PETS – process electron tilt series: a brief tutorial

1. Introduction

PETS is a program for processing a series of diffraction images. It is intended for electron diffraction. In principle it could be used for x-rays or neutrons, too, but for these sources dedicated software is available with much more options and better interface.

PETS is not an officially distributed piece of software, and therefore it can have all problems you can think about: bugs, lack of internal tests for user errors, lack of diagnostic messages if something goes wrong, lack of documentation... On the other hand, I am using the program myself, and therefore there is a reasonable guarantee that the program is not producing a complete junk (at least not always...).

PETS is an old fashioned command line program with no graphical interface of its own. However, it can call ImageJ and send the images to ImageJ, which shows them in real time. This combination is a viable alternative to a genuine graphical interface. To enable the communication between PETS and ImageJ you have to install ImageJ and insert the path to ImageJ into the system variable PATH (not trivial under Windows...). Then when PETS is started, it creates a macro for ImageJ named pets_macro.ijm, and ImageJ is automatically invoked with the macro as an argument. When this procedure is successful, a black image with a white circle opens as a sign that the communication has been established.

This brief manual has – apart from the introduction – two parts. In the first part I describe the format of the input file and the parameters that must be determined before PETS is started. In the second part the standard run and additional options of the program are described.

2. The input file

The input file for PETS is a free format ASCII file that specifies the parameters for the program. This is a fairly complete input file using all useful keywords:

```
lambda 0.0335
Aperpixel 0.00223
dstarmax 1.4
omega 12.56
delta 0.01
center AUTO
phi 1.0
bin 2
reflectionsize 22
pixelsize 0.015
I/sigma 5.0
graphics yes
centering I
noiseparameters 3.5 38
reconstruction
hk0 1 0 0 0 1 0 0 0 0
hk1 1 0 0 0 1 0 0 0 1
hhl 1 1 0 0 0 1 0 0 0
endreconstruction
reconstructionparams 0.003 0.01
```

```
beamstop yes beamstop.xyz 995 1010
mask
971 1000
1112 942
1257 975
2048 975
2048 1063
1257 1063
1112 1080
971 1050
971 1000
endmask

imagelist
data\PED-0000.tif -00.00  0.00
data\PED-0100.tif -01.00  0.00
data\PED-0199.tif -01.99  0.00
etc. etc.
endimagelist
```

The name of the input file can be arbitrary, but I recommend giving it an extension „.pts“. AFAIK this extension does not clash with any standard extension, it is thus easy to locate the files, and, if you want, you can easily create a file association between pts files and PETS. The part of the name before the extension is used as the base for all files produces by PETS.

The keywords are case sensitive. Their description follows:

- **lambda**: relativistic wavelength of the incident electrons in angstroms. 100kV=0.0370A, 120kV=0.0335A, 200kV=0.0251A, 300kV=0.0197A
- **Aperpixel**: Scale of the images given as the size of one pixel in reciprocal angstroms
- **dstarmax**: Resolution limits as d*, in Ångstroms
- **omega**: orientation of the tilt axis of the sample holder with respect to the positive horizontal axis of the image. Must be estimated either by calibration, or by visual inspection of the series of images. Should be close to the correct value within +-20 deg. Can be refined later in PETS.
- **delta**: Inclination of the of the sample holder axis from the horizontal plane. Should be set to zero, can be refined in PETS, usually very close to zero even after refinement (+-1deg or less).
- **center**: Position of the primary beam on the image in pixels. Value “AUTO” means PETS tries to determine the center automatically. This often works, but occasionally fails. In such case you have to give explicitly the coordinates of the center in the image, horizontal first, vertical second. Should be read on the image at the lowest tilt angle (the first image to be treated by PETS). The center is then refined for each image separately during peak search.
- **phi**: Precession angle in degrees used during the data collection. Can be zero, of course.
- **bin**: Binning of the images prior to any treatment. Saves time, makes nicer pictures, and seems to make the whole analysis somewhat more robust. bin 2 is a good choice from my experience for most data sets. ***Important:*** All parameters in the input file that are given in pixels refer to the original, unbinned images. This concerns Aperpixel, center, reflectionsize, mask.
- **reflectionsize**: Diameter of the spots on the images in pixels. Used for peak search and integration, should be large enough to encompass also the strong spots, but small enough to avoid overlaps of neighboring spots.
- **pixelsize**: size of pixels in the 3D reconstruction of reciprocal space. In reciprocal angstroms. Values between 0.005 and 0.015 are usually reasonable.

