Introduction to TVX

WHAT IS TVX?

TVX is a control program running under Linux that can both operate x-ray detectors and process their 2-D x-ray image outputs. The hardware specific routines needed to run a particular detector reside in CAMSERVER, a companion application not covered in this document. While either SPEC or TVX could interface with CAMSERVER, this document focuses on the processing of x-ray images once they have been acquired

TVX has a robust system for displaying images, and operating on them. It provides all the image arithmetic needed to perform background subtractions, “zinger” removal, and image intensity and distortion corrections. Tools are included to select regions of interest and perform operations such as radial and azimuthal integrations. One can use the internal scripting language to correct and analyze a sequence of images. Macros can be defined, and externally generated scripts can be executed. TVX reads and writes TIFF files of various formats (8, 16, 32 bit signed or unsigned integers as well as 32 and 64 bit floating point numbers). Binary images created with other formats can also be read.

TVX and CAMSERVER are organized as sets of "core" commands and facilities that are common to all versions of the programs, and sets of installation-specific subdirectories, which can be tailored to a specific installation. TVX has evolved from an earlier program TV6, developed by Eric Eikenberry, Sol Gruner, and Mark Tate for experimental control of x-ray detectors with analysis for small angle diffraction. Eric F. Eikenberry (Paul Scherrer Institute) has primarily been responsible for porting the code to Linux. Mark Tate (Cornell University, mark@bigbro.biophys.cornell.edu) is responsible for Cornell specific content and should be the primary contact for suggestions for software improvements from the Cornell community. Darol Chamberlain (dc35@cornell.edu) should be contacted for general questions regarding the operation of TVX.

USING TVX:

A. Starting and exiting TVX

1. On a computer with TVX installed, open a Linux terminal. Use the **cd** command to navigate to the directory containing the TVX program. Then start TVX:

   cd /home/<*TVX directory path*>

   ./tvx

This will start TVX, with each command line starting with the TVX prompt (*****).

2. To exit TVX, enter either:

   ***exit** or

   ***quit**

B. Getting help

1. To get a complete list of TVX commands enter:

   * **menu**

TVX returns a complete listing of:

a) **Reserved words**, the words used in the scripting language
b) **External Procedures** and **User commands**, the executable routines (which will often take arguments on the command line)
c) **Defined variables and strings**, the current user defined macros
d) **User variables**, the system variables that can be queried and set from the command line

2. To get an explanation of these commands, type:

   ***help** *TVX_command*

   For example, ***help deleteobj** returns:

   --- KEYWORD = DeleteObj :
   DeleteObj IM : Delete object descriptor associated with IM from memory. This does not delete the file IM if IM is not a Memory file (e.g., Disk and Vmem objects are not erased from disk or Display memory).

.

C. Setting the directory path for image, graph, correction and configuration files

1. TVX needs to know which directory the user's data is in. The command ***imagepath** [<*image directory path*>] both shows and sets the directory TVX is using for the reading and writing of image files (the [ ] brackets used here to denote optional arguments). For example, entering:

   ***imagepath**

   returns: **imagepath='**<*current image directory path*>**'**

The ***imagepath** command is also used to set the default path. Entering:

***imagepath /home/Max_von_Laue**

sets the default image directory to /home/Max_von_Laue. Unless specified otherwise, TVX will read and write images to that directory.

2. The ***grafpath** [<*graph directory path*>] command is likewise used to show (when called without an argument) and set the directory TVX is currently using for the reading and writing of graph files.

3. The ***correctionpath** [<*correction directory path*>] command is used to show and set the path to the directory containing the distortion and intensity correction files. The correction files are set using

   ***setdist** <*xdistortionfile*> <*ydistortionfile*>

   ***setint** <*intensitycorrectionfile*>

4. The ***configpath** [<*config directory path*>] command shows and sets the path to configuration files.

These paths are often set in a TVX startup file or script.

.

D. Reading TIFF images into TVX

TVX is designed to work on grayscale images in the TIFF format (8, 16, 32 bit signed/unsigned plus 32 and 64 bit floats). A TIFF image in the directory specified by the ***imagepath** command can be referred to by name in TVX commands without a path description or the .tif extension. If the image files im1.tif and im2.tif are in the default image directory, then the command:

***move im3=im1+im2**

adds the two files together and writes the sum into a new TIFF file called im3.tif in the same directory. One can also explicitly specify the paths as in:

***move /home/data/im4=/home/temp/im5+/home/temp/im6**

NOTE: While these TIFF images of varied data types (16 and 32 bits) are valid constructs from the TIFF specification, not all programs that read TIFF images will know what to do with these formats. In addition, there are TIFF tags that are defined in these images that other programs should ignore, but they often display error messages instead.

E. Importing non-TIFF images into TVX

Many detectors (e.g. ADSC Quantum detectors) use binary file formats other than TIFF to record images. Non-TIFF images must be imported using the ***RawImageIn** command. The command has the format:

***rawimagein** *diskfile* [*IM*] [*header length*] [*width*] [*height*] [*bits/pix*]

The command imports a binary *diskfile* into the TIFF image [*IM*] with parameters:

[*header length*]- number of bytes to skip at beginning of file
[*width*]- Image width in pixels
[*height*]- Image height in pixels
[*bits/pix*]- Bits per pixel, usually 8 or 16

For example, assume an image named SampleImage.img has been taken by the ADSC Quantum 4 detector and copied into the imagepath directory. Quantum 4 images start with a 512 bytes long header, followed by a 2304 pixel by 2304 pixel image, with 16 bits per pixel. The command:

***rawimagein SampleImage.img im01 512 2304 2304 16**

imports the image into TVX by writing the TIFF file im01.tif into the imagepath directory.

The ***RawImageIn** command can also import images from outside of the imagepath directory by including a path description with the file name. For example:

***rawimagein /home/Dfiles/Q1image.img im02 512 1152 1152 16**

reads the 1152x1152 ADSC Quantum 1 file Q1image.img in the directory Dfiles into the TVX file im02.tif in the imagepath directory.

Note: Some binary files will have the opposite byte order (PC's vs. Mac's for instance). The command:

***byteswap** [*IM*]

will swap the high and low order bytes of a 16-bit image after it has been read in by ***RawImageIn**.

F. Displaying image files

Images are most often displayed using the power law method, with the following format.

***disp** [*IM*] [*Min*] [*Max*] [*Scal*] {cycles through 3 concurrent image displays}

- or- ***disp1** [*IM*] [*Min*] [*Max*] [*Scal*] {displays into most recently used image display}

Here the grayscale image is displayed with all pixel values equal or less than [*Min*] being black, all those equal or greater than [*Max*] being white, and those in between displayed with a grayscale according to:

Grayscale# = (# of grays)*((value - *Min*)/(*Max* - *Min*))**(*Scal*/15)

A small value of *Scal* (~3) gives a very steep transfer function at low values, and very little contrast at high values. *Scal* has a maximum value of 20 and *Scal* = 15 gives a linear transfer function.

