User interfaces at the PolLux STXM

Benjamin Watts, Simone Finizio and Jörg Raabe

Paul Scherrer Institute, Swiss Light Source, PSI-Villigen, Switzerland
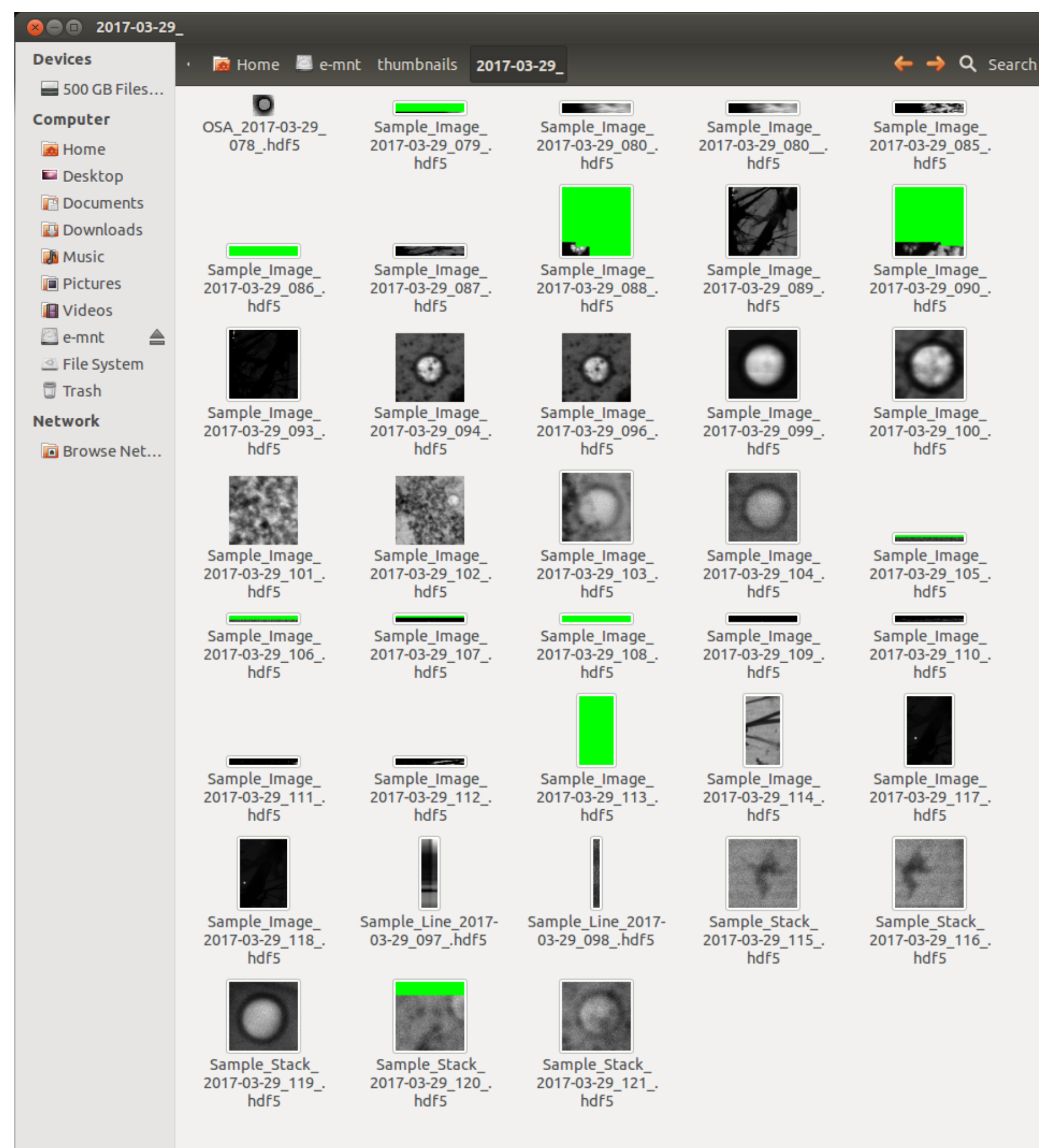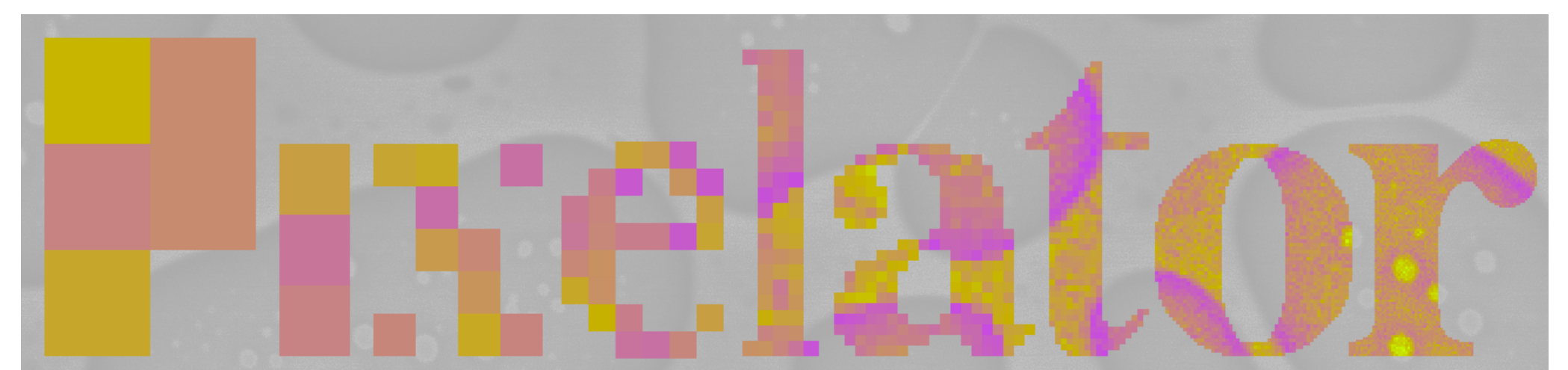
Introduction

