STXM Data Analysis Package

–

MATLAB Script Collection

T.R. Henn* R.C. Moffet#

* *Physikalisches Institut, Julius-Maximilians-Universität, Würzburg, 97074, Germany* **tobias.henn@physik.uni-wuerzburg.de**

# *Chemical Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA* **rcmoffet@lbl.gov**

Contents

1 Introduction

Much of the current research focuses on structures and phenomena on a length scale from nanometers to micrometers which creates a growing need for detailed compositional and chemical analysis at high spatial resolution. Hence, Scanning Transmission X-ray Microscopy (STXM) is therefore becoming an increasingly important instrument by providing spatially resolved compositional information using *Near Edge X-ray Absorption Fine Structure* (NEXAFS) spectroscopy.

STXM produces 3D-datasets containing spectral information at high spatial resolution. A typical STXM *stack* consists of ~ 100 single absorption images with $\sim 100 \times 100$ pixels, i.e. $\sim \mathcal{O}(10^6)$ data points. STXM studies can produce a large number of stacks, thereby giving rise to the need of automated data analysis.

The purpose of this MATLAB script collection is the automated processing of large sets of STXM raw data. The scripts allow batch processing of STXM raw data folders with automated *stack alignment* and *optical density conversion*. After the import and conversion process MATLB built-in functions can be used for further data analysis. The script collection also provides a MATLAB GUI application called **STACKLab** which allows the exploration of individual stacks and the extraction of spectra from user-defined regions of interest.

Examples for the advanced automated analysis of STXM data using MATLAB are generation of size distributions of the chemical composition of atmospheric particles, mapping of chemical components in samples using singular value decomposition or the creation of radial distance scans showing the oxidation state of iron containing dust particles as a function of distance from the particle's center of mass.

1.1 Software requirements and acknowledgement

All Scripts were tested with MATLAB R2008b (Version 7.7.0.471). Most of the scripts (aside from the raw data import script) require the MATLAB **Image Processing Toolbox**. The alignment script falls back on the implementation of an efficient subpixel image registration algorithm by M. Guizar-Sicairos. The design of the data structure array is inspired by S. Takahama.

2 Scanning Transmission Microscopy Data

2.1 STXM raw data

One of the most basic data sets recorded by a Scanning Transmission X-ray Microscope (STXM) is a single energy image. The sample is raster-scanned by a focused, monochromatic X-ray beam and the transmitted photon intensity through the sample is recorded by a computer at discrete sample positions y_n and x_m (*pixels*) as a function of the sample position. The information thereby acquired can be stored in an intensity matrix:

$$I_{nm} = I(y_n, x_m) \tag{2.1}$$

To obtain spectral information, that is, the energy dependence of the sample's photon absorption, the image recording process is repeated over the same sample region at different X-ray energies. The resulting data is referred to as a *stack*. The stack is a set of intensity matrices that combined contain the information about the transmitted photon intensity as a function of photon energy E for each pixel (the absorption spectrum of the sample for each pixel position).

$$I_{nmE} = I(y_n, x_m, E) \tag{2.2}$$

The transmitted intensity depends not only on the intensity of the incident X-ray beam and the sample properties (e.g. thickness, chemical composition, and mass density), but also on the substrate absorption properties. The data described above is therefore considered to be *raw data*. Before carrying out spectral image data analysis, the raw data is converted to *optical density*.

2.2 Conversion of STXM raw data to optical density

When X-ray radiation passes through matter its absorption depends on the elementary composition of the matter, its thickness and the density of the sample. The exponential attenuation of an incident beam with intensity I_0 transmitted through a homogenous sample of thickness t and mass density ρ with an mass absorption coefficient $\mu(E, Z)$ is given by Beer-Lambert's Law:

$$I(t) = I_0 e^{-\mu(E,Z)\rho t} \tag{2.3}$$

Here $I(t)$ is the transmitted X-ray intensity. The mass absorption coefficient and therefore the transmitted X-ray intensity strongly depends on the photon energy $E = \hbar\omega$ and the atomic number Z. Introducing the atomic density n_a the above expression can also be written in terms of the X-ray absorption cross section σ_X showing the linear proportionality of the absorption cross section to the mass absorption coefficient ([1]):

$$I(t) = I_0 e^{-n_a \sigma_X t} \tag{2.4}$$

In X-ray absorption spectroscopy and microscopy quantitative analysis is carried out by converting the transmitted intensity to the dimensionless *optical density* (OD) which is given by the relationship [2]:

