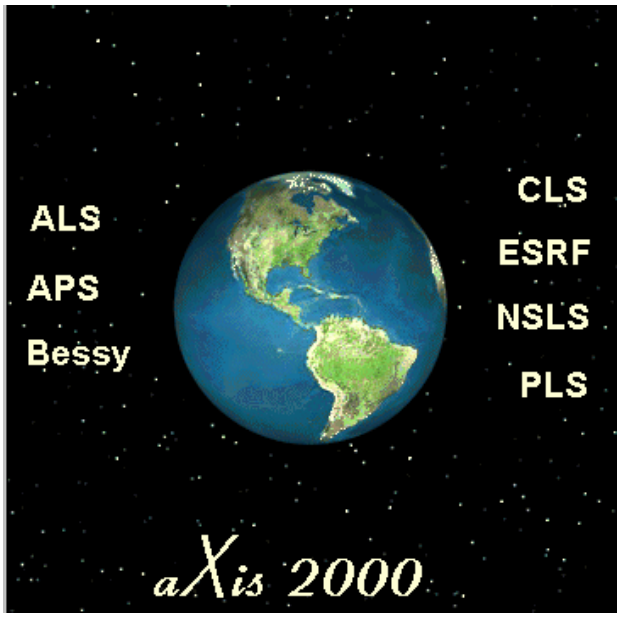
aXis 2000

Analysis of X-ray microscopy Images and Spectra (25-Mar-2005)



AXIS2000 - **A**nalysis of **X**-ray microscopy **I**mages and **S**pectra - is an IDL widget for viewing, comparing and processing X-ray microscopy images and spectra. IDL stands for Interactive Data Language, a product of Research Systems Inc (RSI). It is based on scripts developed by a large number of people at the NSLS and ALS X-ray microscopy facilities, by Carl Zimba (Photons Unlimited) and by Adam & Peter Hitchcock. It operates on Windows (WIN), Unix (X) and Macintosh (MAC) versions of IDL. In 2002 the code was extensively adapted to improve cross-system performance, especially for Macintosh OS. Currently it runs fully with IDL 6.1 and most features work with IDL5.x. If you run IDL 4, you need AXIS version 1.6a or earlier.

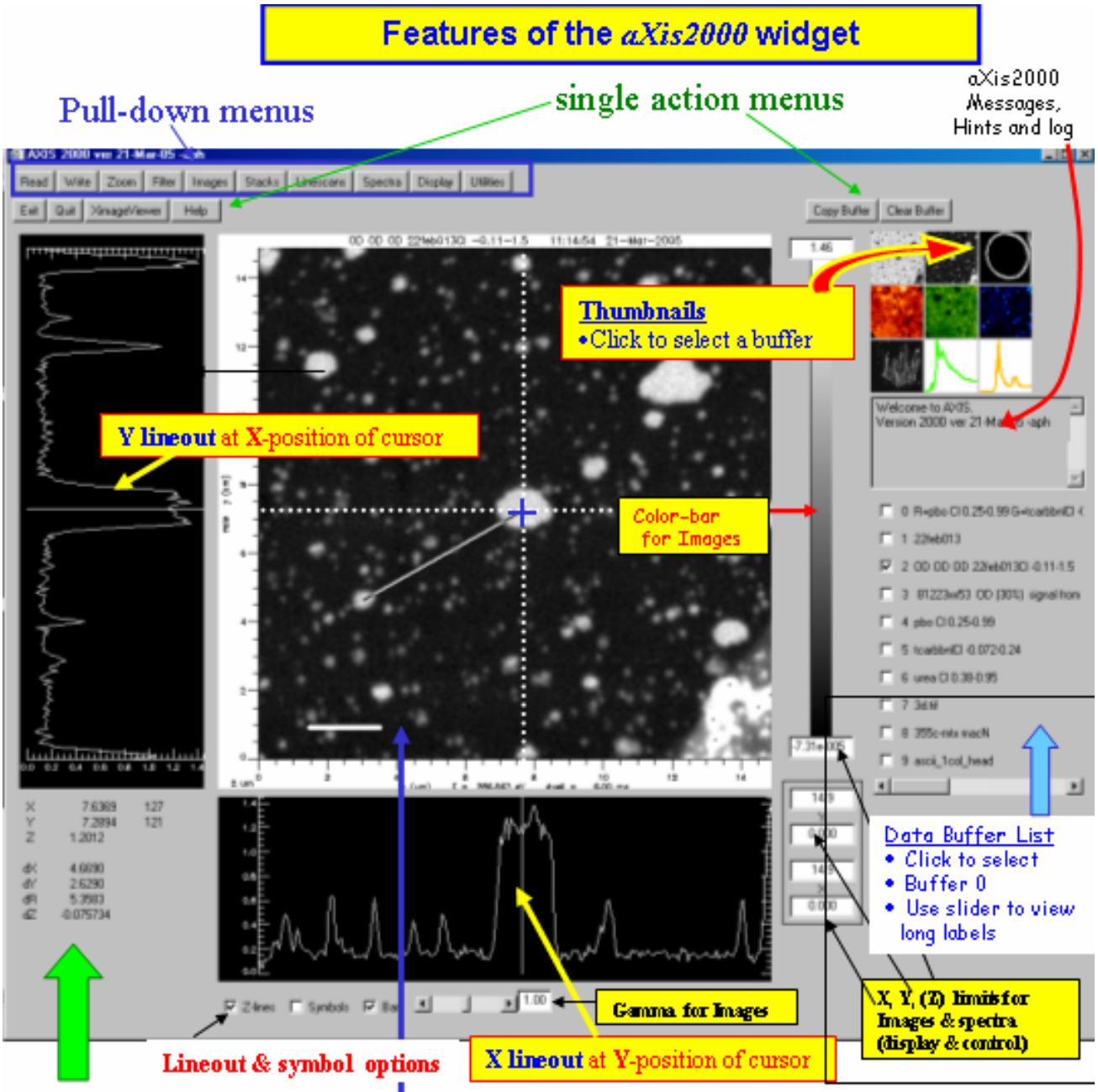
Since May-04 a compiled version **(aXis2000.sav**) for use with IDL Virtual Machine has been available. This allows access to the power of aXis2000 without needing to purchase an IDL license. The aXis2000.sav file distributed from the aXis2000 web site only works with the IDL6.1 Virtual Machine code, so if you want to access the latest improvements to aXis2000, please load the IDL 6.1 virtual machine code from the RSI web site (rsinc.com)

I would appreciate it if you would notify me by email (aph@mcmaster.ca) about problems with the code or with suggestions for improvements. If you make extensions or corrections, I would appreciate receiving a copy of your code revisions to incorporate in future versions.

I thank all the people who have written scripts that went into this. ***Carl Zimba*** *(Photons Unlimited) who supplied ZSTACK and has extensively improved the package overall; my son,* ***Peter*** *who helped set up the basic widget structure; Eli Rotenberg, Jonathan Denlinger, Stefano Cerasari, Tolek Tyliszczak, Andy Smith, and many others. SPECIAL thanks to* ***Chris Jacobsen*** *(Stony Brook, nsls) for sharing his STACK codes,* ***Rick Kneedler****, for providing the basis for the* **stack-fit** *routine, and* ***Billy Loo*** *(UCSF) for providing SF, the Henke mass absorption routine.*

NEW FEATURES in 25-Mar-05 version since May-02 version are HIGHLIGHTED

TO START *aXis2000*: *after installing aXis2000* (see end of this file)
Windows and Mac OS:
Start **IDL** ;
If you have set the Preferences (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**



Cursors

(X,Y,Z) – at cursor

(dX,dY,dZ) – change over line (images) or between cursors (spectra)
dR – distance along line (images only)

Main Image

- Displays currently selected image or selected spectrum (or group of spectra, if Spectra~Overplot used)
- Size of AXIS can be adjusted from 0.5 to 2.0 of its nominal size (360x360 pixels in Main Image) by size parameter in **axis.ini**

Mouse

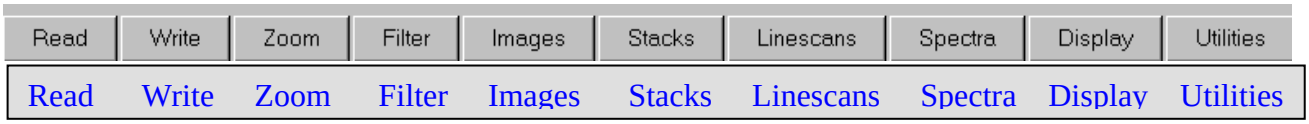
- First click – cursor and lineout; arms line generator
- Second click – draws and documents line (image) ;
- reports difference in cursors (spectra)
- Third click – clears line and cursor information

Hard copy: Printing the displayed signals is achieved by using *Utilities~print*. aXis2000 prints via the IDL Printer virtual device, or by writing Postscript, or PCL-5 format files and copying to appropriate printer, depending on settings in the **axis.ini** file. The system-specific print command is defined in the AXIS.INI initialization file. The user can set this and other parameters using the *Utilities~preferences* command in AXIS.

