

View.gtk manual

Lin Yang
lyang@bnl.gov
Dec. 22, 2010

VIEW.GTK MANUAL.....	1
1. Overview	3
1.1 Background information on X-ray Scattering	3
1.2 Availability and OS Compatibility.....	3
1.3 Updates and future plans	4
2. Before you start	4
2.1 Required supporting software packages.....	4
2.2 The .view.gtk directory	4
3. Operations on a single image	5
3.1 Loading data	5
3.2 Changing the appearance of the data.....	5
3.3 Annotating the scattering pattern.....	6
3.4 Calibration of experimental parameters	6
3.5 Grazing incidence scattering	8
3.6 Simple 2D image arithmetic.....	8
3.7 Line profile	9
4. Batch operations.....	9

1. Overview

1.1 Background information on X-ray Scattering

View.gtk is a program for visualizing and processing 2D X-ray scattering data. In X-ray scattering measurements, the scattering intensity is essentially the Fourier transform of the structure (electron density distribution) in the sample. It is important to represent the scattering intensity as a function of the scattering vector, q , which is defined as $q = \frac{4\pi}{\lambda} \sin \theta$, where λ is the wavelength of the X-ray and 2θ is the scattering angle (between the incident X-ray and the scattered X-ray). The scattering patterns collected from area detectors, however, contain the scattering intensity as a function of the pixel position. It is therefore necessary to translate the pixel positions into the corresponding values of the scattering vector, and reorganize the data if necessary so that they can be presented as a function of the scattering vector instead.

If the sample contains structures that do not have a preferred orientation, the scattering pattern is isotropic, or a powder pattern in which the intensity does not depend on the azimuthal angle ϕ . The scattering intensity therefore can be represented as a one-dimensional scattering profile of intensity vs. q . On the other hand, if the structure within the sample has a preferred orientation (e.g. a strand of fiber or structures supported by a substrate), the scattering vector usually need to be break down in to components that are parallel and perpendicular to this preferred orientation. In particular, in grazing incident measurements (GISAXS and GID) substrate-supported samples, this preferred orientation is defined by the normal of the substrate. The components of the scattering vector are often referred to as q_r and q_z , respectively parallel to and perpendicular to the plane of the substrate.

View.gtk provides capabilities to convert powder patterns into 1D scattering profiles (sometimes known as the azimuthal average) and grazing incidence scattering patterns into maps in q_r - q_z coordinates. It also provides functions that are useful for X-ray scattering data analysis, including line profile extraction, simple 2D image arithmetic, annotation in the scattering pattern and batch processing.

Clearly, the translation from the pixel position to the corresponding q or (q_r, q_z) value depends on the geometry of the scattering measurement. The parameters that define the scattering geometry, such as the sample-to-detector distance and the beam position on the detector, must be first defined. The details of the conversions between the pixel position and the q value are described in [ref]. An example will be given below on how to extract these parameters from a scattering pattern from a sample of known structure.

1.2 Availability and OS Compatibility

View.gtk has been written to support user experiments at beamlines X21 and X9 at the National Synchrotron Light Source. It can be downloaded from the website of beamline X9: <http://x9.nsls.bnl.gov/software/view.htm>.

View.gtk is written in C using the gtk+-2 libraries. It has been compiled and run successfully on the following platforms: Linux, Mac OS X (requires fink and X11), Windows 2000.

1.3 Updates and future plans

View.gtk is being updated periodically when bugs are found. The most recent version should be posted on X9 website as soon as changes are made. Clearly the functionalities of view.gtk are limited and because of the interactive nature of the GUI. Many of the functionalities of view.gtk are also implemented in a python package pyXS, which is also downloadable from X9 website. PyXS is more flexible but the use of it will require some knowledge of python programming. While there is no plan to add further functionalities to view.gtk, you can contact if you would like to have something specific added.

2. Before you start

2.1 Required supporting software packages

View.gtk depend on the gtk+-2 libraries (gtk+2, glib2, pango, etc.) for building the GUI and libtiff for reading some data files. Under linux, you can install these packages using a package manager. Under Mac OS X, fink can be used for the same purpose (note that I have only used view.gtk using the X11 packages). Under Windows, your best bet might be to get the gtk+-2 bundle from <http://www.gtk.org/download-windows.html>. Libtiff is available from <http://www.libtiff.org/>. Remember to add the libraries into your environment variable PATH after installation under windows.

You can compile the package yourself if there isn't already an executable file in the package from the X9 website. The Makefile utilizes pkg-config to get library dependencies. Just make sure the paths are correct in the Makefile.

