FiB-SEM (BSE) slices segmentation – Guidelines

Scope and summary

The scope of this code is the analysis of FiB-SEM slices to crop/register them (*i.e.*, aligning them), reduce the difference in brightness between the slices (optional), and segment them (*i.e.*, assigning every pixel to a given phase – here cathode active material, solid electrolyte, or pores).

The code uses backscattered SE images (BSE).

The code has been originally developed for the case of solid-state electrodes, but it can be directly translated to the classical Li-ion batteries scenario (by knowing that what is called “solid electrolyte – SE” by the code, in this scenario would rather be the carbon-binder domain (CBD), i.e., the agglomerates of conductive additive and binder).

This code has functionalities that at the moment cannot be included in the executable, therefore it should be run directly the code – but don’t worry, it is really easy and you want a section to explain you how to do it below.

To run the code, you should fill in the associated input Excel (“Inputs_FIB-SEM_ASSB.xlsx”), which offers also a description of each of the parameters you can control, as well as reference/default values. Afterward, you can simply run the code (the Excel and the code should be placed in the same folder). This will lead to a first message (that will close itself automatically after 5 seconds) informing you that the code started, and a second message (which will not close automatically) when the analysis is finished. As a function of the number of slices you want to analyze, and as a function of your computer characteristics, the analysis can take quite some time – up to several hours.

Phases Segmentation

The segmentation is performed in two steps:

1) An Otsu thresholding to distinguish the cathode active material (CAM) from the rest (solid electrolyte + pores);
2) A Gabor filter + manual thresholding to discriminate the pores from the solid electrolyte (SE). The SE phase location is then determined by exclusion.

The procedure above is exemplified in Figure 1.

First, the original image is divided in two (CAM, and SE+pores) through an Otsu thresholding (an approach based on identifying the threshold value, in terms of pixel brightness, between different phases in the image – as shown in Figure 2). Afterward, a Gabor Filter (Figure 3) is applied on the reversed pore+SE image (now the SE and pores are brighter, while the CAM is fully dark – *i.e.*, with an imposed pixel intensity of 0). The Gabor filter is dependent on a number of parameters (Figure 3) and can be indeed considered an "infinite" filter (meaning that with an infinite combination of parameters, you will get an infinite number of different filters), and can make arise certain image features and hide others. This filter is very powerful, but there is no way to know in advance which set of parameters can work for your specific scenario. In my case I got lucky after a few hundred tests I found a set of parameters that allow spotting the pore position while hiding all the SE (Figure 1, "4) Pores – Gabor"). Once having identified also the pores, it is possible to define the SE by exclusion with respect to the already identified location of CAM and pores.

This process is performed iteratively on all the slices (which were previously cropped and aligned), which are finally recombined in the 3D segmented volume.

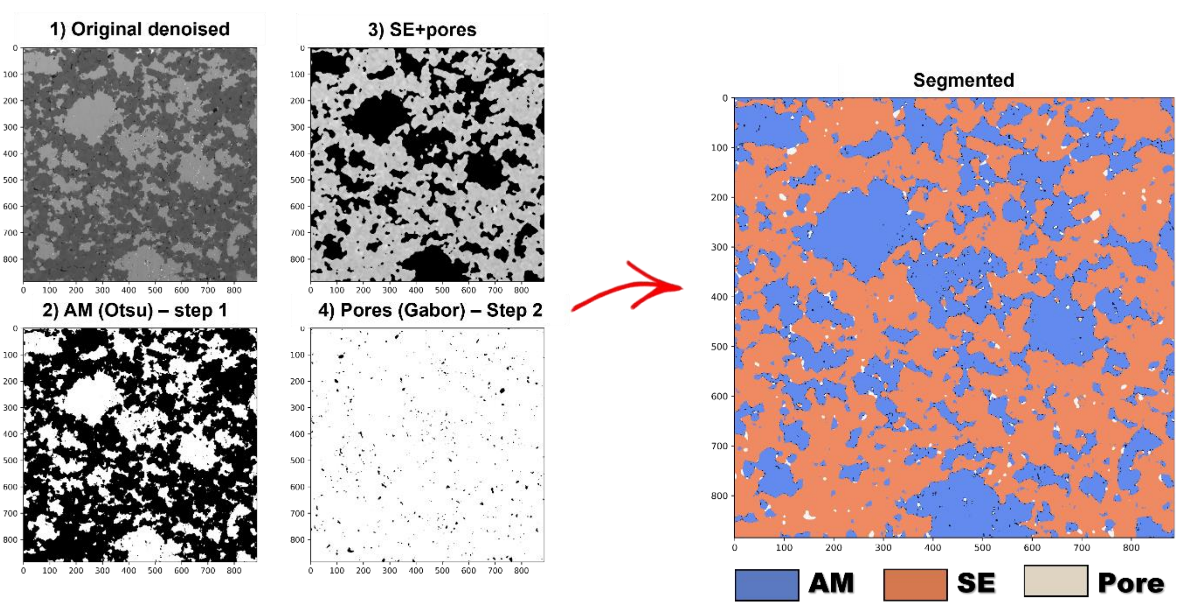


Figure 1. Example of segmentation for a single BSE image.