Notation conventions used in this reference manual:

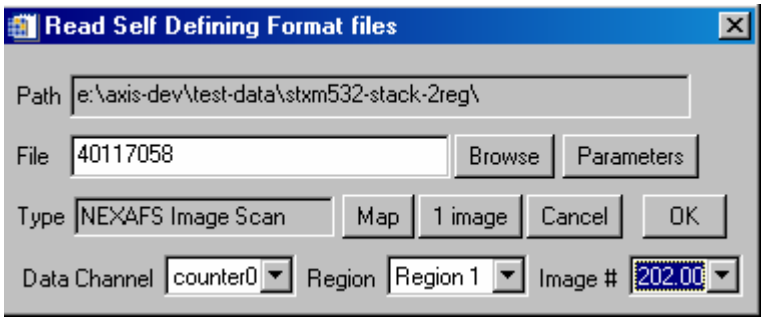
BOLD indicates a pull-down menu command;
TOP LEVEL items are bold, underlined, and light blue highlighted
Utilities~print~annotated is a third level pull-down menu under Utilities first level button
Linescans indicates an in-manual hyperlink

TOP ROW PULL DOWN MENUS



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

ALS-STXM - all types of self defining files containing data from 5.3.2 STXM 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)



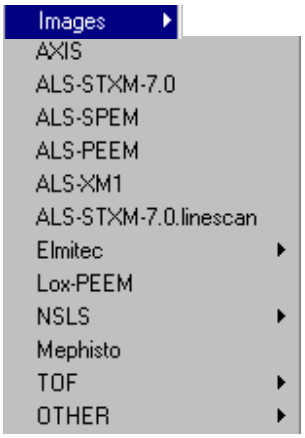
All data generated by the 5.3.2 STXM is described by a self-defining header file (*.hdr) which contains the information needed to identify the type of data contained in the associated data files. After selecting a file, the widget indicated above can be used to :

Browse – to select the file (or edit the filename box)
Parameters - list the header, which contains all microscope parameters
Data channel - select from pull down list, generated from header
Region - select from pull down list, generated from header
Image# - select specific image of multi-image NEXAFS Image scan
Map – (only for NEXAFS Image Scan) – convert a 2-imagestack into a OD difference map
1 image – (only for NEXAFS Image Scan) read only the selected image
Cancel – dismiss the ALS STXM read in widget

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

READ (*continued*)

Images

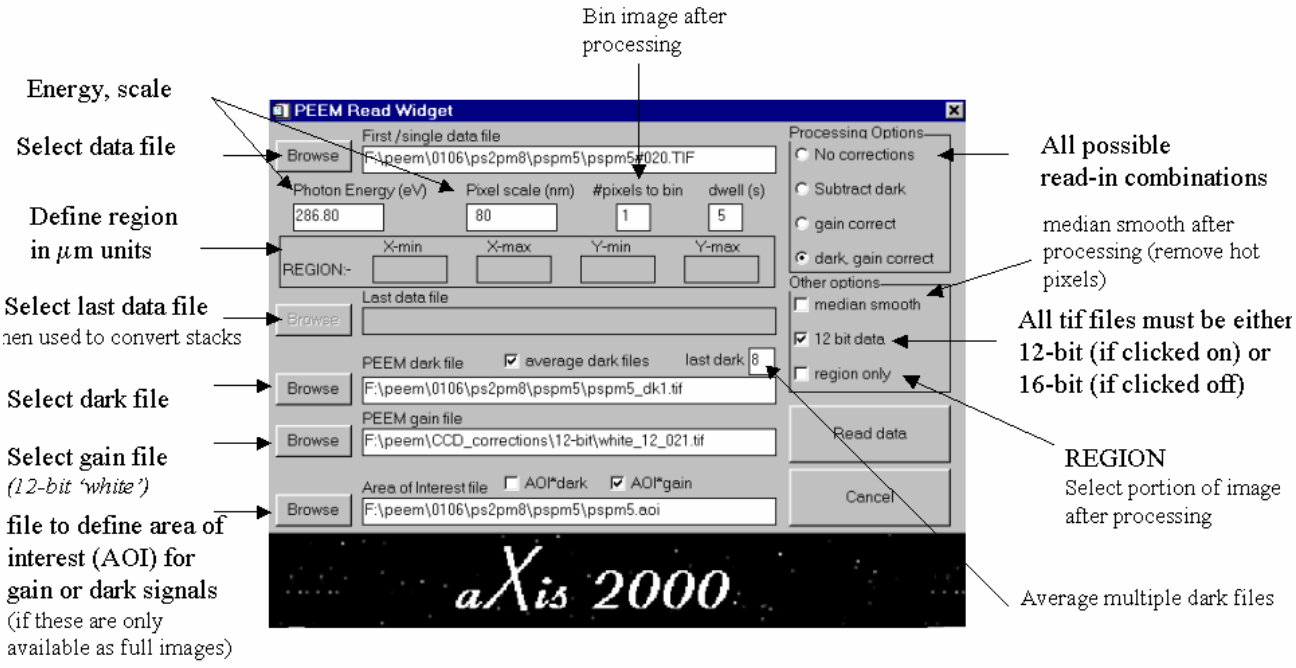


AXIS – images written in aXis2000 **binary** format (*.axb)
This is useful for saving derived results and is the required input for some routines

ALS-STXM-7.0 image files (ascii) – data channels:
- im0 = OSA signal (Io);
- im1 = transmitted signal;
- im2 = other (e.g TEY, luminescence, etc)

ALS-SPEM – read in images recorded in XPS or NEXAFS mode using the ALS SPEM

ALS-PEEM – 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)



ALS-XM1 – read in images (*.tif) from ALS XM1 full field transmission X-ray microscope

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

NSLS – read netCDF (bsif implementation) format image files from Brookhaven X1A STXM
- **old (*.nc)** - format of old STXM files (also used for modifying stacks)
- **stxmIV** - format of new STXM (netCDF but different variables)
- **cryo** - format for cryo-STXM (*.sxm)

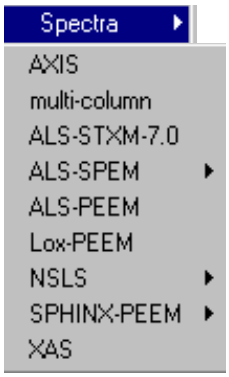
Mephisto – read PEEM data from Mephisto (old binary 512x512 array from a CCD)

TOF – read McMaster time-of-flight 2-d data in Z-matrix format
- **all (pTa)b –** 2-d as written by the pTa acquisition program
- **img –** Z-matrix data

READ (*continued*)

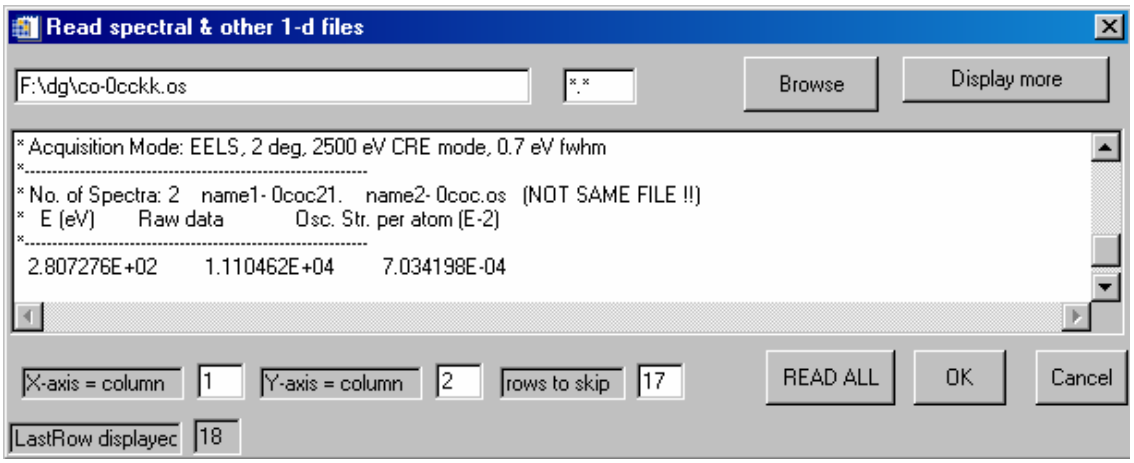
OTHER - standard graphics formats (**BMP**, **GIF**, **PNG, TIF**).
TIF format handles 1-channel and 3-channel (RBG) formats
For each image format one can read the image as an **image** (un-modifiable)
or as **data** (using numerical values for z-value, and pixel indices as (x,y)-values)

Spectra (READ)



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**)

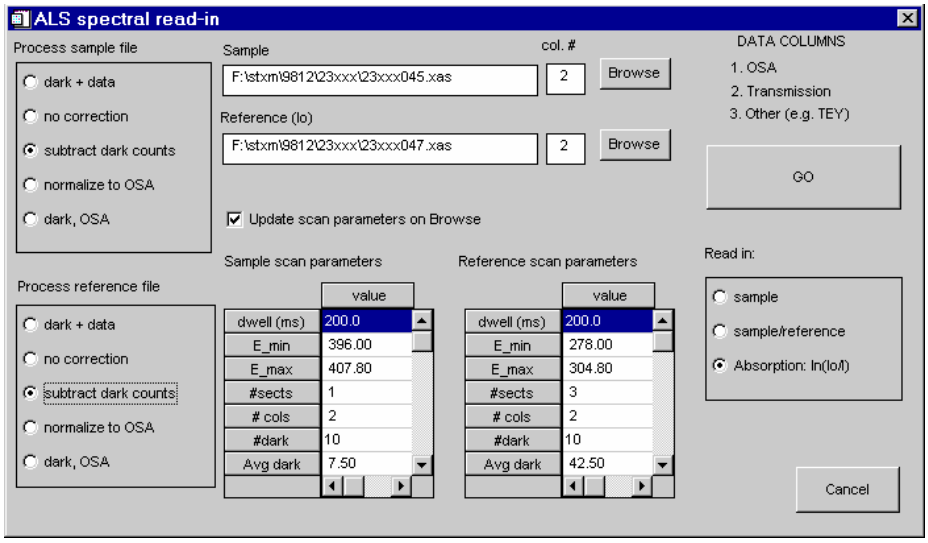
multi-column – guided read-in of any ascii data file, with ability to skip lines, choose columns, multi-column read-in, etc



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

READ - Spectra (*continued*)

ALS – STXM-7.0 read in files from ALS BL 7.0 STXM (HISTORICAL)
using the following widget to select spectral processing options.



Select file name for the sample (and reference if you wish to compute absorption from transmission data, or to normalize yield data); set the data column and process options for the sample and reference; set the read in mode (data, ratio, absorption); then press GO.

