

A Thesis Submitted for the Degree of PhD at the University of Warwick

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/152350>

Copyright and reuse:

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it.

Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk



**Structural refinement of single crystals
using digital-large angle convergent beam
electron diffraction patterns**

by

Alexander James Morley Hubert

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Physics

July 2019

Contents

List of Tables	iv
List of Figures	v
Acknowledgments	xi
Declarations	xiii
Abstract	xiv
Abbreviations	xv
Chapter 1 Introduction	1
1.1 Parallel beam diffraction methods	3
1.1.1 Structure solution	4
1.1.2 Structure refinement	6
1.2 Convergent beam diffraction methods	7
1.2.1 Qualitative CBED	7
1.2.2 Quantitative CBED	9
1.3 Potential uses for D-LACBED	11
1.3.1 Symmetry analysis	11
1.3.2 Structure solution	13
1.3.3 Structure of this thesis	13
Chapter 2 From sample preparation to structural refinement	14
2.1 Sample preparation	15
2.2 Alignment	16
2.3 D-LACBED data collection	16
2.3.1 Focused beam and image shift correction	17
2.3.2 CBED stack collection	19

2.3.3	D-LACBED image processing	21
2.4	Simulation	24
2.5	Fit-index calculation	34
2.6	Minimisation algorithm	36
Chapter 3	Atomic refinement	37
3.1	Alpha-corundum	38
3.2	D-LACBED	39
3.3	CBED	45
3.4	D-LACBED g-vector series	46
3.5	Future work	53
Chapter 4	Debye-Waller factor refinement	55
4.1	Copper	56
4.2	Gallium Arsenide	62
4.3	Alpha-corundum and Indium Phosphide	69
4.4	Discussion	70
4.4.1	The Gibbs-Wilbraham phenomenon	71
Chapter 5	D-LACBED Sensitivity	72
5.1	Simulation Sensitivity	72
5.2	Convergence angle	77
5.3	Atomic coordinates	78
5.4	Thickness Determination	79
5.4.1	Indium phosphide	79
5.5	Accelerating voltage determination	85
5.6	Occupancy	87
5.7	Unit Cell	87
5.8	Debye-Waller factor	88
5.9	Point spread function	89
5.10	Simulation Bloch wave convergence	91
5.11	Summary	93
Chapter 6	Conclusions and future work	94
Appendix A	Corundum bond lengths and angles	97

Appendix B	Guide for D-LACBED alignment	99
B.1	Sample preparation	99
B.2	Alignment	99
Appendix C	Derivation of periodic Schrödinger equation	101
Appendix D	Listing of D-LACBED code	103
D.1	Calibration 1	103
D.2	Calibration 2	108
D.3	Collect	110
D.4	Process	112
D.5	Align	122
Appendix E	In Memoriam	133
Bibliography		134

List of Tables

5.1	maximum delta values for a selection of g-patterns using the A221 dataset (Fig 3.2). Cyan and red highlights the $\bar{1}14$ and 012 systematic rows respectively. Units are in standard deviation	75
5.2	Max deltas from Fig 5.1 adjusted to 1 standard deviation for comparison of parameters	76
A.1	Bond angle comparison between this thesis and reference values (Kondo et al.) for the AlO_6 octahedra seen in fig A.1. The top triplet bond lengths are: 1.850 Å (This study) & 1.854 Å (Kondo). The bottom triplet bond lengths are: 1.976 Å (This study) & 1.971 Å (Kondo) .	98

List of Figures

1.1	top) simplified ray diagrams for named electron diffraction techniques. The back focal plane (BFP) is conjugate with the screen. bottom) example diffraction patterns for each ray diagram. The purple, orange and black discs correspond to their corresponding colours in the ray diagrams.	2
1.2	a) schematic of the 1937 convergent-beam electron interference instrument designed by Möllenstedt b) specimen holder and c) convergent beam diffraction pattern of a 10nm thick silver specimen captured with the aforementioned instrument.	8
1.3	CBED pattern of RKTP in the [010] direction	11
1.4	Dark-field TEM and D-LACBED analysis of beam induced domain motion on a RKTP sample. In the top frame, a surface dimple marks the location of a former domain which has retracted off of the sample's top edge after uniform irradiation. D-LACBED analysis confirms the domain is continuous across this region. The bottom panel shows the retracted domain. D-LACBED analysis confirms a symmetry reversal (vertical mirror) across the imaged domain wall.	12
2.1	Schematic showing the experimental procedure to obtain a refined structure of a single crystal specimen.	15
2.2	a) Schematic showing position of an array of 000 CBED discs during the first tilt calibration raster b) Schematic showing position of the 000 CBED disc in the serpentine raster used in D-LACBED data collection. Colours simply indicate different CBED disc positions. Numbers next to discs indicate its place in the series. The orange box in (a) shows the area of shift corrections (utilising linear interpolation) used for the raster in (b). Scalings are approximate.	18

2.3	Sample recording of image plane shift positions on the camera from D-ED_Calibration2 script	18
2.4	Figure 2 from Beanland et al. [7] (a) Sixteen CBED patterns from [110] silicon with varying beam tilts. The 000 (red), 111 (blue) and 220 (yellow) beams are highlighted in each. (b) Digital reconstruction of LACBED patterns from many individual CBED patterns, highlighting the components from the patterns in (a). (c) The on-axis CBED pattern.	20
2.5	a) Average CBED pattern, b) raw 000 D-LACBED g-pattern and c) montage from [2 $\bar{2}$ 1] alpha-corundum)	23
2.6	Zone axis CBED patterns from [2 $\bar{2}$ 1] alpha-corundum a) without and b) with background subtraction. The dynamic range in both images has been equally shifted to low intensity values to see the diffuse scattering	24
2.7	a) Raw, uncropped D-LACBED g-pattern vs. b) symmetrically averaged and distortion corrected g-pattern.	25
2.8	top) Schematic of the simulation set up in <i>felix</i> over the central 2D pixel line of a D-LACBED image (red line on bottom images). The blue and orange lines indicate the direction of incident wavefunctions (with wavevectors \bar{K}_0^1 and \bar{K}_0^2) for the 1 st and 2 nd pixels respectively. After transmission through the sample (see Figure 2.9) with thickness t , the lines indicate the exit wavefunctions with wavevectors \bar{k}_{g0}^1 and \bar{k}_{g0}^2 for $g0$ ($g = 0$), \bar{k}_{g1}^1 and \bar{k}_{g1}^2 for $g1$. The purple lines show the p^{th} wavefunction in each case. bottom) Sample simulated D-LACBED g-patterns of [001] Copper. The red line is analogous to the red line in the top image. The green and yellow highlights show example assignments of pixel allocations for each processor up to the n^{th} core	30
2.9	Reciprocal space representation of the Bloch wave problem through an infinite potential for the two-beam case. n represents the vector normal to the specimen surface. Other variables are defined within the text	31

2.10	a) simulated (top) and experimental (bottom) 000 D-LACBED images of Copper in the [001] direction. The red and blue lines indicate the pixels used for the line plots in b,c and d. b) line plot (left) of the raw D-LACBED images with associated histogram (right). The red and blue dotted lines indicate mean values of 9733 and 7770 for the simulated and experimental images respectively. c) line and histogram plot from b, but with each image mean subtracted from the intensity values. The black dotted line indicates zero mean for both images d) line and histogram plot from c, but with the intensity values divided by the standard deviation of each image. The black dotted line is again the zero mean.	35
3.1	Crystal structure of α -Al ₂ O ₃ . The symmetry of the space group $R\bar{3}c$ dictates that the aluminium atom can only move along the c direction with the oxygen atom only along the a or b vectors dependent on the Wyckoff position	39
3.2	D-LACBED montage of alpha-corundum in the $[2\bar{2}1]$ direction (A221 dataset). Each pattern has an angular range of 55.14 mrad.	41
3.3	fit index f as a function of x_O and z_{Al}	42
3.4	α -Al ₂ O ₃ $[2\bar{2}1]$ $\delta I/\delta x_0$ to G_3	43
3.5	α -Al ₂ O ₃ $[2\bar{2}1]$ $\delta I/\delta z_{Al}$ to G_3	43
3.6	α -Al ₂ O ₃ $[2\bar{2}1]$ residual (using delta method) of best-fit simulation to experiment.	44
3.7	Refined values of x_O and z_{Al} in α -Al ₂ O ₃ from studies using X-ray (black), neutron (blue) and electron (red) radiation. Error bars for This Study too large for graph (see text)	44
3.8	The A221 dataset up to second order. Blue and orange squares show the cropped region used to approximate a single CBED dataset; the intensity within the orange squares is low and subsequently were not used in refinement.	45
3.9	α -Al ₂ O ₃ $[2\bar{2}1]$ D-LACBED data colour coded for various G_x : G_0 = red, G_1 = purple, G_2 = blue, G_3 = orange	47
3.10	Comparison of refined z_{Al} and x_O values between CBED (0.9%), G_0 (0.9%), G_1 (0.6%) and G_2 (0.6%) datasets. Fit values, f , are in brackets	48
3.11	left) Set of four datasets right) Their corresponding fit spaces. An evenly spaced grid of 32 by 32 simulations were carried out and a 2D spline applied	49

3.12	Corundum projection down the $[2\bar{2}1]$ direction. 036 lattice planes have been overlaid.	50
4.1	Room temperature D-LACBED from Cu (a) Nine patterns from Cu $[001]$, angular range 41.3 mrad. The cyan lines highlight the 400 type deficit lines, the yellow oval highlights a 'flower petal' band (see text) (b) Thirteen patterns from Cu $[114]$, angular range 44.2 mrad. The red lines highlight the $1\bar{5}1$ type deficit features. The yellow hexagon highlights the 'lozenge' (see text). All patterns are normalised for display here to the visible display range and have applied gamma 1.5 to allow features in darker parts of the image to be seen more easily.	57
4.2	Upper: the experimental 000 D-LACBED patterns and best fit simulations at different temperatures for Cu $[001]$. Centre: fit indices f (all patterns) for a range of simulations at each temperature. Lower: simulated 000 patterns (1) – (4) and the difference between them, where yellow shows a decrease, and blue an increase, in relative intensity with increasing Debye-Waller factor B	59
4.3	Upper: the experimental 000 D-LACBED patterns and best fit simulations at different temperatures for Cu $[114]$. The small purple circles show the fading of the $1\bar{5}1$ deficit feature as temperature increases Lower: fit indices f (all patterns) for a range of simulations at each temperature.	60
4.4	Experimental determinations of Debye-Waller factor B in copper D-LACBED data from $[001]$ (solid stars, green) and $[114]$ (hollow stars, red). Errors are smaller than the data points, typically $\sim 0.01\text{\AA}^2$. Many previous measurements of B using X-ray diffraction and the Mössbauer effect are shown in black and white (from ref. [242]) . . .	61
4.5	Eighty-five D-LACBED patterns from $[1\bar{1}0]$ GaAs. Left: room temperature (29°C). Right: 200°C. Each pattern has an angular width of 46.2 mrad	63

4.6	a, b: fit index f (49 unique patterns) for [110] GaAs data at room temperature and 200°C as a function of B_{Ga} and B_{As} . c: Experimental measurements of B_{Ga} and B_{As} . Green and gray = X-ray diffraction; purple = electron diffraction; orange = theory . Unattributed measurements are those collated in ref.. d,e: $\delta I/\delta B_{Ga}$ and $\delta I/\delta B_{As}$ in units of standard deviation. (see also Figs. 4.7 & 4.8) f: the residual between best-fit simulation and experiment, calculated in the same way as d and e. (see also Fig. 4.9)	65
4.7	Enlargement of Fig 4.6d ($\delta I/\delta B_{Ga}$ in units of standard deviation)	66
4.8	Enlargement of Fig 4.6e ($\delta I/\delta B_{As}$ in units of standard deviation)	67
4.9	Enlargement of Fig 4.6f (Residual between best-fit simulation and experiment)	68
4.10	X-ray, neutron, theoretical and electron (black, green, orange and purple respectively) Debye-Waller factor measurements for InP (left) and Corundum (right) at room temperature. The Pre-1990 XRD values (grey) were taken from Ballirano et al.	69
5.1	delta images for a selection of g-patterns using the alpha-corundum dataset in Chapter 3 (A221) over a range of parameters (see text for details). The scale bar units are given in standard deviation.	74
5.2	D-LACBED experimental dataset of InP from the [001] zone axis. The figure contains only the positive index g-patterns of the dataset.	81
5.3	left) Simulations of InP D-LACBED patterns over a range of thicknesses and their corresponding fit values to the experimental image (See Fig. 5.2) right) Figure 10 taken from Muller et al. of a CBED thickness scan of GaAs data. The inset shows the $\bar{4}00$ disc.	82
5.4	DI/DT plot of InP. Units of scale bar are in standard deviation	83
5.5	InP DI/DT 022 g-pattern from Fig 5.4. Clear and faint delta thickness fringes are prevalent throughout the background of the image (e.g. the darkened area around the "022" label)	84
5.6	left) Experimental central CBED disk of the $[37\ 37\ 4]$ zone axis of silicon. The thin cyan lines highlight the location of the HOLZ lines. The thin red line is a crossing point guide. right) Simulated Kikuchi and HOLZ line pattern at 200kV under the same conditions. In both figures lengths A & B correspond to the ratios of HOLZ line crossing points $(7\ 9\ 11)$ & $(9\ 7\ 11)$, $(10\ \bar{1}0\ 8)$ & $(\bar{1}0\ 10\ 8)$, $(\bar{1}\ 3\ \bar{1}3)$ & $(3\ \bar{1}\ \bar{1}3)$ (from left to right)	86

5.7	A and B simulated ratios (see Fig 5.6) at set voltages using the central disk of the silicon [37 37 4] zone axis. Black line is a least squares fit. Blue dashed lines show measured experimental ratio and associated voltage (200.74 ± 0.53 kV).	86
5.8	a) Bright field image at 6000x magnification of the JEOL TM 2100+ beam stopper. b) it's associated binary image after thresholding (up-scaled)	89
5.9	a) Average of 1D MTF measurements from beam stopper images (Fig. 5.10a) up to the maximum spatial frequency (ν_s) of 1.0 and b) it's associated Fourier transformed 1D PSF function (both the MTF and PSF are rotationally symmetric)	90
5.10	Simulation tests for the [001] and [114] Cu D-LACBED data of Fig. 4.1, showing the maximum difference (arbitrary units using 16-bit images) in intensity of any pixel with respect to a reference simulation using 250 Bloch waves. Each line is a least squares fit which is extrapolated to meet the desired criterion for accuracy (horizontal dotted line).	92
A.1	AlO ₆ octahedra within corundum structure - see Fig. 3.1. The bonds within each of the top and bottom triplets are equal in length	98

Acknowledgments

Dear Reader,

Behind the words on the following pages is a story. It is one of desperation - to feel any semblance of peace, any freedom of thought; to recapture the mental gifts I once possessed. It is one of pain – the agonising wait for seconds to turn into minutes, days and months. It is one of frustration – at the whole situation, the culture of misunderstandings flying at you left, right and centre.

But most importantly it is also one of compassion. In the sanctuaries gluing the scientific community together, I have been met with kindness I did not think existed. The generosity from those within the physics and microscopy group throughout my time at Warwick. From the ear Reza lent me countless times on the bus ride home, the conversations over coffee with James and then Sam. From Susan and her determination to stand with me in my push for survival. Tom and his courageous actions without question. From the friends I have made in the physics social group, led passionately by the hard work of Will. The members of Mind aware and Korfbal club. From Ali and Saede for their friendship and hospitality. The friendships I have made with James, Alex, Cantug, Mark, Liz, Rachel, Celia, Andy and Naomi. My sister for her continued love of life. My brother for his support from day one. My closest friend of all, Fynn.

When I was applying for PhDs, a piece of advice kept re-surfacing. The most important part of the PhD is the student-supervisor relationship. Oh my, were they correct. Over these tumultuous years Richard and I have definitely pushed each other's buttons. But in doing so we have grown together. His faith in my ability

has been undeniable, especially during the dark days, when the vast majority would have immediately cast me aside. For that I cannot thank him enough. I can only hope this piece of work shows the extent of our fruitful partnership.

Every child needs their mother. If it were not for her love and seemingly boundless perseverance, for her fight when I could not, for her un-selfishness and patience, I would not be here. Period. I do not think I will ever realise how lucky I am, that she never let me fall down inescapable paths. I am grateful to an extent I don't think she will ever fully realise. Every child needs their mother.

If I was granted one wish, it would be to see more of the compassion I have witnessed out in the open. In the daily workings of research life, the pubs and coffees. In meetings with professors and experts. In the pointless politics that surrounds a research career. In talks, reviews and papers.

I wish for compassion because, dear reader, we do not know the story behind the words.

Declarations

I declare that the work presented in this thesis carried out by the author under the supervision of Dr. Richard Beanland at the Department of Physics, University of Warwick. The research reported here has not been previously submitted, wholly or in part, at this or any other academic institution for admission to a higher degree.

Some of this work presented in this thesis has been previously published by the author in the following chapters

- In Chapter 1, Figure 1.4 has been published in:
J. Hart, S. Liu, A. Lang, A. Hubert, A. Zukauskas, C. Canalias, R. Beanland, A. Rappe, M. Arredondo and M. Taheri, “Electron-beam-induced ferroelectric domain behavior in the transmission electron microscope: Toward deterministic domain patterning,” *Physical Review B*, vol. 94, no. 17, 2016
- In Chapters 3 & 4 the experimental data and findings have been published in:
A. J. M. Hubert, R. Römer, and R. Beanland, “Structure refinement from ‘digital’ large angle convergent beam electron diffraction patterns,” *Ultramicroscopy*, vol. 198, pp. 1–9, 2019.

All of the work in this thesis was undertaken by the author apart from the following:

- In Chapter 4 the experimental data and some findings from Figures 4.5-4.7 were undertaken by R. Beanland
- The code in Appendix D was written by R. Beanland

Abstract

We explore the capability of digital-large angle convergent beam electron diffraction (D-LACBED) data for the structural refinement of single crystals. To achieve this, we use three materials as test cases. We use corundum for atomic position refinement, copper and gallium arsenide for Debye-Waller factor (DWF) refinement. D-LACBED patterns are found to be extremely sensitive to atomic position, within 0.4 pm of reference X-ray values. The patterns are less sensitive to DWF (using the independent atom model - IAM) but nonetheless give good agreement to X-ray and Mössbauer radiation values for copper. We find the IAM to be insufficient for accurate refinement of gallium arsenide due to the influence of previously suggested strong anharmonicity and bonding within the material. Finally, we use simulation to explore the sensitivity of D-LACBED patterns through most refineable structural parameters, providing context to the aforementioned results. During the analysis we see that higher g-vector patterns within the D-LACBED data may be more sensitive to structural parameters in general.

Abbreviations

1D	one dimensional
2D	two dimensional
2D	two dimensional
ACR	atomic coordinate refinement
CBED	convergent beam electron diffraction
CCD	charge coupled device
cRED	continous rotational eletron diffraction
D-LACBED	digital - large angle convergent beam electron diffraction
DWF	Debye-Waller factor
DM	Digital Micrograph TM
EDT	electron diffraction tomography
HOLZ	higher-order Laue zone
IAM	independent atom model
LACBED	large angle convergent beam electron diffraction
MTF	modulation transfer function
PSF	point spread function

PED	precessional electron diffraction
PEDT	precessional electron diffraction tomography
PACBED	position averaged convergent beam electron diffraction
QCBED	quantitative convergent beam electron diffraction
RED	rotational electron diffraction
RKTP	rubidium doped potassium titanyl phosphate
SAED	selected area electron diffraction
TEM	transmission electron microscope/microscopy
ZAP	zone axis pattern
ZNCC	zero-mean normalised cross correlation
ZOLZ	zero-order Laue zone

Chapter 1

Introduction

“To understand [why black-body radiation is connected to electron diffraction] it is necessary to go back to the early 1920’s, perhaps the most exciting period in the whole of physics, and I venture to say the most baffling to those who experienced it”

- Sir George Thomson

The Early History of Electron Diffraction [1]

“Where one door is shut, another is opened”, the wise words first written by de Cervantes in his seminal book Don Quixote [2]. If only the reality of a researcher was based upon a two-door system. Behind the open door lay a myriad of hidden passageways leading to dead ends, more doors (some locked, bricked up or broken) and the odd u-turn. Generally, when a new technique arrives which unlocks more information it can take time to figure out how to use it. It is not clear which paths will lead to fruition and which will result in a dead end. Perhaps the most well-known example is the successful sequencing of the human genome. It may take decades before this achievement reaps its full scientific benefit.

While convergent beam electron diffraction (CBED) has been an extremely useful technique since its inception in the late 1930s (Section 1.2.2), it has a significant limitation. The geometry of selected area electron diffraction (SAED) is based upon a parallel beam, created using an aperture in a conjugate image plane (a virtual aperture - Fig 1.1 SAED), resulting in a spot pattern. The geometry of CBED however, is based upon a convergent beam, creating discs (Fig. 1.1 CBED). While this means the technique can probe nanometre areas of the sample, above a certain angle of convergence the intensities within portions of each disc will overlap. This can be useful for techniques like coherent - CBED [3] or position averaged CBED (PACBED) [4]. It is however generally seen as a problem (the ‘overlap problem’).

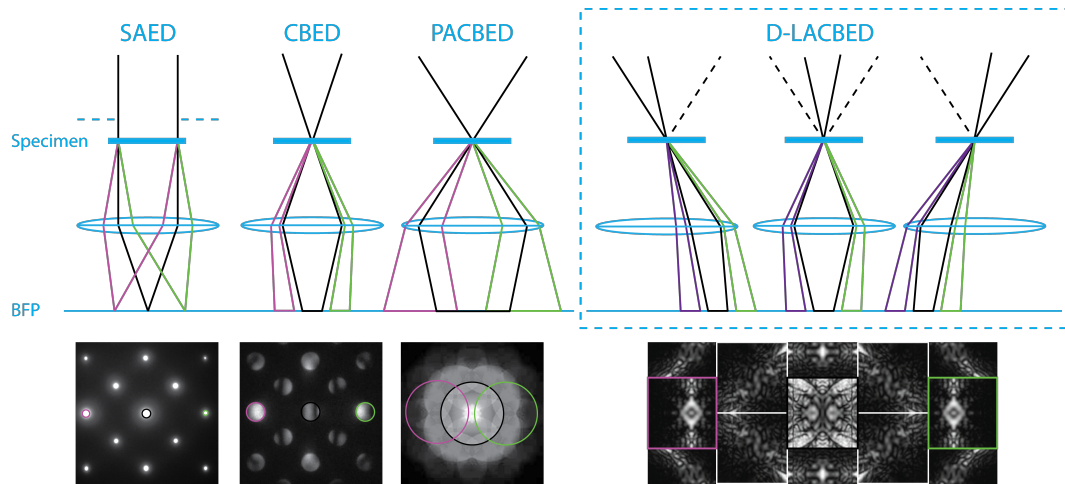


Figure 1.1: top) simplified ray diagrams for named electron diffraction techniques. The back focal plane (BFP) is conjugate with the screen. bottom) example diffraction patterns for each ray diagram. The purple, orange and black discs correspond to their corresponding colours in the ray diagrams. The (PACBED) pattern has been taken from LeBeau et al. [4]

Tanaka, Ueno and Harada presented a partial solution using the large angle convergent beam electron diffraction (LACBED) technique [5]. By converging the beam above or below the specimen they could select one beam (diffracted or transmitted) using the selected area aperture. This technique has been particularly successful at studying imperfect crystal structures [3]. However, the technique required a skilled microscopist, and could only record one g-vector disc, or g-pattern, at a time. A full set of patterns could be recorded but the process would be very time consuming, and the operator could never be sure each g-pattern originated from the same area of sample.

A full solution was found through two, nearly parallel studies. Koch used hardware control of the microscope to introduce the large-angle rocking-beam electron diffraction technique [6]. Beanland et al. [7] on the other hand, used computer software control. By automatically tilting, recording and stitching a set of CBED patterns together over a wide convergence angle, a full LACBED pattern (containing all g-patterns) could be stitched together – the digital-LACBED (D-LACBED) technique [7] (see Chapter 2). The technique probes a vastly superior portion of reciprocal space compared to CBED patterns (Fig 1.1), is easy to use and patterns can be collected in minutes.

This thesis will explore some of the possible applications of D-LACBED's expanded dataset to determine paths that may lead to the most 'open doors'. In

order to understand which of the methods we should initially explore, which door to open first, we must look back at the history of electron diffraction. This will show how past examples of information expansion led to new applications.

1.1 Parallel beam diffraction methods

As for many notable discoveries in science, that of electron diffraction was somewhat unintended [1]. Up until the mid-1920's electrons were predominantly thought of as particles due to the work by JJ Thomson [8] and other researchers [9, 10] (having previously been thought of as waves ie. cathode rays). It was through the work of de Broglie [11, 12] on black-body radiation where he postulated wave-particle duality (using the now famous de Broglie relation) that attention fell on the possible wave-particle duality of electrons. After the accidental creation of large single crystals in very low energy electron reflection of nickel producing diffraction like patterns in 1925 [1] and the resulting proposition for the wave-nature of electrons by Elsasser [13], Davisson and Germer [14, 15] published the first evidence of electron diffraction by reflection using low energy electrons. This was confirmed by G. P. Thomson and Reid [16] as well as Kikuchi [17, 18] for high energy electron diffraction. Each was later described theoretically through dynamical diffraction theory by Bethe [19] and extended by Blackman [20] for the reflection and transmission case respectively. Both pieces of experimental and theoretical work led to two separate fields: Low energy electron diffraction for the characterisation of crystal surfaces [21] and high energy electron diffraction (now part of transmission electron microscopy or TEM) for bulk single crystals [22] which we are interested in for this thesis.

Early transmission electron microscopes (TEMs) were rudimentary, custom made machines [23]. Selected area diffraction was first proposed by Boersch [24, 25] in 1936 to show Abbe's theory of optics held for the electron microscope. It took over a decade before the selected area aperture was introduced to TEMs [26, 27]. Around the early 1950s, the aperture was built into commercial transmission electron microscopes meaning SAED was used to study crystals for the first time [28–33]. This allowed diffraction through individual, isolated grains of the specimen, leading to its now very broad usage in electron microscopy.

The advent of SAED and CBED (see Section 1.2) meant unit cell scale structural analysis of single crystals was now possible. Pioneered by Vainshtein and colleagues [34] from the 1940's to 70's, numerous unknown structures of both inorganic and organic substances were solved by several electron diffraction methods through structure solution.

1.1.1 Structure solution

Originating from X-ray crystallography a structure solution method uses the intensity and position of the experimental diffraction intensities to directly obtain the electron density of the crystal. This is possible under the kinematic approximation

$$I_{\bar{g}} = |F_{\bar{g}}|^2 \quad (1.1)$$

which states that the integrated intensities of the diffraction peaks, $I_{\bar{g}}$ (where \bar{g} is a reciprocal lattice vector) are proportional to the square modulus of the X-ray structure factors $F_{\bar{g}}$ which are the Fourier components of the electron density in the crystal structure. Because the phase of the structure factors cannot be directly measured, the well-known phase problem in crystallography, several phasing methods exist to extract them, for example: direct or Patterson methods [35]. In X-ray diffraction, where the kinematic approximation applies, these have been used to extract complete structure factors allowing electron density maps and therefore (if the space group is known – Section 1.2.1) atomic positions to be obtained.

These methods can also be applied to SAED patterns to extract X-ray structure factors. In electron diffraction the Fourier components $V_{\bar{g}}$ describe the electron potential rather than the electron density (see Section 2.4). However, they can be converted to X-ray structure factors $F_{\bar{g}}$ using the Mott-Bethe relation in reciprocal space or to an electron density function using Poisson’s equation [36] in real space. However, since electrons mostly scatter dynamically (i.e. more than once) due to their much stronger interaction with matter than X-rays [37], only quasi-kinematic intensities are possible for materials thicker than one atomic layer – and even then only for thin specimens. Since a general direct relation between dynamical intensities and structure factors does not currently exist [38], structure solution using electron diffraction fell out of favour during the latter half of the 20th century, despite the success of Vainshtein.

Two developments near the end of the 20th century reinvigorated the desire for structure solution from electron diffraction spot patterns. The first was the introduction of the zone axis precessional electron diffraction (PED) method by Vincent and Midgley [39]. Through simple hardware add-ons they were able to collect integrated electron spot patterns by tilting the parallel incident beam through a hollow cone around a zone-axis. By the early 2000’s it was shown that PED data could be used within direct methods [40] and robust examination of the technique had taken place [41]. To the point it was subsequently commercialised as an add-on system for TEMs [42]. It is now a reliable technique for structure solution as well

as a number of other analysis methods [43–45].

The second development was due to the widespread introduction of personal computers and charge coupled device (CCD) cameras to transmission electron microscopes. Processing of data could now happen as the experiment was taking place. In the mid-1980s, Hovmöller and collaborators demonstrated it was possible to use image processing of high resolution electron microscopy data [46] to obtain atomic positions of several inorganic crystals [47,48]. It was later shown that the technique could be more accurate if combined with diffraction data [49] for both light [50] and heavy elements [51] including complex structures [52]. However, complicated structures require multiple datasets with separate projections in order to obtain 3D information of the sample. It is very time-consuming and requires a highly skilled operator [53]. X-ray diffractometers on the other hand require the opposite. They are easy to use and 3D information can be obtained in minutes. If electron diffraction was ever to compliment X-ray diffraction a much easier to use technique was required.

In 2007 Kolb et al. [54] demonstrated electron diffraction tomography (EDT) [55]. Using computer control of the microscope the sample holder is tilted through a predetermined set of angles. Scanning transmission electron microscopy is then used at every tilt interval to take an image. This means the beam can be re-shifted back to the same area of sample. The 3D reciprocal space data gives better 3D real space data as well as averaging out some of the dynamical effects (as long as the specimen is thin). The introduction of tomography to electron diffraction initiated a boom (hundreds of publications) in structure solution studies for a wide range of complex materials including zeolites [56–60], porous materials in general [61], organics [62,63] (such as pharmaceuticals [64]) as well as many more [65–67]. Two other prominent tomographic techniques were developed after Kolb’s initial paper, precessional electron diffraction tomography (PEDT) [68] and rotational electron diffraction (RED) [53]. PEDT uses goniometer tilts in between PED pattern acquisitions (also effective for thick samples), RED rotates the goniometer along a specific rotation direction between zone axes utilising beam tilt for even finer sampling. Development has continued on all three techniques, each becoming more precise and robust as the years progress [69,70]. New variations on tomographic methods are also continuously being made [71]. It is now the ‘go to’ technique for structure solution of nanostructures unable to be evaluated by X-ray powder diffraction.

1.1.2 Structure refinement

Studies utilising structural refinement using parallel beam methods are far less common than structure solution studies. This is the next stage of structural determination once the atomic potential is known. A theoretical model of the potential (the structure factors) is used to simulate a complimentary diffraction pattern usually using either the Bloch wave or multislice method (see Chapter 2). Through minimisation algorithms via a fitting function between the experimental and simulated patterns certain parameters of the model can be found. For instance, the structure of the atomic potential can be refined (atomic coordinates - Chapter 3, temperature factors - Chapter 4, unit cell and occupancy - Chapter 5) usually used to discover deviations from previously known structures or new structural phases. Specific structure factors can be refined to ascertain deviations from the modelled neutral atom values - therefore uncovering the bonding influences between atoms. Finally, experimental conditions can be refined, for instance thickness of the sample (Chapter 5).

The small total number of intensities in parallel beam methods, even with EDT, does not provide as many as convergent beam methods where refinement studies are more prevalent [72] (This Section compared to Section 1.2.2). EDT methods do sample a wider angular range. The reason for the prevalence of CBED to be used for structural refinement is unclear. Perhaps distortion correction is easier with the features present in CBED patterns. Nevertheless, since dynamical intensity varies dramatically over a wide range of variables and experimental conditions (see Chapter 5) the greater number of intensities over wider angles seems likely to lead to a greater achievable accuracy.

Dynamical refinements are necessary for both SAED and PED patterns as intensities can change significantly over many parameters and experimental conditions (see Chapter 5). Kinematical refinements have been carried out using PED data to give reasonable results ([73] and references therein), however it is clear dynamical intensity calculations are required [41, 73–75]. While there have been studies with successful dynamical refinements of SAED [76, 77] and PED [78] data using multislice [79] and Bloch wave [80] simulations throughout the 2000s, the most thorough study on dynamical refinement was undertaken by Palatinus et al in 2015 [81, 82]. They refined, using PEDT data, the atomic positions and occupancies of several complex inorganic single crystals, including kaolinite which has up to 39 degrees of freedom. With average deviations of 0.02\AA compared to reference X-ray data. This work, using the greater amount of information available in tomographic data lead to several recent successful refinements of previously unknown structures and phases [83–86].

1.2 Convergent beam diffraction methods

Due to observations of Kossel patterns from X-ray diffraction and the divergence of a parallel beam to wider angles to form Kikuchi lines [17], Kossel asked Möllenstedt to build a microscope capable of producing electron interference from a convergent beam for his master's thesis [87]. Möllenstedt achieved the first CBED patterns in 1938 [88] with state of the art instrumentation, including a wine bottle electron gun [87, 89] (Figure 1.2).

Use of the technique was very stagnant in the early years due to the poor data quality – caused in particular by poor vacuums leading to high sample contamination by the intense convergent beam [72]. Nevertheless, the dynamical diffraction theory including two and three beam methods¹ developed throughout the 1950's and 1960's, with an emphasis on symmetry determination. Perhaps the most powerful method of CBED.

1.2.1 Qualitative CBED

The dynamical scattering of electrons combined with the convergent beam geometry produces discs of varying intensity. Unlike spot pattern methods, qualitative analysis of the symmetry within each individual disc as well as the pattern as a whole is possible. This has many advantages, for example the easy determination of non-centrosymmetric crystals [90, 91].

Determining the symmetry of a crystal helps to confine the possible atom locations for structure solution. Work began to firstly determine how to take the 2D symmetries present in a CBED pattern and relate it to the 3D symmetry within the crystal (point group). Through the work of Goodman [92], Tinnappel [93] and the comprehensive study by Buxton [94] it was shown any crystallographic point group could be found from at most two or three orientations of CBED patterns if completed systematically [95].

Under the correct conditions, translational symmetry elements, only viewable due to dynamical diffraction, can be characterised using CBED patterns. Gjønnes-Moodie lines [96–99] form from the destructive interference when screw axes, glide planes or both align to produce zero intensity within kinematically forbidden reflections. The characterisation of these symmetry elements helps to determine the vast majority of the 230 space groups [100, 101].

The ability to probe very small areas meant that symmetries could be found

¹A comprehensive review of two and three beam dynamical theory can be found in Spence and Zuo [36]

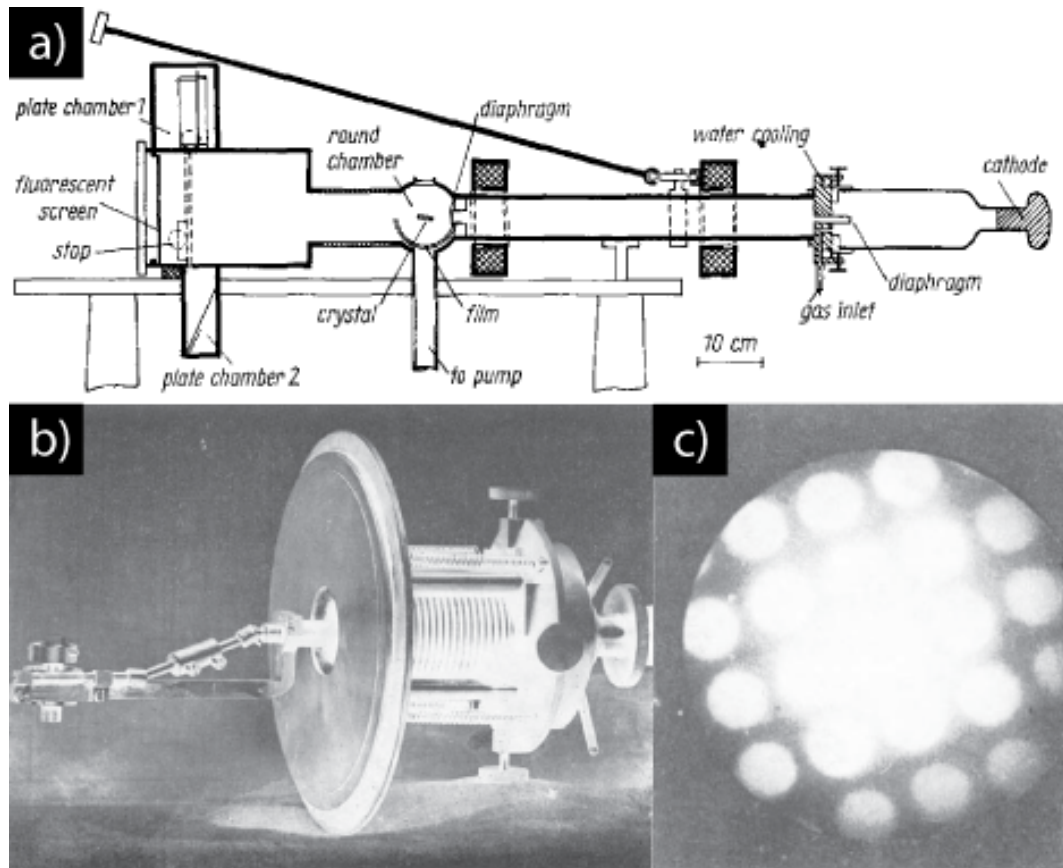


Figure 1.2: a) schematic of the 1937 convergent-beam electron interference instrument designed by Möllenstedt b) specimen holder and c) convergent beam diffraction pattern of a 10nm thick silver specimen captured with the aforementioned instrument. All figures taken from [87]

of separate grains leading to a whole multitude of studies [102] such as twinning, phase transformations and piezoelectric/pyroelectric effects [95]. Studies involving symmetry determination using CBED are still present at the time of writing [103–105], albeit with much less abundance. Recent advances include quantitative characterisation using image processing methods [106].

1.2.2 Quantitative CBED

The Quantitative use of CBED patterns began with the work of Macgillavry [107] using two-beam dynamical theory to attempt extraction of structure factors from Möllenstedt's original CBED patterns (see Fig. 1.2). After numerous attempts of structure factor extraction from CBED patterns using two-beam theory [108], a new method emerged in the late 1960's. Uyeda et al. [109] and Hoier [110] used the ratios between Kikuchi lines in Kikuchi patterns to determine the accelerating voltage of the transmission electron microscope (see Chapter 5). As a consequence of this work it was discovered that the intensities at the Bragg condition of second order conditions becomes vanishingly small at specific voltages [111, 112]. The critical voltage technique has been used for scattering factors [113, 114], lattice parameter changes [114] and Debye-Waller factor (DWF) determination [115]. It is extremely sensitive to low order structure factors which can be extracted using three beam theory [36]. This would eventually lead to the structural refinement of low-order structure factors using CBED patterns in the 1980's and 1990's.

One of the first comprehensive attempts to solve a complete structure (AuGeAs) using CBED was by Vincent et al. [116, 117]. They determined the space group from CBED patterns, the lattice parameters from spot patterns and then higher-order Laue zone (HOLZ) lines for better accuracy. Using a trial and error methodology in comparison to a structure of the same space group and lattice parameters (NiP₂), they ruled out potential structures with bond length arguments and a technique of quasi-kinematic HOLZ reflection refinement (developed by the Bristol group in the previous few years) until the final structure was found. Incidentally it was this search for quasi-kinematic intensities in electron diffraction that led to the PED method.

As computer memory, speed and cost decreased during the late 1980s and early 90s, interest in refining structure factors using quantitative CBED rose. Spence and Zuo produced a number of comprehensive studies using the Bloch wave method to simulate and the simplex method to automatically refine structure factors from CBED data. They used the systematic geometry and the sensitivity of three-phase

invariants² to extract full structure factors [119]. Bird and Saunders used an alternative zone axis orientation with a quasi-Newtonian algorithm to achieve a similar feat [120]. Each method used line scans or ‘rocking curves’ extracted from the CBED discs.

It was perhaps the work of Høier et al. [121] and Tsuda et al. [122] in the mid 1990’s which showed the potential the Quantitative CBED (QCBED) method had when compared to other electron diffraction refinement techniques. Høier and colleagues were the first group to use a two-dimensional image of intensities in a multi-parameter refinement. Tsuda and Tanaka produced a comprehensive study on strontium titanate refining the atomic coordinates and DWFs – instead of structure factors themselves. Both studies showed the QCBED technique had the potential to be a panacea method i.e. the broad range of dynamical intensities provided in a CBED pattern could be used to determine many different structural properties of a crystal. In contrast to the previous methods of refinement (critical voltage and HOLZ reflection) where the limited amount of information available meant only specific parameters could be measured.

To the present day, the QCBED technique has been used to determine a range of quantitative parameters ([123] and references therein) including bonding [124–130], lattice parameters [123], atomic positions [131, 132] and Debye-Waller factors [133–136]. However, the method has not ‘taken off’ in the same fashion as tomography in parallel beam methods. Saunders [137] postulates that zone axis CBED patterns do not cover enough of reciprocal space to refine the required number of structure factors to build a complete picture of the bonding charge distribution. The ‘overlap problem’ only adds to this issue, restricting the technique to materials with small lattice parameters [36].

Perhaps the most tantalising prospect for Quantitative CBED lies in direct inversion of the patterns [138]. First identified by Bird et al. [139] and Peng et al. [140] a general method to obtain structure factors (with phase intact) from the intensities was proposed almost at the same time by Allen [141] and Spence [142]. While progress has been made since [143–145], it is currently unable to be used practically [38].

²For the interested reader, three-phase invariants have also been used in CBED structure solution to directly determine potential structure factor phases – a comprehensive review is given in Guo [118]

1.3 Potential uses for D-LACBED

From the previous two sections we can conclude three initial routes for D-LACBED investigations. Symmetry analysis, structure solution and structure refinement.

1.3.1 Symmetry analysis

In materials with even moderate lattice parameters (>1 nm) the diameter of the CBED discs may be too small to qualitatively determine the symmetry. For example, in a recent study of rubidium doped potassium titanyl phosphate (RKTP) [146] D-LACBED was used to confirm the electron beam had created a domain that was a twin of the original structure. From the CBED pattern in Fig. 1.3 we see that the moderately long lattice parameters ($a = 1.28$ nm, $b = 0.64$ nm, $c = 1.06$ nm) in RKTP produces a dense reciprocal lattice in the $[010]$ direction (minimum g-vectors: $200 = 0.16 \text{ \AA}^{-1}$, $001 = 0.10 \text{ \AA}^{-1}$) with limited CBED disc area.

It is therefore not only difficult to determine any symmetry within the pattern but also the centre of the zone axis. This can be rectified using the D-LACBED technique.

As shown in Fig. 1.4 D-LACBED patterns were taken on the bulk and newly created domain, where the flipping of the entire pattern (vertical mirror) clearly displays twinning. D-LACBED has an advantage over LACBED in this regard. The technique is arguably easier to use and the entire set of g-patterns are collected at once, meaning symmetry concerning the entire pattern can be analysed.

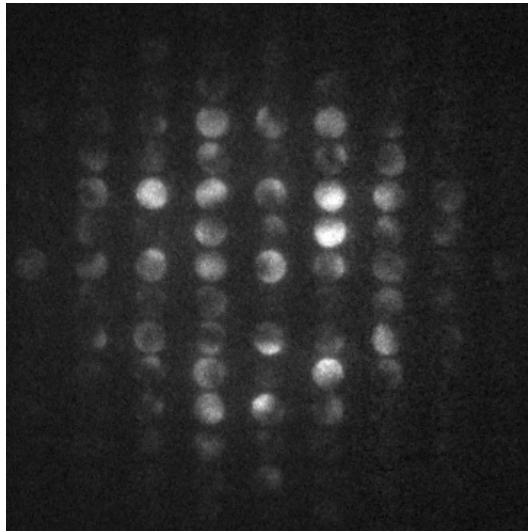


Figure 1.3: CBED pattern of RKTP in the $[010]$ direction

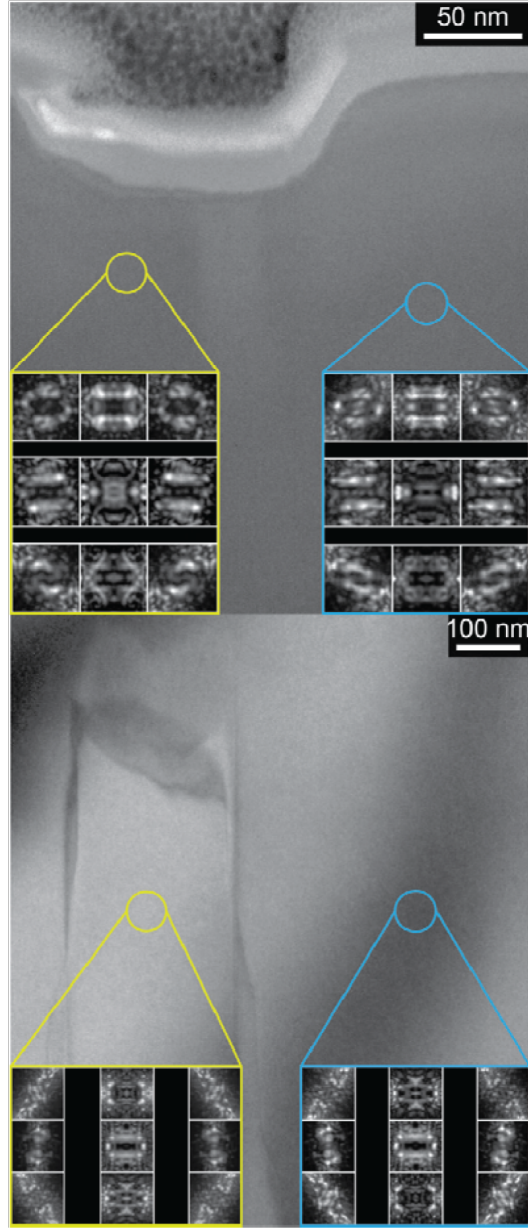


Figure 1.4: Dark-field TEM and D-LACBED analysis of beam induced domain motion on a RKTP sample. In the top frame, a surface dimple marks the location of a former domain which has retracted off of the sample's top edge after uniform irradiation. D-LACBED analysis confirms the domain is continuous across this region. The bottom panel shows the retracted domain. D-LACBED analysis confirms a symmetry reversal (vertical mirror) across the imaged domain wall. From [146]

1.3.2 Structure solution

A form of D-LACBED has been recently used to help confine possible structure factor phases using the three-phase invariant geometry from dynamical three beam theory [118, 147]. The large convergence angle makes it easier to identify three-beam conditions, which are generally off zone axis orientation where four or more beam dynamical conditions are present. If a general direct inversion theory were to materialise (discussed in Section 1.2.2) it may be possible to completely solve structures (ab initio and refinement) from a single zone axis pattern.

1.3.3 Structure of this thesis

We see from the history of parallel and convergent beam electron diffraction that as the datasets become larger and more diverse the more precise and accurate the structural refinements become. While successful refinement studies have been carried out in both incident beam geometries, it has yet to have the same impact as structure solution and symmetry analysis. Given, to the authors knowledge, no previous studies of structural refinement using LACBED patterns exist we choose this route for the thesis. We describe the methodology we use to complete a successful refinement in Chapter 2. We then look at the simplest question in structural refinement: where are the atoms? Where we look at atomic refinement in Chapter 3. In Chapter 4 we move on to analysing further properties in the crystal structure, eventually deciding upon a thorough analysis of the Debye-Waller factor. In Chapter 5 we apply context to the obtained results, by conducting a study on the sensitivity of D-LACBED patterns through simulation.

Chapter 2

From sample preparation to structural refinement

“While fancy-ass instruments such as aberration corrected microscopes with sophisticated detectors make PED work better, all that is necessary is a microscope that has well-implemented shift-tilt purity controls with sufficient range, a pair of post-specimen deflectors, and an electron film magazine”

- Chris Own

Systems Design and Verification of the Precession Electron Diffraction
Technique [41]

The paper by Beanland et al. [7] is the only piece of literature explicitly describing the digital - large angle convergent beam electron diffraction (D-LACBED) technique to this date. Submitted in 2013, many changes and improvements have been made since. Most were made to accommodate structure refinement, the basis of this thesis and the paper by the author and colleagues, published in 2019 [148]. The details of the changes in technique were only briefly mentioned. No paper exists yet for the simulation software *felix* [149]. Therefore, this chapter will focus solely on the D-LACBED technique and the theory used in *felix* to achieve the results we see in the following chapters and the accompanying paper [148]. The other procedures or knowledge required for this study will be briefly mentioned and referenced, although detailed explanations will be omitted due to their prevalence within the literature.

The chronology of this chapter will follow the experimental procedure from preparing a sample to the refinement algorithm we use as shown in Figure 2.1. The sample preparation and alignment procedures are very similar to standard TEM and

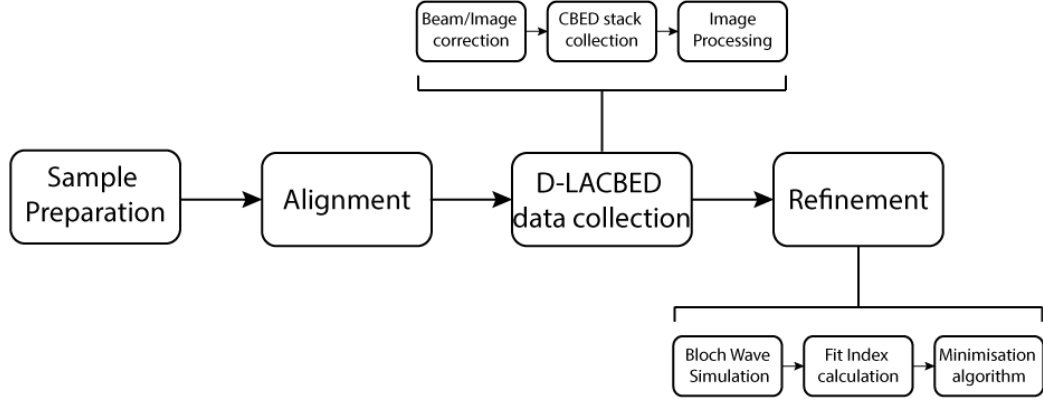


Figure 2.1: Schematic showing the experimental procedure to obtain a refined structure of a single crystal specimen.

CBED methodologies. See Williams and Carter [150] for a comprehensive review. For the collection of D-LACBED data we shall focus on the Digital MicrographTM (DM) scripts we use for the zone axis patterns (ZAPs) we see in this thesis. For a review on the geometry of CBED patterns, and their use in quantitative CBED refinement see Spence and Zuo [36, 151]. The refinement procedure will briefly outline the Bloch wave theory we use to simulate the D-LACBED patterns as well as the maximum gradient refinement algorithm we use. A detailed derivation of Bloch wave theory is provided by Metherell [152].

2.1 Sample preparation

The TEM samples for this thesis were prepared using conventional methods [150, 153, 154]. This included gluing to a copper grid, mechanical grinding with lapping papers and ion beam thinning [155] using a GatanTM Precision Ion Polishing System (PIPS). For ion beam thinning the two argon ion guns were set at angles of 4 degrees above and below the sample plane. The ions were accelerated at 6kV until a hole appeared, then 2kV for approximately 10 mins to remove any surface damage or contaminants.

Depending on the experiment the samples were transplanted into either a GatanTM double tilting or heating holder and then inserted into the JEOLTM 2100 microscope. If the samples were still not clean, they were annealed in situ or to approximately 50°C in a baking station overnight.

2.2 Alignment

The alignment process follows conventional CBED procedure [156] for the most part. However, a few considerations should be made for D-LACBED which are described below in section 2.3 with more detail, including DM scripts, given in Appendix D.

All alignments were performed at 40 or 50kX magnification. The camera length was chosen as 25cm and 50cm for the Gatan Orius and OneView cameras respectively. These values gave optimal resolutions for D-LACBED patterns. A high enough camera length so that the distortions within the patterns can be detected and therefore corrected but low enough that we are able to collect a D-LACBED dataset with a large convergence angle.

After the user has manually aligned the microscope for a conventional CBED pattern a beam and image shift calibration is performed using computer control of the microscope (Sections 2.3.1 & 2.3.2 respectively). These calibrations are required because of the displacement of the incident beam due to the spherical aberration of the pre-field and post-field objective lenses. These displacements are cancelled for the most part by the pivot point alignment ('tilt compensators') of the microscope at small beam tilts, but above a moderate value of tilt they grow in proportion with the cube of the beam tilt angle. Since in practice the displacements depend upon the tilt compensator settings and the excitation of the objective lenses, it is generally necessary to perform a measurement of the displacements for a range of beam tilts immediately before a D-LACBED experiment. The beam shift deflector is used to apply an equal and opposite displacement so that the electron beam remains in the same place on the specimen, to a precision of 1 or 2 nm.

Although similar phenomena might be expected to affect the position of the beam in the diffraction pattern, in practice these are small enough to be ignored for the tilt ranges employed here. Also, as long as the shift compensator is properly aligned the use of the beam shift deflector has no appreciable effect on the 000 beam position in the diffraction pattern.

2.3 D-LACBED data collection

A sequential tilt series of the beam along a serpentine raster (Fig 2.2b) is used to collect a stack of CBED images. The raster is set so that the CBED discs overlap on sequential images, so that they can be stitched together to form the D-LACBED pattern (Fig 2.4).

2.3.1 Focused beam and image shift correction

As mentioned in Section 2.2, we see significant shifts of both the focused beam [7] and image on the specimen and image plane respectively (up to ~ 100 nm) as the electron beam is tilted. Therefore, two separate DM scripts (Calibration 1 & 2 - see Appendix D) are used to measure and correct for the shift seen in both cases as the beam undergoes the serpentine raster used in D-LACBED collection. The focused beam should then remain in one spot on the sample for the entirety of data collection (See the videos in the supplementary information of Beanland et al. [7]).

The focused beam shift calibration (D-ED_Calibration1) works with no sample in the field of view, and measures the position of the focused electron probe in the image for an array of incident beam tilts. It also measures the displacement of the 000 disc in the diffraction pattern produced by a unit change in the x- and y-beam tilt digital-to-analogue converters (DACs), the displacement of the electron probe in the image plane produced by a unit change in the x- and y-beamshift DACs, and the radius of the 000 CBED disc, is set to be the maximum tilt distance (80% of the camera width). This sets out the basic geometry of both the calibration and data collection raster shown in Fig. 2.2.

A pre-determined 8x8 tilt increment series is mapped on to pixel locations in the camera image to create the alignment raster. During both calibrations the beam will be tilted to produce the positions shown in Fig 2.2a. Using linear interpolation¹ this will be used to translate the recorded shift corrections to any pixel location within the camera image.

The image shift calibration (D-ED_Calibration2) requires the incident electron spread beam over the full field of view and the presence of a distinct feature on the sample at exact focus. The second calibration measures (using cross-correlation) the displacement of the image on the camera (corresponding to a shift of the diffraction pattern in the back focal plane) during the same array of beam tilts. The shift is measured using a cross-correlation between a reference and shifted image. They are both Sobel filtered to reduce the effect of uneven illumination induced by coma at large beam tilts, which can cause problems in cross correlation. The Sobel filter removes this broad variation in background intensity and accentuates the sample boundaries within the image.

If the user has microscope has been aligned correctly the recorded shifts should roughly form a ‘star shape’ (see Fig 2.3), where the large beam shifts in the corners of the star indicate the highest tilt angles along the diagonals of the square tilt calibration raster.

¹While a good approximation, the shift is visibly cubic at higher tilt angles as mentioned previously in section 2.2. This will be included in future scripts.

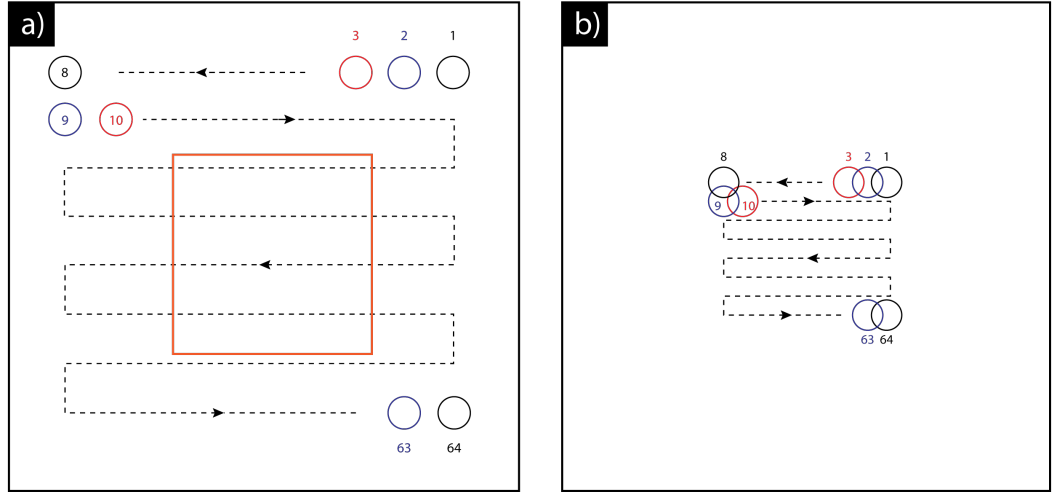


Figure 2.2: a) Schematic showing position of an array of 000 CBED discs during the first tilt calibration raster b) Schematic showing position of the 000 CBED disc in the serpentine raster used in D-LACBED data collection. Colours simply indicate different CBED disc positions. Numbers next to discs indicate its place in the series. The orange box in (a) shows the area of shift corrections (utilising linear interpolation) used for the raster in (b). Scalings are approximate.

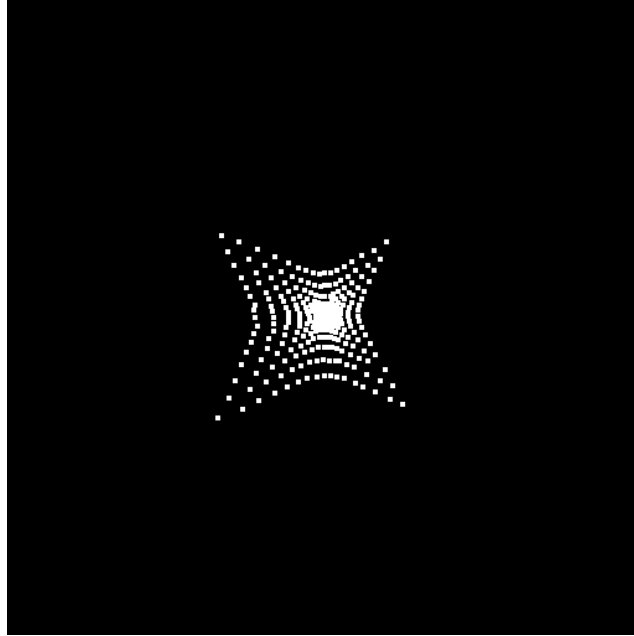


Figure 2.3: Sample recording of image plane shift positions on the camera from D-ED_Calibration2 script

2.3.2 CBED stack collection

Once aligned and calibrated the collection of the CBED disk tilt series can begin. The smaller the radii of the CBED discs and the larger the number of tilts per raster length (the maximum allowed is determined by the camera length – which in turn is restricted by the camera field of view/resolution) the longer the collection process. Most D-LACBED data collections are complete within 10 minutes.

The camera continuously runs at the user specified exposure (in our case 0.01 seconds), but a frame is only grabbed once the tilt has been executed. Most of the delay between frame acquisition is caused by latency between the GatanTM DM script command and the JEOLTM software which controls the lenses. A significant speed up is possible if the commands were controlled through hardware additions rather than software (like in precessional electron diffraction - PED [39]). However, it would make the technique less portable. When each disc within over every frame in the CBED tilt series is stitched together they form our D-LACBED patterns covering a much larger angle of incidence than their individual CBED pattern counterparts as shown in Fig. 2.4.

Future improvements could include a spiral raster, where the tilt series starts in the centre and works out to the outer edge in a spiral sequence. Because data quality generally worsens the longer the beam is on the sample (time taken dependent on sample) due to beam damage and contamination (most prominent tends to be contamination leading to an increase in background contrast) from the electron beam [157] the higher quality data would reside in the centre of the pattern which contains most of the dynamical scattering. At the moment the raster takes a 'top down' approach. Thus if the sample becomes more and more damaged/contaminated the bottom of the pattern will show these effects more prominently than the top. This intensity change could 'break' known symmetry in the pattern. By using a spiral raster sample damage and contamination present in the outer edges of the pattern is more likely to be symmetrical, potentially preserving the symmetry of the pattern.

The quality of the pattern is linked to the speed of collection, therefore any improvements linked to the focused beam spending less time on the sample are beneficial. This includes a faster camera, hardware additions mentioned earlier or continuous collection using post-processing to frame grab.

