

Real-time Diagnostic Tool for the Scanning Electron Microscope Technical Milestone Report

Liuchuyao Xu, Robinson College

Supervisor: Dr David Holburn

January 8, 2020

Abstract

RESERVED LINE.
RESERVED LINE.
RESERVED LINE.
RESERVED LINE.
RESERVED LINE.

1 Introduction

The Scanning Electron Microscope (SEM) is an important technology for micro-surface observation. Conventional optical microscopes have resolutions limited to the wavelength of light (~ 200 nm), whereas SEMs offer greater resolutions (~ 0.5 nm) [1].

SEMs work in a similar manner as optical microscopes. An electron beam is first generated and then shone onto the specimen. The reflected electrons are captured and used to produce an image. The resolution depend on the wavelength of the electrons, which in turn depends on the voltage used for accelerating the electrons.

The images produced by SEMs often have a variety of disturbances, such as lack of sharpness, low image quality, noises and image distortion and deformation. Sources of the disturbances could be interaction between the specimen and the electron beam, selection of observation conditions, specimen preparation or problems with the instrument itself.

Because of the disturbances, it is often not straightforward for an SEM operator to judge whether an image is good. The aim of the project is to develop a GUI-based software, which aids operators by providing real-time diagnostic information of the images. The initial requirement for the software is that it must display results of Fast Fourier Transform (FFT) on the images at a minimum of 10 frames per second. An SEM operator will constantly review the software, give feedback and come up with new features to be added. So far, the additional features added are displaying the histograms of the images and the results of histogram equalization on the images.

2 Progress

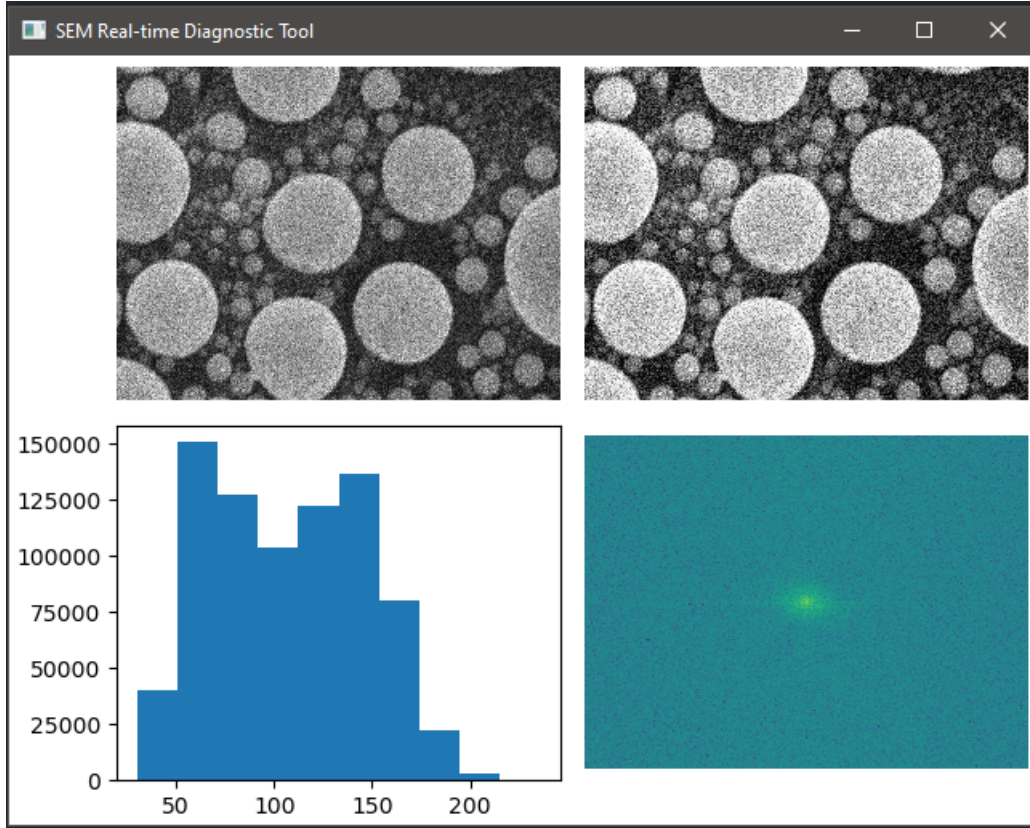


Figure 1: Preliminary GUI

Figure 1 shows the preliminary GUI. The top left plot shows the image grabbed from the SEM and the bottom left plot shows the histogram of it. The top right plot shows the result of histogram equalization on the image and the bottom right plot shows the result of FFT on the image.