$$OD = \ln\left(\frac{I_0}{I(t)}\right) \tag{2.5}$$

The conversion to optical density removes the raw data's dependence on the incident X-ray intensity and substrate properties and allows physically meaningful comparison of data. The optical density is linearly related to the sample properties by the mass absorption coefficient $\mu(E, Z)$ and to the X-ray absorption cross section σ_X:

$$OD(E) = \mu(E, Z)\rho t = n_a \sigma_X(E, Z)t \tag{2.6}$$

The optical density of a sample containing j different, noninteracting compounds is given by the linear superposition of the optical densities of the single compounds:

$$OD(E) = \sum_{k=1}^{j} \mu_k(E, Z)\rho_k t_k \tag{2.7}$$

where t_k denotes the “relative path” through each constituent with mass absorption coefficient μ_k and density ρ_k. The concept of ”relative paths” is used when the components are not in pure phases but rather are mixed [2].

3 Data preparation

Before starting the actual STXM data analysis, three main steps are necessary:

$$\textit{raw data import} \longrightarrow \textit{stack alignment} \longrightarrow \textit{conversion to optical density}$$

This section describes the methods used to automate the above steps and explains the usage of the corresponding MATLAB scripts.

3.1 STXM raw data import

The STXM instruments at the Advanced Light Source beamlines 11.0.2 and 5.3.2 store their stack raw data as a collection of ASCII files in individual folders. The folder name comprises information about the beamline number, the date the stack was recorded and a continuous number that indicates the number of scans taken on a particular day. The folder contains a header file (*.hdr*) and one file for each image recorded at a certain X-ray energy (*.xim*):

```
11_081025012.hdr
11_081025012_a000.xim
11_081025012_a001.xim
11_081025012_a002.xim
....
```

The header file contains information about important storage ring and beamline parameters (storage ring current, exit slits etc.), but also about sample position, image size in μm, number of pixels for each axis and the energy range of the stack. The actual raw photon counts are stored separately for each photon energy as a plain text matrix in the *.xim* files.

The STXM raw data can be imported to MATLAB using the script **LoadStackRaw.m**. The following example shows the usage of **LoadStackRaw** to import a stack from the folder */Users/tobi/Desktop/example_stack/* in which the *.hdr* and *.xim* files are located:

```
>> Sraw=LoadStackRaw('/Users/tobi/Desktop/example_stack')

Sraw =

eVenergy: [95x1 double]
Xvalue: 7.2540
Yvalue: 6.3200
spectr: [158x181x95 double]
```

The output of the import script is a four field structure array. The *eVenergy* field is a vector containing the k photon energies used to record the STXM absorption images. The *Xvalue* and *Yvalue* fields correspond to the length of the horizontal and vertical image axis in μm. The actual image data is stored in the $n \times m \times k$ array *spectr*, a data cube with one dimension in X-ray energy and two dimensions in spacial position (x and y) [3]. Here n and m are the number of pixels along the vertical and horizontal axis. k is the total number of images and is therefore equal to the length of the *eVenergy* vector. The lth element in the *eVenergy* vector corresponds to the *spectr* data points with the third index equal to l.

3.1.1 Examples: basic stack structure array operations

The following basic examples are intended to illustrate the usage of the script collection's standard stack data structure array. First, the photon energies (in eV) are printed to the screen:

```
>> Sraw.eVenergy

ans =

  278.0000
  278.5000
  279.0000
  279.5000
  ...
```

The example stack was recorded at energies around 290 eV at the carbon *K-edge*. Next, the index of the first image recorded at photon energies above 289 eV is calculated:

```
>> idx=find(Sraw.eVenergy >= 289 ,1, 'first')

idx =

    44
```

This absorption image is then plotted in grayscale with a color bar using this index:

```
>> figure
>> imagesc([0,Sraw.Xvalue],[0,Sraw.Yvalue],Sraw.spectr(:,:,idx))
>> colormap gray
>> colorbar
```

The information about the x and y image axis lengths was used for axis labeling. The result of this plot command is shown in Figure 3.1.

