Acta Cryst A
FOUNDATIONS
ADVANCES

ISSN: 2053-2733
journals.iucr.org/a

Indexing of diffraction patterns for determination of crystal orientations

Adam Morawiec

Acta Cryst. (2020). A**76**, 719–734

Acta Cryst A
FOUNDATIONS
ADVANCES

ISSN 2053-2733

Indexing of diffraction patterns for determination of crystal orientations

Adam Morawiec*

Polish Academy of Sciences, Institute of Metallurgy and Materials Science, Reymonta 25, Krakow 30-059, Poland.
*Correspondence e-mail: nmmorawi@cyf-kr.edu.pl

Received 29 April 2020
Accepted 21 September 2020

Edited by I. A. Vartaniants, DESY, Germany

Keywords: diffraction; crystal orientation; indexing; crystallographic computing.

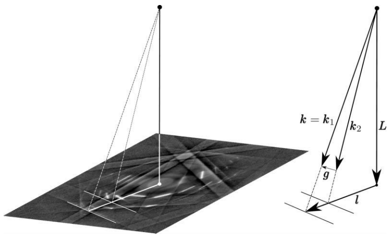
The task of determining the orientations of crystals is usually performed by indexing reflections detected on diffraction patterns. The well known underlying principle of indexing methods is universal: they are based on matching experimental scattering vectors to some vectors of the reciprocal lattice. Despite this, the standard attitude has been to devise algorithms applicable to patterns of a particular type. This paper provides a broader perspective. A general approach to indexing of diffraction patterns of various types is presented. References are made to formally similar problems in other research fields, *e.g.* in computational geometry, computer science, computer vision or star identification. Besides a general description of available methods, concrete algorithms are presented in detail and their applicability to patterns of various types is demonstrated; a program based on these algorithms is shown to index Kikuchi patterns, Kossel patterns and Laue patterns, among others.

1. Introduction

Indexing of diffraction patterns is a process of ascribing Miller indices to diffraction reflections. In general, indexing is performed without knowledge of the crystal structure [see *e.g.* Kabsch (1988), Duisenberg (1992) and Morawiec (2017)]. Here, we consider end-indexing, *i.e.* indexing of patterns originating from known structures. In particular, it is assumed that the unit-cell parameters are known, and that one knows which reflections are forbidden and which can show up in the pattern. End-indexing is used mainly for the determination of crystal orientations. Knowing the locations of reflections and their Miller indices, one can easily get the orientation.

A large number of papers on indexing have been published, but they are fragmentary, with a focus on particular types of patterns. They usually describe indexing procedures together with those for automatic detection of reflections. In effect, solutions to the same formal problem are dispersed over many loosely related publications. This paper gives a more panoramic view of indexing-based orientation determination. It reviews the most effective methods of crystal orientation determination, but without focusing on diffraction patterns of a specific type. The point is to concentrate attention on the essence of the orientation determination problem without distraction by peripheral issues of pattern processing. Moreover, techniques used in other research areas to solve related problems are briefly described. The point is to indicate fields from which ideas for creating more efficient indexing algorithms can be adapted.

The general approach to orientation determination opens the opportunity to create algorithms applicable to patterns of various types. If a program code specialized to a certain type is needed, one may write it by starting from the general frame-



https://doi.org/10.1107/S2053273320012802

work and being aware of the variety of existing algorithmic solutions. This approach also helps to avoid unnecessary repetitions; in the past, the same methods have been independently invented by researchers working in different fields or using different diffraction methods. For instance, indexing based on triplets of reciprocal-lattice vectors was used for solving Laue patterns (Ohba *et al.*, 1981). Then it was invented again as 'triplet indexing' of electron backscatter diffraction (EBSD) patterns (Wright & Adams, 1992; Adams *et al.*, 1993). In the meantime, voting by matching triangles was independently introduced for star identification (Groth, 1986). Similarly, indexing methods based on accumulation along curves in the space of rotations were described by Beyerlein *et al.* (2017) for monochromatic serial crystallography data and by Gevorkov *et al.* (2020) for polychromatic patterns, but the authors were unaware of a similar approach proposed much earlier for indexing *K*-line patterns (Morawiec & Bieda, 2005).

Besides the general description of pattern-type independent indexing methods, the paper contains example indexing algorithms. An implementation of these algorithms is shown to index Kikuchi patterns, Kossel patterns or Laue patterns, and it is shown to be quite efficient when applied to EBSD patterns. To our knowledge, solving such a broad spectrum of patterns with the same code has not been attempted before, but there is still room for creating more universal indexing programs. It is believed that further progress can be facilitated by combining the methods of computational geometry, computer science and other research fields. Thus, the paper is addressed primarily to crystallographers writing indexing software, and it may also be of interest to users of automatic systems for orientation determination who look for better understanding of the indexing engines installed in those systems.

The issue of crystal orientation determination arises in many fields, *e.g.* in conventional electron microscopic analysis of individual crystallites or crystal interfaces and in diffractometric measurements, but its principal application is orientation mapping. Most orientation mappings rely on explicit indexing. An indexing engine is at the heart of an orientation determination system. The technique of orientation mapping is widely applied in studies of the microstructure and texture of polycrystalline materials. Automatic systems collect diffraction patterns originating from small areas of a specimen, determine local orientations, and provide digital maps with orientation parameters ascribed to individual pixels or grains. The automatically generated orientation maps complement conventional contrast images with quantitative data; besides showing the topography of the crystallites, the maps also contain information on grain orientation. Contemporary orientation mapping methods originated from techniques relying on Kossel patterns (Inokuti *et al.*, 1980, 1985, 1987). Nowadays, most orientation maps are created based on step-by-step scans on computer-controlled instruments equipped with digital cameras. At each step, a diffraction pattern is acquired and an orientation is determined. Particularly successful are systems using scanning electron microscopy (SEM) and EBSD patterns (Adams *et al.*, 1993). Although less often, orientation maps are also created using other types of patterns. Similar to EBSD is the technique relying on transmission Kikuchi diffraction in SEM (Keller & Geiss, 2012; Fundenberger *et al.*, 2016). Orientation maps are acquired by conventional transmission electron microscopy (TEM) using Kikuchi patterns and spot patterns (Fundenberger *et al.*, 2003; Rauch *et al.*, 2008; Morawiec *et al.*, 2014). TEM-based maps were obtained by so-called 'conical dark-field scanning', with the orientation at a given point determined from images corresponding to various incident-beam directions (Wright & Dingley, 1998); with this approach, step-by-step beam scanning is avoided. Orientation maps were also acquired by high-energy X-ray diffraction (Poulsen, 2004). The aforementioned mapping methods are based on monochromatic radiation, but polychromatic X-rays have also been used [see *e.g.* Miyamoto *et al.* (2011) and Whitley *et al.* (2015) (laboratory X-ray setups), and Tamura *et al.* (2003) and Ice & Pang (2009) (synchrotron setups)]. Besides 2D maps, diffraction-based determination of local orientations is also used for reconstruction of 3D microstructures by resolving diffraction patterns generated by a synchrotron beam transmitted through a multi-grain specimen (*e.g.* Poulsen, 2004), but methods of multi-grain indexing, *i.e.* solving patterns with reflections from multiple differently oriented grains, are not considered here.

Clearly, computer programs for orientation determination are expected to be general so that they can be applied to arbitrary crystal structures. The other important aspects are orientation resolution and accuracy. However, the two critical features which determine the quality of programs for automatic orientation determination are robustness and speed. Indexing of diffraction patterns from polycrystalline materials can be challenging, especially in the case of a small grain size or a large density of defects, as the quality of the patterns may be poor. Moreover, with thousands or millions of patterns used for creating a pixelized orientation map, what matters is the time efficiency of the pattern processing. For speed, patterns are collected to be of a quality sufficient for indexing, and the pattern-solving software is expected to be capable of dealing with coarse patterns. Generally, the software must be able to handle poorly localized and spurious reflections. Orientation determination algorithms are only as good as their ability to cope with imperfections in the input data.

The paper is organized as follows. It begins with basic concepts of crystal orientation determination via assignment of reciprocal-lattice vectors to experimental scattering vectors (*i.e.* via end-indexing) and with a reminder of ways to get the coordinates of these vectors. Formal aspects of the indexing problem are then summarized, and some conceptually similar problems considered in computer science and in computational geometry are listed. Subsequent sections review practical methods of solving the indexing problem. The most effective methods rely on the accumulation of contributions either in a discrete space (Section 5) or in the space of orientations (Section 6). The described schemes are illustrated by example indexing algorithms. The algorithms are formulated without reference to particular types of diffraction data.

An implementation of these algorithms is shown to index patterns of various types. Before the closing remarks, there are brief sections on the testing of indexing algorithms and on the figures of merit used in end-indexing.

One needs to note that crystal orientation can also be determined without identification of individual reflections, by matching an experimental pattern as a whole to simulated patterns (Rauch *et al.*, 2008; Gupta & Agnew, 2009; Chen *et al.*, 2015). These methods of direct matching differ considerably from those involving indexing. They are briefly described in Appendix *A*.

2. Reflection-identifying vectors

Orientation determination systems based on explicit pattern indexing perform detection and identification of individual reflections. Methods for detecting reflection positions depend on the type of pattern. Clearly, the locations of diffraction spots are determined using peak-detection techniques. To get the parameters of bands or lines in EBSD or Kikuchi patterns, various variants of the Hough transform are applied. The positions of detected reflections are used to get the coordinates of the corresponding scattering vectors. To get the Miller indices of reflections, these scattering vectors are matched with reciprocal-lattice vectors calculated from crystallographic data. Ultimately, indexing algorithms rely on two lists: a list of scattering vectors and a list of reciprocal-lattice vectors.

Before going on to a more detailed description of these lists, one needs to note the issue of vector discrimination. If two vectors on one of the lists are closer than a permissible uncertainty in vector determination, they could be matched with the same vector on the other list. This complication can be avoided by eliminating such close vectors at the outset when constructing the lists.

2.1. List of scattering vectors

The scattering vector $\mathbf{g}$ corresponding to a diffraction spot can be calculated by simple application of the Laue equation $\mathbf{g} = \mathbf{k} - \mathbf{k}_0$, where $\mathbf{k}_0$ and $\mathbf{k}$ are the wavevectors of the incident beam and the reflected beam, respectively. If the magnitudes of the wavevectors are not accessible (as in the case of spots in Laue patterns), the direction $\hat{\mathbf{g}}$ of the scattering vector can be obtained from the location of a spot by normalizing the difference $\hat{\mathbf{k}} - \hat{\mathbf{k}}_0$ ($\propto \hat{\mathbf{g}}$). Here and below, symbols with a circumflex $\hat{}$ denote normalized vectors.

For *K*-line patterns,[1] the primary issue is getting scattering vectors from individual *K* lines. The relevant procedure was described by Morawiec (1999), but it works only for sufficiently curved conics, *e.g.* for Kossel lines. With electron diffraction, due to the shortness of the wavelength λ, *K* lines are usually seen as straight lines and the expression given by Morawiec (1999) is not applicable, but getting the scattering vector from the line positions is simple. Take a Kikuchi pattern with pairs of deficit and excess (dark and bright) lines. If $\mathbf{L}$ denotes the vector from the radiation source to the pattern center, and $\mathbf{l}$ is the vector perpendicular to the Kikuchi line from the pattern center to the line, then $\mathbf{k} = (\mathbf{L} + \mathbf{l})/(\lambda\,|\mathbf{L} + \mathbf{l}|)$ is a wavevector to the point on the Kikuchi line nearest to the pattern center (Fig. 1). Having two such wavevectors $\mathbf{k}_1$ and $\mathbf{k}_2$ for the bright and dark lines of one pair, respectively, one can calculate the scattering vector $\mathbf{g} = \mathbf{k}_1 - \mathbf{k}_2$. In EBSD patterns, the locations of lines approximating whole bands are usually determined. In such cases, only directions $\hat{\mathbf{g}}$ of the scattering vectors can be obtained. For a given band, $\hat{\mathbf{g}}$ is the normal to the plane through the line approximating the band and the source of radiation.

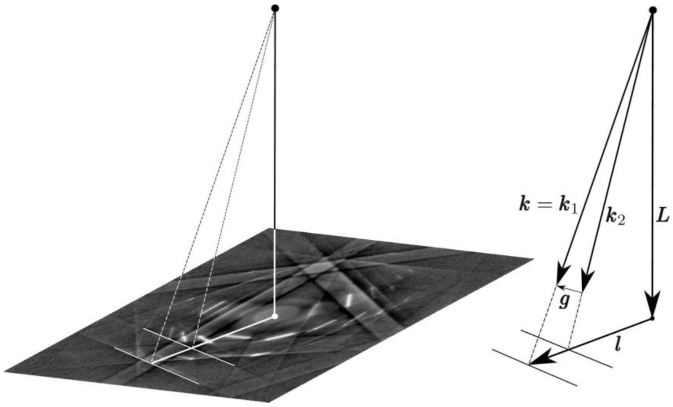


Figure 1
The construction used for calculating the scattering vector $\mathbf{g}$ corresponding to a pair of Kikuchi lines.

