Hardware Triggered Scanning: Introduction

Philip Taylor, Emma Arandjelovic

Observatory Sciences Limited

Course Aims

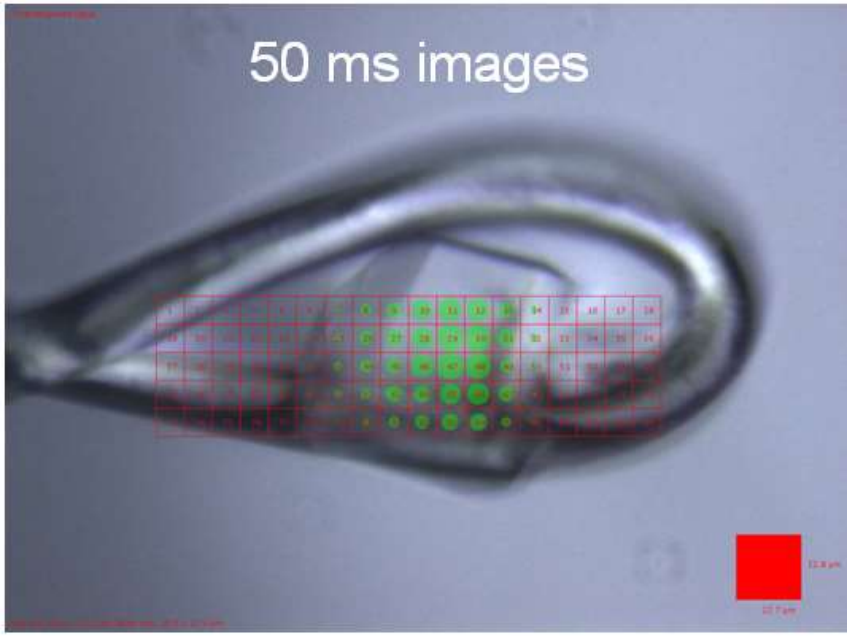
1. Get to know the functionality provided by the hardware triggered scanning stack
2. Understand the architecture, and acquire basic knowledge of all the components
3. Get hands-on experience of setting up scans
4. Learn how to debug common problems!

Course Content

1. Introduction and demo
2. Overview of the hardware
3. Low level control and data acq. concepts
4. Scan configuration
5. Experiment control and data processing
6. Practical exercises 

Grid Scanning Overview

- Move sample in X and Y to scan it through the beam
- Rewind at the end of each row or reverse direction (‘alternate/snake’ scan)



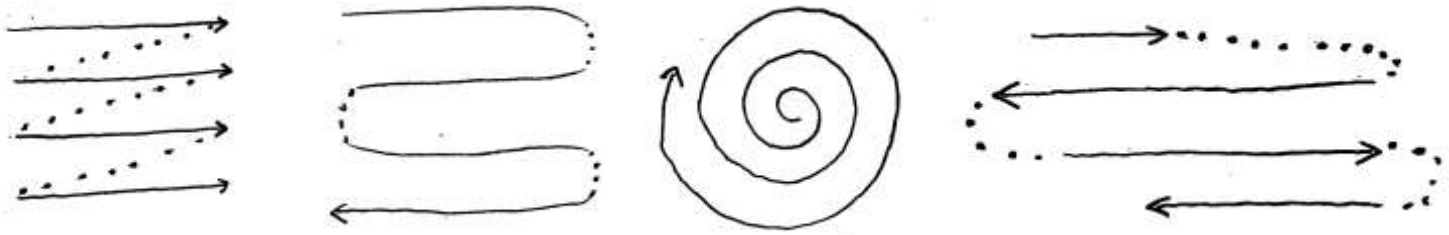
Data acquisition can be done using:

1. Software step scan
2. Hardware step scan
3. Continuous scan

Continuous scans are much faster! But also complex…

Framework Motivation

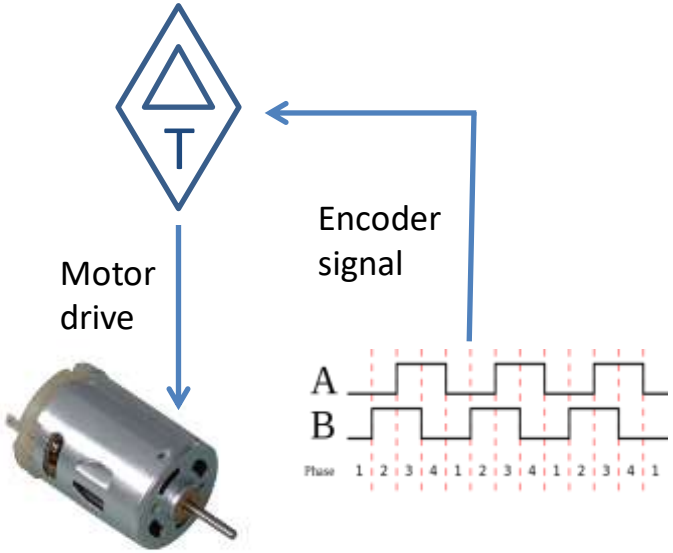
- Growing need for fast, continuous scanning
- Complex trajectories – not just grids!



