Clip User Manual

Olaf J. Schumann

March 25, 2009

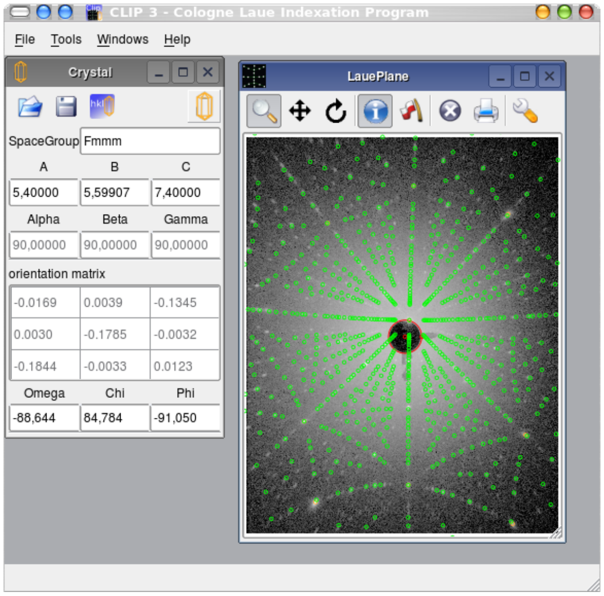
Contents

Clip, short for the Cologne Laue Indexation Program, is a software for the analysis of Laue images. It is provides a semi-automatically indexing function, the possibility to further refine a solution and allows to calculate the required rotations to reorient the crystal. It supports multiple crystals and projections at the same time. A unique feature is the native support for the 12-bit gray scale images of the Fuji BAS image plate scanner, which is used at the institute Laue camera. Here it has been used very successful since 2006.

The software is written in the Python language [1] for a reduced development time. Most code was first implemented in Python and only speed critical parts were rewritten in C++ and exposed to Python with the SIP interface generator [2]. The graphical user interface is realized with the help of the Qt [3] toolkit and it's python bindings PyQt [4]. The use of the cross-platform toolkit Qt and Python allows the use on Linux, Windows and Max OS X computers. For the Windows platform, a stripped down install package including an embedded version of the python interpreter is available. This withdraws the need for a full python installation including PyQt, numpy, scipy and the python image library. The program is licensed under the terms of the Gnu General Public License [5] and is publicly available at http://clip.berlios.de.

1 The Program

1.1 Main Window



SpaceGroup Fmmm

A	B	C
5,40000	5,59907	7,40000

Alpha	Beta	Gamma
90,00000	90,00000	90,00000

orientation matrix

-0.0169	0.0039	-0.1345
0.0030	-0.1785	-0.0032
-0.1844	-0.0033	0.0123

Omega	Chi	Phi
-88,644	84,784	-91,050

Figure 1: Clip main window with crystal definition and a Laue Plane

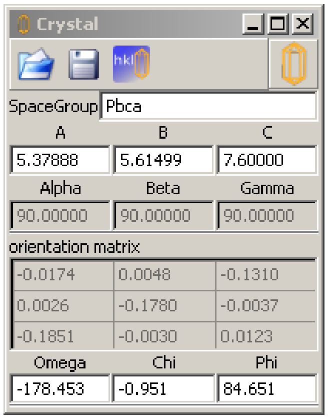
Figure 1.1 shows the main window of Clip with the two principal subwindows, the crystal definition and the Laue projection plane.

The work paradigm is to define one or several crystals and connect it to projectors, which then display a pattern according to their type. One crystal could be connected to possibly many projectors, while a projector is only connected to a single crystal. The majority of cases will only employ a single crystal and a single projector. As said before, the projector determines the type of projection. Up to know, a stereographic projection of reciprocal lattice vectors and a planar projection of the emergent wave vector (the Laue photograph on a flat film) are Implemented. But other types of projectors could be written, like a projection on a cylindrical detector or a gnomonic projection.

The Crystal and projector windows could be played from the File Menu. Additionally, the complete workspace could be saved and restored. The Workspace includes all crystals and the projectors connected to them. The window layout and all relevant data are saved to a xml-file. This allows to save the result of an indexing, including the refined parameters and the marked spots.

