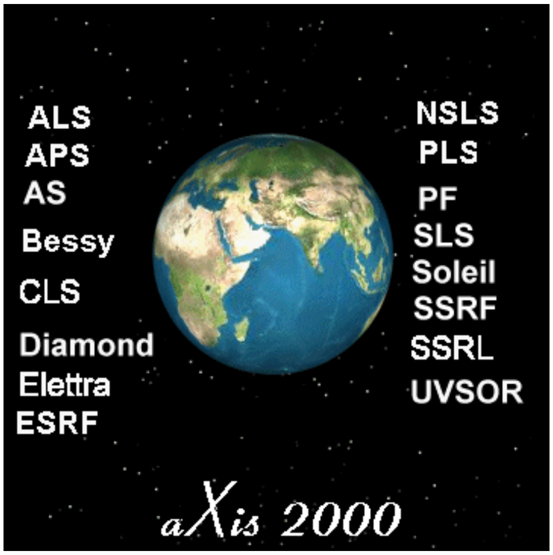
aXis 2000

<u>A</u>nalysis of <u>X</u>-ray microscopy <u>I</u>mages and <u>S</u>pectra
(20 Feb 2025)

aXis2000 - <u>A</u>nalysis of <u>X</u>-ray microscopy <u>I</u>mages and <u>S</u>pectra - is an IDL widget for viewing, comparing and processing X-ray microscopy images, spectra and spectromicroscopy (combined image & spectra data). IDL stands for Interactive Data Language, a scientific computing platform developed by Research Systems Inc (RSI), currently owned by L3Harris. aXis2000 contains scripts developed by Chris Jacobsen, Carl Zimba, Adam Hitchcock and others. The widget platform was written by Adam & Peter Hitchcock. It is maintained and frequently updated by Adam Hitchcock. It can be obtained from unicorn.mcmaster.ca/aXis2000. It operates on Windows (WIN), Unix (X) and Macintosh (MAC) versions of IDL.



A compiled version **(aXis2000.sav**) for use with IDL Virtual Machine is available. This allows access to the power of aXis2000 without needing to purchase an IDL license. Please note that there are some features of aXis2000 that work with the licensed version, but do not work with the VM version. The specific details of these problems depend on the versions of both IDL and aXis2000. I would appreciate it if you would notify me by email (aph@mcmaster.ca) about problems with the code or if you wish to make suggestions for improvements. If you make extensions or corrections to the distributed source files, or develop new IDL routines for specific data processing functionality, I would appreciate receiving a copy of your code revisions <u>with sample data</u>, so I can evaluate and incorporate in future versions. Your contribution will be acknowledged.

*I thank all the people who have written scripts that went into this, includng my son, **Peter Hitchcock** who helped set up the basic widget structure; **Carl Zimba** (Photons Unlimited) who supplied ZSTACK and extensively improved the overall package in 2000; **Eli Rotenberg, Jonathan Denlinger, Stefano Cerasari, Tolek Tyliszczak, Andy Smith, Andreas Scholl, Göran Johansson, Jacob Stewart Ornstein**, and many others. SPECIAL thanks to **Chris Jacobsen** (Stony Brook, nsls) for sharing his STACK_ANALYZE and PCA_GUI codes, **Rick Kneedler,** for providing the basis for the stack-fit routine, and **Billy Loo** for providing the SF X-ray spectral predictor based on, the Henke mass absorption coefficients, and the Conjugate Gradient Optimization routine (ax_cgo).*

TO START *aXis2000*: *after installing aXis2000* (see end of this file) on Windows, Unix or Mac OS: Start **IDL** ;

If you have set the <u>Preferences</u> (in IDL) so that **axis2000_batch.pro** is the start file, aXis2000 will launch automatically.

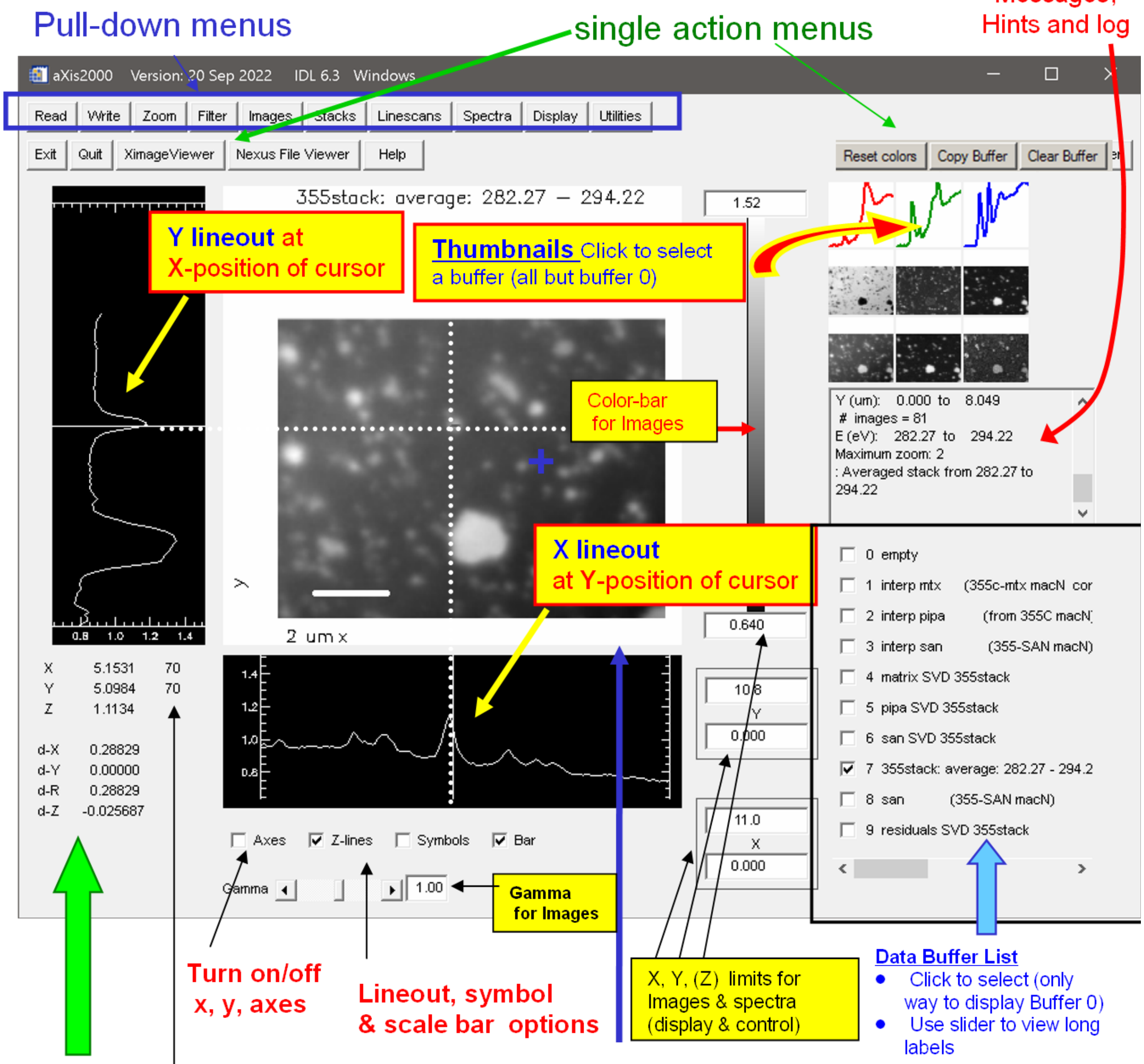
Otherwise, type **axis2000** on the IDL command line.

If you quit aXis2000 and stay in IDL, you can restart by typing **axis2000**

Feb 2025

Features of the *aXis2000* widget

aXis2000 Messages, Hints and log



Pull-down menus

single action menus

X	5.1531	70
Y	5.0984	70
Z	1.1134	
d-X	0.28829	
d-Y	0.00000	
d-R	0.28829	
d-Z	-0.025687	

```
Y (um):   0.000 to   8.049
 # images = 81
E (eV):   282.27 to   294.22
Maximum zoom: 2
: Averaged stack from 282.27 to
294.22
```

- □ 0 empty
- □ 1 interp mtx (355c-mtx macN cor
- □ 2 interp pipa (from 355C macN
- □ 3 interp san (355-SAN macN)
- □ 4 matrix SVD 355stack
- □ 5 pipa SVD 355stack
- □ 6 san SVD 355stack
- ☑ 7 355stack: average: 282.27 - 294.2
- □ 8 san (355-SAN macN)
- □ 9 residuals SVD 355stack

Y lineout at X-position of cursor

Thumbnails Click to select a buffer (all but buffer 0)

Color-bar for Images

X lineout at Y-position of cursor

Gamma for Images

Turn on/off x, y, axes

Lineout, symbol & scale bar options

X, Y, (Z) limits for Images & spectra (display & control)

Data Buffer List

- Click to select (only way to display Buffer 0)
- Use slider to view long labels

Cursors

(X,Y,Z) – at cursor pixel indices

(dX,dY,dZ) – change over line (images) or between cursors (spectra)

dR – distance along line (images only)

Main Image

- Displays currently selected image or spectrum (or group of spectra, if *display~overplot* ; or group of images, if *display~thumbnails* is used)
- Size of AXIS2000 can be adjusted from **0.5 to 2.0** of its nominal size (360x360 pixels in Main Image) by the size parameter in **axis.ini**

Mouse

- First click – cursor and lineout; arms line generator
- Second click – draws and documents line (image) ;
  - reports difference in cursors (spectra)
- Third click – clears line and cursor information

Notation conventions used in this reference manual for aXis2000

BOLD indicates a pull-down menu command

TOP LEVEL items are bold, underlined, and light blue highlighted

Utilities~print~annotated is a third level pull-down menu under **Utilities** first level button

Linescans indicates an internal bookmark or a hyperlink to a webpage.

Hard copy: Printing the Main Image is achieved by using *Utilities~print*. aXis2000 prints via the IDL Printer virtual device to the default printer on the computer it is running on.

Default parameters: The user can define a number of properties of the aXis2000 widget by editing the axis initialization file (**axis_win.ini, axis_macos.ini** or **axis_unix.ini**).

I find it convenient to document the data analysis I perform with aXis2000 by taking snapshots of the display [e.g. **Snipping Tool** in Windows10 or **MWsnap** (freeware, http://www.mirekw.com/), and transferring the snapshot via the clipboard to **Powerpoint, Keynote** or similar program. Note that clicking on any sub-component of the aXis2000 widget (or any pop-up GUIs) will copy that sub-component to the clipboard.

In Windows I use the free, open source program, paint.net, to modify snapshots from aXis2000.

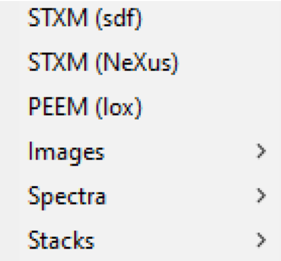
TOP ROW PULL DOWN MENUS (Click on the menu name to go to that part of the manual)

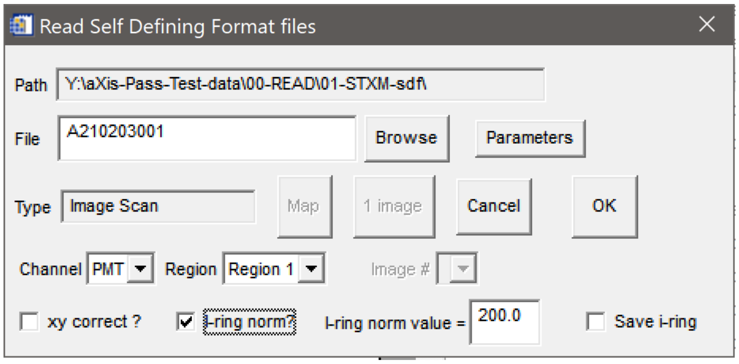
Read	Write	Zoom	Filter	Images	Stacks	Linescans	Spectra	Display	Utilities

READ

for all types of data (stacks, images, linescans, spectra, ROIs etc)

STXM (sdf) - self defining files containing all types of data from files written by **STXM_Control** (the STXM acquisition program at ALS, Bessy-Maximus, CLS_aSTXM (pre-2025), Solaris, MaxIV-SoftiMAX, UVSOR and NSRRC STXMs) are read by the following widget. (single images; multi-region images, single spectra, multiple point spectra, linescans, image sequences for one region or multiple regions; DAQ, all types of motor scans, etc)




Read Self Defining Format files

Path: Y:\aXis-Pass-Test-data\00-READ\01-STXM-sdf\

File: A210203001

Type: Image Scan

Channel: PMT

Region: Region 1

Image #:

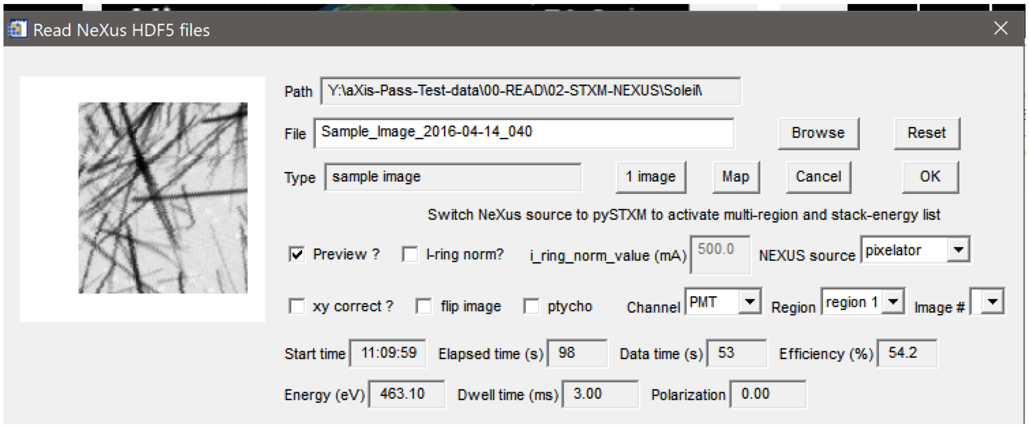
□ xy correct ?   ☑ I-ring norm?   I-ring norm value = 200.0   □ Save i-ring

The output of a STXM_control measurement is described by a self-defining header file (***.hdr**) which contains the information needed to identify the type of data contained in the associated data files (***.xim** or ***.xsp**). After selecting a file, the widget indicated above can be used to :

Browse – to select a file (you can also edit the file name– hit '**ENTER**' after changing)
Parameters - display the header, which contains microscope, beamline & source parameters
Map – generates a component map from the first 2 images of a stack [OD(2)-OD(1)]
To do this from any 2 images in a >2 image stack use *stacks~analyze~stack process*
1 image (only for stacks) – read in one image from a stack (select the energy using the **Image#** pull down list)
Cancel – dismiss the STXM (sdf) read in widget
OK - read the file, using the selected options
Channel - select detector channel from pull down list, generated from header
Region - select spatial region (area or point) from pull down list, generated from header
xy correct ? - if checked, and the data was recorded, shifts position of each pixel to that reported by the interferometer guidance system
Image# - select specific image of multi-image stack
I-ring norm? – if checked, all data intensities are adjusted as if Ring current is the I-ring norm value
I-ring norm value – Ring current: defaults CLS – 200 mA (default); ALS – 500 mA
Save Iring – save the I-ring values of a multi-energy measurement as a aXis2000 *.txt file.

Selecting **OK** then reads the information in the associated image (*.xim) or spectral (*.xsp) ascii (text) files into one or more aXis2000 buffers. Stacks are converted directly to *.ncb files

STXM (NeXus~pixelator) - for SLS, Soleil, Bessy-II Mystiic *.hdf5 datafiles



Path Y:\aXis-Pass-Test-data\00-READ\02-STXM-NEXUS\Soleil\
File Sample_Image_2016-04-14_040
Type sample image

Switch NeXus source to pySTXM to activate multi-region and stack-energy list

☑ Preview ? □ I-ring norm? i_ring_norm_value (mA) 500.0 NEXUS source pixelator

□ xy correct ? □ flip image □ ptycho Channel PMT Region region 1 Image #

Start time 11:09:59 Elapsed time (s) 98 Data time (s) 53 Efficiency (%) 54.2

Energy (eV) 463.10 Dwell time (ms) 3.00 Polarization 0.00

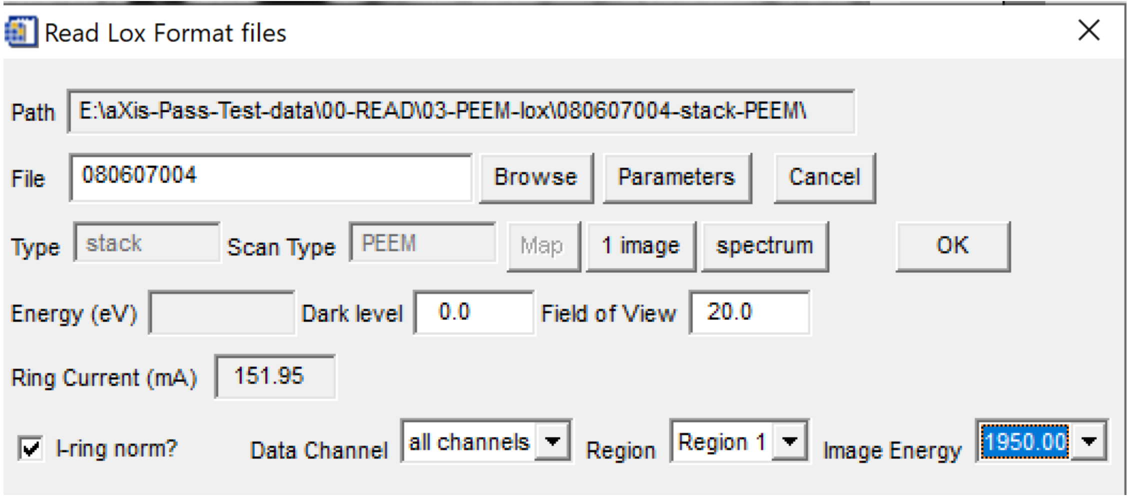
Browse - to select a file
Reset – clears Path, File, Type
image – (only for stacks) read in one image from a stack (select using **Image#)-1**
Map - generates a component map from the first 2 images of a stack [OD(2)-OD(1)]
To do this from any 2 images in a >2 image stack use *stacks~analyze~stack process*
Cancel – dismiss the STXM (NeXus~pixelator) read in widget
OK - read the file, using the selected options
Preview – if set, display the NeXus *.hdf5 data in the viewing panel
I-ring norm ? – if checked, all data intensities are adjusted as if ring current is I_ring value
I-ring norm value (mA): defaults CLS (cryo-STXM) – 200 mA; Soleil – 500 mA
NEXUS source pixelator (SLS, Bessy, Soleil, Diamond) or pySTXM (CLS cryro)
xy correct ? - if checked, and the data was recorded, shifts position of each pixel to that reported by the interferometer guidance system (experimental)
flip image – [x➔x, y➔-y] required for some versions of pySTXM image files
ptycho - read in STXM type images generated when measuring in ptychography mode
Channel - select detector channel from pull down list, generated from header
Region - select spatial region (area or point) from pull down list, generated from header
Image # - select specific image of multi-image stack (*toggle NEXUS source to display*)
Start time - clock time at start of acquisition
Elapsed time - time of acquisition
Data time – time predicted from dwell and line delays, assuming no other delays
Efficiency – ratio of data / elapsed time
Energy (eV) – photon energy
Dwell time (ms) – dwell time
Polarization – parameter for photon polarization (converted to LCP, RCP, Lin-H, lin-V (and displayed on y-axis label)

STXM (pySTXM2.5) – as for **Read~STXM~pixelator**, with parameters etc set for files written by the pySTXM2.5 version used for CLS cryo-STXM data acquisition

READ (*continued*)

return to top menu

PEEM (Lox) – Reads data written by **Lox**, the Visual Basic data acquisition for the CLS PEEM (called CaPeRS = **Ca**nadian **P**hoto**e**mission **R**esearch **S**pectromicroscope).