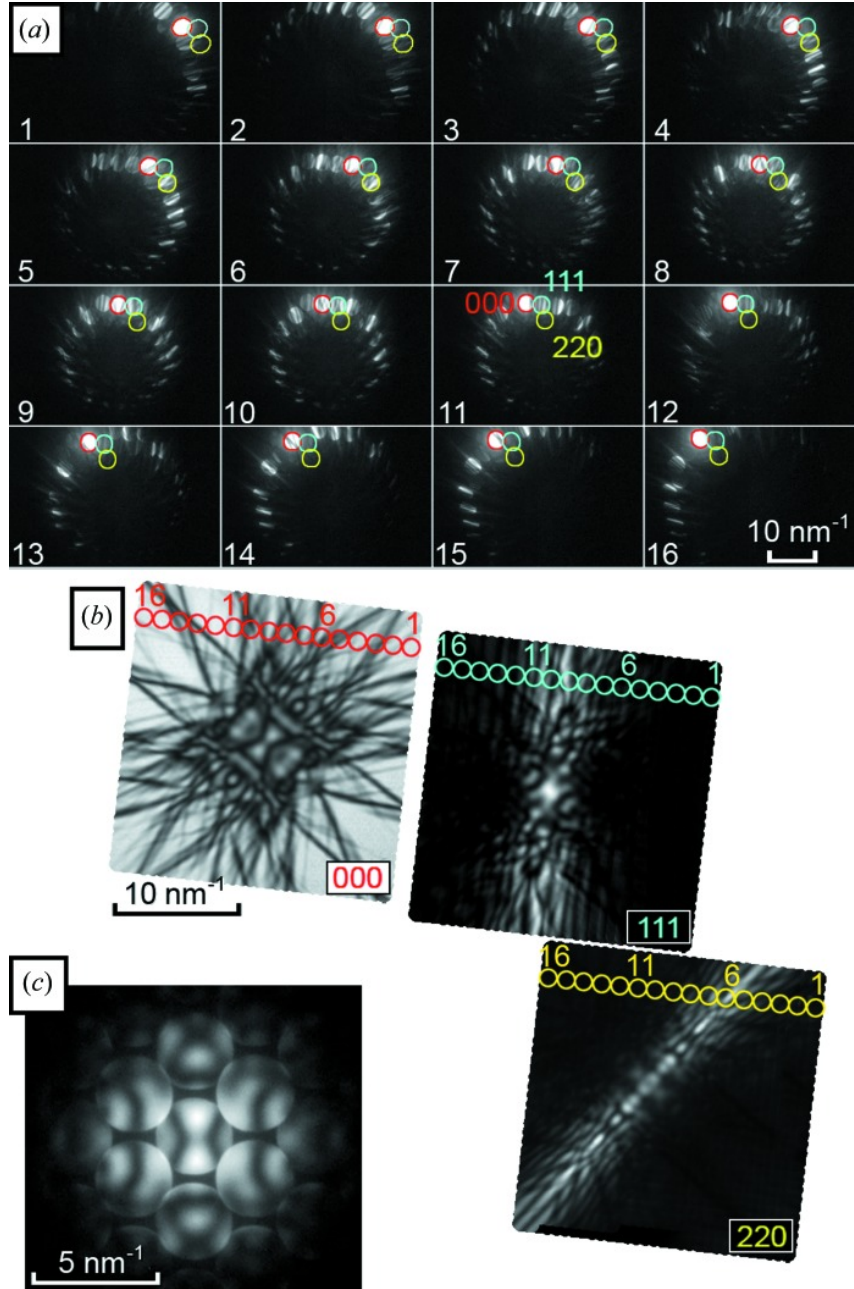


Figure 2.4: Figure 2 from Beanland et al. [7] (a) Sixteen CBED patterns from $[110]$ silicon with varying beam tilts. The 000 (red), 111 (blue) and 220 (yellow) beams are highlighted in each. (b) Digital reconstruction of LACBED patterns from many individual CBED patterns, highlighting the components from the patterns in (a). (c) The on-axis CBED pattern.

2.3.3 D-LACBED image processing

The raw D-LACBED patterns are created from the CBED stack using the DM script `D-ED_Process`. To ascertain the reciprocal lattice vectors, each of the CBED images within the tilt series are shifted (using the calibrated measurements of CBED pattern displacement with beam tilt) so that the central 000 discs are all aligned in the middle. The intensities within the entire stack are summed to produce the averaged CBED image we see in Fig 2.5a. It has some similarities to spot PED diffraction patterns, but over a full square solid angle instead of a hollow one. The averaged CBED image reduces the error in measuring the g-vector basis of the reciprocal lattice from the diffraction pattern. The unit cell dimensions are checked against the reference structure for the material under investigation. However, for each study in this thesis we have used the unit cell parameters from the reference structure (X-ray and neutron source). Electron diffraction unit cell measurements from reciprocal lattice spacings are far less precise than their X-ray equivalents due to distortion. We have also found unit cell parameter refinement using D-LACBED patterns is only precise to around 1pm, on the order of 100-1000 times worse than X-rays. Methods to obtain precise unit cell parameters in-situ (from electron diffraction data) are discussed in Chapter 5. The average CBED image could potentially be used to obtain an initial structure solution before a refinement as, like PED, the average intensity of each CBED can be modelled as pseudo-kinematical. It is a possibility to explore in the future.

The CBED discs corresponding to the minimal g-vectors are identified by the user if the automatic procedure fails to pick up the CBED disks correctly. The indices are obtained from an indexing software such as `CrystalMaker` [158]. The g-vectors are then swapped to make a right-handed pair which will form the reciprocal lattice basis. If the lattice is not centred (i.e. a 90 degree angle is present in the pattern that is not captured by the smallest pair of g-vectors), the user identifies the face centred g-vector. These are stored for later use in the alignment script. For future efficiency, the disc indices could potentially be obtained from the average CBED pattern using kinematical theory.

The background is subtracted through measurement of the background over a small area in between each disc in the average CBED pattern. Ideally a measurement would be taken for every tilted CBED image, however this is currently too computationally expensive using DM scripts. A 2D cubic spline is used to extrapolate the background intensity between each of the measured points in the average CBED image to its entirety. We now have the background intensity for every pixel in the image which is subsequently subtracted from each of the CBED images in

the tilt series (Fig. 2.6).

After background subtraction, a percentage of every CBED disc (usually 75%) is cut out of each image and placed in separate g-pattern images. This is done sequentially. The overlapping intensities between each CBED spot are then averaged together to produce the raw D-LACBED g-patterns (Fig 2.5b).

A montage is also created by pasting all of the raw D-LACBED g-pattern into one image as seen in Fig 2.5c. To do so the basis g-vectors are lengthened and are therefore no longer absolute. Their ratios and angle are however maintained. The montage is useful to observe the quality of the data (through any unexpected symmetry breaking in the pattern caused by contamination and/or distortion).

The raw D-LACBED images are sufficient for symmetry analysis. For quantitative analysis we can use the symmetry of the D-LACBED patterns to average the data. This reduces the error caused by distortion, contamination or small thickness differences. Manual averaging requires days of processing. The D-ED_Align script achieves this in minutes.

Firstly, the centre of the 000 D-LACBED g-pattern is found by creating a 180°rotated copy. Then, through cross-correlation, the distance between the original and rotated image centres can be found, and therefore the image centre location. This is possible due to the two-fold rotational symmetry implicit in every zero-order Laue zone (ZOLZ) D-LACBED g-pattern [94]. The current rectangular images are cropped to squares relative to the g-pattern centres.

Each g-pattern is then rotated so that the 1st g-vector basis in the right-handed pair lies horizontally (g-vectors are re-selected if one is face centred). The centres of every g-pattern are now found to average each individual g-pattern with its rotated 2-fold symmetric compatriot.

If enough mirror symmetry is present in the pattern the linear distortion and skew of the 000 g-pattern in the x and y direction is measured with a cross-correlation using a copy that is stretched and skewed until the best fit is found. This is then applied to every g-pattern. If there isn't enough symmetry the stretch and skew is found later using a thickness refined simulation using the separate D-ED_Stretch script.²

The D-LACBED g-patterns are then averaged according to any rotational and/or mirror symmetry present in the montage. After cropping to a user specified size and convergence angle estimated the D-LACBED data is now ready for structural refinement.

²There is another subsequent calibration script called D-ED_SubPixel which use a thickness refined simulated D-LACBED pattern to align the pattern centres of the experimental and simulated g-patterns.

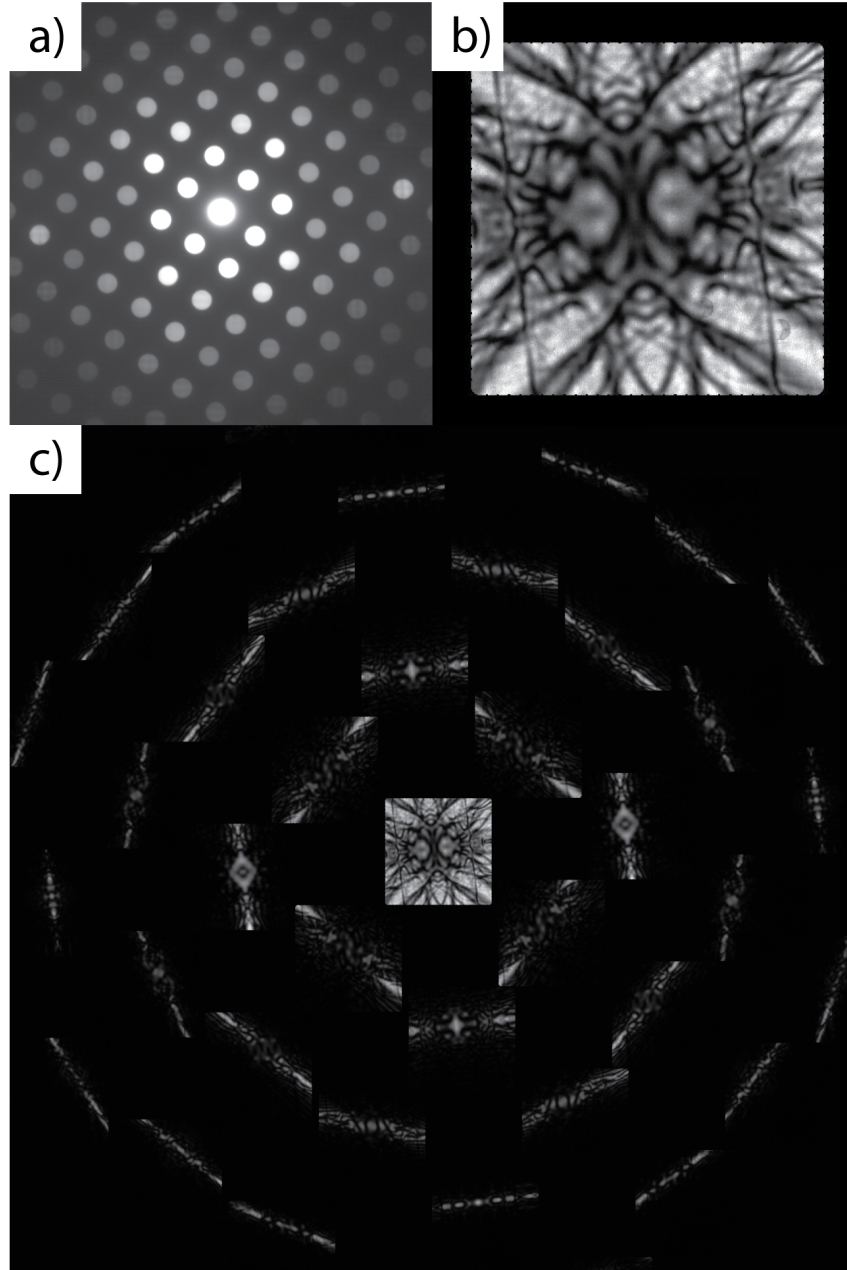


Figure 2.5: a) Average CBED pattern, b) raw 000 D-LACBED g-pattern and c) montage from $[2\bar{2}1]$ alpha-corundum)

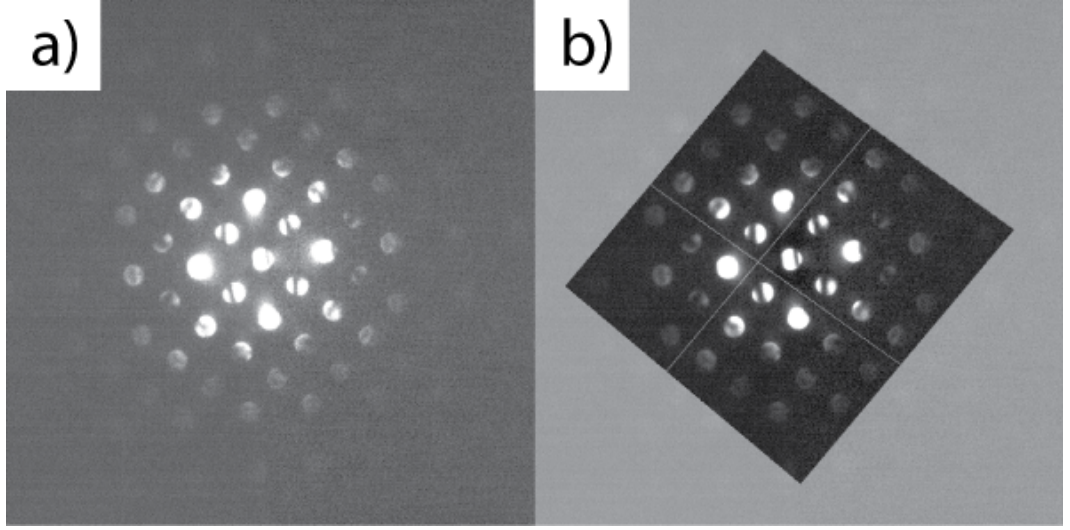


Figure 2.6: Zone axis CBED patterns from $[2\bar{2}1]$ alpha-corundum a) without and b) with background subtraction. The dynamic range in both images has been equally shifted to low intensity values to see the diffuse scattering

The cropped and processed experimental images seen in Figure 2.7b are read into the program *felix* [149], a Fortran MPI [159] parallelised program capable of running on supercomputers.

2.4 Simulation

To simulate our D-LACBED patterns we may choose one of two methods. The first is the Multislice method [160–162]. This is where we split the crystal potential into two dimensional slices and project the potential onto the next slice in the crystal. Using wave-optics, we can propagate the wavefunction of the electron from one slice to the next. The second is the Bloch wave method, which uses the periodicity of the crystal in conjunction with the Schrödinger equation to obtain a linear combination of electron wavefunctions bound by the periodicity of the crystal. These are then solved using the eigenfunction method.

We have chosen to use the Bloch wave method. This is because we are initially interested in defect free single crystals. While the multislice algorithm is quicker [163, 164] and represents the full distribution of electrons in our diffraction pattern, i.e. intensity from inelastic effects can be modelled, the Bloch wave method is both easily parallelisable and can easily accommodate standard unit cell axes [165]. The Bloch wave method can also calculate HOLZ scattering, of which multislice struggles [162]. While HOLZ scattering is neglected in this thesis, the

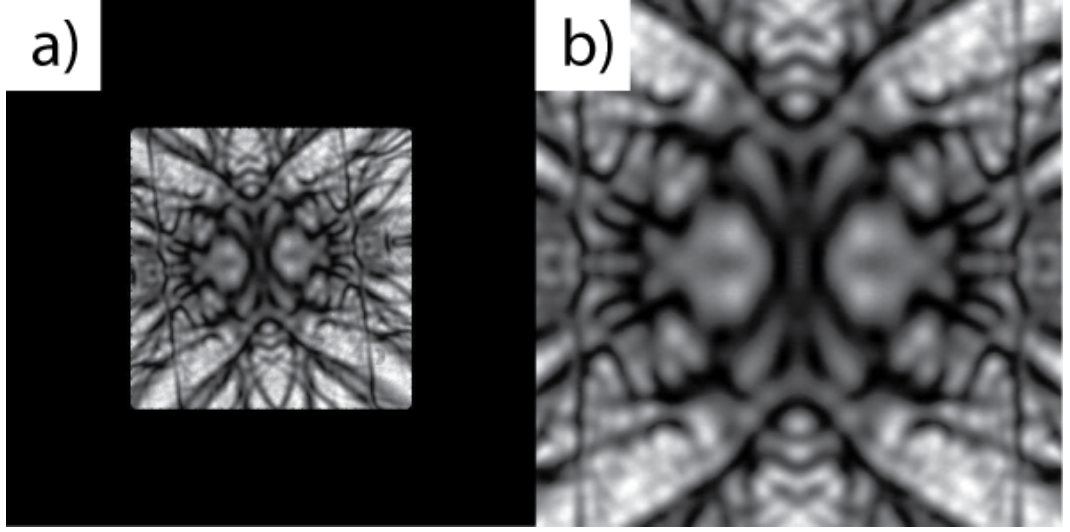


Figure 2.7: a) Raw, uncropped D-LACBED g-pattern vs. b) symmetrically averaged and distortion corrected g-pattern.

arguments in Chapter 5 suggest higher order g-vectors may hold key influence for future D-LACBED refinement.

We therefore give a brief summary of the Bloch wave problem, set out for parallel programming of D-LACBED patterns. We assume absorption (the reduction of Bragg intensity due to inelastic scattering) is always present, meaning we shall not refer to the special quantum mechanical properties of the Bloch wave matrices found in most texts. For the interested reader reviews can be found by Shmueli et al. [165], Self et al. [163], Kirkland [164], Spence and Zuo [36, 151], Metherell [152], Humphreys [166] and Hirsh et al. [167], based on the original paper by Bethe [19] extended by MacGillavry [107], Heidenreich [168] and Kato [169] (for the hardcore enthusiasts).

We must first model the properties of the incident fast electrons in the vacuum of the TEM. Our electron waves incident on the sample have wavevector, $\bar{K}_0 = 1/\lambda$, with wavelength λ . Our electron is travelling at around 70% the speed of light at an accelerating voltage of, $A = 200kV$. We must therefore make relativistic corrections to both the rest mass m_0 and wavelength of the travelling electron according to Fujiwara [170].

$$m = m_0 \left(1 - \left(\frac{v}{c} \right)^2 \right)^{-\frac{1}{2}} \quad (2.1a)$$

$$\lambda = \frac{h}{2|e|m_0A\left(1 + \frac{|e|A}{2m_0c^2}\right)^{\frac{1}{2}}} \quad (2.1b)$$

Where m is our relativistically corrected mass, v is the velocity of our electron, c is the speed of light, h is Planck's constant and $|e|$ is the magnitude of the electrons charge. Since our electrons have a fixed kinetic energy $E = |e|A$ (there is a very small increase when the electron passes through the potential of the crystal) and we are only interested in elastic scattering in the Bloch wave method, the Heisenberg uncertainty principle states that we cannot know the time our fast electron passes through the transmission electron microscope [152]. We can therefore use the time-independent Schrodinger wave equation to describe the motion of our fast electrons throughout their travel in the TEM vacuum

$$\left(-\frac{\hbar^2}{2m}\nabla^2 - |e|V(\vec{r})\right)\Psi_0(\vec{r}) = E\Psi_0(\vec{r}) \quad (2.2)$$

where $\hbar = h/2\pi$ is the reduced Planck's constant, $V(\vec{r})$ is a yet to be assigned potential describing the periodic real space crystal lattice³ and $\Psi_0(\vec{r})$ is the incident wavefunction of the electron. In free space there is no crystal potential, so our electron experiences no potential energy $|e|V(\vec{r}) = 0$. Our incident wavefunction, $\Psi_0(\vec{r})$ is therefore chosen to have solutions to Eq. 2.2 in the form of an infinite plane wave

$$\Psi_0(\vec{r}) = \exp(2\pi i \vec{K}_0 \cdot \vec{r}) \quad (2.3)$$

where we have used the imaginary number i .

To model our electron wavefunction through the crystal we first assume our crystal model has no boundary conditions, i.e. the periodic crystal lattice extends to infinity. The boundary conditions (the crystal surfaces) will be added later. We define a modified potential energy of the fast electron within the periodic structure of the crystal $U(\vec{r}) = (2m|e|/h^2)V(\vec{r})$ to simplify insertion into the Schrodinger equation. The modified potential energy can be represented as a Fourier series with Fourier amplitudes (or modified structure factors) $U_{\vec{g}}$ and phase $\exp(2\pi i \vec{g} \cdot \vec{r})$

$$U(\vec{r}) = \sum_{\vec{g}} U_{\vec{g}} \exp(2\pi i \vec{g} \cdot \vec{r}) \quad (2.4)$$

where the $(2m|e|/h^2)$ has been absorbed into the $U_{\vec{g}}$ term to create the modified

³ $\vec{r} = m\vec{a} + n\vec{b} + p\vec{c}$ for the r^{th} atom in the unit cell with vectors \vec{a}, \vec{b} and \vec{c} where m, n and p are constants. It is this vector which varies during atomic refinement.

structure factor⁴. The modified potential is a summation over the currently infinite number of Bragg reflecting planes in the crystal which we represent by introducing a reciprocal lattice with basis \bar{g} (See introductory solid state textbooks [171–173] for comparison between real and reciprocal space lattice).

The modified structure factors describe the amount of scattering to each reflection due to the crystal structure with unit cell volume Ω

$$U_{\bar{g}} = \left(\frac{2m|e|}{h^2} \right) \left(\frac{h^2}{2\pi m_0 e \Omega} \right) \sum_i [f_{ei}(\bar{q}) + i f'_{ei}(\bar{q})] \exp(-2\pi i \bar{g} \cdot \bar{r}) O_i T_i \quad (2.5)$$

This is achieved through summation of all the individual atoms (index i) elastic and absorptive electron scattering factors $f_{ei}(\bar{q})$ and $f'_{ei}(\bar{q})$ respectively (where q is over any g-vector undefined by the crystal lattice), the site occupancy O_i and each atoms deviation from its mean position due to temperature T_i within the unit cell.

There are a number of tabulations for the elastic scattering factors $f_{ei}(\bar{q})$ available in the literature [164, 174–176] which are included in *felix*. For this thesis, the majority of calculations were with Kirkland’s tabulations [164]. The absorptive scattering factors originate from the complex model of the potential, $U_{\bar{g}} = (U_{\bar{g}}^r + i U_{\bar{g}}^a)$, where we have a real potential $U_{\bar{g}}^r$ (of only elastic scattered electrons) and an absorptive potential $U_{\bar{g}}^a$ (defining the loss of elastically scattered electrons in the diffraction pattern due to inelastic processes). It has been shown that most of the lost electrons is caused by thermal diffuse scattering (TDS) [177, 178]. Other contributions do exist (ionisation [179, 180], Compton scattering [181] etc.) but are complex to model [182]. Therefore they are not included in this thesis. They do however provide an avenue for future studies. The TDS absorption can be modelled using the Einstein approximation as inelastic scattering factors, $f'_{ei}(\bar{q})$, of varying complexity [183–187]. We have used the Bird and King [183] model due to its simplicity. We have not, however, used their tabulated values, instead fully calculating each absorptive scattering factor in each simulation for increased precision. This is

⁴Under normal circumstances, a potential independent of a scatterer is defined, ie. $V(\bar{r}) = \sum_{\bar{g}} V_{\bar{g}} \exp(2\pi i \bar{g} \cdot \bar{r})$ instead of a potential energy, $(2m|e|/h^2)V(\bar{r})$. The structure of a crystal found through separate scattering experiments can then be easily compared through the structure factors $V_{\bar{g}}$. In electron diffraction we can include the $2m|e|/h^2$ factor within the structure factors because electron diffraction experiments are usually conducted at fixed standard accelerating voltages, e.g. 80 kV, 100 kV, 200 kV etc. (which determines the velocity and therefore relativistic mass of the electron). Our modified structure factors $U_{\bar{g}}$ are therefore not ‘genuine’ structure factors as they depend on experimental conditions, but can easily be converted to the standard convention for comparison with other experiments or even for X-rays and other types of scattering [36]

a substantial calculation, requiring a 2D numerical integration over the entirety of reciprocal space, therefore it is MPI parallelised.

The temperature factors T_i describe the loss of intensity of the Bragg beams due to the thermal motion of the atom about its mean lattice position

$$T_i = \exp(-B_i \bar{q}^2) \quad (2.6a)$$

$$B_i = 8\pi^2 \langle u^2 \rangle \quad (2.6b)$$

where B_i is the isotropic DWF and $\langle u^2 \rangle$ is the mean squared displacement for each atom. It is derived by introducing a displacement vector \bar{u} to the phase term $\exp(-2\pi i \bar{g} \cdot \bar{r} + \bar{u})$ in Eq. 2.5 [188]. A review on the effect of the DWF in diffraction patterns can be found in Warren (Chapter 11) [189]. For most studies in this thesis we use the independent atom model (IAM) and harmonic (Einstein) approximation to calculate the isotropic version of the DWF. We show in Chapter 4 how this model starts to break down as the crystal structure becomes more complicated.

For both the atomic coordinate and DWF refinements in Chapters 3 & 4 respectively, the structure factors $U_{\bar{g}}$ are recalculated after every iteration as the variables \bar{r} and B_i are optimised (See section 2.6 for refinement algorithm).

We can reduce the possible number of vectors for atomic coordinate refinement by using the space group symmetry of the unit cell. Only certain movements of atoms down specific vectors will be allowed. This is determined by each atom's Wyckoff position, which have been tabulated in *felix* (currently only for corundum - however the structure is in place for future tabulations of any material).

So far, we have modelled the electron in free space with an ambiguous direction \bar{K}_0 as well as the infinite potential of the crystal sample with the expectation of an electron passing through but without a wave-equation describing it. To this point, each processor has been carrying out the same calculation ie. the code is running in serial. It is possible to split the structure factor calculation into separate parallel calculations for each g-vector. However, the previous calculations are all computationally inexpensive. Setting up in parallel requires needlessly complex communication between processors for a negligible increase in computation time.

We shall see that the model for the wave equation in a crystal (with solutions $\Psi_B(\bar{r})$) for a D-LACBED pattern is computationally expensive but easily parallelisable. We can describe our D-LACBED pattern as a collection of spot patterns separated out into pixels of the final image (Fig 2.8 top). Each pixel p has a separate incident wavefunction $\Psi_0^p(\bar{r})$ with wavevector \bar{K}_0^p , wave equation calculation and resulting exit wavefunction $\psi_g^p(z)$ leading to the pixel intensity I_g^p for each

g-vector. We have therefore used MPI to assign cores separate sets of pixel calculations to send back to the host processor to assemble each image independently (Fig 2.8 bottom), bypassing the overlap problem (See Chapter 1).

Since each pixel calculation is independent the pixel notation p will be dropped in the following explanation of the Bloch wave problem. We shall focus on the calculation of one arbitrary pixel in the D-LACBED pattern.

We now modify our free space Schrodinger wave equation (Eq. 2.2) to include our periodic potential energy of the crystal from Eq. 2.4 and rearrange (see Appendix C).

$$(\nabla^2 + 4\pi^2 k_0^2)\Psi_B(\vec{r}) = -4\pi^2 U(\vec{r})\Psi_B(\vec{r}) \quad (2.7)$$

where we have introduced the magnitude of a new incident wavevector \bar{k}_0 due to the slight increase in the electron's energy as it travels through the negative potential of the crystal. We have chosen Bloch wave solutions for our wavefunction $\Psi_B(\vec{r})$. Like in free space they are also plane waves, however their form is restricted by the periodicity of the potential, and can therefore also be represented as a Fourier series

$$\Psi_B(\vec{r}) = b_j(\bar{k}_j, \vec{r}) = \sum_{\vec{g}} C_{\vec{g}j} \exp [2\pi i(\bar{k}_j + \vec{g}) \cdot \vec{r}] \quad (2.8)$$

Where in principle the sum is over the infinite number reciprocal lattice vectors. For practical purposes the user in *felix* specifies an upper limit to the number of reciprocal lattice vectors. This is achieved by ranking each reflection due to its distance from the zeroth g-vector [190]. Each Bloch wave $b_j(\bar{k}_j, \vec{r})$ indicates one of many degenerate states j (with wavevector \bar{k}_j) of slightly differing kinetic energy but same total energy defining a dispersion surface (See Figure 2.9). The Bloch wave coefficients $C_{\vec{g}j}$ determine the contribution each degenerate state has to the total wavefunction $\Psi_B(\vec{r})$, and are one of the unknowns to be found by the eigenfunction method. The Bloch wavevectors \bar{k}_j are also unknown⁵.

Inserting Eq. 2.7 into Eq. 2.8

⁵Note: we have now described three sets of k-vectors describing three separate total wavefunctions of the fast electron. The wavefunction through free-space: $\Psi_0(\vec{r})$ with wavevector \bar{K}_0 , the Bloch wavefunction through an infinite periodic potential: $\Psi_B(\vec{r})$ with modified incident wavevector \bar{k}_0 and scattered Bloch wavevectors \bar{k}_j where $j = 1, 2, 3, \dots, n$ for n Bloch waves and finally the g-dependent exit Bloch wavefunction: $\psi_{\vec{g}}(z)$ with wavevectors (implicit in Eq. 2.20) $\bar{k}_{\vec{g}n}$ where n is the integer g-vector index.

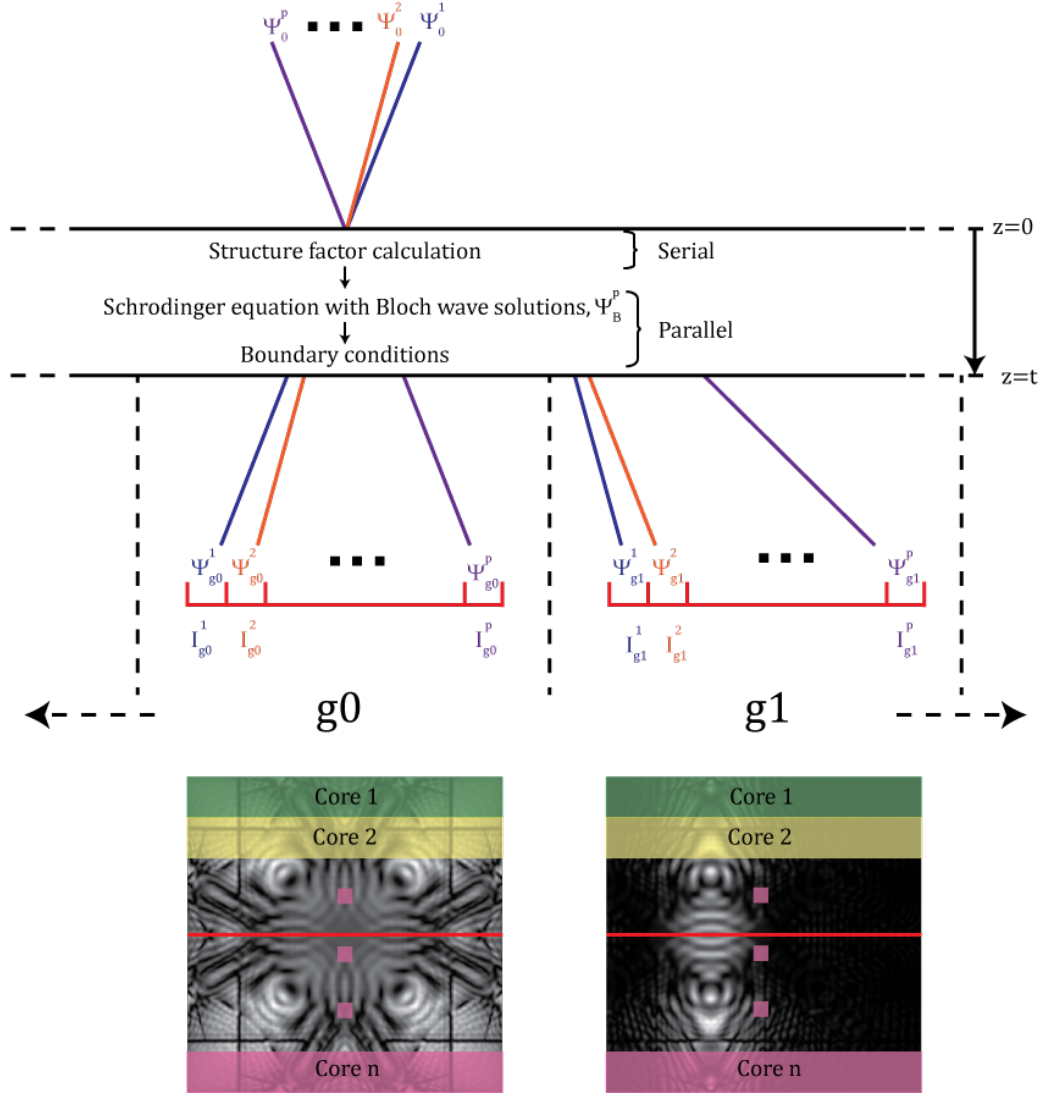


Figure 2.8: top) Schematic of the simulation set up in *felix* over the central 2D pixel line of a D-LACBED image (red line on bottom images). The blue and orange lines indicate the direction of incident wavefunctions (with wavevectors \bar{K}_0^1 and \bar{K}_0^2) for the 1st and 2nd pixels respectively. After transmission through the sample (see Figure 2.9) with thickness t , the lines indicate the exit wavefunctions with wavevectors \bar{k}_{g0}^1 and \bar{k}_{g0}^2 for $g0$ ($g = 0$), \bar{k}_{g1}^1 and \bar{k}_{g1}^2 for $g1$. The purple lines show the p^{th} wavefunction in each case. bottom) Sample simulated D-LACBED g-patterns of [001] Copper. The red line is analogous to the red line in the top image. The green and yellow highlights show example assignments of pixel allocations for each processor up to the n^{th} core

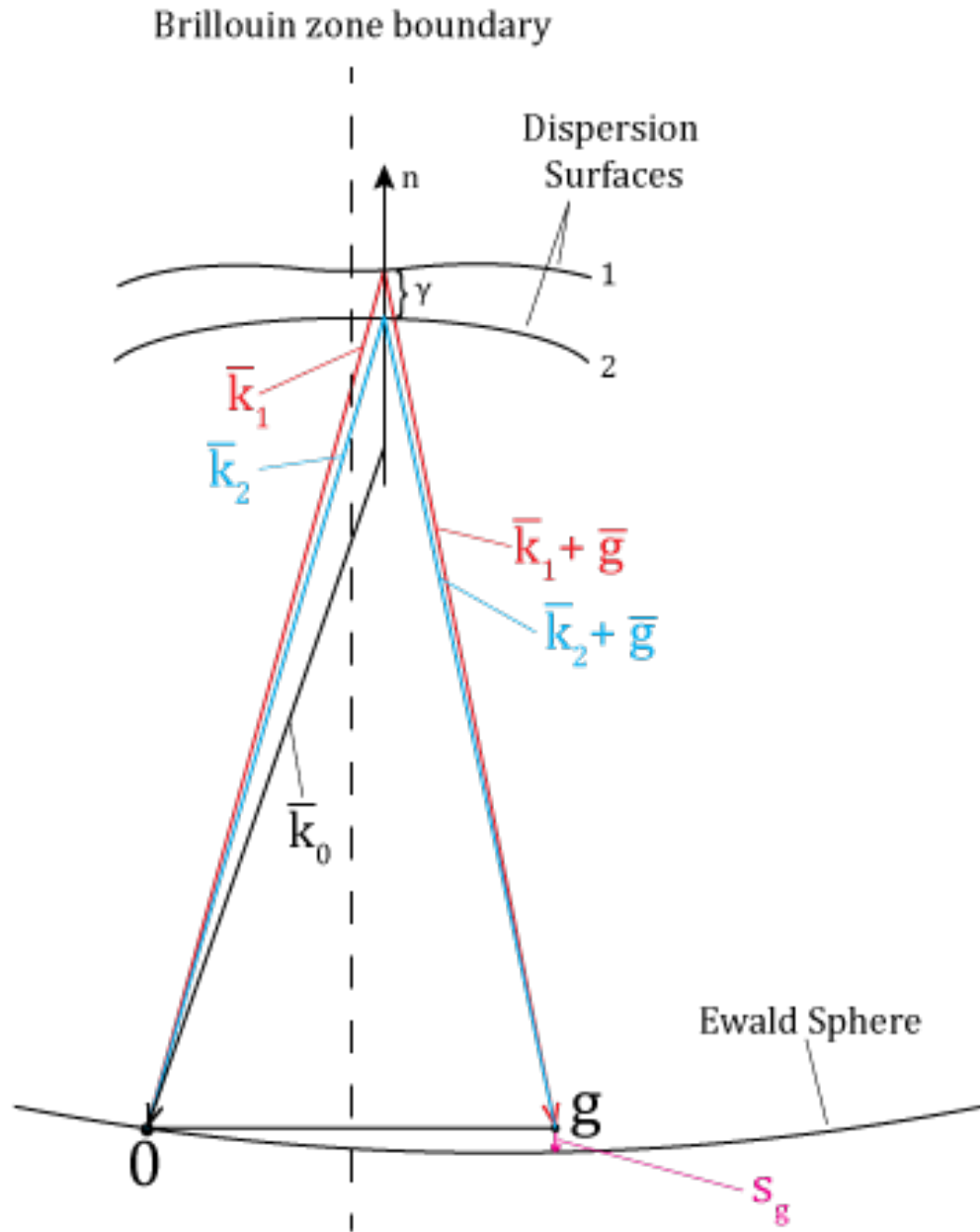


Figure 2.9: Reciprocal space representation of the Bloch wave problem through an infinite potential for the two-beam case. n represents the vector normal to the specimen surface. Other variables are defined within the text

$$\sum_{\bar{g}} (k_0^2 - |\bar{k}_j + \bar{g}|^2) C_{\bar{g}j} \exp [2\pi i(\bar{k}_j + \bar{g}) \cdot \bar{r}] = - \left[\sum_{\bar{h} \neq 0} U_{\bar{h}} \exp (2\pi i \bar{h} \cdot \bar{r}) \right] \left[\sum_{\bar{g}} C_{\bar{g}j} \exp [2\pi i(\bar{k}_j + \bar{g}) \cdot \bar{r}] \right] \quad (2.9)$$

where we have introduced the reciprocal g-vector series \bar{h} to discern between the two series on the right-hand side of Eq. 2.9. The zeroth structure factor term U_0 has been absorbed into the incident wavevector magnitude k_0 for a small computational advantage. By re-indexing and equating exponential coefficients [164] we arrive at the well-known Bloch wave dispersion relation [36]

$$(k_0^2 - |\bar{k}_j + \bar{g}|^2) C_{\bar{g}j} + \sum_{\bar{h} \neq \bar{g}} U_{\bar{g}-\bar{h}} C_{\bar{h}j} \quad (2.10)$$

We wish to make $k_0^2 - |\bar{k}_j + \bar{g}|^2$ linear in order to make Eq. 2.10 solvable by the eigenfunction method. Therefore, we make two approximations. The high energy approximation, $\bar{k}_j = k_0 + \gamma_j$ where we assume that the range of degeneracies for each Bloch state j is very small ($k_0 \sim |\bar{k}_j|$). Also, the x & y electron wave vector components (and their derivatives) must be continuous at the surface of the specimen. We are only allowed a change of the z -component from our incident wavevector $k_{0,z}$ to our Bloch wavevector \bar{k}_j due to the mean inner potential U_0 . Our Bloch wavevector can therefore be approximated using the incident wavevector plus a small term, γ_j ($\gamma_j \ll k_0$ - see Fig 2.9 for geometric interpretation), in the z -direction, $\bar{k}_j = k_0 + \gamma_j \hat{z}$ where \hat{z} is a unit vector. We also only consider zeroth order g-vectors using the ZOLZ approximation $\bar{g} \cdot \hat{z} = 0$.

Through these two approximations we can write [167]

$$k_0^2 - |\bar{k}_j + \bar{g}|^2 = -2\gamma_j \bar{k}_0 \cdot \hat{z} - 2(\bar{k}_0 \cdot \bar{g}) - g^2 \quad (2.11)$$

$$= -2\gamma_j k_{0,z} + 2k_0 s_{\bar{g}} \quad (2.12)$$

where [165]

$$s_{\bar{g}} = -\frac{1}{2k_0} (2(\bar{k}_0 \cdot \bar{g}) + g^2) \quad (2.13)$$

The deviation parameter $s_{\bar{g}}$ is the distance from the reciprocal lattice point to the Ewald sphere (defined by the incident k-vector, \bar{K}). It is used as a beam selection

parameter to discard g-vectors in Eq. 2.10 which have negligible contributions to the total wavefunction [190]. With the addition of Bethe parameters [190,191] the size of our eigenvalue problem can be reduced dramatically with minimal loss of accuracy.

With the advent of Eq. 2.11 we can now form the Bloch wave matrix equation from Eq. 2.10

$$\sum_{\bar{h}} [U_{\bar{g}-\bar{h}} + 2k_0 s_{\bar{g}}] C_{\bar{h}j} = 2k_{0,z} \gamma_j C_{\bar{g}j} \quad (2.14)$$

Or in matrix notation

$$\begin{bmatrix} 0 & U_{-\bar{g}1} & U_{-\bar{g}2} & \cdots \\ U_{\bar{g}1} & 2k_0 s_{\bar{g}1} & U_{\bar{g}1-\bar{g}2} & \cdots \\ U_{\bar{g}2} & U_{\bar{g}2-\bar{g}1} & 2k_0 s_{\bar{g}2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} C_{\bar{g}0j} \\ C_{\bar{g}1j} \\ C_{\bar{g}2j} \\ \vdots \end{bmatrix} = 2k_{0,z} \gamma_j \begin{bmatrix} C_{\bar{g}0j} \\ C_{\bar{g}1j} \\ C_{\bar{g}2j} \\ \vdots \end{bmatrix} \quad (2.15)$$

Using the numerical procedure LAPACK [192] we obtain the eigenvectors, the set of $C_{\bar{g}j}$, and eigenvalues $2k_{0,z} \gamma_j$. We can now solve for an electron travelling through an infinite periodic potential.

To obtain solutions for our total wavefunction $\Psi_T(\bar{r})$ describing a fast electron through a finite periodic potential in the TEM, we impose boundary conditions. In effect the sample acts as a filter, only allowing certain forms of plane wave through dependent on the structure and orientation of the crystal. To do this we introduce an excitation coefficient α_j for the total wavefunction at the entrance surface $\Psi_{TU}(\bar{r})$. We can also introduce a separate thickness-based coefficient $\psi_{\bar{g}}(z)$ to represent the total wavefunction at any thickness throughout the sample for each g-vector $\Psi_{TL}(\bar{r})$. This has been called the ‘Darwin’ representation [167] of the total wavefunction

$$\Psi_{TU}(\bar{r}) = \sum_j \alpha_j b_j(\bar{k}_j, \bar{r}) \quad (2.16a)$$

$$\Psi_{TL}(\bar{r}) = \sum_{\bar{g}} \psi_{\bar{g}}(z) \exp [2\pi i(\bar{k}_0 + \bar{g}) \cdot \bar{r}] \quad (2.16b)$$

Since both Eq. 2.16a and Eq. 2.16b are equally valid representations of the wavefunction at any point in the crystal we can equate them both to obtain the total wavefunction

$$\Psi_T(\bar{r}) = \sum_j \alpha_j b_j(\bar{k}_j, \bar{r}) = \sum_{\bar{g}} \psi_{\bar{g}}(z) \exp [2\pi i(\bar{k}_0 + \bar{g}) \cdot \bar{r}] \quad (2.17)$$

Equate exponential coefficients and rearrange for $\psi_{\bar{g}}(z)$

$$\psi_{\bar{g}}(z) = \sum_j \alpha_j C_{\bar{g}j} \exp(2\pi i \gamma_j z) \quad (2.18)$$

We can now obtain the wavefunction for each g-vector at the exit thickness t once we solve for the excitation coefficients using the special circumstance where $z = 0$. We take the inverse of the Bloch wave coefficient matrix and apply to both sides to obtain an expression for the excitation coefficients

$$\alpha_j = \sum_{\bar{g}} C_{\bar{g}j}^{-1} \psi_{\bar{g}}(0) \quad (2.19)$$

where $\psi_{\bar{g}}(0)$ is unity for $g = 0$ and zero for all other g-vectors⁶. By inserting into Eq. 2.18 we obtain the Bloch wave

$$\psi_{\bar{g}}(z) = \sum_j C_{\bar{g}j} \exp(2\pi i \gamma_j z) C_{\bar{g}j}^{-1} \psi_{\bar{g}}(0) \quad (2.20)$$

the intensity of each pixel is then determined by the relation

$$I_{\bar{g}}^p = |\psi_{\bar{g}}^p(z)|^2 \quad (2.21)$$

where we have re-introduced the pixel notation p .

2.5 Fit-index calculation

We now require a method to fit our simulated D-LACBED images with our experimental ones. We have chosen to use a zero-mean normalised cross correlation (ZNCC) [193]. Fits were obtained for each D-LACBED pattern individually, removing any dependence on relative intensities of different reflections.

To give some resemblance to the R-factor commonly used to indicate experiment/simulation fit quality in structure solution methods the fit index for n experimental patterns x and simulated patterns y , each with N pixels is

⁶At the top surface of the crystal (where the electron beam enters) all diffracted beams have zero intensity while the transmitted beam has an intensity of unity.

$$f = \sum_{j=1}^n \left(1 - \sum_{i=1}^N \frac{(x_{i,j} - \bar{x}_j)(y_{i,j} - \bar{y}_j)}{\sigma_{xj}\sigma_{yj}N} \right)_j \quad (2.22)$$

where the first sum is over all patterns and the second sum is the ZNCC: x_j and y_j are the means of the j^{th} experimental and simulated patterns respectively; $x_{i,j}$ and $y_{i,j}$ are their pixel values; σ_{xj} and σ_{yj} are their standard deviations. Eq. 2.22 gives $f = 0$ for a perfect fit.

The ZNCC has the advantage of being insensitive to background offsets and scaling. This is achieved by bringing both simulated and experimental patterns (Fig 2.10b) onto the same mean (Fig 2.10c) and scale (Fig 2.10d).

The simulated 000 g-pattern is not blurred in Fig 2.10, with a noticeable difference to the experimental g-pattern. This is accounted for by adding a Gaussian

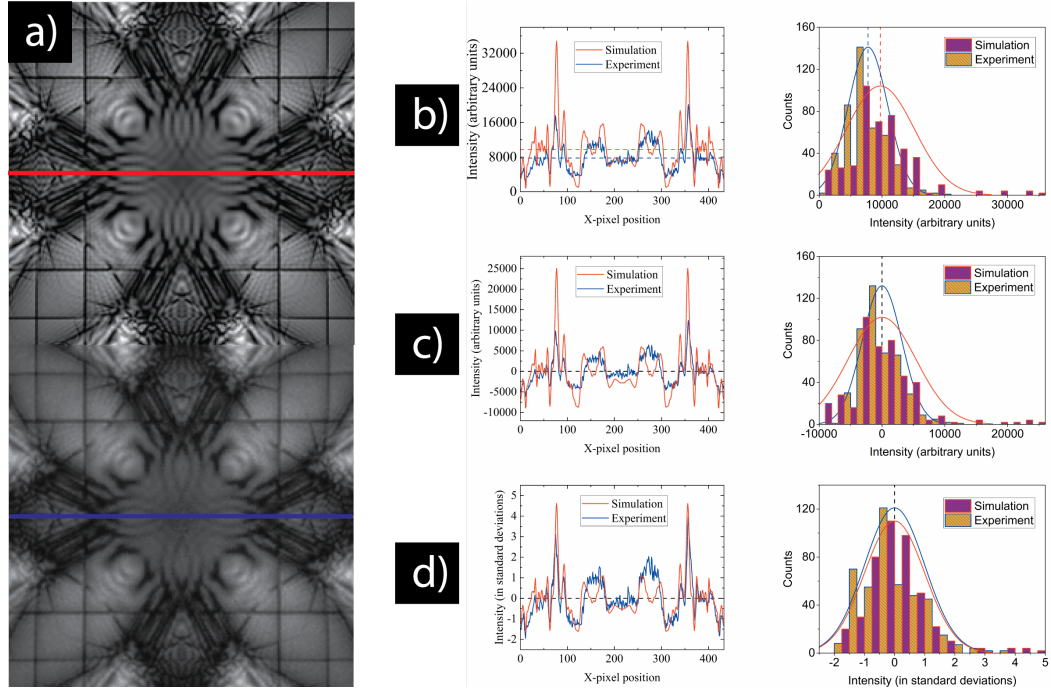


Figure 2.10: a) simulated (top) and experimental (bottom) 000 D-LACBED images of Copper in the [001] direction. The red and blue lines indicate the pixels used for the line plots in b,c and d. b) line plot (left) of the raw D-LACBED images with associated histogram (right). The red and blue dotted lines indicate mean values of 9733 and 7770 for the simulated and experimental images respectively. c) line and histogram plot from b, but with each image mean subtracted from the intensity values. The black dotted line indicates zero mean for both images d) line and histogram plot from c, but with the intensity values divided by the standard deviation of each image. The black dotted line is again the zero mean.

blur to the simulated image before obtaining the ZNCC. See Chapter 5 for more detail.

Currently, each g-pattern is weighted evenly i.e. each g-pattern gives the same contribution to the averaged global fit value. It is shown in Chapter 5 that in future perhaps more weighting should be given to higher order g-patterns.

2.6 Minimisation algorithm

In the very first iterations of *felix* we used the simplex algorithm [194] to refine crystal structural parameters. However, this proved to be a very inefficient procedure for D-LACBED refinement. We found the global minima in 2D fit spaces to be approximately parabolic (see Figs 4.2 & 4.3). Therefore, we used three simulations to apply Cramer’s rule [195,196] in order to extrapolate the minimum of a parabola in the negative gradient direction. We then simulate at that point and repeat the process until a user set difference is met between sequential parabola minimums. This method was applied for the copper data in Chapter 4. For greater than two dimensions we use $n + 1$ points (where n is the number of dimensions) to determine which direction between the points has the maximum gradient. A 2D parabola is then calculated along that direction and the process is repeated. This method was applied to the corundum, gallium arsenide and indium phosphide data in Chapters 3,4 & 5 respectively. More recently, we have found that under certain circumstances the maximum gradient method can be ineffective due to the topology of the fit space (for example ‘valleys’ to the global minimum). This provides scope for future alterations to the maximum gradient algorithm. So far we have found singular, parabolic global minimums for the every refineable variable. However, should a circumstance arise with a more complex fit space/global minima the minimisation algorithm can be changed (e.g. a genetic algorithm).

Chapter 3

Atomic refinement

“It would be very pleasing, given that electron diffraction can give us the unit cell and the crystal symmetry so easily, to be able to determine atomic positions too. There is clearly enough information present to be able to do so.”

- J.A. Eades

Microdiffraction’s Contribution to Microcharacterization [197]

Measurement of crystal structure usually takes place in two stages: first, structure solution, in which the unit cell and symmetry of the material is determined, as well as the atoms in the unit cell and their approximate locations; followed by structure refinement, in which time-averaged atomic coordinates and thermal vibration parameters are optimised to give the best fit to experimental data. The CBED technique has been used abundantly in the past for most types of structural refinement study as mentioned in Chapter 1. Although, surprisingly there have been no atomic coordinate refinement (ACR) studies using CBED in the previous decade.

Reasons for this are difficult to discern. Atom coordinates determined from structure refinement and solution studies using electron diffraction have been around for at least the last thirty [117] and sixty [34] years respectively. In the evolution of such techniques there has been a trend to obtain greater amounts of information for better results. The introduction of ‘diffraction tomography’ (EDT [54, 198, 199], PEDT [68, 70, 81, 82], RED [53, 70, 200], cRED [64, 69, 201]) is a clear example. It is therefore strange that the technique which currently produces the greatest amount of information at a zone axis, CBED, has not been completely utilised.

The most recent and thorough study of CBED ACR to date (in 2004), Ogata et al. [202] refines 21 independent atomic coordinates in BaTiO_3 , a jump of 20 from the authors previous paper on LaCrO_3 [132]. The introduction of parallel computing

was given as the reason for the increased capability of their refinement. However, there appears to have been no CBED ACR studies since. Studies on Debye-Waller and refinement of individual structure factors using CBED are abundant in comparison [136]. Most work uses computationally heavy dynamical theory, sometimes with double-digit degrees of freedom. Perhaps a perceived lack of competitiveness with X-ray diffraction may be a possible reason. Or it may simply be a loss of interest in ACR [129] due to the previous success of CBED for electrostatic potential and/or electron density studies.

Here, we explore ACR refinement using D-LACBED data from alpha-corundum (α -Al₂O₃). We find that D-LACBED data can produce accurate and precise results within 1 picometre of the nominal values with a well-defined global minimum. In the following sections we explore how the increase in information provided by D-LACBED data improves ACR in comparison to CBED and suggest ways a convergent beam could be employed for similar studies in the future.

3.1 Alpha-corundum

Alpha-corundum, the thermodynamically stable phase of aluminium oxide, is a well-studied material [203] with very well-characterised fractional atomic coordinates [204, 205] and electron density [127, 206]. It has a trigonal structure (space group $R\bar{3}c$, Figure 3.1) and is one of a class of materials with composition X_2O_3 , where X is a metal. For each material, the metal atomic coordinate is slightly different from that of aluminium in α -Al₂O₃. Sapphire, the term sometimes used interchangeably with corundum, describes in fact the weakly doped varieties of alpha-corundum; the most prominent example is ruby (chromium doped α -Al₂O₃).

The atomic coordinates in the alpha-corundum basis have only two degrees of freedom (i.e. coordinates not fixed by symmetry), one each for Aluminium [$0, 0, z_{Al}$] and Oxygen [$x_O, 0, 1/4$]. Although D-LACBED using only ZOLZ data is insensitive to atomic coordinates along the beam direction, due to the symmetry of the space group, refinement of these coordinates always changes coordinates of some atoms that are perpendicular to any zone axis we select. High quality room temperature X-ray data from Kondo et al. [204] (here referred to as SK) gives the fractional coordinates $z_{Al} = 0.352156(17)$ and $x_O = 0.69364(7)$ respectively and these values are taken here as reference values. The simple and well-studied nature of the structure along with the future possibility to measure small atomic displacements in the X_2O_3 series, means α -Al₂O₃ is an ideal test case for ACR using D-LACBED.

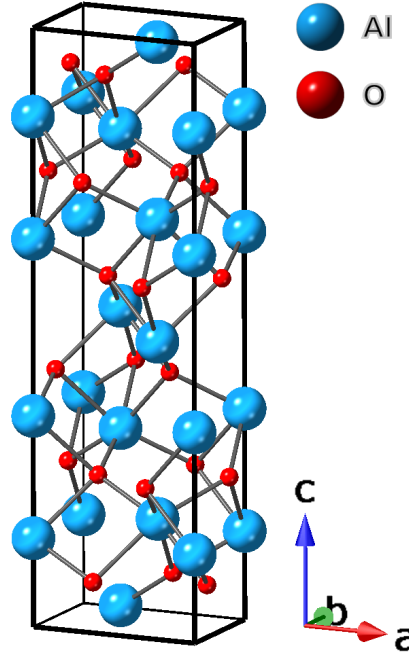


Figure 3.1: Crystal structure of $\alpha\text{-Al}_2\text{O}_3$. The symmetry of the space group $R\bar{3}c$ dictates that the aluminium atom can only move along the c direction with the oxygen atom only along the a or b vectors dependent on the Wyckoff position (See [203])

3.2 D-LACBED

The first two goals were to ascertain the accuracy of a single D-LACBED zone axis pattern (ZAP) and the suitability of the IAM for ACR. While evidence suggests the IAM does suffice [207], most other electron diffraction techniques currently use tomographic or multiple zone axes to determine atomic coordinates (see Chapter 1). However, no other technique has the superior angular range D-LACBED provides. A D-LACBED ZAP will therefore give the baseline accuracy.

A TEM sample of corundum was made using standard procedures.

D-LACBED data were collected from the $[2\bar{2}1]$ zone axis at room temperature (we will call it the A221 dataset – Figure 3.2) with a smallest Bragg angle of $2\theta_{10\bar{2}} = 7.2\text{mrad}$ at 200 kV. 961 CBED patterns were collected in approximately 3.2 minutes with a CBED disc beam half-convergence angle of 1.7 mrad, giving D-LACBED g-patterns extending beyond 50 mrad. The reconstructed images had dimensions of 296 x 296 pixels. A221 provides the basis for the entirety of this

chapter. After processing, the DWF ($B_{Al} = 0.15\text{\AA}^2$ and $B_O = 0.44\text{\AA}^2$ – see Section 4.4), thickness (87.3 nm) and D-LACBED g-pattern half-convergence angle (27.57 mrad) were refined using the simple gradient descent algorithm within *felix* [149] (described in Section 2.6).

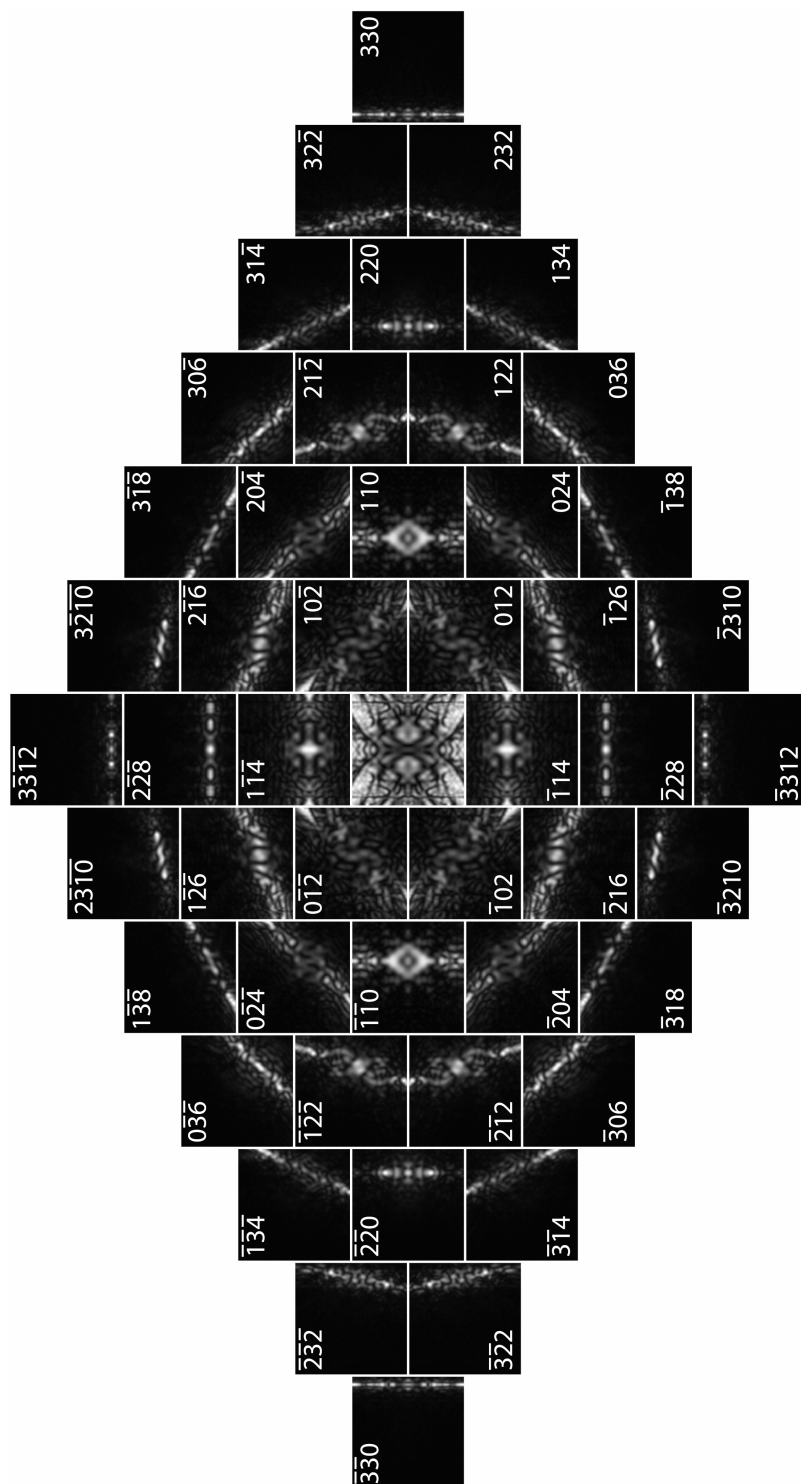
Refinement of the atomic coordinates from this gave $z_{Al} = 0.3525(4)$ and $x_O = 0.693(1)$ [148]¹, i.e. an approximate difference of 0.4 pm and 0.2 pm respectively from the literature values (see Figure 3.7 - bond lengths and angles can be found in Appendix A). As shown in Figure 3.3 the data gives a unique best fit of $f = 0.6\%$ (a ‘perfect’ match gives a 0% fit value).

This is the best fit that has been achieved with D-LACBED experimental data so far. For comparison, fit values for the copper data in Chapter 4 had values in the range of 3 – 10%. This shows that for at least relatively simple structures the IAM model is sufficient for sub-picometre precision atomic coordinate refinement. Strikingly, the final atomic coordinate values suggest that a D-LACBED ZAP probes enough of reciprocal space for sub-pm accuracy, well within the accuracy of the other electron diffraction ACR techniques [81, 82, 202].

To illustrate the sensitivity and goodness of fit of the full D-LACBED data to atomic coordinates we show the changes in the patterns using a form of differential in Figs 3.4, 3.5 and 3.6 that is appropriate for our use of the ZNCC as a fit index. (Intensity differences are calculated between simulations for small changes δx_0 and δz_{Al} , 1pm each, for Figs 3.4 and 3.5, between the experimental and refined simulation for Figure 3.6). Each pattern’s mean is subtracted, and intensity normalised to give a standard deviation of unity. The scales of $\delta I/\delta x_0$, $\delta I/\delta z_{Al}$ and the residual are thus in units of standard deviation. It is clear from the $\delta I/\delta x_0$ and $\delta I/\delta z_{Al}$ images that all patterns display strong sensitivity to sub-picometre atomic displacements of both atoms. Some regions in a few of the patterns are very strongly affected (often close to, or at, the pattern centre), while other regions are relatively insensitive. The intricate changes of intensity are quite different for the two parameters, which results in the ability to refine them essentially independent of each other.

Importantly, even with highly sensitive data, the maximum residual in Fig. 3.6 does not exceed 1 standard deviation (σ). We see in Section 5.3 the precision of the D-LACBED patterns is approximately a picometre per 1σ of max normalised difference or ‘delta’ intensity, emphasising the excellent fit achieved. The remain-

¹After publication we decided to undertake the precision analysis presented in Chapter 5 where we have used conservative error values of ± 0.5 standard deviations from the maximum normalised difference or ‘delta’ in simulated montages. This is based upon the difficulty of determining the complex effect of the many influences on the patterns (this includes limitations within the simulated model, such as bonding - as well as optical errors e.g. distortion.) This is something to be examined for the future



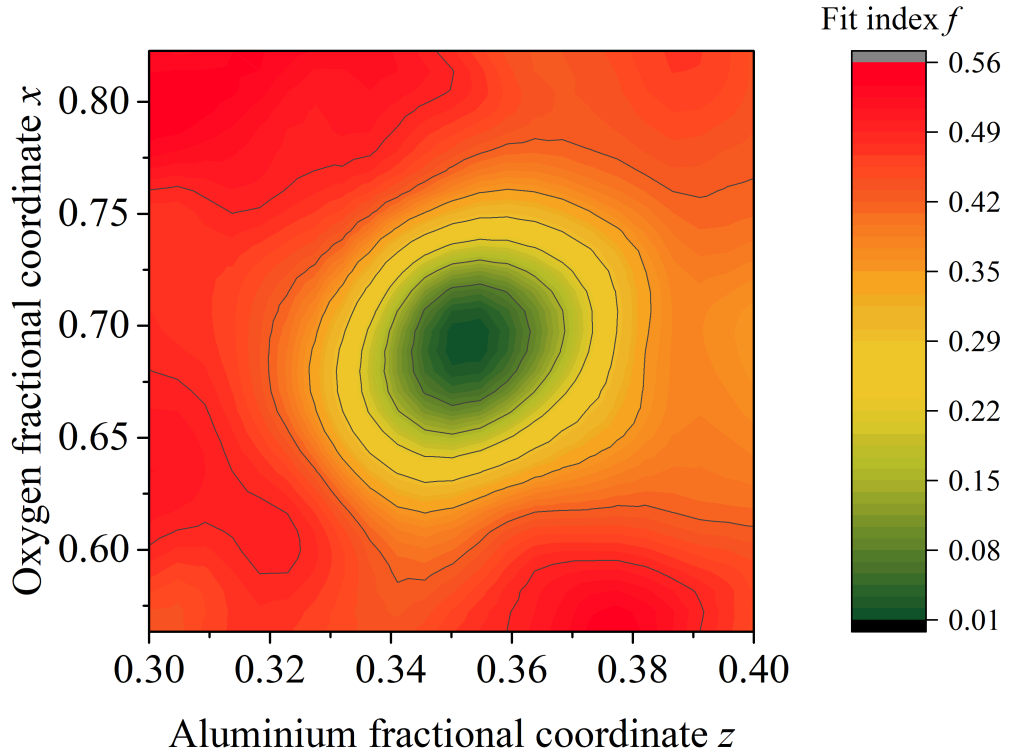


Figure 3.3: fit index f as a function of x_O and z_{Al} .

ing residuals seen could be due to several factors. The pattern of intensity seen in some g-patterns is asymmetric, even though both the experimental and simulated data are not, suggesting very small uncorrected distortions or sub-pixel shifts. The limitations of our distortion correction methods are discussed in Chapter 2. The remaining intensity difference could be put down to limitations within the simulation as well as in the background correction. Even a small level of charge transfer (which we don't account for) between the atoms may affect the atomic potential. Limitations or small errors in the background correction may also lead to small systematic errors in calculation.