- Need for on-the-fly visualization of data
- Desire for a generic solution

Hardware Components

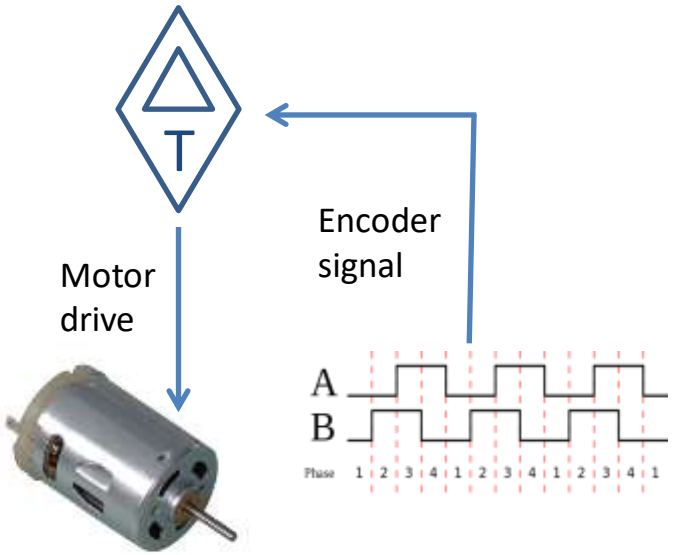
Motion trajectory control

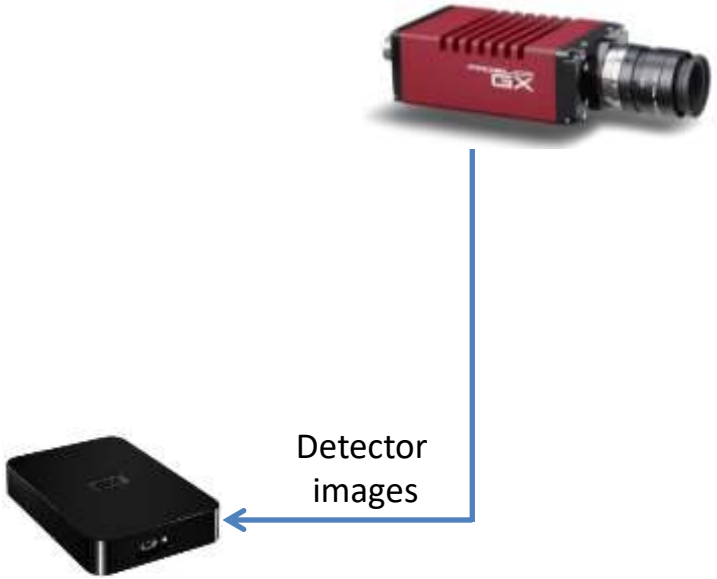


Hardware Components

Motion trajectory control

Data capture



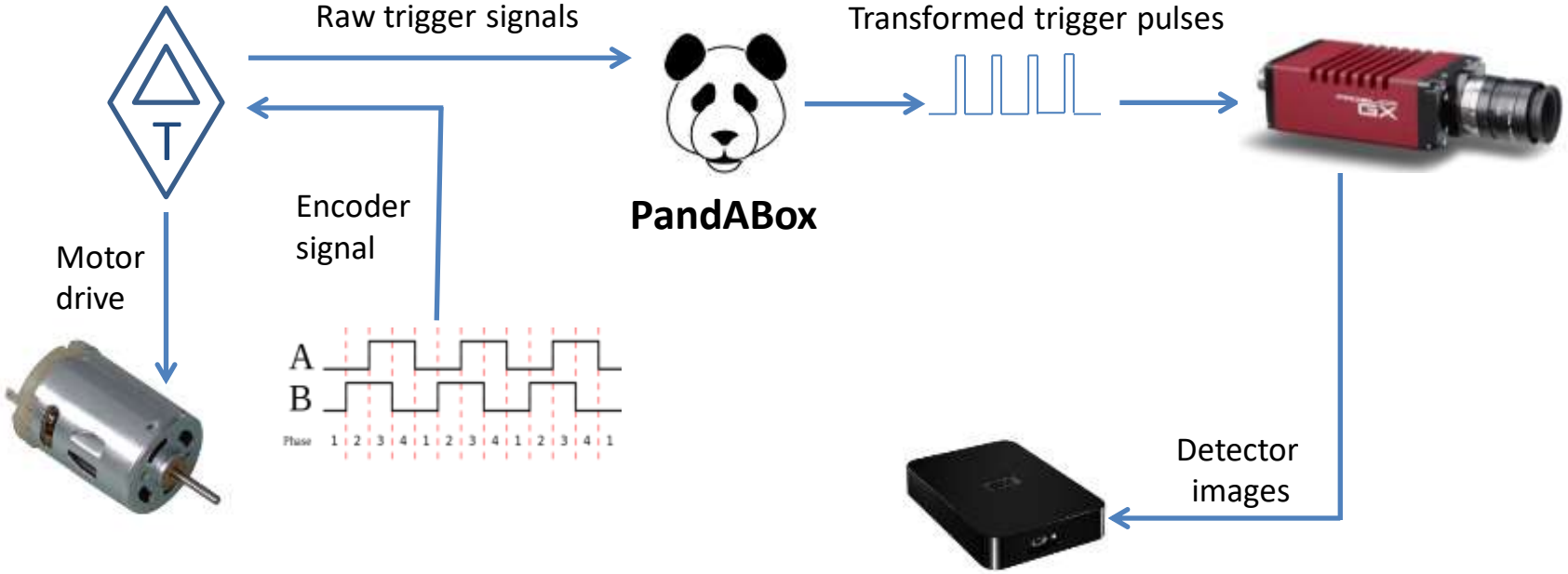


Hardware Components

Motion trajectory control

Flexible triggering, and fast position capture

Data capture

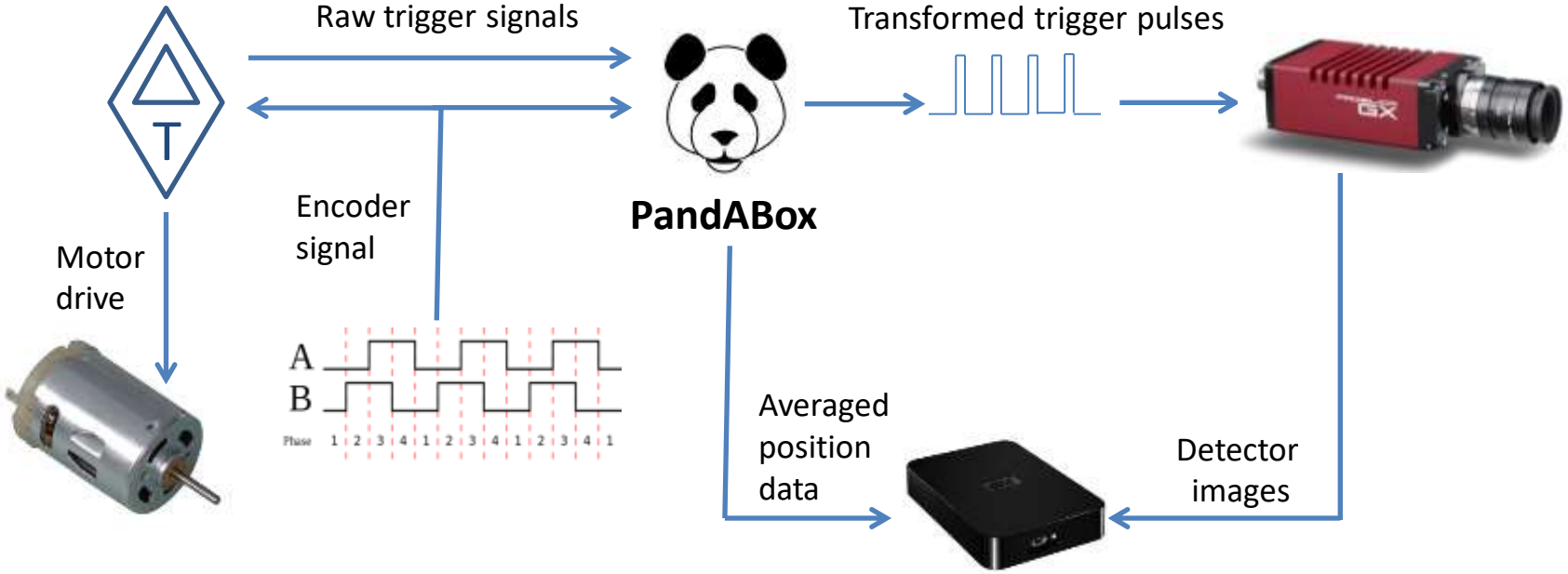


Hardware Components

Motion trajectory control

Flexible triggering, and fast position capture

Data capture



Software Components

Data Analysis WorkbeNch

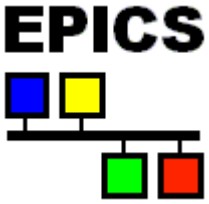
- Analysis and visualization

Generic Data Acquisition

- Experiment setup and supervision

Malcolm