The Tools menu contains the tool windows, which are described below in their occurrence in a typical workflow.

1.2 Crystal Window



SpaceGroup Pbca

A	B	C
5.37888	5.61499	7.60000

Alpha	Beta	Gamma
90.00000	90.00000	90.00000

orientation matrix

-0.0174	0.0048	-0.1310
0.0026	-0.1780	-0.0037
-0.1851	-0.0030	0.0123

Omega	Chi	Phi
-178.453	-0.951	84.651

Figure 2: Crystal parameter input window.

Figure 1.2 shows the input box for the crystal parameters. The three tool buttons allow th save and load the crystal definition to a xml file and to start the indexing procedure respectively. The xml-file contains the space group, the lattice parameters and the three Euler angles. The crystal icon in the upper right corner could be dragged to a projection window to connect them.

Below, one finds the inputs for the space group and the lattice constants. The space group determines the free lattice parameters (e.g. all angle parameters are fixed to 90° for the orthorhombic space groups) and which equivalent solutions in the indexing procedure should be suppressed. The program uses a very relaxed checking of the space group . Everything which is formally correct should be accepted, e.g. F1, F4/mmm, Pbaa but not Paaa. At the moment systematic extinctions, even for the lattice centering, are not handled, therefore e.g. Pmmm, Pnma and Fmmm would give exactly the same result. not handled by the program A special behavior is implemented for the trigonal space groups. If entered as e.g. R3m, the program uses the rhombohedral primitive setting a=b=c, $\alpha = \beta = \gamma$, when entered using the non-standard centering symbol “H” (H3m) the hexagonal setting a=b≠c, $\alpha = \beta = 90°, \gamma = 120°$ is used. When changing directly between these space-group symbols, the lattice constants are converted accordingly and a rotation is employed in order to get an equivalent projection. This rotation ensures, that the $[1\,1\,1]_{rhom}$ and $[1\,\bar{1}\,0]_{rhom}$ coincides with the $[0\,0\,1]_{hex}$ and $[1\,0\,0]_{hex}$ respectively after the transformation.

Beneath the space group input, the inputs for the lattice constants are located. Depending on the space group, some of the fields may be disabled, for a cubic space group, only the a lattice constant is available.

Underneath the lattice constants the orientation matrix and three Eulerian angles follow. The orientation matrix displays the reciprocal lattice vectors in a Cartesian laboratory system. The system employed in the program maps the x-, y- and z-axis to the primary beam axis, the horizontal and the vertical axis respectively. The Euler angles Ω, χ and Φ qualify the rotations around the z-, x- and y-axis respectively. The initial orientations of the crystal, e.g. with all Euler angles set to zero, places the crystallographic **a**-axis on the x-axis, the **b**-axis in the xy-plane and the **c**-axis completes the right handed system.

1.3 Laue Plane

The Laue plane window represents the film/image plate, where the Laue pattern will be recorded on. It displays the recorded image, the primary beam marker, marked reflexes and the calculated spots and allows to zoom in to the image and mark reflexes.

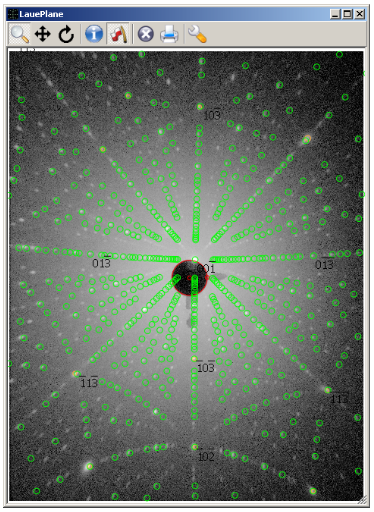
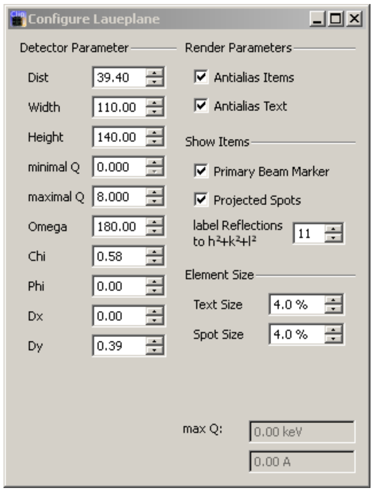


