History and Future of Computerized Data Acquisition: Application to Scanning Microscopy

D. Frank Ogletree, Ed S. Barnard

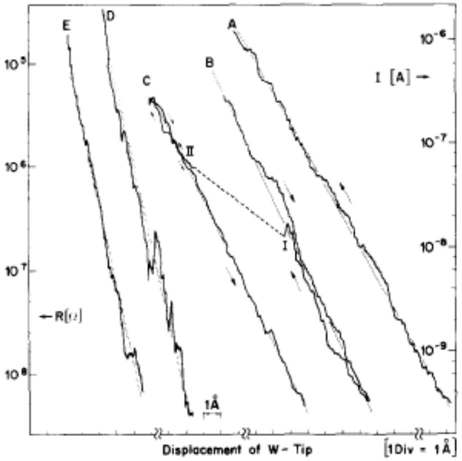
Imaging Facility
Molecular Foundry, Materials Sciences Division
Lawrence Berkeley National Lab

A Short History of Computerized Experiments

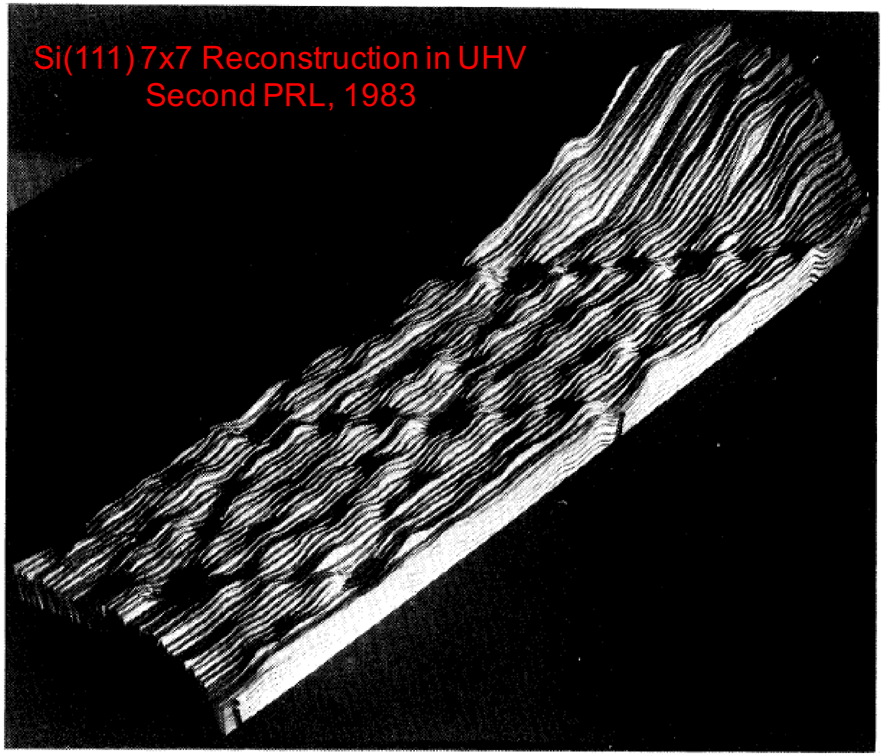
- relatively "recent"
  - only 30 years…
  - STM developed at IBM research was …analog…in early 80's
- mid-80's
  - Artisanal or proprietary, limited hardware, almost no software tools, graphics/visualization, STM/AFM first computerized
- mid-90's
  - crude SEM software, first TEM software without detector integration, CCD detectors for TEM and Spectroscopy…
- mid-2000's
  - internet, much better computers, operating systems, software environments, computer "literacy"
- mid-2010s,
  - high performance computing, fast networks, cheap storage, big data, theory/simulation much faster and more capable…

Invention of the STM, 1981

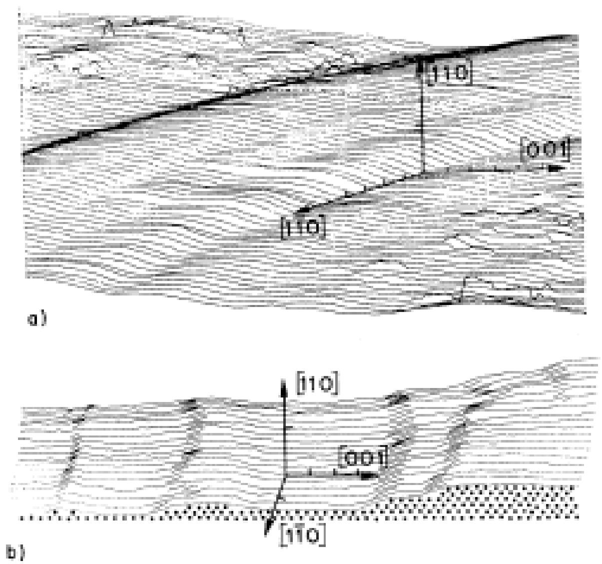
Gerd Binnig & Heine Rohrer, IBM Rüshlikon



Vacuum tunneling between W tip and Pt foil, First APL, Binnig & Rohrer Jan 1982 (results from March 81)