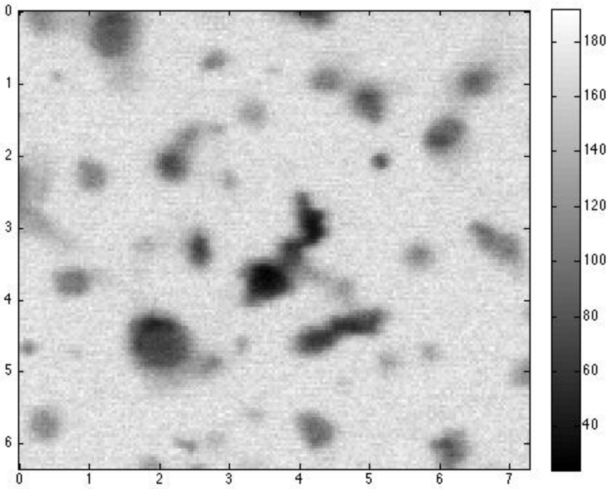

Figure 3.1: Absorption image of atmospheric particles at photon energy E=289.2 eV (data in raw counts, axis scale in μm)

3.2 Image registration (stack alignment)

The process of recording a stack typically takes from a couple of minutes up to several hours. During the recording process the sample holder is moved in the x and y directions by piezo actuators. Thermal drifts and other mechanical processes can cause misalignment of the stack images. Since NEXAFS spectra are generated by analyzing the subsequent intensity values for individual image pixels *stack alignment* is a crucial step in STXM data analysis and needs to be performed before converting the raw data to optical density.

Stacks can be automatically aligned using the script **FTAlignStack.m**. The usage of the script is shown in the following example:

```
>> S_Aligned=FTAlignStack(Sraw)

S_Aligned =

eVenergy: [95x1 double]
Xvalue: 6.8131
Yvalue: 6.2400
spectr: [156x170x95 double]
```

To reduce the signal noise, the mean of three images, taken from the third quarter of the stack, is used for the alignment reference image. Other images are aligned to this reference image. The image shifts used to align the images to the this reference image are calculated with a precision of 0.1 pixels using an efficient subpixel image registration algorithm [4]. The image registration algorithm first estimates the peak position in the *2D cross correlation* of the two images by using a Fast Fourier Transform (FFT) method with an upsampling factor $\kappa_0 = 2$. To reduce calculation time, only a small region around this estimated position is then analyzed using a higher upsampling factor κ to achieve the desired accuracy of locating the precise position of the cross correlation peak. The image shifts are given by the displacement of this peak from the cross correlation origin. These non-integer image shifts are performed applying the *shift theorem* by adding the corresponding linear phases to the image values in Fourier space [5]. The phase modification leads to a circular shift in the real space image. Image regions that enter or leave the original stack image boundaries during the stack recording (due to drift processes during data acquisition) are removed from the stack and the lengths of the image axis are updated. The aligned stack is returned by the **FTAlignStack** script as a new structure array.

3.3 Conversion to optical density

The last step of STXM data preparation is the conversion of the raw data (in counts) into *optical density.* The conversion is performed using the **OdStack.m** script. The core function of the script is the detection of sample-free regions in the stack images. These sample-free regions are used to extract the average background spectrum $I_0(E)$. The script provides two different thresholding methods:

1. constant value thresholding (‘C’)
2. thresholding using Otsu’s method (‘O’)

Constant value thresholding maps the stack’s mean intensity values to grayscale values between 0 and 1. Regions with mean values above 0.985 are considered to be ”sample-free” and are used to extract the average background spectrum. Alternatively, Otsu’s method [6] can be used to automatically calculate the appropriate thresholding value. Our experience indicates that Otsu’s method is particularly useful in the case of torn substrate films where the constant threshold method would use only the torn region to extract the average background spectrum

The first input argument of **OdStack** is the aligned raw data structure array. The second input argument is a string indicating the threshold method (‘O’ or ’C’). The script returns the converted stack structure array and a diagnostic figure that can be used to validate the optical density conversion:

```
>> S=OdStack(S_Aligned,'O')

S =

eVenergy: [95x1 double]
Xvalue: 6.8131
Yvalue: 6.2400
spectr: [156x170x95 double]
Izero: [95x2 double]
```

The optical density conversion script adds a fifth field, *Izero*, to the output structure array. *Izero* is a two column matrix where the average spectrum $I_0(E)$ is stored.

Figure 3.2 shows the output plots produced by **OdStack**. The upper parts of the figure show the stack's mean raw intensity and the mean optical density, the lower parts show the mask defining the region used to extract the background spectrum (white) and the extracted average spectrum of the sample-free region.