In the context of dynamical diffraction Figure 3.7 shows a highly accurate and precise result. It is clear there is still work to be done to achieve the precision and accuracy of X-ray and neutron scattering techniques. D-LACBED as a technique is very young and there are plenty of improvements to be made (see Chapter 2). However, it already matches or exceeds the accuracy of other electron diffraction techniques [43] which have been used to solve highly complex structures requiring far more reflections [73, 84, 86, 208–210].

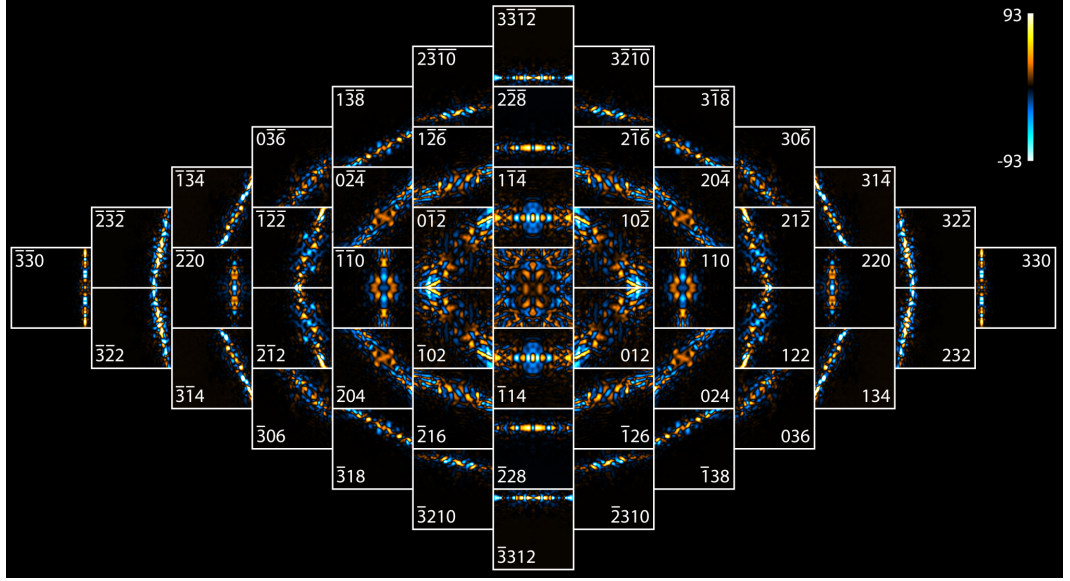


Figure 3.4: $\alpha\text{-Al}_2\text{O}_3$ $[\bar{2}\bar{2}1]$ $\delta I/\delta x_0$ to G_3

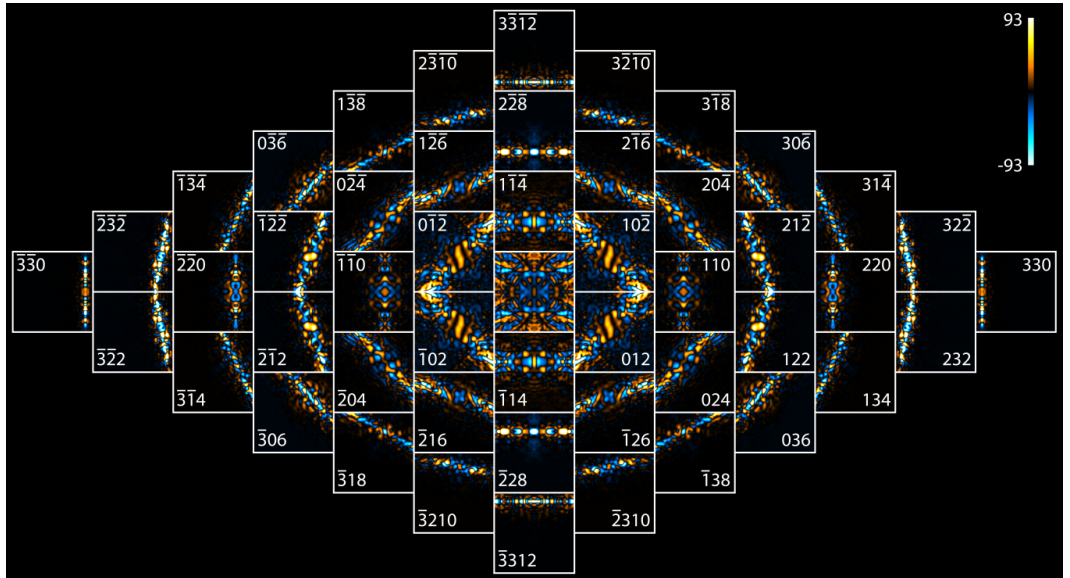


Figure 3.5: $\alpha\text{-Al}_2\text{O}_3$ $[\bar{2}\bar{2}1]$ $\delta I/\delta z_{Al}$ to G_3

3.3 CBED

Given the acceptable accuracy of previous CBED ACR studies [132, 202, 207], one may wonder the reason for using the D-LACBED technique instead of CBED. What advantages does D-LACBED data have over CBED data? In order to examine this question each of the g-patterns from the A221 dataset were cropped down to 104 x 104 pixels to produce data with a similar information content to a conventional CBED pattern with a beam half-convergence angle of 4.92 milliradians. Although, unlike a conventional CBED pattern each g-pattern is a separate image and square, rather than a series of discs in one image (see Figs. 3.8 & 3.11 - CBED). Due to the displacement of the Bragg condition from the centre of the pattern with increasing g-vector magnitude, there is very little intensity in g-patterns past the first order in these cropped areas (compare cyan and orange squares in Figure 3.8). Therefore, only the 000 and first order g-patterns were used in refinement.

Atomic coordinate refinement in *felix* using the cropped data gave $z_{Al} = 0.353$ and $x_O = 0.691$, i.e. 1.13 and 1.27 pm from SK respectively. The accuracy of this result is in line with Ogata et al. SrTiO₃ ACR [202] and within the typical average distance from reference atoms obtained from dynamical PEDT [82] (both on much more complex materials). So far atomic refinement using CBED, as expected,

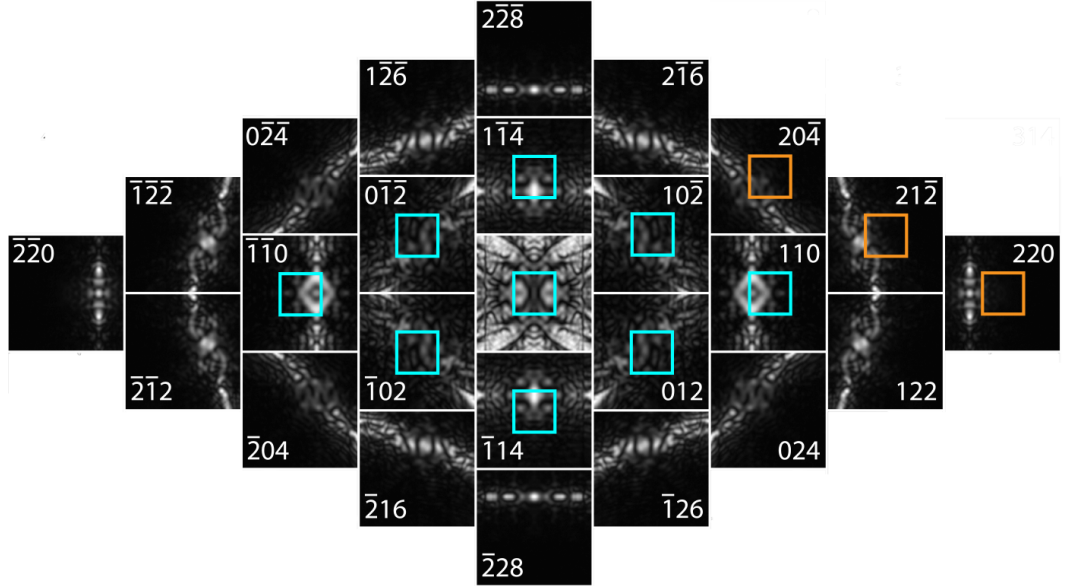


Figure 3.8: The A221 dataset up to second order. Blue and orange squares show the cropped region used to approximate a single CBED dataset; the intensity within the orange squares is low and subsequently were not used in refinement.

seems feasible. However, the favourable refinement conditions should be noted. We have chosen the longest first order g-vector as our basis. This means in a conventional CBED pattern some discs would be overlapping slightly. Also, in a true CBED pattern distortion correction would have to be applied differently, without the benefit of high convergence angle D-LACBED data. While methods exist [125], it is unclear whether this would produce better or worse results. The thickness value was fixed at 87.3 nm previously obtained from the initial DWF refinement using the full A221 dataset. The starting refinement position was chosen close to the SK refinement values. It is uncertain how accurate the initial estimate must be for a successful refinement, especially for materials with many degrees of freedom. This will be covered in the next section.

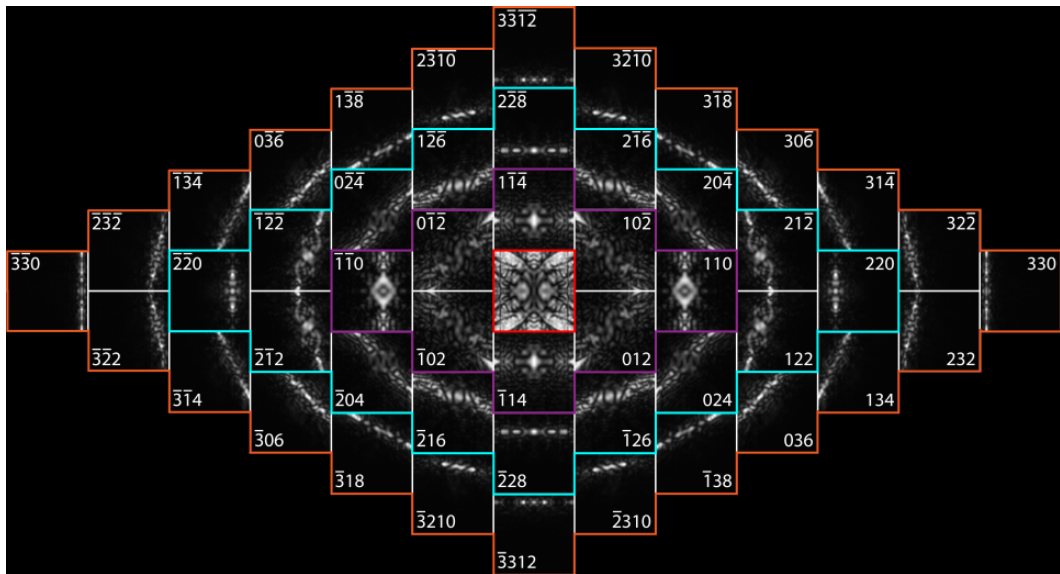
3.4 D-LACBED g-vector series

Given the trend within parallel beam electron diffraction techniques to explore a greater area of reciprocal space for more accurate and reliable results, and the improved accuracy from our CBED to D-LACBED dataset, it is interesting to see the effects of a larger dataset size on ACR. Increasing the effective convergence angle of the incident beam gives more information both in the number of intensities (dependent on Bragg condition) and g-patterns (as long as the the Ewald sphere is tilted far enough to intersect more reflections). The uppermost limit is then only determined by the camera field of view for any given crystal along a particular zone axis². There is some loss of data during the processing stage, where each g-pattern is cropped relative to the size and centre of the central D-LACBED image. At high index g-patterns, the result is a partial or complete loss of data since most diffracted intensity lies outside the image boundary. We can start to see this effect along the (110) systematic row in Figs 3.2, 3.4, 3.5 and 3.6.

Starting with just the central 000 g-pattern of the A221 D-LACBED data set we ran ACR over an increasing number of g-patterns limited by the reciprocal vector range, G_x , where x is the order number (see Figure 3.9). For example, a refinement up to G_1 will involve the central image as well as the first order $1\bar{1}\bar{4}$, $10\bar{2}$ and 110 g-patterns (symmetrically related g-patterns were also included).

All datapoints in Figure 3.10 are within 1.5 pm of SK's result, a reasonable range for an atomic refinement study. The CBED refinement gives the worst result. Interestingly the G_0 refinement gives a closer match to SK than G_1 . This may be

²Technically the camera length gives the ultimate limit to the number of intensities, however this is tied in to the resolution of the camera. This is discussed in Chapter 2 & 5



because the central image is used as the reference for the distortion correction, it is the least likely g-pattern to have distortion errors. Also, high index g-patterns with a large proportion of background to Bragg intensity will be more susceptible to distortion or small misalignment with the simulated g-patterns. Only a small area of the dark field g-patterns have a high level of Bragg intensity which generally has relatively sharp gradients to the almost zero-level background. Therefore any distortions or g-pattern misalignments could contribute to a more erroneous result.. This may be the cause for the discrepancy, however further investigation is required. For quick and efficient ACR, all three g-vector simulations produce acceptable results.

The fit values given in the Fig. 3.10 caption for G_1 and G_2 are very similar and noticeably lower than G_0 . Contrary to the slightly worse fit (i.e. higher percentage) we might expect when initially including more intensities. In the ZNCC we compare all the intensities, including the background (null) intensity which gives an almost perfect fit (limited by experimental noise). Therefore, if we account for the almost perfect fit to background intensity between simulation and experiment in the dark field patterns, we may expect fits to become better compared with the bright field 000 g-pattern as observed. Simply, the background fit is enhancing the total correlation. The relatively large (0.3%) decrease in fit index from the inclusion of the dark field g-patterns indicate this. A thorough study of other zone axis data in corundum and/or with different materials will help to confirm this argument..

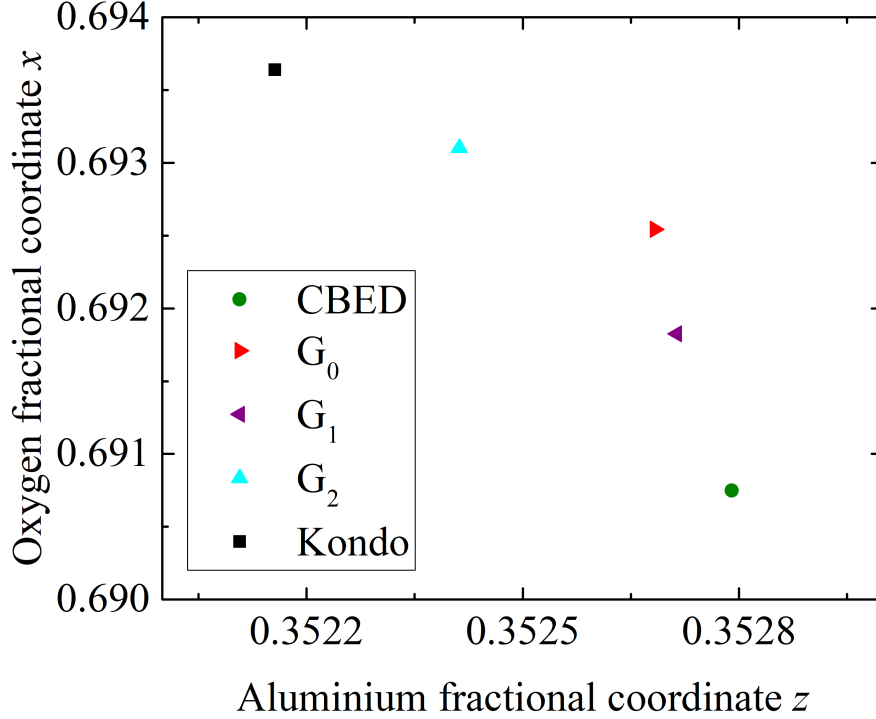


Figure 3.10: Comparison of refined z_{Al} and x_O values between CBED (0.9%), G_0 (0.9%), G_1 (0.6%) and G_2 (0.6%) datasets. Fit values, f , are in brackets

The CBED and G_0 refinement in Figure 3.10 may be of particular interest to beam sensitive material studies where the short electron beam illumination timescales required to sample undamaged material render D-LACBED inadequate. Conventional CBED and LACBED [5] could be used in its place, since it is possible to record a single pattern in a short time. Figure 3.10 also raises the fundamental question of the benefit of more information in D-LACBED ACR, besides an almost negligible increase in accuracy.

To better understand the limitations of CBED data, we explored the fit values around the refined fractional coordinates. This should provide evidence on the sensitivity of each dataset to inaccurate initial models. The fit indices over a grid of 32x32 simulations for each dataset seen in Figs. 3.8 and 3.9 are shown in Fig. 3.11.

It is clear from Figure 3.11 that an initial model with fractional coordinates within ± 0.02 and ± 0.05 (± 26 and $\pm 24 pm$) of the global minimum for z_{Al} and x_O respectively will result in a successful refinement using any of the datasets.

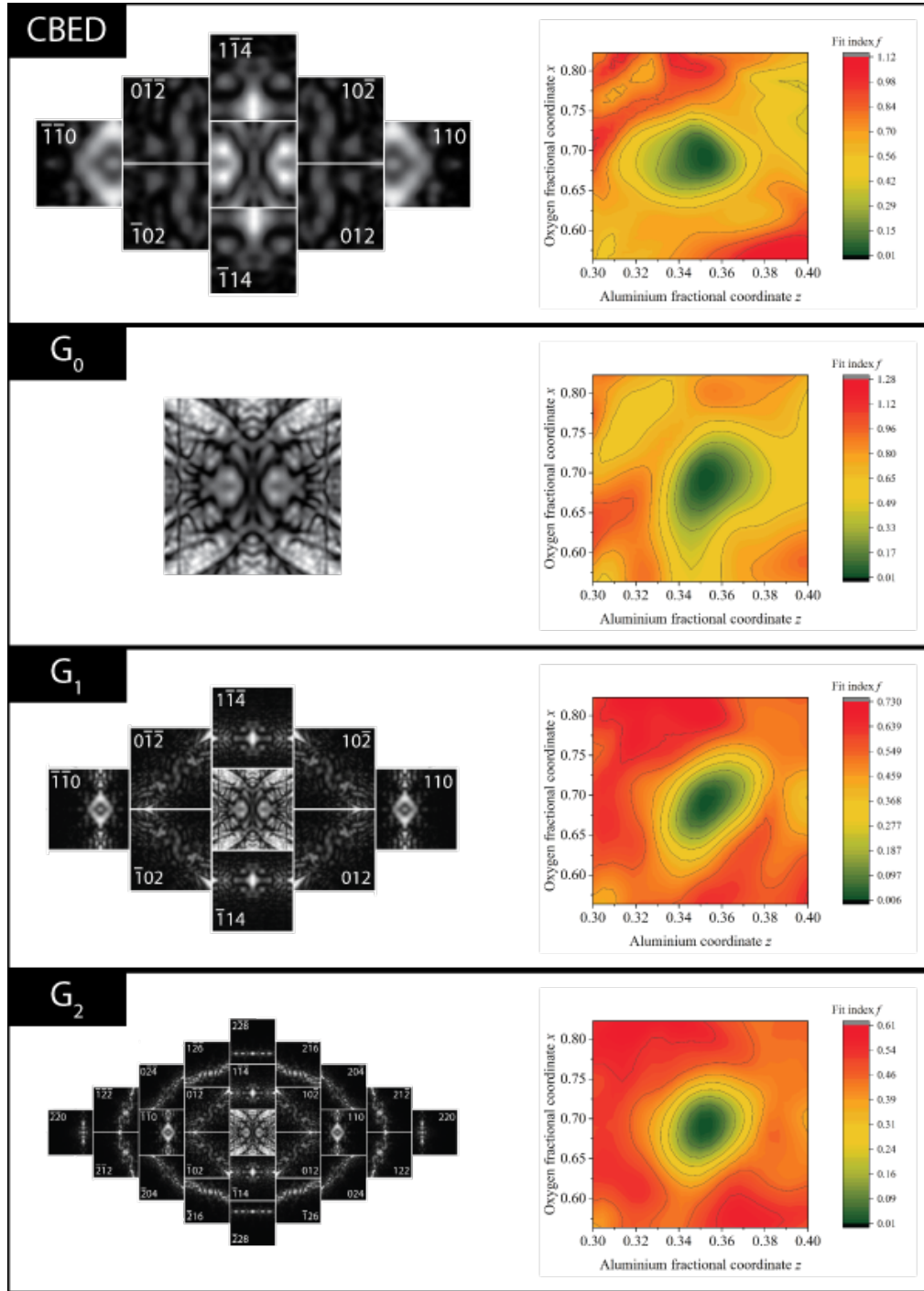


Figure 3.11: left) Set of four datasets right) Their corresponding fit spaces. An evenly spaced grid of 32 by 32 simulations were carried out and a 2D spline applied

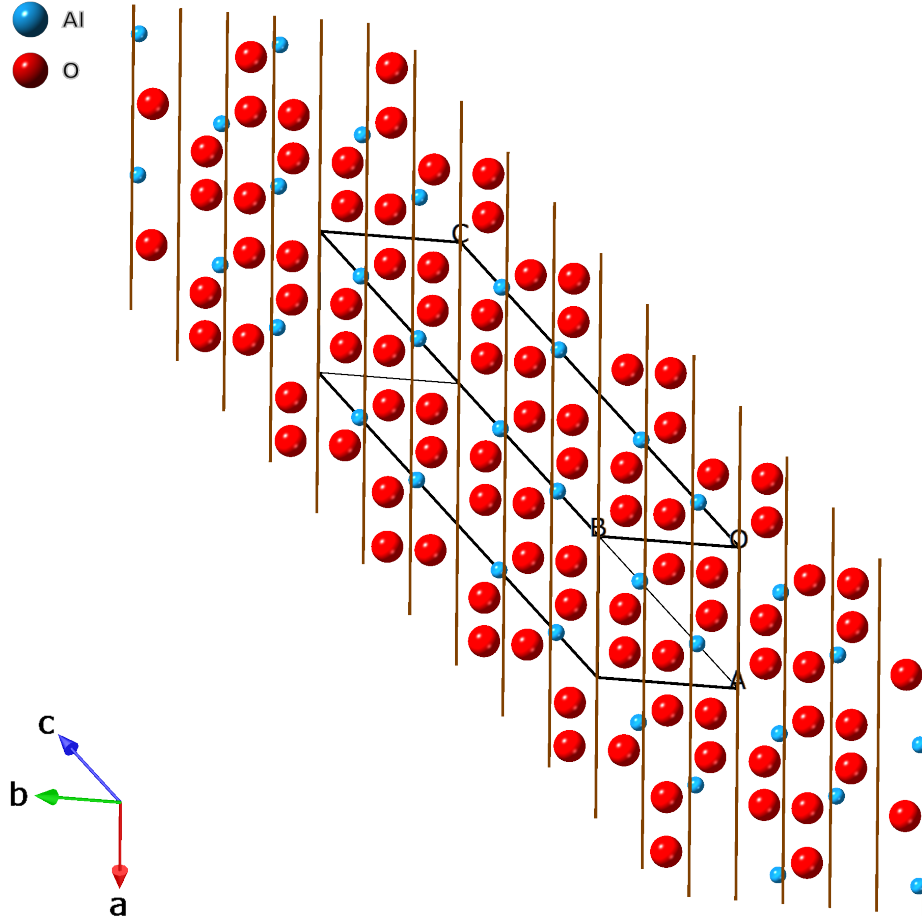


Figure 3.12: Corundum projection down the $[2\bar{2}1]$ direction. 036 lattice planes have been overlaid.

The CBED and G_0 datasets both show the small areas which will result in a successful refinement. Local minima are prevalent throughout the fit space that could trap a refinement using gradient descent algorithm. The correct solution however has a deep and well-defined minimum that could be found by other minimisation methods.

We see a marked improvement in the area that would give a successful starting position for gradient descent as we increase the number of g-patterns to 9 (4 independent) in G_1 and 25 (9 independent) in G_2 . Importantly, the local minima present in both the CBED and G_0 fit areas either disappear or become much shallower, while the global minimum remains steep in relation. The global minimum

curve also appears to be more symmetrical – the gradients of z_{Al} and x_O fits either side of the absolute minimum point appear to become more equal. A reduction in the change of gradient for both directions is also apparent. An increase in convergence angle (i.e. comparing the CBED and G_1 datasets) seems to have a similar effect to the number of g-patterns.

The increase in symmetry of the fit spaces seen in Fig. 3.11 may have further implications. The unit cell dimension along the z-direction is approximately twice as long as the dimension in the x & y directions. A movement down the z fractional coordinate of the aluminium atom in picometers corresponds to approximately twice the distance for the same movement in x & y fractional coordinate. We should therefore see a 'valley' for all our fit spaces along the x-z direction - which while crude, is seen in the fit spaces up to G_1 . However in the G_2 and G_3 plots (Fig. 3.3) the valley seems to progressively disappear. This suggests that reflections down the systematic row $n(012)$ may be more sensitive to the positions, as the planes in the corundum structure are linked to this ratio. To confirm this, fit spaces omitting the $n(012)$ systematic row of reflections as well as their corresponding structure factors should enhance the valley seen in the fit spaces of Fig. 3.11.

We can see from Fig. 3.12 and the aluminium delta plot (Fig. 3.5) that the 036 planes look to be sensitive to z_{Al} . However, as mentioned previously, portions of all g-patterns within Fig. 3.5 seem to show strong sensitivity to z_{Al} . Perhaps most notably, an accurate result can be obtained using only the straight through D-LACBED g-pattern - emphasising the prevalence of dynamical contributions to the intensity within every D-LACBED g-pattern.

From comparison of Figures 3.4 & 3.5 we seem to see the distribution of delta values in z_{Al} to be nearer each maximum than the values in x_O . This may indicate why we see the symmetrical fit space in Fig. 3.3. Down the $[2\bar{2}1]$ direction, D-LACBED patterns may be twice as sensitive to the aluminium position than the oxygen one. Exploration of different zone axes will help determine whether this is the case (see Section 3.5).

Experimentally, it seems advantageous to collect D-LACBED data with as large a convergence angle and as many g-patterns as possible. While shown to improve the accuracy a little, the greatest benefit of an increase in data is the improved reliability of the refinement obtained by removing or reducing the local minima, i.e.

- i) The initial model does not need to be as accurate to produce a successful refinement

- ii) If the refinement does become stuck in a local minimum, they are easier to characterise based on their reduced gradients and depths

For a multidimensional problem with many free parameters, identifying and avoiding local minima may be crucial. While the addition of information flattens out the parameter space making it easier to identify the global minimum once found, Fig. 3.11 shows it may not significantly increase the area where the gradient descent can find the global minima.

Nevertheless, the fit space shows a very slight average gradient towards the global minimum, indicated by the reduction in the maximum fit in Figure 3.11 from the CBED to G_2 datasets. Since one would expect the fit to get worse the further the aluminium and oxygen atoms get from the unique solution, a secondary overall fit curve may be present in all datasets, spanning the entire unit cell. This may become more pronounced at higher convergence angles and g-patterns. It would be interesting to perform a grid simulation over the full unit cell for z_{Al} and x_O to examine this relationship. It is likely that the trend of reduction and flattening of local minima will continue with a greater and more diverse dataset. Therefore, collecting multiple zone axis data will likely improve the accuracy as well as allow more error in the initial model.

This study has only considered the sensitivity of the sets of g-vector orders. An investigation of the sensitivity of individual g-patterns or indeed areas of intensity within the patterns will help future studies in ACR. For example, it may be the case where certain g-patterns are particularly sensitive to certain atomic coordinate movements or transitions.

Another avenue to explore is improvement of the minimisation algorithm. The location of the best-fit solution only improves a small amount with a greater amount of data. A genetic algorithm, or other approach capable of avoiding local minima, would be able to find the global minimum.

Crucially it seems D-LACBED data provides a smooth fit space, broad enough to ensure that the global minimum can be found. This is important for multidimensional refinements where the chance of refining to a local minimum is greatly increased.

3.5 Future work

To solidify D-LACBED ACR as a technique, data from other zone axes of corundum can be refined to explore the sensitivity of different zone axes as well as test the reproducibility. The aim should perhaps not be to find the zone axis which gives the greatest sensitivity to atomic positions, but rather the worst. For instance, a series of zone axes at an increasing angular difference to the [001] direction (e.g. [013], [012] and [011]) for corundum will be progressively less 'blind' to the free z-coordinate of the aluminium atom. A G_3 fit space similar to the one shown in Fig. 3.11 should give an indication of the extent of dynamical diffraction in D-LACBED patterns. We should expect a 'valley' along the well defined oxygen position. The position and shape of this valley will indicate the dynamical sensitivity of D-LACBED patterns. A study like this may show the advantages of D-LACBED over CBED for atomic refinement. It would be interesting to observe the $\langle 001 \rangle$ fit space with HOLZ structure factors included. This would confirm such directions are truly 'blind' to atomic position refinement.

A study of the least sensitive atomic refinement zone axis in corundum should provide better context for future atomic refinement studies using D-LACBED. A baseline precision will be determined i.e. if it is shown that picometre precision (the standard for other electron diffraction techniques - see Chapter 1) can still be achieved with the least sensitive zone axes future studies may need to only avoid blind zone axes.

There is also value in attempting to find the most sensitive zone axes. The Kondo paper [204] describes a very small (sub-pm) movement of the aluminium atom along the z-axis in relation to temperature. At the moment D-LACBED data from the $[2\bar{2}1]$ zone axis does not seem to be precise enough to determine these small deviations in structure. A zone axis like the [110] may provide an interesting comparison, both in the accuracy of the final refinement result and the shape of the fit space (which may help identify sensitive reflections).

A material like chromium oxide could then be studied, which has the exact same structure (with chromium on aluminium sites) apart from very small differences in atomic coordinate. It was hoped this would be a part of the thesis, but due to time constraints the study was postponed.

The next stage may involve a more complex material with large lattice parameters, degrees of freedom and charge transfer between atoms. The large lattice parameter ensures CBED is unviable, the degrees of freedom tests the viability of the ZAP and the strong bonding influences will test the capability of the IAM. The

material should also be well studied to ensure comparable accuracies.

Further to the above, to ascertain the limit and test the ultimate accuracy of D-LACBED, a study similar to SK's [204] could be undertaken, where the position of the aluminium atom in corundum changes very slightly in accordance to temperature.

In the long term, a technique similar to EDT (or Automated DT) [54,198,199] could be developed to probe more reciprocal space. A series of CBED patterns could be taken in a 'line' along one rotational axis of goniometer tilts in the following process. Once a location on the sample for data collection has been identified the first CBED pattern is taken. The convergence angle of the CBED patterns would be adjusted so each disc is just about touching. The goniometer is tilted along a chosen axis to just inside the half-convergence angle (dependent on precision of goniometer). Scanning transmission electron microscopy mode (at low magnification) could then be used to reproduce an image and therefore show the previous CBED site. The microscope is then realigned as shown in Kolb et al. [54,198,199]. A second CBED pattern is then taken in the same location. This process is repeated for x tilts.

Once completed, each CBED disk in the tilt series can be stitched together to produce a CBED 'line' through reciprocal space. Unlike RED or PEDT, there is no need to tilt the beam at any point. There are no 'missing wedges' either along the tilt direction as the reciprocal space has been probed entirely by the focused beam of the CBED pattern. The technique could possibly be extended to different geometries to produce very large angle D-LACBED patterns, potentially over multiple zone axes.

However, many limitations already come to light: the offset of alignments due to changing height, thickness and time on the sample, the data set would be huge and most likely require a large amount of memory and processing power, and potentially many other problems. It is however a direction to travel down towards further probing of reciprocal space.

Another possible future route is using D-LACBED patterns for structure solution before eventual refinement. Past studies of CBED have used the three-phase invariant geometry inherent in dynamical diffraction to extract the phases of the structure factors [211–213]. D-LACBED was then recently used by Guo et al. [118,147]. Potentially, estimations of the unit cell and structure factor phases could lead to initial guesses of the atom coordinates. These coordinates could then be refined to a proper solution. The three-beam phase invariant method relies on quite strict geometry however, often off zone-axis. The phases are also quite imprecise. A general method for extracting the structure factor phases from ED patterns was first proposed by Allen et al. [141,143] and Spence [142], however to this date a concrete generalised theory remains elusive [38].

Chapter 4

Debye-Waller factor refinement

“[Fejér] goes on to challenge Bocher’s other claims, and in fact politely calls Bocher a liar. For those interested in how a gentleman did this in 1914, we quote (translated from the German). ‘I could therefore with pleasure verify that after the publications of Herr T.H. Gronwall (1912) and myself (1913), certain questions can in fact today be handled with the greatest ease, for which however in the year 1906 every trace of a hint was lacking.’ ”

- Edwin Hewitt & Robert E. Hewitt

The Gibbs-Wilbraham phenomenon: an episode in Fourier analysis [214]

Knowledge of the atom positions forms the basis for further exploration of a crystal structure. The analysis can either extend to the refinement of the underlying functions within each of the electron structure factors, Eqs. 4.1 & 4.2, or to the Fourier components as a whole. For the latter, electron structure factors can give insights into charge density (and therefore bonding) using the Mott-Bethe relationship [215], where lower order structure factors taken from reference X-ray datasets are refined using CBED ZAP’s [125–127, 129, 216–218]. However, it is possible the combination of disparate data – i.e. from different samples and techniques – may introduce discontinuities in the Fourier series and thus artefacts in real space (the Gibbs-Wilbraham phenomenon is discussed later in Section 4.4.1) which can be of the same magnitude as the bonding effects being sought [219]. Therefore, we have decided to explore the underlying functions within the structure factors. The proficiency of D-LACBED to determine atomic positions was shown in the previous chapter and the model we use for inelastic scattering (Bird & King [183], see Chapter 2) does not require any variables to be refined. Therefore, the only other property to explore is the temperature factor.

Revisiting equation 2.5 from Chapter 2 we have

$$U_{\bar{g}} = \left(\frac{2m|e|}{h^2} \right) \left(\frac{h^2}{2\pi m_0 e \Omega} \right) \sum_i [f_{ei}(\bar{q}) + i f'_{ei}(\bar{q}, \langle \bar{u}_i \rangle)] \exp [-2\pi i \bar{g} \cdot (\bar{r} + \langle \bar{u}_j \rangle)] O_i \quad (4.1)$$

where $U_{\bar{g}}$ is the modified structure factor, m_0 is the rest mass of the fast electron and m is its associated relativistically corrected mass, e is its charge, h is Planck's constant, Ω is the unit cell volume, $f_{ei}(\bar{q})$ is the electron's scattering factor, $f'_{ei}(\bar{q}, \langle \bar{u}_i \rangle)$ is the electron's absorptive scattering factor, $\langle \bar{u}_i \rangle$ is the thermal displacement vector where $\langle \dots \rangle$ indicates time averaging, \bar{g} is the reciprocal lattice vector with \bar{r} its associated real space vector and O_i is the occupancy.

The Einstein model of independent, harmonic thermal vibrations, allows $\langle \bar{u}_i \rangle$ to be converted to an isotropic temperature factor, $\exp(-B_i \bar{q}^2)$, where the DWF, $B = 8\pi^2 \langle u^2 \rangle$ is determined by the mean square thermal displacements $\langle u^2 \rangle$, i.e.

$$U_{\bar{g}} = \left(\frac{2m|e|}{h^2} \right) \left(\frac{h^2}{2\pi m_0 e \Omega} \right) \sum_i [f_{ei}(\bar{q}) + i f'_{ei}(\bar{q}, B)] \exp [-2\pi i (\bar{g} \cdot \bar{r}_i)] \exp(-B_i \bar{q}^2) O_i \quad (4.2)$$

Here we test the ability of D-LACBED to measure the Debye-Waller factor for simple well-known materials. We study Copper, a simple metal where the IAM (which does not account for bonding and anharmonicity) will likely work. We then move on to GaAs, a covalently bonded material with some ionic character [220], to see how the IAM is affected. We compare our results to the X-ray analysis on GaAs by Stevenson [221] which suggests an accurate measurement of the DWF requires a cubic anharmonic correction and consideration of bonding.

4.1 Copper

The simplest possible test of the capabilities of D-LACBED would be a material that has a limited number of parameters that can be refined and conforms reasonably well to the assumptions of the theoretical model (i.e. spherical, neutral atoms that are well-described by calculated scattering factors, with harmonic thermal vibrations). Most monatomic metals fit this description and we choose copper here due to its ready availability. The copper sample was prepared using standard methods (Chapter 2) the only difference a brief (~ 20 second) etching in nital solution before microscope insertion to remove any oxidation layers.

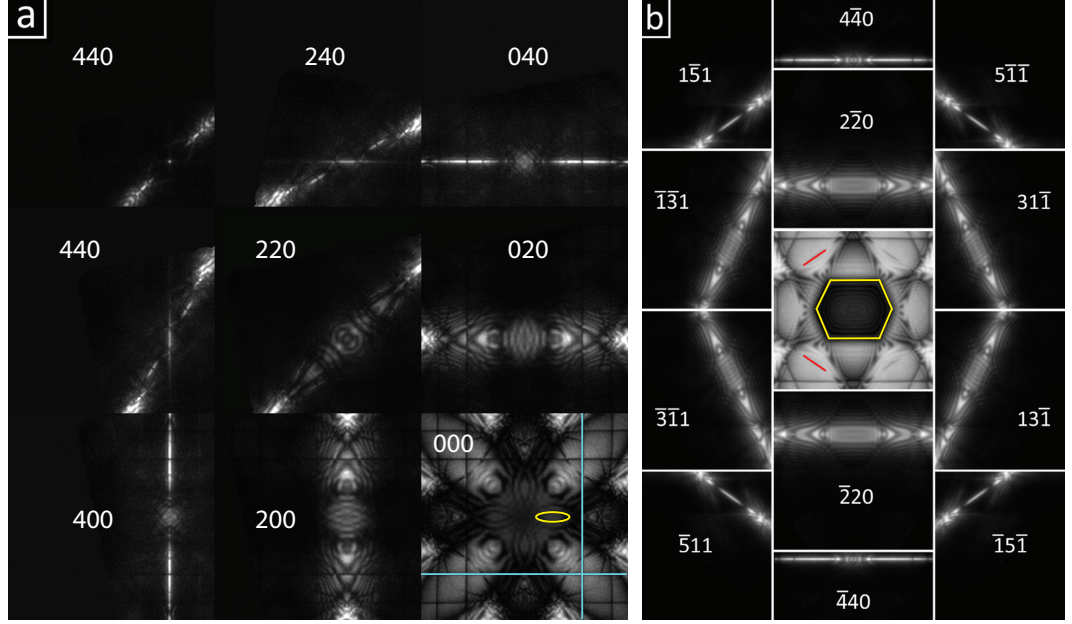


Figure 4.1: Room temperature D-LACBED from Cu (a) Nine patterns from Cu [001], angular range 41.3 mrad. The cyan lines highlight the 400 type deficit lines, the yellow oval highlights a 'flower petal' band (see text) (b) Thirteen patterns from Cu [114], angular range 44.2 mrad. The red lines highlight the $1\bar{5}1$ type deficit features. The yellow hexagon highlights the 'lozenge' (see text). All patterns are normalised for display here to the visible display range and have applied gamma 1.5 to allow features in darker parts of the image to be seen more easily.

The atomic coordinates of this fcc crystal are fixed, leaving only Debye-Waller factor as a measurable parameter. We collected data from [001] and [114] zone axes as examples of relatively dense low-index diffraction patterns and more sparse mid-index patterns respectively. The small lattice parameter of Cu results in relatively large Bragg angles (e.g. $2\theta_{002} = 13.8$ mrad), allowing large convergence angles without overlapping discs in the CBED pattern. We thus used a beam half-convergence angle of 5.9 mrad, requiring only ~ 120 CBED patterns to produce D-LACBED data extending beyond 40 mrad. Figure 4.1 shows some of the D-LACBED patterns collected from Cu at room temperature (303 K) from the [001] and [114] zone axes. Data similar to Fig. 4.1 was collected at 373K and then at temperature increments of 100K to a maximum of 753K.

It was found that optimised simulations required > 210 Bloch waves for [001], but only > 120 for [114] data (see Section 5.10). The 400x400 pixel simulations at a single thickness required 1 minute 10 sec for Cu [001] and 50 seconds for Cu [114] using 256 cores.

The fit to simulation was refined by varying the isotropic Debye-Waller factor B . The plots in Figs. 4.2 & 4.3 show the fit index (Eq. 2.22 – Chapter 2) for the different temperatures as a function of B . This demonstrates a well-behaved parameter space with a single minimum; fit indices were typically below 5% but increased slightly for higher temperatures. The simple gradient-descent algorithm from *felix* (Chapter 2 & [149]) was used, typically requiring less than ten iterations to obtain a precision of 0.01\AA^2 .

Experimental 000 patterns and best fit simulations for the Cu [001] and [114] data are also shown in Fig. 4.2 & 4.3 respectively. We see thermal scattering's primary effect, the intensity out of the Bragg condition and into diffuse background, through the subtle weakening of sharp features as the temperature increases. For example the horizontal and vertical 400-type (Fig 4.2) or diagonal $1\bar{5}1$ type (Fig 4.3) deficit lines (Fig. 4.1). These types of lines have been used for measurement of the DWF via the critical voltage method [115, 222], normally using second order reflections. See Chapter 1 for more detail. The most obvious change with increasing temperature however, is the darkening of the 'flower petal' bands in the [001] data (see Fig. 4.1). We can also attribute this to increased thermal scattering, but on the absorptive potential (see equation 4.2). A larger amount of thermal scattering gives increased absorption near the centre of each g-pattern where strong channelling occurs. That is, as the electron beam propagates through the crystal, channelling (and the scattering of the electron beam to high angles i.e. absorption) occurs more strongly along atom columns with a smaller interatomic distance along the direction of propagation. Since this distance is larger for higher index beams, the darkening of the central bands is less obvious in the mid-index [411] data, although we see it is still significant in the central lozenge-shaped region (see Fig. 4.1) along the precise zone axis.

Simulated 000 patterns at fixed specimen thickness with B (1) to (4) are also shown in Fig. 4.2. The difference images beneath show the changes that affect the fit index f , i.e. variations in relative intensity when each pattern is normalised to the same range, rather than changes in absolute intensity. This is useful here since it corresponds to the normalised cross-correlation used to calculate the fit index (Chapter 2). These complicated fringe patterns exhibit both increases and decreases in relative intensity. Additionally, the 400-type deficit lines show a relative decrease in intensity at low B that becomes a relative increase at higher B , showing an interplay between absorption and thermal effects. The initial darkening of the line is caused by absorptive scattering, which affects large g-vectors more readily than small ones at reduced values of B [183]. At higher temperatures however,

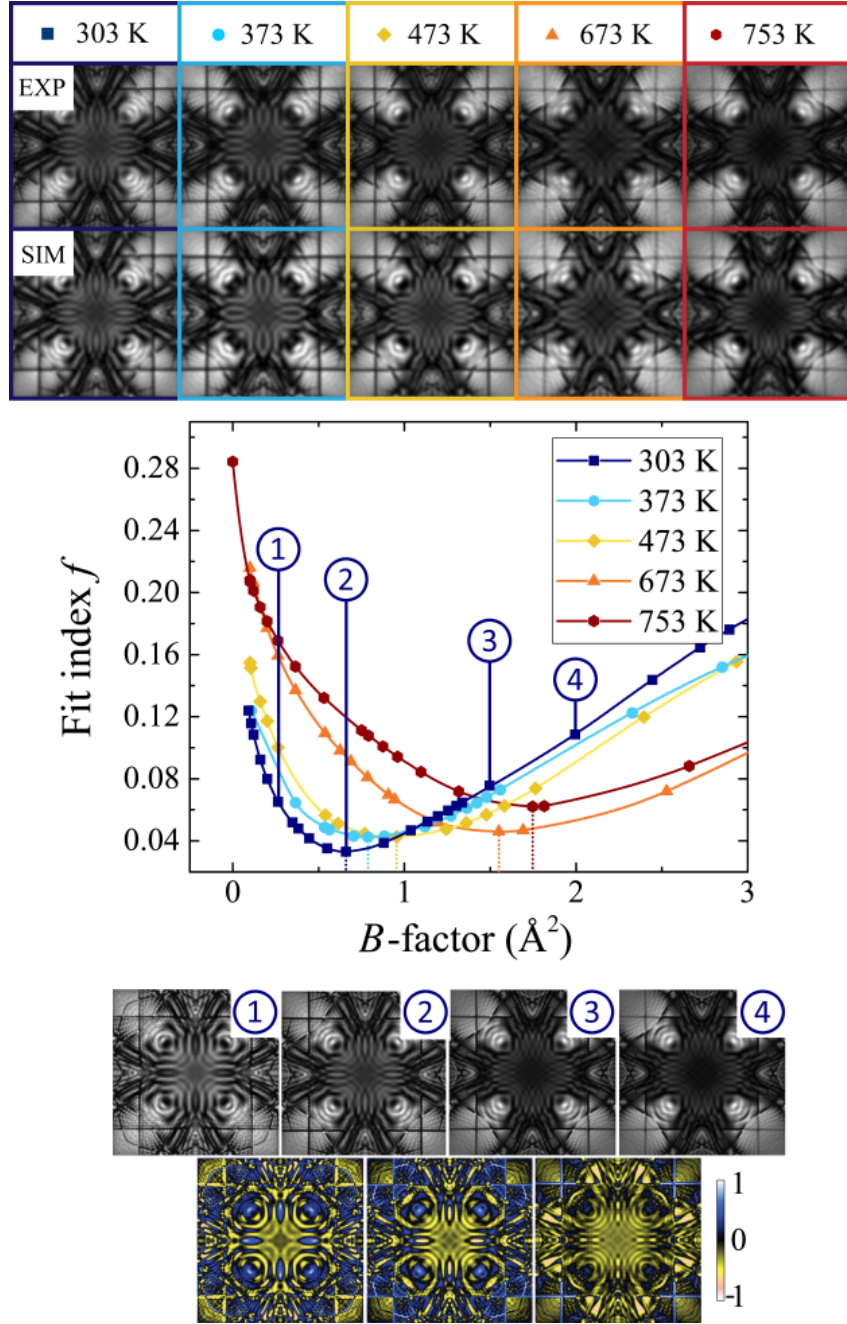


Figure 4.2: Upper: the experimental 000 D-LACBED patterns and best fit simulations at different temperatures for Cu [001]. Centre: fit indices f (all patterns) for a range of simulations at each temperature. Lower: simulated 000 patterns (1) – (4) and the difference between them, where yellow shows a decrease, and blue an increase, in relative intensity with increasing Debye-Waller factor B .

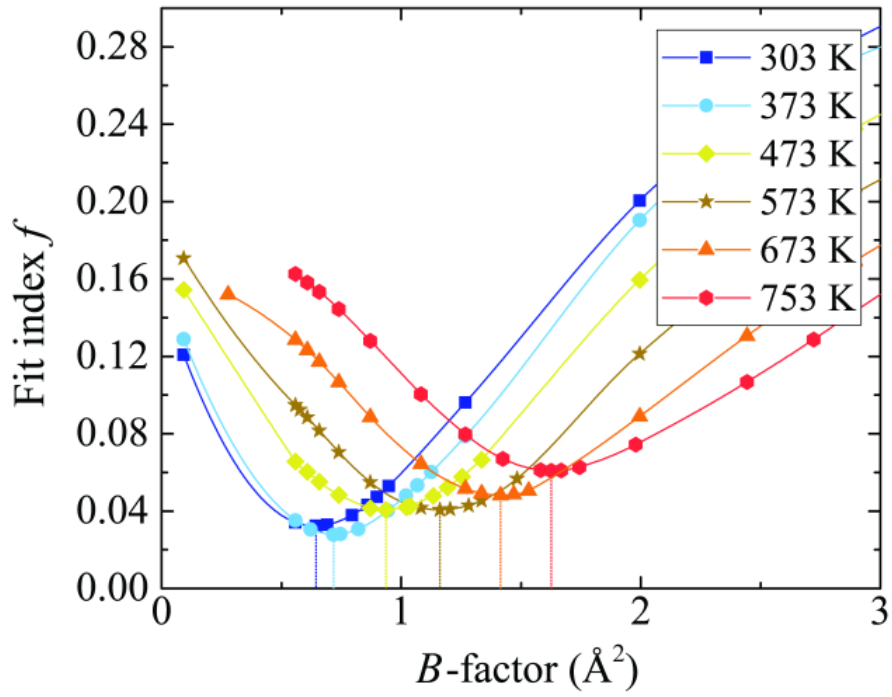


Figure 4.3: Upper: the experimental 000 D-LACBED patterns and best fit simulations at different temperatures for Cu [114]. The small purple circles show the fading of the $1\bar{5}1$ deficit feature as temperature increases Lower: fit indices f (all patterns) for a range of simulations at each temperature.

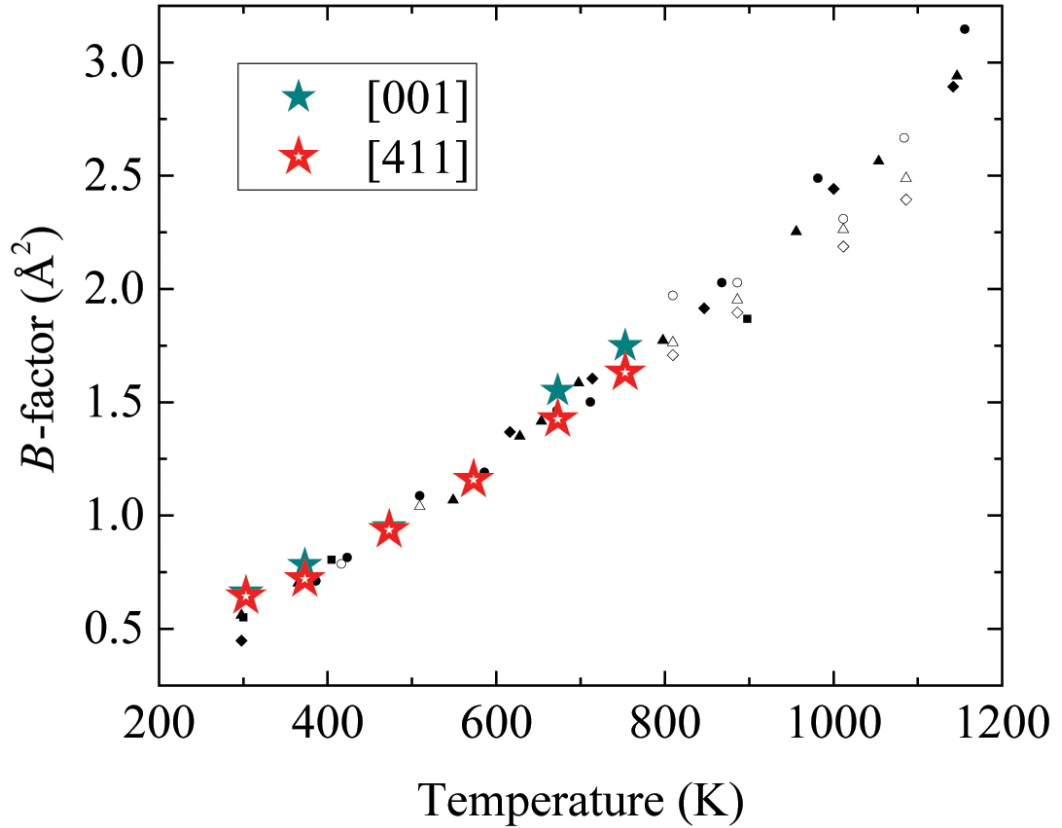


Figure 4.4: Experimental determinations of Debye-Waller factor B in copper D-LACBED data from [001] (solid stars, green) and [114] (hollow stars, red). Errors are smaller than the data points, typically $\sim 0.01\text{\AA}^2$. Many previous measurements of B using X-ray diffraction and the Mössbauer effect are shown in black and white (from ref. [223])

these diffraction features become continuously weaker (brighter) as expected from Eq. 4.2. It would be interesting to explore simulations using more recent models of absorption that include ionisation and Compton scattering for their effect on D-LACBED patterns (see Section 2.4)

The measured Debye-Waller factors for copper as a function of temperature are shown in Fig. 4.4, together with historical data collated by Shukla [223]. Excellent agreement is found, showing that D-LACBED data gives accurate temperature factors in this simple metal. Nevertheless, there is a small disagreement between measurements from the [001] and the [114] measurement, particularly at higher temperatures. It is well-known that the deviation of the data in Fig. 4.4 from a straight line is mainly due to anharmonic thermal vibrations, which become more significant at higher temperatures [224]. This cannot be captured by the single DWF, which

only describes harmonic vibrations; attempts to fit experimental data from an anharmonic material using a Debye-Waller factor will effectively give slightly differing answers for different g-vectors. This can be used to advantage in measurements using the Mössbauer effect [224], where comparison of Debye-Waller factors obtained with first and second order diffraction can be used to calculate anharmonic components. Such an approach is unlikely to be successful in the case of D-LACBED data, since dynamical diffraction mixes intensities between the different reflections (for example, in Fig. 4.1a the transfer of the ‘flower petal’ shaped intensity from the centre of the 200 type patterns to the centre of the 000 pattern is quite clear). Here, the DWFs are derived from a fit to all the D-LACBED patterns at a zone axis and are thus some kind of average measurement. Nevertheless, since the patterns all lie in the ZOLZ, the [001] and [114] D-LACBED data cover different parts of reciprocal space and it is perhaps not surprising that small differences in measured Debye-Waller factors appear when anharmonic thermal vibrations are known to be present.

4.2 Gallium Arsenide

We thus examine a material that has smaller, but still significant, bonding effects without the complication of any atomic coordinate refinement, i.e. GaAs. The smallest Bragg angle in the GaAs $[1\bar{1}0]$ pattern is $2\theta_{111} = 7.7$ mrad at 200 kV, requiring much smaller convergence angles than Cu to avoid overlapping discs in the CBED pattern. We used a beam half-convergence angle of 1.16 mrad and 1681 CBED patterns to produce D-LACBED data extending beyond 40 mrad. The reconstructed D-LACBED patterns had dimensions of 320x320 pixels. Figure 4.5 shows eighty-five patterns taken at room temperature (29°C) and 200°C. The vertical (110) mirror symmetry reduces the number of unique patterns to forty-nine.

Optimising the fit between simulation and experiment for B_{Ga} and B_{As} using two-dimensional gradient descent gives a unique solution at $B_{Ga} = 0.83(2)$, $B_{As} = 0.68(2)\text{\AA}^2$ for the room temperature data with a fit $f = 4.61\%$ and $B_{Ga} = 1.01(2)$, $B_{As} = 0.80(2)\text{\AA}^2$ ($f = 4.96\%$) at 200°C. The normalised differentials $\delta I/\delta B_{Ga}$ and $\delta I/\delta B_{As}$ for the central few D-LACBED patterns are shown in Figs. 4.6d and 4.6e with the full symmetrically independent patterns in Figs. 4.7 & 4.8 respectively. The influence of DWFs here on the ZNCC, while significant, is roughly two orders of magnitude smaller than the effect of atomic coordinates in the corundum refinement. As observed for the above refinements, changes in the parameters B_{Ga} and B_{As} produce a complicated pattern of increases and decreases in intensity resulting in an

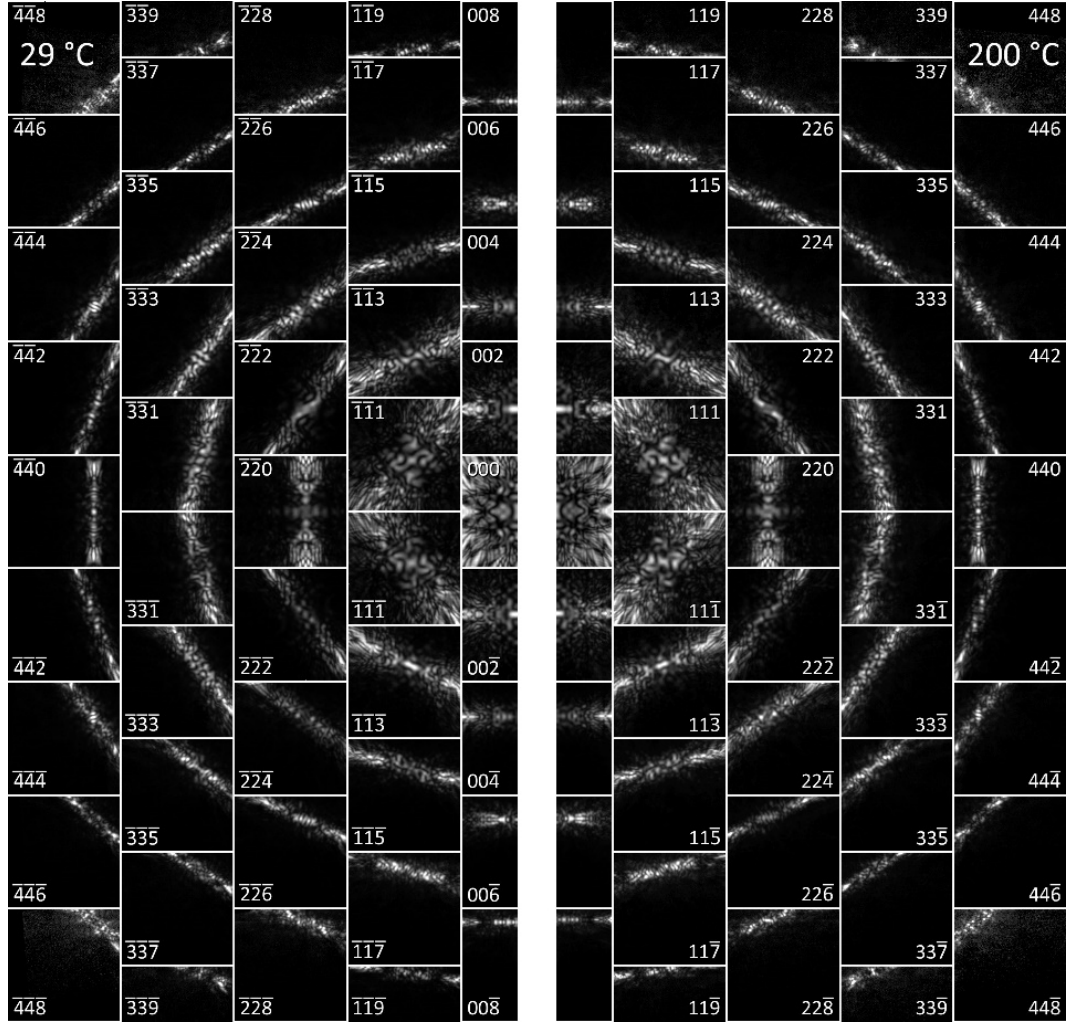


Figure 4.5: Eighty-five D-LACBED patterns from $[1\bar{1}0]$ GaAs. Left: room temperature (29°C). Right: 200°C . Each pattern has an angular width of 46.2 mrad

effective independence of the fits shown in Figs. 4.6a and 4.6b. Typically, fewer than twenty iterations were required to find the best fit to a precision better than 0.01 \AA^2 . While the time to obtain a result is acceptable for this simple two-parameter problem, it is still quite long if more complex problems are to be tackled. Thus, we tested the reproducibility of the result using the 200°C data binned by 2 (160×160 pixels) and binned by 4 (80×80 pixels), giving simulation times of ~ 22 seconds and ~ 12 seconds respectively using the same simulation conditions. These gave $B_{Ga} = 1.04$, $B_{As} = 0.78 \text{ \AA}^2$ and $B_{Ga} = 1.11$, $B_{As} = 0.88 \text{ \AA}^2$ respectively, i.e. an error of up to 10% for a reduction in time of roughly eight times.

Comparison of these results with literature values (Fig. 4.6c) shows considerably worse agreement than was obtained in the case of Cu. The most accurate measurement of Debye-Waller factors in GaAs using X-ray diffraction gives $B_{Ga} = 0.622(3)$, $B_{As} = 0.483(5) \text{ \AA}^2$ at room temperature [221]. It was noted in this X-ray study by Stevenson that the inclusion of a cubic anharmonic parameter, β , significantly affected both B_{Ga} and B_{As} . We therefore decided to include β using Dawson's structure factor formulism [225] (Terms beyond cubic anharmonicity were found to produce negligible effects [226]). However, we found an insignificant change in both fit and Debye-Waller factor. It is possible the reason is because Stevenson's analysed many diffraction spots (5058). The majority of these had high-order g-vectors, where the intensities are known to be less sensitive to bonding, and more sensitive to the DWF (and therefore anharmonic effects). In our study however, all g-vectors are associated with low-order g-vectors where bonding is known to have a large influence, especially in electron diffraction. It was shown in Stevenson that bonding had to be considered for the quasi-forbidden reflections, $h + k + l = 4n + 2$, in order to ascertain accurate structure factors (with correct DWFs).

Interestingly, all experimental measurements are clustered along a single trend line and our measurement is fairly close to another measurement made by electron diffraction [130]. Figs. 4.6f and 4.9 show the difference between best-fit simulation and experiment at room temperature. We take the systematic nature of the residual intensity to indicate that refinement against parameters such as bonding would produce a better fit.

