XXIII ENMC
ENCONTRO NACIONAL DE
MODELAGEM COMPUTACIONAL
XI ECTM
ENCONTRO DE CIÊNCIA E
TECNOLOGIA DE MATERIAIS

A NEW SOFTWARE FOR MA-XRF DATA VISUALIZATION

Sergio A. Barcellos Lins[1] - sergio.lins@roma3.infn.it
Boris Bremmers[1]
Giovanni E. Gigante[1]
[1]Università degli Studi di Roma ”La Sapienza” - Rome, Italy

Abstract. *Macro X-ray fluorescence (MA-XRF) analysis has experienced an exponential increase in usage during the last decade, especially in the field of conservation sciences. This was supported, in parts, by the unceasing development in robotics and electronics, being now possible to develop in-house systems at a much lower cost than commercially available ones. Developing home-made systems entails the need for complex data analysis routines or standalone software to evaluate the large amount of data obtained (sometimes in the order of few gigabytes). The software herein proposed presents itself as an ad hoc alternative for currently existing software, providing a simple and fast way of analyzing multiple datasets and stitch them into a larger set if needed. Moreover, it creates a local user database for easy navigation through the datasets. The software was written in Python and its interface with the Tcl/Tk package. High-performance python routines and parallel processing were implemented to speed up calculation-intensive steps. Automated data extraction routines were written to simplify the evaluation step, dismissing a deep understanding of the underlying physics process.*

Keywords: *X-Ray Fluorescence, Imaging, MA-XRF, Data Visualization*

1. INTRODUCTION

X-ray fluorescence (XRF) is a well known and established analytical technique, widely used in the most diverse fields of science and industry **(Guerra, 2000; Ridolfi, 2019)**. In conservation and heritage sciences its use can be traced back to the late 60's **(Keene Congdon, 1967)**, being now routinely employed. During the last decade, XRF technique started being used in a scanning fashion, gathering data representing large surfaces instead of a handful of spots **(Dik, 2008)**. It first started in synchrotron facilities around 2008 and, thanks to the advancements and miniaturization of robotics and electronics, it is now possible to perform MA-XRF scanning analyses in-situ with mobile or portable home-made instrumentation **(Ravaud, 2016; Cesareo, 2018; Campos, 2019)**.

This transition entailed the need for new data analysis methods and algorithms. The amount of data collected through scanning is significantly larger than that obtained with traditional spot analysis. While for the latter it is possible to evaluate each obtained spectrum manually and individually, for the former this approach is impossible. Few software and routines were developed in time to tackle this issue. However, only few are available as software possessing a distribution package. Currently the default (and only) open-source software used is PyMca **(Sole, 2007)**. It was developed in 2007 and updated ever since. However, it was not conceived initially as an *ad hoc* tool for MA-XRF, and, using it exclusively for this purpose may be a bit confusing or counter-intuitive at first. Another program used for MA-XRF data processing is Datamuncher **(Alfeld, 2015)**, built on top of WinAxil routines and running on IDL virtual machines. It lacks a distribution package and therefore can be difficult to run, especially if the user has no knowledge or practice with virtual machines.

In this scope, a new and *ad hoc* software for MA-XRF data processing was developed (named XISMuS), including different data evaluation routines for a fast, balanced or precise calculation of elemental maps, all automated. The software provides image manipulation tools and a data stitching module to merge different datasets together. The latter being extremely useful when analysing large samples that need to be sub-divided into several parts. Moreover, the software automatically builds a local sample database, updated through usage, providing a fast way to change between datasets to load, compare and cross-normalize outputs.

The software implementation and few results are presented and discussed.

2. IMPLEMENTATION

As a MA-XRF analysis software, the routines required for evaluating the data are mostly based on the extrapolation of firmly established methods for XRF data analysis. These comprise peak deconvolution, energy calibration (through linear or polynomial regression), peak fitting, and continuum estimation, to cite a few. The main difference lies in the automatic application of this series of methods to several spectra at once. This process can be extremely time consuming depending on the dataset size, and, keeping that in mind, three methods were developed to calculate the elements' peaks net area, creating 2D images of the element's spatial distribution. Each method has an application, for a fast (first-approach), balanced or precise analysis. They are labelled in the software as *simple roi*, *auto roi*, and *auto wizard*, respectively.