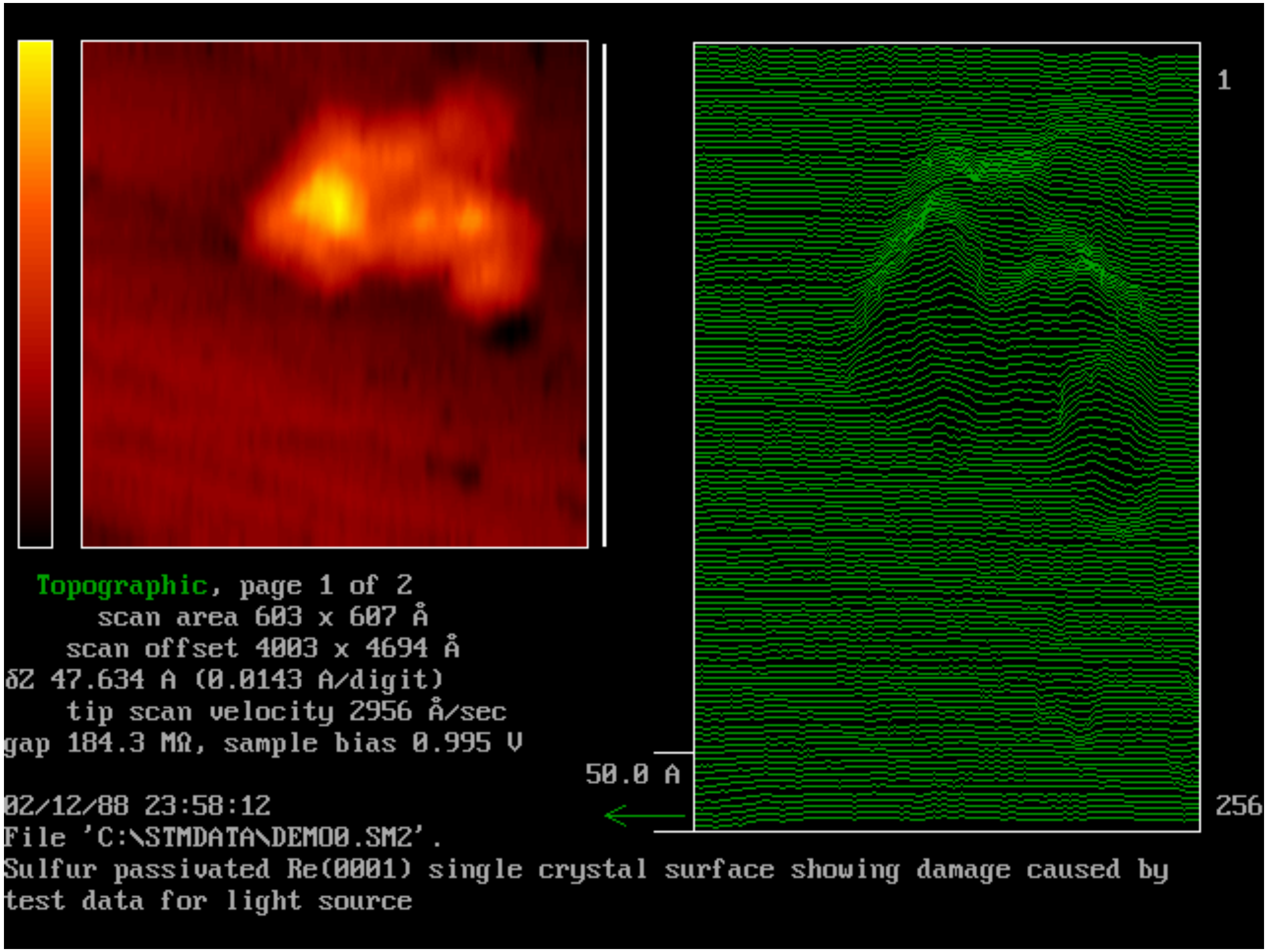


Si(111) 7x7 Reconstruction in UHV
Second PRL, 1983



Atomic Steps on Au(110) in UHV
First PRL, July 1982

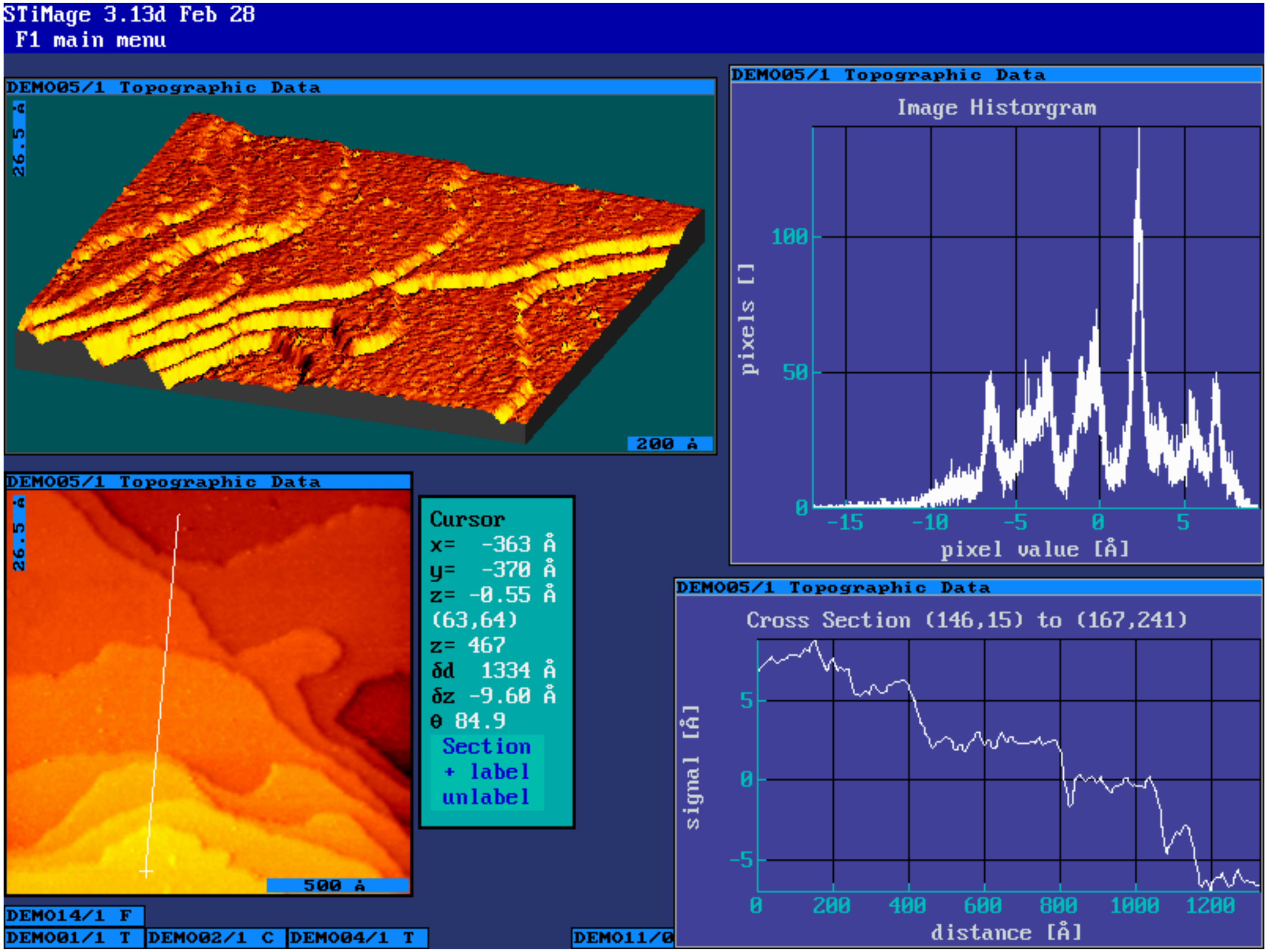
STM Software



Screen capture of first STM program developed at LBL in 1987

Fortran on DEC LSI-11 minicomputer,
5 MB disk, 64 kB RAM, $6,000 display system, 640x480 pixels

STM Software



STM program in 1993, C and Assembler on
Compaq 80386 ($19 k), 0.02 GHZ 1 MB RAM
32 bit CPU, SVGA display, extended DOS

Scanning Microscopy

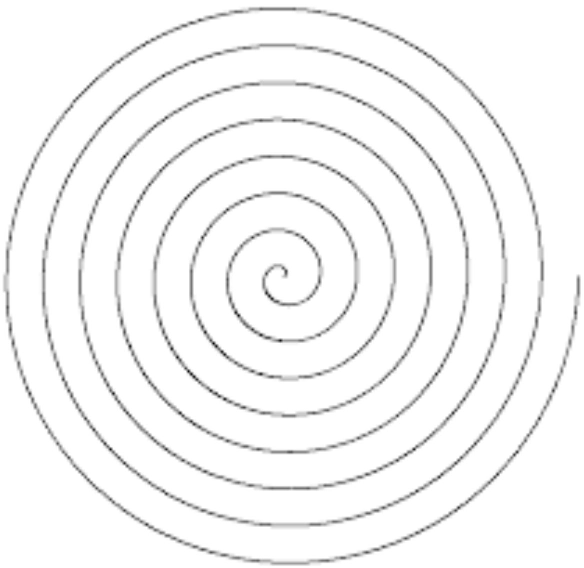
- Scanning Probe STM/AFM
  - I-V, F-z, electrochemical, dissipation, acoustics, friction, piezo-responsive…
- Confocal/Near Field Optical
  - hyperspectral Raman, PL, PLE, lifetime, pump-probe, transient absorption, polarization, epifluoresence…..
- Analytic SEM
  - Cathodoluminesence, Quantitative current imaging/EBIC, Reflection EELS, Auger Spectroscopy, XRF/EDS/WDS, EBDC…
- Analytic STEM
  - EELS, XRF, CBED, BF/DF, SE, HAADF…
- X-ray synchrotron methods
  - STXM, SFXM…

Nanomaterials Characterization

Data to Damage Ratio!

- SEM
  - heating, radiation damage, contamination, charging (image and electronic properties)…
- STEM
  - SEM modes plus lattice damage/atom displacement, ice radiolysis…
- STM/AFM
  - tip change/wear, sample wear/contamination, tip-induced dynamic processes, vibrational excitations causing chemistry, diffusion…
- Optical
  - thermal damage, melting/ablation, flurophore bleaching…

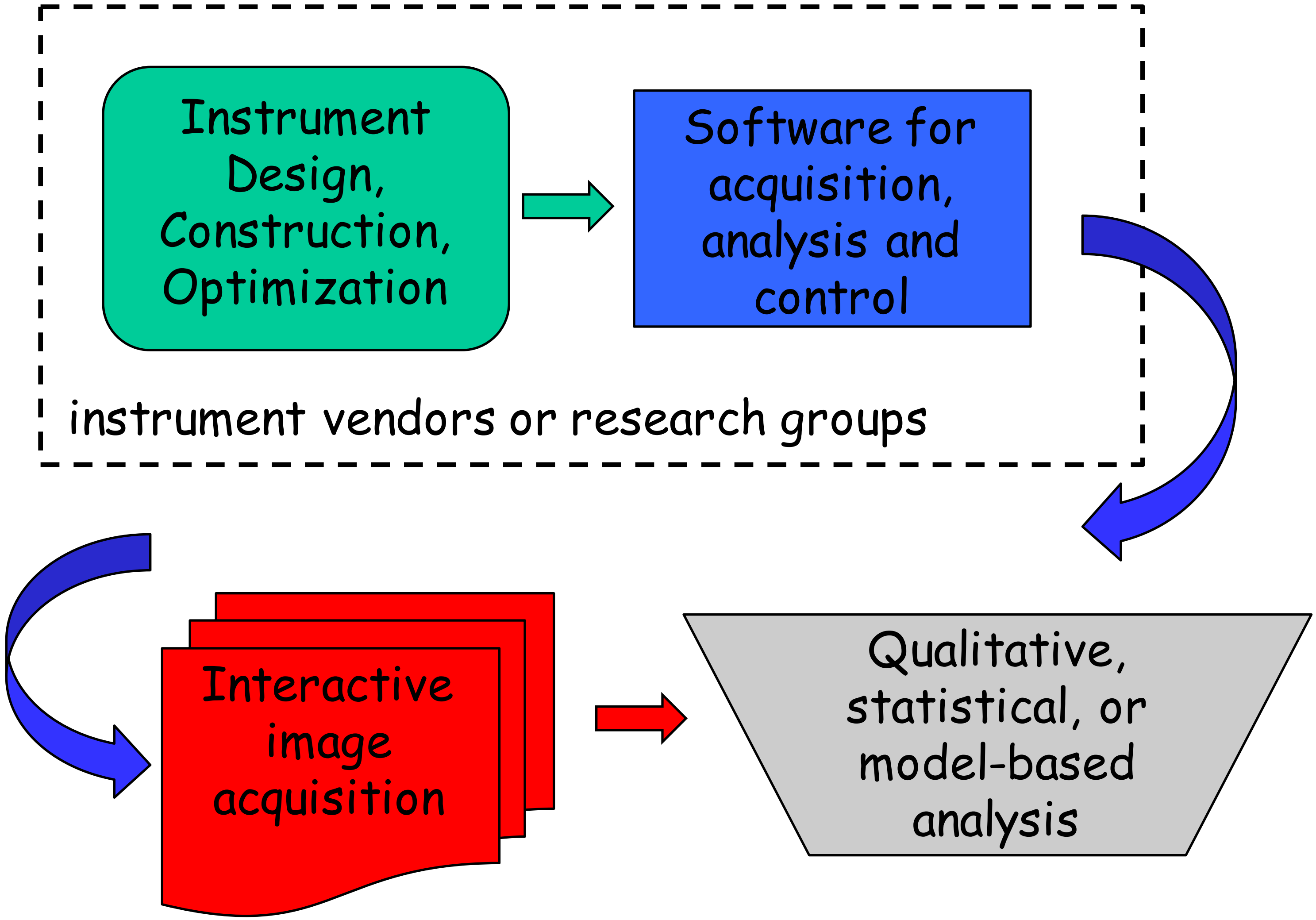
Smarter Acquisition - Front End



- experiments not just images
- fast images - slow spectra
  - scan region once per spectral point
  - feature tracking during acquisition
  - depends on relative speed of instruments
- "adaptive" acquisition
  - automatic object finding, detail where its needed
  - low SNR image to find regions for hyperspectral mapping
  - SNR threshold not fixed time for spectra
- spiral scanning - Paul Ashby
  - edge detection/following