For example, take the Quantum 4 image imported as im01.tif in the last section. Entering

***disp im01**

displays the image using the most recently used values of [*Min*] [*Max*] and [*Scal*]. The window that ***disp** opens has various options to change the display, select various image regions, and provide intensity information.[1]

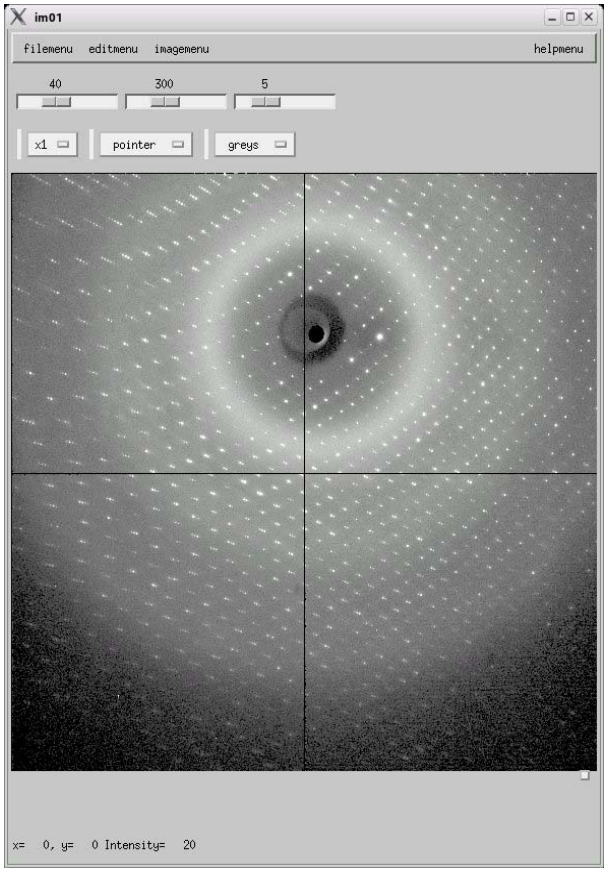


Image information: Notice that when the cursor is over an image pixel, the value and x-y coordinates of the pixel are displayed in the lower left hand corner of the window. In this image some bright spots reach saturation, while the rest of the image ranges between 30 and 160.

Min/Max/Scal Sliders: After displaying the image, [*Min*] was set to 40 (value<40 gives black), [*Max*] was set to 300 (value>300 gives white) and [*Scale*] was set to 5. Sliders for Min/Max only have coarse adjustment with the mouse. Fine adjustments can be made to these numbers by clicking on the slider and then using the arrow keys to adjust the value. The same result could be generated by entering:

[1] im01, Thaumatin crystal, courtesy of Chae Un Kim

***disp im01 40 300 5**

Zoom menu: Selecting a zoom factor of **x2** through **x10** will bring up a secondary zoom window with the same display parameters as the parent.

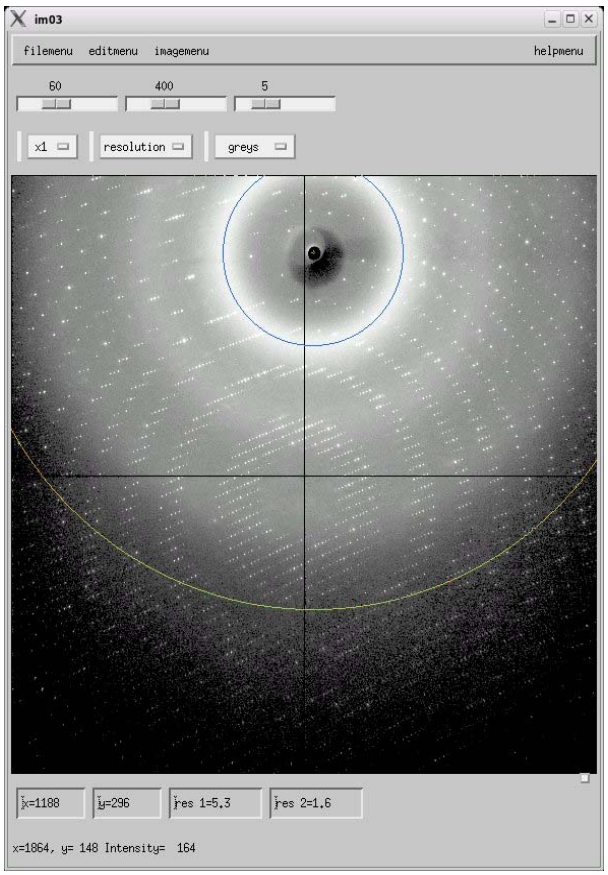
Cursor menu: The pull-down cursor menu changes the region of interest selection tool. Specific tool parameters (location, size, etc) will be shown in editable boxes just below the image in the main display. These parameters are passed to TVX commands.

Pointer: single pixel value display.
Annulus: sets 2 concentric circles of radius rad_1 and rad_2 at x,y. Used to set region of interest for ***AnnularInt**, ***Spot**, ***Integrate** and ***PixlFill**
Box: sets a rectangular box with coordinates of opposite corners x1,y1 x2,y2. Used for ***Box**, ***Integrate**, ***Histogram** and ***PixlFill**.
Butterfly: sets a "butterfly" shaped region, most often used for radial integration. Center of butterfly at x,y an angular splay (in degrees), a direction pointer (in degrees) and a width at the origin (in pixels). Used with ***Dens**, ***Integrate**, ***PixlFill**.
Resolution: allows the user to translate image positions to d-spacings in angstroms. To make these calculations, the user must set three system variables, **lambda** (x-ray wavelength in angstroms), **pixel_size** (pixel size) and **det_dist** (detector to sample distance). The variables **pixel_size** and **det_dist** must be in the same units, with millimeters being the most commonly used. A typical command sequence would run:

***lambda=0.9186**	{set wavelength to 0.9186 angstroms}
***pixel_size=0.0816**	{F2 Quantum 4 pixel size = 81.6 μm}
***det_dist=165**	{sample to detector distance = 165 mm)
***disp im03**	{display image im03}

The user then selects the **resolution** button, and centers the concentric circles on the beam. The d-spacings which correspond to the radius of each circle are displayed in the two boxes at the bottom of the window. The example below has a ring at ~5 Å caused by the cryo-protectant with resolution down to ~1.6 Å.[2]

[2] im03, Thaumatin crystal, courtesy of Chae Un Kim



Look-Up-Table menu: Modifies current display look-up-table

grays/spectral/thermal: switch between grayscale and false color schemes

decades: Linear lookup table with multiple (set by [*Scal*]) wrap-arounds

power: power-law lookup (*Scal* sets power with *Scal*=15 being linear)

reverse: switch lookup table to be a negative image. Selecting reverse again will toggle back to normal.

Filemenu/Editmenu/Helpmenu: Placeholders- not implemented yet.

Imagemenu: - menu option for button selection.

Display window capture: The displayed images (or a selected image with the name *IM*) can be saved to a .ppm file (*outfilename*) in the imagepath directory using the command:

***captureim** *outfilename* [*IM*]

G. Manipulating images (and graphs) with the *move command

The **move** command is a general object manipulator that works on both images and graphs. It has the form of:

***move** *DestIM*= [-]*A* [*Operation*][*B*]

where *DestIM* is the destination image, *A* is the source image or a numerical value, [*Operation*] is the operation being performed on the source image(s) and [*B*] is either another image or a number, depending on the operation.

Valid operations are:

+ addition

- subtraction (or negation of a single argument)

Frequently x-ray images are accompanied by background exposures of the same duration taken with no signal. If the image files rawdata01.tif and bkgrd01.tif are in the default image directory, then the command:

***move data01=rawdata01-bkgrd01**

takes the difference between rawdata01.tif and bkgrd01.tif and writes it into a new file called data01.tif in the same directory. The file data01.tif contains just the signal from scattered x-rays.

***** multiplication

/ division

***move im2=im1/2** {sets **im2** equal to **im1** divided by 2}

>> bitshift right [*B*] must be an integer

<< bitshift left [*B*] must be an integer

***move im2=im1>>2** {**im2** is set to **im1** right shifted by 2}

% modulo [*B*] must be an integer

***move im2=im1%10** {**im2** is set to remainder of **im1**/10}

: dezinger *A* and [*B*], remove statistically sig. differences from *A* +[*B*]

***move im3=im1:im2**

Resulting image im3 depends on the value of the user variable **dezngtype**:

0 – im3 will be the dezingered sum of im1 and im2 (default).

