
I		Intensity

2.3 Abbreviations and Acronyms

symbol	description
3D	Three-dimensional
A	Assumption
CCD	Charge-Coupled Device
CL	Cathodoluminescence Spectroscopy
DD	Data Definition
EELS	Electron Energy Loss
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
SEM	Scanning Electron Microscope
SpectrumImageAnalysisPy	[put your program name here —SS]
T	Theoretical Model

[Add any other abbreviations or acronyms that you add —SS]

3 Introduction

T

3.1 Purpose of Document

This document details the requirements of the software SpectrumImageAnalysisPy.

3.2 Scope of Requirements

The scope of the requirements for the software SpectrumImageAnalysisPy is limited to the import; removal of instrumentation artifacts; visualization and navigation through; and the 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 the data processing methods used to remove effects of the acquisition system from the data acquired. A basic knowledge of convolution theory will be helpful to the reader and an understanding of the characteristics of 3D datasets. The reader should understand EELS and CL, as relevant to the sections of the document.

3.4 Organization of Document

The document follows the organizational scheme laid out by Smith *et al* [1, 2].

4 General System Description

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

4.1 System Context

Use of the software SpectrumImageAnalysisPy infers the following responsibilities on the part of the user, and confers the following responsibilities from the program.

- User Responsibilities:
  - Provide the correct input data to the program;
  - Be capable of interacting with the software via a mouse and keyboard;
  - Have the necessary dependencies for the program installed;

  - 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;

- SpectrumImageAnalysisPy Responsibilities:
  - 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 as appropriate, including visualization and data processing commands
  - 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.

5 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models. [Add any project specific details that are relevant for the section overview. —SS]

5.1 Problem Description

SpectrumImageAnalysisPy is a software to allow users to import, process, navigate, and export 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:

- Charge-Coupled Device: the camera used in the spectrometer to collect signal and output to the microscope acquisition software
- Spectrum Image
- Point Spread Function (PSF)
- Sample

5.1.2 Physical System Description

The physical system of SpectrumImageAnalysisPy, as shown in Figure ?, includes the following elements:

PS1: Sample under study.

PS2: STEM equipped with EEL spectrometer. [Find an image/schematic of Titan —Author]

PS3: SEM or STEM equipped with CL collection system. [Find an image/schematic of SEM-CL system —Author]

[A figure here may make sense for most SRS documents —SS]

5.1.3 Goal Statements

Given the [inputs —SS], the goal statements 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; normalization and background subtraction for a CL SI

GS3: Extract slices from the dataset as desired by the user and communicated through the user interface, export these as desired by the user

5.2 Solution Characteristics Specification

The instance models that govern SpectrumImageAnalysisPy are presented in Subsection 5.2.5. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

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 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 that SpectrumImageAnalysisPy is based on. [Modify the examples below for your problem, and add additional models as appropriate. —SS]

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	

Number	T2
Label	**Signal**
Equation	$S(k) = \int_{pixel} d\rho \lfloor \int f(\rho')h(\rho, \rho')d\rho' \rfloor$ $S(k) = \sum_{k'} f(k')h(k, k')$
Description	The signal ($S(k)$) from the microscope, as read at one pixel on the CCD (k), is an integral over the area of one pixel. Inside the integral 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). 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

5.2.3 General Definitions

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

Number	GD1
Label	Scanning grid
Units	nm
Equation	$P(x, y) = \{\langle x, y\rangle \in A^2 \mid x \in \{0..X\}, y \in \{0..Y\}\}$
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).
Ref. By	DD2

5.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given. [Modify the examples below for your problem, and add additional definitions as appropriate. —SS]

Number	DD1
Label	**Spectrum**
Symbol	$I(E)$
Units	Electron counts (EELS) Photon counts (CL) or arbitrary units (a.u.) for EELS or CL
Equation	$E = f(k)$ $I(E) = \zeta(k = 0..K)$
Description	Each pixel (k) on the spectrometer CCD has an energy loss (EELS) or wavelength (CL) value associated with it (E), obtained from the spectrometer acquisition settings (denoted here by $f(k)$), and a collected signal ($\zeta(k)$, T2). The intensity of the spectrum ($I(E)$) is obtained from the collected signal over the whole CCD, from the 0th pixel to the final (Kth) pixel.
Sources	[4]
Ref. By	DD2

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	$I(x, y, E)$ is a dataset composed of Spectra (DD1) collected by the spectrometer detector at each value of $P(x, y)$ (GD1).
Sources	[5]
Ref. By	

Number	DD3
Label	**Point Spread Function**
Symbol	$I_{PSF}(E)$
Units	Electron counts or arbitrary units (a.u.) (EELS)
Equation	$I(E) = \zeta(k = 0..K) = g(k)S'(k) + b(k) + N(k)$
Description	$I_PSF(E)$ is a reference spectrum ($I(E)$, DD1) representing the effect of 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	IM??