Figure 3: Laue plane window with a back reflection Laue image of $SmTiO_3$ from [6]


Configure Laueplane

Detector Parameter

Dist	39.40
Width	110.00
Height	140.00
minimal Q	0.000
maximal Q	8.000
Omega	180.00
Chi	0.58
Phi	0.00
Dx	0.00
Dy	0.39

Render Parameters

☑ Antialias Items

☑ Antialias Text

Show Items

☑ Primary Beam Marker

☑ Projected Spots

label Reflections to h²+k²+l² 11

Element Size

Text Size 4.0 %

Spot Size 4.0 %

max Q: 0.00 keV

0.00 A

Figure 4: Configuration panel for a Laue plane window.

Two concentric red circles, the outer one with a handle, are the primary beam marker. With the help of the handle, the outer circle could be resized to increase the accuracy, if the Laue image provides an appropriate feature, like the image plate system at does.

The mouse action depends on the selected button in the upper toolbar.

A left mouse button click and subsequent drag may either zoom into the image or may rotate the crystal. Two rotation modes are available, the first uses rotations around arbitrary axes in order to allow to freely drag reflexes on the plane while the second rotates the crystal around a fixed axis. This could be specified by the context menu or the rotate window 1.6

A simple left mouse click either creates a info panel for the nearest reflex or adds a marker for indexing and fitting purposes. Info panels and markers could be dragged around with the mouse. The info panels will also be displayed on a print out and will disappear when the crystal orientation is changed.

A short left click restores the previous zoom level and a longer right mouse click opens a context menu. Here the rotation axis could be set to reciprocal lattice vector of the nearest[1] (calculated) reflex, all reflex informations could be cleared and the nearest or all marked reflexes could be cleared.

The three rightmost buttons on the toolbar allow to open or close a Laue image, to print the Laue plane including all decorations and to open the configuration dialog for this Laue plane. The image file may be any image format known to the Python Image Library. Additionally the program is able to read the format which is returned by the Fuji BAS image plate reader. This format has a higher dynamic range of 12 bits, compared to 8 bits of conventional gray scale image formats. In order to take advantage of this improved range, the image is internally represented with the full information and mapped to screen colors by a user modifiable transfer curve, see 1.9 If an image is loaded, four more buttons for image rotation and flipping appear.

Figure 1.3 shows the configuration dialog. On the left are the parameters of the projection. From top to the bottom they are:

Dist Distance from the crystal to the projection plane.

Width Width of the projection plane. If a BAS image is read, this parameter is taken from the image.

Height Height of the projection plane. If a BAS image is read, this parameter is taken from the image.

minimal Q Lower bound of the wave vector distribution in Å^{-1}. Could be normally set to 0.

maximal Q Upper bound of the wave vector distribution in Å^{-1}. A higher value produces more spots and may slow down the program. The upper bound is also shown in the lower right corner as xray tube voltage[2] and x-ray wavelength.

Omega, Chi spherical coordinates of the plane normal. Ω is the rotation around the z-axis, therefore ω=0° corresponds to a transmission Laue configuration, while ω=180° corresponds to a back reflection Laue configuration. χ is th polar angle.

[1] to the actual mouse position

[2] also corresponds to the electron energy

Phi Rotation of the plane around the plane normal.

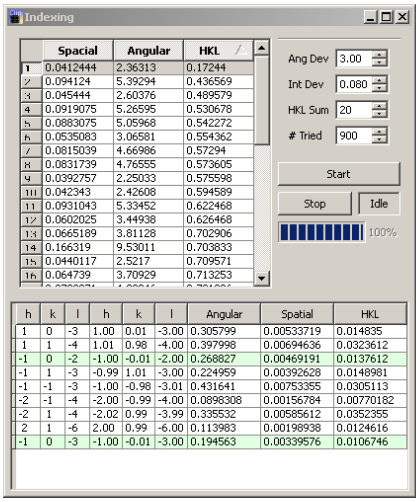
Dx and Dy Position of the piercing point of the plane and the plane normal, connecting the plane with the crystal.