2.2. List of reciprocal-lattice vectors

Knowing the crystal structure, one can easily get the reciprocal-lattice vectors. Because of crystal symmetry there are families of symmetrically equivalent reflecting planes. To get all symmetrically equivalent members of the family, one needs to apply the symmetry operators of the crystal's Laue group to a given reciprocal-lattice vector. The primary criterion for including a family of reciprocal-lattice vectors is the set of expected reflection intensities obtained from the structure factors.[2] In practice, it makes sense to support this criterion by visual inspection of indexing results, to find out which reflections are actually detectable and which are not. Clearly, if only the directions of the vectors are to be matched, there is no distinction between families corresponding to parallel lattice planes, *i.e.* between families $\{nh\ nk\ nl\}$ differing by n.

In order to match the scattering vectors to reciprocal-lattice vectors, one needs to express the latter in a Cartesian coordinate frame $\mathbf{e}_i$ ascribed to the crystal lattice. With a reciprocal-lattice vector $\mathbf{h}$ specified by Miller indices $(hkl) = (h_1 h_2 h_3)$, the ith Cartesian coordinate of $\mathbf{h}$ is $\sum_j h_j\, \mathbf{a}^j \cdot \mathbf{e}_i$,

[1] The family of *K*-line diffraction patterns (Cowley, 1981) comprises among others Kikuchi patterns, EBSD patterns, electron channeling patterns, convergent-beam electron diffraction (CBED) patterns, and X-ray Kossel and pseudo-Kossel patterns.

[2] For electron diffraction patterns from complex crystal structures, Wright *et al.* (2019) proposed the use of a criterion based on the integrated intensities of reflections in dynamically simulated patterns.

where $\mathbf{a}^j$ is the jth basis vector of the reciprocal lattice (*cf.* Busing & Levy, 1967).

2.3. Vector magnitudes and reflection intensities

For complex crystal structures, to match scattering vectors to reciprocal-lattice vectors one needs to take into account the vector magnitudes. With some patterns, *e.g.* Laue patterns or most EBSD patterns, the magnitudes of the scattering vectors are not accessible. Depending on the accessibility of the vector magnitudes, there are two basic types of input data. One type relies on the measured coordinates of complete scattering vectors, the other on the directions of the scattering vectors. Clearly, the case with known vector magnitudes can be reduced to one with known directions by normalizing all vectors.

Moreover, in some cases, reflection intensities need to be taken into account (*cf.* Chen *et al.*, 2012), but it is common to index diffraction patterns based on their geometry, with intensities used only for dichotomous division of reflections into 'on' and 'off' types.

Crystal structures encountered in metallurgy and materials science are often relatively simple, and both vector magnitudes and reflection intensities can be ignored. The initial considerations below account for vector magnitudes, but the algorithms are limited to the case of normalized vectors. This simplifies the procedures and their description. It is also clear that such procedures still allow for indexing a broad range of diffraction patterns. In particular, they index bands in patterns obtained via EBSD – the technique that is nowadays the main tool for orientation mapping.

The described methods can also be applied to more complex structures (say, in macromolecular crystallography) assuming that the accuracy of the input data is adequate to the complexity; clearly, the temporal effectiveness of the algorithms is affected by the large number of reflections appearing in such cases.

3. Formal aspects of end-indexing

3.1. Basic relationships

As was mentioned before, the key idea for indexing procedures is to match the scattering vectors obtained from the diffraction pattern and the vectors of the crystallographic reciprocal lattice. The reciprocal-lattice vectors corresponding to detectable reflections will be denoted by $\mathbf{h}_m$, with m ranging from 1 to a certain M. The coordinates of these vectors are given in the Cartesian frame attached to the crystal. On the other hand, by analyzing diffraction reflections one gets scattering vectors $\mathbf{g}_n$ enumerated by $n = 1, \ldots, N$, with Cartesian coordinates in the sample reference system.

Each legitimate vector $\mathbf{g}_n$ corresponds to a certain $\mathbf{h}_m$, and the task is to match the scattering vectors $\mathbf{g}_n$ with some properly oriented reciprocal-lattice vectors $\mathbf{h}_m$. In principle, each $\mathbf{g}_n$ ($n = 1, \ldots, N$) is to be matched to a unique $\mathbf{h}_m$ ($m = 1, \ldots, M$), *i.e.* indexing is the procedure of determining an injective mapping ϖ:$\{1, \ldots, N\} \rightarrow \{1, \ldots, M\}$ ascribing subscripts enumerating vectors $\mathbf{h}_m$ to subscripts of vectors $\mathbf{g}_n$. With known ϖ, Miller indices $(hkl) = (h_1h_2h_3)$ of the nth reflection (or the scattering vector $\mathbf{g}_n$) are $h_i = \mathbf{a}_i \cdot \mathbf{h}_{\varpi(n)}$, where $\mathbf{a}_i$ is the ith basis vector of the direct lattice. Point-symmetry operations map the set of $\mathbf{h}_m$ vectors onto itself and induce permutations of the elements of the set $\{1, \ldots, M\}$. If σ is such a permutation, the assignment ϖ is equivalent to $\sigma\varpi$. Knowing one representative of the class of equivalent assignments, it is straightforward to get all other elements of that class.

Apart from experimental errors, there exists a proper rotation, say O, transforming all vectors $\mathbf{g}_n$ onto vectors $\mathbf{h}_m$, *i.e.* $\mathbf{h}_{\varpi(n)} = O\mathbf{g}_n$ for all n in $\{1, \ldots, N\}$.[3] Let the Cartesian coordinates of the vectors $\mathbf{g}_n$ and $\mathbf{h}_m$ constitute columns of matrices G and H, respectively. The relationship between vectors $\mathbf{g}_n$ and $\mathbf{h}_m$ can be briefly expressed as

$$OG = HP,$$

where P is an unknown $M \times N$ matrix representing ϖ. P has zero entries everywhere except a value of 1 at each entry (mn) such that the vector $\mathbf{h}_m$ corresponds to the vector $\mathbf{g}_n$, *i.e.* there is exactly one '1' in each column of P. The search for O corresponds to the orientation determination problem, and the search for P solves the indexing problem.

It is obvious that random errors in the locations of detected reflections are inevitable. These errors affect the scattering vectors and the entries of G, and the matrix OG is then only approximately equal to HP. Therefore, one can portray the considered problem (with error-affected but legitimate scattering vectors) in the following form: find P (ϖ) and O minimizing

$$\sum_{n=1}^{N} \left| \mathbf{h}_{\varpi(n)} - O\mathbf{g}_n \right|^2 = \|HP - OG\|^2, \tag{1}$$

where $\|X\| = [\mathrm{tr}(X^{\mathrm{T}}X)]^{1/2}$ denotes the Frobenius norm of matrix X. In this formulation, the problem lies in the domain of combinatorial (P or ϖ) and continuous (O) optimization.

3.2. Related solvable problems

It is instructive to break down the problem of minimization of equation (1) into simpler problems concerning the case of $N = M$, *i.e.* with the sets of matched vectors having the same cardinality, and the matrices G on H having the same dimensions. These simplified partial problems are, first, how to get the permutation matrix P minimizing $\|HP - G\|^2$, and second, how to get the special orthogonal matrix O minimizing $\|H - OG\|^2$.

The first problem is directly related to the linear assignment problem: given a real $N \times N$ cost matrix C, find a permutation matrix P minimizing $\mathrm{tr}(PC^{\mathrm{T}})$. It is solved by the classic Kuhn Hungarian algorithm (Kuhn, 1955). The minimization of $\|HP - G\|^2$ is easily reduced to a linear assignment problem with the cost matrix $C = -H^{\mathrm{T}}G$.

[3] With the matrix O in front of $\mathbf{g}$, the convention of Bunge (1982) is followed. It is usually used in texture analysis and in orientation mapping systems.

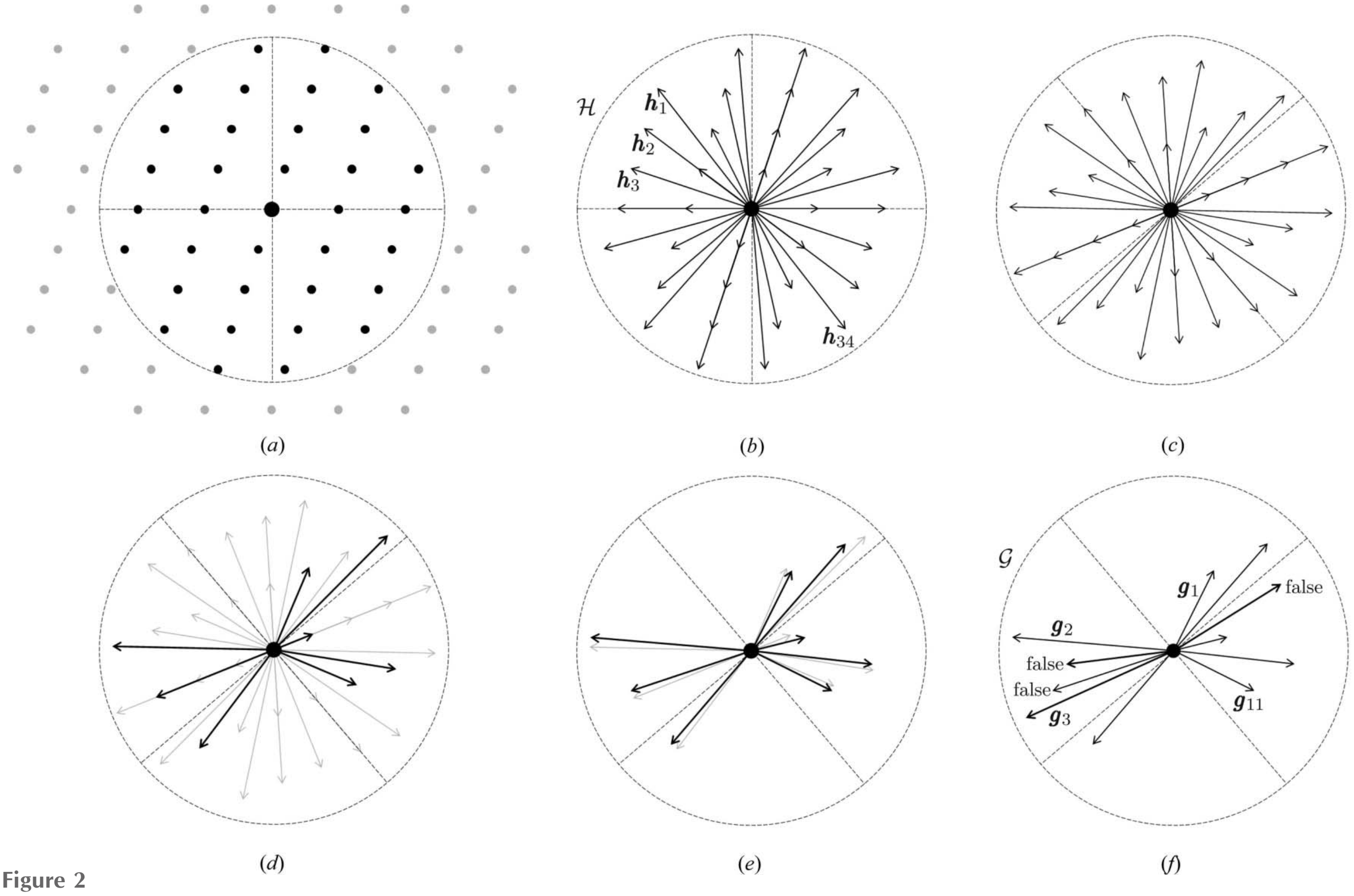


Figure 2
A two-dimensional illustration of the relationship between the reciprocal-lattice and scattering vectors corresponding to detected reflections. (*a*) The nodes of the reciprocal lattice. (*b*) Vectors $\mathbf{h}_m$ with endpoints at nodes corresponding to detectable reflections, *i.e.* figure $\mathcal{H}$. (*c*) Rotated vectors $\mathbf{h}_m$. (*d*) Rotated vectors $\mathbf{h}_m$ corresponding to detected reflections, *i.e.* error-free scattering vectors. (*e*) The scattering vectors affected by experimental errors. (*f*) Error-affected and false scattering vectors, *i.e.* figure $\mathcal{G}$.

The minimization of $\|H - OG\|^2$ with respect to O is related to the problem of determination of the special orthogonal matrix nearest to a square matrix: given an invertible square matrix A, determine the special orthogonal matrix $\mathcal{SO}(A)$ minimizing $\|A - O\|^2$. A general way of getting $\mathcal{SO}(A)$ is via singular value decomposition of A. With $A = U\Sigma V^{\rm T}$, one has

$$\mathcal{SO}(A) = USV^{\rm T},$$

where $S = {\rm diag}[+1, +1, {\rm sign}(\det(A))]$. Alternative methods of getting $\mathcal{SO}(A)$ rely on polar decomposition of A (Morawiec, 2004) or use the link between special orthogonal matrices and quaternions (Morawiec, 1998). The problem of getting a special orthogonal matrix matching corresponding vectors is frequently referred to as determination of absolute orientation. Given two sets of corresponding vectors in the form of the matrices G and H such that $HG^{\rm T}$ is invertible, the function $\|H - OG\|^2$ has a minimum at

$$O = \mathcal{SO}(HG^{\rm T}),$$

i.e. the problem of getting the absolute orientation is reduced to finding the special orthogonal matrix nearest to $HG^{\rm T}$ (MacKenzie, 1957; Wahba *et al.*, 1966; Kabsch, 1976, 1978; Stephens, 1979). Each advanced orientation determination program uses this or another form of the operator $\mathcal{SO}$.