1 – im3 will be the dezingered average.

2 – im3 will be a zinger map im1 and im2 will be replaced by dezingerd versions of themselves.

An alternate instruction for removing zingers is the ***dezing** command:

***dezing im3 im1 im2** [*additional image files*]

{**im3** set to the <u>average</u> of **im1**,**im2** and any additional image files after the zingers have been removed}

! perform image corrections to *A*, where [*B*] is one of the following:

imi -- intensity correction -- prompts for intensity correction file

imd -- distortion correction -- prompts for distortion map files

imc -- performs intensity then distortion correction

All detectors have some geometric distortions and some variation in sensitivity with position across the face of the detector. These faults can be compensated for using distortion and intensity correction routines. Once correction files have been made for the particular detector being used (see section J) and the paths to these files set with the ***setdist** and ***setint** commands (see section H), then the ***move** command makes these corrections, as in:

***move im2=im1!imc** {**im2** set to intensity and distortion corrected version of **im1**}

H. Avoiding overflows and underflows

The raw data files of most detectors at CHESS use 16 bit unsigned integer values from 0 – 65535. Once image manipulations begin, choices must be made to handle potential overflows and underflows. Adding images at this point gives a possibility of integer overflow. When a subtraction is performed, TVX converts the data type from unsigned integer to a signed integer (-32768 - +32767 for a 16-bit image). This avoids any possible underflow (e.g. a -3 value being treated as +65535), at the expense of possibly truncating data at the high end. This problem can be solved in two ways:

1. The Short data words can be converted to another data type, such as Long, using the *convert command:
   ***convert** *SourceIM DestIM datatype*

TVX data types	Char	Short	Long	Double	Float
Bits per pixel	8	16	32	64	32

   Image manipulations can then be performed on the image files with a greater range of values.

2. The entire image (and its companion background images) can be divided by 2 (or another appropriate scale factor) using either:
   ***move im2=im1/2** {**im2** is set to ½ **im1**}
   ***move im2=im1>>1** {**im2** is set to **im1** right shifted by one}

   Care should be taken to assure that overflows are avoided at each step when performing multiple image operations

NOTE: Often at CHESS, stored data is kept as a 16 bit unsigned integer, even after image correction. Data is processed at a higher precision, a pedestal is added so that the noise about zero signal will not be artificially truncated (the pedestal is often the value 20), then the result converted to a 16-bit unsigned number. Such a strategy is easily implemented in TVX as well.

I. Sample sequence of commands for processing x-ray image data

For a properly corrected image, one should have the following images available:

1. **Raw x-ray images:** One may take multiple, nominally identical images to increase statistics and allow for the removal sporadic bright events due to stray cosmic radiation (as well as other high-energy radiation sources). Assume we have a set of nominally identical images
   imag1.tif , imag2.tif, ... imagn.tif

2. **Background images:** These should match the exposure length of your raw x-ray images. Taking multiple images will reduce the noise of the measurement. One set of background images can be used for a series of x-ray images. One should take a new set periodically, especially if the experiment is sensitive to an absolute measurement of the zero dose level. Again assume we have a set of background images:
   bkg1.tif, bkg2.tif, ... bkgn.tif

The following is a typical sequence of commands for processing this set of images. Note that several intermediate images will be created. One can reuse these temporary image filenames again.

Initialization of TVX:

Open a Linux terminal

cd /home/<*directory path*>	{go to directory containing TVX}
./tvx	{start TVX}
***imagepath /home/data**	{set image path to directory containing image data}

The files used for distortion (both x and y) and intensity corrections must first be set by the ***setdist** and ***setint** commands. Assuming the distortion correction files XDistCorrF.tif and YDistCorrF.tif as well as the intensity correction file IntCorrF.tif are stored in the folder /home/Cfiles, then:

***correctionpath /home/Cfiles**	{set correction files directory path}
***setdist XDistCorrF YdistCorrF**	{set distortion correction files}
***setint IntCorrF**	{set intensity correction file}

sets files for the distortion and intensity corrections.

Display raw image:

***disp imag1 4000 14000 5**

will display the first image, imag1, with "black level" offset of 4000.

Dezinger raw images:

***dezing imagavg imag1 imag2 ... imagn**

will set imagavg to the average of imag1 through imagn with the zingers removed. If there is only one image, skip directly to background subtraction.

Dezinger background images:

***dezing bkgavg bkg1 bkg2 ... bkgn**

will set bkgavg to average background with zingers removed.

Convert dezingered output to type long:

***convert imagavg long1 long** {create long1.tif (32bits)}

***convert bkgavg long2 long** {create long2.tif (32bits)}

This conversion to 32 bits is only one way to avoid image underflow/overflow.

Subtract background from data:

***move long3=long1-long2** {subtract long2 from long1}

Intensity/Distortion correction:

***move result=long3!imc**

uses correction files specified at initialization to correct image.

Display result:*disp result 0 10000 5

displays result with no "black level" offset after background subtraction.

Convert data type:

One can convert result.tif from 32 bits back to 16 bits (to save on storage space for example). If converting to an unsigned short, one should add a small offset to the data before conversion.

***move temp=result+20**

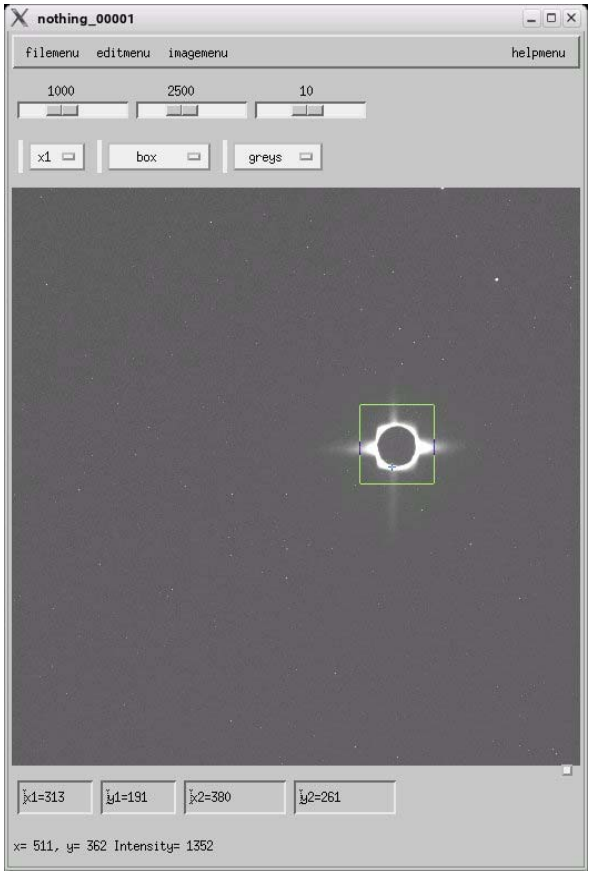
***convert temp short_result short**

J. Using TVX tools for image analysis

TVX comes with many image analysis tools. The selection tools **box**, **butterfly** and **annulus**, in combination with the TVX commands ***integrate** and ***peak**, are used to analyze image data.

1. The **box** tool is used to analyze pixel values within a rectangular area of an image displayed with the ***disp** command. After choosing **box** from the middle pull down menu, the mouse and arrow keys can be used to size and locate the selected rectangle on the image. Entering ***integrate** returns statistics on pixel values within the box, including mean, standard deviation, minimum, maximum, etc. In the image below, the maximum pixel value of ~15K provided by the box tool allows the user to set an exposure time that won't saturate the detector.[3] The tool can also be used to quantify scattering while aligning slits.