On the right side, on finds parameters which alter the appearance of the display. The render parameters allow to switch on or off the anti-aliasing for the text and the items like projected reflex markers. Anti-aliasing provides better quality but may be slower.

Next category allows to show or hide the primary beam marker and the projected spots. The latter is very useful when one marks spots for indexing. Additionally, one could specify, up to which value of $h^2+k^2+l^2$ the hkl should be displayed beside the reflexes. Contrary to the reflex info panels, these indicators move along with the reflexes and allow a quick navigation in the reciprocal space.

Finally the Element Size category allows th specify the size for the calculated spots and all text at the plane.

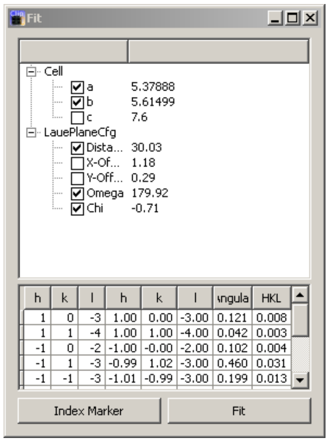
1.4 Index Search



Ang Dev 3.00
Int Dev 0.080
HKL Sum 20
Tried 900

	Spacial	Angular	HKL
1	0.0412444	2.36313	0.17244
2	0.094124	5.39294	0.436569
3	0.045444	2.60376	0.489579
4	0.0919075	5.26595	0.530678
5	0.0883075	5.05968	0.542272
6	0.0535083	3.06581	0.554362
7	0.0815039	4.66986	0.57294
8	0.0831739	4.76555	0.573605
9	0.0392757	2.25033	0.575598
10	0.042343	2.42608	0.594589
11	0.0931043	5.33452	0.622468
12	0.0602025	3.44938	0.626468
13	0.0665189	3.81128	0.702906
14	0.166319	9.53011	0.703833
15	0.0440117	2.5217	0.709571
16	0.064739	3.70929	0.713253

h	k	l	h	k	l	Angular	Spatial	HKL
1	0	-3	1.00	0.01	-3.00	0.305799	0.00533719	0.014835
1	1	-4	1.01	0.98	-4.00	0.397998	0.00694636	0.0323612
-1	0	-2	-1.00	-0.01	-2.00	0.268827	0.00469191	0.0137612
-1	1	-3	-0.99	1.01	-3.00	0.224959	0.00392628	0.0148981
-1	-1	-3	-1.00	-0.98	-3.01	0.431641	0.00753355	0.0305113
-2	-1	-4	-2.00	-0.99	-4.00	0.0898308	0.00156784	0.00770182
-2	1	-4	-2.02	0.99	-3.99	0.335532	0.00585612	0.0352355
2	1	-6	2.00	0.99	-6.00	0.113983	0.00198938	0.0124616
-1	0	-3	-1.00	-0.01	-3.00	0.194563	0.00339576	0.0106746



⊟ Cell
☑ a 5.37888
☑ b 5.61499
□ c 7.6
⊟ LauePlaneCfg
☑ Dista... 30.03
□ X-Of... 1.18
□ Y-Off... 0.29
☑ Omega 179.92
☑ Chi -0.71

h	k	l	h	k	l	ıngula	HKL
1	0	-3	1.00	0.00	-3.00	0.121	0.008
1	1	-4	1.00	1.00	-4.00	0.042	0.003
-1	0	-2	-1.00	-0.00	-2.00	0.102	0.004
-1	1	-3	-0.99	1.02	-3.00	0.460	0.031
-1	-1	-3	-1.01	-0.99	-3.00	0.199	0.013

Figure 5: Indexing tool window.