3.3. Rotations *versus* proper rotations

There is an issue of the domain of O in equation (1): one may ask whether this domain comprises all rotations or just proper rotations. A couple of facts need to be recalled. First, orientation of an object (a crystallite) corresponds to a proper rotation and is determined by a special orthogonal matrix. Second, almost all materials studied by orientation mapping have centrosymmetric crystal structures. Third, the figure built of vectors $\mathbf{h}_m$ is centrosymmetric due to Friedel's law. In the case of centrosymmetric crystals, with the matrix O relating G to HP assumed to be special orthogonal (*i.e.* the matrix may be a result of using $\mathcal{SO}$), one gets a single solution modulo the proper operations of the Laue group. If the matrix relating G to HP was allowed to be orthogonal, one would get a single solution modulo *all* operations of the Laue group. If the crystal structure is non-centrosymmetric, then because of Friedel's law the unequivocal crystallite orientation is indeterminable by conventional orientation mapping methods (see Appendix A).

3.4. Spurious scattering vectors and computational context

The crystallographic problem described in Section 3.1 is formally similar to the problem of registering two figures of

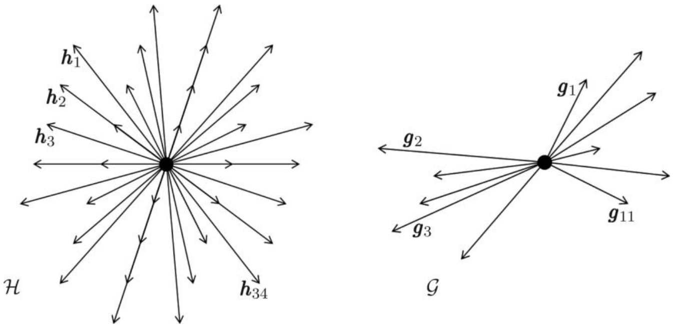


Figure 3
A schematic 2D illustration of the orientation determination problem based on Fig. 2. It shows the input data available when the magnitudes of the scattering vectors $\mathbf{g}_n$ are known. The task is to find the rotation of $\mathcal{G}$ leading to the 'best match' between 'large' subfigures of $\mathcal{G}$ and $\mathcal{H}$.

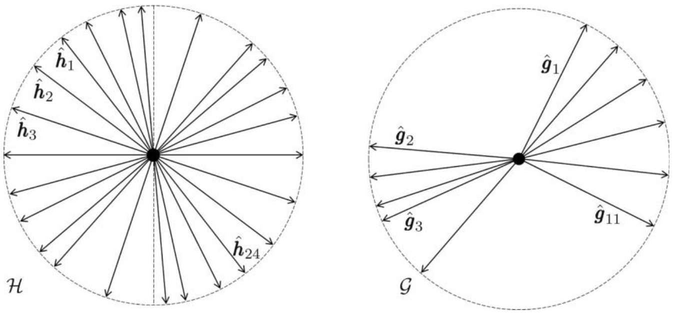


Figure 4
A schematic 2D illustration of the orientation determination problem in the case when the magnitudes of the scattering vectors are not known and all vectors are normalized to 1 (*i.e.* illustration of the constellation problem). It is based on Fig. 2.

spatially distributed points, which arises in other fields. In computer science, with points as graph vertices and distances between points as attributes (weights) of edges, the problem can be seen as inexact edge-attributed subgraph matching [see *e.g.* chapter 2 of the work by Bengoetxea (2002)]. If reflection intensities are taken into account, it turns into edge-attributed (by distance) vertex-attributed (by intensity) subgraph matching. In computational geometry, one considers geometric point-pattern matching. It is usually formulated in terms of point 'sets' and in a more general framework of rigid motion: given two sets $\mathcal{G}$ and $\mathcal{H}$ of spatially distributed points, determine the smallest ε_H such that there is a rigid motion of the 'query' figure $\mathcal{G}$ that brings each of its points within a distance ε_H of a point in $\mathcal{H}$ [see *e.g.* Goodrich *et al.* (1999)]. In other words, the task is to match the figure $\mathcal{G}$ to a subfigure of $\mathcal{H}$.[4] With $\mathcal{G}$ being a figure composed of endpoints of the scattering vectors $\mathbf{g}_n$ and $\mathcal{H}$ denoting a figure composed of endpoints of reciprocal-lattice vectors $\mathbf{h}_m$, the indexing problem is to determine an optimal subfigure of $\mathcal{H}$ nearly congruent to the complete figure $\mathcal{G}$.

However, the account of Section 3.1 is not complete as it ignores an important aspect of automatic detection of reflections: the input data are affected not only by small inaccuracies in reflection positions, but also by gross errors, leading to illegitimate scattering vectors. For a graphic illustration of the problem, see Figs. 2 and 3. With spurious reflections present, instead of equation (1) it is more suitable to consider the function

$$N_c^{-1}\sum_{n=1}^{N} w_n \left| \mathbf{h}_{\varpi(n)} - O\mathbf{g}_n \right|^2, \qquad (2)$$

where the binary weight $w_n = 1$ if $\mathbf{g}_n$ is legitimate, $w_n = 0$ otherwise, and the number of non-zero terms $N_c = \sum_{n=1}^{N} w_n$ is not smaller than an assumed limit. If $\mathbf{g}_n$ is spurious, the vector $O\mathbf{g}_n$ at the true orientation is likely to be distant from all vectors $\mathbf{h}_m$, *i.e.* the nth term is likely to be large. It is therefore likely to be eliminated because the value of function (2) is reduced most by setting to zero the weights of the largest terms.

The simplest approach to this complication is to eliminate spurious scattering vectors by considering subsets of the complete set of these vectors, and to use some additional criterion of accepting a solution for a given subset. Allowing for spurious vectors is an integral part of an eminent issue of computational geometry known as the largest common point set (LCPS) problem: given two point sets $\mathcal{G}$ and $\mathcal{H}$ and a positive number ϵ, determine the largest subfigure $\mathcal{G}'$ of $\mathcal{G}$ such that there is an isometry carrying $\mathcal{G}$ to a position in which each point of $\mathcal{G}'$ is not further than ϵ from a point of $\mathcal{H}$ (*e.g.* Ambühl *et al.*, 2000). The subfigure of $\mathcal{H}$ nearly congruent to $\mathcal{G}'$ will be denoted by $\mathcal{H}'$. With the isometry being a proper rotation and the elements of $\mathcal{G}$ and $\mathcal{H}$ identified with, respectively, endpoints of $\mathbf{g}_n$ and $\mathbf{h}_m$ vectors bound to the fixed center of rotation, the indexing problem can be seen as a simplified version of the algorithmic LCPS problem.

The problem of orientation determination in the presence of spurious reflections is close to the practical problem of registering two clouds of points studied in the field of computer vision. Algorithms used for object recognition and for aligning 3D shapes are not necessarily directly applicable to indexing, but numerous approaches are considered in that field and ideas used there can be inspiring. Again, in general, these methods involve translation, whereas in orientation determination the translation term is fixed to zero.

Important for indexing is the case where the magnitudes of the scattering vectors are ignored and the magnitudes of all involved vectors are set to 1. The corresponding restricted case of LCPS is referred to as the constellation problem (Cardoze & Schulman, 1998): given two sets $\mathcal{G}$ and $\mathcal{H}$ of points on the unit sphere, determine a rotation carrying the largest subset of $\mathcal{G}$ to a position approximating a subset of $\mathcal{H}$. The constellation problem is also portrayed as the star identification problem which arises in the determination of satellite orientation (attitude) (Zhang, 2017). A graphic illustration of the constellation problem is given in Fig. 4.

The matching problems in particular fields have specific features, and there are similarities and some differences. The most important characteristic aspect of the crystal orientation

[4] Formally, ε_H is the one-sided Hausdorff distance from transformed $\mathcal{G}$ to $\mathcal{H}$.

determination problem is that, unlike most other matching problems, it is affected by crystal symmetry, *i.e.* the figure $\mathcal{H}$ is symmetric. Other differences depend on the compared cases. Compare, for instance, the indexing of diffraction patterns with star identification. The solid angle (field of view) covered by a star tracker is small and, in consequence, the angles between star-indicating vectors are small. Moreover, the number of false or undetected stars is relatively small and the identification can be based on nearest-neighbor separation. This is not true in indexing: the angles between scattering vectors in $\mathcal{G}$ can be arbitrarily large, the fraction of false reflections can be large, and many reflections can be undetected. In star identification, often the scale of the star patterns is to be determined (Groth, 1986); in indexing, there is no need to scale the patterns.

4. Accumulation

As was noted above, to deal with spurious scattering vectors, one needs to match test subfigures of $\mathcal{G}$ and $\mathcal{H}$ and specify criteria for accepting a match as the solution. In principle, one may use arbitrary k-tuples from $\mathcal{G}$ and $\mathcal{H}$ as matching elements. However, since the fraction of spurious scattering vectors in $\mathcal{G}$ can be large, the test subfigures must be small. In practice, the matching algorithms are based on pairs (*e.g.* Wenk *et al.*, 1997; Chung & Ice, 1999) triplets (*e.g.* Tamura *et al.*, 2003; Tamura, 2014) or quadruplets (*e.g.* Mortari *et al.*, 2004; Heyl, 2013) of vectors.

The most reliable matching algorithms rely on accumulation. Matched elements, *i.e.* pairs or triplets, cast votes (Ohba *et al.*, 1981; Groth, 1986), *i.e.* they contribute to an accumulation space, and the final solution is based on the results of the voting. The advantage of accumulation-based methods lies in their robustness and conceptual simplicity. The algorithms are relatively easy to implement. Accumulation can be made in the discrete space of assignments or in the continuous space of crystal orientations, and these two cases are considered in Sections 5 and 6, respectively.

4.1. Rotation relating two *k*-tuplets of vectors

With $k \geq 3$, the rotation transforming a k-tuple $(\hat{\mathbf{g}}_{n_1}, \ldots, \hat{\mathbf{g}}_{n_k})$ to a position closest to that of $(\hat{\mathbf{h}}_{m_1}, \ldots, \hat{\mathbf{h}}_{m_k})$ is given by the formula for absolute orientation,

$$O^{(k)} = \mathcal{SO}\Big(\big[\hat{\mathbf{h}}_{m_1}\ \hat{\mathbf{h}}_{m_2}\ \ldots\ \hat{\mathbf{h}}_{m_k}\big] \big[\hat{\mathbf{g}}_{n_1}\ \hat{\mathbf{g}}_{n_2}\ \ldots\ \hat{\mathbf{g}}_{n_k}\big]^{\mathrm{T}} \Big). \quad (3)$$

Clearly, the above expression and its interpretation are valid if $\hat{\mathbf{h}}_{m_i}$ and $\hat{\mathbf{g}}_{n_i}$ are replaced by $\mathbf{h}_{m_i}$ and $\mathbf{g}_{n_i}$, respectively, but from now on the focus will be on normalized vectors, because the strategies described below rely on such vectors. Equation (3) cannot be directly extended to get a rotation matching two 2-tuples. An easy way of circumventing this problem is to use equation (3) for $k = 3$ with the third linearly independent vectors being the cross products $\mathbf{g}_\times = \hat{\mathbf{g}}_{n_1} \times \hat{\mathbf{g}}_{n_2}$ and $\mathbf{h}_\times = \hat{\mathbf{h}}_{m_1} \times \hat{\mathbf{h}}_{m_2}$; one has

$$\begin{aligned} O^{(2)} &= \mathcal{SO}\Big(\big[\hat{\mathbf{h}}_{m_1}\ \hat{\mathbf{h}}_{m_2}\ \mathbf{h}_\times\big] \big[\hat{\mathbf{g}}_{n_1}\ \hat{\mathbf{g}}_{n_2}\ \mathbf{g}_\times\big]^{\mathrm{T}} \Big) \\ &= \big[\hat{\mathbf{r}}_1^h\ \hat{\mathbf{r}}_2^h\ \hat{\mathbf{r}}_3^h\big] \big[\hat{\mathbf{r}}_1^g\ \hat{\mathbf{r}}_2^g\ \hat{\mathbf{r}}_3^g\big]^{\mathrm{T}}, \end{aligned} \quad (4)$$

where $\hat{\mathbf{r}}_1^g$, $\hat{\mathbf{r}}_2^g$ and $\hat{\mathbf{r}}_3^g$ represent normalized versions of $\mathbf{g}_\times$, $\hat{\mathbf{g}}_{n_1} + \hat{\mathbf{g}}_{n_2}$ and $\hat{\mathbf{g}}_{n_1} - \hat{\mathbf{g}}_{n_2}$, respectively, and the $\hat{\mathbf{r}}_i^h$ vectors are defined in an analogous way based on $\hat{\mathbf{h}}_{m_i}$ vectors (Morawiec, 2015). The essential point is that $O^{(2)}$ given by equation (4) transforms the pair $\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2}$ to the position closest to that of $\hat{\mathbf{h}}_{m_1}$, $\hat{\mathbf{h}}_{m_2}$. With the last part of equation (4), the computation of this optimal rotation matrix involves only elementary arithmetic operations, *i.e.* there is no need to apply the operator $\mathcal{SO}$.

