doi:10.1017/S1431927622001416
Microsc. Microanal. 28 (Suppl 1), 2022

Extensible Real-Time Data Processing with Python in DigitalMicrograph

Benjamin K. Miller[1*], Bernhard Schaffer[1], Winnie Lei[1], Cory Czarnik[1]

[1.] Gatan, Inc. Pleasanton, CA, United States.
*Corresponding author Benjamin.Miller@ametek.com

Modern electron microscopy can generate large amounts of data. While this may reveal details of the sample in unprecedented detail, collecting more data can also obscure what's most important. In-situ experiments where data is acquired continuously further increase the data volume. In-situ experiments may also require the user to make decisions about what to do next based on the changes observed in the microscope. However, depending on the sample and the software setup, these changes are not always obvious from a default live view of the data displayed in real-time. Given the large data volume, enhanced flexibility to process and visualize data in a variety of ways is valuable, especially if this can be done in real-time to understand the changes occurring to the sample. This enables informed decision making during the experiment, increasing the chances that valuable data is collected.

For many years Gatan has supported live data processing via the DM scripting language [1] and the "live view" window which makes real data, with full resolution and bit depth, available to the user (sometimes at a lower framerate than the camera writes to disk). Recently, Gatan has introduced Python integration in DigitalMicrograph (DM), so that py files can be edited and run directly from within DM, giving easy access to the data in the live view window as well as most other objects in DigitalMicrograph [2]. This should make this kind of live processing more accessible to researchers since many more people are familiar with writing or using Python, now the most popular language according to the TIOBE index [3].

In this work, we demonstrate live processing of the live view data in DM using several standard Python packages including numpy, scipy, scikit-image, and matplotlib. These examples cover a number of applications including particle tracking, phase change monitoring, and crystallite mapping. They also demonstrate output to data of several dimensions. Examples show how an input image frame can be processed to produce and display a single scalar value, a 1D profile, or a 2D image. In each of these scripts, data is taken from a user specified region of interest (ROI) in the live view, producing a numpy array. Processing is then applied to this numpy array, and the result displayed in a new image window in DM. When the live view updates, the new data is captured, processed, and the result is updated. Multiple processing scripts can be run simultaneously if desired, as demonstrated in Figure 1. If processing takes a long time, then the live view slows accordingly (to one update every 1.5 s in Figure 2). Since these scripts are run on the Gatan PC during acquisition, there are some limits on their speed. These will be discussed within the context of an in-situ experiment, and performance enhancing workarounds presented.

These example scripts will be made freely available on the Gatan website. Since the source code of the scripts is available, it is easy to modify the processing functions to act on the numpy array as desired, leaving the rest of the mechanics of the script that interact with DigitalMicrograph unchanged. While it is hoped that these few examples may be useful in their present form, the extensibility of the scripts is a key feature. The primary goal for these scripts is that they can be used as a starting point for scripts tailored specifically to the needs of individual researchers.