- **I/sigma**: Intensity over sigma ratio for peak search. Only peaks with integrated intensity over sigma larger than the given value are saved in the peak list. Default value is 10. Decrease this number, if you are seeing too few picked peaks, and increase it, if you see noise being picked in the images.
- **graphics**: yes/no. Switches on/off the sending of the images to ImageJ.
- **centering**: Lattice centering. Used in integration to skip reflections absent due to the centering. All standard symbols are acceptable: A, B, C, I, F, Robv, Rrev. User-defined centerings are also possible (ask me if you need it).
- **background:** a constant that is subtracted from each image. Use "background 32768" to correct for the offset present in unsigned tifs.
- **Noiseparameters:** two parameters for the determination of $\sigma(I)$. The first value is $G\gamma$, the second is ψ. The uncertainty on each pixel γ is then calculated as:

  $$\sigma^2(p)=G\gamma p+\psi$$

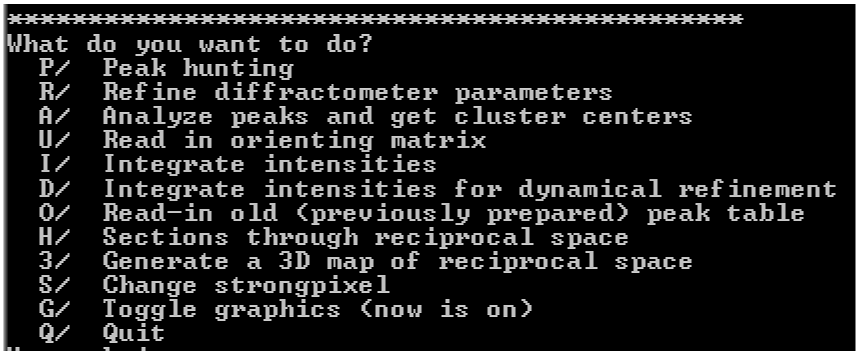
  For more information see Waterman, D. & Evans, G. (2010). *J. Appl. Cryst*. **43**, 1356-1371.
- **reconstruction – endreconstruction**: multiline keyword. Each line contains a specification of one layer in the reciprocal space that should be reconstructed by PETS. The format is filename basvec1 basvec2 origvec, where each vector is given by three reciprocal-space coordinates. The first two vectors are the two vectors defining the orientation of the plane, and the third vector defines the origin of the plane. Example of the definition is shown above in the sample input file.
- **reconstructionparams**: Parameters for the reconstruction of the reciprocal-space layers. Two values, both in reciprocal angstroms. The first is the pixel size in the reconstructions, the second is the half-thickness of the slab around the exact layer position that is projected onto the layer. Taking a too small pixel size and/or too small slab thickness leads to "holey" reconstructions, where some pixels do not have any intensity. The default value for pixel size is equal to the value of the pixelsize keyword, default for the slab thickness is 2*pixelsize.
- **beamstop**: yes/no. Specifies, if the beamstop was used during the data collection. If no, the center of each pattern is located from the strong primary beam. If yes, Friedel pairs are used to locate the center of the image. The latter option can lead to problems, especially if only a few reflections are visible in the patterns. Manual intervention might be required by the program in such cases.