2.2 The .view.gtk directory

View.gtk saves and reads information from configuration files under a directory named .view.gtk (note the . in the beginning of the directory name). This directory is located in the home directory (denoted as HOME_ENV in this document), which is defined by the environment variable \$HOME on unix-like platforms and \$USERPROFILE on Windows. View.gtk will create this directory for you if it does not already exist. You may need to modify files under this directory for some of the operations described below.

3. Operations on a single image

3.1 Loading data

Individual images can be read into view.gtk from the File/Open menu on the control panel. You will need to specify the format of the data file under the Fomat menu. View.gtk currently can read data files in one of the following formats:

- Mar CCD, 16-bit tiff
- Photonic Science CCD, 16-bit tiff (X9 WAXS data , flipped diagonally)
- Roper/Princeton Instrument, 16-bit tiff
- PLATUS, 32-bit tiff
- User-defined raw binary

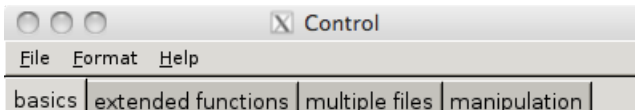


Fig.1. Menu on the control panel

For raw binary data files, the format of the data file must be specified in HOME_ENV/.view.gtk/data_format. View.gtk will get the data format from the first non-comment (not starting with a #) in that file. This line must be in the following format:

```
image_width image_height pixel_format byte_order
```

For instance, the data_format file might read:

```
# this is for the PointGrey firewire camera saved by Coriander
640 480 16I LE
```

The byte order can be either LE (Little Endian) or BE (Big Endian). Currently supported pixel formats are 8I, 16I, 32I (integer) and 32F (floating point).

3.2 Changing the appearance of the data

The scattering intensity in the data file I is mapped to a value $0 \leq v \leq 1$ using one of these formulae:

$$v_1 = \frac{I - \min}{\max - \min} \text{ (linear scale) and } v_2 = 1 - \gamma \log v_1 \text{ (log scale).}$$

The RGB values of the color used to represent the intensity is then interpolated from the colors shown in the 5 color tabs, represent $v=0$,

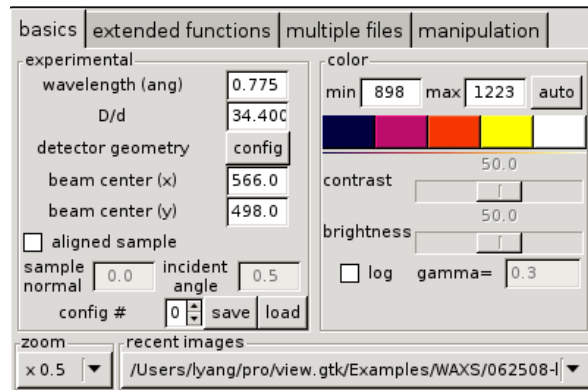


Fig.2. Controls on the “basics” tab for the appearance of the data and the experimental parameters used to define the scattering geometry.

0.25, 0.5, 0.75 and 1.0. When the “auto” button is toggled, view.gtk will try to determine the optimal “max” and “min” values based on the intensity within the part of the data that is visible.

The displayed data can be scaled using the “zoom” drop-down menu. When the displayed image is larger than the size of the display window, you can move the image by clicking on the image and dragging it with the mouse.

3.3 Annotating the scattering pattern

It is sometimes useful to draw reference patterns on the displayed scattering pattern. The reference pattern is defined in HOME_ENV/.view.gtk/ref_pattern, which contain entries of the following format:

(1) A point given by (q_r, q_z)

```
P      qr    qz    color(0-7)
```

(2) A ring corresponding to constant q

```
R      q      color(0-7)
```

(3) A straight (on the q_r - q_z plane) line from $(q_{r,1}, q_{z,1})$ to $(q_{r,2}, q_{z,2})$

```
L      qr,1  qz,1  qr,2  qz,2  N      color(0-7)
```

Here is an example of the ref_pattern file:

```
# beam center
P      0      0      1
# first column of diffraction peaks
L      0.734  0      0.734  3.      50      3
```

Note that positions of the patterns are in coordinates of q , not pixel position. To show this pattern, select “user defined” under the “reference pattern” pull-down list.

