

Upgraded *LauePt4* for rapid recognition and fitting of Laue patterns from crystals with unknown orientations

Vincent W. Huang, Yafei Liu, Balaji Raghothamachar* and Michael Dudley

Department of Materials Science and Chemical Engineering, Stony Brook University, Stony Brook, New York 11790, USA. *Correspondence e-mail: balaji.raghothamachar@stonybrook.edu

Received 2 May 2023

Accepted 11 September 2023

Edited by A. H. Liu, HPSTAR and Harbin Institute of Technology, People's Republic of China

Keywords: Laue diffraction; *LauePt4*; X-ray crystallography; indexing.

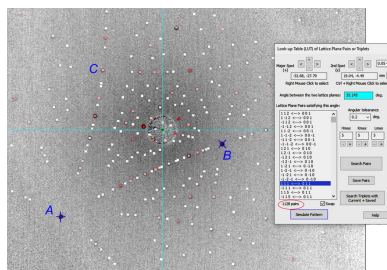
Supporting information: this article has supporting information at journals.iucr.org/j

The *LauePt* program is a popular and easy-to-use crystallography tool for indexing and simulating X-ray Laue patterns, but its previous versions lack search functions for recognizing Laue patterns taken from crystals with unknown orientations. To overcome this obstacle, a major upgrade of the program, called *LauePt4*, is presented with three robust search schemes implemented: (i) crystal rotation along a single diffraction vector, (ii) a look-up method to search for reflection pairs matching the interplanar angle of two selected diffraction spots, and (iii) a more efficient look-up scheme to search for reflection triplets matching three interplanar angles. Extensive tests show that all these schemes, together with the convenient graphical user interfaces and highly optimized computing algorithms, are reliable and powerful for recognizing and fitting Laue patterns of any crystal taken under any diffraction geometry.

1. Introduction

The *LauePt* computer program is an interactive Windows application for simulating white-beam X-ray Laue patterns of general crystals. Since publication of its Version 3 (Huang, 2010), *LauePt* has become a very popular crystallography tool used by many researchers worldwide, particularly for synchrotron white-beam topography or diffraction, determination of crystal orientations by the Laue method, and crystallography education (Oriwol *et al.*, 2013; Woodworth *et al.*, 2014; Asadchikov *et al.*, 2018; Danilewsky, 2020; Cui *et al.*, 2020; Zolotov *et al.*, 2020). As a small program with all the functions integrated into a single executable file, *LauePt* has a complete set of convenient and straightforward graphical user interfaces (GUIs) with which one can quickly simulate a complicated Laue pattern with just a few mouse clicks. Therefore, it can be used even by non-specialists with little crystallography knowledge or training.

Laue patterns generated by *LauePt* can be directly compared with real patterns (Laue images) recorded by X-ray imaging detectors or X-ray films. *LauePt* does not have a procedure to process raw Laue images into maps of diffraction spot positions. Instead, it allows the user simply to import a recorded Laue image as the background, and the simulated pattern is plotted directly on top of it for interactive visual comparison. The program can handle any ordinary images in TIFF (8-bit), JPEG and BITMAP formats. This feature dramatically reduces the complexity of Laue pattern simulation and fitting compared with the large-scale synchrotron Laue microdiffraction programs (Barabash *et al.*, 2001; Liu *et al.*, 2004; Gupta & Agnew, 2009; Ulrich *et al.*, 2011; Robach &



Micha, 2015; Zhang *et al.*, 2017; Örs *et al.*, 2018; Morawiec, 2020; Purushottam Raj Purohit *et al.*, 2022; Gürsoy *et al.*, 2022; Seret *et al.*, 2022;). *LauePt* has another unique advantage, namely accurate intensity calculations (based on the kinematic diffraction model). Consequently, the visual comparison of both diffraction spot positions and intensities is clear and vivid, and the pattern fitting, once achieved, has no ambiguity even for poor-quality recorded Laue images. In addition to the *hkl* Miller indices, each spot of the simulated pattern also gives information about the wavelengths, structure factors and intensities of all the harmonics based on the incident spectrum. The fast and accurate intensity calculations of *LauePt* may potentially enable simple and rapid white-beam diffraction techniques to have new capabilities for verifying or refining crystal structures, or even solving crystal structures.

Unfortunately, however, the previous versions of *LauePt* have a severe shortcoming in a lack of searching/fitting procedures. To fit a recorded image, the user must make initial guesses of two perpendicular directions of the crystal in laboratory space. Starting from these directions, *LauePt* can freely rotate the crystal around three orthogonal axes and the simulated pattern changes instantly with rotation. For crystals with a roughly known orientation, one can quickly find a simulation matching the real image by these rotation operations, from which the exact 3D crystal orientations and the surface miscut can be derived (Huang, 2010). Unfortunately, for crystals with unknown orientations, it becomes extremely difficult, if not impossible, for *LauePt* to find from millions of possible patterns a correct one that matches the recorded Laue image. This has been a major obstacle to its many applications.