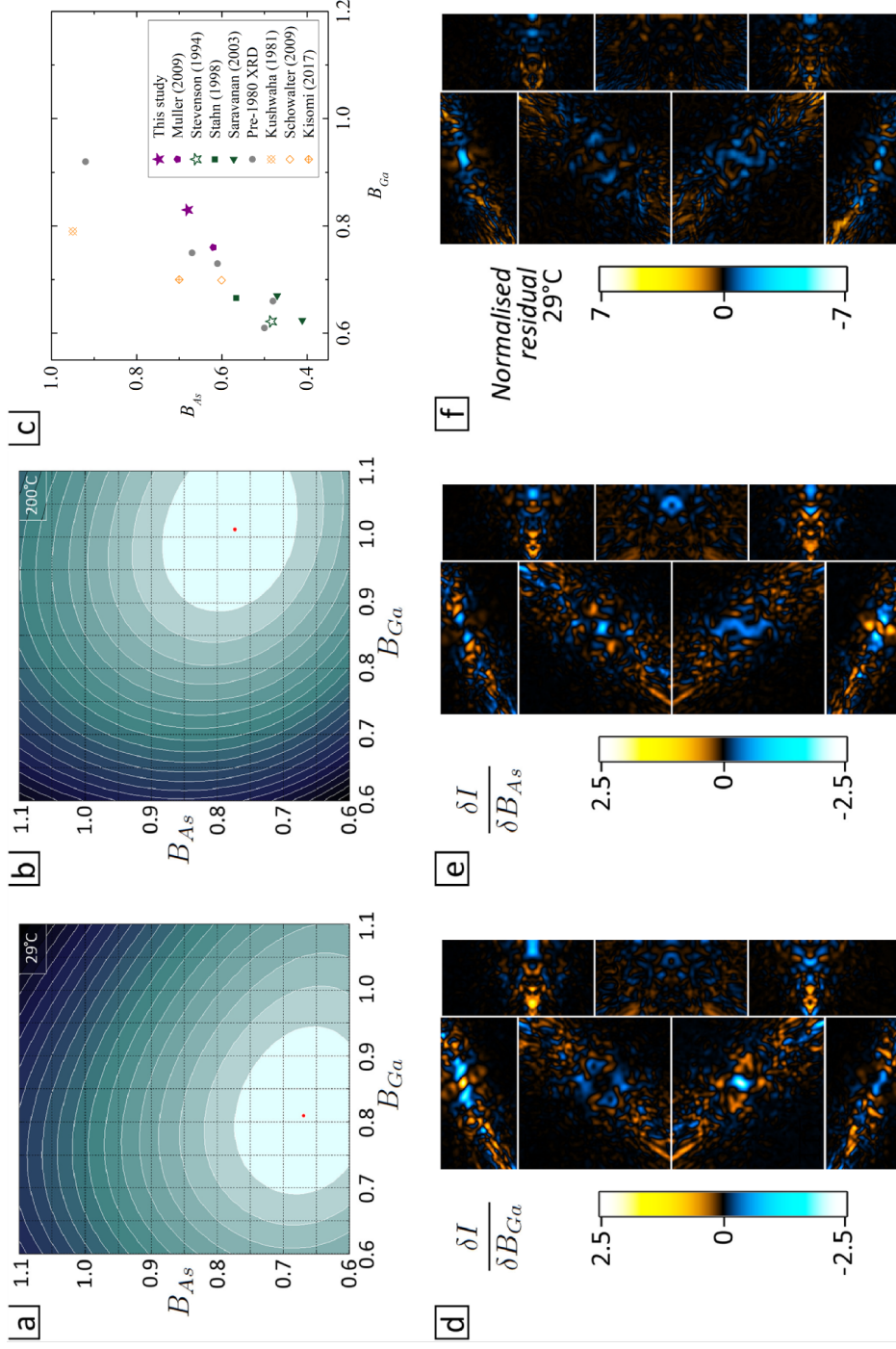


Figure 4.6: a, b: fit index f (49 unique patterns) for [110] GaAs data at room temperature and 200°C as a function of B_{Ga} and B_{As} . c: Experimental measurements of B_{Ga} and B_{As} . Green and gray = X-ray diffraction [220, 221, 227–231]; purple = electron diffraction [130]; orange = theory [232–235]. Unattributed measurements are those collated in ref. [221]. d,e: $\delta I/\delta B_{Ga}$ and $\delta I/\delta B_{As}$ in units of standard deviation. (see also Figs. 4.7 & 4.8) f: the residual between best-fit simulation and experiment, calculated in the same way as d and e. (see also Fig. 4.9)

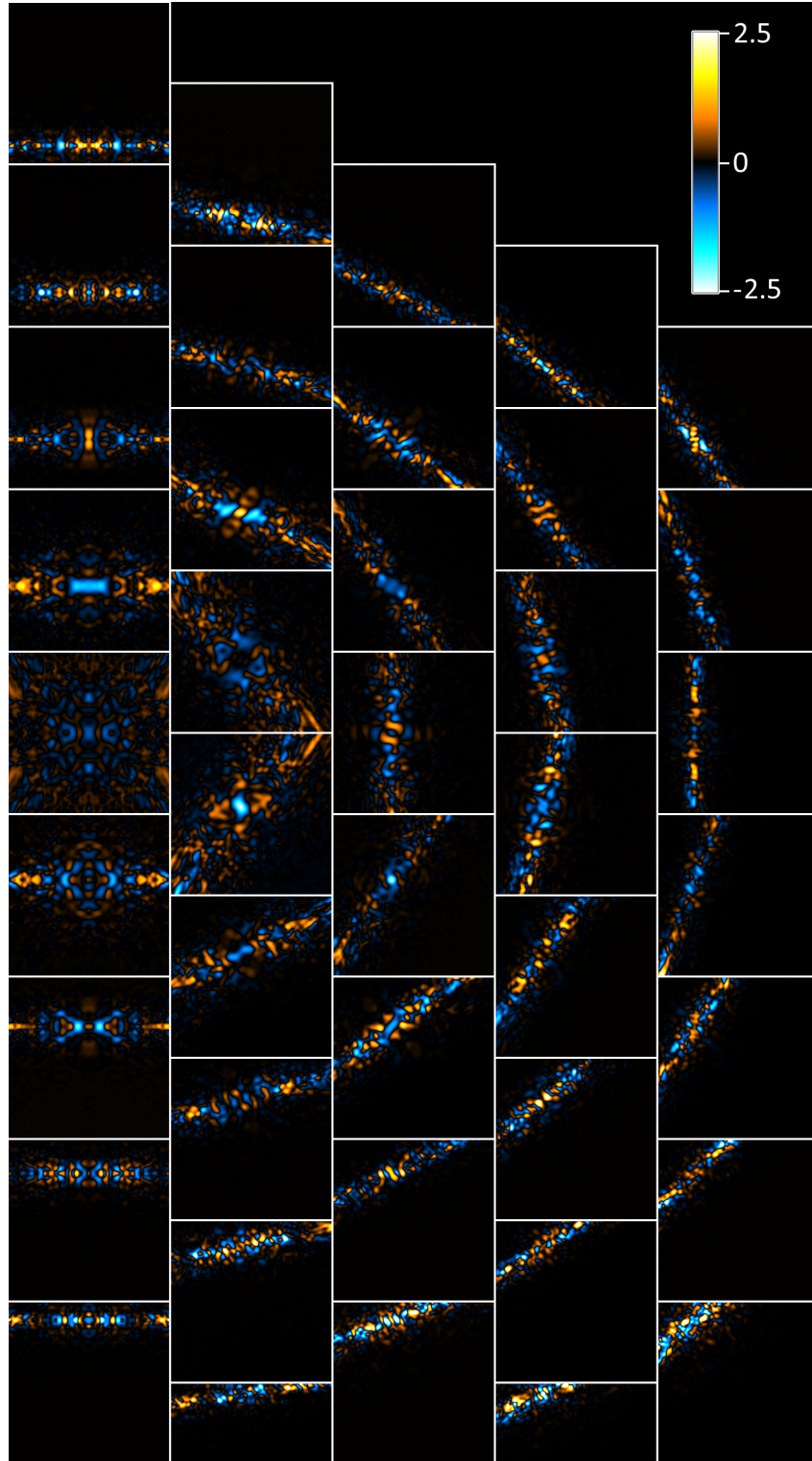


Figure 4.7: Enlargement of Fig 4.6d ($\delta I / \delta B_{Ga}$ in units of standard deviation)

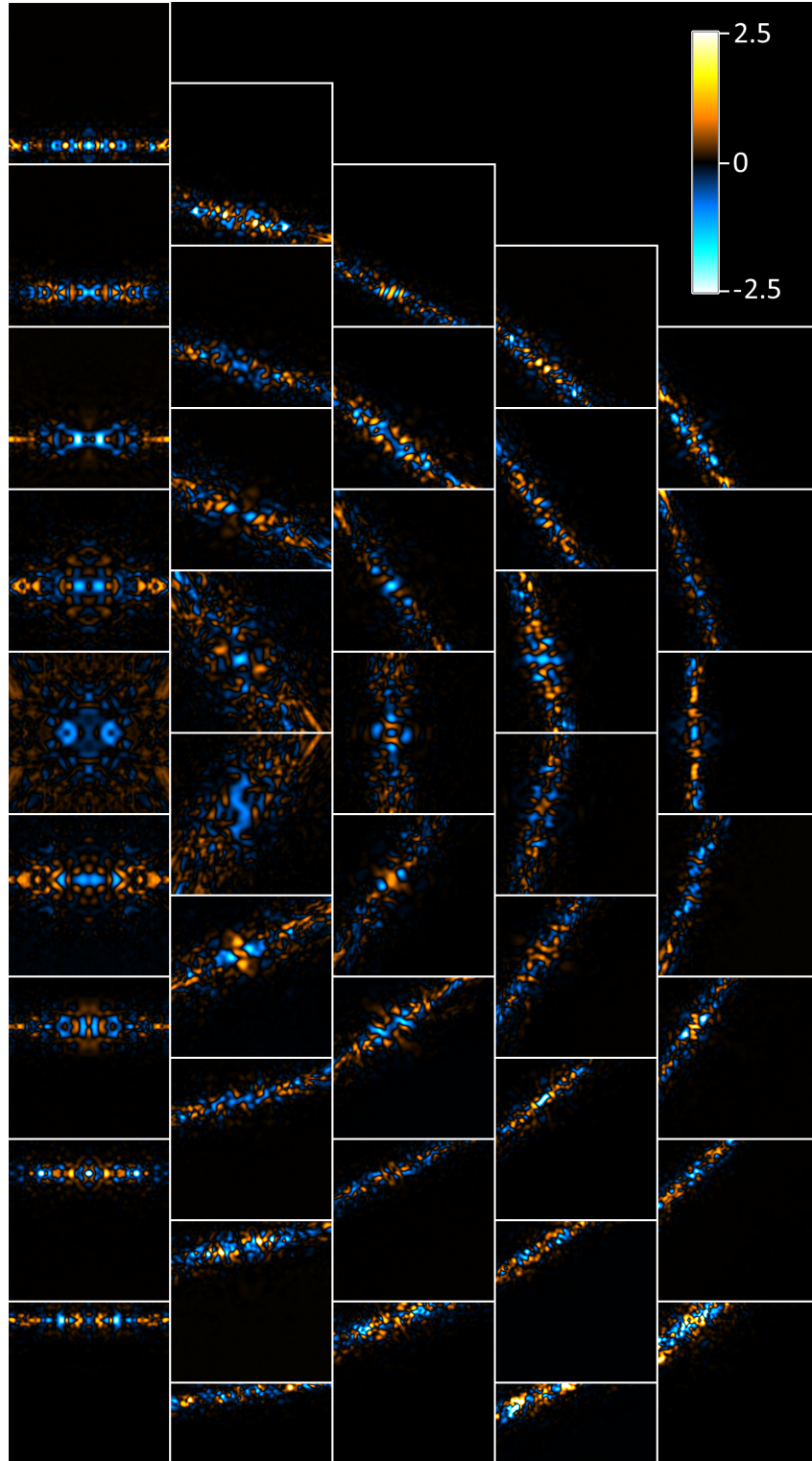


Figure 4.8: Enlargement of Fig 4.6e ($\delta I / \delta B_{As}$ in units of standard deviation)

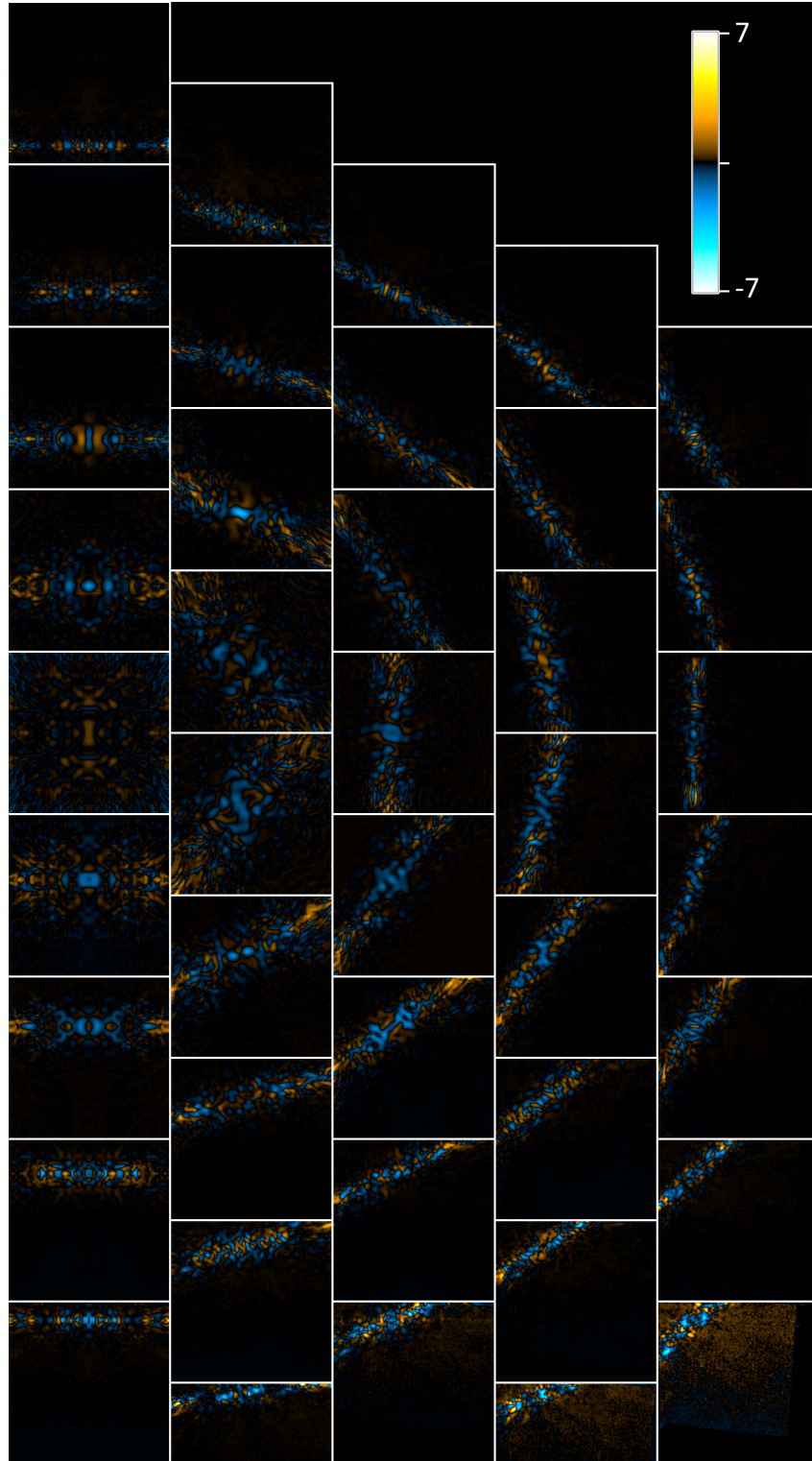


Figure 4.9: Enlargement of Fig 4.6f (Residual between best-fit simulation and experiment)

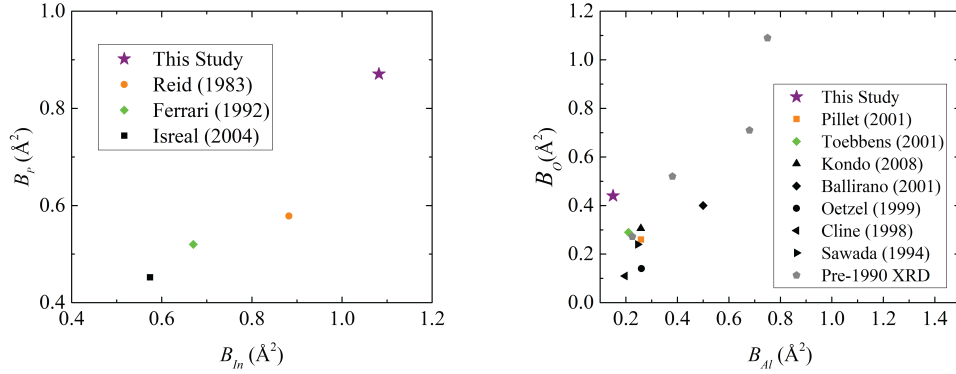


Figure 4.10: X-ray, neutron, theoretical and electron (black, green, orange and purple respectively) Debye-Waller factor measurements for InP (left) [235, 238, 239] and Corundum (right) [204, 236, 237, 240–244] at room temperature. The Pre-1990 XRD values (grey) were taken from Ballirano et al. [241]

4.3 Alpha-corundum and Indium Phosphide

In Chapter 5 we shall examine the accuracy of D-LACBED for thickness determination using an InP [100] dataset. During this study a Debye-Waller factor refinement out to G_3 was also carried out. The values obtained were $B_{In} = 1.0830$ and $B_P = 0.8716 \text{ Å}^2$ respectively. A similar refinement was carried out on the $\alpha\text{-Al}_2\text{O}_3$ A221 dataset (used in Chapter 3) which gave values of $B_{Al} = 0.15$ and $B_O = 0.44 \text{ Å}^2$. Like GaAs (Fig 4.6c), we see a marked difference between the nominal and our measured values in Fig. 4.10. For InP, which has the same structure (Zinc-blende) of GaAs, we see a similar degree of error following the familiar trend seen in the GaAs analysis. This suggests the mechanism causing the discrepancy may be the same in both materials. Any correction applied to the DWF measurement of the GaAs dataset would also likely correct the InP measurement. For corundum, which has a different structure (space group = $R\bar{3}c$) to InP & GaAs, the measurement is closer to the nominal values (Kondo et al., Toebbens et al., Pillet et al. etc. [204, 236, 237] - Fig 4.10) but is on the periphery of the trend seen from the other analyses in this chapter and is therefore worth further study.

4.4 Discussion

Often, the goal of CBED studies has been to obtain information about f_i , in particular structure factors that indicate deviations from spherical atomic symmetry [126, 245–247]. It is well-known that one has to go beyond the neutral atom approach for CBED fitting. The examples here show there is much to be gained by examining the other parameters that influence dynamical electron diffraction patterns.

Since both electron orbitals and thermal vibrations are averaged over the timescale of the measurement, it is not generally possible to separate their influence on the data. The presence of strong bonding effects is probably the reason for the inaccurate measurement of Debye-Waller factors in GaAs, InP and α -Al₂O₃ here. There are opportunities to apply multipole modelling [248, 249] or Hirshfield refinement [250] methods to electron data. While the initial investigations here of anharmonic vibrations in GaAs was shown to have limited influence on the measured DWF from D-LACBED data, further investigation should conclude whether this is the case. Other factors that can influence the result include shortcomings within the Bird and King absorption model and e.g. deviations from perfect crystal symmetry due to the presence of point defects.

Before the CBED pattern tilt series is stitched together in the D-LACBED technique (Chapter 2), the diffuse background found outside every CBED disk is measured and used to subtract the intensity caused by inelastic scattering within it. However, after application the patterns remain blurred. This is due to the PSF, symmetry averaging and any limitations found in the background subtraction process (e.g. inelastic effects such as plasmons known to produce intensity inside the disk which is uncorrelated to outside it). To compensate, a gaussian blur was applied to the simulated data with radius determined through refinement with the experimental dataset, before fitting in *felix*. While the blur was found to have a limited effect on atomic refinement, we see later it has appreciable effects in DWF refinement (Chapter 5). This may explain the small difference in DWF between this study and Muller et al. [130] in GaAs, and also a portion of the difference between the different zone axes in copper.

Since the literature for the DWF of GaAs is more thorough than InP or corundum, it makes sense to further explore analysis on the GaAs data before continuing to the other two materials. A key point of interest lies in the comparison of the GaAs and InP quasi-forbidden reflections ($h + k + l = 4n + 2$). The bonding correction first applied by Saravanan [228] in GaAs and then corrected by Steven-

son [221], was used on these reflections, which only appear in X-ray diffraction due to the difference in atomic number between the two atom types. Since the atomic number is vastly different in InP (In=49, P=15) and only slightly different in GaAs (Ga=31, As=33), comparison of these D-LACBED g-patterns between InP and GaAs may provide insight into the effect of bonding on the DWF.

4.4.1 The Gibbs-Wilbraham phenomenon

In an ideal experiment one would measure all Fourier components $U_{\bar{g}}$ up to infinite \bar{g} . In reality, of course, only a finite number of g-vectors can be sampled. A discontinuity in a Fourier series (such as abrupt truncation) produces oscillatory artefacts in the real space reconstruction of the crystal potential. Unfortunately, these artefacts are quite stubborn and a large number of Fourier components is required to reduce their amplitude, a problem known as the Gibbs-Wilbraham phenomenon [214]. This problem can restrict X-ray diffraction [249], but is even more serious for electron diffraction, which generally is very limited in the number of g-vectors that can be accurately sampled [126]. All refinements based on electron diffraction to date have used a small number of CBED patterns in comparison with X-ray data. The number of g-vectors accessible in a single CBED pattern is very limited indeed – only one or two diffracted beams can be set in the Bragg condition and dark field pattern centres [94] (see Chapter 3) are rarely accessed. Despite this limitation, precise measurement of individual structure factors was demonstrated by Zuo and Spence in the 1980s [36, 37, 119, 245, 251–253], and it has become common practice to measure a handful of the lowest-order structure factors – which are most sensitive to bonding effects – and use X-ray or neutron diffraction to supply hundreds or thousands of higher order structure factors to complete the picture [37, 126, 246, 254]. However, the combination of disparate data – i.e. from different samples and techniques – may introduce discontinuities in the Fourier series and thus artefacts in real space, which can be of the same magnitude as the bonding effects being sought [219]. Conversely, a theoretical model of the potential is not restricted in the same way and it is straightforward to use functions that are continuous and unbounded. We avoid the Gibbs phenomenon in this chapter by fitting experimental functions to 4.2 rather than extracting individual Fourier components i.e. the diffracted intensities predicted by a model are adjusted to fit experiment through variation of structural parameters [255]. While simplistic here, it has been used with more sophistication in X-ray and neutron diffraction for many years in studies of atomic bonding, [256] by modelling electron shells with pseudopotentials [248]. Currently, there is no equivalent framework for electron diffraction.

Chapter 5

D-LACBED Sensitivity

“Hindsight is a wonderful thing, foresight is better...”

- William Blake

Unknown

The results from Chapters 3 and 4 were taken throughout the PhD. Naturally, changes were made to the technique, image processing and simulation as an ongoing process. As more was learned about the advantages and limitations of D-LACBED, additions and changes were made to the scientific procedure (the current version is given in Chapter 2).

For this Chapter we wish to examine the sensitivity of the technique using simulations to put the previous results into context and indicate possible directions for future work to D-LACBED procedure. This is to ensure the technique is reproducible. We also look at factors that could induce error such as the point spread function (PSF) and Bloch wave convergence.

5.1 Simulation Sensitivity

When conducting a structural refinement of a parameter, ideally one would limit the degrees of freedom to the parameter(s) under question. The remaining parameters are fixed at their nominal values obtained from previous research at the most agreed upon value. This method is the most practical way of obtaining quantitative results, as otherwise the problem size simply becomes so vast a result would take far too long to obtain.

Therefore, selection of the variables to refine becomes key. A researcher undertaking a structural refinement study using D-LACBED would wish to obtain an accurate and precise result without needing to vary too many indirect parameters.

To determine how to select the parameters to refine, the sensitivity of D-LACBED patterns to every parameter must be investigated. This is more difficult than it first appears. The sensitivity of a single parameter can be deduced by calculating the deltas between two D-LACBED patterns at slightly different values. However, equating this difference to other parameters (potentially on a completely different measurement scale) becomes an issue e.g. how does a small change in accelerating voltage compare to a small change in Debye-Waller factor? And what should the small change for each be?

For this study we have tried to use the accepted precisions found within the literature. For length scales 1 picometre has been used. This includes unit cell size, atomic coordinates and DWF (for the DWF we used a mean displacement of 1 picometre, and then calculated the resulting mean squared displacement). For thickness and accelerating voltage we decided to use 1 nm¹ and 1 kV respectively (see [122] as example precision). For the convergence angle we have decided to use a change of one pixel as the limiting factor.

The delta (or normalised difference) images for a select number of reflections of the corundum A221 dataset can be found in Fig. 5.1. Chosen because it gave the most precise and accurate result so far. The unit cell, atomic refinement and DWF (to a point) delta patterns should be directly comparable as each have been changed by the same length scale within the unit cell. Accelerating voltage, thickness and convergence angle cannot be directly comparable. Thickness while given in a length scale, does not equate to a unit cell measurement, and is not within the dynamical Bloch wave matrix calculation. Rather it can be more thought of as the number of unit cells in the Bloch wave calculation. The dynamic range is determined by finding the absolute maximum delta for each parameter, given in units of standard deviation.

From Fig. 5.1 we see a low-high g-vector divide from several of the parameters: thickness/accelerating voltage, occupancy, unit cell and DWF. At low index g-patterns such as the 000 and 012 reflections, most of the patterns show very similar deltas. The g-patterns display the same features (especially near the centre) with similarly scaled normalised intensities. At high g's we see this relationship break down. If we look at the $\bar{3} 3 12$ reflection we see significantly different features in each. We find that as the g-vector increases so does the qualitative difference of g-patterns thus meaning each parameter becomes ever more independent of each

¹Some of the thickness refinements in previous chapters have used 1 Å precision, in line with previous quantitative convergent beam electron diffraction (QCBED) studies (e.g. [122]), we have found that the sensitivity of the D-LACBED patterns is on the nanometer scale. An angstrom scale precision gives unviable errors (1590% increase in Table 5.2

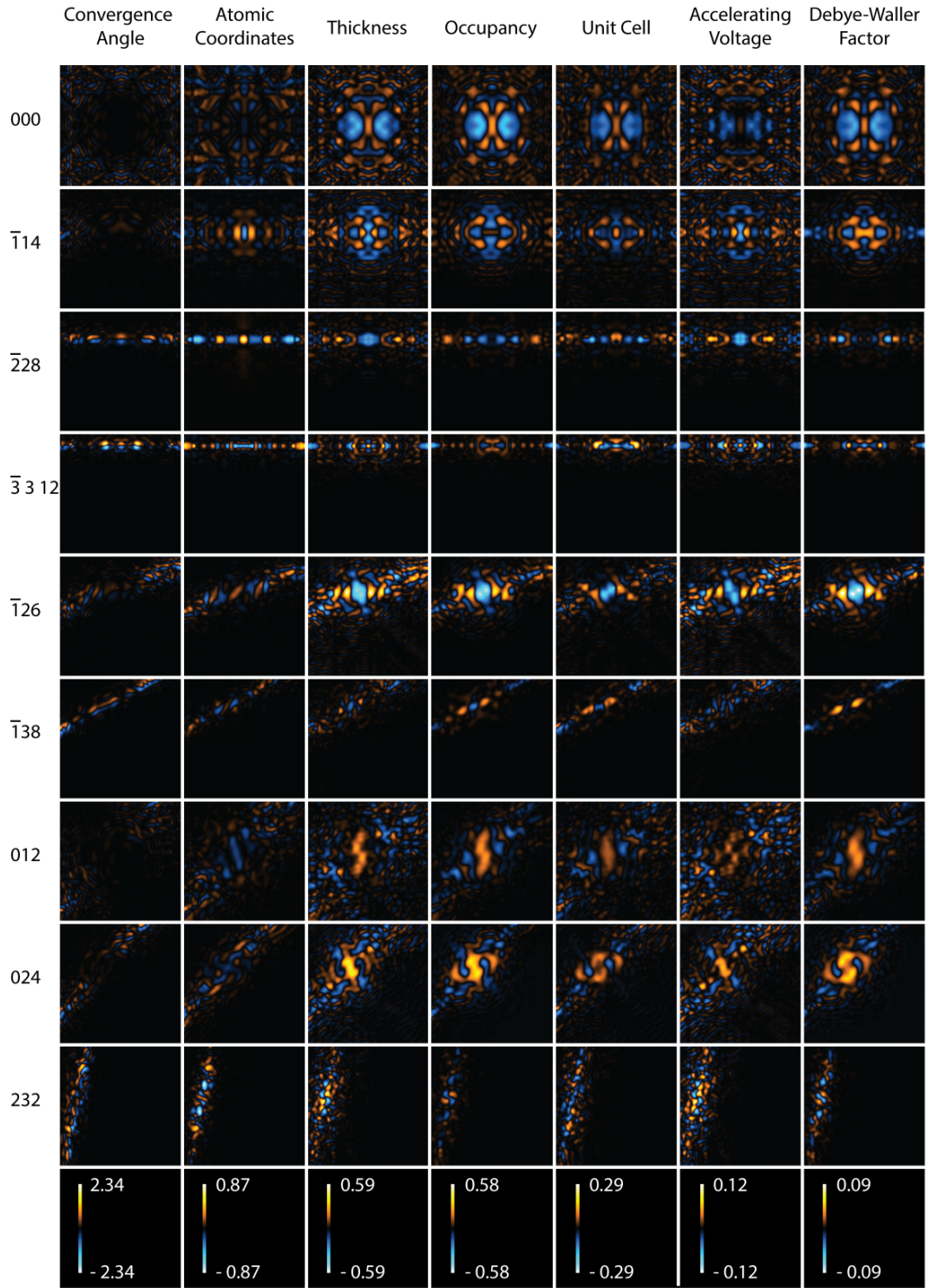


Figure 5.1: delta images for a selection of g-patterns using the alpha-corundum dataset in Chapter 3 (A221) over a range of parameters (see text for details). The scale bar units are given in standard deviation.

Index	Convergence angle	Atomic co- ordinates	Thickness	Occupancy	Unit cell	Accelerating voltage	DWF
000	0.51	0.20	0.32	0.34	0.11	0.04	0.04
$\bar{1}14$	0.55	0.46	0.33	0.24	0.11	0.07	0.04
$\bar{2}28$	0.95	0.83	0.40	0.27	0.20	0.09	0.06
$\bar{3} \ 3 \ 12$	2.34	0.63	0.41	0.36	0.29	0.10	0.08
$\bar{1}26$	1.25	0.40	0.57	0.53	0.24	0.11	0.09
$\bar{1}38$	1.13	0.26	0.24	0.24	0.13	0.05	0.05
012	1.11	0.33	0.27	0.23	0.09	0.06	0.03
024	0.99	0.26	0.47	0.38	0.15	0.08	0.05
036	0.96	0.37	0.52	0.27	0.16	0.12	0.06
232	0.96	0.87	0.55	0.21	0.16	0.12	0.05

Table 5.1: maximum delta values for a selection of g-patterns using the A221 dataset (Fig 3.2). Cyan and red highlights the $\bar{1}14$ and 012 systematic rows respectively. Units are in standard deviation

Parameter	Current Precision	Calculated precision at 1σ	Percentage increase
convergence angle	0.01	0.01	0%
atomic coordinates	1.00 pm	1.15 pm	15%
thickness	1.0 nm	1.69 nm	69%
occupancy	0.01	0.02	100%
unit cell	1.00 pm	3.39 pm	239%
accelerating voltage	1.00 kV	8.52 kV	752%
DWF	0.01\AA^2	0.09\AA^2	800%

Table 5.2: Max deltas from Fig 5.1 adjusted to 1 standard deviation for comparison of parameters

other. The sensitivity of each of these parameters seems to increase with g-vector. This is evident in Table 5.1. The maximum quantitative delta values down the $\bar{1}14$ and 012 systematic rows seem to increase with g-vector for most parameters (barring atomic coordinates).

This suggests that the dominant low order structure factors, associated with low-order reflections change similarly, and the unique characteristics of each parameter within the structure factor equation are difficult to determine. At high-order structure factors, the uniqueness of the parameters become more noticeable. This is perhaps because in low index g-patterns there are many dynamical contributions to intensity from high index structure factors. In high index g-patterns however the same may not apply, ie. there are not many dynamical contributions from low order structure factors. Comparison of future simulations excluding certain low-order or high order structure factors to Fig. 5.1 may indicate whether this is the case. Nevertheless, for future D-LACBED refinement outside of atomic coordinate, the most number of g-vectors should be found. In terms of a fitting function, a weighting term for high order indexes would likely improve the efficiency and independence of each of the parameter refinements. Perhaps leading to an increase in accuracy.

The convergence angle and atomic coordinate delta pattern features are different irrespective of index and shall be discussed later in the chapter.

To compare each parameters sensitivity to each other, a reference maximum

pixel delta value (1 standard deviation) is selected. The change in parameter needed to obtain this max delta value gives an indication of the sensitivity. For difference simulations which have not achieved a max normalised pixel difference of 1, a linear gradient is assumed to extrapolate the value. Comparison can then be made with the accepted literature precision to determine the sensitivity. This is shown in Table 5.2.

Table 5.2 also gives an indication of the precision we should expect from each parameter through a range of 1 standard deviation (σ). It is likely a benchmark smaller than 1σ is detectable in D-LACBED refinements, and therefore means smaller precisions are possible compared to the ones we present in Table 5.2. It is however a good statistical starting point to compare each of the parameters. The current precision is based upon values used in previous chapters, nominal literature values or our own estimations. The percentage increase gives an indication of how our expectation of precision is compared with the one obtained at 1σ .

It is important to notice the disparity between the DWF and atomic coordinates. Given both are on similar length scales, we see atomic coordinates are on the order of 10x more sensitive than DWFs. This is in good agreement with the original normalised difference plots seen in Chapters 3 & 4 for two different materials. The sensitivity of each parameter is discussed in Sections 5.2 through to 5.8.

5.2 Convergence angle

The convergence angle determines how much reciprocal space is probed for the selected zone axis direction. A decrease in the convergence angle will result in a decrease in the radius of the D-LACBED g-patterns. For the experimental g-patterns, we see a simple reduction of the image size. For simulated images with fixed resolution and image size however, a change in convergence angle produces a change in the magnification centred on the central pixel of the 000 g-pattern as well as each individual g-pattern image. This phenomenon is seen clearly using the three central Bragg peaks in the $\bar{3} \ 3 \ 12$ reflection of Fig 5.1. The three central blue peaks (representing the loss of intensity from the initial convergence angle) shift towards the 000 pattern as well as outwards to their new position seen as the central orange peaks. High index g-patterns show the highest degree of difference as the intensity along the Bragg condition is closer to being a single line. This means these g-patterns are extremely sensitive to the ZNCC. A change in convergence angle will shift the intensity immediately on to the background resulting in a large ZNCC value. As seen in table 5.2, the ZNCC more than suffices to determine a precise

value. In addition, one may only need to refine against a small portion of a high index g-pattern to determine an accurate convergence angle. If initially estimated from the average CBED stack (Chapter 2), the convergence angle could then be refined by cropping a high-index g-pattern to only the narrow Bragg line intensity. This would save both on computational memory and processing time, meaning it could potentially take place on a normal desktop. Future tests will show if this is possible.

The sensitivity of the convergence angle refinement is tied in to the resolution of the experimental image. An experimental image with a low resolution and high convergence angle will suffer feature loss due to pixellation. Currently with relatively low convergence angles this has not been a problem. However, a resolution/convergence angle limit exists for D-LACBED patterns which may be useful to determine if very large angle D-LACBED is ever used.

5.3 Atomic coordinates

The sensitivity of D-LACBED patterns to atomic coordinate refinement was covered in Chapter 3. Therefore, our focus shall remain on the comparison with other parameters. From Table 5.2, we see D-LACBED patterns are the most sensitive to atomic coordinate changes out of the parameters we wish to measure (the convergence angle is an experimental condition). The atomic coordinate g-patterns in Fig 5.1 show many complicated feature changes, irrespective of the index. These features tend to be narrow, with intensity variation throughout the pattern, indicating both peak shift and intensity changes. Ideal for a ZNCC. This is unlike DWFs or occupancy g-patterns which produce broad peak changes with even intensity illumination. High index g-patterns ($\bar{2}28, \bar{3} 3 12, 232$) seem to show the strongest sensitivity when considering the entire g-pattern, however certain features of low index patterns still show strong sensitivity, e.g the central part of the $\bar{1}14$ pattern. Perhaps most importantly, the features in the atomic coordinate delta g-patterns are almost completely different from any of the other parameters. This indicates that atomic coordinate refinement has a high level of independence from every other parameter refinement. This was observed during analysis of the atomic coordinate refinements of corundum in Chapter 3. Changes of the Debye-Waller factors (up to 40-50 percent) produced negligible differences to the final coordinate values. In a similar fashion we found that large changes in the applied blur radius also seemed to produce negligible differences. This suggests that atomic coordinate refinement does not require an energy filtered microscope. However, we can see that the thick-

ness, occupancy and convergence angle should be paid close attention due to their relative proximity within Table 5.2 if operating at the current chosen precisions.

We noticed that changes in the sub-pixel alignment or stretch distortion correction in the atomic coordinate refinements in Chapter 3 produced a significant change in the final coordinate values. It was hoped simulations of both distortions could be carried out to show the effect, but time unfortunately ran out. This is something in the future to explore.

5.4 Thickness Determination

CBED has been frequently used to determine the crystal thickness in a variety of electron diffraction studies [119, 120, 202, 257–263]. This is because of the highly sensitive thickness fringes [264] (see Fig 5.6b inset for example [130]) found in the patterns. While there are several methods to use these fringes to determine the thickness [257, 258, 265–269] dating back to the 1940’s [270, 271], the common method in QCBED is to use thickness as a refineable parameter using either rocking curves or the whole pattern [37, 130, 207, 252]. However, as highlighted in Koch [6] and seen in Muller [130] (Fig 5.6b) QCBED thickness refinement is often restricted to thick, small lattice parameter crystals (if the thickness is not the primary scientific aim of the study, due to the overlap problem - See Chapter 1). Otherwise position averaged convergent beam electron diffraction (PACBED) has been shown to be a useful technique for thickness determination. The overlap problem may cause the thickness fit space to have too many local minima, perhaps even to the point where thickness determination is unviable. For the purposes of this study thickness is an experimental condition and so, like the convergence angle, we would ideally like the thickness to be easily determined and highly reliable. We therefore chose to conduct a separate study using InP, to highlight the effectiveness and ease of D-LACBED at determining the thickness.

5.4.1 Indium phosphide

The dataset of InP (Fig 5.3) was taken to test the capabilities of the newly acquired Gatan OneView camera. The camera has a larger field of view in comparison to the previous Gatan Orius allowing wide beam half-convergence angles (26.10 mrad for InP) without loss of resolution. This dataset will therefore best highlight the benefits of the D-LACBED technique for thickness refinement in comparison to CBED. We show that D-LACBED gives a well-defined global minimum, and reduces the number of local minima in comparison to CBED.

The zinc-blende structure of InP is the same as GaAs, indium replaces the gallium positions and phosphorus replaces arsenic. Because all the atoms remain fixed due to symmetry, the only quantitative parameter to refine is the DWF, which was done in the previous chapter.

Thickness refinement is not an independent mode in *felix* as it is a relatively inexpensive calculation. Instead it is consistently refined during other refinement modes. This ensures the thickness is decoupled as much as possible from other parameters. Only one Bloch wave calculation is needed to obtain the thickness fit parameter space shown in Fig 5.4a.

Fortunately, the thickness of GaAs from the CBED Muller data ([130] & Fig 5.3 right) were very similar to the InP D-LACBED data so was used for comparison². We see from Fig 5.4 the significant effect of the convergence angle. While both fit spaces contain global minima, the D-LACBED minimum is much broader, meaning it would be easier to find. We only see one local minimum at 275 nm, easily characterisable due to the very shallow gradient compared to the global minimum. The gradient of the fit space at thinner sections is greater than at thicker ones. This is likely due to the broad thickness fringes seen at thinner simulations rather than the narrow fringes seen at thicker simulations. If the amplitude of the fringes are assumed to be the same we can see that a fringe wavelength much larger than the correct wavelength will produce a worse fit than a much smaller fringe wavelength. For an efficient algorithm, thickness refinements should begin at low thicknesses. The high fit values and initial gradient may be able predict where the global minimum will be found, as a relationship may exist between the initial fit gradient at low thicknesses and the correct fringe frequency. The thickness refinement can then be constrained quickly due to the easily identifiable, broad global minimum.

The CBED global minimum on the other hand has many local minima, importantly with similar gradients, which makes characterisation difficult. In a simulation method like multislice where thickness is more computationally expensive, this would make characterisation of the global minima difficult. Even with Bloch wave simulations we can see that materials with larger lattice parameters may indeed produce many deep local minima, where a unique solution is unclear.

To determine the reasons for the smoother parameter space, a DI/DT plot similar to Fig 5.1 was made for InP. We see from both figures 5.5 and 5.1 that thickness has a profound effect on the diffraction pattern intensities. The peaks within

²Ideally, we would crop down the InP dataset for the best comparison. However due to issues with the supercomputer this could not be achieved in time. Since GaAs has the same structure as InP we believe there should not be too much difference between the fit spaces, even with a different fitting function.

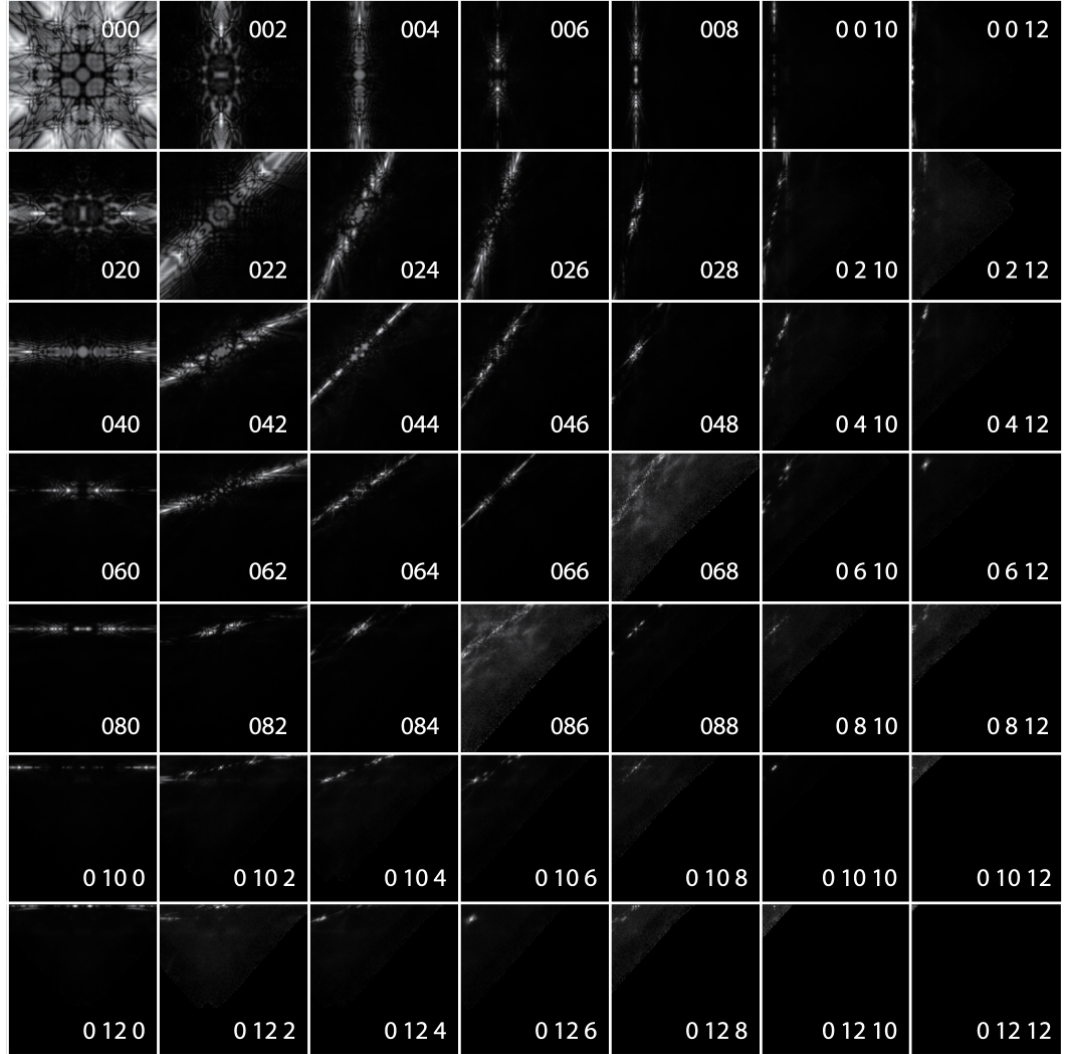


Figure 5.2: D-LACBED experimental dataset of InP from the $[001]$ zone axis. The figure contains only the positive index g-patterns of the dataset.

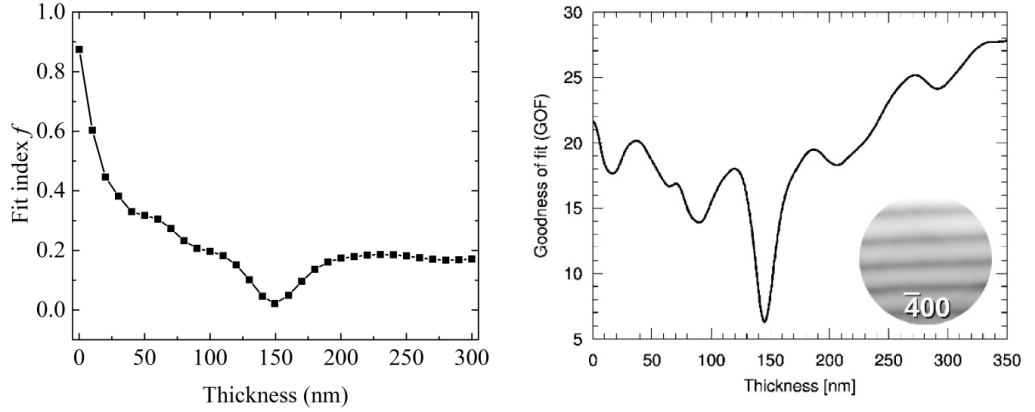


Figure 5.3: left) Simulations of InP D-LACBED patterns over a range of thicknesses and their corresponding fit values to the experimental image (See Fig. 5.2) right) Figure 10 taken from Muller et al. [130] of a CBED thickness scan of GaAs data. The inset shows the $\bar{4}00$ disc.

the normalised difference image in Fig. 5.5 are mostly narrow in nature indicating a shift of peaks rather than a change in amplitude. The peaks are broader near the centre of the g-patterns and along the Bragg condition where thickness fringes are not found. It is upon close inspection of the 022 reflection in Fig. 5.5 that we clearly see the advantage of the wide convergence angles D-LACBED provides. We see clear but faint delta thickness fringes outside of the intense Bragg condition, extending throughout the entirety of the g-pattern. As the thickness changed, the narrow faint fringes wavelength and subsequent position changed significantly, indicating high sensitivity. In an experimental image a large proportion of these fringes are hidden by the diffuse background. As shown in Table 5.2 the thickness fringes near the Bragg condition suffice for nanometre precision. However, it is encouraging to see that a higher precision is possible if energy filtering and inelastic modelling of the diffuse background is implemented.

Since the thickness is known to be more sensitive to peak position than amplitude, a Sobel, or similar edge finding filter could be applied in order to make the measurement more independent from other parameters, especially if there is plenty of diffuse scattering. From Fig. 5.1 we see the centre of the thickness delta images look similar to other parameters.

We note that the voltage delta g-patterns in Figure 5.1 are almost identical to the thickness delta g-patterns apart from a scaling factor. A change in the electron wavelength produces the same result as a change in the thickness using ZOLZ structure factors. The inclusion of HOLZ reflections will perhaps show the difference

between the two parameters. This means that in order to determine the thickness without a statistical error the voltage of the microscope must be determined accurately.

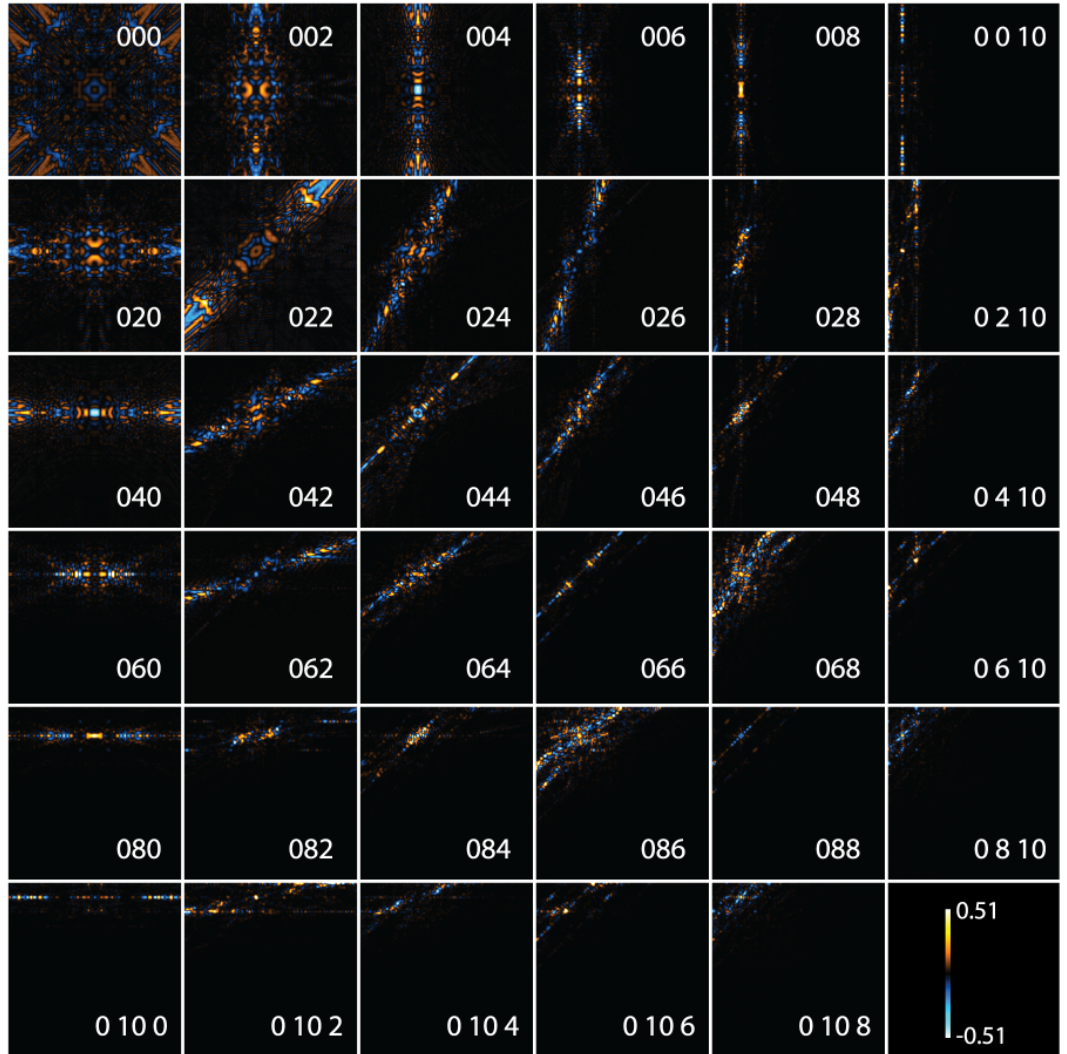


Figure 5.4: DI/DT plot of InP. Units of scale bar are in standard deviation

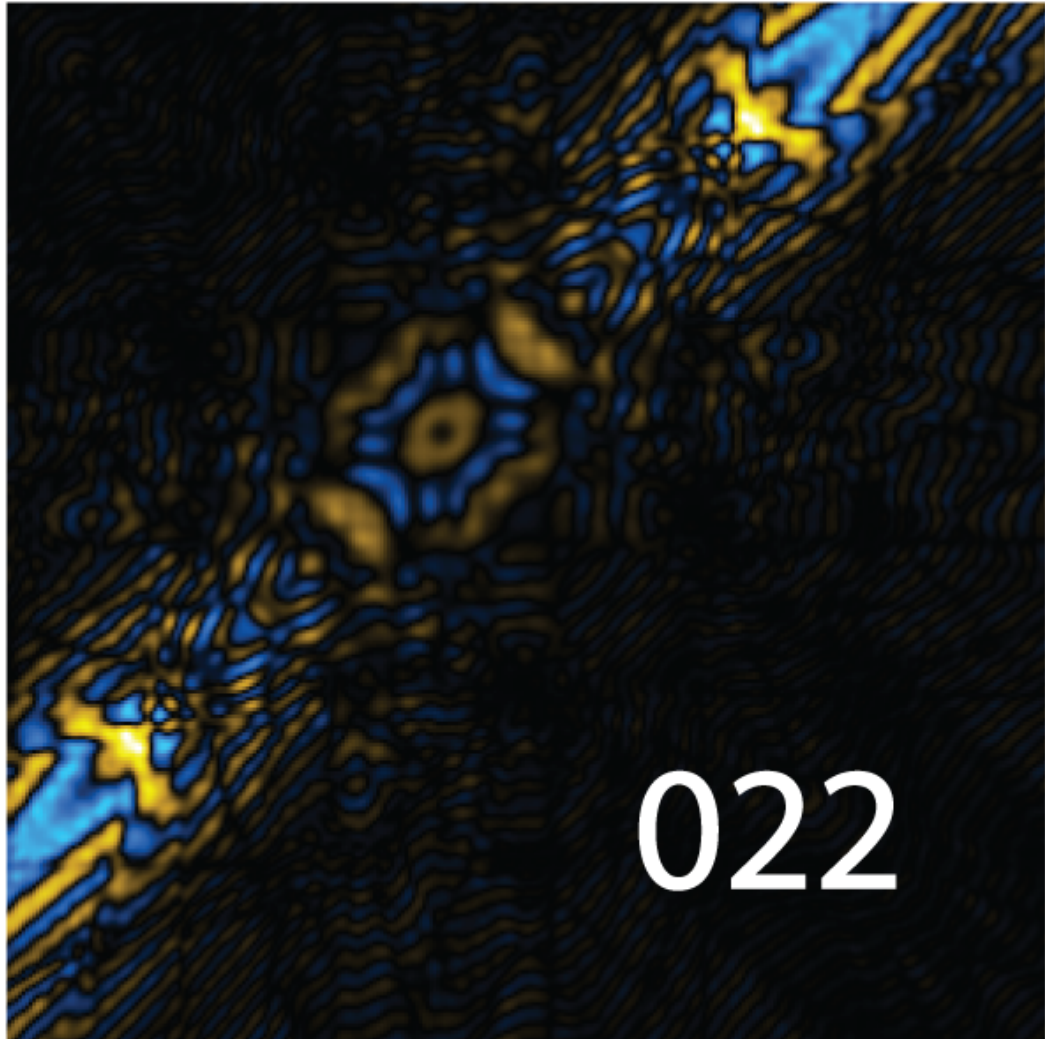


Figure 5.5: InP DI/DT 022 g-pattern from Fig 5.4. Clear and faint delta thickness fringes are prevalent throughout the background of the image (e.g. the darkened area around the "022" label)

5.5 Accelerating voltage determination

The operating voltage for the entirety of the D-LACBED studies has remained fixed at 200kV, the standard for inorganic single crystal electron diffraction studies. However, it is known the operating voltage set by the microscope software may not correspond to the applied voltage [272]. Because of this, past studies have used the ratios of voltage sensitive HOLZ line crossing points in experimental CBED patterns, to determine the actual voltage [122, 130, 273].

As shown in Table 5.2, a small change in voltage influences D-LACBED patterns to the approximate level of a small DWF change (using only ZOLZ structure factors). In the previous section, we saw from Fig. 5.1 the variations in voltage and thickness influence the simulated patterns almost identically. Therefore, precise and accurate measurement of the voltage is important for accurate thickness measurement.

Rao [272] demonstrated from the $[37\ 37\ 4]$ zone axis of silicon, that the crossing points of the $(\bar{1}\ 3\ \bar{1}3)$ $(3\ \bar{1}\ \bar{1}3)$ and $(10\ \bar{1}0\ 8)$ $(\bar{1}0\ 10\ 8)$ HOLZ lines in the central disk could be used to precisely and accurately determine the microscope voltage. Using single crystal silicon prepared through standard sample preparation methods (Chapter 2) we imaged the same zone axis (Fig. 5.7 left), albeit at room temperature. Rao noted room temperature measurements would result in blurred HOLZ lines due to too much TDS (even with an energy filter) for accurate measurement of the voltage. However, with the aid of background subtraction and line guides, we found the lines were clear enough to estimate the voltage (Fig. 5.6 right).

The HOLZ line pattern was simulated using the software TEM strain [274]. The lines were produced using the kinematic approximation, validated by a subsequent dynamical simulation. Simulations at four different voltages were carried out, their A,B ratios measured (Fig 5.6) and plotted, as shown in Fig 5.7. We assumed linearity over a short voltage range and used the plotting software Origin to fit the data with a linear least squares, Levenburg-Marquart algorithm. Using the A,B ratio from the experimental CBED disk we then determined the voltage of the 2100 microscope to be 200.7 (± 0.5) kV. While some human error has to be taken into account due to still present blurring of the HOLZ lines, the minimum ratio (i.e. measured from the edge of the seen lines) is still over 200 kV. Therefore, we can say with confidence the microscope is slightly over 200kV.

However, the accelerating voltage is known to fluctuate per session. Hopefully once HOLZ structure factors have been implemented into *felix*, the voltage refinement can be taken straight from the D-LACBED g-patterns (for patterns where HOLZ lines are visible).

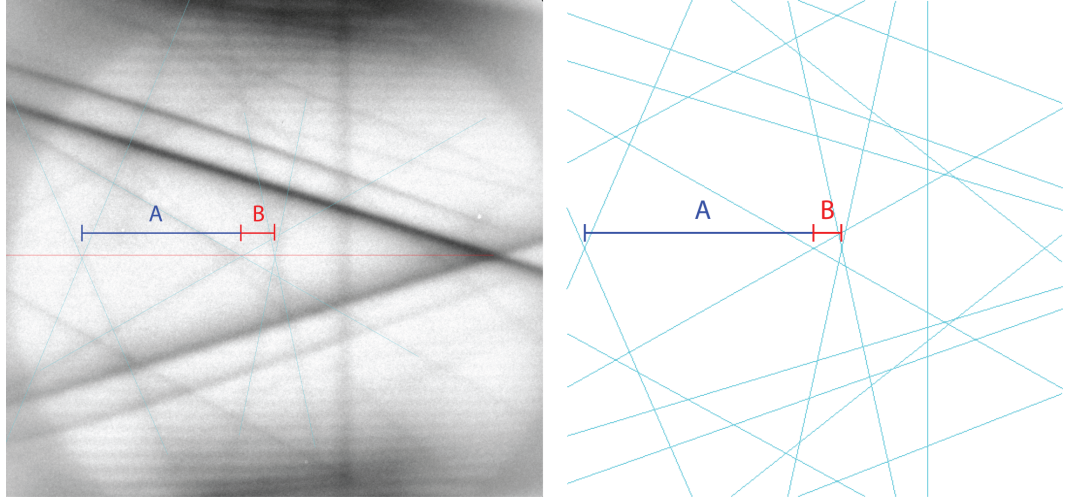


Figure 5.6: left) Experimental central CBED disk of the $[37\ 37\ 4]$ zone axis of silicon. The thin cyan lines highlight the location of the HOLZ lines. The thin red line is a crossing point guide. right) Simulated Kikuchi and HOLZ line pattern at 200kV under the same conditions. In both figures lengths A & B correspond to the ratios of HOLZ line crossing points $(7\ 9\ 11)$ & $(9\ 7\ 11)$, $(10\ \bar{10}\ 8)$ & $(\bar{10}\ 10\ 8)$, $(\bar{1}\ 3\ \bar{13})$ & $(3\ \bar{1}\ \bar{13})$ (from left to right)

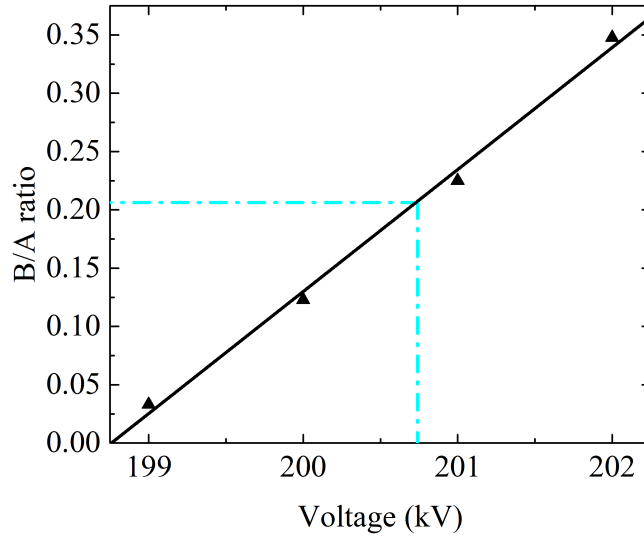


Figure 5.7: A and B simulated ratios (see Fig 5.6) at set voltages using the central disk of the silicon $[37\ 37\ 4]$ zone axis. Black line is a least squares fit. Blue dashed lines show measured experimental ratio and associated voltage (200.74 ± 0.53 kV).

5.6 Occupancy

This thesis does not cover the use of D-LACBED for compositional analysis, so therefore this section will be short. The low index delta g-patterns for the DWF and occupancy are very similar. Both the occupancy and DWF act as a multiplicative factor on each of the structure factors. At low index g-patterns and therefore structure factors it seems both have a very similar effect on the intensity. As discussed in Chapter 4, the Debye-Waller factor scales with g-vector. It is at these high index g-vectors where we see the difference between the two patterns.

