

Automatic Debye–Scherrer elliptical ring extraction via a computer vision approach

Saadia Shahzad,^a Nazar Khan,^a Zubair Nawaz^a and Claudio Ferrero^{b*}

^aPUCIT, Allama Iqbal Campus, University of the Punjab, Lahore, Pakistan, and ^bThe European Synchrotron, 71 avenue des Martyrs, 38000 Grenoble, France. *Correspondence e-mail: ferrero@esrf.fr

Received 18 July 2017

Accepted 7 January 2018

Edited by A. F. Craievich, University of São Paulo, Brazil

Keywords: powder diffraction; calibration; 2D patterns; computer vision.

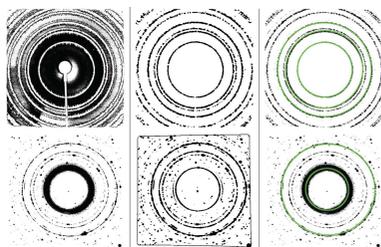
The accurate calibration of powder diffraction data acquired from area detectors using calibration standards is a crucial step in the data reduction process to attain high-quality one-dimensional patterns. A novel algorithm has been developed for extracting Debye–Scherrer rings automatically using an approach based on computer vision and pattern recognition techniques. The presented technique requires no human intervention and, unlike previous approaches, makes no restrictive assumptions on the diffraction setup and/or rings. It can detect complete rings as well as portions of them, and works on several types of diffraction images with various degrees of ring graininess, textured diffraction patterns and detector tilt with respect to the incoming beam.

1. Introduction

The increasingly wider use of area detectors for X-ray powder diffraction (XRPD) measurements at advanced synchrotron radiation sources has led to numerous innovative experiments exploiting nanosized X-ray beams (Dinnebier & Hinrichsen, 2012). The considerable advantages of 2D data collection are the excellent data statistics and the very short exposure times (down to some microseconds), which entail extremely reduced experiment durations, as compared with just a very few years ago. Nowadays, in the frame of an XRPD mapping experiment one may collect good quality frames at frame rates as high as 3 kHz in continuous mode or 9 kHz in 30 s bursts by using, for example, one of the Eiger detectors (Johnson *et al.*, 2012; Gorfman, 2014).

Although in most measurements the detector is centred orthogonally to the incident beam, in the most general case the detector is tilted and offset with respect to the primary beam axis. In this case the diffracted Debye–Scherrer rings can take the form of any conical section (ellipse, hyperbola and parabola or even pairs of straight lines). In the majority of cases the powder patterns are actually ellipses. The customary practice is to azimuthally integrate the image along the ellipses (Hammersley *et al.*, 1996), thus reducing the amount of information by the square root of the number of pixels and consequently the disk storage space (Dinnebier & Hinrichsen, 2012).

Since the experimental setup for powder diffraction measurement has been extensively described elsewhere [see, for example, the review by Lavina *et al.* (2014)], we will focus essentially on the calibration procedure. To this effect, one needs to determine and refine numerically the experimental parameters of the beam centre, the sample-to-detector distance, in some particular cases the X-ray wavelength and the spatial orientation of the detector with respect to the beam



(De Nolf *et al.*, 2014). Typically, high-quality patterns of standard samples are used to determine and refine the required parameters. Several methods have been used for refining the calibration parameters [see, for example, Dinnebier & Hinrichsen (2012), and references therein; Hart *et al.* (2013); De Nolf *et al.* (2014); Tantau *et al.* (2014); Lutterotti *et al.* (2014)]. The calibration process was totally manual for early experiments. Some semi-manual approaches were developed later [*Fit2d* (Hammersley, 2016) and *PyFAI* (Kieffer & Karkoulis, 2013)], which estimate parameters on rings chosen manually. Well known computer programs commonly employed for calibration purposes and for analysing diffraction images include *Two2One* (Vogel & Knorr, 2005), *Powder3d* (Hinrichsen *et al.*, 2006), *Datasqueeze* (Heiney, 2005) and the aforementioned *Fit2d* and *PyFAI*.

Pattern recognition techniques were developed to estimate calibration parameters featuring high accuracy along with some degree of automation (Rajiv *et al.*, 2007; Cervellino *et al.*, 2006; Hart & Drakopoulos, 2013). However, these approaches make restrictive assumptions about the diffraction setup and/or the diffraction rings.

Several synchrotron sources are currently following update programs to face the new challenges posed by science and technology. The size of the beam is becoming of the same order of magnitude as the grains in the powder used. In this case, the problem tends to become that of diffraction from single crystals, with less random orientations of the crystals. The diffraction pattern shows more scattered peaks than well defined rings. As it is very difficult to reduce even further the size of grains in the powder, one has to deal with this problem at the image-processing level.

The basic task of the present work has been to develop a novel algorithm for extracting diffraction rings, which should not need human intervention to capture the rings or portions thereof. Another point is that it should work on several types of image, with various degrees of graininess. The basic idea is to use a computer vision approach automatically to analyse the two-dimensional diffraction frames and search and classify the ring-like shapes by sub-families of conic sections, the geometric parameters of which would enable the refinement of the instrumental parameters.

In the following sections the different steps toward the automatic extraction of the Debye–Scherrer rings are illustrated, starting from the image pre-processing, followed by morphological dilation and thinning operations (§2.1). These morphological operations are parts of an algorithm to control the growth of regions bearing conic candidates (§2.2) and use an incremental detection mechanism (§2.3). The algorithm has been validated for ellipses. The main features of the ellipse harvesting strategy are then explained introducing suitable convergence and ranking criteria (§3). We remove false detections (§4) and recover missed detections by finding ‘families’ of detected ellipses (§5). Finally, a comparison between the suggested method and existing ring detection schemes is discussed (§6). Details of detectors and beamlines used to collect data for use in this paper are presented in Appendix A.

2. Computer-vision aided ring extraction

As mentioned in the *Introduction*, Debye–Scherrer ring fitting is the core of XRPD data calibration. We propose a computer-vision-aided algorithm to automatically extract Debye–Scherrer rings in XRPD images. The detection of these rings allows the calibration of the experiment (Ashiotis *et al.*, 2015). The calibration process is based on obtaining individual elliptic rings. Manual marking of the rings can be slow as well as error prone. Our algorithm can be used to automate this process.

As shown in Fig. 1, segments of Debye–Scherrer rings can appear in a wide variety of ways from almost circular to elliptical, parabolic and even almost linear. The modular structure of certain detectors (*e.g.* Pilatus detectors from Dectris) may produce rings which do not appear to be contiguous. This makes the automatic detection of such rings or ring portions a challenging task.

Considerable research exists on the ellipse fitting problem when points representing an ellipse are provided *a priori* (Gander *et al.*, 1996; Fitzgibbon *et al.*, 1999; Kanatani & Sugaya, 2008). However, the problem of grouping points that represent elliptical regions has received less attention (Qiao & Ong, 2007; Chia *et al.*, 2011; Wong *et al.*, 2012; Pătrăucean *et al.*, 2012).

If there are multiple ellipses in one image, it is not easy to automatically identify which set of points belong to which ellipse. This could be done by viewing the image and marking points manually but it is tedious and very time-consuming to identify the boundary points of one ellipse from a list of all points in the image. As demonstrated in Fig. 2, if we can identify the points lying on one ellipse, ellipse fitting is trivial. On the other hand, if we know the parameters of an ellipse, then obtaining the points that lie on the ellipse is trivial. However, in the absence of both (points as well as ellipse), ellipse detection is a challenging task and can be treated as a latent variable problem.

We approach this problem in an incremental fashion, similar in spirit to the incremental nature of the expectation

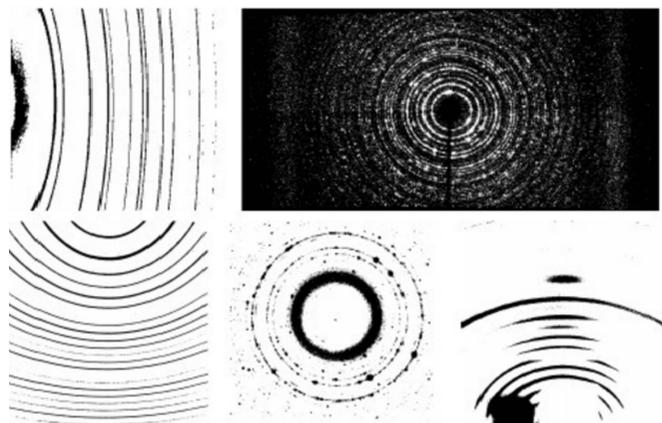


Figure 1
Some examples of Debye–Scherrer ring images. Segments of Debye–Scherrer rings can appear in a wide variety of ways from almost circular to elliptical, parabolic and even almost linear.