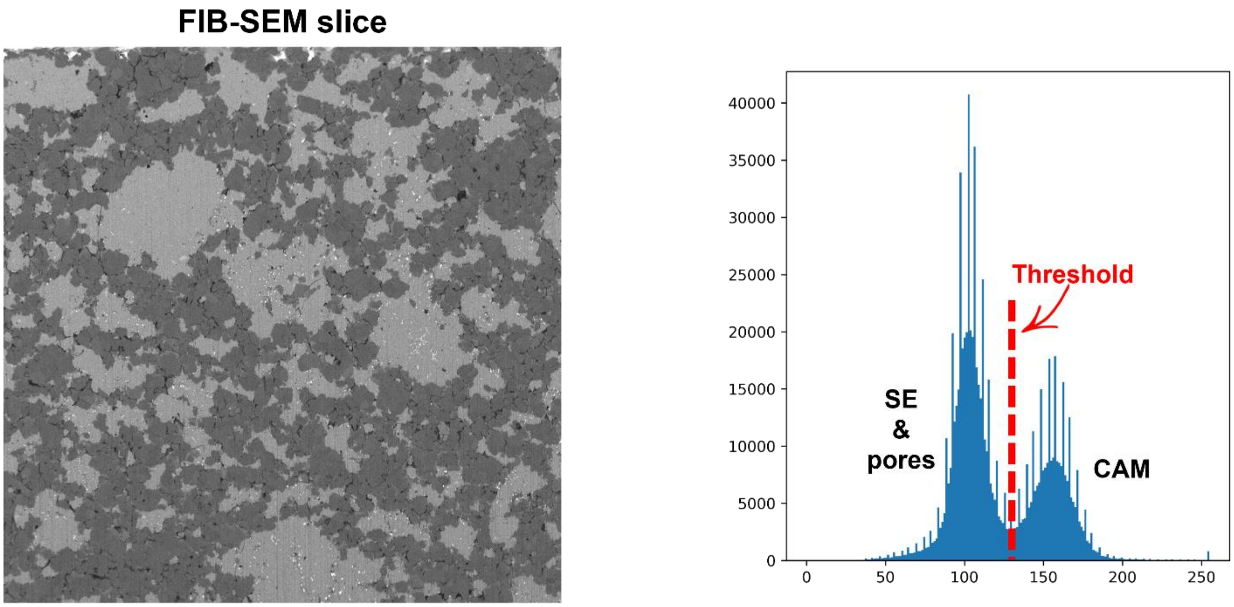
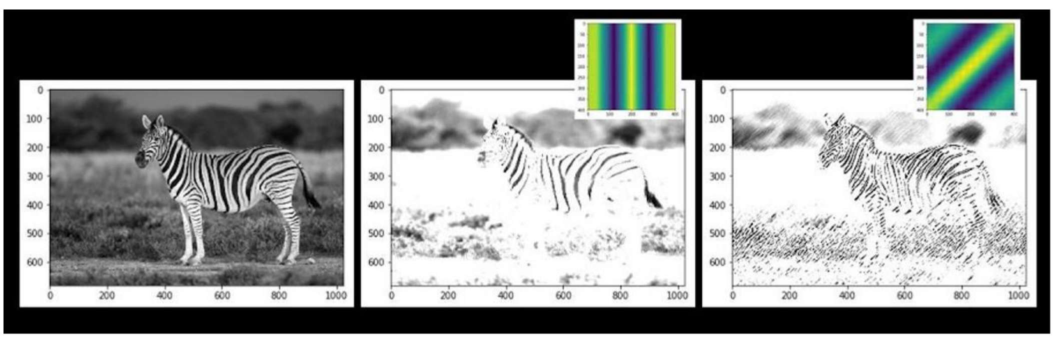


Figure 2. Example of Otsu thresholding. If you want to discover more about this approach, you can check the following YT video: https://youtu.be/YdhhiXDQDl4



$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

Figure 3. Example of Gabor filter, with the equation behind the definition this class of filter on the bottom right of the image. If you want to discover more about this type of digital filter, you can check the following YT video: https://youtu.be/QEz4bG9P3Qs

In case this segmentation approach does not lead to satisfactory results, for your case study, I leave you also a convolutional neural network + Extreme boosting (decision tree-based) code that you can use to train a model for doing the segmentation you need. For understanding how this code work and how to implement/use it, you can refer to this YT video: https://youtu.be/vgdFovAZUzM

Results

The main result is the segmented and aligned 3D volume you sampled through FiB-SEM, which you will get in both 2D (png) and 3D (tif) format. An example of a 3D segmented structure is reported in Figure 4.