From Table 5.2 we see that the occupancy delta images are on a similar level of sensitivity as thickness, which as shown in Section 5.4 can be resolved accurately, therefore it is well within possibility that D-LACBED could be used for compositional analysis.

Our simulations increased the occupancy to beyond 1.0 the normal physical limit. A greater than one occupancy was chosen here in order to more easily compare with the other parameters. An occupancy decrease would reverse the parity of the intensities we see in Fig. 5.1. While unphysical we have observed this phenomenon frequently for refinements of occupancy in the ACR and DWF studies. It is possible the occupancy value may be compensating for the lack of bonding and/or insufficient electron density in the IAM. Although this has very little grounding and requires much more investigation.

5.7 Unit Cell

Most of the past measurements of the unit cell using electron diffraction has been achieved using HOLZ reflection/line refinement [123]. Since we do not currently have the capability to simulate HOLZ reflections we shall focus on ZOLZ changes from unit cell difference images.

Each of the unit cell parameters were increased by one picometre. In parallel illumination, the diffraction spot spacing is related to the lattice parameter lengths dependent on zone axis. In a CBED disk this is also true, but since it is an image rather than a spot pattern, the relative spacings of the point intensities in each of the discs will decrease or increase dependent on the lattice spacings. This has a similar effect to the convergence angle as we can see in Fig 5.1 hence why both sets of delta patterns look similar, are asymmetric and entire Bragg condition shifts are observable at high indexes. However, we also see that there are other intensity changes. At low g's, the entire pattern sees an even spread of intensity, unlike in the conver-

gence angle patterns where only the edges see significant change. Feature changes are also apparent indicating a unit cell refinement will display some independence from a convergence angle refinement and could therefore be used. However, it seems unlikely the precision or accuracy equal that of HOLZ pattern refinement.

5.8 Debye-Waller factor

The DWF was covered in Chapter 4. We learn from Table 5.2 that it is possibly the least sensitive of all the parameters in D-LACBED patterns. This is given some context by the wide range of measurements from other techniques seen in the literature eg. Figure 4.6c. The delta peaks in the g-patterns are broad in nature meaning little or no peak shifts. This is expected as the DWF only affects the amplitude of peaks within diffraction patterns. It is again at high g-patterns where we see the most amount of difference between DWF and other parameters indicating that a high-g weighted ZNCC might be the best form of fitting function.

During the DWF refinements of copper in Chapter 4 we frequently encountered large errors related to the amount of gaussian blur and distortion (both sub-pixel shifts of the entire g-patterns and linear stretches within the g-patterns) applied. We found the blur (giving a rough approximation of the diffuse background) and distortion correction could change the refined DWF by as much as 10-15 percent. It was hopeful that these effects could be simulated and shown, however due to lack of time this was not possible. It is recommended the background and distortion should be looked at closely if accurate and precise DWF's are desired using D-LACBED.

The whole g-pattern sub-pixel shift distortion can be easily corrected using the convolution form of the ZNCC rather than the point form currently used in *felix*. For the point form it is assumed the images are already aligned perfectly, therefore only one correlation value is output. Since the ZNCC was originally purposed for template matching [275], the correlation calculation can be performed over multiple sub-pixel (using bicubic interpolation) shifts. This effectively removes the sub-pixel shift processing step. While the ZNCC is an inexpensive computational calculation, care should be taken to avoid unreasonable numbers of sub-pixel instances. i.e. a constant number of ZNCC calculations over multiple refinement steps may contribute to significant refinement times.

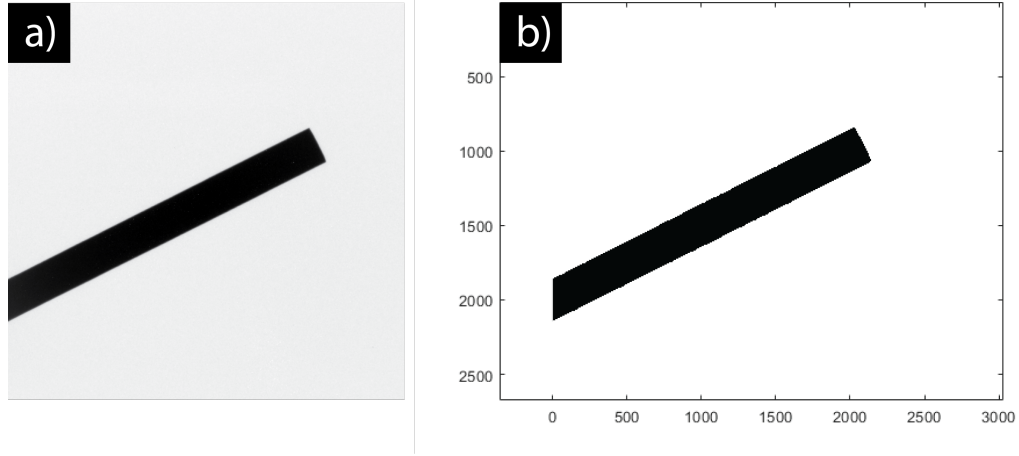


Figure 5.8: a) Bright field image at 6000x magnification of the JEOL™ 2100+ beam stopper. b) it's associated binary image after thresholding (upscaled)

5.9 Point spread function

The introduction of CCD cameras to the TEM was made much later than the inclusion of the CCD detector in other arenas [276]. A scintillator layer is required directly in front of the camera, to turn each incident electron into several photons for CCD detection [277]. This process however, results in a spread of photons and thus causes a blurring of the image: the camera PSF. It is however measurable. The final blurred image can be described as a convolution of the actual image with a rotationally symmetric PSF.

There are two main methods to determine the PSF or its Fourier equivalent, the modulation transfer function (MTF), of the scintillator crystal: the noise method [278, 279] and the edge method [279, 280]. There are many studies exploring the different ways of implementing each and their advantages and disadvantages [281–287].

We have decided to use the method first presented by Thust [288] and then automated by Van den Broek et al. [289]. This is because of its ease of use without loss of accuracy. The method involves taking an image of the beam stopper over parallel illumination. Through upsampling and thresholding, a binary image of the beamstopper is produced to represent the non-blurred ‘ideal’ image (Fig. 5.8b). The MTF of the scintillator crystal can be found by minimising the MTF function applied to the Fourier transformed ideal binary image with the Fourier transform of the recorded ‘actual’ blurred image (Fig. 5.8a).

Using the program MTFEstimate [289] the MTF was measured and shown in

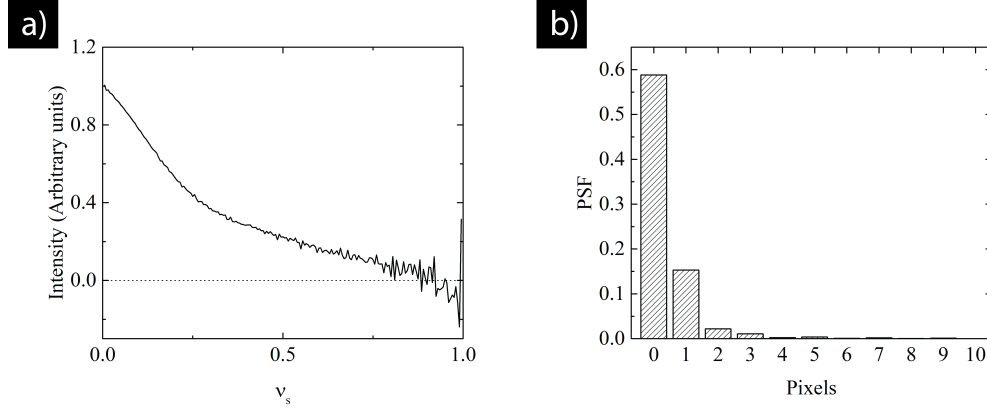


Figure 5.9: a) Average of 1D MTF measurements from beam stopper images (Fig. 5.10a) up to the maximum spatial frequency (ν_s) of 1.0 and b) it's associated Fourier transformed 1D PSF function (both the MTF and PSF are rotationally symmetric)

Fig. 5.9a. This is an average of six separate MTF measurements from 5 or 6 second exposure beam stopper images to average out noise contributions. Once Fourier transformed we obtain the PSF shown in Fig. 5.9b. Only the spatial frequencies up to $0.75 \nu_s$ were selected due to excessive noise above this value. It is however above the Nyquist frequency ($0.5 \nu_s$). While the bars are too small to see in the figure at higher pixel values, the tails are quite long and Lorentzian as mentioned by Thust and Van den Broek et al. [288, 289]. However, most of the blur is characteristic over only two or three pixels. The amplitude of the PSF is higher than the plot shown in Thust, this is probably due to the noise found in the MTF. We found as we increased number of spacial frequencies used from the MTF the PSF amplitude decreased. Therefore, a non-linear least squares fit and/or further averaging of beam stopper images will reduce the noise level and ensure higher frequencies can be measured.

A non-linear least squares fit was planned for the MTF scintillator measurement, however there was difficulty in fitting to the proposed function identified by Van den Broek et al. (the sum of an exponential and gaussian). Since this is required to implement the MTF to the D-LACBED images, the study was stopped here. Once a fit is obtained, the rotationally symmetric MTF of the scintillator can be multiplied to the Fourier transform of a simulated D-LACBED image (The MTF by pixellation does not apply as long as both simulated and experimental images are the same size) to show whether the extent of MTF blurring when compared to the experimental image. Several important questions on the effectiveness of background subtraction, symmetrising data, whether energy-filters and diffuse scattering models are needed, can then be examined.

5.10 Simulation Bloch wave convergence

The poor scaling of simulation time t the number of Bloch waves N (t proportional to N^3) mandates some effort to use the minimum possible number of Bloch waves while maintaining the necessary accuracy in calculated diffracted intensities [190,191]. We find that improvements in accuracy accrue logarithmically with N (Fig. 5.10). We use the difference between a reference pattern with $N = 250$ Bloch waves ($N_{weak} = N_{strong}$) and simulations with different N to determine the behaviour of different zone axes. Fig. 5.10 shows the largest difference found in any pixel of the complete simulation, as a percentage of the maximum intensity of each D-LACBED pattern. For Cu [001] in Chapter 4, 25 D-LACBED patterns with dimensions 400x400 were simulated, giving 4.0×10^6 pixels, while for Cu [114] 13 D-LACBED patterns were simulated giving 2.1×10^6 pixels. Least-squares fits to the measurements can then be extrapolated to find the minimum value of N that gives the required accuracy (here set to be one grey level in an 8-bit image, or $1/256$). The difference in behaviour of the [001] and [114] data is due to the different densities of points in the ZOLZ section through the reciprocal lattice; g-vectors in the [114] pattern are larger, meaning that fewer Bloch waves are necessary to capture the behaviour up to a given scattering angle.

Ideally the ratio of weak beams to strong beams and their subsequent selection criteria would be studied (as achieved in Birkeland et al. [191]). This has been planned for the future.

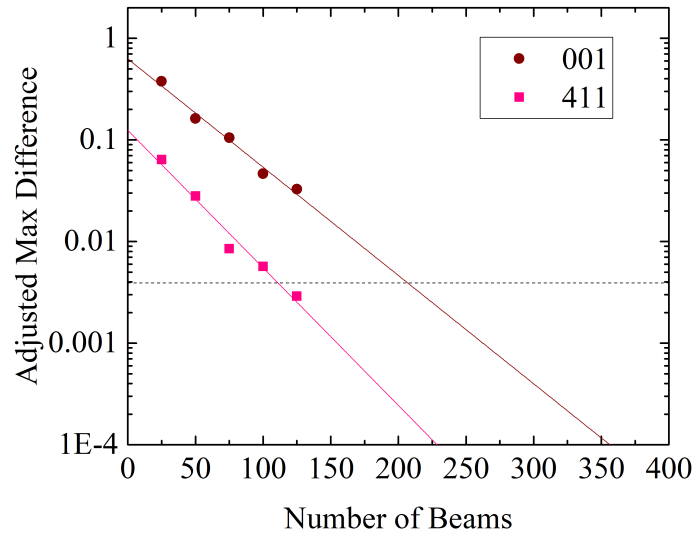


Figure 5.10: Simulation tests for the [001] and [114] Cu D-LACBED data of Fig. 4.1, showing the maximum difference (arbitrary units using 16-bit images) in intensity of any pixel with respect to a reference simulation using 250 Bloch waves. Each line is a least squares fit which is extrapolated to meet the desired criterion for accuracy (horizontal dotted line).

5.11 Summary

We find that D-LACBED patterns are up to 10 times more sensitive to atomic coordinate changes than DWF changes, bringing context to the previous two Chapters. From Table 5.2 we see that atomic coordinate refinements should be able to reach sub-pm precision. We see from Fig. 5.1 that atomic coordinate refinements are almost completely independent from all the other parameters. We hypothesise that atomic coordinate refinement will most likely produce accurate and precise results without the need for energy filters but will perhaps need distortion correction for highly accurate results.

It is possible to refine other parameters including the DWF, but for highly accurate and precise results a large dataset is needed perhaps utilising a fitting function weighted to high index g-patterns. The camera PSF is known to have an influence on the CCD in electron diffraction studies and was found to be similar to other camera PSFs. The analysis of the PSF on D-LACBED images will open further investigation on the effectiveness of the background subtraction and the effect of diffuse scattering on the data. A Bloch wave convergence condition was deduced and used for an initial investigation for the number of beams required to reach an acceptable simulated D-LACBED pattern.

Chapter 6

Conclusions and future work

"As a coda should be, this part of our essay is short"

- **Edwin Hewitt & Robert E. Hewitt**

The Gibbs-Wilbraham phenomenon: an episode in Fourier analysis [214]

D-LACBED provides a solution to the ‘overlap problem’ in CBED, increasing the amount of information possible for zone axis patterns. We find that D-LACBED is indeed more accurate and precise than conventional CBED for structural refinement, in line with the history of electron diffraction.

This is especially the case for atomic positional refinement. In Chapter 3 we showed that the dynamical information of a D-LACBED zone axis pattern contained enough 3D information to obtain atomic coordinates with excellent agreement (sub-picometre range) to X-ray measurements. We explored potential future methods to achieve the accuracies of X-ray diffraction, once thought improbable due to the dynamical nature of electrons [290].

Exploration of well-studied materials in the future should add a greater weight of argument for the use of D-LACBED in ACR studies. The process may start with studies over several zone-axes of a single material (e.g. corundum) to examine how orientation affects the refined atomic positions. It could then be extended to similar structures with different atoms of slightly different positions to explore the sensitivity of the technique (e.g. chromium oxide). After such studies, the consistency of the technique would be clear. This could then be extended to well-studied complex materials, known to have several minor atomic displacements leading to property changes (e.g. perovskites). This will determine if the technique is capable at handling the many displacements required for most modern atomic position refinement studies.

In Chapter 4 we explored the refinement of the DWF. We found that while the IAM model used to refine DWFs from D-LACBED patterns achieved accurate results for simple metals like copper, materials like GaAs with a greater amount of interaction between atoms (through anharmonicity and bonding) required more accurate models. If the Gibbs-Wilbraham phenomenon is explored in more depth, D-LACBED could be used to refine structure factors as a whole. Which would be a very useful experimental counterpart to density functional theory studies of bonding in single crystal materials. There is also future scope using GaAs to explore the model of structure factors. For instance, using pseudopotentials [248] to model the charge transfer between atoms. If a value for the DWF can be obtained for GaAs which agrees with the literature, then exploration of materials with more complex bonding between atoms (e.g. strontium titanate) could be possible.

In Chapter 5 we looked at the sensitivity of D-LACBED Bloch wave simulations within the context of structural parameters. We found that in general, g-patterns at higher indices seemed to be more sensitive to changes of structure and could possibly be used for weighting of future refinements.

It has been shown in Chapter 2 that a significant amount of image processing is implemented to correct the distortions and blur seen in each of the D-LACBED patterns. It has been mentioned but not fully explored in Chapter 5 that uncorrected distortions or blur can significantly contribute to errors in structural refinements. While there are plenty of other contributions to error (i.e. simulation model, dislocations etc.) the distortion and blur are easier to identify, correct and perhaps most importantly made reliable. If structural refinement measurements are to approach the precision of X-ray and neutron ones, distortion and blur correction are essential areas to explore.

In this thesis we only explore singular zone axes chosen because of their ease of access. We see from the discussion in Chapter 3 that certain g-patterns may be more sensitive to atomic position, and postulate searching for the least sensitive zone axes may give an indication of the extent of dynamical scattering in D-LACBED patterns. Exploration of different zone axes allows better understanding of dynamical effects to structural refinements. For instance in Chapter 4 the [110] direction of gallium arsenide is known to have strong dynamical scattering in the $n111$ systematic row g-patterns. A structural refinement of the DWF on a zone axis without strong dynamical g-patterns may help to determine whether these strong dynamical couplings at low-order g-patterns help or hinder the result. Perhaps the conclusion drawn in Chapter 5 that more sensitive g-patterns are at higher indices may mean mid-range zone axes are best for DWF refinements where perhaps more

reflections display 'uniqueness'.

Comparison of D-LACBED structural refinements have only been to ex-situ data so far. It may be useful to conduct an experiment using several techniques similar to D-LACBED (e.g. SAED, PED, CBED, LACBED etc.) on the same area of sample (i.e. in situ) and comparing structural refinement results over many variables. This will give a more reliable indication of the strengths and weaknesses of D-LACBED.

From this thesis we see that D-LACBED could potentially be used for a wide range of structural refinement studies at the nano-level, similarly to parallel beam methods in structural solution studies. Experimentally, there is plenty of future scope to improve the efficiency, ease of use and post-processing of the technique, potentially leading to acquisition of high-quality datasets in seconds.

It is perhaps the simulation where the major limitation of the technique lies. As the size of data sets and camera resolutions increase, the required number of pixels to simulate does also. The structural refinements carried out in this thesis use computationally heavy dynamical theory, which require supercomputers and can take hours to complete. There is only minor scope to improve the efficiency of these calculations through traditional methods [291]. This is compared to X-ray kinematical refinements where a full structure can be output in minutes.

A solution may lie in machine learning. Already used within the electron microscopy community [292–294], machine learning has the capability to reduce the time of complex calculations from hours to minutes or seconds, without loss of accuracy (it may even improve accuracy [295]). If a general solution is found for direct inversion in dynamical diffraction and the application of computer control to transmission electron microscopes continues to improve, the tantalising prospect of a full structure calculation (solution and refinement) may be possible from a single zone axis pattern and achievable in minutes.

At the moment it is unclear whether D-LACBED will be a part of the future of electron microscopy, however, it is clear big data in electron microscopy is here to stay.

Appendix A

Corundum bond lengths and angles

Bond	Bond Angle (This thesis)	Bond Angle (Kondo)
O1 - Al - O2	79.587°	79.642°
O1 - Al - O6	90.675°	90.779°
O2 - Al - O6	86.372°	85.378°
O4 - Al - O5	101.284°	101.161°
O4 - Al - O1	164.119°	164.239°

Table A.1: Bond angle comparison between this thesis and reference values (Kondo et al. [204]) for the AlO_6 octahedra seen in fig A.1. The top triplet bond lengths are: 1.850 Å (This study) & 1.854 Å (Kondo). The bottom triplet bond lengths are: 1.976 Å (This study) & 1.971 Å (Kondo)

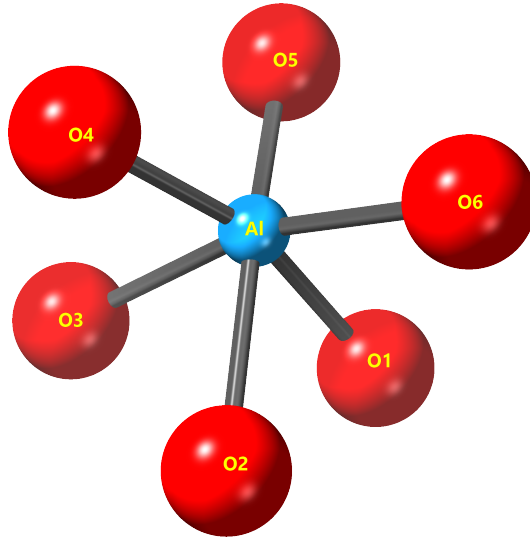


Figure A.1: AlO_6 octahedra within corundum structure - see Fig. 3.1. The bonds within each of the top and bottom triplets are equal in length

Appendix B

Guide for D-LACBED alignment

B.1 Sample preparation

Sample preparation is one of the major influencers on the quality of the final data. A well-made flat (i.e. parallel surfaced) sample, with few defects, contaminants and large average grain size will produce adequate data even with poor calibration and alignment. Further, when ion milling, if the energy is switched from 6kV to 2kV as soon as a hole appears in the sample there are more likely to be flat areas to take D-LACBED data from (a larger hole is more likely to produce a wedge shape sample), thereby reducing the time of the experiment.

B.2 Alignment

During collection of D-LACBED data the author has discovered many tips which may help future users.

- (i) A large sample area of even thickness without bending is ideal for D-LACBED (however it is possible to take data in and around 20nm regions – dependent on the type of microscope). Choose an experimental site before any alignments, otherwise you may find that you need to redo them. Currently a distinct feature (like a hole, or abnormally shaped edge) produces the best image shift correction, therefore a sample area near a distinct feature is preferable.
- (ii) During collection of the data, the incident electron beam is automatically tilted and shifted. Therefore, it is good practice to check the tilt, shift and voltage

centre alignments multiple times throughout the alignment process.

- (iii) To align for the intermediate lens astigmatism, move the beam off the sample and the spread it as widely as possible for parallel illumination. Switch to diffraction mode and adjust the diffraction focus until a caustic is seen within the diffraction disc. To correct for the astigmatism, make the caustic circular¹. Do this alignment before the diffraction focus alignment so it does not have to be repeated.
- (iv) Determination of the correct diffraction focus using a CBED pattern was initially found to be quite difficult. Eventually it was discovered that Kikuchi lines are good indicators of focus. If acquiring a zone axis pattern, tilt to just off zone axis for the sharpest Kikuchi lines. Adjust the focus until the Kikuchi lines are at their sharpest.
- (v) With small CBED discs, sometimes it is difficult to find the centre of the zone axis pattern. Using dark field tilt mode (normally used for dark field imaging) the area around the zone axis can be explored without consequence to previous alignments (as long as the user switches back to bright field tilt mode when tilting to the zone axis).

¹Notes for the rookie microscopist: If the transmission electron microscope has an angled phosphor screen, a perfectly round circle on the angled screen is not a round circle on the flat screen/camera. The author forgot about this many times!

Appendix C

Derivation of periodic Schrödinger equation

In Chapter 2, we use the Schrödinger equation (Eq. 2.2)

$$\left(-\frac{\hbar^2}{2m}\nabla^2 - |e|V(\vec{r})\right)\Psi_B(\vec{r}) = E\Psi_B(\vec{r}) \quad (\text{C.1})$$

with Bloch wave solutions, to describe a wavefunction of fast electrons through a periodic potential (Eq. 2.7). Here we show how this is derived.

The kinetic energy $E = \hbar^2 k_0^2 / 2m$, is inserted into Eq. C.1

$$\left(-\frac{\hbar^2}{2m}\nabla^2 - |e|V(\vec{r})\right)\Psi_B(\vec{r}) = \frac{\hbar^2 k_0^2}{2m}\Psi_B(\vec{r}) \quad (\text{C.2})$$

Multiply Eq. C.2 by $2m$

$$(-\hbar^2\nabla^2 - 2m|e|V(\vec{r}))\Psi_B(\vec{r}) = \hbar^2 k_0^2 \Psi_B(\vec{r}) \quad (\text{C.3})$$

Divide Eq. C.3 by \hbar^2

$$\left(-\frac{\nabla^2}{4\pi^2} - \frac{2m|e|V(\vec{r})}{\hbar^2}\right)\Psi_B(\vec{r}) = k_0^2 \Psi_B(\vec{r}) \quad (\text{C.4})$$

Multiply Eq. C.4 by $4\pi^2$

$$\left(-\nabla^2 - 4\pi^2 \frac{2m|e|V(\vec{r})}{\hbar^2}\right)\Psi_B(\vec{r}) = 4\pi^2 k_0^2 \Psi_B(\vec{r}) \quad (\text{C.5})$$

Expand brackets

$$-\nabla^2 \Psi_B(\vec{r}) - 4\pi^2 \frac{2m|e|V(\vec{r})}{\hbar^2} \Psi_B(\vec{r}) = 4\pi^2 k_0^2 \Psi_B(\vec{r}) \quad (\text{C.6})$$

Swap signs and rearrange

$$\nabla^2 \Psi_B(\bar{r}) + 4\pi^2 k_0^2 \Psi_B(\bar{r}) = -4\pi^2 \frac{2m|e|}{h^2} V(\bar{r}) \Psi_B(\bar{r}) \quad (\text{C.7})$$

Using $(2m|e|/h^2)V(\bar{r}) = U(\bar{r})$, we arrive at Eq 2.7

$$(\nabla^2 + 4\pi^2 k_0^2) \Psi_B(\bar{r}) = -4\pi^2 U(\bar{r}) \Psi_B(\bar{r}) \quad (\text{C.8})$$

Appendix D

Listing of D-LACBED code

D.1 Calibration 1

```
// $BACKGROUND$
//**** Calibrate tilts1 ****\\
//** Richard Beanland r.beanland@warwick.ac.uk **\\
//** Coding started June 2012 **\\

// 1.0, 23 June 2012
// 1.1 - added fast data collection March 2013
// 1.2 - fast data collection as a subroutine April
2013
// 1.3 - bug fixes April/May 2013
// 1.4 - added image shift calibration Summer 2014
// 1.5 - bug fixes Oct 2016
// 2.0 - split off from shift calibration Nov 2016
// 2.1 - updated for OneView camera Dec 2018

//Global variables
number true=1, false=0;
number Camb=4096, Camr=4096//Global values, for
oneViewIS camera
number binning=8;//Could be a dialogue
number expo=0.01;//Could be a dialogue
number _t=0;
number _l=0;
number _b=Camb/binning;
number _r=Camr/binning;
number rspot=2
number sleeptime=0.0;//delay while waiting for
microscope to respond
image img1,img0,img2,imgCC;//live image, reference
image, sobel filtered and cross-correlation
respectively
object img_src

//*****//
// Subroutines.
//*****//
//UpdateCameraImage

//RemoveOutliers
//GetCoordsFromNTilts
//EMChangeMode
//EMBeamCentre
//TiltsSize
//ShiftSize
//DiscSize
//Sobel

//Function UpdateCameraImage
//Gets next available frame from the camera
void UpdateCameraImage(object img_src, image img)
{
    // Wait for next frame
    number acq_params_changed=0;
    number max_wait = 0.1;
    if
    (img_src.IMGSRV_AcquireTo(img,true,max_wait,acq_params
    changed))
    {
        while
        (img_src.IMGSRV_AcquireTo(img,false,max_wait,acq_param
        s changed))
        {
        }
    }
} //End of UpdateCameraImage

//FUNCTION remove outliers
//Remove outliers in the selection by comparing with an
identical image with median filter applied
void RemoveOutliers(image img, number thr)
{
    image medimg=img[];
    medimg=medianFilter(img,3,1);
    // replace ONLY those pixels in IMG which are >thr in
    comparison with median image
}
```

```

img=tert((abs(img-medImg)>thr),medImg,img);
//tidy up
medImg.DeleteImage();
} //end of RemoveOutliers

//Function GetCoordsFromNTilts
//Gets x-y coords from the index currentPoint
void GetCoordsFromNTilts(number nTilts, number
currentPoint, number &i, number &j)
{
    number side=2*nTilts+1;
    j=floor(currentPoint/side)-nTilts;
    i=((currentPoint%side)-nTilts)*((-1)**(j%2)); //NB %
    means modulo, flips sign every row
} //End of GetCoordsFromNTilts

//Function EMChangeMode
//Asks user to change mode - will keep doing so until
they comply
//Space bar to exit loop
void EMChangeMode(string mode want)
{
    string mode_is=EMGetImagingOpticsMode();
    number clickedOK=true;
    while (!(mode_is==mode_want)) //not in diffraction
mode
    {
        clickedOK=true;
        external sem=NewSemaphore();
        try
        {
            ModelessDialog("Please put the microscope in
"+mode_want+" mode", "OK", sem);
            GrabSemaphore(sem);
            ReleaseSemaphore(sem);
            FreeSemaphore(sem);
        }
    }

    catch
    {
        FreeSemaphore(sem);
        clickedOK=false;
        break;
    }
    mode_is=EMGetImagingOpticsMode();
}
// Give user some way out
if (spacedown()) throw("User aborted")
} //End of EMChangeMode

//Function EMBeamCentre
//Puts the beam in the centre of the image given
measured position x0,y0
void EMBeamCentre(string Tag_Path)
{
    number ShiftX0,ShiftY0,xShpX,xShpY,yShpX,yShpY;
    EMGetBeamShift(ShiftX0,ShiftY0);
    //Get beam shift calibration
    GetPersistentNumberNote(Tag_Path+"xShpX",xShpX);
    GetPersistentNumberNote(Tag_Path+"xShpY",xShpY);
    GetPersistentNumberNote(Tag_Path+"yShpX",yShpX);
    GetPersistentNumberNote(Tag_Path+"yShpY",yShpY);
    number x0,y0;//coords of untilted beam
    number maxval=img1.Max(x0,y0);
    number xl=(x0)-x0;//xl,yl are the no. of pixels to
move [x,y]
    number yl=(y0)-y0;
    number xCentre=round(ShiftX0+xl*xShpX+yl*yShpX);
    number yCentre=round(ShiftY0+x1*xShpY+yl*yShpY);
    EMSetBeamShift(xCentre,yCentre);
    sleep(sleeptime);
} //End of EMBeamCentre

//Function Tiltsize

//Changes tilt increment dTilt to be 1/4 of image
height
number TiltSize(number dTilt, number &TlX, number &TlY,
number &T2X, number &T2Y)
{
    //auto-correlation to find the coords of untilted beam
x0y0
    number tiltX0,tiltY0;
    EMGetBeamTilt(tiltX0,tiltY0);
    UpdateCameraImage(img_src,img1);
    // sleep(sleeptime);
    // UpdateCameraImage(img_src,img1);//zero tilt image
img0=img1;//put into img0
    imgCC=img1.CrossCorrelate(img0);
    number x0,y0;//coords of untilted beam
    number maxval=imgCC.Max(x0,y0);
    //tilt the input guessed dTilt along X-DAC
    EMSetBeamTilt(tiltX0+dTilt,tiltY0);
    UpdateCameraImage(img_src,img1);
    sleep(sleeptime);//give the microscope time to respond
    UpdateCameraImage(img_src,img1);//X-DAC tilted image
    imgCC=img1.CrossCorrelate(img0);
    number x,y;//coords of tilted beam
    maxval=imgCC.Max(x,y);
    number tPix=((x-x0)**2+(y-y0)**2)**0.5;//the spot
movement in pixels
    //calculate accurate dTilt
    dTilt=dTilt*0.25*b/tPix;

    //Now measure tilt(pixels) per DAC
    EMSetBeamTilt(tiltX0+dTilt,tiltY0);
    UpdateCameraImage(img_src,img1);
    sleep(sleeptime);
    UpdateCameraImage(img_src,img1);//X-DAC tilted image
    imgCC=img1.CrossCorrelate(img0);
    maxval=imgCC.Max(x,y);
    TlX=(x-x0)/dTilt;
    TlY=(y-y0)/dTilt;
    EMSetBeamTilt(tiltX0,tiltY0+dTilt);

    UpdateCameraImage(img_src,img1);
    sleep(sleeptime);
    UpdateCameraImage(img_src,img1);//Y-DAC tilted image
    imgCC=img1.CrossCorrelate(img0);
    maxval=imgCC.Max(x,y);
    T2X=(x-x0)/dTilt;
    T2Y=(y-y0)/dTilt;

    //reset tilt to zero again
    EMSetBeamTilt(tiltX0,tiltY0);
    sleep(sleeptime);

    return dTilt
} //End of TiltSize

//Function Shiftsize
//Changes shift increment dShift to be 1/4 image height
number ShiftSize(number dShift, number &Sh1X, number
&Sh1Y, number &Sh2X, number &Sh2Y)
{
    //auto-correlation to find the coords of unshifted
beam x0y0
    number shiftX0,shiftY0;
    EMGetBeamShift(shiftX0,shiftY0);
    UpdateCameraImage(img_src,img1);
    sleep(sleeptime);
    UpdateCameraImage(img_src,img1);//zero shift image
img0=img1;//put into img0
    imgCC=img0.CrossCorrelate(img0);
    number x0,y0;//coords of unshifted beam
    number maxval=imgCC.Max(x0,y0);
    //Shift the input guessed dShift along X-DAC
    EMSetBeamShift(shiftX0+dShift,shiftY0);
    sleep(sleeptime);//have to give the microscope time to
respond
    UpdateCameraImage(img_src,img1);
    sleep(sleeptime);

```



```

UpdateCameraImage(img_src,img1);//shifted image
imgCC=img1.CrossCorrelate(img0);
number x,y;//coords of shifted beam
maxval=imgCC.Max(x,y);
number tPix=((x-x0)**2+(y-y0)**2)**0.5;//the spot
movement in pixels
//calculate accurate dShift
dShift=round(dShift*0.25*b/tPix);

//now measure shift(pixels) per DAC
EMSetBeamShift(shiftX0+dShift,shiftY0);
sleep(sleeptime);
UpdateCameraImage(img_src,img1);
sleep(sleeptime);
UpdateCameraImage(img_src,img1);//X-DAC shifted image
imgCC=img1.CrossCorrelate(img0);
maxval=imgCC.Max(x,y);
ShiX=(x-x0)/dShift;
ShiY=(y-y0)/dShift;
EMSetBeamShift(shiftX0,shiftY0+dShift);
sleep(sleeptime);
UpdateCameraImage(img_src,img1);
sleep(sleeptime);
UpdateCameraImage(img_src,img1);//Y-DAC shifted image
imgCC=img1.CrossCorrelate(img0);
maxval=imgCC.Max(x,y);
Sh2X=(x-x0)/dShift;
Sh2Y=(y-y0)/dShift;

//reset tilt to zero again
EMSetBeamShift(shiftX0,shiftY0);
sleep(sleeptime);

return dShift
//End of ShiftSize

//Function DiscSize
//Gives radius of CBED disc

number DiscSize(image cbed)//, number &x0, number &y0
{
    result("Finding radius of CBED disc...\n");
    number imgX,imgY;
    cbed.GetSize(imgX,imgY);
    image disc = cbed.ImageClone()*0;
    number Rmax=round(min(imgX,imgY)/10);
    number Rr=0,Cc1=0,Cc2,dCc=1;

    while (dCc>0)
    {
        Rr+=1
        disc=ttert(iradius<Rr,1,0);//trial image
        Cc2=max(disc.CrossCorrelate(cbed));//cross correlate
        dCc=Cc2-Cc1;//if better than the last one, dCc is
        +ve, otherwise exit
        Cc1=Cc2;
    }
    return Rr-1
} //End of DiscSize

//Function Sobel
//3x3 Sobel filter (should be ^0.5 at the end, but not
done for speed)
void Sobel(image img)
{
    number imgX,imgY;
    img.GetSize(imgX,imgY);
    image diffX=img.ImageClone()*0;
    image diffY=diffX;
    //x gradient
    diffX[1,1,imgY-1,imgX-1] = 3*img[0,0,imgY-2,imgX-2];
    diffX[1,1,imgY-1,imgX-1] += 10*img[0,1,imgY-2,imgX-1];
    diffX[1,1,imgY-1,imgX-1] += 3*img[0,2,imgY-2,imgX ];
    diffX[1,1,imgY-1,imgX-1] -= 3*img[2,0,imgY ,imgX-2];
    diffX[1,1,imgY-1,imgX-1] -= 10*img[2,1,imgY ,imgX-1];
    diffX[1,1,imgY-1,imgX-1] -= 3*img[2,2,imgY ,imgX ];
    img=diffX*diffX+diffY*diffY;
} //End of Sobel

//*****//
// Main program
//*****//
number tstart = GetHighResTickCount();

//Start in diffraction mode
EMChangeMode("DIFF")

//Get Basic stuff to start
number spot=EMGetSpotSize()+1;
number mag=EMGetMagnification(); //Sometimes gives null
answer, why??
number camL=EMGetCameraLength();
//And prompt input for alpha
number alpha=3;
if (!GetNumber("Alpha?",alpha,alpha)) exit(0)
//set up tag and file paths
string
Tag_Path="DigitalDiffraction:Alpha="+alpha+":Binning="+
binning+"CamLe="+CamL+"mm";
string pathname=PathConcatenate(
GetApplicationDirectory("common app data",0),"Reference
Images\\");
string file_name=pathname+"D-
ED TiltCal A"+Alpha+" B"+binning+" C"+CamL+".dm4";
string datetime;
//number of points +/- to measure

diffX[1,1,imgY-1,imgX-1] = 3*img[0,0,imgY-2,imgX-2];
diffX[1,1,imgY-1,imgX-1] += 10*img[0,1,imgY-2,imgX-1];
diffX[1,1,imgY-1,imgX-1] += 3*img[0,2,imgY-2,imgX ];
diffX[1,1,imgY-1,imgX-1] -= 3*img[2,0,imgY ,imgX-2];
diffX[1,1,imgY-1,imgX-1] -= 10*img[2,1,imgY ,imgX-1];
diffX[1,1,imgY-1,imgX-1] -= 3*img[2,2,imgY ,imgX ];
img=diffX*diffX+diffY*diffY;
} //End of Sobel

number nTilts=8;
//make or load calibration data, image TiltCal
GetPersistentStringNote(Tag_Path+"Date",datetime);
image TiltCal;
number data_type=2;
//datetime="";
//if (datetime=="")
//{
    result("No tilt calibration: making new calibration
file");
    TiltCal := NewImage("Tilt/shift
calibration",data_type,(2*nTilts)+1,(2*nTilts)+1,4);
//}
//else
//{
    result("\nLast calibration "+datetime+"\n");
    TiltCal := NewImageFromFile(file_name);
//}

//update tags
number f;
string date;
GetDate(f,date);
string time;
GetTime(f,time);
datetime=date+"."+time;
result ("\nStarting tilt calibration "+datetime+"\n");
SetPersistentStringNote(Tag_Path+"Date",datetime);
SetPersistentNumberNote(Tag_Path+"Spot size",spot);
SetPersistentNumberNote(Tag_Path+"nCals",nTilts);

// Stop any current camera viewer
number close_view=1, stop_view=0;
try
{
    cm_stopcurrentcameraViewer(stop_view);
}
catch
{
}

```

```

        throw("Couldn't stop camera properly, try again!");
    }

    //set up images to contain data
    number nPts=((2*nTilts)+1)**2;
    //set up arrays holding shift correction
    image Xsh=RealImage("X-shift with
    tilt",4,(2*nTilts)+1,(2*nTilts)+1 );
    //Xsh.DisplayAt(655,30);
    //Xsh.SetWindowSize(200,200);
    image Ysh=RealImage("Y-shift with
    tilt",4,(2*nTilts)+1,(2*nTilts)+1 );
    //Ysh.DisplayAt(875,30);
    //Ysh.SetWindowSize(200,200);

    //Start the camera running in fast mode
    //Use current camera
    object camera = CMGetCurrentCamera();
    // Create standard parameters
    number kUnprocessed = 1;
    number kDarkCorrected = 2;
    number kGainNormalized = 3;
    number processing = kGainNormalized;
    // Define camera parameter set
    object acq_params =
    camera.CM_CreateAcquisitionParameters_FullCCD(processin
    g,expo,blinking,binning);
    acq_params.CM_SetDoContinuousReadout(true);
    acq_params.CM_SetQualityLevel(0); //what does this do?
    object acquisition =
    camera.CM_CreateAcquisition(acq_params);
    object frame set info =
    acquisition.CM_ACQ_GetDetector().DTCTR_CreateFrameSetIn
    fo();
    img_src =
    alloc(CM_AcquisitionImageSource).IMGSRC_Init(acquisitio
    n,frame set info,0);
    CM_ClearDarkImages() //why?

    // Create and display live image
    img1=acquisition.CM_CreateImageForAcquire("Live");
    img1.DisplayAt(10,30);
    img1.SetWindowSize(500,500);

    //Set up reference image img0
    img0=img1.ImageClone();
    img0.DisplayAt(15,30);
    img0.SetWindowSize(200,200);
    img0.SetName("Reference");
    //and img2 to display the measured beam positions
    img2=img1.ImageClone()*0;
    img2.DisplayAt(525,30);
    img2.SetWindowSize(500,500);
    img2.SetName("Progress");
    //and cross correlation image imgCC
    imgCC= img1.CrossCorrelate(img0);
    imgCC.DisplayAt(445,30);
    imgCC.SetWindowSize(200,200);
    imgCC.SetName("Cross correlation");

    ///////////////////////////////////
    // Start acquisition
    //NB define variables outside try/catch
    number
    Rr,TlX,TlY,T2X,T2Y,dTilt,detT,xTpX,xTpY,yTpY,yTpY;
    number
    ShlX,ShlY,Sh2X,Sh2Y,dShift,detSh,xShpX,xShpY,yShpX,yShp
    Y;
    number ShiftX0,ShiftY0,TiltX0,TiltY0,pt,maxval;
    number i,j,tX,tY,x,y,dx,dy,tIncX,tIncY,prog;
    try
    {
        img_src.IMGSRC_BeginAcquisition()
        UpdateCameraImage(img_src,img1);
        sleep(sleeptime);
        UpdateCameraImage(img_src,img1);
        //put the untilted image into img0

        detSh=ShlX*Sh2Y-Sh2X*ShlY; //determinant
        xShpX= Sh2Y/detSh; //EMSetBeamShift(xShpX,xShpY)
        shifts the beam i x-pixel
        xShpY= ShlY/detSh; //EMSetBeamShift(yShpX,yShpY)
        shifts the beam i y-pixel
        yShpX= ShlX/detSh;
        //Save them to the global tag group
        SetPersistentNumberNote(Tag_Path+"xShpX",xShpX);
        SetPersistentNumberNote(Tag_Path+"xShpY",xShpY);
        SetPersistentNumberNote(Tag_Path+"yShpX",yShpX);
        SetPersistentNumberNote(Tag_Path+"yShpY",yShpY);

        //Centre beam shift and display start point
        UpdateCameraImage(img_src,img1);
        sleep(sleeptime);
        UpdateCameraImage(img_src,img1);
        EMBeamCentre(Tag_Path);
        EMGetBeamShift(ShiftX0,ShiftY0);
        result ("Centred beam shift:
        "+ShiftX0+","+ShiftY0+"\n");
        SetPersistentNumberNote(Tag_Path+"ShiftX0",ShiftX0);
        SetPersistentNumberNote(Tag_Path+"ShiftY0",ShiftY0);
        EMGetBeamTilt(TiltX0,TiltY0);
        result ("Initial beam tilt: "+TiltX0+","+TiltY0+"\n");
        SetPersistentNumberNote(Tag_Path+"TiltX0",TiltX0);
        SetPersistentNumberNote(Tag_Path+"TiltY0",TiltY0);

        ///////////////////////////////////
        //Map beam shift as a function of tilt
        //Go to first point
        pt=1;
        GetCoordsFromNTilts(nTilts,pt,i,j);
        //increment in position of disc, in pixels
        tIncX=( z/(2*nTilts))*0.8; //max beam tilt is 80% of
        diffraction pattern width from centre
        tIncY=( b/(2*nTilts))*0.8; //max beam tilt is 80% of
        diffraction pattern height from centre

        img0=img1;

        //Get radius of CBED disc
        Rr=DiscSize(img0);
        result ("Disc radius="+Rr+" pixels\n");
        SetPersistentNumberNote(Tag_Path+"Disc Radius",Rr);

        //Set magnitude of tilt
        // dTilt gives diffraction pattern shift 1/4 of
        camera height
        // [TlX,TlY] is the beam tilt in pixels [X,Y] per x
        DAC

        dTilt=TiltSize(500000/CamL,TlX,TlY,T2X,T2Y); //40000/Cam
        L is initial guess
        result ("Set beam tilt: "+TlX+","+T2X+"\n");
        detT=TlX*T2Y-T2X*TlY; //determinant
        xTpX= T2Y/detT; //EMSetBeamTilt(xTpX,xTpY) shifts the
        disc 1 x-pixel
        xTpY= TlY/detT;
        yTpX= T2X/detT; //EMSetBeamTilt(yTpX,yTpY) shifts the
        disc 1 y-pixel
        yTpY= TlX/detT;
        //Save them to the global tag group
        SetPersistentNumberNote(Tag_Path+"xTpX",xTpX);
        SetPersistentNumberNote(Tag_Path+"xTpY",xTpY);
        SetPersistentNumberNote(Tag_Path+"yTpX",yTpX);
        SetPersistentNumberNote(Tag_Path+"yTpY",yTpY);

        //Go to imaging mode
        EMChangeMode("MAG1")

        //Calibrate beam shift - assume it is linear so a
        single measurement of x and y DAC shift is fine.
        // [ShlX,ShlY] is the beam shift in pixels [X,Y] per x
        DAC
        dShift=ShiftSize(10,ShlX,ShlY,Sh2X,Sh2Y); //first
        guess at shift increment is 20, changed to 1/4 of
        camera height

```

```

        tX=TiltX0 + (i*xTpX + j*yTpX)*tIncX;//convert from
        pixels to DAC for point i,j
        tY=TiltY0 + (i*xTpY + j*yTpY)*tIncY;
        EMSetBeamTilt(tX,tY);
        sleep(sleeptime);
        UpdateCameraImage(img_src,img1);
        sleep(sleeptime);
        UpdateCameraImage(img_src,img1);//throw this image
    away
    //measure beam shifts and put into XSh,ySh
    while (pt<nPts)
    {
        //set tilt
        GetCoordsFromNTilts(nTilts,pt,i,j);
        tX=TiltX0 + (i*xTpX + j*yTpX)*tIncX;
        tY=TiltY0 + (i*xTpY + j*yTpY)*tIncY;
        EMSetBeamTilt(tX,tY);
        //get image
        sleep(sleeptime);
        UpdateCameraImage(img_src,img1);
        sleep(sleeptime);
        UpdateCameraImage(img_src,img1);
        img0=img1;
        img0.RemoveOutliers(50);
        //measure beam position
        maxval=img0.max(x,y);
        result ("max " "xx", "yy" "\n")
        //mark&save it
        img2[y-rspot,x-rspot,y+rspot,x+rspot]=100;
        img2.UpdateImage();
        dx=x-(r/2);
        dy=y-(b/2);
        //      result ("X shift
        "+(i+nTilts)+", "+(j+nTilts)+", "+dx+" "\n")
        Xsh.SetPixel(i+nTilts,j+nTilts,-dx);//NB negative
        of measured value so shift cancels tilt
        Ysh.SetPixel(i+nTilts,j+nTilts,-dy);
        pt++;
    }

    //Put XSh and YSh into TiltCal layers 0 and 1 for
    saving
    TiltCal[0,0,0,(2*nTilts)+1,(2*nTilts)+1,1]=Xsh;
    TiltCal[0,0,1,(2*nTilts)+1,(2*nTilts)+1,2]=Ysh;
    result ("Beam shift calibration collected\n")
}
catch
{// We are here because an error happened, stop the
acquisition
    img_src.IMGSRG FinishAcquisition();
}
//stop acquisition
img_src.IMGSRG FinishAcquisition();

//Save calibration
SetPersistentStringNote(Tag_Path+"Calibration file
path",pathname);
TiltCal.SetStringNote("Info:Path",pathname)
TiltCal.SaveAsGatan(file name)
//Add tags & save Calibration (stack of 4 images)
TiltCal.CM_WriteAcquisitionTagsToImage(camera,acq_param
s)
TiltCal.SetStringNote("Info:Date",datetime)
TiltCal.SetNumberNote("Info:Camera Length",CamL)
TiltCal.SetNumberNote("Info:Magnification",mag)
TiltCal.SetNumberNote("Info:Alpha",Alpha)
TiltCal.SetNumberNote("Info:Spot size",spot);
TiltCal.SetNumberNote("Info:Disc Radius",Rr);
TiltCal.SetNumberNote("Tilts:xTpX",xTpX);
TiltCal.SetNumberNote("Tilts:xTpY",xTpY);
TiltCal.SetNumberNote("Tilts:yTpX",yTpX);
TiltCal.SetNumberNote("Tilts:yTpY",yTpY);
TiltCal.SetNumberNote("Shifts:xShpX",xShpX);
TiltCal.SetNumberNote("Shifts:xShpY",xShpY);
TiltCal.SetNumberNote("Shifts:yShpX",yShpX);
TiltCal.SetNumberNote("Shifts:yShpY",yShpY);

//tidy up
//reset tilts to original values

EMSetBeamTilt(TiltX0,TiltY0);
img0.DeleteImage();
img1.DeleteImage();
img2.DeleteImage();
imgCC.DeleteImage();
//End of main program

number tend = GetHighResTickCount();
result ("Elapsed time =
"+CalcHighResSecondsBetween(tstart,tend)+" seconds\n");
result ("Calibration complete, ding dong\n\n")

```

D.2 Calibration 2

```
// $BACKGROUNDS$
//*** Calibrate tilts2 ***\\
//** Richard Beanland r.beanland@warwick.ac.uk **\\
//** Coding started June 2012 **\\

// 1.0, 23 June 2012
// 1.1 - added fast data collection March 2013
// 1.2 - fast data collection as a subroutine April
2013
// 1.3 - bug fixes April/May 2013
// 1.4 - added image shift calibration Summer 2014
// 1.5 - bug fixes Oct 2016
// 2.0 - split off from tilt calibration Nov 2016

//Global variables
number true=1, false=0;
number CamB=4096, CamR=4096//Global values, for
oneViewIS camera
number binning=8;//Could be a dialogue
number expo=0.005;//Could be a dialogue
number _t=0;
number _l=0;
number _b=CamB/binning;
number _r=CamR/binning;
number rspot=2;
number sleeptime=0.00;//delay while waiting for
microscope to respond
image imgRef,img0,img1,img2,imgCC;//reference, sobel
filtered, live, progress and cross-correlation image
respectively
object img_src

//*****
// Subroutines.
//*****
//*****
//UpdateCameraImage
//RemoveOutliers

//GetCoordsFromNTilts
//EMChangeMode
//EMBeamCentre
//Tiltsize
//Shiftsize
//DiscSize
//BlurG
//Sobel

//*****
// Main program
//*****
number tstart = GetHighResTickCount();

//Get Basic stuff to start
number spot=EMGetSpotSize()*1;
number mag=EMGetMagnification(); //Sometimes gives null
answer, why??
//Switch to in diffraction mode
EMChangeMode("DIFF")
number camL=EMGetCameraLength();
EMChangeMode("MAG1")

//And prompt input for alpha
number alpha=3;
if (!GetNumber("Alpha?",alpha)) exit(0)
//set up tag and file paths
string
Tag_Path="DigitalDiffraction:Alpha="+alpha+":Binning="+
binning+":CamL="+CamL+"mm:";
string pathname=PathConcatenate(
GetApplicationDirectory("common_app_data",0),"Reference
Images\");
string file_name=pathname+"D-
ED_TiltCal_A"+Alpha+"_B"+binning+"_C"+CamL+".dmd";

string datetime;

//make or load calibration data, image TiltCal
GetPersistentStringNote(Tag_Path+"Date",datetime);
image TiltCal;
if (datetime=="")
{
throw("No tilt calibration: Run TiltCalibration
first!");
}
else
{
result("\nLast calibration "+datetime+"\n");
TiltCal := NewImageFromFile(file_name);
}

//update tags
number f;
string date;
GetDate(f,date);
string time;
GetTime(f,time);
datetime=date+"_"+time;
result("\nStarting shift calibration "+datetime+"\n");
SetPersistentStringNote(Tag_Path+"Date",datetime);
//load tilt calibration
number Rr,xTpX,xTpY,yTpX,yTpY,nTilts,TiltX0,TiltY0;
GetPersistentNumberNote(Tag_Path+"Spot size",spot);
GetPersistentNumberNote(Tag_Path+"nCells",nTilts);
GetPersistentNumberNote(Tag_Path+"xTpX",xTpX);
GetPersistentNumberNote(Tag_Path+"xTpY",xTpY);
GetPersistentNumberNote(Tag_Path+"yTpX",yTpX);
GetPersistentNumberNote(Tag_Path+"yTpY",yTpY);
GetPersistentNumberNote(Tag_Path+"TiltX0",TiltX0);
GetPersistentNumberNote(Tag_Path+"TiltY0",TiltY0);
EMSetBeamTilt(TiltX0,TiltY0);
result ("Initial beam tilt: "+TiltX0+","+TiltY0+"\n")

// Stop any current camera viewer
number close_view=1, stop_view=0;
try
{
cm_stopcurrentcameraViewer(stop_view);
}
catch
{
throw("Couldn't stop camera properly, try again!");
}

//set up images to contain data
number nPts=((2*nTilts)+1)**2;
//set up arrays holding shift correction
image Ysh=RealImage("Y-shift with
tilt",4,(2*nTilts)+1,(2*nTilts)+1);
//Xsh.DisplayAt(655,30);
//Xsh.SetWindowSize(200,200);
image Yshw=RealImage("Y-shift with
tilt",4,(2*nTilts)+1,(2*nTilts)+1);
//Ysh.DisplayAt(875,30);
//Ysh.SetWindowSize(200,200);

//Works in imaging mode
EMChangeMode("MAG1")
external sem = NewSemaphore();
try
{
ModelessDialog("Please spread the beam, large
condenser aperture, feature in the centre","OK",sem)
GrabSemaphore(sem)
ReleaseSemaphore(sem)
FreeSemaphore(sem)
}
catch
{
FreeSemaphore(sem)
break;
}
```

```

}

//Start the camera running in fast mode
//Use current camera
object camera = CMGetCurrentCamera();
// Create standard parameters
number data_type=2;
number kUnprocessed = 1;
number kDarkCorrected = 2;
number kGainNormalized = 3;
number processing = kGainNormalized;
// Define camera parameter set
object acq_params =
camera.CM CreateAcquisitionParameters FullCCD(processin
g,expo,binning,binning);
acq_params.CM_SetDoContinuousReadout(true);
acq_params.CM_SetQualityLevel(0);//what does this do?
object acquisition =
camera.CM CreateAcquisition(acq_params);
object frame_set_info =
acquisition.CM_ACQ_GetDetector().DTCTR_CreateFrameSetIn
fo();
img_src =
alloc(CM AcquisitionImageSource).IMGSRC Init(acquisitio
n,frame_set_info,0);
CM_ClearDarkImages(0)//why?

// Create and display live image
img1:=acquisition.CM CreateImageForAcquire("Live");
img1.DisplayAt(10,30);
img1.SetWindowSize(500,500);
//Set up reference image imgRef
imgRef:=img1.ImageClone();
//imgRef.DisplayAt(225,30);
//imgRef.SetWindowSize(200,200);
//imgRef.SetName("Reference");
//Set up filtered image img0
img0:=img1.ImageClone();
img0.DisplayAt(10,570);

img2[y-rspot,x-rspot,y+rspot,x+rspot]=1;

//Go to first point
pt=0;
GetCoordsFromNTilts(nTilts,pt,i,j);
tx=TiltX0 + (i*TpX + j*TpY)*tIncX;//convert from
pixels to DAC for point i,j
ty=TiltY0 + (i*TpX + j*TpY)*tIncY;
EMSetBeamTilt(tx,ty);
UpdateCameraImage(img_src,img1);//throw this image
away
UpdateCameraImage(img_src,img1);
//measure image shifts and put into XSh,ySh
while (pt<nPts)
{
    OpenAndSetProgressWindow("Measuring image
displacement","Image "+(pt+1)+" of "+nPts," "+prog+"
%");
    //set tilt
    GetCoordsFromNTilts(nTilts,pt,i,j);
    tx=TiltX0 + (i*TpX + j*TpY)*tIncX;
    ty=TiltY0 + (i*TpX + j*TpY)*tIncY;
    EMSetBeamTilt(tx,ty);
    //get image
    UpdateCameraImage(img_src,img1);
    UpdateCameraImage(img_src,img1);
    img1.RemoveOutliers(50);
    img0=img1.BlurG(2).Sobel();
    //measure beam position
    imgCC=img0.CrossCorrelate(imgRef);
    maxval=imgCC.Max(x,y);
    result(pt+1, displacement=["+x+", "+y+"]\n");
    //mark&save it
    img0[y-rspot,x-rspot,y+rspot,x+rspot]=max(img0);
    img2[y-rspot,x-rspot,y+rspot,x+rspot]=max(img2);
    img0.UpdateImage();
    dx=x-( r/2);
    dy=y-( b/2);

    img0.SetWindowSize(600,600);
    img0.SetName("Sobel filtered");
    //and img2 to display the measured beam positions
    img2:=img1.ImageClone()*0;
    img2.DisplayAt(530,30);
    img2.SetWindowSize(500,500);
    img2.SetName("Progress");
    //and cross correlation image imgCC
    imgCC:= img1.CrossCorrelate(img0);
    //imgCC.DisplayAt(445,30);
    //imgCC.SetWindowSize(200,200);
    //imgCC.SetName("Cross correlation");

    ///////////////////////////////////
    // Start acquisition
    //NB define variables outside try/catch
    number
    Sh1X,Sh1Y,Sh2X,Sh2Y,dShift,detSh,xShpX,xShpY,yShpX,yShp
    Y;
    number ShiftX0,ShiftY0,pt,maxval;
    number i,j,tX,tY,x,y,dx,dy,prog;
    //increment in position of disc, in pixels
    number tIncX=( r/(2*nTilts))*0.8;//max beam tilt is 80%
    of diffraction pattern width from centre
    number tIncY=( b/(2*nTilts))*0.8;//max beam tilt is 80%
    of diffraction pattern height from centre

    try
    {
        img_src.IMGSRC_BeginAcquisition()
        //Set up reference image in imgRef, 3x3 Sobel filter
        applied
        UpdateCameraImage(img_src,img1);
        UpdateCameraImage(img_src,img1);
        img1.RemoveOutliers(50);
        imgRef=img1.BlurG(2).Sobel();
        imgCC=imgRef.CrossCorrelate(imgRef);
        maxval=imgCC.Max(x,y);

        Xsh.SetPixel(i+nTilts,j+nTilts,-dx);//NB negative
        of measured value so shift cancels tilt
        Ysh.SetPixel(i+nTilts,j+nTilts,-dy);
        pt++;
    }
    //Put XSh and YSh into TiltCal layers 2 and 3 for
    saving
    TiltCal[0,0,2,(2*nTilts)+1,(2*nTilts)+1,3]=Xsh;
    TiltCal[0,0,3,(2*nTilts)+1,(2*nTilts)+1,4]=Ysh;
}
catch
{
    // We are here because an error happened, stop the
    acquisition
    img_src.IMGSRC_FinishAcquisition();
}
img_src.IMGSRC_FinishAcquisition();

//Save calibration
SetPersistentStringNote(Tag Path+"Calibration file
path",pathname);
TiltCal.SetStringNote("Info:Path",pathname)
TiltCal.SaveAsGatan(file_name)
//Add tags & save Calibration (stack of 4 images)
TiltCal.CM WriteAcquisitionTagsToImage(camera,acq_param
s);
TiltCal.SetStringNote("Info:Date",datetime)
TiltCal.SetNumberNote("Info:Camera Length",CamL);
TiltCal.SetNumberNote("Info:Disc Radius",Rr);
TiltCal.SetNumberNote("Info:Magnification",mag)
TiltCal.SetNumberNote("Info:Alpha",Alpha)
TiltCal.SetNumberNote("Info:Spot size",spot);
TiltCal.SetNumberNote("Info:Disc Radius",Rr);
TiltCal.SetNumberNote("Tilts:xTpX",xTpX);
TiltCal.SetNumberNote("Tilts:yTpY",yTpY);
TiltCal.SetNumberNote("Tilts:yTpX",yTpX);
TiltCal.SetNumberNote("Tilts:yTpY",yTpY);
TiltCal.SetNumberNote("Shifts:xShpX",xShpX);
TiltCal.SetNumberNote("Shifts:xShpY",xShpY);
TiltCal.SetNumberNote("Shifts:yShpX",yShpX);
TiltCal.SetNumberNote("Shifts:yShpY",yShpY);