  Optionally, beamstop mask can be defined in an external file (see keyword mask-endmask below for the format). The coordinates of the points of the polygon defining the beamstop should be defined relative to a reference point – typically the tip of the beamstop. Then it is possible to give the file name of the beam stop followed by the coordinates of the reference point, and to avoid defining the beamstop mask explicitly. The full syntax is thus beamstop yes filename.ext x-coordinate y-coordinate (see real example in the sample file above).
- **mask – endmask**: Multiline keyword. Each line contains the coordinates of one vertex of the polygon defining a mask. Necessary if you have unpleasant features in the images that you want to avoid. Not necessary for masking just the strong primary beam – this is avoided automatically. The polygon need not be closed – if it is not, PETS automatically closes adds the last edge of the polygon.
- **imagelist – endimagelist**: Multiline keyword. Each line contains one image. The format is: name alpha beta. Name is the file name of the image (including possible relative or absolute path), alpha is the tilt angle, and beta is the possible second tilt value of the double tilt holder. The last value should be zero, if you tilt only around one axis, but it must always be given. The only format PETS can read is 16bit unsigned TIF. If you have signed TIFs (the diffraction image

appears grey, if you open it in a normal image viewer), add a line “background 32768” in the input file to subtract the false background that is created by reading a signed TIF as unsigned.

3. Description of the program

You can start PETS either from the command line, or by associating the input file extension with PETS, and then double clicking the input file. In any case you will get this menu:



```
**********************************************
What do you want to do?
  P/  Peak hunting
  R/  Refine diffractometer parameters
  A/  Analyze peaks and get cluster centers
  U/  Read in orienting matrix
  I/  Integrate intensities
  D/  Integrate intensities for dynamical refinement
  O/  Read-in old (previously prepared) peak table
  H/  Sections through reciprocal space
  3/  Generate a 3D map of reciprocal space
  S/  Change strongpixel
  G/  Toggle graphics (now is on)
  Q/  Quit
```

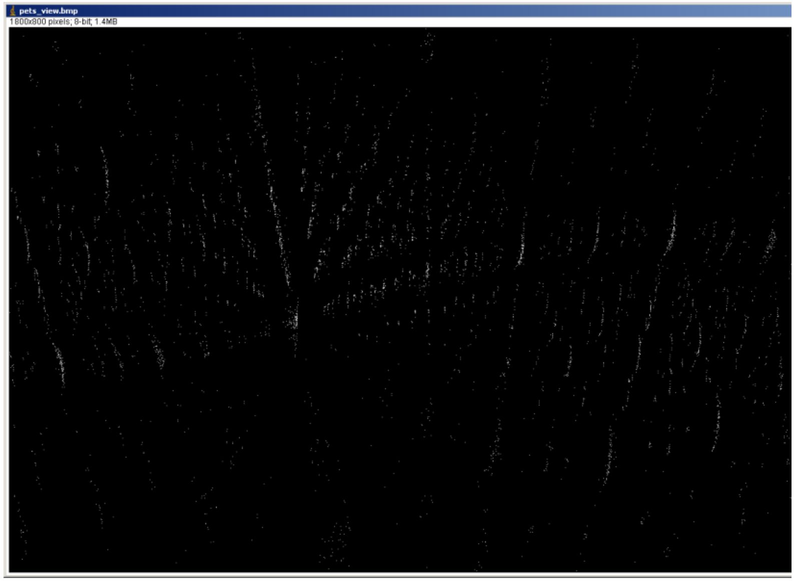
PETS offers the following possibilities: peak hunting, refinement of the angles omega and delta, peak analysis (clustering and difference space generation), integration of intensities, integration of intensities for dynamical refinement, calculation of a 3D distribution of intensities in reciprocal space, and reconstructions of sections through the reciprocal space (also known as reconstructed precession images or unwarping). The one important item missing in the list – the indexing – can be done using Jana2006, and PETS reads in the output from Jana2006.