For example, to read only the data from column 2, with removal of the dark count signal from the detector which is pre- and post-appended to the spectral data, but without dark count subtraction, one would chose:
1. Process sample file: no correction
2. Sample: file name (select via browse or simply type in the name - useful if working through a sequence of files);
3. col # 2 ;
4. Read in: sample;
5. GO

ALS-SPEM – (ALS BL7.0)
- **XPS** – multi-region XPS spectra recorded with Phi electron spectrometer
- **NEXAFS** – sample current NEXAFS recorded with BL 7.0 SPEM

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

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

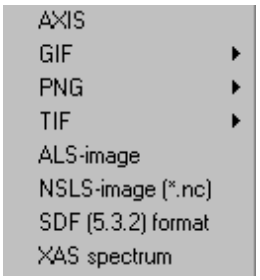
NSLS – read spectral data files in formats used at X1A
- **nc** - as-recorded (BSIF binary)
- **stxmIV** – stxm IV format (nc, but different than old-stxm)
- **ascii** - converted spectral format (generated by an X1A nsls utility)
- **map** - mapper style files (e.g. stack alignment shifts)

SPHINX-PEEM - – read in multi-column spectra written by Sphinx (*.dat files)
- **1 –** read 1 column
- **all** – read all columns

READ - Spectra (*continued*)

XAS – XAS format with user settable header (used for reference spectra, ZSTACK and NSLS)

WRITE to store images and spectra for later use



AXIS - automatically selects 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)**
- if you wish to write images in ascii, use **utilties~Write image ascii**

GIF – writes the 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)

PNG – writes the 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 the 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-image – writes ALS BL7 image format **(*.img)**

NSLS-image – writes NSLS netCDF image format **(*.nc)**

SDF (5.3.2) format – write self defining file format [*.hdr, *.xim (*.xsp) files for images (spectra)]

XAS spectrum - write spectrum in XAS format with optional definition of detailed header

---

ZOOM

(**for** images **and** spectra)

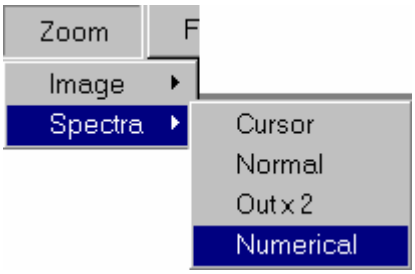


Image *(NB –since expanding the scale is performed by cutting out the region of interest, the cursor-cut and numerical-cut commands can also be used to extract sub images)*

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 continuous 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 - numerical selection of range. Data is cut from the displayed image and shifted to the working buffer (0).

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

Numerical - select numerical limits

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 (*)

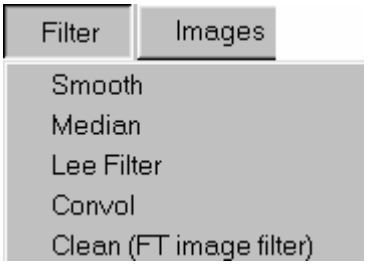
Z: Z-scale of images are indicated at the bottom and top of the color bar

.

* 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.**

FILTER (images and spectra)



5 types of smoothing with selectable parameters. The first four apply to BOTH images and spectra, but **Clean** works only with images.

Smooth – Boxcar average over n-points (right to edge of images)

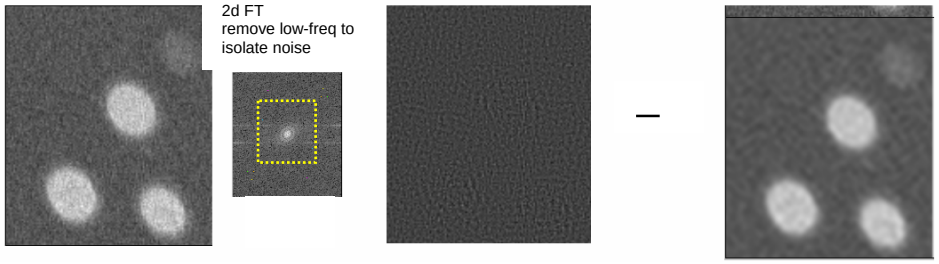
Median – n-point Savitsky-Golay averaging

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

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

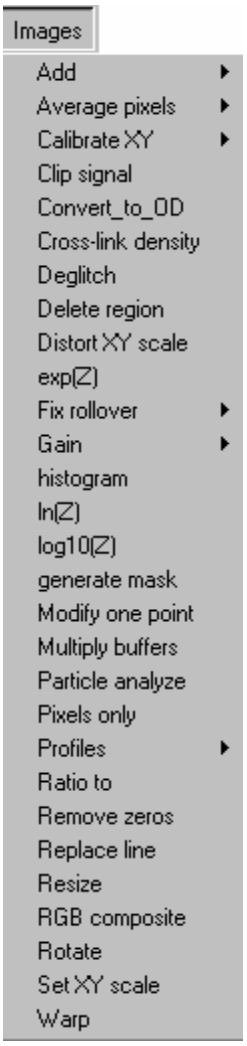
Clean (2d Fourier transform image filter) - 2d-FT filter. The FT is displayed on a 1:1 pixel format. Use the rubber-band style cursor (click, drag mouse, click a second time) to define the data in the complementary frequency domain to delete. The reverse transform of all data but the rejected data is displayed in buffer 9. The centre of the FT image corresponds to 0 (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 sampling can be cleanly removed by deleting strong (typically linear) 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.

IMAGES (processing only valid for images)



Add

- **Append** - append 2 images, matched on basis of (x,y) scales - images can be **tessellated** using append.
- **Buffer** – weighted addition of two buffers *(use a -ve 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 of selected regions of an image.

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.

Clip signal - replace all values outside user-selected limits with a fixed value (default is the average of the in-bounds data)

Convert_to_OD - compute OD (optical density) representation of a transmission image with user defined Io value. The default 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

Cross-link density - use Flory-Huggins equation to compute cross-link density from a polymer volume fraction image.

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

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

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

exp(Z) - exponential of z-values (useful to back-convert Absorption to Transmittance)

Images (*continued*)

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.
1 - fix a single image
many - fix rollover on a series of images (*.lst)

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

Histogram – compute the histogram of the pixels

Ln(Z) - natural log of z-values (use to convert Transmittance to Absorbance)

Log10(Z) - base-10 log of z-values

generate mask – convert image pixels to 1 if Z > threshold and 0 if Z < threshold
- used in conjunction with **Multiply buffers** to select portions of image
- optional write of a region_of_interest file to allow selection in stack_analyze
- use in conjunction with **Particle Analyze**

Multiply buffers – take product of current and a second 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.

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

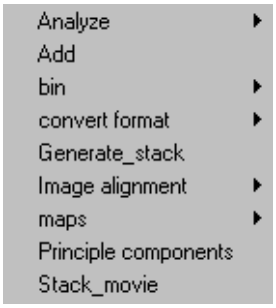
Profiles - generate and save intensity profiles from images
Linear – intensity along line defined by two user selected points
Radial – intensity as a function of angle in a circular region. The user defines a diameter which is then rotated about its center. Optionally, the resulting 'unfolded' radial 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 2 buffers with optional scaling. Images are matched by (x,y) scales

Remove zeros – replaces zero value pixels with a local average. (despeckles images with dropouts)

Replace line – use cursor to identify horizontal lines one wishes to remove. It can be replaced by the average of 2 adjacent lines (default) or by any selected line. The line suggested is the (n+1) line (one above the selected line). Cursor line value is indicated during replace line selection. This is useful for the ALS STXM data which is acquired line-at-a-time and tends to have errors which affect whole lines. Numerical selection of the line to replace allows access to un-displayed lines in large images

Images (*continued*)



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*)

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 is saved to disk in a 3-component TIF format which reads into Paint Shop 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.

Rotate - rotate image about a user defined point, by user defined angle. This can be used in conjunction with Images~Distort X,Y scale to remove image distortion by symmetrizing an object of known shape (e.g. circular particles).

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

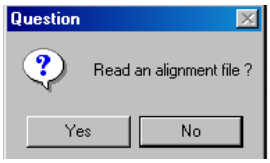
Warp –uses multiple matching point "morphing" capability of IDL. Intended to handle cases where a 2-point manual or simple shift will not track changes in the sample shape in a stack (these typically occur because of radiation damage).

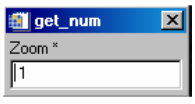
STACKS manipulation of image sequences

Analyze

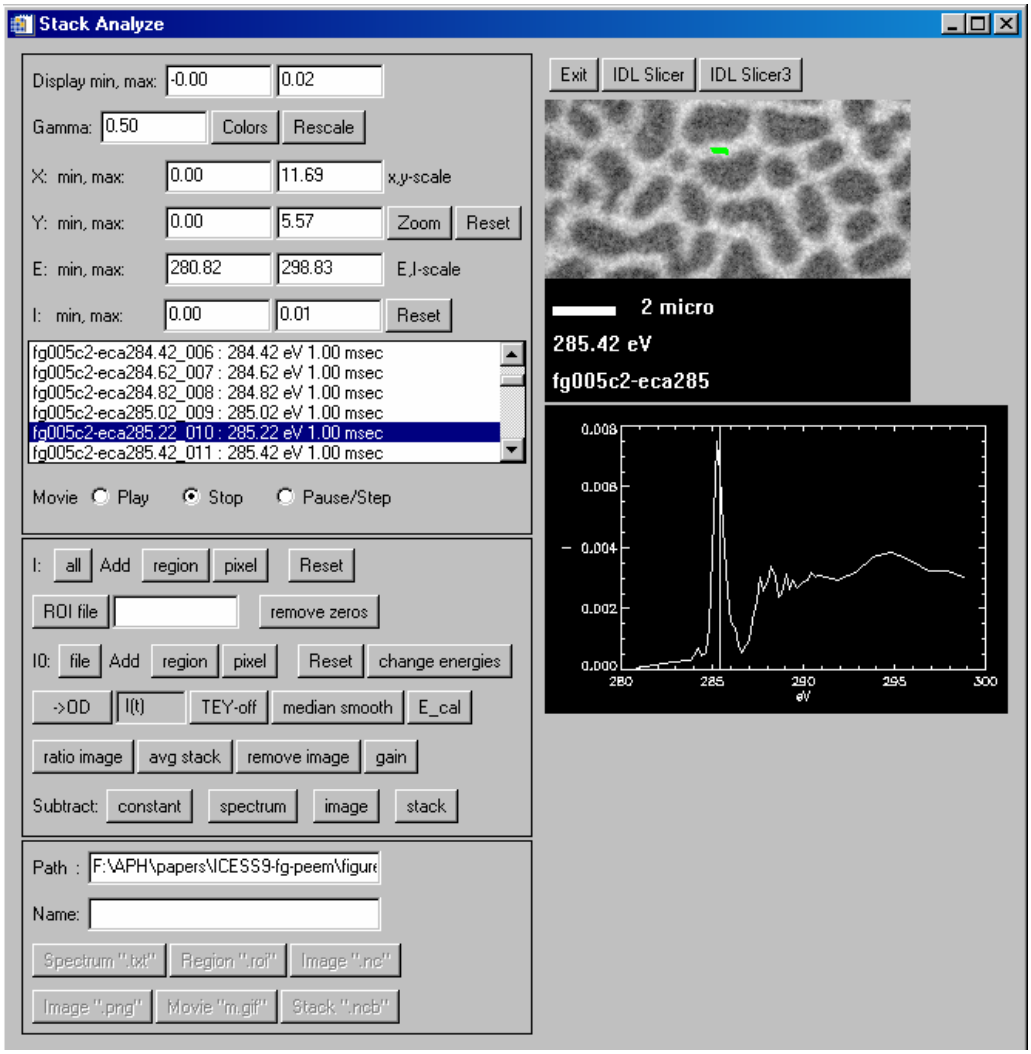
AXIS binary - loads an AXIS binary stack format file (*.ncb) into the stack_analyze widget. Note that it is convenient to store image sequence data in this binary representation since it is much more compact (by three times) than the original *.nc or sdf formats. Since it is possible to extract images 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 (to incorporate alignment shifts into data) and widget size scale factor, before the following widget appears.