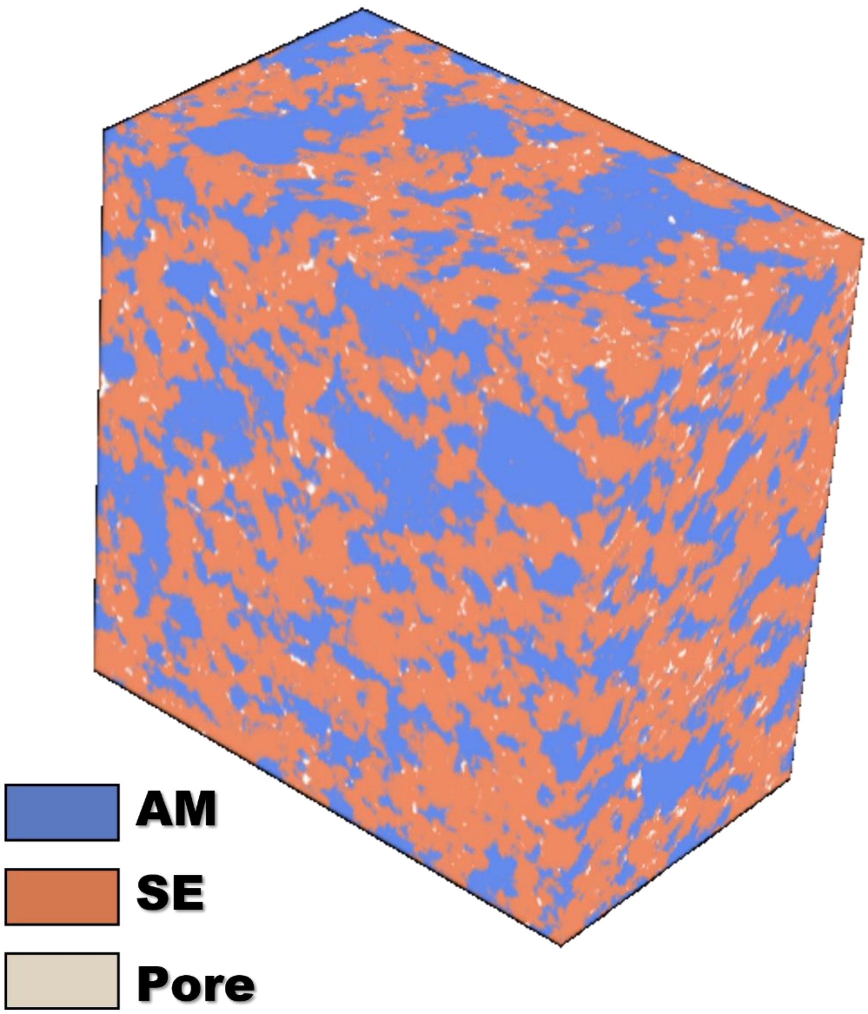


Figure 4. Example of 3D segmented volume. Orange = CAM ; blue = SE ; white = pores.

Running the code

This code has functionalities that at the moment cannot be included in the executable, therefore it should be run directly the code – but don't worry, it is really easy.

I advise you to use the Jupyter Notebook version of the code, as this will allow you to re-launch (if needed) only some part of the code and not all of it (saving time) and because the Jupyter version of the code offers some interactive plotting to check the goodness of each processing step (cropping and alignment, light enhancement, segmentation).

The easiest path for using Jupyter Notebook is probably installing Anaconda (containing Jupyter Notebook, among other things):

https://www.anaconda.com/products/distribution

After that, you can launch the Anaconda Navigator (just search it after the installation) and from there you can run Jupyter Notebook.

The libraries to be installed are the ones reported in the very top of the code ("import Xxx" or "from Xxx"). It is very easy to install them (you can also google, library by library, how to install them, but typically it works for all in the same way):

https://youtu.be/Yr_ihLKq_yY

If you want to understand a bit better and learn more about Python (that can really be handy, so I definitely advise it!), you can look at the following videos series:

https://youtu.be/7uE6hypji0o

Contact for problems/doubts

If you have any problem using the code, feel free to contact me on my personal email: teo.lombardo3@gmail.com