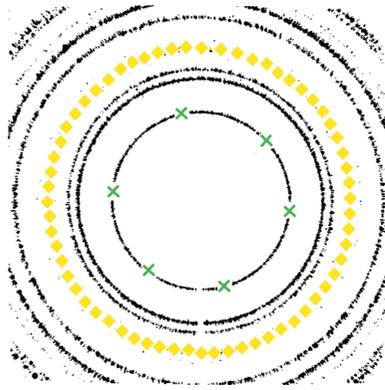


Figure 2
The chicken-and-egg nature of the ellipse detection problem. Upon identifying some points on the boundary of a candidate ellipse (green crosses), an ellipse can be fitted through these points. Conversely, knowing the parameters of an ellipse, all points on the boundary can be identified (yellow diamonds).

maximization (EM) algorithm (Dempster *et al.*, 1977) used for solving latent variable problems. Accordingly, we name our algorithm Incremental Ellipse Detection (IED).

We start with an initial region (set of points) and fit an ellipse to it. Then we select additional points that lie close to both the region and the corresponding fitting ellipse and add these to the region. A new fitting ellipse is then adjusted to this expanded region and the process is repeated until no more points can be added to the region. Other regions in the image can similarly be processed in sequence to obtain all elliptical region estimates, *i.e.* the Debye–Scherrer rings.

We now describe the process in detail. After initial pre-processing (§2.1), we merge sets of points to identify potential elliptical regions (§2.2). Then we detect ellipses using the IED algorithm (§2.3). Pseudo-code of the algorithm is presented in Appendix B.

2.1. Gap filling

Debye–Scherrer rings can comprise unconnected sets of pixels. There can be gaps along the rings. We pre-process the data to fill such gaps. This gap-filling step enables our algorithm to find connected elliptic arcs more accurately.

Let I denote an image containing Debye–Scherrer rings. Since the range of pixel intensities in such images can vary significantly, we force the intensities to lie between 0 and 255 *via* an affine rescaling as

$$I(x, y) = \frac{I(x, y) - \min(I)}{\max(I) - \min(I)} \times 255. \quad (1)$$

Then we smooth the image I convolving with a rotationally symmetric Gaussian low-pass filter of size 2×2 with standard deviation $\sigma = 1$. After that, spatial derivatives in vertical and horizontal directions are computed using equations (2) and (3) [as done by von Gioi *et al.* (2012)] to obtain the gradient vector $\nabla I = I_x$ at every pixel. The pixel locations corresponding to equations (2) and (3),

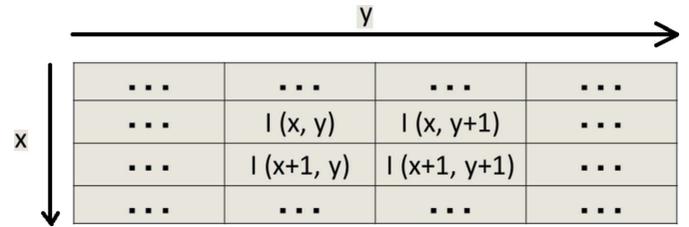


Figure 3
For the pixel $I(x, y)$, we use its neighbouring pixels to compute the derivatives in the x and y directions.

$$I_x(x, y) = \frac{[I(x + 1, y) - I(x, y)] + [I(x + 1, y + 1) - (I(x, y + 1))]}{2}, \quad (2)$$

$$I_y(x, y) = \frac{[I(x, y + 1) - I(x, y)] + [I(x + 1, y + 1) - (I(x + 1, y))]}{2}, \quad (3)$$

are illustrated in Fig. 3. The magnitude and orientation of the gradient vector ∇I can be computed as

$$m(x, y) = \left\{ [I_x(x, y)]^2 + [I_y(x, y)]^2 \right\}^{1/2}, \quad (4)$$

$$\theta(x, y) = \tan^{-1} \left[\frac{I_y(x, y)}{I_x(x, y)} \right], \quad (5)$$

respectively. Then we select pixels whose gradient magnitudes are greater than or equal to the 90th percentile of all gradient magnitudes in the image. This yields a binary image that represents ring pixels.

The connectivity among pixels along the rings can be improved through a simple morphological operation of dilation. We dilate our thresholded binary image by using a structuring element of size 3×3 containing all ‘1’s. While dilation can fill gaps between pixels, it can also lead to unnecessarily thicker rings. Therefore, dilation is followed by a thinning procedure to neutralize the thickening effect of dilation while retaining its gap-filling effect. Details of morphological dilation and thinning are given by Gonzalez & Woods (2001).

Gap filling enhances the image quality and makes the image more suitable for harvesting connected regions in the region-growing procedure (§2.2). Fig. 4 demonstrates that subsequent region growing produces much better results when gaps are processed by this dilation and thinning sequence. Without gap filling the grown regions are still relatively unconnected and not very elliptical. After gap filling, the regions become better connected, contiguous and more elliptical. Better initial ellipses can be fitted to such regions. This leads to more accurate and faster ellipse detection.

2.2. Region growing

The task here is to identify elliptical candidate regions in an image. The procedure of collecting points in the form of regions is called *region growing*. We group neighbouring pixels

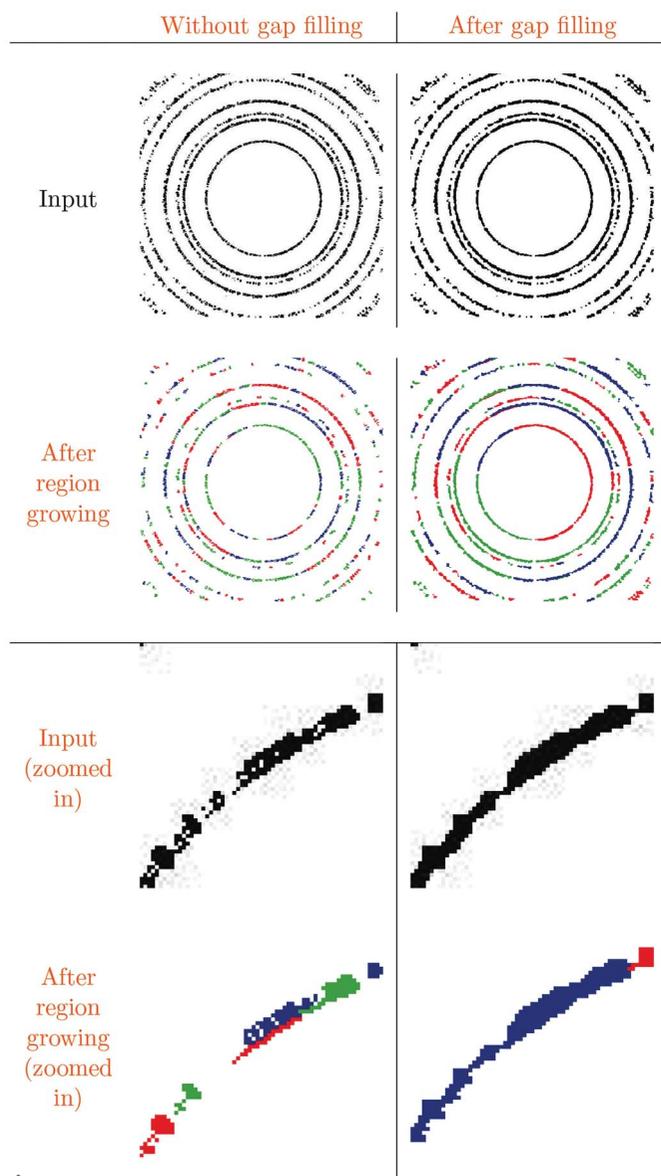


Figure 4 Gap filling improves region growing. Region growing produces much better results when gaps along Debye–Scherrer rings are filled by a dilation and thinning sequence. Without gap filling the grown regions are still relatively unconnected and not very elliptical. After gap filling the regions become better connected, contiguous and more elliptical.

on the basis of similar gradient orientations as discussed by Pătrăucean *et al.* (2012). We explain the method here for completeness.