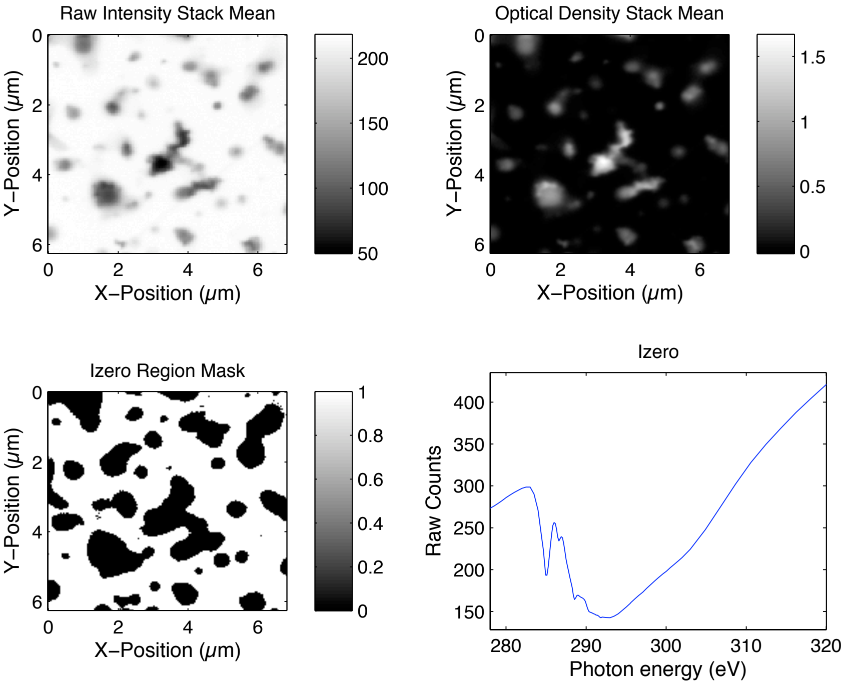

Figure 3.2: Diagnostic figure produced by the optical density conversion script **OdStack**

4 StackMovie - stack image sequence movies

The MATLAB script collection contains the function **StackMovie**. It can be used to view a movie by subsequently showing the single absorption images. Figure 4.1 shows the output of the script. The left half of the MATLAB figure shows the current STXM image. The right half of the figure displays the mean absorption spectrum of the sample. This spectrum is extracted by locating the image background using Otsu's method and calculating the average absorption spectrum of all non-background pixels.

The absorption image sequence movie is played by passing the data structure of the stack as the single argument to the function **StackMovie**:

```
>> StackMovie(S)
```

StackMovie is a useful tool to inspect the result of the stack alignment process. It can also be used to quickly locate components with known absorption peak energies in the sample. The components will appear at the corresponding energy positions during the stack movie.

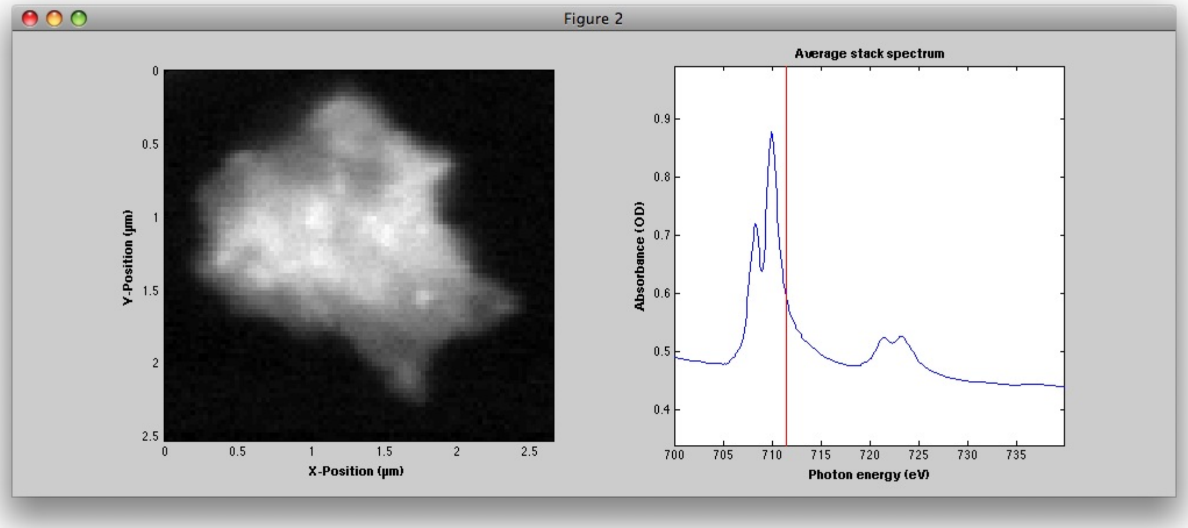

Figure 4.1: A snapshot of the stack movie generated by the **StackMovie** script

5 Quantitative component mapping

One of the widely used applications of X-ray spectro-microscopy is the generation of (quantitative) component maps. The distribution of different chemical components over a region of interest can be mapped by analyzing the stack data on a pixel-by-pixel basis using *reference spectra*. The STXM Data Analysis Package provides the script **stackSVD.m** that can be used to perform the necessary calculations by applying a *Singular Value Decomposition* (SVD) analysis on the stack data. The following section describes the working principle of the algorithm used in the **stackSVD** script, an example of the usage of the script is shown in the subsequent section.