Frank Ogletree and Ed Barnard

Imaging Instrument Paradigm



vendor instrument software

- often the "weak link",
  - less capable than hardware
  - lags behind in software engineering, exponential growth in computing power
- SEM modify data before digitization/storage
  - no quantitative data, "contrast and brightness", "channel mixing", limited data channels
  - poor or no drift correction, no concept of spectroscopy, low dose imaging, copy "analog" video burn edge/corners
  - very minimal data visualization, pay extra for contrast…
- SPM
  - generally more powerful software but proprietary formats, can be unstable/crashes
  - limited scripting/programming (zB Asylum Igor, Nanonis Lab View)
- Optical microscopy
  - software mostly for bio imaging applications, sophisticated turn-key instruments, or build it yourself

instrument software

- *researcher developed solutions*
  - artisanal, strong integration scientce/function
  - re-inventing the wheel, undocumented or oral tradition, user hostile…
- *vendor software*
  - instruments with large customer/application base and competitive markets can have decent software for typical applications
  - Often full power of hardware is "locked out", unintended consequence or captive markets…
  - scientific "niche" markets stuck with long software redesign cycles, "locked in" to bad/proprietary choices…
- *Commercial software environments*
  - NI/Labview, Matlab…
- *what is to be done?*

One Hardware/Software Challenge Cathodoluminescence

- SEM
  - beam current/energy/focus
    - » SmartSEM GUI (computer #1), serial interface
    - » TTL beam blanker
  - scanning/image acquisition
    - » external analog scan control inputs
  - electron detectors
    - » analog and/or pulse count
    - » "classical" SEM single data stream
- extra acquisition/control
    - » RHK SPMpro scanning, counter, multichannel data (computer #2)
    - » Labview CCD, spectrometer, heater, (computer #2)
    - » SRS electronics modules
    - » Andor, Acton, Attocube, Camera, etc vendor software
- Optical Components
  - collection mirror – attocube nano-translators
    - » TTL inputs (old)
    - » closed loop USB-DLL
  - Acton grating spectrometer
    - » USB text commands
  - Andor spectroscopic CCD
    - » USB-DLL
  - Acton OMA V IR diode array
    - » USB-DLL
  - optical point detectors
    - » PMTs, APDs, pulse train
    - » IR photodiodes, analog
  - CMOS imaging camera
- Sample
  - thermocouples, heaters, cryostat, Lakeshore controller
    - » GPIB, voltage programmed

"ScopeFoundry" for Experiments

- **Emerging platform for Experiments**
  - Developed by Ed Barnard (last talk) for confocal spectro-microscopy experiments
  - Extended to fast experiments/acquisition on SEM/CL/Auger , NCEM
  - Separate processes for instrument control, user GUI, data handling
- **Include real instrument response functions ?**
- **Couple to HPC/Bigger data ?**
  - ORNL Beams??
- **Include (real time) simulations of probe-sample interactions ??**
  - Physics mostly known, tools for calculation of different aspects mostly exist, rarely used (activation barrier, learning curve…)

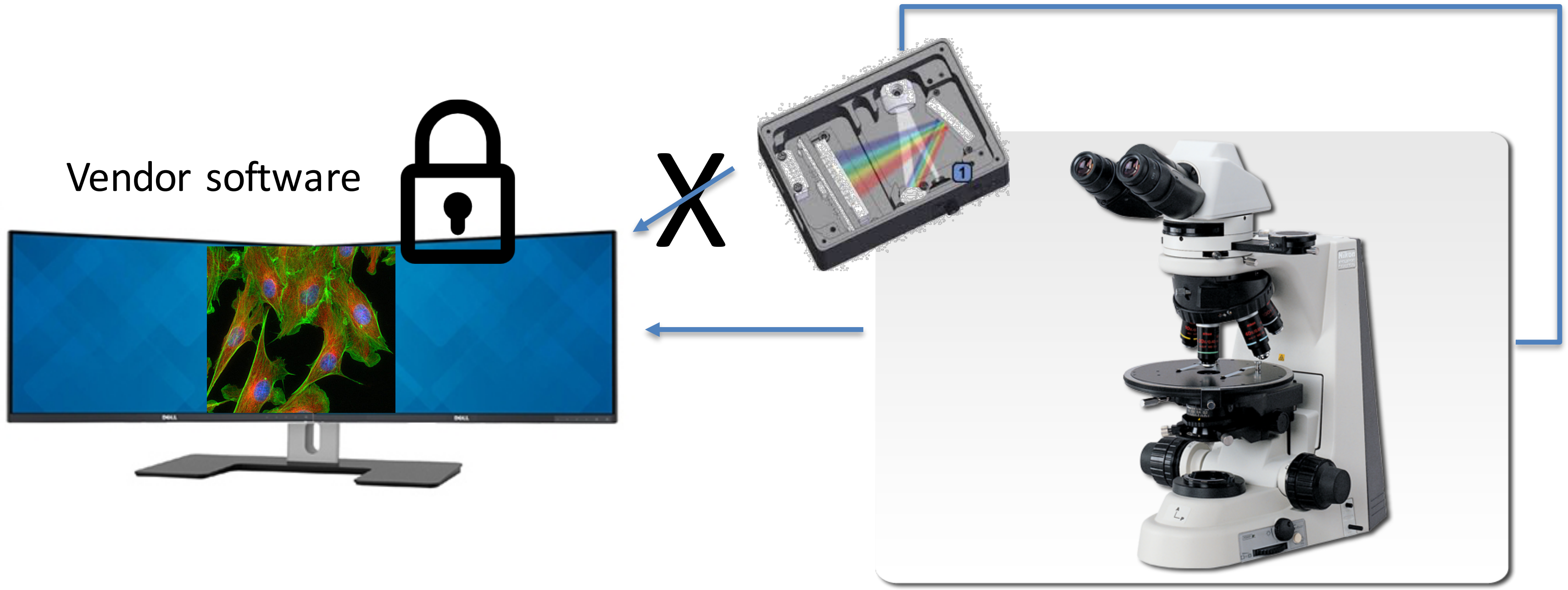
"ScopeFoundry" EcoSystem

- **Scientific Python**
  - Anaconda for Mac, Windows, Ubuntu, almost pain-free setup
  - Rapidly expanding open source toolset, connect to good numeric libraries, Device independent graphics Qt-Pyside
  - Debug on the fly during experiments (Eclipse editor)
- **Instrument control**
  - Support most common and obscure instrument interfaces
  - call DLL drivers, Serial (GBIP, USB, RS-232, etc)
- **Hardware**
  - National Instruments (DAQmx-Python)
  - NI PXI-hosted FPGA fast decision making (C DLL-Python)
  - Fast data transfer PXIe
- **Data**
  - HDF5 (Python library) images/metadata/experiments

Open Source Success ?

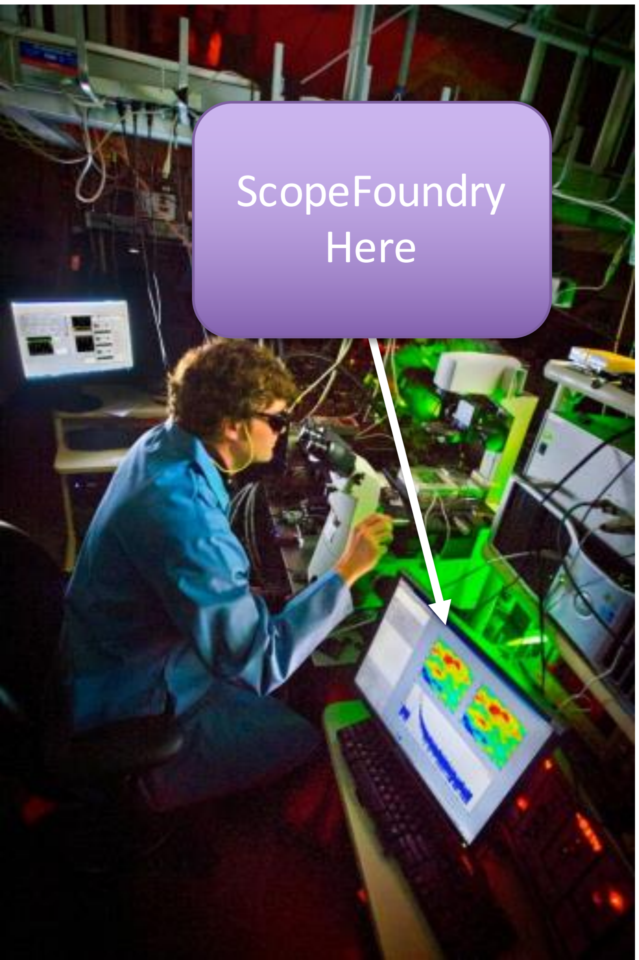
- How can viable software communities be created ?
  - many good intentions, efforts and extinct projects, standards, environments..
- Example of ImageJ
  - open source, multi-platform, extensible
  - many 100s of contributors, many 1000s of users
  - core of dedicated developers/coordinators, supported to some extent by NIH…
  - "Quantum Espresso", "NanoHub", other academic projects….
- Examples of Anaconda, WSxM
  - supported by commercial entities (Continium, Nanotec Electronica) and offered to research communities (for now)
- Role for NSRCs, National Labs, BES…?
  - support projects? joint efforts?
  - push vendors for low-level API's