[3] nothing_00001, Slit scatter, courtesy of Gil Toombes



2. The ***histogram** command makes a histogram of a region specified by the **box** tool, or by coordinates entered by the user. For example, once an area of an image has been selected with the **box** tool, the command:

   ***histogram 750 900 50**

generates a histogram graph with five bins labeled 700, 750, … 900. Events below 725 go into bin '700', events from 725 to 774 go into bin '750', etc. The command:

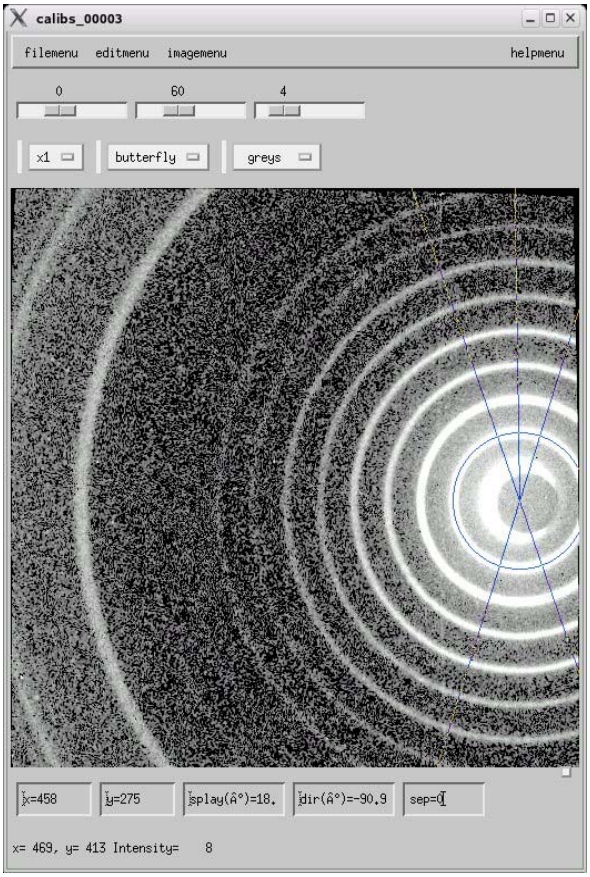
   histogram 750 900 50 100 200 150 250 Hist1

generates the same histogram taken from a box with x1=100, y1=200, x2=150, y2=250, and puts the results in graph Hist1 in the location set by ***grafpath**.

3. The **butterfly** tool is designed to analyze axially symmetric images (e.g. rings) displayed with the ***disp** command. In the following example, the command is used on an image taken with a silver behenate lamellar calibration sample.[4] After choosing **butterfly** from the middle pull down menu, the mouse and arrow keys can be used to size and locate the butterfly pattern on the image. When the mouse cursor displays a cross, clicking and dragging on the butterfly changes its position. Fine positional adjustments are

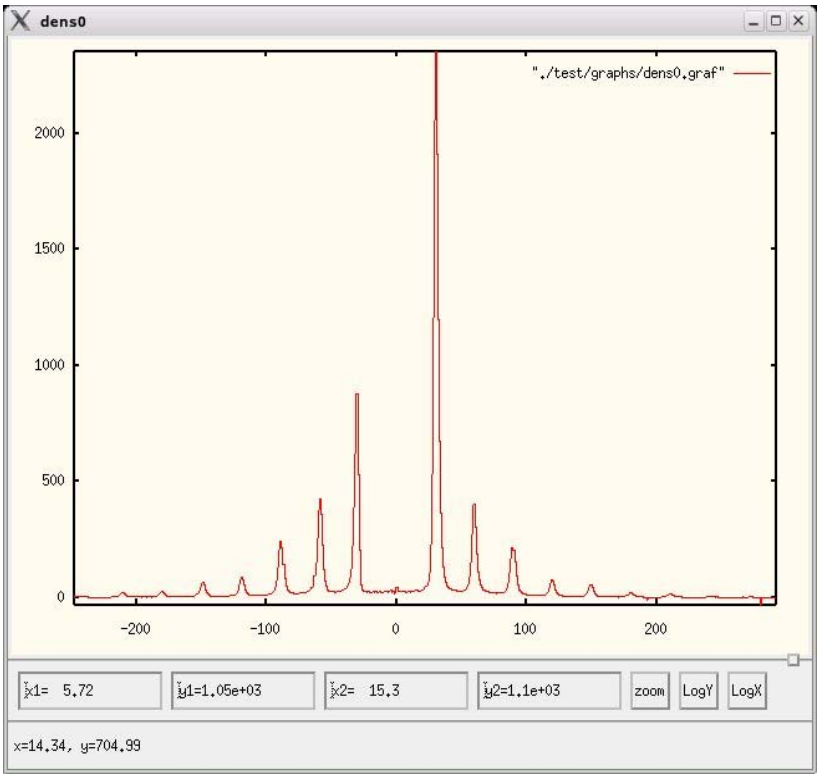
[4] calibs_00003, Silver behenate lamellar calibration sample, courtesy of Gil Toombes

best done with the arrow keys. The concentric circle displayed with the symmetric angles is there solely for aligning the pattern with the data. The size of the circle can be adjusted with the mouse when the cursor is over the circle and a circle cursor is displayed.



Three additional adjustments are used to define the butterfly tool, **splay**, **direction** and **separation**. **Splay** controls the angle enclosed by the lines radiating from the center of the tool, and can be adjusted when the mouse cursor displaying opposed arrows while above the lines. **Direction** is used to angularly rotate the butterfly around its center, and can be adjusted when the green double arrows are displayed with the cursor above the extra line controlling orientation. **Separation** controls the distance between the two opposite angles, and is controlled when the cursor is a double arrow above the center of the tool. **Splay**, **direction** and **separation** can also be changed by typing into their respective value boxes at the bottom of the ***disp** window.

Once the butterfly tool has been set in the desired position, entering ***integrate** generates a graph of average pixel value vs. radial position in pixels, as shown below. Note the **zoom**, **LogY** and **LogX** buttons in the lower right-hand corner, which can be used to zoom in on a section of the graph as well as convert either axis to a log scale.



Entering ***peak** or ***peak dens0** (where dens0 is the name of the graph file displayed in the upper left-hand corner of the graph) starts the peak position fitting routine (note that the TVX ***** prompt is no longer present). Typing **R** (for return) exits the routine and typing **H** (for help) regenerates a listing of the command line alternatives. First, type **X** and enter the x-ray wavelength in angstroms. Then in this case type **L**, to put peak into the lamellar analysis mode. The peaks can then be assigned index numbers. Typing in **1** results in the response: 'left button to set, right to unset, middle to quit'. Click on the first peak, and then with the mouse cursor still over the graph window, click on the middle button. The peak position has been recorded when the two line response 'The center of the peak is at <*peak center in pixels*>' and 'index= <*peak index number*>' has been given. These steps need to be repeated for each of the peaks included in the calibration. Note that peaks to the left of the 0^{th} pixel are given negative indices of -1, -2, etc.

For non-lamellar calibrants, the ***peak** routine has several alternate modes, including Squares Mode, Index Mode, d-Spacing Mode and Spherical Calibrant Mode. Squares Mode (choose by entering **S**) is used when there are peaks from subplanes that are spaced $1/\sqrt{(\text{integer})}$ times the unit cell distance. In Index Mode (choose by entering **I**), one (same as lamellar), two or three indices can be specified, depending on the reflections present. The d-Spacing Mode (choose by entering **J**) allows for reflections from plane spaced at an arbitrary fraction of the unit cell. The mode accepts the inverse of this fraction entered in decimal form. As an example, the second peak of a

hexagonally symmetric calibrant could be indexed with ±3 in Squares Mode, with h=1, k=1 or h=1, k=1, l=1 in Index Mode, as well as ±1.7321 in d-Spacing Mode. Again, a negative value indicates a peak on the opposite side of the middle pixel.