In this paper, we introduce a major upgrade of *LauePt* to Version 4 by implementing three fast and reliable search schemes to overcome this obstacle. These improvements make *LauePt4* a full-function program without affecting its compactness and ease of use.

The Laue images in this paper were recorded with a laboratory Laue camera in the back-reflection geometry [$\Phi = 0$ in Fig. 1(a)]. The source is an AEG Cu X-ray tube (model FK 60-10) operated at 30 kV and 30 mA, and the incident beam is limited by a 1 mm diameter pinhole. The field of view of the imaging detector (a 2009 Photonic Science X-ray backscattering dual CCD detector) is 145 mm by 96 mm and the pixel size is $\sim 70 \mu\text{m}$. The Laue images presented here are 8-bit grayscale TIFF images (with 256 intensity levels) recorded with a typical exposure time of around 3 min. State-of-the-art Laue cameras may have much shorter exposure times and better image contrast. Currently *LauePt4* only supports 8-bit color depth, which is adequate for computer screen display and visual comparison.

2. Crystal rotation along a specific diffraction vector

As shown in Fig. 1(a), *LauePt* sets up a fixed $x'y'z'$ laboratory coordinate system (CS), where the z' axis is opposite to the incident direction and the x' axis is along the horizontal direction. Crystal rotation is performed around any of the three axes X , Y and Z that initially coincide with x' , y' , z' , respectively. Here a brute-force algorithm for searching a Laue pattern is a triple-nested loop of rotation around X , Y , Z with a step size of 2° , but that would require $(180 \times 180 \times 180) \simeq 6\,000\,000$ simulations and visual comparisons, which is impractical.

A common feature of Laue diffraction is that a low-index reflection (such as 100, 110, 111 *etc.*) usually appears as an isolated diffraction spot, which is also the intersection of different hyperbolas or ellipses formed by diffraction spots of different zones. For example, Fig. 2(a) is a back-reflection Laue image of an Si wafer with an initially unknown orientation, where A , B and C are such isolated spots. On the basis of this phenomenon, our first strategy is that, if a Laue image contains a well isolated spot, we assign it a low index as a trial. (If there are no obvious isolated spots in the image, one can easily rotate the crystal during the experiment to bring one or more isolated spots onto the imaging detector.) Next, we design a scheme, called the G-rotation method, to rotate the crystal around the diffraction vector \mathbf{G} of this spot. As indicated in Fig. 1(b), crystal rotation around \mathbf{G} does not change the specific diffracted wavevector $\mathbf{k}_g (= \mathbf{k}_0 + \mathbf{G})$ since both \mathbf{G} and the incident wavevector \mathbf{k}_0 remain unchanged during rotation. Consequently, the position of the selected spot P also remains fixed on the detector plane in Fig. 1(a) during rotation. However, other spots will (dramatically) change positions during rotation.

LauePt4 allows the user to right click on a diffraction spot P to select it. On the basis of the mouse click position on the screen, the program calculates the real position (x_p, y_p) of P in the Cartesian CS $X_f Y_f$ of the detector plane. From the

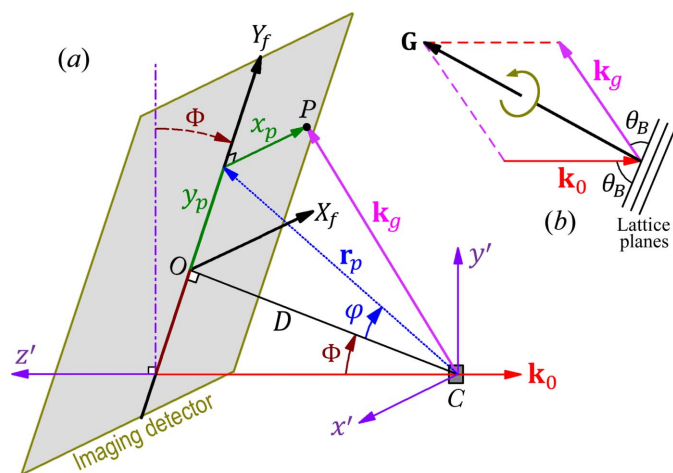


Figure 1
(a) A schematic diagram of Laue diffraction geometry. C is the crystal. O is the perpendicular foot of C on the imaging detector plane. D is the imaging distance from C to O . The X_f axis is along the opposite direction to x' . *LauePt* allows the detector to be rotated around x' by an angle Φ ($-180^\circ \ll \Phi \ll 180^\circ$). [$\Phi = 90^\circ$ corresponds to the reflection geometry widely used in Laue microdiffraction where the X-ray imaging detector is placed above the crystal (Gürsoy *et al.*, 2022; Ulrich *et al.*, 2011), while $\Phi = 180^\circ$ corresponds to the forward transmission geometry.] (b) Crystal rotation along the diffraction vector \mathbf{G} without affecting the diffracted wavevector \mathbf{k}_g .