- Scan configuration

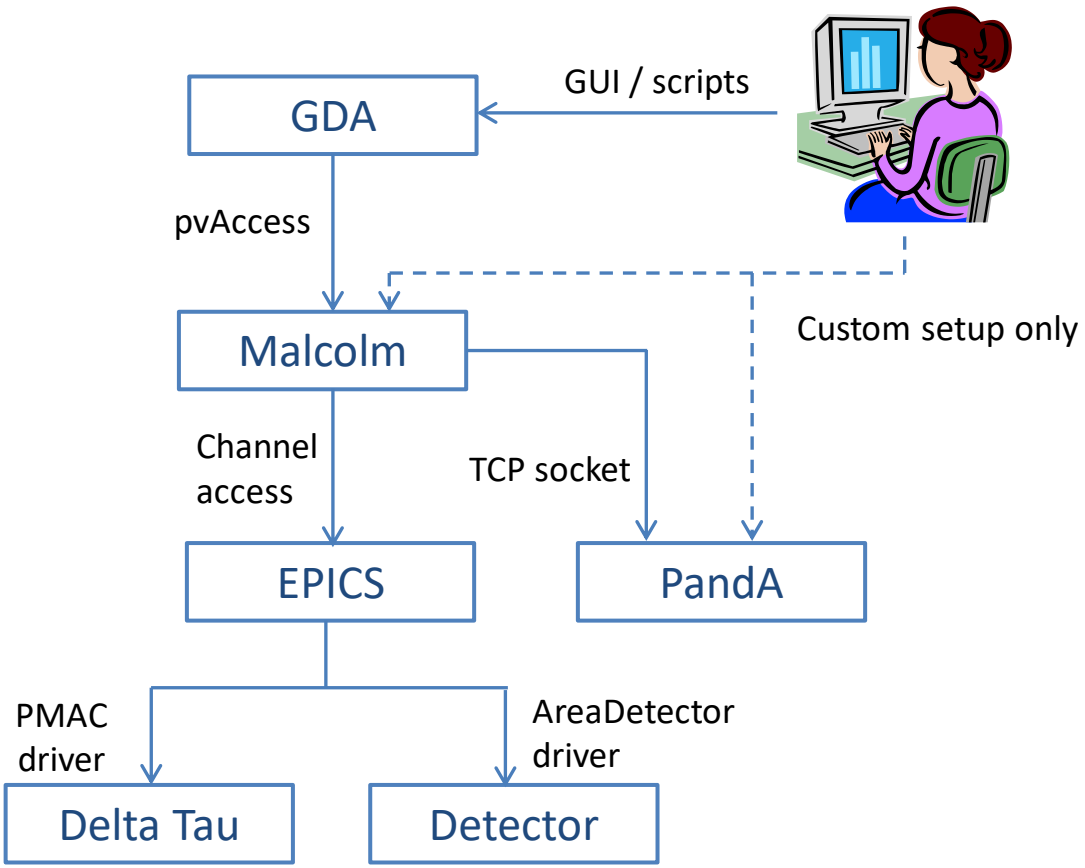


Experimental Physics & Industrial Control System

- Low level control of hardware

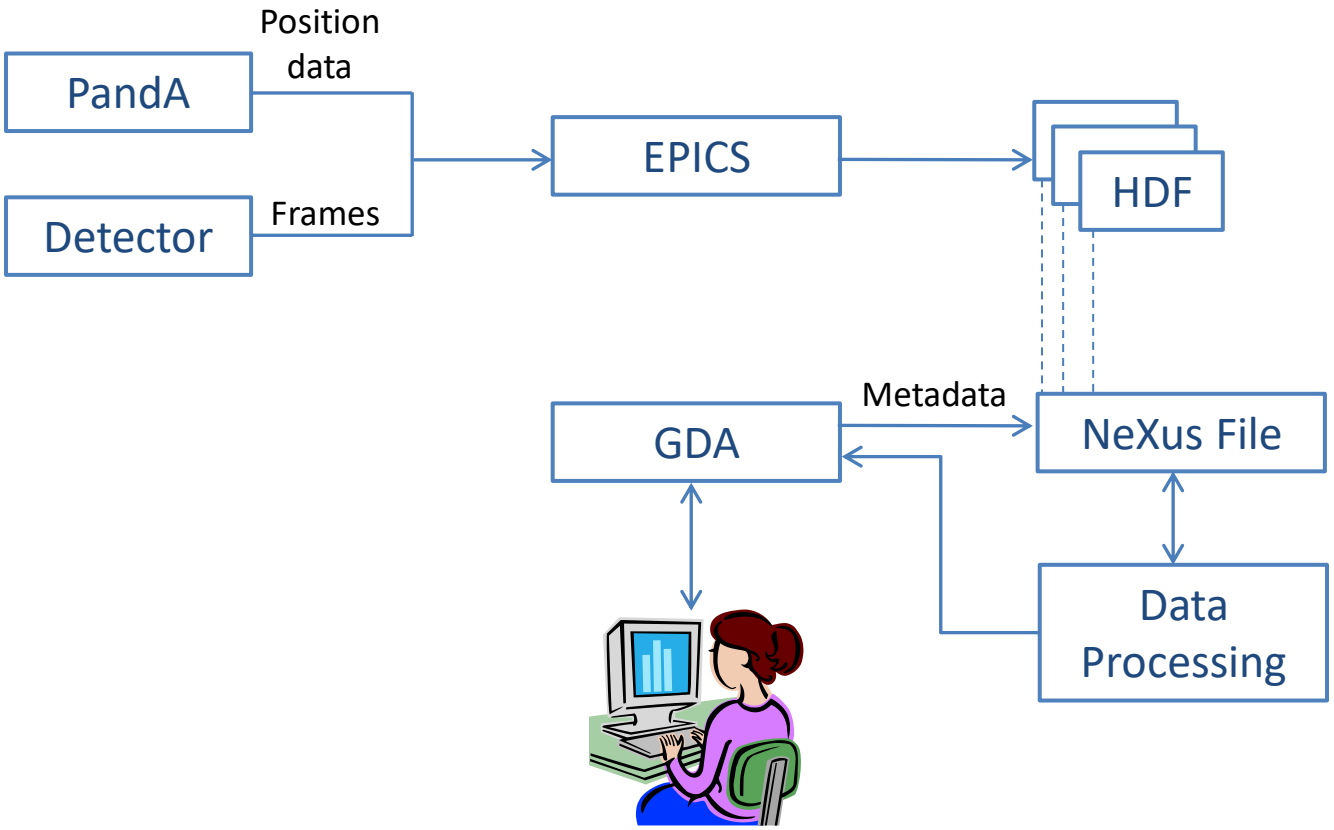
System Architecture

Control Flow

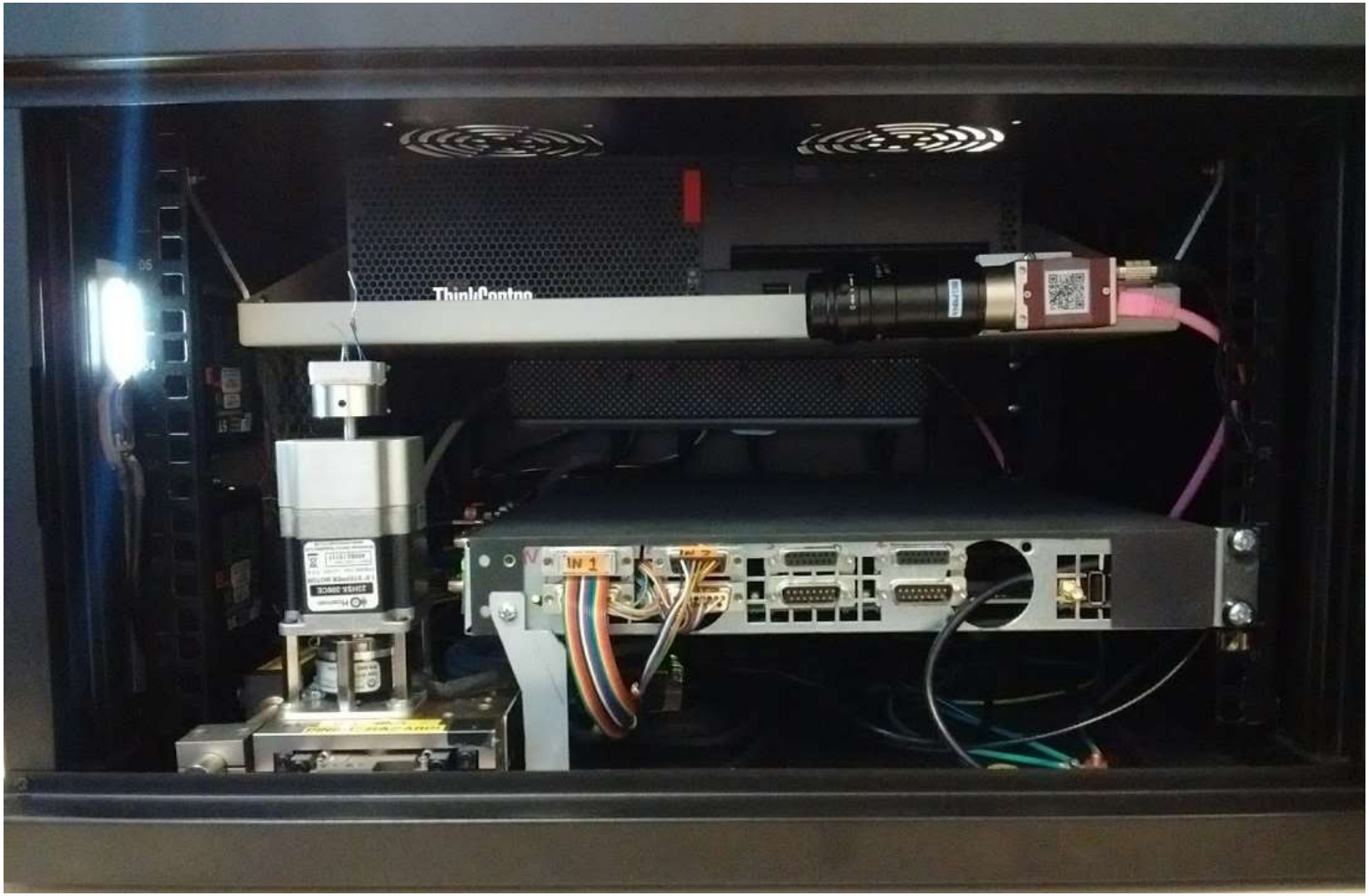


System Architecture

Data Flow



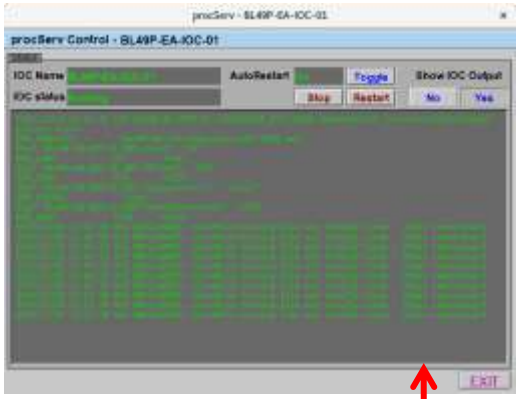
Test Rig

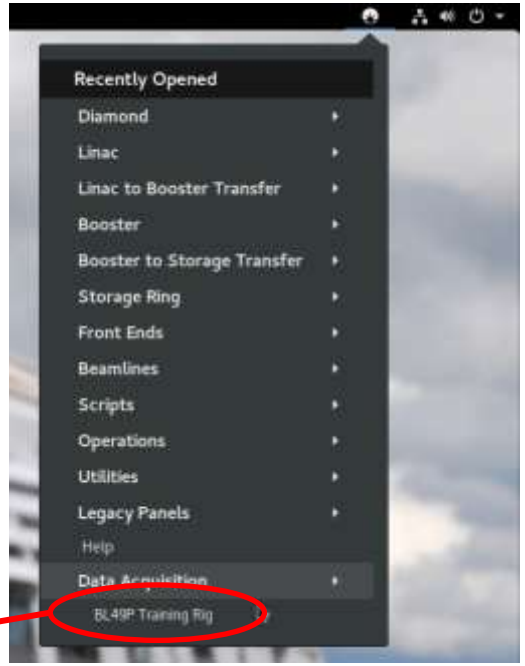


Test Rig

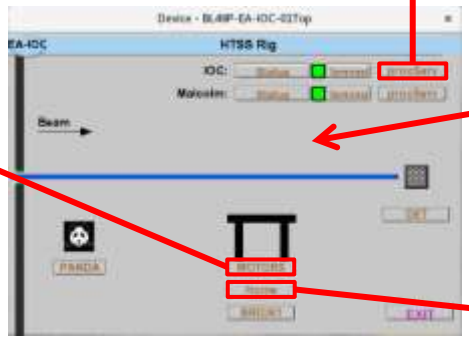
- Beamline in a box!
  - Light illuminated ‘sample’
  - Two stepper motors (θ and X)
  - DT Turbo Clipper controller
  - Allied Vision Mako camera
  - PandA electronics
  - PC with full software stack for testing and training

Test Rig


procServ Control - BL49P-EA-IOC-01
EXIT


Recently Opened
Diamond
Linac
Linac to Booster Transfer