Read Lox Format files

Path E:\aXis-Pass-Test-data\00-READ\03-PEEM-lox\080607004-stack-PEEM\

File 080607004

Type stack   Scan Type PEEM

Energy (eV)   Dark level 0.0   Field of View 20.0

Ring Current (mA) 151.95

☑ I-ring norm?   Data Channel all channels   Region Region 1   Image Energy 1950.00

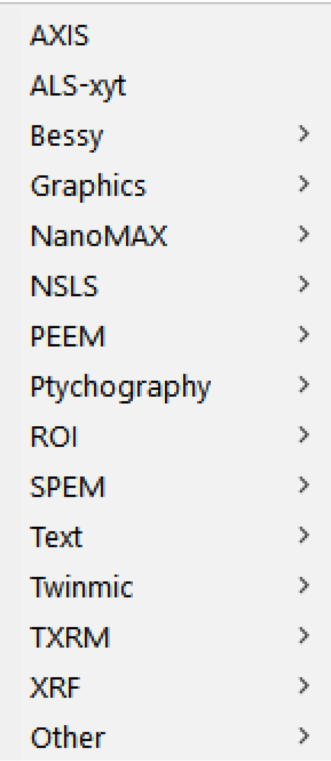
Lox data is described by a header file (*.lox) which contains the information needed to identify the type of data contained in the associated data files, along with all data acquisition and PEEM parameters. After selecting a file via its associated *.lox file, the Read Lox widget can be used to :

Browse – to select the file (or edit the filename box)
Parameters - list the header, which contains all microscope parameters
Cancel – exit the Read~PEEM~Lox GUI
Map – (only for a stack) – convert a 2-imagestack into an OD difference map
1 image – (only for NEXAFS Image Scan) read only the image identified in the image energy pull down list (which displays the photon energy)
OK – read in data file as described by selected parameters
Energy (eV) – energy of Lox image measurement (read from file)
Dark Level – set camera dark level
Field of View - set field of view of the image (microns)
I-ring norm? – if checked, all data intensities are adjusted as if Ring current is 500 mA
Ring Current (mA) (read from file)
Data Channel - select from pull down list, generated from header
Region - select from pull down list, generated from header
Image Energy (eV) - list of energies of all the images in a stack - select specific image

Selecting **OK** then reads the information in the associated binary image (*.tif) or ascii spectral (*.lox) files into one or more aXis2000 buffers. Image sequences are converted directly to the aXis2000 binary format (*.ncb).

return to top menu

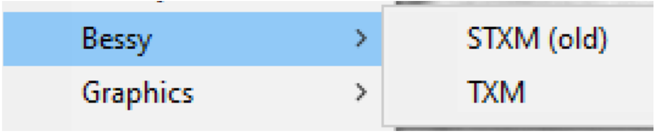
Images



AXIS – images written in aXis2000 **binary** format (*.axb)
This is useful for saving derived results. It is the default output format, and the required input for some routines

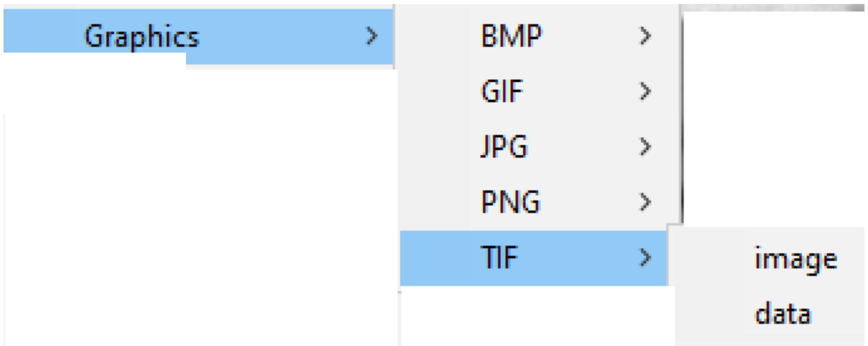
ALS-xyt – file format for pattern generator (STXM)

Bessy – read 1 image or a stack from files of old Bessy-I STXM

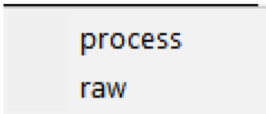


Graphics - standard formats (**BMP**, **GIF**, **JPG**, **PNG**, **TIF**).
TIF format handles 1-channel & 3-channel (RBG) formats
For each image format one can read the image as
image (un-modifiable, preserves colors)
or as **data** (modifiable, pixel indices as (x,y)-values)

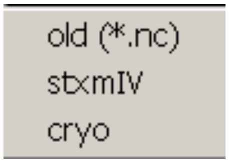
CAUTION: there are known bugs in these routines, with some inversions and strange color distortions



NanoMAX – read images recorded in hdf5 format at the MaxIV NanoMAX beamline
process - *.h5 file format read in
raw - not written as of Feb2025

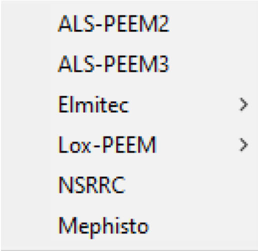


NSLS – read images written by NSLS I STXMs (pre 2012 or so)
old (*.nc) – prior to ~2002 (NetCDF)
stxmIV – last used data format
cryo - format from cryo-STXM



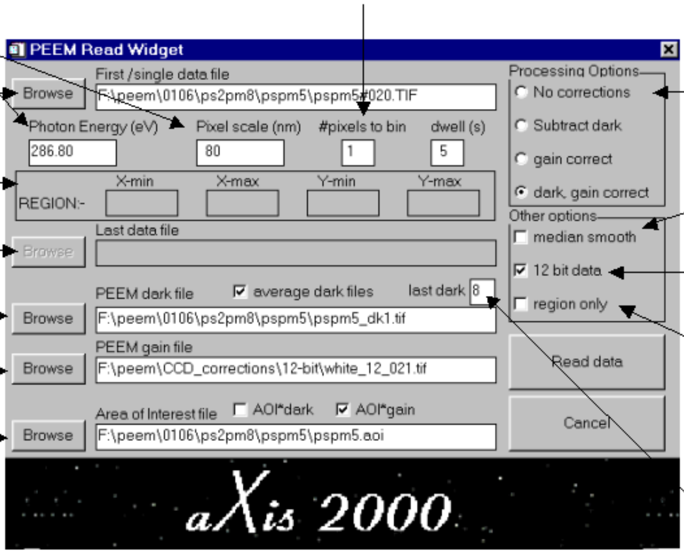
READ~Images *(continued)*

return to top menu



PEEM

ALS-PEEM2 – reads in 12-bit and 16-bit ALS-specific formats with a **widget.** (NB the **Elmitec** read command uses the same structure to define image and image sequence parameters)
NB – *this routine may not work in the Virtual Machine (*.sav) version*


PEEM Read Widget

First /single data file
Browse | F:\peem\0106\ps2pm8\pspm5\pspm5#020.TIF

Photon Energy (eV): 286.80
Pixel scale (nm): 80
#pixels to bin: 1
dwell (s): 5

REGION:- X-min X-max Y-min Y-max

Last data file
Browse

PEEM dark file ☑ average dark files last dark 8
Browse | F:\peem\0106\ps2pm8\pspm5\pspm5_dk1.tif

PEEM gain file
Browse | F:\peem\CCD_corrections\12-bit\white_12_021.tif

Area of Interest file ☐ AOI*dark ☑ AOI*gain
Browse | F:\peem\0106\ps2pm8\pspm5\pspm5.aoi

Processing Options
○ No corrections
○ Subtract dark
○ gain correct
● dark, gain correct

Other options
☐ median smooth
☑ 12 bit data
☐ region only

Bin image after processing

Energy, scale

Select data file

Define region in μm units

Select last data file
hen used to convert stacks

Select dark file

Select gain file
(12-bit 'white')

file to define area of interest (AOI) for gain or dark signals
(if these are only available as full images)

All possible read-in combinations

median smooth after processing (remove hot pixels)

All tif files must be either 12-bit (if clicked on) or 16-bit (if clicked off)

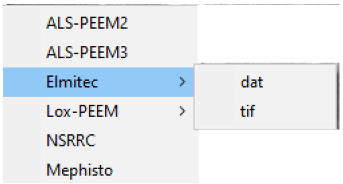
REGION
Select portion of image after processing

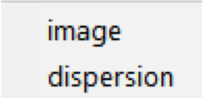
Average multiple dark files

ALS-PEEM3 – reads data format for ALS PEEM3 (implemented Jan-2007, Andreas Scholl)

Elmitec - read PEEM files.

- **dat** – read in 16-bit (12-bit encoded) files written by UView2002
- **tif** - read=-in 8-bit (3-channel grayscale) exported by UView2002





Lox-PEEM – reads Lox format images
image - reads in *.lox images
dispersion - not written as of Feb2025

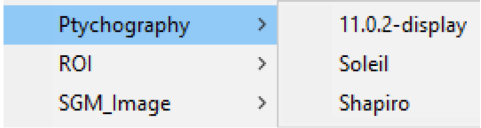
NSRRC – PEEM data written by Taiwan TLS PEEM

Mephisto – read PEEM data from Mephisto (SRC - old binary 512x512 from a CCD)

READ~Images (*continued*)

return to top menu

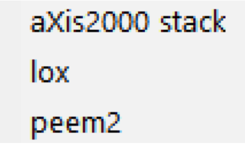
Ptychography



11.0.2 display – animation of ptycho data from 2011-2012.
Soleil tif file generated by Soleil ptycho acquisition (2019 version) – superseded by *.hdf5 file
Shapiro - output of Nov-2012 ptychography image analysis

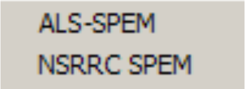
ROI – region of interest definitions for

aXis2000 stack – ROI for stack region selection (CJJ, CZ)
Lox – ROI for Lox format PEEM
Pem2 – ROI (=AOI) for peem2 data

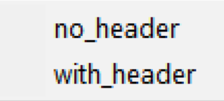


SPEM

ALS-SPEM – read in images recorded in XPS or NEXAFS mode using the ALS SPEM
NSRRC SPEM read in images recorded with Taiwan PEEM

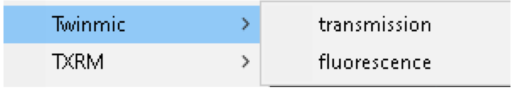


Text - read in images written as an array of Ascii (ie readable with a text editor like Notepad) , with or without standard (5-line) aXis2000 header



Twinmic - read in images from Twinmic (Elettra) STXM

Transmission
Fluorescence



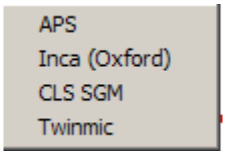
TXRM – read in images written by X-radia software

Alba.Mistral
Xradia [STXM at Shanghai - stacks]

NB this only works on a Windows system & needs special codes (IDL-Java bridge) which are provided in the aXis2000 source package. *Contact Adam Hitchcock if you need instructions on how to set that up*

XRF read in X-ray fluorescence (XRF) maps from different systems

APS – from GoeCARS XRF microprobe
Inca (Oxford) – from McMaster SEM (Tescan) running Xmax SDS
CLS SGM – from Amptek (spectral mode as an array)
Twinmic – XGLabs data from Twinmic

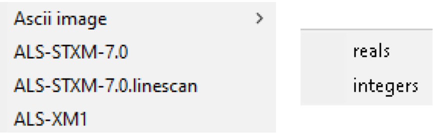


OTHER - mostly OLD formats

Ascii image (Reals or Integers)
ALS-STXM-7.0 image files (ascii) – data channels:

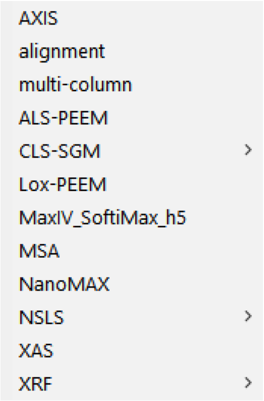
- im0 = OSA signal (Io);
- im1 = transmitted signal;
- im2 = other (e.g TEY, luminescence, etc)

ALS-STXM-7.0 linescan – linescan files from ALS-STXM7.0.1
ALS-XM1 – data files from full field microscope at the ALS (rudimentary)



READ~Spectra

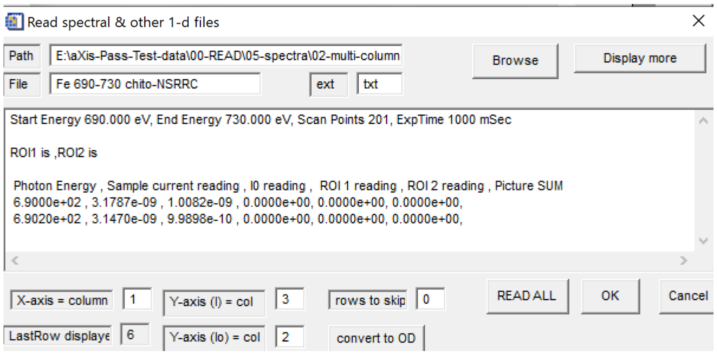
return to top menu



AXIS – read spectra from ascii format files **(*.txt)**
This can be ascii 1-d data written by aXis2000
or many generic formats of multi-column data
with or without headers. (default ext. = ***.txt**)

alignment – alignment (x,y – shifts) from Zimba, or Jacobsen aligners

multi-column – guided read-in of any multi-column, header ascii data file, with ability to skip lines, choose columns, multi-column read-in, etc



Path: E:\aXis-Pass-Test-data\00-READ\05-spectra\02-multi-column
File: Fe 690-730 chito-NSRRC
ext: txt

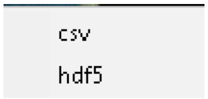
```
Start Energy 690.000 eV, End Energy 730.000 eV, Scan Points 201, ExpTime 1000 mSec

ROI1 is ,ROI2 is

 Photon Energy , Sample current reading , I0 reading ,  ROI 1 reading , ROI 2 reading , Picture SUM
 6.9000e+02 , 3.1787e-09 , 1.0082e-09 , 0.0000e+00, 0.0000e+00, 0.0000e+00,
 6.9020e+02 , 3.1470e-09 , 9.9898e-10 , 0.0000e+00, 0.0000e+00, 0.0000e+00,
```

X-axis = column: 1
Y-axis (I) = col: 3
rows to skip: 0
LastRow displaye: 6
Y-axis (Io) = col: 2

Browse – select file
Display more – add another 6 lines to the display buffer (to get to end of headers)
X-axis column – select data column for x-axis
Y-axis column – select data column for y-axis
Rows to skip – use LastRow displayed to identify which line to start read in
Read all – read all columns in a multicolumn file (x, many y)
OK – read only the identified (x,y) columns
Cancel – abort widget
Y-axis(Io) – column for Io intensities
Convert to OD - use 2 columns to generate OD as ln(Io/I)

ALS-PEEM – spectra from ascii file written by ALS PEEM 2

CLS-SGM



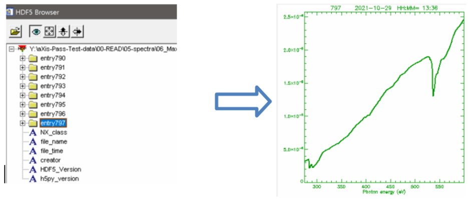
csv – comma separated variables format files
written from web-based CLS-SGM converter software
[https://sgmdata.lightsource.ca/users/]

hdf5 – hierarchical data format version 5 – SGM TEY/ transmission and XRF-yield data as stored by the acquisition software. NB this routine only reads the first entry (as of May 2022).

Lox-PEEM – read in multi-column spectra written by Lox (*.lox files)

return to top menu

MaxIV_SoftiMax_h5 – read in spectra or beamline monitoring data files written in hdf5 format recorded by SoftiMAX beamline software e.g. 20211029.h5 (click on an entry# of the multi-entry hdf5 file)

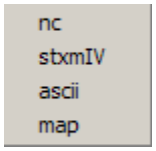


MSA - read in Digital Micrograph spectra (TEM-EELS)

NanoMAX – read in *.h5 spectra from MaxIV NanoMAX beamline

NSLS – read spectral data files in formats used at X1A

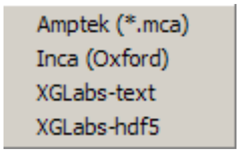
- **nc** - as-recorded (BSIF binary)
- **stxmIV** – stxm IV format (nc, but different than old-stxm)
- **ascii** - converted spectral format (generated by an X1A nsls utility)
- **map** - mapper style files (e.g. stack alignment shifts)



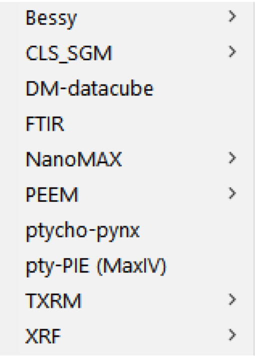
XAS – (**X**-ray **A**bsorption **S**pectra) format with user settable header set up for bibliography (used for reference spectra, ZSTACK, NSLS, PCA_GUI and COREX bibliography) single spectrum or multiple spectra written to a common energy sampling (x-axis)

XRF

- **Amptek (*.mca)** – XRF spectra written from the Amptek software
- **Inca (Oxford)** – XRF spectra written from Inca (SEM, TEM)
- **XGLabs (*.dta)** – XRF spectra written from XGLabs software
- **XGLabs-hdf5** – XRF spectra written in hdf5 format from XGLabs software



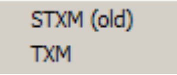
Read~Stacks



Bessy

STXM (old) read stacks from Thieme / Gottingen scanning TXM

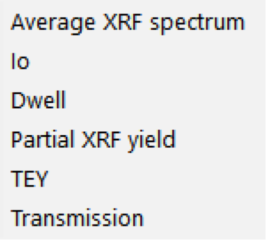
TXM read stacks from cryo-TXM



CLS-SGM

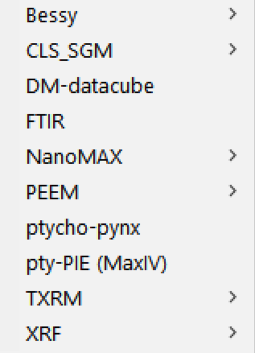
extract stacks from *.hdf5 files written by CLS SGM

Average XRF spectrum
Io
Dwell
Partial XRF yield
TEY
Transmission



return to top menu

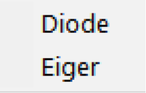
Read~Stacks (*continued*)



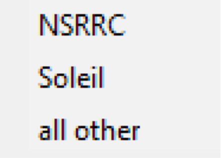
DM-datacube – converts binary data exported from Digital Micrograph image sequences into a aXis2000 *.ncb stack

FTIR – converts Nicolet Thermo (version 7) map files (sets of spectra in spatial array) to *.ncb stack (for Soleil FTIR)

NanoMAX read stacks from diode (STXM) or Eiger (ptycho)
Diode – integrating STXM detector
Eiger - Dectris ptychography camera



PEEM
NSRRC - PEEM stacks from Taiwan
Soleil – NeXus format stacks written from Soleil PEEM
only versions of IDL above 7.0 can read Soleil NeXus (.nxs) files*
All other – converts PEEM stack data from various instruments (Lox, Sphinx, ALS-PEEM2, ALS-PEEM3, ALS-pre-Sep02 (peem2 old format), Other).
see BELOW: STACKS~convert format ~ PEEM for view of widget and full description)

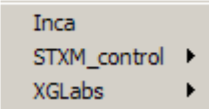


Pytcho – PYNX –(Soleil) reads in selected type of tif files (object or probe, amplitude or phase) from sets of folders (1 / energy) OR all-from-1-folder

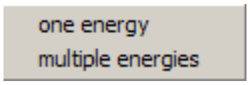
Pty-PIE (MaxIV) - reads in multiple reconstructions as sequence of *.txt files (spectro-ptycho at MaxIV SoftiMax; under development as of May 2023)

TXRM – stacks from full-field microscopes at Alba and SSRF
Alba.Mistral
Xradia (SSRF, Shanghai)

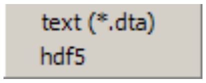
XRF – read in stacks of X-ray fluorescence images
Inca – converts X-ray fluorescence (XRF) maps written from the Bruker Inca raw export into a *.ncb stack



STXM control
one energy - converts XRF-maps measured at a single photon energy using STXM_control to a *.ncb stack
multiple energies - converts XRF-maps measured at multiple photon energies using STXM_control (stack-of-stacks) to a *.ncb stack with summing over user selected X-ray fluorescence energy range to make an XRF-yield stack

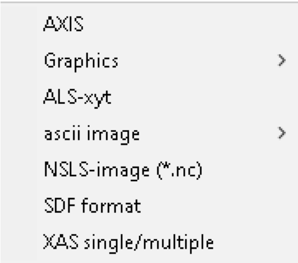


XGLabs – converts XRF-maps written by the Labview software controlling the Eletrra Twinmic STXM to a *.ncb stack starting from **text (*.dta)** or **hdf5** data files



WRITE store results (images, spectra, etc) for later use or transfer to other programs.

return to top menu



AXIS - automatically writes file type depending on buffer contents

– spectra written with header in ascii **(*.txt)**

- images written with header in Z-matrix binary format using IDL system-independent binary coding **(*.axb)**

Graphics

GIF – writes current Main Image (with any annotation etc) as a GIF file
- **image** (exactly as seen on main screen of aXis2000)
- **data** (the image without axes or labels)

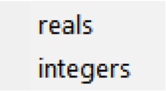
JPG – writes current Main Image (with any annotation etc) as a JPG file
- **image** (exactly as seen on main screen of aXis 2000)
- **data** (the image without axes or labels)

PNG – writes current Main Image (with any annotation etc) as a PNG file
- **image** (exactly as seen on main screen of aXis 2000)
- **data** (the image without axes or labels)

TIF - writes current Main Image (with any annotation etc) to a TIF file.
- **image** (exactly as seen on main screen of aXis 2000)
- **data** (the image without axes or labels)

ALS-xyt – file with point-by-point (x,y,t) values to create patterns using **pattern_generation** function of STXM_control. NB after reading in a graphics imge (tif, jpg etc), the *Images~Set XY Scale Utilities~change mesh* and *Spectra_gain* functions can be used to adjust size, point spacing and intensities (0-1) in the *.xyt file this routine writes. At the STXM, the **pattern_generation** routine can then adjust exposure time through a multiplier.

ascii image – write image tas an array of **Real** or **Integers** for export to other programs (e.g. excel)



NSLS-image (*.nc) – writes NSLS netCDF image format **(*.nc)**
- used in the Jacobsen *stack_analyze* package

SDF format – write self defining file format [*.hdr, *.xim (*.xsp) files for images (spectra)]

XAS single/multiple - writes one or more spectra which are displayed in the main viewing window of axis2000 in XAS format with optional definition of detailed header. This is a useful way to transfer multi-spectral data for figures in a scientific plotting program like Sigma Plot, Origin etc. The default energy scale spacing is the finest spacing in the set of files plotted. If this is very small the resulting files can be very large – set to a value relevant to the energy resolution (e.g. 0.1 eV).

return to top menu

ZOOM (for images and spectra)

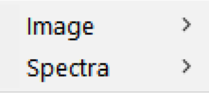
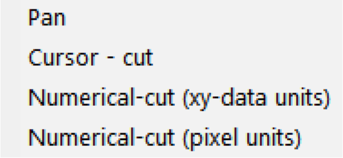


Image *(NB –since expanding the scale is performed by cutting out the region of interest, the cursor-cut and numerical-cut commands can also be used to extract sub images)*

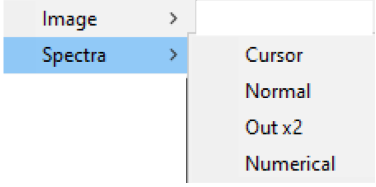


Pan – pops-up a zoom window with a ~3x expansion of the region around the cursor. Move the cursor on the main image to look at different areas. Left click to change zoom factor. Right clock to end zoom. If the **Z-lineouts** option button at the bottom of the aXis2000 screen is OFF, then the zoomed image is only updated on each left mouse click.

Cursor – cut - stretchable box cursor used to define region. Data is cut from the displayed image and shifted to the working buffer (0).

Numerical-cut (xy-data units) - numerical selection of range in xy-data units. Data is cut from the displayed image and shifted to the working buffer (0). The last values used to define the limits are saved and used the next time *zoom~image_numerical-cut (xy-data units)* is used, which allows accurate extraction of regions from many images (e.g. a set of component maps from fitting)

Numerical – cut (pixel units) - numerical selection of range in pixel units. Data is cut from the displayed image and shifted to the working buffer (0). The last values used to define the limits are saved and used the next time *zoom~image_numerical-cut (pixel units)* is used, for accurate extraction of regions from many images (e.g. a set of component maps from fitting)



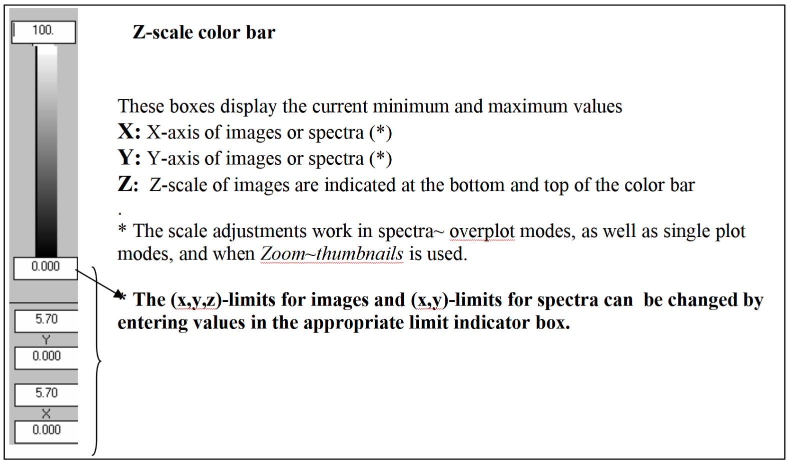
Spectra

Cursor -Stretchable box cursor

Normal - return plot to full X-scale, same Y-scale

Out x2 - useful to "make white space" for comparison overplotting (also can use z-limits)

Numerical - select numerical limits – as with *zoom~image_numerical*, the limits are preserved



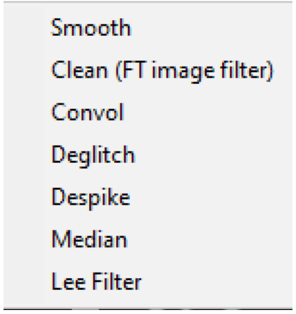
Z-scale color bar

These boxes display the current minimum and maximum values
X: X-axis of images or spectra (*)
Y: Y-axis of images or spectra (*)
Z: Z-scale of images are indicated at the bottom and top of the color bar
.
* The scale adjustments work in spectra~ overplot modes, as well as single plot modes, and when *Zoom~thumbnails* is used.