geometry in Fig. 1(a) the diffracted wavevector is $\mathbf{k}_g = -x_p \hat{\mathbf{x}}' + r_p \sin(\Phi + \varphi) \hat{\mathbf{y}}' + r_p \cos(\Phi + \varphi) \hat{\mathbf{z}}'$ in the $x'y'z'$ laboratory CS, where $r_p = (D^2 + y_p^2)^{1/2}$, $\varphi = \tan^{-1}(y_p/D)$ and $\hat{\mathbf{x}}', \hat{\mathbf{y}}', \hat{\mathbf{z}}'$ are unit vectors along the x', y', z' axes, respectively. Here only the direction of \mathbf{k}_g is valid, represented by the unit vector $\hat{\mathbf{k}}_g = \mathbf{k}_g/|\mathbf{k}_g|$. The unit vector along the incident direction is $\hat{\mathbf{k}}_0 = -\hat{\mathbf{z}}'$. The \mathbf{G} vector in Fig. 1(b) is parallel to the vector $\mathbf{k}_g - \hat{\mathbf{k}}_0$. *LauePt4* then obtains a unit vector $\hat{\mathbf{G}}' = (\hat{\mathbf{k}}_g - \hat{\mathbf{k}}_0)/|\hat{\mathbf{k}}_g - \hat{\mathbf{k}}_0|$ along \mathbf{G} in the $x'y'z'$ laboratory CS. The program also defines another unit vector $\hat{\mathbf{U}}' = \hat{\mathbf{z}}' \times \hat{\mathbf{G}}'$ perpendicular to $\hat{\mathbf{G}}'$.

After the selection, the user can assign a Miller index hkl to P , i.e. the diffraction vector of P is $\mathbf{G} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$ in the crystal CS, where $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$ are the three primitive reciprocal-lattice vectors in the crystal CS. The program also automatically selects another diffraction vector $\mathbf{G}_u = h_u\mathbf{a}^* + k_u\mathbf{b}^* + l_u\mathbf{c}^*$ parallel to $\hat{\mathbf{U}}'$. Afterwards, the two directions \mathbf{G} and \mathbf{G}_u completely determine the crystal orientation in the laboratory CS (see the supporting information for the transformation between the crystal and laboratory CSs). *LauePt4* simulates the pattern based on (hkl) and $(h_u k_u l_u)$ in the laboratory CS. Then \mathbf{G}_u can be rotated around \mathbf{G} by 360° , which corresponds

to a crystal rotation around \mathbf{G} . Therefore, if the guessed index hkl is correct, one can find the matching simulation within the 360° rotation. If the search fails, one can try another low index to repeat this process.

For example, the simulation of Fig. 2(b) is generated by right clicking spot *A* in Fig. 2(a) and assigning it the Miller index 111. Here the simulation does not match the recorded image, but the calculated 111 reflection position marked by the '+' symbol indeed coincides with spot *A*. When the crystal is rotated along $\mathbf{G} = 111$, the simulated pattern changes, but the 111 reflection spot '+' is fixed at spot *A*, as expected. After rotation of 20° , the simulated pattern successfully matches the recorded pattern in Fig. 2(c). Under this condition, *LauePt4* gives the 'absolute orientation' of the crystal plane perpendicular to z' as $(1, 0.739, 0.254)$. In other words, the z' direction is along vector $\mathbf{a}^* + 0.739\mathbf{b}^* + 0.254\mathbf{c}^*$ in reciprocal space. (We allow the indices to be fractions in order to express arbitrary orientations that are not necessarily along discrete reciprocal-lattice vectors.) The plane $(1, 0.739, 0.254)$ is 0.46° away from the true lattice plane (431) . Since the Si wafer was perpendicular to z' in the experiment, the wafer surface is determined to be very close to (431) only with a small miscut angle of 0.46° . *LauePt4* also gives the orientations of the crystal along the x' and y' directions. Here spot *B* is determined to be reflection 011 (actually its harmonics) in Fig. 2(c). One may alternatively select this spot and assign 011 to it. Then rotating the crystal around the corresponding vector $\mathbf{G} = 011$ can also easily reach the same matching pattern.

Our extensive tests show that the simple G-rotation method is surprisingly successful even for complicated patterns with high-density spots, although it may take a number of attempts to try the index of the selected spot before the fitting is achieved. It works particularly well for separating different sets of Laue patterns (diffracted from crystal twinning, domains or grains) superimposed on a single Laue image, since this method only needs a single spot to simulate an entire Laue pattern. In contrast, in the following methods one can possibly select two or three spots belonging to different Laue patterns in the superimposed image.