5.1 Singular Value Decomposition analysis of STXM data

Section 2.2 introduced the concept of ”relative paths”. For a sample that contains j different, noninteracting compounds, the optical density of the sample at a photon energy E_i is the linear superposition of the optical densities of the single compounds:

$$OD(E_i) = \sum_{k=1}^{j} \mu_k(E_i)\rho_k t_k \tag{5.1}$$

SVD analysis is used to calculate the spatial distribution of the product $\rho_k t_k$ for each of the j compounds at each stack pixel. The stack data consists of a set of vectors $\overrightarrow{OD}$, each containing the absorption spectrum for one pixel at position (y, x). Defining the matrix $\mathbf{M}$ to hold the information about the mass absorption coefficients $\mu_k(E_i)$ for each compound at each photon energy

$$M_{ik} = \mu_k(E_i) \tag{5.2}$$

allows to write a set of equations similar to (5.1) for each pixel and energy:

$$\mathbf{M} \cdot \overrightarrow{\rho t} = \overrightarrow{OD} \tag{5.3}$$

where the vector $\overrightarrow{\rho t}$ is defined as $(\overrightarrow{\rho t})_k = \rho_k t_k$.

SVD STXM data analysis aims at solving the overdetermined (the number of compounds in the sample is typically lower than the number of different photon energies used to record a stack) set of linear equations (5.3) for the solution vector $\overrightarrow{\rho t}$ on a pixel-by-pixel basis. The set of all the solution vectors is then used to generate the component maps. Since one cannot expect (5.3) to have an exact solution vector for real STXM data and non-ideal mass absorption coefficients data, one is rather interested in a solution vector that minimizes the Euclidian norm

$$||\mathbf{M} \cdot \overrightarrow{\rho t} - \overrightarrow{OD}|| \tag{5.4}$$

This problem has a solution given by the *pesudoinverse* of the mass absorbtion coefficient matrix $\mathbf{M}^{+}$ [7]:

$$\overrightarrow{\rho t} = \mathbf{M}^{+} \cdot \overrightarrow{OD} \tag{5.5}$$

A efficient way to calculate the pseudoinverse $\mathbf{M}^{+}$ is to perform a SVD on the mass absorption coefficient, that is to calculate the matrices $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}$ with:

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{*} \tag{5.6}$$

For j sample compounds and a stack recorded at m different photon energies, $\mathbf{M}$ is a $m \times j$ matrix. $\mathbf{U}$ is a $m \times m$ unitary matrix, $\mathbf{\Sigma}$ is a $m \times j$ diagonal matrix with nonnegative real numbers, the *singular values*, on its diagonal, and $\mathbf{V}^{*}$ denotes the conjugate transpose of the unitary $j \times j$ matrix $\mathbf{V}$.

The pseudoinverse $\mathbf{M}^{+}$ that minimizes equation (5.4) in a least square sense is then given by the matrix product $\mathbf{M}^{+} = \mathbf{V}\mathbf{\Sigma}^{+}\mathbf{U}^{*}$ where $\mathbf{\Sigma}^{+}$ is the matrix $\mathbf{\Sigma}$ whith every nonzero entry replaced by its reciprocal [7]. The optimal solution vector can be calculated as

$$\overrightarrow{\rho t} = \mathbf{V}\mathbf{\Sigma}^{+}\mathbf{U}^{*} \cdot \overrightarrow{OD} \tag{5.7}$$

Generating an array containing the component $(\overrightarrow{\rho t})_k$ of the optimal solution vector for each stack pixel now yields a map showing the spatial distribution of the kth sample compound.

5.2 SVD analysis using stackSVD

The STXm data analysis package provides **stackSVD** to generate component maps. The script takes the standard stack structure array S and an arbitrary number of two-column vectors that store the reference spectra as an argument (first column: photon energy, second column: corresponding mass absorption coefficient) and returns a $n \times m \times j$ matrix array where n and m are given by the stack's number of pixels along both spatial axis and j corresponds to the number of reference spectra provided to the script:

```
ComponentMaps=stackSVD(S, spec1, spec2, spec3, ...)
```

The reference spectra can either be tabulated literature values or be extracted from sample regions that are known to be pure in one compound using the **STACKLab** program's spectrum export function. The information stored in the returned matrix array can be used to generate *RGB color maps* showing the distribution of the different components. The stack data and the reference spectra do not need to be recorded using the same energy resolution or energy range, the number of stack images must at least be equivalent to the number of reference spectra (while it typically will be higher by a factor of $10-100$). The following section provides an example of the usage of the **stackSVD** script and shows how to generate a RGB component map from the SVD analysis results.