```

D.3 Collect

```

// $BACKGROUND$
//*** Collect Digital Diffraction Pattern ***\\
//** Richard Beanland r.beanland@warwick.ac.uk **\\
//* Coding started March 2011 *\\

// Version 1.4a, 13 Apr 2012
// Orius SC600A on 2100 LAB6
// 1.5 - serpentine raster, 17 May 2012
// 1.7 - collect into memory and process, 26 Feb 2013
// 1.8 - added image shift calibration Summer 2014
// 1.8a - created interp function
// 2.0 - added material and other tags

//Realignment 11/4/14
//TEM mode Camera length 25cm ILL=320A ***for all
Alpha=***
//Alpha1, Camera length 20 cm, ILL=4AE5
//Alpha9, Camera length 20 cm, ILL=5040
//Alpha1, Camera length 25 cm, ILL=4B25
//Alpha3, Camera length 20 cm, ILL=5171 (51A7)

//Global variables
number true = 1, false = 0;
number CamL=4096, CamR=4096//Global values, for
OneViewIS camera
number binning=6; //Could be a dialogue
number exp=0.01; //Could be a dialogue
number sleeptime=0.00; //delay while waiting for
microscope to respond
number t = 0;
number i = 0;
number b = CamR/binning;
number r = CamR/binning;
image imgL; //live image
object img_src

//*****
//*****

number f;
string date;
getDate(f,date);
string time;
getTime(f,time);
string datetime=date+" "+time;
result("Starting acquisition "+datetime+"\n");
datetime="";
// Get alpha
number alpha=3;
if (!GetNumber("Alpha?",alpha,alpha));
exit(0);
//Check Calibration
number mag=EMGetMagnification(); //Sometimes gives null
answer, why??
number camL=EMGetCameraLength();
string
Tag_Path="DigitalDiffraction:Alpha="+alpha+"Binning="+
binning+"CamL="+CamL+"mm:";
GetPersistentStringNote(Tag_Path+"Date",datetime);
if (datetime=="")
{
throw("No tilt calibration - please recalibrate");
}
else
{
result("\nLast calibration "+datetime+"\n");
}
//*****
//Load calibration
number
Rr,spot,nCals,xTpX,xTpY,yTpX,yTpY,xShpX,xShpY,yShpX,yShpY;
string pathname,file_name,material;
GetPersistentNumberNote(Tag_Path+"Disc Radius",Rr);
GetPersistentNumberNote(Tag_Path+"Spot size",spot);
GetPersistentNumberNote(Tag_Path+"nCals",nCals);
GetPersistentNumberNote(Tag_Path+"xTpX",xTpX);
GetPersistentNumberNote(Tag_Path+"xTpY",xTpY);

// Subroutines.
//*****
//GetCoordsFromNTilts
//UpdateCameraImage
//Interp

//Function Interp
//Gives linear interpolation between four values in a
2x2 image A
number Interp(image A, number dX, number dY)
{
//Coefficients _a,_b,_c,_d for linear interpolation
//r=a+bx+cy+dxy
number _a=A.GetPixel(0,0);
number _b=(A.GetPixel(1,0)-_a);
number _c=(A.GetPixel(0,1)-_a);
number _d=(_a-A.GetPixel(0,1)-
A.GetPixel(1,0)+A.GetPixel(1,1));
number corA=_a+_b*dX+_c*dY+_d*dX*dY;
return corA
} //End of Interp

//*****
//*****
// Main program
//*****
//*****
//Check microscope status before starting
string mode=EMGetImagingOpticsMode();
if (! (mode=="DIFF"))//mag mode
{
throw ("TEM not in diffraction mode")
}

//*****
//Set up parameters with user input

GetPersistentNumberNote(Tag_Path+"yTpX",yTpX);
GetPersistentNumberNote(Tag_Path+"yTpY",yTpY);
GetPersistentNumberNote(Tag_Path+"xShpX",xShpX);
GetPersistentNumberNote(Tag_Path+"xShpY",xShpY);
GetPersistentNumberNote(Tag_Path+"yShpX",yShpX);
GetPersistentNumberNote(Tag_Path+"yShpY",yShpY);
GetPersistentStringNote(Tag_Path+"Calibration file
path",pathname);
file_name=pathname+"D-
EB TiltCal A"+Alpha+" B"+binning+" C"+CamL+".dnd";
GetPersistentStringNote(Tag_Path+"Material",material);
if (material=="")
{
if (!GetString("Material?",material,material));
exit(0);
}
//Set tilt increment to give a displacement of 40% of
disc diameter
number tInc=Rr*0.8; //in pixels
SetPersistentNumberNote(Tag_Path+"TiltIncrement",tInc);
//NB sets a limit on the max number of tilts since the
beam shift calibration
//is only correct for up to 80% of the diffraction
pattern width.
image TiltCal := NewImageFromFile(file_name);
image
Xsh=TiltCal[0,0,0,(C*nCals)+1,(C*nCals)+1,1]+TiltCal[0,
0,2,(C*nCals)+1,(C*nCals)+1,3];
Xsh.SetName("X-shift with tilt");
//Xsh.DisplayAt(655,30);
//Xsh.SetWindowSize(200,200);
image
Ysh=TiltCal[0,0,1,(C*nCals)+1,(C*nCals)+1,2]+TiltCal[0,
0,3,(C*nCals)+1,(C*nCals)+1,4];
Ysh.SetName("Y-shift with tilt");
//Ysh.DisplayAt(875,30);
//Ysh.SetWindowSize(200,200);
number xCal=(0.8*_z)/(C*nCals); //distance between
calibration points

```

```

number yCal=(0.8*b)/(2*nCals);//in units of pixels in
the diffraction pattern
//Get nTilts
number nTilts=99;
string prompt = "Number of beam tilts (+ & -): ";
result(prompt+" ");
if (!GetNumber(prompt,nTilts,nTilts))
exit(0)
//check that max number of beam tilts doesn't go
outside 80% of image
if (nTilts*tInc > nCals*xCal)//NB work on smallest
camera dimension
{
nTilts=floor((nCals*xCal)/tInc);
}
result(nTilts+"\n")
number nPts = (2*nTilts+1)**2;
//////////
//Get initial state
number TiltX0,TiltY0,ShiftX0,ShiftY0
EMGetBeamTilt(TiltX0,TiltY0)
result("Initial beam tilts: TiltX = "+TiltX0+", TiltY =
"+TiltY0+"\n")
EMGetBeamShift(ShiftX0,ShiftY0);
result("Initial beam shift: ShiftX = "+ShiftX0+",
ShiftY = "+ShiftY0+"\n")

//////////
// Stop any current camera viewer
number close_view=1, stop_view=0;
cm_stopcurrentcameraViewer(stop_view);

//////////
//Start the camera running in fast mode
//Use current camera
object camera = CMGetCurrentCamera();
// Create standard parameters
number kUnprocessed = 1;
number kDarkCorrected = 2;

number kGainNormalized = 3;
number processing = kGainNormalized;
// Define camera parameter set
object acq_params =
camera.CM CreateAcquisitionParameters FullCCD(processin
g,expo,binning,binning);
acq_params.CM_SetDoContinuousReadout(true);
acq_params.CM_SetQualityLevel(0);//what does this do?
object acquisition =
camera.CM CreateAcquisition(acq_params);
object frame_set_info =
acquisition.CM_Acq_GetDetector().DTCTR_CreateFrameSetIn
fo();
img_src =
alloc(CM AcquisitionImageSource).IMGSRG_Init(acquisitio
n,frame_set_info,0);
CM_ClearDarkImages(0)//why?
// Create and display fast image
img1=acquisition.CM CreateImageForAcquire("Acquired");
img1.DisplayAt(15,30);
img1.SetWindowSize(800,800);
number data_type=img1.GetDataTypes();
number imgX,imgY
img1.Get2DSize(imgX,imgY);

//////////
// Create 3D destination data stack
image CBED_stack := NewImage("CBED
Stack",data_type,imgX,imgY,nPts);
CBED_stack = 0;
CBED_stack.DisplayAt(835,30)
CBED_stack.SetWindowSize(800,800)

//////////
//Go to first point
number i,j,pX,pY;
number pt=0;
GetCoordsFromNTilts(nTilts,pt,i,j);
//tilts to be applied, in pixels

pX=i*tInc;
pY=j*tInc;
//tilt to be applied, in DAC numbers
number tX=TiltX0 + xTpX*pX + yTpX*pY;
number tY=TiltY0 + xTpY*pX + yTpY*pY;
//linear interpolation for beam shift correction
number dX=pX/xCal & 1;//fractional coords
number dY=pY/yCal & 1;
//The 4 surrounding shift calibrations
image
xX=XSh[floor(pY/yCal)+nCals,floor(pX/xCal)+nCals,floor(
pY/yCal)+nCals+2,floor(pX/xCal)+nCals+2];
image
yY=YSh[floor(pY/yCal)+nCals,floor(pX/xCal)+nCals,floor(
pY/yCal)+nCals+2,floor(pX/xCal)+nCals+2];
//interpolated values
number corX=Interp(Xx,dX,dY);
number corY=Interp(Yy,dX,dY);
//shift correction vector
number sX=ShiftX0 + corX*xShpX + corY*yShpX;
number sY=ShiftY0 + corX*xShpY + corY*yShpY;
//set tilt and shift
EMSetBeamShift(sX,sY);
EMSetBeamTilt(tX,tY);
sleep(sleeptime);
number sleepcounter=0.00
//////////
// Start acquisition
//NB define variables outside try/catch
number prog
try
{
img_src.IMGSRG_BeginAcquisition()

while(pt < Npts)
{
prog=round(100*(pt+1)/nPts);
OpenAndSetProgressWindow("Data collection","Image
"+(pt+1)+" of "+nPts," "+prog+" %");

GetCoordsFromNTilts(nTilts,pt,i,j);
//tX and tY as a fraction of image width
pX=i*tInc;//in pixels
pY=j*tInc;
//tilt correction vector
tX=TiltX0 + xTpX*pX + yTpX*pY;
tY=TiltY0 + xTpY*pX + yTpY*pY;
//linear interpolation for beam shift correction
dX=pX/xCal & 1;
dY=pY/yCal & 1;

xX=XSh[floor(pY/yCal)+nCals,floor(pX/xCal)+nCals,floor(
pY/yCal)+nCals+2,floor(pX/xCal)+nCals+2];
yY=YSh[floor(pY/yCal)+nCals,floor(pX/xCal)+nCals,floor(
pY/yCal)+nCals+2,floor(pX/xCal)+nCals+2];
corX=Interp(Xx,dX,dY);
corY=Interp(Yy,dX,dY);
//shift correction vector
sX=ShiftX0 + corX*xShpX + corY*yShpX;
sY=ShiftY0 + corX*xShpY + corY*yShpY;
//set tilt and shift
EMSetBeamShift(sX,sY);
EMSetBeamTilt(tX,tY);
UpdateCameraImage(img_src,img1);
UpdateCameraImage(img_src,img1);
CBED_stack[0,0,pt,imgX,imgY,pt+1]=img1;
pt++;
}
}
catch
{
// We are here because an error happened, stop the
acquisition
img_src.IMGSRG_FinishAcquisition();
}
//reset tilts to original values
EMSetBeamTilt(TiltX0,TiltY0);
EMSetBeamShift(ShiftX0,ShiftY0);

```

D.4 Process

```
// $BACKGROUND$
//*** Process Digital Diffraction Pattern ***\\
//** Richard Beanland r.beanland@warwick.ac.uk **\\
//** Coding started March 2013 **\\

// 2.8, 17 Apr 2013
// Orius SC600A on 2100 LaB6
// Complete new version to match compensated data
// collection
// 2.9 bug fixes April/May 2013
// 2.10 added image shift compensation
// 3.0 added several subroutines for cubic spline of
// background
// 3.1 background fit done on average CBED image rather
// than central CBED pattern

//Global variables
number true = 1, false = 0;
number pi=3.1415926535897932384626433832795;

//*****
//Subroutines.
//*****
//CalculateSplineConstants
//CubicSpline
//SplineRows
//SplineInterp
//DiscMask
//ROIpos
//UserG
//okGdialog
//GetCoordsFromNTilts
//AddYellowArrow
//GetMeanG
//GetG_vectors

// Function CalculateSplineConstants
image CalculateSplineConstants(image dataset)
// Calculates the cubic spline constants - by David
// Mitchell
{
    number n, sizeX, sizeY, minx, maxx, yspline, i,
    prevval, thisx,m,j
    dataset.getSize(sizeX,sizeY)
    // the number of data points
    n=sizeX-1
    minmax(dataset[0,0,1,sizeX], minx, maxx)

    // Arrays to store the data points
    // note the data start at pixel position 1 - pixel
    // position 0 is not used
    image x=realimage("",4,sizeX+1, 1)
    image a=realimage("",4,sizeX+1, 1)
    image xw=realimage("",4,sizeX+1, 1)
    image b=realimage("",4,sizeX+1, 1)
    image xl=realimage("",4,sizeX+1, 1)
    image xw=realimage("",4,sizeX+1, 1)
    image xz=realimage("",4,sizeX+1, 1)
    image b=realimage("",4,sizeX+1, 1)
    image c=realimage("",4,sizeX+1, 1)
    image d=realimage("",4,sizeX+1, 1)
    image constantarray=realimage("",4,sizeX+1,5) //
    stores all the constants and the x values

    // set the x and y arrays to the respective values in
    // the dataset passed in
    x[0,1,1,sizeX+1]=dataset[0,0,1,sizeX]
    a[0,1,1,sizeX+1]=dataset[1,0,2,sizeX]
    m=n-1
    for(i=0; i<m+1;i++)
    {
        h[0,i+1,1,i+2]=getpixel(x,i+2,0)-getpixel(x, i+1,0)
    }
    for(i=1; i<m+1;i++)

constantarray[4,0,5,sizeX+1]=x[0,0,1,sizeX+1]

    return constantarray
}

//End of calculatesplineconstants

// Function CubicSpline
number CubicSpline(image constantarray, number xvalue,
number extrapolate)
//Input an array of spline constants, an x value
// from which to interpolate a yvalue, and a boolean
// (interpolate). If interpolate is set to 1 then
// interpolation outside the range will be performed,
// otherwise values outside this range return a zero.
// the function returns the interpolated y value. As
// the constants are only calculated once in the
// other function, the computation in this function is
// very fast and suitable for use in a loop
// to calculate a range of y values
{
    number minx, maxx, sizeX, sizeY, yspline, i, n
    getsizes(constantarray, sizeX, sizeY)
    // Get the minimum and maximum values in the x data -
    // bottom row of the array from position 1 to n
    minmax(constantarray[4,1,5,sizeX], minx, maxx) //
    ignore position 0
    // Check that the passed in xvalue is within the range
    // of the values in the data set supplied or zero.
    // If the extrapolate option is turned off (0) then
    // the function returns a y value of zero
    if(extrapolate==0) // do not extrapolate, any out of
    range values of x result a y value of zero
    {
        if(xvalue<minx || xvalue>maxx)
        {
            yspline=0
            return yspline
        }
    }
}
```



```

    }
    // loop through the x data (row 5 - position 4 in the
    constantarray) to find which interval the passed in
    xvalue lies in
    n=sizeX-2 // note - 2 because pixel position 0 is
    unused.
    for(i=1;i<n;i++)
    {
        if(xvalue<getpixel(constantarray,i,4)) break
        if(xvalue>=getpixel(constantarray,i,4) &&
        xvalue<getpixel(constantarray,i+1,4)) break
    }
    // Calculate the distance between the lower bound x
    data point for the interval and the passed in xvalue
    number xcalc=xvalue-getpixel(constantarray,i,4)
    // Compute the spline a=row 0, b=row 1, c=row 2 and
    d=row 3
    // y=a +bxcalc+c*xcalc^2+dxdcalc^3
    y spline=getpixel(constantarray,i,0)+getpixel(constantarray,i,1)*xcalc+getpixel(constantarray,i,2)*xcalc**2+getpixel(constantarray,i,3)*xcalc**3

    return y spline
} //end of CubicSpline

// Function SplineRows
image SplineRows(image RowsOut, image BackNumbers,
number gmag)
// Calculates a set of cubic splines for several rows
of measurement
//Input is the array of background measurements and an
image to be filled with
//a cubic spline in one dimension. gmag the magnitude
of the g-vector
//
{
    number nMeas1,nMeas2,LenSp,ind,jnd,yspline,s0,s1,s,ra;
    image SplineConstants;
    BackNumbers.GetSize(nMeas1,nMeas2);
    RowsOut.GetSize(LenSp,nMeas2);
    image Spline := RealImage("SplineFit",4,LenSp,1);
    //data to send to spline constant calculation routine,
    top row
    image SplineInput := RealImage("spline
    input",4,nMeas1,2);
    SplineInput[0,0,1,nMeas1]=round(icol*gmag);
    //calculate cubic spline for rows and put into RowsOut
    for (jnd=0; jnd<nMeas2; jnd++)
    {
        //fill in the bottom row of the input array and
        calculate the spline
        SplineInput[1,0,2,nMeas1]=BackNumbers[icol,jnd];

    SplineConstants=SplineInput.CalculateSplineConstants()
    //SplineConstants.DisplayAt(100,100);
    for(ind=0; ind<LenSp;ind++)
    {
        yspline=SplineConstants.CubicSpline(ind,0)
        Spline.setpixel(ind,0,yspline)
    }
    //Spline.UpdateImage();
    //put it in the appropriate row
    RowsOut[icol,jnd]=Spline;
}

return RowsOut
} //End of SplineRows

// Function SplineInterp
image SplineInterp(image BackOut, image SplineRow,
image SplineCol, number gimag, number g2mag )
//Takes two SplineRow images and produces a 2D
interpolation of them
{
    Ishift=(SplineCol.GetPixel(ind,row)+SplineCol.GetPixel(
    ind+1,row)-avSpline.GetPixel(0,0)-
    avSpline.GetPixel(round(gimag)-1,0))/2;
    BackOut[row,1,row+1,r]=avSpline+Ishift
}
}
return BackOut
} //End of SplineInterp

// Function DiscMask
image DiscMask(image LocalMask, number Rdisc, number
g1X, number g1Y, number g2X, number g2Y)
// Makes a mask to cover the CBED discs
{
    LocalMask=1;
    number LrX=round((abs(g1X)+abs(g2X))/2);
    number LrY=round((abs(g1Y)+abs(g2Y))/2);
    image Disc:= RealImage("Disc",4,2*Rdisc,2*Rdisc);
    Disc=tert( (radius<Rdisc), 0,1); //black disc
    number t,l,b,r,tt,ll,bb,rr
    //Note two possible cases, origin on the left (g1Y<0)
    or the top (g1Y>0)

    //half disc on left
    tt=(round(LrY-(g1Y+g2Y)/2)-Rdisc)*(g1Y<0)+(round(LrY-
    g1Y-g2Y)/2)-Rdisc)*(g1Y>0);
    bb=(round(LrY-(g1Y+g2Y)/2)+Rdisc)*(g1Y<0)+(round(LrY-
    g1Y-g2Y)/2)+Rdisc)*(g1Y>0);
    t=(tt>0)?tt:(tt<0)?0;
    l=0;
    b=(bb<(2*LrY+1))*bb+(bb>(2*LrY))*2*LrY;
    r=Rdisc;
    LocalMask[t,l,b,r]=LocalMask[t,l,b,r]*Disc[t-
    tt,Rdisc,b-bb+2*Rdisc,2*Rdisc];

    //half disc on right

```

```

        tt=(round(LrY+(g1Y+g2Y)/2)-
Rdisc)*(g1Y<0)+(round(LrY+(g1Y-g2Y)/2)-Rdisc)*(g1Y>0);

bb=(round(LrY+(g1Y+g2Y)/2)+Rdisc)*(g1Y<0)+(round(LrY+(g
1Y-g2Y)/2)+Rdisc)*(g1Y>0);
t=(tt>0)*tt+(tt<0)*0;
l=2*LrX-Rdisc;
b=(bb<(2*LrY+1))*bb+(bb>(2*LrY))*2*LrY;
r=2*LrX;
LocalMask[t,l,b,r]=LocalMask[t,l,b,r]*Disc[t-tt,0,b-
bb+2*Rdisc,Rdisc];

//half disc on top
ll=(round(LrX+(g1X-g2X)/2)-Rdisc)*(g1Y<0)+(round(LrX-
(g1X+g2X)/2)-Rdisc)*(g1Y>0);
rr=(round(LrX+(g1X-g2X)/2)+Rdisc)*(g1Y<0)+(round(LrX-
(g1X+g2X)/2)+Rdisc)*(g1Y>0);
t=;
l=(ll>0)*ll+(ll<0)*0;
b=Rdisc;
r=(rr<(2*LrX+1))*rr+(rr>(2*LrX))*2*LrX;
LocalMask[t,l,b,r]=LocalMask[t,l,b,r]*Disc[Rdisc,l-
ll,2*Rdisc,r-rr+2*Rdisc];

//half disc on bottom
ll=(round(LrX-(g1X-g2X)/2)-
Rdisc)*(g1Y<0)+(round(LrX+(g1X+g2X)/2)-Rdisc)*(g1Y>0);
rr=(round(LrX-(g1X-
g2X)/2)+Rdisc)*(g1Y<0)+(round(LrX+(g1X+g2X)/2)+Rdisc)*(
g1Y>0);
t=2*LrY-Rdisc;
l=(ll>0)*ll+(ll<0)*0;
b=2*LrY;
r=(rr<(2*LrX+1))*rr+(rr>(2*LrX))*2*LrX;
//result("t,l,b,r="+t+","+l+","+r+","+b+","+n");
LocalMask[t,l,b,r]=LocalMask[t,l,b,r]*Disc[0,l-
ll,Rdisc,r-rr+2*Rdisc];

//Radius

LocalMask=ttert(iradius<1*Rdisc,LocalMask,0)

return LocalMask
} //End of DiscMask

//Function ROIpos
//Gives top,left,bottom,right of an ROI drawn by a user
void ROIpos(image img, string prompt, number &t, number
&l, number &b, number &r)
{
    number IsizX,IsizY,clickedOK;
    img.GetSize(IsizX,IsizY);
    ImageDisplay img disp = img.ImageGetImageDisplay(0);
    roi theROI = NewROI();
    theROI.ROISetRectangle(t,l,b,r);
    img_disp.ImageDisplayAddROI(theROI);
    clickedOK = true;
    external sem = NewSemaphore();
    try
    {
        ModelessDialog(prompt,"OK",sem);
        GrabSemaphore(sem);
        ReleaseSemaphore(sem);
        FreeSemaphore(sem);
    }
    catch
    {
        FreeSemaphore(sem);
        clickedOK = false;
        break;
    }
    img_disp.ImageDisplayDeleteROI(theROI);
    if(clickedOK)
    {
        theROI.ROIGetRectangle(t,l,b,r);
    }
} //End of function ROIpos

//Function AddText
void AddText(image img, number x, number y, string
text)
{
    number bigly=50
    component imgdisp=img.imagegetimagedisplay(0);
    component words=NewTextAnnotation({x+3},{y-
bigly/2},text,bigly)
    words.componentsetforegroundColor(1,1,0)
    words.componentsetfontface("Microsoft Sans Serif")
    imgdisp.componentaddchildatend(words)
} //end of AddText

//Function AddBlueCircle
void AddBlueCircle(image img, number t, number l,
number b, number r)
{
    component imgdisp=img.imagegetimagedisplay(0);
    component ov=NewOvalAnnotation(t,l,b,r); //create a
circle defined by top, left etc.
    ov.componentsetforegroundColor(0,0,1); //make the
circle blue
    imgdisp.componentaddchildatend(ov); //add the
circle to the image
} //end of AddBlueCircle

//Function AddYellowArrow
void AddYellowArrow(image img, number x0, number y0,
number x1, number y1)
{
    component imgdisp=img.imagegetimagedisplay(0);
    component arrowannot=NewArrowAnnotation(y0,x0,y1,x1);
    // create a single-headed arrow bounded by the
rectangle defined by top, left etc.
    arrowannot.componentsetforegroundColor(1,1,0); // make
the arrow yellow

arrowannot.componentsetDrawingMode(1); // turn
background fill to on
arrowannot.componentsetBackgroundColor(0,0,0); // set
the background fill to black
imgdisp.componentaddchildatend(arrowannot); // add the
arrow to the image
} //end of AddYellowArrow

//Function DeleteStuff
//Deletes circles, lines and text from an image
void DeleteStuff(image img)
{
    component imgdisp=img.imagegetimagedisplay(0);
    number i,n,j
    for (i=0; i<15; i++)
    {
        n=imgdisp.componentcountchildrenoftype(i);
        // result(i+": "+n+" kids\n");
        for (j=0; j<n; j++)
        {
            component
            id=imgdisp.componentgetnthchildtoftype(i,0)
            id.componentremovefromparent(0)
        }
    }
} //End of DeleteStuff

//Function UserG
//Gives top,left,bottom,right of an ROI drawn by a user
void UserG(image Avg, number Rr, number &g1X, number
&g1Y, number &g2X, number &g2Y, number &pXavg, number
&pYavg,)
{
    number t,l,b,r,IsizX,IsizY;
    string prompt;
    image AvgTemp;
    Avg.GetSize(IsizX,IsizY);

```

```

//Clean up average image
Avg.DeleteStuff();
//Get g-vectors manually
prompt = "Position ROI on central beam and hit OK";
result(prompt+"...");
number i=0;
number j=0;
t=pYavg+g1Y*i+g2Y*j-Rr;
l=pXavg+g1X*i+g2X*j-Rr;
b=pYavg+g1Y*i+g2Y*j+Rr;
r=pXavg+g1X*i+g2X*j+Rr;
Avg.ROIpos(prompt,t,l,b,r);
result(" done\n");
pXavg=(l+r)/2;
pYavg=(t+b)/2;
//First diffraction vector
prompt = "Position ROI on 1st g and hit OK";
result(prompt+"...");
i=0;
j=0;
t=pYavg+g1Y*i+g2Y*j-Rr;
l=pXavg+g1X*i+g2X*j-Rr;
b=pYavg+g1Y*i+g2Y*j+Rr;
r=pXavg+g1X*i+g2X*j+Rr;
Avg.ROIpos(prompt,t,l,b,r);
number order=2;
if (!GetNumber("Diffraction order?",order,order))
exit(0)
result(" done\n");
g1X=((l+r)/2)-pXavg)/order;
g1Y=((t+b)/2)-pYavg)/order;
//Second diffraction vector
prompt = "Position ROI on 2nd g and hit OK";
result(prompt+"...");
i=0;
j=0;
t=pYavg+g1Y*i+g2Y*j-Rr;
l=pXavg+g1X*i+g2X*j-Rr;
b=pYavg+g1Y*i+g2Y*j+Rr;
r=pXavg+g1X*i+g2X*j+Rr;
Avg.ROIpos(prompt,t,l,b,r);
number order=2;
if (!GetNumber("Diffraction order?",order,order))
exit(0)
result(" done\n");
g1X=((l+r)/2)-pXavg)/order;
g1Y=((t+b)/2)-pYavg)/order;
//Function GetMeanG
void GetMeanG(image MagTheta, image cCluster, number
cMeanMag, number cMeanTheta, number cMeanVx, number
cMeanVy, number cNMeas)
{
//incoming mean values are a single vector
//outgoing mean values are an average of the vectors
deemed to be in the cluster
//incoming mag/theta is the same for +/- vectors
[since tan(q+p)=tan(q)]
//so flip the vectors if the x-component has an
opposite sign
}

number tolMag=5;//tolerance in pixels to say it's in a
cluster
number tolAng=5;//tolerance in degrees to say it's in
a cluster
number
n,nVecs,i,dTheta,dMag,ThetaSum,VxSum,VySum,x,y,signX;
Cluster.Get2DSize(n,nVecs);
nMeas=0;
number MagSum=0;
for (i=0; i<nVecs; i++)
{
dMag=abs(MeanMag - MagTheta.GetPixel(0,i));
dTheta=abs(MeanTheta - MagTheta.GetPixel(1,i) );
if ( ( dTheta<tolAng ) && (dMag<tolMag ) )
{
nMeas++;
MagSum+=MagTheta.GetPixel(0,i);
ThetaSum+=MagTheta.GetPixel(1,i);
x=MagTheta.GetPixel(2,i);
y=MagTheta.GetPixel(3,i);//
signX=abs(x)/x;
VxSum+=x*(abs(MeanVx)/MeanVx)*signX;//second part
here reverses
VySum+=y*(abs(MeanVx)/MeanVx)*signX;//sign if x is
opposite sign
Cluster[i,0,i+1,1]=1;//it is in the cluster
MeanMag=MagSum/nMeas;
MeanTheta=ThetaSum/nMeas;
MeanVx=VxSum/nMeas;
MeanVy=VySum/nMeas;
}
}
}
//End of GetMeanG

//Function GetG vectors
//Gets the two smallest g-vectors [g1X,g1Y],[g2X,g2Y],
given an image Avg and disc radius Rr
//will return null values if there are not enough peaks
to analyse

r=pXavg+g1X*i+g2X*j+Rr;
Avg.ROIpos(prompt,t,l,b,r);
order=2;
if (!GetNumber("Diffraction order?",order,order))
exit(0)
result(" done\n");
g2X=((l+r)/2)-pXavg)/order;
g2Y=((t+b)/2)-pYavg)/order;
//show discs and g-vectors on average image
for (i=-2; i<3; i++)
{
for(j=-2; j<3; j++)
{
Avg.AddBlueCircle(pYavg+g1Y*i+g2Y*j-
Rr,pXavg+g1X*i+g2X*j-
Rr,pYavg+g1Y*i+g2Y*j+Rr,pXavg+g1X*i+g2X*j+Rr);
}
}
Avg.AddYellowArrow(
pXavg,pYavg,(pXavg+g1X),(pYavg+g1Y) );
Avg.AddYellowArrow(
pXavg,pYavg,(pXavg+g2X),(pYavg+g2Y) );
Avg.AddText((pXavg+g1X),(pYavg+g1Y),"1");
Avg.AddText((pXavg+g2X),(pYavg+g2Y),"2");
}
//End of function UserG

void GetG vectors(image Avg, number Rr, number g1X,
number g1Y, number g2X, number g2Y, number pXavg,
number pYavg,)
{
number npeaks=25;//maximum number of peaks to measure
in the cross correlation
number IsizX,IsizY;
Avg.GetSize(IsizX,IsizY);
IsizX=IsizX/2;//using an average image twice the width
of the CBED stack
IsizY=IsizY/2;//using an average image twice the
height of the CBED stack
//start by getting a cross-correlation image
//image of a blank disk, radius Rr
image Disc=realimage("Disc",4,2*IsizX,2*IsizY);
Disc=tert(iradius<Rr,1,0);
//Disc.DisplayAt(225,30);
//Disc.SetWindowSize(200,200);

//Cross correlation between average image and the
blank disc image
//gives the position of the central beam
image AvCC=Avg.CrossCorrelate(Disc);
number maxval,xp,yp;
maxval=AvCC.max(xp,yp);
pXavg=xp;
pYavg=yp;
result("000 beam is at ["+pXavg+","+pYavg+"]\n");

//////////THIS WILL PROBABLY HAVE TO BE A USER
INPUT//////////
number dSize=3.5//a multiplying factor for the disc
size when deleting peaks which have already been
measured. Ideally dSize*Rr should be half of the
smallest g-vector.
number DdSize=round(dSize*Rr);
Disc.deleteimage();//tidy up
//delete all info below 2% of best correlation
number _top=max(AvCC)*0.02;

```

```

AvCC=tert( (AvCC> top),AvCC,0.001); //make background
not quite zero (mainly for debug, so can see the
deleted peaks)
//AvCC.DisplayAt(445,30);
//AvCC.SetWindowSize(200,200);

//x- and y-coords as column vectors
image X:= RealImage("X-coords",4,1,npeaks);
//X.DisplayAt(665,30);
//X.SetDisplayType(5); //show as spreadsheet
image Y:= RealImage("Y-coords",4,1,npeaks);
//Y.DisplayAt(885,30);
//Y.SetDisplayType(5); //show as spreadsheet
image TempImg:=realimage("Deleting
disc",4,2*DdSize,2*DdSize); //A dark...
TempImg=tert(iradius<dSize*Rr,0,1); //...circle
//TempImg.DisplayAt(225,230);
number i;
number Nmax=nPeaks;
number flag=0;
for (i=0; i<nPeaks; i++)
//get peak position, in descending order of
correlation/intensity
if (max(AvCC)> top)
//only keep going if there are peaks to be found
maxval=AvCC.max(xp,yp);
X.SetPixel(0,i,xp); //x-coord of peak
Y.SetPixel(0,i,yp); //y-coord of peak
Avg.AddBlueCircle(yp-Rr,xp-Rr,yp+Rr,xp+Rr);
//result("Spot "+i+ " = "+maxval+", at
"+xp+", "+yp+",\n");
SetSelection(AvCC, (yp-DdSize), (xp-
DdSize), (yp+DdSize), (xp+DdSize));
AvCC[i]*=TempImg; //this peak is done, delete it
ClearSelection(AvCC);
}
else
{ //No peaks left
if (flag==0)

{
Nmax=i; //reduce number of peaks
flag=1;
}
}
nPeaks=Nmax;
if (nPeaks<3)
{//there are not enough detected spots, return null
values
//g1X=0;
//g1Y=0;
//g2X=0;
//g2Y=0;
return
}
//Find difference vectors Vx, Vy, by replicating X and
Y into square matrices Xx, Yy, and subtracting the
transpose
image Xx:= RealImage("Xx",4,npeaks,npeaks) //
image Vx:= RealImage("Vx",4,npeaks,npeaks)
image Yy:= RealImage("Yy",4,npeaks,npeaks)
image Vy:= RealImage("Vy",4,npeaks,npeaks)
Xx=X[0,irow];
Vx=Xx[irow,icol]-Xx;
Vy=Y[0,irow];
Vy=Yy[irow,icol]-Yy;
//Polar coordinates, Vmag and Vtheta
image Vmag:= RealImage("Vmag",4,npeaks,npeaks);
Vmag=(( (Vx*Vx)+(Vy*Vy)
)*0.5)*(irow*icol); //irow*icol gives bottom left
diagonal half;
image Vtheta:= RealImage("Vtheta",4,npeaks,npeaks);
number big=1000000; //an arbitrary number larger than
anything else
Vx=tert((Vx==0),big,Vx); //get rid of divide by zero
error
Vtheta=atan(Vy/Vx)*(irow*icol);
Vtheta=Vtheta*180/pi;

//Sort by magnitude ascending into new column vector
MagTheta
number nVecs=(npeaks*npeaks-npeaks)/2; //number of
different vectors
image MagTheta:= RealImage("Mag-Theta-X-Y",4,4,nVecs);
//MagTheta.DisplayAt(645,30);
//MagTheta.SetName("Mag-Theta-X-Y");
//MagTheta.SetDisplayType(5); //show as spreadsheet
//MagTheta.SetWindowSize(120,500);
Vmag=tert((Vmag==0),big,Vmag); //replace zeroes with
this big number
number Mag=Vmag.min(xp,yp); //lowest magnitude in the
list
i=0;
while (Mag < big) //go through list until all are
replaced by the large number
{
MagTheta[i,0,i+1,1]=Vmag[yp,xp,yp+1,xp+1]; //first
col=magnitude
MagTheta[i,1,i+1,2]=Vtheta[yp,xp,yp+1,xp+1]; //second
column=theta
MagTheta[i,2,i+1,3]=Vx[yp,xp,yp+1,xp+1]; //third
column=Vx
MagTheta[i,3,i+1,4]=Vy[yp,xp,yp+1,xp+1]; //fourth
column=Vy
Vmag[yp,xp,yp+1,xp+1]=big; //this point is done,
eliminate from Vmag
i=i+1;
Mag=Vmag.min(xp,yp);
}
//set sign of theta [not needed, just flip them in the
sum
//MagTheta[0,1,nVecs,2]=MagTheta[0,1,nVecs,2]*MagTheta[
0,3,nVecs,3]/abs(MagTheta[0,3,nVecs,4]); //second
column=theta

//Find clusters - similar g-vectors in mag-theta space
image Cluster:= RealImage("Cluster",4,1,nVecs);

//Cluster.DisplayAt(645,30);
//Cluster.SetName("Cluster");
//Cluster.SetDisplayType(5); //show as spreadsheet
//Cluster.SetWindowSize(120,500);
image gVectors:= RealImage("g's",4,5,nVecs);
//gVectors.DisplayAt(855,30);
//gVectors.SetName("g-vectors");
//gVectors.SetDisplayType(5); //show as spreadsheet
//gVectors.SetWindowSize(120,500);
number MeanMag=MagTheta.GetPixel(0,0); //start with mag
of first point
number MeanTheta=MagTheta.GetPixel(1,0); //and with
angle of first point
number MeanVx=MagTheta.GetPixel(2,0); //and with X of
first point
number MeanVy=MagTheta.GetPixel(3,0); //and with Y of
first point
number j=1;
number k=0;
number nMeas=1; //number of measured points to give an
average g-vector
//Go through and get clusters
while (sum(Cluster)<nVecs)
{
GetMeanG(MagTheta,Cluster,MeanMag,MeanTheta,MeanVx,Mean
Vy,nMeas);
gVectors[k,0,k+1,1]=MeanMag;
gVectors[k,1,k+1,2]=MeanTheta;
gVectors[k,2,k+1,3]=MeanVx;
gVectors[k,3,k+1,4]=MeanVy;
gVectors[k,4,k+1,5]=nMeas;
//Find next unmatched point
i=0;
while (j==1)
{
i++;
j=Cluster.GetPixel(0,i);
}
}

```

```

j=1;
MeanMag=MagTheta.GetPixel(0,i);//next point
MeanTheta=MagTheta.GetPixel(1,i);//next point
Cluster[i,0,i+1,1]=1;//next cluster
k++
}
//Output - the two smallest g-vectors
glX=gVectors.GetPixel(2,0);
glY=gVectors.GetPixel(3,0);
g2X=gVectors.GetPixel(2,1);
g2Y=gVectors.GetPixel(3,1);
result("Found "+nPeaks+" different CBED disks,\n")
result("giving "+k+" different g-vectors\n")
//show g-vectors on average image
Avg.AddYellowArrow(
pXavg,pYavg,(pXavg+glX),(pYavg+glY));
Avg.AddYellowArrow(
pXavg,pYavg,(pXavg+g2X),(pYavg+g2Y));
}
//End of GetG_vectors

//Function GetGids
//Gets HKL of g-vectors from user
void GetGids(number &glH, number &glK, number &glL,
number &g2H, number &g2K, number &g2L, number &glAg2,
number &glMg2, number &gC)
{
if (!GetNumber("First g index H?",glH,glH)) exit(0);
if (!GetNumber("First g index K?",glK,glK)) exit(0);
if (!GetNumber("First g index L?",glL,glL)) exit(0);
if (!GetNumber("Second g index H?",g2H,g2H)) exit(0);
if (!GetNumber("Second g index K?",g2K,g2K)) exit(0);
if (!GetNumber("Second g index L?",g2L,g2L)) exit(0);
result("g1: "+glH+", "+glK+", "+glL+"\n");
result("g2: "+g2H+", "+g2K+", "+g2L+"\n");
if (!GetNumber("Angle between them?",glAg2,glAg2))
exit(0);
if (!GetNumber("Ratio of magnitudes
gl/g2?",glMg2,glMg2)) exit(0);

result("Ratio of g-vector magnitudes = "+glMg2+",
angle= "+glAg2+" degrees\n");
if (!GetNumber("Centring g-vector (0,1,2)?",gC,gC))
exit(0);
if (gC==0) result("The pattern is not face-
centred\n");
if (gC==1) result("g1 is a face-centring vector\n");
if (gC==2) result("g2 is a face-centring vector\n");
}
//End of function GetGids

//*****
//Main program
//*****
result("\nD-ED Process v3.3.0\n")

number f;
string date;
getDate(f,date);
string time;
getTime(f,time);
string datetime=date+" "+time;
result("\nStarting processing "+datetime+"\n")

//*****
//Get 3D data stack
image CBED_stack:=GetFrontImage();
number IsizX,IsizY,nPts;
CBED_stack.Get3DSize(IsizX,IsizY,nPts);
number data type = CBED_stack.GetDataTypes();
number
Rr,tInc,CamL,mag,Alpha,spot,xTpX,xTpY,yTpX,yTpY,xShpX,x
ShpY,yShpX,yShpY;
string material;
//get image tags
{
CBED_stack.GetStringNote("Info:Date",datetime);

number disX,disY,pt;//vector describing displacement of
a tilted CBED pattern
number minX=IsizX,minY=IsizY;//most negative
displacement values
for (pt=0; pt<nPts; pt++)
{
prog=round(100*(pt+1)/nPts)
OpenAndSetProgressWindow("Average CBED
pattern","Image "+(pt+1)+" of "+nPts," "+prog+" %");
GetCoordsFromNTilts(nTilts,pt,i,j);
disX=round(i*tInc)+round(0.5*IsizX);
if (disX<minX) minX=disX;
disY=round(j*tInc)+round(0.5*IsizY);
if (disY<minY) minY=disY;

Avg[disY,disX,IsizY+disY,IsizX+disX]+CBED_stack[0,0,pt
,IsizX,IsizY,pt+1];
AvgC[disY,disX,IsizY+disY,IsizX+disX]+1;
Avg.UpdateImage();
}
//make average
Avg=tert((Avg>0),Avg/AvgC,0);
//fill outside with representative background value
number rBack=Avg.GetPixel(minX+5,minY+5);
//result("\nrBack="+rBack+"\n");
//result("minX="+minX+", minY="+minY+"\n");
Avg=tert((Avg>0),Avg,rBack);
result(" done\n");
Avg.DisplayAt(30,30);
Avg.SetWindowSize(600,600);

//*****
//Find g-vectors
number
glX,glY,glmag,g2X,g2Y,g2mag,pXavg,pYavg,theta;
Avg.GetG_vectors(Rr,glX,glY,g2X,g2Y,pXavg,pYavg);
number pX=pXavg-round(0.5*IsizX);//there is a
difference between the centre

```

```

number pY=pYavg-round(0.5*IsizY);//in the avg image and
the CBED stack
//Check for failure to get g-vectors
if (g1X**2 > (IsizX**2)/4) g1X=0
if (g1Y**2 > (IsizX**2)/4) g1Y=0
if ( (g1X+g1Y+g2X+g2Y)==0 )
{
    result("Cannot find g-vectors!");
}
else
{
    //output g-vector statistics
    Avg.AddText((pXavg+g1X), (pYavg+g1Y), "1");
    Avg.AddText((pXavg+g2X), (pYavg+g2Y), "2");
    g1mag=(g1X**2+g1Y**2)**0.5;
    g2mag=(g2X**2+g2Y**2)**0.5;
    result("g1 = [" +g1X+", "+g1Y+"], magnitude
    "+g1mag+"\n");
    result("g2 = [" +g2X+", "+g2Y+"], magnitude
    "+g2mag+"\n");
    ratio=g1mag/g2mag;
    theta=180*acos((g1X*g2X+g1Y*g2Y)/(g1mag*g2mag))/pi;
    result("Ratio of g-vector magnitudes = "+ratio+",
    angle= "+theta+"\n");
}

//////////
//Check g-vectors are acceptable, if not do them
manually
number t,l,b,r;
string prompt;
while (!TwoButtonDialog("Are the measured g-vectors
good?", "Yes", "No") )
{
    Avg.UserG(Rr, g1X, g1Y, g2X, g2Y, pXavg, pYavg);
    pX=pXavg-round(0.5*IsizX);
    pY=pYavg-round(0.5*IsizY);
    g1mag=(g1X**2+g1Y**2)**0.5;
    g2mag=(g2X**2+g2Y**2)**0.5;
    result("000 beam is at [" +pX+", "+pY+"]\n");
}

g2Y=swap;
swap=g1mag;
g1mag=g2mag;
g2mag=swap;
//result("g1 (new)=-g2 (old), g2 (new)=g1 (old)\n")
redo=1;
}
else
{
    if (sgn(cross)<0)//270<theta<360
    //swap g1 and g2
    result("Swapping g-vectors to make a right-handed
    pair\n");
    result("Note the new values: x=right, y=down\n");
    swap=g1X;
    g1X=g2X;
    g2X=swap;
    swap=g1Y;
    g1Y=g2Y;
    g2Y=swap;
    swap=g1mag;
    g1mag=g2mag;
    g2mag=swap;
    //result("g1 (new)=g2 (old), g2 (new)=g1 (old)\n")
    redo=1;
}
}
//make g1X positive
if (g1X<0)
{
    result("Swapping g-vectors to make a right-handed
    pair\n");
    result("Note the new values: x=right, y=down\n");
    g1X=-g1X;
    g1Y=-g1Y;
    g2X=-g2X;
    g2Y=-g2Y;
    redo=1;
}

result("g1: "+g1X+", "+g1Y+", magnitude
"+g1mag+"\n");
result("g2: "+g2X+", "+g2Y+", magnitude
"+g2mag+"\n");
ratio=g1mag/g2mag;
theta=180*acos((g1X*g2X+g1Y*g2Y)/(g1mag*g2mag))/pi;
result("Ratio of g-vector magnitudes = "+ratio+",
angle= "+theta+"\n");
}

//////////
//g-vector calculations
//make sure a perfect alignment (g1Y=0) never happens
if (g1Y==0) g1Y=g1Y+0.000000001
number dot=(g1X*g2X+g1Y*g2Y)/(g1mag*g2mag);//gives
cos(theta)
number cross=(g1X*g2Y-g1Y*g2X)/(g1mag*g2mag);//g1 x g2
gives sin(theta)
number swap;
number redo=0;//flag to redraw annotations
if (sgn(dot)<0)//90<theta<270
//if the angle between g1 and g2 is not clockwise
0<theta<180 degrees, swap them
result("Swapping g-vectors to make a right-handed
pair\n");
result("Note the new values: x=right, y=down\n");
if (sgn(cross)<0)//180<theta<270
//change the sign of g2
g2X=-g2X;
g2Y=-g2Y;
//result("changed sign of g2\n")
}
else//90<theta<180
//g1=-g2 (old), g2=g1 (old)
swap=g1X;
g1X=-g2X;
g2X=swap;
swap=g1Y;
g1Y=-g2Y;

}
//redraw discs and g-vectors on average image
if (redo==1)
{
    Avg.DeleteStuff();
    number i,j;
    for (i=-2; i<3; i++)
    {
        for (j=-2; j<3; j++)
        {
            Avg.AddBlueCircle(pYavg+g1Y*i+g2Y*j-
            Rr, pXavg+g1X*i+g2X*j-
            Rr, pYavg+g1Y*i+g2Y*j+Rr, pXavg+g1X*i+g2X*j+Rr);
        }
        Avg.AddYellowArrow(
        pXavg, pYavg, (pXavg+g1X), (pYavg+g1Y) );
        Avg.AddYellowArrow(
        pXavg, pYavg, (pXavg+g2X), (pYavg+g2Y) );
        Avg.AddText((pXavg+g1X), (pYavg+g1Y), "1");
        Avg.AddText((pXavg+g2X), (pYavg+g2Y), "2");
    }
    //recalculate dot & cross products
    dot=(g1X*g2X+g1Y*g2Y)/(g1mag*g2mag);//gives cos(theta)
    cross=(g1X*g2Y-g1Y*g2X)/(g1mag*g2mag);//g1 x g2 gives
    sin(theta)
    theta=(acos(dot)*sgn(cross));
    //result("theta="+180*theta/pi+"\n");
    result("g1 = [" +g1X+", "+g1Y+"]: g2 =
    [" +g2X+", "+g2Y+"]\n");
    //Angle phi between g1 and the x-axis
    number dotX=g1X/g1mag;//gives cos(phi)
    number crossX=-g1Y/g1mag;//g1 x [100] gives sin(phi)
    number phi=(acos(dotX)*sgn(crossX))
    //result("phi="+180*phi/pi+"\n");

    //Get g-vector details
    number nG1=3;
}

```

```

if (!GetNumber("Number of spots for 1st g (+/-
)?",nG1,nG1)) exit(0);
number nG2=3;
if (!GetNumber("Number of spots for 2nd g (+/-
)?",nG2,nG2)) exit(0);
//Diameter of circular selection
number Wfrac=75
if (!GetNumber("Percentage of spot used, 67-
100%",Wfrac,Wfrac))
exit(0);
number g1H=1;
number g1K=0;
number g1L=0;
number g2H=0;
number g2K=1;
number g2L=0;
number g1Ag2=90;//nominal angle between g1 and g2
number g1Mg2=1;//nominal ratio of magnitudes, g1/g2
number gC=0;//for centred patterns, 0=no centring,
otherwise the g-vector in the centre
GetGids(g1H,g1K,g1L,g2H,g2K,g2L,g1Ag2,g1Mg2,gC);
while (!TwoButtonDialog("Are the g-vector HKLs
correct?", "Yes", "No"))
{
result("Repeating g-vector HKLs\n")
GetGids(g1H,g1K,g1L,g2H,g2K,g2L,g1Ag2,g1Mg2,gC);
}
//Put tags into Avg
{
Avg.SetStringNote("Info:Date",datetime);
Avg.SetNumberNote("Info:Camera Length",CamL);
Avg.SetNumberNote("Info:Magnification",mag);
Avg.SetNumberNote("Info:Alpha",Alpha);
Avg.SetNumberNote("Info:Spot size",spot);
Avg.SetNumberNote("Info:Disc Radius",Rr);
Avg.SetStringNote("Info:Material",material);
Avg.SetNumberNote("g-vectors:nG1",nG1);
Avg.SetNumberNote("g-vectors:nG2",nG2);
Avg.SetNumberNote("g-vectors:g1X",g1X);
Avg.SetNumberNote("g-vectors:g1Y",g1Y);
Avg.SetNumberNote("g-vectors:g2X",g2X);
Avg.SetNumberNote("g-vectors:g2Y",g2Y);
Avg.SetNumberNote("g-vectors:g1H",g1H);
Avg.SetNumberNote("g-vectors:g1K",g1K);
Avg.SetNumberNote("g-vectors:g1L",g1L);
Avg.SetNumberNote("g-vectors:g2H",g2H);
Avg.SetNumberNote("g-vectors:g2K",g2K);
Avg.SetNumberNote("g-vectors:g2L",g2L);
Avg.SetNumberNote("g-vectors:g1Ag2",g1Ag2);
Avg.SetNumberNote("g-vectors:g1Mg2",g1Mg2);
Avg.SetNumberNote("g-vectors:gC",gC);
Avg.SetNumberNote("g-vectors:pXavg",pXavg);
Avg.SetNumberNote("g-vectors:pYavg",pYavg);
}

//////////
// Background calculation
// the number of background measurements needed for g1
and g2
number nMeas1=2*nG1+2;//from -nG1 to +nG1, plus the
zero column and the final row
number nMeas2=2*nG2+2;
//size of spline fits are |g1|*(nMeas1-1) & (nMeas2-
1)*|g2|
number LenSpl1=round(g1mag*(nMeas1-1));
number LenSpl2=round(g2mag*(nMeas2-1));
//size of LocalBackImg
number LrX=round((abs(g1X)+abs(g2X))/2);
number LrY=round((abs(g1Y)+abs(g2Y))/2);
number Rdisc=round(min(g1mag,g2mag)/2);

//images for calculation of 2D background
image LocalBackImg := RealImage("Local
area",4,2*LrX,2*LrY);//image for measurement of local
background
image LocalMask:= RealImage("Local
mask",4,2*LrX,2*LrY);//same size image LocalMask

image BackNumbers := RealImage("Background
measurements",4,nMeas1,nMeas2);//image to store array
of background measurements
image BackNumbers1 := RealImage("Background
flip",4,nMeas1,nMeas2);//to generate numbers for zero
measurements
image BackNumbersTr:= RealImage("Transposed
measurements",4,nMeas2,nMeas1);//transposed version for
column calculation
image Rows := RealImage("Spline
rows",4,LenSpl1,nMeas2);//interpolated on rows
image RowsTr := RealImage("Tr(Spline
rows)",4,nMeas2,LenSpl1);//transposed rows
image Cols := RealImage("Spline
columns",4,nMeas1,LenSpl2);//interpolated on columns
image ColsTr := RealImage("Tr(Spline
columns)",4,LenSpl2,nMeas1);//transposed to use row calc
image Back :=
RealImage("Background",4,LenSpl1,LenSpl2);//average
image BackTr := RealImage("Background (Cols
only)",4,LenSpl2,LenSpl1);//transposed interpolated
number LenW1=round(LenSpl1+abs(LenSpl2*cos(theta)));
image BackShear := RealImage("Background
shear",4,LenW1,LenSpl2);//first shear the image
image BackFlip := RealImage("Background
flipped",4,LenW1,LenSpl2);//horizontal flip needed for
theta>90
number LenW2=round(LenSpl2*sin(theta));
image BackWarp := RealImage("Background
warped",4,LenW1,LenW2);//second compress it, maintains
g2 length
image BackRot = BackWarp.Rotate(phi);//rotated to match
the image
BackRot.SetName("Background rotated");
number RsizX,RsizY;
BackRot.GetSize(RsizX,RsizY);
//Make the mask
DiscMask(LocalMask,Rdisc,g1X,g1Y,g2X,g2Y);
number X,Y,ind,jnd,knd,t1,l1,b1,r1,inside,row,ra,Ishift

//LocalMask.DisplayAt(550,50);
//Back.DisplayAt(30,30);
//Back.SetWindowSize(300,300);
//BackTr.DisplayAt(30,350);
//BackTr.SetWindowSize(300,300);
//BackShear.DisplayAt(300,30);
//BackShear.SetWindowSize(300,300);
//BackFlip.DisplayAt(330,350);
//BackFlip.SetWindowSize(300,300);
//BackWarp.DisplayAt(660,30);
//BackWarp.SetWindowSize(300,300);
//BackRot.DisplayAt(660,350);
//BackRot.SetWindowSize(300,300);

//////////
// Measure background values from array of areas
between discs in average image
BackNumbers=0;//reset measurements
number gNo=0;
for (ind=-nG1; ind<nG1+2; ind++)
{
for (jnd=-nG2; jnd<nG2+2; jnd++)
{
prog=round(100*(gNo/((2*nG1+1)*(2*nG2+1))));
OpenAndSetProgressWindow("Measure
background...", "Image *gNo+" of
"+(2*nG1+1)*(2*nG2+1)," "+prog+" %");
//GetCoordsFromNTilts(nTilts,pt,i,j);
//appropriate point -(g1+g2)/2 from disk jnd,ind
X=round(pXavg+ind*g1X+jnd*g2X-(g1X+g2X)/2);
Y=round(pYavg+ind*g1Y+jnd*g2Y-(g1Y+g2Y)/2);
i=Y-LrY;
b=Y+LrY;
l=X-LrX;
r=X+LrX;
inside=((l<0)+(r>2*IsizX)+(t<0)+(b>2*IsizY));//could

```

```

be more sophisticated here and use part of the mask
when it goes outside the image
if (inside)//mask off CBED discs and get mean value
of what remains
{
    LocalBackImg=Avg[t,l,b,r]*LocalMask;
    //Put the measurement into BackNumbers

BackNumbers.SetPixel(ind+nG1,jnd+nG2,sum(LocalBackImg)/
sum(LocalMask));
}
gNo++;
}

//////////
// Make 2D cubic Splines
{
    OpenAndSetProgressWindow("Making 2D cubic
    spline...", "rows", " ");
    //Construct a background in an orthogonal image
    //Make an set of spline rows
    Rows.SplineRows(BackNumbers, g1mag);
    //Make a (transposed) set of spline columns
    BackNumbersTr=BackNumbers[irow,icol];
    OpenAndSetProgressWindow("Making 2D cubic
    spline...", "columns", " ");
    ColsTr.SplineRows(BackNumbersTr, g2mag);
    //transpose back to columns again
    Cols=ColsTr[irow,icol];
    //Combine the two 1D solutions for each reflection
    OpenAndSetProgressWindow("Making 2D cubic
    spline...", "rows+columns", " ");
    //use row splines for the image and column splines to
    match intensities
    Back.SplineInterp(Rows, Cols, g1mag, g2mag);
    //use column splines for the image and row splines to
    match intensities (all transposed)
    RowsTr=Rows[irow,icol];

    OpenAndSetProgressWindow("Background removal", "Image
    "+(pt+1)+" of "+nPts, " +prog+ " %");
    //get the coordinates to put the background into.
    GetCoordsFromTilts(nTilts, pt, i, j);
    //Datum background point is given by -nG1, -nG2
    X=floor(pX+ j*tInc-(nG1+0.5)*g1X-(nG2+0.5)*g2X);
    Y=floor(pY+ j*tInc-(nG1+0.5)*g1Y-(nG2+0.5)*g2Y);
    //offset of origin in rotated background
    if (phi<0)//NB we avoid rotations larger than +/-90
    by choosing g-vectors correctly
    {
        X=X-round(LenW2*abs(sin(phi)));
    }
    else
    {
        Y=Y-round(LenW1*abs(sin(phi)));
    }
    //check edges of ROI in CBED stack
    t=(Y>0)*Y+(Y<0)*0;
    l=(X>0)*X+(X<0)*0;
    b=((Y+RszY)<IsizY)*(Y+RszY)+((Y+RszY)>=IsizY)*(IsizY
    );
    r=((X+RszX)<IsizX)*(X+RszX)+((X+RszX)>=IsizX)*(IsizX
    );
    //check edges of ROI in BackRot
    t1=(Y>0)*0+(Y<0)*(-Y);
    l1=(X>0)*0+(X<0)*(-X);
    b1=((Y+RszY)<IsizY)*RszY+((Y+RszY)>=IsizY)*(IsizY-
    Y);
    r1=((X+RszX)<IsizX)*RszX+((X+RszX)>=IsizX)*(IsizX-
    X);
    //result("t,l,b,r="+t+", "+l+", "+b+", "+r+", "+n);
    //result("t,l,b,r1="+t1+", "+l1+", "+b1+", "+r1+", "+n);
    CBED stack[l,t,pt,r,b,pt+1]=CBED stack[l,t,pt,r,b,pt+1]
    -BackRot[t1,l1,b1,r1];

    OpenAndSetProgressWindow("Making 2D cubic
    spline...", "columns+rows", " ");
    BackTr.SplineInterp(ColsTr, RowsTr, g2mag, g1mag);
    //take the average of the two solutions
    Back=(Back+BackTr[irow,icol])/2;
    OpenAndSetProgressWindow("Making 2D cubic
    spline...", "scaling", " ");
}

//////////
// Warp to match g-vectors
//Deform the background image to match the g-vectors
{
    if (theta<pi/2)
    //deformations leave the top left pixel [0,0]
    unchanged
    BackShear=Back[icol-irow*cos(theta), irow]; //shear
    BackWarp=BackShear[icol, irow/sin(theta)]; //squash=rota
    tion
}
else
//the same with flips to leave the top right pixel
unchanged
    BackFlip=Back[LenSpl-icol, irow];
    BackShear=BackFlip[icol-irow*cos(pi-theta), irow];
    BackFlip=BackShear[LenSpl+LenSp2*cos(pi-theta)-
    icol, irow];
    BackWarp=BackFlip[icol, irow/sin(theta)];
}
BackRot=BackWarp.Rotate(phi);
}

//////////
// Subtract from CBED stack
{
    for (pt=0; pt<nPts; pt++)
    {
        prog=round(100*(pt+1)/nPts);
        //subtract from avg
        X=floor(pXavg-(nG1+0.5)*g1X-(nG2+0.5)*g2X)-
        round(LenW2*abs(sin(phi)))*phi<0);
        Y=floor(pYavg-(nG1+0.5)*g1Y-(nG2+0.5)*g2Y)-
        round(LenW1*abs(sin(phi)))*phi<0);
        //check edges of ROI in CBED stack
        t=(Y>0)*Y+(Y<0)*0;
        l=(X>0)*X+(X<0)*0;
        b=((Y+RszY)<2*IsizY)*(Y+RszY)+((Y+RszY)>2*IsizY)*(2*
        IsizY);
        r=((X+RszX)<2*IsizX)*(X+RszX)+((X+RszX)>2*IsizX)*(2*
        IsizX);
        //check edges of ROI in BackRot
        t1=(Y>0)*0+(Y<0)*(-Y);
        l1=(X>0)*0+(X<0)*(-X);
        b1=((Y+RszY)<2*IsizY)*RszY+((Y+RszY)>2*IsizY)*(2*Isi
        zY-Y);
        r1=((X+RszX)<2*IsizX)*RszX+((X+RszX)>2*IsizX)*(2*Isi
        zX-X);
        //Avg[t,l,b,r]=Avg[t,l,b,r]-BackRot[t1,l1,b1,r1];
    }

    //////////
    // Create 3D data stack for D-LACBED images
    result("Creating stack of D-LACBED images...");
    image DLACBEDimg:=NewImage("D-
    LACBED Stack", data_type, IsizX, IsizY, ((2*nG1+1)*(2*nG2+1
    )));
    DLACBEDimg=0;
    DLACBEDimg.DisplayAt(0, 625);
    DLACBEDimg.SetWindowSize(200, 200);

    //Create scratch image for calculation of average
    image ScratImg := RealImage("Average", 4, IsizX, IsizY)
    number Rr2=round(Wfrac*Rr/100);
    //other images for cut and copy
    image TempImg := RealImage("Disk", 4, 2*Rr2, 2*Rr2);
    image vTempImg := RealImage("Temp", 4, 2*Rr2, 2*Rr2);

```



```

vTempImg=tent((iradius<Rr2), 1,0);

//loop over DLACBED stack and build the patterns
gNo=0;
for (ind=-nG1; ind<nG1+1; ind++)
{
    for (jnd=-nG2; jnd<nG2+1; jnd++)
    {
        prog=round(100*( (gNo+1)/((2*nG1+1)*(2*nG2+1)) ));
        OpenAndSetProgressWindow("Creating D-LACBED patterns", "Image "+(gNo+1)+" of "+(2*nG1+1)*(2*nG2+1), "
        "+prog+" %");
        //loop over CBED stack
        for (pt=0; pt<nPts; pt++)
        {
            GetCoordsFromNTilts(nTilts,pt,_i,_j);
            //appropriate vector for disk
            X=round(pX+ i*inc*(ind*g1X)+(jnd*g2X));
            Y=round(pY+ j*inc*(ind*g1Y)+(jnd*g2Y));
            inside=((X-Rr2<0)+(X+Rr2>IsizX)+(Y-
            Rr2<0)+(Y+Rr2>IsizY));
            if (inside)
            {
                TempImg=CBED_stack[X-Rr2,Y-Rr2,pt,
                X+Rr2,Y+Rr2,pt+1]; //The disk of interest
                TempImg=tent((iradius<Rr2),
                TempImg,0); //Cropped to be circular
                DLACBEDImg[X-Rr2,Y-Rr2,gNo, X+Rr2,Y+Rr2,gNo+1]
                += TempImg; //Add it to the LACBED pattern
                ScratImg[Y-Rr2,X-Rr2,Y+Rr2,X+Rr2] +=
                vTempImg; //Update mask which keeps count of the number
                of images in one pixel
                TempImg=tent((vTempImg>TempImg), vTempImg,
                TempImg)
            }
            ScratImg+=(SratImg==0); //make pixels with zero
            values equal 1
        }
    }
    gNo++;
}

DLACBEDImg[0,0,gNo, IsizX,IsizY,gNo+1] /=
SratImg; //divide by mask
SratImg=0;
gNo++;
}

DLACBEDImg.SetLimits(DLACBEDImg.min(),DLACBEDImg.max())
//Tidy up
TempImg.DeleteImage();
vTempImg.DeleteImage();
CBED_stack.DeleteImage();

//Put tags into LACBED_stack
{
    DLACBEDImg.SetStringNote("Info:Date",datetime);
    DLACBEDImg.SetNumberNote("Info:Camera Length",CamL);
    DLACBEDImg.SetNumberNote("Info:Magnification",mag);
    DLACBEDImg.SetNumberNote("Info:Alpha",Alpha);
    DLACBEDImg.SetNumberNote("Info:Spot size",spot);
    DLACBEDImg.SetNumberNote("Info:Disc Radius",Rr);
    DLACBEDImg.SetStringNote("Info:Material",material);
    DLACBEDImg.SetNumberNote("g-vectors:nG1",nG1);
    DLACBEDImg.SetNumberNote("g-vectors:nG2",nG2);
    DLACBEDImg.SetNumberNote("g-vectors:g1X",g1X);
    DLACBEDImg.SetNumberNote("g-vectors:g1Y",g1Y);
    DLACBEDImg.SetNumberNote("g-vectors:g2X",g2X);
    DLACBEDImg.SetNumberNote("g-vectors:g2Y",g2Y);
    DLACBEDImg.SetNumberNote("g-vectors:g1H",g1H);
    DLACBEDImg.SetNumberNote("g-vectors:g1K",g1K);
    DLACBEDImg.SetNumberNote("g-vectors:g1L",g1L);
    DLACBEDImg.SetNumberNote("g-vectors:g2H",g2H);
    DLACBEDImg.SetNumberNote("g-vectors:g2K",g2K);
    DLACBEDImg.SetNumberNote("g-vectors:g2L",g2L);
    DLACBEDImg.SetNumberNote("g-vectors:g1Ag2",g1Ag2);
    DLACBEDImg.SetNumberNote("g-vectors:g1Mg2",g1Mg2);
    DLACBEDImg.SetNumberNote("g-vectors:gC",gC);
}
result(" done\n");
}

{
    //a2 is the centre of the rectangle where the D-
    LACBED image comes from in the stack
    a2X=round(pX+ind*g1X+jnd*g2X);
    a2Y=round(pY+ind*g1Y+jnd*g2Y);
    //result("centre: "+a2X+", "+a2Y+"\n")
    //Bounding rectangle for each D-LACBED image
    t2=round((a2Y-wid)*(1-((a2Y-wid)<0))); //could
    also be *((!(a2Y-wid)<0))
    l2=round((a2X-wid)*(1-((a2X-wid)<0)));
    b2=round((a2Y+wid)*(1-((a2Y+wid)>IsizY)) +
    ((a2Y+wid)>IsizY)*IsizY);
    r2=round((a2X+wid)*(1-((a2X+wid)>IsizX)) +
    ((a2X+wid)>IsizX)*IsizX);
    //result("copy from:
    "+t2+", "+l2+", "+b2+", "+r2+"\n")
    //a1 is the location of the rectangle where the D-
    LACBED image will go in the montage
    a1X=round(Lx + (ind*g1X*F+jnd*g2X*F));
    a1Y=round(Ly + (ind*g1Y*F+jnd*g2Y*F));
    //result("centre: "+a1X+", "+a1Y+"\n")
    t1=round(a1Y-a2Y+t2);
    l1=round(a1X-a2X+l2);
    b1=round(a1Y+b2-a2Y);
    r1=round(a1X+r2-a2X);
    //result("paste to: "+t1+", "+l1+", "+b1+", "+r1+"\n")
    inside=((l1<0)+(r1>Fsiz)+(t1<0)+(b1>Fsiz));
    //inside=(t1>0)*(l1>0)*(b1<Fsiz)*(r1<Fsiz);
    if (inside)
    {
        //Montage[t1,l1,b1,r1] = DLACBEDImg[l2,t2,gNo
        ,r2,b2,gNo+1];
        Montage[t1,l1,b1,r1] = tent(
        (DLACBEDImg[l2,t2,gNo
        ,r2,b2,gNo+1]==0), Montage[t1,l1,b1,r1],DLACBEDImg[l2,t2
        ,gNo ,r2,b2,gNo+1])
    }
    gNo++;
}
}