3. Looking up lattice plane pairs with a selected interplanar angle

The second fitting procedure added to *LauePt4* is the classical method of using a look-up table (LUT) method to search for pairs of lattice planes (reflections) satisfying the interplanar angle of two selected diffraction spots (Heizmann & Baro, 1967; Riquet & Bonnet, 1979). Here we call it the LUT-P method. Implementation of LUT-P in *LauePt4* is very simple. First, the user selects two spots (low-index isolated spots are always preferable, if possible) in the imported Laue image. *LauePt4* immediately calculates the two diffraction vectors \mathbf{G}_1 and \mathbf{G}_2 in the laboratory CS using the method in Section 2, from which the interplanar angle $\theta_{12} = \cos^{-1}[(\mathbf{G}_1 \cdot \mathbf{G}_2)/(|\mathbf{G}_1| |\mathbf{G}_2|)]$ is obtained. *LauePt4* actually does not have an LUT calculated in advance. Instead, it instantly calculates the angle θ between each possible reflection pair

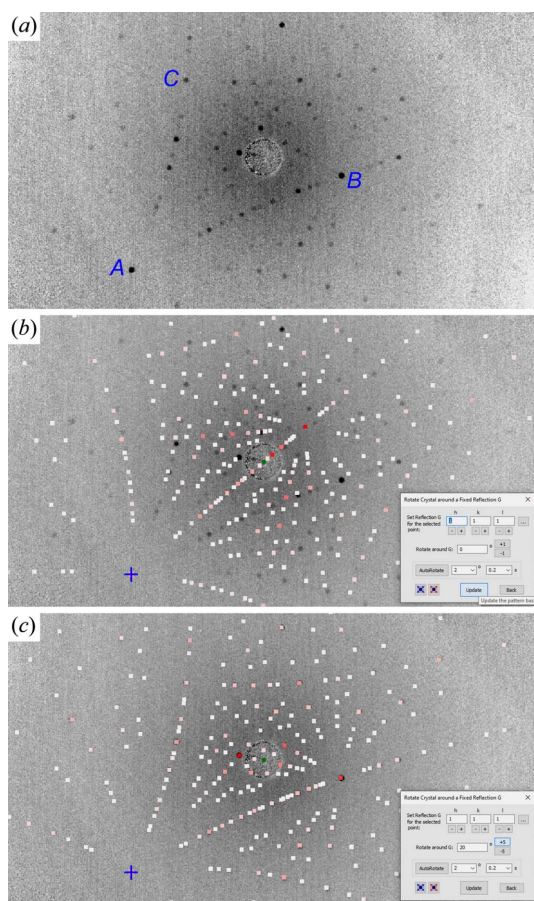


Figure 2 (a) Back-reflection Laue image taken from an Si wafer. (b) Laue pattern simulated by assigning the $\mathbf{G} = 111$ reflection to spot *A*. (c) A matching pattern simulated with 20° crystal rotation around \mathbf{G} from panel (b), which determines the wafer surface to be (431) . Image width 141 mm and imaging distance $D = 36$ mm. The white to red spot colors indicate weak to strong diffraction intensities.

$\mathbf{H}_1 = h_1\mathbf{a}^* + k_1\mathbf{b}^* + l_1\mathbf{c}^*$ and $\mathbf{H}_2 = h_2\mathbf{a}^* + k_2\mathbf{b}^* + l_2\mathbf{c}^*$ (in the crystal CS) within the index range limited by $|h_{1,2}| \leq H_{\max}$, $|k_{1,2}| \leq K_{\max}$, $|l_{1,2}| \leq L_{\max}$. If $|\theta - \theta_{12}| \leq \Delta\theta$, *LauePt4* adds the pair $(h_1k_1l_1, h_2k_2l_2)$ to the candidate list. Here $\Delta\theta$ is the angular tolerance set by the user.

After all the candidate pairs have been found, *LauePt4* allows the user to select any pair $(\mathbf{H}_1 = h_1k_1l_1, \mathbf{H}_2 = h_2k_2l_2)$ from the list to simulate the corresponding pattern. Since assigning \mathbf{H}_1 and \mathbf{H}_2 to the selected \mathbf{G}_1 and \mathbf{G}_2 vectors completely decides the crystal orientation in the laboratory CS, the simulation is unique for each pair, but it will not necessarily match the recorded image, except that the simulated \mathbf{H}_1 and \mathbf{H}_2 reflection spots will always coincide with the two selected spots. Therefore, the user must find the correct pair $(\mathbf{H}_1, \mathbf{H}_2)$ with which the entire simulation matches the recorded image. Here \mathbf{H}_1 and \mathbf{H}_2 must also be in the correct order, *i.e.* swapping the reflection indices between the two spots will give a different pattern. The simulation process is similar to that used in Section 2, where one only needs to replace \mathbf{G} and \mathbf{G}_u in Section 2 with \mathbf{H}_1 and $\mathbf{H}_1 \times \mathbf{H}_2$, respectively (see the supporting information).