5.2.1 Example: Mapping iron oxidation states in mineral dust

The following example illustrates the usage of the **stackSVD** script. The stack used in this examples consists of STXM images of an iron containing mineral dust particle recorded at the Fe $L_{2,3}$-edge, that is at photon energies between 700 and 740 eV. SVD analysis is used to map the distribution of the two iron oxidation states (Fe(II) and Fe(III)).

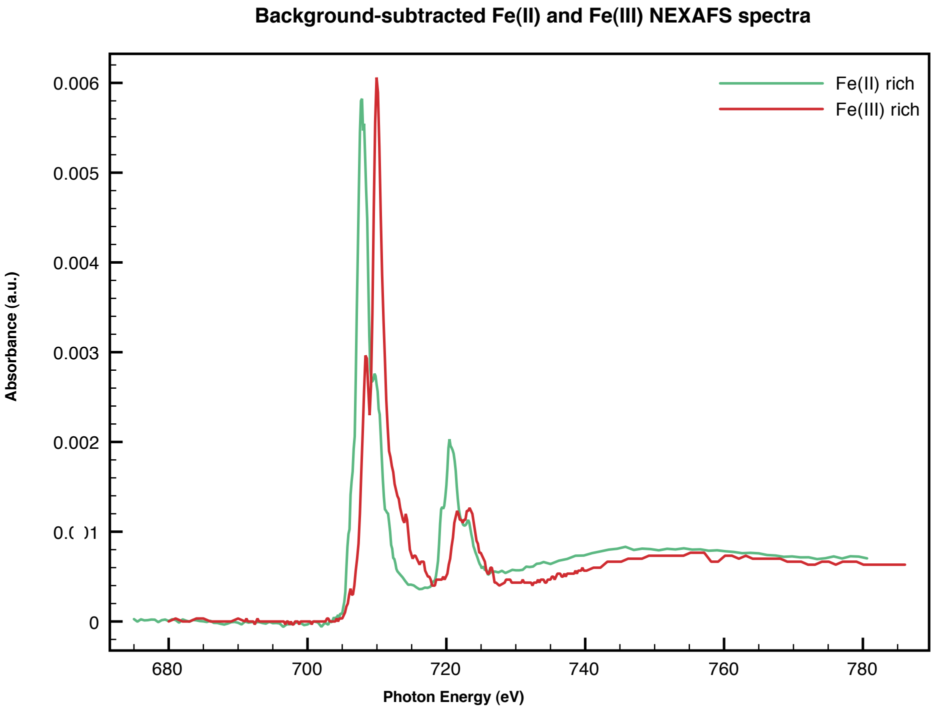

Figure 5.1: Fe(II) rich and Fe(III) rich NEXAFS reference spectra recorded at the Fe $L_{2,3}$-edge

Since the mineral dust particle contains several other elements besides iron, the recorded Fe NEXAFS spectrum is superimposed with background absorption. To perform the SVD analysis, three different reference spectra are provided to the **stackSVD** script. The variables $s1$ and $s2$ contain the Fe(III) and Fe(II) rich reference spectra which have been baseline-subtracted before performing the SVD analysis. Figure 5.1 shows these reference spectra. The background absorption is modeled using a linear background with a negative slope that is stored in the variable *base*.