The logical status of each pixel is initially set to false. Starting from a seed pixel with the largest gradient magnitude, neighbour harvesting is performed by collecting neighbouring pixels which have *almost* the same orientation. Similarity in orientation is scored up to an angular threshold τ , and each time this occurs the status of selected neighbours toggles into true. In the present work, since we are in quest of elliptic regions, we set $\tau = 45^\circ$. Each selected neighbour is, in turn, further polled to find its neighbour candidates. This search stops when no more neighbours exist whose gradient orientation difference with the region’s overall orientation is less

than or equal to τ . The selected pixels are labelled as one region.

Among the remaining pixels with false status, the pixel with largest gradient magnitude is chosen as the next seed pixel and the process is repeated. This continues until there are no more pixels with status false. It can be seen from Fig. 4 that regions grown in this way are better suited for fitting ellipses. This is especially true when region growing follows a gap-filling step.

2.3. Incremental ellipse detection (IED)

Once regions corresponding to elliptical segments are found, we can merge multiple constituent regions to capture valid ellipses. However, which regions to merge is not known *a priori*. Given correct regions, ellipse fitting is easy and, conversely, given a correct ellipse, finding its constituent regions is also easy. However, in our case, neither the ellipse nor the constituent regions are known. Therefore, we detect ellipses by incrementally improving upon elliptical regions and fitted ellipses. We name this algorithm Incremental Ellipse Detection (IED). A flowchart for the IED algorithm is presented in Fig. 5 and pseudocode is presented in Appendix B.

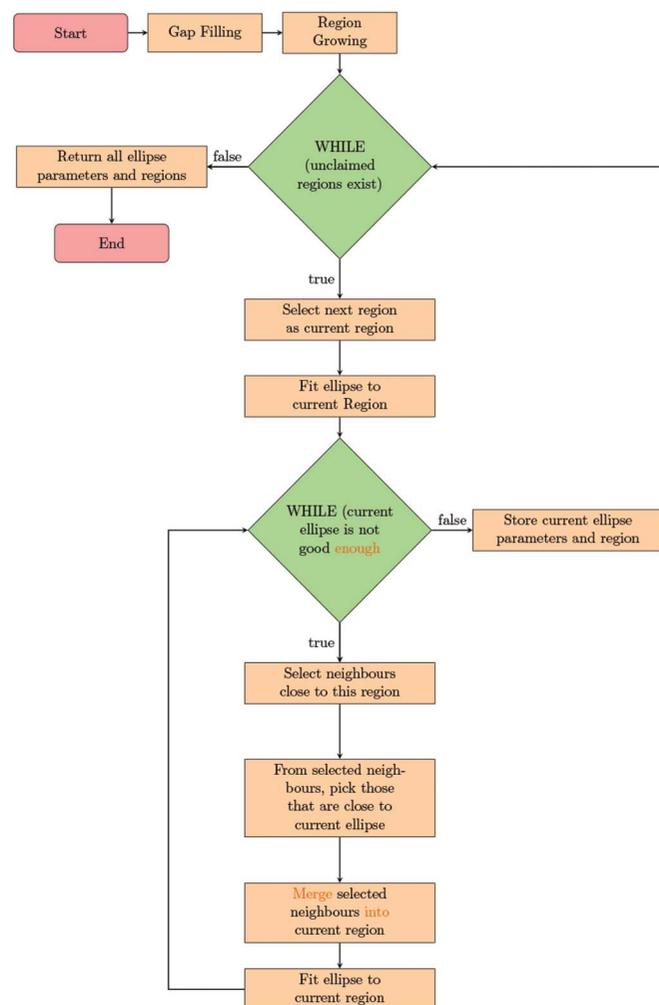


Figure 5 Flowchart for the incremental ellipse detection (IED) algorithm.

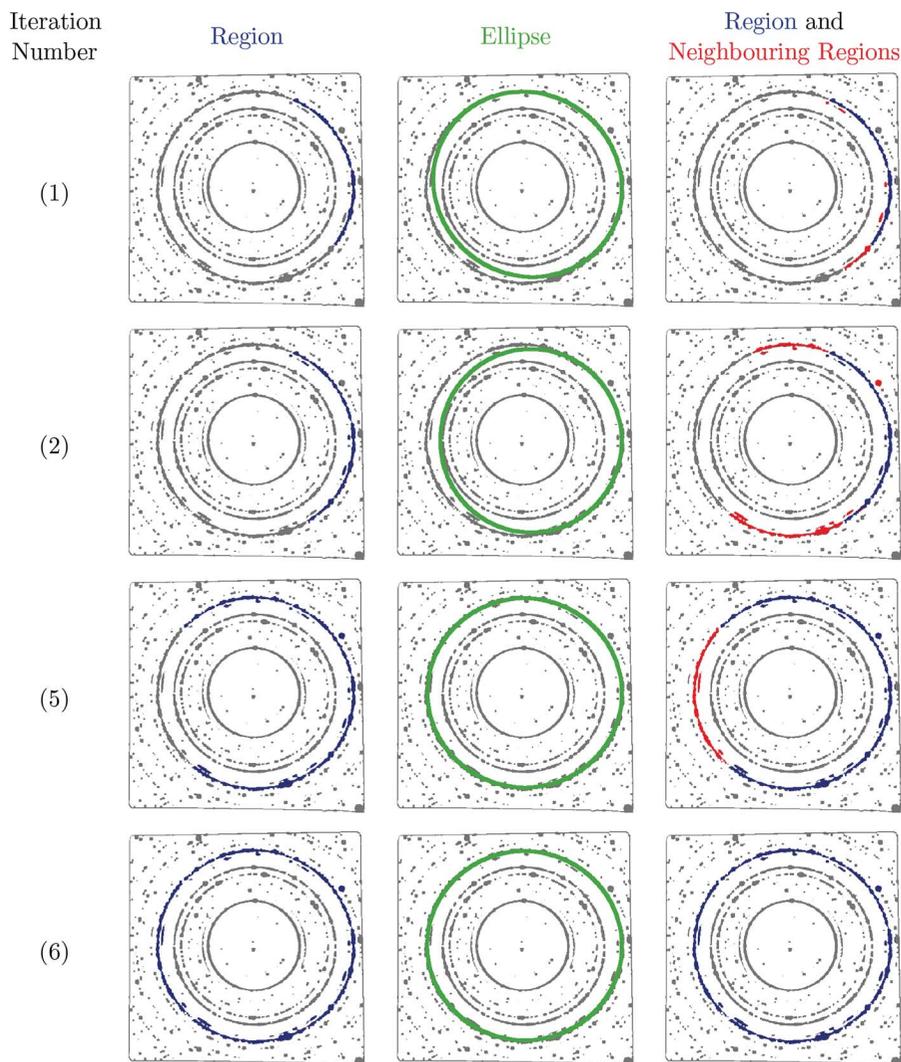


Figure 6 Incremental ellipse detection (IED). Each row shows successive iterations. The first column shows the current region (in blue). The second column shows the ellipse fitted to this region (in green) and the third column shows neighbouring regions (in red) close to the current region and also close to the fitted ellipse. The neighbouring regions are merged into the current region for the next iteration. The process continues until convergence.

A working example of the IED algorithm can be seen in Fig. 6. We start with the largest region (shown in blue) and fit an ellipse (in green) to it. Then, we find other new regions (red) that lie close to the current region (blue) as well as the fitted ellipse (green) and add them to the current region. In the second iteration, a new ellipse is fitted to this enlarged region and the process is repeated until convergence (sixth iteration in the example of Fig. 6). Then we pick the next largest unprocessed region and perform the incremental fitting process. This is repeated until no more unprocessed regions remain.

Fig. 7 shows results on various types of challenging patterns and demonstrates that our algorithm is able to detect significant rings despite beamstop shadows, diffuse scattering, graininess, blobs and non-orthogonality of detectors.

3. Ellipse evaluation

In the present context, ellipse evaluation means finding the fitness of a fitted ellipse with respect to a region. Instead of just visual analysis, we have devised an evaluation method to check numerically the quality of fitted ellipses. The following sections describe the measures for ellipse evaluation and the criteria based on these measures.

3.1. Ellipse evaluation measures

In order to accept or reject an ellipse and then rank all accepted ellipses, we use the following two evaluation measures. Both measures depend upon the set of points R on which ellipse E is fit and the set of points lying on the boundary of ellipse E . These boundary points can be generated *via* the parametric representation of the ellipse.