*** The (x,y,z)-limits for images and (x,y)-limits for spectra can be changed by entering values in the appropriate limit indicator box.**

return to top menu

FILTER (for images and spectra)

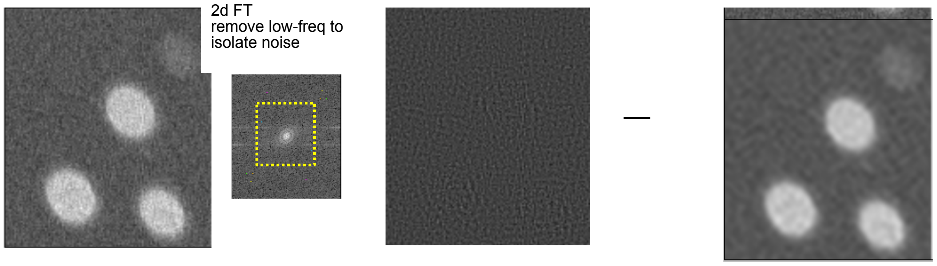


Five types of smoothing with selectable parameters. **Clean (FT image filter)** works only with images. The other four apply to BOTH images and spectra.

Smooth – Savitsky-Golay average over n-points (right to edge of images), n is odd, if n=even selected, uses n+1

Clean (FT image filter) - 2d Fourier transform (FT) filter. The 2d-FT is displayed on a 1:1 pixel format. Use the rubber-band style cursor (click, drag mouse, click a second time) to define the data in the complementary frequency domain to delete. The reverse transform of all data but the rejected data is displayed in buffer 9. The centre of the FT image corresponds to 0 frequency (dc), while positions farther from the centre correspond to higher frequency. Periodic (moiré) noise associated with aliasing (beating) of a systematic noise signal with STXM raster scanning can be cleanly removed by deleting strong signals in the FT.

The **Clean (2d-FT Image filter)** can be used with subtraction of the filtered result to perform high-pass or selective band-pass filtering. For example -



Difference gives smoothed signal; can isolate frequency components in x,y differently to remove systematic noise associated with scanning etc.

Convol - convolutes with user-definable digital filter (definable frequency response)

Deglitch - -graphics-driven deglitch routine. Cursors identify lower and upper bounds of data.

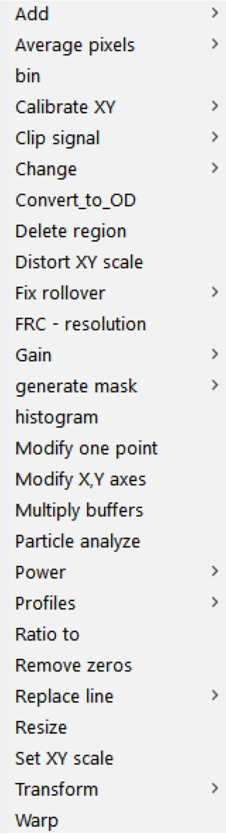
Despike – local filter to remove point (spike) artifacts. User selects width of a median filter to identify the spikes. (NB the "**Despike stack**" routine in *Stacks~Analyze~***Jacobsen's stack_analyze** works better)

Median – median averaging, except at edges

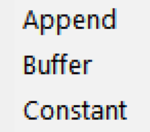
Lee Filter – Lee filtering smooths image noise by generating statistics in a local neighborhood and comparing the result to the expected values

return to top menu

IMAGES (processing of images)



Add

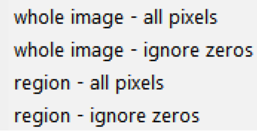


- Append - append 2 images, matched using (x,y) scales
- images can be **tessellated** using append.

– **Buffer** – weighted addition of two buffers
(use a negative weight to subtract buffers)

- Constant - Add a constant to each pixel of an image
(*use a negative constant for subtraction*)

Average pixels - compute average intensity in a user-defined region



whole image – all pixels - full image

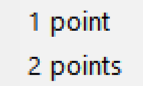
whole image - ignore zeros - same, but does not include zeros

region- all pixels - select pixel region of interest; report average Z-value with statistics

region- ignore zeros - as for all pixels, but does not include zero values. This is useful if an image has been multiplied by a masked image and the areas of interest are not contiguous. Use to count pixels and thereby determine areas and average intensities of selected regions of an image.

bin – reduce image pixelation by binning. i.e. convert (nxn) pixels to 1 pixel using Sum or Average e.g. 3 converts (3x3) regions pixels to 1 pixel

Calibrate XY - calibrate X,Y scales of images



– **1-point** – shifts current image to make selected point have user-defined (x,y)
– **2 points -** allows linear stretching as well as shifting

NB: The calibration routines are similar to those used in *Stacks~Image alignment.* They can be used to manually process the first and last images of a sequence in order to ensure that the x,y limits selected will be present in all files.
Note that once the parameters have been entered to calibrate ONE image, these are saved, so it is easy to apply the same calibration to a number of images, as a manual alignment process. For example, if a few-image stack is measured one energy at a time, **Calibrate XY** *i*s an easy way to generate roughly aligned images. One can then (i) write out the xy-calibrated images, (ii) make a stack list *(*.sl)* file using a text editor, like NotePad, (iii) use *stacks~Analyse~stack list input* to assemble the set of xy-calibrated images into a stack.(iv) *Stacks~analyze~Jacobsen stack analyze* or *Stacks~analyze~Zimbacan then be used* to further align routines.

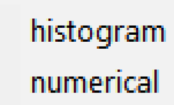
Example of a **stack list file** (*.sl): [1st line is path to data; successive lines are the file names]

```
Y:\aXis-Pass-Test-data\05-STACKS\00-ANALYZE\01_STACK_LIST_INPUT\
12_055.axb
12_056.axb
12_057.axb
```

Images (*continued*)

return to top menu

Clip signal

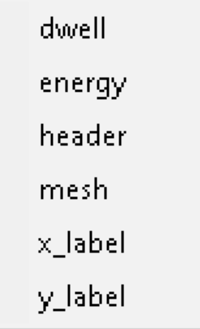


replace all values outside user-selected limits with a fixed value (default is the average of the in-bounds data) . select the lower & upper limits by

histogram – use cursors on a plot of a histogram of z-values

numerical – use text input

Change



dwell – change dwell per pixel (ms)

energy – change photon energy (eV) or equivalent parameter

header – modify header (this is what will be saved to disk)

mesh - redefine numbers of pixels (images) or points (spectra). This is useful to match pixelation of different mages. Many routines in aXis2000 can modify pixel count by 1 or 2 pixels. *Change~mesh* can be used to make the (x,y) pixilation of each image of a set of images the same

x_label – change X- label (scale bar, energy, dwell)

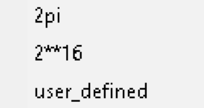
y_label – change Y-label (polarization, etc)

Convert_to_OD - compute OD (optical density) representation of a transmission image with user defined Io value. The default Io is the maximum value pixel in the image (typically this is too large – should use *image~average pixels~region- all pixels* in a hole area of the image to get a more precise OD)

Delete region – use a box cursor to define a region to replace with a user selected value. (this is useful to remove glitches or to remove part of a masked region).

Distort XY scale – distort (x,y) scaling. Pixels are interpolated to square in new co-ordinates

Fix rollover - for some image files (e.g. ALS PEEM) it is possible to have signed or unsigned formats which leads to 'roll-over' for pixels with above 16 K counts (e.g. if an unsigned integer data set is read using a signed integer). This replaces all pixels with a -ve value with 65,535 plus that value. Similarly, in phase outputs from ptychography reconstructions, many reconstruction programs automatically 'fold back' phase values such they lie between –π and +π



2pi – replace points after increase of 2pi by previous value -3.14159
- replace points after decrease of 2pi by previous value +3.14159

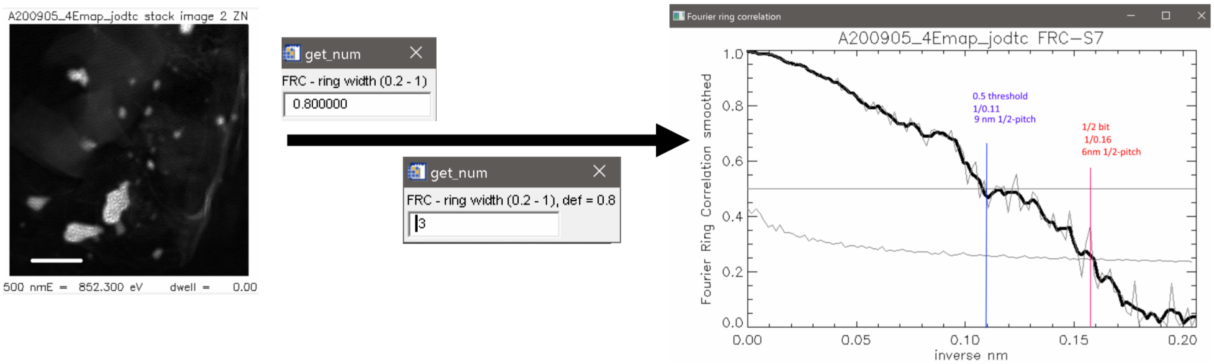
216** replaces all pixels with a -ve value with 65,535 plus that value

user_defined – user specifies the'rollover' threshold criteria and the value to increment (if pixel value goes –ve across threshold) or decrement (if value goes +ve across threshold)

Images (*continued*)

return to top menu

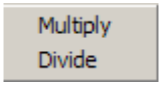
FRC – resolution – apply Fourier Ring Correlation algorithm to a square image to estimate spatial resolution. The inverse of the frequency where the FRC curve crosses 0.5 or the ½-bit line are an estimate the ½- pitch spatial resolution. For the equivalent full width at half maximum (FWHM) multiply by 2.



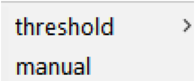
FRC - ring width (0.2 - 1): 0.800000

FRC - ring width (0.2 - 1), def = 0.8: 3

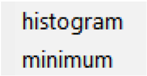
Gain - **Multiply** - multiply the Z-values by a user-supplied constant
Divide - divide the Z-values by a user-supplied constant



generate mask – generates a **0/1 map** (1 - selected pixels; 0 = not-selected pixels) and writes a ROI file. These maps be used in conjunction with *Multiply buffers* to select portions of images or to write a region_of_interest file (***.roi**) to allow extraction of the spectrum of that roi using *stacks~analyze~stack_analyze*



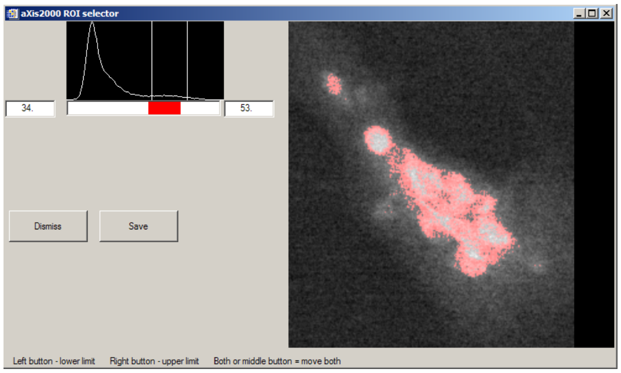
threshold - histogram
pop-up GUI which allows adjustment of lower and upper limits while displaying the selected (red overlay) regions change dynamically



threshold –minimum user inputs the lower limit only (NB must be >0) – default is 0.5

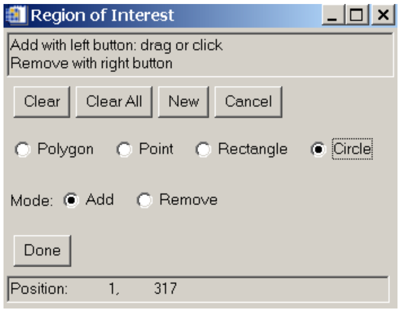
manual – use region definition tool to select a particular region for stack area selection.
NB this ROI selector is also used in e.g. *images~delete region* and *stacks~analyze~Zimba*

threshold region selector



Left button - lower limit   Right button - upper limit   Both or middle button = move both

manual region selector



Add with left button: drag or click
Remove with right button

Position: 1, 317

Images (*continued*)

return to top menu

histogram – generate and display the histogram of all the pixels in an image

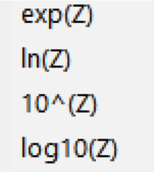
Modify one point – allows modification of a pixel (NB use **Delete region** for area modification)
Left click to add, right click to delete; in each case prompt is a local average

Modify x,y axes - applies a scale factor to the numerical axes of an image
(useful if there is a known distortion of an image scanning system)

Multiply buffers – take product of the current buffer and a second, user selected buffer

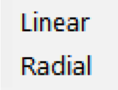
Particle analyze - when applied to a masked image (0/1 pixel values only) it analyzes the areas of contiguous regions and reports a distribution of diameters, assuming circular areas.

Power



exp(Z) – exponential of the z values of the displayed image
ln(Z) – natural log of z-values
(can be used to convert Transmittance to Absorbance scales
- **10^(Z)** – generate power of 10 of the z-values of the displayed image
- log10(Z) – generate base-10 log of z-values

Profiles - generate and save intensity profiles from images

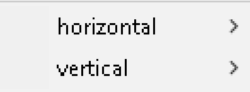


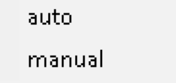
Linear – intensity along line defined by two user selected points
Radial – intensity as a function of angle in a circular region. The user defines a diameter which is then rotated about its center. Optionally, the resulting 'unfolded' radial distribution can be symmetrized by auto-seeking a threshold level and aligning at that common level. This routine was written to explore radial distributions in particles. It is also useful for analysis of azimuthal orientation effects probed by linear dichroism.

Ratio to - computes ratio of the displayed image to that in a user-identified buffer with optional scaling. Images are matched by their (x,y) scales.

Remove zeros – replaces zero value pixels with a local average.
(used to de-speckles images with dropouts)

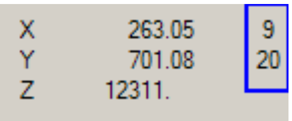
Replace line – for each direction (H, V) replace the current line with that of another line, or average of two lines. Used to correct for missing (all z-values are 0) or x-shifted lines. If auto is selected the routine identifies bad lines as those containing a zero,.





Auto automatically identifies 'bad lines' (ones with pixels that deviate beyond expected statistical fluctuation (3 sigma) & replaces them with average of adjacent line(s).

Manual - use cursor to identify horizontal lines to remove. The selected line will be replaced by the average of 2 adjacent lines (default) or by any selected line, as the user chooses. The default selectedline is the (n+1) line (the one above the selected line). Cursor line value is indicated during the **replace line** selection. This is useful for images which are acquired line-at-a-time which can fail on the triggering of a line resulting in a line with only '0' values. Numerical selection of the line to replace allows access to un-displayed lines in large images. Use the **pixel indicator** to determine the line number



Images (*continued*)

return to top menu

Resize - set size of image in terms of # of pixels explicitly (this is needed to correct rounding errors in re-pixelation steps (e.g. shifts, truncates etc) whenever 2 or more images have to have identical numbers of (x,y) pixels for subsequent processing (same as *utilities~change mesh*)

Set XY scale - calibrate distance scale by defining distance between two user-selected points

Transform – modify the displayed image by:

Flip – invert image vertically through a horizontal centre line
Mirror – invert image horizontally through a vertical centre line
Rotate - rotate image about a user defined point, by user defined angle.
This can be used in conjunction with I*mages~Distort X,Y scale* to remove image distortion by symmetrizing an object of known shape (e.g. circular particles).

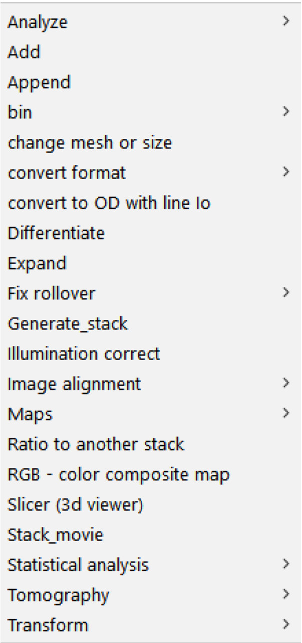
Warp –uses multiple matching point "morphing" capability of IDL. Intended to handle cases where a 2-point or 1-point manual shift will not compensate for changes in the sample shape in a stack (the 'warping' usually occurs because of radiation damage).

---

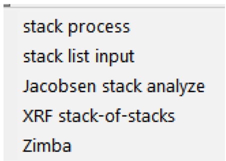
return to top menu

STACKS - manipulation of image sequences (stacks).

Note, the ability to simultaneously manipulate multi-dimensional data sets (3,4,5 & more) is one of the strengths of aXis2000, combining aspects of both image and spectral manipulation.



Analyze



Stack process

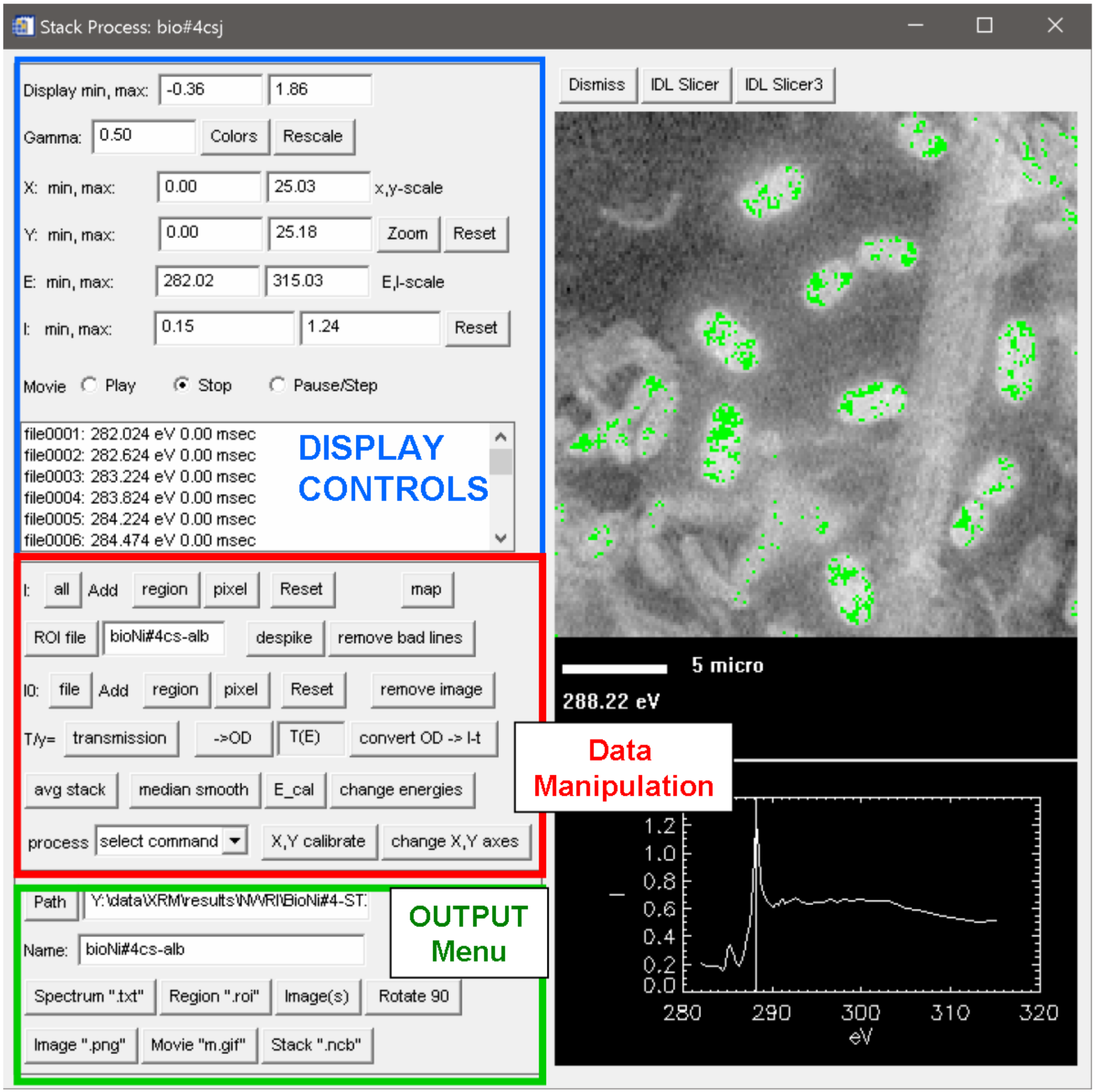
loads an AXIS binary stack format file (*.ncb) into the **stack_process** widget. Note that it is convenient to store image sequence data in this binary representation since it is much more compact (by three times or more) than original data formats. Since it is possible to extract images from the binary file and since the IDL cross platform binary formatting is used, no information is lost.

The user is promoted for an alignment file (*.aln) to incorporate alignment shifts into the stack data) and a **widget size scale factor** (before the following widget appears. The *.aln files are written by the Zimba_align or stack_align (early Jacobsen) procedures

The default **widget size scale factor** will make the stack_process widget fill about ½ of the computer screen. For very large stacks, one can use a fractional **widget size scale factor**. This will display and allow visualization. HOWEVER, when the scale factor is < 1, many of the data modification actions of *Stack_process* do not work.