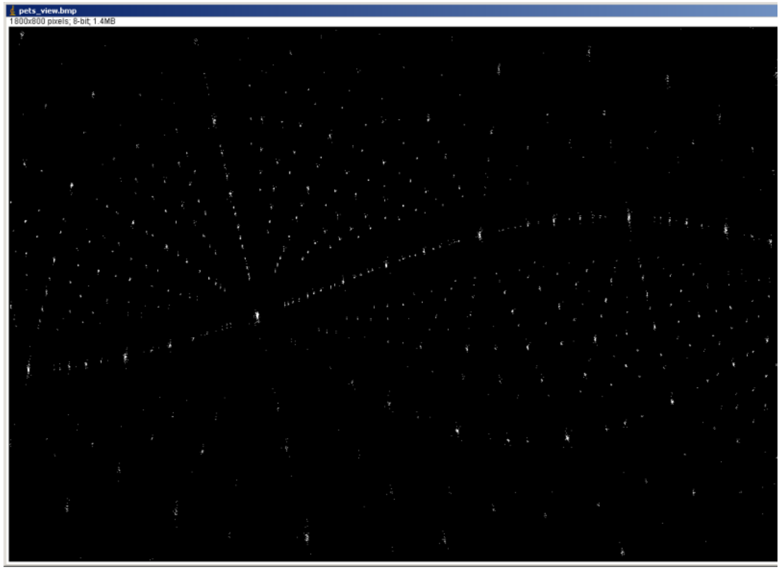
The standard procedure is: Make peak hunting; refine diffractometer parameters (angles omega and delta); analyze peak table and prepare input for Jana2006; Index the lattice in Jana2006; read-in the orientation matrix in PETS; integrate intensities; make reconstructions of some layers of reciprocal lattice and/or 3D map of intensities for checking purposes.

The individual items in the menu will now be described in detail:

- **P/ Peak hunting**: Searches for peaks in each image, using the threshold I/sigma. The critical issue is the location of the center of the diffraction image. Without the beam stop there is no problem and PETS locates the center automatically. With the beam stop, PETS locates the center by identifying Friedel opposites. For various reasons the center can move from one image to the other, and it is non-trivial to recognize such shift correctly and robustly. Therefore, if the program cannot locate the center, or locates a considerably shifted center, then it writes out a warning, and waits for user input. If the center shown on the current frame (grey encircled cross) is on the correct position, pressing enter confirms the center and the program continues. Otherwise pres „n“, and enter the correct coordinates of the center. It is usually possible to determine the center from the image in ImageJ.
  After peak hunting two files are created: jobname.rlp contains the raw peak list that is used for all subsequent analysis, file jobname.cor is a peaklist corrected for the drift of the pattern center, and jobname.cenloc contains the refined positions of the centers of each image.

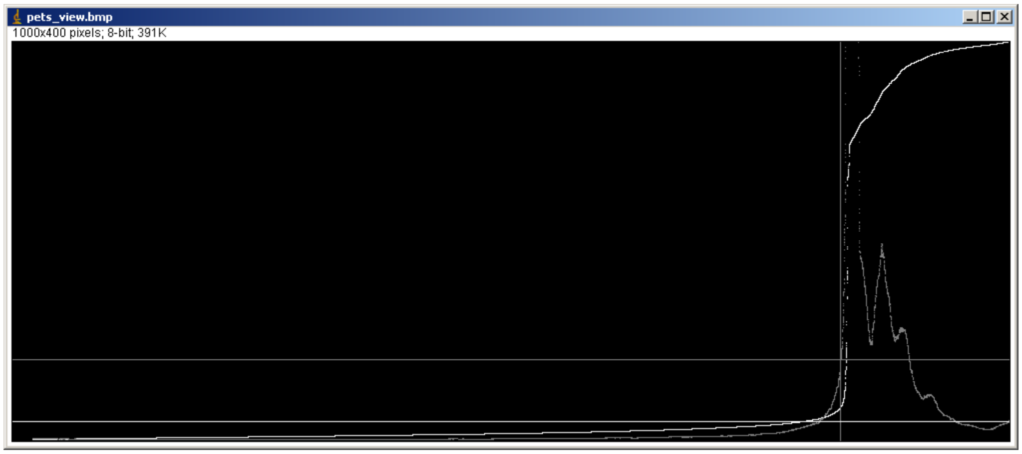
- **R/ Refine difractometer parameters:** The angle omega (angle between the projection of the tilt axis on the image and the horizontal axis) depends on the camera length and on the exact focusing conditions. Therefore it cannot be exactly calibrated, and must be refined for each data set. The details of how this works are beyond the scope of this tutorial, but you can easily asses the quality of the data and accuracy of the parameters by looking at the cylindrical projection of the peak positions that is shown by PETS during the refinement. Below are two examples of such plots. The first one is made with an omega that is off by 10 deg, the second with the correct omega:





  The procedure is long, but once you do it, you can enter the correct omega into the input file, and you do not have to re-refine the parameters for the same data again.
- **A/ Analyze peaks and get cluster centers**: The raw peak list is usually too noisy to be used for indexing directly. Therefore, PETS analyzes the peak list. This analysis proceeds in two steps. In

the first step all reflections measured on subsequent frames are clustered, and the centers of the clusters are used instead of individual peak positions. For this step the limit for the clusters must be determined. To check that the program did this correctly, the program shows a plot of inter-peak distances sorted from the shortest to the longest. The plot looks like this:

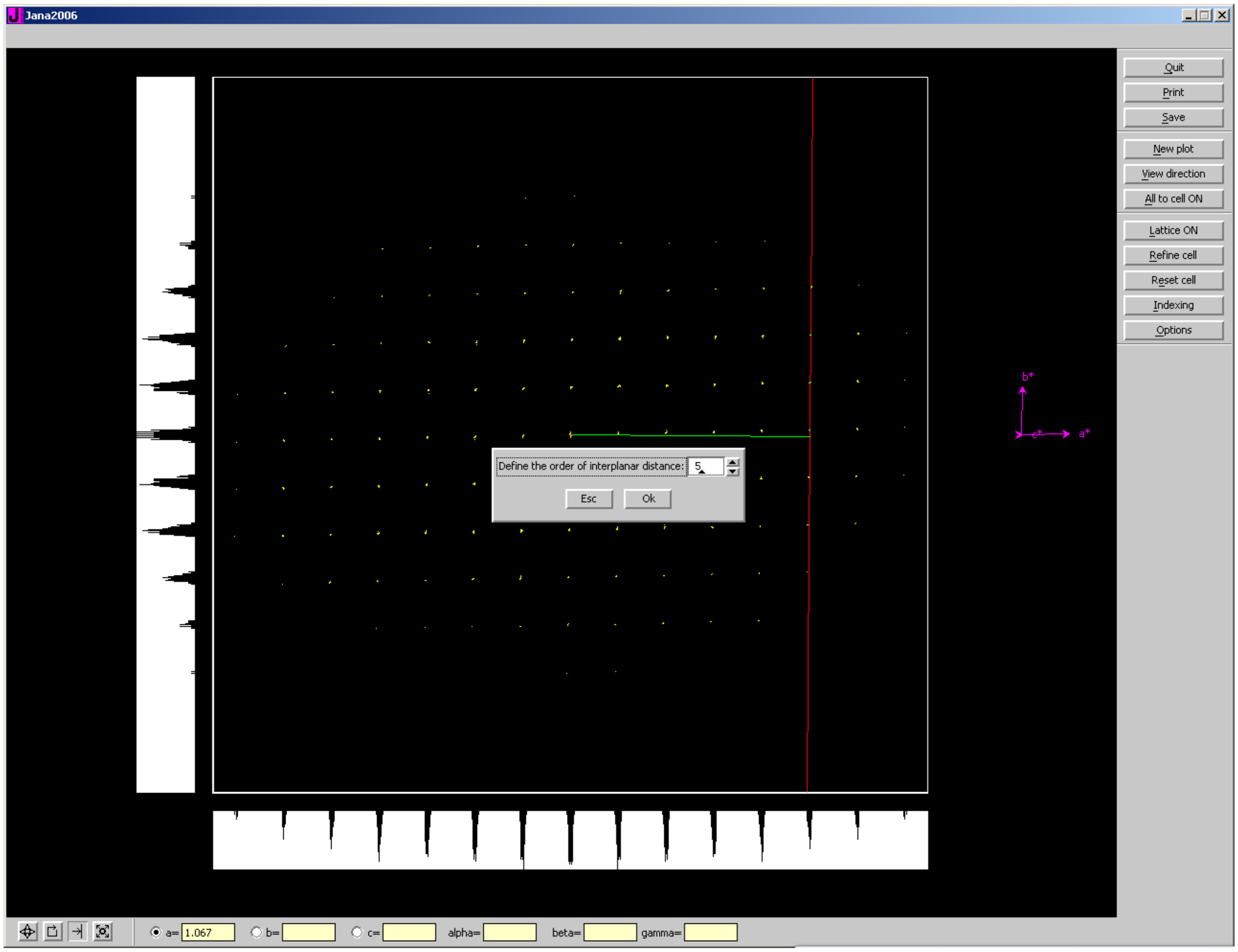


If everything works fine, you should see a clear jump in the distances accompanied by a peak in the derivative (grey in the plot).