3.1.1. Percentage of claimed angles.

In order to estimate the fitness score of an ellipse, the angular coverage of an ellipse E by the region R is measured. More coverage means greater fitness. We divide angular extent of the ellipse into 360 discrete angles.

With reference to Fig. 8, let x be a 2D point belonging to region R . Let l be the line segment connecting x with the centre of an ellipse E . The intersection of l with the boundary of ellipse E is denoted by \hat{x}_α where α is the corresponding angle in a parametric representation of the ellipse. Now we can state that region point x *claims* angle α of ellipse E . For all points in region R , let $A(R)$ be the set of corresponding claimed angles with respect to ellipse E . Finally, we define an indicator function

$$v(x) = \begin{cases} 1, & \text{if } x \text{ is viewable in the image,} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Note that for ellipses fit to elongated elliptical or parabolic Debye–Scherrer rings (see last row of Fig. 7), most points on fitted ellipse boundaries will not be viewable within the image. Now we can define an evaluation measure based on the percentage of angles claimed in ellipse E by points in region R as

$$\mathcal{C}(R, E) = \frac{\sum_{\alpha \in A(R)} v(\hat{x}_\alpha)}{\sum_{\alpha=1}^{360} v(\hat{x}_\alpha)} \times 100, \quad (7)$$

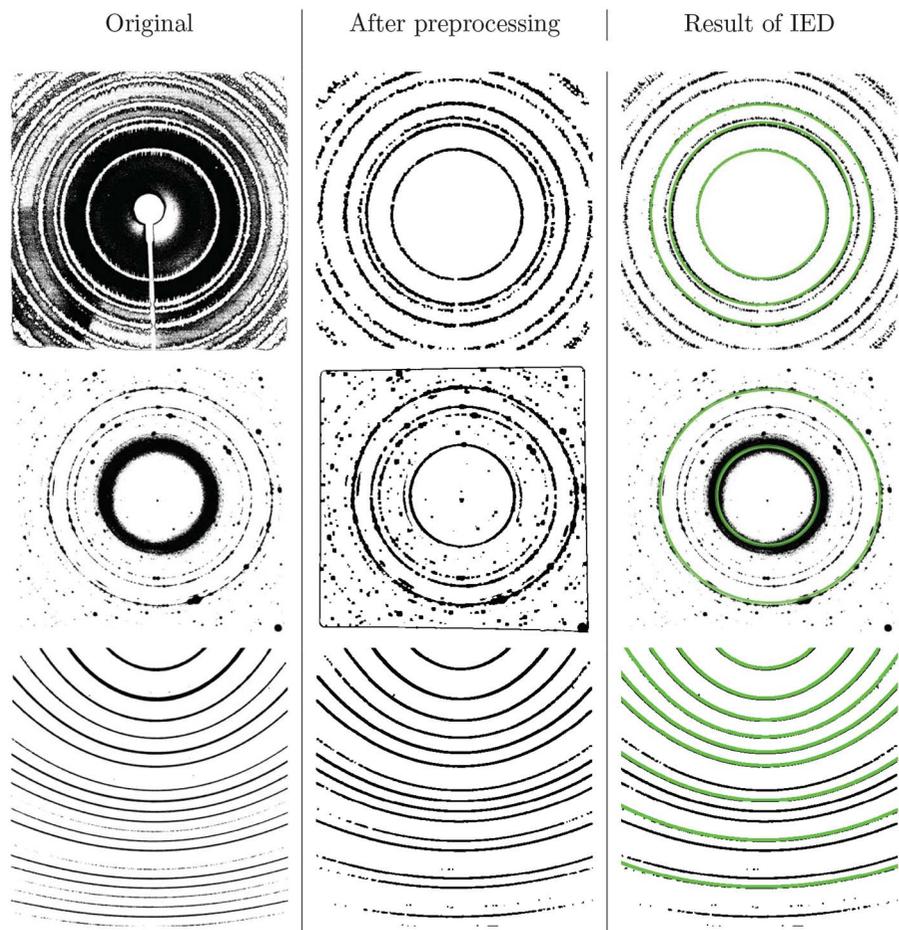


Figure 7 Different challenges in detection of Debye–Scherrer rings are handled by the proposed IED algorithm. Significant rings are detected despite beamstop shadows, diffuse scattering, graininess, blobs and non-orthogonality of detectors.

where the numerator counts the total number of unique viewable angles of ellipse E that are claimed by the points in region R . The denominator is the total number of viewable points on ellipse E .

Fig. 9 shows $\mathcal{C}(R, E)$ values for the (R, E) pairs as the region is incrementally grown. It can be seen that the $\mathcal{C}(R, E)$ value is small when the region is not fully elliptical and large when the region covers an entire ellipse.

3.1.2. Average points-to-ellipse distance. The second measure that we use to assess the goodness of fit between

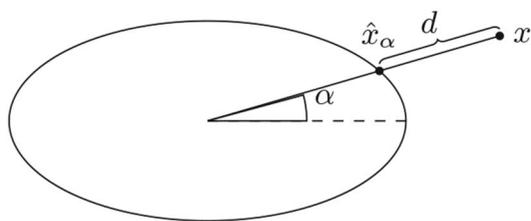


Figure 8 Geometry for a point x lying close to an ellipse. We deem point x to claim angle α on the ellipse and \hat{x}_α is the point on the ellipse boundary corresponding to angle α .

region R and ellipse E is the average distance of points in R from the boundary of ellipse E . This can be computed as

$$\mathcal{D}(R, E) = \frac{1}{|R|} \sum_{x \in R} d(x), \quad (8)$$

where $d(x)$ is the Euclidean distance illustrated in Fig. 8. Fig. 10 displays the values for $\mathcal{D}(R, E)$ for three different (R, E) pairs. It can be seen that the value is large for poor ellipse fits and small for good ellipse fits.

3.2. Ellipse evaluation criteria

We use measures of $\mathcal{C}(R, E)$ and $\mathcal{D}(R, E)$ as described in §3.1 to evaluate the ellipses at certain stages of the IED algorithm. The following subsections describe the use of these measures.

3.3. Convergence criteria

Our ellipse growing algorithm is incremental and iteratively keeps on merging regions to generate an ellipse until

- (i) either the claimed angle percentage $\mathcal{C}(R, E)$ is greater than a threshold value τ_C ,
- (ii) or no more candidate points for region R are found.

This helps in faster execution by stopping the ellipse-growing process as soon as an acceptable level of fitness is achieved.

3.4. Acceptance/rejection criteria

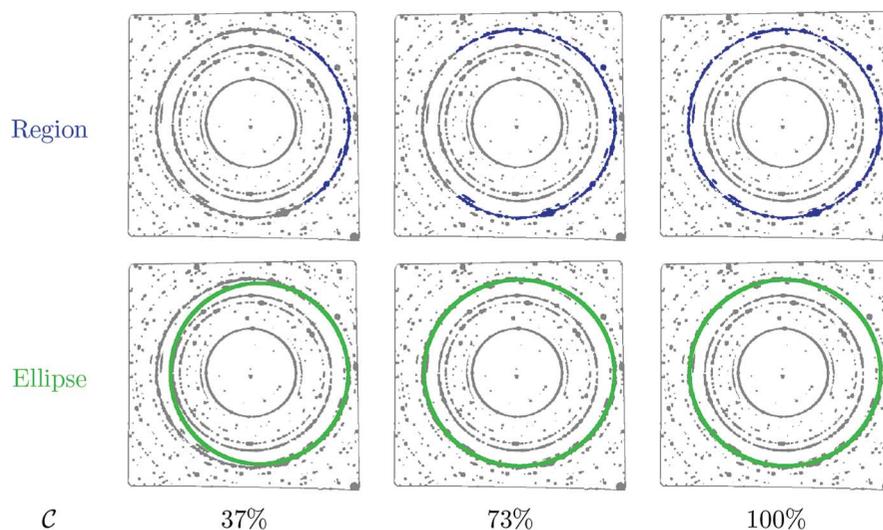
After convergence, it is decided whether an ellipse is valid or not. This is done by validating ellipses in the following three ways. A grown ellipse is accepted if

- (i) the claimed angle percentage $\mathcal{C}(R, E)$ is greater than or equal to a threshold τ_{Cmin} , and
- (ii) the average distance $\mathcal{D}(R, E)$ between region and ellipse is less than a threshold τ_D , and
- (iii) the length ratio between the major axis and the minor axis is less than a threshold τ_R .