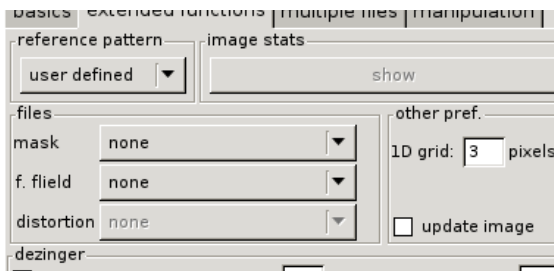


Fig.3. Reference and mask pull-down menus under the “extended functions” tab. The “update image” check box can be used to monitor a data file that is being updated, e.g. during data collection.

3.4 Calibration of experimental parameters

The display of the reference pattern requires the experimental parameters that define the conversion between a pixel position and the corresponding q or (q_r, q_z)

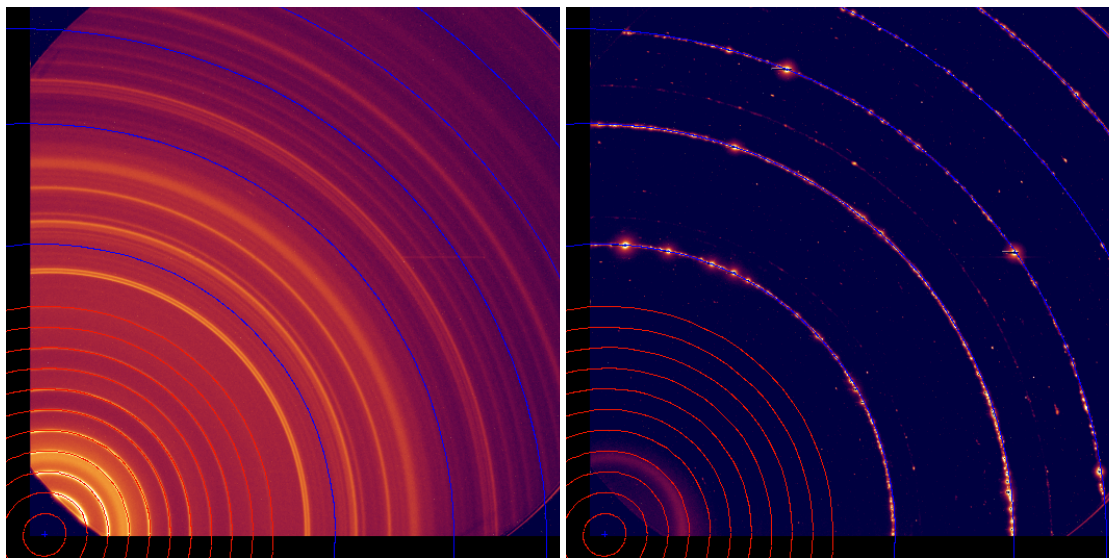


Fig.4. Measured scattering patterns from silver behenate (left) and LaB₅ (right) annotated with the simulated scattering patterns. Note that the rings are not circular since the detector is positioned off-axis from the incidence beam.

value. These values can be found by comparing the scattering pattern from a standard sample with the displayed reference pattern.

The wavelength of the X-ray is usually already known (e.g. known target in laboratory sources, selected using the monochromator at a synchrotron beamline). At X9, for most transmission small angle scattering measurements, the position of the beam can be read direct from the scattering pattern since the beam stop is semi-transparent to X-rays. A standard sample, usually silver behenate, is then used to determine the value D/d , which is defined as the ratio of the sample-to-detector distance (D) divided by the dimension of the detector (d). Note that only the ratio of the two goes into the calculation of the q values, not the values of D and d .

Silver behenate powder produces a series of diffraction rings at q values that equal to multiples of 0.1076 \AA^{-1} . The value of D/d should be adjusted until the simulated reference silver behenate pattern matches the measured scattering pattern. The intensity of these rings drops off quickly beyond 1 \AA^{-1} . For wide angle scattering that sometimes cover q values up to 4 \AA^{-1} , another standard sample is therefore necessary to give more accurate calibration. For instance LaB₄ powder produces rings at $q(\text{\AA}^{-1}) = 1.5115, 2.1376, 2.6180, 3.0230, 3.3798, 3.7024, 4.2751$ and CeO₂ powder produces rings at $2.0154, 2.3262, 3.2877, 3.8539, 4.0253$.

Calibration for a detector that is oriented off-axis from the incident beam (e.g. the WAXS detector X9) requires additional parameters to describe the orientation of the detector. The pop-up dialog for these parameters appears when the “config” button is clicked.

Note that the parameters may not go into effect until you hit the return key after you modify the values. The experimental parameters can be saved by clicking the “save” button at the bottom of the “experimental” box and later restored by clicking “load” button (Fig.2.). There are 6 different configurations that can be saved.