The case of $k = 2$, *i.e.* pairwise matching, is particularly convenient. It is in a sense elemental among those providing discrete solutions. In matching based on k larger than two, additional vectors sharpen the criteria for the congruence of given subsets. On the other hand, using higher-order k-tuples increases the computational costs. To avoid combinatorial explosion, k needs to be kept small. Moreover, numerical experiments show that, with error-affected data, higher-order k-tuples do not really improve the quality of indexing (*cf.* Fig. 10 below.)

Critical for the temporal performance of the above algorithms is the number of k-tuples. In practice, the number of actual matchings is reduced by considering only k-tuples with certain characteristics. One needs to use features allowing for the elimination of as many redundant k-tuples as possible. Most of the matchings can be omitted by observing that the vectors $\hat{\mathbf{g}}_{n_i}$ can match $\hat{\mathbf{h}}_{m_i}$ only if the angles between the vectors of the k-tuples differ by less than a given threshold; if not, the k-tuples are rejected as a possible match. This is particularly simple when $k = 2$: the vectors $\hat{\mathbf{g}}_{n_1}$ and $\hat{\mathbf{g}}_{n_2}$ can match $\hat{\mathbf{h}}_{m_1}$ and $\hat{\mathbf{h}}_{m_2}$ only if the angle between $\hat{\mathbf{g}}_{n_1}$ and $\hat{\mathbf{g}}_{n_2}$ is close to that between $\hat{\mathbf{h}}_{m_1}$ and $\hat{\mathbf{h}}_{m_2}$. Additionally, due to crystal symmetry, some k-tuples of reciprocal-lattice vectors are equivalent, and only one representative of these k-tuples needs to be considered. In the simplest approach, it is enough to use only one vector of a family of symmetrically equivalent reciprocal-lattice vectors as the first vector $\hat{\mathbf{h}}_{m_1}$ of a k-tuple.

5. Accumulation in discrete space

Of all possible assignments, one looks for the one with the largest support from matching of k-tuples. Briefly, pairs of k-tuples vote for assignments, and the one with the largest number of votes is assumed to solve the problem. The quality of the solutions is reflected in the number of matched vectors (*i.e.* the cardinality of the sets $\mathcal{G}'$ and $\mathcal{H}'$). Knowing the assignment P, the resulting orientation is $O = \mathcal{SO}(HPG'^{\mathrm{T}})$, where G' is the matrix with vectors of $\mathcal{G}'$. Orientation determination by accumulation in a discrete space is similar to geometric hashing – a voting-based method used for object recognition (Wolfson & Rigoutsos, 1997). To give an example, a strategy (A) based on pair-matching and accumulation in a discrete space is described in detail. Two other strategies of this kind (B and C) are listed in Appendix *B*. As was noted above, the strategies are limited to the case of normalized $\hat{\mathbf{h}}_m$ and $\hat{\mathbf{g}}_n$ vectors.

5.1. Example strategy (A)

The preliminary step is to create a list (hereafter referred to as `LIST`) of angular distances between $\hat{\mathbf{h}}_m$ vectors; with the angle between $\hat{\mathbf{h}}_{m_1}$ and $\hat{\mathbf{h}}_{m_2}$ being $d_h = d(\hat{\mathbf{h}}_{m_1} \cdot \hat{\mathbf{h}}_{m_2}) = \arccos(\hat{\mathbf{h}}_{m_1} \cdot \hat{\mathbf{h}}_{m_2})$, the entry added to the list has the form of the quintuplet $(d_h, f_1, k_1, f_2, k_2)$, where f_i indicates the family (*i.e.* a set corresponding to certain $\{hkl\}$) of symmetrically equivalent vectors to which $\mathbf{h}_{m_i}$ belongs, and k_i indicates the particular vector of the family. Clearly, each pair (f_i, k_i) identifies a reciprocal-lattice vector. The list is shortened thanks to crystal symmetry; for symmetrically equivalent pairs separated by the same distance, only one representative quintuplet $(d_h, f_1, k_1, f_2, k_2)$ needs to be saved.[5] The list is sorted by the distances d_h. With this, determining the items matching the angular distance between a pair of $\hat{\mathbf{g}}_n$ vectors is fast. `LIST` is created once for all patterns of a given phase.

For a given pattern, an accumulation (vote count) table $V(n, f)$ will be needed. It is indexed by n enumerating the scattering vectors $\hat{\mathbf{g}}_n$ obtained from the pattern, and by f (= 1, . . . , M_f) enumerating the families of equivalent reciprocal-lattice vectors. The table V is initialized with zero entries.

The indexing of the pattern begins with accumulation loops. For each pair $(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$ such that $n_1 < n_2$ get the angular distance $d(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$. From the `LIST` of quintuplets $(d_h, f_1, k_1, f_2, k_2)$ retrieve all items with d_h deviating from $d(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$ by less than a given threshold. This threshold, say p_1, is a parameter of strategy A. The families f_i which appear in the retrieved items constitute a set S (without duplicates). Each element of the set casts votes for both $\hat{\mathbf{g}}_{n_1}$ and $\hat{\mathbf{g}}_{n_2}$, *i.e.* for each f_i in S the values of $V(n_1, f_i)$ and $V(n_2, f_i)$ are increased by 1.

With the accumulation completed, the larger the value of $V(n, f)$ for a given n, the larger the probability that $\hat{\mathbf{g}}_n$ matches a vector from the fth family. At this point, one has the families (or indices $\{hkl\}$) expected to correspond to each scattering vector, whereas the task is to get the concrete vector [or indices (hkl)] modulo the Laue symmetry of the crystal. A simple approach is to assume that $\hat{\mathbf{g}}_n$ matches a vector of the single family $\arg\max_f V(n, f)$, but clearly one can ascribe two or more families with the highest number of votes. The number p_d of ascribed families is another parameter of strategy A.[6] In the particular implementation described below, this parameter is fixed at $p_d = \max(\lfloor M_f/2 \rfloor, 1)$.

```
A Input: lattice parameters, $M_f$ sets of Miller indices of reflectors, symmetry
         operations, parameters $p_i$ $(i = 1, 2, d)$
/* Initialization                                                              */
get $M$ reciprocal lattice vectors $\boldsymbol{h}_m$ grouped in $M_f$ families
normalization $\hat{\boldsymbol{h}}_m = \boldsymbol{h}_m / |\boldsymbol{h}_m|$
create LIST
/* Processing of patterns                                                      */
for each diffraction pattern do
    Input: $N$ normalized scattering vectors $\hat{\boldsymbol{g}}_n$
    $V = 0$
    /* Accumulation                                                            */
    for each pair $(\hat{\boldsymbol{g}}_{n_1}, \hat{\boldsymbol{g}}_{n_2})$ such that $n_1 < n_2$ do
        $d(\hat{\boldsymbol{g}}_{n_1}, \hat{\boldsymbol{g}}_{n_2}) = \arccos(\hat{\boldsymbol{g}}_{n_1} \cdot \hat{\boldsymbol{g}}_{n_2})$
        retrieve from LIST quintuplets $(d_h, f_1, k_1, f_2, k_2)$ such that
        $|d(\hat{\boldsymbol{g}}_{n_1}, \hat{\boldsymbol{g}}_{n_2}) - d_h| \leq p_1$
        $S = \bigcup$ (indices $f$ of families present in the retrieved quintuplets)
        for each $f$ in $S$ and $i = 1, 2$ do
            $V(n_i, f) \longleftarrow V(n_i, f) + 1$
    /* Only $p_d$ $(\geq 1)$ high-vote families are ascribed to $\hat{\boldsymbol{g}}_n$            */
    for each $n$ do
        Set the $M_f - p_d$ lowest values of $V(n, \cdot)$ to 0
    /* Final stage                                                             */
    $HighestNumberOfAssignedVects \longleftarrow 0$
    for each pair $(\hat{\boldsymbol{g}}_{n_1}, \hat{\boldsymbol{g}}_{n_2})$ such that $n_1 < n_2$ do
        Retrieve vectors $\hat{\boldsymbol{h}}_{m_i}$ $(i = 1, 2)$ of the families $f$ for which $V(n_i, f) \neq 0$
        /* Getting trial orientation                                           */
        for each pair $(\hat{\boldsymbol{h}}_{m_1}, \hat{\boldsymbol{h}}_{m_2})$ of the retrieved vectors do
            $O^{(2)} \longleftarrow$ result of eq.(4) applied to $(\hat{\boldsymbol{g}}_{n_1}, \hat{\boldsymbol{g}}_{n_2})$ and $(\hat{\boldsymbol{h}}_{m_1}, \hat{\boldsymbol{h}}_{m_2})$
        /* Trial assignment                                                    */
        $NumberOfAssignedVects \longleftarrow 0$;  $H'$ and $G' \longleftarrow$ empty matrices
        for each $n$ do
            $\hat{\boldsymbol{h}}_{m_x} \longleftarrow$ vector $\hat{\boldsymbol{h}}_m$ $(m = 1, \ldots, M)$ nearest to $O^{(2)}\hat{\boldsymbol{g}}_n$
            if $d(\hat{\boldsymbol{h}}_{m_x}, O^{(2)}\hat{\boldsymbol{g}}_n) \leq p_2$ then
                append $\hat{\boldsymbol{g}}_n$ and $\hat{\boldsymbol{h}}_{m_x}$ to matrices $G'$ and $H'$, respectively
                $NumberOfAssignedVects \longleftarrow NumberOfAssignedVects + 1$
        /* Saving the result                                                   */
        if $NumberOfAssignedVects > HighestNumberOfAssignedVects$ then
            $O_{res} \longleftarrow \mathcal{SO}(H'G'^T)$ ;   $quality(O_{res}) \longleftarrow \mathrm{tr}(H'^T O_{res} G')$
            $HighestNumberOfAssignedVects \longleftarrow NumberOfAssignedVects$
        else if $NumberOfAssignedVects = HighestNumberOfAssignedVects$ then
            $O \longleftarrow \mathcal{SO}(H'G'^T)$ ;   $quality(O) \longleftarrow \mathrm{tr}(H'^T O G')$
            if $quality(O) > quality(O_{res})$ then
                $O_{res} \longleftarrow O$ ;   $quality(O_{res}) \longleftarrow quality(O)$
    Result: Orientation $O_{res}$ and $quality(O_{res})$
```

Figure 5
Algorithm A: a simplified description of strategy A for computing crystal orientation from normalized scattering vectors $\hat{\mathbf{g}}_n$.

In the final stage, one needs to discriminate between members of families of equivalent vectors. This stage can be performed in various ways. One could look for a consistent solution by considering only angular distances, but more convenient is a method involving orientation: for every pair $(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$ satisfying $n_1 < n_2$, construct all pairs of vectors $(\hat{\mathbf{h}}_{m_1}, \hat{\mathbf{h}}_{m_2})$ such that only the vectors $\hat{\mathbf{h}}_{m_i}$ from the family (familes) ascribed to $\hat{\mathbf{g}}_{n_i}$ in the previous step are allowed to correspond to $\hat{\mathbf{g}}_{n_i}$. Based on $(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$ and $(\hat{\mathbf{h}}_{m_1}, \hat{\mathbf{h}}_{m_2})$, using function (2), get a trial orientation $O^{(2)}$. Having the trial orientation, one can perform a trial indexing: for each vector $O^{(2)}\mathbf{g}_n$ (n = 1, . . . , N) one finds the closest of unmatched vectors $\hat{\mathbf{h}}_m$. If the distance between $O^{(2)}\mathbf{g}_n$ and the closest vector, say $\hat{\mathbf{h}}_{m_x}$, does not exceed a given threshold (parameter p_2 of strategy A), this closest vector is ascribed to $\mathbf{g}_n$, *i.e.* $\varpi(n) = m_x$; otherwise, $\mathbf{g}_n$ is considered to be spurious. The problem is solved by the assignment ϖ leading to the smallest number of spurious $\mathbf{g}_n$ vectors. At the end, one needs to apply $\mathcal{SO}$ to

[5] The above issue is simple but a word of caution is in order. Matrices $O^{(2)}$ given by equation (4) represent proper rotations. This affects the allowed relationships of equivalence between pairs of $\hat{\mathbf{h}}_m$ vectors. Take the $m\bar{3}m$ crystal symmetry. The angular distance between the vectors $\mathbf{h}_1 = (101)$ and $\mathbf{h}_2 = (221)$ is $\pi/4$. The same distances separate the vectors $\mathbf{h}_1' = (011)$ and $\mathbf{h}_2' = (212)$, and the vectors $\mathbf{h}_1'' = (011)$ and $\mathbf{h}_2'' = (221)$, but of these two only the pair $(\mathbf{h}_1', \mathbf{h}_2')$ is related to $(\mathbf{h}_1, \mathbf{h}_2)$ by a symmetry operation being a proper rotation; the same concerns the pair $(-\mathbf{h}_1'', -\mathbf{h}_2'')$ but not the pair $(\mathbf{h}_1'', \mathbf{h}_2'')$.