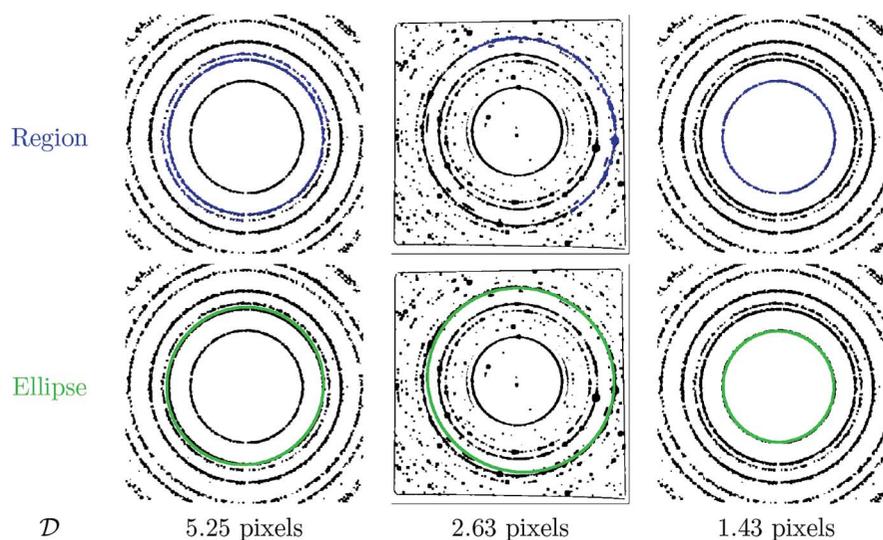
The last condition prevents detection of extremely elongated ellipse configurations.

3.5. Ranking of ellipses

Once all valid ellipses are found, it is useful to rank them so that subsequent processing can be done at a high confidence level. The ranking is done by sorting all valid ellipses in ascending order based on their distance values $\mathcal{D}(R, E)$. The ellipse with the lowest average $\mathcal{D}(R, E)$ value is ranked best. Fig. 11 shows ranking of ellipses in our results. The numbers


Figure 9

Claimed angle percentage $C(R, E)$ as R and E are incrementally grown. As the region becomes more elliptical, the percentage of claimed angles increases.


Figure 10

Average distance to ellipse $D(R, E)$ values for three different (R, E) pairs. It can be seen that the value is large for poor ellipse fits and small for good ellipse fits and is therefore a measure of closeness between the region and the fitted ellipse.

written next to the ellipses indicate the ranks of the respective ellipses. For all calculations in this paper, the threshold values were set as $\tau_C = 90\%$, $\tau_{C_{\min}} = 50\%$, $\tau_D = 10$ and $\tau_{\text{ratio}} = 2$. Notice that both τ_C and $\tau_{C_{\min}}$ are thresholds on the same measure $C(R, E)$. Threshold τ_C is used to decide if an ellipse being incrementally grown has already become good enough. If so the IED algorithm stops improving that ellipse further in order to save time. The other threshold $\tau_{C_{\min}}$ is applied once an ellipse has reached its final form. This final form could be good or bad. Therefore, all ellipses are thresholded to filter out badly fitted ellipses.

It can be observed from all results shown so far that, while good ellipses are detected by our algorithm, there exist some

false detections and also some missed detections. Both problems are addressed in the next two sections. It should be noted, however, that these improvements are for orthogonal detectors only and do not apply to Debye–Scherrer ring images obtained from tilted detectors.

4. Removal of false detections

As can be seen in the last row of Fig. 11, there are many ellipses detected. Some ellipses are true and some are false detections. These false detections occur because of noise in the image and/or very little gap between two ellipses. In this section, we will describe how to remove these false detections. This is done in two phases:

(i) classification of images, *i.e.* whether an image comes from an orthogonal or a tilted detector.

(ii) identification and removal of false ellipses in images from orthogonal detectors.

4.1. Classification of images

The false ellipse detections in the last row of (Fig. 11) belong to a frame issued by an orthogonal detector only. We can easily classify whether a pattern comes from an orthogonal detector or a tilted one by reviewing the parameters of the detected ellipses. One can see that Debye–Scherrer rings for orthogonal detectors are generally scaled versions of each other. Therefore, their major-to-minor axis length ratios are very similar. This allows us to mark any detected ellipse with a significantly different ratio as a false detection.

Specifically, the major-to-minor axis length ratio is recorded for all detected ellipses. Then we compute a histogram of these ratios as shown in Fig. 12. The centre of the histogram bin with maximum count is considered as the mode of these ratios. By thresholding this mode value, we are able to decide whether a given image results from an orthogonal or a tilted detector. The outliers are found and removed in images from orthogonal detectors only.

4.2. Identification and removal of false detections

A key observation here is that, for orthogonal detectors, Debye–Scherrer rings are more or less concentric and false detections can be spread over the span of the detector. Thus,

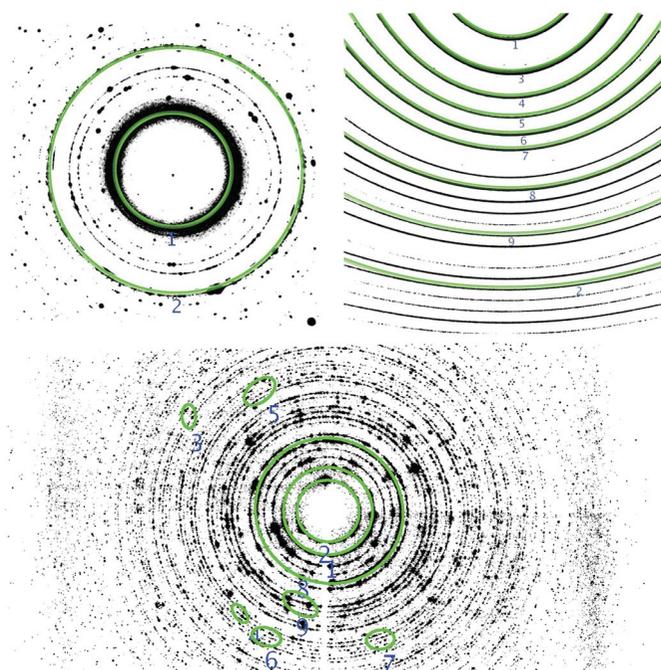


Figure 11
 Top left: two ellipses detected in the presence of blobs and gaps along contours. Top right: valid ellipses detected for partial Debye-Scherrer rings. Bottom: detections in the presence of blobs, gaps, beamstop shadow and densely located Debye-Scherrer rings. Poorly fitted and false detections receive a lower rank based on equation (8).

we want to find a representative centre \mathbf{c} which will represent the centre of the correct ellipses. By finding the deviation of each ellipse centre from \mathbf{c} , we can classify whether an ellipse is an outlier or not. Finding this representative centre cannot be done using the mean value because the mean is always sensitive to outliers and false ellipses' centres are quite far off from \mathbf{c} .

One might be tempted to use the median which is robust to outliers. However, such robustness is only when there are relatively few outliers. When the number of outliers approaches or exceeds the number of inliers, then even the median does not remain robust.

The mode is a good choice because, as stated earlier, the centres of outliers are dispersed and the centres of correct ellipses are close to each other. We use a two-dimensional histogram for finding the mode of the ellipse centres as shown in Fig. 13. This mode centre, \mathbf{c} , works as a representative centre for all correct ellipses.

After finding \mathbf{c} , we calculate the Euclidean distance of all ellipses' centres from it. By thresholding the Euclidean distance, our algorithm automatically decides whether an ellipse is an outlier or a correct one. Fig. 14 displays images before and after removal of false detections. After removal, the remaining ellipses are ranked according to equation (8).