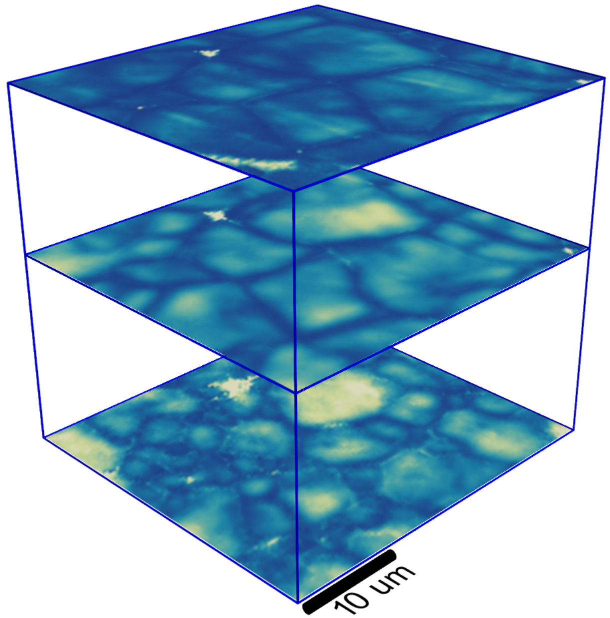
Custom Microscope Software – Why You Need It

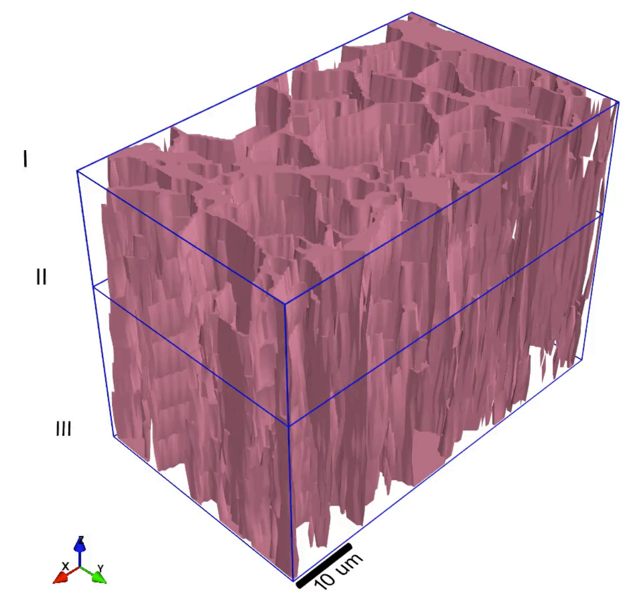


Molecular Foundry Imaging Facility: 3D mapping of lifetime in solar cells

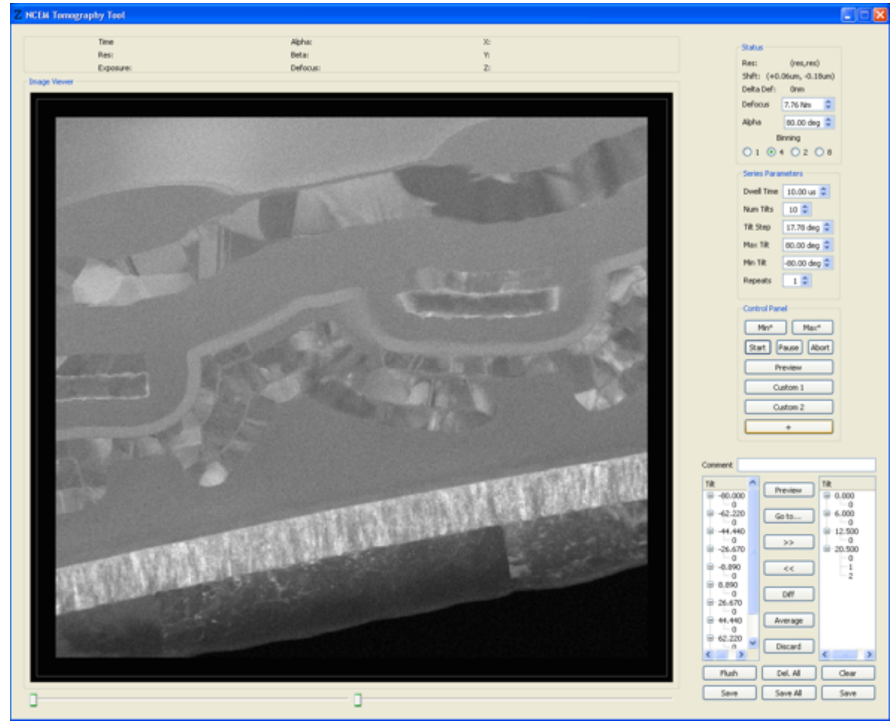
- Custom confocal Microscope
- 12 different vendor hardware pieces
- 4D (3D + t) data sets







ScopeFoundry at NCEM



TEAM Microscopes

- Smart drift correction during tomography
- New imaging modalities

Colin Ophus
Peter Ercius

Inside the Black Box:
What does Microscope Software Do?

Microscope
Software

Takes user input

- Integration time
- For series: delta time, number of images
- For scanning: scan rate, area

Takes measurements

- Moving stage to [x,y,z]
- Measures a specified property
- Storing the value associated with [x,y,z]