STACKS (*continued)*



NB click on spectrum to select image at that energy

Display min, max – Z-axis limits
Gamma – gamma for Z-axis
Colors – select from palette of standard IDL color tables
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
E: min, max – use to set the energy scale of the spectral display
I: min, max – use to set the intensity scale of the spectral display
Reset – reset (E, I) scales to full scale values
Movie – play – play the move
- stop – stop playing movie
- pause/step – advance one frame

I: all - display average spectrum of all pixels (treated as I in Beer's Law or TEY)
Region - display average spectrum of region selected using
Pixel - display average spectrum of a selected pixel
Reset – reset selection of I pixels

STACKS (*continued)*

ROI file – browse to select *.roi (region of interest) file [click in box to activate]
Remove zeros - replaces zero value pixels with a local average. (despeckle)
Io: file - browse to select spectrum as Io (treated as Io in Beer's Law or TEY)
Region - display average spectrum of Io region selected using
Pixel - display average spectrum of a selected pixel as Io
Reset – reset selection of Io pixels
change energies – exchange current energies of stack for those in a selected file
->OD – convert Z-values from transmission to optical density using currently defined Io
TEY-off [**TEY-on**] toggle between absorption (-ln(I/Io) and TEY (I/Io) normalization
(the current status is displayed between ->OD and TEY boxes)
median smooth – apply 3-point median smooth to all images
E-cal – calibrate the energy (linear shift by user selected amount)
ratio image – ratio each image in stack to image (*.nc format) selected by user from disk
avg. stack – compute average of all images & stire in current aXis2000 buffer
remove image – remove currently displayed image from the stack
gain – multiply z-scale by user defined value
Subtract – constant – subtract a user defined constant from z-values of each image
- **image** – subtract image from file (*.nc) from each image in the stack
- **spectrum** – subtract the selected spectrum (*.txt) from spectrum at each pixel
- **stack** – subtract a user selected stack (*.ncb) from this stack (NB stacks can be generated from an image and aspectrum using the *stacks~Generate_stack* command)

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 sepctrum
Region ".roi" - write currently selected pixel set to file (mutiple regions are supported)
Image ".nc" – write displayed or all images to *.nc files
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 has licensing problems for use in other applications. The AVIMaker from Platypus software (http://www.c-point.com/) makes avi files which are much smoother, at least when used in power point for windows. The AVI Maker can also convert the avif files to other types of movie formats. Note that the individual gif images are required as input to AVImaker so answer 'N' when asked, *"delete all *.gif files?"*
Stack ".ncb" – write the currently processed stack data as a *.ncb stack
The user is asked to define the lower and upper row and column indices for the saved data, thus allowing removal of the lost regions from the alignment procedure. The *.ncb binary file generated is need for the **stack-fit** and **stack-SVD** routines. The save stack ".ncb" routine saves whatever is currently displayed in stack_analyze. It is strongly recommended that you normalize to Io (i.e. conversion to OD scale for transmission measurements, or divide by an appropriate reference for yield data)
Exit – exit stack_analyze widget
IDL slicer – enter widget to manipulate 3d display of stack (early version)
IDL slicer3 – advanced 3d manipulation of stack using latest IDL Slicer widget

STACKS (*continued)*

AXIS - allows user to pick the stack_list (*.sl) and the pixel shift (*.aln) files. If you cancel after first pickfile, then stack_build_list is started. If you select a stack_list and then cancel after the second pickfile, then stack_align is started. Once these choices are made the stack_analyze widget (described above) is started)
This command can be used to re-assemble a stack, which has been written out as separate *.nc files, and has some modified images.

Zimba – Carl Zimba's version of the stack analysis code. Currently these are the highest performing codes for stack analysis, although the code, when run from aXis 2000 often locks up for unknown reasons. Note this widget can be run outside of aXis2000 if you have an IDL license by running *zstack.pro*. Some features:

- Reliable alignment (stack_analyze does have an associated alignment widget but the code in Zstack_align is more capable)
- 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 files suitable for avi_maker

Note: The Zimba *.sl and *.aln file formats differ from those for the Jacobsen set. 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 help file.

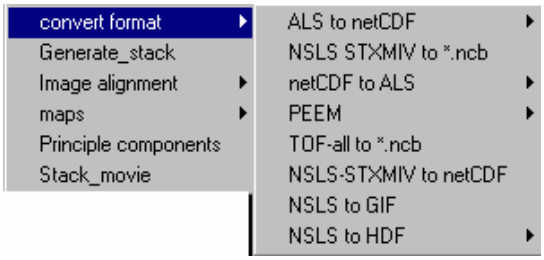
Add – add two stacks (useful to take differences to remove dominant contributions to better display minority signals by using negative weighting)

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 the processing.
- **current** image
- **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

convert format

For most conversions, the user selects the first file then the last file in a sequence, and the intervening files are converted assuming a standard file numbering convention (columns 6,7,8 of the name being file number).

- **ALS to netCDF** – converts 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 Stack_analyze automatically
  - **1** converts one file selected by pickfile
  - **many** converts many files selected by (first, last) pickfile routine
  - **file** converts files listed in the user-defined file (e.g. *.lst from ALS)



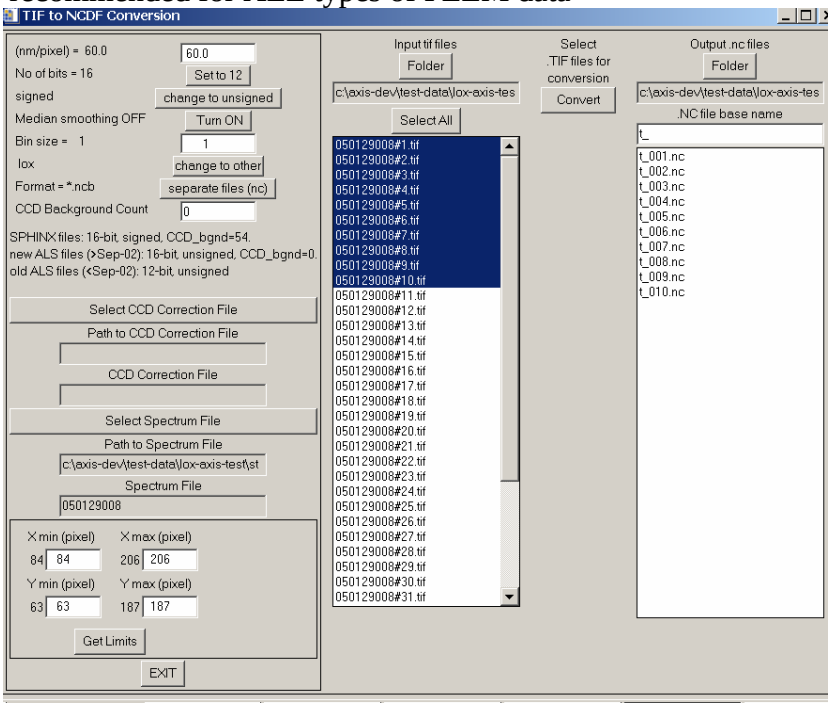
-**NSLS STXMIV to *.ncb** – converts nsls STXMIV image sequence directly to a binary stack file (all images in one file). NB One can use Zimba or Jacobsen align procedures on *.ncb files.

- 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)

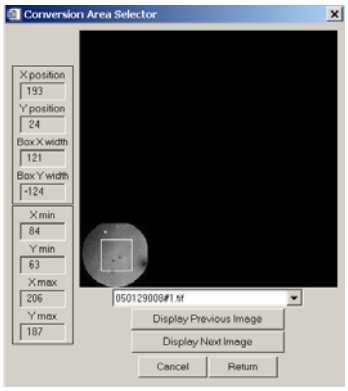
PEEM - converts PEEM stacks written by various instruments either to a set of *.nc files or a *.ncb binary stack file

General – recommended for ALL types of PEEM data



STACKS /convert format (*continued)*

The default conditions for SPHINX, ALS-PEEM2 [new (>Sep-02) & old (<Sep-02)] 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 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.



- **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

- **TOF all to *.ncb** converts sequence of time-of-flight PEPICO files to a 3d stack

- **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

STACKS / (*continued)*

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

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 'tempermental'.*

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

Stretch/ shift - 2-point calibration of image (x,y-scale)

After shifting or shift/stretching 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. cuts 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 1st letter of the original filename converted to **s** (shift) or **a** (shift/stretch).
5. generates ***.sl*** file of filenames to be used in stack_analyze. In addition it stores 2 (**s**) or 4 (**a**) files which are the shifts in real-space units (microns) (from buffers 5,6 and 8,9).

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.

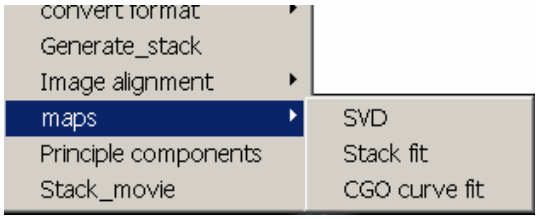
Notes for image_alignment*:*

- Clear buffers 5-9 (using the **Clear Buffer** *above the thumbnails) before starting as they 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 the 4 or more fiducial points identified on the first image, and on each subsequent image. This can be useful when the sample distorts during a measurement due to radiation damage

maps



SVD - convert an image sequence (stack) to 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). The user is prompted to either read the list of reference spectra from a *.par parameter file, or, after identifying the individual spectra, write the list to a *.par file. This is helpful in cases where the same set of reference spectra might be applied to a number of stacks.