Booster
Booster to Storage Transfer
Storage Ring
Front Ends
Beamlines
Scripts
Operations
Utilities
Legacy Panels
Help
Data Acquisition
BL49P Training Rig


Device - MOTORS
MOTORS Top
Move
STOP
EXIT

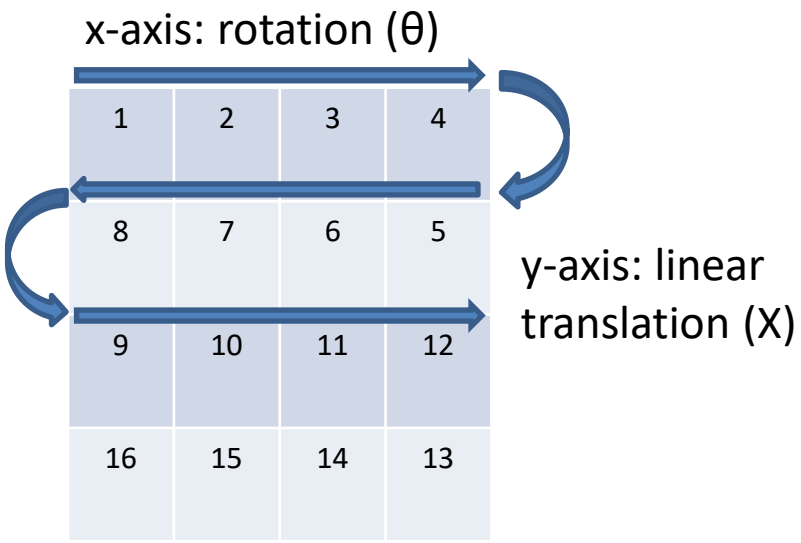

HTSS Rig
Beam
EXIT


Motor - BL49P-MO-MAP-01:STAGEX
Position
20.000 mm
Stop
Ready
Moving
Forward
Homed
Hi Limit
Lo Limit
Soft Limit
Missed
Home
Reverse
Forward
Jog
Tweak
Tweak Step
1.000 mm
Kill
Sync VAL = RBV
Direction
Pos
Set/Use
Use
Offset
Variable
Use Encoder
Yes
Calibration
Resolution
Motion
Links
Limits
Commands
Drive
Status
Servo
Alarms
Misc
Exit

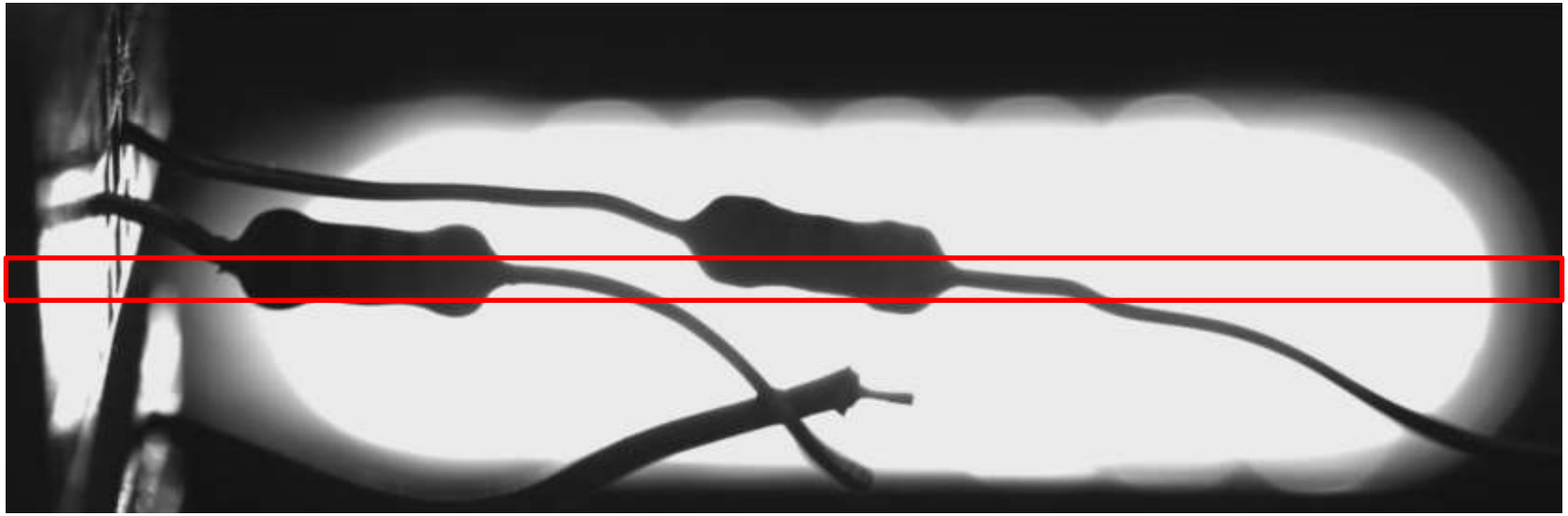

Homing - BL49P-MO-MAP-01:STAGE
Homing - BL49P-MO-MAP-01:STAGE
Homing State
Homing Status
Move timer
Group of axes to home
Home
Abort
EXIT