2.1 Loading and preparing the data

Currently the software supports *.mca and *.txt spectra files. The batch of spectra, representing one sample (dataset), is loaded either by the user or automatically by the software when it detects a dataset in an user-defined subfolder. The dataset is packed into a *.cube file, containing all the necessary information to finally extract the elemental maps. The file-reading process can become a bottleneck for large datasets readout, significantly increasing the computing time. Because of that, the reading process is split into several threads, speeding up the operation. Following the file-reading process, necessary information is calculated. This information comprises derived spectra, calibration, global image, spectra matrix and continuum matrix. The latter can be calculated using the Statistical Non-Linear Iterative Peak Clipping (SNIPBG) or orthogonal polynomial methods **(Van Grieken, 2002)**, chosen by the user.

2.2 Elemental maps extraction

The element's peaks usually follow a Gaussian distribution, deviations from this profile are observed when a high number of photons are detected (roughly 10^7) in the peak, or when fitting K-lines of high Z atoms, such as barium **(Van Grieken, 2002)**. Therefore, in a MA-XRF perspective, one can set a region of interest (ROI) equal to the peak's full-width at half-maximum (FWHM) centred at the peak's centre.

By "slicing" both the spectra and continuum matrices synthetically packed in the *.cube file, in the ROI range, it is possible to obtain a 2D representation of that ROI. Traditionally, this range is set manually by the user, where he/she gives the start and end indexes. This option is available in the software, but, to automate the process, the user can simply select chemical elements from a periodic table and the software will take care of finding those elements' ROI.

In *simple roi* method, each polled element is assigned two energy values, Kα and Kβ (or Lα and Lβ depending on Z). These values are tabulated and the software extracts them from either xraylib **(Brunetti, 2004)** (if available in the host machine) or from its internal database. Next, the energy values distance from the observed corresponding peaks in the summation spectrum is verified. If needed, the peak centre index is updated and the ROI finally set. The peak's FWHM is calculated based on the following relation: $FWHM = 2.3548 \cdot \sigma$, where the standard deviation σ can be defined as a function of the peak energy (in eV), the electronic noise contribution to the peak's width, silicon's fano factor and the energy required to produce an electron-hole pair in silicon **(Sole, 2007; Van Grieken, 2002)**:

$$\sigma(E_{line}) = \sqrt{\left(\frac{NOISE}{2.3548}\right)^2 + (w_{Si} * FANO * E_{line})} \tag{1}$$

In *auto roi* method, the difference is that the peak's centre is verified in each and every spectrum instead of the summation derived spectrum. This approach results in several identified ROI's, one for each pixel-spectrum. With this method it is possible to extract information from elements that are only present in small portions of the sample and consequently suppressed in the summation spectrum. The existence of a peak is assessed with signal-to-noise ratio (SNR) criteria and second differential curve. Each pixel-spectrum is smoothed with a 3^{rd} order Savitzky-Golay filter to enhance peak detection. If a peak is detected, it's net area is calculated over the non-smoothed (raw) data.

Finally, when using *auto wizard* method, the net peak areas are calculated through a Gaussian curve fitting. The software initially runs a peak detection algorithm to identify which elements are present in the dataset. This algorithm will first smooth the spectra matrix and then recalculate the continuum matrix. An auxiliary summation spectrum and corresponding continuum are calculated, where each added spectrum has its values below 1 clipped to 1. In this way, noise from individual pixel-spectrum is overlooked, thus avoiding the mis-identification of peaks. Next, the recalculated summation spectrum is convoluted and its convolution variance, together with its continuum, are used as a criteria to select which peaks will be fitted.

Following, the list of selected and identified peaks is re-checked with *simple roi*'s algorithm for peak centring. The user is prompted to add more elements to the fitting pool (if needed) and then start the operation, calculating the peaks net area by fitting the following equation for each pixel-spectrum:

$$f(peaks) = \left[\sum_{p}^{peaks}\sum_{i}^{nchan}\frac{A_p * gain}{\sqrt{2\pi}\sigma_p}\cdot exp\left(-\frac{(E(i)-E_p)^2}{2{\sigma_p}^2}\right)\right] + continuum \tag{2}$$

Where $E(i)$ is a function that correlates each channel to an energy value, $peaks$ is an array containing the energies of the matched peaks in electronvolts, and A_p, E_p, and σ_p are the amplitude, energy, and standard deviation of peak p, respectively.