STACKS/map (*continued)*

The **output** is a set of **composition maps**, automatically written to files with names constructed from a root and the component names (it is wise to keep the component names SHORT to avoid excessively long file names in directory listings). In addition the **residua**l 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 residuals** signal can be saved for later examination.
If the input stack is in OD units, and the reference files are on absolute linear absorption scales, 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 for further details.

Stack-fit – Performs a linear regression analysis (linear least squares fit) of the spectrum at each pixel to a sum of (1 to 8) user-defined model spectra and a constant.
NOTE: Relative to SVD maps, stack-fit adds an additional component to the analysis. CONSTANT' is a parameter which is flat spectrally (same at all energies) but different at each pixel. This may lead to additional uncertainty in the result. However in some cases, where there can be offsets of reference spectra relative to the data, due to Io errors, stack-fit is the more logical choice.

CGO curve fit . This uses a conjugate gradient optimization method (from Numerical recipes) to perform spectrum-by-spectrum curve fits to 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

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 reference spectra
Buffers **4-6(8)** – the component images - i.e. the spatial distribution of the component
Buffer **7** – the map of the linear term (for **Stack-fit** only)
Buffer **9** – the map of the chi square values of the fit at each pixel (~ fractional uncertainty)

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 coalesce any 3 of the component maps into a single **color-coded composition map**, by using the *Images~RGB* command. *(If you only want to combine 2 maps, select one of them for 2 colors then use an image processing program such as Paint Shop Pro to set the duplicate color to zero intensity)*

Stacks / maps *(continued)*

==========

Notes on evaluating the significance of maps derived using SVD, Stack fit or CGO fit.

The user of *stacks~maps* must realize that the fitting code will always give a result, but the result may not be valid. It is important to evaluate the quality of the fit. AXis2000 provides a number of tools to do this.

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. Using the ROI files in *stack_analyze* of the stack allows the analyst to extract the spectrum of the poor fit region, which may be a 'missing component'.

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 poor fit regions of the residuals stack.

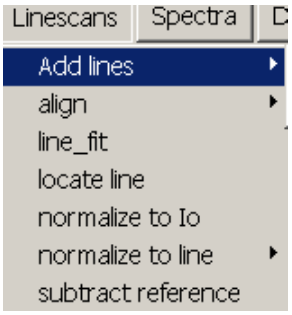
One can apply *images~generate_mask* on the component maps to identify those pixels with large amounts of components of interest. After extracting the spectrum of those pixels using the ROI files in *stack_analyze* of the stack, the quality of the spectral fit can be examined by applying *spectra~curve fit~{linear regression, or CGO fit}*

==========

Principle components - applies the 'canned' IDL principle component analysis package (PCOMP.SAV) to spectra from a set of OD images (binary stack in an *.ncb file). The routine implemented in aXis2000 provides the eigenvalues and eigenvectors (component spectra) 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 eigen images are readily generated by using the eigenspectra as model spectra in a SVD map

Stack_movie calls Jacobsen stack_movie routine. Files defined by user-selectable stack list (*.sl) file. If 'cancel' is selected, the stack_build_list routine is initiated.

LINESCANS



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 orthogonal directions
- remove low frequency noise from images (average all horizontal, or all vertical, then ratio)

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)

Linescans / *continued*

align - 'tilts' data to user specified line
- **Linear** – straight line defined by 2 points
- **Curve** – curved line defined by multiple section straight lines

line_fit – apply curve fit (as in stack-fit) to a linescan. This is an excellent method to analyze and visualize chemical composition along a line.

locate line – draws (x,y) position of a linescan on the image currently displayed. Be sure to display an image on which the linescan was defined (or at least without any base scan changes).

normalize to Io – divides each **horizontal** line by user-selected buffer 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*.

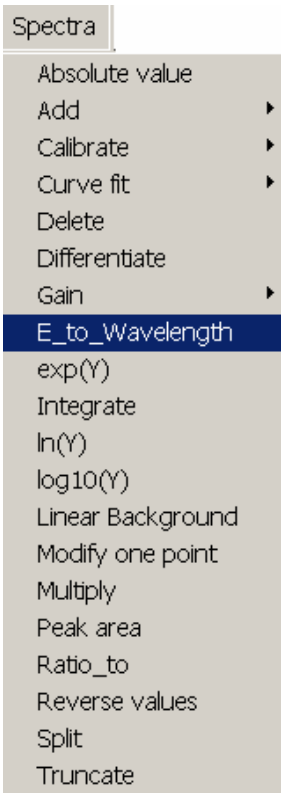
subtract reference computes (Image – ref), The reference 1-d profile signal is taken from user selected buffer (this does same thing as *normalize to line ~horizontal*)

SPECTRA (procedures only valid for spectra)

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

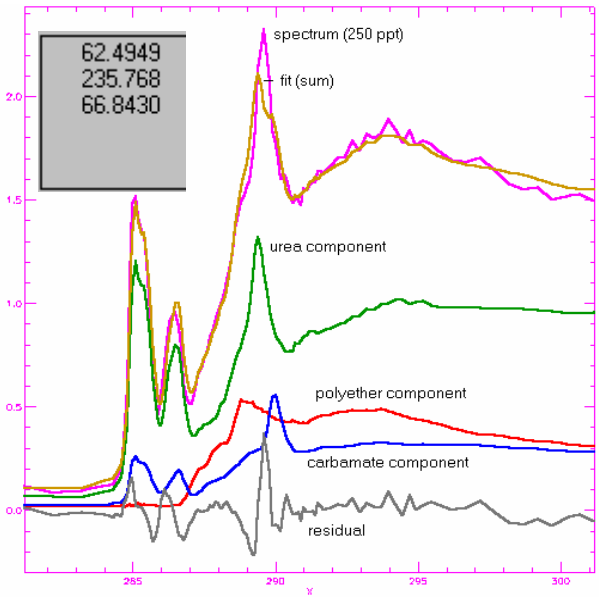


Spectra */continued*

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.



Delete - delete all data between 2 cursor-identified positions

Differentiate – take derivative of displayed signal (simple $\Delta Y/\Delta X$ only)

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 (symmetric conversion)

exp(Y) - computes exponential

Integrate – determine integral of displayed signal

Ln(Z) - natural log of z-values (use to convert Transmittance to Absorbance)

Log10(Z) - base-10 log of z-values

Linear background - subtract user defined line

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

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

Reverse values – reverses the (x,y)-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

---

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 buffer, with bottom y-position selected by cursor; no rescale (ie y-scale is set by previously plotted data)

Hint: 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 either the thumbnail plot, the buffer label, or the 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

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

Display / *continued*

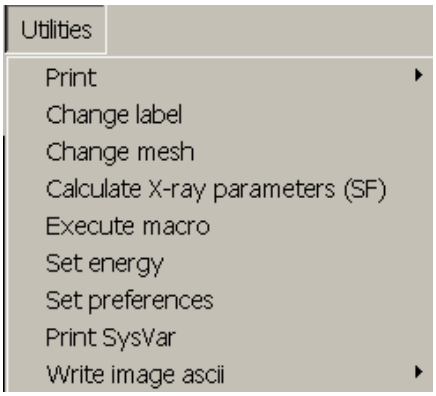
Show color scheme - display current colors assigned. Note that the bottom 16 colors in the 256-color table are assigned to 'hard' colors to allow ready differentiation of different buffers.

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)

UTILITIES



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* ***PrtSc*** *(WIN) to copy the monitor screen to the clipboard, then use an image processing program (e.g. Paint Shop Pro) to cut out the area of interest, and paste that into a presentation program, such as powerpoint.*

Hint: *(2)* Use *Utilities~Preferences* to define the Print_command which will transfer the print output file to your local or network printer. Some examples are provided in the AXIS.INI file. The PRINTER option allows pass through to the default system printer.

Change label - modify label (this is what will be saved to disk)

Change mesh - redefine numbers of pixels (images) or points (spectra). This is useful to allow matching of sampling of different data sets. Same as *images~resize*

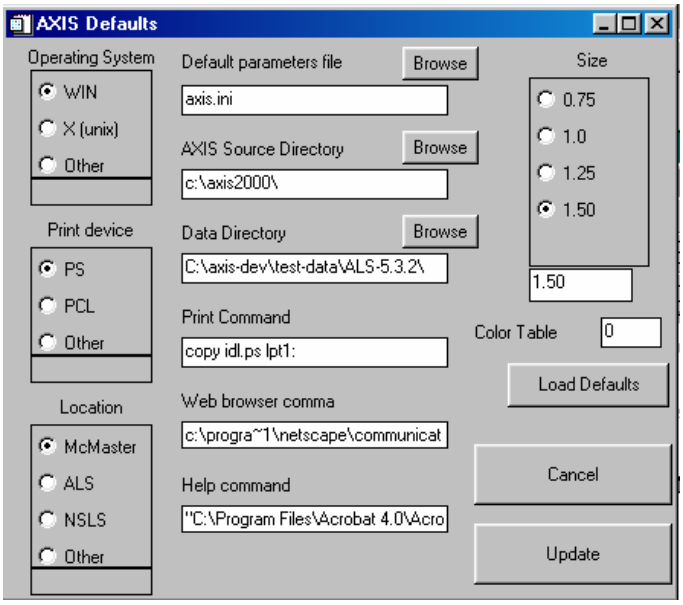
Calculate X-ray parameters (SF) - Determine the **mass absorption** or **transmission** for user defined elemental compositions. The mass absorption atomic signal provides normalization for model spectra used in the SVD map and stack_fit routines The transmission for a user-defined composition, density 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. Run SF at the IDL command line to access all features. (use *sf, /help* for a list of the capabilities)

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.

Utilities / *continued*

Set energy - set energy of an image - useful to calibrate the energy of images for, eg. SVD. Also, in some cases *Write~nsls* does not store the correct energy and this fixes those files.

Set preferences – allows user to modify the default parameters which are contained in a file called **axis.ini**. 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 to allow the user to provide the required details.