The PolLux scanning transmission soft X-ray microscopy (STXM) beamline has been operating at the Swiss Light Source (SLS) since 2006. With a photon energy range of 270 to 1600 eV (from the C K-edge up to the Al K-edge), experiments at PolLux have tended to focus on organic electronic materials, magnetism of the transition metals (Fe, Co and Ni) and studies of the environment. PolLux has seen many upgrades over the years and we continue to develop both the hardware and software to enable cutting-edge science by our user-groups. Avoiding and removing carbon contamination from the beamline optics and endstation are a priority at PolLux in order to preserve flux at the C K-edge (280 - 320 eV).

Instrument Control

PolLux is controlled by the *Pixelator* software, which has been developed in collaboration with the *Max-Planck-Institut für Intelligente Systeme* and *Semafor Informatik & Energie AG* for reliable, efficient and user-friendly operation of experiments. Many hardware interface modules have been implemented for common beamline control systems (EPICS and TANGO), motor controllers (Newport, PI and Smaract) and I/O cards (Agilent and NI). Implementations of *Pixelator* are also running on the HERMES (Soleil) and MAXYMUS (BESSY II) STXM beamlines, and we invite other STXMs to join the Pixelator community. Recent updates to *Pixelator* have focused on efficiency and reliability, as well as the development of a Python module for scripting experiments. The Python module interacts with the *Pixelator* server via the same JSON/zero message queue interface as used by the *Pixelator* GUI and so has access to all functions. In order to tightly integrate complex user-experiments into the STXM measurements, *Pixelator* can be easily configured to record user-defined data sources on a pixel-by-pixel basis via analog or counter inputs, as well as digital channels such as EPICS.

https://www.psi.ch/en/microspec/pixelator

Client
- Qt based GUI
- Define paramters for standa
  scan types
- Data presentatio

Client
- Scripting
- Complex scans
- Batch scanning

JSON

Server

Zero Message Queue

Orchestra Control Engine
- Realtime system
- Positioning feedback
- Fast data sampling

StxmController C++
- Hardware interfaces
- All scan logic
- Data handling

NeXus API (hdf5 data files)

Hardware Modules

PCI Bus

USB

TCP/IP

EPICS

Interferometer

Fast Piezo Scanner

NeXus

HDF

XMP

Data Files

All data are stored in NeXus-compliant HDF5 files, including the interferometer positions and ring current values on a pixel-by-pixel basis, allowing accurate normalization and re-gridding of images. Thumbnail images are included in the HDF5 user-block (following the XMP standard commonly known for including metadata in graphics files) and can be viewed with standard file browsers (via plugins) for the major operating systems as a convenience for users sorting through large numbers of data files.