STACKS~*stack process* (continued)

return to top menu



Stack Process: bio#4csj

Display min, max: -0.36 1.86

Gamma: 0.50

X: min, max: 0.00 25.03 x,y-scale

Y: min, max: 0.00 25.18

E: min, max: 282.02 315.03 E,I-scale

I: min, max: 0.15 1.24

Movie ○ Play ● Stop ○ Pause/Step

file0001: 282.024 eV 0.00 msec
file0002: 282.624 eV 0.00 msec
file0003: 283.224 eV 0.00 msec
file0004: 283.824 eV 0.00 msec
file0005: 284.224 eV 0.00 msec
file0006: 284.474 eV 0.00 msec

ROI file bioNi#4cs-alb

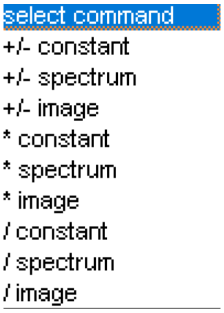
process select command

Path Y:\data\XRM\results\NWRI\BioNi#4-ST:

Name: bioNi#4cs-alb

Notes

1. click on the displayed spectrum to select image at that energy (or click in the image list)
2. process ‘select command’ pull down menu contains many full-stack actions:



select command
+/- constant
+/- spectrum
+/- image
* constant
* spectrum
* image
/ constant
/ spectrum
/ image

return to top menu

DISPLAY CONTROLS

Display min, max – Z-axis limits
Gamma – gamma for Z-axis
Colors – select from palette of standard IDL color tables
(*select* B-W linear *& the ROI will be colored green for I, and red for Io*)
Rescale – [ON] min/max each image. [OFF] – use the display min, max settings
X: min, max – use to set the x-scale of image (for precise region selection)
Y: min, max – use to set the y-scale of image
Zoom – graphical expansion of image **Reset** restore full stack (x,y)
E: min, max – use to set the energy range of the spectral display
I: min, max – use to set the intensity range of the spectral display
Reset – reset (E, I) scales to full scale values
Movie – **play** – play the move
– **stop** – stop playing movie
– **pause/step** – pause, or advance one frame

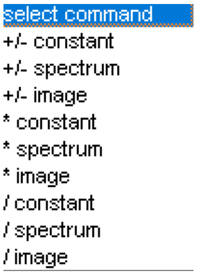
Image parameter list – select image by clicking. Use arrow keys or mouse wheel to scroll up/down

DATA MANIPULATION CONTROLS

I: all - display average spectrum of all pixels (treated as I in Beer's Law or TEY)
Add **region** - display average spectrum of an I region selected by polygon draw selector
pixel - display average spectrum of a selected pixel
Reset – reset selection of I pixels
map – use 2 (A-B) or 3 (A-(B+C)/2) selected images to take make a chemical map
ROI file – click to browse & select a *.roi (region of interest) file
despike - replaces zero value pixels with a local average.
remove bad lines – replace lines ~ auto applied to all images in a stack
Io: file - browse to select spectrum as Io (treated as Io in Beer's Law or TEY)
Add **region** - display average spectrum of an I region selected by polygon draw selector
pixel - display average spectrum of a selected pixel as Io
Reset – reset selection of Io pixels
remove image – remove currently displayed image from the stack
T/y = click displayed mode to switch between **transmission** & **yield** (electron, XRF etc) modes
→ **OD** – convert Z-values from transmission to optical density using currently defined Io
T(E) toggle between absorption (-ln(I/Io) and YIELD type TEY (I/Io) normalization
(the current status is displayed between the →OD and **convert OD → I-t** boxes)
convert OD → I-t – convert absorbance to transmittance using user-supplied Io
avg. stack – compute average of all images **<u>within the DISPLAYED E-min/ E-max range</u>**
& store as a file and in a selected aXis2000 buffer
median smooth – apply 3-point median smooth to all images
E_cal – calibrate the energy scale (linear shift by user selected amount). Must save stack
change energies – exchange current energies of stack for those in a selected spectrum file
process **select command** – modify stack (see next page for description of commands)
X,Y calibrate – define width of stack area (aspect ratio is preserved)
change X,Y axes – replace present (x,y) values with those from an external file

STACKS~stack process (*continued*)

return to top menu



process **select command** :

+/- constant – subtract a constant from z-values of each image

+/- spectrum – subtract a spectrum (*.txt) from spectrum at each pixel

+/- image – subtract image from file (*.axb) from each image in the stack

*** constant** – multiply each image by a constant

*** spectrum** – multiply stack by a spectrum (*.txt)

*** image** –multiply each image in stack by an image (*.axb) (use for masking)

/ constant – divide each image by a constant

/ spectrum – divide stack by a spectrum (*.txt)

/ image –divide stack by an image (*.axb) (use for masking)

OUTPUT MENU NB until name is identified, the commands are inactive. Type name & hit 'return'

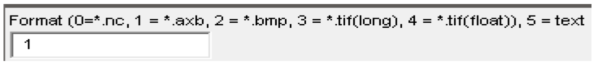
Path – defines the path for input / output files (default is folder that contained the stack)

Name – filename to be used for output

Spectrum ".txt" – write displayed spectrum

Region ".roi" – write currently selected pixel set to file (multiple regions are supported)

Image(s) – write displayed image OR all the images to a variety of formats


Format (0=*.nc, 1 = *.axb, 2 = *.bmp, 3 = *.tif(long), 4 = *.tif(float)), 5 = text
1

Rotate 90 – write out stack after rotating the stack clockwise or counterclockwise by 90°

Image ".png" – write current display (image and spectrum) to a *.png format file

Movie "m.gif" – write images or {images & spectra"} as a movie

The stack_analyze routines generate a multi-gif file which is 'clunky' and the *.mgif file written does not play due to licensing issues. The **AVIMaker** from Platypus software (http://www.c-point.com/) can be used to make *.avi files from the set of *.gif files written by the Movie command. Alternatively a ***.gif movie** ca be created using the Easy Gif Animator (https://www.easygifanimator.net/). The individual gif images are required as input to the Easy Gif Animator so answer 'N' when asked, *"delete all *.gif files?"Hint*: while the routine usually stores the images in the default folder (the one containing the stack), it sometimes ends up in a different folder, typically the source code folder (C:\aXis2000), or the (n-1) active folder. Good luck hunting for the *.gif files !!!

Stack ".ncb" – write the stack data **within the displayed E-range** as a *.ncb stack.

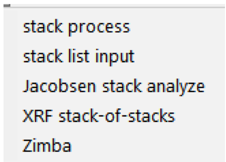
Thus to extract a subset of the energies of a stack, modify the E-range by setting E-min/E-max in the 'Display Menu". The spatial region stored is selected by defining the column / row pixel limits (default is all area of the stack). The user is asked to define the lower and upper column and then row indices for the saved data, thus removing unwanted parts of a stack area, such as the edges of a stack which might be distorted by e.g. the alignment procedure. The *.ncb binary file generated is needed for the **stack-fit** and **stack-SVD** routines. The save stack ".ncb" routine saves whatever is currently displayed in **stack process** over the energy range defined by the **E min, max** values in the display area.

Hint: In order to use curve fitting routines with reference spectra(see **Stacks~Maps**) it is necessary to normalize to Io and convert to OD scale (transmission measurements) or divide by an appropriate Io spectrum for yield data and then save the Io-normalized stack.

return to top menu

COMMANDS AT THE TOP RIGHT OF THE **STACK PROCESS** WIDGET

Dismiss – exit **stack process** widget
IDL slicer – start widget for 3d display of stack (early version of the slicer widget)
IDL slicer3 – advanced 3d manipulation of stack using latest IDL Slicer widget



==============================

Stacks~Analyse (continued)

stack list input

- generate a stack from a stack_list file (*.sl) which has the format:

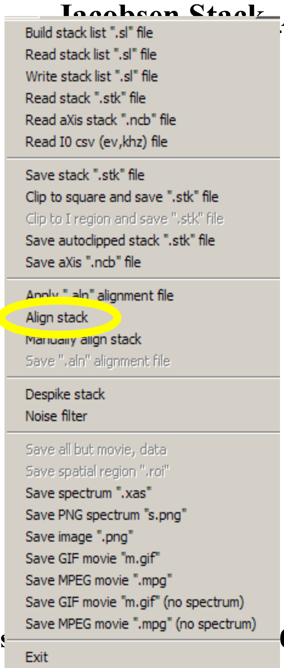
```
Y:\aXis-Pass-Test-data\05-STACKS\00-ANALYZE\01_STACK_LIST_INPUT\
12_055.axb
12_056.axb
. . . .  (each image listed in energy sequence
```

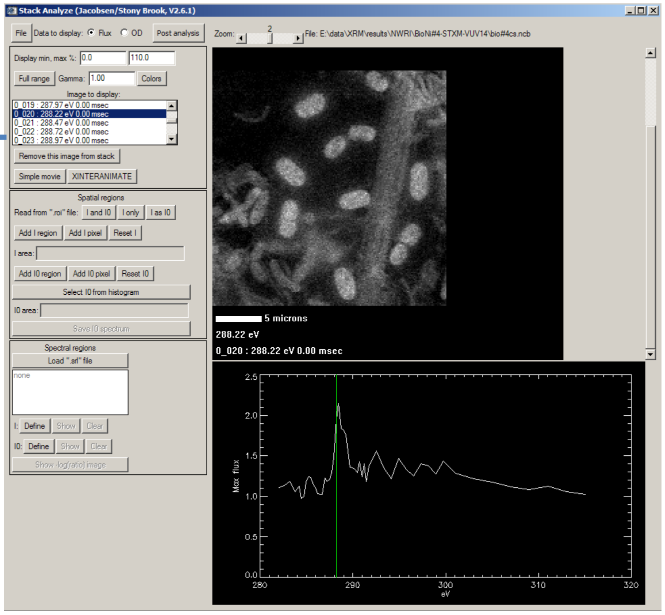
Then the **stack_process** widget (described above) is started. This command is used to assemble a stack from individual images or to re-assemble a stack, which has been written out as separate files, and has, for example, some modified images. Remember to save the re-assembled stack !!

Jacobsen stack analyse – starts the Jacobsen stack_analyse widget, which is a **complete spectromicroscopy data processing package**. The version accessed by aXis2000 is from Sep 2008. More recent versions may exist. See the documentation for that widget, from Chris Jacobsen's web site. The capabilities of this package are extraordinary. *The alignment routine is exceptional good and easy to use.* Apply it many times until the (x,y) deviations are less than 0.1 pixel. After alignment, trim to the aligned area using **Save autoclipped stack '.stk' file**)and then **Save axis '.ncb. file**. Note: the autoclip routine usually keeps an area which includes alignment artifacts in the 1 or 2 columns at left, and 1 or 2 rows at top. These can be removed when the stack is written from *Stack Process*.

Menu under **'File'**

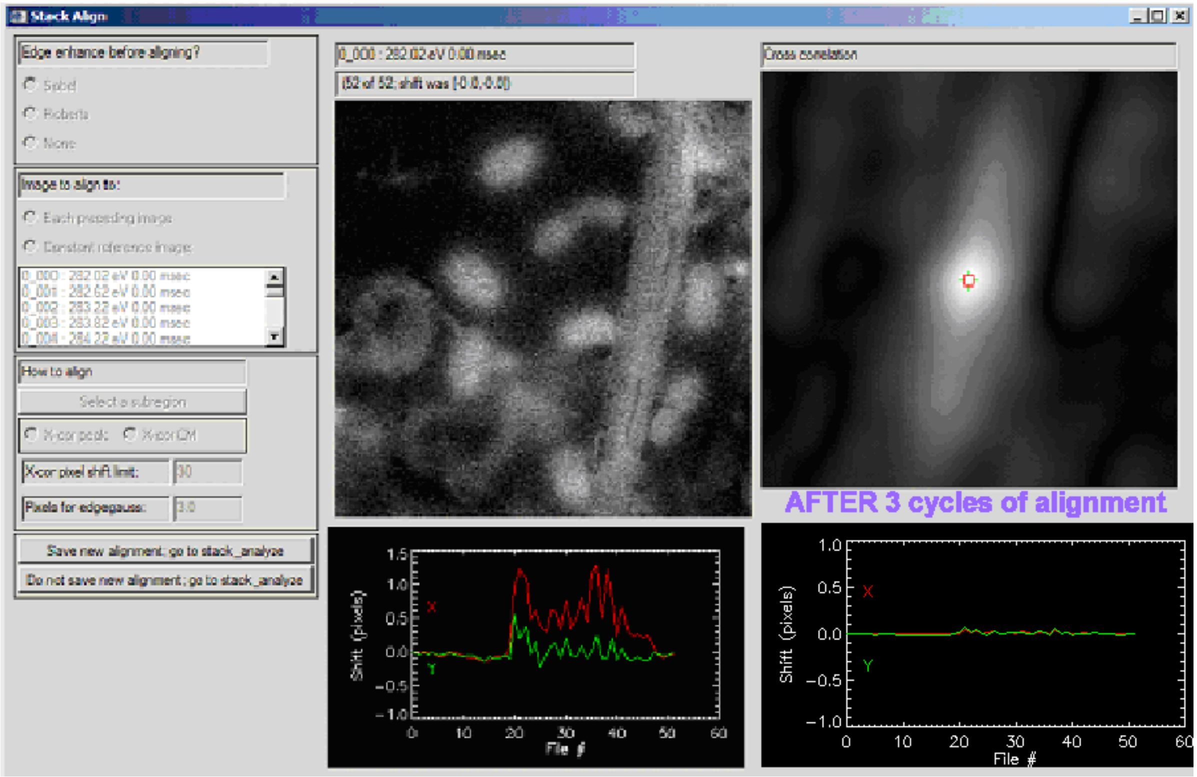
Jacobsen Stack–Analyze


Build stack list ".sl" file
Read stack list ".sl" file
Write stack list ".sl" file
Read stack ".stk" file
Read aXis stack ".ncb" file
Read I0 csv (ev,khz) file
Save stack ".stk" file
Clip to square and save ".stk" file
Clip to I region and save ".stk" file
Save autoclipped stack ".stk" file
Save aXis ".ncb" file
Apply ".aln" alignment file
Align stack
Manually align stack
Save ".aln" alignment file
Despike stack
Noise filter
Save all but movie, data
Save spatial region ".roi"
Save spectrum ".xas"
Save PNG spectrum "s.png"
Save image ".png"
Save GIF movie "m.gif"
Save MPEG movie ".mpg"
Save GIF movie "m.gif" (no spectrum)
Save MPEG movie ".mpg" (no spectrum)
Exit



STACKS Analyze ~ Jacobsen stack analyse (*continued*)

return to top menu



Stack Align

0_000 : 282.02 eV 0.00 msec

(52 of 52; shift was [-0.0,-0.0])

0_000 : 282.02 eV 0.00 msec
0_001 : 282.62 eV 0.00 msec
0_002 : 283.22 eV 0.00 msec
0_003 : 283.82 eV 0.00 msec
0_004 : 284.22 eV 0.00 msec

X-cor pixel shift limit: 30

Pixels for edgegauss: 3.0

AFTER 3 cycles of alignment

XRF stack-of-stacks – allows extraction of one multi-incident-photon-energy stack from an array of XRF or polarization stacks recorded at a set of incident photon energies. (Ultimately to be evolved into a true 4D viewer and processor widget – ***anyone want to help*** ?)

Zimba

Carl Zimba's version of the **Jacobsen stack analyze** code. It has a very nice alignment which is able to align stacks for which the Jacobsen alignment routine fails.

Hint: For unknown reasons, when called from aXis2000 the Zimba routines often lock up if complex combinations of commands are used. In such cases the full functionality of the Zimba routines ca be accessed by running *zstack.pro* outside of aXis2000 (if you have an IDL license) in which case you can access all the many features for display and storage, including saving a movie of a stack with display of the spectra from multiple regions.

Some features of the Zimba stack analyze package:

- Reliable alignment (which differs from the routine in Jacobsen’s stack analyze).
- Many different formats for Io data are supported (*.csv, *.txt, *.xas)
- Simultaneous display of original and aligned data
- Excellent manual alignment (ztune)
- Multiple spectral regions defined
- Provision to write full, aligned stack to a single file (*.ncb, *.stk)
- Provision to write image sequences as MPEGs, or a set of image file

STACKS **Analyze ~Zimba** *(continued)*

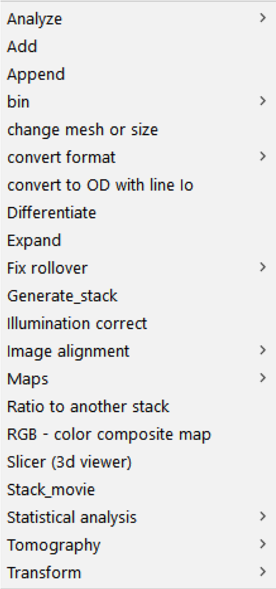
return to top menu

Note: The Zimba *.sl and *.aln file formats differ from those written by the *Jacobsen stack analyse*. It is not generally possible to 'mix-and-match' the two sets without using a text editor for conversion. In particular, it is essential to delete all letter and non-numerical symbols from the lines listing the shifts in the *.aln file written by Zimba, before using it in the read-in procedure for stack_analyze. The ZSTACK codes can be run independently from AXIS.

Please see separate documentation for ZSTACK supplied by Carl Zimba included at the end of this manual.

==

STACKS (*continued*)

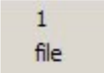


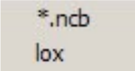
Add – add two stacks. This is useful to take differences to remove dominant contributions to better display minority signals by using negative weighting. Use **Generate_stack** to make a stack of a fitted component from the component reference spectrum and the component map.

Append – combine 2 stacks along the energy axis. (The stacks must have the same pixel dimensions !). The images in the appended stack are ordered according to the photon energy (or equivalent variable)

bin   **-** bins (2-2x2 to1 pixel, 3 = 3,3 to 1 pixel, etc) image data for better statistics. The first character of the original file name is converted to a **b** to indicate a bin processing.

– **energy** - bin the data along the energy axis
– **NSLS - 1** – bin 1 user-selected *.nc file
– **file** – bin all files listed in the supplied file (typically *.sl files).
- stack   ***.ncb** – bin all images in a stack
*. **lox** – bin all images in a Lox stack (PEEM from CLS)



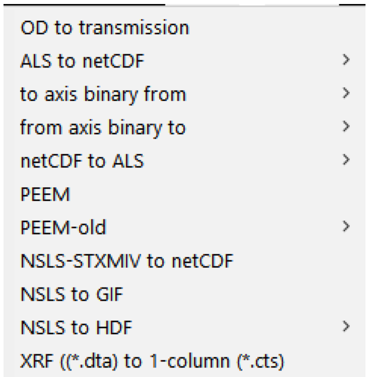


change mesh or size – adjust pixilation or the real-space dimensions

convert to OD with line Io – generate OD version of a transmission stack for the case where there is a *set of columns all in an Io region*, which can then be used to generate an Io on a line-by-line basis (normally a single Io value for each stack image/energy is used in normalization). This routine is useful if the stack images are acquired so slowly that there is significant change during acquisition of each image (e.g. a drop in the ring current fro decay mode light sources).

return to top menu

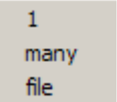
convert format



For most conversions, the user selects the first file then the last file in a sequence; intervening files are converted assuming a standard file name convention (e..g last 3 positions before '.' are an index number).

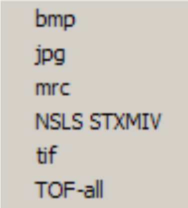
- OD to transmission – convert to transmission using "white" Io (1 at each energy)

- **ALS to netCDF** – converts (old) ALS images to NSLS netCDF files, with E-scale shifts and optional binning (adds squares of 2x2 or 3x3 etc pixels). Generates ***.sl** file for input to *Jacobsen stack_analyze* or *Zimba*
  - **1** converts one file
  - **many** converts multiple files selected by clicking on first then last
  - **file** converts files listed in the user-defined file (e.g. *.lst from ALS)



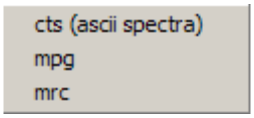
- to axis binary from

- **bmp**
- **jpg**
- **mrc** (format used for tomography in, e.g. IMOD)
- **NSLS STXMIV**
- **tif**
- **TOF-all** (time-of-flight PEPICO)



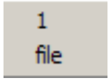
- from axis binary to

- **cts (ascii spectra)**
- **mpg**
- **mrc**



- netCDF to ALS - converts NSLS netCDF format files to ALS format files

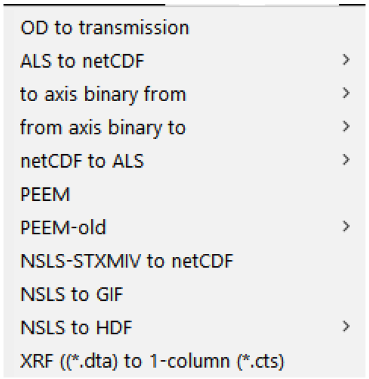
- **1** converts one file selected by pickfile
- **file** converts files listed in the user-defined file (e.g. *.lst from ALS)



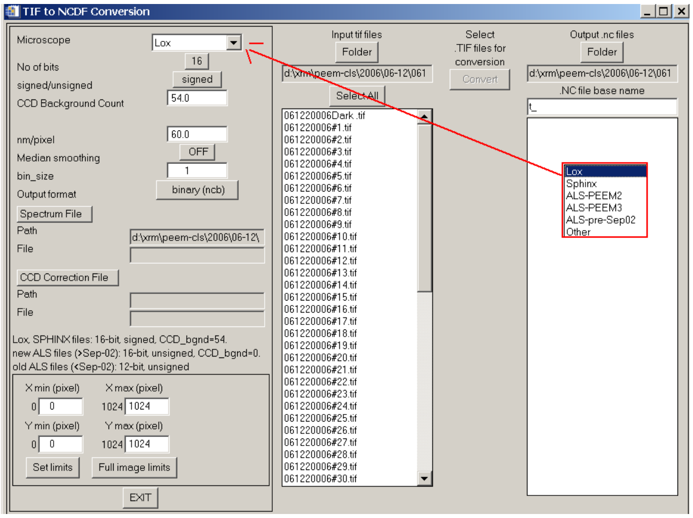
STACKS~**convert format** *(continued)*

return to top menu

PEEM - converts PEEM stacks written by various instruments (Lox, Sphinx, AALS-PEEM2, ALS-PEEM3, ALS-pre-Sep02 (peem2 old format), Other)



Recommended for ALL types of PEEM data



Microscope: Lox
No of bits: 16
signed/unsigned: signed
CCD Background Count: 54.0
nm/pixel: 60.0
Median smoothing: OFF
bin_size: 1
Output format: binary (ncb)
Path: d:\xrm\peem-cls\2006\06-12\
File:
Path:
File:

Lox, SPHINX files: 16-bit, signed, CCD_bgnd=54.
new ALS files (>Sep-02): 16-bit, unsigned, CCD_bgnd=0.
old ALS files (<Sep-02): 12-bit, unsigned

X min (pixel): 0 | 0
X max (pixel): 1024 | 1024
Y min (pixel): 0 | 0
Y max (pixel): 1024 | 1024

d:\xrm\peem-cls\2006\06-12\061

061220006Dark .tif
061220006#1.tif
061220006#2.tif
061220006#3.tif
061220006#4.tif
061220006#5.tif
061220006#6.tif
061220006#7.tif
061220006#8.tif
061220006#9.tif
061220006#10.tif
061220006#11.tif
061220006#12.tif
061220006#13.tif
061220006#14.tif
061220006#15.tif
061220006#16.tif
061220006#17.tif
061220006#18.tif
061220006#19.tif
061220006#20.tif
061220006#21.tif
061220006#22.tif
061220006#23.tif
061220006#24.tif
061220006#25.tif
061220006#26.tif
061220006#27.tif
061220006#28.tif
061220006#29.tif
061220006#30.tif

d:\xrm\peem-cls\2006\06-12\061
.NC file base name: t_

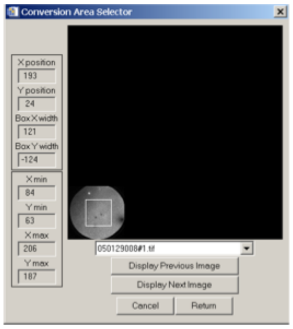
The output is either a binary stack (*.ncb) (read with *stack_process*), or as a set of {*.nc} images (read in with *Stacks~Analyze~stack list* input (or *Stacks~Analyze~Zimba*).The default conditions for SPHINX, ALS-PEEM2 [new (>Sep-02) & old (<Sep-02)], ALS PEEM3 and Lox are indicated in the left centre of the widget.