$\frac{d}{d_{h,k,l}}$	Lamellar Mode **L**	Squares Mode **S**	Index Mode (h,k,l) **I**			d-Spacing Mode **J**
			(h)	(h,k)	(h,k,l)	
1	±1	±1	(1)	(1,0)	(1,0,0)	±1.00
$\sqrt{2}$	-	±2	-	-	(1,1,0)	±1.4142
$\sqrt{3}$	-	±3	-	(1,1)	(1,1,1)	±1.7321
2	±2	±4	(2)	(2,0)	(2,0,0)	±2.00
$\sqrt{5}$	-	±5	-	-	(2,1,0)	±2.2361
$\sqrt{6}$	-	±6	-	-	(2,1,1)	±2.4495

Choosing **Y** enters spherical Mode. As in Lamellar Mode, each peak is assigned an index (±1, ±2, etc.) with the 0^{th} order peak in the middle.

Once the peaks have been indexed, entering **F** (for fit) generates a separate box with a list of all the peaks. With each peak is a button to remove any unwanted ones. Checking how closely the peaks are aligned to the expected positions can be done by entering **T** (for tic), which generates vertical lines at the expected peak position in the intensity graph. If the zoom button is used before entering **T**, the tic lines will also appear in the zoom window, giving an enlarged view of the fit to a given peak. Entering **U** (for undo tics) removes the tic lines.

Next enter the calibration routine by entering **C**. The information needed to complete the calibration depends on the type of calibrant. For most calibrants, enter the d spacing of the unit cell in angstroms (e.g. d= 58.37Å for silver behenate). For spherical calibrants, enter the radius of the spheres.

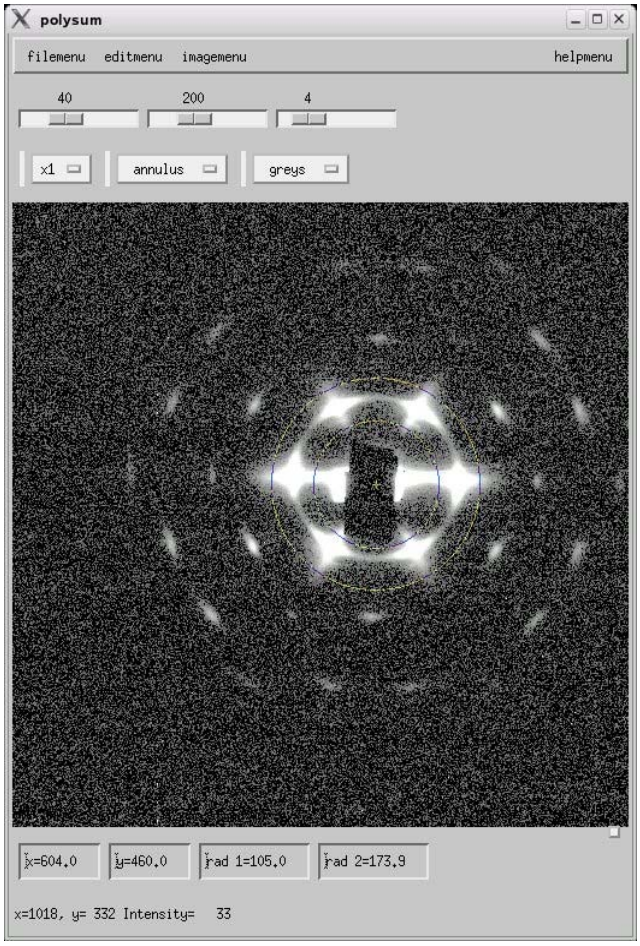
4. The **annulus** tool is designed to analyze the amount of x-ray signal in a circular area while subtracting out the background level in a surrounding annulus. After choosing **annulus** from the middle pull down menu, the mouse and arrow keys can be used to size and locate the pattern on the image. As with the butterfly, when the mouse cursor displays a cross, clicking and dragging on the figure changes its position. Fine positional adjustments are best done with the arrow keys. The size of the inner circle can be adjusted when the mouse is over it and a white single arrow cursor is displayed. The size of the outer circle can be adjusted when a green pointer cursor is displayed. The position and size of the annulus can also be changed by typing values into the x, y, rad1 and rad2 boxes at the bottom of the display window.

Once the annulus tool has been placed in the desired position, entering ***integrate** generates a description of the position and radii of the two circles, as well as a sum of the x-rays in the inner circle and the outer annulus. The TVX variable **intens** is set to the total signal within the inner circle after the average signal between the circles has been subtracted from each pixel.

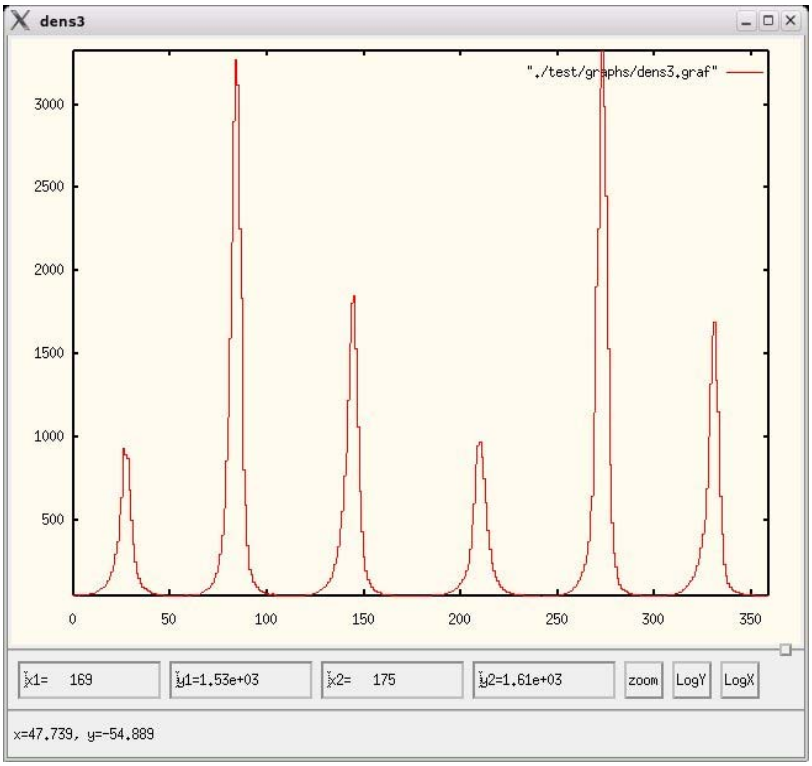
In some cases the user may want to know x-ray intensity as a function of angle around a circular pattern. After using the **annulus** tool to surround the pattern, entering:

***annularint** [*IM*][*graphfile*]

Generates a graph of integrated intensity vs. angle (0º is vertical) as shown in the following two illustrations:[5]



[5] polysum, Hexagonal phase of Kraton D-1102, courtesy of Mark W. Tate



5. The ***dens** command performs the same operations as ***integrate** with some additional controls. The two commands have the formats of:
   ***integrate** [*IM* [*Graph_name*]]
   ***dens** [*IM* [*Graph_name*]] [*Xorigin Yorigin splay direction separation*]

The ***integrate** command takes the current selection tool and performs a context specific integration on the image file *IM* (or the default image if no *IM* is specified) and generates the graph *Graph_name*. If, for example, the butterfly tool is being used, parameters such as the separation, splay, position [*Xorigin Yorigin*] and direction are set by manually manipulating the selection tool on the screen. The ***dens** command allows these parameters to be entered numerically, an especially useful feature when it is used in a script.