Figure 6: Fit window

Image 1.4 shows the indexing tool. In order to start the indexing procedure, the correct space group and lattice constants should be specified, the detector-sample distance should be correctly entered in the configuration dialog[3], the primary beam marker should be moved to the appropriate position and some likely low indexed spots should have been marked[4]. When the setup is complete,

[3] a distance of 30 mm is default

[4] Choosing the suitable one depends very much on the experience of the experimenter

one could start the indexing and hope, the correct solution is among the calculated ones. Otherwise one should mark different spots and/or try to modify the lattice constants and detector distance. If this does not lead to the proposed result, ask an expert.

The algorithm used in the indexing procedure works as follows. For each marked spot, the direction $\mathbf{n}_i$ of the scattering vector is calculated. The magnitude is not accessible because of the Laue geometry, so the n_i are unit vectors. If k_i denotes the length of the scattering vector, then indexing the pattern stood for determine a rotation matrix R and a set of integer vectors $\mathbf{h}_i$ so that

$$\mathrm{k}_i \cdot \mathbf{n}_i = \mathrm{R} \cdot \sum_j \mathbf{a}_j^* \cdot \mathrm{h}_{i,j} = \mathrm{R} \cdot \Re \cdot \mathbf{h}_i$$

is fulfilled with the reciprocal lattice vectors $\mathbf{a}_i^*$ and the matrix of reciprocal basis vectors $\Re$. This could be transformed to

$$s_i \cdot \Re^{-1} \cdot \mathrm{R}^{-1} \cdot \mathbf{n}_i = (\mathrm{h\ k\ l}) \tag{1}$$

with appropriate scale factors s_i. In the actual procedure, the hkl will only be almost integer. To determine the rotation matrix R, a set of reciprocal lattice vectors $\mathbf{t}_i$ is created, sorted according to h^2+k^2+l^2 and normalized. The first N vectors are used as test vectors, the number is adjustable with the parameter **# Tried**. Then for each pair of test vectors ($\mathbf{t}_i$, $\mathbf{t}_j$) the angle between them is calculated and compared to the angles between scattering vectors ($\mathbf{n}_i$, $\mathbf{n}_j$). If pairs are found, whose angles differ by no more than the parameter **Ang Dev**, an optimal rotation matrix R_t is calculated, so that $\mathbf{t}_i = \mathrm{R}_t^{-1} \cdot \mathbf{n}_i$ and $\mathbf{t}_j = \mathrm{R}_t^{-1} \cdot \mathbf{n}_j$.

Know for the rest of the scattering vectors $\mathbf{n}_i$, the vector $\Re^{-1} \cdot \mathrm{R}^{-1} \cdot \mathbf{n}_i$ is calculated. If the rotation matrix R_t represents a feasible indexing, this vector could be scaled to the (almost) integer indices $\mathbf{h}_i$ of the reflex. Therefore the vector is divided by it's largest component[5] and successively multiplied by integer values ranging from 1 to **max HKL**. The difference of the remaining two components from integer has to be less than **Int Dev** for a match. If all scattering vectors could be indexed, the rotation matrix R_t is optimized and stored together with the scaling factors and indices as a possible solution. For each solution, three score factors are calculated, which are list in the table as Angular, Spacial and HKL. HKL is the summed difference from integer of hkl in equation 1, Spacial and Angular are defined as follows:

$$\mathrm{Spacial} = \sum_i \mathbf{n}_i - \mathbf{h}_i \qquad \mathrm{Angular} = \sum_i \mathrm{acos}(\mathbf{n}_i \cdot \mathbf{h}_i)$$

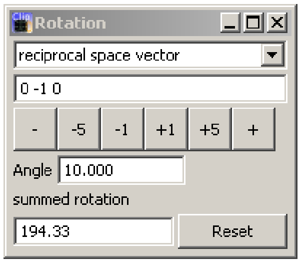
with $\mathbf{h}_i = \frac{\mathrm{R} \cdot \Re \cdot \mathbf{h}_i}{\|\mathrm{R} \cdot \Re \cdot \mathbf{h}_i\|}$

[5] which then becomes 1

1.5 Parameter Refinement

Image 1.4 shows the fit tool window. At the top, the parameters which might be varied along with their values are shown. For the cell constants, it is ensured, that at least one of a, b, and c is kept constant, as it is not possible to determine all three with the Laue technique. With the bottom-left button, all marked spots are indexed analog to the procedure described in subsection 1.4. With the right button, the orientation and all selected parameters will be varied in order to minimize the HKL score factor. For a good fit, some spots at the border of the Laue image should be marked. In the first step, only the orientation should be refined, subsequently one could refine the distance, Omega and Chi and then the lattice constants. If Omega and Chi are refined, the Dx and Dy parameters are adjusted in order to keep the primary beam marker at it's position.