https://github.com/nexusformat/hdf5xmp/releases

Helper Programs

Synchrotron beamlines and instruments typically include many adjustable parameters that strongly affect the various qualities of the experiments conducted. These parameters often interact in complicated ways that make it difficult to properly educate visiting users about how they can choose the instrumental parameters to optimise the instrument performance for the specific requirements of their experiment. Providing GUI tools that simplify the choice of paramter values while giving understandable feedback on their physical meaning can greatly reduce the support workload on the beamline scientists
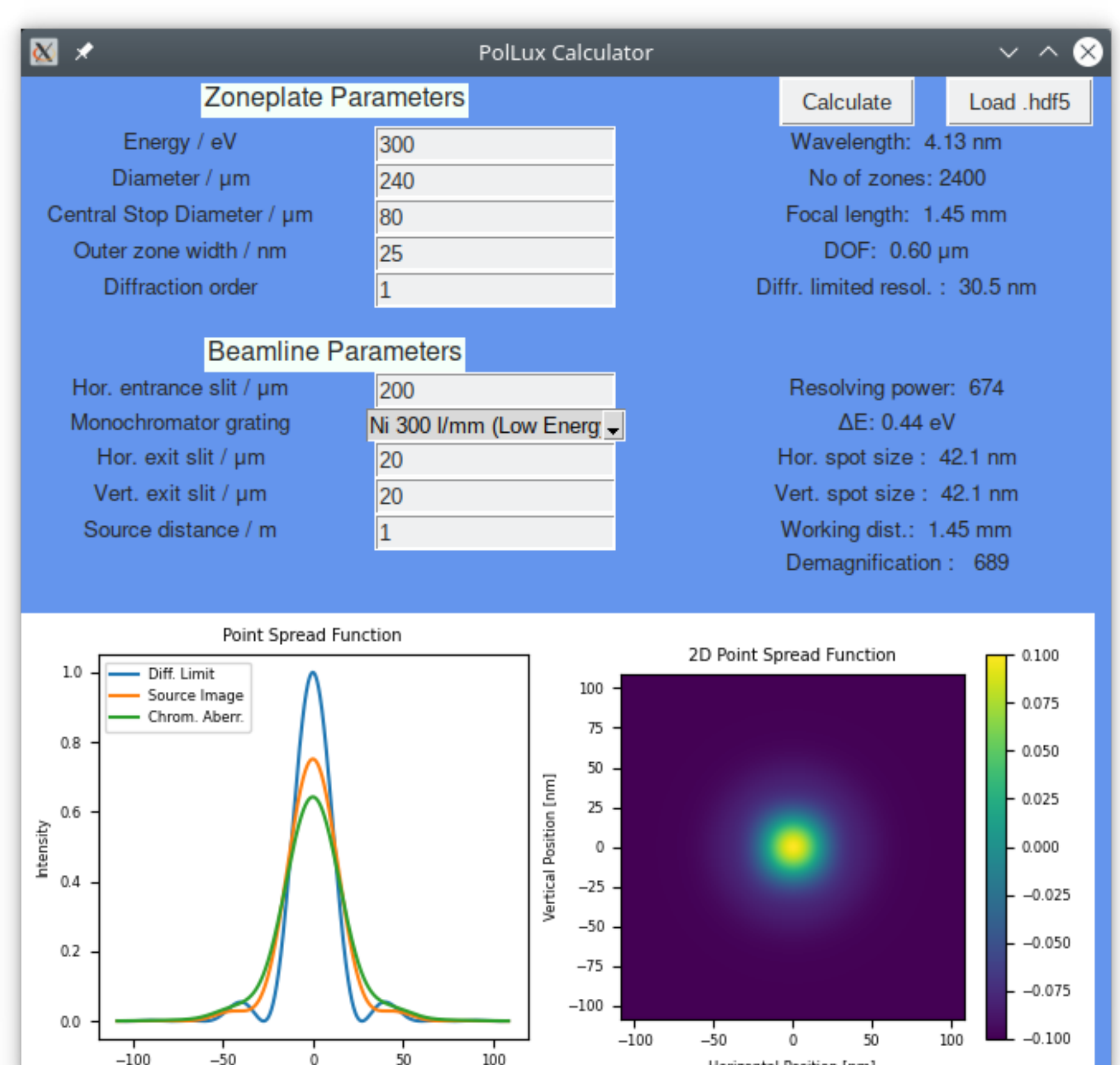
Focus Finder
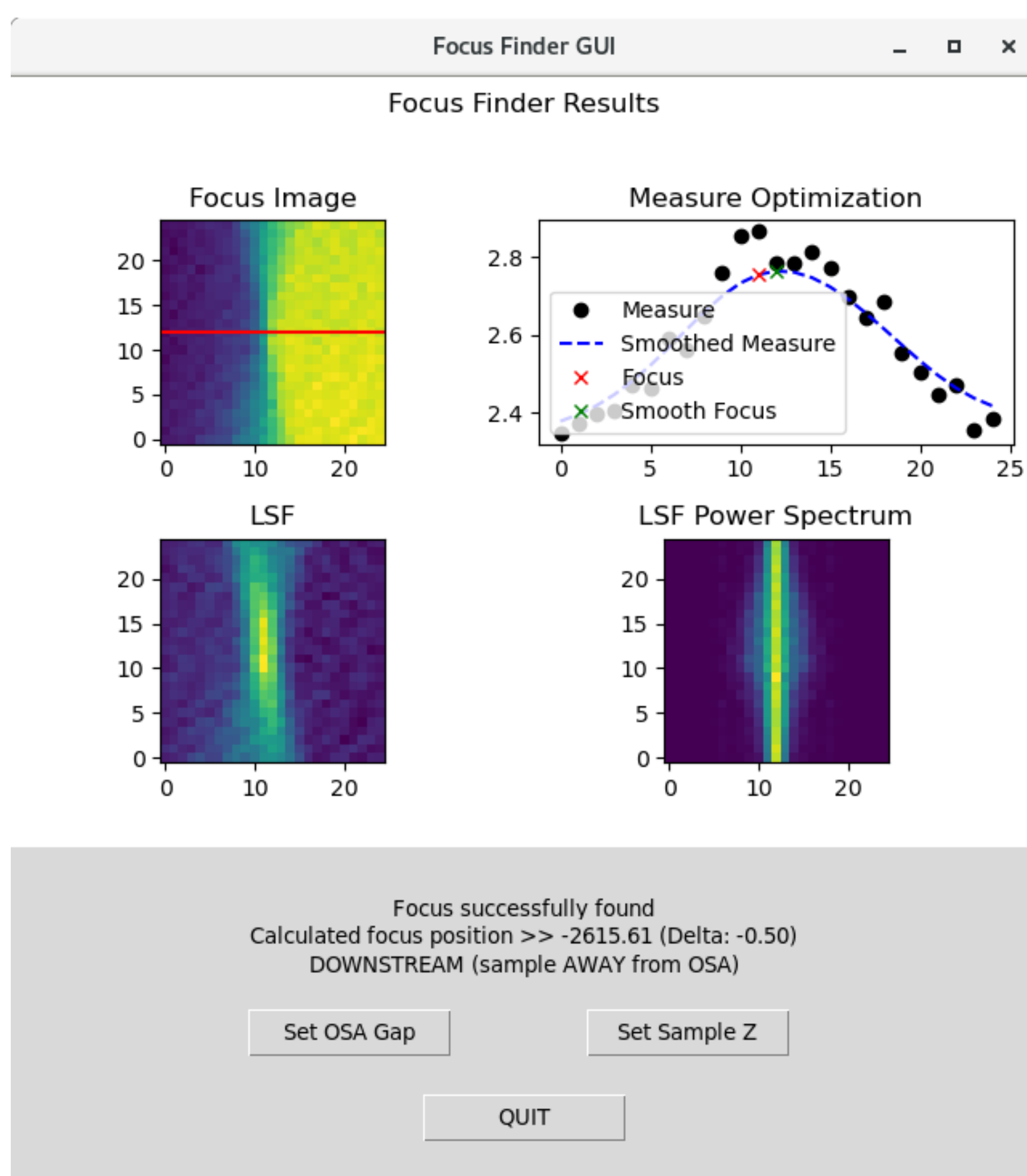
The PolLux Calculator program provides a simple graphical interface to scripts that calculate the critical performance metrics of the scanning transmission X-ray microscopy (STXM) experiments being performed at the PolLux Beamline (Swiss Light Source). The interactive nature of the program allows users to explore the impact of adjusting parameters, such as the monochromator slit widths, before making hardware adjustments. The program is written in Python as a single file so that it is easy to download and run.

https://gitlab.psi.ch/microspectro-public/pollux-calculator

One task that PolLux users often struggle with is the interpretation of focusing scans (performed as a "knife-edge scan"). To address this problem, we provide a *Focus Finder* program that analyses each focusing scan by calculating the entropy of the line spread functions frequency distribution - in other words, the best focus should correspond with the sharpest contrast changes. The GUI further advises on the noise level and provides simple buttons for accepting the new focus position.

PolLux Calculator

Acknowledgements