Fig. 3 shows the Laue pattern simulated by selecting spots *A* and *B* in Fig. 2(a). Here the two spots are selected by simple right mouse clicks on the computer screen. *LauePt4* then calculates the angle between the two corresponding reflections as $\theta_{12} = 35.245^\circ$. Here, with $H_{\max} = K_{\max} = L_{\max} = 5$ and $\Delta\theta = 0.2^\circ$, *LauePt4* finds 1128 reflection pairs satisfying this interplanar angle, which is a large collection (reducing the tolerance to $\Delta\theta = 0.1^\circ$ can reduce the number of pairs to 768). The indices of *A* and *B* have been identified as 111 and 011, respectively, in Fig. 2, and the pair (111, 011) is indeed included in the list as a candidate. Afterwards, the user can try any pair in the list to let *LauePt4* simulate the corresponding pattern. The simulation in Fig. 3 is based on the pair (111, 011), *i.e.* *A* and *B* are assigned indices 111 and 011, respectively. The simulation almost perfectly matches the Laue image and all the results are consistent with those of Fig. 2(c).

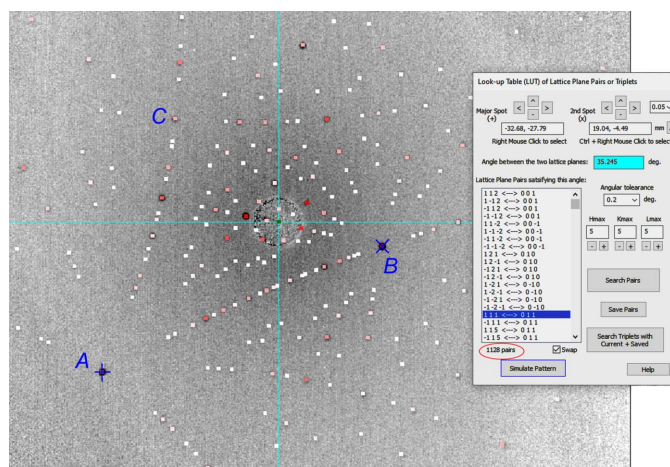


Figure 3
LUT-P search performed on the diffraction spots *A* and *B* gives the correct simulation matching the recorded image [the same as Fig. 2(a)].

Note that all the symmetry-related equivalent pairs are included in the list of Fig. 3 [*e.g.* (111, 011), (111, 110), (111, 101), ($\overline{111}$, $0\overline{11}$), ...]. If the user selects any pair other than these equivalents, the simulation will not match the Laue image. For the current LUT-P method, therefore, the user needs to test the candidates in the list one by one (including the swapped pairs). The user can use the mouse to click any pair in the list to select it (see the highlighted pair in the list of Fig. 3), and then *LauePt4* immediately simulates the Laue pattern based on the selected pair and updates the computer screen with the current simulation on top of the recorded image. In addition to selection by mouse click, *LauePt4* also allows the user to use the UP (\uparrow) and DOWN (\downarrow) arrow keys of the computer keyboard to change the selection in the list. In other words, the user can keep pressing the UP or DOWN arrow key to let the selection move up or down quickly over each item in the list. During this scrolling process, *LauePt4* instantly makes the corresponding simulation and screen update based on the transient selection. Since it computes the patterns extremely quickly due to the highly optimized algorithms and code, *LauePt4* is always capable of providing an instant screen update of the changing simulation even if the scrolling is very fast. This feature greatly reduces the time to search for the correct pair when the list is very long. Therefore, LUT-P runs fast in *LauePt4*. More importantly, this method is reliable and seldom fails.

4. Advanced look-up method for matching triplets

To improve the LUT-P efficiency further, we have developed an advanced look-up technique, called the LUT-T method, to search for *reflection triplets* satisfying three interplanar angles. The principles are again very simple. As an example, let us first select spots *A* and *C* in Fig. 2(a) to perform the LUT-P search described in Section 3. *LauePt4* gives a list of 1392 pairs (not including the swapped pairs). *LauePt4* can save this list of pairs as $\{\mathbf{H}_A^{(I)}, \mathbf{H}_C^{(I)}\}$ and the interplanar angle θ_{AC} between *A* and *C*. Next, we keep *A* selected (as the major spot) and select *B* as the second spot (with *C* deselected) to perform LUT-P again. As has been shown in Fig. 3, this step gives 1128 potential pairs, denoted $\{\mathbf{H}_A^{(II)}, \mathbf{H}_B^{(II)}\}$. The interplanar angle between *A* and *B* is θ_{AB} . *LauePt4* also calculates the interplanar angle between *B* and *C*, θ_{BC} .