1.6 Rotation


Rotation

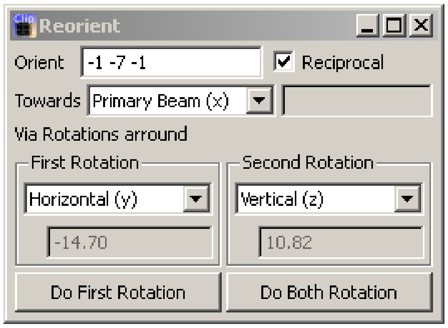
reciprocal space vector

0 -1 0

Angle 10.000

summed rotation 194.33

Figure 7: Rotation tool window


Reorient

Orient -1 -7 -1 ☑ Reciprocal

Towards Primary Beam (x)

Via Rotations arround

First Rotation: Horizontal (y) -14.70

Second Rotation: Vertical (z) 10.82

Figure 8: Reorientation tool window

Figure 1.6 shows the rotation tool. It allows to select one of the principal laboratory frame axes (primary beam, horizontal or vertical) or to specify an arbitrary axis in the laboratory frame, the reciprocal space or the direct space. As long as the rotation axis is not changed, the rotations done around it are summed up. Additionally provides separate buttons to quickly rotate through ±1°, ±5° or ± a user defined value.

1.7 Reorientation

The reorientation tool 1.6 allows to calculate the angles, which are necessary to transfer the actual orientation to the desired on and also to apply the rotations. In the top input box, the real or reciprocal lattice vector is defined, which should be rotated towards the direction specified in the second line. Here, the three principal laboratory axis and arbitrary real and reciprocal lattice vectors are possible. The lower part of the window allows to define the goniometer axis, around which the rotations will be done.

1.7.1 Algorithm

To reorient two unit vectors, say $\mathbf{v}_1$ to $\mathbf{v}_2$ with two rotations around the ϕ and χ unit vectors, **Clip** uses the following approach.

We we need to determine two rotation matrices R_ϕ and R_χ, so that $v_2 = R_\phi \cdot R_\chi \cdot v_1$, or equivalently $R_\phi^{-1} \cdot v_2 = R_\chi \cdot v_1$. $R_\phi^{-1} \cdot v_2$ and $R_\chi \cdot v_1$ lie on a circle on the unit sphere. These circles are the intersections of the unit sphere with planes having ϕ and χ as normals and distances from the origin of $v_2 \cdot \phi$ and $v_1 \cdot \chi$ respectively. These equations of these planes are: $\mathbf{x} \cdot \phi = v_2 \cdot \phi$ and $\mathbf{x} \cdot \chi = v_1 \cdot \chi$. The intersection of these two planes is a line and the vectors on this line with a length of 1 represent the possible intermediate vectors for the rotation: $v_i = R_\phi^{-1} \cdot \mathbf{v_2}$.

The equation of the line is

$$u_1 + \varepsilon \cdot u_2 \tag{2}$$

where u_2 could be chosen to be perpendicular to ϕ and χ and therefore to be simply $u_2 = \phi \times \chi$ and u_1 could be chosen to lie in the $\phi\,\chi$-plane. This gives $u_1 = \lambda \cdot \phi + \mu \cdot \chi$. The length of u_1 is the minimal distance of the line from the origin and therefore determines the number of solutions. If it is larger than one, the intersection of the two planes lie entirely outside of the unit sphere and no rotation around ϕ and χ from v_1 to v_2 is possible. If the length is exactly one, then there is one solution and corresponding one possible rotation. If the length is smaller than one, are two possible solutions.