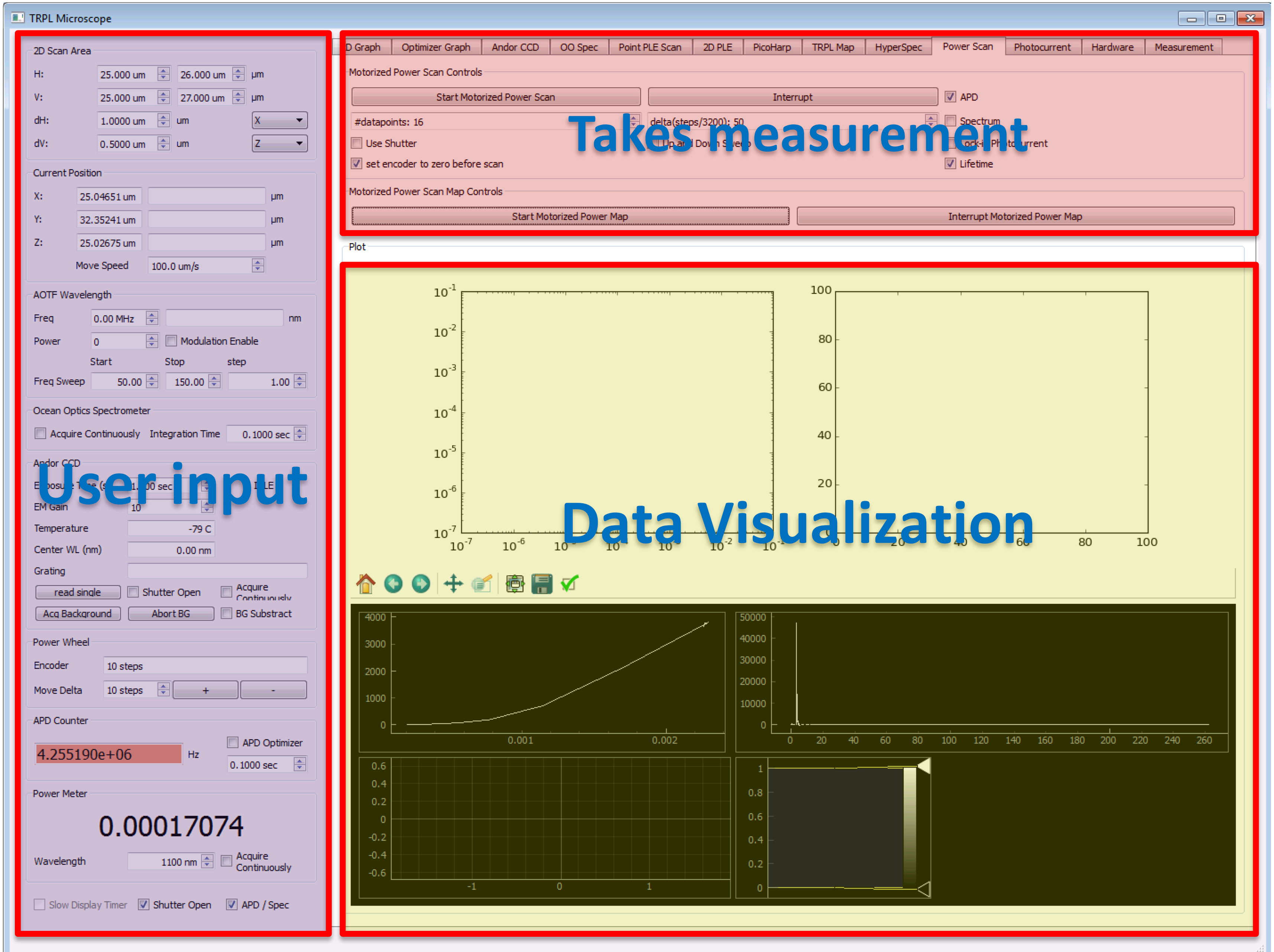
Magic

- Distortion corrections
- Background subtraction

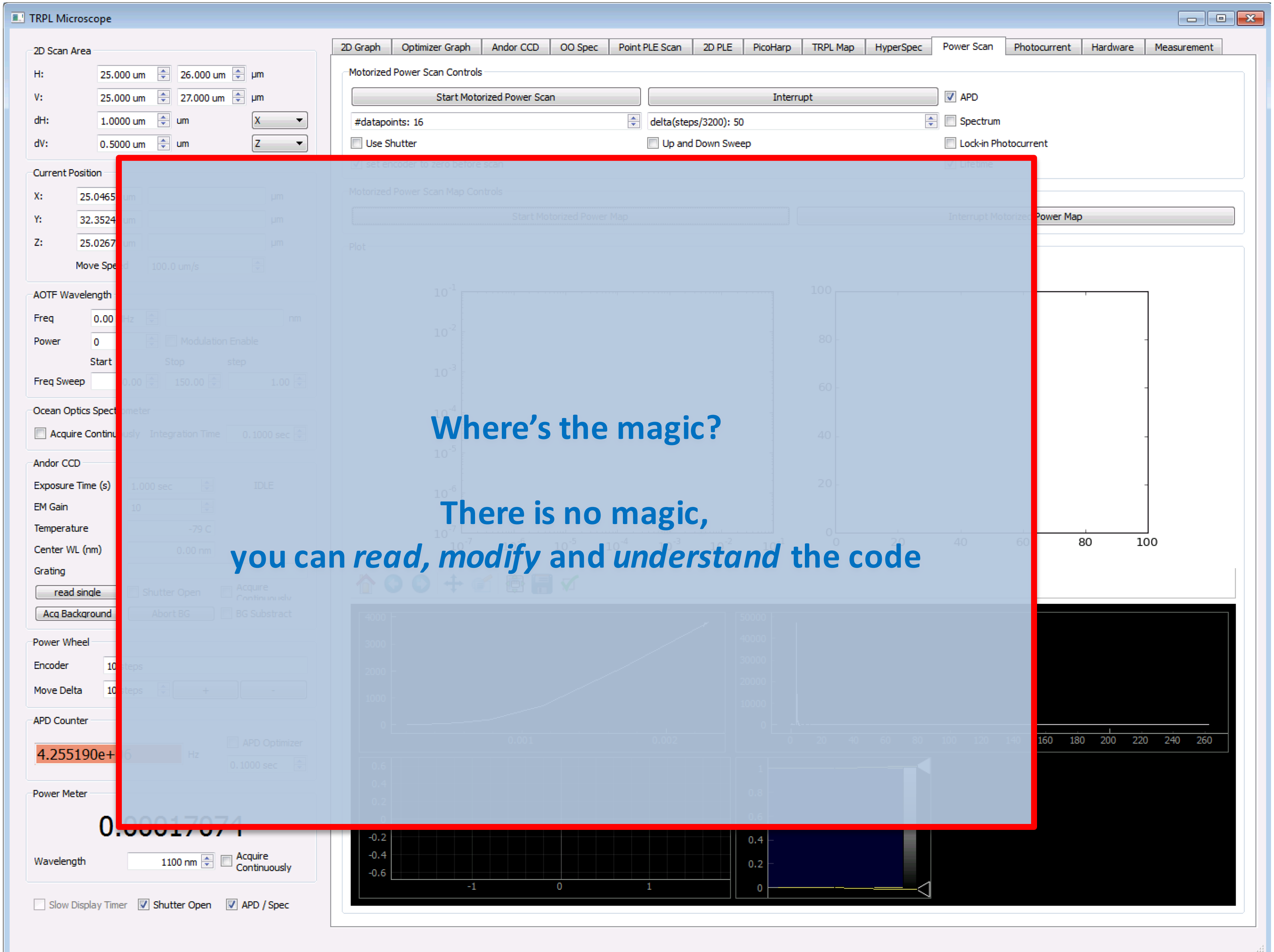
Data visualization / Post-processing / Analysis

- Flattening
- Planarization
- Statistics

ScopeFoundry: A custom microscopy control platform



ScopeFoundry: A custom microscopy control platform



Where's the magic?

There is no magic,

you can *read, modify* and *understand* the code

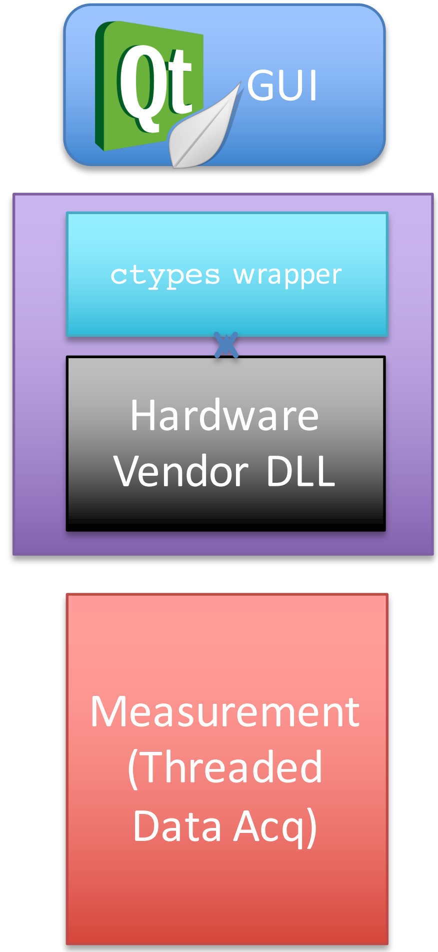
ScopeFoundry: A custom microscopy control platform

Flexible open-source tools for microscopy and lab equipment control and data acquisition

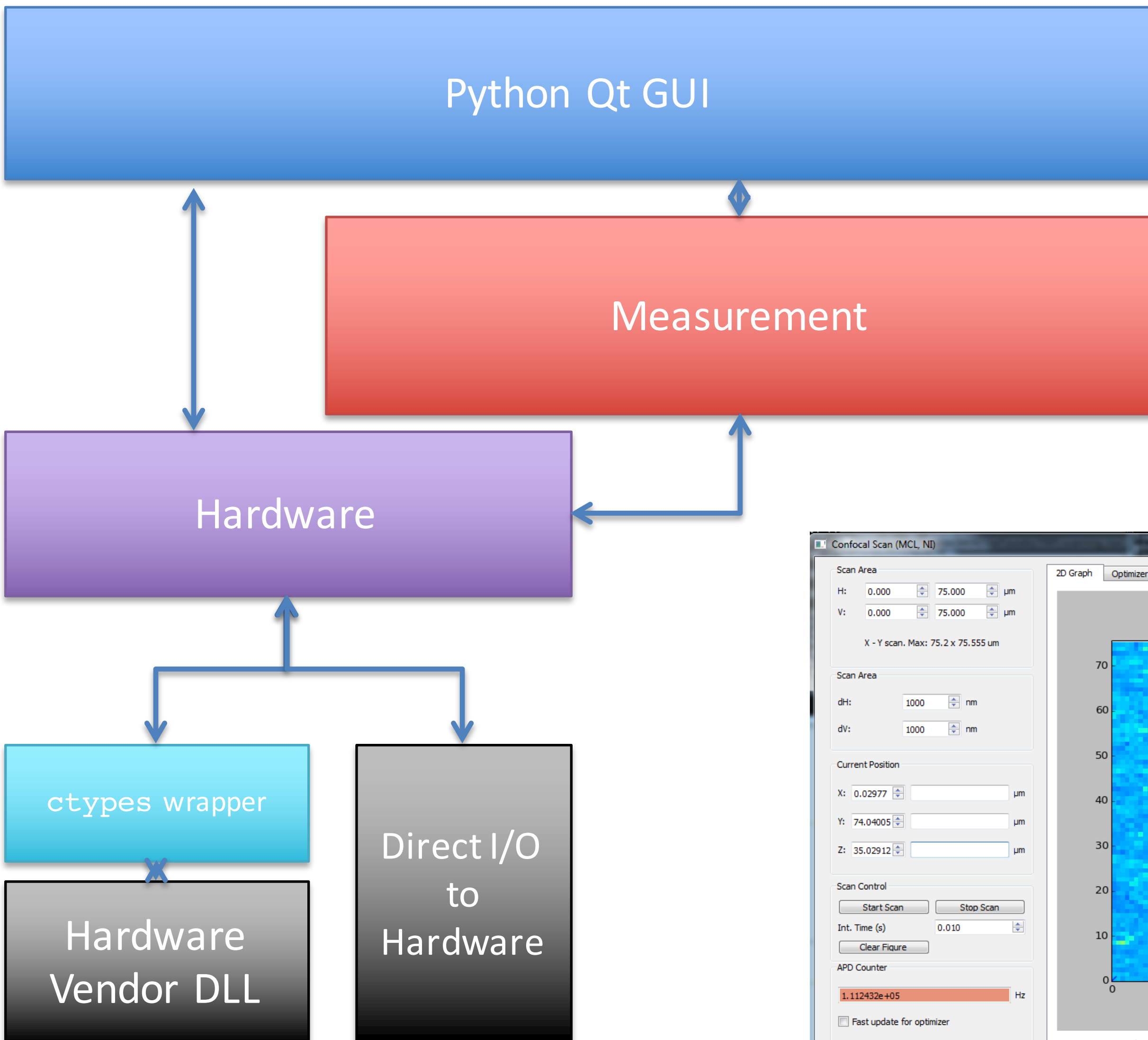
- Modular, multi-threaded Python GUI allows for fast data acquisition and visualization
- Rapid GUI builder with QT Creator
- Live updates of code for fast development and debugging
- Python bindings to C hardware driver APIs

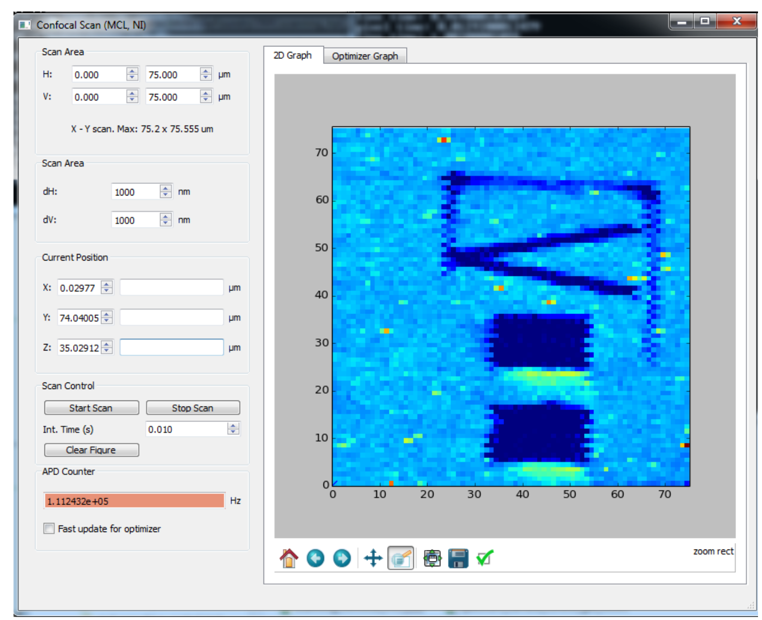
Components Needed for Microscope Software