5.2.5 Instance Models

This section transforms the problem defined in Section 5.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 5.2.4 to replace the abstract symbols in the models identified in Sections 5.2.2 and 5.2.3.

The goals [reference your goals —SS] are solved by [reference your instance models —SS]. [other details, with cross-references where appropriate. —SS] [Modify the examples below for your problem, and add additional models as appropriate. —SS]

Number	IM1
Label	**Normalization to the integral**
Input	k_1 k_2 $I(E_k)$ The input is constrained such that $k_1 >= 0$, and $k_2 <= N$
Output	$I_{norm}(E_k) = \frac{I(E_k)}{\sum_{E_{k=k_1}}^{E_{k=k_2}} I(x,y,E_k)}$
Description	$I(x, y, E_k)$ is the 3D spectrum image, 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: Pixel index along first spatial axis y: Pixel index along second spatial axis $I_{norm}(x, y, E_k)$: Spectrum image with each (x, y) pixel normalized independently along the spectral axis
Sources	
Ref. By	

Number	IM2
Label	**Richardson-Lucy Deconvolution**
Input	$I_PSF(E)$ is the point spread function of the instrument (see DD3) $I^0_{real}(x, y, E)$ is the initial guess for the "real" spectrum (DD1) $I_{measured}(x, y, E)$ is the measured spectrum, as obtained from the spectrometer software (DD1) R is the number of iterations of the deconvolution algorithm applied to the spectrum
Equation	$I^{c+1}_{real(b)}(E) = I^c_{real(b)}(E) \sum_a \frac{I_{PSF}(E) I_{measured}(E)}{\sum_d I_{PSF}(E) I^c_{real(d)}(E)}, while\ c < R$
Output	$I_{deconvolved}(E_k) = I^R_{real(b)}(E)$
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.
Sources	[6, 7]
Ref. By	

Number	IM3
Label	**Background subtraction**
Input	$I_{collected}$ is the (see DD3) $I^0_{real}(x, y, E)$ is the initial guess for the "real" spectrum (DD1)
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).
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 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 (eg, 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 the data input is 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. 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 upon user command: either to an integrated portion of a chosen range along the spectral axis or to a single chosen channel.

R8: [It isn't always required, but often echoing the inputs as part of the output is a good idea. —SS]

R9: [Calculation related requirements. —SS]

R10: [Verification related requirements. —SS]

R11: [Output related requirements. —SS]

6.2 Nonfunctional Requirements

The nonfunctional priorities for the software are the 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. 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 this simple and understandable. 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. A user may 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 1 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 2 shows the dependencies of instance models, requirements, and data constraints on each other. Table 3 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

[You will have to modify these tables for your problem. —SS]

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure **??** shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Figure **??** shows the dependencies of instance models, requirements, and data constraints on each other.

	T??	T??	T??	GD??	GD??	DD??	DD??	DD??	DD??	IM??	IM??	IM??	IM
T??													
T??			X										
T??													
GD??													
GD??	X												
DD??				X									
DD??				X									
DD??													
DD??								X					
IM??					X	X	X				X		
IM??					X		X		X	X			X
IM??		X											
IM??		X	X				X	X	X		X		

Table 1: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM??	IM??	IM??	IM??	5.2.6	R??	R??
IM??		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R8	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R10			X	X			
R??		X					
R??		X					

Table 2: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
T??	X																		
T??																			
T??																			
GD??		X																	
GD??			X	X	X	X													
DD??							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM??											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 3: Traceability Matrix Showing the Connections Between Assumptions and Other Items

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.

9 Appendix

[Your report may require an appendix. For instance, this is a good point to show the values of the symbolic parameters introduced in the report. —SS]

9.1 Symbolic Parameters

[The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. —SS]