3.5 Grazing incidence scattering

For grazing incident scattering patterns, additional parameters are needed to describe the geometry of the sample: the incident angle of X-ray onto the sample and the possible misalignment of the sample normal from the upright direction (Fig.2.). These parameters only become active when the “aligned sample” box is checked. Once this box is checked, 3 red lines will be shown in the displayed scattering pattern to indicate the expected q_r and q_z axis and the position of the ridge that correspond to zero exit angle for the scattered X-rays (assuming the q_c value for silicon), below which there should not be significant scattering intensity. These lines are useful to help refine the “incident angle” and “sample normal” parameters. See the example in Fig.5.

Each grazing incident scattering pattern can also be converted into a map of intensity in coordinates of (q_r, q_z) instead the pixel position. This operation is triggered by checking the “recp. spc.” box under the “manipulation” tab (Fig.5. right).

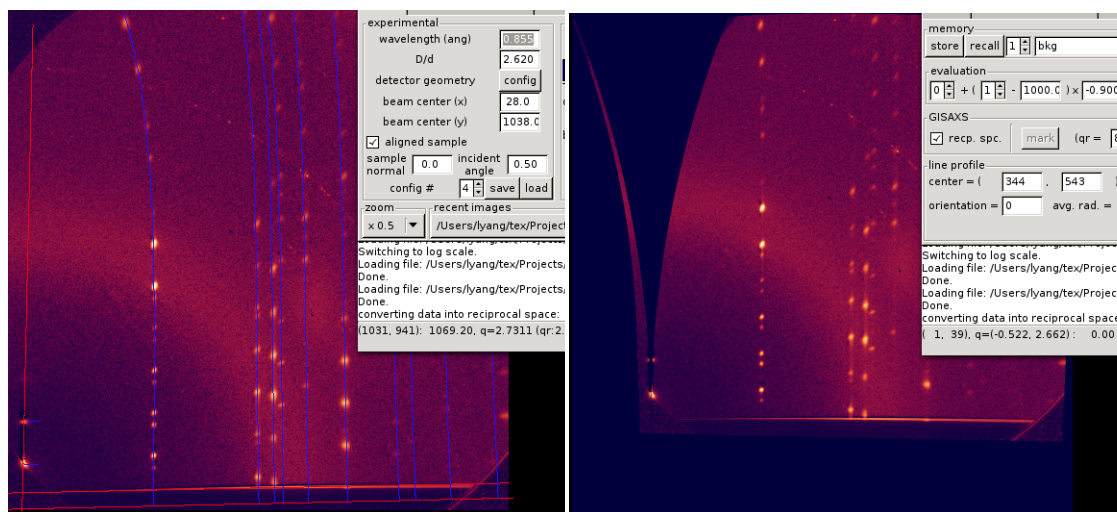


Fig.5. An example of grazing incident scattering pattern displayed (left) with user-defined reference pattern and the visual aids that indicate the q_r and q_z axis and (right) after converted into a q_r - q_z map. Note that while the reference pattern contain lines that correspond to constant q_r 's, the “columns” of diffraction peaks only become apparent after the data is converted to a q_r - q_z map. Also note that due to the off-axis alignment of the detector, the q_r and q_z axes do not appear orthogonal to each other.

3.6 Simple 2D image arithmetic

Scattering patterns can be saved into 5 memory buffers, each can be named to help the user remember the contents of the buffers. Simple arithmetic can then be

applied to the saved data. This is useful, for instance, for subtracting scattering background. The resulted image can be saved into a raw binary data file of type 32F.

3.7 Line profile

Line profiles can be extracted from the displayed 2D image (could be the result from image arithmetic and q_r - q_z conversion). The line is defined by the “center” (in pixel position), the “length” (actual half length of the line, in the unit of pixels), the “orientation” (0 for a horizontal line) and the “avg. rad.” (half width of the line). The intensity in all the pixels along the width of the line is averaged together. Once the user click the “extract” button, the line will be highlighted briefly on the screen and the extracted intensity will be written into the file HOME_ENV/view-extract.dat. The data file contains columns that correspond to the pixel index, the corresponding q value(s) and the averaged intensity, as shown in the example below:

```
# index    intensity    q or (qr,qz)
# center=(401.0, 450.0), len=400, radius= 2.0
-399      1026.14      0.7552    0.0288
-398      1019.42      0.7552    0.0320
-397      1019.10      0.7552    0.0352
... ..
```