- Graphical User Interface
  - Plots / Visualizations
  - parameter entry: Hardware & Measurement
  - Actions – start/stop, calibrate
- Hardware
  - Wrapper for vendor supplied driver
- Measurement
  - Threaded data acquisiton
    - Hardware control and coordination
    - Independent of user interface
    - Store data and write it disk
  - GUI output/visualization of data

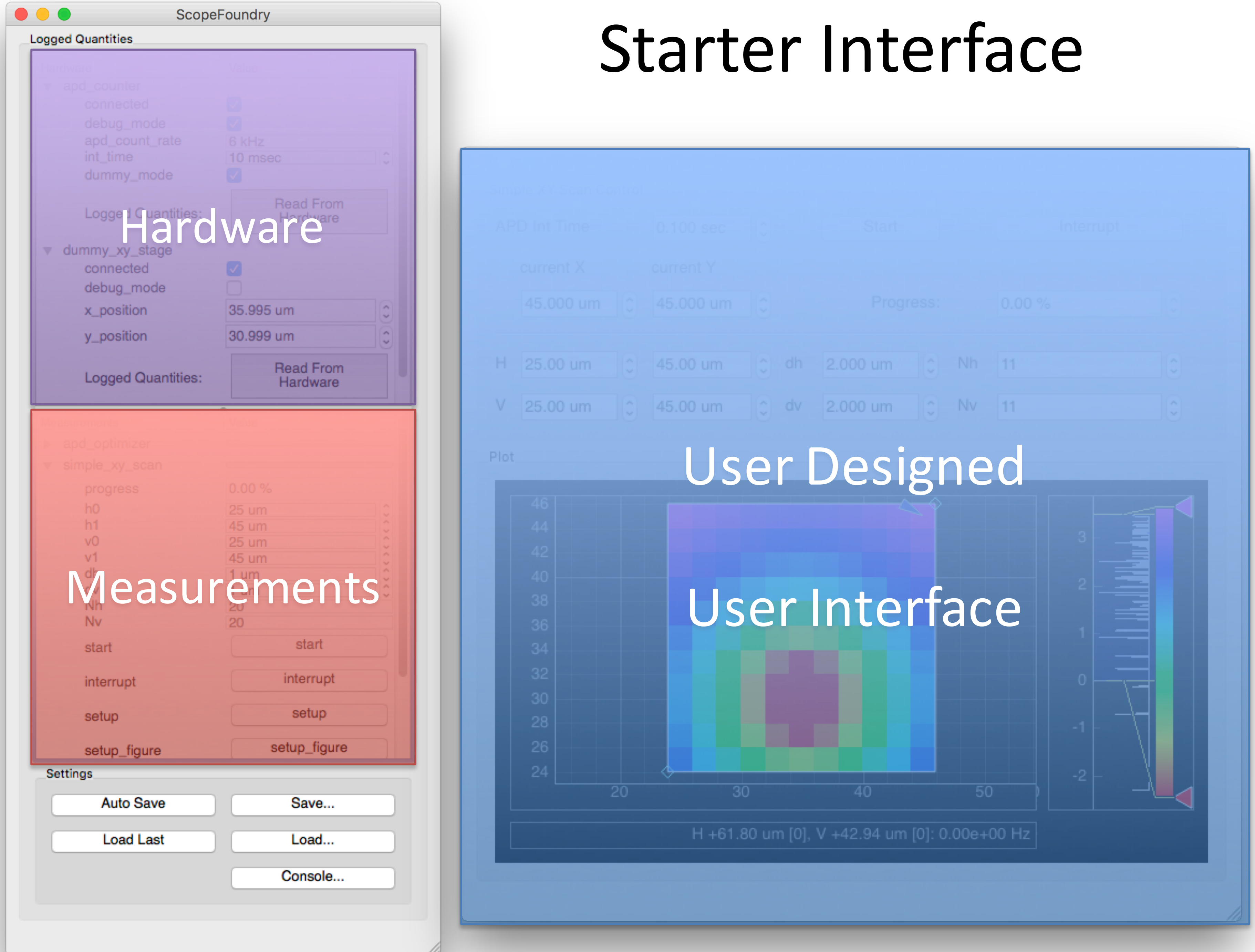


ScopeFoundry Modules

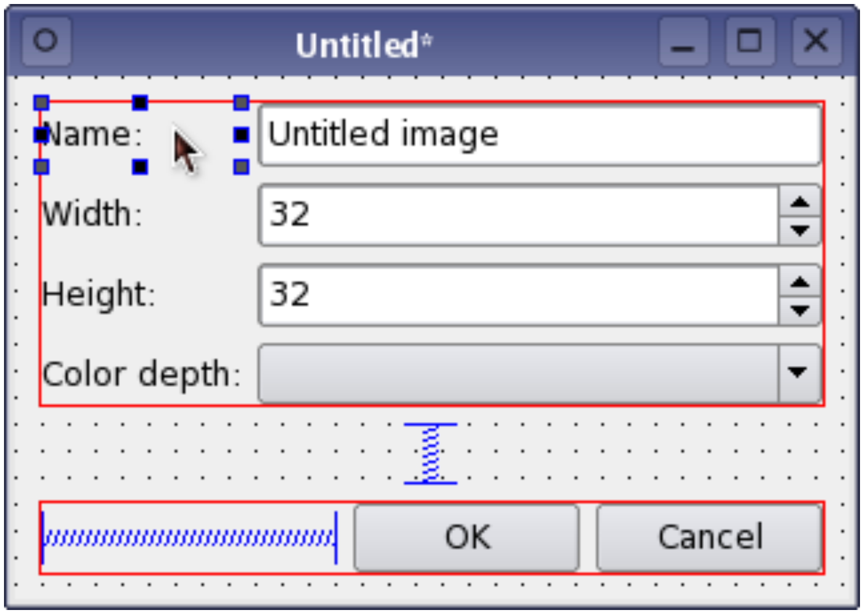




Starter Interface

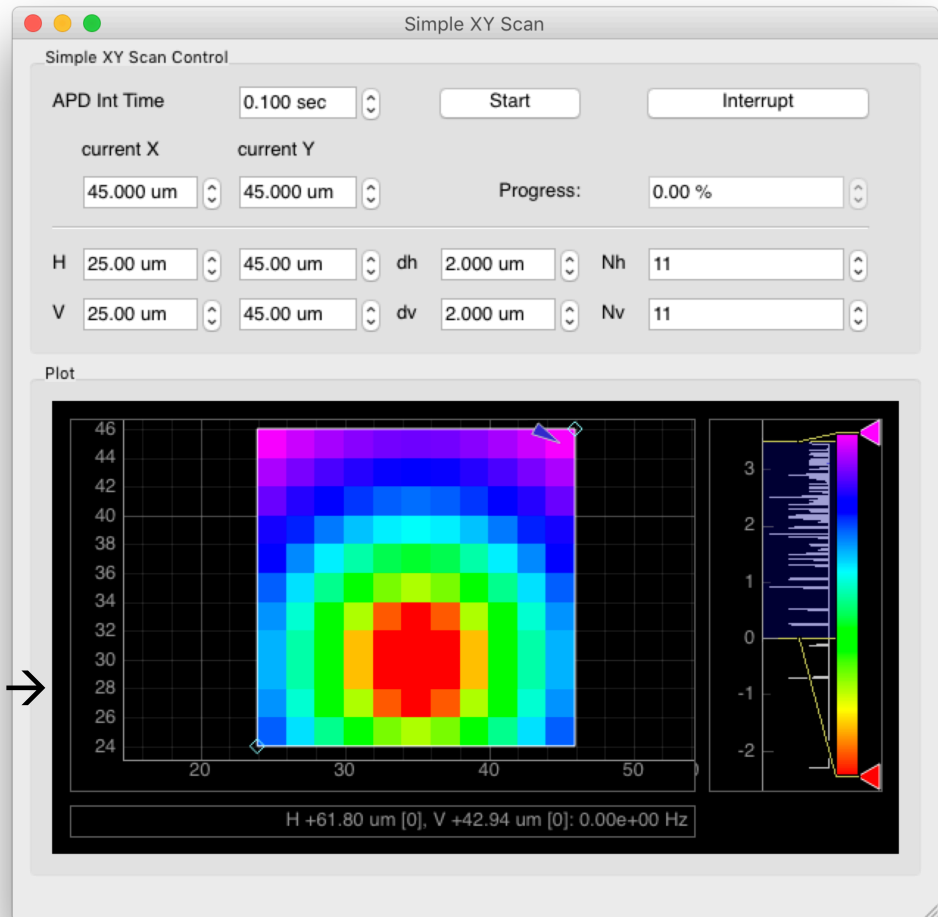


Custom Graphical Interface

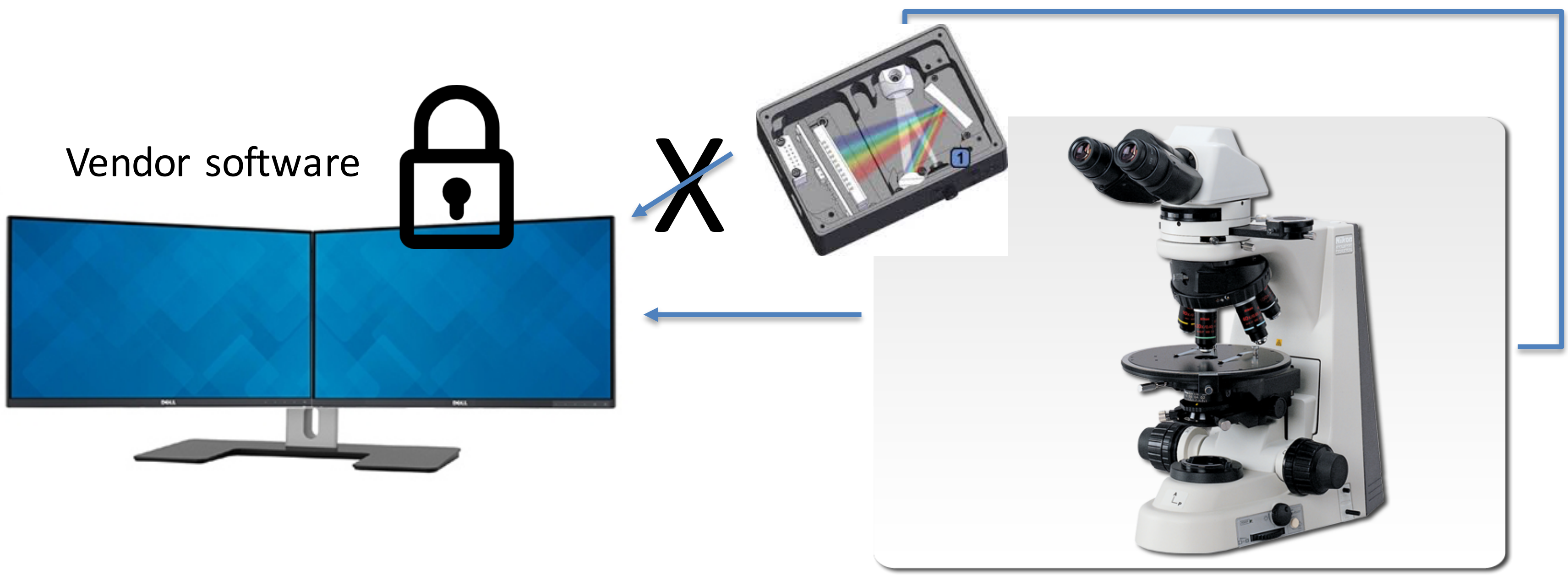


Qt Designer RAD

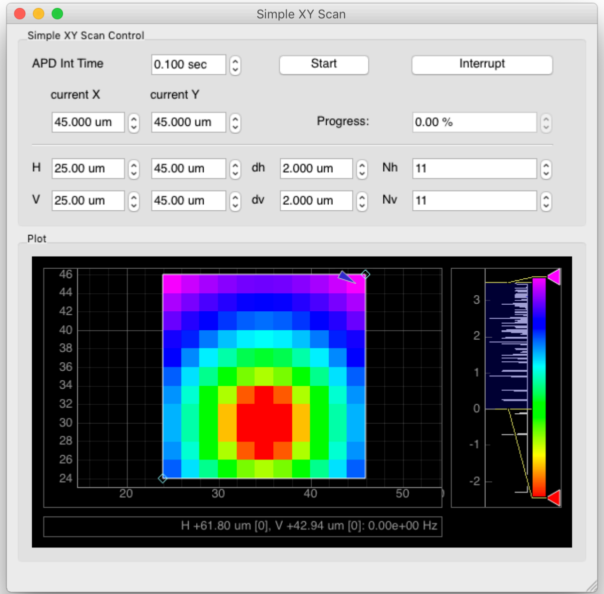
PyQtGraph plots →



Power of control over your microscope



Measurement: Simple scanning example



Threaded Run Loop:

```
for jj in range(self.Ny):
        y = self.y_array[jj]
        self.stage.y_position.update_value(y)
        self.h5_file.flush() # flush data to file every line

        for ii range(self.Nx):
            self.stage.x_position.update_value(self.x_array[ii])
            # each pixel:
            # acquire signal and save to data array
            self.pixel_i += 1
            self.apd_count_rate.read_from_hardware()
            self.apd_map_h5['data'][jj,ii] = self.apd_count_rate.val
            spectrum = self.andor_ccd.read_spectrum()
            self.spec_h5['data'][jj,ii,:] = spectrum[:]
```