[6] In this step, $M_f - p_d$ families are rejected as potential ascriptions to $\hat{\mathbf{g}}_n$. One may build a more sophisticated rejection algorithm, in which the accumulation stage is repeated, but this time only the pairs consisting of vectors from earlier-accepted families are allowed to cast votes. The results of the second voting round are saved in V. There is an analogy between this second voting and an extension to the Hough transform known as back-mapping (Gerig & Klein, 1986). See also Kolomenkin *et al.* (2008).

legitimate $\mathbf{g}_n$ vectors and their partners $\hat{\mathbf{h}}_{\varpi(n)}$ to get the orientation O. In the case of a draw (*i.e.* the same number of spurious vectors for different orientations), the result with the smallest residue is used as the final solution. The residue can be defined as *e.g.* $\sum_n w_n |\hat{\mathbf{h}}_{\varpi(n)} - O\hat{\mathbf{g}}_n|^2$ or $\sum_n w_n \arccos(\hat{\mathbf{h}}_{\varpi(n)} \cdot O\hat{\mathbf{g}}_n)$; like in function (2), $w_n = 1$ if $\mathbf{g}_n$ is legitimate, and $w_n = 0$ if $\mathbf{g}_n$ is spurious. A simplified description of strategy A is given in Fig. 5.

5.2. Triplet voting

A natural extension of pair-based voting is voting by triplets. Such a method was applied in a program for indexing of Laue patterns (Ohba *et al.*, 1981). There is a family of algorithms used for autonomous star recognition based on triplets of stars (*e.g.* Groth, 1986); they are also known as the 'triangle algorithms' (Zhang, 2017). Triplet voting is used for indexing EBSD patterns in the commercial OIM Analysis system (Wright & Adams, 1992; Adams *et al.*, 1993).

Triplet-based accumulation can be implemented by simply modifying a pair-based method. To cast votes, instead of pairs, vectors $\hat{\mathbf{g}}_n$ must constitute triplets nearly congruent to triplets of $\hat{\mathbf{h}}_m$ vectors. Other aspects of indexing are, *mutatis mutandis*, the same as in pairwise voting. For illustration, such a modification of strategy A is described in Appendix *B* as strategy D.

5.3. Example implementation

The algorithms A–D have been implemented in a (Fortran90) program called *KiKoCh2*. It stems from *KiKoCh* (Morawiec *et al.*, 2002) which was designed for indexing of Kikuchi patterns, but besides the name little of this predecessor is left in *KiKoCh2*. The applicability of the procedures and of the program for indexing of various patterns are illustrated in the figures below. *KiKoCh2* can be used to index EBSD and standard Kikuchi patterns (Fig. 6), but it also works for patterns recorded with much smaller acquisition angles. Fig. 7 shows an example result with indexing of *K* lines in the central disk of a CBED pattern in which the diameter of the disk is about 1/10 of the camera length. It is also applicable to Laue patterns originating from simple structures (Fig. 8) and to indexing of Kossel patterns (Fig. 9). *KiKoCh2* is included in the package TEM*Strain* for analysis of CBED patterns (Morawiec, 2007) and in the package KSL*Strain* for analysis of Kossel patterns (Morawiec *et al.*, 2008; Morawiec, 2016). The program is available at http://imim.pl/personal/adam.morawiec/.

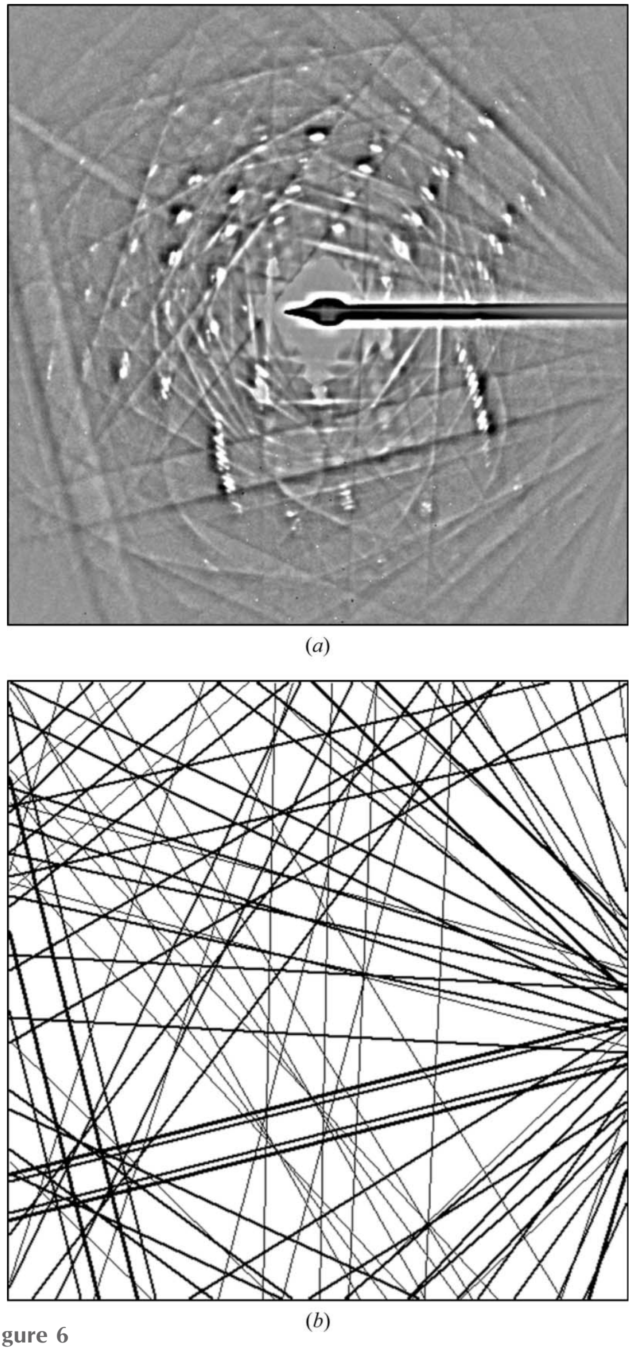


Figure 6
(*a*) Experimental Kikuchi pattern of chromium carbide Cr_3C_2 alloy (courtesy of E. Bouzy). (*b*) Simulated Kikuchi lines in orientation determined using strategy A based on marking lines manually positioned on the experimental pattern.

6. Accumulation in rotation space

Contributions can also be accumulated in the space of orientations (or more precisely, in the symmetry-induced fundamental region of the orientation space; Morawiec, 2004). Contributions can be made by individual vectors (one vector from each $\mathcal{G}$ and $\mathcal{H}$), and in this case accumulation takes place along continuous curves in the parameter space. With two or more vectors from each set, accumulation takes place at discrete points (Morawiec & Bieda, 2005).

6.1. Accumulation at points of the rotation space

For the k-tuple ($k \geq 2$) of vectors $\hat{\mathbf{g}}_{n_i}$ from $\mathcal{G}$ and the k-tuple of vectors $\hat{\mathbf{h}}_{m_i}$ from $\mathcal{H}$ ($i = 1, \ldots, k$), the special orthogonal matrix representing the best rotation of $\hat{\mathbf{g}}_{n_i}$ to $\hat{\mathbf{h}}_{m_i}$ is given by equations (3) and (4). To determine the rotation carrying the largest subset of $\mathcal{G}$ to a subset of $\mathcal{H}$, all k-tuples from $\mathcal{G}$ and from $\mathcal{H}$ are considered, and each pair of k-tuples contributes to a point in the rotation space. The amount of contribution made to the parameter space by a given pair of k-tuples can be

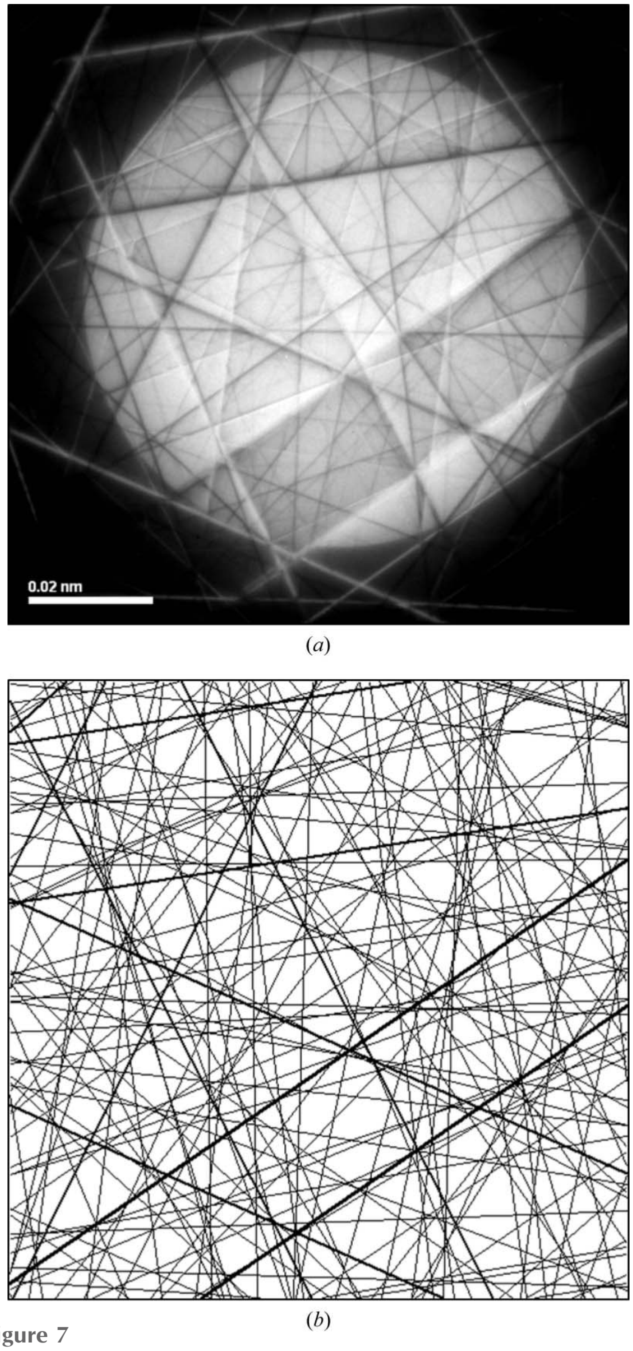


Figure 7
(*a*) The central disk of an experimental CBED pattern of a TiAl alloy (courtesy of G. Brunetti). (*b*) Simulated lines in orientation determined using strategy A based on marking lines manually positioned on the experimental pattern.

constant, but it may also be weighted by the quality of the match between the k-tuples.

Assuming exact congruence between the largest matching subsets of $\mathcal{G}$ and $\mathcal{H}$, the rotation relating them gets the largest number of contributions, and the problem of orientation determination is reduced to locating a maximum of the contributions. The same approach is applicable to experimental patterns, but a tolerance needs to be allowed. At the end, knowing the crude orientation, the matching figures $\mathcal{G}'$ and $\mathcal{H}'$ and the assignment between their elements are determined, and the operator $\mathcal{SO}$ is applied to them to get the final orientation. This method has been implemented a number of times; see *e.g.* strategy 4 for indexing Kikuchi patterns in the report by Morawiec (1999) or a strategy for indexing Laue patterns described by Kalinowski *et al.* (2011).