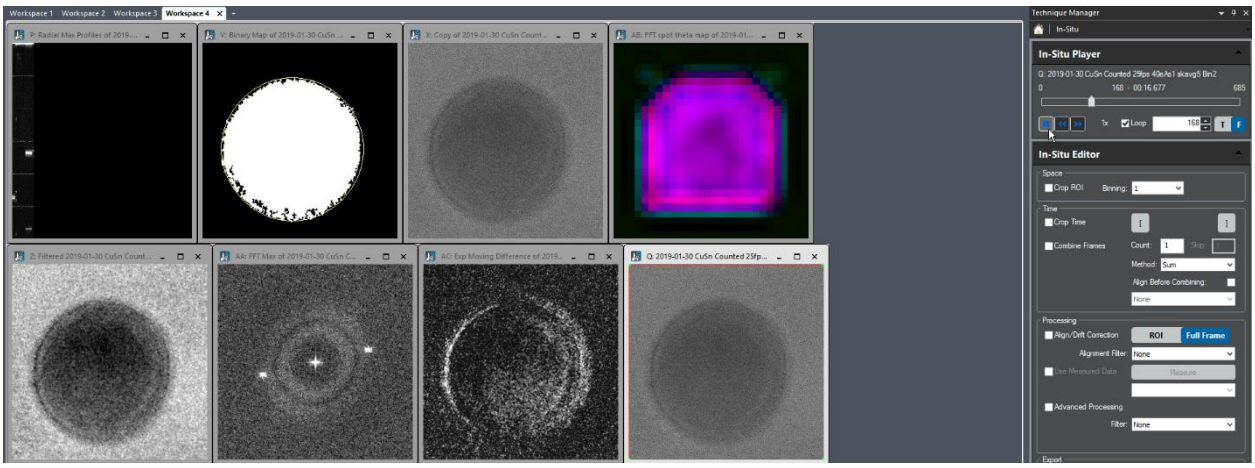

Figure 1. Screenshot from DigitalMicrograph showing the original data (bottom right window) as well as the simultaneous output from 7 different live processing scripts. In this example, with an image size of just 870x870, all processing occurred in less than 1 second per frame.

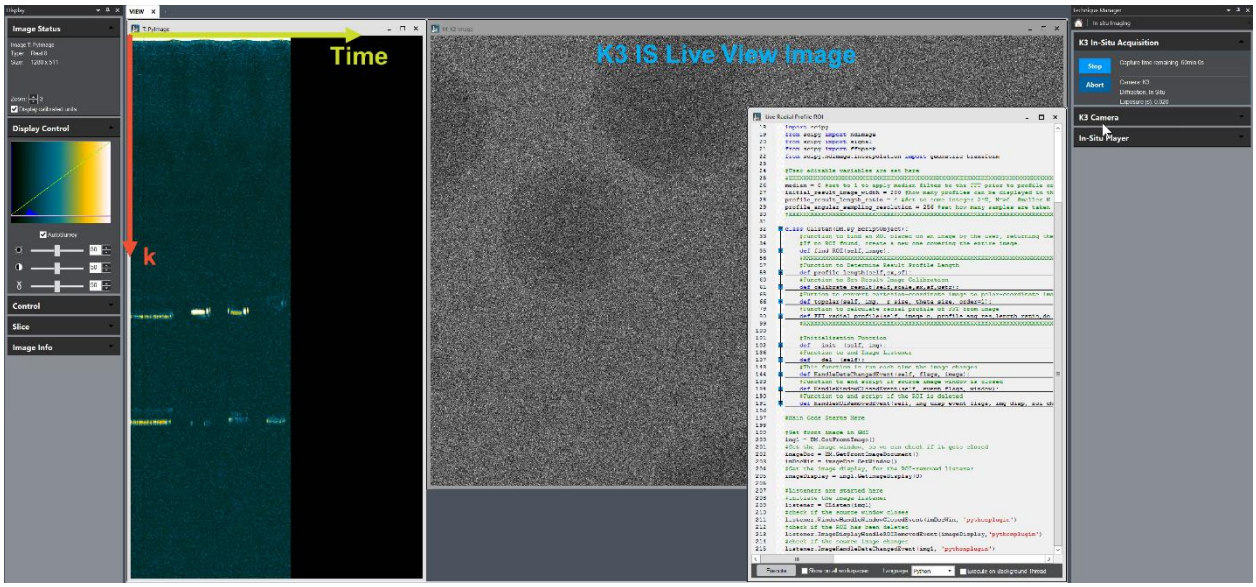

Figure 2. Screenshot from DigitalMicrograph showing real-time processing of in-situ video data as it is being acquired. Radial profiles of diffractograms computed from each high-resolution image frame are placed continuously into a result window on the left. Each profile took about 1.5 s to produce, but data is still saved to disk at 50 fps.

References:
[1] B Schaffer, DigitalMicrograph in “Transmission Electron Microscopy Diffraction, Imaging, and Spectrometry”, eds. B Carter, DB Williams, (Springer International Publishing), p. 167.
[2] https://www.gatan.com/resources/python-scripts (Accessed Feb 2022).
[3] https://www.tiobe.com/tiobe-index (Accessed Feb 2022).

https://doi.org/10.1017/S1431927622001416 Published online by Cambridge University Press