5. Refinement via elliptical family constraint

From the results shown so far, not all Debye-Scherrer rings are detected by our algorithm. Sometimes valid Debye-

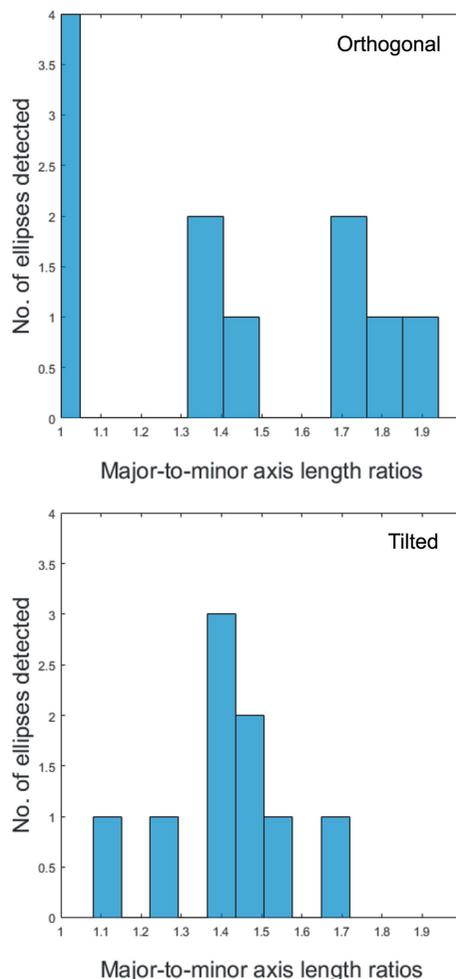


Figure 12
 Histograms of major-to-minor axis length ratio for orthogonal and tilted detectors. Most Debye-Scherrer rings in orthogonal detectors are close to circular and therefore the mode of their axis-length ratios is close to 1. Tilted detectors have a larger ratio since the rings produced are more elliptical. This difference can be used to classify the type of the detector.

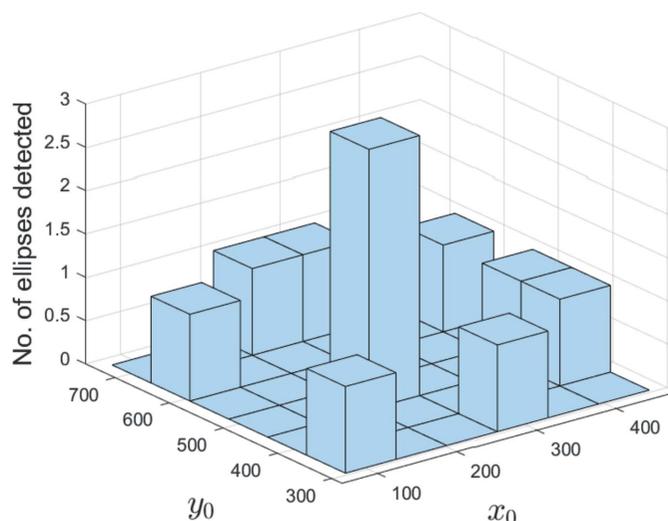


Figure 13
 Two-dimensional histogram of ellipse centres for an orthogonal detector. The mode of this histogram can be used as a representative centre location.

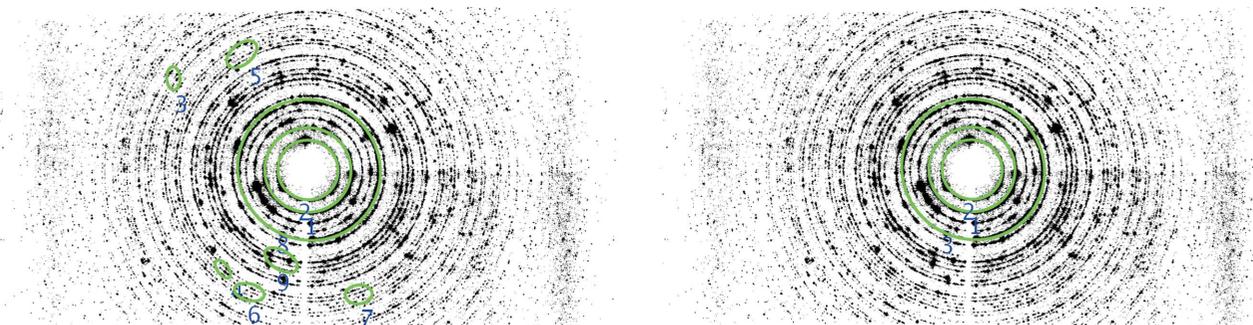


Figure 14 Left: result of IED algorithm with false detections. Right: false detections eliminated by removing outliers with respect to ellipse centres. The remaining ellipses are ranked again according to equation (8).

Scherrer rings are rejected since the detected corresponding ellipses fail to pass all thresholds. This is a trade-off between correct and missing detections and cannot be avoided. Although calibration is possible from only a few rings, it is beneficial to detect as many rings as possible. For this purpose we note that Debye–Scherrer rings can be seen as *families* of ellipses. For orthogonal detectors, a family can be obtained from equally scaled versions of the major and minor ellipse axes such that their ratio is preserved. Consider an ellipse $\mathbf{e}_1 = [e_{\bar{x}}, e_{\bar{y}}, e_{\theta}, e_a, e_b]^T$ with centre $(e_{\bar{x}}, e_{\bar{y}})$, angle e_{θ} , half-major axis length e_a and half-minor axis length e_b . Its family members can be generated as $\mathbf{e}_k = [e_{\bar{x}}, e_{\bar{y}}, e_{\theta}, ke_a, ke_b]^T$ for $k \in \mathbb{R}^+$. This ensures that all members share the same centre, angle and shape (*i.e.* ratio of axis lengths e_a/e_b). Note that this definition of a family is not suitable for non-orthogonal detectors. Therefore, we focus in this section on Debye–Scherrer rings obtained through orthogonal detectors only.

For every detected ellipse \mathbf{e} , the proposed refinement involves generating boundary points E_k corresponding to ellipses \mathbf{e}_k for different¹ $k \in \mathbb{R}^+$. For each such ellipse, we find the set R_k of Debye–Scherrer ring pixels that intersect the boundary E_k . Finally, the validity of set R_k as a Debye–Scherrer ring can be determined by thresholding the ratio $|R_k|/|E_k|$.

A large interval in scales k may cause our algorithm to miss some available ellipses because of the unavailability of intersections. This problem can be alleviated by using a smaller interval between scales but this leads to redundant detections. Therefore, it is important to post-process the results. For this, we compare the geometric representations of ellipses. Specifically, two ellipses \mathbf{e}_1 and \mathbf{e}_2 are considered distinct if the absolute difference $|e_{i_1} - e_{2i}|$ for any component

¹ $k \simeq 1$ is avoided to prevent duplicate detections.

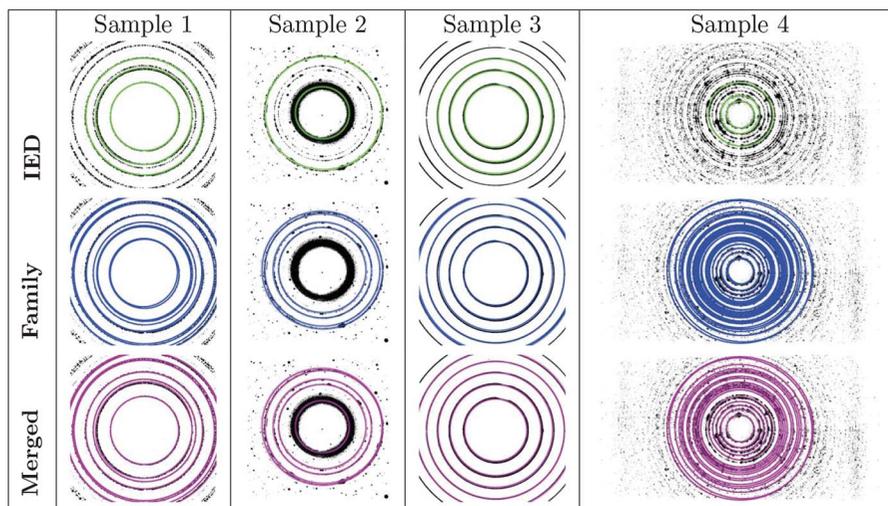


Figure 15 Refinement *via* family constraint for the four different Debye–Scherrer ring images. The top row shows results of the IED algorithm, the middle-row shows results of refinement *via* family constraint and the bottom row shows the merged results.

$i \in \{\bar{x}, \bar{y}, \theta, a, b\}$ is greater than the corresponding threshold τ_i . If all five absolute differences are within their thresholds, then only one ellipse out of \mathbf{e}_1 and \mathbf{e}_2 is retained.

Fig. 15 displays refinement steps on orthogonal detector patterns using the family constraint. It can be noticed that the refinement procedure recovers almost all Debye–Scherrer rings that could have been recovered manually. Since the constraint is valid for orthogonal detectors only, refinement results for non-orthogonal detectors are obviously not so good and this is demonstrated in Fig. 16.