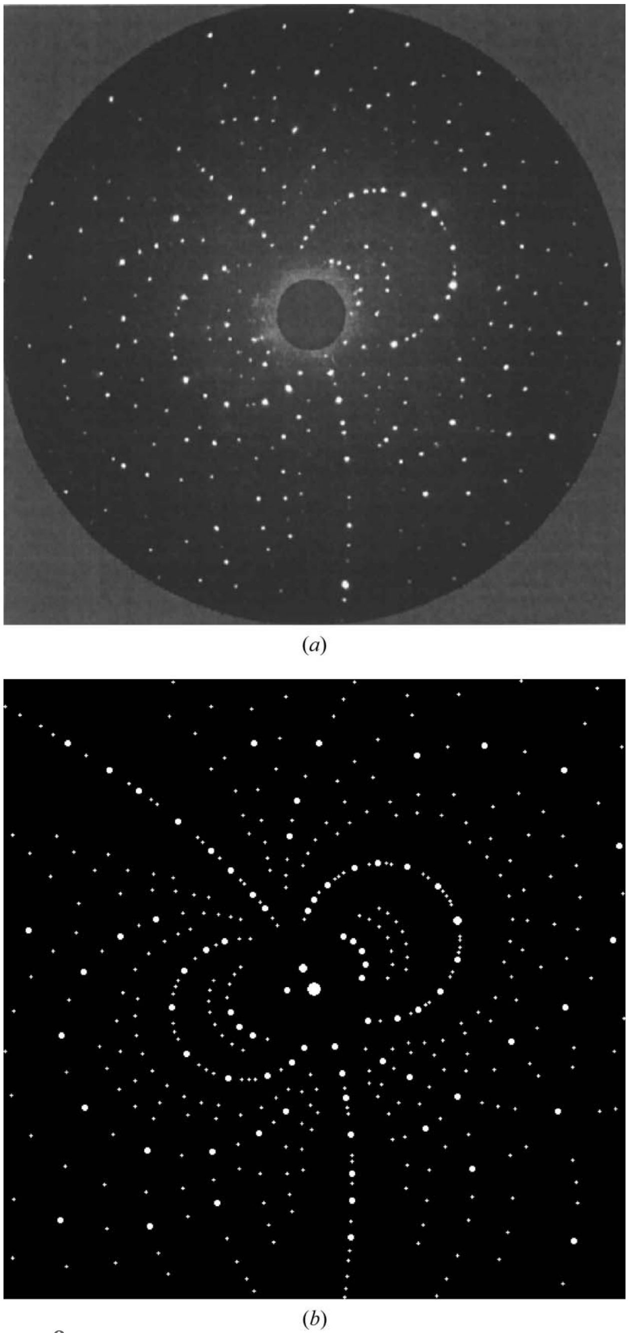
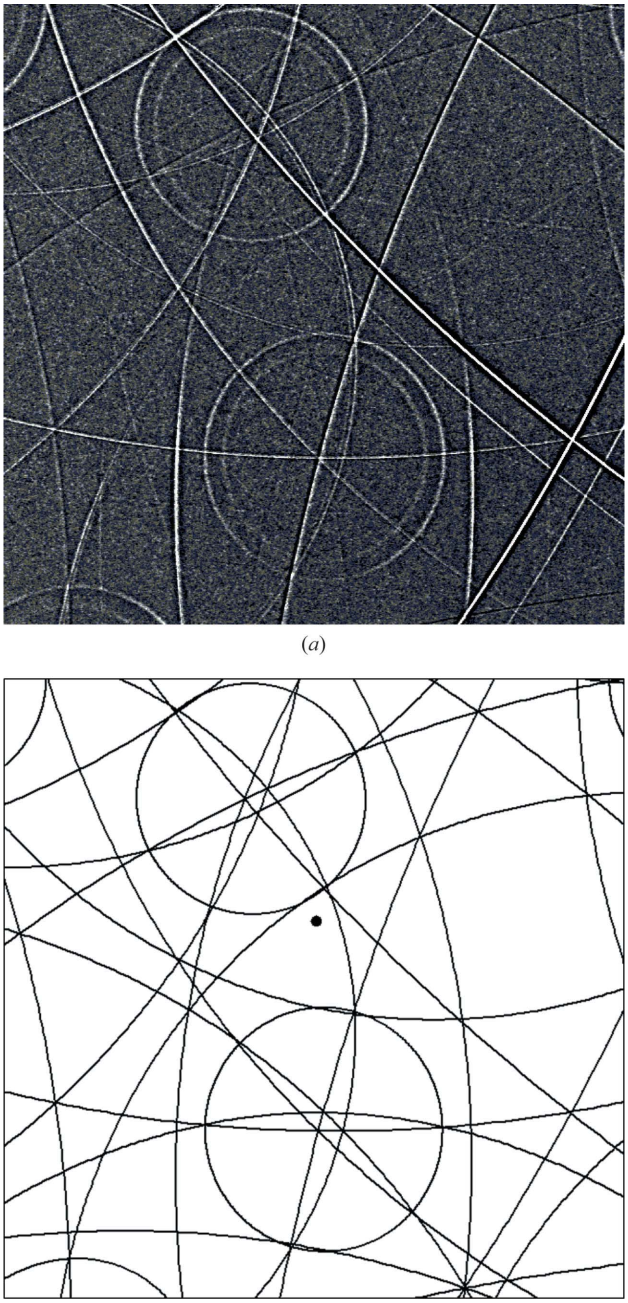


Figure 8
(*a*) An experimental Laue pattern of an organic compound published by Ravelli *et al.* (1996). Reproduced with permission of the International Union of Crystallography. (*b*) A Laue pattern simulated for orientation determined using strategy A based on spots automatically detected on the experimental pattern.

6.2. Accumulation along curves in the space of rotations

Contributions can also be accumulated based on individual vectors, *i.e.* with $k = 1$ (Morawiec & Bieda, 2005). Let $R(\mathbf{v}, \omega)$



(*a*)

(*b*)

Figure 9
(*a*) An experimental Kossel pattern of pattern of a CuBeAl alloy (Bouscaud *et al.*, 2014) (courtesy of D. Bouscaud). (*b*) Simulated Kossel $K\alpha_1$ lines in orientation determined using strategy A based on marking points manually positioned on the experimental pattern.

be the special orthogonal matrix of the rotation by the angle ω about the axis determined by $\mathbf{v}$. Given two vectors $\hat{\mathbf{g}}_n$ and $\hat{\mathbf{h}}_m$ in $\mathcal{G}$ and $\mathcal{H}$, respectively, one looks for all rotations transforming $\hat{\mathbf{g}}_n$ on $\hat{\mathbf{h}}_m$. Assuming that $\hat{\mathbf{g}}_n \neq -\hat{\mathbf{h}}_m$, these rotations are represented by the matrices

$$O^{(1)}(\omega) = R(\hat{\mathbf{g}}_n + \hat{\mathbf{h}}_m, \pi)\, R(\hat{\mathbf{g}}_n, \omega). \qquad (5)$$

They constitute an ω-parameterized geodesic in the rotation space. Each pair $(\hat{\mathbf{g}}_n, \hat{\mathbf{h}}_m)$ contributes to rotations located on one of such curves. In practice, the contributions are made along the curves at some small intervals of ω.

As in the case of $k \geq 2$, the rotation relating the largest matching subsets of $\mathcal{G}$ and $\mathcal{H}$ gets the largest number of contributions, and again the problem is reduced to locating the global maximum of the contributions in the accumulator space. The maximum corresponds to the parameters of the sought rotation. This approach to orientation determination and indexing is a form of the generalized Hough transform. An implementation of a variant of this method is described by Schmidt (2014), and see also Beyerlein *et al.* (2017) and Gevorkov *et al.* (2020).

6.3. Maxima in rotation space

As in all accumulation-based methods, at the end one needs to determine the locations of maxima in the accumulator space, *i.e.* in the symmetry-induced fundamental region in the space of proper rotations. Since the match between vectors is only approximate, some tolerance must be allowed. A straightforward way of evaluating the contributions is by partitioning the space into equivolume bins of a size linked to the precision of the experimental data and the resolution of the resulting orientations. The center of the bin with the largest accumulation is the sought rotation matching the largest subsets of $\mathcal{G}$ and $\mathcal{H}$. With this approach, however, extra measures are needed to take account of contributions distributed in neighboring bins or near borders of the fundamental region.

Instead of binning, methods of cluster analysis can be implemented. One approach (particularly suitable for resolving the discrete cases with $k \geq 2$) is to use a list of potential solutions; every new orientation obtained from equations (3) or (4) is compared with already saved potential solutions. If the orientation deviates from an earlier-saved solution by an angle smaller than a given threshold, the ranking number of that solution is increased, and the solution is corrected by taking the weighted average (Morawiec, 1998) of the solution and the new orientation. Otherwise, the orientation is appended to the list as a new potential solution.

Contributions can also be made by adding continuous orientation components, *e.g.* represented by coefficients of an expansion of a peak into (symmetrized) generalized spherical harmonics. In this case, one needs a method of searching for global extrema of functions on the rotation space given by analytical expressions. One way is to search for local extrema using, say, one of the gradient methods, starting from points on a nearly uniform grid (Roşca *et al.*, 2014; Larsen & Schmidt, 2017; Quey *et al.*, 2018). The highest of these local maxima is considered to be the global maximum. Clearly, the grid density must correspond to the width of the peaks of the function.

6.4. Other orientation-based algorithms

For completeness, one needs to the mention orientation-based methods used in computer vision for 3D registration. The most popular iterative closest point (ICP) algorithm (Besl & McKay, 1992) has "difficulty correctly registering 'sea urchins' and 'planets"', and these are the shapes of $\mathcal{G}$ and $\mathcal{H}$ figures in the indexing problem.[7] There are, however, extensions of ICP such that each point of the 'model' $\mathcal{H}$ interacts with all 'data' points of $\mathcal{G}$ (Tsin & Kanade, 2004). For example, in the 'softassign matching' algorithm of Rangarajan *et al.* (1997), the mixed (combinatorial + continuous) problem is replaced by a purely continuous nonlinear problem; the matrix representing the combinatorial part of the problem is allowed to have continuous entries, and its evolution in the optimization process is guided by additional constraints and barrier functions.

A simple way of turning the registration into minimization of a continuous function is known as the kernel correlation method (Tsin & Kanade, 2004). Briefly, in the case without translation, the problem of orientation determination is reduced to finding the proper rotation O minimizing

$$f(O) = -\sum_{m=1}^{M}\sum_{n=1}^{N}\exp\left(\frac{-|\mathbf{h}_m - O\mathbf{g}_n|^2}{2\sigma^2}\right). \qquad (6)$$

With unit vectors $\hat{\mathbf{h}}_m$ and $\hat{\mathbf{g}}_n$, this function takes the form $-\sum_m\sum_n\exp(\hat{\mathbf{h}}_m\cdot O\hat{\mathbf{g}}_n/\sigma^2)$, *i.e.* it is a sum of von Mises–Fisher functions (Morawiec, 2004). The above expressions are based on the Gaussian kernel, but clearly other kernels can be used.

It is worth noting that, similar to equation (6), one could get the orientation by locating the global minimum of the function

$$f(O) \propto N_c^{-1}\sum_{n=1}^{N} w_n(O)\min_m|\mathbf{h}_m - O\mathbf{g}_n|^2,$$

where $w_n(O) = 1$ if $\min_m|\mathbf{h}_m - O\mathbf{g}_n| < \epsilon$ and $w_n(O) = 0$ otherwise, and $N_c = \sum_{n=1}^{N} w_n(O)$ is the number of non-zero terms. With unit vectors $\hat{\mathbf{h}}_m$ and $\hat{\mathbf{g}}_n$, the function can be expressed as

$$f(O) \propto N_c^{-1}\sum_{n=1}^{N} w_n(O)\max_m \hat{\mathbf{h}}_m\cdot O\hat{\mathbf{g}}_n,$$

where $w_n(O) = 1$ if $\max_m \hat{\mathbf{h}}_m\cdot O\hat{\mathbf{g}}_n > 1-\epsilon$ and $w_n(O) = 0$ otherwise. This is in a sense a trial-and-error approach accounting for all orientations (f as a function of O) and all assignments (search for $\min_m|\mathbf{h}_m - O\mathbf{g}_n|^2$ or $\max_m \hat{\mathbf{h}}_m\cdot O\hat{\mathbf{g}}_n$ for each $O\hat{\mathbf{g}}_n$). Strategy B and the final stage of strategy A are based on the same principle, but instead of considering all orientations, only those matching pairs of vectors are tested. Clearly, with this and similar approaches, an additional computational device promoting solutions with large N_c is needed.

7. Testing of indexing algorithms

The standard method of checking the quality of indexing algorithms is to generate artificial data based on known initial orientations, add errors, run the program, and check the deviations between the resulting orientations and the true initial orientations.

Both the indexing success rate and speed depend on the types of patterns and the lengths of the lists of matched vectors. Example results (success rate *versus* random errors) of tests on simulated EBSD bands for α-Ti are illustrated in Fig. 10. The figure shows the performance of algorithms A, B, C, D and a triplet-based accumulation in orientation space (Morawiec, 1999). For these particular data, the algorithms A–D outperform the accumulation in the orientation space in terms of robustness to large random errors. The speeds of indexing with strategies A–D were similar, with the fastest (A) processing more than 15 000 patterns per second and the slowest (C) nearly 10 000 patterns per second. (CPU times on a 2.6 GHz personal computer, computation without any form of parallelism.) Indexing by accumulation in the orientation space was much slower; the speed was about 200 patterns per second.