Fitting is performed using *scipy*'s curve fitting routine. Fitting independent peaks allows the deconvolution of overlapping ones, not possible with the two previously described methods. Obviously, this large advantage comes with a drawback in computing time. To speed up the operation, fitting is performed in as many available cores possible (when multicore option is enabled by the user), splitting the data into several chunks and launching the processes sequentially. Fano and noise factors are globally fitted during the dataset first compilation, further speeding the fitting process, as less variables have to be fitted in each pixel-spectrum. The processes progress is monitored using a lock and killed if data is left hanging.

2.3 Dataset merge

Datasets can be stitched together using a dedicated module called *Mosaic*. This module lists all *.cube files already compiled by the user and gives the possibility to add any number of datasets to a canvas. The canvas size is set when starting the module. The datasets are represented by either the sum-map or any elemental map previously calculated. In the canvas, the user can rotate and drag the datasets around, positioning them accordingly.

Canvases states can be saved and loaded, a feature useful when analysing data *in-situ*, where datasets are acquired sequentially and the image is assembled during the scanning campaign. This option allows the user to load the previous canvas state with *n* datasets and easily add the newly acquired *n+1* dataset, visualizing the current overall progress of the scanning campaign.

Figure 1 shows a schematic representation of how the *Mosaic* module works. The canvas image is refreshed every time the user performs an action (rotate, drag, move layer). The image is drawn by reading each pixel and checking its priority, given by the layer position. To avoid redrawing the whole image when dragging, only the modified part is refreshed (refreshed area in Fig. 1). Then, when compiling the datasets loaded on canvas into one new *.cube file, the render area will exclude the borders where no data is present.

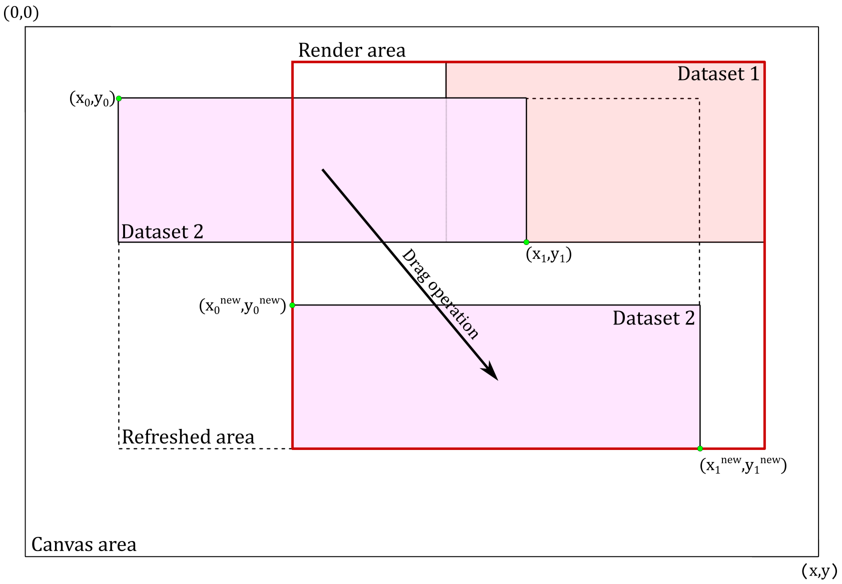

Figure 1- Schematic representation of the Mosaic module functioning.

The rendering process can be performed creating or not a scaling mask, according to user input. In short, the scaling mask is used to correct issues (as discontinuities in the final images) created from the difference in signal intensity from the different datasets loaded on canvas. There are three algorithms available: (a) one that averages the datasets as blocks, by assigning each dataset a scaling factor, (b) one that automatically matches every dataset histogram (based on the sum map) to the largest dataset histogram, and one last method (c) that applies a variant of the contrast stretching filter used in image processing, enhancing the contrast in darker areas. Lastly, the user can manually adjust each dataset histogram if none of the above algorithms suits his/her needs or do not perform well.