Instant Hyper-spectral Imaging

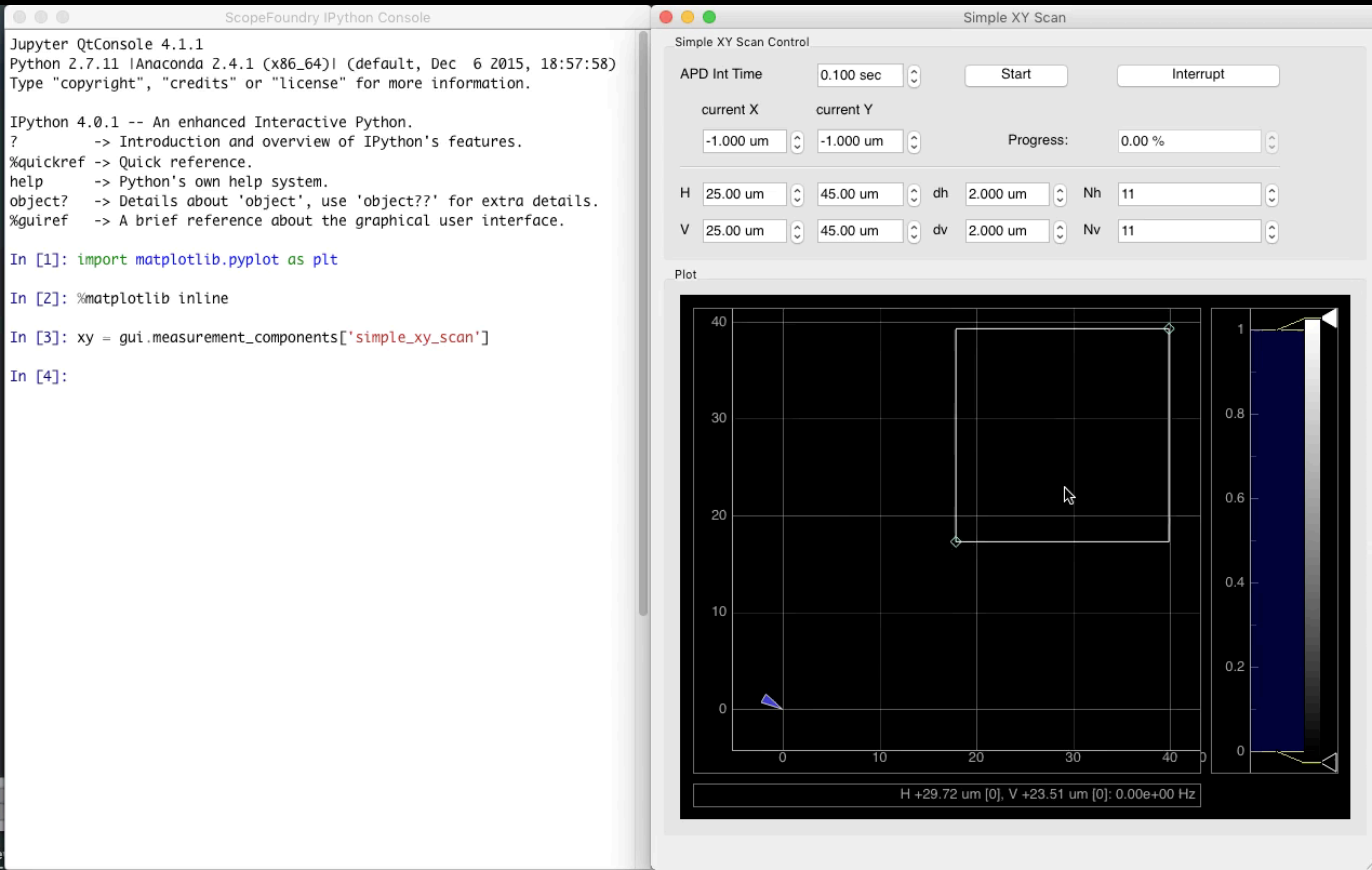
Demos

1. Interactive User Interface

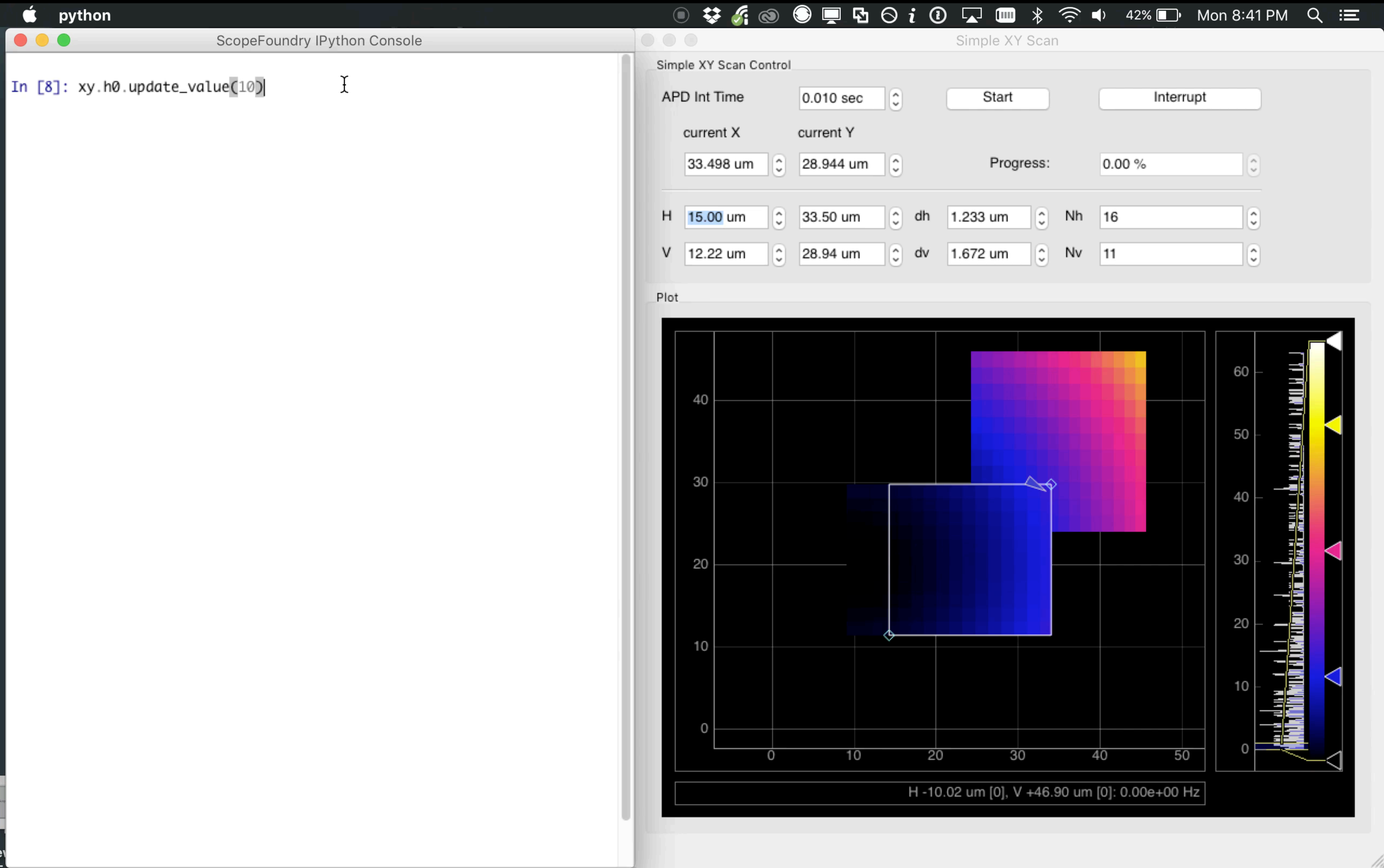
2. In-depth online data access and control

3. Live code updates – great for debugging!

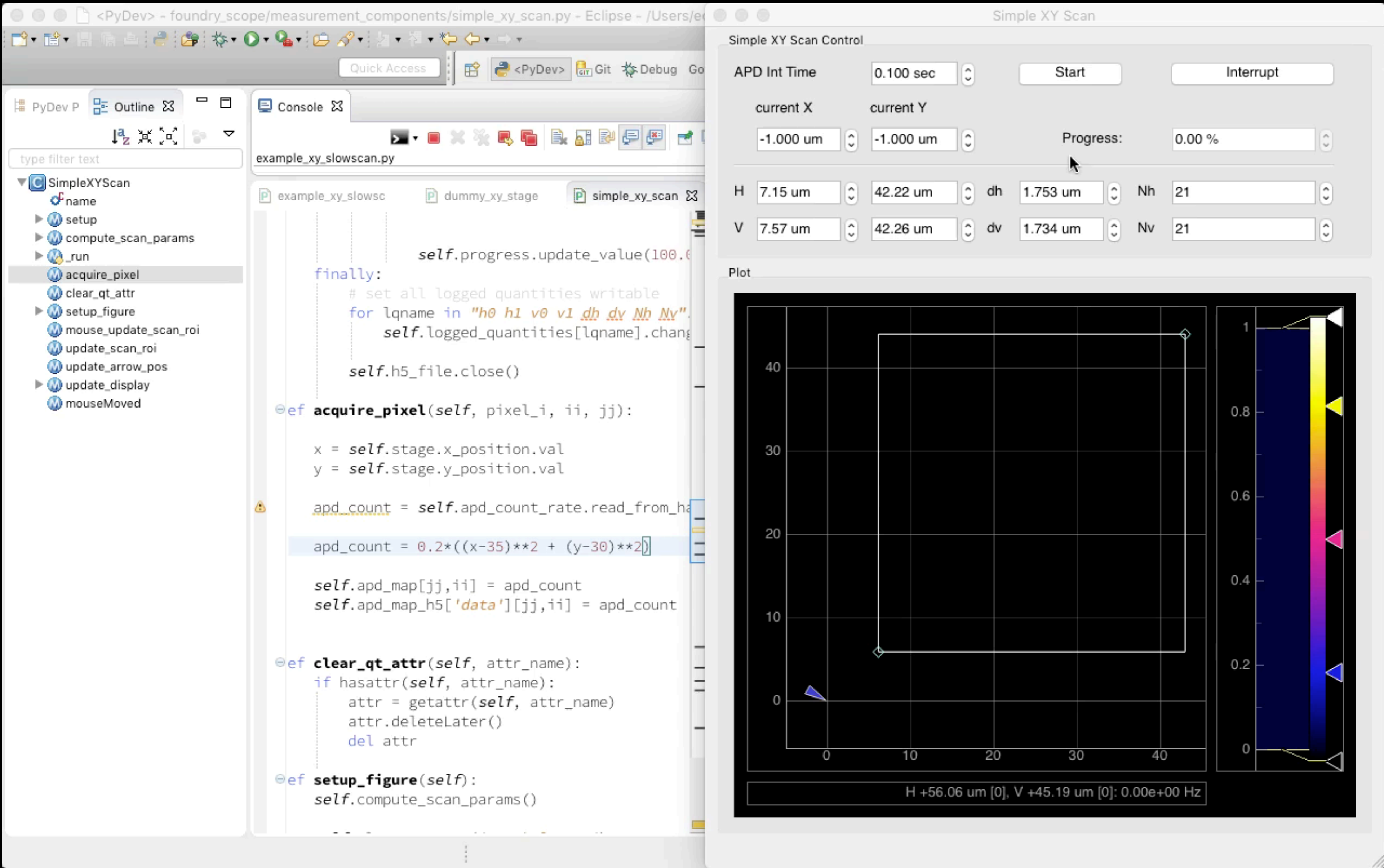
Interactive User Interface

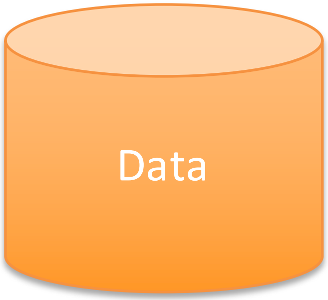


IPython interactive data access

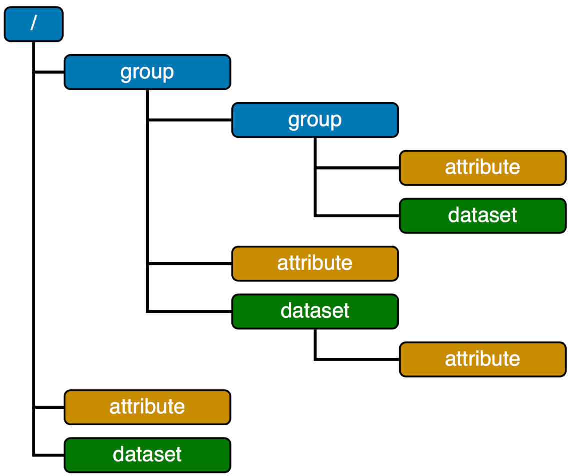


Live Code Update





Standarizing data formats: HDF5



HDF file root group.
Group belonging to root group.
Group belonging to another group.
Attribute beloning to a group.
Dataset belonging to a group.
Attribute belonging to a group.