In the LUT-T procedure, *LauePt4* compares each reflection pair in $\{\mathbf{H}_A^{(II)}, \mathbf{H}_B^{(II)}\}$ with all the saved pairs in $\{\mathbf{H}_A^{(I)}, \mathbf{H}_C^{(I)}\}$. During the comparison, if two pairs $(\mathbf{H}_A^{(I)}, \mathbf{H}_C^{(I)})$ and $(\mathbf{H}_A^{(II)}, \mathbf{H}_B^{(II)})$ have a common reflection, *LauePt4* calculates the angle θ_{23} between the other two reflections [for example, between $\mathbf{H}_C^{(I)}$ and $\mathbf{H}_B^{(II)}$ if $\mathbf{H}_A^{(I)} = \mathbf{H}_A^{(II)}$]. If θ_{23} is the same as θ_{BC} within the angular tolerance, the three different reflections form a triplet satisfying the three interplanar angles θ_{AC} , θ_{AB} and θ_{BC} . Then the pair $(\mathbf{H}_A^{(II)}, \mathbf{H}_B^{(II)})$ is kept in $\{\mathbf{H}_A^{(II)}, \mathbf{H}_B^{(II)}\}$. If θ_{23} is different from θ_{BC} , the pair is discarded. *LauePt4* also discards the pair $(\mathbf{H}_A^{(II)}, \mathbf{H}_B^{(II)})$ if all the pairs in $\{\mathbf{H}_A^{(I)}, \mathbf{H}_C^{(I)}\}$ do not have a common reflection with this pair. As shown in Fig. 4, after this filtering process, the number of pairs in $\{\mathbf{H}_A^{(II)}, \mathbf{H}_B^{(II)}\}$ is dramatically

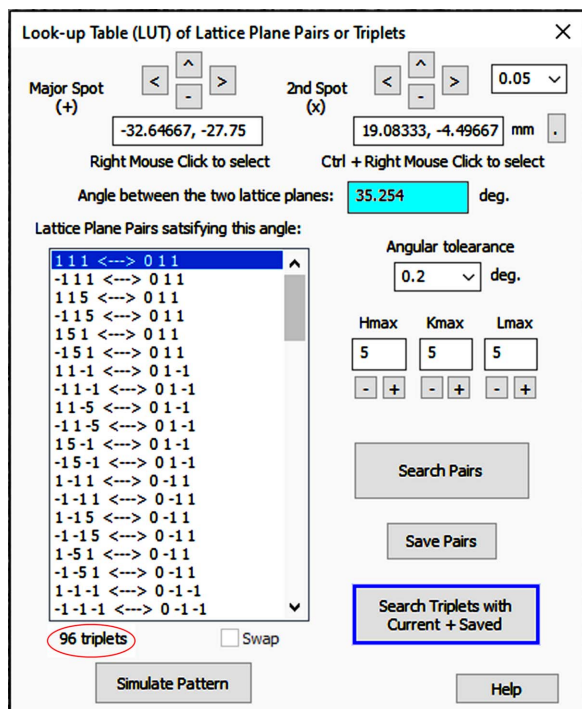


Figure 4 The dialog for LUT-P and LUT-T searches. The list box has the 96 triplets obtained from the LUT-T search performed on spots *A*, *B* and *C* in Fig. 2(a).

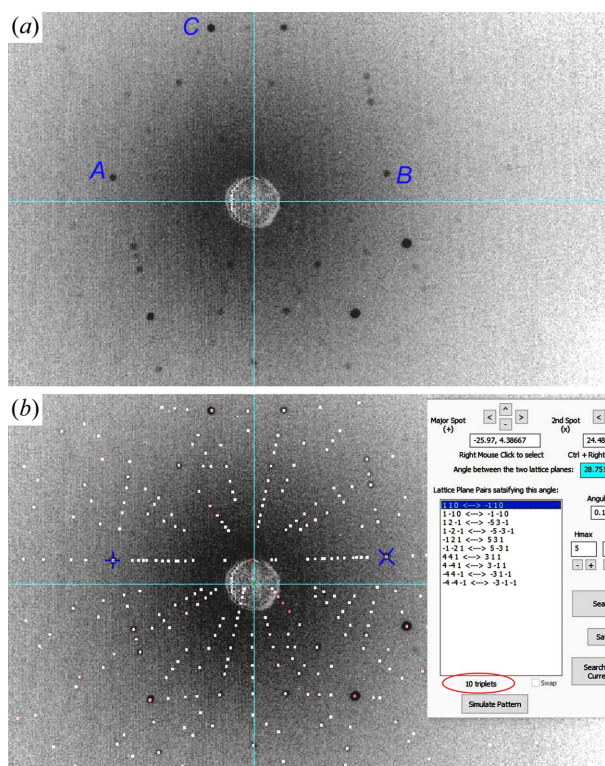


Figure 5 The LUT-T search and pattern simulation of a β -Ga₂O₃ crystal. (a) Recorded back-reflection Laue image of the crystal. (b) Ten reflection triplets obtained from the LUT-T search for spots *A*, *B*, *C* in panel (a) and the matching simulation based on the valid solution (110, $\bar{1}10$). Imaging distance $D = 45.8$ mm, $H_{\max} = K_{\max} = L_{\max} = 5$ and $\Delta\theta = 0.1^\circ$. Image width 112 mm.

reduced from 1128 to 96. Further examination shows that the list in Fig. 4 only contains three different configurations: ($\mathbf{H}_A^{(II)} = 111$, $\mathbf{H}_B^{(II)} = 011$), (115, 011) and (115, $\bar{1}14$). All the other pairs are symmetry-related equivalents to these three pairs. Now we can let *LauePt4* simulate the patterns of these three pairs, from which the (115, 011) and (115, $\bar{1}14$) pairs are readily excluded since the corresponding patterns do not match the recorded Laue image. The remaining pair (111, 011) is the correct solution, which is consistent with Figs. 2 and 3.