The upper left part of the widget sets the conversion parameters

nm/pixel – sets spatial scale
No of bits – 12 or 16 bit
Signed/unsigned
Median smoothing – Yes/ No
Bin - reduce size and improve statistics by 2x2 => 1; 3x3 => 1 etc
Lox - Yes / No
Output format {*.nc} set with *.sl stack list, or *.ncb file
CCD Background count – signal in absence of X-rays

Input files **Folder** – select folder with data to convert to stack (lists *.tif files)
Select all – highlights all files (alternatively select those you wish to convert

Select CCD Background file – for patterned dark signals
Select spectrum file – get energies of stack from spectrum generated during stack acquisition
Get Limits – identify pixel range to convert
Convert – this button will only become active after an energy spectrum is identified and the files to be converted have been selected. After pushing this button the files are converted to *.nc files, either temporary, or permanent. If Output format is set to *.ncb, only the binary stack is written to disk.



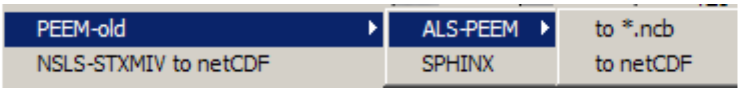
X position: 193
Y position: 24
Box X width: 121
Box Y width: -124
X min: 84
Y min: 63
X max: 206
Y max: 187
050129008#1.tif

STACKS~**Convert format** *(continued)*

return to top menu

PEEM-old *(ignore unless desperate!)*

- ALS-PEEM



to *.ncb – convert to binary stack file, without writing individual netCDF files

to netCDF – convert to set of *.nc (netCDF) files and a stack list (.sl) file

This activates the ALS PEEM read-in widget (see Read~images~ALS PEEM) and the user selects the first and last files, all conversion options, then executes the conversion.

Note: this widget does not work in the IDL VM version

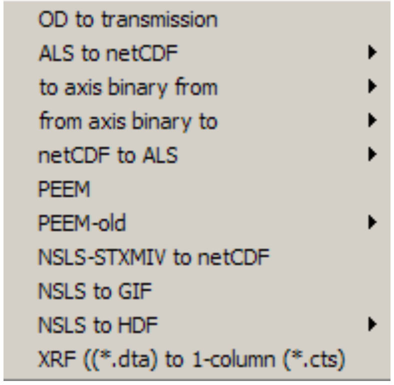
- SPHINX - convert 16-bit TIF files from SPHINX LabView program (Elmitec at SRC)

to *.ncb – convert to binary stack file, without writing individual netCDF files

to netCDF – convert to set of *.nc (netCDF) files and a stack list (.sl) file

This activates the ALS PEEM read-in widget (see Read~images~ALS PEEM) and the user selects the first and last files, all conversion options, then executes the conversion.

Note: this widget does not work in the IDL VM version



NSLS STXMIV to *.ncb – converts nsls STXMIV image sequence directly to a binary stack file (all images in one file). NB One can use Zimba or Jacobsen align procedures on *.ncb files.

NSLS-STXMIV to netCDF - convert NSLS stxmIV format files to netCDF format files

NSLS to GIF – converts one *.nc file to a GIF image

NSLS to HDF - converts *.nc files to HDF format data files

1 converts one file selected by pickfile

many - converts a sequence of files with first, last identified by pickfile

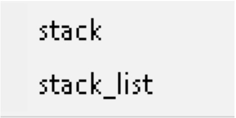
XRF ((*.dta) to 1-column (*.cts)

Converts XRF spectra from Oxford system to readable values

Differentiate – compute the derivative of the spectra at each pixel in a stack

Expand - add additional pixels on left/right and top/bottom to a selected stack.

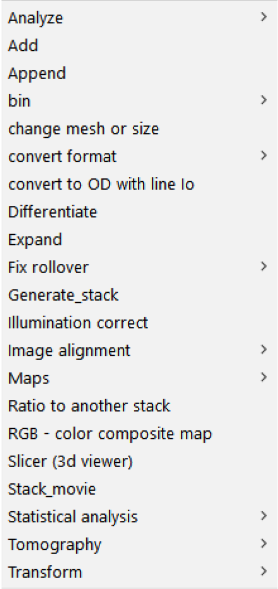
Fix rollover



– as for Images, but applied to all images of a stack. for some image files (e.g. ALS PEEM) it is possible to have signed or unsigned formats which leads to 'roll-over' for pixels with above 16 K counts (e.g. if an unsigned integer data set is read using a signed integer). This replaces all pixels with a -ve value by 65,535 plus that value. Similarly, in phase outputs from ptychography reconstruction, many reconstruction programs automatically ‘fold back’ phase values such they lie between –π and +π. NB the phase roll-over code only partially works.

STACKS (*continued)*

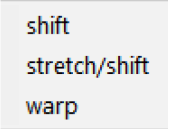
return to top menu



Generate_stack - a stack is computed and saved from an image and a spectrum, with the image and spectrum selected by the user from aXis2000 buffers.

Y_Illumination correct ratio the first selected stack by a second stack (which is the illumination, which can be generated by heavy smoothing all the images in the first stack)

Image alignment - tools to manually align a series of sequential files (either ALS or NSLS format; read in after defining first and last of the sequence). *Note this always works whereas the manual align in ZSTACK is somewhat 'temperamental'*. There is a nice manual alignment tool in *Stacks~Analyze~Jacobsen stack analyse*



shift - 1-point calibration of image (x,y-scale).

stretch/ shift - 2-point cali. of each image (x,y-scale)

After shifting or stretch/ shifthing the (x,y) scales for image alignment, this routine then:

1. grids to a user-defined pixel size (should be similar or smaller than the recorded pixel size to avoid loss of information)
2. truncates to a fixed user-defined [xmin, ymin, xmax, ymax].
3. bins to a user defined factor (1=no binning) to trade off S:N versus spatial resolution
4. writes a netCDF file. The names are FORCED to be the same as the input but with the 1^{st} letter of the original filename converted to **s** (shift) or **a** (shift/stretch).
5. generates ***.sl** file of filenames (read using *Stacks~Analyze~stack list input*). In addition it stores 2 (**s**) or 4 (**a**) files (in buffers 5,6 and 8,9) which are the shifts in real-space units (microns) **The parameters defined in the first pass are saved and applied automatically to all subsequent files**. The X,Y shifts and stretch terms for each file are accumulated in buffers 7 (stretch) 5,6 (point 1,x,y) and 8,9 (point 2,x,y) for later use to explore how the microscope or the sample is changing.

Notes for image_alignment*:*

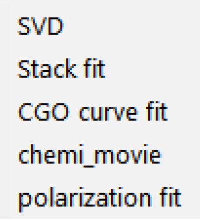
- Clear buffers 5-9 (using the **Clear Buffer** *above the thumbnails) before starting as buffers 5-9 are used for results*

- Check the extreme limits of a stack sequence to ensure the [xmin, ymin, xmax, ymax] values used to define the common part of all images are valid for all files.

Warp – Image alignment using a polynomial 2-d transformation to align successive images to 4 or more fiducial points identified on the first image, by manually (position cursor and click) on the same fiducial points in each subsequent image. This can be useful when the sample distorts during a measurement due to e.g. radiation damage to an organic film the sample sits on (formvar).

STACKS (*continued)*

return to top menu



maps

SVD - convert an image sequence (stack) to a set of component maps using **singular value decomposition** (SVD) procedures. When accessed from this menu item, the **input must be an AXIS format binary file stack (*.ncb)**. Typically the **input** stack consists of a set of images prepared on an optical density (OD) scale with careful alignment. The reference intensities at each energy of the stack are extracted from user identified reference spectra (*.txt, read from disk), which are interpolated / extrapolated as needed to match the stack energies. The user is prompted to either (i) provide a ***.par** parameter file which contains a list of the reference spectra, or, (ii) identify the filenames of the individual reference spectra, and, if desired, save a list to a ***.par** file. This is helpful in cases where the same set of reference spectra might be applied to a number of different stacks, containing the same species..

The **output** is a set of **chemical component maps**, automatically written to files with names constructed from a root and the component names (it is wise to keep the component names *SHORT and without spaces* to avoid excessively long file names). In addition the **residual** signals averaged over all energies is saved as an 'image'. Comparison of the magnitude and spatial distribution of the residual signal is a useful way to evaluate the validity of the analysis. Optionally a **stack of the residual images** (difference of data and fit at each energy) can be saved for later examination.

If the input stack is in OD units, and the reference files are on absolute linear absorption scales (**OD1**, or optical density per nm thickness at the standard density of the component), the Z-values of the resulting component maps are in absolute thickness units (in nm, assuming the reference spectra are in units of nm^{-1}).

NB the SVD code can also be run from the IDL command line and has different features not implemented directly in aXis2000. See the code file and IDL help for further details.

Stack fit – Performs a fit of the spectrum at each pixel to a sum of (1 to 8) user-defined model spectra and a constant. Stack fit uses the SAME SVD routine as the stacks~Maps~SVD routine, only differing by including an additional **constant** component (same value at each energy, i.e. flat spectrally (same at all energies) but different at each pixel. This always leads to a better fit statistically (smaller residual, often by 2 orders of magnitude) but may lead to systematic error in the result. *Stack-fit* rather than *SVD* should be used where there are offsets of reference spectra relative to the data, due to e.g. Io errors, or the area of the stack contains signal from component which do not contain the element whose X-ray edge is being measured.

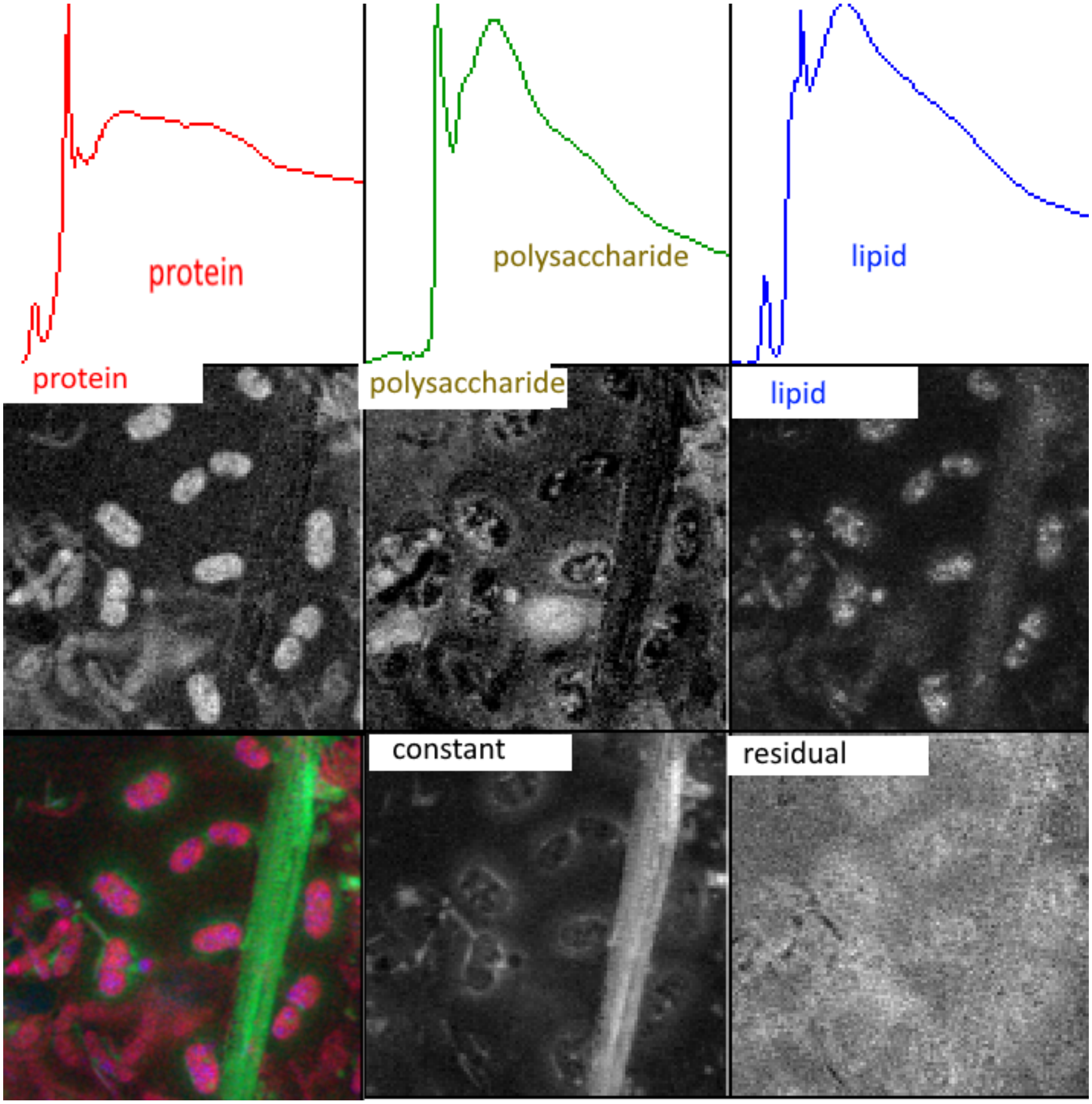
CGO curve fit - This uses a conjugate gradient optimization method (from Numerical recipes) to perform spectrum-by-spectrum linear regression curve fit to user supplied reference spectra. The user dialog is similar to those used in the *SVD* and *stack fit* routines. The fit of a single spectrum can be carried out using this method with the *spectra~curve fit* command.

In tests on low noise data sets with valid OD scales and with accurate spectral models, essentially identical component maps are generated by SVD maps, stack fit and CGO curve fit

return to top menu

EXAMPLE of using *stacks~maps~SF* to convert a C 1s stack to a set of component maps.

Data: wet biofilm (courtesy John Lawrence, National Water Research Institute, Saskatoon)



Reference spectra are quantitative (OD1) so z-scales on the **protein, polysaccharide** and **lipid** maps are nm thickness. **constant** represents signals from water, a silicate diatom, and $CaCO_3$ and K at the envelope of the cells. The color coded composite (buffer 7) combines the **protein**, **constant** and **lipid** component maps. The image area is 25 µm x 25 µm. (Lawrence et al, Applied Environmental Microbiology <u>69</u> (**2003**) 5543)

STACKS~maps (*continued)*

For **SVD, Stack fit** and **CGO-fit** analyses with less than 5 reference species, at completion the aXis2000 buffers contain:

Buffers **1-3(5)** – the r*eference spectra*
Buffers **4-6(8)** – the *component images* - i.e. the spatial distribution of the component
Buffer **7** – *constant*: the map of the constant term (for **Stack-fit** only)
Buffer **9** – *residual***:** the map of the chi square values of the fit at each pixel, averaged over all energies. To evaluate fit quality, the residual magnitude should be compared to the average OD of the stack.

If there are more than 4 components, the spectra are displayed, but, after the fit is complete, the component maps are displayed in sequence, starting in buffer #1.

For analyses with more than 5 reference species, the component maps are stored in the same buffer number (**1-8**) as the reference spectra and buffer 9 contains the residuals map.

After making the component maps it is often useful to explore the spatial correlations of components by combining any 3 of the component maps into a single **color-coded composition map**, by using the *Display~RGB composite* command or (if you are running a IDL licensed version) *stacks~RGB color composite map*

HINT: If you want to create a 1-color or 2-color composite, select a buffer that is empty or one that contains a spectrum for the colors you do not wish to include.

<u>How to evaluate the significance of maps derived using SVD, Stack fit or CGO fit.</u>

The user of *stacks~Maps* must realize that the fitting code **will always give a result, but the <u>result may not be valid</u>**. **<u>It is ESSENTIAL to evaluate the quality of the fit</u>**.

aXis2000 provides a number of tools to do this.

* Examination of the **residual stack** can help to evaluate the validity of the fit. Spectra extracted from various regions of the residual stack should be only noise; if there is a missing chemical component one can sometimes obtain its spectrum in the poorly fit regions of the residuals stack.

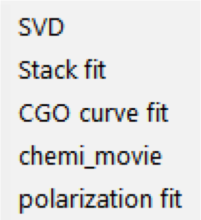
* One can apply *images~generate_mask* on the **residual map** (in buffer #9) to write out a region_of_interest (ROI) file corresponding to the poor fit regions. The ROI files can be used in *stack_process* to extract the spectrum of the poor fit region, which may be a 'missing component'.

* One can apply *images~generate_mask* **on the component maps** to generate the ROI of the pixels with large amounts of components of interest. After extracting the spectrum of those pixels using the ROI files in *stack_process* the **<u>quality of the spectral fit</u>** can be examined by using *spectra~curve fit~{linear regression (=stack fit) , or CGO(=SVD)}*. The average amount of each component, along with residual and a constant (if stack fit) are displayed in the aXis2000 log window.

* An alternate method, such as a multivariate statistical analysis (MSA) approach- e.g. PCA_GUI (*stacks~statistical analysis~PCA_GUI (CJJ Dec 2005*) or **Mantis** (https://spectromicroscopy.com/) can be explored, If the results of the forward fitting approach (maps~SVD or stackfit) (often called linear combination fitting (LCF) in XAS spectroscopy) and MSA methods are similar, greater confidence can be placed in the resulting component maps.

Chemi_movie

Performs a SVD or stack fit analysis of multiple stacks recorded with same parameters, and processed to a standard format. Stacks are in separate folders, and are identified by a list file: *Cu_CO2R_pot_series.lst*. The resulting component maps can then be combined to follow how the morphology and/or chemistry changes under applied stimulus (e.g. applied potential (echem) or time, etc.



Polarization fit

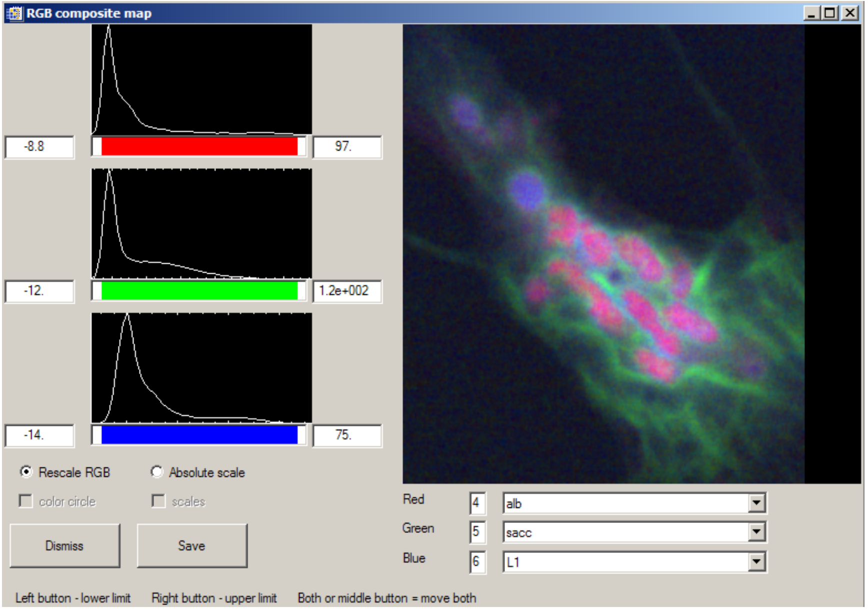
Fits a sequence of images where the control variable is the angle between the E-vector and the image. Such data can be generated by azimuthal angle scanning (e.g. at STXM532), or by rotating the E-vector with an EPU (at STM1102 and CLS-STXM). The polarization signal is fit to

$$I(\theta) = C + A * \cos^2(\theta - B)$$

where C is a non-angle dependent constant, A is the amplitude of the dichroic signal and B is the 'director' (angle of maximum intensity)

Note that the quality of the fit can be evaluated by selecting a region of interest, extracting the angle-dependent signal then using s*pectra~curve fit~pol fit*

RGB Color composite map – uses a widget with lower / upper limit controls and selector of any of the aXis2000 buffers to visualize spatial arrangements of chemical components (e.g. as derived from curve fitting (stacks~maps) or multivariate statistical analyses (or simply, images at 3 different energies)



-8.8 97.

-12. 1.2e+002

-14. 75.

● Rescale RGB ○ Absolute scale

□ color circle □ scales

Red 4 alb

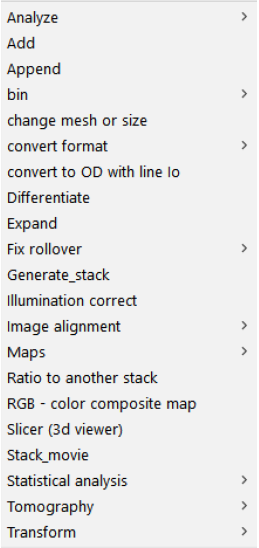
Green 5 sacc

Blue 6 L1

Left button - lower limit Right button - upper limit Both or middle button = move both

Note: The **RGB Color composite map** does not function properly when executed from the virtual machine (VM) version (tested Feb 2025, on IDL6.3 and IDL8.6). Use *Display~RGB composite*.

return to top menu



Ratio to another stack – divide the stack by a user selected stack (*.ncb) (NB stacks can be generated from an image and spectrum using *stacks~Generate_stack*)

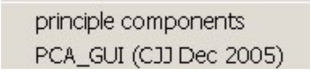
Rotate – rotate the whole stack by user selected angle. Boundaries are filled with the minimum over each image

Slicer (3d viewer) – IDL's 3d viewer with read in of a binary stack

Stack_movie calls *Jacobsen stack_movie* routine. Files defined by user-selectable **stack list file (*.sl)**.

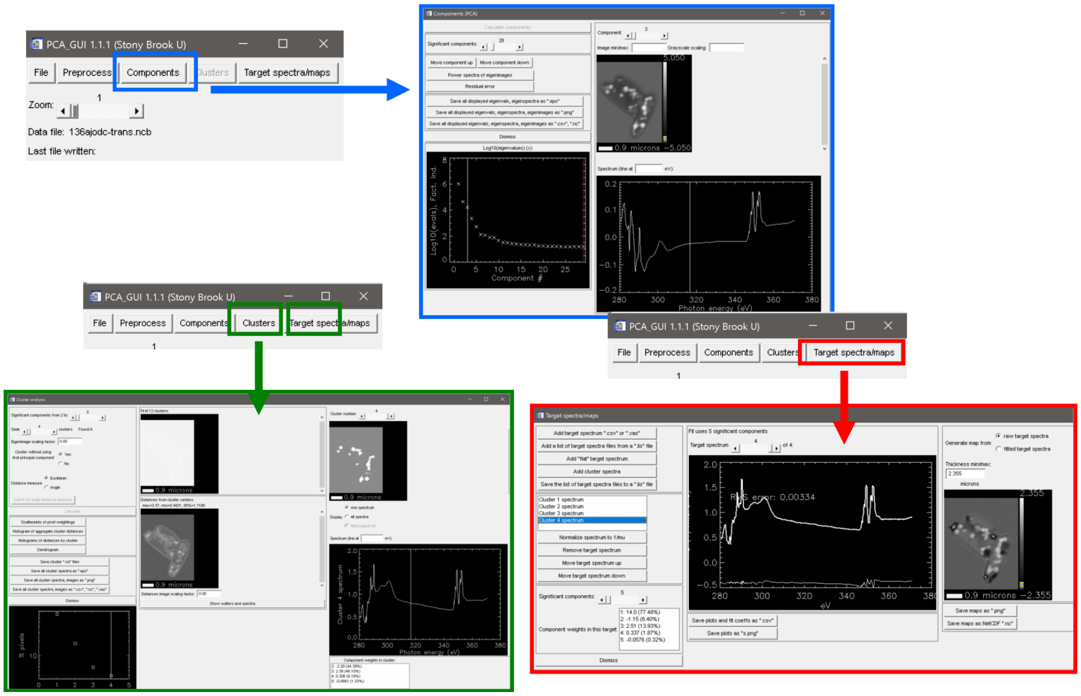
If 'cancel' is selected, the *stack_build_list* routine is initiated.