Demo Application

- Grid scan in θ and X
- ‘Snake’ style



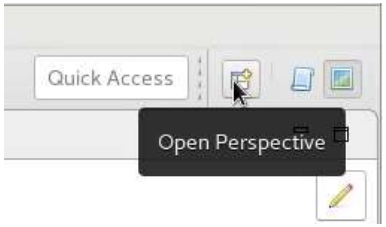
Camera View

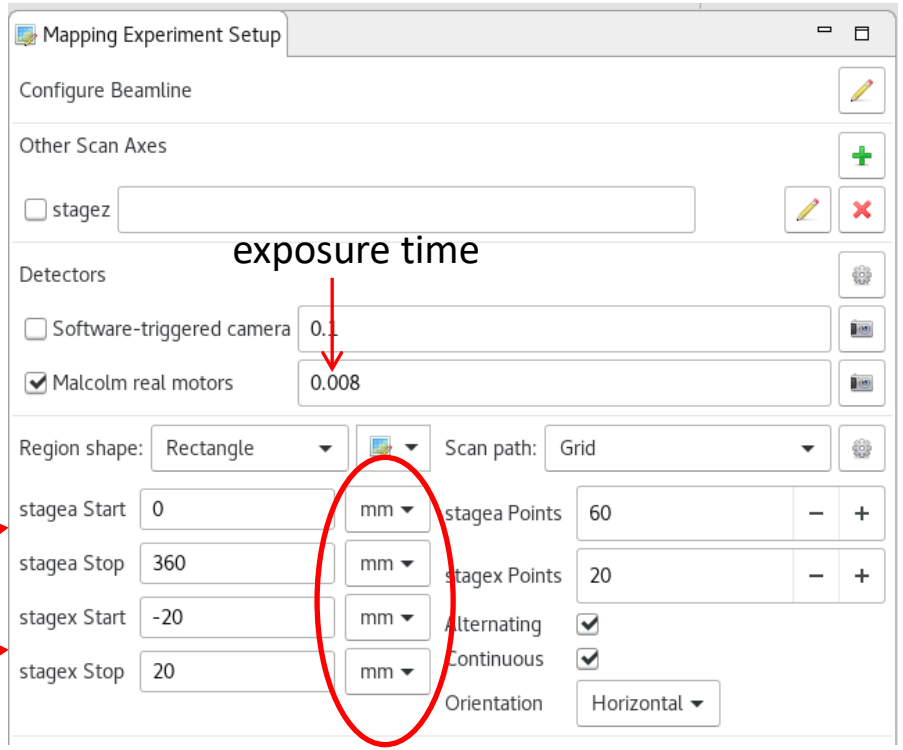


- Camera configured to capture a thin slice – each image is 1936 x 20 array
- Images used to reconstruct the sample

Stage 1: Setup scan

Open GDA mapping perspective





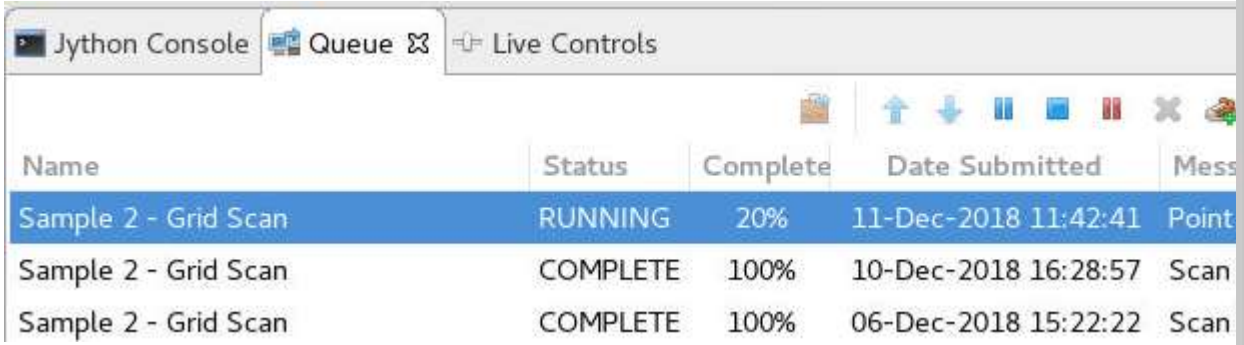
rotation axis

translation axis

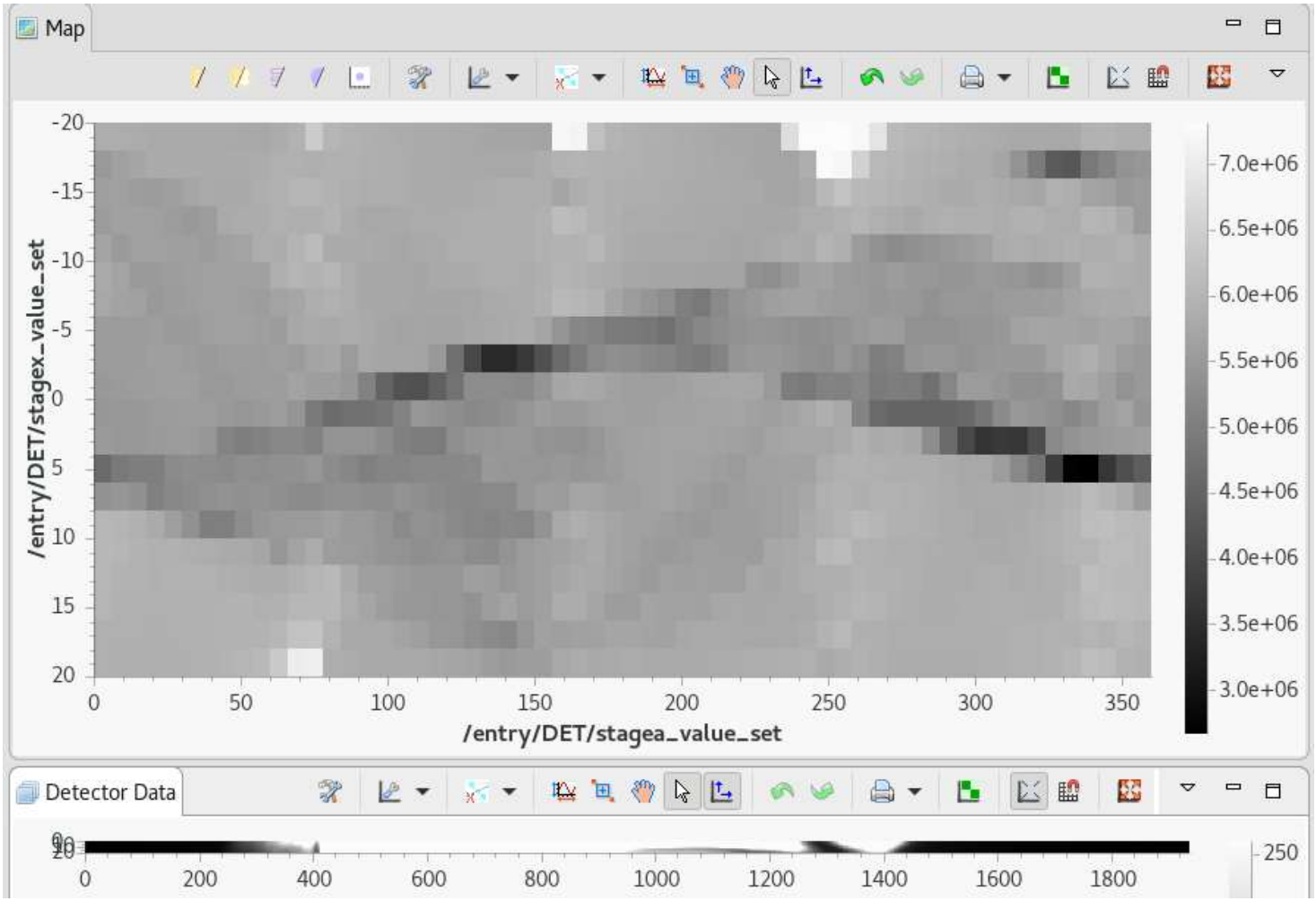
Open DET panel from the Test Rig panel and select MJPG tab