Indexing algorithms usually involve some parameters, and particular values of these parameters are to be fixed for given diffraction data. The numbers will depend on the type of pattern (*e.g.* Kossel *versus* Laue), the settings of the pattern acquisition system (*e.g.* sample-to-detector distance), the crystal structure, the level of errors affecting the scattering vectors *etc.* The optimal values of these parameters can be

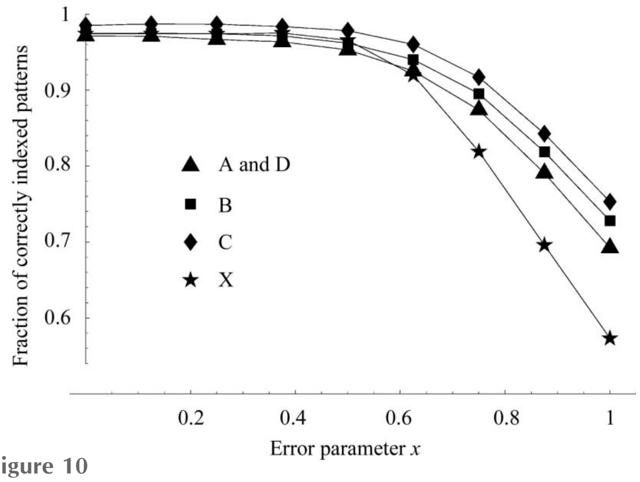


Figure 10
Example results of tests on simulated EBSD bands for α-Ti. The success rate (fraction of correctly indexed patterns) is plotted versus random errors added to band positions. The number of bands per pattern was seven, of which three were spurious. The level of errors is quantified by parameter x. The error range for the distance of a band from the pattern center (band shift) was $x \times 8\%$ of the pattern diameter. The error range for band inclination (band slope) was $x \times 4^\circ$. The errors were uniformly distributed in their ranges. Indexing of an individual pattern was considered correct if the obtained orientation deviated from the true one by less than 5°. The results for strategies A and D are not the same but are visually indistinguishable. The data X marked by stars were obtained using an algorithm based on accumulation in the orientation space (Morawiec, 1999).

[7] The algorithm is iterative: first, it finds points of the model $\mathcal{H}$ closest to points of $\mathcal{G}$, *i.e.* one gets a certain trial assignment ϖ, and second, with this assignment, the standard registration is applied (minimization of squared distances between corresponding points, and in our case that would be the $\mathcal{SO}$ operation). It is easy to see that the above procedure leads to a local minimum. To find a global minimum, the algorithm is run multiple times for a number of appropriate initial states, but it is still susceptible to local minima.

determined by checking the quality of indexing on simulated data having characteristics similar to those of the experimental data. Clearly, with a properly defined goal function, this can be done in an automatic way using a minimization program. In the case of the data shown in Fig. 10, the parameters of strategies A–D were the same; the success rate could be improved by tuning the parameters to the strategy and error level.

In orientation mapping, the quality of the indexing can be evaluated based on real experimental maps. The better the routine, the smaller the number of unsolved cases and misindexing ‘spikes’ on the maps, and the smaller the average misorientation between neighboring pixels within individual grains. To estimate the reliability of indexing a single pattern, one needs a figure of merit.

8. Figures of merit

After indexing an individual experimental pattern, one ultimately has the crystal orientation and the assignment between some vectors of $\mathcal{G}$ and $\mathcal{H}$. With this, one can specify nearly congruent subfigures $\mathcal{G}'$ and $\mathcal{H}'$ of $\mathcal{G}$ and $\mathcal{H}$, respectively. There is a question about the quality of this congruency and the reliability of the solution. A number of indicators can be used and combined to get a final figure of merit. The simplest measures of quality are the number of detected reflections which were actually indexed (*i.e.* the cardinality of the sets $\mathcal{G}'$ and $\mathcal{H}'$), the ratio of the number of indexed reflections to the number of detected reflections (*i.e.* the ratio of the cardinalities of the sets $\mathcal{G}'$ and $\mathcal{G}$) and the quality of the match between $\mathcal{G}'$ and $\mathcal{H}'$ [*e.g.* the actual value of the expression in function (2)]. One may also use indicators specific to the used method, *e.g.* the number of votes in the accumulation space.

The other issue is how the best solution compares with other potential solutions, in particular how the best solution compares with the second-best solution. Quantitative measures of the dominance of the best solution can be constructed from the factors listed in the previous paragraph. A good example of such a parameter is the confidence index, defined as $(v_1 - v_2)(v_1 + v_2)^{-1}$, where v_i is the number of votes for the ith best solution (Field, 1997). Clearly, the confidence index is equally suitable in all accumulation-based algorithms.

Figures of merit can be shown on corresponding maps, *e.g.* on confidence index maps. They are sometimes correlated to physical aspects of diffraction from differently oriented crystallites, and the maps visualize the distribution of these properties. In particular, the figures of merit are indispensable for phase discrimination.

9. Final remarks

Indexing of diffraction patterns originating from crystals of known structures is a key element of most systems for crystal orientation determination. Such indexing can be reduced to matching experimental scattering vectors to vectors of the crystal reciprocal lattice. The paper refers to formally similar problems (of registering endpoints of vectors) which arise in other research fields and to methods of solving them. The indexing is considered from a relatively broad perspective without limitation to a particular type of pattern. The paper is focused on indexing methods based on the accumulation of votes in a parameter space. The latter can be continuous (a fundamental region in the orientation space) or discrete (the number of matching vectors). Besides discussing the general principles of indexing, particular algorithms are described in detail.

For simplicity, parts of the text above and the algorithms concern the constellation problem with all involved vectors having the same magnitude, but it is worth noting that the described procedures can be relatively easily generalized to matching vectors with arbitrary magnitudes. One way to account for vector magnitudes is by using weighted contributions: the larger the difference between magnitudes of potentially matching vectors the smaller the contribution. It is also sensible to ban contributions when the difference is larger than a given threshold.

The accumulation-based indexing methods differ by their parameter space, nature of contributions (*e.g.* pairs *versus* triplets of vectors), and ways of collecting and counting the contributions. Depending on demands, one may consider using various combinations of these methods. In particular, to improve the reliability of indexing, one may apply multiple contributions of different types and multiple ways of counting the contributions. Moreover, whenever the accumulation can be seen as the generalized Hough transform, its reliability can be improved by adapting known enhancements of that transform (Leavers, 1993).

In the above considerations, it was assumed that all vectors are bound to a fixed center of rotation which corresponds to the position of the source of diffraction. This position is known only approximately, and additionally it changes during orientation mapping when the incident beam scans the specimen. With known indexing, the experimental scattering vectors can be used to refine the position of the source. In this case, the goal function depends on the pattern center, the sample-to-detector distance and the orientation parameters.

Finally, one needs to stress again that testing on simulated data is crucial for the development of orientation determination software. The data need to have error characteristics similar to those of input data from real diffraction patterns. The tests allow for fine-tuning of flexible software parameters, and they give estimates of the reliability, robustness and accuracy to be expected in real cases.

APPENDIX *A*
Direct matching

Crystal orientation can be determined without indexing by matching an experimental pattern to simulated patterns. The approach is similar to solving diffraction patterns by human inspection of an atlas of simulated patterns [see *e.g.* Krahl-Urban *et al.* (1973), Preuss *et al.* (1974) and Edington (1975)]. In large-scale mappings, experimental patterns are automatically compared with pre-computed simulated patterns. For each experimental pattern, the best match is obtained based on a suitably chosen similarity measure. An effective and standard method of matching intensity images is correlation. Computer-based 'template matching' is applicable to diffraction patterns of various types [see *e.g.* Rauch *et al.* (2008), Gupta & Agnew (2009) and Chen *et al.* (2015)]. It is particularly suitable for dealing with electron spot diffraction patterns; since their dependence on crystal orientation is relatively weak, the number of templates needed is relatively small. Note that it is not necessary to cover all possible orientations. Because of crystal symmetry, it is sufficient to use an asymmetric domain (fundamental region) of the orientation space plus a margin of a width resulting from the pattern acquisition angle.

The natural domain for a complete simulated diffraction pattern is a sphere. That is why it is natural to match patterns projected onto a sphere with spherical cross correlation as the pattern similarity measure (Hielscher *et al.*, 2019; Lenthe *et al.*, 2019). The correlation of real functions $f_{\rm s}$ (simulated intensity function) and $f_{\rm p}$ (experimental pattern) on the unit sphere S^2 is given by $(f_{\rm s} \star f_{\rm p})(O) = \int_{S^2} f_{\rm s}(\hat{\mathbf{n}}) f_{\rm p}(O\hat{\mathbf{n}})\, {\rm d}\hat{\mathbf{n}}$. The correlation reaches a maximum at the orientation corresponding to the best match of $f_{\rm s}$ and $f_{\rm p}$, and the problem is reduced to finding the maximum of $f_{\rm s} \star f_{\rm p}$. The above objective function is defined via a single pattern simulated over the entire sphere, whereas the domain of the experimental pattern $f_{\rm p}$ is limited to a window of the detector. This issue is addressed by modifying $f_{\rm p}$ so there is a smooth decay from the values inside the window to zero outside it, and by using a correction term accounting for the shape of the window (Hielscher *et al.*, 2019). The integrals can be estimated by direct numerical computation, but this approach is slow. It is more time-efficient to use expansion of the functions $f_{\rm s}$ and $f_{\rm p}$ into spherical harmonics, and to express $f_{\rm s} \star f_{\rm p}$ as a series with coefficients given by expansion coefficients of $f_{\rm s}$ and $f_{\rm p}$. See Hielscher *et al.* (2019) for details.

Clearly, knowing the crystal structure and its orientation, one can make an additional effort and ascribe Miller indices to the reflections, but this step is unnecessary when the goal is to get an orientation.

Direct matching of patterns has some advantages over orientation determination via reflection indexing. It may be preferable in the case of patterns with diffuse reflections, *e.g.* patterns originating from highly deformed samples. Moreover, direct matching accounts for intensities and this makes a difference in phase discrimination. In particular, it can discriminate between phases with the same types of Bravais lattices (*e.g.* Ram & De Graef, 2018). In principle, it allows for discriminating chiralities (Winkelmann & Nolze, 2015) or polarities (Naresh-Kumar *et al.*, 2017) of crystallites. Its main disadvantage is the high computational cost.

APPENDIX *B*
Other strategies

B1. Strategy B

At the outset, as in strategy A, the preliminary step is to create the `LIST` of quintuplets $(d_h, f_1, k_1, f_2, k_2)$.[8] Indexing strategy B is similar to the final stage of strategy A. It relies on trial orientations from the beginning. The steps are as follows.

(i) For each pair $(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$ $(n_1 < n_2)$, get the angular distance $d(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$ and retrieve from the `LIST` all items with distances d_h within a threshold p_1 from $d(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$.

(ii) For every retrieved item $(d_h, f_1, k_1, f_2, k_2)$, using equation (4), get a trial orientation $O^{(2)}$ based on the pairs $(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$ and $(\hat{\mathbf{h}}_{m_1}, \hat{\mathbf{h}}_{m_2})$, where m_i is determined by the pair (f_i, k_i).

(iii) Having the trial orientation, perform a trial indexing: for each vector $O^{(2)}\hat{\mathbf{g}}_n$ $(n = 1, \ldots, N)$, get the closest of unmatched vectors $\hat{\mathbf{h}}_m$. If the distance between $O^{(2)}\hat{\mathbf{g}}_n$ and the closest vector $\hat{\mathbf{h}}_{m_x}$ does not exceed a threshold p_2, $\hat{\mathbf{h}}_{m_x}$ is ascribed to $\mathbf{g}_n$, *i.e.* $\varpi(n) = m_x$; otherwise, $\hat{\mathbf{g}}_n$ is considered to be spurious. The problem is solved by the assignment ϖ leading to the smallest number of spurious $\mathbf{g}_n$ vectors.

(iv) At the end, one needs to apply $\mathcal{SO}$ to legitimate $\mathbf{g}_n$ vectors and their partners $\hat{\mathbf{h}}_{\varpi(n)}$ to get the orientation O. In the case of a draw (the same number of spurious vectors for different orientations), the result with the smallest residue is used as the final solution.

This strategy is suitable for early exit. If a solution reached at a certain stage of matching has a satisfactory figure of merit, further search for even better solutions can be abandoned.

B2. Strategy C

The first steps of C are similar to strategy B. One constructs the `LIST`, and one gets the admissible trial orientations $O^{(2)}$ based on pairs of vectors. As above, this involves a threshold p_1. Moreover, like in strategy A, one needs an accumulation table $V(n, f)$.

Having the trial orientation, a trial indexing is performed in the following way: for each vector $O^{(2)}\mathbf{g}_n$ $(n = 1, \ldots, N)$, get the closest of vectors $\hat{\mathbf{h}}_m$. If the distance between $O^{(2)}\mathbf{g}_n$ and this closest vector, say $\hat{\mathbf{h}}_{m_x}$, does not exceed a threshold p_2, $\mathbf{g}_n$ is considered legitimate and $\varpi(n) = m_x$. Let N_L denote the number of legitimate vectors obtained for this trial orientation. If $N_L \geq 3$, there is the accumulation step

$$V(n, f) \leftarrow V(n, f) + N_L^{p_3},$$

where f represents the family of the vector $\hat{\mathbf{h}}_{\varpi(n)}$ and p_3 is a parameter. At the end of this stage, having the accumulation table V, p_d families with the highest number of votes in $V(n, \cdot)$ are ascribed to $\hat{\mathbf{g}}_n$. (In *KiKoCh2*, $p_d = 1$.)

The final stage of getting the orientation and assignment is the same as in the case of strategy A; this involves another parameter (which in *KiKoCh2* is set equal to p_2).