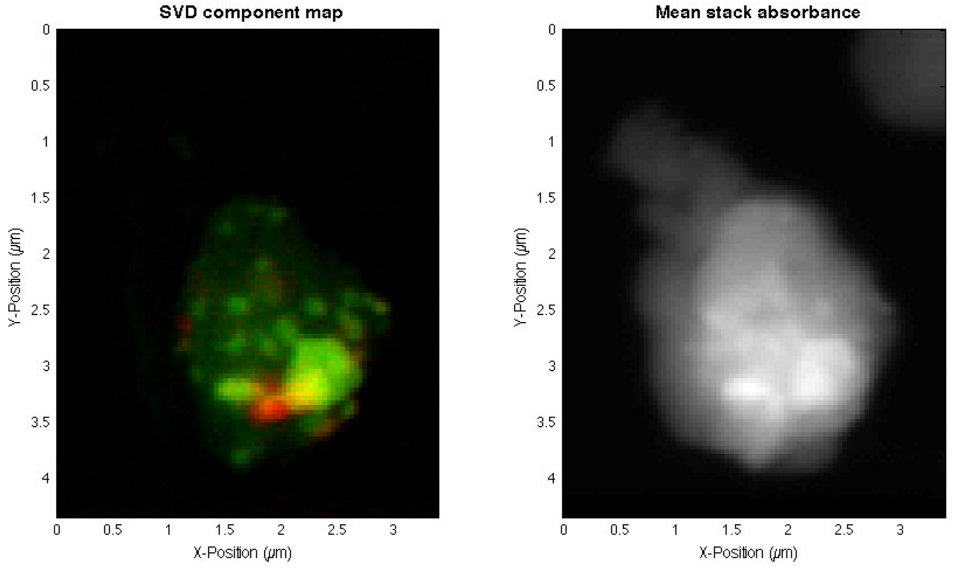

Figure 5.2: Distribution of Fe(II) (red) and Fe(III) (green) in a mineral dust particle

As a first step, the matrix array containing the component maps is generated with the **stackSVD** script:

```
>> Cmaps=stackSVD(S,s1,s2,base);
```

The second step of constructing the RGB color map of the spatial iron oxidation state distribution is to scale the SVD component maps to the maximum value of a RGB image, that is 255. The maximum SVD component value is calculated, the Fe(III) and Fe(II) maps stored in the $Cmap(:,:,1)$ and $Cmap(:,:,2)$ variable are scaled to the RGB image range and copied in the corresponding color channels of the final image variable $RGBMap$:

```
>> if max(max(Cmaps(:,:,1)))>=max(max(Cmaps(:,:,2)))
       scale=max(max(Cmaps(:,:,1)));
   else
       scale=max(max(Cmaps(:,:,2)));
   end

 >> RGBMap(:,:,1) = uint8(255/scale*Cmaps(:,:,1));
    RGBMap(:,:,2) = uint8(255/scale*Cmaps(:,:,2));
    RGBMap(:,:,3)=0;
```

The resulting image is shown together with a grayscale image of the mean of the stack absorption in figure 5.2.

6 STACKLab - GUI tool for STXM data exploration

The STXM Data Analysis Package contains a GUI tool for stack exploration called **STACKLab**. The program is started by calling the function with the desired stack structure array as the only argument:

```
>> STACKLab(S)
```

The graphical user interface of the tool is shown in Figure 6.1. The user interface is divided into two main components: the *StackViewer* at the left side and the *SpecViewer* at the right side.