6. Comparisons

6.1. Comparison with the method by Rajiv *et al.*

The work of Rajiv *et al.* (2007) extracts Debye–Scherrer rings for orthogonal detectors only under rather restrictive assumptions using a modified Hough transform. Since a 5D Hough transform for ellipse detection is computationally demanding, the authors suggest to decompose the Hough transform in order to work in 1D only. This is achieved by making the following restrictive assumptions:

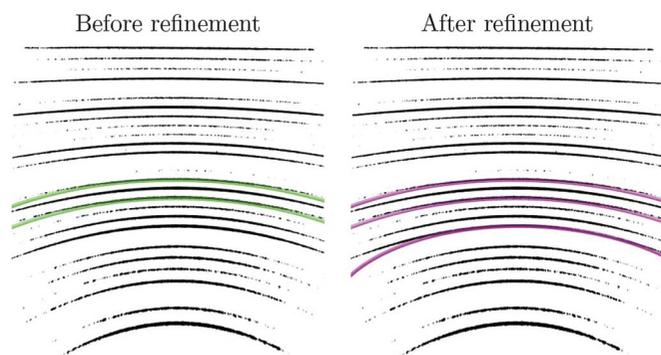


Figure 16
Refinement *via* orthogonal family constraint is not very useful for non-orthogonal detectors which yield highly eccentric Debye–Scherrer rings.

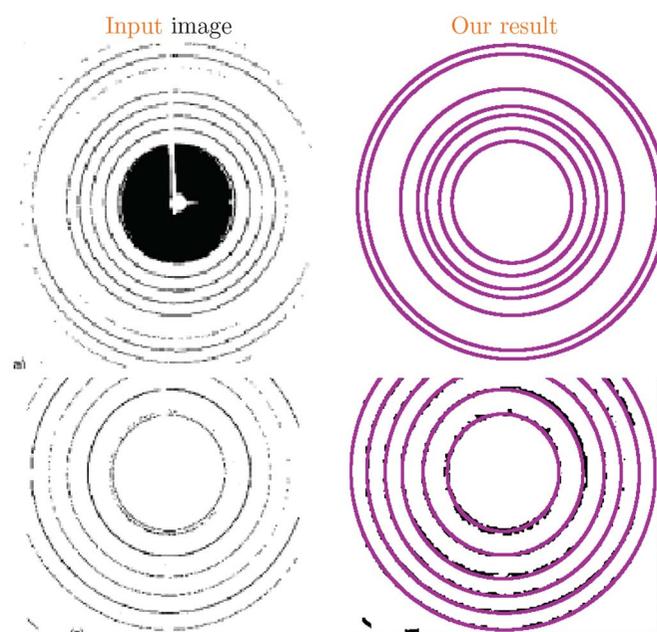


Figure 17
Comparison with Rajiv *et al.* (2007). Our algorithm recognizes a number of ellipses higher than their method. As we could work only on the images extracted from the PDF file of Rajiv *et al.*'s paper, we had to slightly adapt our pre-processing to it.

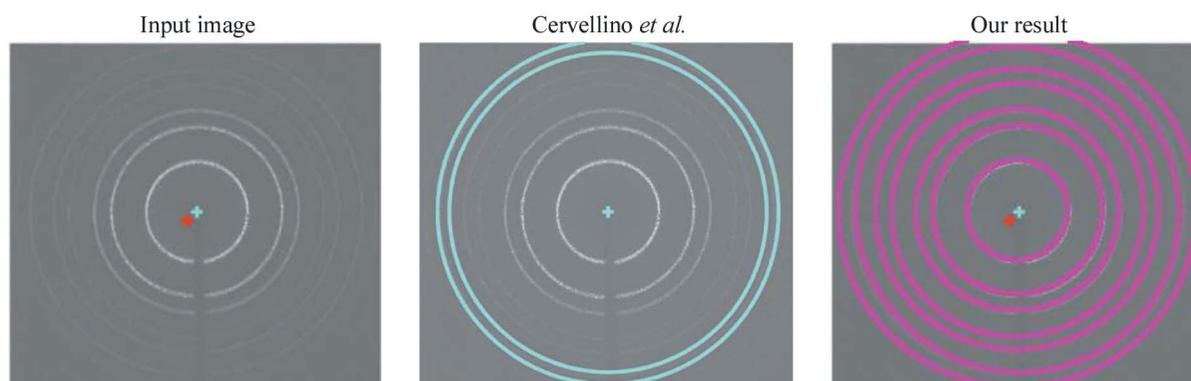


Figure 18
Comparison with the method by Cervellino *et al.* Our algorithm was able to detect all the Debye–Scherrer rings available in the image.

- (i) All Debye–Scherrer rings share the same centre.
- (ii) The centre of the Debye–Scherrer rings has to be within the image.
- (iii) The centre of the ellipse has to be close to the centre of the image.

In addition, their algorithm also depends on carefully chosen radii for Debye–Scherrer rings and this selection of radii is not automatic.

In contrast, our algorithm does not make any of these restrictive assumptions and does not use prior information of the sample or diffraction setup. Our algorithm is able to detect partial Debye–Scherrer rings with centres located outside the image which is often the case when non-orthogonal detectors are used. Fig. 17 compares the result of our algorithm on two of the images used by this method (Rajiv *et al.*, 2007). Our algorithm was able to detect all the Debye–Scherrer rings available in the image.

6.2. Comparison with the method by Cervellino *et al.*

Cervellino *et al.* (2006) describe a method to identify ellipses with Debye–Scherrer rings imposing the following two constraints:

- (i) Images should be taken from an (almost) orthogonal detector.
- (ii) The sample-to-detector distance should be known.

The Debye–Scherrer rings resulting from such detectors are nearly circular and most of the rings are completely visible in the images they show. In contrast, the algorithm we propose can work for orthogonal as well as for tilted detectors and does not require any parameter specific to the sample or detector.

Fig. 18 compares results of method by Cervellino *et al.* (2006) with the result of our proposed method on two Debye–Scherrer ring images. Our algorithm was able to detect all the Debye–Scherrer rings available in the image.

6.3. Comparison with the methods by Hart *et al.*

A method for calibration using manually extracted Debye–Scherrer rings and minimal *a priori* knowledge of the calibration setup is presented by Hart *et al.* (2013). Since the extraction of the rings is carried out manually, the method is not directly comparable with our automatic method. In

a separate work, Hart & Drakopoulos (2013) provide a procedure for fitting an ellipse to complete as well as spotty rings. This method is based upon an approximate ellipse centre, identification of peak locations along each radial direction of the ring and the weighted least-squares algebraic fit of an ellipse.

They define circular annuli (a ring-shaped region bounded by two concentric circles) such that each diffraction ring lies completely within its own annulus. This definition of annuli is based on the approximate peak locations. While finding approximate peak locations, false positives may exist in the case of spotty rings. In this case, the authors let the reader decide whether the best fit of an ellipse is representative of the underlying data or not.

In the case of tilted detectors, it is recommended to use elliptic annuli instead of circular ones. The selection of elliptic annuli again depends on approximate instrument calibration parameters. An underlying assumption is that this mechanism will work on orthogonal and slightly tilted Debye–Scherrer rings only. In contrast, our method does not make any assumption and also works for a wider range of tilted Debye–Scherrer rings.

7. Conclusion

We have proposed a new ellipse extraction method which is specialized in the automatic detection of Debye–Scherrer rings in 2D XRPD patterns. This method provides a mechanism to detect complete as well as partial Debye–Scherrer rings in a given image. The first part of our algorithm, called Incremental Ellipse Detection (IED), starts by grouping connected contours to find small elliptic arcs. Then it gradually generates full ellipses by grouping unconnected elliptic arcs in an incremental way. In the second step, these initial results are refined by removing false positives and recovering missed detections by finding families of detected ellipses. These refinements are applicable to orthogonal detectors in which Debye–Scherrer rings appear in elliptical shapes. For non-orthogonal detectors, the rings can appear in parabolic or even hyperbolic shapes. Refinement of results for these cases requires further insights and is left as future work.