In the next step, PETS calculates difference vectors between the peaks. Then the same clustering procedure is applied to the group of difference vectors, and again a plot of distances is shown for checking the correctness of the choice. The jump in the second plot is usually even sharper than in the first plot.

At the end of this procedure three files are produced: jobname.xyz contains the “cleaned” peak list, where individual peaks are replaced by the cluster centers. jobname.diff contains the difference vectors, and jobname.clust contains the list of cluster centers from the difference vectors. As a rule, if everything works fine, the file jobname.clust is the best for indexing. At the end, however, the lattice should be checked against jobname.xyz to make sure nothing was missed or messed up during the clustering procedure. Two notable cases where jobname.clust will not work are twins and (strongly) incommensurately modulated crystals. Do not forget to check the xyz file by reading it in Jana2006 (see the next paragraph) and comparing with the lattice determined from the clust file. Only strong peaks are used for clusters, and there is a danger of missing some superlattice reflections.

- **Indexing:** This can be done manually in Jana2006. No need to close and reopen PETS! Just open Jana2006 with any structure, and make sure the working directory is the same as the location of the pts file. I have a dummy structure (m40 and m50 of any structure is enough) that I copy to the current working directory. Once Jana2006 is open, go to Tools>Graphic>Indexing. Press “New plot” and open the file jobname.clust. You will see a cloud of points, and it should be fairly easy to locate the special directions by rotating the cloud. Once you find a nice projection, you can switch to indexing mode, and drag the directions of special planes by mouse to define the lattice parameters. This is a screenshot of this stage, when the lattice parameter *a* is specified:

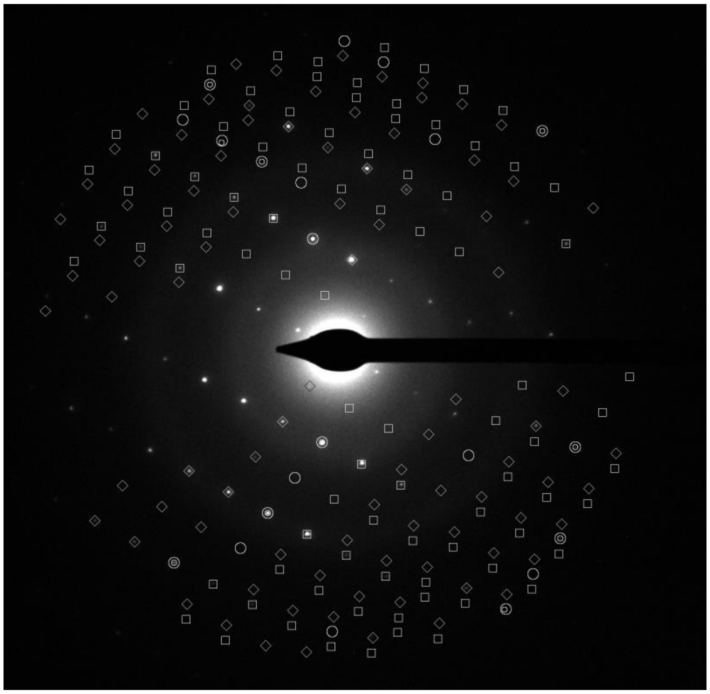


Define the order of interplanar distance: 5

● a= 1.067 ○ b= ○ c= alpha= beta= gamma=

Once you define all three directions, you can refine the cell by pressing the button “refine cell”, and then again “refine cell” in the dialog. There are many options in this dialog as well as in the main window. Most of them are fairly easy to understand, if you play a bit with the tools. If you get stuck with this point, just write me an e-mail and I will write you more detailed instructions. Once you are happy with the refined cell, press Quit and exit the indexing routine. At this point a file jobname.smr is created that contains the orientation matrix.