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

Second row of single command buttons

EXIT write axis.ini file with parameters then return to main IDL command window.

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

NOTE:
From the Command Window of IDL type axis2000 to (re-)start AXIS. If you wish to display the rotating AXIS logo, restart AXIS by typing **axis2000, /spin** at the IDL prompt. With or without rotation, if you have a net connection, and you have the command to start your browser identified correctly in the axis.ini file, then when you click on the indicated synchrotron sites or the aXis2000 label, a web connection is made to the X-ray microscopy facility or aXis2000 web site.

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 files in the selected folder. This is useful when you do not have detailed information on data files.

HELP

Calls Adobe Acrobat reader, using the Unix (Acroread) or Windows command (AcroEx32) to bring up **AXIS.pdf** (a pdf version of **AXIS.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 and is set by the **help_cmd** line in the **set_preferences** dialog.

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 *to transfer it to buffer 1-9 to avoid overwriting the result in the next processing step*.

CLEAR buffer pops up multi-buffer widget to let user select those buffers they wish to clear. (The same function is also available as *utilities~clear*).

CONTROLS BELOW THE MAIN IMAGE WINDOW

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. Continuous updating of the lineouts and the zoom/pan, as well as symbol plotting, take large amounts of cpu/graphics resources and can slow aXis2000 to a to a crawl on slow computers.

Bar – ON – display white scale bar of correct size (1,2,5,10 pattern; within 5-10% of image)
The position of the scale bar can be defined using *Display~scale bar position*

Gamma control – use either the slider or the number box to change the image gamma. Note this control is only active if an image is displayed.

********** XXXXXXXXXXXXXXXXXXXXXXXXX**********

INSTALLATION

********** XXXXXXXXXXXXXXXXXXXXXXXXX**********

Copy all code into a directory. (use **WinZip** or similar to **expand axis2000.zip using folder names** into your preferred source directory). – I use c:\aXis2000\.
The directory structure should be: C:\aXis2000\ \logo\
\sfdata\

To set-up IDL so it starts aXis2000 every time:

A.1 Start IDL I use version 5.2)

A.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

A.3 Shut down IDL and restart.

To use IDL without automatically starting axis2000

B. 1 Type **AXIS2000,/dialog** at the IDL command line.

Several different versions of the INI file are included. Try 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 (to just fill your graphics screen), the codes used to launch acrobat, command that will run your web browser and command to transfer a postscript to file to your printer. See below for an example and more details.

A. For PC (or other WINDOWS device) 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)

B. For Unix (X-windows) based systems

Since Unix is case sensitive, it is important that the lower case of the names of the 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.

- 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_com.mon are used for include files. The COMMON include files are now:

axis_com.pro	aligncom.pro
analcom.pro	bsif_com.pro
img_com.pro	

The stack codes which are located in the same subdirectory as the rest of the axis code differ significantly from those currently being supplied by Chris Jacobsen on his web site. These codes are based on the Jun-98 version of his stack widgets, but with considerable enhancements.

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

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

IDL Virtual machine version
The aXis2000 widget can be run without an IDL license by downloading the IDL VM (virtual machine) code, and running the aXis2000.sav file contained in the aXis2000 package. Instructions for setting up a Windows desktop shortcut to this are given at
http:/unicorn.mcmaster.ca/aXis2000.html

Adam P. Hitchcock
Canada Research Chair in Materials Research – CLS/CCRS
Professor of Chemistry 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 THE BEGINNING USERS, a number of **USEFUL DOCUMENTS** are available by download from http:/unicorn.mcmaster.ca/aXis2000.html

TUTORIALS – a set of unit operations for manipulating X-ray microscopy data.

STACK_ANALYZE_MANUAL.doc - manual to walk-through a typical stack analysis based on ALS STXM or PEEM data. [DATED]

STACK_FIT.DOC - manual to walk through stack fit from a prepared *.ncb aligned image sequence.

FOR THE PROGRAMMERS – SOME USEFUL ADD-INS

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 cover the formats AXIS is able to process (*contact APH to obtain*)

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

```
c:\axis\
WIN
PS
C:\axis_dev\test-data\
 1.00000
 0
ALS
idl.ps
Arial*12
copy idl.ps lpt1:
c:\acrobat3\exchange\AcroEx32 c:\axis-dev\axis.pdf
c:\progra~1\netscape\communicator\program\netscape
---------------------------------
written: Mon Jan 31 14:17:18 2001

; ********** Initialization file for AXIS Widget ******************
;
; (aph 31-Jan-00)  Please adapt to your hardware
; Items set                    devices currently supported
; --------------------------  --------------------
0 CodePath                   (e.g. c:\axis\) - include final separator
1 Screen display device      (WIN, X)
2 Print file format             (PCL, PS)
3 Default data directory     (any - last path wil be saved on exit of AXIS
4 graphics scale factor      (0.5 - 2) (1 = 360x360 pixels)
5 color table                (any IDL supports) 0 = B/W; 3 = red
temperature
6 location                   (NSLS has special meaning = spooler switch)
7 spooler or print file      (c:\tmp\x1a at nsls - special !!)
8 default font for widget    (windows - Arial*14; unix - 6x10 or 8x13 )
9 Printer command line       (command line needed to transfer to printer)
10 Command to view Help pdf  (get acrobat reader free from www.adobe.com)
11 command to run browser    (WIN: must be executable from MS-DOS window)

   Win95/win98    = "copy idl.ps lpt1: "
 some examples of printer command
   NSLS (spooler) = "WLPRSPL -L c:\program files\wlprspl\wlprspl.qs
idl.ps"
   ALS (Unix)    = "lpr -S XRAYS-1 -P 6-BL7LJ4 idl.ps"
   ALS (BL7-Dow) = "lpr -S XRAYS-1 -P 6-BL7LJ4 idl.ps"
   ALS (Dodgson) = "lpr -S XRAYS-1 -P 7-210HP idl.ps"

form of Help command on Dodgson.ALS.lbl.gov
    c:\progra~1\acrobat3\reader\acroRd32 c:\axis\axis.pdf

form of Browser command on jabberwocky (standard Netscape install)
    c:\progra~1\netscape\communicator\program\netscape
```

A complete set of sample data files for aXis2000 is available.

GLOSSARY of terms associated with aXis2000

axis.ini - text file containing default parameters for aXis2000. May need to be edited to optimize performance

component maps - spatial distributions of a chemical species, which can be generated from multiple images (selected energies or a full image sequence) using SVD maps or 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 Virtual Machine – a version of IDL which allows execution of compiled IDL script. This can be downloaded for free from Research Systems Inc (rsinc.com)

IDL widget - a graphical user interface with pre-programmed data manipulation or other capabilities written in Interactive Data Language, from Research Systems Inc (rsinc.com)

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 without prior knowledge of the data structure. Read and write routines are available in C++ and IDL.

ZSTACK

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

Dear Zstack User:

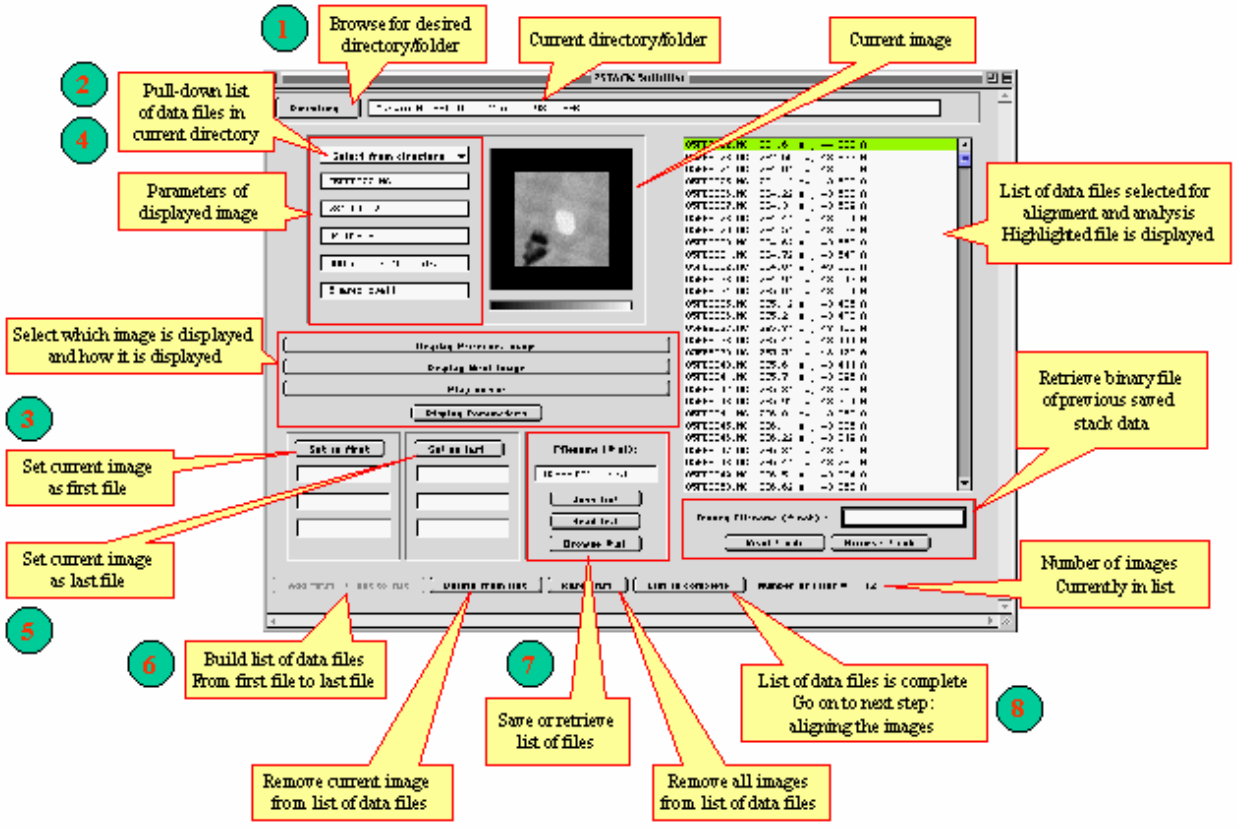
ZSTACK is a suite of IDL procedures for alignment and analysis of a series of x-ray microscopy images that have been acquired at different x-ray energies. It operates on Windows, Macintosh, and UNIX platforms using IDL v5.0 or later.