3. SOFTWARE OUTPUTS

To demonstrate few of the software capabilities, two artificially corroded copper coupons were scanned with a portable MA-XRF acquisition system developed *in-house*. The system comprises an x-y scanning stage and a detachable head. The latter supports the X-ray tube and detector(s), chosen according to need. The system is built in a modular fashion, allowing a quick exchange of parts, including the scanning stage, if larger samples are to be analysed.

For the data acquisition process, the tube operated with a voltage of $35\,\mathrm{kV}$ and a current of $17\,\mu\mathrm{A}$. The step resolution used was of 1 mm and the dwell-time set to 5 seconds. This resulted in a total scanned area of $43 \times 13\,\mathrm{mm}^2$ acquired in approximately 54 minutes, each sample. These samples were thoroughly analysed and interpreted elsewhere **(Barcellos Lins, 2019)**. Here, they are used with the sole purpose of giving the reader a general overview of the software outputs and capabilities. Furthermore, the system and software have been tested with real archaeological bronze samples **(Barcellos Lins, 2020)**, gilded and painted leathers and other materials.

The datasets were then loaded into the software and compiled each into a *.cube file. Following, they were loaded into a *Mosaic* canvas and merged together by using the counts per pixel scaling algorithm. Elemental distribution maps were extracted with *auto wizard* method using the default configuration values. Elemental maps for sulphur and iron are shown in the analysis screen in Figure 2.

In Figure 3, some other functionalities of the software can be seen. The proposed software can apply a series of routines to the obtained outputs, as image addition and subtraction (Fig. 3a), application of imaging filters (Fig. 2), sub-ROI analysis and correlation plots (Fig. 3b). The latter takes into consideration any filters and sub-ROI applied, therefore being useful to verify possible correlations between specific regions in the image. For the sub-ROI, the derived summation spectrum representing the selected area is displayed in live-time and can be saved separately for further investigation. Image addition/subtraction takes into consideration any filters previously applied.

To test the software outputs reliability, the first sample's datacube (dataset 1 - Fig. 2) was analysed with the default software for MA-XRF analysis. There, the dataset was configured and then fitted with the fast XRF stack fitting tool. Results were nearly identical except for the iron elemental distribution map, shown in Figure 4, where a slightly higher contrast image was obtained with the default software. Sulphur signal is found to be very low, almost suppressed in the summation spectrum. However, both software managed to fit the low-signal peak and extract consistent images.

The information provided by the software could give insight on the sulphur-based corrosion

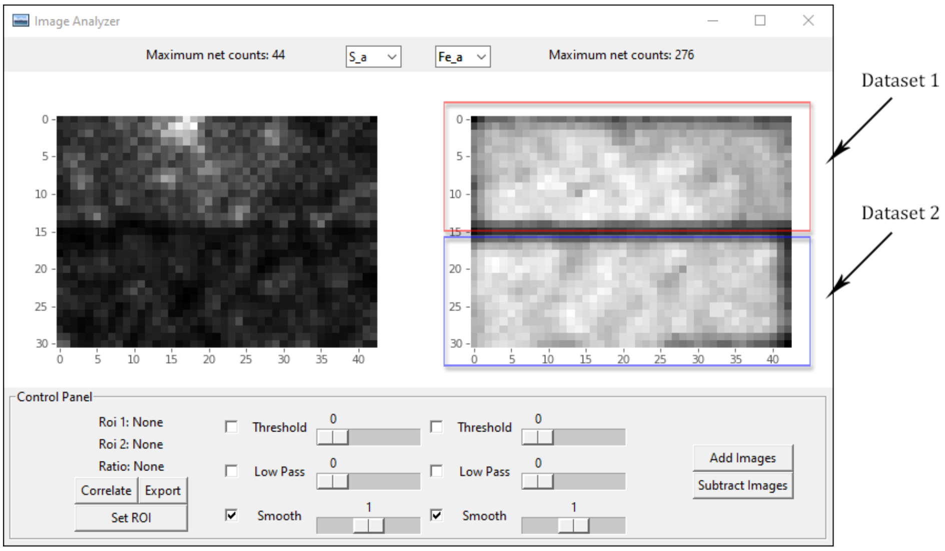

Image Analyzer

Maximum net counts: 44 S_a Fe_a Maximum net counts: 276

Control Panel

Roi 1: None
Roi 2: None
Ratio: None

□ Threshold 0 □ Threshold 0
□ Low Pass 0 □ Low Pass 0
☑ Smooth 1 ☑ Smooth 1

Figure 2- Image analyzer screenshot from the software GUI. Elemental distribution maps obtained from the rendered datacube of sulphur and iron Kα lines are shown (left and right, respectively).