```

D.5 Align

```

// $BAGGROUND$
// **** D-IACBED aligner ****
// *** Richard Beandland r.beandland@warwick.ac.uk ***
// ** Coding started Jan 2018 **

// 1.0, 27 Jan 2018
// 1.2, 8 19 Apr 2018 using parabola to find the best
// angle and shear

// Global variables
number true = 1, false = 0;
number pi=3.1415926535897932384626433832795;
number tiny=0.000001; // a small number
number window=50; // size of window around pattern
// centre for cross correlation
image mask=RealImage("Mask",4,window,window);
mask=exp(-(iradius/(0.3*window))**2);
number te,le,be,re; // the edges of the data

////////////////////
// Subroutines.
////////////////////
//FindEdges
//Merge2
//Merge2
//Parabola
//PatternCentre
//SymmetryMatrix
//ApplySym
//SymmetryAdd
//BlurG
//Sobel
//Linwrap
//FindEdges
//skewfit
//totfit
//Median

//Function Merge2
//Shifts image 2 and puts it in image 1 with sub-pixel
//accuracy, linear interpolation
//adapted from Merge2
image Merge2(image img1, image img2, number x, number
y, number SumNo)
{
//stupidly this code shifts the image the wrong way
number negX=-x
number negY=-y
number sizX,sizY;
img1.GetSize(sizX,sizY);
img1=SumNo; //weight the destination image by the
number averaged
image countSum=SumNo*tert((img1==0),0,1); //mask for
averaging
number fx=floor(negX);
number fy=floor(negY);
//result("x:",fx,"y:",fy,"fx+fx","fy+fy","\n");
//bounding rectangle for copy from img2
number tl=abs(round(-fy*(fx<0))); //abs needed to avoid
problems with -0 (?)
number ll=abs(round(-fx*(fx<0)));
number bl=round(sizY-fy*(fx>0));
number rl=round(sizX-fx*(fx<0));
number h=bl-tl;
number w=rl-ll;
//result("tbl:",tl*","+ll*","+bl*","+rl*","\n");
//numbers for linear interpolation
number dx = negX - floor(negX);
number dy = negY - floor(negY);
number tla = (l-dx) * (l-dy);
number bra = dx * dy;
number bla = (l-dx)*dy;
number tra = dx * (l-dy);
//Add 4x image2 into temp image using linear
interpolation
image temp=img1[tl,1,bl,rl]*0
temp[0,0,h-1,w-1]+= tla*img2[tl,1,bl,rl-1];

temp[0,1,h-1,w]= tra*img2[tl,1,bl-1,rl-1];
temp[1,0,h,w-1]= bla*img2[tl,1,bl-1,rl-1];
temp[1,1,h,w]= bra*img2[tl,1,bl-1,rl-1];
//bounding rectangle for paste to img1
number t2=abs(round(fy*(fy>0)));
number l2=abs(round(fx*(fx>0)));
number b2=round(sizY+fy*(fy<0));
number r2=round(sizX+fx*(fx<0));
//result("tbl2:",t2*","+l2*","+b2*","+r2*","\n");
img1[t2+1,l2+1,b2-1,r2-1]= temp[tl,1,h-1,w-1];

return img1
} //End of Merge2

//Function Merge2
//Merges image 2 onto image 1 with sub-pixel accuracy,
//linear interpolation
image Merge2(image img1, image img2, number x, number
y, number SumNo)
{
number sizX,sizY;
img1.GetSize(sizX,sizY);
img1=SumNo; //weight the destination image by the
number averaged
image countSum=SumNo*tert((img1==0),0,1); //mask for
averaging
number fx=floor(x);
number fy=floor(y);
//bounding rectangle for copy from img2
number tl=abs(round(fy*(fy>0))); //abs needed to
avoid problems with -0 (?)
number ll=abs(round(-fx*(fx<0)));
number bl=round(sizY+fy*(fy<0));
number rl=round(sizX+fx*(fx<0));
//bounding rectangle for paste to img1
number t2=abs(round(sizY-b1)*(t1<0));
number l2=abs(round(sizX-r1)*(t1<0));
number b2=round(sizY-tl);
number r2=round(sizX-ll);
//numbers for linear interpolation
number xinter = x - floor(x);
number yinter = y - floor(y);
number tla = (l-xinter) * (l-yinter);
number bra = xinter * yinter;
number bla = (l-xinter)*yinter;
number tra = xinter * (l-yinter);
//Add 4x image2 into temp image using linear
interpolation
image temp=img1[t2,l2,b2-1,r2-1]*0 +
tla*img2[tl,1,bl-1,rl-1]+tra*img2[tl,1,bl-1,rl-1]+
bla*img2[tl+1,1,bl,rl-1]+bra*img2[tl+1,1,bl,rl];
image tempSum=tert((temp==0),0,1); //mask for
subtracting background and intensity correction
number h=b2-l2;
number w=r2-l2;
//border to remove aliased pixels
number bor=w;
//Delete one-pixel border
temp[0,0,h-bor,w-bor]*=tempSum[bor,bor,h,w];
temp[bor,0,h,w-bor]*=tempSum[0,bor,h-bor,w];
temp[0,bor,h,w-bor]*=tempSum[0,0,h-bor,w];
temp[bor,bor,h,w-bor]*=tempSum[0,0,h-bor,w];
img1[t2,l2,b2-1,r2-1]= temp;
//Calculate mask
tempSum=tert((temp==0),0,1);
countSum[t2,l2,b2-1,r2-1]=tempSum;
img1=tert((img1==0), 0, (img1/countSum));

return img1
} //End of Merge2

//Function Parabola
//Gives sub-pixel peak position in 2D using Kramers
rule
void Parabola(image img, number fx, number fy)
{
number maxVal=img.max(x,v);

```

```

        number a1=img.GetPixel(x,y-1);
        number a2=maxval;
        number a3=img.GetPixel(x,y+1);
        y+=0.5*(a3-a1)/(2*a2-a1-a3);
        a1=img.GetPixel(x-1,y);
        a3=img.GetPixel(x+1,y);
        x+=0.5*(a3-a1)/(2*a2-a1-a3);
    } //end of Parabola

    //Function PatternCentre
    //finds the displacement [x,y] of the pattern centre
    //from the image centre
    //by applying a 180 degree rotation and cross-
    //correlating
    void PatternCentre(image img, number &x, number &y)
    {
        image img180=img.rotate(pi);
        image imgCC=img180.CrossCorrelate(img);
        //imgCC.DisplayAt(500,50)
        imgCC.Parabola(x,y)
        number sizX,sizY
        img.GetSize(sizX,sizY);
        x-=sizX/2;
        y-=sizY/2;
    } //end of PatternCentre

    //Function symmetry matrix
    //returns the matrix relating equivalent patterns in
    //the LACBED stack
    void SymmetryMatrix(string SymType, number &m11, number
    &m12, number &m21, number &m22, string &SymName)
    {
        m11=0;
        m12=0;
        m21=0;
        m22=0;
        if (SymType=="2")
        {
            m11=-1;
        }

        m22=-1;
        SymName="2-fold";
    }
    if (SymType=="4+")
    {
        m12=-1;
        m21=1;
        SymName="4-fold(+)";
    }
    if (SymType=="4-")
    {
        m12=1;
        m21=-1;
        SymName="4-fold(-)";
    }
    if (SymType=="mx")
    {
        m11=-1;
        m22=1;
        SymName="x mirror";
    }
    if (SymType=="mx1")
    {
        m11=-1;
        m21=1;
        m22=1;
        SymName="x mirror";
    }
    if (SymType=="mx2")
    {
        m11=-1;
        m12=-1;
        m22=1;
        SymName="x mirror";
    }
    if (SymType=="my")
    {
        m11=1;
        m22=-1;
    }

    SymName="y mirror";
    }
    if (SymType=="my1")
    {
        m11=1;
        m21=-1;
        m22=-1;
        SymName="y mirror";
    }
    if (SymType=="my2")
    {
        m11=1;
        m12=1;
        m21=0;
        m22=-1;
        SymName="y mirror";
    }
    if (SymType=="mxy")
    {
        m12=-1;
        m21=-1;
        SymName="x-y mirror";
    }
    if (SymType=="myx")
    {
        m12=-1;
        m21=-1;
        SymName="y-x mirror";
    }
    }
} //end of symmetry matrix

//Function ApplySym
//Applies the symmetry operation given by SymOp to a
//square image
void ApplySym( image &img2, string SymOp)
{
    image img0=img2;
    //img2.DisplayAt(225,30);
    //img2.SetName("img2");

    //img2.SetWindowSize(200,200);
    if (SymOp=="4+")
    {
        img2=img0.rotate(pi/2);
    }
    if (SymOp=="4-")
    {
        img2=img0.rotate(-pi/2);
    }
    if (SymOp=="2")
    {
        img2=img0.rotate(pi); //no need if individual 2-folds
        have been applied
    }
    if (SymOp=="mx")
    {
        img2=img0[iwidth-icol,irow];
    }
    if (SymOp=="my")
    {
        img2=img0[icol,iheight-irow];
    }
    if (SymOp=="mxy")
    {
        img2=img0[iwidth-irow,iheight-icol];
    }
    if (SymOp=="myx")
    {
        img2=img0[irow,icol];
    }
}

//End of ApplySym

//Function SymmetryAdd
//Adds symmetrically related patterns
//requires window defined as a global variable, assumes
//a SQUARE input image [siz x siz]

```

```

void SymmetryAdd(image &Lstack, string SymOp, number
nG1, number nG2, number g1X, number g1Y, number g2X,
number g2Y, number SumNo)
{
    image Lstack2=Lstack.ImageClone();
    number data_type = Lstack.GetDataTypes();
    number siz,nPatt;
    Lstack.GetSize(siz,siz,nPatt);
    number med=round((siz/2));
    image img1=NewImage("img1",data_type,2*siz,2*siz);
    image img2=img1;
    image imgCC=img1*0;
    //img1.DisplayAt(5,30);
    //img1.SetName("img1");
    //img1.SetWindowSize(200,200);
    //img2.DisplayAt(220,30);
    //img2.SetName("img2");
    //img2.SetWindowSize(200,200);
    //imgCC.DisplayAt(440,30);
    //imgCC.SetName("imgCC");
    //imgCC.SetWindowSize(200,200);
    //Get the matrix describing equivalent patterns
    string SymName;
    number m11,m12,m21,m22
    SymOp.SymmetryMatrix(m11,m12,m21,m22,SymName);
    if (SymOp=="my1" || SymOp=="my2") SymOp="my";
    if (SymOp=="mx1" || SymOp=="mx2") SymOp="mx";
    number
ind,jnd,indS,jndS,gNoS,prog,t1,l1,b1,r1,t2,l2,b2,r2,x,y
;
    number it=0.1;
    number gNo=0;
    for (ind=-nG1; ind<nG1+1; ind++)
    {
        for (jnd=-nG2; jnd<nG2+1; jnd++)
        {
            prog=round(100*( (gNo+1)/nPatt ))

            //Apply symmetry operation to image
            img2.ApplySym(SymOp);
            imgCC=img2.CrossCorrelate(img1);
            imgCC.Parabola(x,y);
            x=-siz;
            y=-siz;
            //result("x,y:"*x*","*y*"\n");
            //use the full patterns to make the average

            img1[med,med,med+siz,med+siz]=Lstack[0,0,gNo,siz,siz,gN
ot+1];

            img2[med,med,med+siz,med+siz]=Lstack2[0,0,gNoS,siz,siz,
gNoS+1];
            //Apply symmetry operation to image
            img2.ApplySym(SymOp);
            img1=Merge2(img1,img2,x,y,SumNo);

            Lstack[0,0,gNo,siz,siz,gNo+1]=img1[med,med,med+siz,med+
siz];
            gNo++
        }
    }
}

//end of SymmetryAdd

//Function BlurG
//A Gaussian blur of radius r applied to image img
image BlurG(image img, number r)
{
    number ind,jnd;
    //get size, min and max of input image
    number siz2,siz
    img.GetSize(siz2,siz2);
    siz=round(siz2/2);
    number Rmin=img.min();
    number Rmax=img.max();
    //set up a 1D kernel of appropriate size

    number kernel=round(5*r);
    image gauss1D=NewImage("Kernel",2,1,2*kernel+1);//1
pixel wide
    //gauss1D.DisplayAt(0,30);
    //gauss1D.SetWindowSize(20,200);
    number Rsum=0;
    for (ind=-kernel; ind<kernel+1; ind++)
    {
        Igauss=exp(-(ind*ind)/(2*r*r));
        gauss1D[ind+kernel,0,ind+kernel+1,1]=Igauss;
        Rsum=Rsum+Igauss;
    }
    gauss1D=gauss1D/Rsum;//normalise
    image img0=img;//duplicate of input image, will be
returned as output
    image temp=img*0;//temporary image to apply the blur
    //img0.DisplayAt(30,30);
    //img0.SetName("img0");
    //img0.SetWindowSize(200,200);
    image shift=temp;//for edge filling
    //blur y
    for (ind=-kernel; ind<kernel; ind++)
    {
        if (ind<0)
        {
            shift[0,0,siz2+ind,siz2]=img0[-ind,0,siz2,siz2];
            for (jnd=0; jnd<-ind+1; jnd++)//edge fill on
bottom
            {
                shift[siz2-jnd-1,0,siz2-jnd,siz2]=img0[siz2-
1,0,siz2,siz2];
            }
        }
        else
        {
            shift[ind,0,siz2,siz2]=img0[0,0,siz2-ind,siz2];
            for (jnd=0; jnd<ind+1; jnd++)//edge fill on top
            {

```

```

        shift[jnd,0,jnd+1,siz2]=img0[0,0,1,siz2];
    }
    temp=temp+shift*gauss1D.getPixel(0,ind+kernel);
}
//make the y-blurred image the input for x-blur
img0=temp;
temp=0;//reset the temp image

//blur x
for (ind=-kernel; ind<kernel+1; ind++)
{
    if (ind<0)
    {
        shift[0,0,siz2,siz2+ind]=img0[0,-ind,siz2,siz2];
        for (jnd = 1; jnd<-ind+1; jnd++)//edge fill on
right
        {
            shift[0,siz2-jnd,siz2,siz2-jnd+1]=img0[0,siz2-
1,siz2,siz2];
        }
    }
    else
    {
        shift[0,ind,siz2,siz2]=img0[0,0,siz2,siz2-ind];
        for (jnd=1; jnd<ind+1; jnd++)//edge fill on left
        {
            shift[0,jnd,siz2,jnd+1]=img0[0,0,siz2,1];
        }
    }
    temp=temp+shift*gauss1D.getPixel(0,ind+kernel);
}
//set intensity range of outpt image to match that of
the input image
temp=temp*(Rmax-Rmin)/temp.max()+Rmin
//return the blurred image
return img0=temp;
}

//End of BlurG

//Function linwarp
//Applies linear distortions to a square image
image linwarp(image img, number XoX, number XoY, number
YoX, number YoY)
{
    image warped=img*0;
    number siz,med;
    img.GetSize(siz,siz);
    med=siz/2;
    warped=img[icol+(icol-med)*XoX+(irow-
med)*XoY,irow+(irow-med)*YoY+(icol-med)*YoX];
    return warped
}

//Function FindEdges
void FindEdges(image img, number &te, number &le,
number &be, number &re)
{
    //finds the edges of the LACBED data in Lstack,
te,le,be,re are global variables
    number siz;
    img.GetSize(siz,siz);
    number med=siz/2;
    number ind=0;
    number dat=0;
    // number wf=0.75;//shift of border inwards to allow
window to be used, must be >0.5
    number wf=1.5;//shift of border inwards to allow
window to be used, must be >0.5
    while (dat==0)
    {
        //count until reach a non-zero pixel
        dat=img.GetPixel(ind,med);
        ind++;
    }
    te=ind+wf*window;
    dat=0;
    ind=0;

while (dat==0)
{
    dat=img.GetPixel(med,ind);
    ind++;
}
le=ind+wf*window;
dat=0;
ind=siz-1;
while (dat==0)
{
    dat=img.GetPixel(ind,med);
    ind--;
}
be=ind-wf*window;
if ((be-med)<(med-te))
{
    //be is closer to middle, move te
    te=med-(be-med);
}
//te is closer to middle, move be
else
{
    be=med+(med-te);
}
dat=0;
ind=siz-1;
while (dat==0)
{
    dat=img.GetPixel(med,ind);
    ind--;
}
re=ind-wf*window;
if ((re-med)<(med-le))
{
    //re is closer to middle, move le
    le=siz-re;
}
//le is closer to middle, move re
else
{
    re=siz-le;
}
}
//result("tlbr:"+te+", "+le+", "+be+", "+re+"\n");

}

//end of FindEdges

//Function skewfit
//Finds best skew for a square image and a transformed
version
//XoX etc. are the best fit coefficients for linear
distortion
void skewfit(image img, string SymOp, number &XoX,
number &XoY, number &YoX, number &YoY)
{
    number siz;
    img.GetSize(siz,siz);
    number med=siz/2;
    image Dimg=img;//copy of input image (to be Deformed)
    image Timg=Dimg;
    Timg.ApplySym(SymOp);//the transformed image
    //we deform the copy of the original image to match
the transformed one.
    //the final distortion will be half of that needed to
transform one into the other
    //edges of the data inside the image
    number te,le,be,re,x,y;
    img.FindEdges(te,le,be,re);
    image Mimg=img*0;//mask to enhance fit

    //XoX skew
    Mimg[med-window/2,le-
window/2,med+window/2,le+window/2]=mask;
    Mimg[med-window/2,re-
window/2,med+window/2,re+window/2]=mask;
    image TimgM=Timg*Mimg;//masked Timg
    image DimgM=Dimg*Mimg;//masked Dimg
    DimgM.DisplayAt(630,30);
    DimgM.SetName("Deformed image");
    DimgM.SetWindowSize(300,300);
    // Timg.DisplayAt(630,330);
    // Timg.SetName("Transformed image");
    // Timg.SetWindowSize(300,300);
    //Starting point, zero skew
}

```

```

number bestfit=max(TimgM.CrossCorrelate(DimgM));
number bestYoX=YoX;
number Cshow=round(YoX*10000)/100;
OpenAndSetProgressWindow("Fitting x-skew,"+Cshow,"
",bestfit+" ");
//result("x-skew=0, "+bestfit+"\n");
number lastfit=bestfit;
number del=0.0025;//increment by 0.25%
//work out which way to go
DimgM=Dimg.linarwarp(XoX,XoY,(YoX+del),YoY)*Mimg;
number fit=max(TimgM.CrossCorrelate(DimgM));
//result("x-skew=0.0025, "+fit+"\n");
if (fit<lastfit)
{//fit is worse, go the other way
del*=-1;
}
while (lastfit>=bestfit)
{//now get approximate minimum
YoX+=del;
DimgM=Dimg.linarwarp(XoX,XoY,YoX,YoY)*Mimg;
fit=max(TimgM.CrossCorrelate(DimgM));
//result("x-skew="+YoX+", "+fit+"; last="+lastfit+";
best="+bestfit+"\n");
if (fit>=bestfit)
{
bestYoX=YoX;
bestfit=fit;
}
lastfit=fit;
Cshow=round(YoX*10000)/100;
OpenAndSetProgressWindow("Finding x-skew,"+Cshow,"
",fit+" ");
}
//3-point parabola to get best fit
del=abs(del);
DimgM=Dimg.linarwarp(XoX,XoY,bestYoX-del,YoY)*Mimg;
number a1=max(TimgM.CrossCorrelate(DimgM));
number a2=bestfit;
DimgM=Dimg.linarwarp(XoX,XoY,bestYoX+del,YoY)*Mimg;

//result("y-skew="+XoY+", "+fit+"; last="+lastfit+";
best="+bestfit+"\n");
if (fit>=bestfit)
{
bestXoY=XoY;
bestfit=fit;
}
lastfit=fit;
Cshow=round(XoY*10000)/100;
OpenAndSetProgressWindow("Finding y-skew,"+Cshow,"
",fit+" ");
}
del=abs(del);
DimgM=Dimg.linarwarp(XoX,bestXoY-del,YoX,YoY)*Mimg;
a1=max(TimgM.CrossCorrelate(DimgM));
a2=bestfit;
DimgM=Dimg.linarwarp(XoX,bestXoY+del,YoX,YoY)*Mimg;
a3=max(TimgM.CrossCorrelate(DimgM));
//result("3 x "+(bestXoY-del)+"", "+bestXoY+",
"+(bestXoY+del)+"\n");
//result("3 y "+a1+", "+a2+", "+a3+"\n");
XoY=bestXoY+del*0.5*(a3-a1)/(2*a2-a1-a3);//best skew
Cshow=round(XoY*10000)/100;
OpenAndSetProgressWindow("Best fit y-skew,"+Cshow,"
",fit+" ");
result("done\n");
DimgM.DeleteImage();
TimgM.DeleteImage();
//End of skewfit

//function rotfit
//Gives fit index for rotation of a square image by
checking it against a horizontally flipped version
number rotfit(image img, number phi)
{
number fit
image Rot000=img.rotate(phi).BlurG(2).Sobel();
image Rot2=Rot000*0;
//Rot000.DisplayAt(630,330);

number a3=max(TimgM.CrossCorrelate(DimgM));
//result("3 x "+(bestYoX-del)+"", "+bestYoX+",
"+(bestYoX+del)+"\n");
//result("3 y "+a1+", "+a2+", "+a3+"\n");
YoX=bestYoX+del*0.5*(a3-a1)/(2*a2-a1-a3);//best skew
Cshow=round(YoX*10000)/100;
OpenAndSetProgressWindow("Best fit x-skew,"+Cshow,"
",fit+" ");
}
//XoY skew
Mimg=0;
Mimg[te>window/2,med-
window/2,te+window/2,med+window/2]=mask;
Mimg[be>window/2,med-
window/2,be+window/2,med+window/2]=mask;
TimgM=Timg*Mimg;
DimgM=Dimg*Mimg;
//Starting point, zero skew
bestfit=max(TimgM.CrossCorrelate(DimgM));
number bestXoY=XoY;
Cshow=round(XoY*10000)/100;
OpenAndSetProgressWindow("Fitting y-skew,"+Cshow,"
",bestfit+" ");
//result("y-skew=0, "+bestfit+"\n");
lastfit=bestfit;
//work out which way to go
DimgM=Dimg.linarwarp(XoX,XoY+del,YoX,YoY)*Mimg;
fit=max(TimgM.CrossCorrelate(DimgM));
if (fit<lastfit)
{//fit is worse, go the other way
del*=-1;
}
while (lastfit>=bestfit)
{//now get approximate minimum
XoY+=del;
DimgM=Dimg.linarwarp(XoX,XoY,YoX,YoY)*Mimg;
fit=max(TimgM.CrossCorrelate(DimgM));

//Rot000.SetName("Rotated image");
//Rot000.SetWindowSize(300,300); image
Rot2=Rot000*0;
number Rsiz,te,le,be,re;
Rot000.GetSize(Rsiz,Rsiz);
Rot000.FindEdges(te,le,be,re);
Rot2[(Rsiz-window)/2,(le-
window/2),(Rsiz+window)/2,(le+window/2)]=mask;
Rot2[(Rsiz-window)/2,(re-
window/2),(Rsiz+window)/2,(re+window/2)]=mask;
Rot000*=Rot2;
Rot2=Rot000[iwidth-icol,irow];
image CC=Rot000.CrossCorrelate(Rot2);
fit=1-max(CC);
Rot000.DeleteImage();
return fit
} //end of rotfit

//Function median
//gives the median value of an image
number median(image img)
{
number sizX,sizY
img.GetSize(sizX,sizY);
number nPix=sizX*sizY;
number odd=round(nPix/2);
number midpoint=round((1+(nPix-odd)/2)); //e.g. gives 6
if nPix=10 or 11
image sorted = NewImage("list",2,1,midpoint)
//fill up the list to the midpoint
number ind,x,y;
for (ind=0; ind<midpoint; ind++)
{
sorted[ind,0,ind+1,1]=img.min(x,y);
img.SetPixel(x,y,img.max());
}
number ibar
if(odd)
{

```

```

        ibar=sorted.getPixel(0, (midpoint-1))
    }
    else
    {
        ibar=(sorted.getPixel(0, (midpoint-1))+sorted.getPixel(0, (midpoint-2)))/2;
    }
    return ibar
} //End of median

//Function ReCentre
//puts the centre of the 000 image in the middle again
void ReCentre(image &Lstack)
{
    OpenAndSetProgressWindow("Re-centring", "", "");
    number siz, nPatt, x, y;
    Lstack.Get3DSize(siz, siz, nPatt);
    number med=siz/2;
    number N000=round((nPatt-1)/2);
    image L000=Lstack[0, 0, N000, siz, siz, N000+1]*0;
    number t1=(siz-window)/2;
    number l1=(siz-window)/2;
    number b1=t1+window;
    number r1=l1+window;

L000[t1, l1, b1, r1]=Lstack[l1, t1, N000, r1, b1, N000+1]*mask;
L000.PatternCentre(x, y);
//roi in instack
    number height=round(siz-abs(y));
    if ((height % 2)==1) height-=1//make height even if
it's odd
    number width=round(siz-abs(x));
    if ((width % 2)==1) width-=1//make width even if it's
odd
    t1=0*(y>0)+(siz-height)*(y<=0);
    l1=0*(x>0)+(siz-width)*(x<=0);
    b1=height*(y>0)+siz*(y<=0);
    r1=width*(x>0)+siz*(x<=0);
    //roi in Lstack

//Lstack will be square, [siz x siz], to contain the
averaged patterns
image Lstack := NewImage("Averaged
Stack", data_type, siz, siz, nPatt);
Lstack.DisplayAt(5, 30);
Lstack.SetWindowSize(600, 600);
number med=round((siz/2)); //a useful number
//remove negative pixels
L_Instack=tert((L_Instack<0), 0, L_Instack);

//get image tags
number
Rr, tInc, CamL, mag, Alpha, spot, nG1, nG2, g1X, g1Y, g2X, g2Y, g1H,
g1K, g1L, g2H, g2K, g2L, g1Ag2, g1Mg2, gC;
string material;
{
    L_Instack.GetStringNote("Info:Date", datetime);
    L_Instack.GetNumberNote("Info:Camera Length", CamL);
    L_Instack.GetNumberNote("Info:Magnification", mag);
    L_Instack.GetNumberNote("Info:Alpha", Alpha);
    L_Instack.GetNumberNote("Info:Spot size", spot);
    L_Instack.GetNumberNote("Info:Disc Radius", Rr);
    L_Instack.GetStringNote("Info:Material", material);
    L_Instack.GetNumberNote("g-vectors:nG1", nG1);
    L_Instack.GetNumberNote("g-vectors:nG2", nG2);
    L_Instack.GetNumberNote("g-vectors:g1X", g1X);
    L_Instack.GetNumberNote("g-vectors:g1Y", g1Y);
    L_Instack.GetNumberNote("g-vectors:g2X", g2X);
    L_Instack.GetNumberNote("g-vectors:g2Y", g2Y);
    L_Instack.GetNumberNote("g-vectors:g1H", g1H);
    L_Instack.GetNumberNote("g-vectors:g1K", g1K);
    L_Instack.GetNumberNote("g-vectors:g1L", g1L);
    L_Instack.GetNumberNote("g-vectors:g2H", g2H);
    L_Instack.GetNumberNote("g-vectors:g2K", g2K);
    L_Instack.GetNumberNote("g-vectors:g2L", g2L);
    L_Instack.GetNumberNote("g-vectors:g1Ag2", g1Ag2);
    L_Instack.GetNumberNote("g-vectors:g1Mg2", g1Mg2);
    L_Instack.GetNumberNote("g-vectors:gC", gC);
    if (material=="")

        number t2=abs(round(med-height/3));
        number l2=abs(round(med-width/2));
        number b2=t2+height;
        number r2=l2+width;
        image Ltemp=Lstack;

Lstack[l2, t2, 0, r2, b2, nPatt]=Ltemp[l1, t1, 0, r1, b1, nPatt];
}

//*****//
//*****//
// Main program
//*****//
result("\nD-ED Align v1.2.9\n")

number f ;
string date ;
GetDate(f, date);
string time ;
GetTime(f, time);
string datetime=date + " "+time ;
result("\nStarting processing "+datetime+"\n")
OpenAndSetProgressWindow("Starting processing", "", "");

//*****//
// Get 3D data stack
image L_Instack := GetFrontImage();
number sizX, sizY, nPatt, siz;
L_Instack.Get3DSize(sizX, sizY, nPatt);
//check for image type
if (nPatt==0) throw("Exiting: input should be a 3D
LACBED stack");
number data_type = L_Instack.GetDataType();
//result("data type="+data_type+"\n");
siz=round(1.4*max(sizX, sizY)); //1.4 is big enough to
accomodate a 45 deg rotation
if ((siz % 2)==1) siz+=1//make siz even if it's odd

{
    if (!GetString("Material?", material, material));
    exit(0);
    L_Instack.SetStringNote("Info:Material", material)
}
result("Material is "+material+"\n");
//g-vector outputs
number g1mag=sqrt(g1X*g1X+g1Y*g1Y);
number g2mag=sqrt(g2X*g2X+g2Y*g2Y);
number dot=(g1X*g2X+g1Y*g2Y)/(g1mag*g2mag); //gives
cos(theta)
number cross=(g1X*g2Y-g1Y*g2X)/(g1mag*g2mag); //g1 x g2
gives sin(theta)
number theta=(acos(dot)*sgn(cross));
result("g1: "+g1H+", "+g1K+", "+g1L+", +/-"+nG1+"";
["+g1X+", "+g1Y+", magnitude "+g1mag+"\n"];
result("g2: "+g2H+", "+g2K+", "+g2L+", +/-"+nG2+"";
["+g2X+", "+g2Y+", magnitude "+g2mag+"\n"];
//result("Nominal ratio of magnitudes g1/g2 =
"+g1Mg2+"\n");
//result("Actual ratio of magnitudes g1/g2 =
"+(g1mag/g2mag)+"\n");
//result("Nominal angle between g1 and g2 ="+g1Ag2+"
degrees\n");
//result("Actual angle between g1 and g2
="+((180*theta/pi)+" degrees\n");

//*****//
//Get cropped stack radius
number w=200;
if (!GetNumber("Cropped stack radius?", w, w)) exit(0);
result("Cropped stack size "+(2*w)+"x"+(2*w)+"\n");

//*****//
//montage
number montage=0;
if (TwoButtonDialog("Make montage?", "Yes", "No") )
{

```

```

        result("Will make montage\n");
        montage=1;
    }
    else
    {
        result("Will not make montage\n");
        montage=0;
    }

    ////////////
    //Get symmetry
    string symmetry;
    if (!GetString("Rotational symmetry?", "1", symmetry))
    exit(0);
    result("Applying rotational symmetry "+symmetry+"\n");

    ////////////
    //use 000 image to find centre and set up averaged
    stack
    number N000=round((nPatt-1)/2);
    number x,y,x0,y0;//cross correlation peak for 000
    image avl000=L_Instack[0,0,N000,sizX,sizY,N000+1];
    //avl000.DisplayAt(0,30);
    //avl000.SetName("Current pattern");
    //avl000.SetWindowSize(200,200);
    image maL000=avl000*0;
    maL000=exp(-(i-radius/(0.25*sizX))**2);
    //maL000=exp(-(i-col-329)/(0.15*sizX))**2)*exp(-(i-row-
    338)/(0.15*sizX))**2);
    avl000*=maL000;
    //sleep(5);
    avl000.PatternCentre(x,y);
    avl000.merge2(avl000.rotate(pi),x,y,1);
    x0=x;//we can only expect a rough estimate without
    using a window
    y0=y;
    //put central part of 000 pattern into Lstack
    number t1=round((siz-window)/2);
    number l1=round((siz-window)/2);

    number b1=t1+window;
    number r1=l1+window;
    number t2=round((sizY-y-window)/2);
    number l2=round((sizX-x-window)/2);
    number b2=t2+window;
    number r2=l2+window;
    Lstack[l1,t1,N000,r1,b1,N000+1]=avl000[t2,l2,b2,r2]*mas
    k;
    avl000.DeleteImage();
    maL000.DeleteImage();
    //fine correction using windowed pattern
    Lstack[0,0,N000,siz,siz,N000+1].PatternCentre(x,y);
    x0+=x;//correct [x0,y0]
    y0+=y;
    result("Whole pattern centre displaced by
    [ "+(x0/2)+", "+(y0/2)+" ]\n");
    //roi in instack
    number height=round((sizY-abs(y0)));
    number width=round((sizX-abs(x0)));
    t1=0*(y0>0)+(sizY-height)*(y0<=0);
    l1=0*(x0>0)+(sizX-width)*(x0<=0);
    b1=height*(y0>0)+sizY*(y0<=0);
    r1=width*(x0>0)+sizX*(x0<=0);
    //roi in Lstack
    t2=round((med-height)/2);
    l2=round((med-width)/2);
    b2=t2+height;
    r2=l2+width;
    //Copy instack across to Lstack
    Lstack[l2,t2,0,r2,b2,Npatt]=L_Instack[l1,t1,0,r1,b1,Npa
    tt];
    //from now on everything is done with Lstack, which is
    large, square and has 000 centred
    image Limg=NewImage("Current
    pattern",data_type,siz,siz);
    //Limg.DisplayAt(0,30);
    //Limg.SetName("Current pattern");
    //Limg.SetWindowSize(200,200);

    ////////////
    //Measure all the pattern centres in Lstack
    result("Measuring pattern centres...");
    number n1=2*nG1+1;
    number n2=2*nG2+1;
    image PattCen=NewImage("Centres",data_type,n1,n2,2);
    //PattCen.DisplayAt(30,30);
    //PattCen.SetName("Centres");
    //PattCen.SetWindowSize(200,200);
    number ind,jnd,prog,gNo;
    gNo=0;
    for (ind=-nG1; ind<nG1+1; ind++)
    {
        for (jnd=-nG2; jnd<nG2+1; jnd++)
        {
            prog=round(100*( gNo/Npatt ))
            OpenAndSetProgressWindow("Measuring pattern
            centres","Image "+(gNo+1)+" of "+Npatt," "+prog+" %");
            t1=round((ind*g1Y+jnd*g2Y+siz-window)/2);//window
            around pattern centre
            l1=round((ind*g1X+jnd*g2X+siz-window)/2);
            b1=t1+window;
            r1=l1+window;
            Limg=J;

            Limg[t1,l1,b1,r1]=Lstack[l1,t1,gNo,r1,b1,gNo+1]*mask;
            Limg.PatternCentre(x,y);

            PattCen[(ind+nG1),(jnd+nG2),0,(ind+nG1+1),(jnd+nG2+1),1
            ]=x;
            PattCen[(ind+nG1),(jnd+nG2),1,(ind+nG1+1),(jnd+nG2+1),2
            ]=y;
            gNo++;
        }
    }
    result("done\n");
    //update g-vectors

    result("\n");
    //median is probably better (less sensitive to bad
    measurements)
    //alternative using mean
    g1X=mean(PattCen[0,0,0,(n1-1),n2,1]-
    PattCen[1,0,0,n1,n2,1]);
    g2X=mean(PattCen[0,0,0,n1,(n2-1),1]-
    PattCen[0,1,0,n1,n2,1]);
    g1Y=mean(PattCen[0,0,1,(n1-1),n2,2]-
    PattCen[1,0,1,n1,n2,2]);
    g2Y=mean(PattCen[0,0,1,n1,(n2-1),2]-
    PattCen[0,1,1,n1,n2,2]);
    result("Mean g1: [ "+g1X+", "+g1Y+" ]\n");
    result("Mean g2: [ "+g2X+", "+g2Y+" ]\n");
    g1X=median(PattCen[0,0,0,(n1-1),n2,1]-
    PattCen[1,0,0,n1,n2,1]);
    g2X=median(PattCen[0,0,0,n1,(n2-1),1]-
    PattCen[0,1,0,n1,n2,1]);
    g1Y=median(PattCen[0,0,1,(n1-1),n2,2]-
    PattCen[1,0,1,n1,n2,2]);
    g2Y=median(PattCen[0,0,1,n1,(n2-1),2]-
    PattCen[0,1,1,n1,n2,2]);
    g1mag=sqrt(g1X*g1X+g1Y*g1Y);
    g2mag=sqrt(g2X*g2X+g2Y*g2Y);
    dot=(g1X*g2X+g1Y*g2Y)/(g1mag*g2mag);//gives cos(theta)
    cross=(g1X*g2Y-g1Y*g2X)/(g1mag*g2mag);//g1 x g2 gives
    sin(theta)
    theta=(acos(dot)*sgn(cross));
    result("Median g1: [ "+g1X+", "+g1Y+" ]\n");
    result("Median g2: [ "+g2X+", "+g2Y+" ]\n");
    result("Nominal ratio of magnitudes g1/g2 =
    "+g1Mag/g2Mag+"\n");
    result("Actual ratio of magnitudes g1/g2 =
    "+(g1mag/g2mag)+"\n");
    result("Nominal angle between g1 and g2 ="+"g1Ag2+"
    degrees\n");
    result("Actual angle between g1 and g2
    ="+"(180*theta/pi)+" degrees\n");
    }

```



```

//////////*****
//Averaging using user-input symmetry
//to keep track of the number of images that have been
added
number SumNo=1;

//Apply individual 2-fold pattern symmetries and put
into Lstack
number XoX,XoY,YoX,YoY;
image L000=Lstack[0,0,N000,siz,siz,N000+1];
image Dimg=L000;
if (!(symmetry=="0"))
{
    result("    2-fold pattern averaging...");
    gNo=0;
    for (ind=-nG1; ind<nG1+1; ind++)
    {
        for (jnd=-nG2; jnd<nG2+1; jnd++)
        {
            prog=round(100*( (gNo+1)/Npatt ));
            OpenAndSetProgressWindow("2-fold pattern
            averaging", "Image "+(gNo+1)+" of "+Npatt, " "+prog+"
            %");
            t1=round((ind*g1Y+jnd*g2Y+ siz-window)/2); //window
            around pattern centre
            l1=round((ind*g1X+jnd*g2X+ siz-window)/2);
            b1=t1>window;
            r1=l1>window;
            Limg=0;
            Limg[t1,l1,b1,r1]=Lstack[l1,t1,gNo,r1,b1,gNo+1];
            Limg.PatternCentre(x,y);
            //result("x,y: "+x+", "+y+"\n");
            //use the full patterns to make the average
            Limg=Lstack[0,0,gNo,siz,siz,gNo+1];
            Limg=Merge2(Limg, (Limg.rotate(p1)), x, y, SumNo);
            Lstack[0,0,gNo,siz,siz,gNo+1]=Limg;
            gNo++;
        }
    }

    if (gC==2)
    {
        result("g2 is a face-centring vector: centred lattice
        is\n");
        h2X=g2X-g1X;
        h2Y=g2Y-g1Y;
        h2mag=sqrt(h2X*h2X+h2Y*h2Y);
        h2H=round(2*g2H-g1H);
        h2K=round(2*g2K-g1K);
        h2L=round(2*g2L-g1L);
        h1Ah2=90; //centred patterns are always rectangular
        h1Mh2=g1Mg2/(2*sin(g1Ag2*pi/180));
    }
    if (abs(h1X)<tiny) h1X=0;
    if (abs(h1Y)<tiny) h1Y=0;
    if (abs(h2X)<tiny) h2X=0;
    if (abs(h2Y)<tiny) h2Y=0;
    result("h1: "+h1H+", "+h1K+", "+h1L+" :
    ["+h1X+", "+h1Y+", "magnitude "+h1mag+"\n"];
    result("h2: "+h2H+", "+h2K+", "+h2L+" :
    ["+h2X+", "+h2Y+", "magnitude "+h2mag+"\n"];
    result("      \n");

    ////////////
    //Rotate stack to put h1 horizontal
    if (!(symmetry=="0"))
    {
        //Calculate angles and new g-vectors
        //Angle phi between h1 and the x-axis
        number dotX=h1X/h1mag; //gives cos(phi)
        number crossX=-h1Y/h1mag; //gl x [100] gives sin(phi)
        number phi=acos(dotX)*sgn(crossX);
        //Use the 000 image to get an accurate rotation angle
        number dphi=pi/720; //quarter degree angle increment
        //Get initial fit at the nominal rotation angle
        number
        bestfit=Lstack[0,0,N000,siz,siz,N000+1].rofit(phi+tiny);
        number bestphi=phi;

        OpenAndSetProgressWindow("Measuring rotation
        angle", ""+(phi*180/pi), ""+bestfit);
        //result("initial angle="+ (phi*180/pi) +",
        "+bestfit+"\n");
        number lastfit=bestfit;
        //Get fit at incremented angle
        number
        fit=Lstack[0,0,N000,siz,siz,N000+1].rofit((phi+dphi+ti
        ny));
        //result("angle= "+ ((phi+dphi)*180/pi) +",
        "+fit+"\n");
        OpenAndSetProgressWindow("Measuring rotation
        angle", ""+(phi+dphi*180/pi), ""+fit);
        //work out which way to go
        if (fit>lastfit)
        { //fit is worse, go the other way
            dphi*=-1;
        }
        while (lastfit<=bestfit)
        { //now get approximate minimum
            phi+=dphi; //increment angle
            lastfit=fit;
            fit=Lstack[0,0,N000,siz,siz,N000+1].rofit(phi+tiny);
            //result((phi*180/pi) +", "+fit+"\n");
            OpenAndSetProgressWindow("Measuring rotation
            angle", ""+(phi*180/pi), ""+fit);
            if (fit<=bestfit)
            {
                bestphi=phi;
                bestfit=fit;
            }
        }
        //now do a 3-point parabola to get an interpolated
        fit
        dphi=abs(dphi);
        number
        a1=Lstack[0,0,N000,siz,siz,N000+1].rofit(bestphi-
        dphi+tiny);
    }

    }
    //SumNo++
    result("done\n");
}

//////////
//centred lattice: default is that vectors h1 and h2
are the same as g1 and g2
number h1X=g1X;
number h1Y=g1Y;
number h1H=g1H;
number h1K=g1K;
number h1L=g1L;
number h1mag=g1mag;
number h2X=g2X;
number h2Y=g2Y;
number h2H=g2H;
number h2K=g2K;
number h2L=g2L;
number h2mag=g2mag;
number h1Ah2=g1Ag2;
number h1Mh2=g1Mg2;
if (gC==0) result("The pattern is not face-
centred\n"); //and so h=g
if (gC==1)
{ //set up new vectors h1 and h2 to describe the centred
pattern
    result("g1 is a face-centring vector: centred lattice
    is\n");
    h1X=g1X-g2X;
    h1Y=g1Y-g2Y;
    h1mag=sqrt(h1X*h1X+h1Y*h1Y);
    h1H=round(2*g1H-g2H); //relying on the user here to
    have put in indices that work
    h1K=round(2*g1K-g2K);
    h1L=round(2*g1L-g2L);
    h1Ah2=90; //centred patterns are always rectangular
    h1Mh2=g1Mg2*2*sin(g1Ag2*pi/180);
}

}

```

```

        number a2=bestfit;
        number
a3=Lstack[0,0,N000,siz,siz,N000+1].rotfit(bestphi+dphi+
tiny);
        phi=bestphi+dphi*0.5*(a3-a1)/(2*a2-a1-a3);
        OpenAndSetProgressWindow("Rotating by", (phi*180/pi)+"
degrees", "+fit");
        result("Rotating by "+(phi*180/pi)+" degrees\n");
        image Lcorrected=Lstack.rotate(phi);
        number sizXr,sizYr;
        Lcorrected.GetSize(sizXr,sizYr,nPatt);
        //put it back into Lstack
        t1=round((sizYr-siz)/2);
        l1=round((sizXr-siz)/2);
        b1=t1+siz;
        r1=l1+siz;

Lstack[0,0,0,siz,siz,nPatt]=Lcorrected[l1,t1,0,r1,b1,nP
att];
        //rotated vectors
        number glXr=glX*cos(phi)+glY*sin(phi);
        number glYr=glY*cos(phi)-glX*sin(phi);
        number g2Xr=g2X*cos(phi)+g2Y*sin(phi);
        number g2Yr=g2Y*cos(phi)-g2X*sin(phi);
        number h1Xr=h1X*cos(phi)+h1Y*sin(phi);
        number h1Yr=; //by definition
        number h2Xr=h2X*cos(phi)+h2Y*sin(phi);
        number h2Yr=h2Y*cos(phi)-h2X*sin(phi);
        glX=glXr;
        glY=glYr;
        h1X=h1Xr;
        h1Y=h1Yr;
        g2X=g2Xr;
        g2Y=g2Yr;
        h2X=h2Xr;
        h2Y=h2Yr;
        //re-centre
        Lstack.ReCentre();

        L000=L000.Blur(2).Sobel();
        L000.skewfit(SymOp,XoX,XoY,YoX,YoY);
        result(" Measured skew =
"+(round(XoY*10000)/100)+"%\n");
        //result("dx="+((round(XoX*10000)/100)+"% +
"+(round(XoY*10000)/100)+"%\n");
        //result("dy="+((round(YoX*10000)/100)+"% +
"+(round(YoY*10000)/100)+"%\n");
        //Take the average
        XoY=(XoY+XoY1)/2;
        YoX=(YoX+YoX1)/2;
        //Measure y-stretch from xy-mirror or 4-fold rotation
        (3-fold and 6-fold to be added)
        if (Msymmetry=="mxy" || symmetry=="4")
        {
            image mask1:=RealImage("Mask",4,2*w>window); //short
wide mask
            mask1=exp(-(irow-window/2)/(0.3*window)**2)*exp(-
((icol-w)/(0.5*w))**2);
            //we deform the copy of the original image to match
the transformed one.
            L000=Lstack[0,0,N000,siz,siz,N000+1];
            L000=L000.Blur(2).Sobel();
            image Mimg=L000*0.5; //top & bottom masks to enhance
fit
            Mimg[(med-w),(med-w),(med-w+window),(med+w)]=mask1;
            Mimg[(med+w>window),(med-w),(med+w),(med+w)]=mask1;
            image DimgM=L000*Mimg; //masked input image (to be
Deformed)
            DimgM.DisplayAt(630,30);
            DimgM.SetName("Deformed image");
            DimgM.SetWindowSize(300,300);
            if (Msymmetry=="mxy") SymOp="mxy";
            if (symmetry=="4") SymOp="4+";
            L000.ApplySym(SymOp)
            image TimgM=L000*Mimg; //transformed masked
reference image
            number bestfit=max(TimgM.CrossCorrelate(DimgM));
            //result("y-stretch=0.0000, "+bestfit+"\n");

        }

        //////////////////////////////////////////////////
        //Apply mirrors
        XoX=0; //start off with no distortion
        XoY=0;
        YoX=0;
        YoY=0;
        if (!(symmetry=="0"))
        {
            string Msymmetry,SymOp;
            if (!GetString("mirror symmetry?", "mx", Msymmetry))
            exit(0);
            result(" Applying mirror symmetry "+Msymmetry+"\n");
            //first measure distortions, skews from x-mirror
            if (Msymmetry=="mx" || Msymmetry=="mxy" ||
Msymmetry=="mm")
            {
                result("Measuring distortions, x-mirror... ");
                SymOp="mx";
                L000=Lstack[0,0,N000,siz,siz,N000+1];
                L000=L000.Blur(2).Sobel();
                L000.skewfit(SymOp,XoX,XoY,YoX,YoY);
                result(" Measured skew =
"+(round(XoY*10000)/100)+"%\n");
                //result("dx="+((round(XoX*10000)/100)+"% +
"+(round(XoY*10000)/100)+"%\n");
                //result("dy="+((round(YoX*10000)/100)+"% +
"+(round(YoY*10000)/100)+"%\n");
            }
            //store first set of measurements
            number XoY1=XoY;
            number YoX1=YoX
            //measure skews from y-mirror, should give the same
            result(" Measuring distortions, y-mirror...");
            string SymOp="my";
            L000=Lstack[0,0,N000,siz,siz,N000+1];

            number lastfit=bestfit;
            number del=0.0025; //Increment by 0.25%
            number bestYoY=YoY;
            number Cshow=round(YoY*10000)/100;
            OpenAndSetProgressWindow("Finding y-
stretch, "+Cshow+" ", bestfit+"\n");
            //work out which way to go
            DimgM=(L000.linwrap(XoX,XoY,YoX,YoY+del))*Mimg;
            number fit=max(TimgM.CrossCorrelate(DimgM));
            //result("y-stretch=0.0025, "+fit+"\n");
            if (fit<lastfit)
            { //fit is worse, go the other way
                del*=-1;
            }
            while (lastfit>=bestfit)
            { //now get approximate minimum
                YoY+=del;
                DimgM=(L000.linwrap(XoX,XoY,YoX,YoY))*Mimg;
                fit=max(TimgM.CrossCorrelate(DimgM));
                //result("y-stretch="+YoY+" ", "+fit+");
                last="+lastfit+"; best="+bestfit+"\n");
                if (fit>=bestfit)
                {
                    bestYoY=YoY;
                    bestfit=fit;
                }
                lastfit=fit;
                Cshow=round(YoY*10000)/100;
                OpenAndSetProgressWindow("Finding y-
stretch, "+Cshow+" ", fit+"\n");
            }
            //3-point parabola to get best fit
            del=abs(del);
            DimgM=L000.linwrap(XoX,XoY,YoX,bestYoY-del)*Mimg;
            number a1=max(TimgM.CrossCorrelate(DimgM));
            number a2=bestfit;
            DimgM=L000.linwrap(XoX,XoY,YoX,bestYoY+del)*Mimg;
            number a3=max(TimgM.CrossCorrelate(DimgM));

```

```

//result("3 x "+(bestYoY-del)+"", "(bestYoY+",
"+(bestYoY+del)+"\n")
//result("3 y "+a1+", "+a2+", "+a3+"\n")
YoY=bestYoY+del*0.5*(a3-a1)/(2*a2-a1-a3); //best
stretch
result(" Measured y-
stretch="+round(YoY*10000)/100)+"%\n");
Cshow=round(YoY*10000)/100;
OpenAndSetProgressWindow("Best fit y-
stretch, ", Cshow+" ", fit+" ");
DimgM.DeleteImage();
TimgM.DeleteImage();
}
else
{
result("Not enough symmetry to determine y-stretch,
ensure that you run B-ED_Stretch to match the
simulation later\n")
}
result("Distortion applied:\n")
result("dx="+round(XoX*10000)/100)+"x% +
"+round(XoY*10000)/100)+"y%\n");
result("dy="+round(YoX*10000)/100)+"x% +
"+round(YoY*10000)/100)+"y%\n");
//Correct distortions
gNo=0;
for (ind=-nG1; ind<nG1+1; ind++)
{
for (jnd=-nG2; jnd<nG2+1; jnd++)
{
prog=round(100*( gNo+1)/Npatt ))
OpenAndSetProgressWindow("Correcting
distortions", "Image "+(gNo+1)+" of "+Npatt, " "+prog+"
%");
L000=Lstack[0,0,gNo,siz,siz,gNo+1];
Limg=L000.linwrap(XoX,XoY/2,YoX/2,YoY);
Lstack[0,0,gNo,siz,siz,gNo+1]=Limg;
gNo++
}
}

if (gC==0)
SymmetryAdd(Lstack, "mxy", nG1, nG2, g1X, g1Y, g2X, g2Y, SumNo)
;
if (gC==1) result("xy-mirror symmetry incompatible
with centred lattice, redefine basis vectors\n");
SumNo++;
result("done\n");
Lstack.ReCentre();
}
//other mirrors for 3m and 6mm needed here
}

//Averaging using rotational symmetry
//number m1,m2,m21,m22;
string SymOp;
//2-fold symmetry
if (symmetry=="2" || symmetry=="4" || symmetry=="6")
{
result(" 2-fold symmetry averaging...");
SymmetryAdd(Lstack, "2", nG1, nG2, g1X, g1Y, g2X, g2Y, SumNo);
SumNo++;
result("done\n");
}

//4+ symmetry
if (symmetry=="4")
//NB do not need 4(-) averaging if we have already
done a 2-fold
SumNo=1
result(" 4-fold symmetry averaging...");

SymmetryAdd(Lstack, "4+", nG1, nG2, g1X, g1Y, g2X, g2Y, SumNo);
SumNo++;
result("done\n");
//re-centre
Lstack.ReCentre();
}
//3 and 6 fold needed here!

}
}
//mirror symmetry averaging
if (Msymmetry=="mx" || Msymmetry=="mxy" ||
Msymmetry=="mm")
{
result(" x-mirror symmetry averaging...");
if (gC==0)
SymmetryAdd(Lstack, "mx", nG1, nG2, g1X, g1Y, g2X, g2Y, SumNo);
if (gC==1)
SymmetryAdd(Lstack, "mx1", nG1, nG2, g1X, g1Y, g2X, g2Y, SumNo)
;
if (gC==2)
SymmetryAdd(Lstack, "mx2", nG1, nG2, g1X, g1Y, g2X, g2Y, SumNo)
;
SumNo++;
result("done\n");
Lstack.ReCentre();
}
if (Msymmetry=="my" || Msymmetry=="mxy" ||
Msymmetry=="mm")
{
result(" y-mirror symmetry averaging...");
if (gC==0)
SymmetryAdd(Lstack, "my", nG1, nG2, g1X, g1Y, g2X, g2Y, SumNo);
if (gC==1)
SymmetryAdd(Lstack, "my1", nG1, nG2, g1X, g1Y, g2X, g2Y, SumNo)
;
if (gC==2)
SymmetryAdd(Lstack, "my2", nG1, nG2, g1X, g1Y, g2X, g2Y, SumNo)
;
SumNo++;
result("done\n");
Lstack.ReCentre();
}
if (Msymmetry=="mxy")
{
result(" xy-mirror symmetry averaging...");//NB
this can only be present in 4mm symmetry
}
}

//////////////////
//Measure all the pattern centres in Lstack
result("Measuring pattern centres...");
n1=2*nG1+1;
n2=2*nG2+1;
gNo=0;
for (ind=-nG1; ind<nG1+1; ind++)
{
for (jnd=-nG2; jnd<nG2+1; jnd++)
{
prog=round(100*( gNo/Npatt ))
OpenAndSetProgressWindow("Measuring pattern
centres", "Image "+(gNo+1)+" of "+Npatt, " "+prog+" %");
t1=round((ind*g1Y+jnd*g2Y+ siz-window)/2); //window
around pattern centre
l1=round((ind*g1X+jnd*g2X+ siz-window)/2);
b1=t1+window;
r1=l1+window;
Limg=0;
Limg[t1,l1,b1,r1]=Lstack[l1,t1,gNo,r1,b1,gNo+1]*mask;
Limg.PatternCentre(x,y);

PattCen[(ind+nG1), (jnd+nG2), 0, (ind+nG1+1), (jnd+nG2+1), 1
]=x;

PattCen[(ind+nG1), (jnd+nG2), 1, (ind+nG1+1), (jnd+nG2+1), 2
]=y;
gNo++
}
}
result("done\n");
//update g-vectors
result("_____ \n");
//median is probably better (less sensitive to bad
measurements)
//alternative using mean

```

```

g1X=mean(PattCen[0,0,0,(n1-1),n2,1]-
PattCen[1,0,0,n1,n2,1]);
g2X=mean(PattCen[0,0,0,n1,(n2-1),1]-
PattCen[0,1,0,n1,n2,1]);
g1Y=mean(PattCen[0,0,1,(n1-1),n2,2]-
PattCen[1,0,1,n1,n2,2]);
g2Y=mean(PattCen[0,0,1,n1,(n2-1),2]-
PattCen[0,1,1,n1,n2,2]);
result("Mean g1: [" +g1X+" "+g1Y+""]\n");
result("Mean g2: [" +g2X+" "+g2Y+""]\n");
g1X=median(PattCen[0,0,0,(n1-1),n2,1]-
PattCen[1,0,0,n1,n2,1]);
g2X=median(PattCen[0,0,0,n1,(n2-1),1]-
PattCen[0,1,0,n1,n2,1]);
g1Y=median(PattCen[0,0,1,(n1-1),n2,2]-
PattCen[1,0,1,n1,n2,2]);
g2Y=median(PattCen[0,0,1,n1,(n2-1),2]-
PattCen[0,1,1,n1,n2,2]);
g1mag=sqrt(g1X*g1X+g1Y*g1Y);
g2mag=sqrt(g2X*g2X+g2Y*g2Y);
dot=(g1X*g2X+g1Y*g2Y)/(g1mag*g2mag); //gives cos(theta)
cross=(g1X*g2Y-g1Y*g2X)/(g1mag*g2mag); //g1 x g2 gives
sin(theta)
theta=(acos(dot))*sgn(cross));
result("Median g1: [" +g1X+" "+g1Y+""]\n");
result("Median g2: [" +g2X+" "+g2Y+""]\n");
result("Nominal ratio of magnitudes g1/g2 =
"+g1Mg2+"\n");
result("Actual ratio of magnitudes g1/g2 =
"+(g1mag/g2mag)+"\n");
result("Nominal angle between g1 and g2 =" +g1Ag2+"
degrees\n");
result("Actual angle between g1 and g2
="+ (180*theta/pi)+" degrees\n");
}

//additional rotation
//{
//image Lrotated2=Lstack.rotate(45*pi/180);

//number sizXr,sizYr;
//Lrotated2.GetSize(sizXr,sizYr,nPatt);
//t1=round((sizYr-siz)/2);
//l1=round((sizXr-siz)/2);
//b1=t1+siz;
//r1=l1+siz;
//Lstack[0,0,0,siz,siz,nPatt]=Lrotated2[l1,t1,0,r1,b1,n
Patt];
//Lrotated2.DeleteImage();
//}

//////////
//cropped stack
result("Cropping to [" +(2*w)+"x" +(2*w)+""]\n");
t1=med-w;
l1=med-w;
b1=med+w;
r1=med+w;
image LACBED_reduced_stack=Lstack[l1,t1,0,r1,b1,nPatt];
LACBED_reduced_stack.SetName("Cropped Stack");
LACBED_reduced_stack.DisplayAt(5,30);
LACBED_reduced_stack.SetWindowSize(200,200);
LACBED_reduced_stack.SetLimits(LACBED_reduced_stack.min
(),LACBED_reduced_stack.max())
result("Convergence angle =
"+(w/min(g1mag,g2mag))+ "\n");

//////////
//montage
if (montage==1)
{
    image
    Dmontage=NewImage("Montage",data_type,(n1*2*w),(n2*2*w)
);
    image
    temp=LACBED_reduced_stack[0,0,0,2*w,2*w,1]; //temp image
    for normalising
    number back=5; //a background to subtract, if desired
    Dmontage.DisplayAt(225,30);

Dmontage.SetWindowSize(500,500);
Dmontage.SetName("Montage");
gNo=nPatt-1;
for (ind=-nG1; ind<nG1+1; ind++)
{
    for (jnd=-nG2; jnd<nG2+1; jnd++)
    {
        prog=round(100*(1-(gNo+1)/NPatt));
        OpenAndSetProgressWindow("Making montage","Image
"+(gNo+1)+" of "+NPatt," "+prog+" %");
        t1=round((jnd+nG2)*2*w);
        l1=round((ind+nG1)*2*w);
        b1=t1+2*w;
        r1=l1+2*w;
        //result("t1br:" +t1+" "+l1+" "+b1+" "+r1+"\n");
        temp=LACBED_reduced_stack[0,0,gNo,2*w,2*w,gNo+1]-
back;
        temp=tert((temp<0),0,temp);
        temp=-temp.min();
        temp/=temp.max();
        Dmontage[t1,l1,b1,r1]=temp;
        gNo--
    }
}
Dmontage.SetLimits(Dmontage.min(),Dmontage.max())
result("done\n");
}

//set image tags
{
    LACBED_reduced_stack.SetStringNote("Info:Date",datetime
);
    LACBED_reduced_stack.SetNumberNote("Info:Camera
Length",CamL);
    LACBED_reduced_stack.SetNumberNote("Info:Magnification"
,mag)
    LACBED_reduced_stack.SetNumberNote("Info:Alpha",Alpha);

LACBED_reduced_stack.SetNumberNote("Info:Spot
size",spot);
LACBED_reduced_stack.SetNumberNote("Info:Disc
Radius",Rr);
LACBED_reduced_stack.SetStringNote("Info:Material",mate
rial);
LACBED_reduced_stack.SetNumberNote("g-
vectors:nG1",nG1);
LACBED_reduced_stack.SetNumberNote("g-
vectors:nG2",nG2);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g1X",g1X);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g1Y",g1Y);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g2X",g2X);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g2Y",g2Y);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g1H",g1H);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g1K",g1K);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g1L",g1L);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g2H",g2H);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g2K",g2K);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g2L",g2L);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g1Ag2",g1Ag2);
LACBED_reduced_stack.SetNumberNote("g-
vectors:g1Mg2",g1Mg2);
}
GetDate(f,date);
GetTime(f,time);
datetime=date+" "+time;

```

Appendix E

In Memoriam



GatanTM double-tilt heating holder - oops

Bibliography

- [1] G. Thomson, “The early history of electron diffraction,” *Contemporary Physics*, vol. 9, no. 1, pp. 1–15, 1968.
- [2] T. Shelton and M. de Cervantes, *Don Quixote*. 1620.
- [3] D. Cherns, J. W. Steeds, and R. Vincent, “Convergent Beam Electron Diffraction,” *Handbook of Microscopy Set: Applications in Materials Science, Solid-State Physics and Chemistry*, vol. 1, no. SUPPL. 1, pp. 467–485, 2008.
- [4] J. M. LeBeau, S. D. Findlay, L. J. Allen, and S. Stemmer, “Position averaged convergent beam electron diffraction: Theory and applications,” *Ultramicroscopy*, vol. 110, no. 2, pp. 118–125, 2010.
- [5] M. Tanaka, R. Saito, K. Ueno, and Y. Harada, “Large-angle convergent-beam electron diffraction,” *Microscopy*, vol. 29, no. 4, pp. 408–412, 1980.
- [6] C. T. Koch, “Aberration-compensated large-angle rocking-beam electron diffraction,” *Ultramicroscopy*, vol. 111, no. 7, pp. 828–840, 2011.
- [7] R. Beanland, P. J. Thomas, D. I. Woodward, P. A. Thomas, and R. A. Roemer, “Digital electron diffraction—seeing the whole picture,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 69, no. 4, pp. 427–434, 2013.
- [8] J. J. Thomson, “Xl. cathode rays,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 44, no. 269, pp. 293–316, 1897.
- [9] I. Falconer, “Corpuscles, electrons and cathode rays: Jj thomson and the ‘discovery of the electron’,” *The British journal for the history of science*, vol. 20, no. 3, pp. 241–276, 1987.
- [10] J. Z. Buchwald and A. Warwick, *Histories of the electron: the birth of microphysics*. MIT Press, 2004.