Statistical analysis



- **Principle components** - applies the IDL principle component analysis package (PCOMP.SAV) to derive the **eigenspectra** (power weighted representation of the data) from a set of OD images (binary stack in an *.ncb file). The routine implemented in aXis2000 provides the **eigenvalues** (fractional contribution of each eigenspectrum) and eigenspectra of the first 8 components . This is useful to obtain a sense of how many independent chemical species might be present in a give data set. The **eigenimages** are readily generated by using the eigenspectra as model spectra in a SVD map

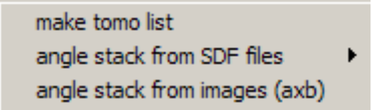
- **PCA_GUI (CJJ Dec 2005)** – combination of principle component, cluster and target analysis allows for unsupervised analysis to extract a set of potential NEXAFS spectra, with extensive capabilities to evaluate and optimize. **See the manual for this package for further details.**



return to top menu

Stacks *(continued)*

Tomography Routines to analyse **4d (x,y, E, θ)** data.

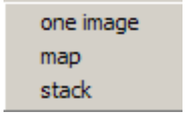


Make tomo list – write a list of the folders containing tomo data.
This can then be edited to include the angles at which each stack was measured.

Angle stack from SDF files

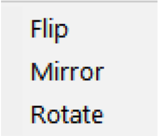
reads a set of angle varying images to generate an **(x,y, θ)** stack

one image – select one image from a set of images (stack) at each angle
map – select a map from each folder
stack - read full 4-d (x,y,, E, θ) data and convert to a format suitable for read in to IMOD.



Transform – perform standard transformations to each image of a stack

Flip – invert image vertically through a horizontal centre line
Mirror – invert image horizontally through a vertical centre line
Rotate - rotate image about a user defined point, by user defined angle.



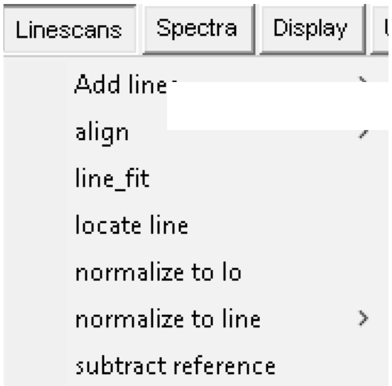
LINESCANS (processing of linescan spectra)

return to top menu

Processing of linescan spectra. Linescan processing methods are also useful for a variety of image analysis tasks such as:

- obtain averaged **profiles** from an image along X or Y
- remove low frequency noise from images (average all horizontal, or all vertical, then ratio)

HINT: If the direction of interest in the image is not oriented along vertical or horizontal, use *Image~rotate* to rotate the direction of interest to horizontal or vertical

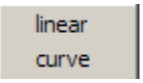


Add lines - sums all lines selected with cursor (lower / upper limits)

- Horizontal (for Linescans, sum spectra over a range of length along the line)
- Vertical (for Linescans, sum line profiles over a range of E)

align - 'tilts' data to user specified line

- Linear – straight line defined by 2 points
- Curve – curved line defined by multiple straight line sections



line_fit – apply SVD-based curve fit to a linescan (same method as in *stacks~maps~stack-fit*). This is an excellent method to analyze and visualize chemical composition along a line. Quantitative thickness along the line can be generated if quantitative reference spectra (OD1) are used.

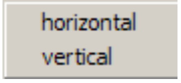
Linescans *(continued)*

return to top menu

locate line – draws the (x,y) position of a linescan on the image currently displayed using information from the parameter file for the linescan (Jan2023: only set up for SDF format data written by STXM_control). Be sure to load and display the image on which the linescan was defined (or at least one which includes the spatial region of the linescan).

normalize to Io – divides each **horizontal** line by user-selected buffer containing the Io, then computes –log(Image/Io). Get Io from *Add lines~Horizontal* if there is an open region in the linescan or from a separately recorded point or line spectral scan.

normalize to line



Horizontal - divides each **horizontal** line by content of user-selected buffer
Vertical - divides each **vertical** line by content of user-selected buffer

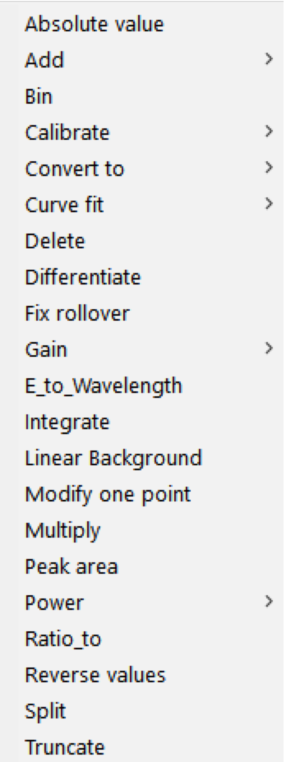
> ***Hint***: To remove line-by-line periodic scan noise, generate a 1-d profile of periodic noise by *Add lines~vertical* (or *Add lines~horizontal*, as appropriate) over all or part of an image then subtract that profile from the image using *Normalize to line*.

subtract reference computes (Image – ref), The reference 1-d profile signal is taken from user selected buffer (this routine does the same thing as *normalize to line ~horizontal*)

==

SPECTRA

return to top menu



Procedures for processing 1-d data (y as a function of x). This is often spectra, but can also be intensity profiles along a line, Io signals, etc

Absolute value – computes absolute value

Add

Append - append 2 data sets - all data points in overlapping region are kept
Buffer - generate SUM of 2 spectra (interpolation used)
Constant - add (or subtract) constant to y-axis

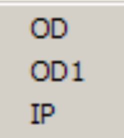
Calibrate (for each of **X** and **Y** axes):

Auto - uses last axis recalibration parameters
1 point – shift values by user defined amount from cursor selected point (shift)
2 point – shift vales based on 2 points (shift & stretch)
Numerical 1- or 2-point with numerical input

Spectra *(continued)*

return to top menu

Convert_to



OD – automatically generate OD spectrum from a transmission spectrum by identifying a second buffer with Io

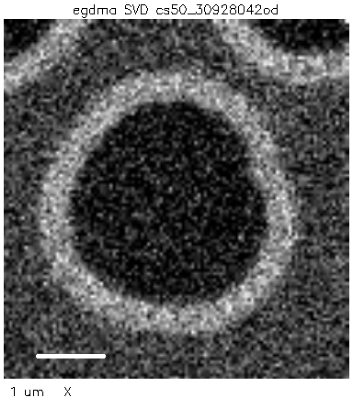
OD1 – converts to intensity for 1 nm of material (needs XX-sf.od1)

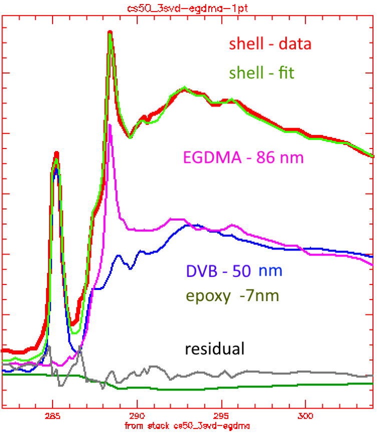
IP – converts E-scale from kinetic energy to binding energy with user supplied photon energy. IP = E_{photon} - $E_{kinetic}$

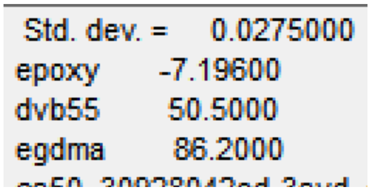
Curve fit

linear regression – use IDL regress function to fit the spectrum in the current buffer to a set of reference spectra. The reference spectra can be identified using a *.par parameter file (identical to that used in *stacks~map* and *linscan~fit*)

CGO fit - use conjugate gradient optimization method (CGO) to fit the spectrum currently displayed in a buffer to a set of reference spectra which the user is prompted to select from the disk. This is a useful method to check and display the fits that are obtained from image stacks. The fit components are listed on the IDL log as well as in the aXis2000 log display.







```
Std. dev. =   0.0275000
epoxy       -7.19600
dvb55        50.5000
egdma        86.2000
```

Pol fit – fit a linear polarization dependent signal to

$$I(\theta) = C + A * \cos^2(\theta - B)$$

Spectra *(continued)*

return to top menu

Delete - delete all data between 2 cursor-identified positions

Differentiate – take derivative of displayed signal (simple ΔY/ΔX only)

Fix rollover – identifies data with modulo 2^{16}-1 range, and adds units of to successive rolled over portions of the spectra

Gain - **multiply** – multiply y-axis values by a factor
- **divide** – divide y-axis values by a factor

E_to_wavelength - X-axis transformed by 12398/X (both way conversion)

Integrate – determine integral of displayed signal

Linear background - subtract user defined line

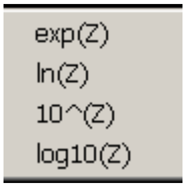
Modify one point – left click – add a point, right click – delete the point

Multiply – multiply the spectrum in the current buffer with that from a used selected buffer.

Peak area – determine area under curve between user-selected limits

Power

- **exp(Z)** – exponential of z
- **ln(Z)** - natural log of z-values
  (use to convert Transmittance to Absorbance)
- **10^(Z)** – raise z-values to 10
- **log10(Z)** - base-10 log of z-values



Ratio_to - takes ratio of 2 buffers (interpolated to same x-axis scale)

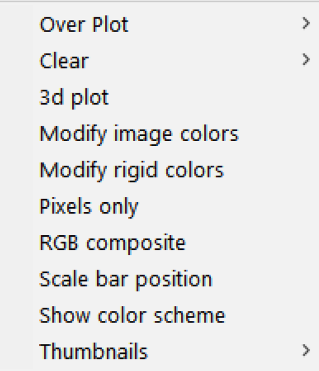
Reverse values – reverses the (x) values in a spectrum.

Split – separates a multi-valued spectrum into single-valued regions, placed in successive buffers. Data of this type can be generated by acquiring multiple regions which are not in strict increasing-energy order. This can be useful as a means to track radiation damage. Recording the most chemical sensitive part of a NEXAFS spectrum at the beginning and end of a point spectrum is an effective way of having an internal check on radiation damage.
Hint: AXIS auto-detects multi-section spectra recorded with overlapping multiple regions and plots all components in the same buffer. The separate single-valued sections are placed in sequential buffers by this command.

Truncate - truncate spectrum to data between 2 cursor-identified positions

return to top menu

DISPLAY



Over Plot - display **multiple spectra**

No Rescale - select multiple buffers (0-9)
Y-axis of data is preserved

Rescale – select multiple buffers (0-9)
each spectrum is rescaled to full screen

Window - add a single buffer, with data rescaled within y-limits selected by the user using the cursor

Shift - add a single buffe → bottom y-position selected by cursor; no rescale (ie y-scale is set by previous data)

All modes of *Display~Over Plot* can be combined in any order. All previous *Display~Over Plot* processing is preserved in the main plot window until a single buffer is selected for plotting, by left-clicking on: thumbnail plot, buffer label, or indicator box.

Clear **Lines** – removes cursor-related lines from Main Image

Current – erases only the currently selected buffer

Selected - any of 0-9 buffers can be erased (also as second row button)

All - resets all buffers to zero (like starting a new version of aXis2000)

NB the **Clear Buffer** single line command accesses the last 2 commands

Hint: If the colors go 'crazy', restore the default color scheme by **Clear Buffer** or *Display~Clear*. This often occurs on the first use of *stacks~analyze~zimba* – a black display is generated when displaying the first selected spectrum. Use the **Clear Buffer** command above the thumbnails to reset the color scale, and then erase and re-select the regions in *stacks~analyze~zimba*

3d-plot - generate 3a -d shade surface plot from an image with x,y,z axes.

Modify image colors - pop-up widget (XLOADCT) that selects color scheme and adjusts (top, bottom, gamma) variables. *In **Win** systems, if 256 colors is set, this updates all graphic windows dynamically. If color is set to higher value (16-bit, 24-bit, or true color) one needs to redisplay a graph to change the color scaling. Since each thumbnail sketch (upper left of screen) is displayed independently, the color scale of the thumbnail sketches will only change after they are selected.*

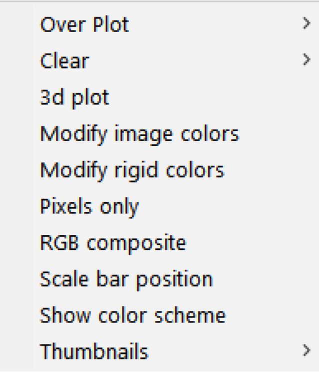
Modify rigid colors - select custom colors for buffer specific spectral colors, background, foreground, etc

Pixels only - plot using 1 display pixel per pixel of data

RGB composite – takes images in 3 buffers and assigns then to red, green and blue color components of a single image. It is necessary that all images are the same size and are of the same physical region. The composite image can be saved to disk in a 3-component TIF format which reads into Paint.net and Powerpoint. The individual R, B, G components can be read back using the *Read~Images~Other~TIF~data* command. User is given option to use a common scale for all three images (thus preserving information about relative intensities of components), or autoscaling each image independently, which will give equal visibility to each component.

Display *(continued)*

return to top menu



Scale bar position – use cursor to define the left end of the scale bar.

Show color scheme - display current colors assigned. Note that the bottom 16 colors in the 256-color table are assigned to 'hard' colors to allow ready differentiation of different buffers.

Thumbnails – display multiple buffers on main window

4 – user selects any 4 buffers

- **common scale** - plot using identical scale for all images (in B/W for printing)
- **rescale each** – plot with each image byte scaled to its data values (in color)

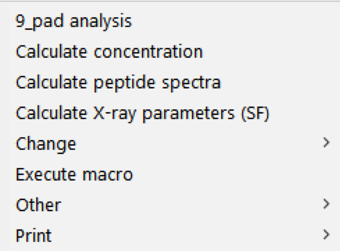
9 – display all 9 buffers

- **common scale** - plot in B/W (for printing)

rescale each – plot with each image byte scaled to its data values (in color)

return to top menu

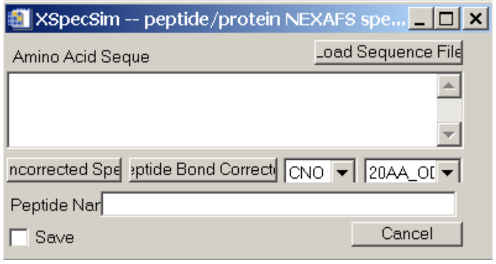
UTILITIES



9_pad analysis automated analysis of 9-pad radiation damage patterns. Templates for 10 areas are adapted to the spatial scale of an experimental 9-pad image, multiplied and integrated to generate the signal as a function of dose.

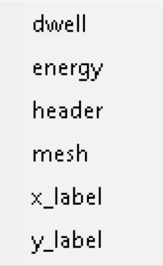
Calculate concentration computes equivalent concentration in aqueous solution (in Molarity). Use with *utilities~calculate X-ray absorption parameters (SF)* to ESTIMATE visibility of spectral features in aqueous solution, and estimate the solution concentration and water thickness to guide sample preparation. [see: Axis2000-manual-calc_conc.ppt 04-Jun-18]

Calculate peptide spectra – launches a widget to generate the NEXAFS spectrum of a user specified amino acid sequence from the C, N, O spectra of the constituent amino acids. Note, the AA sequence can be taken from the protein data bank (PDB) or generated from standard codes for the amino acids.



Calculate X-ray parameters (SF) - Determine the **mass absorption** or **transmission** for user defined elemental compositions. The mass absorption elemental signal for a given chemical formula and density provides normalization for reference spectra to the quantitative OD1 (optical density response for 1 nm thickness) which can be used in the SVD map and stack_fit routines to obtain quantitative thickness maps. In addition, the transmission for a user-defined composition, density and thickness can be useful to evaluate feasibility of a sample. NB The ***sf.pro*** program (in the AXIS directory) is a full implementation of the old CXRO SF program for X-ray constants, which corrected some errors. Routine written by Billy Loo, 1996. The elemental X-ray mass absorption coefficients are from 1996. There my have been updates since then. See https://henke.lbl.gov/optical_constants/ for latest values.

Change



dwell – change dwell per pixel (ms)

energy – change photon energy (eV) or equivalent parameter

header – modify header (this is what will be saved to disk)

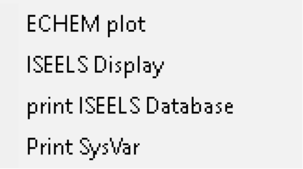
mesh - redefine numbers of pixels (images) or points (spectra). This is useful to match pixelation of different mages. Many routines in aXis2000 can modify pixel count by 1 or 2 pixels. *Change~mesh* can be used to make the (x,y) pixilation of each image of a set of images the same

x_label – change X- label (scale bar, energy, dwell)

y_label – change Y-label (polarization, etc)

Execute macro – execute a file of standard aXis2000 commands (with parameters) So far only a few of the axis commands are set up to be used in this way.

Other – commands not directly related to spectromicroscopy



ECHEM plot – generate composites of images and electrochemical results (e.g. cyclic voltammogram) to make a movie of in situ electrochemical results

ISEELS Display – too to plot spectra from an inner shell electron energy loss spectrometer.

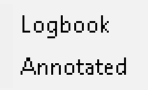
print ISEELS Database – print all selected ISEELS spectra from the gas phase core excitation data base (http://unicorn.mcmaster.ca/corex/cedb-title.html)

Print Sysvar - display on the IDL log window (so, need licensed IDL) the current values of all IDL graphics parameters (used in code development).

Utilities *(continued)*

return to top menu

Print



Logbook (default: 3" x 3") – no annotation
Annotated (default: 4" x 4") – brings up the IDL annotate widget.

Hint: *(1) A fast alternative to printing is to use a screen capture program such as **MWSnap** http:/ www.mirekw.com/ to copy the screen area you want to the clipboard, then use an image processing program such as Paint.net to further modify (e.g. annotate). The modified screen captured image can then be transferred into a presentation program, such as powerpoint or keynote.*
Hint: *(2)* Use The PRINTER option in *axis_xxxx.ini* (xxxx = win, unix, macos) to set up your default system printer

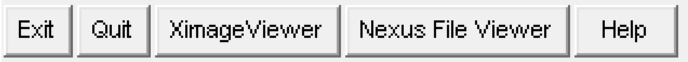
Set preferences – allows user to modify the default parameters which are contained in a file called **axis_win.ini** (for Windows OS) or **axis_macos.ini** (for MacOS). This uses a widget (*see fig*) to let the user define a large number of different parameters. The modified initialization file is written at the end of each normal exit to aXis2000 so that it starts in the same configuration (default directories etc) as in the last use.

On start-up, if AXIS cannot find the default initialization file (**axis.ini** in the subdirectory where the AXIS code files are located), it starts this widget.

NB in post 2023 versions of aXis2000 this has been removed as it does not work right.

Second row of single command buttons

Above the Main graphical window



Exit write **axis.ini file** with parameters then return to main IDL command window. It is a good idea to use Exit rather than Quit, since Exit saves the last path used, which can save time navigating to a folder you were processing, the next time you launch aXis2000.

QUIT return to main IDL command window, without saving current parameters

NOTES:
If you have an internet connection, and you have the command to start your browser identified correctly in the **axis.ini** file, then you can click on the indicated synchrotron sites or the aXis2000 label to connect to the X-ray microscopy facility. Clicking the label at the bottom of the splash screen takes you to the aXis2000 web site

Additionally, for those using a licensed version of IDL.

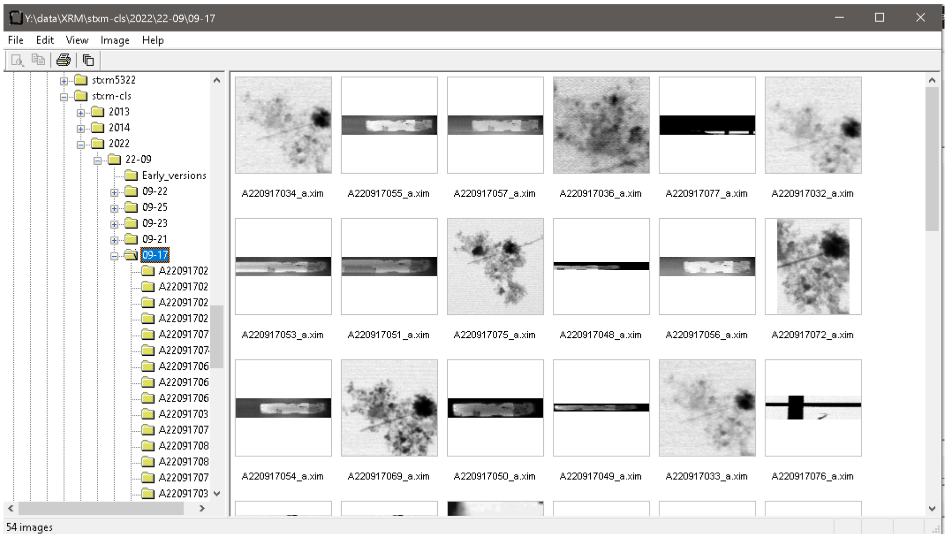
1. From the Command Window of IDL type *axis2000* to (re-)start AXIS.
2. If you wish to display the **rotating AXIS logo**, restart AXIS by typing **axis2000, /spin** at the IDL prompt.

return to top menu

XimageViewer

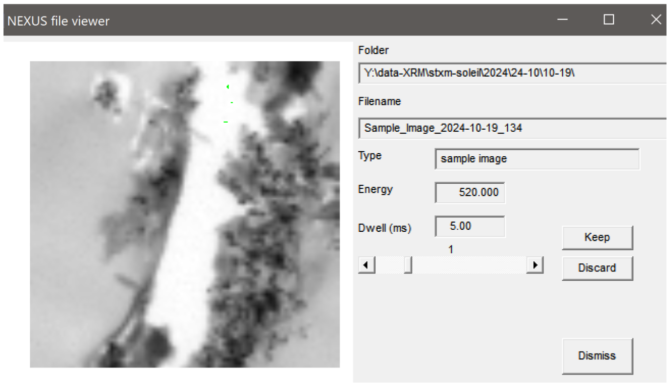
Launch a viewer of image files. In addition to standard image files (jpg, bmp, gif, tif, png, etc) this displays ALS_STXM (SDF) format file (*.hdf,*.xim) pairs in the selected folder. This is useful when you do not have detailed information on data files.