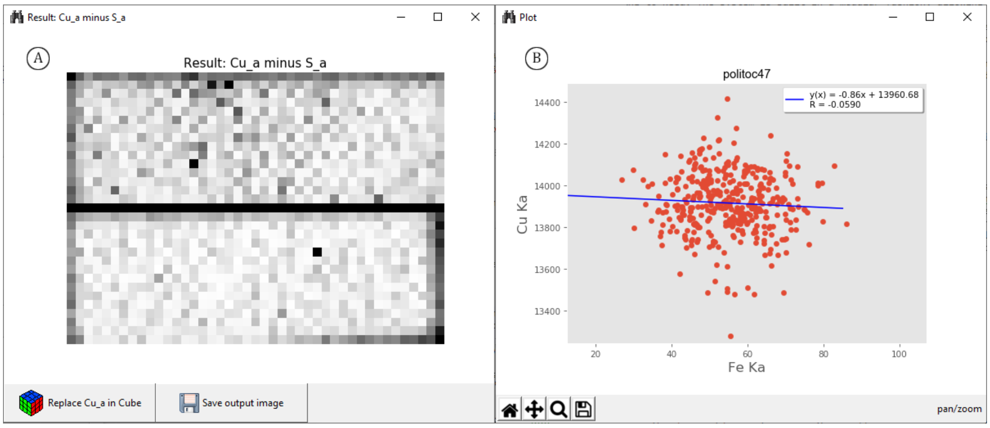

Figure 3- Image subtraction output between Cu-Kα and S-Kα from the rendered datacube (a) and correlation plot example from the central portion of dataset one for Cu-Kα and Fe-Kα (b).

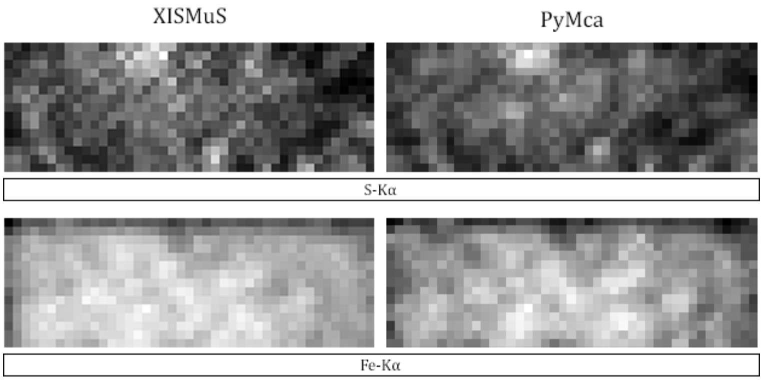

Figure 4- Comparison between PyMca and XISMuS outputs for sulphur and iron Kα lines from dataset one.

products spatial distribution, even though the sulphur concentration in the sample's surface, as a result of the artificial corrosion process is considerably low (± 2% wt.). Datacube merging, by using the *Mosaic* module was useful solely for a more direct comparison and visualization of both datasets together. It's full power is only highlighted when dealing with large samples which have been sub-divided into two or more sub-datasets. Furthermore, the correlation plot shown in Fig. 3b suggests that there is no correlation between iron and copper in this case, being iron rather a contamination. This is somewhat plausible, as the samples are 99.96% pure.

4. CONCLUSIONS

The software herein presented demonstrated to be a comparable alternative to the standard software for MA-XRF data analysis. It provides most of the staple tools required for a throughout interpretation of data in a fast and automatic way. The simple GUI was developed to render the (sometimes overwhelmingly confusing) data analysis process as simple and intuitive as possible. Moreover, most of the software functionalities were presented to the reader, showing how outputs can be obtained, and the overall functioning of the software and its underlying methods were explained.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 766311. The authors would like to thank Prof. Dr. Sabrina Grassini and Dr. Elisabetta Di Francia for kindly providing the datasets used to demonstrate the software functionalities. Furthermore, the authors also acknowledge Prof. Dr. Roberto Cesareo, Prof. Dr. Antonio Brunetti and Dr. Stefano Ridolfi for the fruitful discussions, suggestions and comments during the development of this software.