As another example, Fig. 5 shows the application of LUT-T to a β -Ga₂O₃ crystal with much lower symmetry than Si. β -Ga₂O₃ has a monoclinic structure with unit-cell parameters $a = 12.214$ Å, $b = 3.0371$ Å, $c = 5.7981$ Å, $\alpha = \gamma = 90^\circ$ and $\beta = 103.83^\circ$ (Goldberg, 2001). The crystal studied in Fig. 5 was grown by the edge-defined film-fed growth method. During growth, unpredictable twinning can happen multiple times, even at very early stages in the seed region (Ueda *et al.*, 2016). As a result, the as-grown crystal may exhibit various orientations other than the intended one. In Fig. 5(a), the scattering of the Ga atoms is strong, producing strong background noise such that many of the weak diffraction spots are hardly visible.

Similarly to the above example, here we first select spots *A* and *C* to perform the LUT-P search, which gives 768 reflection pairs satisfying the interplanar angle 17.974° between *A* and *C*. Since β -Ga₂O₃ has many more allowed Bragg reflections than Si, here we set a smaller angular tolerance, $\Delta\theta = 0.1\%$. After saving the 768 pairs, we select spot *B* in Fig. 5(a) to perform LUT-P between *A* and *B*, which gives 996 reflection pairs. Next, the LUT-T filtering process is carried out between the two sets of reflection pairs, which leaves only ten pairs in the list of Fig. 5(b) that satisfy the reflection triplet conditions. Simulation with the first pair ($A = 110$, $B = \bar{1}10$) immediately gives the Laue pattern that matches the recorded image in Fig. 5(b). Meanwhile, the crystal orientation along the z' axis (nearly perpendicular to the crystal surface) is determined as (0.0223, -1 , -0.0903), which is 2.73° away from the (0 $\bar{1}0$) lattice plane. Here ($A = 110$, $B = \bar{1}10$) is the only solution, since trials of the other nine pairs do not produce matching patterns.

Obviously, LUT-T is a faster and more efficient procedure compared with the above G-rotation and LUT-P methods. Our numerous tests show that LUT-T is quite reliable and robust even if high-index spots are selected. To increase the success rate effectively while reducing the number of triplets in the list, the geometric parameters in Fig. 1, such as D and Φ , should be as accurate as possible (by means of calibration) so that the user can set a small angular tolerance and assign large values to H_{\max} , K_{\max} and L_{\max} to perform the search. However, inaccuracy in these parameters does not destabilize the process, and the user only needs to make the search over a broader range or try slightly different values for these parameters.

5. Summary

LauePt is a general-purpose application for simulating, indexing and analyzing Laue patterns. It has a single small

executable file (~2.5 MB) written with Visual C++ and Microsoft Foundation Classes. It can be run on any Windows computer or laptop without the need for large memory or powerful CPUs. Previous versions of *LauePt* do not have an efficient algorithm for recognizing or fitting Laue images taken from crystals of unknown orientation. In this paper we have introduced a major upgrade of *LauePt* to Version 4, to which three powerful search and fitting schemes have been added that completely overcome this difficulty.

The first scheme is a single-spot G-rotation method, which allows the user to select a single (low-index) diffraction spot from a Laue image and assign a trial reflection index $\mathbf{G} = hkl$ to it. *LauePt4* is then able to simulate all Laue patterns that have this reflection fixed along the specific direction in space. Determination of a 3D crystal orientation requires two directions. Here *LauePt4* allows the user to choose the other direction perpendicular to \mathbf{G} , which corresponds to a 360° rotation of the crystal around \mathbf{G} , to search for the correct simulation. The second scheme is a two-spot look-up method, LUT-P, that allows the user to select two diffraction spots. *LauePt4* then searches for all the reflection pairs in the user-defined range that satisfy the interplanar angle of these two spots. Afterwards, *LauePt4* can simulate the Laue patterns of all the candidate pairs to find the correct pair and simulation. The third scheme is a three-spot look-up method, LUT-T, in which *LauePt4* searches for reflection triplets satisfying the three interplanar angles between three user-selected diffraction spots and performs the simulation based on each triplet.

Our extensive tests show that the three schemes are all powerful for recognizing Laue patterns, with the LUT-T method being the most efficient. It can rapidly recognize almost any Laue images recorded from any crystal, even if the image quality is poor.

Overall, the current upgrade makes *LauePt4* a reliable crystallography tool with comprehensive functions. Since the schemes are implemented with highly efficient algorithms and optimized C++ code, the program size remains small and the running speed is extremely fast. Almost all search or simulation procedures (including intensity calculations) are executed and finished instantly without any noticeable delay. The fast speed and responsiveness of *LauePt4* greatly facilitate quick and arbitrary crystal rotation (with instant pattern update) and rapid screening of reflection pairs by simulations in LUT-P and LUT-T, even if the list contains thousands of pairs.