The software is developed using Python and Qt5, a GUI framework. Both of them are widely used in the industry and well-supported. This ensures that the dependencies of the software will not easily go out-of-date, and it is straightforward for others to continue developing the software in future.

The SEM images are grabbed by calling an Application Programming Interface (API), provided by Carl Zeiss, the manufacturer of the SEM being used. The API converts the images into Python *numpy* arrays, which are then plotted using Python *matplotlib* and displayed using Qt. Both of the Python packages are highly optimized and the whole process takes less than 1 millisecond to complete on the test PC.

The histograms are calculated from the Python *numpy* arrays, and plotted in a similar manner as the SEM images. This process does not take a long time either.

Histogram equalization is the latest feature added to the software, and is used for improving the contrast and brightness of the images. The algorithm is adapted from torywalker's work on Github [2]. The algorithm takes 2 seconds to complete on the test PC, which prevents the software from being real-time. Methods for improving the performance should be investigated.

Performing FFT is the main feature of the software. Originally it was done using the Python package *numpy*, and it took about 0.3 seconds to update each frame on the test PC. Various methods for improving the performance were investigated and the final solution is to utilise the Nvidia CUDA Toolkit, which can take advantage of the parallel processing capability of the GPU to perform the FFT. On the test PC, a speed boost of about 5 times was achieved, which mean it takes only about 0.06 seconds to update each frame. This makes it possible to meet the requirement of 10 frames per second.

3 Next Steps

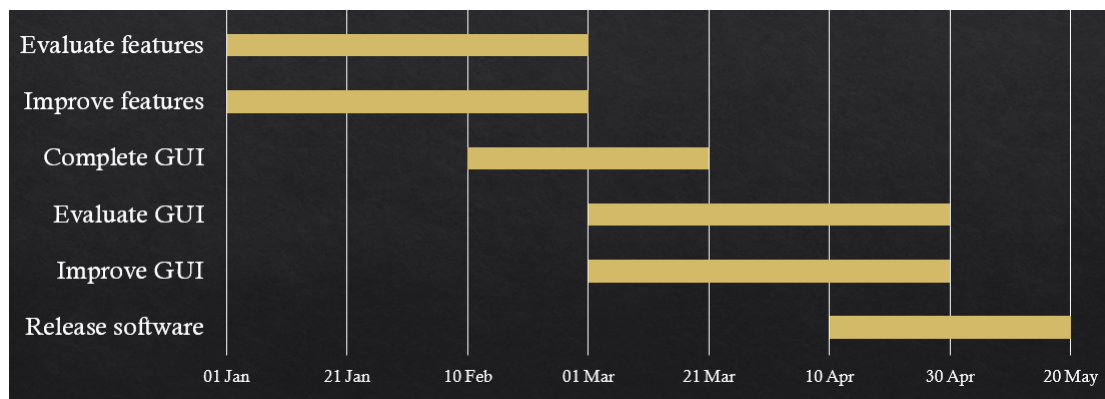


Figure 2: Next Steps

Figure 2 shows the time-line of the rest of the project. The most important step now is to evaluate the features with an SEM operator, to assess whether the features are useful and determine how they can be improved. Then time should be spent on improving the aesthetic aspects of the GUI and the user experience.

References

- [1] JEOL. *Scanning Electron Microscope*. 2019. URL: <https://www.jeol.co.jp/en/science/sem.html>.
- [2] torywalker. *histogram-equalizer*. 2019. URL: <https://github.com/torywalker/histogram-equalizer>.