REFERENCES

Alfeld, M. and Janssens, K. (2015), "Strategies for processing mega-pixel X-ray fluorescence hyperspectral data: A case study on a version of Caravaggio's painting Supper at Emmaus", *Journal of Analytical Atomic Spectrometry*, Vol. 30, No. 3, 777-789.

Barcellos Lins, S. A., Di Francia, E., Grassini, S., Gigante, G. E. and Ridolfi, S. (2019) "MA-XRF measurement for corrosion assessment on bronze artefacts", *Proc. of IMEKO International Conference on Metrology for Archaeology and Cultural Heritage*, Florence, Italy, 4-6 December, 2019, 538-542.

Barcellos Lins, S. A., Ridolfi, S., Gigante, G. E., Cesareo, R., Albini, M., Riccucci, C., Di Carlo, G., Fabbri, A., Branchini, P., and Tortora, L. (2020) "Differential X-Ray Attenuation in MA-XRF Analysis for a Non-invasive Determination of Gilding Thickness", *Frontiers in Chemistry*, Vol. 8, March, 1-8.

Brunetti, A., Sanchez Del Rio, M., Golosio, B., Simionovici, A., and Somogyi, A. (2004) "A library for X-ray-matter interaction cross sections for X-ray fluorescence applications", *Spectrochimica Acta - Part B Atomic Spectroscopy*, Vol. 59, No. 10-11, 1725-1731.

Campos, P. H. O. V., Appoloni, C. R., Rizzutto, M. A., Leite, A. R., Assis, R. F., Santos, H. C., Silva, T. F., Rodrigues, C. L., Tabacniks, M. H. and Added, N. (2019), "A low-cost portable system for elemental mapping by XRF aiming in situ analyses", *Applied Radiation and Isotopes*, Vol. 152, 78-85.

Cesareo, R., Lopes, R. T., Gigante, G. E., Ridolfi, S. and Brunetti, A. (2018), "First results on the use of a EDXRF scanner for 3D imaging of paintings", *Acta Imeko*, Vol. 7, No. 3, 8-12.

Dik, J., Janssens, K., Van Der Snickt, G., Van Der Loeff, L., Rickers, K. and Cotte, M. (2008), "Visualization of a lost painting by vincent van gogh using synchrotron radiation based X-ray fluorescence elemental mapping", *Analytical Chemistry*, Vol. 80, No. 16, 6436-6442.

Guerra, M. F. (2000), "The study of the characterisation and provenance of coins and other metalwork using XRF, PIXE and Activation Analysis", *Radiation in Art and Archeometry*, 378-416.

Keene Congdon, L. O. (1967), "Metallic Analyses of Three Greek Caryatid Mirrors", *American Journal of Archaeology*, Vol. 71, No. 2, 149-153.

Ravaud, E., Pichon, L., Laval, E., Gonzalez, V., Eveno, M. and Calligaro, T. (2016), "Development of a versatile XRF scanner for the elemental imaging of paintworks", *Applied Physics A: Materials Science and Processing*, Vol. 122, No. 1, 1-7.

Ridolfi, S., Laurenzi Tabasso, M., Askari Chaverdi, A. and Callieri, P. (2019), "The Finishing Technique of the Stone Monuments of Persepolis: Further Studies and New Findings Through the Use of Non-Destructive Analytical Techniques", *Archaeometry*, Vol. 61, No. 2, 272-281.

Solé, V. A., Papillon, E., Cotte, M., Walter, P. and Susini, J. (2007), "A multiplatform code for the analysis of energy-dispersive X-ray fluorescence spectra", *Spectrochimica Acta - Part B Atomic Spectroscopy*, Vol. 62, No. 1, 63-68.

Van Grieken, R. E. and Markowicz, A. A. (2002), "Handbook of X-ray spectrometry", Marcel Dekker, Inc., pp.983.