ZSTACK was written originally to satisfy my own needs for analysis of STXM spectral image stacks. This is based upon the original STACK code developed by Chris Jacobsen (SUNY - Stony Brook) but it has been 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 aXis2000 data analysis package. ZSTACK can also be used independently of aXis2000. The development of ZSTACK has been an evolutionary process. Please notify me of any bugs, problems, comments, or suggestions for improvements or new features.

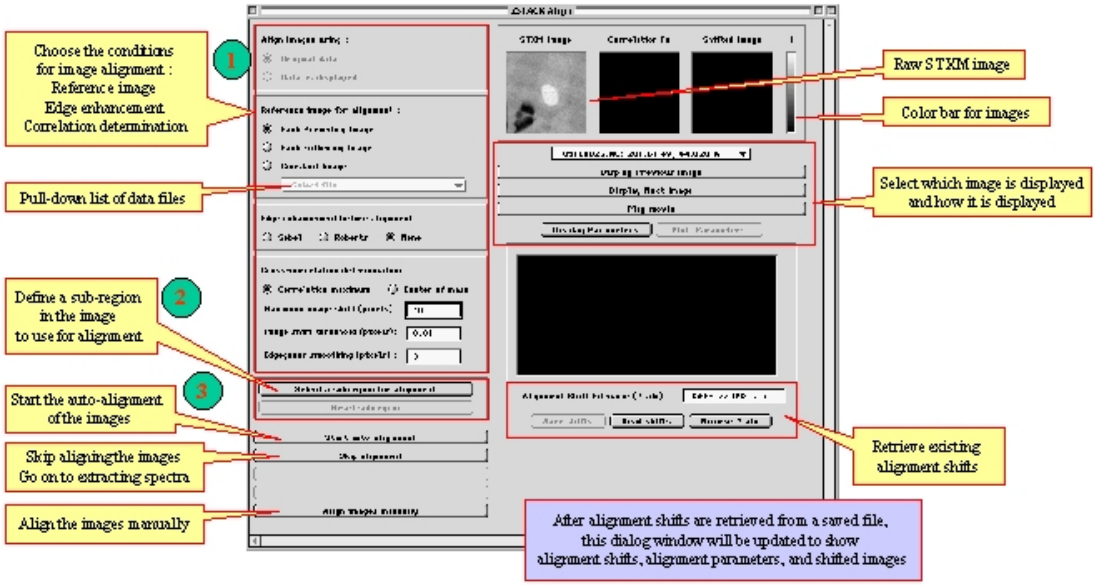
Carl G. Zimba
Photons UnLimited
84 Oxford Street
Arlington, MA
photons@mediaone.net

ZSTACK Buildlist



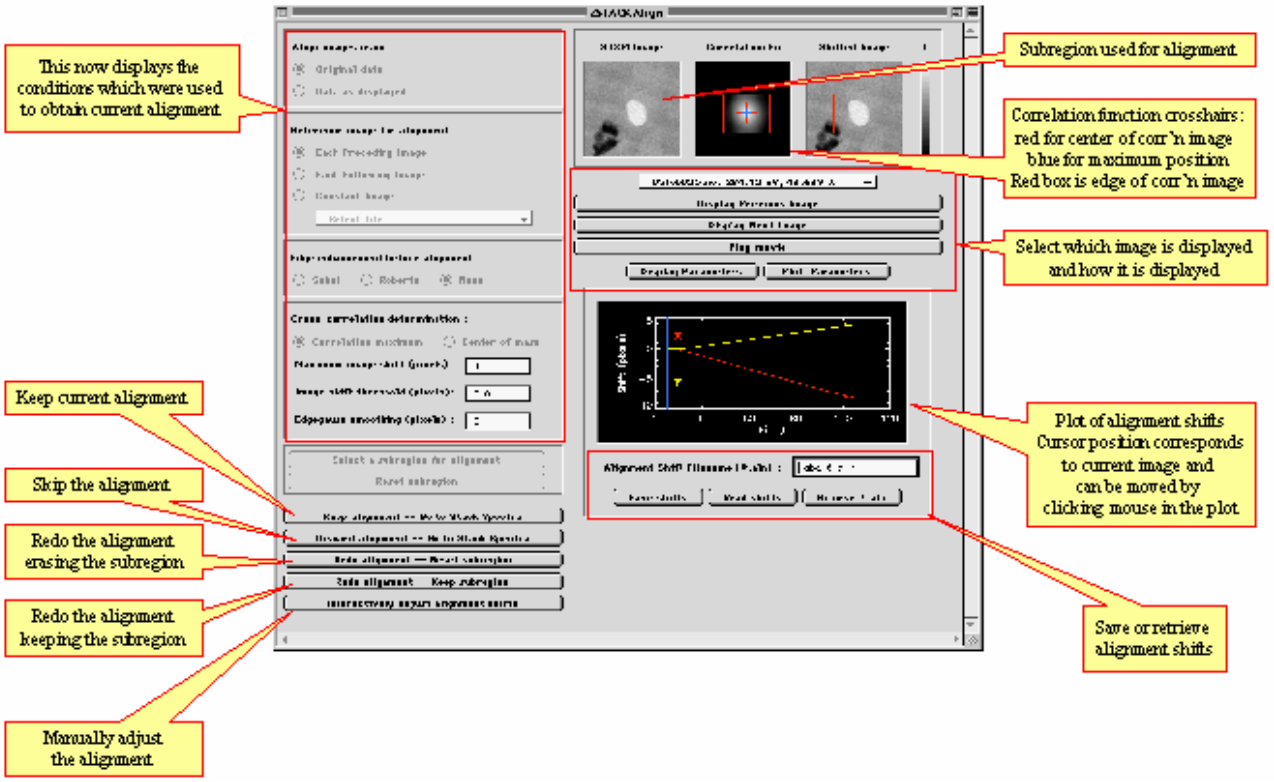
ZSTACK Align

(before alignment)

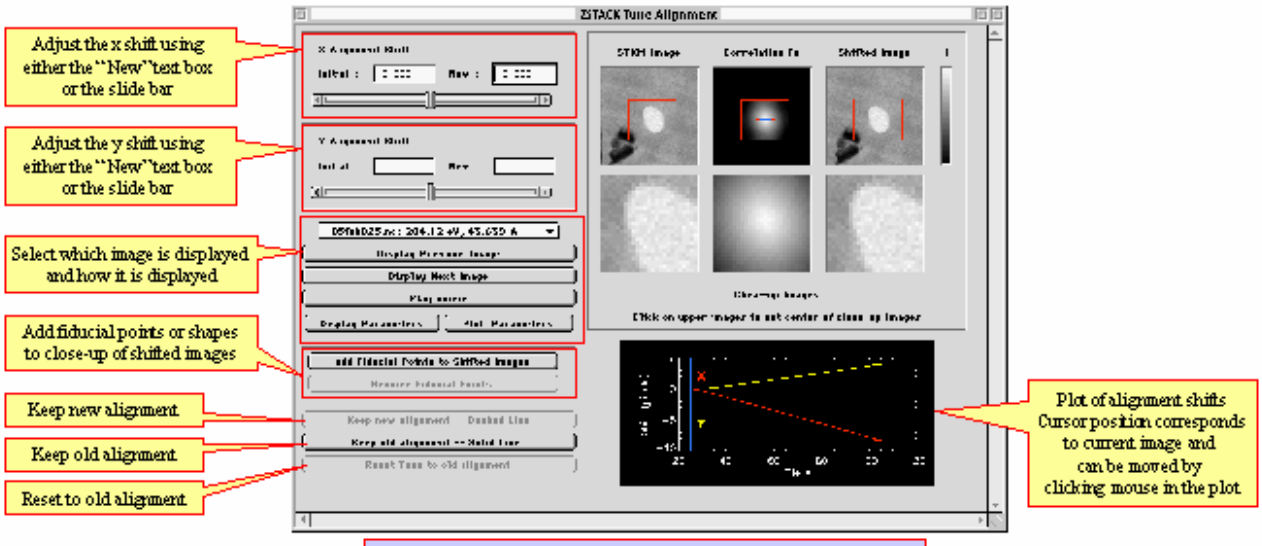


ZSTACK Align

(after alignment)