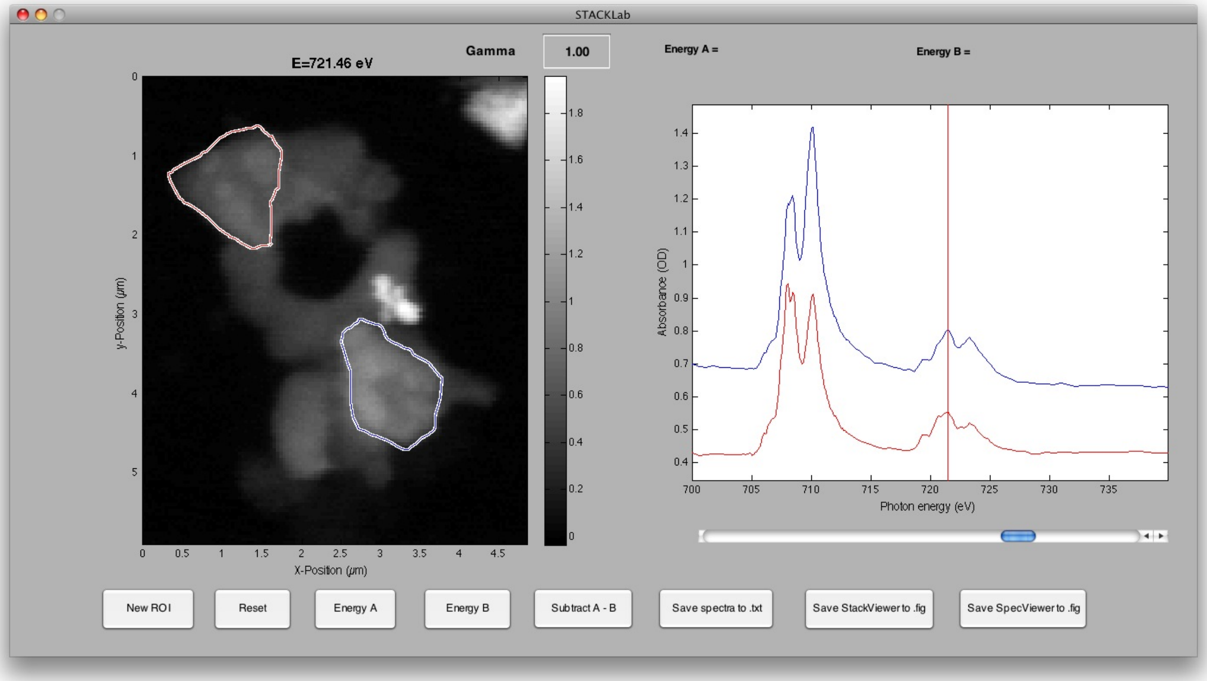

Figure 6.1: Graphical user interface of the stack exploration tool **STACKLab**

The right window, the *SpecViewer* is used to display NEXAFS spectra extracted from the stack. After the program starts, an image analysis algorithm identifies regions with high optical density (e.g. particles on the substrate) by applying Otsu's method on the mean of the stack images and extracts the average spectrum of the "non-background" regions.

The *StackViewer* is used to show the individual absorption images of the stack. The corresponding photon energy can be adjusted using the slider at the lower right or by clicking with the mouse at the desired position of the spectrum in the *SpecViewer* window.

A common way to identify stack areas that are rich in one chemical component is using a *difference map*: Two images at energies A (at a strong absorption peak of the chemical component of interest) and B (usually a pre-edge image) are subtracted, the resulting image is the difference map. Generating difference maps is implemented in **STACKLab**. The energies A and B are set by clicking at the respective buttons. The difference map is shown in the *StackViewer* after clicking the *Subtract A - B* button.

NEXAFS spectra can be extracted from arbitrary stack regions using the *New ROI* button. After clicking the button the mouse can be used to capture the region in the *SpecViewer* window. The corresponding spectrum (i.e. the average spectrum over the region of interest) is displayed on the right side immediately after the regions of interest are selected.

Gamma Correction can be used to adjust the image contrast and to emphasize regions with higher or lower optical density by entering a new γ value in the text box next to the *StackViewer*. The gamma corrected image is generated in a two step process:

1. The original optical density values of the image are mapped to the standard interval $[0, 1]$ by a linear transformation.

2. The gamma correction is achieved by a non-linear intensity transformation

The new intensity values are calculated using the power law:

$$I_{new} = I_{old}^{\gamma} \tag{6.1}$$

From Equation 6.1 it is apparent that choosing $\gamma < 1$ emphasizes regions with lower optical density while $\gamma > 1$ highlites strongly absorbing regions.

The *Save spectra to .txt* button is used to export extracted spectra into ASCII text files. Each file contains one absorption spectrum as a two column array (photon energy in the first column, optical density in the second column). A ”Save spectra as” dialogue asks the user to specify a file path and file name for the spectra. The .txt files are named using the specified filename and a ascending number:

```
STACKLab_Spec_1.txt
STACKLab_Spec_2.txt
...
```

Two buttons *Save SpecViewer to .fig* and *Save StackViewer to .fig* can be used to export the content of the two plot windows for further analysis or presentation. The file path and file name are chosen in a ”Save axes object as” dialogue, the files are stored using MATLAB’s standard figure file format *.fig*. Data can be extracted from these files, axis labels, figure titles and line properties can be changed and different color maps can be applied to the stack images using MATLAB’s plot tools.

Bibliography

[1] D. Attwood. *Soft X-Rays and extreme ultraviolet radiation.* Cambridge University Press, 2007.

[2] I. Koprinarov, A. Hitchcock, C. McCrory, and R. Childs. Quantitative mapping of structured polymeric systems using singular value decomposition analysis of soft x-ray images. *Journal Of Physical Chemistry B*, Jan 2002.

[3] C. Jacobsen, S. Wirick, G. Flynn, and C. Zimba. Soft x-ray spectroscopy from image sequences with sub-100nm spatial resolution. *Journal of Microscopy*, Jan 2000.

[4] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup. Efficient subpixel image registration algorithms. *Opt Lett*, 33(2):156–158, Jan 2008.

[5] R. N. Bracewell. *The Fourier Transform and its Applications.* McGraw-Hill, New York, 2000.

[6] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[7] R. Penrose. On best approximate solution of linear matrix equations. *Proceedings of the Cambridge Philosophical Society*, 52:12–19, 1956.

List of Figures