This DLL was provided by **Ivo Koprinarov.**



Nexus File Viewer

Launch a tool to review NeXus format files to allow rapid discard of unwanted files (the STXMs using pixelator acquisition save all files, 30-50% are junk). This is also useful to identify data of interest when the log book notes are insufficient. The data viewed in NEXUS file viewer is the default file name in *read~STXM (NeXus)*



Folder: Y:\data-XRM\stxm-soleil\2024\24-10\10-19\

Filename: Sample_Image_2024-10-19_134

Type: sample image

Energy: 520.000

Dwell (ms): 5.00

1

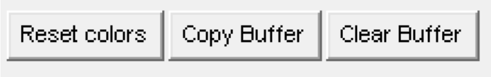
Help

Calls Adobe Acrobat reader, using the Unix (Acroread) or Windows command (AcroEx32) to bring up **AXIS.pdf** (a pdf version of the **aXis2000-manual.doc**). Since the location of the Adobe reader depends on the system, the system command that will start the Acrobat reader and display the AXIS.PDF file is part of the AXIS.INI file. These details ares set by the **help_cmd** line in the **set_preferences** dialog, or by using a text editor to edit, axis-win,ini (Windows) or axis-unix.ini (unix, MacOS) file.

Second row of single command buttons *(continued)*

return to top menu

above the 9-thumbnail display



Reset colors

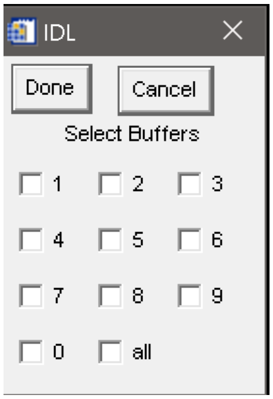
Changes Display color scheme to the current default. Often the color scheme is distorted after running one of the **stack~analyze** widgets. Use this button to reset to the default color scheme to that for aXis2000.

COPY buffer

- moves contents of the current buffer to one of the 10 data buffers in AXIS. The destination buffer is selected by clicking on the thumbnail, or by clicking on the label of the buffer to which you wish to move the data. Note the latter is the only way to move data into buffer 0.

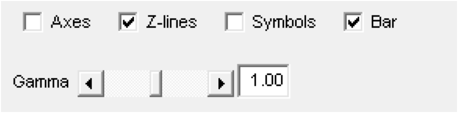
Hint: *To preserve a processed spectrum or image located in Buffer 0 you must use* **Copy Buffer** *to transfer it to buffer 1-9 to avoid overwriting the result in the next processing step.*

CLEAR buffer pops up the *multi-buffer selector widget* to let the user select those buffers they wish to clear.



CONTROLS BELOW THE MAIN IMAGE WINDOW

return to top menu



☐ Axes ☑ Z-lines ☐ Symbols ☑ Bar

Gamma 1.00

Axes

If selected (**ON**), the X and Y scales are displayed.
If not selected (**OFF,** the default) only the image, X-axis and Y-axis labels are displayed.

Z-lines

– **ON** (default) - X, Y lineouts generated at every mouse move
- **OFF** - X, Y lineouts generated only on left mouse click

Note: The operation of *Zoom~pan* is also switched with this button. If ON, the zoomed image updates on every mouse motion. If OFF, the zoomed image updates only after a mouse click on the main image

Symbols

ON - plot symbols on lineouts and spectra (main image window only)
OFF (default) – no symbols on lineouts or spectra

Note these options allow one to adapt to the speed of the computer. On large images, continuous updating of the lineouts and the zoom/pan, as well as symbol plotting, can take large amounts of cpu/graphics resources and can slow aXis2000.

Bar

ON (default) display white scale bar [size is (1,2,5,10 pattern; within 5-10% of image]
The position of the scale bar can be defined using *Display~scale bar position*
OFF – scale bar and value are not displayed

Gamma

use either the slider or the number boxes to change the image gamma (for the aficionados, se https://en.wikipedia.org/wiki/Gamma_correction). Note this control is only active if an image is displayed.

************ XXXXXXXXXXXXXXXXXXXXXXXXXX************** return to top menu

INSTALLATION

************ XXXXXXXXXXXXXXXXXXXXXXXXXX**************

AAData
java
logo
sfdata

Copy all code into a directory. (use **WinZip** (or similar) to **expand axis2000.zip using folder names** into your source directory). –
I STRONGLY RECOMMEND using the folder name **c:\aXis2000\** on Windows systems, and **users\aXis2000** on MacOS systems as the main directory / folder.

There should be 4 folders (AAData, java, logo, and sfdata) and over 500 files (mostly *.pro).

<u>To set-up a **licensed IDL system** so it starts aXis2000 every time:</u>

1. Start IDL (I use version 6.3 for Windows and 8.8 for MacOS)
2. Set up preferences
   Path – add the directory with aXis2000 files
   Start-up- working directory: c:\axis2000\ (or whatever your source directory is)
   - start-up file: axis2000_batch
3. Shut down IDL and restart.

To use IDL without automatically starting axis2000, type **aXis2000** at the IDL command line.

Several different versions of the INI file are included:

axis_win.ini (for Microsoft Windows);
axis_macos.ini (for Apple OS),
axis_unix.ini (for unix OS).
Use the one identified for your operating system. If necessary use a text editor to edit the entries for your needs. Typically, this will involve changes to the SIZE parameter (I choose a value that fills about ½ of my graphics screen), the location of your pdf viewer, and the location of your web browser. See below for an example and more details.

Hint1: There are often problems if you try to use a different drive letter, or folder name than c:\aXis2000. A particularly **BAD CHOICE** would be C:\Documents and Settings\Default User\aXis2000\ **since the IDL code in aXis2000 automatically TRUNCATES text strings at the first space. Thus the program would be looking for files (or writing files to)** C:\Documents, a folder which does not exist.

Hint2: I may be possible to execute aXis2000 just from the aXis2000.sav, compiled version. However, many parts of aXis2000 would not work properly as it needs information from the distributed source code and the data files in the sub-folder.

return to top menu

The **aXis2000 initialization file** (axis_win.ini, axis_unix.ini, axis_macos.ini)

A sample AXIS.INI file (on a WIN system):

```
LINE#
1.  c:\aXis2000\
2.  WIN
3.  PS
4.  Y:\Backup\papers-old\X-ray_microscopy\CNTs\MWCNT-Najafi\
5.  1.00000
6.  0
7.  mac
8.  idl.ps
9.  Arial*14
10. copy idl.win lpt1:
11. c:\progra~1\acrobat3\reader\acroRd32 c:\axis\axis.pdf
12. C:\Program Files (x86)\Google\Chrome\Application\chrome.exe
-----------------------------------
written: Mon Jan 02 14:58:04 2023
```

```
; ********** Initialization file for AXIS Widget ******************
;
; (aph 31-Jan-00)  Please adapt to your hardware
; Items set                     devices currently supported
; ---------------------------  --------------------
1 CodePath                   (e.g. c:\aXis2000\) - include final separator
2 Screen display device      (WIN, X)
3 Print file format          (PRINTERPCL, PS)   ; Printer uses system default
4 Default data directory     (any - last path will be saved on Exit
5 graphics scale factor      (0.5 - 2) (1 = 360x360 pixels for image window)
6 default color table        (any IDL supports) 0 = B/W; 3 = red temperature
7 location                   (NSLS has special meaning = spooler switch)
8 spooler or print file      (c:\tmp\x1a at nsls - special !!)
9 default font for widget    (windows - Arial*14; unix - 6x10 or 8x13 ) NOT USED
10 Printer command line       (command line needed to transfer to printer)
                              NOT used if PRINTER is set on line 2
11 Command to view Help pdf  (get acrobat reader free from www.adobe.com)
12 command to run browser    (WIN: must be executable from MS-DOS window)
```

A. For WINDOWS machines

Color schemes greater than 256 work fine. In most cases, data is redisplayed, but occasionally you need to refresh a display to get a new color table to be implemented (With color tables above 256 - 16-bit, 32-bit or true color) IDL does not automatically update the display). *Reset colors* does this.

B. For Unix (X-windows) based systems

Since Unix is case sensitive, it is important that the case of the names of files be preserved. All the AXIS file names are strictly LOWER CASE. Some FTP programs force the file names to upper case which is a problem. If you have error messages saying a given pro file cannot be found please check the case of the filename and rename it to exactly what the IDL error reports.
Use ASCII transfer for all files (except *axis.bmp*, *axis.doc*, *axis.pdf* associated with the stack codes, which must be transferred in binary mode) when transferring code from Windows to UNIX or Macintosh systems to avoid problems with the differences in end-of-line coding in the 3 systems.

return to top menu

- the UNIX version of IDL only likes @bsif_com or @axis_com in the AXIS.PRO file. On the Linux-PC version the batch processing would not work when long names such as axis_common.pro are used for include files. Some examples of the COMMON include files are:

axis_com.pro	ax_peem_com.pro
bsif_com.pro	img_com.pro

in total, there are 39 common files in aXis2000 (as of Jan 2023)

Run time hints:
If you quit AXIS but stay in IDL, restart AXIS by executing ***axis2000*** *at the IDL command line.*

==

IDL Virtual machine version

return to top menu

This allows you to run aXis2000 without purchasing an IDL license :

1. download IDL from the download website, which, as of January 20203 is:
https://www.l3harrisgeospatial.com/Software-Technology/IDL
(NB you need to register in order to access the download the code (version 8.6 is ~2 Gb)

2. Download the aXis2000.zip package from unicorn.mcmaster.ca\aXis2000.html

3. Extract (unzip) all contents into c:\aXis2000 (Windows) or /Users/aXis2000 (Mac OS X)
(Make sure 'Use folder names' is turned on when you unzip)

4. set up a **desktop icon** for convenience. If it is your first time setting up aXis2000, you need to associate the aXis2000.sav file with the IDL runtime engine which on windows is
(IDL6.3) C:\RSI\IDL63\bin\bin.x86\idltrt.exe
(IDL8.6) C:\Program Files\Harris\IDL86\bin\bin.x86_64\idltrt.exe (for 64-bit computers)
or C:\Program Files\Harris\IDL86\bin\bin.x86\idltrt.exe (for 32-bit computers)
Once the association of *.sav files with IDLRT is made the desktop icon can be created by (Windows) right click on the desktop; click New~shortcut; browse to, then click on
C:\aXis2000\aXis2000.sav; click Next; define name of shortcup.

5. If you have other *.sav files you do NOT want to associate with IDLRT, you can define the desktop item as outlined at http://unicorn.mcmaster.ca/axis/aXis2000-IDLVM.html

Hints: **You must have all the source code files in the folder to make the Virtual Machine version work properly**

Note: If you use a folder other than c:\aXis2000 or /Users/aXis2000 there may be problems running the code.

Do NOT use folder (directory) names with blanks in the folder or file names for the source code or your data !!!
A common problem is to put aXis2000 in 'My Documents' or in 'Program Files' Note that both of those folder names contain a blank. You are warned.

return to top menu

Please provide feedback by email on how you would like to see this program evolve to be more useful to the X-ray and non X-ray SpectroMicroscopy community.

If you report a bug, I will try to fix it as soon as possible. It is most useful if you can send data file(s) which exhibit the bug, and, if you are running a full version of IDL, the IDL log window error report.

If you want to contribute with additional capabilities, please contact me. I am open to coding reasonable (preferably, small) extensions of aXis2000's capabilities.

Adam P. Hitchcock
Emeritus Professor of Chemistry & Chemical Biology aph@mcmaster.ca
Brockhouse Institute for Materials Research Tel: (905) 525-9140 ext. 24749
McMaster University Fax: (905) 521-2773
Hamilton, Ont. L8S 4M1 CANADA

FOR BEGINNING USERS, a number of **USEFUL DOCUMENTS** are available by download from http:/unicorn.mcmaster.ca/aXis2000.html

TUTORIALS – **SIMPLE** describes unit operations for manipulating X-ray microscopy data.
- **ADVANCED** - walks through a STXM data analysis using curve fitting

STACK_ANALYZE - manual to walk-through a typical stack analysis based on ALS STXM or PEEM data. *[Warning – this document is quite dated and oriented to the analysis of PEEM data measured at ALS_PEEM2 (on the 7.1 bend magnet beamline]*
STACK_FIT - manual to walk through stack fit from a prepared *.ncb aligned image sequence.

aXis2000 guides to specific procedures

http://unicorn.mcmaster.ca/axis/aXis2000-procedures.html

has a lot of "work flow" or single step guides to help new users of Xis2000

FOR PROGRAMMERS – SOME USEFUL ADD-INS (in aXis2000.zip)

Content.lst - lists all AXIS routines with a one-line description

Run **ax_make_html** at the IDL prompt. This will extract the header documentation from all the files to generate an html document with internal hyper-links etc. (there is an HTML file in the aXis2000 distribution with this output)

TEST-DATA – a set of files that can be used to test each of the aXis2000 routines

(*contact Adam Hitchcock to obtain*)

GLOSSARY of terms associated with aXis2000

return to top menu

***axis_XXX.ini* -** (XXX=win, unix, macOS) text file containing default parameters for aXis2000. Edit to adapt to your system

component maps - spatial distributions of a chemical species, which can be generated from multiple images (selected energies or a full image sequence) using *stacks~maps~SVD* or *stacks~maps~stack fit*)

dark count – signal (image or spectral) recorded without X-rays. This is detector specific (e.g. CCD camera leakage, light leakage into PMT etc) and must be subtracted from real signal before any data processing involving ratios (yield or absorption determinations).

IDL - Interactive Data Language is a high performance scientific computing platform, optimized for arbitrary sized multi-dimension array processing and image display. IDL was originally developed by David Stern and his team from Research Systems Inc. Wikipedia has a short history of IDL. The current owner is L3Harris

IDL Virtual Machine – a version of IDL which allows execution of compiled IDL script. This can be downloaded for free from L3Harris

IDL widget - a graphical user interface with pre-programmed data manipulation or other capabilities written in Interactive Data Language.

ini file - a text file which contains values of parameters used to customize aXis2000 for your environment

intensity profile - Plot of pixel value across a line defined on an image. Rectilinear profiles are available automatically, updated either on each mouse move, or after a right click, depending on the setting of the continuous lineouts control.

line scan spectra – a 2-d data set in which the intensity along a line is recorded at a series of photon energies. aXis2000 displays the data with energy along the horizontal axis and position along the line as the vertical axis. Thus horizontal lineouts (lower side panel) display spectra at a point; vertical lineouts (left side panel) display the contrast along the physical line at a given photon energy.

masked image - Output of *Images~generate_mask.* It is an image consisting of only 0 or 1 value pixeks, based on whether the original pixel was above (1) or below (0) a user-defined threshold. This, along with *Images~multiply buffers* and *Images~average pixels*, is useful to evaluate the intensity in a selected region of a component map.

netCDF – a standard binary scientific data format. The data is read and written by platform dependent routines provided by a standards body. Used for NSLS STXM data.

OD – optical density, or absorbance. In a transmission measurement (STXM, TXM) the recorded data is the intensity of the transmitted. X-rays. It is converted to OD by – ln(I/Io) where I is the signal transmitted through the sample and Io is the signal without the sample in place.

self defining files - A structured, ascii format in which a syntax is used to allow a single program to read (and write) complex data structures without prior knowledge of the data structure. Read and write routines are available in C++ and IDL.

Stack – a set of images at a sequence of energies (in other spectromicroscopies, sometimes called a ‘map’)

return to top menu

ZSTACK
Data Analysis for Hyperspectral X-ray Microscopy Imaging
v2.1 (5 jan 2001)

ZSTACK is a suite of IDL procedures for alignment and analysis of a series of x-ray microscopy images that have been acquired at different x-ray energies. It operates on Windows, Macintosh, and UNIX platforms using IDL v5.0 or later.

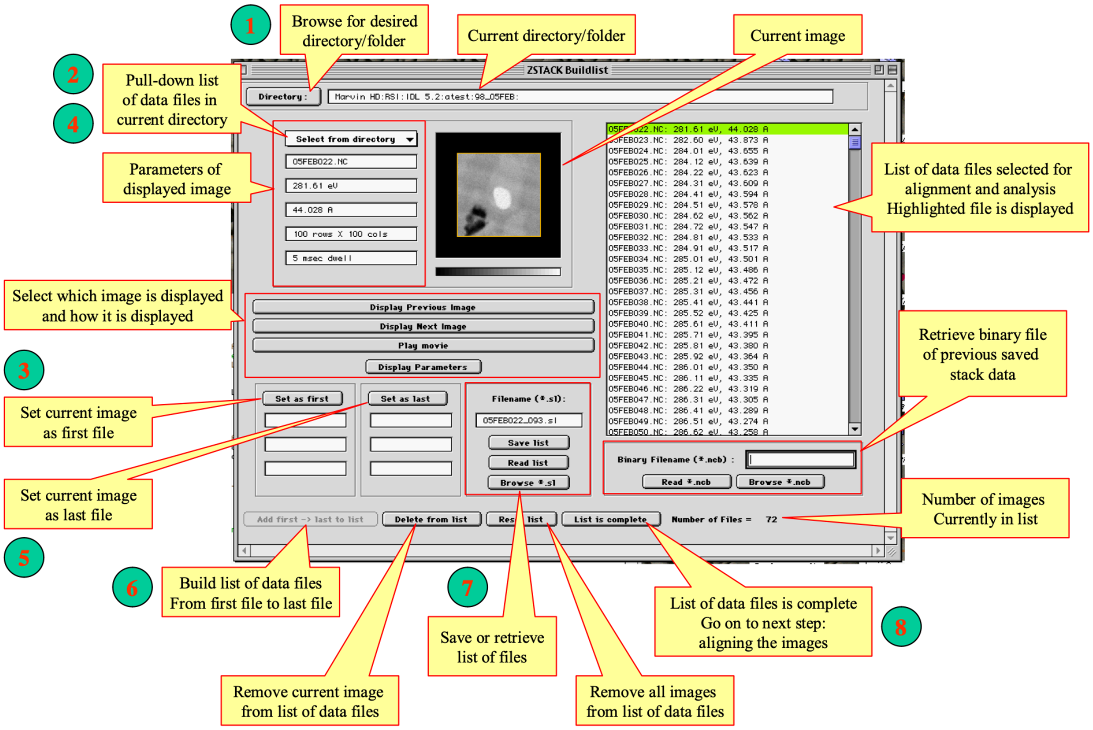
ZSTACK was written originally to satisfy my own needs for analysis of STXM spectral image stacks. The original version was based upon the original STACK code developed by Chris Jacobsen (SUNY - Stony Brook). It has expanded considerably beyond the initial scope, largely due to my analysis needs and feedback from fellow users.

In a collaborative effort with Adam Hitchcock (McMaster University), ZSTACK has been bundled into his AXIS data analysis package. ZSTACK can also be used independently of AXIS.

The development of ZSTACK has been an evolutionary process. Please notify me of any bugs, problems, comments, or suggestions for improvements or new features.

Carl G. Zimba
Photons UnLimited
84 Oxford Street
Arlington, MA
photons@mediaone.net

ZSTACK Buildlist



Browse for desired directory/folder

Current directory/folder

Current image

Pull-down list of data files in current directory

Parameters of displayed image

Select which image is displayed and how it is displayed

Set current image as first file

Set current image as last file

Build list of data files From first file to last file

Remove current image from list of data files

Save or retrieve list of files

Remove all images from list of data files

List of data files is complete Go on to next step: aligning the images

List of data files selected for alignment and analysis Highlighted file is displayed

Retrieve binary file of previous saved stack data

Number of images Currently in list

Directory : Marvin HD:RSI:IDL 5.2:atest:98_05FEB:

05FEB022.NC
281.61 eV
44.028 A
100 rows X 100 cols
5 msec dwell

```
05FEB022.NC: 281.61 eV, 44.028 A
05FEB023.NC: 282.60 eV, 43.873 A
05FEB024.NC: 284.01 eV, 43.655 A
05FEB025.NC: 284.12 eV, 43.639 A
05FEB026.NC: 284.22 eV, 43.623 A
05FEB027.NC: 284.31 eV, 43.609 A
05FEB028.NC: 284.41 eV, 43.594 A
05FEB029.NC: 284.51 eV, 43.578 A
05FEB030.NC: 284.62 eV, 43.562 A
05FEB031.NC: 284.72 eV, 43.547 A
05FEB032.NC: 284.81 eV, 43.533 A
05FEB033.NC: 284.91 eV, 43.517 A
05FEB034.NC: 285.01 eV, 43.501 A
05FEB035.NC: 285.12 eV, 43.486 A
05FEB036.NC: 285.21 eV, 43.472 A
05FEB037.NC: 285.31 eV, 43.456 A
05FEB038.NC: 285.41 eV, 43.441 A
05FEB039.NC: 285.52 eV, 43.425 A
05FEB040.NC: 285.61 eV, 43.411 A
05FEB041.NC: 285.71 eV, 43.395 A
05FEB042.NC: 285.81 eV, 43.380 A
05FEB043.NC: 285.92 eV, 43.364 A
05FEB044.NC: 286.01 eV, 43.350 A
05FEB045.NC: 286.11 eV, 43.335 A
05FEB046.NC: 286.22 eV, 43.319 A
05FEB047.NC: 286.31 eV, 43.305 A
05FEB048.NC: 286.41 eV, 43.289 A
05FEB049.NC: 286.51 eV, 43.274 A
05FEB050.NC: 286.62 eV, 43.258 A
```

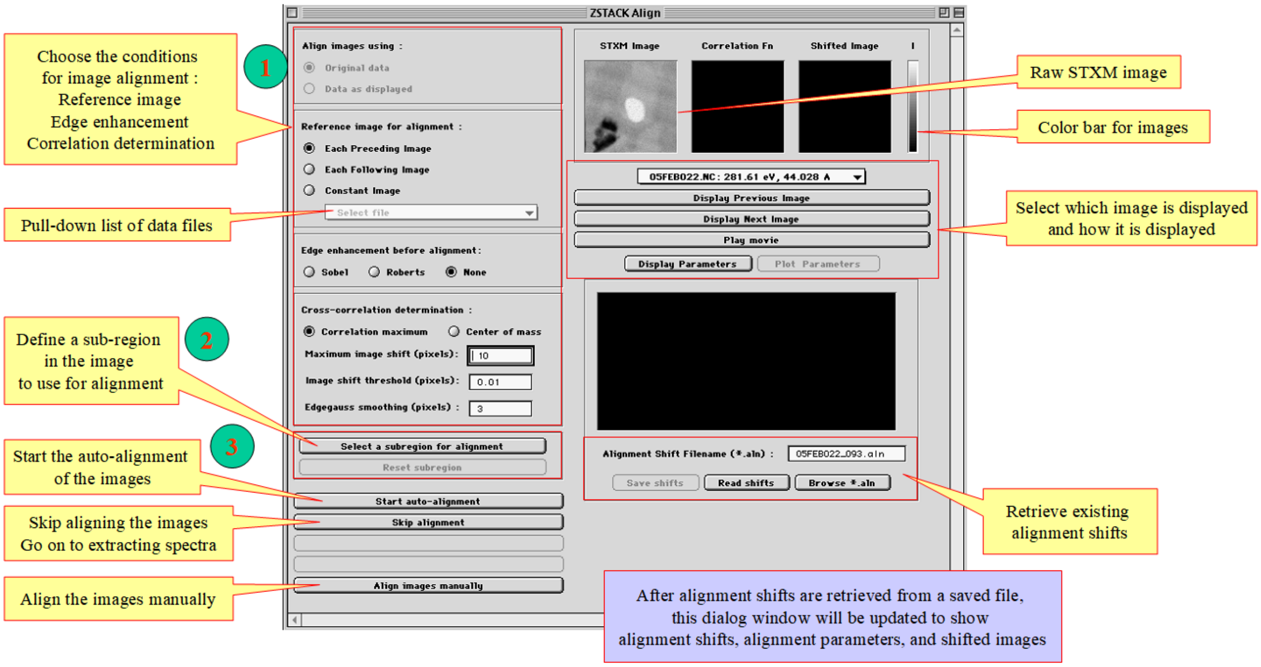
Filename (*.sl): 05FEB022_093.sl

Binary Filename (*.ncb) :