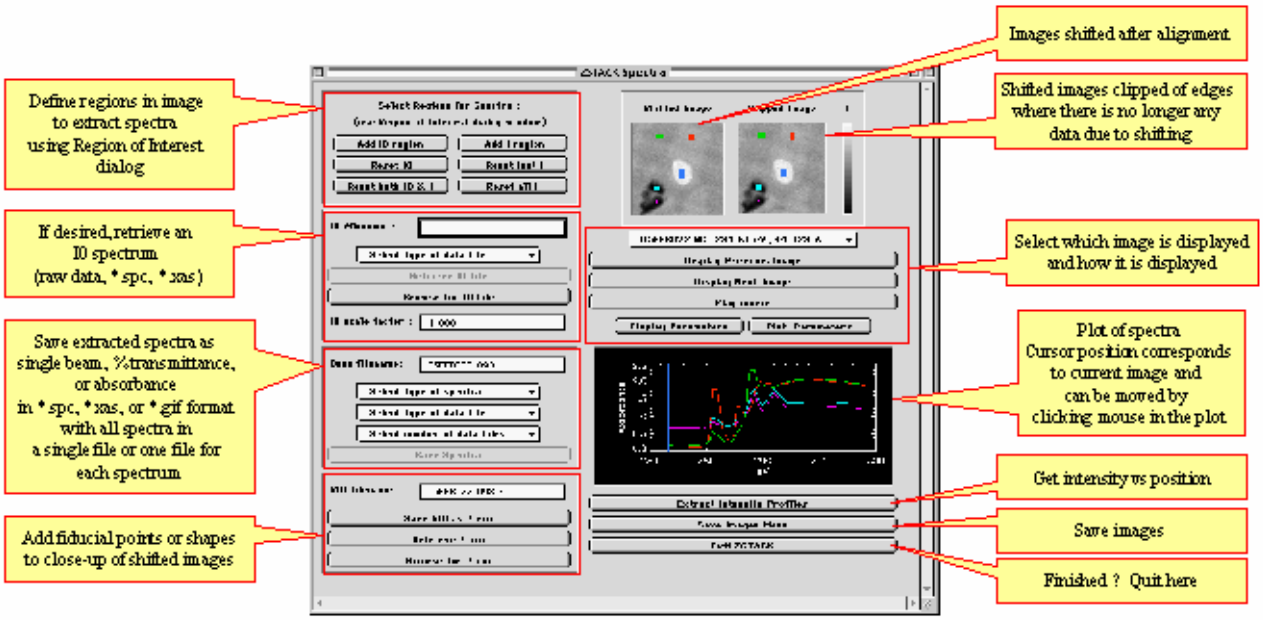


ZSTACK Tune

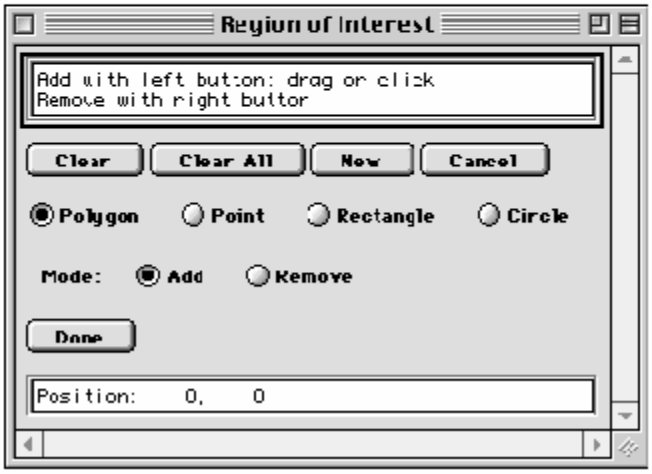


By placing some fiducial points in the close-up of the shifted images, the quality of the alignment can be inspected while playing the images as a movie

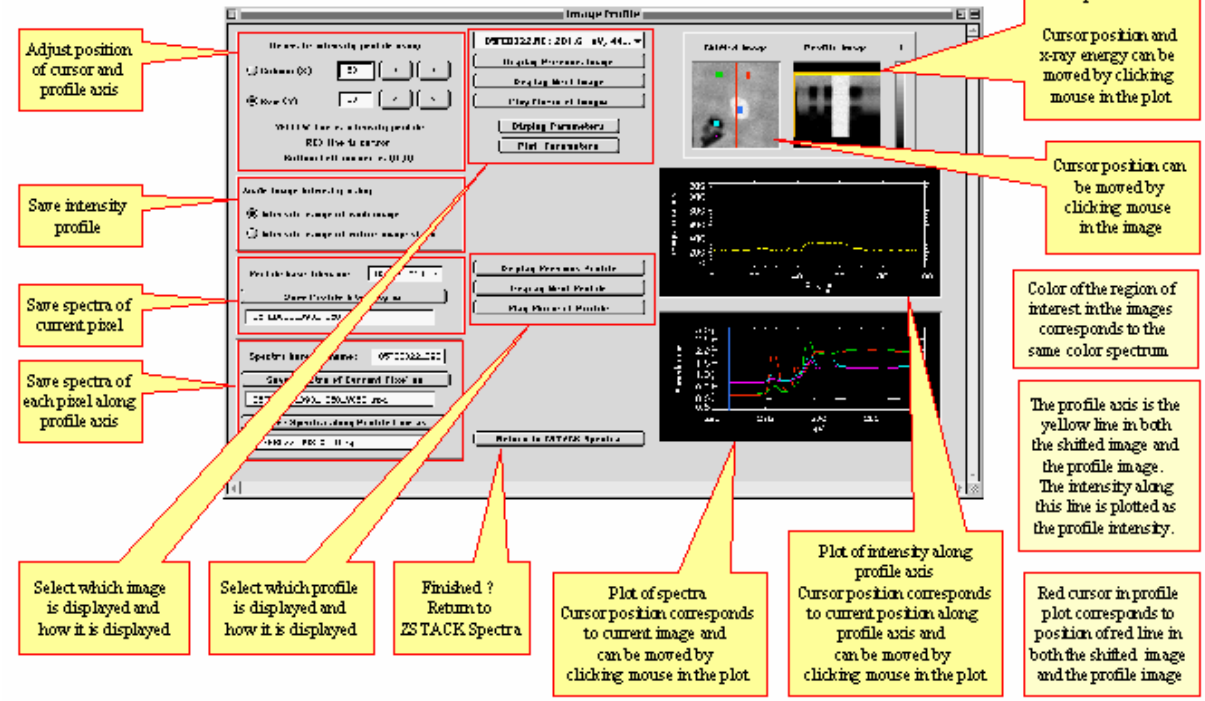
ZSTACK Spectra



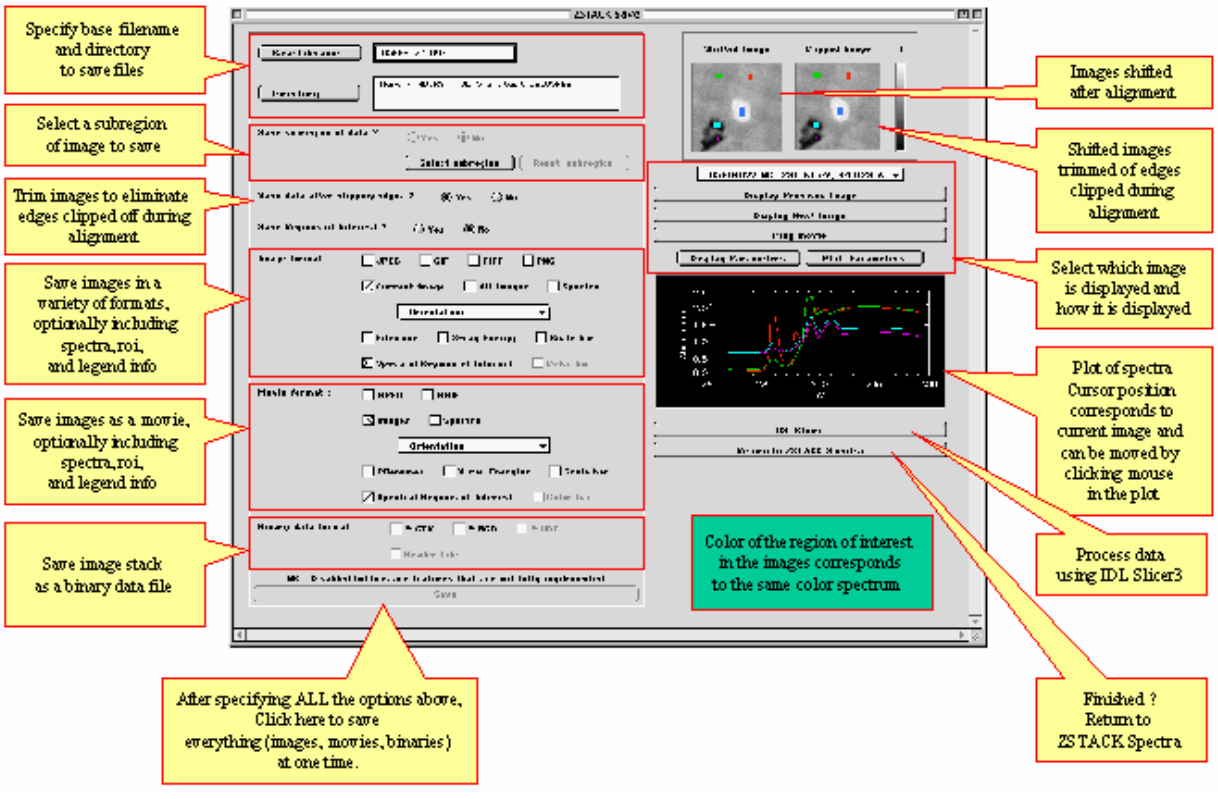
Region of Interest



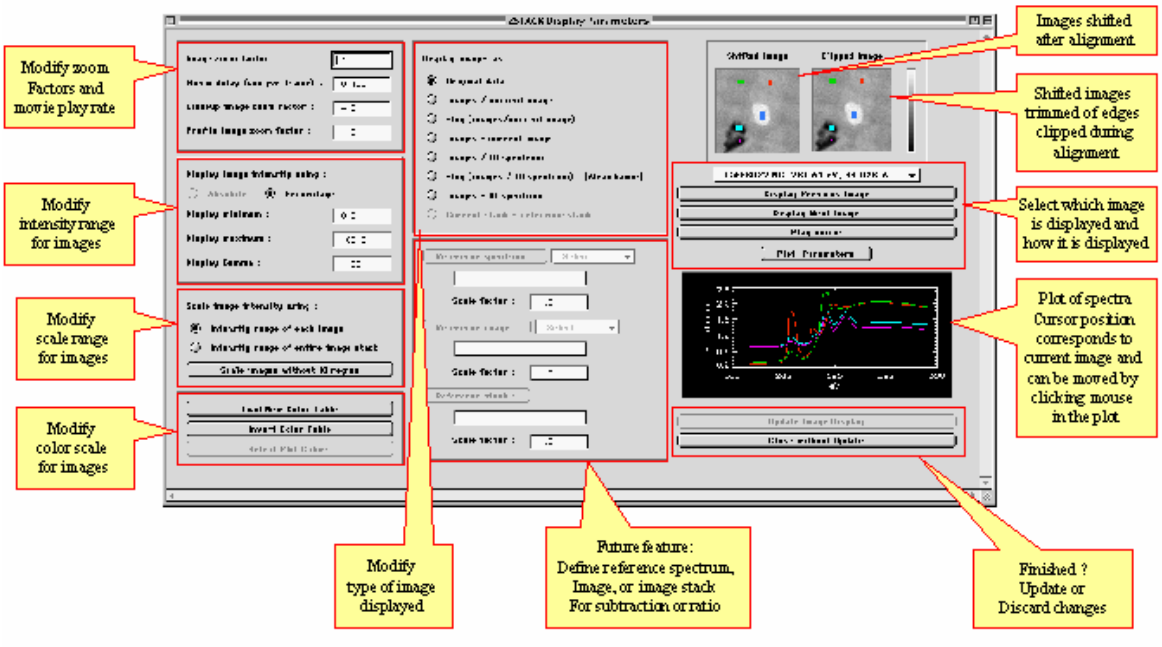
ZSTACK Profile



ZSTACK Save



ZSTACK Display



ZSTACK Plot