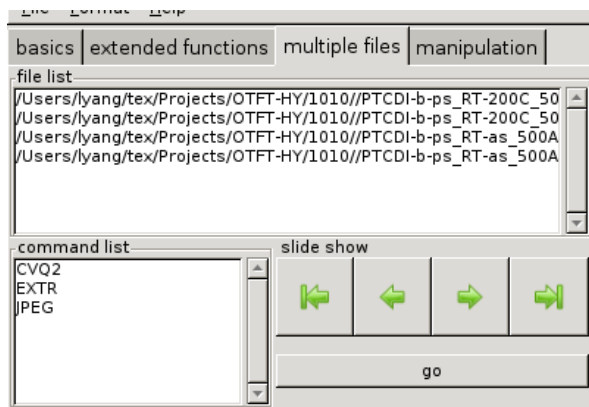


Fig.7. Batch operations under the “multiple file” tab.

created using any text editor and should have only one entry on each line.

Load the list of files to be processed by clicking in the “file list” area, a file dialog will pop up. Do the same for the batch “command list”. Once the file list is loaded, the user can browse through the list using the buttons in the “slide show” box. The permitted batch commands are:

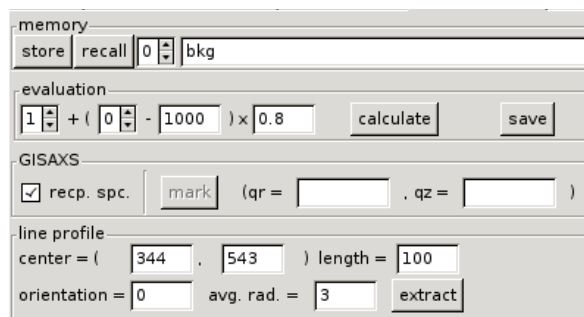


Fig.6. The “memory”, “evaluation” boxes for simple arithmetic operations and the “line profile” box for extracting the scattering intensity on a line. All under the “manipulation” tab.

4. Batch operations

View.gtk provide some capabilities to quickly process a large set of data files. Two separate text files must be defined to perform batch operations: one contains the name of the data files to be processed, and the other contains a list of operations to be performed for each data file. Both files should be both simple text files that can be

(1) CVQ1

This is the command to perform azimuthal average for powder patterns. The results will be saved in a text file, with the same file name but extension .ext in the same directory as the 2D data file. The data file contain 3 columns: q , averaged intensity and error bar. The error bar is the standard deviation calculated from the intensity in pixels included in the data point.

It is important to exclude regions of the scattering pattern that do not include isotropic scattering intensity, for instance behind the beam stop and in the corners of CCD data that do not register scattering intensity. This can be achieved by using a mask file, loaded using the “mask” pull-down menu shown in Fig.3. The mask file is a plain text file containing entries the represent geometric shapes. The following shape are currently supported:

(a) A circle: center at (cx, cy)

c c_x c_y radius

(b) A hole (inverse of a circle):

h c_x c_y radius

(c) A rectangle: center at ,

h c_x c_y width height rotation

(d) A fan:

f c_x c_y azimuthal_start azimuthal_end

Note that all entries in the mask file use coordinates of pixel position, not q . Here's an actual example:

```
# mask for Mar CCD data collected from protein solution samples
h      512      512      510      # corners of the image
c      708      489.5    16.5      # beam stop
r      708      490      150       10      0      # parasitic scattering
r      708      490      13        150     0      # parasitic scattering
r      863      524      330       7       -17     # support for the bs
```

(2) ANG1

Instead of performing the azimuthal average, which re-organize the scattering intensity into a 1D function of intensity vs. q , ANG1 reorganize the data into a 1D function of intensity vs. the azimuthal angle. The result is saved into a text file with extension .ang. It is often useful to create a mask to include only pixels within a band between concentric circles (use c and h entries) in the ANG1 operation.

(3) EXTR and EXTB

These two commands are similar to the function of clicking on the “extract” button under the “manipulation” tab and will produce files of extension .ext in the same directory as the 2D data files. The difference between EXTB and EXTR is that EXTB will subtract a background intensity for each data point, which is calculated from the intensity in the pixels located immediately outside of both sides of the line and within half width of the line.

(4) JPEG

Save the displayed data as JPEG files.

(5) SAVE

Save the displayed data. Same function as clicking the “save” button under the “manipulation” tab.

(6) SUMB

Like EXTB, but sum up the intensity within all pixels under the line and write the result into HOME_ENV/view-sumb.dat . This command may be useful to track the intensity change in a series of scattering patterns.

(7) SUMM

Sum up all the scattering patterns in the file list.

(8) CVQ2

Performs the same function as clicking the “recp. spc.” checkbox under the “manipulation” tab.