Number of Files = 72

ZSTACK Align

(before alignment)



Choose the conditions for image alignment : Reference image Edge enhancement Correlation determination

Pull-down list of data files

Define a sub-region in the image to use for alignment

Start the auto-alignment of the images

Skip aligning the images Go on to extracting spectra

Align the images manually

Raw STXM image

Color bar for images

Select which image is displayed and how it is displayed

Retrieve existing alignment shifts

Align images using :
◉ Original data
○ Data as displayed

Reference image for alignment :
◉ Each Preceding Image
○ Each Following Image
○ Constant Image
Select file

Edge enhancement before alignment:
○ Sobel ○ Roberts ◉ None

Cross-correlation determination :
◉ Correlation maximum ○ Center of mass
Maximum image shift (pixels): 10
Image shift threshold (pixels): 0.01
Edgegauss smoothing (pixels) : 3

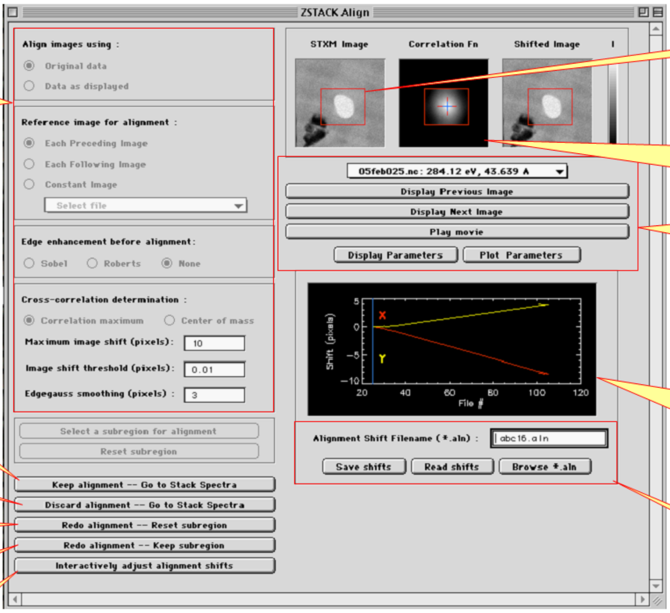
05FEB022.NC : 281.61 eV, 44.028 A

Alignment Shift Filename (*.aln) : 05FEB022_093.aln

After alignment shifts are retrieved from a saved file, this dialog window will be updated to show alignment shifts, alignment parameters, and shifted images

ZSTACK Align

(after alignment)



ZSTACK Align

Align images using :
● Original data
○ Data as displayed

Reference image for alignment :
● Each Preceding Image
○ Each Following Image
○ Constant Image
Select file

Edge enhancement before alignment:
○ Sobel ○ Roberts ● None

Cross-correlation determination :
● Correlation maximum ○ Center of mass
Maximum image shift (pixels): 10
Image shift threshold (pixels): 0.01
Edgegauss smoothing (pixels) : 3

05feb025.nc: 284.12 eV, 43.639 A

Alignment Shift Filename (*.aln) : abc16.aln

This now displays the conditions which were used to obtain current alignment

Keep current alignment

Skip the alignment

Redo the alignment erasing the subregion

Redo the alignment keeping the subregion

Manually adjust the alignment

Subregion used for alignment

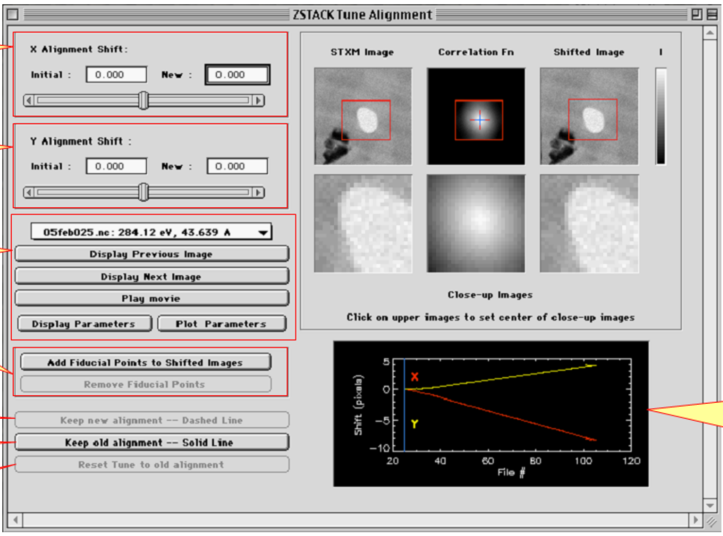
Correlation function crosshairs:
red for center of corr'n image
blue for maximum position
Red box is edge of corr'n image

Select which image is displayed and how it is displayed

Plot of alignment shifts
Cursor position corresponds to current image and can be moved by clicking mouse in the plot

Save or retrieve alignment shifts

ZSTACK Tune



ZSTACK Tune Alignment

X Alignment Shift:
Initial : 0.000 New : 0.000

Y Alignment Shift :
Initial : 0.000 New : 0.000

05feb025.nc: 284.12 eV, 43.639 A

Adjust the x shift using either the "New" text box or the slide bar

Adjust the y shift using either the "New" text box or the slide bar

Select which image is displayed and how it is displayed

Add fiducial points or shapes to close-up of shifted images

Keep new alignment

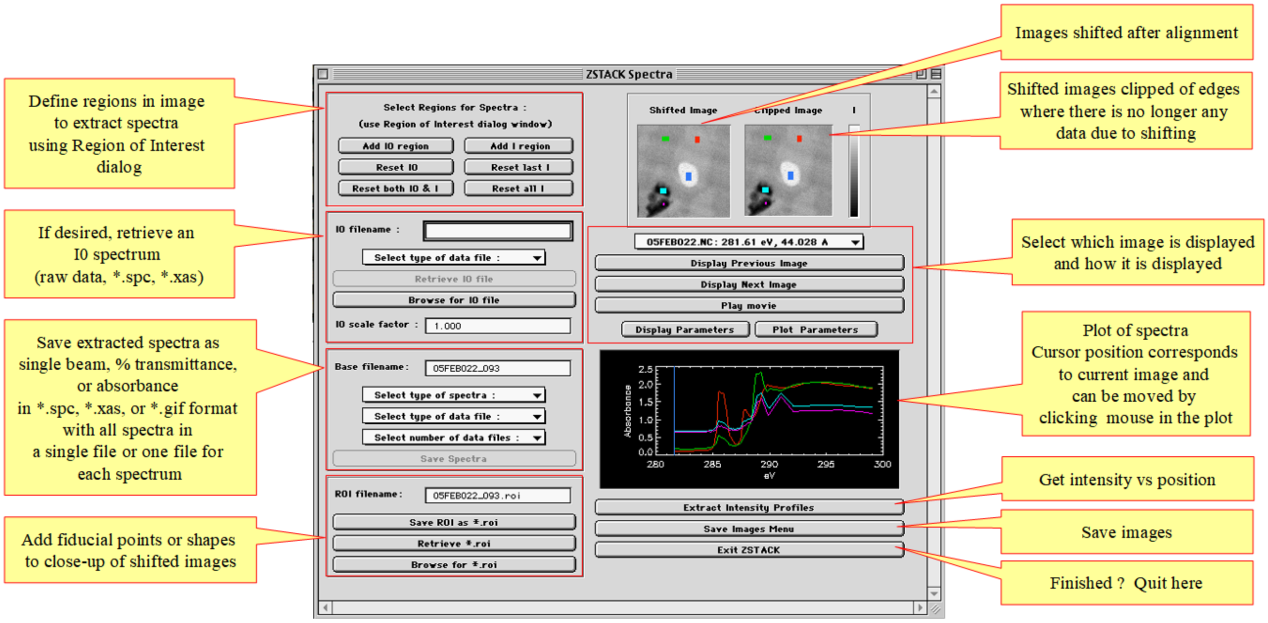
Keep old alignment

Reset to old alignment

Plot of alignment shifts
Cursor position corresponds to current image and can be moved by clicking mouse in the plot

By placing some fiducial points in the close-up of the shifted images, the quality of the alignment can be inspected while playing theimages as a movie

ZSTACK Spectra



I0 scale factor : 1.000

Base filename : 05FEB022_093

ROI filename : 05FEB022_093.roi

05FEB022.NC : 281.61 eV, 44.028 A

Define regions in image to extract spectra using Region of Interest dialog

If desired, retrieve an I0 spectrum (raw data, *.spc, *.xas)

Save extracted spectra as single beam, % transmittance, or absorbance in *.spc, *.xas, or *.gif format with all spectra in a single file or one file for each spectrum

Add fiducial points or shapes to close-up of shifted images

Images shifted after alignment

Shifted images clipped of edges where there is no longer any data due to shifting

Select which image is displayed and how it is displayed

Plot of spectra
Cursor position corresponds to current image and can be moved by clicking mouse in the plot

Get intensity vs position

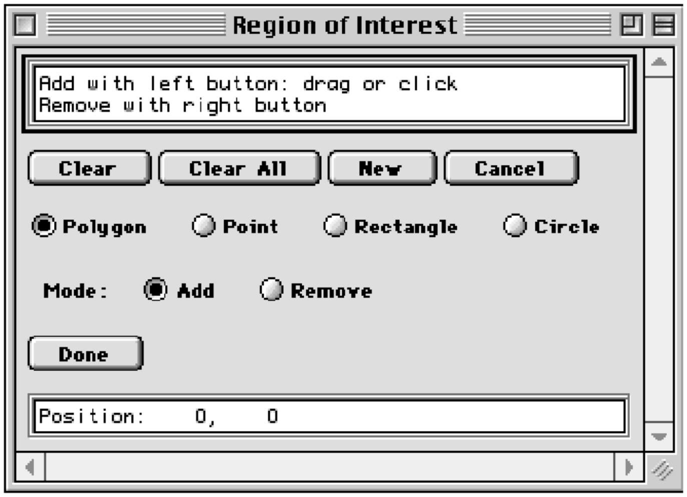
Save images

Finished ? Quit here

Fourteen separate regions of interest can be specified, yielding fourteen I spectra.
In addition, a region of interest can be specified in an empty region of the sample to yield an I0 spectrum.

Color of the region of interest in the images corresponds to the same color spectrum

Region of Interest



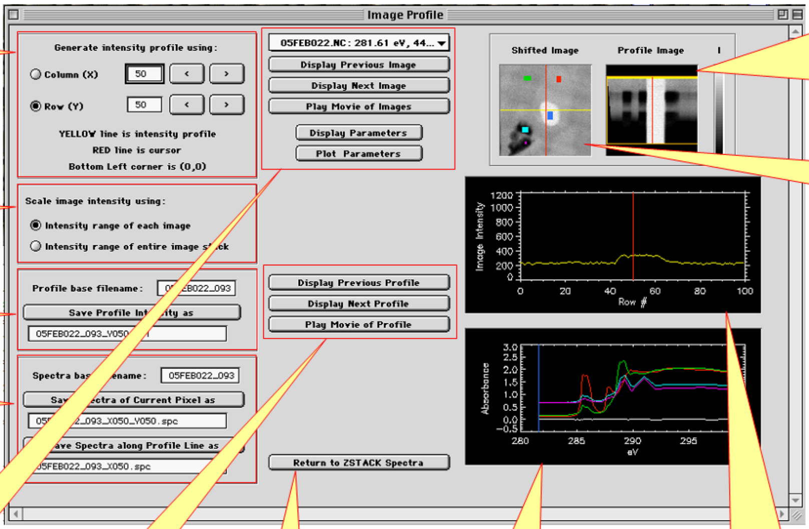
Add with left button: drag or click
Remove with right button

● Polygon ○ Point ○ Rectangle ○ Circle

Mode : ● Add ○ Remove

Position: 0, 0

ZSTACK Profile



Generate intensity profile using:

○ Column (X) 50

● Row (Y) 50

YELLOW line is intensity profile

RED line is cursor

Bottom Left corner is (0,0)

Scale image intensity using:

● Intensity range of each image

○ Intensity range of entire image stack

Profile base filename: 05FEB022_093

05FEB022_093_Y050

Spectra base filename: 05FEB022_093

05FEB022_093_X050_Y050.spc

05FEB022_093_X050.spc

05FEB022.NC: 281.61 eV, 44...

Map of intensity as a function of x-ray energy and position along the profile axis

Cursor position and x-ray energy can be moved by clicking mouse in the plot

Adjust position of cursor and profile axis

Save intensity profile

Save spectra of current pixel

Save spectra of each pixel along profile axis

Cursor position can be moved by clicking mouse in the image

Color of the region of interest in the images corresponds to the same color spectrum

The profile axis is the yellow line in both the shifted image and the profile image. The intensity along this line is plotted as the profile intensity.

Select which image is displayed and how it is displayed

Select which profile is displayed and how it is displayed

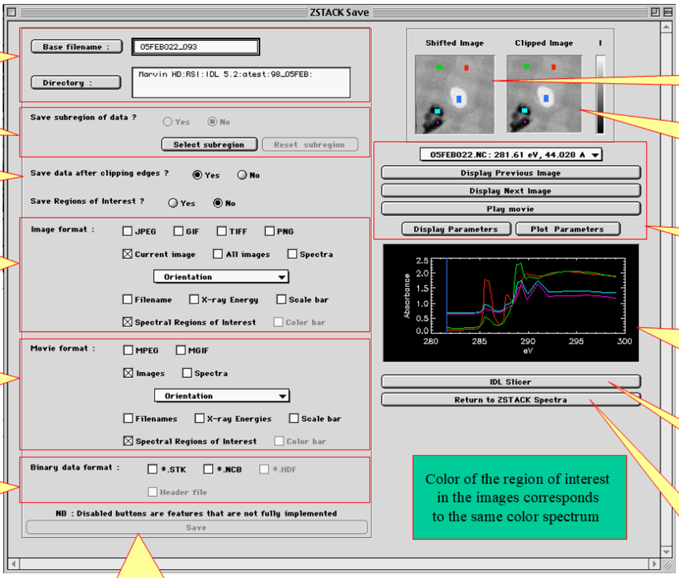
Finished ? Return to ZSTACK Spectra

Plot of spectra Cursor position corresponds to current image and can be moved by clicking mouse in the plot

Plot of intensity along profile axis Cursor position corresponds to current position along profile axis and can be moved by clicking mouse in the plot

Red cursor in profile plot corresponds to position of red line in both the shifted image and the profile image

ZSTACK Save



Base filename : 05FEB022_093

Directory : Marvin HD:RSI:IDL 5.2:atest:98_05FEB:

Save subregion of data ? ○ Yes ● No

Save data after clipping edges ? ● Yes ○ No

Save Regions of Interest ? ○ Yes ● No

Image format : □ JPEG □ GIF □ TIFF □ PNG

☒ Current image □ All images □ Spectra

Orientation

□ Filename □ X-ray Energy □ Scale bar

☒ Spectral Regions of Interest □ Color bar

Movie format : □ MPEG □ MGIF

☒ Images □ Spectra

Orientation

□ Filenames □ X-ray Energies □ Scale bar

☒ Spectral Regions of Interest □ Color bar

Binary data format : □ *.STK □ *.NCB □ *.HDF

□ Header file

NB : Disabled buttons are features that are not fully implemented

05FEB022.NC: 281.61 eV, 44.028 A

Color of the region of interest in the images corresponds to the same color spectrum

Specify base filename and directory to save files

Select a subregion of image to save

Trim images to eliminate edges clipped off during alignment

Save images in a variety of formats, optionally including spectra, roi, and legend info

Save images as a movie, optionally including spectra, roi, and legend info

Save image stack as a binary data file

Images shifted after alignment

Shifted images trimmed of edges clipped during alignment

Select which image is displayed and how it is displayed

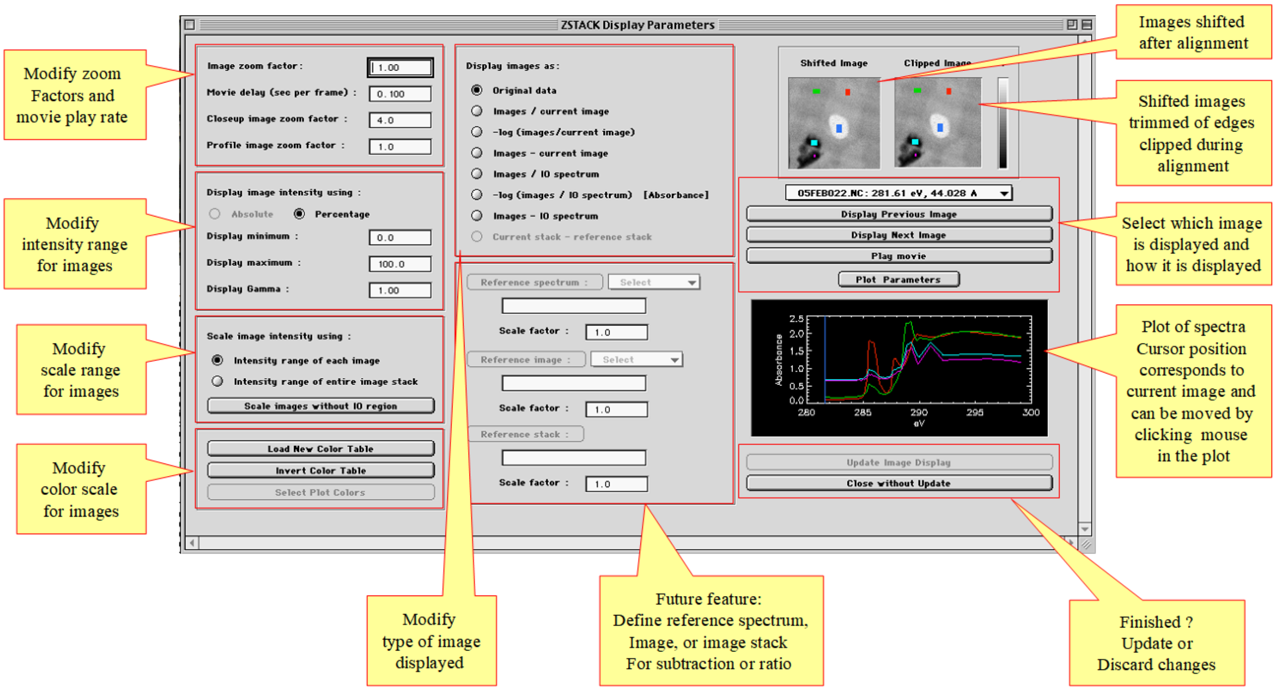
Plot of spectra Cursor position corresponds to current image and can be moved by clicking mouse in the plot

Process data using IDL Slicer3

After specifying ALL the options above, Click here to save everything (images, movies, binaries) at one time.

Finished ? Return to ZSTACK Spectra

ZSTACK Display



ZSTACK Display Parameters

Image zoom factor : 1.00
Movie delay (sec per frame) : 0.100
Closeup image zoom factor : 4.0
Profile image zoom factor : 1.0

Display image intensity using :
○ Absolute ● Percentage
Display minimum : 0.0
Display maximum : 100.0
Display Gamma : 1.00

Scale image intensity using :
● Intensity range of each image
○ Intensity range of entire image stack

Display images as :
● Original data
○ Images / current image
○ -log (images/current image)
○ Images - current image
○ Images / I0 spectrum
○ -log (images / I0 spectrum) [Absorbance]
○ Images - I0 spectrum
○ Current stack - reference stack

Reference spectrum : Select
Scale factor : 1.0
Reference image : Select
Scale factor : 1.0
Reference stack :
Scale factor : 1.0

05FEB022.NC : 281.61 eV, 44.028 A

Modify zoom Factors and movie play rate

Modify intensity range for images

Modify scale range for images

Modify color scale for images

Modify type of image displayed

Future feature: Define reference spectrum, Image, or image stack For subtraction or ratio

Images shifted after alignment

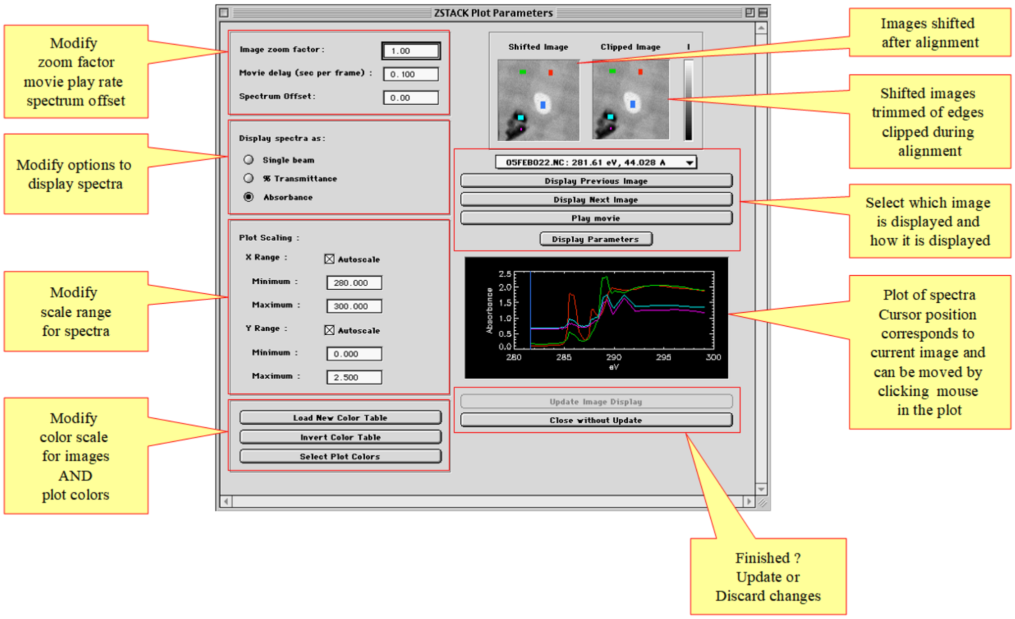
Shifted images trimmed of edges clipped during alignment

Select which image is displayed and how it is displayed

Plot of spectra Cursor position corresponds to current image and can be moved by clicking mouse in the plot

Finished ? Update or Discard changes

ZSTACK Plot



ZSTACK Plot Parameters

Image zoom factor : 1.00
Movie delay (sec per frame) : 0.100
Spectrum Offset : 0.00

Display spectra as :
○ Single beam
○ % Transmittance
● Absorbance

Plot Scaling :
X Range : ☒ Autoscale
Minimum : 280.000
Maximum : 300.000
Y Range : ☒ Autoscale
Minimum : 0.000
Maximum : 2.500

05FEB022.NC : 281.61 eV, 44.028 A

Modify zoom factor movie play rate spectrum offset

Modify options to display spectra

Modify scale range for spectra

Modify color scale for images AND plot colors

Images shifted after alignment

Shifted images trimmed of edges clipped during alignment

Select which image is displayed and how it is displayed

Plot of spectra Cursor position corresponds to current image and can be moved by clicking mouse in the plot

Finished ? Update or Discard changes