K. Using the graphing commands

So far graphs have been generated using the ***integrate** and ***dens** commands. TVX comes with many other commands used to add to and manipulate graphs.

1. The ***grafpath** command is used to show and set the directory TVX is currently using for the reading and writing of graph files. It has the form:
   ***grafpath** [<*graph directory path*>]

2. The ***graf** command can be used to graph up to 3 files using the current graphing mode and has the form:
   ***graf** *filename1* [*filename2* [*filename3*]] [*xmin xmax* [*ymin ymax*]]
   ***graf1** *filename1* [*filename2* [*filename3*]] [*xmin xmax* [*ymin ymax*]]

(graf1 will display in the active graf window, graf will cycle through up to 3 active windows)
If *xmin* and *xmax* are not given, or are both 0, the data is scanned for appropriate limits (same for *ymin* and *ymax*). To specify just *ymin* and *ymax*, set *xmin* and *xmax* to 0 0 as a placeholder.

3. The ***grafset** command sets the graphing mode and line type according to Gnuplot conventions:

***grafset p**	{set line type to points}
***grafset his**	{set line type to histogram}
***grafset lp**	{set line type to lines and points}
***grafset peaker**	{line type points for first file, followed by lines}

Many useful commands have the form ***grafset "***custom string***",** such as:

***grafset "set logscale y"**	{change graph's y coordinate to log scale}
***grafset "set nologscale y"**	{change graph's y coordinate to linear}
***grafset "set grid"**	{turn on graph's grid plot}
***grafset "set nogrid"**	{turn off graph's grid plot}
grafset "set xrange** *xmin xmax"**	{set graph's x range min and max}
***grafset "set autoscale y"**	{turn on autoscale for y coordinate}

4. The displayed graph (or a selected graph with the name *GR*) can be saved to a .ppm file (*outfilename*) in the grafpath directory using the command:
   ***capturegr** *outfilename* [*GR*]

L. Masking images

Often an image contains unwanted artifacts that need to be masked. This can be done in TVX by creating a mask image that has all 1's where the target image is to be kept, and 0's where it will be erased. Once a mask has been made and turned on, operations such as ***integrate** and ***dens**, as well as the selection tools **box**, **butterfly** and **annulus**, apply only to the pixels that aren't masked off.

1. TVX has two commands useful for creating masks, ***pixlfill** and ***mkmask**. Pixlfill fills the pixels in a **box**, **butterfly** or **annulus** selection tool with pixels of a given value. It has the form of:
   ***pixlfill** [*IM*] *val* or ***pixlfill** [*IM*] *val* [*region*]

where [*IM*] is the name of the mask file, *val* is the assigned value (1 or 0) and [*region*] specifies the region of an image that is to be set to *val*. The instruction operates on the specified mask image with the last selection tool used on any image. This feature is useful since the selection tool can be aligned on the image that needs masking while the mask is generated in another file.

When used with the **box** tool, ***pixlfill** sets the pixels inside the box to *val*. When used with the **butterfly** tool, the pixels enclosed by the two sets of outgoing rays are set to *val*. When used with **annulus**, the pixels inside the inner circle are set to *val* unless [region] is set to 2, in which case the pixels between the two circles

are set to *val*. Assigning (x1,y1)(x2,y2) to the [*region*] parameter allows the user to input the x-y coordinates of a box that is set to *val*.

2. The ***mkmask** command takes an image file and for all pixels between two threshold values, sets the corresponding pixels in a mask image to 1, while setting the other mask pixels to 0. It has the form:
   ***mkmask** [*IM* [*IMout*]] *low high*

where *IM* is the image, *IMout* is the mask, *low* is the low threshold and *high* is the high threshold. Having two thresholds has advantages. Setting the *low* threshold to zero results in a mask with all 1's below the *high* threshold. Likewise setting the *high* threshold to the maximum value results in a mask with all 1's above the *low* threshold.

3. In some cases the users may want to combine two mask images using logical operators. The logical function AND (both inputs must be 1 to get a 1 output) can be performed by multiplying the masks together, as in:
   ***move mout=m1*m2** {logical AND operation with result in mout}

The logical function OR (either input can be 1 to get a 1 output) can be performed by merging to masks, as in:

***merge m1 m2** {logical OR operation with result in m2}

A mask can be negated (0's go to 1's, 1's go to 0) with the following:

***move mout=1-m1**

4. The ***maskimg** command is used both to declare and turn on a mask, to inquire about which mask is active and to turn off the current mask. It has the form:
   ***maskimg** [*IM*] or ***maskimg 0**

Entering ***maskimg** *IM* sets the active mask to *IM*. Entering ***maskimg** with no argument returns the file name of the active mask. Entering ***maskimg 0** turns off the active mask.

5. The following is a typical sequence of commands used to mask an unwanted feature from an image file titled im1.

***move msk1=im1**	{make msk1 a copy of im1 so they have the same dimensions}
***move msk1=1**	{set msk1 to all 1's)
***disp im1 500 10000 4**	{display im1 with appropriate values of [*Min*], [*Max*] and [*Scal*]}

If using the ***pixlfill** command, the **box**, **butterfly** or **annulus** tool would then be used to select the area to be masked off.

***pixlfill msk1 0**	{set pixels in msk1 to 0 using position of selection tool in im1}

If using the ***mkmask** command, pixels in msk1 can be set to 1 or 0 depending on the value of the same pixel in im1.

***mkmask im1 msk1 0 8000** {set pixels in msk1 to 0 where pixels in im1 are greater than 8000}

***maskimg msk1** {set active mask to msk1}

Once the msk1 has been created and turned on with the ***maskimg** command, commands such as ***integrate** and ***dens** only apply to the pixels in the image files where the msk1 has a value of 1.

M. Making distortion and intensity correction files (for CHESS staff)

1. Two sets of images are needed to make distortion and intensity correction files. An exposure of a precisely spaced grid of small holes is used to make a distortion correction file. The input values given below are for a mask with an array of 75 μm holes spaced 1 mm apart taken with a detector with a pixel spacing of 58 μm. It is important to have the spot array aligned with the edges of the detector field as closely as possible. A flood exposure taken with a uniform x-ray source held at least one meter from the detector is needed to create the intensity correction file.

2. To make the distortion correction file, enter:
   ***mkdistcorr** /home/<*path*>/<*hole pattern file*> {replaces Sqc in TV6}

Next the routine lists the current values of parameters used to make the correction files. Below is a list of these parameters with values appropriate for the detector described above underlined:

Half-width of box to hold spot: 5 {5<8.6=(1000/58)/2}
Annulus width for spot background subtraction: 0 {not used here}
Allowed variation in spot integral intensity: 200% {more than enough}
Minimum distance (in pix) between spots: 10 {10<17=1000/58}
Minimum number of pixels to perimeter: 5 {close to edge}
Diameter of holes in mask (in mm): 0.075 {75 μm}
Spacing of holes in mask (in mm): 1.0
Demagnification: 55.0 {ratio of source-mask : mask-detector distances (for PSF calc.)}
Focal spot size (mm): 0.10 {size of x-ray source}

Once these values have been modified and accepted, the routine asks the user to define a perimeter (region with useful spots). The union of a circle and a rectangle defines the perimeter. First type E (for edges), then click on the appropriate boundary after typing T, B, L and R (for top, bottom, left and right respectively) to define the rectangle. Then type M (to return to a menu) and then C (for center of the circle). After clicking on a central point, type M and then R (for radius). Clicking on the furthest corner sets the radius to include the entire image.