To determine the coefficients in the representation of u_1 we consider the equations for the two planes (remember that u_2 is perpendicular to both χ and ϕ and therefore the product vanishes). Here we get the equations $(\lambda{\cdot}\phi{+}\mu{\cdot}\chi){\cdot}\phi = v_2{\cdot}\phi$ and $(\lambda{\cdot}\phi{+}\mu{\cdot}\chi){\cdot}\chi = v_1{\cdot}\chi$. The first equation simplifies to $\lambda = (v_2{-}\mu{\cdot}\chi){\cdot}\phi$ and inserted in the second gives the solutions

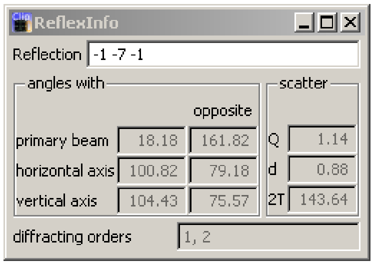
$$\lambda = \frac{v_2 \cdot \phi - (\phi \cdot \chi)(v_1 \cdot \chi)}{1 - (\phi \cdot \chi)^2} \tag{3}$$

$$\mu = \frac{v_1 \cdot \chi - (\phi \cdot \chi)(v_2 \cdot \phi)}{1 - (\phi \cdot \chi)^2} \tag{4}$$

If we now normalize u_2, we could rewrite equation (2) as $\varepsilon = \pm\sqrt{1 - |u_1|^2}$

1.8 Reflex Information

The reflex info tool 1.8 shows the angles between a reflex and the x-, y- and z-axis. If is in scattering position, that is the angle to the primary beam is less than 90 °, it shows Q in Å^{-1}, the interplane distance d in Å and the scattering angle in °. The reflex could be either entered manually or clicked with an activated info mode in the Laue Plane.

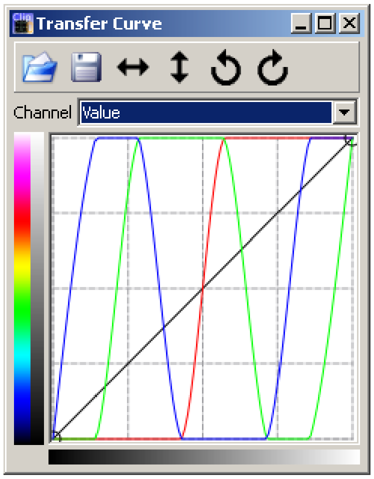


Reflection -1 -7 -1

angles with		opposite	scatter	
primary beam	18.18	161.82	Q	1.14
horizontal axis	100.82	79.18	d	0.88
vertical axis	104.43	75.57	2T	143.64

diffracting orders 1, 2

Figure 9: Reflex information window



Channel Value

Figure 10: Transfer curve tool window

1.9 Transfer Curve

The image transfer curve 1.8 allows to adjust the contrast of an image. It works like the corresponding tools in various graphics softwares, like The Gimp or Adobe Photoshop. Curves could be edited, and saved to or loaded from disk. Additionally the image might be rotated in 90 ° steps or flipped horizontally or vertically. In contrast to the similar function of the projector window, only the image itself is affected, neither the aspect ratio not the marked spots are updated. This serves for the case, where the image orientation is changed and afterwards spots are marked and indexed. If know the workspace is saved and later loaded, the spot positions does not match these of the image. Here, only (!) the image has to be transformed. As this behavior is somewhat special, it will be reimplemented in future versions.

2 Future developments

All required features for a successful treatment of Laue exposures are basically available in Clip and the user interface is quite self consistent. Therefore only a few features are scheduled in the short term. These include the handling of systematic extinctions from the space-group, the possibility to display some crystal informations on the projection plane, an improved marker handling and performance improvements.

Once a marker is placed on the projector plane, the user has no chance to tell, which marker belongs to which score, e.g. in the indexing tool. Thus, it would be very helpful to select in the indexing window a marker line with a bad score and get a get the corresponding marker highlighted.

The crystal informations would include lattice constants, orientation and some user defined text such as sample description/identification and are most useful on printing as a documentation. Up to now, users manually write these data on the hardcopy.