We also provide measures for evaluating the goodness of fit between points representing Debye–Scherrer rings and the ellipses fitted to them. Every fitted ellipse is evaluated based on the angular coverage of the fitted portion and on the average distance of the points to the fitted ellipse. This allows the experimenter to rank and choose from the detected ellipses based on a numerical criterion. The proposed method for automatic detection of Debye–Scherrer rings can dramatically accelerate the calibration process for batches of diffraction images. MATLAB code for the proposed algorithm

Table 1
Detectors and beamlines used to collect data.

Figure number	File(s)	Detector	Beamline/Synchrotron Facility
2	max1	Frelon CCD	ID16B/ESRF
4	max1	Frelon CCD	ID16B/ESRF
6	Si12_002	Frelon CCD	ID11/ESRF
7	1. max1	1. Frelon	1. ID16B/ESRF
	2. Si12_0002	2. Frelon CCD	2. ID11/ESRF
	3. tilted_002	3. Titan CCD (from Agilent Technologies)	3. BL711/MAX II, Lund
9	Si12_0002	Frelon CCD	ID11/ESRF
10	1. max1	1. Frelon CCD	1. ID16B/ESRF
	2. Si12_0001	2. Frelon CCD	2. ID11/ESRF
11	1. Si12_0002	1. Frelon CCD	1. ID11/ESRF
	2. tilted_002	2. Titan CCD (from Agilent Technologies)	2. BL711 / MAX II, Lund
	3. Lab6_0021	3. (Half active surface) Frelon CCD	3. ID11 / ESRF
14	Lab6_0021	(Half active surface) Frelon CCD	ID11 / ESRF
15	1. max1	1. Frelon CCD	1. ID16B / ESRF
	2. Si12_0002	2. Frelon CCD	2. ID11 / ESRF
	3. tilted_002	3. Titan CCD (from Agilent Technologies)	3. BL711 / MAX II, Lund
	4. Lab6_0021	4. (Half active surface) Frelon CCD	4. ID11 / ESRF
16	tilted_006	Titan CCD (from Agilent Technologies)	BL711 / MAX II, Lund

can be downloaded from <https://github.com/saadiasm/dsr/archive/master.zip>.

APPENDIX A

Table of sources

Details of detectors and beamlines used to collect data for use in this paper are presented here in Table 1.

APPENDIX B

Incremental ellipse detection (IED) algorithm

Pseudo-code of the algorithm discussed in §2 is shown here in Fig. 19.

Acknowledgements

We are thankful to Dr Jerome Kieffer from the ESRF, Grenoble, for his kind support in problem definition, data acquisition and invaluable suggestions about the presented algorithmic workflow. Grateful acknowledgement is also expressed to Dr Andy Fitch from the ESRF, Grenoble, for critical review of the manuscript.

References

- Ashiotis, G., Deschildre, A., Nawaz, Z., Wright, J. P., Karkoulis, D., Picca, F. E. & Kieffer, J. (2015). *J. Appl. Cryst.* **48**, 510–519.
- Cervellino, A., Giannini, C., Guagliardi, A. & Ladisa, M. (2006). *J. Appl. Cryst.* **39**, 745–748.
- Chia, A. Y.-S., Rahardja, S., Rajan, D. & Leung, M. K. (2011). *IEEE Trans. Image Process.* **20**, 1991–2006.
- Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). *J. R. Stat. Soc. Ser. B*, pp. 1–38.
- De Nolf, W., Vanmeert, F. & Janssens, K. (2014). *J. Appl. Cryst.* **47**, 1107–1117.
- Dinnebier, R. E. & Hinrichsen, B. (2012). *Uniting Electron Crystallography and Powder Diffraction*, pp. 251–257. Dordrecht: Springer.

```

1: procedure INCREMENTALELLIPSEDETECTION ▷ Detect ellipses via incremental
   region and ellipse improvement
   Rinitial,                                ▷ regions returned by REGIONGROW
   Efinal                                  ▷ valid fitted ellipses
   Rfinal                                  ▷ regions corresponding to fitted ellipses
   Cfinal                                  ▷ claimed angle percentages for each ellipse-region pair
   Dfinal                                  ▷ average point to ellipse distances for each ellipse-region pair
   ranking                                  ▷ ranking of fitted ellipses
2:   eNum ← 1
3:   UR ← Rinitial
4:   while UR ≠ ∅ do
5:     CR ← largest unclaimed region
6:     CE ← FITELLIPSE(CR)
7:     [C, D] ← EVALUATEELLIPSE(CR, CE)
8:     while (C < τC) do
9:       N ← GETREGIONNEIGHBOURS(CR, Rinitial)
10:      N ← GETELLIPSENEIGHBOURS(CE, N)
11:      CR ← CR ∪ N
12:      CE ← FITELLIPSE(CR)
13:      [C, D] ← EVALUATEELLIPSE(CR, CE)
14:     end while
15:     Efinal ← Efinal ∪ CE
16:     Rfinal ← Rfinal ∪ CR
17:     Cfinal ← Cfinal ∪ C
18:     Dfinal ← Dfinal ∪ D
19:     UR ← UR \ CR
20:   end while
21:   ranking ← SORT(Dfinal, ascending)
22: end procedure

```

Figure 19
The incremental ellipse detection (IED) algorithm.

Fitzgibbon, A., Pilu, M. & Fisher, R. B. (1999). *IEEE Trans. Pattern Anal. Machine Intell.* **21**, 476–480.
 Gander, W., Golub, G. H. & Strebel, R. (1996). *Bull. Belg. Math. Soc. Simon Stevin*, **3**(5), 63–84.
 Gioi, R. G. von, Jakubowicz, J., Morel, J.-M. & Randall, G. (2012). *Image Process. On Line*, **2**, 35–55.
 Gonzalez, R. C. & Woods, R. E. (2001). *Digital Image Processing*, 2nd ed. Boston: Addison-Wesley.

Gorfman, S. (2014). *Crystallogr. Rev.* **20**, 210–232.
 Hammersley, A. P. (2016). *J. Appl. Cryst.* **49**, 646–652.
 Hammersley, A., Svensson, S., Hanfland, M., Fitch, A. & Hausermann, D. (1996). *Intl J. High. Press. Res.* **14**, 235–248.
 Hart, M. L. & Drakopoulos, M. (2013). *arXiv*: 1311.5430.
 Hart, M. L., Drakopoulos, M., Reinhard, C. & Connolly, T. (2013). *J. Appl. Cryst.* **46**, 1249–1260.
 Heiney, P. (2005). *Newsl. IUCr Commission Powder Diffr.* **32**, 9.
 Hinrichsen, B., Dinnebier, R. & Jansen, M. (2006). *Z. Kristallogr. Suppl.* **2006**, 231–236.
 Johnson, I., Bergamaschi, A., Buitenhuis, J., Dinapoli, R., Greiffenberg, D., Henrich, B., Ikonen, T., Meier, G., Menzel, A., Mozzanica, A., Radicci, V., Sathapathy, D. K., Schmitt, B. & Shi, X. (2012). *J. Synchrotron Rad.* **19**, 1001–1005.
 Kanatani, K. & Sugaya, Y. (2008). *19th International Conference on Pattern Recognition (ICPR 2008)*, 8–11 December 2008, Tampa, FL, USA, pp. 1–4. IEEE.
 Kieffer, J. & Karkoulis, D. (2013). *J. Phys. Conf. Ser.* **425**, 202012.
 Lavina, B., Dera, P. & Downs, R. T. (2014). *Rev. Mineral. Geochem.* **78**, 1–31.
 Lutterotti, L., Vasin, R. & Wenk, H.-R. (2014). *Powder Diffr.* **29**, 76–84.
 Pătrăucean, V., Gurdjos, P. & Grompone Von Gioi, R. (2012). *Proceedings of the European Conference on Computer Vision (ECCV 2012)*, pp. 572–585. Springer.
 Qiao, Y. & Ong, S. (2007). *Pattern Recognit.* **40**, 1990–2003.
 Rajiv, P., Hinrichsen, B., Dinnebier, R., Jansen, M. & Joswig, M. (2007). *Powder Diffr.* **22**, 3–19.
 Tantau, L., Islam, M., Payne, A., Tran, C., Cheah, M., Best, S. P. & Chantler, C. (2014). *Radiat. Phys. Chem.* **95**, 73–77.
 Vogel, S. & Knorr, K. (2005). *Newsl. IUCr Commission Powder Diffr.* **32**, 23–25.
 Wong, Y., Lin, S., Ren, T. & Kwok, N. (2012). *21st International Symposium on Industrial Electronics (ISIE 2012)*, pp. 1105–1110. IEEE.