Next the user will be asked to locate a square of 4 adjacent fiducial spots within the image. The spots should be located near the center of the image and entered in the following order:

(3) (4)
(1) (2)

Once any bad spots have been removed, first enter Y {for all remaining spots are good} and then enter –l to generate a fit to all spots. Next come some additional questions:

Output array of spot intensities: N {can see an array of spot intensities}
Smooth array of spots: Y {smoothing most often helpful}
Statistical graphs: N {can generate statistical graphs}
Calculate distortion correction files (Y/N): Y
Name for delta X image: /home/<*path*>/<*X dist. correction filename*>
Name for delta Y image: /home/<*path*>/<*Y dist. correction filename*>

Now the user has the distortion correction files the use of which is described earlier in this document.

3. To make the intensity correction file, enter:
***mkflatcorr** /home/<*path*>/<*flat field file*> {replaces Cath in TV6}

Again the routine lists the current values of parameters used to make the correction files. Below is a list of these parameters with values appropriate for the detector described above underlined:

Distance from source to detector (cm): 100
Pixel width (cm): 0.006 {0.006 cm ≈ 58 μm}
Pixel height (cm): 0.006
Accept values [Y]?

After these values have been entered and accepted, TVX responds with:

Compute area correction from distortion files?: Y {effect of distortion corrections on intensity corrections should be included}
Enter destination image: /home/<*path*>/<*intensity correction filename*>
Enter x distortion file: home/<*path*>/<*X dist. correction filename*>
Enter y distortion file: home/<*path*>/<*Y dist. correction filename*>

Again the routine asks the user to define a perimeter defined by the union of a circle and a rectangle (see distortion correction section above). Next comes;

Air absorption coeff/cm (enter 0 to ignore): 0 {not important if the correction is less than the nonuniformity of the source}
Silicon absorption coeff/mm (enter 100 to ignore): 100

Routine then makes the intensity correction file and indicates it has been written.

Appendix 1 Filenames, Numbers, Strings, Operators and Special Characters

1. **Image and Graph Filenames**:
   Image and graph filenames can use alphanumeric characters (plus underscore), but must start with an alpha character. The default extension for images is .tif, and the default for graph files is .graf. Filenames are case sensitive, but avoid using variants with the same spelling but different capitalization (the internal hash table will see them as the same file- so you can't have both open at the same time). Likewise, files with the same name, but in different directories cannot be open at the same time.

   The underscore (_) character is also valid in a filename, but ~ ! @ # $ % ^ & * ( ) { } [ ] + = - > < / \ ; : ' " or spaces are not.

   Images filenames will be passed as arguments to various TVX routines. "move" is the general image manipulation routine using operator notation.

2. **Numbers:**
   Integers: 1, -3, etc.
   Floating point numbers: 1.2, 3.4e-10, 0.876
   NOTE: .876 would be invalid, you must have the 0 before the decimal point
   Hexadecimal: 0x2a

   Numbers with leading zeros can be produced by using the Format command

   Format 0 - default variable formatting- no leading zeros on integers
   floats to 4 significant digits
   Format N - integers padded to at least N columns by adding leading zeros.
   Example:
   Format 3
   4 -> 004, 54 -> 054, 6578 -> 6578
   Format N.X - pad integers with leading zeros to be at least N columns
   - floats are given to X significant digits

   Numbers may be manipulated directly on the command line: e.g.
   1+2
   33%5
   with the result printed after pressing return.

3. **Strings:**
   Strings are generally any combination of characters. If there are any special characters included (spaces, operators, etc), the string should be enclosed in quotes.
   abc

"abc"
"abc def%^*"

4. **Variables and macros:**
Variables can be defined using alphanumeric characters, avoiding names already defined by the system (see **menu** output). They can be of type integer, float or string. To define a persistent variable name use

define variablename1=1 for an integer
define floatvariable=10.3 for a floating point number
define stringvariable="abcd" for a string

Variables created with define are in system memory. One can save current definitions to a file with

save "/path/filename.gl"

(.gl is a naming convention adopted for these types of files, although any name could be used)

get "/path/filename.gl" will read these definitions back into TVX
type will list all definitions to the screen
show *variablename* will print the definition of the variable
edit *variablename* will allow modification of the variable on the command line.
forget *variablename* removes the variable from the defined list.

A temporary variable may be defined within a command line just by using an assignment statement, as in

i=32;j=430;disp im0 i j 5

Simply by defining a string to be a valid TVX command, one can create executable macros. For instance

define m="menu"

creates a one letter shortcut for executing the menu command.

5. **Loops and branching:**
Loops and branching can be executed following a C programming language syntax. These commands can be nested.

For Loops:
for(i=0;i<10;i++){*command1;command2;...***}**
Do while:
do{*command1;comand2;...***}while(***condition***)**
While:
while(*condition***){***commands***}**
If – else
if(*condition***){***commands***}else{***commands2***}**
Goto Label:
goto a particular branch in the command line. e.g.
i=1;do{move im0=im[i];if(i>4){goto out}}while(i<20){i++); out:

nokey can be used as a condition when the operator wants to wait for a keystroke, e.g.,

for(i=0;i<100;i++){disp1 im[i];while(nokey)}

would display files im0 to im99 in succession, waiting for a keystroke from the operator.

monitor allows an opportunity for the operator to type at the command line.

for(i=0;i<100;i++){disp1 im[i];monitor}

would allow the command to be typed after each file was displayed.

6. **Operators: reserved characters for command line operations**

		Images	Numbers
+	addition	x	x
-	subtraction, negation	x	x
=	assignment	x	x
/	division	x	x
*	multiplication	x	x
%	modulus	x	x
>>	bitshift right	x	x
<<	bitshift left	x	x
:	dezinger operation	x	
!	image correction operator	x	
|	image size specifier	x	
>	greater than		x
>=	greater than or equal		x
<	less than		x
<=	less than or equal		x
++	increment		x
&	bitwise and		x
&&	logical and		x
|	bitwise or		x
||	logical or		x
^	exclusive bitwise or		x
==	logical compare		x
!	logical not		x

7. **Special characters:**

/ path-level delimiter
; command delimiter
$ comment line- characters after the $ are treated as a comment
\ line continuation – continue command on next line
“” string delimiter
[] indirection

TVX will evaluate an expression within [] and replace with the value on the command line. For example:

i=0;move im6=im[i] will be executed as if one typed

move im6=im0 on the command line

The indirection can evaluate a string or number and can be concatenated.

j="abc";i=6;move im6=a[j]b[i]

will evaluate to "move im6=aabcb6".

(),{} brackets for for/while loops, nested computation

for(i=0;i<10;i++){move im6=im[i]}

(2+3)*4

8. **Glossary files and external scripts (filename.gl)**

Definitions and scripts can be saved and imported via glossary files. In the most basic form, string and macro definitions are saved in these files. One can save the current set of user defined variables and macros via

save *"/path/filename.gl"*

One can read definitions using

get *"/path/filename.gl"*

Note: on reading the file, TVX is actually executing "define" statements. In reality, one can include any executable statement in a glossary file. TVX will then execute lines as a script. One can generate an external script file from the TVX command line by using

appendfile *filename any_text*

When used with indirection notation (evaluation within brackets []), one can generate scripts with sequentially numbered operations

Appendix 2 Listing of TVX commands and system variables

TVX has many other commands, which may prove useful in various applications. Instruction syntax can be obtained in most cases by using the **help** command. Note that all commands are case insensitive.