Long term developments depend strongly on required features. Some possible enhancements are outlined below.

As multicore processors are more and more common, certainly more parts of the code will ported to multithreading algorithms which in turn would greatly improve performance. More image processing features could implemented, like cropping the image, background reduction or new image formats. Up to now, no handling of twinned crystals is possible. It would be nice, if the twinning law could be entered to clip and spots for different twins are marked e.g. by different colors.

The indexing routine might be updated. Indexing relies on marking at least two low indexed reflexes. The algorithm itself allows to use direct space vectors instead of reciprocal space ones as input. This would result in marking direct space vectors, or equivalently zones in the reciprocal space. As the low index zones are clearly visible as curved lines of bright spots, normally with clear envelope, this might give better results in pathological cases. In fact the second edition of clip uses this approach but it was abandoned as selecting spots separately for indexing and refining decelerates the workflow.

Another improvement in indexing would be a fully automatic algorithm, at least for simple problems. Wenk *et al.* [7] propose an algorithm where the Laue image is transformed by a gnomonic projection. This projection transfers great circles viz. zones to straight lines. Low indexed zones with lots of bright spots appear as pronounced lines, which in turn are detected by a subsequent Hough transform and a peak search algorithm. First test showed indeed human visible straight lines, but the background of the image plate images prevents a reliable recognition by the computer.

A model of the crystal structure could be used to calculate the spot intensities. This would allow to handle weak reflexes which might not be visible on the exposure. They might just be masked if below a certain threshold or the size of the calculated spots could scale with calculated intensity.

At the very end, clip might support the integration of spots, scaling them according to the wavelength distribution etc. and output in a form suitable for single crystal refinement.

3 Summary

Clip has proven to be a very useful and hopefully userfriendly program. The development basically followed the suggestions and needs of the users [6], therefore some simple features like systematic extinctions are not yet available, while others like printing are.

[6] and the authors interests

Clip has been extensively used in the institute of physics II for orientation and checking of a whole bunch of crystals. It's algorithm showed good performance in automatically determine the orientation for most of the samples.

The fit procedure allows to distinguish between critical cases, if one adopts some common crystallographic sense. The pseudo-cubic perovskites may here serve as an example. They often crystallizes in the orthorhombic space-group Pbnm, which emerges from the cubic base structure by a rotation of 45 ° around the c-axis and doubling of the unit cell in c-direction. Thus, the lattice constants in terms of the cubic ones a are roughly $\sqrt{2}$a, $\sqrt{2}$a, and 2a, especially with a small difference between the a and b lattice constants. If one records a Laue pattern with the $(0\,0\,1)$ in direction of the primary beam, marks some reflexes and let the a and b constants run during the refinement, one of them gets larger. This allows to distinguish between the short and long in-plane axis and works surprisingly well, even for small orthorhombic splittings. There is one catch one should be aware of. The long and short axis might be reversed compared to the literature Pbnm data which corresponds to a description in the Pnam space-group or equivalently a renaming of a to b and vice versa. Then, if the $(0\,0\,1)$ is located near the center, the (literature) a axis lies in the direction of all $(0\,k\,l)$ reflexes, not $(h\,0\,l)$ as in the case where long and short axis match between clip and the literature.

Clip might not be finished yet, but most features for normal work use are present.

References

[1] G. van Rossum, Python reference manual, Technical report, 1995. 1

[2] Riverbank Computing Limited, *SIP - A Tool for Generating Python Bindings for C and C++ Libraries*, 2008, Version: 4.7.9. 1

[3] Nokia Corporation, *Qt Reference Documentation*, 2008, Version: 4.4. 1

[4] Riverbank Computing Limited, *PyQt v4 - Python Bindings for Qt v4*, 2008, Version: 4.4.4. 1

[5] Gnu general public license. 2

[6] H. Roth, PhD thesis, Universität zu Köln, in Vorbereitung. 4

[7] H. R. Wenk, F. Heidelbach, D. Chateigner, and F. Zontone, *Laue Orientation Imaging*, Journal of Synchrotron Radiation **4**, 95 (1997). 11