Stage 2: Start scan

- Hit ‘Queue scan’
- Motors move to initial positions
- Scan starts
- Images are acquired
- Watch the data acq. in GDA



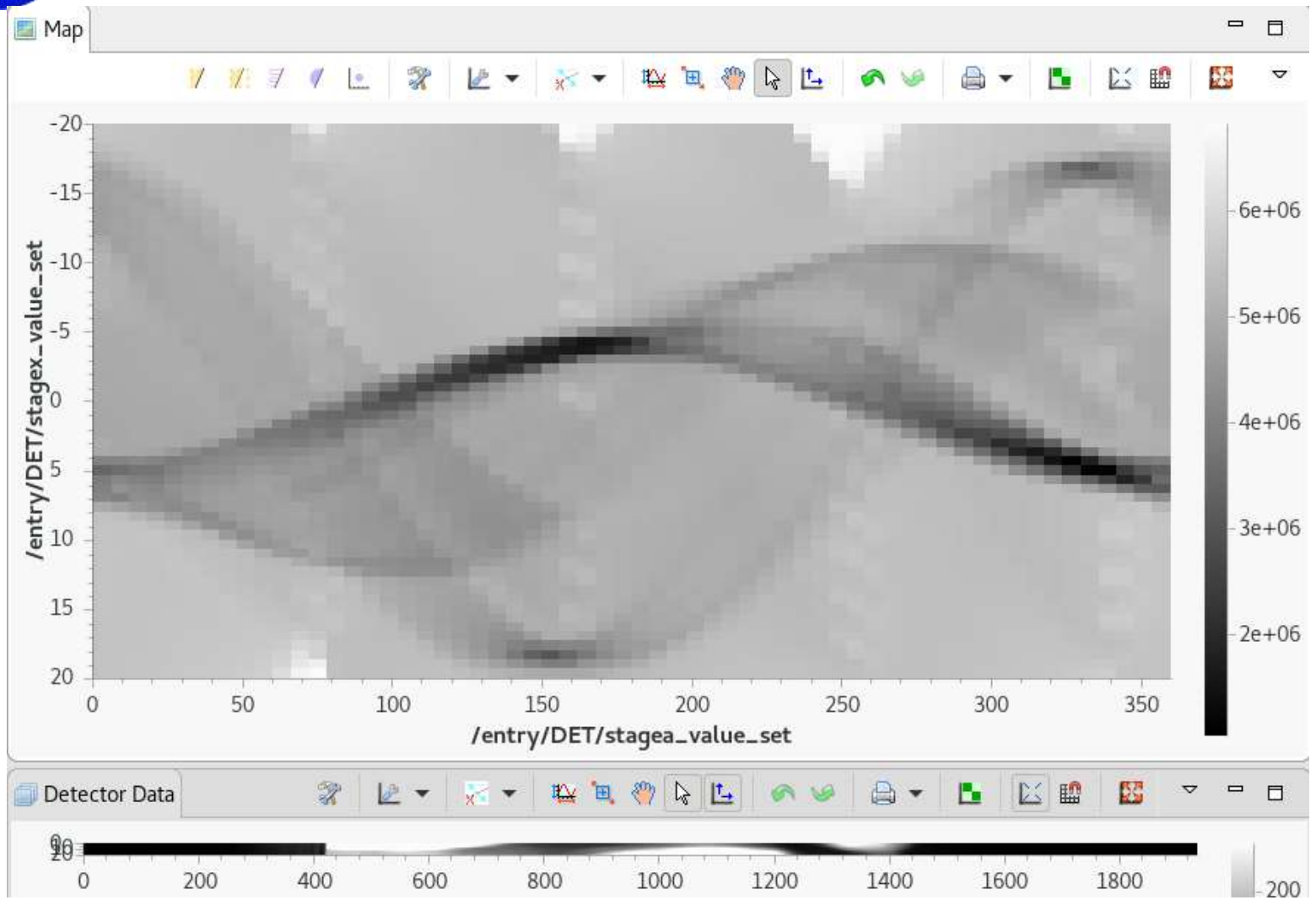
Stage 3: Check the data



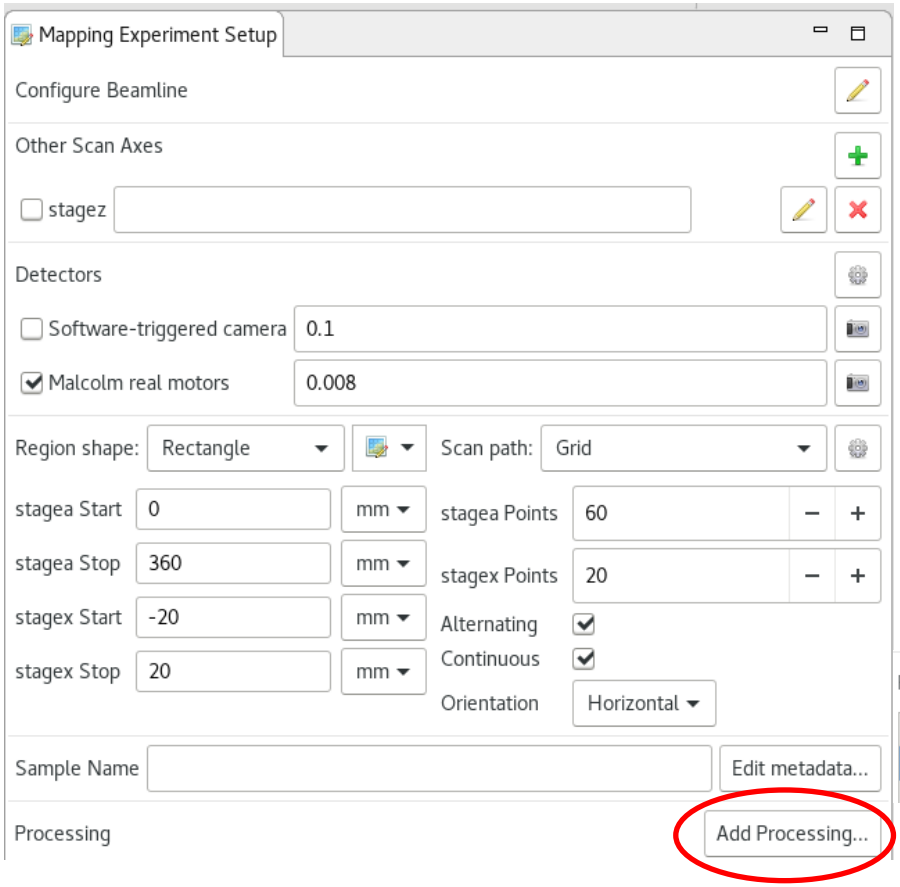
To improve the image:

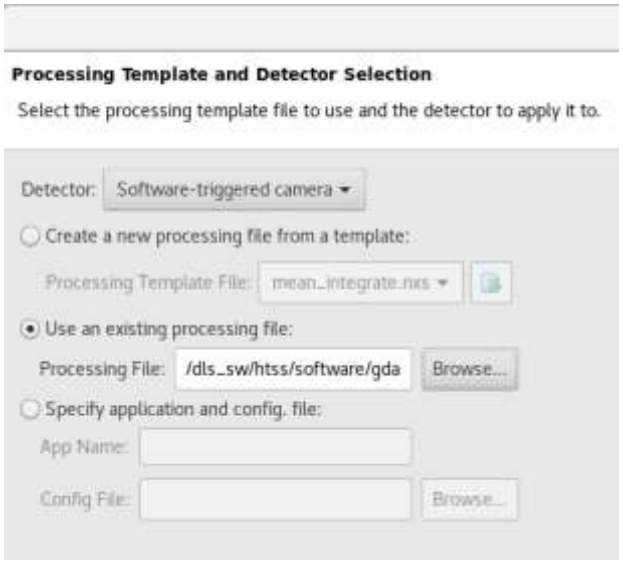
- Higher resolution (AKA more images)
  - Increase stagea points ➔ 180
  - Increase stagex points ➔ 40
- Process the image
- If you see “jaggy edges” on the sinogram, try turning off alternating

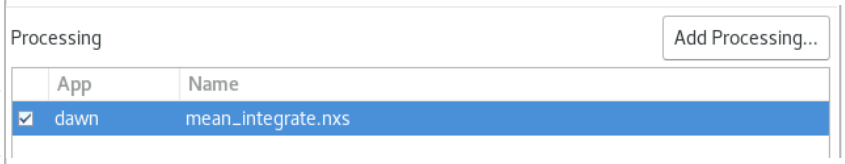
Stage 4: Check the data



Stage 5: Add processing

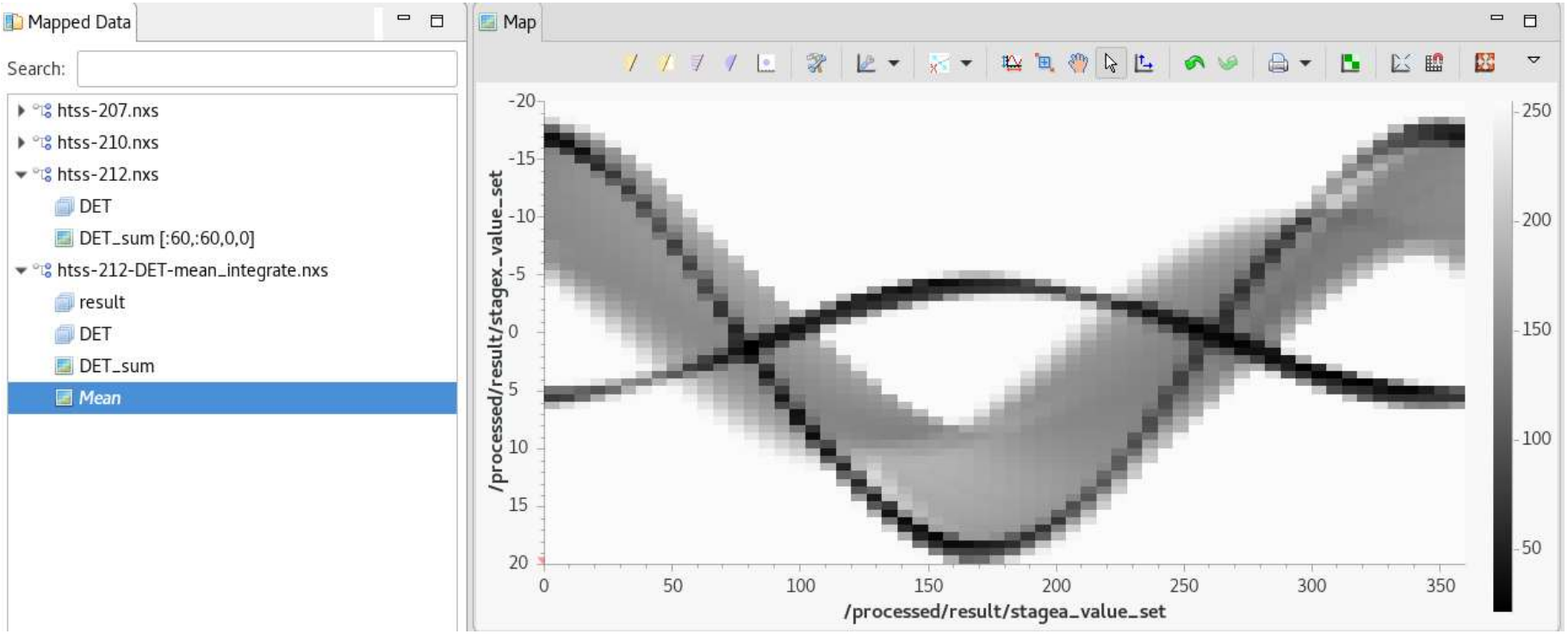




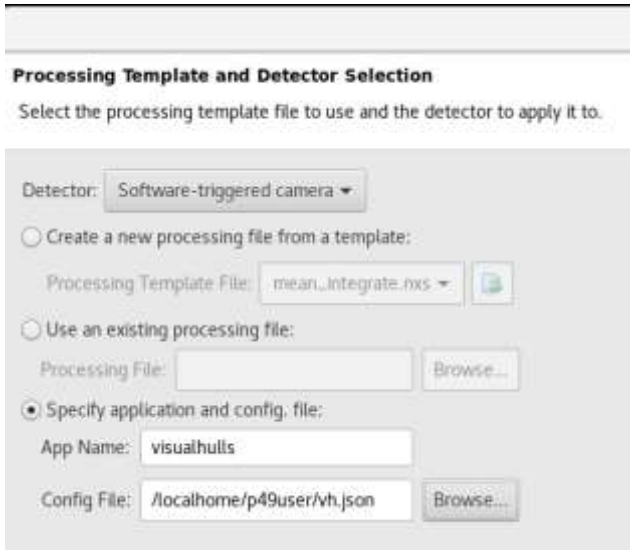


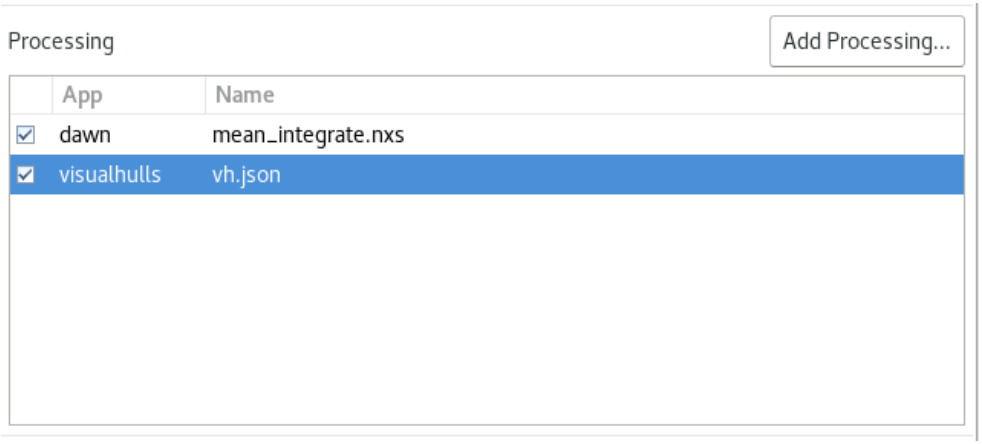
/dls_sw/htss/software/gda_var/processingTemplates/mean_integrate.nxs