1. Memory image object input and manipulation commands:
   move - general image input and manipulation routine, also for distortion correction and removing zingers
   captureim - capture an image (and its zoom) as displayed in a .ppm file
   deleteallobjects - clean house without saving
   deleteobj - delete an object from hash table
   dezing - average a series of images with zingers removed
   immirrorh - flip image across a horizontal line
   immirrorv - flip image across a vertical line
   imrotate180 - rotate image 180º
   imrotatel - rotate image 90º left
   imrotater - rotate image 90º right
   listobjs - lists the names of all image objects currently being used
   rawimagein - read a foreign (e.g. non-TIFF) file as an image
   typeheader - type out the header of an object
   typeobj - type out object descriptor

2. Image examination and analysis commands:
   annularint - graph integrated intensity vs. angle around annulus
   box - set tool to box, set x1,y1,x2,y2 and perform an integration
   closedisp - undisplay an image
   dens - make a plot from integrating a strip, bowtie or annulus
   disp - display an image in a new window, if available. Otherwise, reuse oldest
   disp1 - display reusing a window (most recent, if more than one)
   displastimg – display last exposure image taken
   examine - look at pixel values from an 11 x 11 area of an image
   expand - expand an image
   getpixel - gets the value of a pixel given x,y coordinates
   histogram - make a histogram of a region specified by the **box** tool, or by coordinates
   histset - sets the mode of the **histogram** command
   imagepath - set path for image storage
   integrate - perform an integration using the current tool
   maskimg - declare a mask image for use by **box**, **integrate** (with **box** tool), **spot** and **histogram**
   peak - find peak positions and fit to a lattice from rfile data
   psf - convolve an image with a PSF (point spread function)
   regress - regress one image on another
   regresssq - regress the square of an image on another
   showres - show parameters of resolution circles on image

spot - integrate a spot over background selected by annulus tool

3. Image editing commands
   annularavg - make an image that is the annular average of a source image. Use to fill bad pixels in diffraction patterns for illustrations
   bin - bin an image
   byteswap - swap the bytes in a 16-bit image
   clipimg - clip an image to lie between two bounds
   convert - change the image data type (char, short, long and float)
   cut - cut out a box from an image
   editobj - edit the object descriptor
   merge - merge two images by combining nonzero pixels
   mfilter - image or graph smoothing using a filter
   mkimage - make an arbitrary-size image and fill with a value
   mkmask - convert an image to a mask
   mknoiseimg - fill an image with Gaussian noise with specified mean and variance
   noisefill - fill selection with Gaussian noise with specified mean and variance
   paste - paste an image fragment into an image
   pasteraw - paste a foreign (e.g. non-TIFF) file into an image
   pixlfill - fill pixels in a box or annulus with a given value
   rectify - take the absolute value of an image
   setpixel - sets the value of a pixel given x,y coordinates
   smooth, smoothim - smooth an image

4. Image drawing commands
   drawline - draw a line on an image, user supplied coordinates
   drawlines - linedrawing demo
   undrawlines - remove all drawn lines

   drawbox - draw a box on an image, user supplied coordinates
   drawboxes - boxdrawing demo
   boxes - draw boxes on an image under mouse control
   undrawboxes - remove all drawn boxes

   drawcircle - draw a circle on an image
   drawcircles - circle drawing demo
   circles - draw circles on an image under mouse control
   undrawcircles - remove drawn circles

   drawplus - draw a fiducial on an image
   drawpluses - fiducial drawing demo
   pluses - draw fiducials on an image under mouse control
   undrawpluses - remove drawn pluses

   drawpoly - demo of 3 polygons

drawpolys - polygon drawing demo
undrawpolys - remove all drawn polygons

5. Distortion correction commands
   move - general image manipulation routine, also distortion correction
   correctionpath - set path for correction data
   mkdistcorr - make distortion correction files from mask image
   mkflatcorr - make flat field correction file from flood field image
   oblique - alter a flat field file for the obliquity factor
   setdist - set path and name of distortion correction files
   setint - set path and name of intensity correction files

6. Graphing commands
   add2graf - add a point to each of the graphs started by **startgraf**
   capturegr - capture a graph (and its zoom) as displayed as a .ppm file
   closegraf - close a graph
   drawtic - draw a tic mark - position from keyboard
   drawtics - draw tics on a graph demo
   finishgraf - finish the graph(s) and show the result
   graf - graph up to 3 functions, with zoom
   graf1 - make a graph reusing the previous window
   grafline - specify the line style
   grafpath - set path for graph data storage
   grafset - specify the graphics mode
   mfilter - image or graph smoothing using a filter
   smooth, smoothgr - smooth a graph
   startgraf - start 1, 2 or 3 graphs to show data yet to be generated
   tics - draw tics on a graph under mouse control
   undrawtics - remove drawn tics

7. Detector control commands, to be used with CAMSERVER
   bkg - start a background of duration '**expt**'
   cam - send a string to the camera for interpretation by its parser
   cammenu - print the camera menu known to the client
   camreset - reset the camera and stop an exposure in progress
   camstat - display camera status from the camstat file system
   clkrd - reads the clock
   clkset - sets the clock for t seconds
   closecamera - disconnect the camera socket connection
   configpath - set or view the path to config files - a facility for installation-specific code
   connect - attempt to connect or reconnect to camera server
   exp - start an exposure of duration '**expt**'
   exposepath - set the path to be used by the camera
   resetetime - reset to zero the elapsed timer
   telemetry - get telemetry values from camera

timestamp - show the current date and time to 1 ms
wait - wait for exposure to finish, or wait for length of time t seconds
wait4img - wait until the image has been taken
waitauto - automatic wait
waitshowt - wait for length of time t seconds, show a countdown

8. Miscellaneous TVX commands
help - invoke the help system
man - a synonym for help
appendfile - append a line of text to a given file
autoname - set auto-increment file base name and path
beep - sound the system beep
noop - report command sequence text and do nothing
protect - write-protects a directory - useful as last line in a script
round - round floating point value to nearest integer
exit - self explanatory
quit

9. System variables

The following system variables are set by TVX routines (e.g. **box** and then ***integrate**) or are can be set by the user with:

variable=value

camera_process_running - [**MARK TO FILL IN**]
centroid_x - x dimension of counts centroid, set by **box;*integrate**
centroid_y - y dimension of counts centroid, set by **box;*integrate**
counts - total number of counts, set by **box;*integrate**
dbglvl - debug level, normally 0. Level 1 shows passing of variables
dens_dot_product - [**MARK TO FILL IN**]
det_dist - distance from sample to detector, in the same units as **pixel_size**
dezngtype - sets response when dezingering with ***move**, see section G
display_time – [**MARK TO FILL IN**]
etime – actual time shutter was open, used by CAMSERVER
expt - detector exposure time, used by CAMSERVER
intens - after **annulus**; ***integrate**, **intens** is the total intensity within the inner circle after the average intensity between the circles has been subtracted from each pixel.
lambda - x-ray wavelength in angstroms
maximum - maximum pixel value, set by **box;*integrate**
mean - average pixel value, set by **box;*integrate**
minimum - minimum pixel value, set by **box;*integrate**
pixel_size - size of detector pixels in the same units as **det_dist**
queue_busy - [**MARK TO FILL IN**]
spot_int_sq_bkg - [**MARK TO FILL IN**]
stdev - standard deviation of pixel values, set by **box;*integrate**
value – returned value used by several routines

var - variance of pixel values, set by **box;*integrate**
verbose - sets verbosity of TVX's responses to the user, can be 0,1 or 2
xcen - integer version of **centroid_x**
ycen - integer version of **centroid_y**
zngcntrl - [**MARK TO FILL IN**]
zngkut - [**MARK TO FILL IN**]

10. Indirection notation
    Indirection ('[]' notation) allows names and numbers to be substituted in expressions. For example:
    format 3;for(i=1;i<=50;i++){appendfile cvrt mv im[i].tif im[i-1].tif}

will make a shell script called '**cvrt**' that will renumber 500 files from (1 - 500) to (0 - 499)