- [11] L. d. Broglie, “Xxxv. a tentative theory of light quanta,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 47, no. 278, pp. 446–458, 1924.
- [12] P. Weinberger, “Revisiting louis de broglie’s famous 1924 paper in the philosophical magazine,” *Philosophical magazine letters*, vol. 86, no. 7, pp. 405–410, 2006.
- [13] W. Elsasser, “Bemerkungen zur quantenmechanik freier elektronen,” *Naturwissenschaften*, vol. 13, no. 33, p. 711, 1925.
- [14] C. Davisson and L. H. Germer, “The scattering of electrons by a single crystal of nickel,” *Nature*, vol. 119, no. 2998, p. 558, 1927.
- [15] C. Davisson and L. H. Germer, “Diffraction of electrons by a crystal of nickel,” *Physical review*, vol. 30, no. 6, p. 705, 1927.
- [16] G. P. Thomson and A. Reid, “Diffraction of cathode rays by a thin film,” *Nature*, vol. 119, no. 3007, p. 890, 1927.
- [17] S. Kikuchi, “Diffraction of cathode rays by mica,” *Proceedings of the Imperial Academy*, vol. 4, no. 6, pp. 271–274, 1928.
- [18] S. Nishikawa and S. Kikuchi, “Diffraction of cathode rays by mica,” *Nature*, vol. 121, no. 3061, p. 1019, 1928.
- [19] H. Bethe, “Theorie der beugung von elektronen an kristallen,” *Annalen der Physik*, vol. 392, no. 17, pp. 55–129, 1928.
- [20] M. Blackman, “On the intensities of electron diffraction rings,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 173, no. 952, pp. 68–82, 1939.
- [21] J. Pendry, “Leed and the crystallography of surfaces,” *Surface science reports*, vol. 19, no. 3-6, pp. 87–97, 1993.
- [22] P. Midgley and M. Saunders, “Quantitative electron diffraction: from atoms to bonds,” *Contemporary Physics*, vol. 37, no. 6, pp. 441–456, 1996.
- [23] T. Mulvey, “Origins and historical development of the electron microscope,” *British journal of applied physics*, vol. 13, no. 5, p. 197, 1962.

- [24] H. Boersch, “Über das primäre und sekundäre Bild im Elektronenmikroskop. II. Strukturuntersuchung mittels Elektronenbeugung,” *Annalen der Physik*, vol. 419, pp. 75–80, jan 1936.
- [25] F. Haguénau, P. Hawkes, J. Hutchison, B. Satiat-Jeunemaître, G. Simon, and D. Williams, “Key events in the history of electron microscopy,” *Microscopy and Microanalysis*, vol. 9, no. 2, pp. 96–138, 2003.
- [26] J. B. Le Poole, “A new electron microscope with continuously variable magnification,” *Philips Tech. Rev.*, vol. 9, no. 2, 1947.
- [27] M. E. Haine, R. S. Page, and R. G. Garfitt, “A Three-Stage Electron Microscope with Stereographic Dark Field, and Electron Diffraction Capabilities,” *Journal of Applied Physics*, vol. 21, no. 2, pp. 173–182, 1950.
- [28] E. Suito, K. Takiyama, and N. Uyeda, “Paper iv colloid morphological and crystalline studies of” bikini dust” fallen on the no. 5 fukuryu maru by electron microscopy and diffraction methods.(the radioactive dust from the nuclear detonation),” *Institute for Chemical Research, Kyoto University*, 1954.
- [29] E. Suito and N. Uyeda, “Study of the thickness of single micro-crystals of gold by electron diffraction method,” *Proceedings of the Japan Academy*, vol. 29, no. 7, pp. 331–336, 1953.
- [30] E. Suito and N. Uyeda, “Study of single micro-crystals of gold by a three-stage electron microscope,” *Proceedings of the Japan Academy*, vol. 29, no. 7, pp. 324–330, 1953.
- [31] A. C. van Dorsten, W. J. Oosterkamp, and J. B. LePoole, “An experimental electron microscope for 400,” *Philips. techn. Rev.*, vol. 9, p. 193, 1947.
- [32] L. Reimer, “Electron diffraction methods in tem, stem and sem,” *Scanning*, vol. 2, no. 1, pp. 3–19, 1979.
- [33] G. Honjo and K. Mihama, “A study of clay minerals by electron-diffraction diagrams due to individual crystallites,” *Acta Crystallographica*, vol. 7, no. 6-7, pp. 511–513, 1954.
- [34] B. K. Vainshtein, *Structure analysis by electron diffraction*. Elsevier, 1964.
- [35] M. F. C. Ladd, R. A. Palmer, and R. A. Palmer, *Structure determination by X-ray crystallography*. Springer, 1985.

- [36] J. M. Zuo and J. C. H. Spence, *Electron microdiffraction*. Springer Science & Business Media, 1992.
- [37] J. Spence, “On the accurate measurement of structure-factor amplitudes and phases by electron diffraction,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 49, no. 2, pp. 231–260, 1993.
- [38] J.-M. Zuo and J.-L. Rouvière, “Solving difficult structures with electron diffraction,” *IUCrJ*, vol. 2, no. 1, pp. 7–8, 2015.
- [39] R. Vincent and P. Midgley, “Double conical beam-rocking system for measurement of integrated electron diffraction intensities,” *Ultramicroscopy*, vol. 53, no. 3, pp. 271–282, 1994.
- [40] L. Marks and W. Sinkler, “Sufficient conditions for direct methods with swift electrons,” *Microscopy and Microanalysis*, vol. 9, no. 5, pp. 399–410, 2003.
- [41] C. Own, *System Design and Verification of the Precession Electron Diffraction Technique*. PhD thesis, Department of Materials Science and Engineering, Northwestern, Evanston, 2005.
- [42] S. Nicolopoulos and D. Bultreys, “Precession electron diffraction and tem applications,” *NanoMEGAS SPRL, Blvd Edmond Machtens*, vol. 79, 2004.
- [43] P. A. Midgley and A. S. Eggeman, “Precession electron diffraction—a topical review,” *IUCrJ*, vol. 2, no. 1, pp. 126–136, 2015.
- [44] A. S. Eggeman and P. A. Midgley, “Precession electron diffraction,” in *Advances in Imaging and Electron Physics*, vol. 170, pp. 1–63, Elsevier, 2012.
- [45] J. Gjønnes, V. Hansen, S. Andersen, C. Marioara, and X. Li, “Electron crystallography of aluminum alloy phases,” *Zeitschrift für Kristallographie-Crystalline Materials*, vol. 218, no. 4, pp. 293–307, 2003.
- [46] S. Hovmöller, “Crisp: crystallographic image processing on a personal computer,” *Ultramicroscopy*, vol. 41, no. 1-3, pp. 121–135, 1992.
- [47] S. Hovmöller, A. Sjögren, G. Farrants, M. Sundberg, and B.-O. Marinder, “Accurate atomic positions from electron microscopy,” *Nature*, vol. 311, pp. 238–241, sep 1984.
- [48] T. E. Weirich, R. Ramlau, A. Simon, S. Hovmöller, and X. Zou, “A crystal structure determined with 0.02 Å accuracy by electron microscopy,” *Nature*, vol. 382, no. 6587, p. 144, 1996.

- [49] X. Zou, Y. Sukharev, and S. Hovmöller, “Quantitative electron diffraction—new features in the program system eld,” *Ultramicroscopy*, vol. 52, no. 3-4, pp. 436–444, 1993.
- [50] D. Edwards, T. O. Mason, W. Sinkler, L. Marks, F. Goutenoire, and K. R. Poeppelmeier, “A structural investigation of $\text{Ga}_3\text{Xin}_5\text{Xsn}_2\text{O}_{16}$,” *Journal of Solid State Chemistry*, vol. 140, no. 2, pp. 242–250, 1998.
- [51] T. E. Weirich, X. Zou, R. Ramlau, A. Simon, G. L. Cascarano, C. Giacovazzo, and S. Hovmöller, “Structures of nanometre-size crystals determined from selected-area electron diffraction data,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 56, no. 1, pp. 29–35, 2000.
- [52] X. Zou, Z. Mo, S. Hovmoller, X. Li, and K. Kuo, “Three-dimensional reconstruction of the ν -alcrfe phase by electron crystallography,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 59, no. 6, pp. 526–539, 2003.
- [53] D. Zhang, P. Oleynikov, S. Hovmöller, and X. Zou, “Collecting 3d electron diffraction data by the rotation method,” *Zeitschrift für Kristallographie International journal for structural, physical, and chemical aspects of crystalline materials*, vol. 225, no. 2-3, pp. 94–102, 2010.
- [54] U. Kolb, T. Gorelik, C. Kübel, M. Otten, and D. Hubert, “Towards automated diffraction tomography: Part i—data acquisition,” *Ultramicroscopy*, vol. 107, no. 6-7, pp. 507–513, 2007.
- [55] E. Mugnaioli, “Single nano crystal analysis using electron diffraction tomography,”
- [56] J. Su, E. Kapaca, L. Liu, V. Georgieva, W. Wan, J. Sun, V. Valtchev, S. Hovmöller, and X. Zou, “Structure analysis of zeolites by rotation electron diffraction (red),” *Microporous and Mesoporous Materials*, vol. 189, pp. 115–125, 2014.
- [57] P. Guo, L. Liu, Y. Yun, J. Su, W. Wan, H. Gies, H. Zhang, F.-S. Xiao, and X. Zou, “Ab initio structure determination of interlayer expanded zeolites by single crystal rotation electron diffraction,” *Dalton Transactions*, vol. 43, no. 27, pp. 10593–10601, 2014.

- [58] S. Smeets, L. B. McCusker, C. Baerlocher, E. Mugnaioli, and U. Kolb, "Using focus to solve zeolite structures from three-dimensional electron diffraction data," *Journal of Applied Crystallography*, vol. 46, no. 4, pp. 1017–1023, 2013.
- [59] Y. Lorgouilloux, M. Dodin, E. Mugnaioli, C. Marichal, P. Caullet, N. Bats, U. Kolb, and J.-L. Paillaud, "Im-17: a new zeolitic material, synthesis and structure elucidation from electron diffraction adt data and rietveld analysis," *Rsc Advances*, vol. 4, no. 37, pp. 19440–19449, 2014.
- [60] J. Jiang, J. L. Jorda, J. Yu, L. A. Baumes, E. Mugnaioli, M. J. Diaz-Cabanas, U. Kolb, and A. Corma, "Synthesis and structure determination of the hierarchical meso-microporous zeolite itq-43," *Science*, vol. 333, no. 6046, pp. 1131–1134, 2011.
- [61] E. Mugnaioli and U. Kolb, "Applications of automated diffraction tomography (adt) on nanocrystalline porous materials," *Microporous and Mesoporous Materials*, vol. 166, pp. 93–101, 2013.
- [62] U. Kolb, T. E. Gorelik, E. Mugnaioli, and A. Stewart, "Structural characterization of organics using manual and automated electron diffraction," *Polymer Reviews*, vol. 50, no. 3, pp. 385–409, 2010.
- [63] M. Warren, "Atomic structures solved in minutes - cross-disciplinarity led to method's use on small molecules.," *Nature*, vol. 563, pp. 16–17, 2018.
- [64] E. Van Genderen, M. Clabbers, P. Das, A. Stewart, I. Nederlof, K. Barentsen, Q. Portillo, N. Pannu, S. Nicolopoulos, Gruene, *et al.*, "Ab initio structure determination of nanocrystals of organic pharmaceutical compounds by electron diffraction at room temperature using a timepix quantum area direct electron detector," *Acta Crystallographica Section A: Foundations and Advances*, vol. 72, no. 2, pp. 236–242, 2016.
- [65] E. Mugnaioli, I. Andrusenko, T. Schüler, N. Loges, R. E. Dinnebier, M. Panthöfer, W. Tremel, and U. Kolb, "Ab initio structure determination of vaterite by automated electron diffraction," *Angewandte Chemie International Edition*, vol. 51, no. 28, pp. 7041–7045, 2012.
- [66] C. S. Birkel, E. Mugnaioli, T. Gorelik, U. Kolb, M. Panthöfer, and W. Tremel, "Solution synthesis of a new thermoelectric zn_{1+x}sb nanophase and its structure determination using automated electron diffraction tomography," *Journal of the American Chemical Society*, vol. 132, no. 28, pp. 9881–9889, 2010.

- [67] A. Rabkin, S. Samuha, R. E. Abutbul, V. Ezersky, L. Meshi, and Y. Golan, “New nanocrystalline materials: a previously unknown simple cubic phase in the sns binary system,” *Nano letters*, vol. 15, no. 3, pp. 2174–2179, 2015.
- [68] E. Mugnaioli, T. Gorelik, and U. Kolb, “‘ab initio’ structure solution from electron diffraction data obtained by a combination of automated diffraction tomography and precession technique,” *Ultramicroscopy*, vol. 109, no. 6, pp. 758–765, 2009.
- [69] Y. Wang, T. Yang, H. Xu, X. Zou, and W. Wan, “On the quality of the continuous rotation electron diffraction data for accurate atomic structure determination of inorganic compounds,” *Journal of Applied Crystallography*, vol. 51, no. 4, 2018.
- [70] M. Gemmi, M. G. La Placa, A. S. Galanis, E. F. Rauch, and S. Nicolopoulos, “Fast electron diffraction tomography,” *Journal of Applied Crystallography*, vol. 48, no. 3, pp. 718–727, 2015.
- [71] S. Smeets, X. Zou, and W. Wan, “Serial electron crystallography for structure determination and phase analysis of nanocrystalline materials,” *Journal of applied crystallography*, vol. 51, no. 5, 2018.
- [72] C. Sung and D. B. Williams, “Principles and applications of convergent beam electron diffraction: A bibliography (1938-1990),” *Journal of electron microscopy technique*, vol. 17, no. 1, pp. 95–118, 1991.
- [73] L. Palatinus, D. Jacob, P. Cuvillier, M. Klementová, W. Sinkler, and L. D. Marks, “Structure refinement from precession electron diffraction data,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 69, no. 2, pp. 171–188, 2013.
- [74] C. Own, L. Marks, and W. Sinkler, “Precession electron diffraction 1: multi-slice simulation,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 62, no. 6, pp. 434–443, 2006.
- [75] C. A. Corrêa, M. Klementová, and L. Palatinus, “Structural Analysis of Nanocrystals by 3D Electron Diffraction Tomography,” in *WDS’14 Proceedings of Contributed Papers—Physics*, 2014.
- [76] C. Satto, J. Jansen, C. Lexcellent, and D. Schryvers, “Structure refinement of l21 cu–zn–al austenite, using dynamical electron diffraction data,” *Solid state communications*, vol. 116, no. 5, pp. 273–277, 2000.

- [77] W. Tirry, D. Schryvers, K. Jorissen, and D. Lamoen, “Quantitative determination of the crystal structure of ni₄ti₃ precipitates,” *Materials Science and Engineering: A*, vol. 438, pp. 517–520, 2006.
- [78] A. Dudka, A. Avilov, and G. Lepeshov, “Crystal structure refinement from electron diffraction data,” *Crystallography Reports*, vol. 53, no. 3, pp. 530–536, 2008.
- [79] J. Jansen, D. Tang, H. Zandbergen, and H. Schenk, “Msls, a least-squares procedure for accurate crystal structure refinement from dynamical electron diffraction patterns,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 54, no. 1, pp. 91–101, 1998.
- [80] A. Dudka, A. Avilov, and S. Nicolopoulos, “Crystal structure refinement using bloch-wave method for precession electron diffraction,” *Ultramicroscopy*, vol. 107, no. 6-7, pp. 474–482, 2007.
- [81] L. Palatinus, V. Petříček, and C. A. Corrêa, “Structure refinement using precession electron diffraction tomography and dynamical diffraction: theory and implementation,” *Acta Crystallographica Section A: Foundations and Advances*, vol. 71, no. 2, pp. 235–244, 2015.
- [82] L. Palatinus, C. A. Corrêa, G. Steciuk, D. Jacob, P. Roussel, P. Boullay, M. Klementová, M. Gemmi, J. Kopeček, M. C. Domeneghetti, *et al.*, “Structure refinement using precession electron diffraction tomography and dynamical diffraction: tests on experimental data,” *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, vol. 71, no. 6, pp. 740–751, 2015.
- [83] G. Steciuk, A. David, V. Petříček, L. Palatinus, B. Mercey, W. Prellier, A. Pautrat, and P. Boullay, “Precession electron diffraction tomography on twinned crystals: application to cati₃ thin films,” *Journal of Applied Crystallography*, vol. 52, no. 3, 2019.
- [84] M. Klementová, M. Karlík, P. Novák, and L. Palatinus, “Structure determination of a new phase ni₈ti₅ by electron diffraction tomography,” *Intermetallics*, vol. 85, pp. 110–116, 2017.
- [85] C. A. Corrêa, M. Klementová, V. Dříněk, J. Kopeček, and L. Palatinus, “Crystal structure redetermination of ϵ -ni₃si₂ from a single nanowire by dynamical refinement of precession electron diffraction data,” *Journal of Alloys and Compounds*, vol. 672, pp. 505–509, 2016.

- [86] L. Palatinus, P. Brázda, P. Boullay, O. Perez, M. Klementová, S. Petit, V. Eigner, M. Zaarour, and S. Mintova, “Hydrogen positions in single nanocrystals revealed by electron diffraction,” *Science*, vol. 355, no. 6321, pp. 166–169, 2017.
- [87] G. Möllenstedt, “My early work on convergent-beam electron diffraction,” *physica status solidi (a)*, vol. 116, no. 1, pp. 13–22, 1989.
- [88] W. Kossel and G. Möllenstedt, “Elektroneninterferenzen in konvergentem bündel,” *Naturwissenschaften*, vol. 26, no. 40, pp. 660–661, 1938.
- [89] G. I. Finch and H. Wilman, “The study of surface structure by electron diffraction,” in *Ergebnisse der exakten naturwissenschaften*, pp. 353–436, Springer, 1937.
- [90] D. Eaglesham, C. Humphreys, N. M. Alford, W. Clegg, M. Harmer, and J. Birchall, “New phases in the superconducting y: Ba: Cu: O system,” *Applied physics letters*, vol. 51, no. 6, pp. 457–459, 1987.
- [91] P. t. Goodman and G. Lehmpfuhl, “Observation of the breakdown of friedel’s law in electron diffraction and symmetry determination from zero-layer interactions,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 24, no. 3, pp. 339–347, 1968.
- [92] P. Goodman, “A practical method of three-dimensional space-group analysis using convergent-beam electron diffraction,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 31, no. 6, pp. 804–810, 1975.
- [93] A. Tinnappel and K. Kambe, “Group-theoretical analysis of symmetry problems in electron diffraction,” *Acta Crystallographica Section A*, vol. 31, no. S, p. 6, 1975.
- [94] B. F. Buxton, J. Eades, J. W. Steeds, and G. M. Rackham, “The symmetry of electron diffraction zone axis patterns,” *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 281, no. 1301, pp. 171–194, 1976.
- [95] J. Eades, “Convergent-beam techniques in transmission electron microscopy,” *Applied surface science*, vol. 26, no. 3, pp. 280–293, 1986.
- [96] J. Gjønnes and A. Moodie, “Extinction conditions in the dynamic theory of electron diffraction,” *Acta Crystallographica*, vol. 19, no. 1, pp. 65–67, 1965.

- [97] P. Goodman and G. Lehmpfuhl, “Elektronenbeugungsuntersuchungen im konvergenten bündel mit dem siemens elmiskop i,” *Zeitschrift für Naturforschung A*, vol. 20, no. 1, pp. 110–114, 1965.
- [98] M. Tanaka, H. Sekii, and T. Nagasawa, “Space-group determination by dynamic extinction in convergent-beam electron diffraction,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 39, no. 6, pp. 825–837, 1983.
- [99] M. Tanaka, H. Sekii, and T. Nagasawa, “Space-group determination by dynamic extinction in convergent-beam electron diffraction: errata,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 40, no. 6, pp. 721–721, 1984.
- [100] M. Tanaka, “Point-group and space-group determination by convergent-beam electron diffraction,” *International Tables of Crystallography section B*, 2010.
- [101] M. Tanaka, M. Terauchi, K. Tsuda, and K. Saitoh, *Convergent-beam Electron Diffraction IV*. JEOL, 2002.
- [102] D. Eaglesham, “Applications of convergent beam electron diffraction in materials science,” *Journal of electron microscopy technique*, vol. 13, no. 1, pp. 66–75, 1989.
- [103] K. Tsuda and M. Tanaka, “Direct observation of the symmetry breaking of the nanometer-scale local structure in the paraelectric cubic phase of batio3 using convergent-beam electron diffraction,” *Applied Physics Express*, vol. 9, no. 7, p. 071501, 2016.
- [104] K. Tsuda and M. Tanaka, “Nanometer-scale local structural study of the paraelectric cubic phase of knbo3 by convergent-beam electron diffraction,” *Japanese Journal of Applied Physics*, vol. 56, no. 10S, p. 10PB09, 2017.
- [105] Y.-T. Shao and J.-M. Zuo, “Nanoscale symmetry fluctuations in ferroelectric barium titanate, batio3,” *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, vol. 73, no. 4, pp. 708–714, 2017.
- [106] K.-H. Kim and J.-M. Zuo, “Symmetry quantification and mapping using convergent beam electron diffraction,” *Ultramicroscopy*, vol. 124, pp. 71–76, 2013.
- [107] C. H. MacGillavry, “Dynamical theory of electron diffraction,” *Physica (Utrecht)*, vol. 7, pp. 329–343, 1940.

- [108] J. M. Cowley, “Crystal structure determination by electron diffraction,” *Progress in Materials Science*, vol. 13, pp. 267–321, 1968.
- [109] R. Uyeda, M. Nonoyama, and M. Kogiso, “Determination of the wavelength of electrons from a kikuchi pattern,” *Microscopy*, vol. 14, no. 4, pp. 296–300, 1965.
- [110] R. Høier, “A method to determine the ratio between lattice parameter and electron wavelength from kikuchi line intersections,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 25, no. 4, pp. 516–518, 1969.
- [111] F. Nagata and A. Fukuhara, “222 electron reflection from aluminium and systematic interaction,” *Japanese Journal of Applied Physics*, vol. 6, no. 10, p. 1233, 1967.
- [112] R. Uyeda, “Dynamical effects in high-voltage electron diffraction,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 24, no. 1, pp. 175–181, 1968.
- [113] A. Fox and R. Fisher, “A summary of low-angle x-ray atomic scattering factors measured by the critical voltage effect in high energy electron diffraction,” *Australian Journal of Physics*, vol. 41, no. 3, pp. 461–468, 1988.
- [114] J. Lally, C. Humphreys, A. Metherell, and R. Fisher, “The critical voltage effect in high voltage electron microscopy,” *Philosophical Magazine*, vol. 25, no. 2, pp. 321–343, 1972.
- [115] E. Menon and A. Fox, “Debye—waller factors of stoichiometric and al-rich γ -tial alloys,” *Philosophical Magazine A*, vol. 77, no. 3, pp. 577–592, 1998.
- [116] R. Vincent, “Structure of AuGeAs determined by convergent-beam electron diffraction. I. Derivation of basic structure,” *Philos. Mag.*, vol. 50, pp. 745–763, 1984.
- [117] R. Vincent, “Structure of AuGeAs determined by convergent-beam electron diffraction. II. Refinement of structural parameters,” *Phil Mag A*, vol. 50, pp. 765–786, 1984.
- [118] Y. Guo, *Three-beam electron diffraction for structural investigations*. PhD thesis, Department of Materials Science and Engineering, Monash University, 2016.

- [119] J.-M. Zuo, J. Spence, and R. Høier, “Accurate structure-factor phase determination by electron diffraction in noncentrosymmetric crystals,” *Physical review letters*, vol. 62, no. 5, p. 547, 1989.
- [120] D. Bird and M. Saunders, “Sensitivity and accuracy of cbed pattern matching,” *Ultramicroscopy*, vol. 45, no. 2, pp. 241–251, 1992.
- [121] R. Høier, L. N. Bakken, K. Marthinsen, and R. Holmestad, “Structure factor determination in non-centrosymmetric crystals by a two-dimensional cbed-based multi-parameter refinement method,” *Ultramicroscopy*, vol. 49, no. 1-4, pp. 159–170, 1993.
- [122] K. Tsuda and M. Tanaka, “Refinement of crystal structure parameters using convergent-beam electron diffraction: the low-temperature phase of SrTiO_3 ,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 51, no. 1, pp. 7–19, 1995.
- [123] R. Holmestad, C. R. Birkeland, K. Marthinsen, R. Høier, and J. M. Zuo, “Use of quantitative convergent-beam electron diffraction in materials science,” *Microscopy research and technique*, vol. 46, no. 2, pp. 130–145, 1999.
- [124] X. Sang, *Electron density determination and bonding in tetragonal binary intermetallics by convergent beam electron diffraction*. PhD thesis, University of Pittsburgh, 2012.
- [125] P. N. Nakashima, “Improved quantitative cbed structure-factor measurement by refinement of nonlinear geometric distortion corrections,” *Journal of applied crystallography*, vol. 38, no. 2, pp. 374–376, 2005.
- [126] P. N. Nakashima, A. E. Smith, J. Etheridge, and B. C. Muddle, “The bonding electron density in aluminum,” *Science*, vol. 331, no. 6024, pp. 1583–1586, 2011.
- [127] V. A. Streltsov, P. N. Nakashima, and A. W. Johnson, “A combination method of charge density measurement in hard materials using accurate, quantitative electron and x-ray diffraction: The α - Al_2O_3 case,” *Microscopy and Microanalysis*, vol. 9, no. 5, pp. 419–427, 2003.
- [128] M.-Y. Wu, J. Zhu, S.-Y. Li, Z.-H. Du, and S.-L. Li, “Measurement of the fine structure factors and extinction distance by convergent-beam electron diffraction in Co_3Ti with and without b doping,” *Journal of applied crystallography*, vol. 33, no. 4, pp. 1119–1121, 2000.

- [129] Y. Ogata, K. Tsuda, and M. Tanaka, “Determination of the electrostatic potential and electron density of silicon using convergent-beam electron diffraction,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 64, no. 5, pp. 587–597, 2008.
- [130] K. Müller, M. Schowalter, J. Jansen, K. Tsuda, J. Titantah, D. Lamoën, and A. Rosenauer, “Refinement of the 200 structure factor for GaAs using parallel and convergent beam electron nanodiffraction data,” *Ultramicroscopy*, vol. 109, no. 7, pp. 802–814, 2009.
- [131] F. Feng, J. Zhu, and A. Zhang, “Refinement of the crystal structural parameters of $\text{La}_{2/3}\text{Ca}_{1/3}\text{MnO}_3$ using quantitative convergent-beam electron diffraction,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 61, no. 4, pp. 453–459, 2005.
- [132] K. Tsuda, Y. Ogata, K. Takagi, T. Hashimoto, and M. Tanaka, “Refinement of crystal structural parameters and charge density using convergent-beam electron diffraction—the rhombohedral phase of LaCoO_3 ,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 58, no. 6, pp. 514–525, 2002.
- [133] M.-Y. Wu, S.-Y. Li, J. Zhu, Z.-H. Du, and S.-L. Li, “Measurement of Debye–Waller factors in Co_3Ti by quantitative CBED,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 56, no. 2, pp. 189–192, 2000.
- [134] X. Sang, A. Kulovits, and J. Wiezorek, “Simultaneous determination of highly precise Debye–Waller factors and multiple structure factors for chemically ordered tetragonal FePd ,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 67, no. 3, pp. 229–239, 2011.
- [135] X. Sang, A. Kulovits, and J. Wiezorek, “Determination of Debye–Waller factor and structure factors for Si by quantitative convergent-beam electron diffraction using off-axis multi-beam orientations,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 66, no. 6, pp. 685–693, 2010.
- [136] X. Sang, A. Kulovits, and J. Wiezorek, “Comparison of convergent beam electron diffraction methods for simultaneous structure and Debye–Waller factor determination,” *Ultramicroscopy*, vol. 126, pp. 48–59, 2013.
- [137] M. Saunders, “Quantitative zone-axis convergent beam electron diffraction: current status and future prospects,” *Microscopy and Microanalysis*, vol. 9, no. 5, pp. 411–418, 2003.

- [138] H. M. L. Faulkner, *Studies in phase and inversion problems for dynamical electron diffraction*. PhD thesis, school of physics, the university of Melbourne, 2003.
- [139] D. Bird and M. Saunders, “Inversion of convergent-beam electron diffraction patterns,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 48, no. 4, pp. 555–562, 1992.
- [140] L.-M. Peng and J.-M. Zuo, “Direct retrieval of crystal structure factors in THEED,” *Ultramicroscopy*, vol. 57, no. 1, pp. 1–9, 1995.
- [141] L. Allen, T. Josefsson, and H. Leeb, “Obtaining the crystal potential by inversion from electron scattering intensities,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 54, no. 4, pp. 388–398, 1998.
- [142] J. Spence, “Direct inversion of dynamical electron diffraction patterns to structure factors,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 54, no. 1, pp. 7–18, 1998.
- [143] L. J. Allen, H. M. L. Faulkner, and H. Leeb, “Inversion of dynamical electron diffraction data including absorption,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 56, no. 2, pp. 119–126, 2000.
- [144] L. Allen, C. Koch, M. Oxley, and J. Spence, “Inversion of dynamical electron scattering to obtain the crystal potential using data from two thicknesses,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 57, no. 4, pp. 473–474, 2001.
- [145] C. T. Koch, “Using dynamically scattered electrons for three-dimensional potential reconstruction,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 65, no. 5, pp. 364–370, 2009.
- [146] J. L. Hart, S. Liu, A. C. Lang, A. J. Hubert, A. Zukauskas, C. Canalias, R. Beanland, A. M. Rappe, M. Arredondo, and M. L. Taheri, “Electron-beam-induced ferroelectric domain behavior in the transmission electron microscope: Toward deterministic domain patterning,” *Physical Review B*, vol. 94, no. 17, p. 174104, 2016.
- [147] Y. Guo, P. N. Nakashima, and J. Etheridge, “Three-beam convergent-beam electron diffraction for measuring crystallographic phases,” *IUCrJ*, vol. 5, no. 6, 2018.

- [148] A. Hubert, R. Römer, and R. Beanland, “Structure refinement from ‘digital’ large angle convergent beam electron diffraction patterns,” *Ultramicroscopy*, vol. 198, pp. 1–9, 2019.
- [149] R. Beanland, K. Evans, A. Hubert, and R. Roemer, “Felix.” <https://github.com/RudoRoemer/Felix>. Accessed: 2019-07-10.
- [150] D. B. Williams and C. B. Carter, *Transmission Electron Microscopy: A Textbook for Materials Science*. Boston, MA: Springer US, 2009.
- [151] J. M. Zuo and J. C. H. Spence, *Advanced transmission electron microscopy*. Springer, 2017.
- [152] A. J. F. Metherell, “Diffraction of electrons by perfect crystals,” *Electron microscopy in materials science*, pp. 397–552, 1975.
- [153] P. J. Goodhew, *Specimen preparation for transmission electron microscopy of materials*. Oxford University Press, 1984.
- [154] V. B. Özdöl, V. Srot, and P. A. van Aken, “Sample preparation techniques for transmission electron microscopy,” *Handbook of Nanoscopy*, pp. 473–498, 2012.
- [155] J. Ayache, L. Beaunier, J. Boumendil, G. Ehret, and D. Laub, *Sample preparation handbook for transmission electron microscopy: techniques*, vol. 2. Springer Science & Business Media, 2010.
- [156] R. Vincent, “Techniques of convergent beam electron diffraction,” *Journal of electron microscopy technique*, vol. 13, no. 1, pp. 40–50, 1989.
- [157] R. Egerton, P. Li, and M. Malac, “Radiation damage in the tem and sem,” *Micron*, vol. 35, no. 6, pp. 399–409, 2004.
- [158] D. Palmer and M. Conley, “CrystalMaker,” *CrystalMaker Software, Bicester, England*, 2007.
- [159] W. Gropp, W. D. Gropp, A. D. F. E. E. Lusk, E. Lusk, and A. Skjellum, *Using MPI: portable parallel programming with the message-passing interface*, vol. 1. MIT press, 1999.
- [160] J. M. Cowley and A. F. Moodie, “The scattering of electrons by atoms and crystals. I. A new theoretical approach,” *Acta Crystallographica*, vol. 10, no. 10, pp. 609–619, 1957.

- [161] P. Goodman and A. Moodie, “Numerical evaluations of n-beam wave functions in electron scattering by the multi-slice method,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 30, no. 2, pp. 280–290, 1974.
- [162] B. G. Mendis, *Electron beam-specimen interactions and simulation methods in microscopy*. John Wiley & Sons, 2018.
- [163] P. G. Self, M. O’keefe, P. Buseck, and A. Spargo, “Practical computation of amplitudes and phases in electron diffraction,” *Ultramicroscopy*, vol. 11, no. 1, pp. 35–52, 1983.
- [164] E. J. Kirkland, *Advanced computing in electron microscopy*. Springer Science & Business Media, 2010.
- [165] U. Shmueli, *International Tables for Crystallography, Volume B: Reciprocal Space*. Springer Science & Business Media, to be published.
- [166] C. Humphreys, “The scattering of fast electrons by crystals,” *Reports on Progress in Physics*, vol. 42, no. 11, p. 1825, 1979.
- [167] P. B. Hirsch, *Electron microscopy of thin crystals*. Plenum Press, 1967.
- [168] R. Heidenreich, “Electron microscope and diffraction study of metal crystal textures by means of thin sections,” *Journal of Applied Physics*, vol. 20, no. 10, pp. 993–1010, 1949.
- [169] N. Kato, “Dynamical Theory of Electron Diffraction for a Finite Polyhedral Crystal I. Extension of Bethe’s Theory,” *Journal of the Physical Society of Japan*, vol. 7, no. 4, pp. 397–406, 1952.
- [170] K. Fujiwara, “Relativistic dynamical theory of electron diffraction,” *Journal of the Physical Society of Japan*, vol. 16, no. 11, pp. 2226–2238, 1961.
- [171] J. R. Hook and H. E. Hall, *Solid state physics*. Wiley, 1991.
- [172] C. Kittel, *Introduction to Solid State Physics*. Wiley, 2005.
- [173] N. Ashcroft and N. Mermin, *Solid State Physics*. Philadelphia: Saunders College, 1976.
- [174] L.-M. Peng, “Electron atomic scattering factors, Debye-Waller factors and the optical potential for high-energy electron diffraction,” *Journal of electron microscopy*, vol. 54, no. 3, pp. 199–207, 2005.

- [175] I. Lobato and D. Van Dyck, “An accurate parameterization for scattering factors, electron densities and electrostatic potentials for neutral atoms that obey all physical constraints,” *Acta Crystallographica Section A: Foundations and Advances*, vol. 70, no. 6, pp. 636–649, 2014.
- [176] P. Doyle and P. Turner, “Relativistic hartree–fock x-ray and electron scattering factors,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 24, no. 3, pp. 390–397, 1968.
- [177] G. Radi, “Complex lattice potentials in electron diffraction calculated for a number of crystals,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 26, no. 1, pp. 41–56, 1970.
- [178] P. H. Dederichs, “Dynamical diffraction theory by optical potential methods,” in *Solid state physics*, vol. 27, pp. 135–236, Elsevier, 1972.
- [179] M. Oxley and L. Allen, “Atomic scattering factors for k-shell electron energy-loss spectroscopy,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 57, no. 6, pp. 713–728, 2001.
- [180] M. Oxley and L. Allen, “Atomic scattering factors for k-shell and l-shell ionization by fast electrons,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 56, no. 5, pp. 470–490, 2000.
- [181] D. Eaglesham and S. Berger, “Energy filtering the “thermal diffuse” background in electron diffraction,” *Ultramicroscopy*, vol. 53, no. 4, pp. 319–324, 1994.
- [182] L. J. Allen, S. Findlay, *et al.*, “Modelling the inelastic scattering of fast electrons,” *Ultramicroscopy*, vol. 151, pp. 11–22, 2015.
- [183] D. Bird and Q. King, “Absorptive form factors for high-energy electron diffraction,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 46, no. 3, pp. 202–208, 1990.
- [184] A. Weickenmeier and H. Kohl, “Computation of absorptive form factors for high-energy electron diffraction,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 47, no. 5, pp. 590–597, 1991.
- [185] A. Weickenmeier and H. Kohl, “The influence of anisotropic thermal vibrations on absorptive form factors for high-energy electron diffraction,” *Acta*

- Crystallographica Section A: Foundations of Crystallography*, vol. 54, no. 3, pp. 283–289, 1998.
- [186] C. Rossouw, P. Miller, J. Drennan, and L. Allen, “Quantitative absorption corrections for electron diffraction: correlation between theory and experiment,” *Ultramicroscopy*, vol. 34, no. 3, pp. 149–163, 1990.
 - [187] L. Allen and C. Rossouw, “Absorptive potentials due to ionization and thermal diffuse scattering by fast electrons in crystals,” *Physical Review B*, vol. 42, no. 18, p. 11644, 1990.
 - [188] R. W. James, S. L. Bragg, and W. L. Bragg, *The Optical Principles of the Diffraction of X-rays*. Bell London, 1948.
 - [189] B. E. Warren, *X-ray Diffraction*. Courier Corporation, 1990.
 - [190] J.-M. Zuo and A. Weickenmeier, “On the beam selection and convergence in the Bloch-wave method,” *Ultramicroscopy*, vol. 57, no. 4, pp. 375–383, 1995.
 - [191] C. Birkeland, R. Holmestad, K. Marthinsen, and R. Høier, “Efficient beam-selection criteria in quantitative convergent beam electron diffraction,” *Ultramicroscopy*, vol. 66, no. 1-2, pp. 89–99, 1996.
 - [192] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users’ guide*, vol. 9. Siam, 1999.
 - [193] R. C. Gonzalez and R. E. Woods, “Digital image processing ,” *Publishing house of electronics industry*, vol. 141, no. 7, 2002.
 - [194] V. Klee and G. J. Minty, “How good is the simplex algorithm,” tech. rep., Washington Univ. Seattle dept. of mathematics, 1970.
 - [195] S. M. Robinson, “A short proof of Cramer’s rule,” *Mathematics Magazine*, pp. 94–95, 1970.
 - [196] J. D. Hoffman and S. Frankel, *Numerical methods for engineers and scientists*. CRC press, 2001.
 - [197] J. Eades, “Microdiffraction’s contribution to microcharacterization,” *Ultramicroscopy*, vol. 24, no. 2-3, pp. 143–154, 1988.

- [198] U. Kolb, T. Gorelik, and M. Otten, “Towards automated diffraction tomography. Part II — Cell parameter determination,” *Ultramicroscopy*, vol. 108, no. 8, pp. 763–772, 2008.
- [199] U. Kolb, E. Mugnaioli, and T. Gorelik, “Automated electron diffraction tomography—a new tool for nano crystal structure analysis,” *Crystal Research and Technology*, vol. 46, no. 6, pp. 542–554, 2011.
- [200] W. Wan, J. Sun, J. Su, S. Hovmöller, and X. Zou, “Three-dimensional rotation electron diffraction: software RED for automated data collection and data processing,” *Journal of applied crystallography*, vol. 46, no. 6, pp. 1863–1873, 2013.
- [201] M. Gemmi, M. Merlini, L. Palatinus, P. Fumagalli, and M. Hanfland, “Electron diffraction determination of 11.5 Å and HySo structures: Candidate water carriers to the Upper Mantle,” *American Mineralogist*, vol. 101, no. 12, pp. 2645–2654, 2016.
- [202] Y. Ogata, K. Tsuda, Y. Akishige, and M. Tanaka, “Refinement of the crystal structural parameters of the intermediate phase of h-BaTiO₃ using convergent-beam electron diffraction,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 60, no. 6, pp. 525–531, 2004.
- [203] K. Syassen, “Ruby under pressure,” *High Pressure Research*, vol. 28, no. 2, pp. 75–126, 2008.
- [204] S. Kondo, K. Tateishi, and N. Ishizawa, “Structural evolution of corundum at high temperatures,” *Japanese Journal of Applied Physics*, vol. 47, no. 1S, p. 616, 2008.
- [205] E. Newnham and Y. De Haan, “Refinement of the α -Al₂O₃, Ti₂O₃, V₂O₃ and Cr₂O₃ structures,” *Zeitschrift für Kristallographie-Crystalline Materials*, vol. 117, no. 1-6, pp. 235–237, 1962.
- [206] E. Maslen, V. Streltsov, N. Streltsova, N. Ishizawa, and Y. Satow, “Synchrotron X-ray study of the electron density in α -Al₂O₃,” *Acta Crystallographica Section B: Structural Science*, vol. 49, no. 6, pp. 973–980, 1993.
- [207] K. Tsuda and M. Tanaka, “Refinement of crystal structural parameters using two-dimensional energy-filtered cbed patterns,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 55, no. 5, pp. 939–954, 1999.

- [208] M. Colmont, L. Palatinus, M. Huve, H. Kabbour, S. Saitzek, N. Djelal, and P. Roussel, “On the use of dynamical diffraction theory to refine crystal structure from electron diffraction data: Application to KLa_5O_5 (v04) 2, a material with promising luminescent properties,” *Inorganic chemistry*, vol. 55, no. 5, pp. 2252–2260, 2016.
- [209] Z. Zhou, L. Palatinus, and J. Sun, “Structure determination of modulated structures by powder x-ray diffraction and electron diffraction,” *Inorganic Chemistry Frontiers*, vol. 3, no. 11, pp. 1351–1362, 2016.
- [210] T. Willhammar, J. Su, Y. Yun, X. Zou, M. Afeworki, S. C. Weston, H. B. Vroman, W. W. Lonergan, and K. G. Strohmaier, “High-throughput synthesis and structure of zeolite zsm-43 with two-directional 8-ring channels,” *Inorganic chemistry*, vol. 56, no. 15, pp. 8856–8864, 2017.
- [211] P. N. Nakashima, A. F. Moodie, and J. Etheridge, “A practical guide to the measurement of structure phases and magnitudes by three-beam convergent beam electron diffraction,” *Ultramicroscopy*, vol. 108, no. 9, pp. 901–910, 2008.
- [212] P. N. Nakashima, A. F. Moodie, and J. Etheridge, “Structural phase and amplitude measurement from distances in convergent-beam electron diffraction patterns,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 63, no. 5, pp. 387–390, 2007.
- [213] P. N. Nakashima, A. F. Moodie, and J. Etheridge, “Direct atomic structure determination by the inspection of structural phase,” *Proceedings of the National Academy of Sciences*, vol. 110, no. 35, pp. 14144–14149, 2013.
- [214] E. Hewitt and R. E. Hewitt, “The Gibbs-Wilbraham phenomenon: an episode in Fourier analysis,” *Archive for history of Exact Sciences*, vol. 21, no. 2, pp. 129–160, 1979.
- [215] N. F. Mott, “The scattering of electrons by atoms,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 127, no. 806, pp. 658–665, 1930.
- [216] M. Saunders, D. Bird, N. Zaluzec, W. Burgess, A. Preston, and C. Humphreys, “Measurement of low-order structure factors for silicon from zone-axis CBED patterns,” *Ultramicroscopy*, vol. 60, no. 2, pp. 311–323, 1995.

- [217] M. Saunders, A. Fox, and P. Midgley, “Quantitative zone-axis convergent-beam electron diffraction (cbcd) studies of metals. i. structure-factor measurements,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 55, no. 3, pp. 471–479, 1999.
- [218] J.-M. Zuo, J. Spence, J. Downs, and J. Mayer, “Measurement of individual structure-factor phases with tenth-degree accuracy: the 00.2 reflection in beo studied by electron and x-ray diffraction,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 49, no. 3, pp. 422–429, 1993.
- [219] J. E. Bernard and A. Zunger, “Bonding charge density in GaAs,” *Physical review letters*, vol. 62, no. 19, p. 2328, 1989.
- [220] R. Saravanan, Y. Ono, M. Isshiki, K. Ohno, and T. Kajitani, “Electron density distribution in GaAs using MEM,” *Journal of Physics and Chemistry of Solids*, vol. 64, no. 1, pp. 51–58, 2003.
- [221] A. Stevenson, “Thermal vibrations and bonding in GaAs: an extended-face crystal study,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 50, no. 5, pp. 621–632, 1994.
- [222] A. G. Fox, “Is it feasible to determine the bonding charge density of stoichiometric γ -TiAl through structure factor measurements?,” *Philosophical magazine letters*, vol. 68, no. 1, pp. 29–37, 1993.
- [223] R. Shukla and E. Sternin, “Debye-Waller factor in Cu: A Green’s function approach,” *Philosophical Magazine B*, vol. 74, no. 1, pp. 1–11, 1996.
- [224] C. Martin and D. O’Connor, “Anharmonic contributions to Bragg diffraction. I. Copper and aluminium,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 34, no. 4, pp. 500–505, 1978.
- [225] B. Dawson, “A general structure factor formalism for interpreting accurate x-ray and neutron diffraction data,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 298, no. 1454, pp. 255–263, 1967.
- [226] W. C. Hamilton, *Statistics in physical science*. New York: Ronald Press, 1964.
- [227] Z. Barnea, G. J. McIntyre, and G. R. Moss, “Diffraction Studies of Real Atoms and Real Crystals,” *Australian Academy of Sciences, Melbourne*, 1974.

- [228] R. Saravanan, S. Mohanlal, and K. Chandrasekaran, “Anharmonic temperature factors, anomalous-dispersion effects and bonding charges in gallium arsenide,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 48, no. 1, pp. 4–9, 1992.
- [229] R. Uno, T. Okano, and K. Yukino, “Electron distribution in GaAs as revealed by the X-ray diffraction,” *Journal of the Physical Society of Japan*, vol. 28, no. 2, pp. 437–442, 1970.
- [230] V. Bublik and S. Gorelik, “Über die röntgenographische analyse der mittleren quadratischen atomverschiebungen in einigen halbleiterkristallen,” *Kristall und Technik*, vol. 12, no. 8, pp. 859–869, 1977.
- [231] J. Stahn, M. Möhle, and U. Pietsch, “Comparison of experimental and theoretical structure amplitudes and valence charge densities of GaAs,” *Acta Crystallographica Section B: Structural Science*, vol. 54, no. 3, pp. 231–239, 1998.
- [232] M. Schowalter, A. Rosenauer, J. Titantah, and D. Lamoen, “Computation and parametrization of the temperature dependence of Debye–Waller factors for group IV, III–V and II–VI semiconductors,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 65, no. 1, pp. 5–17, 2009.
- [233] A. Fazeli Kisomi and S. Mousavi, “Ab initio calculations of the phonon and thermal properties of a (GaAs) 1/(AlAs) 1 superlattice and comparing them with the Ga 0.5 Al 0.5 As alloy,” *Chinese Journal of Physics*, vol. 55, pp. 1062–1066, 2017.
- [234] M. Kushwaha, “Compressibilities, Debye-Waller factors, and melting criteria for II-VI and III-V compound semiconductors,” *Physical Review B*, vol. 24, no. 4, p. 2115, 1981.
- [235] J. S. Reid, “Debye-waller factors of zinc-blende-structure materials—a lattice dynamical comparison,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 39, no. 1, pp. 1–13, 1983.
- [236] D. Többsen, N. Stüßer, K. Knorr, H. Mayer, and G. Lampert, “E9: the new high-resolution neutron powder diffractometer at the Berlin neutron scattering center,” in *Materials Science Forum*, vol. 378, pp. 288–293, Trans Tech Publ, 2001.

- [237] S. Pillet, M. Souhassou, C. Lecomte, K. Schwarz, P. Blaha, M. Rérat, A. Lichanot, and P. Roversi, “Recovering experimental and theoretical electron densities in corundum using the multipolar model: IUCr multipole refinement project,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 57, no. 3, pp. 290–303, 2001.
- [238] S. Israel, R. Saravanan, and R. Rajaram, “Electronic structure of InP at RT, 200 and 100 K,” *Physica B: Condensed Matter*, vol. 349, no. 1-4, pp. 390–400, 2004.
- [239] C. Ferrari, C. Bocchi, O. Moze, C. Wilson, and R. Fornari, “Time-of-flight neutron diffraction investigation of temperature factors in the zn blende semiconductor inp,” *Physica B: Condensed Matter*, vol. 180, pp. 627–628, 1992.
- [240] H. Sawada, “Residual electron density study of α -aluminum oxide through refinement of experimental atomic scattering factors,” *Materials research bulletin*, vol. 29, no. 2, pp. 127–133, 1994.
- [241] P. Ballirano and R. Caminiti, “Rietveld refinements on laboratory energy dispersive X-ray diffraction (EDXD) data,” *Journal of Applied Crystallography*, vol. 34, no. 6, pp. 757–762, 2001.
- [242] M. Oetzel and G. Heger, “Laboratory X-ray powder diffraction: a comparison of different geometries with special attention to the usage of the Cu K α doublet,” *Journal of Applied Crystallography*, vol. 32, no. 4, pp. 799–807, 1999.
- [243] W. Webb, “Atomic displacements in metallic solid solutions,” *Journal of Applied Physics*, vol. 33, no. 12, pp. 3546–3552, 1962.
- [244] J. P. Cline and R. W. Cheary, “The Design, Alignment, Calibration and Performance Characteristics of the Conventional Laboratory Diffractometer,” in *Workshop notes from the 45th and 47th Denver X-ray conference*, 1998.
- [245] J.-M. Zuo, J. Spence, and M. O’keeffe, “Bonding in GaAs,” *Physical review letters*, vol. 61, no. 3, p. 353, 1988.
- [246] J.-M. Zuo, M. Kim, M. O’keeffe, and J. Spence, “Direct observation of d-orbital holes and Cu–Cu bonding in Cu₂O,” *Nature*, vol. 401, no. 6748, p. 49, 1999.
- [247] J. Zuo, M. O’Keefe, and J. Spence, “Have orbitals really been observed?,” *Journal of Chemical Education*, vol. 78, no. 7, p. 877, 2001.

- [248] P. Coppens, *X-ray charge densities and chemical bonding*, vol. 4. International Union of Crystallography, 1997.
- [249] C. Gatti and P. Macchi, *Modern Charge-Density Analysis*. Springer Netherlands, 2012.
- [250] S. C. Capelli, H.-B. Bürgi, B. Dittrich, S. Grabowsky, and D. Jayatilaka, “Hirshfeld atom refinement,” *IUCrJ*, vol. 1, no. 5, pp. 361–379, 2014.
- [251] J.-M. Zuo and J. Spence, “Automated structure factor refinement from convergent-beam patterns,” *Ultramicroscopy*, vol. 35, no. 3-4, pp. 185–196, 1991.
- [252] J.-M. Zuo, “Automated structure-factor refinement from convergent-beam electron diffraction patterns,” *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 49, no. 3, pp. 429–435, 1993.
- [253] J. M. Zuo and J. C. H. Spence, “Experimental test of accuracy in automated structure refinement from CBED,” in *Electron microscopy 1994, VOL 1: Interdisciplinary developments and tools* (Jouffrey, B and Colliex, C, ed.), pp. 851–852, Int Federat Soc Electron Microscopy, 1994.
- [254] J. M. Zuo, R. Holmestad, J. C. H. Spence, and R. Høier, “Quantitative electron diffraction and applications to materials science,” in *Microbeam Analysis 1995: Proceedings of the 29th annual conference of the microbeam analysis society* (Etz, ES, ed.), Microbeam Analysis, (220 E 23rd St, Suite 909, New York, NY 10010), pp. 269–270, Microbeam Anal Soc, V C H Publishers, 1995.
- [255] T. S. Koritsanszky and P. Coppens, “Chemical applications of X-ray charge-density analysis,” *Chemical reviews*, vol. 101, no. 6, pp. 1583–1628, 2001.
- [256] R. Stewart, “Electron population analysis with rigid pseudoatoms,” *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 4, pp. 565–574, 1976.
- [257] S. M. Allen, “Foil thickness measurements from convergent-beam diffraction patterns,” *Philosophical Magazine A*, vol. 43, no. 2, pp. 325–335, 1981.
- [258] P. Kelly, A. Jostsons, R. Blake, and J. Napier, “The determination of foil thickness by scanning transmission electron microscopy,” *Physica status solidi (a)*, vol. 31, no. 2, pp. 771–780, 1975.

- [259] P. N. Nakashima, “Thickness difference: a new filtering tool for quantitative electron diffraction,” *Physical review letters*, vol. 99, no. 12, p. 125506, 2007.
- [260] H. Gong and F. Schapink, “Foil-thickness determination from zone-axis CBED patterns and TEM images for a GaAs/AlAs multilayer in plan view,” *Ultramicroscopy*, vol. 41, no. 4, pp. 375–385, 1992.
- [261] K. Botros, “A method for determining foil thicknesses in TEM by using convergent beam electron diffraction under weak beam conditions,” *Micron*, vol. 26, no. 6, pp. 539–543, 1995.
- [262] M. C. Riglos and A. Tolley, “A method for thin foil thickness determination by transmission electron microscopy,” *Applied Surface Science*, vol. 254, no. 1, pp. 420–424, 2007.
- [263] C. Rossouw, M. Gibson, and C. Forwood, “Dynamical electron diffraction analysis of lattice parameters, Debye—Waller factors and order in Ti-Al and Ti-Ga alloys,” *Ultramicroscopy*, vol. 66, no. 3-4, pp. 193–209, 1996.
- [264] W. Kossel and G. Möllenstedt, “Elektroneninterferenzen im konvergenten bündel,” *Annalen der Physik*, vol. 428, no. 2, pp. 113–140, 1939.
- [265] M. Klinger, M. Němec, L. Polívka, V. Gärtnerová, and A. Jäger, “Automated CBED processing: Sample thickness estimation based on analysis of zone-axis CBED pattern,” *Ultramicroscopy*, vol. 150, pp. 88–95, 2015.
- [266] D. Delille, R. Pantel, and E. Van Cappellen, “Crystal thickness and extinction distance determination using energy filtered CBED pattern intensity measurement and dynamical diffraction theory fitting,” *Ultramicroscopy*, vol. 87, no. 1-2, pp. 5–18, 2001.
- [267] S. Honglong, L. Minting, and W. Wenzhong, “CBED Tools for semi-automatic measurement of crystal thicknesses,” *Journal of Applied Crystallography*, vol. 50, no. 1, pp. 313–319, 2017.
- [268] V. Hou, “A DigitalMicrographTM Script for Crystal Thickness Measurements Using Convergent Beam Electron Diffraction,” *Microscopy and Microanalysis*, vol. 10, no. S02, pp. 1380–1381, 2004.
- [269] J. L. Lábár and J. Morgiel, “A new method for the measurement of thickness in single crystals,” *Micron*, vol. 29, no. 6, pp. 425–430, 1998.

- [270] C. H. Macgillavry, “Diffraction of convergent electron beams,” *Nature*, vol. 145, no. 3666, p. 189, 1940.
- [271] I. Ackermann, “Observations on the dynamical interference phenomena in convergent electron beams. II,” *Ann. Phys.(Leipzig)*, vol. 2, pp. 41–54, 1948.
- [272] D. V. S. Rao, “A simple technique for the estimation of the voltage of a transmission electron microscope,” *Journal of electron microscopy*, vol. 57, no. 4, pp. 119–122, 2008.
- [273] C. J. Rossouw and C. J. Maunders, “Voltage refinement by deficit HOLZ line geometry,” *Microscopy*, vol. 53, no. 4, pp. 381–392, 2004.
- [274] A. Morawiec, “A program for refinement of lattice parameters based on multiple convergent-beam electron diffraction patterns,” *Journal of Applied Crystallography*, vol. 40, no. 3, pp. 618–622, 2007.
- [275] J. Lewis, “Fast normalized cross-correlation,” *Vision Interface*, vol. 10, no. 1, pp. 120–123, 1995.
- [276] G. Fan and M. Ellisman, “Digital imaging in transmission electron microscopy,” *Journal of microscopy*, vol. 200, no. 1, pp. 1–13, 2000.
- [277] J. Spence and J.-M. Zuo, “Large dynamic range, parallel detection system for electron diffraction and imaging,” *Review of scientific instruments*, vol. 59, no. 9, pp. 2102–2105, 1988.
- [278] W. De Ruijter and J. Weiss, “Methods to measure properties of slow-scan CCD cameras for electron detection,” *Review of scientific instruments*, vol. 63, no. 10, pp. 4314–4321, 1992.
- [279] A. Weickenmeier, “Quantitative characterization of point spread function and detection quantum efficiency for a YAG scintillator slow scan CCD camera,” *Optik*, vol. 99, pp. 147–154, 1995.
- [280] I. Daberkow, K.-H. Herrmann, L. Liu, and W. Rau, “Performance of electron image converters with YAG single-crystal screen and CCD sensor,” *Ultramicroscopy*, vol. 38, no. 3-4, pp. 215–223, 1991.
- [281] R. R. Meyer and A. I. Kirkland, “Characterisation of the signal and noise transfer of CCD cameras for electron detection,” *Microscopy Research and Technique*, vol. 49, no. 3, pp. 269–280, 2000.

- [282] P. N. Nakashima and A. W. Johnson, “Measuring the PSF from aperture images of arbitrary shape—an algorithm,” *Ultramicroscopy*, vol. 94, no. 2, pp. 135–148, 2003.
- [283] K. H. Downing and F. M. Hendrickson, “Performance of a 2k CCD camera designed for electron crystallography at 400 kV,” *Ultramicroscopy*, vol. 75, no. 4, pp. 215–233, 1999.
- [284] J.-M. Zuo, “Electron detection characteristics of slow-scan CCD camera,” *Ultramicroscopy*, vol. 66, no. 1-2, pp. 21–33, 1996.
- [285] I. Daberkow, K.-H. Herrmann, L. Liu, W. Rau, and H. Tietz, “Development and performance of a fast fibre-plate coupled CCD camera at medium energy and image processing system for electron holography,” *Ultramicroscopy*, vol. 64, no. 1-4, pp. 35–48, 1996.
- [286] R. Staunton, “Measuring the high frequency performance of digital image acquisition systems,” *Electronics Letters*, vol. 33, no. 17, pp. 1448–1450, 1997.
- [287] R. Staunton, “Edge operator error estimation incorporating measurements of ccd tv camera transfer function,” *IEE Proceedings-Vision, Image and Signal Processing*, vol. 145, no. 3, pp. 229–235, 1998.
- [288] A. Thust, “High-resolution transmission electron microscopy on an absolute contrast scale,” *Physical review letters*, vol. 102, no. 22, p. 220801, 2009.
- [289] W. Van den Broek, S. Van Aert, and D. Van Dyck, “Fully automated measurement of the modulation transfer function of charge-coupled devices above the Nyquist frequency,” *Microscopy and Microanalysis*, vol. 18, no. 2, pp. 336–342, 2012.
- [290] J. M. Cowley, *Electron diffraction techniques*, vol. 2. Oxford University Press, 1992.
- [291] R. S. Pennington, F. Wang, and C. T. Koch, “Stacked-Bloch-wave electron diffraction simulations using GPU acceleration,” *Ultramicroscopy*, vol. 141, pp. 32–37, 2014.
- [292] W. Xu and J. M. LeBeau, “A deep convolutional neural network to analyze position averaged convergent beam electron diffraction patterns,” *Ultramicroscopy*, vol. 188, pp. 59–69, 2018.

- [293] B. H. Martineau, D. N. Johnstone, A. T. van Helvoort, P. A. Midgley, and A. S. Eggeman, “Unsupervised machine learning applied to scanning precession electron diffraction data,” *Advanced Structural and Chemical Imaging*, vol. 5, no. 1, p. 3, 2019.
- [294] A. Ziletti, D. Kumar, M. Scheffler, and L. M. Ghiringhelli, “Insightful classification of crystal structures using deep learning,” *Nature communications*, vol. 9, no. 1, p. 2775, 2018.
- [295] J. M. Ede and R. Beanland, “Improving electron micrograph signal-to-noise with an atrous convolutional encoder-decoder,” *Ultramicroscopy*, vol. 202, pp. 18–25, 2019.