[8] In this case, a simpler list built of triplets (d_h, m_1, m_2) can be used; the indices m_1 and m_2 identify items on the complete list of vectors $\hat{\mathbf{h}}_m$ $(m = 1, \ldots, M)$.

B3. Strategy D

This is an example strategy based on triplets of vectors. It is a modified version of strategy A, and the preliminary steps are the same in both cases. The accumulation stage is different. For each triplet $(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2}, \hat{\mathbf{g}}_{n_3})$, get the angular distances $d(\hat{\mathbf{g}}_{n_1}, \hat{\mathbf{g}}_{n_2})$, $d(\hat{\mathbf{g}}_{n_2}, \hat{\mathbf{g}}_{n_3})$ and $d(\hat{\mathbf{g}}_{n_3}, \hat{\mathbf{g}}_{n_1})$. For each of these three distances, retrieve from the `LIST` the items such that the deviation between the distance and d_h is smaller than a given threshold. The threshold, say p_1, is a parameter of strategy D.

The families which appear in the items retrieved based on $d(\hat{\mathbf{g}}_{n_i}, \hat{\mathbf{g}}_{n_j})$ are collected without duplicates in a set S_{ij}. One gets three such sets S_{12}, S_{23} and S_{31}. Now, the vector $\hat{\mathbf{g}}_{n_1}$ is supported by families common to S_{12} and S_{31}, *i.e.* by those in $S_1 = S_{31} \cap S_{12}$. Similarly, $S_2 = S_{12} \cap S_{23}$ and $S_3 = S_{23} \cap S_{31}$ support $\hat{\mathbf{g}}_{n_2}$ and $\hat{\mathbf{g}}_{n_3}$, respectively. Each element of the set S_i casts a vote for $\hat{\mathbf{g}}_{n_i}$, *i.e.* for each f in S_i, the value of $V(n_i, f)$ is increased by 1. As in A, one ultimately gets the resulting accumulator table $V(n, f)$. For a given n, the larger the number $V(n, f)$, the larger the probability that $\hat{\mathbf{g}}_n$ matches a vector from the fth family. The final stage is the same as in the case of strategy A. Strategy D is controlled by the same parameters as strategy A.

Acknowledgements

The author is grateful to Drs Gert Nolze (BAM, Berlin, Germany) and Stuart I. Wright (EDAX, Draper, Utah, USA) for comments on the manuscript.

References

Adams, B. L., Wright, S. I. & Kunze, K. (1993). *Metall. Trans. A*, **24**, 819–831.

Ambühl, C., Chakraborty, S. & Gärtner, B. (2000). *Algorithms – ESA 2000. Proceedings of the 8th Annual European Symposium on Algorithms*, 5–8 September 2000, Saarbrücken, Germany. *Lecture Notes in Computer Science*, Vol. 1879, edited by M. Paterson, pp. 52–64. Heidelberg: Springer.

Bengoetxea, E. (2002). *Inexact Graph Matching Using Estimation of Distribution Algorithms.* Ecole Nationale Supérieure des Télécommunications, Paris, France.

Besl, P. & McKay, N. D. (1992). *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 239–256.

Beyerlein, K. R., White, T. A., Yefanov, O., Gati, C., Kazantsev, I. G., Nielsen, N. F.-G., Larsen, P. M., Chapman, H. N. & Schmidt, S. (2017). *J. Appl. Cryst.* **50**, 1075–1083.

Bouscaud, D., Berveiller, S., Pesci, R., Patoor, E. & Morawiec, A. (2014). *Adv. Mater. Res.* **996**, 45–51.

Bunge, H. J. (1982). *Texture Analysis in Materials Science.* Oxford: Butterworth.

Busing, W. R. & Levy, H. A. (1967). *Acta Cryst.* **22**, 457–464.

Cardoze, D. E. & Schulman, L. J. (1998). *Proceedings of the 39th Annual Symposium on the Foundations of Computer Science*, 8–11 November 1998, Palo Alto, California, USA, pp. 156–165. Washington DC: IEEE Computer Society Press.

Chen, K., Dejoie, C. & Wenk, H.-R. (2012). *J. Appl. Cryst.* **45**, 982–989.

Chen, Y., Park, S. U., Wei, D., Newstadt, G., Jackson, M., Simmons, J., De Graef, M. & Hero, A. (2015). *Microsc. Microanal.* **21**, 739–752.

Chung, J. S. & Ice, G. E. (1999). *J. Appl. Phys.* **86**, 5249–5255.

Cowley, J. M. (1981). *Diffraction Physics*. Amsterdam: North-Holland.

Duisenberg, A. J. M. (1992). *J. Appl. Cryst.* **25**, 92–96.

Edington, J. W. (1975). *Practical Electron Microscopy in Materials Science*, Monograph Two, *Electron Diffraction in the Electron Microscope.* London: Macmillan.

Field, D. P. (1997). *Ultramicroscopy*, **67**, 1–9.

Fundenberger, J. J., Bouzy, E., Goran, D., Guyon, J., Yuan, H. & Morawiec, A. (2016). *Ultramicroscopy*, **161**, 17–22.

Fundenberger, J. J., Morawiec, A., Bouzy, E. & Lecomte, J. S. (2003). *Ultramicroscopy*, **96**, 127–137.

Gerig, G. & Klein, F. (1986). *Proceedings of the 8th International Conference on Pattern Recognition*, 27–31 October 1986, Paris, France, pp. 498–500. Washington DC: IEEE Computer Society Press.

Gevorkov, Y., Barty, A., Brehm, W., White, T. A., Tolstikova, A., Wiedorn, M. O., Meents, A., Grigat, R.-R., Chapman, H. N. & Yefanov, O. (2020). *Acta Cryst.* A**76**, 121–131.

Goodrich, M. T., Mitchell, J. S. B. & Orletsky, M. W. (1999). *IEEE Trans. Pattern Anal. Mach. Intell.* **21**, 371–379.

Groth, E. (1986). *Astron. J.* **91**, 1244–1248.

Gupta, V. K. & Agnew, S. R. (2009). *J. Appl. Cryst.* **42**, 116–124.

Heyl, J. S. (2013). *Mon. Notes R. Astron. Soc.* **433**, 935–939.

Hielscher, R., Bartel, F. & Britton, T. B. (2019). *Ultramicroscopy*, **207**, 112836.

Ice, G. E. & Pang, J. W. L. (2009). *Mater. Charact.* **60**, 1191–1201.

Inokuti, Y., Maeda, C. & Ito, Y. (1985). *Metall. Trans. A*, **16**, 1613–1623.

Inokuti, Y., Maeda, C. & Ito, Y. (1987). *Trans. Iron Steel Inst. Jpn*, **27**, 302–311.

Inokuti, Y., Shimizu, Y. & Shimanaka, H. (1980). *Kawasaki Steel Giho*, **12**, 89–98.

Kabsch, W. (1976). *Acta Cryst.* A**32**, 922–923.

Kabsch, W. (1978). *Acta Cryst.* A**34**, 827–828.

Kabsch, W. (1988). *J. Appl. Cryst.* **21**, 67–72.

Kalinowski, J. A., Makal, A. & Coppens, P. (2011). *J. Appl. Cryst.* **44**, 1182–1189.

Keller, R. R. & Geiss, R. H. (2012). *J. Microsc.* **245**, 245–251.

Kolomenkin, M., Pollak, S., Shimshoni, I. & Lindenbaum, M. (2008). *IEEE Trans. Aerosp. Electron. Syst.* **44**, 441–456.

Krahl-Urban, B., Butz, R. & Preuss, E. (1973). *Acta Cryst.* A**29**, 86–88.

Kuhn, H. W. (1955). *Naval Res. Logistics*, **2**, 83–97.

Larsen, P. M. & Schmidt, S. (2017). *J. Appl. Cryst.* **50**, 1571–1582.

Leavers, V. F. (1993). *Computer Vision Graphics and Image Understanding: Image Understanding*, Vol. 58, pp. 250–264. Berlin: Springer.

Lenthe, W. C., Singh, S. & De Graef, M. (2019). *Ultramicroscopy*, **207**, 112841.

Mackenzie, J. K. (1957). *Acta Cryst.* **10**, 61–62.

Miyamoto, A., Weikusat, I. & Hondoh, T. (2011). *J. Glaciol.* **57**, 103–110.

Morawiec, A. (1998). *J. Appl. Cryst.* **31**, 818–819.

Morawiec, A. (1999). *J. Appl. Cryst.* **32**, 788–798.

Morawiec, A. (2004). *Orientations and Rotations: Computations in Crystallographic Textures*. Berlin: Springer-Verlag.

Morawiec, A. (2007). *J. Appl. Cryst.* **40**, 618–622.

Morawiec, A. (2015). *Mater. Sci. Eng.* **82**, 012008.

Morawiec, A. (2016). *J. Appl. Cryst.* **49**, 322–329.

Morawiec, A. (2017). *J. Appl. Cryst.* **50**, 647–650.

Morawiec, A. & Bieda, M. (2005). *Arch. Met. Mater.* **50**, 47–56.

Morawiec, A., Bouzy, E., Paul, H. & Fundenberger, J. J. (2014). *Ultramicroscopy*, **136**, 107–118.

Morawiec, A., Fundenberger, J.-J., Bouzy, E. & Lecomte, J.-S. (2002). *J. Appl. Cryst.* **35**, 287.

Morawiec, A., Pesci, R. & Lecomte, J. S. (2008). *Ceram. Trans.* **201**, 163–170.

Mortari, D., Samaan, M. A., Bruccoleri, C. & Junkins, J. L. (2004). *Navigation*, **51**, 171–183.

Naresh-Kumar, G., Vilalta-Clemente, A., Jussila, H., Winkelmann, A., Nolze, G., Vespucci, S., Nagarajan, S., Wilkinson, A. J. & Trager-Cowan, C. (2017). *Sci. Rep.* **7**, 10916.

Ohba, R., Uehira, I. & Hondoh, T. (1981). *Jpn. J. Appl. Phys.* **20**, 811–816.

Poulsen, H. F. (2004). *Three-Dimensional X-ray Diffraction Microscopy.* Berlin: Springer.

Preuss, E., Krahl-Urban, B. & Butz, R. (1974). *Laue Atlas*. Düsseldorf: Bertelsmann Universitätsverlag.

Quey, R., Villani, A. & Maurice, C. (2018). *J. Appl. Cryst.* **51**, 1162–1173.

Ram, F. & De Graef, M. (2018). *Acta Mater.* **144**, 352–364.

Rangarajan, A., Chui, H. & Bookstein, F. L. (1997). *Information Processing in Medical Imaging*, edited by J. Duncan & G. Gindi, pp. 29–42. Berlin: Springer.

Rauch, E. F., Véron, M., Portillo, J., Bultreys, D., Maniette, Y. & Nicolopoulos, S. (2008). *Microsc. Anal.* **22**, S5–S8.

Ravelli, R. B. G., Hezemans, A. M. F., Krabbendam, H. & Kroon, J. (1996). *J. Appl. Cryst.* **29**, 270–278.

Roşca, D., Morawiec, A. & De Graef, M. (2014). *Modell. Simul. Mater. Sci. Eng.* **22**, 075013.

Schmidt, S. (2014). *J. Appl. Cryst.* **47**, 276–284.

Stephens, M. A. (1979). *Biometrika*, **66**, 41–48.

Tamura, N. (2014). *Strain and Dislocation Gradients from Diffraction*, edited by R. Barabash & G. Ice, pp. 125–155. London: Imperial College Press.

Tamura, N., MacDowell, A. A., Spolenak, R., Valek, B. C., Bravman, J. C., Brown, W. L., Celestre, R. S., Padmore, H. A., Batterman, B. W. & Patel, J. R. (2003). *J. Synchrotron Rad.* **10**, 137–143.

Tsin, Y. & Kanade, T. (2004). *Proceedings of the 8th European Conference on Computer Vision*, 11–14 May 2004, Prague, Czech Republic, edited by T. Pajdla & J. Matas, Part III, pp. 558–569. Berlin: Springer.

Wahba, G., Farrell, J. L. & Stuelpnagel, J. C. (1966). *SIAM Rev.* **8**, 384–386.

Wenk, H. R., Heidelbach, F., Chateigner, D. & Zontone, F. (1997). *J. Synchrotron Rad.* **4**, 95–101.

Whitley, W., Stock, C. & Huxley, A. D. (2015). *J. Appl. Cryst.* **48**, 1342–1345.

Winkelmann, A. & Nolze, G. (2015). *Ultramicroscopy*, **149**, 58–63.

Wolfson, H. J. & Rigoutsos, I. (1997). *IEEE Comput. Sci. Eng.* **4**, 10–21.

Wright, S. I. & Adams, B. L. (1992). *Metall. Trans. A*, **23**, 759–767.

Wright, S. I. & Dingley, D. J. (1998). *Mater. Sci. Forum*, **273–275**, 209–214.

Wright, S. I., Singh, S. & De Graef, M. (2019). *Microsc. Microanal.* **25**, 675–681.

Zhang, G. (2017). *Star Identification: Methods, Techniques and Algorithms.* Berlin: Springer.