Stage 6: Check the data



Stage 7: visualhulls

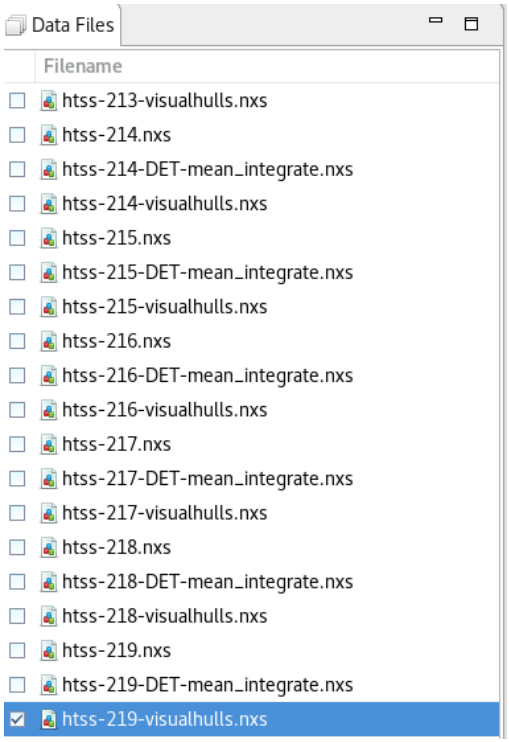


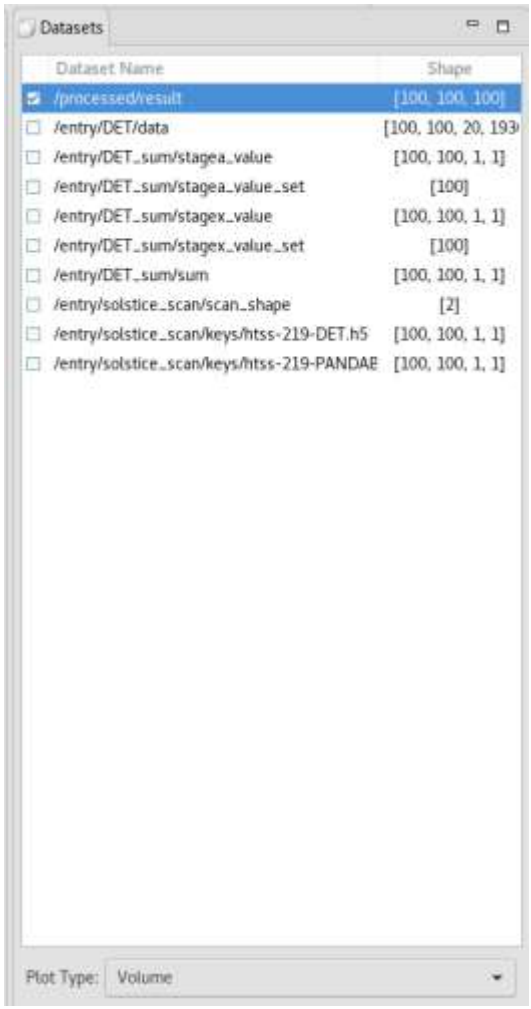


App Name must be **exactly** “visualhulls”
Config File: $HOME/vh.json

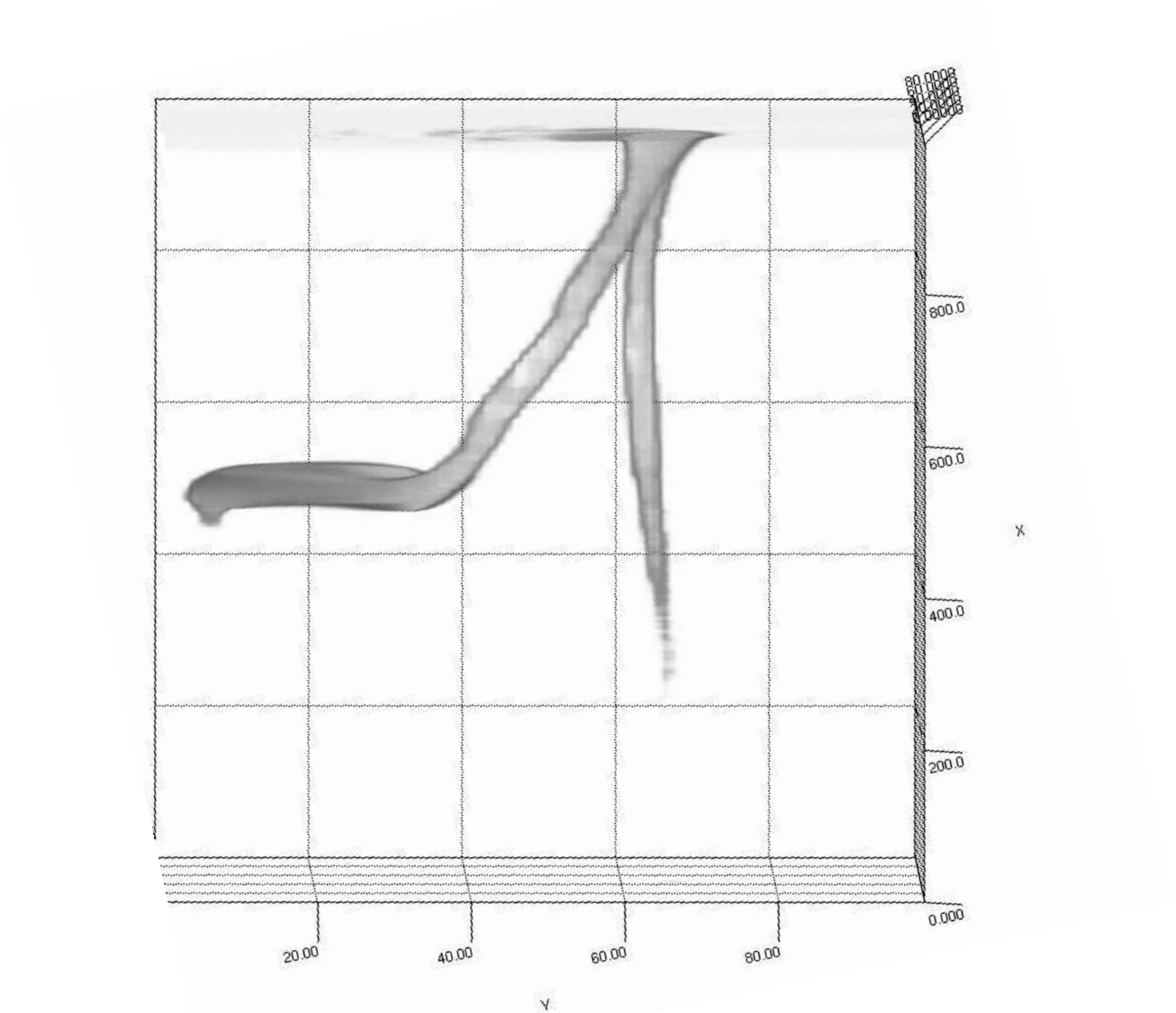
Stage 7: visualhulls

1. Right click on the visualhulls .nxs in *Mapped Data*
2. Transfer > DataVis
3. Check the visualhulls.nxs file just created in *Data Files*
4. Select /processed/result and Plot Type “Volume” in *Datasets*





Stage 8: Check the data

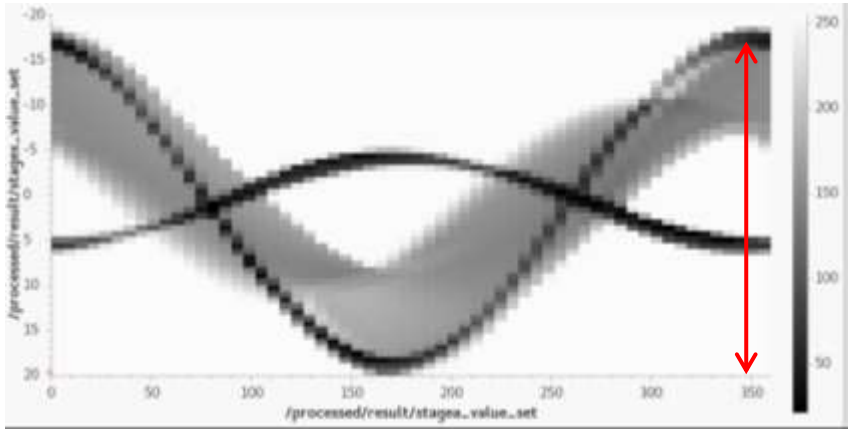


Stage 9: Improve visualhulls

Open vh.json ~/ Save

```
{
        "step": 10,
        "start": 400,
        "stop": 1390,
        "threshold": 230,
        "cor": 0.8
}
```

Tab Width: 8 Ln 1, Col 1 INS



height/2 = cor