Using Python to Balance User-friendliness and Flexibility for FIB-SEM Data Processing

Mike Marsh, Nicolas Piche, Eric Fournier, and Normand Mongeau
Object Research Systems (Montreal, Canada)

Quantitative imaging by FIB-SEM serial section microscopy continues to mature as equipment and expertise grow in the user community. In general, the required data processing tasks are well handled by software platforms that can align the image stacks and address the downstream processing and interpretation needs. Despite sophisticated user interfaces that anticipate many routine use cases, users working on novel research questions often find themselves with processing challenges that don’t fit the standard work flow. Inevitably, programmatically-minded investigators turn to programmable solutions which provide the needed flexibility but at the expense of user interface ease and elegance. Here we address the difficulty in balancing user-friendliness with flexibility.

We submit that the proper balance of flexibility and ease-of-use is met by developing the data processing platform such that the entire user-interface is implemented in a high-level scripting language. With such a platform, the data processing and interpretation functionality is properly exposed in a scriptable interface. This gives users the needed flexibility to program solutions that don’t meet the standard workflow, but additionally empowers users to process aggregate datasets by scripting the processing solution in batch.

We present such a commercial implementation where all of the visualization and data processing routines are implemented as high-performance, compiled solutions, but the user interface is scripted in the popular Python programming language. Developing a solution where all of the user-interface is implemented in Python necessitated that all of the end-user required features could be invoked in Python. This, in turn, made it easy to expose all of that functionality to users for scripting purposes. To minimize end-user development efforts, we take advantage of the Numeric Python data structures for all user image data, which lend themselves to many community image processing solutions which are easy to reuse. To meet the user’s flexibility requirements, we provide multiple Python entry-points, different ways of deploying Python code to accomplish tasks. The best example is a fully integrated, interactive Python console where the user can access all relevant data structures, invoke commands, and even import additional libraries. We also have multiple plugin models that permit users to develop customized re-usable scripts and graphical user interfaces, as well as drop in algorithm alternatives for standard image filters and image alignment routines. Collectively, these entry-points enhance a mature user-interface for standard workflows with the required flexibility to customize and extend user solutions with ease.