Dataset belonging to a group.
Attribute belonging to a dataset.
Attribute belonging to root group.
Dataset belonging to root group.

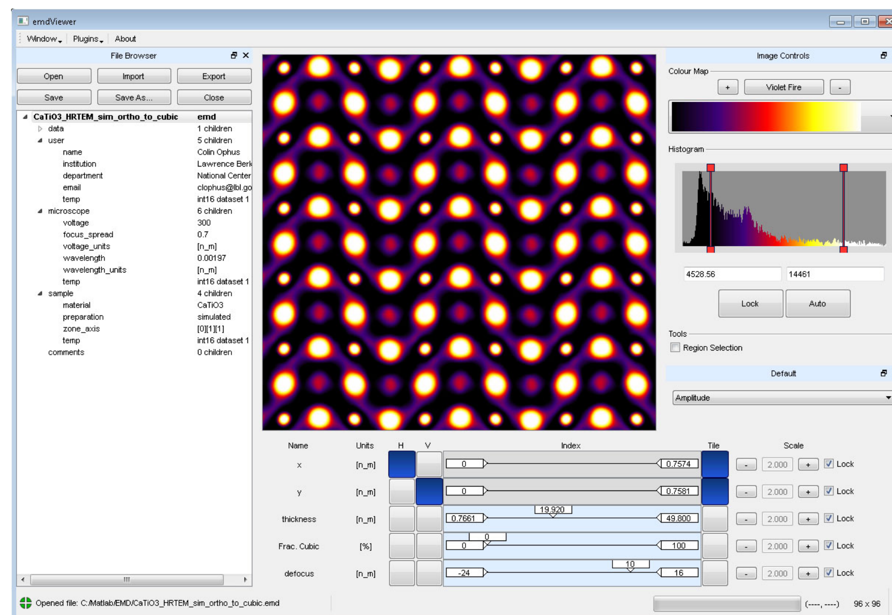
HDF5:

Open source library for handling hierarchical data with 'attributes' (i.e. metadata)

Programming language agnostic

EMDViewer
NCEM is developing an open source viewer for N-dimensional HDF5 data

Colin Ophus
http://emdatasets.lbl.gov/



Conclusions

- ScopeFoundry used in many measurement techniques at the Molecular Foundry, NCEM. Not all are scanning microscopy
- General availability soon!
- Come talk to us about using it for your experiments

Frank: dfogletree@lbl.gov
Ed: esbarnard@lbl.gov