LauePt4 is freely available for scientific use, and currently it can be downloaded from <https://you.stonybrook.edu/xraycharacterization/lauept4/>. It will also be uploaded to other public scientific software websites soon.

Acknowledgements

VWH is a Syosset High School student and is grateful to MD and his collaborators for providing the internship to upgrade

the *LauePt* program during the summers of 2021 and 2022. VWH appreciates all the suggestions, instructions, discussions, testing and feedback provided by MD's group members during coding. Dr X. Huang at Argonne National Laboratory is thanked for sharing the *LauePt3* source code and also for many critical discussions. The *LauePt* program was initiated in MD's group more than two decades ago with contributions from many former postdoctoral researchers and graduate students, which are acknowledged.

References

- Asadchikov, V., Buzmakov, A., Chukhovskii, F., Dyachkova, I., Zolotov, D., Danilewsky, A., Baumbach, T., Bode, S., Haaga, S., Hänschke, D., Kabukcuoglu, M., Balzer, M., Caselle, M. & Suvorov, E. (2018). *J. Appl. Cryst.* **51**, 1616–1622.
- Barabash, R., Ice, G. E., Larson, B. C., Pharr, G. M., Chung, K.-S. & Yang, W. (2001). *Appl. Phys. Lett.* **79**, 749–751.
- Cui, Q., Li, R., Zhong, D., You, F., Xu, K., Xu, H., Wang, T., Liu, J., Cao, L., Tang, J., Huang, W. & Teng, B. (2020). *J. Cryst. Growth*, **547**, 125802.
- Danilewsky, A. N. (2020). *Cryst. Res. Technol.* **55**, 2000012.
- Goldberg, Y. (2001). *Properties of Advanced Semiconductor Materials GaN, AlN, InN, BN, SiC, SiGe*, edited by M. E. Levinshtein, S. L. Rumyantsev & M. S. Shur, pp. 31–47. New York: John Wiley & Sons Inc.
- Gupta, V. K. & Agnew, S. R. (2009). *J. Appl. Cryst.* **42**, 116–124.
- Gürsoy, D., Sheyfer, D., Wojcik, M., Liu, W. & Tischler, J. Z. (2022). *J. Appl. Cryst.* **55**, 1104–1110.
- Heizmann, J.-J. & Baro, R. (1967). *Bull. Mineral.* **90**, 41–43.
- Huang, X. R. (2010). *J. Appl. Cryst.* **43**, 926–928.
- Liu, W., Ice, G. E., Larson, B. C., Yang, W., Tischler, J. Z. & Budai, J. D. (2004). *Metall. Mater. Trans. A*, **35**, 1963–1967.
- Morawiec, A. (2020). *Acta Cryst.* **A76**, 719–734.
- Oriwol, D., Carl, E.-R., Danilewsky, A. N., Sylla, L., Seifert, W., Kittler, M. & Leipner, H. S. (2013). *Acta Mater.* **61**, 6903–6910.
- Örs, T., Micha, J.-S., Gey, N., Michel, V., Castelnaud, O. & Guinebretiere, R. (2018). *J. Appl. Cryst.* **51**, 55–67.
- Purushottam Raj Purohit, R. R. P., Tardif, S., Castelnaud, O., Eymery, J., Guinebretière, R., Robach, O., Ors, T. & Micha, J.-S. (2022). *J. Appl. Cryst.* **55**, 737–750.
- Riquet, J.-P. & Bonnet, R. (1979). *J. Appl. Cryst.* **12**, 39–41.
- Robach, O. & Micha, J.-S. (2015). *LaueTools: Open Source Python Packages for X-ray MicroLaue Diffraction Analysis*, <https://sourceforge.net/projects/lauetools/>.
- Seret, A., Gao, W., Juul Jensen, D., Godfrey, A. & Zhang, Y. (2022). *J. Appl. Cryst.* **55**, 1085–1096.
- Ueda, O., Ikenaga, N., Koshi, K., Iizuka, K., Kuramata, A., Hanada, K., Moribayashi, T., Yamakoshi, S. & Kasu, M. (2016). *Jpn. J. Appl. Phys.* **55**, 1202BD.
- Ulrich, O., Biquard, X., Bleuet, P., Geaymond, O., Gergaud, P., Micha, J. S., Robach, O. & Rieutord, F. (2011). *Rev. Sci. Instrum.* **82**, 033908.
- Woodworth, A. A., Neudeck, P. G., Sayir, A., Solá, F., Dudley, M. & Raghathamachar, B. (2014). *J. Cryst. Growth*, **392**, 34–40.
- Zhang, F. G., Bornert, M., Petit, J. & Castelnaud, O. (2017). *J. Synchrotron Rad.* **24**, 802–817.
- Zolotov, D., Buzmakov, A., Grigoriev, M. & Schelokov, I. (2020). *J. Appl. Cryst.* **53**, 781–788.