- **U/ Read-in orienting matrix**: reads the file jobname.smr and import the orientation matrix UB. The matrix and the lattice parameters calculated from it are written out for check.
- **I/ Integrate intensities**: The climax of your efforts! Using the parameter reflectionsize, the centering (if present), and the imported orientation matrix the reflection positions are predicted and their intensities are integrated. PETS shows an image of each frame with an overlay of diamonds, circles and squares, like this:



A diamond means the reflection is approaching exact Bragg condition, circle is shown on the frame nearest to the Bragg position, and square after the Bragg position. If there is a small circle inside the large circle, PETS found a significant intensity at that position. In such case the large circle is the predicted position and the small circle is centered around the real maximum in the image.

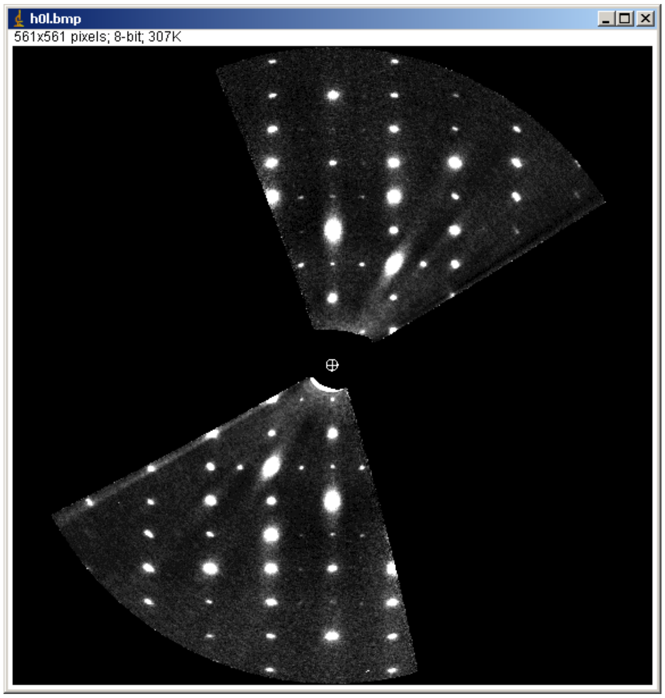
At the end of the procedure a file jobname.hkl is created, where the intensities are written out in the format 3i4,2f10.2, plus some more information after the first five columns. Moreover, a file jobname.cif_pets is created that it directly readable by Jana2006.

- **D/ Integrate intensities for dynamical refinement**: This option integrates the reflection intensities frame by frame (i.e. no integration across frames). The result of the integration is a file jobname.cif_pets with integrated intensities suitable for dynamical refinement (well, if your sample is suitable for this purpose, of course).

Further options:

- **O/ Read-in old (previously prepared) peak table**: If you want to repeat the peak analysis or the refinement of the parameters, you can read-in the peak table, and you do not have to rerun the lengthy peak hunting. However, this will work only if you have not changed the number of images and the binning. You should be able to change other parameters like Aperpix, phi or lambda, though.
- **H/ Sections through reciprocal space:** A very useful option for presentation, symmetry determination and checking the quality of the data. You can specify any section through the reciprocal space – see keyword *reconstruction* above. If you invoke this option without giving a list of layers in the input file, the program will offer you to calculate a standard set: hk0, hk1,

hk2, h0l, h1l, h2l, 0kl, 1kl, 2kl. Thus, if you are interested only in these sections, you do not have to specify them in the input file. An example of such a reconstructed layer is here:



Files are saved as 16 bit greyscale pgm files. Use e.g. ImageJ to view the files.

- **3/ Generate a 3D map of reciprocal space**: Another useful option for getting an overall picture of the reciprocal space. The program will take all images and calculate the intensity distribution in cube with the dimension 2*dstarmax. The data are exported in the format .xplor (large file!) that can be viewed in 3D by software VESTA or UCSF Chimera. The result looks for example like this: